ഹയർ സെക്കന്ററി കോഴ്സ്

കമ്പ്യൂട്ടർ ആപ്ലിക്കേഷൻസ്

(കോമേഴ്സ്)

ക്ലാസ് - XII

കേരളസർക്കാർ
പൊതുവിദ്യാഭ്യാസ വകുപ്പ്

സംസ്ഥാന വിദ്യാഭ്യാസ ഗവേഷണ പരിശീലന സമിതി (SCERT); കേരളം
2019

ദേശീയഗാനം

ജനഗണമന അധിനായക ജയഹേ
ഭാരത ഭാഗ്യവിധാതാ,
പഞ്ചാബസിന്ധു ഗുജറാത്ത മറാഠാ
ദ്രാവിഡ ഉത്ക്കല ബംഗാ,
വിന്ധ്യഹിമാചല യമുനാഗംഗാ,
ഉച്ഛല ജലധിതരംഗാ,
തവശുഭനാമേ ജാഗേ,
തവശുഭ ആശിഷ മാഗേ,
ഗാഹേ തവ ജയ ഗാഥാ
ജനഗണമംഗലദായക ജയഹേ
ഭാരത ഭാഗ്യവിധാതാ
ജയഹേ, ജയഹേ, ജയഹേ,
ജയ ജയ ജയ ജയഹേ!

പ്രതിജ്ഞ

ഇന്ത്യ എന്റെ രാജ്യമാണ്. എല്ലാ ഇന്ത്യക്കാരും എന്റെ സഹോദരീ സഹോദരന്മാരാണ്.
ഞാൻ എന്റെ രാജ്യത്തെ സ്നേഹിക്കുന്നു; സമ്പൂർണവും വൈവിധ്യപൂർണവുമായ അതിന്റെ പാരമ്പര്യത്തിൽ ഞാൻ അഭിമാനം കൊള്ളുന്നു.
ഞാൻ എന്റെ മാതാപിതാക്കളെയും ഗുരുക്കന്മാരെയും മുതിർന്നവരെയും ബഹുമാനിക്കും.
ഞാൻ എന്റെ രാജ്യത്തിന്റെയും എന്റെ നാട്ടുകാരുടെയും ക്ഷേമത്തിനും ഐശ്വര്യത്തിനും വേണ്ടി പ്രയത്നിക്കും.

Prepared by:

State Council of Educational Research and Training (SCERT)
Poojappura, Thiruvananthapuram 695012, Kerala
Website : www.scertkerala.gov.in *e-mail* : scertkerala@gmail.com
Phone : 0471 - 2341883, Fax : 0471 - 2341869
Typesetting and Layout : SCERT
© Department of Education, Government of Kerala

To be printed in quality paper - 80gsm map litho (snow-white)

ആമുഖം

ഏതു വിജ്ഞാനവും മാതൃഭാഷയിൽ പഠിക്കാനും പ്രകാശനം ചെയ്യാനും സാധിക്കും. അതിനുള്ള അവസരം പഠിതാക്കൾക്ക് ഒരുക്കേണ്ടത്, ഏതൊരു പഠന സമ്പ്രദായത്തിന്റെയും അനിവാര്യതയാണ്. അതിന്റെ തുടക്കമെന്ന നിലയ്ക്കാണ് ഹയർസെക്കന്ററി തലത്തിൽ ഭാഷേതര വിഷയങ്ങളിലെ പാഠപുസ്തകങ്ങൾ മലയാളത്തിൽ പ്രസിദ്ധീകരിക്കുന്നത്.

മാതൃഭാഷയിലൂടെയുള്ള വിദ്യാഭ്യാസം, ജ്ഞാനസമ്പാദനത്തിനുള്ള സുഗമമാർഗം എന്നതിനോടൊപ്പം സാംസ്കാരികതനിമയുടെ തിരിച്ചറിയൽ കൂടിയാണ്. അതുകൊണ്ടാണ് വികസിതരാജ്യങ്ങൾ മാതൃഭാഷയെ മുഖ്യബോധന മാധ്യമമായി സ്വീകരിച്ചിരിക്കുന്നത്. ഇന്ത്യയിലാകട്ടെ, ദേശീയതലത്തിലുള്ള പ്രധാന പരീക്ഷകളെല്ലാം പ്രാദേശിക ഭാഷകളിൽക്കൂടി നടത്തുന്നതിനുള്ള സംവിധാനവും ഉണ്ടായിവരികയാണ്. ഈയൊരു സാഹചര്യത്തിൽ നമ്മുടെ കുട്ടികളും മാതൃഭാഷയുടെ ശക്തിസൗന്ദര്യങ്ങൾ തിരിച്ചറിഞ്ഞ് വിവിധ വിഷയങ്ങളിൽ ജ്ഞാനനിർമിതിയിൽ ഏർപ്പെടേണ്ടതുണ്ട്. അതിന് അവരെ സജ്ജരാക്കുകയാണ് ഈ പാഠപുസ്തകങ്ങളുടെ മുഖ്യ ലക്ഷ്യം.

പരിഭാഷപ്പെടുത്തിയ പുസ്തകങ്ങളിൽ അതത് വിഷയങ്ങളിലെ സാങ്കേതിക പദങ്ങൾ പരമാവധി മലയാളത്തിലാക്കിയിട്ടുണ്ട്. നമ്മുടെ ഭാഷയിൽ ചിരപരിചിതമായ ഇംഗ്ലീഷ് പദങ്ങളെ അതേപടി സ്വീകരിച്ചിട്ടുമുണ്ട്. വിവർത്തനത്തിന് തീർത്തും വഴങ്ങാത്ത പദങ്ങളെ അതേരീതിയിൽ തന്നെ ഉപയോഗിച്ചിരിക്കുന്നു. മാതൃഭാഷയിൽ പഠിക്കുന്നവർക്ക് ആശയഗ്രഹണം സുഗമമാക്കുന്ന വിധത്തിലാണ് പാഠപുസ്തകരചന നടത്തിയിരിക്കുന്നത്. അതോടൊപ്പം മലയാളഭാഷയുടെ വളർച്ചയ്ക്കും ഈ പ്രവർത്തനം സഹായകമാകുമെന്ന് കരുതുന്നു.

പാഠപുസ്തകവിവർത്തന രംഗത്ത് നമ്മുടെ രാജ്യത്ത് നടന്ന വലിയൊരു കാൽവെപ്പാണ് ഇത്. പ്രഥമ സംരംഭമെന്നനിലയിൽ പല പരിമിതികളും പരിഭാഷയിൽ വന്നിട്ടുണ്ടാകാം. ക്ലാസ്മുറിയിൽ പ്രയോഗത്തിൽ വരുമ്പോഴാണ് അവയെല്ലാം കൂടുതൽ ബോധ്യപ്പെടുക. തുടർന്ന് വരുന്ന ഘട്ടങ്ങളിൽ അവയൊക്കെ പരിഹരിക്കുന്നതിന് എല്ലാ അഭ്യുദയകാംക്ഷികളിൽ നിന്നും വിശിഷ്യ അധ്യാപകർ, വിദ്യാർത്ഥികൾ എന്നിവരിൽ നിന്നും അഭിപ്രായങ്ങളും നിർദേശങ്ങളും പ്രതീക്ഷിക്കുന്നു.

ഡോ. ജെ. പ്രസാദ്

ഡയറക്ടർ,

എസ്.സി.ഇ.ആർ.ടി. കേരളം

പാഠപുസ്തക നിർമാണ സമിതി

ശ്രീ. ജോയ് ജോൺ
എച്ച്.എസ്.എസ്.ടി.,
സെന്റ് ജോസഫ് എച്ച്.എസ്.എസ്. തിരുവനന്തപുരം
ശ്രീ. എ.എൻ. സത്യൻ
എച്ച്.എസ്.എസ്.ടി., ജി.എം. എച്ച്.എസ്.എസ്.
കൊയിലാണ്ടി
ശ്രീ. മുസ്ഫ ഷംസുൽ ഹക്ക് കെ.കെ.
എച്ച്.എസ്.എസ്.ടി.,
ജി.എച്ച്.എസ്.എസ്. കൂത്തുപറമ്പ്, കണ്ണൂർ
ശ്രീ. എ.എസ്. ഇസ്മൈൽ
എച്ച്.എസ്.എസ്.ടി., പി.ജെ.എം.എസ്
ജി.എച്ച്.എസ്.എസ്., കണ്ടശ്ശാൻകടവ്, തൃശ്ശൂർ
ശ്രീ. വിനോദ് വി.
എച്ച്.എസ്.എസ്.ടി., എൻ.എസ്.എസ്
എച്ച്.എസ്.എസ്.,പ്രാക്കുളം, കൊല്ലം.
ശ്രീ. പ്രശാന്ത് പി.എം
എച്ച്.എസ്.എസ്.ടി.,
സെന്റ് ജോസഫ് ബോയ്സ് എച്ച്.എസ്.എസ്.,
കോഴിക്കോട്

ശ്രീ. സുനിൽ കാര്യത്താൻ
എച്ച്.എസ്.എസ്.ടി.,
ഗവ. ബ്രണ്ണൻ എച്ച്.എസ്.എസ്.
തലശ്ശേരി, കണ്ണൂർ
ശ്രീ. അസീസ് വി.
എച്ച്.എസ്.എസ്.ടി., ഗവ. എച്ച്.എസ്.എസ്.,
വെള്ളിയോട്, കോഴിക്കോട്
ശ്രീ. രാജ്മോഹൻ സി.
എച്ച്.എസ്.എസ്.ടി.,
നവമുകുന്ദ എച്ച്.എസ്.എസ്.,
തിരുനാവായ, മലപ്പുറം
ശ്രീ. നെജീവ് പി.പി.
എച്ച്.എസ്.എസ്.ടി.,
ഹിമായത്തുൽ ഇസ്ലാം എച്ച്.എസ്.എസ്.,
കോഴിക്കോട്

വിദഗ്ധർ

ഡോ. ലജീഷ് വി.എൽ
അസ്സി. പ്രൊഫ. ഡിപ്പാർട്ട്മെന്റ് ഓഫ്
കമ്പ്യൂട്ടർ സയൻസ് , കോഴിക്കോട് സർവകലാശാല
മധു വി.ടി.
ഡയറക്ടർ, കമ്പ്യൂട്ടർ സെന്റർ, സർവകലാശാല
കോഴിക്കോട്.
ഡോ. സുഷീൽ കുമാർ
അസ്സി. പ്രൊഫ. ഡിപ്പാർട്ട്മെന്റ് ഓഫ് ഇംഗ്ലീഷ്,
ഡി.ബി.കോളജ്, ശാസ്താംകോട്ട
ഡോ. വിനയകുമാരൻ നായർ എൻ.
അസ്സി. കമാൻഡന്റ്, ഹൈ-ടെക് സെൽ,
പോലീസ് ഹെഡ് ക്വാർട്ടേർസ്, തിരുവനന്തപുരം
ഡോ. കബീർ.വി
അസ്സി. പ്രൊഫ. & ഹെഡ്, ഡിപ്പാർട്ട്മെന്റ് ഓഫ്
കമ്പ്യൂട്ടർ സയൻസ്, ഫറൂക്ക് കോളേജ്, കോഴിക്കോട്

ഡോ. മധു എസ്സ്. നായർ
അസ്സി. പ്രൊഫ. ഡിപ്പാർട്ട്മെന്റ് ഓഫ്
കമ്പ്യൂട്ടർ സയൻസ് , കേരള സർവകലാശാല
ഡോ. ബിനു. പി. ചാക്കോ
അസ്സി. പ്രൊഫ. ഡിപ്പാർട്ട്മെന്റ് ഓഫ്
കമ്പ്യൂട്ടർ സയൻസ് , പ്രജ്യോതി നികേതൻ കോളജ്,
പുതുക്കാട്, തൃശ്ശൂർ
ഡോ. ദീപ എൽ.സി.
അസ്സി. പ്രൊഫ. ഡിപ്പാർട്ട്മെന്റ് ഓഫ് ഇംഗ്ലീഷ്,
ഗവ.വിമൻസ് കോളജ്, തിരുവനന്തപുരം
ശ്രീ. മഹേശ്വരൻ നായർ വി.
സബ് ഡിവിഷണൽ എഞ്ചിനീയർ,
റീജണൽ ടെലികോം സെന്റർ, കൈമനം,
തിരുവനന്തപുരം

ആർട്ടിസ്റ്റ്

സുധീർ വൈ **വിനീത് വി**

അക്കാദമിക് കോർഡിനേറ്റർ

ഡോ. മീന എസ്.
അസ്സി. പ്രൊഫ., എസ്.സി.ഇ.ആർ.ടി

പാഠപുസ്തക പരിഭാഷ സമിതി (മലയാളം)

ഡോ. ബിനു പി ചാക്കോ
അസോസിയേറ്റ് പ്രൊഫസർ, ഡിപ്പാർട്ട്മെന്റ് ഓഫ് കമ്പ്യൂട്ടർ സയൻസ്
പ്രജോതി നികേതൻ കോളേജ്, പുതുക്കാട്

ഡോ. ഗ്ലാഡ്സ്റ്റൺ എസ്. രാജ്
അസിസ്റ്റന്റ് പ്രൊഫസർ & ഹെഡ്, ഡിപ്പാർട്ട്മെന്റ് ഓഫ് കമ്പ്യൂട്ടർ സയൻസ്
ഗവ. കോളേജ്, നെടുമങ്ങാട്

ഡോ. എസ്. ഗിരീഷ്കുമാർ
അസിസ്റ്റന്റ് പ്രൊഫസർ, മലയാള വിഭാഗം
ശ്രീ.കേരളവർമ്മ കോളേജ്, തൃശൂർ

ശ്രീ. അരുൺ പി മാത്യു
എച്ച്.എസ്.എസ്.ടി, എം.ടി.എം.എച്ച്.എസ്.എസ്
പത്തനംതിട്ട

ശ്രീ. ഇ.എസ്. ഇസ്മൈൽ
എച്ച്.എസ്.എസ്.ടി.,
ജി.എച്ച്.എസ്.എസ്. മുല്ലശ്ശേരി, തൃശൂർ

ശ്രീ. മുകേഷ് വി.കെ.
എച്ച്.എസ്.എസ്.ടി.,
ജി.എച്ച്.എസ്.എസ്. കോക്കല്ലൂർ

ശ്രീ. മുഹമ്മദ് സർഫറാസ്
എച്ച്.എസ്.എസ്.ടി, സി.എച്ച്.എസ്.എസ്. നെല്ലിപുഴ

ശ്രീ. വിനയചന്ദ്രൻ സി
എച്ച്.എസ്.എസ്.ടി, വി.ജി.എച്ച്.എസ്.എസ്.
അംബികോദയം നെടിയവിള, കൊല്ലം

ശ്രീ. സുനിൽകുമാർ
എച്ച്.എസ്.എസ്.ടി, ഗവ. എച്ച്.എസ്.എസ്.
തേമ്പാംമൂട്, തിരുവനന്തപുരം

ശ്രീ. ഷിജു ആർ.
എച്ച്.എസ്.എസ്.ടി.,
ജി.എച്ച്.എസ്.എസ്. വടകര, പുത്തൂർ

ശ്രീ. ശ്രീജിത്ത് പി.
എച്ച്.എസ്.എസ്.ടി.,
സി.ജെ.എച്ച്.എസ്.എസ്. ചെമ്മനാട്, കാസർഗോഡ്

ശ്രീ. ഹരി കെ
എച്ച്.എസ്.എസ്.ടി, ഗവ. വി.& എച്ച്.എസ്.എസ്.
കടപ്പുറം, ചാവക്കാട്, തൃശൂർ

അക്കാദമിക് കോർഡിനേറ്റർ

ശ്രീമതി റിയാന അൻസാരി
റിസർച്ച് ഓഫീസർ, എസ്.സി.ഇ.ആർ.ടി

ഉള്ളടക്കം

1. C++ പ്രോഗ്രാമിങ്ങിനെക്കുറിച്ച് ഒരു പുനരവലോകനം 9
2. അറേകൾ 31
3. ഫങ്ഷനുകൾ 45
4. വെബ് സാങ്കേതികവിദ്യ 81
5. HTML ഉപയോഗിച്ചു കൊണ്ടുള്ള വെബ് രൂപകല്പന 131
6. ജാവ സ്ക്രിപ്റ്റ് ഉപയോഗിച്ചുള്ള ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് 173
7. വെബ് ഹോസ്റ്റിങ് 217
8. ഡാറ്റാബേസ് നിർവഹണ സംവിധാനം 229
9. സ്ട്രക്ചേർഡ് ക്വറി ലാംഗ്വേജ് 255
10. സംരംഭത്തിലെ വിഭവങ്ങളുടെ ആസൂത്രണം 301
11. വിവരവിനിമയ സാങ്കേതികവിദ്യയിലെ പ്രവണതകളും പ്രശ്നങ്ങളും 317

W9Z6I4

പാഠപുസ്തകത്തിൽ ഉപയോഗിച്ചിരിക്കുന്ന സൂചനകൾ

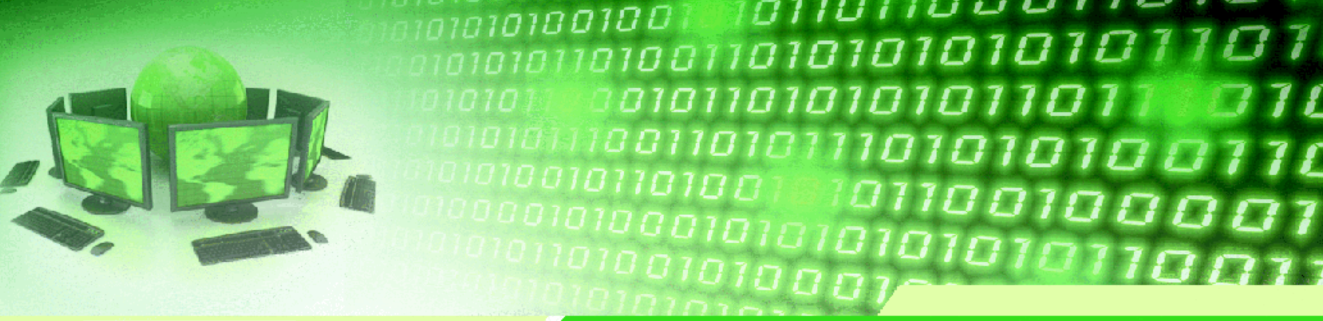
നമുക്ക് ചെയ്യാം

നിങ്ങളുടെ പുരോഗതി അറിയുക

ഇൻഫർമേഷൻ ബോക്സ്

നമുക്ക് പരിശീലിക്കാം

പഠനനേട്ടങ്ങൾ



B1X4C8

1 C++ പ്രോഗ്രാമിങ്ങിനെക്കുറിച്ച് ഒരു പുനരവലോകനം

പ്രധാന പഠനനേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പഠനം പൂർത്തിയാക്കുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ:

- കമ്പ്യൂട്ടറിലേക്ക് ഡാറ്റ കൈമാറുന്നതിനുള്ള ഇൻപുട്ട് പ്രസ്താവനകൾ പ്രോഗ്രാമിൽ ഉപയോഗിക്കുന്നു.
- വിവിധ തരത്തിലുള്ള ഔട്ട്പുട്ട് പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഔട്ട്പുട്ട് പ്രസ്താവനകൾ പ്രോഗ്രാമിൽ ഉപയോഗിക്കുന്നു.
- പ്രശ്ന പരിഹാര സമയത്ത് തീരുമാനങ്ങൾ എടുക്കുന്നതിനുള്ള വിവിധ രീതിയിലുള്ള if പ്രസ്താവനകൾ പ്രയോഗിക്കുന്നു.
- elseif ലാഡറും switch പ്രസ്താവനയും താരതമ്യം ചെയ്യുന്നു.
- C++-ലെ വിവിധ ലൂപ്പ് പ്രസ്താവനകളുടെ വ്യത്യാസം തിരിച്ചറിയുന്നു.
- പ്രോഗ്രാമുകളിൽ പ്രശ്ന പരിഹാരത്തിനനുയോജ്യമായ ലൂപ്പുകളെ തെരഞ്ഞെടുക്കുന്നു.
- പ്രശ്ന പരിഹാരത്തിൽ നെസ്റ്റഡ് ലൂപ്പ് എന്ന ആശയം ഉപയോഗിക്കുകയും ഔട്ട്പുട്ട് പ്രവചിക്കുകയും ചെയ്യുന്നു.
- ലൂപ്പുകളിൽ break, continue എന്നീ പ്രസ്താവനകളുടെ സ്വാധീനം തിരിച്ചറിയുകയും ഇവ പ്രോഗ്രാമിന്റെ ഗതിനിയന്ത്രണത്തിൽ എങ്ങനെ സ്വാധീനിക്കുന്നു എന്നു വിശദീകരിക്കുകയും ചെയ്യുന്നു.

C++ ഭാഷയുടെ അടിസ്ഥാന ആശയങ്ങളെക്കുറിച്ച് പതിനൊന്നാം ക്ലാസിൽ ചർച്ച ചെയ്തു കഴിഞ്ഞു. ഇനിയുള്ള പാഠഭാഗങ്ങളിലെ പ്രധാന പഠന നേട്ടങ്ങൾ കൈവരിക്കുന്നതിന് ഇത്തരം ആശയങ്ങളെക്കുറിച്ചുള്ള നല്ല അറിവ് അത്യാവശ്യമാണ്. പതിനൊന്നാം ക്ലാസിൽ നിങ്ങൾ ആർജിച്ച C++ ഭാഷയുടെ ശേഷികളേയും നൈപുണികളേയും കുറിച്ചുള്ള ഓർമപുതുക്കലാണ് ഈ അധ്യായം. ഓരോ ആശയവും അതിനാവശ്യമുള്ള വിശദീകരണത്തോടു കൂടി അവതരിപ്പിച്ചിരിക്കുന്നു. വളരെ പ്രധാനപ്പെട്ട സെലക്ഷൻ പ്രസ്താവനകൾ, ലൂപ്പ് പ്രസ്താവനകൾ എന്നിവ പ്രോഗ്രാമുകളുടെ സഹായത്തോടെയാണ് വിശദീകരിക്കപ്പെട്ടിട്ടുള്ളത്. ചില നൂതന സവിശേഷതകളായ നെസ്റ്റഡ് ലൂപ്പുകൾ, break, continue എന്നീ പ്രസ്താവനകളുടെ ലൂപ്പുകളിലെ പ്രഭാവവും ഈ അധ്യായത്തിൽ പരിചയപ്പെടുത്തുന്നുണ്ട്. C++ പ്രോഗ്രാം നിർമിക്കുന്നതിന് IDE ഉള്ള ഗ്നു കമ്പയിലർ കളക്ഷൻ (GNU Compiler Collection -GCC) ഉപയോഗിക്കുന്നതിനാൽ, പ്രോഗ്രാമിന്റെ ഘടന, ഹെഡർ ഫയലുകൾ ഉൾപ്പെടുത്തുന്ന രീതി, ഡാറ്റാ ഇനങ്ങളുടെ വലിപ്പം തുടങ്ങിയവയെക്കുറിച്ച് കൃത്യമായ ധാരണ നമുക്കുണ്ടാവണം.

1.1. C++ ന്റെ അടിസ്ഥാന ഘടകങ്ങൾ Basics of C++

ഒരു പ്രോഗ്രാമിങ് ഭാഷ എന്ന രീതിയിൽ C++ -ലെ കാരക്റ്റർ സെറ്റിൽ തുടങ്ങി ടോക്കണുകൾ, പ്രയോഗങ്ങൾ, പ്രസ്താവന എന്നിങ്ങനെയാണ് നാം പഠിച്ചുവന്നത്. വിവിധ ഡാറ്റാ ഇനങ്ങളും അവയുടെ മോഡിഫയറുകളും നാം ചർച്ച ചെയ്തു. പ്രയോഗങ്ങളുടെ (expressions) നിർമ്മാണ സമയത്ത് ഡാറ്റാ ഇനങ്ങളുടെ മാറ്റത്തിന്റെ അനിവാര്യതയും നാം തിരിച്ചറിഞ്ഞു. പട്ടിക 1.1-ൽ ഇത്തരം ഘടകങ്ങളെ കുറിച്ച് ചുരുക്കത്തിൽ പ്രതിപാദിച്ചിരിക്കുന്നു.

C++ ഒരു	
കാരക്ടർ സെറ്റ് (Character set)	C++ ഭാഷയുടെ അടിസ്ഥാന ഘടകം. അക്ഷരങ്ങൾ (a-z, A-Z) അക്കങ്ങൾ (0 - 9), പ്രത്യേക കാരക്ടറുകൾ (#, ; : >}+ തുടങ്ങിയ), വൈറ്റ് സ്പേസ് (സ്പേസ് ബാർ, ടാബ്, ന്യൂലൈൻ). കൂടാതെ പൂജ്യത്തിനും ഇരുന്നൂറ്റി അമ്പത്തഞ്ചിനും (0 - 255) ഇടയ്ക്ക് ASCII കോഡുള്ള (മൂല്യമുളള) മറ്റു പ്രത്യേക കാരക്ടറുകളുകൾ എന്നിങ്ങനെ ഇവയെ തരംതിരിച്ചിരിക്കുന്നു.
ടോക്കണുകൾ (tokens) -	C++ പ്രോഗ്രാമുകളുടെ അടിസ്ഥാന നിർമ്മാണ ഘടകമാണ്. ഒന്നോ അതിലധികമോ കാരക്ടറുകളാൽ നിർമ്മിതം. കീവേർഡ്, ഐഡന്റിഫയർ, ലിറ്ററൽ, പംക്ച്വുവേറ്റർ, ഓപ്പറേറ്റർ എന്നിവയായി തരംതിരിച്ചിരിക്കുന്നു.
കീവേർഡ് (Keywords)-	ഭാഷാ കമ്പയിലറുകൾക്ക് പ്രത്യേക രീതിയിലുള്ള അർഥം കൈമാറുന്ന കരുതിവയ്ക്കപ്പെട്ട വാക്കുകൾ.
ഐഡന്റിഫയറുകൾ (Identifiers)-	മെമ്മറി സ്ഥാനങ്ങൾ, പ്രസ്താവനകൾ, ഫംങ്ഷനുകൾ, ഡാറ്റ ഇനങ്ങൾ തുടങ്ങിയവയെ തിരിച്ചറിയുന്നതിനുള്ള ഉപയോക്തൃ നിർവ്വചിത വാക്കുകൾ. ഐഡന്റിഫയറുകളുടെ സാധുത ഉറപ്പുവരുത്തുന്നതിന് ചില പ്രത്യേക നിയമാവലികൾ അനുവർത്തിക്കേണ്ടതാണ്. വേരിയബിൾ, ലേബൽ, ഫങ്ഷൻ നാമം എന്നിവ ഇതിൽ ഉൾപ്പെടുന്നു.
ലിറ്ററൽ (Literals) -	പ്രോഗ്രാമിന്റെ പ്രവർത്തനസമയത്ത് വിലയ്ക്ക് യാതൊരു മാറ്റവും സംഭവിക്കാത്ത ടോക്കണുകൾ ആണ് ഇവ. സ്ഥിരാംഗങ്ങൾ എന്നും വിളിക്കപ്പെടുന്നു. പൂർണസംഖ്യാ സ്ഥിരാംഗങ്ങൾ, ദശാംശസംഖ്യാ സ്ഥിരാംഗങ്ങൾ, കാരക്ടർ സ്ഥിരാംഗങ്ങൾ, സ്ട്രിങ് സ്ഥിരാംഗങ്ങൾ എന്നിവയായി തരംതിരിച്ചിരിക്കുന്നു. ഡിജിറ്റുകളാണ് പൂർണസംഖ്യാ സ്ഥിരാംഗങ്ങൾ. വേണമെങ്കിൽ + (പ്ലസ്) - (മൈനസ്) എന്നീ ചിഹ്നങ്ങൾ ആദ്യകാരക്ടറുകളായി ചേർക്കാവുന്നതാണ്. ഫ്ളോട്ടിങ് പോയിന്റ് സ്ഥിരാംഗങ്ങളെ ദശാംശ രൂപത്തിലും എക്സ്പൊണന്റ് രൂപത്തിലും സൂചിപ്പിക്കാം. ഓരോ ജോഡി ഒറ്റ ഉദ്ധരണികൾക്കിടയിലുള്ള ഒരു കാരക്ടറിനെ കാരക്ടർ സ്ഥിരാംഗം എന്നു വിളിക്കുന്നു. എസ്കേപ്പ് സീക്വൻസുകൾ എന്ന് വിളിക്കുന്ന ചില പ്രത്യേക കാരക്ടർ സ്ഥിരാംഗങ്ങൾ ഉണ്ട്. പ്രിന്റ് ചെയ്യാൻ കഴിയാത്ത ചില കാരക്ടറുകളായ ന്യൂലൈൻ ('\n') ടാബ് സ്പേസ് ('\t'). പംക്ച്വുവേഷൻ അടയാളങ്ങളായ ഏക ഉദ്ധരണി (' \'), ഇരട്ട ഉദ്ധരണി ('\") ചോദ്യചിഹ്നം ('\? ')തുടങ്ങിയവ ഈ ഗണത്തിൽപെടുന്നു. ഒരു കൂട്ടം കാരക്ടറുകൾ ഇരട്ട ഉദ്ധരണികളുടെ ജോഡിക്കിടയിൽ ഉൾപ്പെടുത്തിയാൽ അവയെ സ്ട്രിങ് സ്ഥിരാംഗം എന്നു പറയുന്നു.
ഓപ്പറേറ്ററുകൾ (Operators)-	കമ്പയിലറിനോട് ചില പ്രവർത്തനങ്ങളെക്കുറിച്ച് സൂചിപ്പിക്കുന്ന ചിഹ്നങ്ങളാണിവ. ഇവ ഓരോന്നും ചില പ്രത്യേക പ്രവർത്തനങ്ങൾക്ക് പ്രേരിപ്പിക്കുന്നു. ഓപ്പറേറ്ററുകളുടെ എണ്ണത്തിനനുസരിച്ച് (ഓപ്പറേഷൻ നടത്തപ്പെടുന്ന ഡാറ്റ) ഓപ്പറേറ്ററുകളെ യൂനറി, ബൈനറി, ടെർനറി എന്നിങ്ങനെ തരംതിരിച്ചിരിക്കുന്നു. ഏതുതരം പ്രവർത്തനം നടത്തുന്നു എന്നതനുസരിച്ച് ഇതിനെ മറ്റൊരു രീതിയിൽ തരം തിരിക്കാം. അരിതമെറ്റിക് (+,-,*,/,%), റിലേഷണൽ (<,>,<=, >=, ==, !=), ലോജിക്കൽ (&&,.... ,!) എന്നീ ഓപ്പറേറ്ററുകളാണ് അവ. പ്രവർത്തനത്തിന്റെ ഫലമായി ഈ ഓപ്പ

C++ ഒരു	
	റേറ്ററുകൾ ചില വിലകൾ നൽകുന്നു. ഇൻപുട്ട് നൽകുന്നതിനായി ഗെറ്റ് ഫ്രം (>>), ഔട്ട്പുട്ട് ലഭിക്കുന്നതിനായി പുട്ട്ടു (<<), ഒരു വേരിയബിളിന് വില നൽകുന്നതിനായി (=) എന്നീ പ്രത്യേക ഓപ്പറേറ്ററുകളുമുണ്ട്. മറ്റൊരുതരം ഓപ്പറേറ്ററുകൾ ഒരു അരിത്മറ്റിക് പ്രവർത്തനത്തിനുശേഷം വില നൽകൽ പ്രവർത്തനം നടത്തുന്നു. ഇവയിൽ നൽകൽ ഓപ്പറേറ്ററുകളായ (+=), (-=), (*=), (/=), (%=) എന്നിവ അടങ്ങിയിരിക്കുന്നു.
പംക്ച്വുവേറ്ററുകൾ (Punchuators):	പ്രോഗ്രാമിന് ഉപയോഗിക്കുന്ന ഭാഷയിലെ പദഘടനാ രീതിയുടെ പൂർത്തീകരണത്തിന് പ്രത്യേക കാരക്ടറുകളായ കോമ (,), അർദ്ധവിരാമം (;) ഹാഷ് (#), ബ്രാക്കറ്റുകൾ ({ }) തുടങ്ങിയവ ഉപയോഗിക്കുന്നു. ഇവ അർത്ഥപരവും ഘടനാപരമായും കമ്പയിലറിന് പ്രത്യേക സൂചനകൾ നൽകുന്നതിന് ഉപയോഗിക്കുന്നു.
ഡാറ്റ ഇനങ്ങൾ (Data types)-	ഡാറ്റ ഇനവും ഈ ഡാറ്റ ഉപയോഗിക്കുന്ന ബന്ധപ്പെട്ട ഓപ്പറേഷനുകളും തിരിച്ചറിയുന്നതിനുള്ള ഒരുപാധിയാണിവ. ഡാറ്റ ഇനങ്ങളെ അടിസ്ഥാന ഡാറ്റാ ഇനങ്ങൾ, ഉപയോക്തൃ നിർവചിത ഡാറ്റ ഇനങ്ങൾ എന്നിങ്ങനെ തരംതിരിച്ചിരിക്കുന്നു. അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ അഭാജ്യ വിലകളെ പ്രതിനിധാനം ചെയ്യുന്നു. int, char, float, double, void എന്നിവ ഇതിൽ ഉൾപ്പെടുത്തുന്നു. ഇവയിൽ void ഡാറ്റ ഇനമൊഴികെ ബാക്കിയുള്ളവയ്ക്കെല്ലാം അതിന്റേതായ വലിപ്പവും പരിധിയും ഉണ്ട്. ശൂന്യഗണ സൂചകമായാണ് void ഡാറ്റ ഇനം ഉപയോഗിക്കുന്നത് എന്നതിനാൽ അതിന്റെ വലിപ്പം പൂജ്യമാണ്.
ടൈപ്പ് മോഡിഫയറുകൾ (Type modifiers) -	കീവേർഡുകളായ signed, unsigned, short, long എന്നിവയാണ് ടൈപ്പ് മോഡിഫയറുകൾ. മെമ്മറി വലിപ്പവും, അടിസ്ഥാന ഡാറ്റ ഇനം പിന്തുണയ്ക്കുന്ന ഡാറ്റയുടെ പരിധിയും പരിഷ്ക്കരിക്കുന്നതിനായി ഡാറ്റ ഇനത്തോടൊപ്പം ഇവ ഉപയോഗിക്കുന്നു.
പദപ്രയോഗങ്ങൾ (Expressions) -	ഓപ്പറേഷനുകൾ നടത്തുന്നതിനാവശ്യമായ ഓപ്പറേറ്ററുകളും ഓപ്പറന്റുകളും ഉപയോഗിച്ചാണ് പദപ്രയോഗങ്ങൾ നിർമ്മിക്കുന്നത്. ഓപ്പറേഷനുകളെ അടിസ്ഥാനമാക്കി ഇവയെ അരിത്മെറ്റിക് പദപ്രയോഗം, റിലേഷണൽ പദപ്രയോഗം, ലോജിക്കൽ പദപ്രയോഗം എന്നിങ്ങനെ തരംതിരിച്ചിരിക്കുന്നു. അരിത്മെറ്റിക് എക്സ്പ്രഷനെ പൂർണസംഖ്യപദപ്രയോഗം, ദശാംശസംഖ്യപദപ്രയോഗം എന്നിങ്ങനെ രണ്ടായി തരംതിരിച്ചിരിക്കുന്നു. ഒരു പൂർണസംഖ്യ പ്രയോഗത്തിന്റെ ഓപ്പറന്റുകളായി പൂർണസംഖ്യ ഡാറ്റ മാത്രമേ ഉണ്ടാവൂ. കൂടാതെ ഇവയുടെ ഫലം എപ്പോഴും ഒരു പൂർണ സംഖ്യ വിലയുമായിരിക്കും ദശാംശസംഖ്യപദപ്രയോഗങ്ങളിൽ ഓപ്പറന്റുകളും തിരിച്ച് നൽകുന്ന വിലയും ദശാംശസംഖ്യ ആയിരിക്കും. റിലേഷണൽ പദപ്രയോഗങ്ങളിൽ സംഖ്യകളോ അക്ഷരങ്ങളോ ആയ ഡാറ്റ ഓപ്പറന്റുകളും ശരി അല്ലെങ്കിൽ തെറ്റ് തിരികെ ലഭിക്കുന്ന ഫലവുമായിരിക്കും. ലോജിക്കൽ പ്രയോഗങ്ങളിൽ ഓപ്പറന്റുകളായി റിലേഷണൽ പദപ്രയോഗങ്ങളാണ് ഉപയോഗിക്കാറുള്ളത്. ഇത്തരം പദപ്രയോഗങ്ങൾ ശരി അല്ലെങ്കിൽ തെറ്റ് എന്നീ വിലകളാണ് ഉത്തരങ്ങളായി തിരിച്ചുനൽകാറുള്ളത്.

C++ ഒരു	
ഇനം മാറ്റൽ (Tyep conversion) -	ഒരു അരിത്മെറ്റിക് പദപ്രയോഗത്തിൽ വിവിധ തരം ഓപ്പറന്റുകൾ ഉൾപ്പെട്ടിട്ടുണ്ടെങ്കിൽ ഇനം മാറ്റൽ പ്രക്രിയ നടത്തപ്പെടുന്നു. വിലയുടെ നിലവിലുള്ള ഡാറ്റ ഇനത്തെ മറ്റൊന്നിലേക്ക് മാറ്റുന്ന പ്രക്രിയാണ് ഇനം മാറ്റൽ. ആന്തരികമായും, ബാഹ്യമായും ഇനം മാറ്റൽ നടത്താവുന്നതാണ്. ആന്തരിക തരംഗമാറ്റത്തിൽ കമ്പയിലറിനാണ് ഇനം മാറലിന്റെ പൂർണ ഉത്തരവാദിത്തം. ഇത്തരം ഇനം മാറ്റത്തിൽ എല്ലായ്പ്പോഴും മെമ്മറി വലുപ്പം കുറഞ്ഞ ഇനത്തെ ഉയർന്ന ഇനത്തിലേക്കാണ് മാറ്റപ്പെടുന്നത്. അതുകൊണ്ട്തന്നെ ഇത് ഡാറ്റ ഇനം ഉയർത്തൽ എന്നും അറിയപ്പെടുന്നു. ബാഹ്യമായ ഇനം മാറ്റത്തിന്റെ പൂർണ്ണ ഉത്തരവാദിത്തം ഉപയോക്താവിൽ അർപ്പിതമാണ്. ഏത് ഡാറ്റ ഇനത്തിലേക്കാണോ മാറ്റേണ്ടതെന്ന് തീരുമാനിക്കുന്നത് ഉപയോക്താവാണ്. അതിനാൽ ഇത് ടൈപ്പ് കാസ്റ്റിങ്ങ് എന്നും അറിയപ്പെടുന്നു.

പട്ടിക 1.1: Basic elements of C language

1.1.1 C++ പ്രോഗ്രാമിലെ വിവിധതരത്തിലുള്ള പ്രസ്താവനകൾ (Various statements in a C++ Program)

സാധാരണയായി പ്രീ പ്രോസസ്സർ നിർദ്ദേശങ്ങളോടു കൂടിയാണ് ഒരു C++ പ്രോഗ്രാം ആരംഭിക്കുന്നത്. ഒരു പ്രോഗ്രാമിൽ ഉപയോഗിച്ചിരിക്കുന്ന മുൻനിർവചിത ഐഡന്റിഫയറുകളും ഫങ്ഷനുകളേയും കുറിച്ച് ഉള്ള വിവരങ്ങൾ അടങ്ങിയ ഹെഡർ ഫയലുകളെ # include എന്നു തുടങ്ങുന്ന പ്രീ പ്രോസസ്സർ നിർദ്ദേശം ഉപയോഗിച്ച് ബന്ധിപ്പിക്കുന്നു. using namespace പ്രസ്താവന, പ്രീ പ്രോസസ്സർ നിർദ്ദേശത്തിനുശേഷം ഉപയോഗിക്കുന്നു. സാധാരണയായി std എന്ന മുൻനിർവചിത namespace ഉപയോഗിച്ച് cin, cout തുടങ്ങിയ ഐഡന്റിഫയറുകളുടെ ഉപയോഗപരിധി നിർവചിക്കുന്നു. അതിനുശേഷം main() ഫങ്ഷൻ ആരംഭിക്കുന്നു. ഒരു C++ ന് അത്യന്താപേക്ഷിതമായ ഫങ്ഷനാണ് ഇത്. പ്രോഗ്രാമിന്റെ പ്രവർത്തനം ആരംഭിക്കുന്നതും അവസാനിക്കുന്നതും ഇതിലാണ്. പ്രഖ്യാപന പ്രസ്താവനകളും, കൃത്യം പരിഹരിക്കുവാൻ ആവശ്യമായ ഒരു കൂട്ടം പ്രവർത്തന പ്രസ്താവനകളും ഇതിൽ അടങ്ങിയിരിക്കുന്നു. ഈ പ്രസ്താവനകളെ കുറിച്ച് നമുക്ക് കൂടുതൽ മനസിലാക്കാം.

പ്രഖ്യാപന പ്രസ്താവനകൾ (Declaration statement)

വേരിയബിൾ എന്നത് മെമ്മറിയിലെ ഒരിടത്തിനെ സൂചിപ്പിക്കുവാനും, പ്രോഗ്രാമുകൾക്ക് ഡാറ്റയെ പ്രതിനിധികരിക്കുവാനും ഉള്ളതാണ്. ഒരു പ്രോഗ്രാമിൽ അവ ഉപയോഗിക്കുന്നതിന് മുമ്പ് തന്നെ നിർവചിക്കുകയും അവയുടെ ഡാറ്റ ഇനം പ്രഖ്യാപിക്കേണ്ടതുമാണ്. താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവനകൾ വേരിയബിൾ പ്രഖ്യാപനങ്ങൾക്ക് ഉദാഹരണങ്ങളാണ്.

```
int n, sum;
float rad, area;
signed int a,b,c;
```

വേരിയബിൾ പ്രഖ്യാപിക്കുന്നതിനോടൊപ്പം തന്നെ അവയ്ക്കുള്ള വില നൽകുന്നതാണ് ചുവടെ കാണിച്ചിരിക്കുന്നത്.

```
int n=10;
```

ഇങ്ങനെയുള്ള പ്രസ്താവനകളെ വേരിയബിളുകൾക്ക് പ്രാരംഭവില നൽകൽ പ്രസ്താവന എന്ന് അറിയപ്പെടുന്നു. വേരിയബിളിൽ നൽകിയിരിക്കുന്ന വില പ്രോഗ്രാമിൽ മറ്റ് വില കൊണ്ട് മാറ്റം

വരാം. എന്നാൽ ചുവടെ ചേർത്തിരിക്കുന്ന പ്രസ്താവന വേരിയബിൾ നൽകിയിരിക്കുന്ന വില മാറ്റം വരുത്തുവാൻ അനുവദിക്കുന്നില്ല.

```
const int n=10;
```

ഇവിടെ പ്രാരംഭവില നൽകൽ ആരംഭിച്ചിരിക്കുന്നത് const ആക്സസ് മോഡിഫയർ കൊണ്ടാണ്, ഈ കീവേർഡ് വേരിയബിളിലേ വില മാറ്റം വരുത്തുന്നത് നിയന്ത്രിക്കുന്നു.

ഇൻപുട്ട് പ്രസ്താവന (Input statement)

എക്സ്ട്രാക്ഷൻ ഓപ്പറേറ്റർ, ഗെറ്റ് ഫ്രം എന്നീ പേരുകളിലറിയപ്പെടുന്ന >> എന്ന ഓപ്പറേറ്റർ C++ ലഭ്യമാക്കിയിരിക്കുന്നു. ഇത് ഒരു ബൈനറി ഓപ്പറേറ്റർ ആയതിനാൽ ഇതിന് രണ്ട് ഓപെറന്റുകളുടെ ആവശ്യകത ഉണ്ട്. ആദ്യത്തെ ഓപെറന്റ് ആയ മുൻ നിർവചിത ഐഡന്റിഫയർ cin കീബോർഡിനെ ഇൻപുട്ട് ഒബ്ജക്റ്റ് ആയി തിരിച്ചറിയുന്നു. രണ്ടാമത്തേ ഓപെറന്റ് നിർബന്ധമായും ഒരു വേരിയബിൾ ആയിരിക്കണം. നമുക്ക് ഒന്നിൽ കൂടുതൽ ഇൻപുട്ട് സ്വീകരിക്കുന്നതിനായി അതേ പ്രസ്താവനയിൽ ഒന്നില് കൂടുതൽ നവേരിയബിൾ നമുക്ക് ഉപയോഗിക്കാം. ഒരു സാധുവായ ഉദാഹരണം ഒരു ചുവടെ നൽകിയിരിക്കുന്നു.

```
cin>>rad;
cin>>a>>b>>c;
```

ഔട്ട്പുട്ട് പ്രസ്താവന (Output statement)

ഔട്ട്പുട്ട് പ്രവർത്തനം നടത്തുന്നതിനായി ഇൻസേർഷൻ ഓപ്പറേറ്റർ അല്ലെങ്കിൽ പുട്ട് ടു ഓപ്പറേറ്റർ എന്ന് അറിയപ്പെടുന്ന ഓപ്പറേറ്ററുകൾ C++ ലഭ്യമാക്കിയിരിക്കുന്നു. ഇത് ഒരു ബൈനറി ഓപ്പറേറ്റർ ആണ്. ഇതിൽ ആദ്യത്തേ ഓപെറന്റ് ആയ മുൻനിർവചിത ഐഡന്റിഫയർ cout മോണിറ്ററിനെ ഔട്ട്പുട്ട് ഒബ്ജക്റ്റ് ആയി തിരിച്ചറിയുന്നു. രണ്ടാമത്തെ ഓപെറന്റ് ഒരു സ്ഥിരാംഗമോ, ഒരു വേരിയബിളോ അല്ലെങ്കിൽ ഒരു പദപ്രയോഗമോ ആകാം. ഔട്ട്പുട്ട് പ്രസ്താവനകൾക്ക് ചില ഉദാഹരണങ്ങൾ ചുവടെ നൽകിയിരിക്കുന്നു.

```
cout<< "hello";
cout<< area;
cout<< 25;
cout<< a+b+c;
cout<< "Sum of " << n << "numbers = " << sum;
```

വില നൽകൽ പ്രസ്താവന (Assignment statement)

മെമ്മറിയുടെ ഒരിടത്തിൽ (വേരിയബിൾ) നിർദ്ദിഷ്ട ഡാറ്റ സൂക്ഷിക്കുവാൻ വില നൽകൽ പ്രസ്താവന ഉപയോഗിക്കുന്നു. '=' ഉൾപ്പെട്ടിട്ടുള്ള പ്രസ്താവനകളെ വില നൽകൽ പ്രസ്താവന എന്ന് അറിയപ്പെടുന്നു. ഇത് ഒരു ബൈനറി ഓപ്പറേറ്റർ ആണ്. = ന്റെ ഇടത്വശത്തെ ഓപെറന്റ് ഒരു വേരിയബിൾ ആയിരിക്കണം. = ന് ശേഷമുള്ള ഓപ്റെന്റ് തരത്തിലുള്ള ഒരു സ്ഥിരാംഗമോ, ഒരു വേരിയബിളോ, ഒരു അക്കങ്ങളുടെ പദപ്രയോഗമോ ആകാം. സാധുവായ വില നൽകൽ പ്രസ്താവനകളുടെ ഉദാഹരണം ചുവടെ നൽകിയിരിക്കുന്നു.

```
n = 253;
area = 3.14*rad*rad;
a = b = c;
```

പ്രത്യേക വില നൽകൽ ഓപ്പറേറ്ററുകളെ അരിത്തമെറ്റിക് വില നൽകൽ ഓപ്പറേറ്റർ എന്ന് അറിയപ്പെടുന്നു. +=, -=, *=, /- ഉം % = ആണിവ. ഇവയെല്ലാം ബൈനറി ഓപ്പറേറ്റേഴ്സ് ആണ്. ഇതിന്റെ വലതുഭാഗത്ത് ഉള്ള ഓപ്പറന്റ് ഒരു വേരിയബിൾ ആയിരിക്കണം. ഇവയുടെ പ്രവർത്തനം ചുവടെ വിശദീകരിച്ചിരിക്കുന്നു.

```
n+=2;          // n=n+2;  എന്നതിനു തുല്യം
a*=b;          // a=a*b;  എന്നതിനു തുല്യം
sum-=n%10;     // sum=sum-n%10;  എന്നതിനു തുല്യം
```

ഓപ്പറേറ്ററുകളായ ++ ഉം -- ഉം C++ ലെ പ്രത്യേക തരം ഓപ്പറേറ്ററുകൾ ആണ്. ഒരു തരത്തിൽ വില നൽകൽ പ്രസ്താവനകൾക്ക് സമാനമാണ്. ഇവ യൂനറി ഓപ്പറേറ്ററുകളാണ്. അവയുടെ ഓപെറന്റ് വേരിയബിൾ ആയിരിക്കണം. ചുവടെ നൽകിയിരിക്കുന്ന പ്രസ്താവനകൾ ഇവയുമായി ബന്ധപ്പെട്ട പ്രവർത്തനങ്ങൾ വിശദമാക്കിയിരിക്കുന്നു.

```
n++;           // n=n+1;  എന്നതിനു തുല്യം
a--;           // a=a-1;  എന്നതിനു തുല്യം
```

ഈ ഓപെറേറ്ററുകൾക്ക് രണ്ട് തരം പതിപ്പുകൾ ആണ് പോസ്റ്റ്ഫിക്സ് രൂപമെന്നും, പ്രിഫിക്സ് രൂപമെന്നും a++; ഉം a--; ഇവ യഥാക്രമം ഇൻക്രിമെന്റിന്റെയും ഡിക്രിമെന്റിന്റെയും പോസ്റ്റ് ഫിക്സ് രൂപം ++a; ഉം --a ഉം പ്രിഫിക്സ് രൂപവും ഏതു രൂപത്തിലായാലും ++ ഓപ്പറേറ്റർ ഓപെറന്റ് വേരിയബിളിന്റെ കൂടെ ഒന്ന് കൂട്ടുകയും അതിന്റെ ഫലം അവിടെത്തന്നെ സൂക്ഷിക്കുകയും ചെയ്യുന്നു.

ഈ രണ്ട് പതിപ്പുകളും വില നൽകൽ പ്രസ്താവനകളുടെ കൂടെയോ ഔട്ട്പുട്ട് പ്രസ്താവനകളുടെ കൂടെ ഉപയോഗിക്കുമ്പോൾ വ്യത്യസ്ത ഫലമാകും കിട്ടുക. 'a' എന്നത് 5 എന്ന വില സൂക്ഷിച്ചിട്ടുള്ള ഒരു ഇന്റിജർ വേരിയബിളാണ് എന്നും, b എന്നത് മറ്റൊരു വേരിയബിൾ ആണ് എന്നും സങ്കൽപ്പിക്കുക. b = a++ എന്ന പ്രസ്താവനയുടെ പ്രവർത്തനത്തിനു ശേഷം b യിലെ വില 5 ഉം a യിലെ വില 6 ഉം ആകുന്നു. അതായത്, b=a++; എന്നത് b=a; a=a+1 ഉം എന്ന തുടർച്ചയായ പ്രസ്താവനകൾക്ക് സമാനമാണ്. ആയതിനാൽ ഇത്തരം വർദ്ധന നൽകൽ രീതിയെ ഉപയോഗിക്കുക. വ്യത്യാസം വരുത്തുക രീതി എന്നറിയപ്പെടുന്നു.

എന്നാൽ b=++a; എന്ന പ്രസ്താവന a=a+1; b=a; എന്ന തുടർച്ചയായ പ്രസ്താവനകൾക്ക് സമാനമാണ്. ആയതിൽ a യുടെയും b യുടെയും വില 6 ആയി മാറുന്നു. ഈ രീതിയെ വ്യത്യാസം വരുത്തുക ഉപയോഗിക്കുക എന്നറിയപ്പെടുന്നു.

അതുപോലെ cout<<1--; എന്ന പ്രസ്താവന 5 എന്ന് പ്രദർശിപ്പിക്കുകയും, എന്നാൽ a യുടെ വില 4 ആയി മാറുകയും ചെയ്യുന്നു. ഇതിന് തുല്യമായ പ്രസ്താവന എന്നത് cout<<a; a=a-1 എന്നാണ്. എന്നാൽ cout<<--a; എന്നതിന് സമാനമായ പ്രസ്താവനകൾ a=a-1; cout<<a എന്നാണ്.

ഇൻപുട്ട്, ഔട്ട്പുട്ട്, വില നൽകൽ ഓപറേറ്ററുകൾ (>>,<< ഉം =) തുടങ്ങിയവ ഇത്തരം പ്രസ്താവനകളിൽ ഒന്നിലധികം തവണ വരുന്നതായി കാണാം. ഇതിനെ കാസ്കാഡിംഗ് (cascading) എന്ന് അറിയപ്പെടുന്നു. ഓരോ വിഭാഗത്തിലെയും കാസ്കാഡിംഗ് ഇൻപുട്ട്, ഔട്ട്പുട്ട് വില നൽകൽ ഓപ്പറേറ്റർ തുടങ്ങിയവയുടെ ഉദാഹരണം ചുവടെ നൽകിയിരിക്കുന്നു.

```
cin >> a >> b >> c;
cout << "Sum of " << n << "numbers = " << sum;
a = b = c;
```

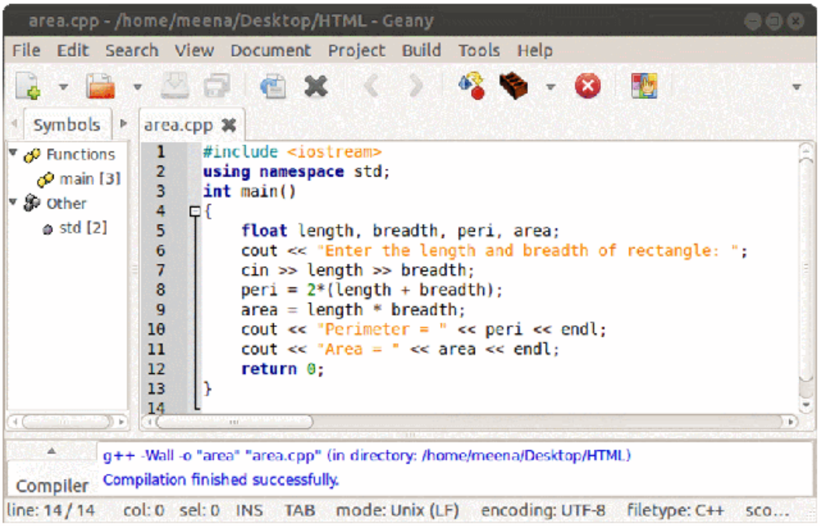
1.1.2 ഒരു C++ പ്രോഗ്രാമിലെ പ്രോഗ്രാമിന്റെ ഘടന (Structure of a C++ program)

ഒരു C++ പ്രോഗ്രാമിന്റെ അടിസ്ഥാന ഘടന പ്രോഗ്രാം 1.1 കാണിച്ചിരിക്കുന്നു. ഒരു ചതുരത്തിന്റെ നീളവും വീതിയും സ്വീകരിച്ച് അതിന്റെ വിസ്തീർണ്ണവും ചുറ്റളവും കണ്ടെത്തുന്നു.

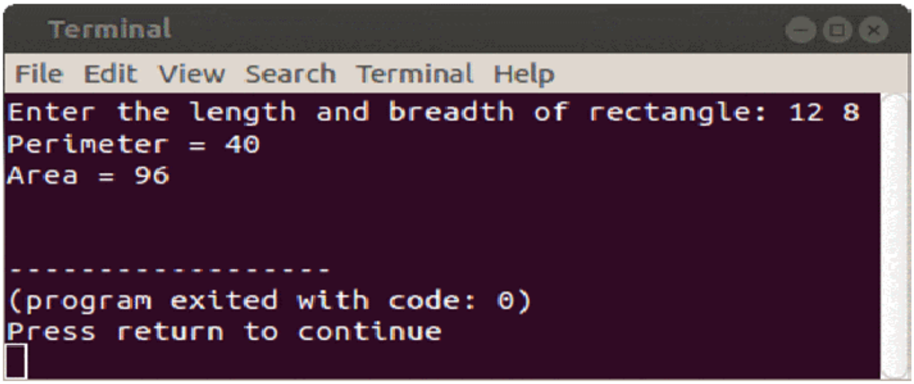
പ്രോഗ്രാം:– 1.1 ഒരു ചതുരത്തിന്റെ വിസ്തീർണവും ചുറ്റളവും കണ്ടുപിടിക്കാൻ

```
#include <iostream>
using namespace std;
int main()
{
   float length, breadth, peri, area;
   cout << "Enter the length and breadth of rectangle: ";
   cin >> length >> breadth;
   peri = 2*(length + breadth);
   area = length * breadth;
   cout << "Perimeter = " << peri << endl;
   cout << "Area = " << area << endl;
   return 0;
}
```

cin; ഉം cout ഉം എന്ന ഐഡന്റിഫയറുകൾ ഉപയോഗിക്കുന്നതിനാൽ പ്രോഗ്രാം 1.1 iostream എന്ന ഹെഡർ ഫയൽ ഉപയോഗിക്കുന്നു. cin ഉം cout ഉം പ്രത്യേകമായി ഉപയോഗിക്കുന്നതിന് രണ്ടാമത്തെ വരി അത്യന്താപേക്ഷിതമാണ്. using namespace std എന്ന പ്രസ്താവന cin ഉം cout ഉം main പ്രോഗ്രാമിൽ ലഭ്യമാക്കുന്നു. GCC യിൽ main () എന്ന ഫങ്ഷൻ നാമത്തിന് മുന്നിലായി int എന്ന ഡാറ്റ തരം ഉപയോഗിക്കുന്നു. float ഡാറ്റ തരം ഉപയോഗിച്ച് വേരിയബിളുകൾ പ്രഖ്യാപിച്ചിരിക്കുന്നു. ഇൻപുട്ട്, ഔട്ട്പുട്ട് ഓപ്പറേറ്ററുകളുടെ കാസ്കാഡിംഗ് ഉപയോഗിച്ചിരിക്കുന്നു. വില നൽകൽ പ്രസ്താവനയിൽ സൂത്രവാക്യങ്ങൾ ഉപയോഗിച്ച് പ്രശ്നം നിർദ്ധാരണം ചെയ്തിരിക്കുന്നു. ഓരോ ഫലത്തിനു ശേഷം പുതിയ വരിയിൽ ഫലം ദൃശ്യമാകുന്നതിനായി '/n' ന് പകരം endl ഉപയോഗിച്ചിരിക്കുന്നു. ചിത്രം 1.1 (a) പ്രോഗ്രാം 1.1ന്റെ Geany IDE യിൽ ഉള്ള സ്ക്രീനിന്റെ ചിത്രവും ചിത്രം 1.1 (b) പ്രോഗ്രാമിന്റെ പ്രവർത്തനഫലം കാണിച്ചിരിക്കുന്നു.



ചിത്രം 1.1(a) : ജീനിIDE യിൽ പ്രോഗ്രാം 1.1



ചിത്രം 1.1(b) : ജീനിIDE യുടെ ടെർമിനൽ വിൻഡോയിൽ പ്രോഗ്രാം 1.1 ന്റെ ഔട്ട്പുട്ട്

ഒരു C++ പ്രോഗ്രാം എഴുതുമ്പോൾ, നമ്മൾ 'using namespace std' എന്ന പ്രസ്താവന ഉപയോഗിക്കുന്നു. എന്തുകൊണ്ട്?

ഒരു പ്രോഗ്രാമിൽ ഒരേ വ്യാപ്തിയിൽ ഒരേ പേരിലുള്ള ഒന്നിലധികം ഐഡന്റിഫയറുകൾ (വേരിയബിളുകൾ അല്ലെങ്കിൽ ഫങ്ഷനുകൾ) ഉണ്ടായിരിക്കാൻ പാടില്ല. നമ്മുടെ വീട്ടിൽ രണ്ടോ അതിലധികമോ ആളുകൾക്ക് (അല്ലെങ്കിൽ ജീവജാലങ്ങൾക്ക്) ഒരേ പേരുണ്ടാവില്ല. അങ്ങനെയുണ്ടെങ്കിൽ തീർച്ചയായും വീട്ടിനുള്ളിൽ അവരെ പേരു കൊണ്ട് തിരിച്ചറിയുക എന്നത് വിഷമകരമാകും. അതുകൊണ്ട് നമ്മുടെ വീട്ടിന്റെ പരിധിയിൽ ഒരു പേരും അനന്യമായിരിക്കണം. എന്നാൽ നമ്മുടെ അയൽപക്കത്തെ വീട്ടിൽ സമാനമായ പേരുള്ള ഒരാൾ അല്ലെങ്കിൽ ജീവജാലം ഉണ്ടായിരിക്കാം. അതാത് പരിധിക്കുള്ളിൽ വ്യക്തികളെ പേരു കൊണ്ട് തിരിച്ചറിയുന്നതിന് ഇത് യാതൊരു ആശയകുഴപ്പവുമുണ്ടാക്കില്ല. പക്ഷെ പുറമെ നിന്നൊരു വ്യക്തിയ്ക്ക് പേരു മാത്രം ഉപയോഗിച്ച് കൊണ്ട് ഇവരെ തിരിച്ചറിയാൻ കഴിയില്ല. അതിനാൽ വീട്ടുപേരും കൂടി പരാമർശിക്കേണ്ടതുണ്ട്.

നെയിം സ്പേസ് എന്ന ആശയം വീട്ടുപേരിനു സമാനമാണ്. ഒരു പ്രത്യേക നെയിം സ്പേസുമായി വ്യത്യസ്ത ഐഡന്റിഫയറുകൾ ബന്ധപ്പെട്ടിരിക്കുന്നു. ഓരോ ഇനവും വ്യത്യസ്തമായിരിക്കുന്ന ഒരു ഗണത്തിന്റെ പേരാണിത്. വേരിയബിളുകൾക്കും ഫങ്ഷനുകൾക്കുമായി പ്രത്യേകം നെയിം സ്പേസുകൾ സൃഷ്ടിക്കുന്നതിന് ഉപയോക്താവിനു അനുവാദമുണ്ട്. ഒരു നെയിം സ്പേസിനു പേരു കൊടുക്കാൻ നമുക്ക് ഒരു ഐഡിന്റിഫയർ ഉപയോഗിക്കാം. പ്രോഗ്രാമിങ്ങിൽ ഉപയോഗിക്കുന്ന ഘടകങ്ങളെ ഏത് നെയിം സ്പേസിൽ തിരയണമെന്ന് using എന്ന കീവേർഡ് സാങ്കേതികമായി കംപൈലറിനോട് പറയുന്നു. C++ ൽ stadard എന്നതിന്റെ ചുരുക്കെഴുത്താണ് std.cin, cout തുടങ്ങിയ മറ്റ് പല ഒബ്ജക്ടുകളും നിർവചിച്ചിട്ടുള്ള ഒരു നെയിം സ്പേസ് ആണിത്. അതിനാൽ ഒരു പ്രോഗ്രാമിൽ ഇവ ഉപയോഗിക്കണമെങ്കിൽ std::cin, std::cout എന്ന മാതൃക നാം പിന്തുടരേണ്ടതാണ്. using namespace std എന്ന പ്രസ്താവന പ്രോഗ്രാമിൽ ഉപയോഗിക്കുന്നതിലൂടെ ഇത്തരത്തിലുള്ള വിശദമായ പരാമർശങ്ങൾ ഒഴിവാക്കാവുന്നതാണ്. അത്തരമൊരു സാഹചര്യത്തിൽ കംപൈലർ cin, cout, endl മുതലായവയ്ക്കായി ഈ നെയിം സ്പേസിൽ തിരയുന്നു. cin, cout, endl അല്ലെങ്കിൽ അതുപോലെയുള്ളവ എപ്പോഴൊക്കെ ഒരു C++ പ്രോഗ്രാമിൽ കമ്പ്യൂട്ടർ കാണുന്നുവോ, അവയെ std::cin, std::cout, std::endl എന്നിങ്ങനെ വ്യാഖ്യാനിക്കുന്നു.

using namespace std എന്ന പ്രസ്താവന യഥാർഥത്തിൽ ഒരു ഫങ്ഷനെ കൂട്ടിച്ചേർക്കുന്നില്ല, cin, cout, endl തുടങ്ങി സമാനമായവയേ കൂട്ടിച്ചേർക്കുന്നത് include<iostream> എന്ന പ്രസ്താവനയാണ്.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. C++ പ്രോഗ്രാമിൽ ടോക്കൺ എന്നാലെന്ത്?
2. താഴെ നൽകിയിരിക്കുന്ന ടോക്കണുകൾ ശ്രദ്ധിക്കുക. ഓരോന്നും ഏത് വിഭാഗത്തിൽ ഉൾപ്പെടുന്നു എന്ന് കണ്ടെത്തുക.

   i. number ii. 23.98 iii. "\0" iv. cin v. ++
   vi. void vii. '\\' viii. ; ix. = x. a
3. C++ പ്രോഗ്രാമിൽ # include പ്രസ്താവനയുടെ പങ്കെന്ത്?
4. cin >> 25; എന്ന പ്രസ്താവനയിലെ തെറ്റ് കണ്ടെത്തുക.
5. C++ ന്റെ ടൈപ്പ് മോഡിഫയറുകൾ പട്ടികപ്പെടുത്തുക?

1.2 നിയന്ത്രണ പ്രസ്താവനകൾ (Control statements)

പ്രശ്നങ്ങൾ നിർദ്ധാരണം ചെയ്യാൻ ഒരു പ്രോഗ്രാമിന്റെ തനത് പ്രവർത്തനക്രമത്തെ ആയി മാറ്റം വരുത്തേണ്ടതായി വരും. ഇത് ചിലപ്പോൾ തെരെഞ്ഞെടുക്കൽ, ഒഴിവാക്കൽ, ഒന്നോ അധിലധികമോ പ്രസ്താവനകളുടെ ആവർത്തിച്ചുള്ള നിർദ്ധാരണം എന്നിവയാകാം. ഇത്തരം തീരുമാനങ്ങൾ കൈക്കൊള്ളുന്നത് ചില നിബന്ധനകളെ അടിസ്ഥാനമാക്കിയാണ്. C++ ഈ ആവശ്യം നിറവേറ്റുന്നത് നിയന്ത്രണ പ്രസ്താവനകളുടെ സഹായത്തോടെയാണ്. നിയന്ത്രണ പ്രസ്താവനകളെ രണ്ടായി തരം തിരിക്കാം. (1) തീരുമാനമെടുക്കൽ/തെരെഞ്ഞെടുക്കൽ പ്രസ്താവനകൾ (2) ആവർത്തന പ്രസ്താവനകൾ എങ്ങനെ ഇത്തരം പ്രസ്താവനകൾ പ്രശ്നങ്ങൾ പരിഹരിക്കുവാൻ സഹായകമാകുന്നു എന്ന് നമുക്ക് കാണാം.

1.2.1 തീരുമാനങ്ങൾ എടുക്കുന്നതിനുള്ള പ്രസ്താവനകൾ (Selection statements)

നിബന്ധനകളെ ആധാരമാക്കി ഒരു കൃത്യത്തെ തിരെഞ്ഞെടുക്കുന്നതിനായി C++ രണ്ടു തരം പ്രസ്താവനകൾ ആണഅ ഉപയോഗിക്കുന്നത്. If, Switch എന്നിവയാണ് ഇവ. If പ്രസ്താവനയ്ക്ക് രണ്ടു തരം പതിപ്പുകൾ ആണ് ഉള്ളത്: ലളിതമായ if, if-else ഉം else. if .... ചുവടെ ചേർത്തിരിക്കുന്ന പ്രോഗ്രാം ഈ പ്രസ്താവനകളുടെ പ്രവർത്തന രീതികൾ വിവരിക്കുന്നു.

പ്രോഗ്രാം:– 1.2 തന്നിരിക്കുന്ന മൂന്നു സ്കോറിൽ നിന്ന് ഏറ്റവും ഉയർന്ന CE സ്കോർ കണ്ടുപിടിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{
   short int ce1, ce2, ce3, final_ce;
   cout<<"Enter three CE scores: ";
   cin>>ce1>>ce2>>ce3;
   if (ce1>ce2)
     final_ce=ce1;//നിബന്ധന ശരി എങ്കിൽ പ്രവർത്തിക്കുന്നു.
   else
     final_ce=ce2;//നിബന്ധന തെറ്റ് എങ്കിൽ പ്രവർത്തിക്കുന്നു.
```

```
    if (ce3>final_ce) final_ce=c3;//No else block for this if
    cout<<"Final CE Score is "<<final_ce;
    return 0;
}
```

പ്രോഗ്രാം 1.2 Short int ഡാറ്റ ഇനമാണ് വേരിയബിൾ പ്രഖ്യാപിക്കുവാനായി ഉപയോഗിച്ചിരിക്കുന്നത്. Gcc-യിൽ int ഡാറ്റ ഇനം 4 ബൈറ്റ്സ് മെമ്മറി ഉപയോഗിക്കുമ്പോൾ short 2 ബൈറ്റ്സ് മാത്രമേ ഉപയോഗിക്കുന്നുള്ളൂ. പ്രോഗ്രാം if-else പ്രസ്താവനയും ലളിതമായ if പ്രസ്താവനയും ഉപയോഗിക്കുന്നു. പ്രാരംഭത്തിൽ ce1>ce2 എന്ന നിബന്ധന പ്രസ്താവന പ്രവർത്തിക്കുന്നു. ഈ പ്രസ്താവന ശരിയെങ്കിൽ final-ce യിലേക്ക് ce1 ന്റെ വില നൽകുന്നു, അത് അല്ലെങ്കിൽ ce2 ന്റെ വിലയാകും final-ce യിലെക്ക് നൽകുക. ഇതിനുശേഷം ce3 യുടെ വില final-ce യിലുള്ള വില യുമായി താരതമ്യം ചെയ്യുന്നു. ce3 യിലെ സ്കോർ വലുതാണെങ്കിൽ, final-ce യിൽ ce3 യുടെ വില ശേഖരിക്കുകയും, അല്ലെങ്കിൽ ഒരു മാറ്റവും final-ce യിൽ ഉണ്ടാകുന്നതുമില്ല.

else if ലാഡർ എന്നത് ബഹുമുഖ ശാഖകൾ ഉള്ള പ്രസ്താവനയാണ്. പ്രോഗ്രാം 1.3 else-if ലാഡറിന്റെ പ്രവർത്തനം വിവരിക്കുന്നു. ഒരു ക്യാരക്റ്റർ സ്വീകരിക്കുകയും അത് അക്ഷരമാണോ, അക്കമാണോ, അതല്ല മറ്റ് ക്യാരക്റ്റർ ആണോ എന്ന് ഔട്ട്പുട്ട് നൽകുന്നു.

പ്രോഗ്രാം 1.3 ക്യാരക്റ്റർ ചെറിയ അക്ഷരമാണോ, വലിയ അക്ഷരമാണോ, അക്ഷരമാണോ, മറ്റു ക്യാരക്റ്റർ ആണോ എന്ന് പരിശോധിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{
    char ch;
    cout<<"Enter a character: ";
    cin>>ch;
    if (ch>='A' && ch<='Z')
        cout<<"Uppercase letter";
            else if (ch>='a' && ch<='z')
                cout<<"Lowercase letter";
                    else if (ch>='0' && ch<='9')
                        cout<<"Digit";
                            else
                                cout<<"Other character";
    return 0;
}
```

പ്രോഗ്രാം 1.3 വ്യത്യസ്ത നാല് വിലകളിൽ നിന്ന് ഒന്നിനെ തിരഞ്ഞെടുക്കുവാൻ else if ലാഡറോ else if സ്റ്റെയർ കെയ്സോ ഉപയോഗിക്കുന്നു.

ചില അവസരങ്ങളിൽ ഇന്റിജർ തുല്യത നിബന്ധനകൾ ഉപയോഗിച്ചാണ് തീരുമാനം എടുത്തിരിക്കുന്നത്.

else if ലാഡറിന് പകരം switch പ്രസ്താവന ഉപയോഗിക്കാവുന്നതാണ്. ക്യാരക്റ്റർ ഡാറ്റ ഇനവും ഒരു ഇൻറ്റിജർ ആയതിനാൽ, തുല്യതാ പരിശോധനയ്ക്ക് അവ ഉപയോഗിക്കാവുന്നതാണ്. പ്രോഗ്രാം 1.4 ഈ ആശയം വിശദീകരിക്കുന്നു. ഈ പ്രോഗ്രാം a, b, c d എന്നീ നാല് അക്ഷരങ്ങളിൽ നിന്ന് ഒന്ന് സ്വീകരിക്കുന്നു. 'a' എന്ന അക്ഷരമാണ് എങ്കിൽ 'Abacus' എന്ന് വാക്ക് പ്രദർശിപ്പിക്കുന്നു. അതുപോലെ 'Binary' എന്ന് 'b' യ്ക്കും 'computer' എന്ന് 'c' യ്ക്കും 'debugging' എന്ന് 'd' യ്ക്കും പ്രദർശിപ്പിക്കുന്നു. തന്നിരിക്കുന്ന പ്രശ്നത്തിന് default പ്രസ്താവനയുടെ ആവശ്യമില്ല. തന്നിരിക്കുന്ന നാല് ക്യാരക്റ്റർ അല്ലാതെ മറ്റൊന്നു ഇൻപുട്ട് നൽകിയാൽ പ്രോഗ്രാം, ഈ അവസരയിൽ പ്രതികരിക്കുകയില്ല. ആയതിനാൽ കൂടുതൽ ഉപയോക്ത സൗഹൃദമാക്കുവാനായി default case കൂടി പ്രോഗ്രാമിൽ ഉപയോഗിച്ചിരിക്കുന്നു.

പ്രോഗ്രാം 1.4 തന്നിരിക്കുന്ന ക്യാരക്റ്ററിനു പകരം ഒരു വാക്ക് പ്രദർശിപ്പിക്കുന്നു

```
#include <iostream>
using namespace std;
int main()
{
    char ch;
    cout<<"Enter a, b, c or d: ";
    cin>>ch;
    switch(ch)
    {
      case 'a': cout<<"Abacus";
                break;
      case 'b': cout<<"Binary";
                break;
      case 'c': cout<<"Computer";
                break;
      case 'd': cout<<"Debugging";
                break;
      default : cout<<"Invalid input!!";
    }
    return 0;
}
```

പ്രോഗ്രാം 1.4 ബഹുമുഖ ശാഖ എന്ന ആശയം ഉപയോഗിച്ചിരിക്കുന്നു. വ്യത്യസ്ത അവസരയിൽ നിന്ന് ഒരെണ്ണം മാത്രം പ്രവർത്തിക്കാനും തിരെഞ്ഞെടുക്കൽ എന്നത് switch ലെ പ്രയോഗം നൽകുന്ന വിലയും, ഏതെങ്കിലും case പ്രസ്താവനകളിൽ കൊടുത്തിരിക്കുന്ന സ്ഥിരാംഗത്തിന് തുല്യമാണോ എന്നതിനെ അടിസ്ഥാനമാക്കിയാണ്. ഒരു സ്ഥിരാംഗവുമാകും തുല്യത കണ്ടെത്തിയില്ലെങ്കിൽ default case ആയിരിക്കും പ്രവർത്തിക്കുക.

നമുക്കു ചെയ്യാം.

പ്രോഗ്രാം 1.4 ലെ switch പ്രസ്താവനയേ മാറ്റി else if ലാഡർ ഉപയോഗിക്കുക. പ്രോഗ്രാം 1.4 ൽ നിന്ന് break പ്രസ്താവന ഒഴിവാക്കിയാൽ എന്താകും ഔട്ട്പുട്ട്?

പ്രോഗ്രാം 1.3 ൽ നിന്ന് else-if ഗ്രാഡറിന് പകരം switch ഉപയോഗിക്കുവാൻ കഴിയില്ല. എന്തുകൊണ്ട്?

കണ്ടീഷണൽ ഓപ്പറേറ്റർ (conditional operator (?))

ഇത് C++ ലെ ഒരു ടേർണറി ഓപ്പറേറ്റർ ആണ് ഇത് ഉപയോഗിക്കുന്നതിന് മൂന്ന് ഓപ്പറന്റുകൾ ആവശ്യമാണ്. പ്രോഗ്രാം 1.2 ൽ ഉപയോഗിച്ചിരിക്കുന്ന if-else പ്രസ്താവനയേ ഒഴിവാക്കി ചുവടെ കൊടുത്തിരിക്കുന്നത് പോലെ എഴുതാം.

```
final-ce = (ce1>ce2)? ce1:ce2;
```

മൂന്നു സ്കോറിൽ നിന്ന് വലിയ സ്കോർ കണ്ടെത്തുന്നതിനായി കണ്ടിഷണൽ ഓപ്പറേറ്ററിന്റെ നെസ്റ്റിംഗ് ഉപയോഗിക്കാവുന്നതാണ്.

```
final - ce = (ce1>ce2) ? (ce1>ce3) ? (cel:ce3) : [(ce2>ce3)?ce2:ce3);
```

ആവർത്തന പ്രസ്താവനകൾ (Looping statements)

C++ ൽ മൂന്ന് തരം ആവർത്തന പ്രസ്താവനകൾ ഉണ്ട് while, for ഉം do-while. ഒരു ആവർത്തന പ്രസ്താവനയക്ക് 4 ഘടകങ്ങൾ ആണ് ഉള്ളഥ്. പ്രാരംഭ വില നൽകൽ (init) പരിശോധന പ്രയോഗം (Test expression), പരിഷ്ക്കരിക്കൽ പ്രസ്താവന (Updation statement), ലൂപ്പിന്റെ ചട്ടക്കൂട് (Body of loop). ആവർത്തിക്കപ്പെടേണ്ട പ്രസ്താവനകൾ ഉപയോഗിച്ച് ലൂപ്പിൻരെ ചട്ടക്കൂട് രൂപപ്പെടുത്തുന്നു. പരിശോധന പ്രയോഗത്തിന്റെ വില ശരിയായി തുടരുന്നത് വരെ ലൂപ്പ് പ്രവർത്തിക്കുന്നു. പരിശോധ പ്രയോഗത്തിൽ ഉപയോഗിച്ചിരിക്കുന്ന വേരിയബിളിനെ ലൂപ്പ് നിയന്ത്രണ വേരിയബിൾ (Loop contraol variable) എന്ന് അറിയപ്പെടുകയും ഇതിന്റെ പ്രാരംഭവില പ്രാരംഭ വില നൽകൽ പ്രസ്താവന വഴി ലഭ്യമാകുകയും ചെയ്യുന്നു. പരിഷ്ക്കരിക്കൽ പ്രസ്താവന ലൂപ്പ് നിയന്ത്രണ വേരിയബിളിന്റെ വിലയിൽ മാറ്റം വരുത്തുന്നു ഈ പ്രസ്താവന അടുത്ത ആവർത്തനത്തിന് മുന്നേ പ്രവർത്തിക്കുന്നു.

ലൂപ്പിംഗ് പ്രസ്താവന അഥവാ ആവർത്തന പ്രസ്താവനകളെ രണ്ട് രീതിയിൽ തരം തിരിക്കാം ആഗമന നിയന്ത്രമ ലൂപ്പ് എന്നും ബഹിർഗമന നിയന്ത്രണ ലൂപ്പ് എന്നും. ആഗമന നിയന്ത്രണ ലൂപ്പിൽ പരിശോധന പ്രയോഗം ലൂപ്പിന്റെ പ്രർത്തനത്തിന് മുൻപ് തന്നേ പരിശോധിക്കപ്പെടുന്നു. ലൂപ്പിന് ഉള്ളിലേക്ക് കടക്കണമെങ്കിൽ പരിശോധ പ്രയോഗത്തിന്റെ ഫലം ശരിയാകണം. while ഉം for ഉം ആഗമന നിയന്ത്രണ ലൂപ്പിന് ഉദാഹരണങ്ങൾ ആണ്. എന്നാൽ ബഹിർഗമന നിയന്ത്രണ ലൂപ്പിൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിച്ച ശേഷമാണ് നിബന്ധകൾ പരിശോധിക്കുന്നത്. ആയതിനാൽ ഒരു തവണയെങ്കിലും സഹിർഗമന നിയന്ത്രണ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിച്ചിരിക്കും. do-while ലൂപ്പ് ഈ വിഭാഗത്തിൽപ്പെടുന്നതാണ്.

ലൂപ്പിലെ ഘടകങ്ങളായ പ്രാരംഭവില നൽകൽ, പരിശോധ പ്രയോഗം, പരിഷ്കരിക്കൽ പ്രസ്താവന എന്നിവ ഒരുമിച്ചാമഞ for പ്രസ്താവനയിൽ നൽകിയിരിക്കുന്നത്. എന്നാൽ while ഉം do-whild ലും പ്രാരംഭവില നൽകൽ പ്രയോഗം ലൂപ്പ് ന് മുമ്പും പരിഷ്കരിക്കൽ പ്രസ്താവന ലൂപ്പിന്റെ ചട്ടക്കൂടിനുള്ളിലും നൽകിയിരിക്കുന്നു. പരിശോധ പ്രയോഗം while എന്ന വാക്കിന്റെ കൂടെ തന്നെ നൽകിയിരിക്കുന്നു. ഇവയുടെ പ്രവർത്തനം മനസ്സിലാക്കുവാനായി ചില പ്രോഗ്രാമുകൾ നോക്കാം.

പ്രോഗ്രാം 1.5 ഒരു നമ്പറിലെ അക്കങ്ങളുടെ തുക കണ്ടുപിടിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{
    int num, sum=0, dig;
    cout<<"Enter a number: ";
    cin>>num;
    while (num>0)
    {
        dig=num%10;
        sum=sum+dig;
        num=num/10;
    }
    cout<<"Sum of the digits of the input number = "<<sum;
    return 0;
}
```

പ്രോഗ്രാം 1.5 ൽ num എന്ന ലൂപ്പ് നിയന്ത്രണ വേരിയബിൾ ആയി പരിശോധന പ്രയോഗത്തിൽ ഉപയോഗിച്ചിരിക്കുന്നു. ഇൻപുട്ട് പ്രസ്താവന വഴി ഈ വേരിയബിളിന്റെ പ്രാരംഭ വില നൽകൽ നടത്തിയിരിക്കുന്നു. ലൂപ്പ് ചട്ടക്കൂടിൽ ആണ് ഇതിന്റെ പരിഷ്കരിക്കൽ പ്രസ്താവന നൽകിയിരിക്കുന്നത്. ഇൻപുട്ട് പ്രസ്താവന വഴി ലഭിക്കുന്ന അക്കം പോസിറ്റീവ് നമ്പർ അല്ലെങ്കിൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കുകയില്ല.

for പ്രസ്താവന ഉപയോഗിച്ച് നിർമിച്ച ലൂപ്പ് ഉള്ള മറ്റൊരു പ്രോഗ്രാം നമുക്ക് കാണാം.

പ്രോഗ്രാം 1.6 ആദ്യത്തെ N എണ്ണൽ സംഖ്യകളുടെ തുക കണ്ടുപിടിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int main()
{
    int n, sum=0;
    cout<<"Enter the limit: ";
    cin>>n;
    for(int i=1; i<=n; i++)
        sum=sum+i;
    cout<<"Sum of the first "<<n<<" natural numbers = "<<sum;
    return 0;
}
```

പ്രോഗ്രാം 1.6 ൽ i എന്നത് ഒരു ലൂപ്പ് നിയന്ത്രണ വേരിയബിൾ ആയും ഇതിനെ ഒരു വില നൽകൽ പ്രസ്താവന വഴി പ്രാരംഭ വില നൽകൽ നടത്തിയിരിക്കുന്നു. ഇതിന്റെ പരിഷ്കരിക്കൽ വില കൂട്ടൽ പ്രസ്താവന വഴി ആണ് ചെയ്തിരിക്കുന്നത്.

ഈ പ്രോഗ്രാമിന് ലൂപ്പ് ആധാരമാക്കിയുള്ള എണ്ണൽ ആണ് നടക്കുന്നത്. for പ്രസ്താവന ഇത് പോലുള്ള അവസരങ്ങളിൽ for പ്രസ്താവനകൾ ആണ് ഏറ്റവും അനുയോജ്യം. നേരത്തെ സൂചിപ്പിച്ചത് പോലെ do while ലൂപ്പ് ലൂപ്പിന്റെ ചട്ടക്കൂട് ഒരു പ്രാവിശ്യം എങ്കിലും പ്രവർത്തിപ്പിച്ചിരിക്കും. ഇതിന്റെ ഉപയോഗം താഴത്തേ പ്രോഗ്രാമിൽ വിശദീകരിച്ചിരിക്കുന്നു.

പ്രോഗ്രാം 1.7 ഒരു കൂട്ടം വിദ്യാർഥികളുടെ ഉയര അളവിന്റെ ശരാശരി കണക്കാക്കുന്നു.

```
#include <iostream>
using namespace std;
int main()
{
    float hgt, sum=0, avg_hgt;
    short n=0;
    char ch;
    do
    {
        cout<<"Enter the height: ";
        cin>>hgt;
        n++;
        sum=sum+hgt;
        cout<<"Any more student (Y/N)? ";
        cin>>ch;
    }while (ch=='Y' || ch=='y');
    avg_hgt=sum/n;
    cout<<"Average Height = "<<avg_hgt;
    return 0;
}
```

പ്രോഗ്രാം 1.7 ന്റെ ലൂപ്പ്ചട്ടക്കൂട് ഉപയോക്താവ് 'y' അല്ലെങ്കിൽ 'y' എന്ന് നൽകുന്നിടത്തോളം അവർത്തിച്ച് പ്രവർത്തിച്ച് കൊണ്ടിരിക്കും.

നമുക്കു ചെയ്യാം.

പ്രോഗ്രാം 1.5 ന്റെ ലൂപ്പ് പ്രസ്താവന for ഉം do-while ഉം ഉപയോഗിച്ച് മാറ്റിയെടുക്കുക.

പ്രോഗ്രാം 1.6 ന്റെ ലൂപ്പ് പ്രസ്താവന whild ഉം do-while ഉം ഉപയോഗിച്ച് മാറ്റിയെടുക്കുക.

പ്രോഗ്രാം 1.7 ന്റെ ലൂപ്പ് പ്രസ്താവന for ഉം while ഉം ഉപയോഗിച്ച് മാറ്റിയെഴുതുക.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. C++ ലെ തിരഞ്ഞെടുക്കൽ പ്രസ്താവനകൾ ഏതെല്ലാം?
2. ലൂപ്പ് പ്രസ്താവനകളുടെ നാല് ഘടകങ്ങൾ ഏന്തൊക്കെയാണ്?
3. ആഗമന നിയന്ത്രണ ലൂപ്പ് പ്രസ്താവനയ്ക്ക് ഉദാഹരണം എഴുതുക.
4. കണ്ടിഷനൽ ഓപ്പറേറ്റർ (?) ക്ക് തുല്യമായ നിയന്ത്രണ കൈമാറ്റ പ്രസ്താവന ഏതാണ്.
5. എല്ലാ തരത്തിലുള്ള switch പ്രസ്താവനകളും if ന്റെ വിവിധ രൂപത്തിലുള്ള പ്രസതാവനകൾ ഉപയോഗിച്ച് മാറ്റിയെഴുതാം. ശരിയോ തെറ്റോ എന്ന് പ്രസ്താപിക്കുക.

1.3 ലൂപ്പുകളുടെ നെസ്റ്റിങ്ങ് (Nesting of loops)

ഒരു ലൂപ്പിനകത്ത് മറ്റൊരു ലൂപ്പ് ഉൾപ്പെടുത്തുന്നതിനെ ലൂപ്പുകളുടെ നെസ്റ്റിങ്ങ് എന്നു പറയുന്നു. രണ്ട് ലൂപ്പുകൾ നാം നെസ്റ്റ് ചെയ്യുമ്പോൾ പുറത്തുള്ള ലൂപ്പ് (Outer loop) അകത്തുള്ള ലൂപ്പ് എത്ര തവണ പ്രവർത്തിച്ചു എന്ന് തിട്ടപ്പെടുത്തുന്നു. ഇവിടെ രണ്ടു ലൂപ്പുകളുടെയും ലൂപ്പ് നിയന്ത്രണ വേരിയബിളുകൾ (Loop control variable) വ്യത്യസ്തമായിരിക്കണം.

നെസ്റ്റഡ് ലൂപ്പ് എങ്ങനെ പ്രവർത്തിക്കുന്നു എന്ന് നമുക്കു നോക്കാം. ഒരു ക്ലോക്കിലെ മിനുട്ട് സൂചിയുടെയും, സെക്കന്റ് സൂചിയുടെയും കാര്യം എടുക്കുക. നിങ്ങൾ ക്ലോക്കിന്റെ പ്രവർത്തനം ശ്രദ്ധിച്ചിട്ടുണ്ടോ? മിനുട്ട് സൂചി ഏതെങ്കിലും ഒരു സ്ഥാനത്ത് നിൽക്കുമ്പോൾ സെക്കന്റ് സൂചി ഒരു ഭ്രമണം പൂർത്തിയാക്കുന്നു (1 മുതൽ 60 വരെ). സെക്കന്റ് സൂചി ഒരു ഭ്രമണം പൂർത്തിയാക്കിയതിനുശേഷം മിനുട്ട് സൂചി അടുത്ത സ്ഥാനത്തേക്ക് മാറുന്നു. മിനുട്ട് സൂചിയുടെ ഓരോ സ്ഥാനത്തിനും അനുസൃതമായി സെക്കന്റ് സൂചി ഭ്രമണം പൂർത്തിയാക്കുന്നു. ഈ പ്രക്രിയ തുടർന്നു കൊണ്ടേയിരിക്കുന്നു. ഇവിടെ സെക്കന്റ് സൂചിയുടെ ചലനം ഉള്ളിലെ ലൂപ്പിന്റെ പ്രവർത്തനമായും മിനുട്ട് സൂചിയുടെ ചലനം ബാഹ്യലൂപ്പിന്റെ പ്രവർത്തനമായും കരുതാവുന്നതാണ്. C++ ലെ എല്ലാ ലൂപ്പുകളും നെസ്റ്റിങ്ങ് അനുവദിക്കുന്നു. താഴെ കൊടുത്തിരിക്കുന്ന ഉദാഹരണം `for` ലൂപ്പിന്റെ നെസ്റ്റിങ്ങ് പ്രവർത്തനം കാണിച്ചുതരുന്നു.

ചിത്രം 1.2 ൽ നൽകിയിരിക്കുന്നതു പോലെ മിനിറ്റു സൂചി മാറ്റില്ലാതെ തുടർന്നു കൊണ്ട് സെക്കന്റ് സൂചിയുടെ മൂല്യം 0 ൽ നിന്ന് 59 ലേക്ക് മാറുന്നു. സെക്കന്റ് സൂചിയുടെ മൂല്യം 59 ൽ എത്തിയാൽ പിന്നെ മാറ്റം മിനിറ്റു സൂചിയിലായിരിക്കും. മിനിറ്റു സൂചിയിലെ മാറ്റത്തിനു ശേഷം സെക്കന്റ് സൂചിയുടെ മൂല്യം പൂജ്യത്തിലേക്കു വീണ്ടുമെത്തുന്നു.

ചിത്രം. 1.2: ഡിജിറ്റൽ വാച്ചിലൂടെ നെസ്റ്റിങ് ലൂപ്പ് എന്ന ആശയം

C++ ലെ എല്ലാ ലൂപ്പുകളും നെസ്റ്റിങ് അനുവദിക്കുന്നു. നെസ്റ്റഡ് ലൂപ്പിന്റെ പ്രവർത്തനം മനസ്സിലാക്കുന്നതിനുള്ള ഒരു ഉദാഹരണം ചുവടെ ചേർത്തിരിക്കുന്നു.

```
for( i=1; i<=2; ++i)        Outer loop
{
     for(j=1; j<=3; ++j)    Inner loop
     {
```

```
            cout<< "\n" << i << " and " << j;
    }
}
```

ബാഹ്യലൂപ്പിലെ വേരിയബിളായ `i` ക്ക് പ്രാരംഭ വിലയായി 1 നൽകുന്നു. അതിന്റെ പരിശോധന പ്രയോഗം വിലയിരുത്തി ശരിയായതിനാൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിക്കുന്നു. ചട്ടക്കൂടിൽ അടങ്ങിയിരിക്കുന്നത് നിയന്ത്രണ വേരിയബിൾ `j` യോടുകൂടിയ ആന്തരിക ലൂപ്പാണ്. `j` ക്ക് പ്രാരംഭ വിലയായ 1 നൽകി അതിന്റെ പ്രവർത്തനം ആരംഭിക്കുന്നു. `j` =1, `j`=2, `j`=3 ആയി ആന്തരികലൂപ്പ് 3 തവണ പ്രവർത്തിക്കുന്നു. ഓരോ തവണയും `j<=3` എന്ന പരിശോധന പ്രയോഗം വിലയിരുത്തുകയും ശരിയായതിനാൽ ഔട്ട്പുട്ട് പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു.

```
1 and 1
1 and 2
1 and 3
```

The first 1 is of `i` and the second 1 is of `j`

പരിശോധന പ്രയോഗം `j<=3` തെറ്റാവുമ്പോൾ പ്രോഗ്രാമിന്റെ നിയന്ത്രണം ആന്തരിക ലൂപ്പിൽ നിന്നും പുറത്തു കടക്കുന്നു. ഇപ്പോൾ ബാഹ്യ ലൂപ്പിന്റെ പുതുക്കൽ പ്രസ്താവന പ്രവർത്തിച്ച് `i`=2 ആക്കുന്നു. പരിശോധന പ്രയോഗമായ `i<=2` പരിശോധിച്ച് ശരിയായതിനാൽ ലൂപ്പിന്റെ ചട്ടക്കൂട് ഒന്നുകൂടി പ്രവർത്തിക്കുന്നു. `j` =1, `j`=2, `j`=3 ആയി ആന്തരിക ലൂപ്പ് വീണ്ടും മൂന്നു തവണ പ്രവർത്തിച്ച് ഔട്ട് പുട്ട് പ്രദർശിപ്പിക്കുന്നു.

```
2 and 1
2 and 2
2 and 3
```

ആന്തരിക ലൂപ്പിന്റെ പ്രവർത്തനം പൂർത്തിയാക്കിയതിനുശേഷം നിയന്ത്രണം പുറത്തെ ലൂപ്പിന്റെ വില പുതുക്കൽ പ്രയോഗത്തിൽ തിരിച്ചെത്തുന്നു. `i` യുടെ വില 1 വെച്ച് വർദ്ധിപ്പിക്കുന്നു. (ഇപ്പോൾ `i`=3) പരിശോധന പ്രയോഗം `i<=2` വിലയിരുത്തുമ്പോൾ തെറ്റാവുന്നു. ആയതിനാൽ ലൂപ്പ് അതിന്റെ പ്രവർത്തനം അവസാനിപ്പിക്കുന്നു. പട്ടിക 7.3 മുകളിൽ കൊടുത്ത പ്രോഗ്രാം ശകലത്തിന്റെ പ്രവർത്തനം വിവരിക്കുന്നു.

Iterations	Outer loop (i)	Inner loop (j)	Output
1	1	1	1 and 1
2	1	2	1 and 2
3	1	3	1 and 3
4	2	1	2 and 1
5	2	2	2 and 2
6	2	3	2 and 3

ഠമയഹല 1.2: ഋഃലരൗശേീി ീള ിലലെേ ഹീീു

നെസ്റ്റഡ് ലൂപ്പുകളിൽ പ്രവർത്തിക്കുന്ന സമയത്ത് ബാഹ്യലൂപ്പിലെ നിയന്ത്രണ വേരിയബിളുകളിൽ അവയുടെ വിലയിൽ മാറ്റം വരുന്നത് ആന്തരികലൂപ്പ് പൂർത്തികരിച്ചതിനുശേഷം മാത്രമാണ്. ഇനി താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിലുള്ള ത്രികോണം പ്രദർശിപ്പിക്കാനുള്ള ഒരു പ്രോഗ്രാം നമുക്കു എഴുതാം.

```
*
* *
* * *
* * * *
* * * * *
```

പ്രോഗ്രാം 7.8: ത്രികോണാകൃതിയിൽ നക്ഷത്രചിഹ്നം പ്രദർശിപ്പിക്കുന്നതിന്.

```
#include<iostream>
using namespace std;
int main()
{
    short int i, j;
    for(i=1; i<=5; ++i)            // ആന്തരിക ലൂപ്പ്
     {
          cout<< "\n" ;
        for(j=1; j<=i; ++j)        // ആന്തരിക ലൂപ്പ്
               cout<< '*';
    }
    return 0;
}
```

നമുക്കു ചെയ്യാം

cout<<i; എന്ന പ്രസ്താവനയ്ക്ക് പകരം cout<<'*'; എന്ന പ്രസ്താവന ആണെങ്കിൽ പ്രോഗ്രാം 1.8 ന്റെ ഔട്ട്പുട്ട് എന്തായിരിക്കും?

അതുപോലെ, cout<<j; എന്ന പ്രസ്താവന ആണെങ്കിൽ എന്തായിരിക്കും ഔട്ട്പുട്ട്.

1.4 ജംപ് പ്രസ്താവനകൾ (Jump Statements)

പ്രോഗ്രാമിന്റെ നിയന്ത്രണം ഒരു ഭാഗത്തുനിന്നും മറ്റൊരു ഭാഗത്തേക്ക് മാറ്റാൻ ഉപയോഗിക്കുന്ന പ്രസ്താവനകളെ ജംപ് പ്രസ്താവനകൾ (Jump statements) എന്നു പറയുന്നു. C++ ൽ പ്രത്യേക നിബന്ധനകളില്ലാതെ പ്രവർത്തിക്കുന്ന നാലുതരം ജംപ് പ്രസ്താവനകൾ ഉണ്ട്. അവ **return**, **goto**, **break**, **continue** എന്നിവയാണ് ഇതിനുപുറമെ, C++ ലെ **exit()** എന്ന സ്റ്റാൻഡേർഡ് ലൈബ്രറി ഫങ്ഷൻ പ്രോഗ്രാമിന്റെ പ്രവർത്തനം അവസാനിപ്പിക്കുന്നതിനും ഉപയോഗിക്കുന്നുണ്ട്.

return പ്രസ്താവന ഫംങ്ഷനിൽ നിന്ന് പുറത്ത് വരുന്നതിനും നിയന്ത്രണം, വിളിച്ച പ്രോഗ്രാമിലേക്ക് തിരിച്ചു കൊണ്ടു പോകുന്നതിനും ഉപയോഗിക്കുന്നു. അധ്യായം 10 ൽ ഇതിനെക്കുറിച്ച് പിന്നീട് വിശദീകരിച്ചിട്ടുണ്ട്. ഇനി നമുക്ക് മറ്റു ജംപ് പ്രസ്താവനകളെക്കുറിച്ച് ചർച്ച ചെയ്യാം.

1.4.1 goto പ്രസ്താവന (goto statement)

goto പ്രസ്താവന ഉപയോഗിച്ച് പ്രോഗ്രാം നിയന്ത്രണത്തെ ഫംങ്ഷനിലെ ഏതു സ്ഥലത്തേക്കും മാറ്റാൻ സാധിക്കും. ഒരു goto പ്രസ്താവനയുടെ ലക്ഷ്യസ്ഥാനം ലേബൽ (ഒരു ഐഡന്റിഫയർ) ഉപയോഗിച്ച് അടയാളപ്പെടുത്തുന്നു. goto പ്രസ്താവനയുടെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
          goto ലേബൽ ;
          ............;
          ............;
     ലേബൽ: ..........;
          ............;
```

goto പ്രസ്താവനക്ക് മുമ്പോ പിൻപോ ഒരു പ്രോഗ്രാമിൽ കാണപ്പെടുന്നു. ലേബലിനുശേഷം ഒരു അപൂർണവിരാമം (:) ചിഹ്നം ആവശ്യമാണ്. ഉദാഹരണത്തിന് 1 മുതൽ 50 വരെ പ്രിന്റ് ചെയ്യാനുള്ള കോഡ് ശകലം പരിഗണിക്കുക.

```
      int i=1;
  start:
      cout<<i;
      ++i;
      if (i<=50)
            goto start;
```

ഇവിടെ cout, പ്രസ്താവന 1 എന്ന വില പ്രിന്റ് ചെയ്യുന്നു. അതിനുശേഷം i യുടെ വില 1 വർദ്ധിപ്പിക്കുന്നു. (ഇപ്പോൾ i=2), ഇപ്പോൾ പരിശോധന പ്രയോഗം i<=50 വിലയിരുത്തുന്നു. നിബന്ധന ശരിയായതിനാൽ start എന്ന ലേബലിലേക്ക് പ്രോഗ്രാം നിയന്ത്രണത്തെ മാറ്റുന്നു. നിബന്ധന തെറ്റാവുമ്പോൾ പ്രവർത്തനം അവസാനിപ്പിച്ച് പ്രോഗ്രാം നിയന്ത്രണം if പ്രസ്താവനക്കു ശേഷം എത്തുന്നു. സ്ട്രക്ചേർഡ് പ്രോഗ്രാമിങ് goto ന്റെ ഉപയോഗം പ്രോത്സാഹിപ്പിക്കുന്നില്ല

1.4.2 ബ്രേക്ക് പ്രസ്താവന (break statement)

ഒരു പ്രോഗ്രാമിൽ break പ്രസ്താവന കാണപ്പെട്ടാൽ പ്രോഗ്രാമിന്റെ നിയന്ത്രണം തൊട്ടടുത്ത ലൂപ്പിനോ (for, while, do...while), switch പ്രസ്താവനയ്ക്കോ പുറത്തേക്ക് മാറ്റുന്നു. കൺട്രോൾ ചട്ടക്കൂടിന് ശേഷമുള്ള പ്രസ്താവന മുതൽ പ്രവർത്തനം തുടരുന്നു. switch പ്രസ്താവനയിൽ break ന്റെ പ്രവാഹത്തെ കുറിച്ച് നാം ഇതിനോടകം ചർച്ച ചെയ്തു കഴിഞ്ഞു. ഇത് ലൂപ്പുകളുടെ പ്രവർത്തനത്തെ എങ്ങനെ സ്വാധീനിക്കുന്നു എന്ന് നമുക്ക് നോക്കാം. താഴെ കൊടുത്തിരിക്കുന്ന രണ്ട് പ്രോഗ്രാം ശകലങ്ങൾ പരിഗണിക്കുക.

കോഡ് ശകലം 1

```
  i=1;
  while(i<=10)
```

```
    {
        cin>>num;
        if (num==0)
            break;
        cout<<"Entered number is: "<<num;
        cout<<"\nInside the loop";
        ++i;
    }
    cout<<"\nComes out of the loop";
```

മുകളിലെ പ്രോഗ്രാം 10 സംഖ്യകളെ ഇൻപുട്ട് ചെയ്യാൻ അനുവദിക്കുന്നു. ഇൻപുട്ട് ചെയ്യുമ്പോൾ ഏതെങ്കിലും ഒരു സംഖ്യ 0 ആണെങ്കിൽ ലൂപ്പ് ചട്ടക്കൂടിലെ ബാക്കി പ്രസ്താവനകളെ ഒഴിവാക്കി പ്രോഗ്രാമിന്റെ നിയന്ത്രണം ലൂപ്പിനു പുറത്ത് വരികയും "Comes out of the loop" എന്ന സന്ദേശം സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു. ഒരു നെസ്റ്റഡ് ലൂപ്പിൽ break പ്രസ്താവനഉപയോഗിക്കുന്ന മറ്റൊരു കോഡ് ശകലം നമുക്കു പരിഗണിക്കാം.

കോഡ് ശകലം 2

```
    for(i=1; i<=5; ++i)          //outer loop
    {
        cout<<"\n";
        for(j=1; j<=i; ++j)      //inner loop
        {
            cout<<"* ";
            if (j==3)
                break;
        }
    }
```

```
*
*  *
*  *  *
*  *  *  *
*  *  *  *  *
```

ഈ കോഡ് ശകലം താഴെ കൊടുത്തിരിക്കുന്ന മാതൃക പ്രദർശിപ്പിക്കുന്നു.

നെസ്റ്റഡ് ലൂപ്പ് സാധാരണയായി i=1, i=2, i=3 എന്നീ വിലകൾക്ക് അനുസരിച്ച് പ്രവർത്തിക്കുന്നു. i യുടെ ഓരോ വിലക്കനുസരിച്ച് j, 1 മുതൽ i വരെയുള്ള വിലകൾ സ്വീകരിക്കും. i യുടെ വില 4 ഓ 5 ഓ ആകുമ്പോൾ ഉള്ളിലുള്ള ലൂപ്പ് j = 1, j=2, j=3 എന്നീ വിലകൾക്കനുസരിച്ച് പ്രവർത്തിച്ച് break നു ശേഷം ലൂപ്പിൽ നിന്നും പുറത്ത് പോകുന്നു.

1.43 കണ്ടിന്യൂ പ്രസ്താവന (continue statement)

continue പ്രസ്താവന മറ്റൊരു ജംമ്പ് പ്രസ്താവനയാണ് അത് ലൂപ്പ് ചട്ടക്കൂടിന്റെ ഒരു ഭാഗം ഒഴിവാക്കി അടുത്ത ആവർത്തനത്തിലേക്ക് എത്തിക്കുന്നതിനു വേണ്ടി ഉപയോഗിക്കുന്നു. break പ്രസ്താവന ലൂപ്പിന്റെ പ്രവർത്തനം നിർത്തി വെയ്ക്കുമ്പോൾ continue പ്രസ്താവന ചില ഭാഗങ്ങൾ ഒഴിവാക്കി അടുത്ത ആവർത്തനം നടത്താൻ നിർബന്ധിക്കുന്നു. താഴെ കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാം ശകലം continue പ്രസ്താവനയുടെ പ്രവർത്തനം വിവരിക്കുന്നു.

```
for (i=1; i<=10; ++i)
    {
        if (i==6)
            continue;
        cout<<i<<"\t";
    }
```

ഈ കോഡ് താഴെ പറയുന്ന ഔട്ട്പുട്ട് തരുന്നു.

```
1     2     3     4     5     7     8     9     10
```

6 ലിസ്റ്റിൽ ഇല്ല എന്നത് ശ്രദ്ധിക്കുക. `i` യുടെ വില 6 ആകുമ്പോഴാണ് continue പ്രസ്താവന പ്രവർത്തിക്കുന്നത്. അതിന്റെ ഫലമായി ഔട്ട്പുട്ട് പ്രസ്താവന ഒഴിവാക്കി പ്രോഗ്രാം നിയന്ത്രണം അടുത്ത ആവർത്തനത്തിലെ പുതുക്കൽ പ്രസ്താവനയിൽ എത്തിച്ചേരുന്നു.

ഒരു ലൂപ്പിനകത്തെ `break` പ്രസ്താവന ലൂപ്പിനെ അവസാനിപ്പിക്കുകയും ലൂപ്പിനു ശേഷമുള്ള പ്രസ്താവനകളിലേക്ക് പ്രോഗ്രാം നിയന്ത്രണത്തെ എത്തിക്കുകയും ചെയ്യുന്നു. `continue` പ്രസ്താവന നിലവിലുള്ള ആവർത്തനത്തിലെ ശേഷിച്ച ഭാഗം ഉപേക്ഷിച്ച് ലൂപ്പിന്റെ അടുത്ത ആവർത്തനം ആരംഭിക്കുന്നു. `While` ലൂപ്പിലും, `do...while` ലൂപ്പിലും `continue` പ്രസ്താവന ഉപയോഗിക്കുമ്പോൾ ലൂപ്പ് അനന്തമാകുന്നത് ഒഴിവാക്കണം എന്നത് ശ്രദ്ധിക്കണം.

നമുക്കു ചെയ്യാം.

പട്ടിക 7.3 `break`, `continue` എന്നീ പ്രസ്താവനകൾ തമ്മിലുള്ള താരതമ്യം കാണിക്കുന്നു. പട്ടിക പൂർത്തിയാക്കുക.

`break` പ്രസ്താവന	`continue` പ്രസ്താവന
• .... • ബ്ലോക്കിലെ അവശേഷിക്കുന്ന പ്രസ്താവനകൾ ഒഴിവാക്കി പ്രോഗ്രാമിന്റെ നിയന്ത്രണം സ്വിച്ചിനോ ലൂപ്പിനോ പുറത്തേക്കു കൊണ്ടു വരുന്നു. •	• ലൂപ്പിന്റെ കൂടെ മാത്രം ഉപയോഗിക്കുന്നു. • .... • പരിശോധന പ്രസ്താവനയുടെ വില തെറ്റാകുമ്പോൾ മാത്രം പ്രോഗ്രാമിന്റെ നിയന്ത്രണം ലൂപ്പിന് പുറത്തു കൊണ്ടു പോകുന്നു.

പട്ടിക 1.3: break, continue എന്നീ പ്രസ്താവനകൾ തമ്മിലുള്ള താരതമ്യം

നെസ്റ്റഡ് ലൂപ്പു ആവശ്യമുള്ള ഒരു പ്രോഗ്രാം നമുക്ക് എഴുതാം.

പ്രോഗ്രാം 1.9: തന്നിരിക്കുന്ന സംഖ്യ അഭാജ്യ സംഖ്യയാണോ അല്ലെയോ എന്ന് പരിശോധിക്കുന്നതിന്.

```
#include<iostream>
using namespace std;
int main()
{   short int n, i, flag;
    cout<<"Prime numbers below 100 are...\n";
    for(n=2; n<=100; n++)   //Outer loop
    {   flag=1;
        for(i=2; i<=n/2; i++)     //Inner loop
            if(n%i==0)
            {   flag=0;
                break;//Takes the control outside the iiner loop
            }
        if(flag==1) cout<<n<<'\t';
    }
    return 0;
}
```

പ്രോഗ്രാം 1.8 ൽ, പുറത്തെ ലൂപ്പിലൂടെ n എന്നാ വേരിയബിൾ 2 മുതൽ 100 വരെയുള്ള വിലകൾ പ്രോഗ്രാമിന് നൽകുന്നു. ഈ ഓരോ വിലകളും അഭാജ്യമാണോ എന്ന് അകത്തെ ലൂപ്പ് ഉപയോഗിച്ച് പരിശോധിക്കുന്നു. 2 മുതൽ n/2 വരെയുള്ള ഏതെങ്കിലും ഒരു വില n ന്റെ ഘടകമായി വന്നാൽ, flag എന്നാ വേരിയബിളിന്റെ വില 1-ൽ നിന്ന് 0-മായി മാറ്റിക്കൊണ്ട് അകത്തെ ലൂപ്പ് പ്രവർത്തനം അവസാനിപ്പിക്കുന്നു. അകത്തെ ലൂപ്പിന്റെ പ്രവർത്തനത്തിന് ശേഷവും flag ന്റെ വില 1 ആയി നിലനിൽക്കുന്ന n നെ അഭാജ്യ സംഖ്യയായി പ്രദർശിപ്പിക്കുന്നു.

നമുക്ക് സംഗ്രഹിക്കാം

C++ ഭാഷയുടെ അടിസ്ഥാന ആശയങ്ങളെക്കുറിച്ചുള്ള അറിവുകൾ നമ്മൾ ഓർമ്മ പുതുക്കിയല്ലോ. ക്യാരക്ടർ സെറ്റ്, ടാറ്റ തരങ്ങൾ, തരങ്ങളുടെ പരീക്ഷണങ്ങൾ, പദപ്രയോഗങ്ങൾ, തരങ്ങളുടെ പരിവർത്തനം തുടങ്ങിയവയുടെ ആശയങ്ങൾ ക്യാപ്സ്യൂൾ രൂപത്തിൽ ഇവിടെ അവതരിപ്പിച്ചു. ഉദാഹരണസഹിതം വിവിധ തരത്തിലുള്ള C++ പ്രസ്താവനകളും അനുസ്മരിച്ചു. വിവിധ തരത്തിലുള്ള നിയന്ത്രണ കൈമാറ്റ പ്രസ്താവനകൾ പ്രോഗ്രാമുകളുടെ സഹായത്താൽ ചുരുക്ക രൂപത്തിൽ വിശദീകരിക്കുകയും, നെസ്റ്റേഡ് ലൂപ്പുകളും, രണ്ടു ജമ്പ് പ്രസ്താവനകൾ ആയ ബ്രേക്ക്, കണ്ടിന്യൂ തുടങ്ങിയവ പുതിയ ആശയങ്ങളായും അവതരിപ്പിച്ചു. ഈ പുസ്തകത്തിലെ അധ്യായം രണ്ടു, മൂന്നു എന്നിവയിലെ ആശയങ്ങൾ മനസിലാക്കുവാൻ ഇവയെക്കുറിച്ചുള്ള വ്യക്തമായ ധാരണ ആവശ്യമാണ്.

നമുക്ക് പരിശീലിക്കാം

1. 100 നും 200 നും ഇടയ്ക്കുള്ള പാലിൻഡ്രോം അക്കങ്ങൾ (മുന്നോട്ടും പിന്നോട്ടും ഒരു വായിക്കാൻ കഴിയുന്നവ) പ്രദർശിപ്പിക്കുന്നതിനുള്ള C++ പ്രോഗ്രാം എഴുതുക.
2. 1000 താഴെയുള്ള ആംസ്ട്രോങ് അക്കങ്ങൾ പ്രദർശിപ്പിക്കുന്നതിനുള്ള C++ പ്രോഗ്രാം എഴുതുക
3. 1000 താഴെയുള്ള പെർഫക്റ്റ് അക്കങ്ങൾ പ്രദർശിപ്പിക്കുന്നതിനുള്ള C++ പ്രോഗ്രാം എഴുതുക.
4. ഒരു അക്കത്തിന്റെ ഗുണനപട്ടിക പ്രദർശിപ്പിക്കുന്നതിനുള്ള C++ പ്രോഗ്രാം എഴുതുക.
5. ഉപഭോക്താക്കളുടെ വൈദ്യുതി ബിൽ പ്രദർശിപ്പിക്കുന്നതിനുള്ള C++ പ്രോഗ്രാം എഴുതുക.

300 യൂണിറ്റ് വരെ	:	Rs. 5.00/- 1 യൂണിറ്റിന്
350 യൂണിറ്റ് വരെ	:	Rs. 5.70/- 1 യൂണിറ്റിന്
400 യൂണിറ്റ് വരെ	:	Rs. 6.10/- 1 യൂണിറ്റിന്
500 യൂണിറ്റ് വരെ	:	Rs. 6.70/- 1 യൂണിറ്റിന്
500 യൂണിറ്റ് മുകളിൽ	:	Rs. 7.50/- 1 യൂണിറ്റിന്

എത്ര ഉപഭോക്താക്കളുടെ വിവരങ്ങളും ഉൾപ്പെടുത്താനുള്ള സൗകര്യം പ്രോഗ്രാമിൽ ലഭ്യമായിരിക്കണം.

നമുക്ക് വിലയിരുത്താം

1. താഴെ നൽകിയിരിക്കുന്ന കോഡ് ശകലത്തിലെ തെറ്റ് കണ്ടെത്തുക.

```
for (short i=1; i<5; ++i)
    for (i=5; i>0; --i)
        cout<<i<<"\t";
```

2. break, continue പ്രസ്താവനകൾ താരതമ്യം ചെയ്യുക

3. താഴെ നൽകിയിരിക്കുന്ന പ്രോഗ്രാം ശകലത്തിന്റെ ഔട്ട്പുട്ട് എഴുതുക.

```
for (outer=10; outer>5; --outer)
    for (inner=1; inner<4; ++inner)
        cout<<outer<<"\t"<<inner<<endl;
```

4. ചുവടെ ചേർത്തിരിക്കുന്ന ഔട്ട്പുട്ട് ലഭിക്കുന്നതായി നെസ്റ്റഡ് ലൂപ്പ് ഉപയോഗിച്ച് പ്രോഗ്രാം എഴുതുക.

```
A
A    B
A    B    C
A    B    C    D
A    B    C    D    E
```

5. ചുവടെ ചേർത്തിരിക്കുന്ന C++ കോഡ് ശ്രദ്ധിക്കുക

```
for (n=1; n<5; ++n)
{
    cout<<i;
    if (i==2) continue;
    if (i%3==0) break;
    cout<<"Hello";
}
```

ഈ കോഡിന്റെ ഔട്ട്പുട്ട് തിരഞ്ഞെടുക്കുക

a. 1Hello2Hello3Hello4Hello
b. 1Hello2Hello3
c. 1Hello23
d. 1Hello23Hello

6. ചുവടെ ചേർത്തിരിക്കുന്ന C++ കോഡ് ശകലം ശ്രദ്ധിക്കുക. ലൂപ്പിങ് പ്രസ്താവന ഉപയോഗിച്ച് മാറ്റിയെഴുതുക.

```
      cin>>n;
loop: r=n%10;
      s=s*10+r;
      n=n/10;
      if (n!=0) goto loop;
      cout<<s;
```

7. C++ ലെ കാരക്ടർ കോൺസ്റ്റന്റ് അല്ലാത്തത് തിരഞ്ഞെടുക്കുക?

   a. '\t' b. 'a' c. '9' d. '9a'

8. ചുവടെ ചേർത്തിരിക്കുന്നതിൽ അസാധുവായ ഐഡന്റിഫയർ തിരഞ്ഞെടുത്ത് കാരണം എഴുതുക.

   a. unsigned b. cpp c. 2num d. cout

9. switch പ്രസ്താവനകളിൽ break ഉപയോഗിച്ചില്ലെങ്കിൽ എന്ത് സംഭവിക്കും?

2 അറേകൾ

പ്രധാന പഠനനേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പഠനം പൂർത്തിയാക്കുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ:

- അറെ ഉപയോഗിക്കേണ്ട വ്യത്യസ്ത സാഹചര്യങ്ങൾ തിരിച്ചറിയുന്നു.
- ഡാറ്റ സമൂഹത്തെ സൂചിപ്പിക്കുന്നതിനായി അറെ ഉപയോഗിക്കുന്നു.
- അറെയുടെ മെമ്മറി നീക്കി വയ്ക്കൽ മനസിലാക്കുന്നു.
- പ്രശ്ന നിർധാരണത്തിനായി അറെ അംഗങ്ങളെ ഉപയോഗിക്കുന്നു.
- സ്ട്രിങ്ങുകൾ പ്രതിനിധികരിക്കുന്നതിനായി ക്യാരക്റ്റർ അറെ ഉപയോഗിക്കുന്നു.
- വ്യത്യസ്ത വേഡ് പ്രോസസ്സിങ് പ്രവർത്തനങ്ങൾക്കായി ക്യാരക്ടർ അറെ ഉപയോഗിക്കുന്നു.

I5D2Z5

പ്രോഗ്രാമുകളിൽ ഡാറ്റ സംഭരിക്കുന്നതിനായി നാം വേരിയബലുകൾ ഉപയോഗിക്കുന്നു. എന്നാൽ ഡാറ്റയുടെ എണ്ണം കൂടുതലാണെങ്കിൽ കൂടുതൽ വേരിയബലുകൾ ഉപയോഗിക്കേണ്ടതായി വരും. ഈ സാഹചര്യത്തിൽ ഡാറ്റ ഉപയോഗിക്കുന്ന രീതി വളരെ ബുദ്ധിമുട്ടുള്ളതായി അനുഭവപ്പെടും. ഇതു മറികടക്കാൻ ഈ അദ്ധ്യായത്തിൽ അറെ (Array) എന്ന പേരിലുള്ള C++ ൽ നിന്നും ഉരുത്തിരിഞ്ഞ ഡാറ്റ ഇനം പരിചയപ്പെടുത്തുന്നു. അറെ എന്നത് കേവലമൊരു ഡാറ്റ ഇനത്തിന്റെ നാമം മാത്രമല്ല, മറിച്ച് ഇത് വളരെ കൂടുതൽ ഡാറ്റ എളുപ്പത്തിൽ കൈകാര്യം ചെയ്യുന്നതിന് വേണ്ടി അടിസ്ഥാനപരമായ ഡാറ്റ ഇനങ്ങളിൽ നിന്നും നിർമിച്ചെടുത്ത മറ്റൊരു തരം ഡാറ്റ ഇനമാണ്. അറെയുടെ പ്രഖ്യാപനം പ്രാഥമിക വിലയിരുത്തൽ (Initialization), കടന്നുപോകൽ (Traversal), ക്രമപ്പെടുത്തൽ (Sorting), തിരയൽ (Searching) പോലുള്ള പ്രവർത്തനങ്ങളെപ്പറ്റി നമുക്ക് ചർച്ച ചെയ്യാം.

2.1 അറെയും അവയുടെ ആവശ്യകതയും (Array and its need)

അറെ എന്നാൽ തുടർച്ചയായ മെമ്മറി സ്ഥാനങ്ങളിൽ ശേഖരിച്ചു വച്ചിട്ടുള്ള ഒരേ ഇനത്തിലുള്ള ഡാറ്റകളുടെ സമൂഹമാണ്. ഒരു പേരിൽ ഒരേ ഇനത്തിലുള്ള ഒരു കൂട്ടം വിലകൾ ശേഖരിക്കുന്നതിനായി അറെകൾ ഉപയോഗിക്കുന്നു. ഒരു അറെയിലെ ഓരോ അംഗങ്ങളേയും അതിന്റേതായ സൂചിക വ്യക്തമാക്കിക്കൊണ്ട് ഉപയോഗിക്കുവാൻ സാധിക്കും.

എന്തുകൊണ്ടാണ് പ്രോഗ്രാമുകളിൽ അറേ ആവശ്യമായിവരുന്നത്. ഒരു ഉദാഹരണത്തിന്റെ സഹായത്തോടെ ഇത് നമുക്ക് പരിശോധിക്കാം. ഒരു ക്ലാസിലെ 20 വിദ്യാർത്ഥികളുടെ മാർക്കുകളുടെ ശരാശരിയെ കണ്ടെത്തണം എന്ന് കരുതുക. ഈ സാഹചര്യത്തിൽ സാധാരണ വേരിയബിളുകൾ ഉപയോഗിച്ചാൽ 20 വിദ്യാർത്ഥികളുടെ മാർക്കുകൾ ശേഖരിക്കുവാൻ 20 വേരിയബിളുകൾ ആവശ്യമായി വരും.

```
int  a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t;
float avg;
cin>>a>>b>>c>>d>>e>>f>>g>>h>>i>>j>>k>>l>>m>>n>>o>>p>>q>>r>>s>>t;
avg = (a+b+c+d+e+f+g+h+i+j+k+l+m+n+o+p+q+r+s+t)/20.0;
```

ഒരു പരിധിവരെ മുകളിൽ കൊടുത്തിരിക്കുന്ന കോഡ് ഉപയോഗിച്ച് 20 കുട്ടികളുടെ മാർക്കുകളുടെ ശരാശരി കണ്ടുപിടിക്കുവാൻ കഴിയും. എന്നാൽ 1000 കുട്ടികളുടെ ശരാശരി മാർക്ക് കണ്ടുപിടിക്കേണ്ട ഒരു സാഹചര്യം ഉണ്ടായാൽ ഈ രീതിയിലുള്ള പ്രവർത്തനം സാധ്യമല്ല. അതായത് ഒരു പ്രോഗ്രാമിൽ 1000 വേരിയബിളുകൾ ഉപയോഗിക്കുന്നതും അവ ഉപയോഗിച്ച് പ്രോഗ്രാം ചെയ്യുന്നതും എളുപ്പമുള്ള കാര്യമല്ല, മാത്രമല്ല ഇങ്ങനെ നിർമിക്കുന്ന പ്രോഗ്രാം വളരെ സങ്കീർണവും മനസ്സിലാക്കുന്നതിന് ബുദ്ധിമുട്ടുള്ളതും ആയിരിക്കും. ഇത്തരം സാഹചര്യങ്ങളിൽ അറേ എന്ന ആശയം നമുക്ക് ഉപകരിക്കും. അറെയിലെ ഓരോ അംഗങ്ങൾക്കും മെമ്മറി സ്ഥാനങ്ങൾ അനുവദിക്കേണ്ടതുണ്ട്. മെമ്മറി നീക്കിവെയ്ക്കുന്നതിന് പ്രഖ്യാപന പ്രസ്താവനകൾ ആവശ്യമാണെന്നും നമുക്കറിയാം. എങ്ങനെയാണ് അറേകൾ പ്രഖ്യാപനം നടത്തി അവ ഉപയോഗിക്കുന്നത് എന്ന് നമുക്ക് നോക്കാം.

2.1.1 അറേകളുടെ പ്രഖ്യാപനം (Array Declaration)

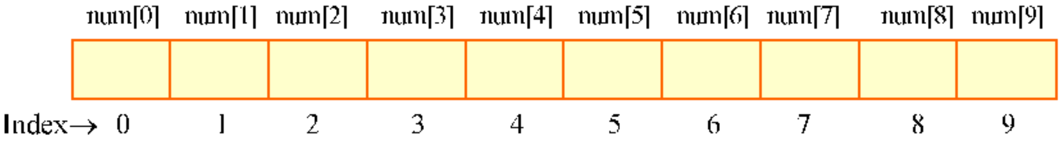
സാധാരണ വേരിയബിളിനെ പോലെ അറെ ഉപയോഗിക്കുന്നതിന് മുമ്പായി പ്രഖ്യാപനം നടത്തേണ്ടതുണ്ട്. C++ൽ അറെ പ്രഖ്യാപനം ചെയ്യുന്നതിനുള്ള വാക്യഘടന താഴെ പറഞ്ഞിരിക്കുന്നു.

```
datatype  array_name[size];
```

വാക്യഘടനയിൽ datatype എന്നത് അറെയിലെ അംഗങ്ങളുടെ ഡേറ്റയുടെ ഇനമാണ് സൂചിപ്പിക്കുന്നത്. array_name എന്നത് അറെയുടെ പേരും size എന്നത് അറെയിലെ ആകെ അംഗങ്ങളുടെ എണ്ണം വ്യക്തമാക്കുന്ന ഒരു പോസിറ്റീവ് സംഖ്യയും ആകുന്നു. താഴെ പറയുന്നത് ഒരു അറെ നിർമാണത്തിന്റെ ഉദാഹരണമാണ്.

```
int num[10];
```

മുകളിലുള്ള പ്രസ്താവന num എന്ന് വിളിക്കുന്ന 10 പൂർണ്ണസംഖ്യകൾ സൂക്ഷിക്കാവുന്ന ഒരു അറെയെ നിർമിക്കുന്നു. ചിത്രം 8.1 കാണിച്ചിരിക്കുന്നതു പോലെ അറെയിലെ അംഗങ്ങൾ മെമ്മറിയിൽ തുടർച്ചയായി സൂക്ഷിക്കുന്നു.



ചിത്രം 2.1 ഒരു അറെയിലെ അംഗങ്ങളുടെ ക്രമീകരണം

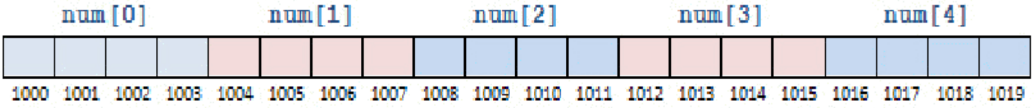
അറെയിലെ അംഗങ്ങൾ ക്രമാനുഗതമായി സൂക്ഷിക്കുന്നതുകൊണ്ട്, ഏത് അംഗത്തേയും അറെയുടെ പേരും അംഗത്തിന്റെ സ്ഥാനവും നൽകി ഉപയോഗിക്കുവാൻ കഴിയും. ഓരോ അംഗത്തെയും സൂചിപ്പിക്കുന്ന സ്ഥാനത്തിന് സൂചിക (index or subscript) എന്നു പറയുന്നു. C++ൽ അറെയുടെ സൂചിക പൂജ്യത്തിൽ ആരംഭിക്കുന്നു. `int num[10]` എന്ന് ഒരു അറെ നിർമ്മിച്ചാൽ അതിൽ സാധ്യമായ സൂചിക വിലകൾ 0 മുതൽ 9 വരെയാകും. ഈ അറെയിലെ ഒന്നാമത്തെ അംഗം `num [0]` ഉം അവസാനത്തെ അംഗം `num [9]` ഉം ആകുന്നു. `num [0]` എന്നത് 'നം ഓഫ് സീറോ' എന്ന് വായിക്കുന്നു. ആയിരം വിദ്യാർഥികളുടെ മാർക്കുകൾ സംഭരിക്കുന്ന പ്രശ്നം താഴെപ്പറയുന്ന പ്രസ്താവന ഉപയോഗിച്ച് പരിഹരിക്കാനാകും.

```
int score[1000];
```

`score` എന്നു പേരുള്ള അറെയിൽ 1000 വിദ്യാർഥികളുടെ മാർക്കുകൾ സംഭരിക്കാം. ആദ്യ വിദ്യാർത്ഥിയുടെ മാർക്ക് `score[0]` ലും അവസാനത്തെ വിദ്യാർത്ഥിയുടെ മാർക്ക് `score[999]` ലും സംഭരിക്കും.

2.1.2 അറെയുടെ മെമ്മറി നീക്കിവെയ്ക്കൽ (Memory Allocation for Arrays)

ഒരു അറെയിൽ അംഗങ്ങളെ സംഭരിക്കുന്നതിന് ആവശ്യമായ മെമ്മറിയുടെ അളവ് അതിന്റെ ഇനവും അംഗങ്ങളുടെ എണ്ണവുമായി ബന്ധപ്പെട്ടിരിക്കുന്നു. ചിത്രം 8.2ൽ `num` എന്ന ഒരു അറെയുടെ മെമ്മറി നീക്കിവെയ്ക്കൽ കാണിച്ചിരിക്കുന്നു, ഇതിൽ ആദ്യ അംഗത്തിന്റെ വിലാസമായി 1000 എന്ന് കാണിച്ചിരിക്കുന്നു. `num` ഒരു പൂർണ്ണസംഖ്യകളുടെ അറെ ആയതിനാൽ, ഓരോ അംഗത്തിന്റെയും വ്യാപ്തി 4 ബൈറ്റുകൾ ആണ് (16 ബിറ്റ് പ്രതിനിധീകരിക്കുന്ന ഒരു സിസ്റ്റത്തിൽ). താഴെക്കൊടുത്തിരിക്കുന്ന ചിത്രം 8.2ൽ `num[0]`ന്റെ വിലാസം `1000`, `num[1]`ന്റെ വിലാസം `1004`, അവസാന അംഗമായ `num[4]` വിലാസം വിലാസം `1016` എന്നിങ്ങനെ ആയിരിക്കും.



ചിത്രം 2.2 ഒരു പൂർണ്ണ സംഖ്യ അറെയുടെ മെമ്മറി അലോക്കേഷൻ

ഒരു ഏകമാന അറെക്ക് (single dimensional array) ആവശ്യമായ മെമ്മറിയുടെ അളവ് താഴെ പറയുന്ന സൂത്രവാക്യം ഉപയോഗിച്ച് കണ്ടുപിടിക്കാം.

ആകെ ബൈറ്റുകൾ = size_of (അറെയുടെ ഇനം) × അറെയിലെ അംഗങ്ങളുടെ എണ്ണം

ഉദാഹരണത്തിന്, `int num [10];` `num` അറെക്കായി നീക്കിവെച്ചിട്ടുള്ള ആകെ ബൈറ്റുകൾ $4 \times 10 = 40$ ബൈറ്റുകൾ ആയിരിക്കും.

2.1.3 അറെയുടെ പ്രാരംഭ വില നൽകൽ (Array Initialization)

സാധാരണ വേരിയബിൾ പോലെ തന്നെ അറെയുടെ പ്രഖ്യാപന പ്രസ്താവനകളോടൊപ്പം അവയുടെ പ്രാരംഭ വിലകൾ നൽകുവാൻ കഴിയും. താഴെപ്പറയുന്ന ഉദാഹരണങ്ങളിൽ കാണിച്ചിരിക്കുന്നതുപോലെ അറെയിലെ അംഗങ്ങളെ ബ്രാക്കറ്റിനുള്ളിൽ എഴുതണം.

```
int score[5] = {98, 87, 92, 79, 85};
char code[6] = {'s', 'a', 'm', 'p', 'l', 'e'};
float wgpa[7] = {9.60, 6.43, 8.50, 8.65, 5.89, 7.56, 8.22};
```

അറെയിലെ അംഗങ്ങളെ അവ എഴുതപ്പെട്ട ക്രമത്തിൽ സൂക്ഷിക്കുന്നു. ഒന്നാമത്തെ അംഗം സൂചിക 0ലും, രണ്ടാമത്തെ അംഗം സൂചിക 1ലും പ്രാരംഭ വിലകളായി സൂക്ഷിക്കുന്നു. ആദ്യത്തെ ഉദാഹരണത്തിൽ, score[0] ലേക്ക് 98, score[1] ലേക്ക് 87, score[2] ലേക്ക് 92, score[3] ലേക്ക് 79, score[4] ലേക്ക് 85 ഉം പ്രാരംഭ വിലകളായി സൂക്ഷിക്കുന്നു. ഒരു അറെയ്ക്ക് അനുവദിക്കപ്പെട്ട അംഗങ്ങളുടെ എണ്ണത്തെക്കാൾ പ്രാരംഭ മൂല്യങ്ങളുടെ എണ്ണം കുറവാണെങ്കിൽ, ആദ്യ സ്ഥാനങ്ങളിൽ അംഗങ്ങൾ സംഭരിക്കും, ശേഷിക്കുന്ന സ്ഥാനങ്ങൾ സംഖ്യാ ഡാറ്റകളുടെ കാര്യത്തിൽ പൂജ്യവും അക്ഷരഡാറ്റ കളുടെ കാര്യത്തിൽ ' ' (സ്പെയിസും) സംഭരിക്കും. ഒരു അറെയിലെ അംഗങ്ങളുടെ പ്രാരംഭ വിലകൾ നൽകുമ്പോൾ അംഗങ്ങളുടെ എണ്ണം ഒഴിവാക്കാവുന്നതാണ്. ഉദാഹരണത്തിന്, താഴെ പറയുന്ന പ്രാരംഭ വില നൽകൽ പ്രസ്താവന അഞ്ച് അംഗങ്ങളുള്ള ഒരു അറെ നിർമ്മിക്കുന്നു.

```
int num[] = {16, 12, 10, 14, 11};
```

2.1.4 അറെയിലെ അംഗങ്ങളെ ഉപയോഗിക്കൽ (Accessing elements of arrays)

ഒരു അറെയിലെ അംഗങ്ങളെ പ്രോഗ്രാമിൽ എവിടെയും ഉപയോഗിക്കാം. ഒരു സമയം ഒരു അംഗത്തിനെ മാത്രമേ ഉപയോഗിക്കാൻ കഴിയൂ. ഓരോ അംഗത്തേയും അറെയുടെ പേരും അവയുടെ സൂചികയും നല്കി ഉപയോഗിക്കുന്നു. score എന്ന അറെയിലെ അംഗങ്ങളെ ഉപയോഗിക്കുന്ന ചില ഉദാഹരണങ്ങൾ താഴെ കൊടുത്തിരിക്കുന്നു.

```
score[0] = 95;
score[1] = score[0] - 11;
cin >> score[2];
score[3] = 79;
cout << score[2];
sum = score[0] + score[1] + score[2] + score[3] + score[4];
```

ബ്രാക്കറ്റിനുള്ളിലെ സൂചിക ഒരു വേരിയബിളോ, ഒരു പൂർണ്ണസംഖ്യയോ, പൂർണ്ണസംഖ്യ നിർദ്ധാരണം ചെയ്യുന്ന ഒരു പ്രസ്താവനയോ ആകാം. ഓരോ സന്ദർഭത്തിലും പ്രസ്താവനയുടെ മൂല്യം അറെയുടെ സൂചികയുടെ സാധുവായ പരിധിക്കുള്ളിൽ ആയിരിക്കണം. ഈ രീതിയിൽ വേരിയബിളോ പ്രസ്താവനയോ ഉപയോഗിക്കുന്നതിന്റെ ഗുണം, അറെയിലുള്ള അംഗങ്ങളെ ഉപയോഗിക്കുന്നതിന് വേണ്ടി ലൂപ്പിന്റെ നിയന്ത്രണ വേരിയബിള് ഉപയോഗിക്കാം എന്നുള്ളതാണ്. ഇത് പ്രസ്താവനകളെ താഴെപ്പറയുന്ന രീതിയിൽ അനുചിതമായി ഉപയോഗിക്കുന്നതിൽ നിന്നും നമ്മെ പിന്തിരിപ്പിക്കുന്നു.

```
sum = score[0] + score[1] + score[2] + score[3] + score[4];
```

മുകളിലുള്ള പ്രസ്താവനയിലെ സൂചികയുടെ മൂല്യങ്ങൾക്കു പകരം ലൂപ്പിന്റെ നിയന്ത്രണ വേരിയബിൾ ഉപയോഗിച്ചുകൊണ്ട് അറെയിലെ അംഗങ്ങളെ ഉപയോഗിക്കാം. താഴെപ്പറയുന്ന പ്രസ്താവനകൾ ഈ ആശയം വിശദമാക്കുന്നു.

```
sum = 0;
for (i=0; i<5; i++)
sum = sum + score[i];
```

താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ ഒരു ഇൻപുട്ട് പ്രസ്താവന ഉപയോഗിച്ചുകൊണ്ടും അറെയിലെ അംഗത്തിന് മൂല്യം നൽകാം.

```
for(int i=0; i<5; i++)
    cin>>score[i];
```

ഈ ലൂപ്പ് പ്രവർത്തിച്ചു കഴിയുമ്പോൾ ആദ്യം സ്വീകരിക്കുന്ന വില അറെയുടെ ഒന്നാമത്തെ അംഗമായ score [0] ലും, രണ്ടാമത്തെ വില score [1] ലും, അവസാന വില score[4] ലും സൂക്ഷിക്കുന്നു.

പ്രോഗ്രാം 8.1 ഒരു അറെയിൽ എങ്ങനെ അഞ്ച് വിലകൾ സ്വീകരിക്കാമെന്നും അവയെ വിപരീത ക്രമത്തിൽ പ്രദർശിപ്പിക്കാമെന്നും കാണിക്കുന്നു. ഈ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തിയിട്ടുള്ള രണ്ട് ലൂപ്പുകളിൽ ആദ്യത്തേത് അറെയുടെ അംഗങ്ങളുടെ വിലകൾ സ്വീകരിക്കുന്നു. അഞ്ച് വിലകൾ സ്വീകരിച്ച് കഴിഞ്ഞാൽ രണ്ടാമത്തെ ലൂപ്പ് സംഭരിച്ച വിലകളെ അവസാനം മുതൽ ആദ്യം വരെ പ്രദർശിപ്പിക്കുന്നു.

പ്രോഗ്രാം 2.1: 5 കുട്ടികളുടെ സ്കോറുകൾ ഇൻപുട്ട് ചെയ്ത്, അവയെ നേർവിപരീത ക്രമത്തിൽ പ്രദർശിപ്പിക്കുക.

```
#include <iostream>
using namespace std;
int main()
{
  int i, score[5];
  for(i=0; i<5; i++)  // Reads the scores
  {
    cout<<"Enter a score: ";
    cin>>score[i];
  }
  for(i=4; i>=0; i--)  // Prints the scores
    cout<<"score[" << i << "] is " << score[i]<<endl;
  return 0;
}
```

ഔട്ട്പുട്ടിന്റെ മാതൃക:

```
Enter a score: 55
Enter a score: 80
Enter a score: 78
Enter a score: 75
```

```
Enter a score: 92
score[4] is 92
score[3] is 75
score[2] is 78
score[1] is 80
score[0] is 55
```

നമുക്ക് ചെയ്യാം.

1. താഴെ പറയുന്നവ സംഭരിക്കുന്നതിനുള്ള അറെ പ്രഖ്യാപന പ്രസ്താവനകൾ എഴുതുക
   i. 100 വിദ്യാർത്ഥികളുടെ മാർക്ക്
   ii. ഇംഗ്ലീഷ് അക്ഷരമാല
   iii. 10 വർഷങ്ങളുടെ പട്ടിക
   iv. 30 ദശാംശ സംഖ്യകളുടെ പട്ടിക
2. താഴെ പറയുന്ന അറെയിൽ പ്രാരംഭ വിലകൾ നല്കുന്നതിനുള്ള പ്രസ്താവനകൾ എഴുതുക
   i. 10 സ്കോറുകളുടെ പട്ടിക 89, 75, 82, 93, 78, 95, 81, 88, 77, 82
   ii. അഞ്ച് അളവുകളുടെ പട്ടിക: 10.62, 13.98, 18.45, 12.68, 14.76 എന്നിവ
   iii. 100 പലിശ നിരക്കുകളുടെ പട്ടിക, ആദ്യ ആറ് പലിശ നിരക്കുകൾ 6.29, 6.95, 7.25, 7.35, 7.40, 7.42.
   iv. മൂല്യം 0 ഉപയോഗിച്ച് 10 മാർക്കിനുള്ള ഒരു അറെ.
   v. VIBGYOR അക്ഷരങ്ങളുള്ള ഒരു അറെ.
   vi. ഓരോ മാസത്തിലുമുള്ള ദിവസങ്ങളുള്ള ഒരു അറെ.
3. `int ar[50];` എന്ന അറെയിലേക്ക് വിലകൾ ഇൻപുട്ട് ചെയ്യുന്നതിനുള്ള C++ കോഡ് ശകലങ്ങൾ എഴുതുക.
4. `float val [100]; val` അറെയുടെ ഇരട്ട സ്ഥാനങ്ങളിലുള്ള അംഗങ്ങൾ പ്രദർശിപ്പിക്കുന്നതിന് C++ കോഡ് ശകലം എഴുതുക:

കുറഞ്ഞത് ഒരിക്കലെങ്കിലും അറെയിലെ ഓരോ അംഗത്തേയും ഉപയോഗിക്കുക എന്നതാണ് കടന്നുപോകൽ എന്നതുകൊണ്ട് ഉദ്ദേശിക്കുന്നത്. അറെയിലെ എല്ലാ അംഗങ്ങളും പ്രദർശിപ്പിക്കുന്നത് കടന്നുപോകൽ ഒരു ഉദാഹരണമാണ്. ഏതെങ്കിലുമൊരു പ്രവർത്തനം അറെയിലെ എല്ലാ അംഗങ്ങളിലും നടക്കുന്നു എങ്കിൽ അതിനെ കടന്നുപോകൽ എന്നു പറയുന്നു. ഒരു പ്രോഗ്രാമിൽ എങ്ങനെയാണ് കടന്നുപോകൽ നടത്തുന്നത് എന്നത് പ്രോഗ്രാം 2.2 ൽ കൊടുത്തിരിക്കുന്നു

പ്രോഗ്രാം 2.2: അറെയിലെ കടന്നുപോകൽ

```
#include <iostream>
using namespace std;
int main()
{
int a[5], i;
cout<<"Enter the elements of the array :";
for(i=0; i<5; i++)
     cin >> a[i];
for(i=0; i<5; i++)
     a[i] = a[i] + 1;
cout<<"\nNow value of the elements in the array are...\n";
for(i=0; i<5; i++)
    cout<< a[i]<< "\t";
return 0;
}
```

കടന്നുപോകൽ

കടന്നുപോകൽ

കടന്നുപോകൽ

ഔട്ട്പുട്ടിന്റെ മാതൃക:

അറെ അംഗങ്ങളെ ചേർക്കുക: 12 3 6 1 8
ഇപ്പോൾ അറെയിലെ ഓരോ അംഗത്തിന്റെയും വില: 1 3 6 8 12

പ്രോഗ്രാം 2.3: അറെ അംഗങ്ങളുടെ തുക കണ്ടെത്തുക

```
#include <iostream>
using namespace std;
int main()
{   int a[5], i, sum;
    cout<<"Enter the elements of the array:";
    for (i=0; i<5; i++)
        cin>>a[i]; // Reading the elements
    sum = 0
    for (i=0; i<5; i++)
        sum = sum + a[i]; // a case of traversal
    cout<<"\nsum of the elements of the array is "<<sum;
    return 0;
}
```

ഔട്ട്പുട്ടിന്റെ മാതൃക:

അറെ അംഗങ്ങളെ ചേർക്കുക: 12 3 6 1 8
അറെ അംഗങ്ങളുടെ തുക: 30

പ്രോഗ്രാം 2.4: അറെയിലെ ഏറ്റവും വലിയ അംഗത്തെ കണ്ടെത്തുക

```
#include <iostream>
using namespace std;
int main()
{
    int a[5],i, big;
    cout<<"Enter the elements of the array:";
    for(i=0; i<5; i++)
        cin>>a[i];
    big = a[0]
    for(i=1; i<5; i++)
            if(a[i] > big)// a case of traversal
                big = a [i]
    cout<<"nThe biggest element is "<< big;
    return 0;
}
```

ഔട്ട്പുട്ടിന്റെ മാതൃക:

അറെ അംഗങ്ങളെ ചേർക്കുക: 23 10 -3 7 11

അറെയിലെ ഏറ്റവും വലിയ അംഗം: 12

2.2. അറെ ഉപയോഗിച്ചുള്ള സ്ട്രിങ് കൈകാര്യം ചെയ്യൽ (String handling using arrays)

C++ ലെ ഒരുതരം ലിറ്ററലാണ് സ്ട്രിങ്. പ്രോഗ്രാമുകളിൽ ഇവ കാണപ്പെടുന്നത് ഉദ്ധരണിക്കുള്ളിൽ (Double quotes) തുടർച്ചയായുള്ള കാരക്റ്ററുകളായാണ്. നിങ്ങളോട് പേര് ശേഖരിക്കുവാനും പ്രദർശിപ്പിക്കുന്നതിനുമുള്ള ഒരു പ്രോഗ്രാം എഴുതുവാൻ ആവശ്യപ്പെട്ടുവെന്നിരിക്കട്ടെ. ഡാറ്റ ശേഖരിക്കുവാൻ വേരിയബിൾ ആവശ്യമാണെന്ന് ഇതിനു മുമ്പ് നാം പഠിച്ചിട്ടുണ്ട്. my_name എന്ന വേരിയബിൾ ഒരു ഐഡന്റിഫയർ ആയി ഇവിടെ നമുക്ക് ഉപയോഗിക്കാം. ഒരു വേരിയബിൾ ഉപയോഗിക്കുന്നതിനു മുമ്പ് അത് പ്രഖ്യാപിക്കണമെന്നുള്ളത് ഈ അവസരത്തിൽ തീർച്ചയായും ഓർമിക്കേണ്ടതാണ്. സ്ട്രിങ് ഡാറ്റയെ സൂചിപ്പിക്കാനുള്ള അടിസ്ഥാന ഡാറ്റ ഇനം നിലവിലില്ലാത്തതിനാൽ ഏതു തരം ഡാറ്റയാണ് സ്ട്രിങ് ഡാറ്റ ശേഖരിക്കുന്ന വേരിയബിൾ പ്രഖ്യാപനത്തിന് ഉപയോഗിക്കാനാവുക എന്ന് പറയാൻ സാധിക്കില്ല? അതുകൊണ്ട് നമുക്ക് char ഡാറ്റ ഇനത്തെക്കുക്കുറിച്ച് ആലോചിക്കാം. എന്നാൽ അവിടെയും ഒരു പ്രശ്നമുണ്ട്. char ഡാറ്റ ഇനത്തിന് ഒരു കാരക്റ്റർ മാത്രമേ ശേഖരിക്കുവാൻ കഴിയുകയുള്ളൂ. അതുകൊണ്ടുതന്നെയാണ് സ്ട്രിങ് എന്നത് തുടർച്ചയായ കാരക്റ്ററുകളുടെ ഇൻപുട്ട് ആയി സ്വീകരിക്കേണ്ടി വരുന്നത് .

"Niketh" എന്ന പേര് പരിഗണിക്കുക. ഇത് ആറ് കാരക്റ്ററുകൾ ഉൾക്കൊള്ളുന്ന ഒരു സ്ട്രിങ് ആണ്. എന്നാൽ ഒരു കാരക്റ്റർ അറെയ്ക്ക് ഒന്നിലധികം കാരക്റ്ററുകളെ ശേഖരിക്കുവാൻ കഴിയുമെന്ന് നമുക്കറിയാം. അതുകൊണ്ടു ഒരു അറേയെ താഴെ കാണുന്നവിധം പ്രഖ്യാപിക്കാവുന്നതാണ്.

```
char my_name[10];
```

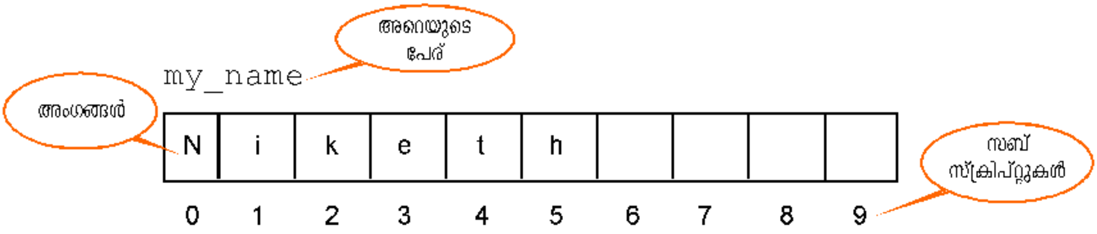
my_name എന്ന് പേരുള്ള അറേയിൽ ഒരു ബൈറ്റ് വീതം വലിപ്പമുള്ള തുടർച്ചയായ പത്ത് മെമ്മറി സ്ഥാനങ്ങൾ നീക്കിവച്ചിട്ടുണ്ട്. ഈ അറേയിലേക്ക് താഴെ കാണുന്നത് പോലെ പ്രാരംഭ വിലകൾ നൽകാവുന്നതാണ്.

```
char my_name[10] = { 'N','i','k','e','t','h'};
```

ചിത്രം 2.1ൽ മേൽ സൂചിപ്പിച്ച കാരക്റ്റർ അറേയുടെ മെമ്മറി നീക്കിവെയ്പ് ചിത്രീകരിച്ചിട്ടുണ്ട്. സ്ട്രിങ്ങിലെ കാരക്റ്ററുകൾ കോമായുപയോഗിച്ച് വേർതിരിച്ചാണ് ശേഖരിക്കുന്നത് എന്ന് പ്രത്യേകം ശ്രദ്ധിക്കേണ്ടതാണ്. ഇതേ ഡാറ്റ ഇൻപുട്ട് ചെയ്യണമെങ്കിൽ താഴെ പറഞ്ഞിരിക്കുന്ന C++ പ്രസ്താവന ഉപയോഗിക്കാം .

```
for  (int i=0;i<6;i++)
          cin>>my_name[i];
```

ഈ കോഡ് പ്രവർത്തിക്കുന്ന സമയത്ത് നാം "Niketh" എന്ന സ്ട്രിങ്ങിനകത്തെ ആറ് കാരക്റ്ററുകൾ ഒന്നിന് പുറകെ ഒന്നായി സ്പേസ് ബാർ, ടാബ് കീ അല്ലെങ്കിൽ എന്റർ കീ എന്നിവയിൽ ഏതെങ്കിലും ഒന്നുപയോഗിച്ച് വേർതിരിച്ച് വേണം ഇൻപുട്ട് ചെയ്യേണ്ടത്. മേൽ സൂചിപ്പിച്ച രണ്ടു രീതിയിലുമുള്ള മെമ്മറി നീക്കിവയ്ക്കലുകൾ താഴെ തന്നിരിക്കുന്ന വിധത്തിലാണ്.



ചിത്രം 2.3: കാരക്റ്റർ അറെയുടെ മെമ്മറി നീക്കിവെയ്പ്

സ്ട്രിങ്ങുകൾ തുടർച്ചയായുള്ള കാരക്റ്ററുകൾ ആയതിനാൽ കാരക്റ്റർ അറേയെ സ്ട്രിങ്ങുകൾ ശേഖരിക്കുന്നതിന് ഉപയോഗിക്കാവുന്നതാണ്. എന്നിരുന്നാലും ഒരു സ്ട്രിങ് നേരിട്ട് ഇൻപുട്ട് ചെയ്യുന്നതായി നമുക്ക് തോന്നുകയേ ഇല്ല എന്നത് ഒരു വസ്തുതയാണ്. പകരം നാം ഒന്നിന് പുറകെ ഒന്നായി കാരക്റ്ററുകൾ ഇൻപുട്ട് ചെയ്ത് അതിനെ ഒരു സ്ട്രിങ് ആക്കി മാറ്റുകയാണ് ചെയ്യേണ്ടത്.

C++ ൽ കാരക്റ്റർ അറേകൾക്ക് ചില പ്രത്യേക സവിശേഷതകൾ ഉണ്ട്. ഒരിക്കൽ ഒരു കാരക്റ്റർ അറെ പ്രഖ്യാപിച്ചാൽ, അറെയുടെ പേര് സ്ട്രിങ് ഡാറ്റ സൂക്ഷിക്കാനുള്ള സാധാരണ വേരിയബിളായിത്തന്നെ പരിഗണിക്കപ്പെടുന്നു. അതുകൊണ്ടു തന്നെ കാരക്റ്റർ അറേയുടെ പേര് സ്ട്രിങ് വേരിയബിളിന് സമാനമാണ് എന്ന് പറയാം. അതിനാൽ നിങ്ങളുടെ പേര് my_name (അറെയുടെ പേര്) ൽ താഴെ കൊടുത്തിട്ടുള്ള പ്രസ്താവന ഉപയോഗിച്ച് സംഭരിക്കാവുന്നതാണ്.

```
cin>>my_name;
```

മറ്റുള്ള ഡാറ്റ ഇനങ്ങളുടെ കാര്യത്തിൽ മേൽ സൂചിപ്പിക്കപ്പെട്ട പ്രയോഗം തെറ്റാണെന്ന് പ്രത്യേകം ശ്രദ്ധിക്കേണ്ടതാണ്. ഇനി നമുക്ക് ഒരു സ്ട്രിങ് ഇൻപുട്ട് ചെയ്തു പ്രദർശിപ്പിക്കുന്നതിനുള്ള പ്രോഗ്രാം പൂർത്തിയാക്കാം. പ്രോഗ്രാം 2.5 ൽ പറഞ്ഞിരിക്കുന്നത് പോലെ ഇത് ചെയ്യാവുന്നതാണ്.

പ്രോഗ്രാം 2.5 ഒരു സ്ട്രിങ് ഇൻപുട്ട് ചെയ്ത് പ്രദർശിപ്പിക്കുക.

```
    #include<iostream>
using namespace std;
int main()
{
    char my_name[10];
    cout << "Enter your name: ";
    cin >> my_name;
    cout << "Hello " << my_name;
}
```

ഈ പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുമ്പോൾ താഴെ കാണുന്നവിധം ഔട്ട്പുട്ട് ലഭിക്കുന്നതാണ്.

```
    Enter your name: Niketh
    Hello Niketh
```

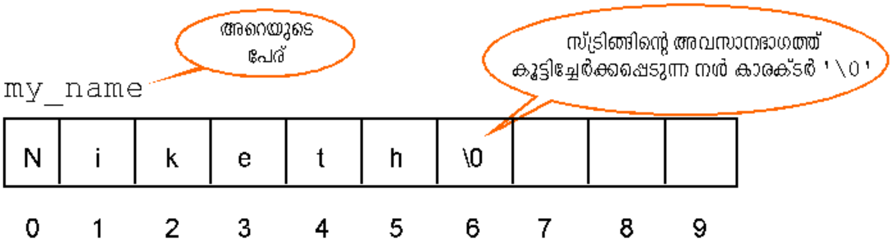
പ്രത്യേകം ശ്രദ്ധിക്കേണ്ടത് ഇവിടെ സ്ട്രിങ് കോൺസ്റ്റന്റ് "Hello" അല്ല "Hello " ആണ് എന്നുള്ളത്. ('o' എന്ന അക്ഷരത്തിനു ശേഷം ഒരു സ്പേസ് നൽകിയിട്ടുണ്ട്).

നമുക്കു ചെയ്യാം

പ്രോഗ്രാം 2.5 പ്രവർത്തിപ്പിച്ച് നിങ്ങളുടെ പേരിന്റെ കൂടെ ഇനിഷ്യലും ഇൻപുട്ട് ചെയ്ത് ഔട്ട്പുട്ട് ശരിയോ തെറ്റോ എന്ന് പരിശോധിക്കുക. പേരിൽ 10 കാരക്റ്ററുകളിലും കൂടുതൽ ഉണ്ടെങ്കിൽ അറെയുടെ വലിപ്പം ആവശ്യത്തിനനുസരിച്ച് വർദ്ധിപ്പിക്കുക .

2.3 സ്ട്രിങ്ങിനു വേണ്ടിയുള്ള മെമ്മറി നീക്കിവെയ്പ് (Memory allocation for strings)

ഒരു അറെയിലുള്ള കാരക്റ്ററുകൾക്ക് എങ്ങനെയാണ് മെമ്മറി അനുവദിക്കുന്നതെന്നു നാം കണ്ടു കഴിഞ്ഞു. ചിത്രം 2.1 ൽ കാണിച്ചിരിക്കുന്നത് പോലെ മെമ്മറി ആവശ്യകത കണക്കാക്കുന്നത് ഇൻപുട്ട് ചെയ്ത കാരക്റ്ററുകളുടെ എണ്ണമനുസരിച്ചാണ്. എന്നാൽ ഒരു കാരക്റ്റർ അറെയിൽ സ്ട്രിങ് ഇൻപുട്ട് ചെയ്യുമ്പോൾ ചിത്രം മറ്റൊന്നാകുന്നു. നമ്മൾ പ്രോഗ്രാം 2.1 പ്രവർത്തിപ്പിച്ച് "Niketh" എന്ന സ്ട്രിങ് ഇൻപുട്ട് ചെയ്താൽ മെമ്മറി നീക്കി വെയ്പ് ചിത്രം 2.5 ൽ താഴെ കാണുന്ന വിധമായിരിക്കും.



ചിത്രം 2.4 : കാരക്റ്റർ അറേയുടെ മെമ്മറി നീക്കിവെയ്പ്.

ഇവിടെ നൾ കാരക്റ്റർ ('\0') സ്ട്രിങ്ങിന്റെ അവസാനഭാഗത്ത് കൂട്ടിച്ചേർക്കപ്പെടുന്നു. ഇത് കാരക്റ്റർ സ്ട്രിങ്ങിന്റെ ടെർമിനേറ്റർ ആയി ഉപയോഗിക്കുന്നു. അതിനാൽ ഒരു സ്ട്രിങ് സംഭരിക്കാനാവശ്യമായ മെമ്മറി എന്നത് സ്ട്രിങ്ങിലെ ആകെ കാരക്റ്ററുകളുടെ എണ്ണവും നൾ കാരക്റ്ററിനു വേണ്ട ഒരു ബൈറ്റും ചേർന്നതാണ്. മേല്പറഞ്ഞ "Niketh" എന്ന സ്ട്രിങ് ശേഖരിക്കുവാൻ ഏഴ് ബൈറ്റ് ആവശ്യമാണ്. (അതായത് 6 കാരക്റ്ററുകൾക്കുള്ള 6 ബൈറ്റ് + നൾ കാരക്റ്ററിനുള്ള 1 ബൈറ്റ്).

താഴെ കാണിച്ചിരിക്കുന്നവിധത്തിൽ നമുക്ക് കാരക്റ്റർ അറെയ്ക്ക് പ്രാരംഭവില നൽകാം.

```
char my_name[10] = "Niketh";

char str[] ="Hello world";
```

ആദ്യത്തെ പ്രസ്താവനയിൽ പത്ത് മെമ്മറി സ്ഥാനങ്ങൾ നീക്കി വെക്കുകയും അതിൽ പ്രാരംഭ വിലയും നൾ കാരക്റ്ററും സംഭരിക്കുകയും ചെയ്യുന്നു. ഇവിടെ അവസാന മൂന്ന് ബൈറ്റുകൾ ഉപയോഗിക്കുന്നില്ല. എന്നാൽ രണ്ടാമത്തെ സ്റ്റേറ്റ്മെന്റിൽ അറെയുടെ വലിപ്പം ഉൾപ്പെടുത്തിയിട്ടില്ല. അതുകൊണ്ട് 11 ബൈറ്റ് സ്ട്രിങ്ങിനും 1 ബൈറ്റ് `'\0'` നും അടക്കം ആകെ 12 ബൈറ്റ് നീക്കിവെയ്ക്കപ്പെടുന്നു.

2.4 സ്ട്രിങ്ങിനു മേലുള്ള ഇൻപുട്ട്/ഔട്ട്പുട്ട് പ്രവർത്തനങ്ങൾ (Input/Output operations on strings)

പ്രോഗ്രാം 2.5 ൽ സ്ട്രിങ് ഡാറ്റ ഇൻപുട്ട്/ഔട്ട്പുട്ട് ചെയ്യുന്നതിനുള്ള പ്രസ്താവനകൾ ഉൾപെടുത്തിയിട്ടുണ്ട്. അറെയുടെ വലിപ്പം 20 ആക്കി പ്രഖ്യാപന പ്രസ്താവനയിൽ ഒരു ചെറിയ മാറ്റം വരുത്തുക. "Maya Mohan" എന്ന പേര് ഇൻപുട്ട് ചെയ്ത് പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുകയാണെങ്കിൽ താഴെ കാണുന്ന വിധത്തിലുള്ള ഔട്ട്പുട്ട് ലഭിക്കുന്നതാണ്.

```
Enter your name: Maya Mohan
Hello Maya
```

സ്ട്രിങ് ശേഖരിക്കുന്നതിന് ആവശ്യമായ വലിപ്പം അറെയ്ക് ഉണ്ടെങ്കിലും നമുക്ക് ഔട്ട്പുട്ടായി `"Maya"` എന്ന് മാത്രമാണ് ലഭിക്കുന്നത്. ഇതെന്തുകൊണ്ട് സംഭവിച്ചു?

നമുക്ക് `cin>>my_name;` എന്ന പ്രസ്താവന സൂക്ഷ്മമായൊന്നു പരിശോധിക്കാം. ഒരു ഡാറ്റ ഇനത്തെ മാത്രമേ ഈ പ്രസ്താവന ഉപയോഗിച്ചു ഇൻപുട്ട് ചെയ്യാൻ കഴിയൂ എന്ന് നമുക്കറിയാം. ഒരു ഡാറ്റയെ മറ്റൊന്നിൽ നിന്ന് വേർതിരിക്കുവാൻ ഉപയോഗിക്കുന്നതാണ് വൈറ്റ് സ്പേസ്. അതുകൊണ്ട് `"Maya Mohan"` എന്നത് രണ്ട് ഡാറ്റയായി പരിഗണിക്കപ്പെടുന്നു. (Maya, Mohan എന്നിവയ്ക്കിടയ്ക്ക് വൈറ്റ് സ്പേസ് ഉള്ളതുകൊണ്ട്). `my_name`ന് മുമ്പ് ഒരു ഇൻപുട്ട് ഓപ്പറേറ്റർ (>>) മാത്രമേയുള്ളൂ. അതിനാൽ ആദ്യത്തെ ഡാറ്റയായ `"Maya"` മാത്രം സംഭരിക്കപ്പെടുന്നു. അതിന് ശേഷമുള്ള വൈറ്റ് സ്പേസ് ഡിലിമിറ്റർ ആയി വർത്തിക്കുകയും ചെയ്യുന്നു. അതിനാൽ ഈ പ്രസ്താവന സംവിധാനം ഉപയോഗിച്ച് വൈറ്റ് സ്പേസ് അടങ്ങിയ സ്ട്രിങ്ങുകൾ മുഴുവനായും ഇൻപുട്ട് ചെയ്യുവാൻ കഴിയുകയില്ല. ഇതിനു പരിഹാരമായി `gets()` എന്ന ഫങ്ഷൻ ഉപയോഗിക്കാവുന്നതാണ്. സ്റ്റാൻഡേർഡ് ഇൻപുട്ട് ഉപകരണങ്ങളിൽ (keyboard) നിന്ന് വൈറ്റ് സ്പേസ് അടങ്ങിയ സ്ട്രിങ്ങുകളെ സ്വീകരിക്കുകയും അതിനെ ഒരു കാരക്റ്റർ അറെയിൽ സംഭരിക്കുന്നതിനുമുള്ള കൺസോൾ ഇൻപുട്ട് ഫങ്ഷനാണ് `gets()`. ഈ ഫങ്ഷനിലേക്ക് സ്ട്രിങ് വേരിയബിൾ (കാരക്റ്റർ അറെയുടെ പേര്) താഴെ കാണുന്നവിധത്തിൽ നൽകാവുന്നതാണ്.

```
gets(character_array_name);
```

ഈ ഫങ്ഷൻ ഉപയോഗിക്കുമ്പോൾ cstdio(stdio.h എന്നത് Turbo C++ൽ) എന്ന ലൈബ്രറി ഹെഡർ ഫയൽ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തേണ്ടതാണ്. പ്രോഗ്രാം 9.1 ൽ include <cstdio> ഉൾപ്പെടുത്തുകയും കൂടാതെ cin>>my_name; എന്ന പ്രസ്താവനയ്ക്ക് പകരം gets(my_name); ഉപയോഗിച്ച് പ്രോഗ്രാം വീണ്ടും പ്രവർത്തിപ്പിച്ചാൽ താഴെ കാണുന്ന ഔട്ട്പുട്ട് ലഭിക്കുന്നതാണ്.

```
Enter your name : Maya Mohan
Hello Maya Mohan
```

ഇപ്പോൾ നാം ഇൻപുട്ട് ചെയ്ത മുഴുവൻ സ്ട്രിങ്ങും ഔട്ട്പുട്ട് ആയി കാണപ്പെടുന്നുണ്ട്. ഇനി നമുക്ക് gets() ഫങ്ഷനും cin ഉം തമ്മിലുള്ള വ്യത്യാസം എന്താണെന്നു നോക്കാം.

സ്ട്രിങ്ങിന്റെ ഇൻപുട്ട്/ഔട്ട്പുട്ട് പ്രവർത്തനങ്ങളിൽ സബ്സ്ക്രിപ്റ്റഡ് വേരിയബിൾ എന്ന ആശയം ഉപയോഗിക്കുന്നില്ലെങ്കിലും, അറെയിലെ ഏതൊരു അംഗത്തെയും അറെയുടെ പേരും സബ്സ്ക്രിപ്റ്റും ഉപയോഗിച്ചു വേർതിരിച്ചുപയോഗിക്കാവുന്നതാണ്. സ്ട്രിങ്ങിലെ ആദ്യത്തെ കാരക്റ്ററിനെ ഉപയോഗിക്കണമെങ്കിൽ my_name[0] എന്നും, അഞ്ചാമത്തെ കാരക്റ്റർ എടുത്തുപയോഗിക്കണമെങ്കിൽ my_name[4] എന്നിങ്ങനെ എന്നും പ്രയോഗിക്കാവുന്നതാണ്. നൾ കാരക്റ്ററും ('\0') നമുക്ക് സബ്സ്ക്രിപ്റ്റ് ഉപയോഗിച്ചു തെരഞ്ഞെടുക്കാം. താഴെ കൊടുത്തിട്ടുള്ള പ്രോഗ്രാം ഈ ആശയം വ്യക്തമാക്കുന്നതാണ്.

പ്രോഗ്രാം 2.6 തന്നിരിക്കുന്ന സ്ട്രിങ്ങിലെ സ്വരാക്ഷരങ്ങളുടെ (Vowels) എണ്ണം കണ്ടുപിടിക്കുക

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
 {
  char str[20];
int vow=0;
  cout<<"Enter a string: ";
  gets(str);
  for(int i=0; str[i]!='\0'; i++)
     switch(str[i])
     {
          case 'a':
          case 'e':
          case 'i':
          case 'o':
          case 'u': vow++;
     }
  cout<<"No. of vowels in the string "<<str<<" is "<<vow;
  return 0;
 }
```

gets() ഫങ്ഷന് വേണ്ടിയുള്ള ഹെഡർ ഫയൽ

നൾ കാരക്ടർ എത്തുന്നതുവരെ തുടർന്നു കൊണ്ടിരിക്കുന്നു.

അറെയിലെ ഓരോ കാരക്ടറും കാരക്ടർ കോൺസ്റ്റന്റുമായി താരതമ്യം ചെയ്യുന്നു

"Hello guys" എന്ന സ്ട്രിങ് ഇൻപുട്ട് ചെയ്ത് പ്രോഗ്രാം 9.2 പ്രവർത്തിപ്പിക്കുകയാണെങ്കിൽ ചുവടെ കൊടുത്തിരിക്കുന്ന ഔട്ട്പുട്ട് കാണാവുന്നതാണ് .

```
Enter a string : Hello guys
No.of vowels in the string Hello guys is 3
```

ഈ പ്രോഗ്രാം പ്രവർത്തിച്ച് ഫലം ലഭ്യമാകുന്നത് എങ്ങനെയെന്ന് നമുക്ക് വിശകലനം ചെയ്യാം.

- തുടക്കത്തിൽ തന്നെ `gets()` ഫങ്ഷൻ ഉപയോഗിച്ച് `"Hello guys"`എന്ന സ്ട്രിങ് ഇൻപുട്ട് ചെയ്യുന്നു .
- 'i' എന്ന സബ്സ്ക്രിപ്റ്റ് ഉപയോഗിച്ചു സൂചിപ്പിക്കുന്ന അറെയിലെ ഓരോ കാരക്റ്ററും, നൾ കാരക്റ്റർ ('\0') അല്ലാത്തിടത്തോളം ഫോർ ലൂപ്പിന്റെ ചട്ടക്കൂട് തുടർച്ചയായി പ്രവർത്തിച്ചുകൊണ്ടിരിക്കുന്നു. അതായത് നൾ കാരക്റ്റർ എത്തുന്നതുവരെ ലൂപ്പിന്റെ ചട്ടക്കൂട് പ്രവർത്തിച്ചുകൊണ്ടിരിക്കും.
- ലൂപ്പ് ചട്ടക്കൂടിനകത്ത് ഒരേയൊരു സ്വിച്ച് പ്രസ്താവന (switch statement) മാത്രമേ ഉള്ളൂ. ആദ്യത്തെ നാലു കേസുകളിലും ഒരു പ്രസ്താവന പോലും നൽകിയിട്ടില്ല. അവസാനത്തെ കേസിന് vow എന്ന വേരിയബിളിന്റെ വില ഒന്ന് വർദ്ധിക്കുന്നു (vow++). എല്ലാ കേസുകൾക്കും ഇതാവശ്യമാണെന്നു ഒരു പക്ഷെ നിങ്ങൾ ചിന്തിക്കുന്നുണ്ടാവും. അത് തികച്ചും ശരിയാണ്. എന്നാൽ അങ്ങനെയാണെങ്കിൽ ഓരോ കേസിനും വെവ്വേറെ ബ്രേക്ക് പ്രസ്താവനകൾ ഉപയോഗിക്കേണ്ടതായി വരും. ഈ പ്രോഗ്രാമിൽ എല്ലാ കേസുകളുടെയും പ്രവർത്തനം ഒരേ പോലെയായതിനാലാണ് ഈ രീതിയിലുള്ള പ്രസ്താവന ഉപയോഗിച്ചിരിക്കുന്നത്.
- ഫോർ ലൂപ്പ് തുടർച്ചയായി പ്രവർത്തിക്കുമ്പോൾ ഓരോ കാരക്റ്ററും ഒന്നിന് പുറകെ ഒന്നായി ലഭ്യമാകുന്നു. അവയെ കേസിലെ ഓരോ കാരക്റ്റർ കോൺസ്റ്റന്റുമായി താരതമ്യം ചെയ്യുന്നു. ഏതെങ്കിലും ഒരു തവണ സമാനത കൈവരിച്ചാൽ vow എന്ന വേരിയബിളിന്റെ വില ഒന്ന് കൂടുന്നു (vow ++).
- നൽകിയിട്ടുള്ള ഇൻപുട്ട് സ്ട്രിങ്ങിന്റെ കാര്യത്തിൽ സമാനത കൈവരിക്കുന്നത് " i " യുടെ വില 1, 4, 7 എന്നിങ്ങനെ ആകുമ്പോഴാണ്. അതുകൊണ്ട് തന്നെ vow ന്റെ വില മൂന്നു തവണ ഓരോന്ന് വെച്ച് വർധിക്കുകയും നമുക്ക് ശരിയായ ഉത്തരം ലഭിക്കുകയും ചെയ്യുന്നു.

സ്ട്രിങ്ങുകൾ ഇൻപുട്ട് ചെയ്യുന്നതിനു `gets()` ഫങ്ഷൻ എങ്ങനെ ഉപയോഗിക്കുന്നുവെന്ന് നാം മനസ്സിലാക്കി. അതുപോലെ സ്ട്രിങ് ഔട്ട്പുട്ട് ചെയ്യുന്നതിന് C++ ൽ `puts()` എന്ന ഫങ്ഷൻ ലഭ്യമാണ്. സ്ട്രിങ് ഡാറ്റയെ സ്റ്റാൻഡേർഡ് ഔട്ട്പുട്ട് ഉപകരണ (മോണിറ്റർ) ത്തിൽ പ്രദർശിപ്പിക്കുവാൻ വേണ്ടിയുള്ള കൺസോൾ ഔട്ട്പുട്ട് ഫങ്ഷനാണ് `put()`. ഇതിന്റെ വാക്യഘടന (syntax) താഴെ കൊടുത്തിരിക്കുന്നു .

```
puts(string data);
```

ഈ ഫങ്ഷനിലേക്ക് സ്ട്രിങ് കോൺസ്റ്റന്റ് അഥവാ വേരിയബിൾ (കാരക്റ്റർ അറെയുടെ പേര്) ആണ് നൽകേണ്ടത്. താഴെ കാണുന്ന C++ കോഡ് നിരീക്ഷിക്കുക .

```
char str[10] ="friends";
puts("hello");
puts(str);
```

മേൽ സൂചിപ്പിച്ച കോഡിന്റെ ഔട്ട്പുട്ട് താഴെ കാണും വിധത്തിലാണ് .

```
hello
friends
```

കാരക്റ്റർ അറെ str[10] ലെ "friends" എന്ന സ്ട്രിങ് അടുത്ത ലൈനിലാണ് പ്രദർശിപ്പിച്ചിരിക്കുന്നത്. puts() ഫങ്ഷനുകൾക്ക് പകരം cout<<"hello";, cout<< str; എന്നീ പ്രസ്താവനകൾ ഉപയോഗിക്കുമ്പോഴുള്ള വ്യത്യാസം ശ്രദ്ധിക്കുക. cout ഉപയോഗിക്കുമ്പോൾ സ്ട്രിങ്ങുകൾക്കിടയിൽ ഒരു സ്പേസ് പോലും ഇല്ലാതെ ഔട്ട്പുട്ട് അതേ വരിയിൽ തന്നെ പ്രദർശിപ്പിക്കപ്പെടുന്നു.

നമുക്കു ചെയ്യാം

പ്രോഗ്രാം 2.6 ൽ "HELLO GUYS" എന്ന ഇൻപുട്ട് നൽകി ഔട്ട്പുട്ട് പ്രവചിക്കുക. പ്രോഗ്രാം പ്രവർത്തിപ്പിച്ചു ഈ ഇൻപുട്ടിന് ശരിയായ ഔട്ട്പുട്ട് ലഭിക്കുന്നുണ്ടോ എന്ന് പരിശോധിക്കുക. ഔട്പുട്ടിലുണ്ടായിരിക്കുന്ന വ്യത്യാസത്തിന് കാരണം കണ്ടെത്തുക. തന്നിരിക്കുന്ന ഏതൊരു സ്ട്രിങ്ങിനും അനുസരിച്ച് കൃത്യമായ ഔട്ട്പുട്ട് ലഭിക്കുന്നതിന് പ്രോഗ്രാമിൽ ആവശ്യമായ മാറ്റങ്ങൾ വരുത്തുക.

നമുക്ക് സംഗ്രഹിക്കാം

അറെ എന്നാൽ തുടർച്ചയായ മെമ്മറി സ്ഥാനങ്ങളിൽ ശേഖരിച്ചു വച്ചിട്ടുള്ള ഒരേ തരത്തിലുള്ള ഡാറ്റകളുടെ സമൂഹമാണ്. ഒരു പേരിൽ ഒരേ തരത്തിലുള്ള ഒരു കൂട്ടം വിലകൾ ശേഖരിക്കുന്നതിനായി അറെകൾ ഉപയോഗിക്കുന്നു. ഒരു അറെയിലെ എല്ലാ അംഗങ്ങളെയും സൂചികയുടെ സഹായത്താൽ ഉപയോഗിക്കുവാൻ കഴിയും. അറെയിലെ മെമ്മറി നീക്കിവയ്പ് ചിത്രങ്ങളുടെ സഹായത്താൽ വിശദീകരിച്ചിട്ടുണ്ട്. ലൂപ്പിങ് പ്രസ്താവനകൾ പ്രത്യേകിച്ചും അറെ അംഗങ്ങളെ മാറ്റുന്നതിനായുള്ളവ പ്രോഗ്രാമുകളുടെ സഹായത്തോടെ വിശദീകരിച്ചിട്ടുണ്ട്. പ്രോഗ്രാമുകളിൽ അറെ ഉപയോഗിച്ച് സ്ട്രിങ്ങുകൾ എങ്ങനെ കാര്യക്ഷമമായി കൈകാര്യം ചെയ്യാമെന്നും നാം പഠിച്ചു.

നമുക്ക് പരിശീലിക്കാം

1. Sales Amt എന്ന അറെയിലേക്ക് 12 മാസത്തെ വിറ്റ്വരവിന്റെ തുക ഇൻപുട്ട് നൽകുവാനുള്ള C++ പ്രോഗ്രാം എഴുതുക. എല്ലാ ഇൻപുട്ടും നൽകിയ ശേഷം ആകെ വിറ്റ് വരവിന്റെ തുകയും ശരാശരിയും കണക്കാക്കുക.
2. N അക്കങ്ങൾ ഉള്ള ഒരു അറെ നിർമിക്കുവാനുള്ള C++ പ്രോഗ്രാം എഴുതുക, ഇതിന്റെ ശരാശരി കണ്ടുപിടിച്ച് ശരാശരിയിൽ കൂടുതൽ ഉള്ള അക്കങ്ങൾ പ്രദർശിപ്പിക്കുക.
3. ഒരു പൂർണസംഖ്യ അറെയുടെ ആദ്യത്തേയും അവസാനത്തേയും അംഗങ്ങളെ ഒത്തു മാറ്റുവാൻ (swap) ഉള്ള C++ പ്രോഗ്രാം എഴുതുക.
4. പത്ത് പൂർണ സംഖ്യകളുടെ ഒരു അറെയിൽ സ്വീകരിക്കുവാനുള്ള C++ പ്രോഗ്രാം എഴുതുകയും ഇവയിൽ നിന്ന് ഏറ്റവും കൂടിയ സംഖ്യയും, കൂടിയ സംഖ്യയും കണ്ടെത്തുക.
5. ഒരു വാചകം അഥവാ സ്ക്രീംഗ് സ്വീകരിക്കുവാനുള്ള ഒരു C++ പ്രോഗ്രാം എഴുതുകയും വലിയ അക്ഷരങ്ങൾ, ചെറിയ അക്ഷരങ്ങൾ, സംഖ്യകൾ, പ്രത്യേക ചിഹ്നങ്ങൾ, ശൂന്യ സ്ഥലം തുടങ്ങിയവയുടെ എണ്ണം കണ്ടെത്തുക.
6. ഒരു വാചകത്തിന്റെ വാക്കുകളുടെ എണ്ണം കണ്ടെത്തുവാനുള്ള C++ പ്രോഗ്രാം എഴുതുക.
7. ഒരു സ്ട്രിംഗിന്റെ അളവ് കണ്ടെത്തുവാനുള്ള C++ പ്രോഗ്രാം എഴുതുക.

നമുക്ക് വിലയിരുത്താം

1. പത്ത് അംഗങ്ങളുള്ള ഒരു അറെയിലെ അംഗങ്ങളെ എണ്ണപ്പെടുന്നത് .... മുതൽ .... വരെയാണ്.
2. ഒരു അറെയിലെ അംഗത്തെ ഉപയോഗിക്കുന്നത് .... ആധാരമാക്കിയാണ്.
3. AR എന്നത് ഒരു അറെയാണ് എങ്കിൽ, AR[7] എന്നത് അറെയിലെ ഏത് അംഗത്തേ പ്രതിനിധീകരിക്കുന്നു.
4. int a[3] = {2,3,4}; എന്ന അറെയുടെ പ്രഖ്യാപനം പരിഗണിക്കുക. a[1] എന്നതിന്റെ വില എന്താണ്?
5. int a[ ] = {1,2,4} എന്ന അറെയുടെ പ്രഖ്യാപനം പരിഗണിക്കുക. എങ്കിൽ a[1] എന്നതിന്റെ വില എന്താണ്?
6. അറെയിലെ എല്ലാ അംഗങ്ങളെയും അച്ചടിക്കുക എന്നത് .... പ്രവർത്തനത്തിന് ഉദാഹരണമാണ്.

7. ചുവടെ തന്നിരിക്കുന്ന കോഡ് ശകലത്തിന്റെ ഔട്ട്പുട്ട് എഴുതുക.

```
puts ("hello");
puts ("friends");
```

8. "GCC" എന്ന വാക്ക് സൂക്ഷിക്കുന്നതിനായുള്ള പ്രാരംഭ വില നൽകൽ പ്രസ്താവന എഴുതുക.

9. അറെ നിർവചക്കുക.

10. int studlist[1000] എന്ന പ്രഖ്യാപനം കൊണ്ട് ഉദ്ദേശിക്കുന്നത് എന്താണ്?

11. എങ്ങനെയാണ് ഏക ത്രിമാനമായ/ഡയമെൻഷണൽ ആയ അറെയ്ക്കായി മെമ്മറി നീക്കി വയ്ക്കുന്നത്.

12. പത്ത് അംഗങ്ങളെ സ്വീകരിക്കുന്നതിനായിട്ടുള്ള C++ പ്രസ്താനവകൾ എഴുതുകയും ഒറ്റ സംഖ്യകളുടെയും ഇരട്ട സംഖ്യകളുടെ എണ്ണം പ്രദർശിപ്പിക്കുക.

13. ചുവടെ ചേർത്തിരിക്കുന്ന പ്രസ്താവനകൾ വായിക്കുക.

```
char namep[20]
cin>>name;
cout<<name;
```

നിങ്ങൾ "sachin tendulkar" എന്ന വാചകമാണ് ഇൻപുട്ടായി നൽകുന്നത് എങ്കിൽ എന്താകും ഇതിന്റെ ഔട്ട്പുട്ട്. നിങ്ങളുടെ ഉത്തരത്തെ സാധൂകരിക്കുക.

14. ഒരേ വലിപ്പമുള്ള രണ്ട് ഏക ഡയമെൻഷണൽ അറെ സ്വീകരിക്കുവാനുള്ള പ്രസ്താവനകൾ എഴുതുകയും തത്തുല്യമായ അംഗങ്ങൾ തമ്മിലുള്ള വ്യത്യാസം കണ്ടെത്തുകയും ചെയ്യുക.

15. ഒരു വാചകം അല്ലെങ്കിൽ ഒരു വാക്ക് അനുലോമവിലോമ പദം (palindrome) ആണോ എന്ന് പരിശോധിക്കുവാനുള്ള പ്രോഗ്രാം എഴുതുക.

F9V8H8

3 ഫങ്ഷനുകൾ

പ്രധാന ആശയങ്ങൾ

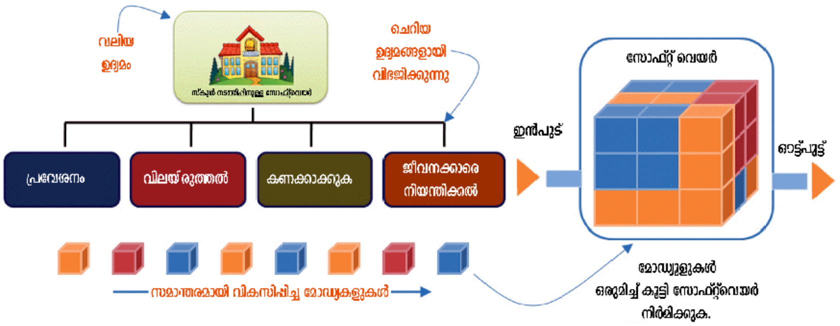
ഈ അധ്യായത്തിന്റെ പൂർത്തീകരണത്തിനു ശേഷം പഠിതാവ്

- പ്രശ്ന പരിഹാരത്തിന് മോഡുലാർ പ്രോഗ്രാമിങ്ങിന്റെ മേന്മകൾ തിരിച്ചറിയുന്നു.
- വിവിധങ്ങളായ ക്യാരക്ടറിനും സ്ട്രിങ് ഡാറ്റയ്ക്കും വേണ്ടിയുള്ള ഇൻപുട്ട്/ഔട്ട്പുട്ട് ഫങ്ഷനുകൾ തരംതിരിക്കുന്നു.
- ക്യാരക്ടർ ഇൻപുട്ട് ഫങ്ഷനുകൾ താരതമ്യം ചെയ്യുന്നു.
- ഇൻപുട്ട്/ഔട്ട്പുട്ട് പ്രവർത്തനങ്ങൾക്ക് യോജിച്ച ക്യാരക്ടറോ, സ്ട്രിങ് ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നു.
- പ്രശ്നങ്ങൾ പരിഹരിക്കുന്നതിന് വേണ്ട മാത്ത്മാറ്റിക്കൽ ഫങ്ഷനുകൾ പ്രയോഗിക്കുന്നു.
- സ്ട്രിങ് ഡാറ്റ കൈകാര്യം ചെയ്യുന്നതിന് വേണ്ടി സ്ട്രിങ് ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നു.
- ക്യാരക്ടർ ഡാറ്റ മുൻനിർവചിത ക്യാരക്ടർ ഫങ്ഷനുകൾ ഉപയോഗിച്ച് കൈകാര്യം ചെയ്യുന്നു.
- ഫങ്ഷനുകൾ നിർമിച്ച് മോഡുലാർ പ്രോഗ്രാമിങ് നടപ്പിലാക്കുന്നു.
- ആർഗ്യുമെന്റുകളുടെ പ്രാധാന്യം തിരിച്ചറിയുകയും വിവിധ തരങ്ങളിലുള്ള ഫങ്ഷൻ വിളികൾ താരതമ്യം ചെയ്യുന്നു.
- ഒരു പ്രോഗ്രാമിലെ വേരിയബിളുകളുടെയും ഫങ്ഷനുകളുടെയും സ്കോപ് മനസ്സിലാക്കുന്നു.

കഴിഞ്ഞ അധ്യായങ്ങളിൽ ലളിതമായ ചില പ്രോഗ്രാമുകൾ നാം ചർച്ച ചെയ്തു. എന്നാൽ സങ്കീർണ്ണമായ പ്രശ്നങ്ങൾ പരിഹരിക്കുന്നതിന് ആയിരക്കണക്കിന് വരികളുള്ള വലിയ പ്രോഗ്രാമുകൾ ആവശ്യമുണ്ട്. അധ്യായം 4 ൽ ചർച്ച ചെയ്തതുപോലെ സങ്കീർണ്ണമായ പ്രശ്നങ്ങൾ ചെറിയ പ്രശ്നങ്ങളായി വിഭജിക്കുകയും അവ ഓരോന്നും പരിഹരിക്കുന്നതിന് വേണ്ട പ്രോഗ്രാമുകൾ എഴുതുകയും ചെയ്യുന്നു. മറ്റൊരു രീതിയിൽ പറഞ്ഞാൽ നാം വലിയ പ്രോഗ്രാമുകളെ ചെറിയ ഉപപ്രോഗ്രാമുകളായി വിഭജിക്കുന്നു. C++ൽ ഫങ്ഷൻ എന്നത് വലിയ പ്രോഗ്രാമുകളെ ചെറിയ ഉപപ്രോഗ്രാമുകളായി വിഭജിക്കുന്നതിനുള്ള ഒരു മാർഗ്ഗമാണ്. നമുക്ക് `main()` ഫങ്ഷൻ പരിചിതമാണ്. ഒരു C++ പ്രോഗ്രാമിൽ `main()` ഫങ്ഷൻ ഒഴിച്ചു കൂടാൻ സാധിക്കാത്താണ് എന്ന് നമുക്കറിയാം. ഒരു പ്രശ്നം പരിഹരിക്കുന്നതിന് ആവശ്യമായ പ്രസ്താവനകൾ `int main()` എന്ന തലക്കെട്ടിന് ശേഷം ഒരു ജോഡി ബ്രെയ്സ് ബ്രാക്കറ്റിനുള്ളിൽ { ഉം } കൊടുത്തിരിക്കുന്നു. നമ്മൾ ഒരു പ്രോഗ്രാമിനെ ഇതുവരെയും ഉപഭാഗങ്ങളായി വിഭജിക്കാറില്ല. അതിനാൽ മൊത്തം ഉദ്യമത്തിനെയും `main()` ഏൽപിച്ചു നൽകി. എന്നാൽ ഉടൻ തന്നെ ലഭ്യമായ ചില ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നുമുണ്ട്. ഇതു പ്രോഗ്രാമിങ് എളുപ്പത്തിൽ ആക്കുന്നു. ഇവയിൽ ഓരോന്നിനും ഒരു പ്രത്യേക ഉദ്യമം ഏൽപിച്ചു നൽകുകയും അവ ഹെഡർ ഫയലുകളിൽ സൂക്ഷിക്കുകയും ചെയ്യുന്നു. ഇതു കൂടാതെ ഇത്തരം ഫങ്ഷനുകൾ പ്രത്യേക ഉദ്യമത്തിനായി നമുക്ക് നിർവചിക്കുവാൻ കഴിയും. ഇവയെ ഉപയോക്തൃ നിർവചിത ഫങ്ഷനുകൾ എന്നു വിളിക്കുന്നു. ഈ അധ്യായത്തിൽ ചില പ്രധാനപ്പെട്ട മുൻ നിർവചിത ഫങ്ഷനുകളെ കുറിച്ച് ചർച്ച ചെയ്യുകയും, നമ്മുടെ സ്വന്തം ഫങ്ഷനുകൾ എങ്ങനെ നിർവചിക്കാമെന്നും പഠിക്കുന്നു. ഇതിലേക്കൊക്കെ കടക്കുന്നതിനു മുമ്പ് മോഡുലാർ പ്രോഗ്രാമിങ് എന്നു വിളിക്കുന്ന പ്രോഗ്രാമിങ് ശൈലിയെക്കുറിച്ച് നമുക്ക് പരിചയപ്പെടാം.

3.1 മോഡുലാർ പ്രോഗ്രാമിങ്ങിന്റെ ആശയം (Concept of Modular programming)

ഒരു സ്കൂളിന് ആവശ്യമായ സോഫ്റ്റ്‌വെയറിന്റെ കാര്യം നമുക്ക് പരിഗണിക്കാം. വ്യത്യസ്ത ഉദ്യമങ്ങൾക്ക് വേണ്ട ധാരാളം പ്രോഗ്രാമുകൾ അടങ്ങിയ വളരെ വലുതും സങ്കീർണ്ണവുമായ ഒരു സോഫ്റ്റ്‌വെയർ ആണിത്. ചിത്രം 10.1 കാണിച്ചിരിക്കുന്നത് പോലെ സങ്കീർണമായ സ്കൂൾ പ്രവർത്തനങ്ങളെ ചെറിയ ഉദ്യമങ്ങളോ മോഡ്യൂളുകളോ ആയി വിഭജിച്ച് സമാന്തരമായി നിർമ്മിച്ചതിന് ശേഷം ഒരുമിച്ച് കൂട്ടി പൂർണ്ണമായ ഒരു സോഫ്റ്റ്‌വെയർ നിർമിക്കുവാൻ കഴിയും. പ്രോഗ്രാമിങ്ങിൽ മുഴുവൻ പ്രശ്നത്തേയും ചെറിയ പ്രശ്നങ്ങളാക്കി



ചിത്രം 3.1 മോഡ്യുലാർ പ്രോഗ്രാമിങ് ശൈലി

വിഭജിച്ച് പ്രത്യേകം പ്രോഗ്രാമുകൾ എഴുതി അവ പരിഹരിക്കുകയും ചെയ്യും. ഈ തരത്തിലുള്ള സമീപനം മോഡുലാർ പ്രോഗ്രാമിങ് എന്നറിയപ്പെടുന്നു. ഓരോ ഉപഉദ്യമത്തേയും ഒരു മോഡ്യുളായി പരിഗണിക്കുകയും ഓരോ മോഡ്യുളുകളിലും നാം പ്രോഗ്രാം എഴുതുകയും ചെയ്യുന്നു. വലിയ പ്രോഗ്രാമുകളെ ചെറിയ ഉപപ്രോഗ്രാമാക്കി വിഭജിക്കുന്ന പ്രവർത്തനത്തെ മോഡുലറൈസേഷൻ എന്ന് വിളിക്കുന്നു.

മോഡുലറൈസേഷൻ നടപ്പിലാക്കുന്നതിന് കമ്പ്യൂട്ടർ പ്രോഗ്രാമിങ് ഭാഷകൾക്ക് വിവിധ മാർഗ്ഗങ്ങൾ ഉണ്ട്. ഉപപ്രോഗ്രാമുകളെ (sub program) സാധാരണയായി ഫങ്ഷനുകൾ എന്നാണ് വിളിക്കുന്നത്. C++ ഉം ഫങ്ഷനുകൾ കൊണ്ട് മോഡ്യുലാർ പ്രോഗ്രാമിങ് നടപ്പിലാക്കുന്നു.

3.1.1. മോഡ്യുലാർ പ്രോഗ്രാമിങ്ങിന്റെ മേൻമകൾ (Merits of modular programming)

മോഡ്യുലാർ ശൈലിയിൽ ഉള്ള പ്രോഗ്രാമിങ്ങിന് നിരവധി ഗുണങ്ങൾ ഉണ്ട്. ഇത് പ്രോഗ്രാമിന്റെ വലിപ്പവും സങ്കീർണ്ണതയും കുറച്ച് പ്രോഗ്രാം കൂടുതൽ വായനാ സുഖമുള്ളതും വീണ്ടും ഉപയോഗിക്കാൻ സാധിക്കുന്നതും, തെറ്റുകൾ കണ്ടുപിടിച്ച് അവ മാറ്റുന്ന പ്രവർത്തനം എളുപ്പത്തിലാക്കുകയും ചെയ്യുന്നു. ഈ പ്രത്യേകതകൾ വിശദമായി നമുക്ക് ചർച്ച ചെയ്യാം.

പ്രോഗ്രാമിന്റെ വലിപ്പം കുറക്കുന്നു: ചില സന്ദർഭങ്ങളിൽ ഒരു പ്രോഗ്രാമിലെ ചില നിർദേശങ്ങൾ പ്രോഗ്രാമിന്റെ വിവിധ ഭാഗങ്ങളിൽ ആവർത്തിച്ചേക്കാം. $\frac{x^5 + y^7}{\sqrt{x} + \sqrt{y}}$; എന്ന പദപ്രയോഗം പരിഗണിക്കുക. x ന്റെയും y യുടേയും വിലകൾ ഉപയോഗിച്ച് ഈ പദപ്രയോഗത്തിന്റെ വില കണ്ടുപിടിക്കുന്നതിന് താഴെകൊടുത്തിരിക്കുന്ന നിർദേശങ്ങൾ നമുക്ക് ഉപയോഗിക്കേണ്ടതുണ്ട്.

1. x ന്റെ അഞ്ചാമത്തെ വർഗ്ഗം കണ്ടുപിടിക്കുക.
2. y യുടെ ഏഴാമത്തെ വർഗ്ഗം കണ്ടുപിടിക്കുക.
3. ഘട്ടം ഒന്നിലും രണ്ടിലും ലഭിച്ച ഫലങ്ങൾ കൂട്ടുക.
4. x ന്റെ വർഗ്ഗമൂലം കണ്ടുപിടിക്കുക.
5. y യുടെ വർഗ്ഗമൂലം കണ്ടുപിടിക്കുക.
6. ഘട്ടം 4 ലും 5 ലും ലഭിച്ച ഫലങ്ങൾ കൂട്ടുക.
7. ഘട്ടം 3 ൽ ലഭിച്ച ഫലത്തെ ഘട്ടം 6 ൽ ലഭിച്ച ഫലം കൊണ്ട് ഭാഗിക്കുക.

ഘട്ടം 1 ന്റെയും ഘട്ടം 2 ന്റെയും ഉത്തരങ്ങൾ കണ്ടുപിടിക്കുന്നതിന് പ്രത്യേകം ലൂപ്പുകൾ ആവശ്യമാണെന്ന് നമുക്ക് അറിയാം. ഒരു സംഖ്യയുടെ വർഗ്ഗമൂലം കാണുന്നതിനാവശ്യമായ യുക്തിയുടെ സങ്കീർണ്ണത നിങ്ങൾക്ക് സങ്കൽപ്പിക്കാൻ കഴിയുമോ?

വ്യത്യസ്ത ഇടങ്ങളിൽ വ്യത്യസ്ത ഡാറ്റ പ്രവർത്തിപ്പിക്കുന്നതിന് ഒരേ നിർദേശങ്ങൾ പ്രോഗ്രാമിന് ആവശ്യമാണ് എന്നത് ഇതിൽ നിന്ന് വ്യക്തമാണ്. ആവർത്തിക്കുന്ന ഉദ്യമങ്ങൾ വേർതിരിക്കുന്നതിനും ഇതിന് വേണ്ട നിർദേശങ്ങൾ എഴുതാനും മോഡുലാർ സമീപനം സഹായിക്കുന്നു. ഇത്തരം നിർദേശങ്ങളുടെ കൂട്ടത്തിന് പേര് നിർദേശിക്കുവാനും ആ പേര് ഉപയോഗിച്ച് ഇവയെ പ്രവർത്തിപ്പിക്കുന്നതിനും നമുക്ക് കഴിയും അങ്ങനെ പ്രോഗ്രാമിന്റെ വലിപ്പം കുറക്കുന്നു.

തെറ്റിനുള്ള സാധ്യത: *പ്രോഗ്രാമിന്റെ വലിപ്പം കുറയുമ്പോൾ സ്വാഭാവികമായും വാക്യഘടനയിലെ തെറ്റുകളും കുറയും.* യുക്തിപരമായ തെറ്റുകൾ ഉണ്ടാവാനുള്ള സാധ്യത പരിമിതപ്പെടും. സങ്കീർണ്ണമായ പ്രശ്നങ്ങൾ പരിഹരിക്കപ്പെടുമ്പോൾ പ്രശ്നത്തിന്റെ എല്ലാവശങ്ങളും നമുക്ക് പരിഗണിക്കേണ്ടതായി വരും. അതിനാൽ പ്രശ്ന പരിഹാരത്തിന് വേണ്ട യുക്തിയും സങ്കീർണമാകും. എന്നാൽ മോഡ്യുലറൈസ് ചെയ്യപ്പെട്ട ഒരു പ്രോഗ്രാമിൽ ഒരു സമയം ഒരു മോഡ്യൂളിൽ മാത്രം നാം ശ്രദ്ധിച്ചാൽ മതിയാകും. ഔട്ട്പുട്ടിൽ എന്തെങ്കിലും തെറ്റ് കണ്ടുപിടിക്കപ്പെട്ടാൽ ബന്ധപ്പെട്ട മോഡ്യൂൾ കണ്ടെത്തി അവിടെ വച്ച് തന്നെ തെറ്റ് തിരുത്തുവാനും നമുക്ക് കഴിയും.

പ്രോഗ്രാമിങ്ങിന്റെ സങ്കീർണ്ണത കുറക്കുന്നു: മുകളിൽ കണ്ടെത്തിയ രണ്ട് ഗുണങ്ങളുടേയും മുഴുവൻ ഫലം പ്രോഗ്രാമിങ്ങിന്റെ സങ്കീർണത കുറയ്ക്കുന്നു എന്നതാകുന്നു. നാം പ്രശ്നത്തെ ചെറിയ ഭാഗങ്ങളായി കൃത്യമായി ഭാഗിക്കുകയാണെങ്കിൽ പ്രശ്ന പരിഹാരത്തിനു വേണ്ട യുക്തിവികസനം ലളിതമാകും. അങ്ങനെ മോഡുലറൈസേഷൻ ഒരു സമയത്ത് ലഘൂകരിക്കപ്പെട്ട ഒരു ഉദ്യമം നമ്മുടെ മനസ്സിലേക്ക് കൊണ്ട് വന്ന് പ്രോഗ്രാമിങ്ങിന്റെ വലിപ്പം കുറക്കുകയും അവയിലുള്ള തെറ്റുകൾ കണ്ടുപിടിച്ച് തിരുത്തുന്ന പ്രവർത്തനം എളുപ്പത്തിലാക്കി പ്രോഗ്രാമിന്റെ സങ്കീർണ്ണത കുറക്കുകയും ചെയ്യുന്നു.

പുനരുപയോഗം മെച്ചപ്പെടുത്തുന്നു: ഓരോ തവണയും പുതിയതായി ഒരു ഫങ്ഷൻ എഴുതുന്നതിന് പകരം ഒരിക്കൽ എഴുതപ്പെട്ട ഒരു ഫങ്ഷൻ മറ്റനവധി പ്രോഗ്രാമുകളിൽ പിന്നീട് ഉപയോഗിക്കപ്പെട്ടേക്കാം. ഇത് പ്രോഗ്രാം വികസനത്തിന് വേണ്ട സമയം കുറക്കുന്നു.

3.1.2 മോഡുലാർ പ്രോഗ്രാമിങ്ങിന്റെ ന്യൂനതകൾ (Demerits of modular programming)

മോഡ്യുലാർ പ്രോഗ്രാമിങ്ങിന് പ്രബലമായ മേൻമകൾ ഉണ്ടെങ്കിലും പ്രശ്നത്തെ ശരിയായ രീതിയിൽ വിഭജിക്കുക എന്നത് ഒരു വെല്ലുവിളി ആണ്. ഓരോ ഉപ പ്രശ്നങ്ങളും മറ്റുള്ളവയിൽ നിന്ന് സ്വതന്ത്രമായിരിക്കണം. മോഡ്യൂളുകളുടെ പ്രവർത്തന ശ്രേണി തയാറാക്കുമ്പോൾ അങ്ങേയറ്റം ശ്രദ്ധപുലർത്തണം.

ചിത്രം 3. 2 കോഫി ഉണ്ടാക്കുന്ന യന്ത്രത്തിന്റെ പ്രവർത്തനം.

3.2 C++ ലെ ഫങ്ഷനുകൾ (Functions in C++)

കോഫി ഉണ്ടാക്കുന്ന യന്ത്രത്തിന്റെ കാര്യം നമുക്ക് പരിഗണിക്കാം. ചിത്രം 10.2 നെ അടിസ്ഥാനമാക്കി അതിന്റെ പ്രവർത്തനങ്ങൾ ചർച്ച ചെയ്യാം. യന്ത്രത്തിലേക്ക് വെള്ളം, പാൽ, പഞ്ചസാര, കാപ്പിപ്പൊടി എന്നിവ കൊടുക്കുന്നു. യന്ത്രം മുൻകൂട്ടി സൂക്ഷിച്ചിട്ടുള്ള നിർദേശങ്ങൾക്ക് അനുസരിച്ച് പ്രവർത്തിപ്പിക്കുകയും ഇവ ഉപയോഗിച്ച് തയ്യാറാക്കുന്ന കോഫി ഒരു കപ്പിൽ ശേഖരിക്കുകയും ചെയ്യുന്നു. അതിനുവേണ്ട നിർദ്ദേശങ്ങൾ താഴെ കൊടുത്തിരിക്കുന്നതു പോലെ ആയിരിക്കും.

1, 60 മി.ല്ലി പാൽ, 120 മി.ല്ലി വെള്ളം 5 ഗ്രാം കാപ്പിപ്പൊടി 20 ഗ്രാം പഞ്ചസാര എന്നിവ യന്ത്രത്തിന്റെ സംഭരണ സ്ഥലത്ത് നിന്ന് എടുക്കുക.

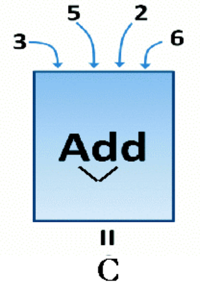
2, മിശ്രിതം തിളപ്പിക്കുക.

3, അവയെ നിർഗമന മാർഗ്ഗത്തിലേക്ക് കൈമാറുക.

ഈ പ്രവർത്തനങ്ങൾ തുടങ്ങുവാൻ യന്ത്രത്തിൽ ഒരു ബട്ടൺ സാധാരണയായി ഉണ്ടാകും. ഈ ബട്ടണിന് make coffee എന്ന പേര് ഉപയോഗിച്ച് നമുക്ക് നാമകരണം ചെയ്യാം. ഈ പ്രവർത്തനം പ്രതീകാത്മമായി താഴെ പറയുന്ന രീതിയിൽ രേഖപ്പെടുത്താൻ കഴിയും.

കപ്പ് = കാപ്പി ഉണ്ടാക്കുക (വെള്ളം, പാൽ, പഞ്ചസാര, കാപ്പിപൊടി).

ഇവയെല്ലാം പ്രോഗ്രാമിലെ ഫങ്ഷനുകളുമായി നമുക്ക് താരതമ്യം ചെയ്യാം. കാപ്പി ഉണ്ടാക്കുക എന്ന പദം ഫങ്ഷന്റെ പേര് ആയും (വെള്ളം, പാൽ, പഞ്ചസാര, കാപ്പിപൊടി) എന്നിവ ഫങ്ഷന് വേണ്ട പാരാമീറ്ററുകളായും "Coffee" തിരിച്ച് കിട്ടുന്ന ഫലവുമാണ്. ഇത് കപ്പിൽ സംഭരിക്കുന്നു. കപ്പിന് പകരം ഗ്ലാസ്സോ, ടംബ്ലറോ അല്ലെങ്കിൽ മറ്റേതെങ്കിലും പാത്രമോ നമുക്ക് ഉപയോഗിക്കാം. അതുപോലെ തന്നെ ഒരു C++ ഫങ്ഷൻ പാരാമീറ്ററുകൾ സ്വീകരിക്കുകയും അതിൽ പ്രവർത്തിച്ച് ഫലം തിരിച്ച് നൽകുകയും ചെയ്യുന്നു. ചിത്രം 10.3 ഒരു ഫങ്ഷനായി കണക്കാക്കാം. *അതിന് 3,5 2,6 എന്നീ വിലകൾ പാരാമീറ്ററുകളായി സ്വീകരിച്ച അവ തമ്മിൽ കൂട്ടുകയും* തുക C എന്ന വേരിയബിളിൽ ശേഖരിക്കുകയും ചെയ്യുന്നു. അത് താഴെ പറയുന്ന രീതിയിൽ എഴുതാം.



ചിത്രം 3.3: കൂട്ടുന്നതിന് ഉള്ള ഫങ്ഷൻ

C = Add (3, 5, 2, 6)

ഒരു പ്രോഗ്രാമിൽ പ്രശ്നപരിഹാരത്തിന്റെ ഭാഗമായി ഒരു പ്രത്യേക ഉദ്യമം നിർവഹിക്കുന്നതിന് നാമകരണം ചെയ്യപ്പെട്ട ഒരു കൂട്ടം നിർദ്ദേശങ്ങളുടെ ഘടകമാണ് ഫങ്ഷൻ എന്ന് നമുക്ക് പറയാം. എല്ലാ ഫങ്ഷനുകൾക്കും പരാമീറ്ററുകൾ ആവശ്യമാണ് എന്നതും അവയെല്ലാം ചില വില തിരിച്ചു നൽകണമെന്നതും നിർബന്ധമില്ല വിവിധ ഉദ്യമങ്ങൾക്ക് എപ്പോഴും ഉപയോഗിക്കാൻ കഴിയുന്ന ഫങ്ഷനുകളുടെ വിലപ്പെട്ട ശേഖരം C++ നൽകുന്നു. (getch(), pow() sqrt()) തുടങ്ങിയ ഫങ്ഷനുകളിൽ അവ ചെയ്യേണ്ട ഉദ്യമങ്ങൾ നേരത്തെ തന്നെ എഴുതപ്പെട്ടതും തെറ്റുകൾ തിരുത്തി കമ്പയിൽ ചെയ്ത് അവയുടെ നിർവ്വചനങ്ങൾ ഹെഡർ ഫയലുകൾ എന്ന് വിളിക്കുന്ന ഫയലുകളിൽ സൂക്ഷിച്ചിരിക്കുകയും ചെയ്യുന്നു. ഉപയോഗത്തിന് തയ്യാറായ ഇത്തരം ഉപപ്രോഗ്രാമുകളെ **മുൻനിർവ്വചിത ഫങ്ഷനുകൾ** അല്ലെങ്കിൽ **അന്തർനിർമിത ഫങ്ഷനുകൾ** (built-in functions) എന്ന് വിളിക്കുന്നു.

വലിയ പ്രോഗ്രാമുകൾ എഴുതുമ്പോൾ മോഡ്യുലറൈസേഷൻ നടത്തുന്നതിന് ഇത്തരം മുൻനിർവചിത ഫങ്ഷനുകൾ മതിയാവില്ല. ചില പ്രത്യേക ഉദ്യമങ്ങൾ നിർവഹിക്കുന്നതിന് നമ്മുടെ സ്വന്തം ഫങ്ഷനുകൾ നിർവചിക്കുന്നതിന് വേണ്ട സൗകര്യം C++ നൽകുന്നു.

നിർവഹിക്കേണ്ട ഉദ്യമം, പേര്, ആവശ്യമുള്ള ഡാറ്റ എന്നിങ്ങനെ ഒരു ഫങ്ഷനുമായി ബന്ധപ്പെട്ട സകലതും ഉപയോക്താവിനാൽ തീരുമാനിക്കപ്പെടുന്നതിനാൽ അവ ഉപയോക്തൃ നിർവചിത ഫങ്ഷനുകൾ (user defined function) എന്ന് അറിയപ്പെടുന്നു.

അപ്പോൾ (main ( )) ഫങ്ഷന്റെ ആവശ്യം എന്താണ്? ഉപയോക്താവ് ഉദ്യമം തീരുമാനിക്കുന്നു എന്ന അർത്ഥത്തിൽ ഇവ ഉപയോക്തൃ നിർമിത ഫങ്ഷൻ ആയി പരിഗണിക്കാവുന്നതാണ്. പ്രോഗ്രാമിന്റെ പ്രവർത്തനം main ( ) ഫങ്ഷനിൽ നിന്ന് ആരംഭിക്കുന്നതിനാൽ ഇത് C++ ലെ ഒഴിച്ചു കൂടാൻ സാധിക്കാത്ത ഫങ്ഷനാണ്. main () ഫങ്ഷനില്ലാതെ C++ പ്രോഗ്രാം പ്രവർത്തിക്കില്ല. എല്ലാ ഫങ്ഷനുകളും ഒരു സ്റ്റേറ്റ്മെന്റിൽ നിന്ന് അവ വിളിക്കുമ്പോഴാണ് പ്രവർത്തിക്കുന്നത്.

3.3 മുൻകൂട്ടി നിർവചിച്ച ഫങ്ഷനുകൾ (Predefind Functions)

വിവിധ ഉദ്യമങ്ങൾക്ക് വേണ്ടി C++ ധാരാളം ഫങ്ഷനുകൾ ലഭ്യമാക്കുന്നു. ഏറ്റവും സാധാരണയായി ഉപയോഗിക്കുന്ന ഫങ്ഷനുകൾ മാത്രം നാം ചർച്ചചെയ്യുന്നു. ഇത്തരം ഫങ്ഷനുകൾ ഉപയോഗിക്കുമ്പോൾ അവയിൽ ചിലതിന് നിശ്ചയിച്ചിട്ടുള്ള ഉദ്യമം നിർവഹിക്കുന്നതിന്, ഡാറ്റ ആവശ്യമാണ്. ഫങ്ഷന്റെ പേരിന് ശേഷം പാരന്തസീസ് എന്ന ഒരു ജോഡി ബ്രാക്കറ്റുകൾക്കുള്ളിൽ ഉപയോഗിക്കുന്ന ഈ ഡാറ്റയെ പരാമീറ്ററുകൾ അല്ലെങ്കിൽ ആർഗ്യുമെന്റുകൾ എന്ന് നാം വിളിക്കുന്നു.

ഉദ്യമങ്ങൾ നിർവഹിച്ചതിനു ശേഷം ഫലങ്ങൾ നൽകുന്ന ചില ഫങ്ഷനുകൾ ഉണ്ട്. ഈ ഫലം ഫങ്ഷൻ തിരിച്ചു നൽകുന്ന വില എന്നറിയപ്പെടുന്നു. ചില ഫങ്ഷനുകൾ ഒരു വിലയും തിരിച്ചു തരുന്നില്ല. പകരം അവയുടെ പ്രത്യേക ഉദ്യമം നിർവ്വഹിക്കുന്നു. തുടർന്നുള്ള ഭാഗങ്ങളിൽ സ്ട്രിങ്ങുകൾ കൈകാര്യം ചെയ്യുന്നതിനും ഗണിത പ്രക്രിയകൾ നടത്തുന്നതിനും ക്യാരക്ടർ ഡാറ്റയിൽ പ്രവർത്തിക്കുന്നതിനുമുള്ള ഫങ്ഷനുകൾ നാം ചർച്ചചെയ്യും. ഇത്തരം ഫങ്ഷനുകൾ ഉപയോഗിക്കുമ്പോൾ അതുമായി ബന്ധപ്പെട്ട ഹെഡർ ഫയലുകൾ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തേണ്ടതാണ്.

3.3.1 കാരക്റ്റർ ഇൻപുട്ട്/ഔട്ട്പുട്ട് നുവേണ്ടിയുള്ള കൺസോൾ ഫങ്ഷനുകൾ (Console functions for character I/O)

സ്ട്രിങ്ങിന് മേലുള്ള ഇൻപുട്ട്/ഔട്ട്പുട്ട് പ്രവർത്തനങ്ങൾ നാം ചർച്ച ചെയ്ത് കഴിഞ്ഞു. കാരക്റ്ററുകൾക്ക് മേൽ പ്രയോഗിക്കുവാനുള്ള ചില ഇൻപുട്ട്/ഔട്ട്പുട്ട് ഫങ്ഷനുകളും C++ ൽ ഉൾപ്പെടുത്തിയിട്ടുണ്ട്. ഇത്തരം ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് **cstdio** (stdio.h എന്നത് Turbo C++ ൽ) എന്ന ഹെഡർ ഫയൽ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തേണ്ടത് അത്യാവശ്യമാണ് .

getchar()

ഈ ഫങ്ഷൻ കീബോർഡിലൂടെ ഇൻപുട്ട് ചെയ്ത കാരക്റ്ററിനെ തിരികെ തരികയാണ് ചെയ്യുന്നത്. താഴെ കൊടുത്തിട്ടുള്ള ഉദാഹരണത്തിൽ കാണുന്ന പോലെ ഒരു കാരക്റ്ററിനെ വേരിയബിലിലിലേക്ക് ശേഖരിക്കാവുന്നതാണ്.

```
char ch=getchar();
```

സ്ട്രിങ് ഔട്ട്പുട്ടിൽ puts() ഫങ്ഷന്റെ മേന്മകൾ നാം കണ്ടു കഴിഞ്ഞു. ഇനി നമുക്ക് കാരക്റ്റർ ഡാറ്റ ഔട്ട്പുട്ടായി ലഭിക്കുവാനുള്ള ഫങ്ഷനെക്കുറിച്ച് പഠിക്കാം.

putchar()

തന്നിരിക്കുന്ന കാരക്റ്റർ ആർഗ്യുമെന്റിനെ സ്റ്റാൻഡേർഡ് ഔട്ട്പുട്ട് ഉപകരണ (മോണിറ്റർ) ത്തിൽ പ്രദർശിപ്പിക്കുകയാണ് ഈ ഫങ്ഷൻ ചെയ്യുന്നത്. ഇവിടെ ആർഗ്യുമെന്റ് ഒരു കാരക്റ്റർ കോൺസ്റ്റന്റോ അല്ലെങ്കിൽ ഒരു വേരിയബിളോ ആവാം. ആർഗ്യുമെന്റായി ഒരു പൂർണ സംഖ്യയാണ് (integer) നൽകുന്നതെങ്കിൽ അതിനെ ഒരു ASCII വിലയായി പരിഗണിക്കുകയും അതിനുനസൃതമായ കാരക്റ്റർ പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു. താഴെ കൊടുത്തിട്ടുള്ള കോഡ് putchar() ഫങ്ഷന്റെ ഉപയോഗം വ്യക്തമാക്കുന്നു.

```
char ch='B';    // വേരിയബിൾ ch നകത്ത് 'B' ശേഖരിക്കപ്പെടുന്നു
putchar(ch);    //'B' സ്ക്രീനിൽ പ്രദർശിപ്പിക്കപ്പെടുന്നു
ptchar('c');    // 'c' സ്ക്രീനിൽ പ്രദർശിപ്പിക്കപ്പെടുന്നു
putchar(97);    // 97 എന്ന ASCII വിലയ്ക്കനുസൃതമായ 'a' സ്ക്രീനിൽ
                പ്രദർശിപ്പിക്കപ്പെടുന്നു.
```

പോഗ്രാം 3.1 ഈ ഫങ്ഷനുകളുടെ പ്രവർത്തനം വ്യക്തമാക്കുന്നതാണ്. ഒരു സ്ട്രിങ് ഇൻപുട്ട് ചെയ്ത് ഒരു കാരക്റ്റർ കണ്ടെത്തുവാൻ ഈ പ്രോഗ്രാമിലൂടെ സാധിക്കുന്നു.

പ്രോഗ്രാം 3.1 തന്നിരിക്കുന്ന കാരക്റ്റർ ഒരു സ്ട്രിങ്ങിനകത്ത് കൺസോൾ ഫങ്ഷൻ ഉപയോഗിച്ച് കണ്ടെത്തുക

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
```

```
{
   char str[20], ch;
   int i, num=0;
   puts("Enter a string:"); //To print '\n' after the string
   gets(str);//To accept a string with white spaces
   cout<<"Enter the character to be searched: ";
   ch=getchar(); //To input the character to be searched
   /* A loop to search for the character and count its
      occurrences in the string. Search will be
      terminated when a null character is found   */
   for(i=0; str[i]!='\0'; i++)
      if (str[i]==ch)
         num++;
   cout<<"\nThe number of occurences of the character ";
   putchar(ch);
   cout<<"\is : "<<num;
   return 0;
}
```

ഈ പ്രോഗ്രാമിന്റെ ഒരു മാതൃക ഔട്ട്പുട്ട് താഴെ കൊടുത്തിരിക്കുന്നു.

Enter the string :

examination

Enter the character to be searched : a

The number of occurences of the character a is : 2

3.3.2 ഇൻപുട്ട് / ഔട്ട്പുട്ട് പ്രക്രിയകൾക്ക് വേണ്ടിയുള്ള സ്ട്രീം ഫങ്ഷനുകൾ (Stream functions for I/O operations)

കാരക്റ്ററുകളിലും സ്ട്രിങ്ങുകളിലും ഇൻപുട്ട്/ഔട്ട്പുട്ട് പ്രക്രിയകൾ ചെയ്യുവാനുള്ള മറ്റൊരു സൗകര്യം C++ ൽ ലഭ്യമാക്കിയിട്ടുണ്ട്. iostream എന്ന ഹെഡർ ഫയലിൽ ഉൾപ്പെടുത്തിയിട്ടുള്ള ഫങ്ഷനുകളാണിവ. മെമ്മറിക്കും ഒബ്ജക്റ്റുകൾക്കുമിടയിൽ പ്രവഹിക്കുവാൻ ബൈറ്റുകളെ (ഡാറ്റ) (stream of bytes) യെ കൈകാര്യം ചെയ്യുന്നതിനാൽ ഇവയെ പൊതുവെ സ്ട്രീം ഫങ്ഷനുകൾ എന്നാണ് വിളിക്കുന്നത്. C++ൽ കീബോർഡ് ,മോണിറ്റർ എന്നിവയെയാണ് സാധാരണയായി ഒബ്ജക്റ്റുകളായി സൂചിപ്പിച്ചിരിക്കുന്നത്. ഇവയിൽ ഏതാനും ചില ഫങ്ഷനുകൾ നമുക്ക് പരിശോധിക്കാം .

A. ഇൻപുട്ട് ഫങ്ഷനുകൾ (Input functions)

കാരക്റ്റർ /സ്ട്രിങ് ഡാറ്റയെ ഇൻപുട്ട് ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്ന ഫങ്ഷനുകളാണിവ. ഒബ്ജക്റ്റുകൾക്കും മെമറിക്കുമിടയിൽ ബൈറ്റുകളെ പ്രവഹിക്കുവാൻ സഹായിക്കുന്ന ഫങ്ഷനുകളാണ് `get()`, `getline()` എന്നിവ. കീ ബോർഡ് ഉപയോഗിച്ച് ഡാറ്റ ഇൻപുട്ട്

ചെയ്യുമ്പോൾ കീബോർഡിനെ സൂചിപ്പിക്കാൻ cin എന്ന ഓബ്ജക്ട് ഉപയോഗിക്കുകയും മേൽപ്പറഞ്ഞ ഫങ്ഷനുകൾ **cin.get()**, **cin.getline()** എന്നീ രീതികളിൽ വിളിക്കുകയോ പ്രയോഗക്ഷമമാക്കുകയോ ചെയ്യുന്നു. ഇവിടെ ഡോട്ട് ഓപ്പറേറ്റർ എന്ന് വിളിക്കുന്ന **പിരിഡ് ചിഹ്നം (dot operator) (.)** ആണ് cin എന്ന ഒബ്ജെക്ടിനും ഫങ്ഷനുമിടയിൽ ഉപയോഗിച്ചിരിക്കുന്നത് .

i.get()

കീബോർഡിലൂടെ ഒരു കാരക്റ്ററിനെയോ ഒന്നിലധികം കാരക്റ്ററുകളെയോ സ്വീകരിക്കുവാൻ ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഒരു സ്ട്രിങ്ങിനെ സ്വീകരിക്കുന്നതിന് ഫങ്ഷന്റെ ആർഗ്യുമെന്റായി അറേയുടെ പേരും വലിപ്പവും നൽകേണ്ടതാണ്. താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് ഈ ഫങ്ഷന്റെ ഉപയോഗം വ്യക്തമാക്കുന്നതാണ്.

```
char ch,str[10];
ch = cin.get(ch);// ഒരു കാരക്റ്റർ സ്വീകരിച്ച് 'ch' ൽ ശേഖരിക്കുന്നു.
cin.get(ch);        // മേൽ സൂചിപ്പിച്ച പ്രസ്താവനയ്ക് സമാനം.
cin.get(str,10); // പരമാവധി 10 കാരക്റ്ററുകളുള്ള സ്ട്രിങ്ങിനെ സ്വീകരിക്കുന്നു.
```

ii. getline()

കീബോർഡിലൂടെ ഒരു സ്ട്രിങ്ങിനെ സ്വീകരിക്കുവാനുള്ള ഫങ്ഷനാണിത്. എന്റർ കീ, കാരക്റ്ററുകളുടെ എണ്ണം അല്ലെങ്കിൽ ഏതെങ്കിലും പ്രത്യേക കാരക്റ്റർ, ഇവയിൽ ഏതെങ്കിലും ഉപയോഗിച്ചാണ് സ്ട്രിങ്ങിന്റെ അവസാനം സൂചിപ്പിക്കുന്നത്. ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നതിനുള്ള രണ്ടുതരം വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
char ch,str[10];
int len;
cin.getline(str,len);      // 2 ആർഗ്യുമെന്റുകൾ സഹിതം.
cin.getline(str,len,ch); // 3 ആർഗ്യുമെന്റുകൾ സഹിതം .
```

ആദ്യത്തേതിൽ getline() ഫങ്ഷന് രണ്ട് ആർഗ്യുമെന്റുകളായ കാരക്റ്റർ അറേയും (ഇവിടെ str) കൂടാതെ ആകെ എത്ര കാരക്റ്ററുകൾ ശേഖരിക്കാമെന്നു സൂചിപ്പിക്കുന്ന ഇന്റിജർ വിലയും (len) ഉണ്ട്. രണ്ടാമത്തേതിൽ സ്ട്രിങ്ങിന്റെ അവസാനം (Delimiter) സൂചിപ്പിക്കുന്ന കാരക്റ്ററും (ch വേരിയബിളിന്റെ വില) കൂടി ആകെ കാരക്റ്ററുകളുടെ എണ്ണത്തിനൊപ്പം നൽകിയിരിക്കുന്നു. സ്ട്രിങ് ഇൻപുട്ട് ചെയ്യുമ്പോൾ ഒന്നുകിൽ കാരക്റ്ററുകൾ മാത്രം (len-1) അല്ലെങ്കിൽ സ്ട്രിങ്ങിന്റെ അവസാനം സൂചിപ്പിക്കുന്ന കാരക്റ്റർ വരെ, ഇവയിലേതാണോ ആദ്യം സംഭവിക്കുന്നത് എന്നതിനെ ആശ്രയിച്ചായിരിക്കും സ്ട്രിങ് ശേഖരിക്കപ്പെടുന്നത് .

B. ഔട്ട്പുട്ട് ഫങ്ഷനുകൾ (Output functions)

മെമ്മറിയ്ക്കും ഒബ്ജെക്റ്റിനുമിടയിൽ ഡാറ്റ ബൈറ്റുകൾ തുടർച്ചയായി പ്രവഹിക്കുവാൻ സഹായിക്കുന്ന ഔട്ട്പുട്ട് ഫങ്ഷനുകളാണ് put(), write() എന്നിവ. ഔട്ട്പുട്ട് ലഭിക്കുവാൻ വേണ്ടി മോണിറ്റർ ഉപയോഗിക്കുന്നതിനാൽ cout എന്ന ഒബ്ജക്റ്റ് ആണ് ഈ ഫങ്ഷനുകളുടെ കൂടെ ഉപയോഗിക്കുന്നത് .

i. put()

ഒരു കാരക്റ്റർ കോൺസ്റ്റന്റോ അല്ലെങ്കിൽ വേരിയബിളോ ആർഗ്യുമെന്റായി സ്വീകരിച്ചു പ്രദർശിപ്പിക്കുവാൻ ഉപയോഗിക്കുന്ന ഫങ്ഷനാണിത്.

```
char ch='c';
cout.put(ch);  // 'c' പ്രദർശിപ്പിക്കപ്പെടുന്നു.
cout.put('B'); // 'B' പ്രദർശിപ്പിക്കപ്പെടുന്നു.
cout.put(65);  // 'A' പ്രദർശിപ്പിക്കപ്പെടുന്നു.
```

ii. write()

ആർഗ്യുമെന്റായി നൽകിയിട്ടുള്ള സ്ട്രിങ്ങിനെ പ്രദർശിപ്പിക്കുവാനാണ് ഇത് ഉപയോഗിക്കുന്നത്. വ്യക്തതയ്ക്ക് വേണ്ടി താഴെ കൊടുത്തിരിക്കുന്ന ഉദാഹരണം കാണുക.

```
char str[10] ="hello";
cout.write(str,10);
```

മേൽപ്പറഞ്ഞ കോഡ് പ്രവർത്തിപ്പിക്കുമ്പോൾ "hello"എന്ന സ്ട്രിങ്ങിന് ശേഷം 5 വൈറ്റ് സ്പേസോടു കൂടിയാണ് പ്രദർശിപ്പിക്കപ്പെടുന്നത്. കാരണം രണ്ടാമത്തെ ആർഗ്യുമെന്റിന്റെ വില 10 ഉം കൂടാതെ സ്ട്രിങ്ങിലെ ആകെ കാരക്റ്ററുകളുടെ എണ്ണം 5 ഉം ആയതിനാലാണ്.

പ്രോഗ്രാം 3.2. സ്ട്രീം ഇൻപുട്ട്/ഔട്ട്പുട്ട് ഫങ്ഷനുകളുടെ പ്രവർത്തനം വിശദമാക്കുന്നതിന്

```
#include <iostream>
#include <cstring> //To use strlen() function
using namespace std;
int main()
{
   char ch, str[20];
   cout<<"Enter a character: ";
   cin.get(ch);  //To input a character to the variable ch
   cout<<"Enter a string: ";
   cin.getline(str,10,'.');      //To input the string
   cout<<"Entered character is:\t";
   cout.put(ch);             //To display the character
   cout.write("\nEntered string is:",20);
   cout.write(str,strlen(str));
   return 0;
}
```

പ്രോഗ്രാം 3.2 പ്രവർത്തിപ്പിക്കുമ്പോൾ താഴെ കാണുന്ന തരത്തിലുള്ള ഔട്ട്പുട്ട് ലഭ്യമാവുന്നതാണ്.

```
Enter a character: p
Enter a string: hello world
Entered character is:     p
```

```
Entered string is:
hello wo
```

ഈ പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുമ്പോൾ എന്താണ് സംഭവിക്കുന്നത് എന്ന് നോക്കാം. തുടക്കത്തിൽ തന്നെ `get()` ഫങ്ഷൻ `'p'` എന്ന കാരക്റ്ററിനെ സ്വീകരിക്കുന്നു. തുടർന്ന് `getline()` ഫങ്ഷൻ ഉപയോഗിച്ച് `"hello world"` എന്ന സ്ട്രിങ് ഇൻപുട്ട് ചെയ്യുന്നു. അതിന് ശേഷം `put()` ഫങ്ഷനുപയോഗിച്ച് `'p'` എന്ന കാരക്റ്റർ പ്രദർശിപ്പിക്കുന്നു. `write()` ഫങ്ഷൻ `"hello wo"` എന്ന് മാത്രമാണ് പ്രദർശിപ്പിക്കുന്നത്. `str` എന്ന അറേയിൽ ശേഖരിക്കാവുന്ന പാമാവധി കാരക്റ്ററുകളുടെ എണ്ണം 10 ആണെന്ന് ഫങ്ഷനിൽ സൂചിപ്പിച്ചിട്ടുമുണ്ട്. ഒരു ബൈറ്റ് നൾ കാരക്റ്ററിന് (`'\0'`– സ്ട്രിങ്ങിന്റെ അവസാന കാരക്റ്റർ) വേണ്ടി മാറ്റിവെക്കപ്പെട്ടതിനാൽ സാധാരണ 9 കാരക്റ്ററുകൾ മാത്രമേ ശേഖരിക്കുവാൻ കഴിയുകയുള്ളു. എന്നാൽ ഇവിടെ ഔട്ട്പുട്ട് ആയി വൈറ്റ് സ്പേസ് ഉൾപ്പെടെ 8 കാരക്റ്ററുകൾ മാത്രമാണ് കാണപ്പെടുന്നത്. ഇതിനു കാരണം `"p"` എന്ന കാരക്റ്ററിന് ശേഷം എന്റർ കീ ഉപയോഗിക്കുമ്പോൾ `'\n'`, `str` എന്ന അറേയുടെ ആദ്യത്തെ അംഗമായി ശേഖരിക്കപ്പെടുന്നതിനാലാണ്. അതുകൊണ്ട് `"hello wo"`എന്ന സ്ട്രിങ് പുതിയ വരിയിലാണ് പ്രദർശിപ്പിക്കപ്പെടുന്നത്.

ഈ പ്രോഗ്രാമിൽ `"hello.world"` എന്ന് ഇൻപുട്ട് ചെയ്ത് പ്രവർത്തിപ്പിച്ചാൽ താഴെ കാണുന്ന വിധത്തിലുള്ള ഔട്ട്പുട്ട് ലഭിക്കുന്നതാണ്.

```
Enter a character : a
Enter a string : hello.world
Entered character string is : a
Entered string is :
hello
```

ഈ മാറ്റം ഔട്ട്പുട്ടിൽ ഉണ്ടായതിന് കാരണം `getline()` എന്ന ഫങ്ഷൻ ഡോട്ട് ചിഹ്നത്തിന് `(dot operator)` മുമ്പുള്ള കാരക്റ്ററുകളെ മാത്രം സ്വീകരിച്ചതിനാലാണ്.

3.3.3 സ്ട്രിങ് ഫങ്ഷനുകൾ (String Functions)

സ്ട്രിങ്ങുകൾ കൈകാര്യം ചെയ്യുന്നതിന് ധാരാളം സ്ട്രിങ് ഫങ്ഷനുകൾ C++ ൽ ലഭ്യമാണ്. അധ്യായം 9 ൽ ചർച്ചചെയ്തത് പോലെ C++ൽ സ്ട്രിങ് ഡാറ്റാടൈപ്പ് ഇല്ലാത്തതിനാൽ സ്ട്രിങ് കൈകാര്യം ചെയ്യുന്നതിന് ക്യാരക്ടറുകളുടെ അറെ ആണ് ഉപയോഗിക്കുന്നത് അതുകൊണ്ട് തുടർന്നു വരുന്ന ചർച്ചകളിൽ സ്ട്രിങ് എന്ന പദം വരുമ്പോഴെല്ലാം അത് ഒരു ക്യാരക്ടർ അറെ ആണെന്ന് അനുമാനിക്കുക. സാധാരണയായി ഉപയോഗിക്കുന്ന സ്ട്രിങ് ഫങ്ഷനുകൾ താഴെ കൊടുക്കുന്നവയാണ്. ഈ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് നമ്മുടെ C++ പ്രോഗ്രാമിൽ `cstring` (ടർബോ C++ൽ `string.h`) എന്ന ഹെഡർഫയൽ ഉൾപ്പെടുത്തേണ്ടതുണ്ട്.

a. `strlen()`

ഒരു സ്ട്രിങ്ങിന്റെ നീളം കണ്ടുപിടിക്കുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. സ്ട്രിങ്ങിന്റെ നീളം കൊണ്ട് അർത്ഥമാക്കുന്നത് സ്ട്രിങ്ങിലെ അക്ഷരങ്ങളുടെ എണ്ണമാണ്

അതിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
int strlen(string);
```

ഈ ഫങ്ഷൻ ഒരു സ്ട്രിങ് ആർഗ്യുമെന്റായി സ്വീകരിക്കുകയും സ്ട്രിങ്ങിന്റെ നീളം തിരിച്ചു നൽകുകയും ചെയ്യുന്നു. താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് ശകലം ഇത് വിവരിക്കുന്നു.

```
char str[] = "Welcome";
int n;
n = strlen(str);
cout << n;
```

ഇവിടെ strlen() എന്ന ഫങ്ഷന് ഒരു സ്ട്രിങ് വേരിയബിളിനെ ആർഗ്യുമെന്റായി സ്വീകരിക്കുകയും അതിൽ അടങ്ങിയിരിക്കുന്ന സ്ട്രിങ്ങിലെ ക്യാരക്ടറുകളുടെ എണ്ണം, അതായത് 7 എന്ന വില n എന്ന വേരിയബിളിലേക്ക് തിരിച്ച് നൽകുന്നു. അതുകൊണ്ട് n എന്ന വേരിയബിളിന്റെ വിലയായി പ്രോഗ്രാം കോഡ് 7 പ്രദർശിപ്പിക്കുന്നു. അറെ ഡിക്ലറേഷൻ താഴെ കൊടുത്തിരിക്കുന്നത് പോലെ ആണെങ്കിലും ഔട്ട്പുട്ട് ഇത് തന്നെ ആയിരിക്കും.

```
char str [10]= "Welcome";
```

ഡിക്ലറേഷനിൽ അറെയുടെ വലിപ്പം കൊടുത്തിരിക്കുന്നത് ശ്രദ്ധിക്കുക. താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ ആർഗ്യുമെന്റ് ഒരു സ്ട്രിങ്ങ് സ്ഥിര വിലയും ആയേക്കാം.

```
n= strlen ("computer");
```

മുകളിലത്തെ നിർദേശം 8 എന്ന വില തിരിച്ച് നൽകുകയും അത് nൽ സംഭരിക്കുകയും ചെയ്യുന്നു.

b. strcpy()

ഒരു സ്ട്രിങ്ങിനെ മറ്റൊരു സ്ട്രിങ്ങിലേക്ക് പകർത്തുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഇതിന്റെ വാക്യ ഘടന താഴെ നൽകുന്നു.

```
strcpy(string1, string2);
```

ഈ ഫങ്ഷൻ string 2 നെ string 1 ലേക്ക് പകർത്തുന്നു. ഇവിടെ സ്ട്രിങ്ങ് 1 ഉം സ്ട്രിങ്ങ് 2 ഉം ക്യാരക്ടറുകളുടെ അറെ അല്ലെങ്കിൽ സ്ട്രിങ് സ്ഥിരാങ്കങ്ങൾ ആണ്. ഫങ്ഷന്റെ പ്രവർത്തനത്തിന് ആവശ്യമായ ആർഗ്യുമെന്റുകളാണ് ഇവ. താഴെ കൊടുക്കുന്ന കോഡ് ഇവയുടെ പ്രവർത്തനം വിശദമാക്കുന്നു.

```
char s1[10]. s2[10] = "Welcome";
strcpy (s1, s2);
cout << s1;
```

സ്ട്രിങ്ങ് വേരിയബിൾ s1 ൽ അടങ്ങിയിരിക്കുന്ന "Welcome" എന്ന സ്ട്രിങ്ങ് സക്രീനിൽ പ്രദർശിപ്പിക്കപ്പെടും. രണ്ടാമത്തെ ആർഗ്യുമെന്റ് ഒരു സ്ട്രിങ് സ്ഥിരാംഗമായി താഴെകൊടുക്കുന്നു.

```
char str [10];
strcpy (str. "Welcome");
```

ഇവിടെ "Welcome" എന്ന സ്ട്രിങ് സ്ഥിരാങ്കം വേരിയബിൾ str ൽ സംഭരിക്കും. str="Welcome" എന്ന അസൈൻമെന്റ് പ്രസ്താവന തെറ്റാണ്. എന്നാൽ ഒരു ക്യാരക്ടർ അറെയിലേക്ക് പ്രഖ്യാപന സമയത്ത്. വില നമുക്ക് നേരിട്ട് നൽകാവുന്നതാണ്.

```
char str [10] = "Welcome" ;
```

c. strcat()

ഒരു സ്ട്രിങ്ങിലേക്ക് മറ്റൊരു സ്ട്രിങ് കൂട്ടിച്ചേർക്കുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. തത്ഫലമായി ലഭിക്കുന്ന സ്ട്രിങ്ങിന്റെ നീളം രണ്ട് സ്ട്രിങ്ങിന്റെയും നീളത്തിന്റെ ആകെ തുക ആകുന്നു. ഫങ്ഷന്റെ വാക്യ ഘടന താഴെകൊടുത്തിരിക്കുന്നു.

```
strcat(string1, string2);
```

ഇവിടെ `string1, string2` എന്നിവ ക്യാരക്ടറുകളുടെ അറെയോ സ്ട്രിങ് സ്ഥിരാങ്കങ്ങളോ ആയിരിക്കും. `string2, string1` ലേക്ക് കൂട്ടിച്ചേർക്കുന്നു അതുകൊണ്ട് ആദ്യത്തെ ആർഗ്യുമെന്റിന്റെ വലിപ്പം രണ്ട് സ്ട്രിങ്ങുകൾ ഒരുമിച്ച് ഉൾക്കൊള്ളാൻ കഴിയുന്നതായിരിക്കണം. ഈ ഫങ്ഷന്റെ പ്രയോഗം കാണിക്കുന്ന ഒരു ഉദാഹരണം നമുക്ക് കാണാം. താഴെ കൊടുത്തിരിക്കുന്ന ഉദാഹരണം ശ്രദ്ധിക്കുക.

```
char s1[20] = "Welcome", s2[10] = " to C++";
strcat(s1,s2);
cout << s1;
```

മുകളിൽ കൊടുത്തിരിക്കുന്ന `C++` കോഡ് പ്രവർത്തിക്കുമ്പോൾ s1 ന്റെ വിലയായ `"Welcome to C++"` എന്ന ഔട്ട്പുട്ട് ലഭിക്കും. s2 എന്ന `string` വൈറ്റ് സ്പേസോടുകൂടിയാണ് ആരംഭിച്ചിരിക്കുന്നത് എന്നത് ശ്രദ്ധിക്കുക.

d. strcmp()

രണ്ട് സ്ട്രിങ്ങുകൾ തമ്മിൽ താരതമ്യം ചെയ്യുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഈ താരതമ്യത്തിൽ സ്ട്രിങ്ങുകളിലെ ക്യാരക്ടറുകളുടെ അക്ഷരമാലാക്രമം (ASCII വില) പരിഗണിക്കപ്പെടുന്നു. ഫങ്ഷന്റെ വാക്യ ഘടന താഴെ കൊടുത്തിരിക്കുന്നു.

```
strcmp(string1, string2)
```

മൂന്ന് വ്യത്യസ്ത സന്ദർഭങ്ങളിൽ ഫങ്ഷൻ താഴെ കൊടുത്തിരിക്കുന്ന ഏതെങ്കിലും വിലകൾ തിരിച്ച് നൽകുന്നു.

- `string1, string2` എന്നിവ ഒരേ പോലെ ആണെങ്കിൽ 0 തിരിച്ചു നൽകും.
- `string1` അക്ഷരമാലാക്രമത്തിൽ `string2` നേക്കാൾ ചെറുതാണെങ്കിൽ ഒരു നെഗറ്റീവ് വില തിരിച്ച് നൽകും.
- `string1` അക്ഷരമാലക്രമത്തിൽ `string2` നേക്കാൾ വലുതാണെങ്കിൽ ഒരു പോസിറ്റീവ് വില തിരികെ നൽകും.

താഴകൊടുത്തിരിക്കുന്ന കോഡ് ശകലം ഈ ഫങ്ഷന്റെ പ്രവർത്തനം കാണിക്കുന്നു.

```
char s1[]="Deepthi", s2[]="Divya";
int n;
n = strcmp(s1,s2);
if(n==0)
    cout<<"Both the strings are same";
else if(n < 0)
    cout<<"s1 < s2";
else
    cout<<"s1 > s2";
```

മുകളിൽ കൊടുത്തിരിക്കുന്ന കോഡ് ശകലം പ്രവർത്തിക്കുമ്പോൾ `s1 < s2` എന്ന ഔട്ട്പുട്ട് പ്രദർശിപ്പിക്കുമെന്ന് വ്യക്തമാണ്.

e. strcmpi()

വലിയ അക്ഷരങ്ങളോ ചെറിയ അക്ഷരങ്ങളോ പരിഗണിക്കാതെ രണ്ട് സ്ട്രിങ്ങുകൾ താരതമ്യം ചെയ്യുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. അതായത് ഈ ഫങ്ഷൻ അപ്പർകേയ്സ് ലോവർകേയ്സ് അക്ഷരങ്ങൾ താരതമ്യത്തിന് ഒരേപോലെ പരിഗണിക്കും. ഈ ഫങ്ഷന്റെ വാക്യഘടനയും പ്രവർത്തനരീതിയും `strcmp( )` പോലെ ആണെങ്കിലും ഇത് കെയ്സ് സെൻസിറ്റീവല്ല. ഈ ഫങ്ഷനും `strcmp( )`യെ പോലെ വിലകൾ തിരിച്ചു നൽകുന്നു താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് ശകലം പരിഗണിക്കുക.

```
char s1[]="SANIL", s2[]="sanil";
int n;
n = strcmpi(s1,s2);
if(n==0)
    cout<<"strings are same";
else if(n < 0)
    cout<<"s1 < s2";
else
    cout<<"s1 > s2";
```

മുകളിൽ കൊടുത്തിരിക്കുന്ന കോഡ് ശകലം ഒരു C++ പ്രോഗ്രാമിൽ പ്രവർത്തിക്കുമ്പോൾ ലഭിക്കുന്ന ഔട്ട്പുട്ട് "strings are same" എന്നായിരിക്കും, എന്തുകൊണ്ടെന്നാൽ `strcmpi()` ലോവർകേയ്സ് അക്ഷരങ്ങളേയും അപ്പർകേയ്സ് അക്ഷരങ്ങളേയും ഒരേ പോലെ കരുതുന്നു പ്രോഗ്രാം 3.3. രണ്ട് സ്ട്രിങ്ങുകൾ താരതമ്യം ചെയ്യുകയും കൂട്ടിച്ചേർക്കുകയും ചെയ്യുന്നു. പുതിയതായി രൂപപ്പെട്ട സ്ട്രിങ്ങിന്റെ നീളവും പ്രദർശിപ്പിക്കുന്നു.

പ്രോഗ്രാം 3.3 രണ്ട് സ്ട്രിങ്ങുകൾ വ്യത്യസ്തമാണെങ്കിൽ ഇവ രണ്ടും കൂട്ടിച്ചേർക്കുന്നതിനും അതിന്റെ നീളം കണ്ടുപിടിക്കുന്നതിനും.

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
char s1[15], s2[15], s3[30];
cout<<"Enter two strings: ";
cin>>s1>>s2;
int n=strcmp(s1, s2);
if (n==0)
    cout<<"\nThe input strings are same";
else
{
```

സ്ട്രിങ് കൈകാര്യം ചെയ്യുന്നതിനുള്ള ഫങ്ഷനുകൾ അടങ്ങിയ ഹെഡർ ഫയൽ

```
    cout<<"\nThe input strings are not same";
    strcpy(s3, s1);//Copies the string in s1 into s3
    strcat(s3, s2); //Appends the string in s2 to that in s3
    cout<<"String after concatenation is: "<<s3;
    cout<<"\nLength of the new string is: "<<strlen(s3);
}
return 0;
}
```

ഔട്ട്പുട്ടിന്റെ മാതൃക

```
Enter tow strings:india
kerala
The input strings are not same
String after concetenation is:indiakerala
Length of the new string is: 11
```

3.3.4 ഗണിത ഫങ്ഷനുകൾ (Mathematical Functions)

ഇനി നമുക്ക് C++ൽ ലഭ്യമായ ഗണിത ഫങ്ഷനുകളെക്കുറിച്ച് ചർച്ച ചെയ്യാം. പ്രോഗ്രാമിൽ ഈ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് Cmath (ടർബോ C++ ൽ math.h) എന്ന ഹെഡർ ഫയൽ നാം ഉപയോഗിക്കണം.

a. abs()

ഒരു പൂർണ്ണ സംഖ്യയുടെ (integer) അഖണ്ഡവില കണ്ടുപിടിക്കുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഇത് ഒരു പൂർണ്ണ സംഖ്യ (integer) ആർഗ്യൂമെന്റ് ആയി എടുത്ത് അതിന്റെ അഖണ്ഡവില തിരിച്ച് നൽകുന്നു. ഇതിന്റെ വാക്യഘടനയാണ്,

```
int abs(int)
```

ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്ന രീതി ഉദാഹരണമായി താഴെ നൽകുന്നു.

```
int n = -25;
cout << abs(n);
```

മുകളിൽ കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാം കോഡ് 25 എന്ന് പ്രദർശിപ്പിക്കും. ഒരു ദശാംശ സംഖ്യയുടെ കേവലവില (absolute value) കണ്ടുപിടിക്കണമെങ്കിൽ fabs() എന്ന ഫങ്ഷൻ മുകളിൽ കൊടുത്തിരിക്കുന്നത് പോലെ നമുക്ക് ഉപയോഗിക്കാം. അത് ദശാംശ സംഖ്യ തിരികെ നൽകുന്നു.

b. sqrt()

sqrt() ഒരു സംഖ്യയുടെ വർഗ്ഗമൂലം കണ്ടുപിടിക്കുന്നതിന് ഉപയോഗിക്കുന്നു. ഈ ഫങ്ഷന്റെ ആർഗ്യൂമെന്റിന്റെ ഡാറ്റ ഇനം int, float or double എന്നിവ ആകാം. പോസിറ്റീവ് ആർഗ്യൂമെന്റിന്റെ വർഗ്ഗമൂലം ഈ ഫങ്ഷൻ തിരിച്ച് നൽകുന്നു. ഇതിന്റെ വാക്യഘടനയാണ്.

```
double sqrt(double)
```

താഴെ കൊടുത്തിരിക്കുന്ന ഉദാഹരണം പരിശോധിക്കുക. ഇത് 5 എന്ന വില പ്രദർശിപ്പിക്കും.

```
int n = 25;
float b = sqrt(n);
cout << b;
```

c. pow()

ഒരു സംഖ്യയുടെ പവർ കണ്ടുപിടിക്കുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. x,y എന്നീ രണ്ട് ആർഗ്യുമെന്റ്കൾ ഇത് ഉപയോഗിക്കുന്നു x,y എന്നിവയുടെ ഡാറ്റാ ഇനം int, float അല്ലെങ്കിൽ double ആകുന്നു. ഈ ഫങ്ഷൻ x^y യുടെ ഫലം തിരിച്ചു നൽകുന്നു. ഇതിന്റെ വാക്യഘടനയാണ്.

```
double pow(double, int)
```

താഴെ കൊടുത്തിരിക്കുന്ന ഉദാഹരണം ഇതിന്റെ പ്രവർത്തനം വിവരിക്കുന്നു.
ഉദാഹരണം:

```
int x = 5, y = 4, z;
z = pow(x, y);
cout << z;
```

pow(x,y) രണ്ട് ആർഗ്യുമെന്റുകൾ ഉപയോഗിക്കുന്നു. മുകളിൽ കൊടുത്തിരിക്കുന്ന ഉദാഹരണത്തിൽ 625 എന്ന് പ്രദർശിപ്പിക്കും.

പ്രോഗ്രാം 3.4 ഒരു ത്രികോണത്തിന്റെ വൃത്തത്തിന്റെയും വിസ്തീർണം ഗണിത ഫങ്ഷനുകൾ ഉപയോഗിച്ച് കണ്ടുപിടിക്കുന്നത്

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    const float pi=22.0/7
    int a,b,c, radius;
    float s, area1, area 2;
    cout<<"Enter the three sides of the triangle:";
    cin>>a>>b>>c;
    s = (a+b+C)/2.0;
    area1 = sqrt (s*(s-a)*(s-b)*(s-c));
    cout<<"The Area of the Triangle is: "<<area1;
    cout<<"nEnter the radius of the circle: ";
    cin>>radius;
    area2 = pi*pow(radius,2);
    cout<<"Area of the Circle is: "<<area2;
    return 0:
}
```

ഗണിത ഫങ്ഷൻ ഉപയോഗിക്കാനാവശ്യമായ ഹെഡർ ഫയൽ

മുകളിൽ തന്നിരിക്കുന്ന പ്രോഗ്രാമിന്റെ ഔട്ട്പുട്ടിന്റെ മാതൃക

```
Enter the three sides of the triangle: 5   7   9
The Area of the Triangle is: 17.4123
Enter the radius of the circle: 2.5
Area of the Circle is: 12.5714
```

3.3.5 ക്യാരക്ടർ ഫങ്ഷനുകൾ (Character functions)

ക്യാരക്ടറുകളിൽ വിവിധ പ്രവർത്തനങ്ങൾ നടത്തുവാൻ C++ ൽ ലഭ്യമായ വിവിധ ക്യാരക്ടർ ഫങ്ഷനുകൾ താഴെ കൊടുത്തിരിക്കുന്നു. ഈ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് cctype (ടർബോ c++ ൽ ctype.h) എന്ന ഹെഡർ ഫയൽ പ്രോഗ്രാമിൽ കൂട്ടിച്ചേർക്കേണ്ടതുണ്ട്.

a. isupper()

തന്നിരിക്കുന്ന ക്യാരക്ടർ വലിയ അക്ഷരത്തിൽ (upper case) ഉള്ളതാണോ, അല്ലയോ എന്ന് പരിശോധിക്കുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഈ ഫങ്ഷന്റെ വാക്യഘടന ആണ്,

```
int isupper(char c)
```

തന്നിരിക്കുന്ന ക്യാരക്ടർ വലിയ അക്ഷരത്തിൽ (upper case) ആണെങ്കിൽ 1 ഉം അല്ലങ്കിൽ പൂജ്യവും ഈ ഫങ്ഷൻ തിരിച്ച് നൽകുന്നു. താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന പൂജ്യം എന്ന വില n ലേക്ക് നൽകുന്നു.

ഉദാഹരണം:

```
int n = isupper('x');
```

താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവനകൾ പരിഗണിക്കുക.

```
char c = 'A';
int n = isupper(c);
```

മുകളിലുള്ള പ്രസ്താവനകളുടെ പ്രവർത്തനത്തിന് ശേഷം n ന്റെ വില 1 ആയിരിക്കും എന്തുകൊണ്ടെന്നാൽ തന്നിരിക്കുന്ന ക്യാരക്ടർ വലിയ അക്ഷരത്തിൽ ഉള്ളതാണ്.

b. islower()

തന്നിരിക്കുന്ന ഒരു ക്യാരക്ടർ ചെറിയ അക്ഷരത്തിൽ (lower case) ഉള്ളതാണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഇതിന്റെ വാക്യഘടനയാണ്.

```
int islower(char c)
```

തന്നിരിക്കുന്ന ക്യാരക്ടർ ചെറിയ അക്ഷരത്തിലാണെങ്കിൽ 1 ഉം അല്ലെങ്കിൽ പൂജ്യവും ഈ ഫങ്ഷൻ തിരിച്ച് നൽകുന്നു. താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവനകൾ പ്രവർത്തിച്ചതിന് ശേഷം വേരിയബിൾ nന്റെ വില 1 ആയിരിക്കും. എന്തുകൊണ്ടെന്നാൽ തന്നിരിക്കുന്ന ക്യാരക്ടർ ചെറിയ അക്ഷരത്തിലുള്ളതാണ്.

ഉദാഹരണം:

```
char ch = 'x';
int n = islower(ch);
```

എന്നാൽ താഴെകൊടുത്തിരിക്കുന്ന പ്രസ്താവനയിൽ n ന്റെവില പൂജ്യമായിരിക്കും. കാരണം തന്നിരിക്കുന്ന ക്യാരക്ടർ അപ്പർ കേയ്സിലുള്ളതാണ്.

```
int n = islower('A');
```

c. isalpha()

തന്നിരിക്കുന്ന ക്യാരക്ടർ ഒരു അക്ഷരമാണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഇതിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
int isalpha(char c);
```

തന്നിരിക്കുന്ന ക്യാരക്ടർ ഒരു അക്ഷരമാണെങ്കിൽ 1 ഉം അല്ലെങ്കിൽ പൂജ്യവും ഈ ഫങ്ഷൻ തിരിച്ചു നൽകുന്നു.

താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന n ലേക്ക് പൂജ്യം എന്ന വില നൽകുന്നു. കാരണം തന്നിരിക്കുന്ന ക്യാരക്ടർ ഒരു അക്ഷരം അല്ല

```
int n = isalpha('3');
```

എന്നാൽ താഴെകൊടുത്തിരിക്കുന്ന പ്രസ്താവന 1 പ്രദർശിപ്പിക്കുന്നു കാരണം തന്നിരിക്കുന്ന ക്യാരക്ടർ ഒരു അക്ഷരമാണ്.

```
cout << isalpha('a');
```

d. isdigit()

തന്നിരിക്കുന്ന ക്യാരക്ടർ ഒരു അക്കം ആണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഇതിന്റെ വാക്യ ഘടന ആണ്

```
int isdigit(char c);
```

തന്നിരിക്കുന്ന ക്യാരക്ടർ അക്കമാണെങ്കിൽ ഈ ഫങ്ഷൻ 1 ഉം അല്ലെങ്കിൽ പൂജ്യവും തിരിച്ച് നൽകുന്നു. താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവനകൾ പ്രവർത്തിക്കുമ്പോൾ n എന്ന വേരിയബിളിന്റെ വില 1 ഉം ആയിരിക്കും. കാരണം തന്നിരിക്കുന്ന ക്യാരക്ടർ ഒരു അക്കം ആണ്.

```
n = isdigit('3');
```

താഴെ നൽകിയിരിക്കുന്ന പ്രസ്താവനകൾ പ്രവർത്തിക്കുമ്പോൾ n എന്ന വേരിയബിളിന്റെ വില പൂജ്യം ആയിരിക്കും കാരണം തന്നിരിക്കുന്ന ക്യാരക്ടർ ഒരു അക്കം അല്ല.

```
char c = 'b';
int n = isdigit(c);
```

e. isalnum()

തന്നിരിക്കുന്ന ക്യാരക്ടർ ഒരു അക്ഷരമോ അക്കമോ ആണോയെന്ന് ഈ ഫങ്ഷൻ പരിശോധിക്കുന്നു. ഇതിന്റെ വാക്യഘടനയാണ്

```
int isalnum (char c)
```

തന്നിരിക്കുന്ന ക്യാരക്ടർ അക്ഷരമോ അക്കമോ ആണെങ്കിൽ 1 ഉം അല്ലെങ്കിൽ പൂജ്യവും തിരിച്ചു നൽകുന്നു.

താഴെ കൊടുത്തിരിക്കുന്ന ഓരോ പ്രസ്താവനയും പ്രവർത്തിക്കുമ്പോൾ 1 തിരിച്ച് നൽകുന്നു.

```
n = isalnum('3');
cout << isalnum('A');
```

എന്നാൽ താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവനകൾ പ്രവർത്തിപ്പിക്കുമ്പോൾ പൂജ്യം എന്ന വില n ലേക്ക് നൽകുന്നു കാരണം തന്നിരിക്കുന്ന ക്യാരക്ടർ ഒരു അക്കമോ അക്ഷരമോ അല്ല.

```
char c = '-';
int n = isalnum(c);
```

f. toupper()

തന്നിരിക്കുന്ന ക്യാരക്ടർ വലിയ അക്ഷരത്തിലേക്ക് മാറ്റുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു ഇതിന്റെ വാക്യഘടന ആണ്

```
char toupper(char c)
```

തന്നിരിക്കുന്ന ക്യാരക്ടറിന്റെ വലിയ അക്ഷരം ഈ ഫങ്ഷൻ തിരിച്ച് നൽകുന്നു. തന്നിരിക്കുന്ന ക്യാരക്ടർ വലിയ അക്ഷരത്തിൽ ഉള്ളതാണെങ്കിൽ ഔട്ട് പുട്ടും അതുതന്നെ ആയിരിക്കും.

താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന 'A' യെ വേരിയബിൾ c യിലേക്ക് നൽകുന്നു.

```
char c = toupper('a');
```

എന്നാൽ താഴെ തന്നിരിക്കുന്ന പ്രസ്താവനയുടെ ഔട്ട്പുട്ട് 'A' തന്നെ ആയിരിക്കും.

```
cout << (char)toupper('A');
```

ഈ പ്രസ്താവനയിലെ (char) ഉപയോഗിച്ച് ഡാറ്റാ തരം മാറ്റിയിരിക്കുന്നത് ശ്രദ്ധിക്കുക. ഈ രീതി ഉപയോഗിച്ചില്ലെങ്കിൽ ഔട്ട്പുട്ട് A യുടെ ASCII വിലയായ 65 ആയിരിക്കും.

g. tolower()

തന്നിരിക്കുന്ന ക്യാരക്ട്റിനെ ചെറിയ അക്ഷരത്തിലേക്ക് മാറ്റുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഇതിന്റെ വാക്യഘടനയാണ്.

```
char tolower(char c)
```

തന്നിരിക്കുന്ന ക്യാരക്ടറിന്റെ ചെറിയ അക്ഷരം ഫങ്ഷൻ തിരിച്ചു നൽകുന്നു. തന്നിരിക്കുന്ന ക്യാരക്ടർ ചെറിയ അക്ഷരത്തിലുള്ളതാണെങ്കിൽ ഔട്ട്പുട്ടും അതുതന്നെ ആയിരിക്കും. ഈ പ്രസ്താവന പരിഗണിക്കുക.

```
c = tolower('A');
```

മുകളിലത്തെ സ്റ്റേറ്റ്മെന്റ് പ്രവർത്തിച്ചതിന്ശേഷം വേരിയബിൾ c യുടെ വില 'a' ആയിരിക്കും. എന്നാൽ താഴെ കൊടുത്തിരിക്കുന്ന സ്റ്റേറ്റ്മെന്റുകൾ പ്രവർത്തിക്കുമ്പോൾ വേരിയബിൾ 'c' യുടെ വില 'a' ആയിരിക്കും.

```
char x = 'a';
char c = tolower(x) ;
```

tolower(), toupper() എന്നീ ഫങ്ഷനുകളുടെ കാര്യത്തിൽ ആർഗ്യുമെന്റ് ഒരു അക്ഷരമല്ലെങ്കിൽ തന്നിരിക്കുന്ന ക്യാരക്ടർ തന്നെ തിരിച്ചു നൽകും.

പ്രോഗ്രാം 10.3 ൽ ക്യാരക്ടർ ഫങ്ഷനുകളുടെ ഉപയോഗം വിവരിച്ചിരിക്കുന്നു. ഈ പ്രോഗ്രാം ഒരു വാചകം സ്വീകരിക്കുകയും സ്ട്രിങ്ങിലെ ചെറിയ അക്ഷരങ്ങൾ, വലിയ

അക്ഷരങ്ങൾ, അക്കങ്ങൾ എന്നിവയുടെ എണ്ണം കണ്ടുപിടിക്കുകയും ചെയ്യുന്നു. ഇത് മൊത്തം സ്ട്രിങ്ങിനെ വലിയ അക്ഷരത്തിലും ചെറിയ അക്ഷരത്തിലും പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു.

പ്രോഗ്രാം 3.5 തന്നിരിക്കുന്ന സ്ട്രിങ്ങിലെ വിവിധ തരത്തിലുള്ള ക്യാരക്ടറുകൾ എണ്ണുന്നതിന്

```
#include <iostream>
#include <cstdio>
#include <cctype>
using namespace std;
int main()
{
char text[80];
int Ucase=0, Lcase=0, Digit=0,
cout << "Enter a line of text: ";
gets(text);
for(int i=0; text[i]!='\0'; i++)
    if (isupper(text[i])) Ucase++;
        else if (islower(text[i])) Lcase++;
            else if (isdigit(text[i])) Digit++;
cout << "\nNo. of uppercase letters = " << Ucase;
cout << "\nNo. of lowercase letters = " << Lcase;
cout << "\nNo. of digits = " << Digit;
cout << "\nThe string in uppercase form is\n";
i=0;
while (text[i]!='\0')
{
    putchar(toupper(text[i]));
    i++;
}
cout << "\nThe string in lowercase form is\n";
i=0;
do
{
    putchar(tolower(text[i]));
    i++;
} while(text[i]!='\0');
return 0;
}
```

text[1] എന്ന അറെയുടെ വില ശൂന്യ ക്യാരക്ടറിൽ എത്തുമ്പോൾ ലൂപ്പ് അവസാനിക്കും.

putchar() ന് പകരം cout<< ഉപയോഗിക്കുമ്പോൾ ക്യാരക്ടറിന്റെ ASCII വില പ്രദർശിപ്പിക്കും

ഒരു മാതൃകാ ഔട്ട്പുട്ട് താഴെ നൽകുന്നു.

```
Enter a line of text : The vehicle ID is KL01 AB101
No. of uppercase letters = 7
No. of lowercase letters = 11
No. of digits = 5
```

```
The string in uppercase form is
THE VEHICLE ID IS KL01 AB101
The string in lowercase form is
the vehicle id is kl01 ab101
```

നമുക്ക് ചെയ്യാം.

താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിൽ ഒരു ചാർച്ച് തയാറാക്കി അതിലെ നിരകളിൽ നാം ഇതുവരെ ചർച്ച ചെയ്ത എല്ലാ മുൻ നിർവചിത ഫങ്ഷനുകളും ചേർക്കുക

ഫങ്ഷൻ	ഉപയോഗം	വാക്യഘടന	ഉദാഹരണം	ഔട്ട്പുട്ട്

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. മോഡ്യുളാർ പ്രോഗ്രാമിങ് എന്നാൽ എന്ത്?
2. C++ ലെ ഒരു ഫങ്ഷൻ കൊണ്ട് അർഥമാക്കുന്നത് എന്താണ്?
3. ക്യാരക്ടർ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് ആവശ്യമായ ഹെഡർ ഫയലിന്റെ പേർ എഴുതുക.
4. cout<<sqrt(49); ന്റെ ഔട്ട്പുട്ട് എഴുതുക.
5. കൂട്ടത്തിൽ ചേരാത്തത് എടുത്തെഴുതി അതിനുള്ള കാരണം നൽകുക.
   (a) `strlen()` (b) `itoa()` (c) `strcpy()` (d) `strcat()`
6. `pow()` എന്ന ഫങ്ഷന് ആവശ്യമായ ഹെഡർ ഫയലിന്റെ പേരെഴുതുക.
7. `strcmpi()` എന്ന ഫങ്ഷൻ ഉപയോഗിച്ച് `"HELLO"` `"hello"` എന്ന സ്ട്രിങ്ങുകൾ താരതമ്യം ചെയ്യുമ്പോൾ ഉണ്ടാകുന്ന ഔട്ട്പുട്ട് എഴുതുക.
8. താഴെ കൊടുത്തിരിക്കുന്ന C++ സ്റ്റേറ്റ്മെന്റിന്റെ ഔട്ട്പുട്ട് എഴുതുക.
   `cout<<strlen("smoking kills");`
9. ഏത് ഫങ്ഷൻ ഉപയോഗിച്ചാണ് `'P'` എന്ന അക്ഷരത്തെ `'p'` ആക്കി മാറ്റുന്നത്.
10. ഫങ്ഷന്റെ ആർഗ്യുമെന്റ് ഒരു അക്ഷരമോ അല്ലെങ്കിൽ സംഖ്യയോ ആണോ എന്ന് പരിശോധിക്കുന്നതിന് വേണ്ട ഫങ്ഷന്റെ പേര് എഴുതുക.

3.4 ഉപയോക്തൃ നിർവചിത ഫങ്ഷനുകൾ (User defind functions)

നമ്മൾ ഇതുവരെ ചർച്ച ചെയ്ത എല്ലാ പ്രോഗ്രാമിലും `main()` എന്ന പേരിലുള്ള ഫങ്ഷൻ ഉണ്ട്. പ്രോഗ്രാമിന്റെ ആദ്യത്തെ വരി പ്രീ-പ്രോസസറിന് നിർദേശം നൽകുന്ന പ്രസ്താവനയാണ് എന്ന് നമുക്കറിയാം. യഥാർത്ഥത്തിൽ അതിന് ശേഷമുള്ളത് ഫങ്ഷന്റെ നിർവചനമാണ്. പ്രോഗ്രാമുകളിലെ `void main()` നെ ഫങ്ഷന്റെ ഫങ്ഷൻ ഹെഡർ (അല്ലെങ്കിൽ ഫങ്ഷൻ ഹെഡിങ്ങ്) എന്ന് വിളിക്കുന്നു. ഇതിനെ തുടർന്ന് { } എന്ന ആവരണങ്ങൾക്കുള്ളിൽ കൊടുത്തിരിക്കുന്ന പ്രസ്താവനകളെ ബോഡി എന്നു വിളിക്കുന്നു.

ഫങ്ഷൻ നിർവചനത്തിന്റെ വാക്യഘടന താഴെ കൊടുത്തിരിക്കുന്നതാണ്.

```
data_type function_name(argument_list)
{
    statements in the body;
}
```

ഡാറ്റ ഇനം എന്നത് C++ ലെ ഏതെങ്കിലും സാധുതയുള്ള ഡാറ്റ ഇനമാണ്. ഒരു ഉപയോക്തൃ നിർവഹിത പദം (ഐഡന്റിഫയർ) ആണ്. function_name ഐശ്ചികമായ പരാമീറ്ററുകളുടെ കൂട്ടമാണ് ആർഗ്യുമെന്റ് ലിസ്റ്റ്. ഡാറ്റ ഇനങ്ങളോട് കൂടിയ ഒരു കൂട്ടം വേരിയബിളുകളെ കോമ ഉപയോഗിച്ച് വേർതിരിക്കുന്നു. ഫങ്ഷന്റെ ചട്ടകൂട്ടിൽ ഉൾക്കൊള്ളിച്ചിരിക്കുന്ന C++ സ്റ്റേറ്റ്മെന്റുകൾ ഫങ്ഷന്റെ നിർവഹണത്തിന് ആവശ്യമാണ്. ഒരിക്കൽ ഒരു ഫങ്ഷൻ നിർമ്മിക്കാൻ നാം തീരുമാനിച്ചാൽ താഴെ കൊടുത്തിരിക്കുന്ന ചോദ്യങ്ങൾക്ക് ഉത്തരം നൽകേണ്ടതുണ്ട്.

(i) ഫങ്ഷൻ ഹെഡറിൽ ഏത് ഡാറ്റ ഇനം ഉപയോഗിക്കും?

(ii) എത്ര ആർഗ്യുമെന്റുകൾ ആവശ്യമുണ്ട്? അവ ഓരോന്നിന്റെയും ഡാറ്റ ഇനം എന്തായിരിക്കും?

getchar(), strcpy(), sqrt() എന്നീ മുൻ നിർവചിത ഫങ്ഷനുകൾ എങ്ങനെയാണ് ഉപയോഗിച്ചതെന്ന് നമുക്ക് അറിയാമല്ലോ. ഒരു C++ പ്രസ്താവനയിൽ ഈ ഫങ്ഷനുകൾ വിളിക്കുമ്പോൾ (ഉപയോഗിക്കുമ്പോൾ) അവ പ്രവർത്തിക്കുമെന്ന് നാം കണ്ടിട്ടുണ്ട്.

getchar() എന്ന ഫങ്ഷൻ ഒരു ആർഗ്യുമെന്റും ഉപയോഗിക്കുന്നില്ല. എന്നാൽ strcpy() പ്രവർത്തിക്കുന്നതിന് രണ്ട് സ്ട്രിങ് ആർഗ്യുമെന്റുകൾ വേണം ഈ ആർഗ്യുമെന്റുകൾ ഇല്ലാതെ ഈ ഫങ്ഷൻ പ്രവർത്തിക്കില്ല. അതിന് കാരണം ഇത് നിർവചിച്ചിരിക്കുന്നത്, രണ്ട് സ്ട്രിങ് (ക്യാരക്ടർ അറെ) ആർഗ്യുമെന്റ് ഉപയോഗിച്ചാണ്. എന്നാൽ sqrt() യ്ക്ക് ആർഗ്യുമെന്റായി ഒരു സംഖ്യ ആവശ്യമാണ്. അതിനോടൊപ്പം തന്നിരിക്കുന്ന ആർഗ്യുമെന്റിൽ മുൻകൂട്ടി തയ്യാറാക്കിയ പ്രവർത്തനം നടത്തിയ ശേഷം ഒരു ഫലം (ഡബിൾ ഡാറ്റാടൈപ്പ്) തിരികെ നൽകുന്നു. മുകളിൽ പരാമർശിച്ച ഫലത്തെ ഫങ്ഷന്റെ റിട്ടേൺ വാല്യൂ (return value) എന്ന് വിളിക്കുന്നു. ഈ വില ഫങ്ഷന്റെ റിട്ടേൺ വിലയെ ആശ്രയിച്ചിരിക്കുന്നു. മറ്റൊരു രീതിയിൽ പറഞ്ഞാൽ ഫങ്ഷന്റെ ഡാറ്റ ഇനത്തിന് അനുസരിച്ചുള്ള വില ആയിരിക്കണം ഫങ്ഷൻ തിരികെ നൽകേണ്ടത്.

അതുകൊണ്ട് ഫങ്ഷന്റെ ഡാറ്റ ഇനത്തെ ഫങ്ഷന്റെ റിട്ടേൺ ഇനം എന്നും പറയാറുണ്ട്. നാം main() ഫങ്ഷനിൽ return 0; എന്ന സ്റ്റേറ്റ്മെന്റ് ഉപയോഗിക്കുന്നത്, GCC യുടെ ആവശ്യാനുസരണം main()ന്റെ തിരികെ നൽകുന്ന വില int ഡാറ്റ ഇനം ആയി നിർവചിച്ചിരിക്കുന്നതിനാലാണ്.

ആർഗ്യുമെന്റുകളുടെ എണ്ണവും ഇനവും (data type) ഫങ്ഷന്റെ പ്രവർത്തനത്തിന് ആവശ്യമായ ഡാറ്റയെ ആശ്രയിച്ചിരിക്കുന്നു. എന്നാൽ setw(), gets() തുടങ്ങിയ ഫങ്ഷനുകൾ വിലകൾ ഒന്നും തിരിച്ച് നൽകുന്നില്ല. ഇത്തരം ഫങ്ഷനുകളുടെ ഹെഡറിൽ void എന്ന് റിട്ടേൺ ഇനമായി ഉപയോഗിക്കുന്നു. ഒരു ഫങ്ഷൻ ഒന്നുകിൽ ഒരു വില തിരിച്ചു നൽകും, അല്ലെങ്കിൽ ഒരു വിലയും തിരിച്ചു നൽകുന്നില്ല.

3.4.1. ഉപയോക്തൃ നിർമ്മിത ഫംങ്ഷനുകൾ നിർമിക്കുന്നു (Creating user defined functions)

മുകളിൽ ചർച്ചചെയ്ത വാക്യഘടനയെ അടിസ്ഥാനമാക്കി നമുക്ക് ഫങ്ഷനുകൾ നിർമിക്കാം. ഒരു സന്ദേശം പ്രദർശിപ്പിക്കാനുള്ള ഫങ്ഷൻ ആണ് താഴെ കൊടുത്തിരിക്കുന്നത്.

```
void saywelcome()
{
    cout<<"Welcome to the world of functions";
}
```

ഫങ്ഷന്റെ പേര് saywelcome() എന്നാണ്. ഇതിന്റെ ഡാറ്റ ഇനം (റിട്ടേൺ ടൈപ്പ്) വോയിഡ് (void) ആണ്. ഇതിന് ആർഗ്യുമെന്റുകൾ ഇല്ല. ഫങ്ഷൻ ചട്ടക്കൂട്ടിൽ ഒരു പ്രസ്താവന മാത്രമേ ഉള്ളൂ.

ഇപ്പോൾ നമുക്ക് രണ്ട് സംഖ്യകളുടെ തുക കണ്ടുപിടിക്കുന്നതിനുള്ള ഒരു ഫങ്ഷൻ നിർവ്വചിക്കാം. ഒരേ ഉദ്യമത്തിനായി നാല് വിവിധ തരത്തിലുള്ള ഫങ്ഷൻ നിർവചനങ്ങൾ നൽകുന്നു. എന്നാൽ അവയുടെ നിർവചന ശൈലികൾ വ്യത്യാസപ്പെട്ടിരിക്കുന്നതിനാൽ അവയിലോരോന്നിന്റെയും ഉപയോഗം മറ്റൊന്നിൽ നിന്നും വ്യത്യാസപ്പെട്ടിരിക്കുന്നു.

ഫങ്ഷൻ 1	*ഫങ്ഷൻ 2*

`void sum1()`
`{`
`   int a, b, s;`
`   cout<<"Enter 2 numbers: ";`
`   cin>>a>>b;`
`   s=a+b;`
`   cout<<"Sum="<<s;`
`}`
	`int sum2()`
	`{`
	`   int a, b, s;`
	`   cout<<"Enter 2 numbers: ";`
	`   cin>>a>>b;`
	`   s=a+b;`
	`   return s;`
	`}`

ഫങ്ഷൻ 3	*ഫങ്ഷൻ 4*

`void sum3(int a, int b)`
`{`
`   int s;`
`   s=a+b;`
`   cout<<"Sum="<<s;`
`}`
	`int sum4(int a, int b)`
	`{`
	`   int s;`
	`   s=a+b;`
	`   return s;`
	`}`

നമുക്ക് ഈ ഫങ്ഷനുകൾ വിശകലനം ചെയ്ത് അവ എങ്ങനെ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു എന്ന് നോക്കാം. എല്ലാ ഫങ്ഷനുകളുടേയും ഉദ്യമം ഒന്നു തന്നെയാണ്. എന്നാൽ പരാമീറ്ററുകളുടെ എണ്ണത്തിലും റിട്ടേൺ ഇനത്തിലും അവ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു. ടേബിൾ 10.1 ൽ കാണിച്ചിരിക്കുന്നത് പോലെ വോയിഡ് അല്ലാത്ത ഡാറ്റ ഇനം ഉപയോഗിച്ച് നിർവചിച്ചിരിക്കുന്ന ഫങ്ഷനുകൾ അതിന് ചേർന്ന തരത്തിലുള്ള വില തിരിച്ചു നൽകും. ഇതിന് വേണ്ടിയാണ് റിട്ടേൺ പ്രസ്താവന ഉപയോഗിച്ചിരിക്കുന്നത്. (ഫങ്ഷൻ 2ഉം, ഫ

ങ്ഷൻ 4ഉം പരിശോധിക്കുക). റിട്ടേൺ (return statement) പ്രസ്താവന വിളിച്ചിരിക്കുന്ന ഫങ്ഷനിലേക്ക് ഒരു വില തിരിച്ച് നൽകുന്നതിനോടൊപ്പം പ്രോഗ്രാം നിയന്ത്രണം തിരിച്ച് കൈമാറുകയും ചെയ്യുന്നു. അതുകൊണ്ട് ഒരു റിട്ടേൺ സ്റ്റേറ്റ്മെന്റ് പ്രവർത്തിച്ചു കഴിഞ്ഞാൽ ഫങ്ഷനിലെ പിന്നീട് വരുന്ന പ്രസ്താവനകൾ പ്രവർത്തിക്കുകയില്ല എന്നത് ഓർക്കുക.

പേര്	ആർഗ്യുമെന്റുകൾ	തിരിച്ച് നൽകുന്ന വില
sum1()	ആർഗ്യുമെന്റുകൾ ഇല്ല	ഒരു വിലയും തിരിച്ച് നൽകുന്നില്ല
sum2()	ആർഗ്യുമെന്റുകൾ ഇല്ല	പൂർണ്ണ സംഖ്യ തിരിച്ച് നൽകുന്നു
sum3()	രണ്ട് പൂർണ്ണ സംഖ്യകൾ	ഒരു വിലയും തിരിച്ച് നൽകുന്നില്ല
sum4()	രണ്ട് പൂർണ്ണ സംഖ്യകൾ	പൂർണ്ണ സംഖ്യ തിരിച്ച് നൽകുന്നു

പട്ടിക 3.1 : ഫങ്ഷനുകളുടെ വിശകലനം

ഒട്ടുമിക്ക ഫങ്ഷനുകളിലും റിട്ടേൺ പ്രസ്താവന ഫങ്ഷന്റെ അവസാനമാണ് നൽകുന്നത്. void ഡാറ്റ ഇനം ഉപയോഗിച്ച് നിർവചിച്ച ഫങ്ഷനുകളുടെ ചട്ടക്കൂടിനുള്ളിൽ റിട്ടേൺ പ്രസ്താവന ഉണ്ടായേക്കാം. എന്നാൽ അതിന് നമുക്ക് ഒരു വിലയും നൽകാൻ കഴിയില്ല. main() ഫങ്ഷന്റെ റിട്ടേൺ ഇനം ഒന്നുകിൽ void അല്ലെങ്കിൽ int ആണ്.

ഇനി നമുക്ക് ഈ ഫങ്ഷനുകൾ എങ്ങനെ വിളിക്കണമെന്നും അവ എങ്ങനെ പ്രവർത്തിക്കുമെന്നും നോക്കാം. main() ഫങ്ഷൻ ഒഴികെ മറ്റൊരു ഫങ്ഷനും സ്വയം പ്രവർത്തിക്കുക ഇല്ല എന്ന് നമുക്ക് അറിയാം. മുൻ നിർവചിതമോ ഉപയോക്തൃ നിർവചിതമോ ആയ മറ്റ് ഉപ ഫങ്ഷനുകൾ main() ഫങ്ഷനിലോ മറ്റ് ഉപയോക്തൃ നിർവചിത ഫങ്ഷനിലോ വിളിക്കുമ്പോൾ മാത്രമേ പ്രവർത്തിക്കൂ. താഴെ കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാമിൽ ചതുരത്തിനുള്ളിൽ നൽകിയിരിക്കുന്ന കോഡ്, ഫങ്ഷൻ വിളിക്കുന്നതിനെ സൂചിപ്പിക്കുന്നു. ഇവിടെ main() വിളിക്കുന്നു ഫങ്ഷനും sum1(), sum2(), sum3(), sum4() എന്നിവ വിളിക്കപ്പെട്ട ഫങ്ഷനുകളുമാണ്.

```
int main()
{
    int x, y, z=5, result;
    cout << "\nCalling the first function\n";
    sum1();
    cout << "\nCalling the second function\n";
    result = sum2();
    cout << "Sum given by function 2 is " << result;
    cout << "\nEnter values for x and y : ";
    cin >> x >> y;
    cout << "\nCalling the third function\n";
    sum3(x,y);
    cout << "\nCalling the fourth function\n";
    result = sum4(z,12);
    cout << "Sum given by function 4 is " << result;
```

```
    cout << "\nEnd of main function"
}
```

പ്രോഗ്രാമിന്റെ ഔട്ട്പുട്ട് ചുവടെ ചേർക്കുന്നു.

```
Calling the first function
Enter 2 numbers: 10  25
Sum=35
Calling the second function
Enter 2 numbers: 5   7
Sum given by function 2 is 12
Enter values for x and y : 8   13
Calling the third function
Sum=21
Calling the fourth function
Sum given by function 4 is 17
End of main function
```

sum1() ഫങ്ഷനിൽ a ക്കും b യുടേയും ഇൻപുട്ട്

sum2() ഫങ്ഷനിൽ a, b യുടേയും ഇൻപുട്ട്

main() ഫങ്ഷനിൽ x, y എന്നിവയുടേയും വേണ്ട ഇൻപുട്ട്

ഫങ്ഷൻ 4 ന് നൽകിയിരിക്കുന്ന ഉദ്യമത്തിന് രണ്ട് സംഖ്യകൾ ആവശ്യമായതിനാൽ രണ്ട് ആർഗ്യുമെന്റുകൾ നാം നൽകുന്നു. ഫങ്ഷൻ ചില കണക്കുകൂട്ടലുകൾ നിർവ്വഹിച്ച് ഒരു ഉത്തരം നൽകുന്നു. ഒരേ ഒരു ഉത്തരം മാത്രം ഉള്ളതിനാൽ അത് തിരിച്ച് നൽകാൻ കഴിയും. രണ്ട് സംഖ്യകളുടെ തുക കണ്ടുപിടിക്കുന്നതിന് ഈ ഫങ്ഷൻ താരതമ്യേന നല്ലതാണ്.

ഇനി നമുക്ക് രണ്ടു സംഖ്യകളുടെ ഗുണനഫലം കാണുന്നതിനുള്ളള C++ പ്രോഗ്രാം എഴുതി നോക്കാം. ഉപയോക്തൃ നിർവചിത ഫങ്ഷൻ ഉപയോഗിച്ചാണ് ഇത് എഴുതുന്നത്. എന്നാൽ C++ പ്രോഗ്രാമിൽ നാം എവിടെയാണ് ഉപയോക്തൃ നിർവചിത ഫങ്ഷൻ എഴുതുന്നത്? താഴെ കൊടുത്തിരിക്കുന്ന പട്ടിക ഉപയോക്തൃ നിർവചിത ഫങ്ഷൻ എഴുതുന്നതിനുള്ള രണ്ട് ശൈലികൾ കാണിക്കുന്നു.

പ്രോഗ്രാം 3.6 പെർഫക്ട് സംഖ്യ ആണോ എന്ന് പരിശോധിക്കുന്നു. പ്രോഗ്രാം 3.7

ഫങ്ഷൻ main() ന് മുൻപ്

```
#include<iostream>
using namespace std;
int product(int a, int b)
{
   int p;
   p = a * b;
   return p;
}
// Definition above main()
int main ()
```

ഫങ്ഷൻ main() ന് ശേഷം

```
#include<iostream>
using namespace std;
int main()
{
   int ans, num1, num2;
   cout<<"Enter 2 Number: ";
   cin>>num1>>num2;
   ans = product (nim1, num2);
   cout<<"Product = "<<ans;
   return 0;
}
```

``{
  int ans, num1, num2;
  count<<"Enter 2 Numbers :";
  cin>>num1>>num2;
  ans = product (num1, num2)
  count<<"Product = "<<ans;
  return 0;
}```
	``//Definition below main()
	int product(int a, int b)
	{ int p;
	  p = a * b;
	  return p;
	}```

പട്ടിക 3. 2 : ഫംങ്ഷനുകളുടെ വിശകലനം.

പ്രോഗ്രാം 3.6 കംപൈയിൽ ചെയ്യുമ്പോൾ അവിടെ ഒരു തെറ്റും ഉണ്ടാവില്ല. എന്നാൽ പ്രോഗ്രാം 3.7 കംപൈയിൽ ചെയ്യുമ്പോൾ അവിടെ ഒരു തെറ്റ് ഉണ്ടാകും 'product was not declared in this scope' എന്താണ് ഈ പിശക് അർത്ഥമാക്കുന്നത് എന്ന് നമുക്ക് നോക്കാം.

3.4.2 ഫങ്ഷനുകളുടെ പ്രോട്ടോടൈപ്പ് (Prototype of Functions)

ഒരു C++ പ്രോഗ്രാമിൽ എത്ര ഫങ്ഷനുകൾ വേണമെങ്കിലും ഉൾപ്പെടുത്താം എന്ന് നാം കണ്ടു കഴിഞ്ഞു. എന്നാൽ അതിന്റെ പ്രവർത്തനം തുടങ്ങുവാൻ ഒരു main() ഫങ്ഷൻ ഉണ്ടായിരിക്കണം. ഫങ്ഷനുകളുടെ നിർവചനങ്ങൾ നാം ആഗ്രഹിക്കുന്ന രീതിയിൽ ഏത് ക്രമത്തിലും എഴുതാം. നമുക്ക് ആദ്യം തന്നെ main() ഫങ്ഷൻ നിർവചിക്കുകയും മറ്റെല്ലാം ഫങ്ഷനുകളും അതിന് ശേഷമോ മുമ്പോ നൽകാവുന്നതാണ്. പ്രോഗ്രാം 3.6ൽ main() ഫങ്ഷൻ മറ്റ് എല്ലാ ഉപയോക്തൃ നിർമ്മിത ഫങ്ഷനുകൾക്ക് ശേഷമാണ് നൽകിയിരിക്കുന്നത്. എന്നാൽ പ്രോഗ്രാം 3.7 ൽ main() ഫങ്ഷൻ മറ്റ് എല്ലാ ഫങ്ഷനുകൾക്കും മുമ്പാണ് നിർവചിച്ചിരിക്കുന്നത്. നാം ഈ പ്രോഗ്രാം കംപൈൽ ചെയ്യുമ്പോൾ അത് ഒരു തെറ്റ് ചൂണ്ടിക്കാണിക്കും. "product was not declared in this scope" ഇത് എന്തുകൊണ്ടെന്നാൽ product() എന്ന ഫങ്ഷൻ വിളിച്ചിരിക്കുന്നത് അതിന്റെ നിർവചനത്തിന് മുമ്പാണ്. main() ഫങ്ഷന്റെ കംപൈലേഷൻ സമയത്ത് കംപൈലർ product() എന്ന ഫങ്ഷൻ വിളി (function call) എത്തുമ്പോൾ അതിന് അങ്ങനെ ഒരു ഫങ്ഷനെ കുറിച്ച് അറിവില്ല. അങ്ങനെ ഒരു ഫങ്ഷൻ ഉണ്ടോ എന്നും അതിന്റെ പ്രയോഗരീതി ശരിയാണോ അല്ലയോ എന്നും കംപൈലറിന് പരിശോധിക്കാൻ സാധിക്കില്ല. അങ്ങനെ അത് ഫങ്ഷൻ പ്രോട്ടോടൈപ്പിന്റെ അഭാവം മൂലമുള്ള ഒരു തെറ്റ് ചൂണ്ടിക്കാണിക്കുന്നു. ഒരു ഫങ്ഷൻ പ്രോട്ടോടൈപ്പ് എന്നത് ഒരു ഫങ്ഷന്റെ പ്രഖ്യാപനം ആണെന്നും അതിലൂടെ ഫങ്ഷന്റെ പേര്, അതിന്റെ റിട്ടേൺ ഇനം, ആർഗ്യുമെന്റുകളുടെ എണ്ണവും ഇനവും അതിന്റെ പ്രാപ്യത എന്നിവ കമ്പയിലറിന് ലഭ്യമാകുന്നു. ഈ വിവരങ്ങൾ പ്രോഗ്രാമിലെ ഫങ്ഷൻ കാൾ ശരിയാണോ എന്ന് പരിശോധിക്കുന്നതിന് അത്യാവശ്യമാണ്. ഈ വിവരങ്ങൾ ഫങ്ഷൻ ഹെഡറിൽ ലഭ്യമാണ്. അതിനാൽ ഫങ്ഷൻ ഹെഡർ (ഫങ്ഷൻ പ്രോട്ടോടൈപ്പ്) ഫങ്ഷനെ വിളിയ്ക്കുന്നതിന് മുമ്പായി പ്രസ്താവനയായി എഴുതാം. ഇതിന്റെ ഘടന താഴെ കൊടുത്തിരിക്കുന്നു.

```
data_type function_name(argument_list);
```

പ്രോട്ടോടൈപ്പിൽ ആർഗ്യുമെന്റിന്റെ പേരുകൾ നൽകേണ്ടതില്ല. അതുകൊണ്ട് താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന `main()` ഫങ്ഷനിലെ ഫങ്ഷൻ വിളിയ്ക്ക് മുൻപ് കൂട്ടി ച്ചേർത്ത് പ്രോഗ്രാം 3.6 ലെ തെറ്റ് തിരുത്തണം.

```
int product(int, int);
```

ഒരു വേരിയബിൾ പ്രഖ്യാപിക്കുന്നത് പോലെ ഫങ്ഷനും അത് പ്രോഗ്രാമിൽ ഉപയോഗി ക്കുന്നതിന് മുൻപേ പ്രഖ്യാപിച്ചിരിക്കണം. പ്രോഗ്രാമിൽ ഒരു ഫങ്ഷൻ അത് ഉപയോഗി ക്കുന്നതിന് മുൻപേ നിർവ്വചിച്ചിട്ടുണ്ടെങ്കിൽ ഫങ്ഷൻ പ്രഖ്യാപനം പ്രത്യേകമായി നട ത്തേണ്ടതില്ല. പ്രഖ്യാപന പ്രസ്താവന `main()` ഫങ്ഷന് പുറത്തും നൽകാവുന്നതാണ്. പ്രോട്ടോടൈപ്പിന്റെ സ്ഥാനം ഫങ്ഷന്റെ പ്രാപ്യതക്കനുസരിച്ച് വ്യത്യസപ്പെട്ടിരിക്കുന്നു. നമുക്ക് ഇത് ഈ അധ്യായത്തിൽ പിന്നീടുള്ള ഭാഗത്ത് ചർച്ച ചെയ്യാം. ഫങ്ഷൻ നിർവ ചനത്തിന്റെ സ്ഥാനം എവിടെ ആയിരുന്നാലും പ്രോഗ്രാമിന്റെ പ്രവർത്തനം `main()` ഫങ്ഷനിൽ നിന്ന് ആരംഭിക്കും.

3.4.3 ഫങ്ഷനുകളുടെ ആർഗ്യുമെന്റുകൾ (Arguments of Functions)

ഫങ്ഷനിലേക്ക് ഡാറ്റ ലഭിക്കുന്നതിനായി ആർഗ്യുമെന്റുകൾ അല്ലെങ്കിൽ പാരാമീറ്ററു കൾ ഉപയോഗിക്കാമെന്ന് നാം കണ്ടു. ഫങ്ഷൻ കാളിൽ ആർഗ്യുമെന്റുകളുടെ പ്രാധാന്യം എന്താണെന്ന് നമുക്ക് നോക്കാം. താഴെ കൊടുത്തിരിക്കുന്ന ഫങ്ഷൻ പരിഗണിക്കുക.

```
float SimpleInterest(long P, int N, float R)
{
    float amt;
    amt = P * N * R / 100;
    return amt;
}
```

ഈ ഫങ്ഷൻ തന്നിരിക്കുന്ന മുതൽ പലിശനിരക്ക്, കാലം എന്നിവ ഉപയോഗിച്ച് സാധാ രണ പലിശ കണക്കാക്കുന്നു.

താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് ഭാഗം വിവിധ ഫങ്ഷൻ വിളികൾ വിവരിക്കുന്നു.

```
cout << SimpleInterest(1000,3,2);//Function call 1
int x, y; float z=3.5, a;
cin >> x >> y;
a = SimpleInterest(x, y, z); //Function call 2
```

ആദ്യത്തെ പ്രസ്താവന പ്രവർത്തിക്കുമ്പോൾ 1000, 3, 2 എന്നീ വിലകൾ ഫങ്ഷൻ നിർവചനത്തിലെ ആർഗ്യൂമെന്റ് ലിസ്റ്റിലേക്ക് അയക്കുന്നു. ആർഗ്യൂമെന്റുകളായ `P,N, R` എന്നിവയ്ക്ക് യഥാക്രമം 1000, 3, 2 എന്നീ വിലകൾ ലഭിക്കുന്നു. അതേപോലെ അവസാ നത്തെ പ്രസ്താവന പ്രവർത്തിക്കുമ്പോൾ `x,y,z` എന്നീ വേരിയബിളുകളുടെ വിലകൾ യഥാക്രമം ആർഗ്യൂമെന്റുകളായ `P,N, R` എന്നിവയിലേക്ക് അയക്കുന്നു.

`x, y, z` എന്നീ വേരിയബിളുകളെ ആക്ചൽ ആർഗ്യൂമെന്റുകൾ അല്ലെങ്കിൽ യഥാർത്ഥ പരാമീറ്ററുകൾ എന്ന് വിളിക്കുന്നു. കാരണം പ്രവർത്തനത്തിനായി ഫങ്ഷനിലേക്ക് അയക്കുന്ന യഥാർത്ഥ ഡാറ്റയാണിവ. ഫങ്ഷൻ ഹെഡറിൽ ഉപയോഗിക്കുന്ന `P,N,R`

എന്നീ വേരിയബിളുകൾ ഫോർമൽ ആർഗ്യുമെന്റുകൾ അല്ലെങ്കിൽ ഫോർമൽ പാരാമീറ്ററുകൾ എന്ന് അറിയപ്പെടുന്നു. വിളിക്കുന്ന ഫങ്ഷനിൽ നിന്നും അയക്കുന്ന ഡാറ്റ സ്വീകരിക്കാൻ വേണ്ടിയാണ് ഈ ആർഗ്യുമെന്റുകൾ ഉപയോഗിച്ചിരിക്കുന്നത്. വിളിച്ച ഫങ്ഷനിൽ നിന്നും വിളിക്കപ്പെട്ട ഫങ്ഷനിലേക്ക് വിലകൾ അയക്കുന്നതിനുള്ള ഉപാധിയാണ് **ആർഗ്യുമെന്റുകൾ** അല്ലെങ്കിൽ **പരാമീറ്ററുകൾ**. ഫങ്ഷൻ നിർവചനത്തിൽ ആർഗ്യുമെന്റുകളായി ഉപയോഗിച്ച വേരിയബിളുകൾ ഫോർമൽ ആർഗ്യുമെന്റുകൾ എന്നറിയപ്പെടുന്നു. ഫങ്ഷൻ കോളിൽ ഉപയോഗിച്ച സ്ഥിരവിലകൾ, വേരിയബിളുകൾ അല്ലെങ്കിൽ പദപ്രയോഗങ്ങൾ എന്നിവ യഥാർത്ഥ ആർഗ്യുമെന്റുകൾ എന്ന് അറിയപ്പെടുന്നു. ഫങ്ഷൻ പ്രോട്ടോടൈപ്പിൽ വേരിയബിളുകൾ ഉപയോഗിക്കുകയാണെങ്കിൽ അവ ഡമ്മി ആർഗ്യുമെന്റുകൾ എന്ന പേരിൽ അറിയപ്പെടുന്നു.

ഇപ്പോൾ നമുക്ക് fact() എന്ന ഫങ്ഷൻ ഉപയോഗിച്ച് ഒരു സംഖ്യയുടെ ഫാക്ടോറിയൽ കണ്ടുപിടിക്കുകയും അത് nCr ന്റെ വില കാണുന്നതിനായി ഉപയോഗിക്കുകയും ചെയ്യാം (പ്രോഗ്രാം 10.7) നമുക്ക് അറിയാവുന്നതു പോലെ N എന്ന സംഖ്യയുടെ ഫാക്ടോറിയൽ ആദ്യത്തെ N എണ്ണൽ സംഖ്യകളുടെ ഗുണനഫലമാകുന്നു. nCr ന്റെ വില $\frac{n!}{r!(n-r)!}$ എന്ന സൂത്രവാക്യം ഉപയോഗിച്ച് കണ്ടുപിടിക്കുന്നു. ഇവിടെ n! സൂചിപ്പിക്കുന്നത് n എന്ന സംഖ്യയുടെ ഫാക്ടോറിയലാണ്.

പ്രോഗ്രാം 3.8 : nCr ന്റെ വില കണ്ടു പിടിക്കുന്നതിന്

```
#include<iostream>
using namespace std;
int fact(int); // Function prototype
int main()
{
    int n,r; ncr;
    cout<<"Enter the values of n and r : ";
    cin>>n>>r;
    ncr=fact(n)/(fact(r)*fact(n-r));
    cout<<n<<"C"<<r<<" = "<<ncr;
    return 0;
}
int fact(int N)  // Function header
{
    int f;
    for(f=1; N>0; N--)
        f=f*N;
    return f;
}
```

യഥാർത്ഥ ആർഗ്യുമെന്റുകൾ

സൂത്രവാക്യത്തിന് അനുസൃതമായ ഫങ്ഷൻ വിളി

യഥാക്രമ ആർഗ്യുമെന്റുകൾ

ഫാക്ടോറിയൽ തിരിച്ച് നൽകുന്നു

താഴെ കൊടുത്തിരിക്കുന്നത് ഒരു മാതൃക ഔട്ട്പുട്ട് ആണ്.

```
Enter the values of n and r : 5    2
5C2 = 10
```

ഉപയോക്താവ് നൽകുന്ന ഇൻപുട്ടുകൾ

3.4.4. തനതു ആർഗ്യൂമെന്റുകളോട് കൂടിയ ഫങ്ഷനുകൾ (Functions with default arguments)

നമുക്ക് താഴെ പറയുന്ന ആർഗ്യൂമെന്റ് പട്ടികയോട് കൂടിയ TimeSec() എന്ന ഒരു ഫങ്ഷൻ പരിഗണിക്കാം. ഈ ഫങ്ഷൻ സമയത്തെ പ്രതിനിധീകരിക്കുന്ന മണിക്കൂറുകൾ, മിനിട്ട്, സെക്കന്റ് എന്നിവക്കുള്ള മൂന്ന് സംഖ്യകൾ സ്വീകരിക്കുന്നു.

```
long TimeSec(int H, int M=0, int S=0)
{
    long sec = H * 3600 + M * 60 + S;
    return sec;
}
SecTime()
```

തന്നിരിക്കുന്ന സമയത്തെ ഫങ്ഷൻ സെക്കന്റുകളിലേക്ക് മാറ്റുന്നു. M, S എന്നീ ആർഗ്യൂമെന്റുകൾക്ക് തനതു വിലയായി പൂജ്യം നൽകിയിരിക്കുന്നത് ശ്രദ്ധിക്കുക. അതുകൊണ്ട് ഈ ഫങ്ഷൻ താഴെ പറയുന്ന രീതികളിൽ വിളിക്കാം.

```
long s1 = TimeSec(2,10,40);
long s2 = TimeSec(1,30);
long s3 = TimeSec(3);
```

ആർഗ്യൂമെന്റുകളുടെ പട്ടികയിലെ തനതുവില നൽകുന്ന എല്ലാ ആർഗ്യൂമെന്റുകളും വലത്തു നിന്ന് ഇടത്തോട്ട് നൽകണം എന്നത് പ്രധാനമായും ശ്രദ്ധിക്കേണ്ടതാണ്. ഒരു ഫങ്ഷൻ വിളിക്കുമ്പോൾ യഥാർത്ഥ ആർഗ്യൂമെന്റുകൾ (actual arguments) ഫോർമൽ ആർഗ്യുമെന്റുകളിലേക്ക് ഇടത് ഭാഗം മുതൽ നൽകുന്നു.

ആദ്യത്തെ പ്രസ്താവന പ്രവർത്തിച്ചപ്പോൾ 2,10,40 എന്നീ വിലകൾ യഥാക്രമം ഫോർമൽ പരാമീറ്ററുകളായ H, M, S എന്നിവയിലേക്ക് ഫങ്ഷൻ വിളിയോടൊപ്പം അയക്കുന്നു. M, S എന്നിവയുടെ തനതു വിലകളെ യഥാർത്ഥ ആർഗ്യൂമെന്റുകൾ തിരുത്തി എഴുതുന്നു. രണ്ടാമത്തെ പ്രസ്താവനയിലുള്ള ഫങ്ഷൻ വിളിക്കുമ്പോൾ H നും M നും യഥാർത്ഥ ആർഗ്യൂമെന്റുകളുടെ വിലകൾ കിട്ടുമ്പോൾ S അതിന്റെ തനതു വിലയായ പൂജ്യത്തിൽ (0) പ്രവർത്തിക്കുന്നു. അതുപോലെ മൂന്നാമത്തെ പ്രസ്താവന പ്രവർത്തിക്കുമ്പോൾ H ന് വിളിച്ചിരിക്കുന്ന ഫങ്ഷനിൽ നിന്നു വിലകിട്ടുന്നു, എന്നാൽ M ഉം S ഉം അവയുടെ തനത് വില ഉപയോഗിക്കുന്നു. അതുകൊണ്ട് ഫങ്ഷൻ വിളികൾക്ക് ശേഷം s1, s2, s3 എന്നിവയുടെ വിലകൾ യഥാക്രമം 7840, 5400, 10800 എന്നിങ്ങനെ ആയിരിക്കും.

ഫങ്ഷനുകളുടെ ആർഗ്യൂമെന്റുകൾക്ക് തനത് വില നൽകി നിർവ്വചിക്കാൻ സാധിക്കും എന്ന് നാം കണ്ടു കഴിഞ്ഞു. തനത് വില നൽകിയ ആർഗ്യുമെന്റുകളെ തനത് (ഡിഫാൾട്ട്) ആർഗ്യൂമെന്റുകൾ എന്നു വിളിക്കുന്നു. *ഇത് ഒരു ഫങ്ഷന് വ്യത്യസ്ത എണ്ണം ആർഗ്യുമെന്റുകൾ കൊണ്ട് വിളിക്കുന്നതിന് ഒരു പ്രോഗ്രാമറെ അനുവദിക്കുന്നു.* അതായത് തനത് ആർഗ്യുമെന്റുകൾക്ക് വിലകൾ നൽകിയോ നൽകാതെയോ ഫങ്ഷനെ വിളിക്കാൻ കഴിയും.

3.4.5 ഫങ്ഷൻ വിളിക്കുന്നതിനുള്ള വിവിധ രീതികൾ (Methods of calling Functions)

നിങ്ങളുടെ ടീച്ചർ ക്ലാസിലെ എല്ലാ വിദ്യാർത്ഥികളുടേയും രക്ഷകർത്താക്കളെ നിങ്ങളുടെ വിദ്യാലയത്തിൽ ഒരു പരിപാടിയിലേക്ക് ക്ഷണിക്കുന്നതിനുള്ള ഒരു കത്ത് തയ്യാറാക്കുവാൻ നിങ്ങളോട് ആവശ്യപ്പെട്ടു എന്ന് കരുതുക. കത്തിന്റെ മാതൃകയും രക്ഷാകർത്താക്കളുടെ പേര് അടങ്ങുന്ന പട്ടികയും നൽകാൻ ടീച്ചർക്ക് കഴിയും. പേരുകളുടെ പട്ടിക രണ്ട് വിധത്തിൽ നൽകാവുന്നതാണ്. ഒന്നുകിൽ യഥാർത്ഥ പട്ടിക അല്ലെങ്കിൽ അതിന്റെ ഫോട്ടോകോപ്പി. ഈ രണ്ട് രീതികളിൽ പേരിന്റെ പട്ടിക വ്യത്യാസം എന്താണ്? ടീച്ചർ യഥാർത്ഥ പട്ടികയാണ് നിങ്ങൾക്ക് നൽകുന്നതെങ്കിൽ, അതിൽ മറ്റൊന്നും എഴുതാതെയും രേഖപ്പെടുത്താതെയും ശ്രദ്ധാപൂർവ്വം ഉപയോഗിക്കണം. എന്തുകൊണ്ടെന്നാൽ ടീച്ചർക്ക് അതേ പട്ടിക തന്നെ ഭാവിയിൽ ആവശ്യമായി വരാം. എന്നാൽ ഫോട്ടോകോപ്പിയാണ് നിങ്ങൾക്ക് ലഭിക്കുന്നതെങ്കിൽ അതിൽ എഴുതുവാനോ രേഖപ്പെടുത്തുവാനോ നിങ്ങൾക്ക് സാധിക്കും. എന്തുകൊണ്ടെന്നാൽ ഫോട്ടോകോപ്പിയിൽ വരുത്തുന്ന വ്യത്യാസങ്ങൾ യഥാർത്ഥ പട്ടികയെ ബാധിക്കുന്നില്ല.

ക്ഷണക്കത്ത് തയ്യാറാക്കുന്ന ജോലി ഒരു ഫങ്ഷനായി നമുക്ക് പരിഗണിക്കാം. പേര് അടങ്ങുന്ന പട്ടിക ഫങ്ഷൻ ആർഗ്യുമെന്റ് ആണ്. ആർഗ്യൂമെന്റ് ഫങ്ഷനിലേക്ക് രണ്ട് രീതിയിൽ അയയ്ക്കാൻ സാധിക്കും. ആദ്യത്തേത് പേര് അടങ്ങുന്ന പട്ടികയുടെ കോപ്പി കൈമാറുക, മറ്റേത്, യഥാർത്ഥ പട്ടികതന്നെ കൈമാറുക. ക്ഷണക്കത്ത് തയ്യാറാക്കുമ്പോൾ യഥാർത്ഥ പട്ടിക തന്നെയാണ് കൈമാറുന്നതെങ്കിൽ പട്ടികയിൽ ഉണ്ടാക്കുന്ന ഏത് മാറ്റവും യഥാർത്ഥ പട്ടികയെ തന്നെ ബാധിക്കും. അതുപോലെ, C++ ലും രണ്ട് രീതിയിൽ ഫങ്ഷനിലേക്ക് ആർഗ്യൂമെന്റുകൾ അയയ്ക്കാം. ആർഗ്യുമെന്റുകൾ അയയ്ക്കുന്ന രീതിയെ അവലംബമാക്കി ഫങ്ഷൻ വിളിക്കുന്ന രീതിയെ കാൾ-ബൈ-വാല്യൂ രീതിയെന്നും കാൾ ബൈ റഫറൻസ് രീതിയെന്നും തരംതിരിക്കാം. ആർഗ്യൂമെന്റ് കൈമാറ്റ രീതികൾ വിശദമായി താഴെ വിവരിച്ചിരിക്കുന്നു.

a. കാൾ-ബൈ-വാല്യൂ (പാസ് ബൈ വാല്യൂ) രീതി (Call by value Method)

ഈ രീതിയിൽ, ആക്ചൽ ആർഗ്യൂമെന്റുകളിൽ അടങ്ങിയ വിലകൾ ഫോർമൽ ആർഗ്യൂമെന്റുകളിലേക്ക് (Formal argument) അയയ്ക്കുന്നു. മറ്റൊരു വിധത്തിൽ പറഞ്ഞാൽ ആക്ചൽ ആർഗ്യൂമെന്റിന്റെ ഒരു പകർപ്പ് ഫങ്ഷനിലേക്ക് അയയ്ക്കുന്നു. അതിനാൽ ഫങ്ഷനകത്ത് യഥാക്രമ ആർഗ്യൂമെന്റ് പുതുക്കപ്പെട്ടാലും, വിളിക്കുന്ന ഫങ്ഷനിലെ ആക്ചൽ ആർഗ്യൂമെന്റിൽ വ്യത്യാസം പ്രതിഫലിക്കുന്നില്ല. മുൻപ് ചർച്ച ചെയ്ത എല്ലാ ഫങ്ഷനുകളിലും ആർഗ്യൂമെന്റുകളുടെ വിലയാണ് അയച്ചത്. താഴെ കൊടുത്തിരിക്കുന്ന ഉദാഹരണം കാണുക:

```
void change(int n)
{
    n = n + 1;
    cout << "n = " << n << '\n';
}
int main()
    {
    int x = 20;
    change(x);
```

change () ഫങ്ഷനിലെ n പരാമീറ്ററിന് 20 എന്ന വില ശേഖരിക്കാൻ അതിന്റെ സ്വന്തം മെമ്മറി സ്ഥലം ഉണ്ട്.

x ന്റെ വില change () ഫങ്ഷനിലെ n ലേക്ക് കൈമാറ്റം ചെയ്യപ്പെടുന്നു.

```
    cout << "x = " << x;
    }
```

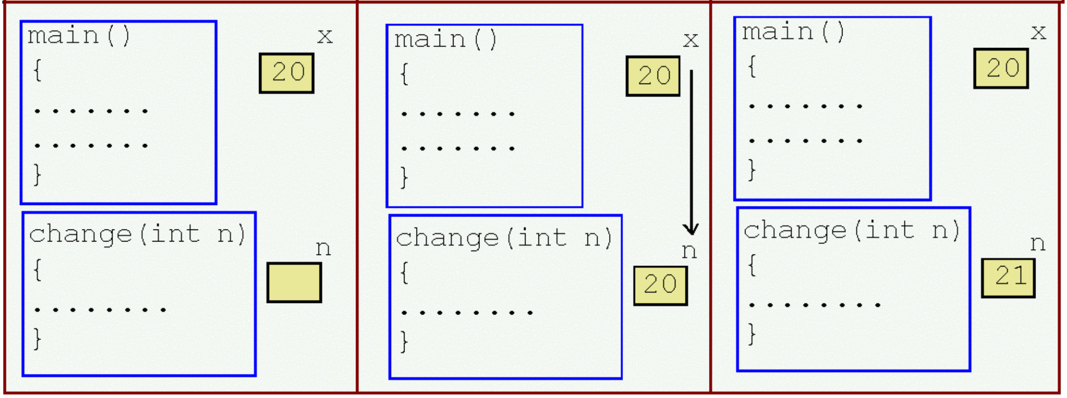
മുകളിലത്തെ പ്രോഗ്രാം ഭാഗത്ത് പരാമർശിച്ചിരിക്കുന്നതു പോലെ, നാം ഒരു ആർഗ്യൂമെന്റ് കൈമാറ്റം ചെയ്യുമ്പോൾ വേരിയബിൾ x ന്റെ ഒരു പകർപ്പ് ഫങ്ഷനിലേക്ക് കൈമാറ്റം ചെയ്യപ്പെടുന്നു.

മറ്റൊരു വിധത്തിൽ പറഞ്ഞാൽ x എന്ന വേരിയബിളിന്റെ വിലമാത്രമേ ഫങ്ഷനിലേക്ക് അയയ്ക്കുന്നുള്ളൂ. അതിനാൽ ഫങ്ഷനിലെ ഫോർമൽ പരാമീറ്ററിന് 20 എന്ന വില ലഭിക്കും. നാം n ന്റെ വില കൂട്ടുകയാണെങ്കിൽ, അത് x എന്ന വേരിയബിളിന്റെ വിലയെ ബാധിക്കുന്നില്ല. ഈ കോഡിന്റെ ഔട്ട്പുട്ട് താഴെ കൊടുത്തിരിക്കുന്നു.

```
n = 21
x = 20
```

ഒരു ഫങ്ഷൻ, കാൾ-ബൈ-വാല്യൂ രീതിയിൽ വിളിച്ചാൽ ആർഗ്യൂമെന്റുകൾക്ക് എന്ത് സംഭവിക്കുമെന്ന് പട്ടിക 3.2 ൽ കാണിക്കുന്നു.

ഫങ്ഷൻ വിളിക്ക് മുൻപ്	ഫങ്ഷൻ വിളിക്ക് ശേഷം	ഫങ്ഷന്റെ പ്രവർത്തനത്തിന് ശേഷം



പട്ടിക 3.2 കാൾ ബൈ വാല്യൂ പ്രവർത്തനം

b. കാൾ ബൈ റഫറൻസ് (പാസ്സ് ബൈ റഫറൻസ്) രീതി (Call by Reference Method)

ഒരു ആർഗ്യൂമെന്റ് റഫറൻസായി അയക്കുമ്പോൾ യഥാർത്ഥ ആർഗ്യൂമെന്റിന്റെ റഫറൻസ് (അഡ്രസ്സ്) ഫങ്ഷനിലേക്ക് അയക്കുന്നു. ഇതിന്റെ ഫലമായി യഥാർത്ഥ ആർഗ്യൂമെന്റിന് അനുവദിച്ച മെമ്മറിസ്ഥലം യഥാക്രമ ആർഗ്യൂമെന്റും കൂടി പങ്കിടും. അതുകൊണ്ട് വിളിച്ച ഫങ്ഷനിലെ യഥാക്രമ ആർഗ്യൂമെന്റിന് എന്തെങ്കിലും മാറ്റം സംഭവിച്ചാൽ ആ മാറ്റം ഫങ്ഷനിലെ യഥാർത്ഥ ആർഗ്യൂമെന്റിലും പ്രതിഫലിപ്പിക്കും. C++ ൽ ആർഗ്യൂമെന്റുകൾ റഫറൻസായി അയക്കുന്നതിന് യഥാക്രമ പരാമീറ്ററായി റഫറൻസ് വേരിയബിൾ ഉപയോഗിക്കുന്നു. ഒരു റഫറൻസ് വേരിയബിൾ മറ്റൊരു വേരിയബിളിന്റെ അപരനാമമാണ്. ഫങ്ഷൻ ഹെഡറിലെ ഡാറ്റ ഇനത്തിനും വേരിയബിളിനും ഇടയിൽ ഒരു ആമ്പർസാന്റ് ചിഹ്നം (&) ഉപയോഗിക്കുന്നു. മറ്റ് വേരിയബിളുകളെപ്പോലെ റഫറൻസ് വേരിയബി

ളുകൾക്ക് പ്രത്യേകമായ മെമ്മറിസ്ഥലം അനുവദിക്കുന്നില്ല. പകരം യഥാർത്ഥ ആർഗ്യുമെന്റിന് അനുവദിച്ച മെമ്മറി സ്ഥലം പങ്കിടും. താഴെ കൊടുത്തിരിക്കുന്ന ഫങ്ഷന്റെ യഥാക്രമം പരാമീറ്ററായി റഫറൻസ് വേരിയബിൾ ഉപയോഗിക്കുന്നു. അതുകൊണ്ട് ഫങ്ഷൻ വിളിക്കായി കാൾബൈ റഫറൻസ് രീതി പ്രാവർത്തികമാക്കുന്നു.

```
void change(int & n)
{
    n = n + 1;
    cout << "n = " << n << '\n';
}
int main()
{
    int x=20;
    change(x);
    cout << "x = " << x;
}
```

> n എന്ന പരാമീറ്റർ ഒരു റഫറൻസ് വേരിയബിൾ ആകുന്നു. അതിനാൽ അതിന് പ്രത്യേകമായ മെമ്മറി അനുവദിക്കുന്നില്ല.

> x ന്റെ റഫറൻസ് change () ഫങ്ഷനിലെ n ലേക്ക് കൈമാറ്റം ചെയ്യപ്പെടും. തൽഫലമായി മെമ്മറി പങ്കുവയ്ക്കുന്നു.

`change()` ലെ ഫങ്ഷൻ ഹെഡറിന് മാത്രമേ മാറ്റം ഉള്ളു എന്നത് ശ്രദ്ധിക്കുക. പരാമീറ്റർ `n` ന്റെ പ്രഖ്യാപനത്തിലെ & ചിഹ്നം അർത്ഥമാക്കുന്നത് ഒരു റഫറൻസ് വേരിബിളാണ് അത് എന്നാണ്. അതുകൊണ്ട് റഫറൻസ് അയച്ചു കൊണ്ട് ഫങ്ഷൻ വിളിക്കപ്പെടും. തന്മൂലം `x` എന്ന നെ `change()` ഫങ്ഷനിലേക്ക് അയക്കുമ്പോൾ `n` ന് `x` ന്റെ അഡ്രസ് കിട്ടുന്നതിനാൽ അവ മെമ്മറിസ്ഥലം പങ്കിടും. മറ്റൊരു തരത്തിൽ പറഞ്ഞാൽ വേരിയബിളുകളായ `n` ഉം `x` ഉം ഒരേ മെമ്മറിസ്ഥലം പരാമർശിക്കുന്നു. നാം `main()` ഫങ്ഷനിൽ `x` എന്ന പേരും `change()` ഫങ്ഷനിൽ `n` എന്ന പേരും ഉപയോഗിച്ച് ഒരേ മെമ്മറിസ്ഥലം പരാമർശിക്കുന്നു. അതുകൊണ്ട് `n` ന്റെ വിലയിൽ വ്യത്യാസം വരുത്തുമ്പോൾ യഥാർത്ഥത്തിൽ `x` വിലയിലാണ് മാറ്റം ഉണ്ടാകുന്നത്. മുകളിലുള്ള പ്രോഗ്രാം പ്രവർത്തിക്കുമ്പോൾ നമുക്ക് താഴെ കൊടുത്തിരിക്കുന്ന ഔട്ട്പുട്ട് ലഭിക്കും.

```
n = 21
x = 21
```

ഫങ്ഷൻ വിളിക്ക് വേണ്ടി കാൾ-ബൈ-റഫറൻസ് ഉപയോഗിക്കുമ്പോൾ ആർഗ്യുമെന്റുകൾക്ക് ഉണ്ടാകുന്ന മാറ്റങ്ങൾ ടേബിൾ 3.3 ൽ വർണ്ണിക്കുന്നു.

ഫങ്ഷൻ വിളിക്ക് മുമ്പ്	ഫങ്ഷൻ വിളിക്ക് ശേഷം	ഫങ്ഷൻ പ്രവർത്തിച്ചതിന് ശേഷം

main()
{
....
....
}
x: 20
change(int &n)
{
....
}
	main()
	{
	....
	....
	}
	x, n: 20
	change(int &n)
	{
	....
	}
		main()
		{
		....
		....
		}
		x, n: 21
		change(int &n)
		{
		....
		}

പട്ടിക 3.3. കാൾ ബൈ റഫറൻസ് പ്രവർത്തനം

ഫങ്ഷൻ വിളിയുടെ രണ്ട് രീതികൾ നമ്മൾ ചർച്ച ചെയ്തു. അടിസ്ഥാനപരമായി ഇവ വ്യത്യാസപ്പെടുന്നത് ആർഗ്യുമെന്റ് അയയ്ക്കുന്ന രീതിയിലാണ്. ഫങ്ഷൻ വിളിയുടെ ഈ രണ്ട് രീതികൾ തമ്മിലുള്ള വ്യത്യാസം പട്ടിക 3.4 ൽ കാണിച്ചിരിക്കുന്നു.

കാൾ ബൈ വാല്യു രീതി	കാൾ ബൈ റഫറൻസ് രീതി
• യഥാക്രമ പരാമീറ്ററുകളായി സാധാരണ വേരിയബിളുകൾ ഉപയോഗിക്കുന്നു.	• യഥാക്രമ പരാമീറ്ററുകളിൽ റഫറൻസ് വേരിയബിളുകൾ ഉപയോഗിക്കുന്നു.
• സ്ഥിരവിലകൾ, വേരിയബിളുകൾ, അല്ലെങ്കിൽ പദപ്രയോഗങ്ങൾ എന്നിവ യഥാർത്ഥ പരാമീറ്ററുകൾ ആയി ഉപയോഗിക്കാം.	• വേരിയബിളുകൾ മാത്രമേ യഥാർത്ഥ പരാമീറ്ററുകൾ ആകൂ.
• യഥാക്രമ പരാമീറ്ററുകളിൽ ഉണ്ടാക്കുന്ന വ്യത്യാസങ്ങൾ യഥാർത്ഥ പരാമീറ്ററുകളിൽ പ്രതിഫലിക്കുന്നില്ല.	• യഥാക്രമ പരാമീറ്ററിന് ഉണ്ടാക്കുന്ന വ്യത്യാസങ്ങൾ യഥാർത്ഥ പരാമീറ്ററുകളിൽ പ്രതിഫലിക്കും.
• യഥാക്രമ ആർഗ്യുമെന്റുകൾക്ക് പ്രത്യേകം മെമ്മറി ആവശ്യമുണ്ട്.	• യഥാർത്ഥ ആർഗ്യുമെന്റുകളുടെ മെമ്മറി യഥാക്രമ ആർഗ്യുമെന്റുകൾ പങ്കിടുന്നു.

പട്ടിക 3.4. കാൾ ബൈ വാല്യൂ v/s കാൾ ബൈ റഫറൻസ്

കാൾ ബൈ റഫറൻസ് രീതി വിശദമാക്കുന്നതിന് അനുയോജ്യമായ ഒരു ഉദാഹരണം നമുക്ക് ചർച്ച ചെയ്യാം. ഈ പ്രോഗ്രാം main() ഫങ്ഷനിലെ രണ്ട് വേരിയബിളുകളുടെ വിലകൾ പരസ്പരം കൈമാറുന്നതിന് കാൾ ബൈ റഫറൻസ് രീതിയിൽ വിളിക്കാൻ കഴിയുന്ന ഒരു ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. രണ്ട് വേരിയബിളുകളുടെ വിലകൾ പരസ്പരം കൈമാറുന്ന പ്രക്രിയയെ സ്വാപ്പിങ്ങ് എന്ന് പറയുന്നു.

പ്രോഗ്രാം 3.9 രണ്ട് വേരിയബിളുകളുടെ വിലകൾ പരസ്പരം കൈമാറുന്നതിന്.

```
#include <iostream>
using namespace std;
void swap(int & x, int & y)
{
    int t = x;
    x = y;
    y = t;
}
int main()
{
    int m, n;
    m = 10;
    n = 20;
    cout<<"Before swapping m= "<< m <<" and n= "<<n;
    swap(m, n);
    cout<<"\nAfter swapping m= "<< m <<" and n= "<<n;
    return 0;
}
```

നമുക്ക് പ്രോഗ്രാം 3.9 ലെ സ്റ്റേറ്റ്മെന്റുകളിലൂടെ കടന്നു പോകാം. ആക്ച്വൽ ആർഗ്യുമെന്റുകളായ m ഉം n ഉം ഫങ്ഷനിലേക്ക് റഫൻസ് ആയി അയക്കുന്നു. swap() ഫങ്

ഷന് ഉള്ളിൽ x ന്റെയും y യുടേയും വിലകൾ പരസ്പരം കൈമാറ്റം ചെയ്യപ്പെടുന്നു. അപ്പോൾ യഥാർത്ഥത്തിൽ m ലും n ലുമാണ് മാറ്റം നടക്കുന്നത്. അതിനാൽ മുകളിലത്തെ പ്രോഗ്രാം കോഡിന്റെ ഔട്ട്പുട്ട്,

```
Before swapping m = 10 and n= 20
After swapping m = 20 and n= 10
```

യഥാക്രമ ആർഗ്യുമെന്റിന് പകരം സാധാരണ വേരിയബിൾ ഉപയോഗിച്ച് മുകളിലെ പ്രോഗ്രാമിൽ മാറ്റം വരുത്തി അതിന്റെ ഔട്ട്പുട്ട് പ്രവചിക്കുക. കമ്പ്യൂട്ടർ ലാബിൽ ഈ കോഡ് പ്രവർത്തിപ്പിച്ച് നിങ്ങളുടെ ഉത്തരം പരിശോധിക്കുക.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. C++ പ്രോഗ്രാമുകളിലെ ഏറ്റവും ഒഴിച്ചു കൂടാനാവാത്ത ഫങ്ഷൻ ഏതെന്ന് തിരിച്ചറിയുക.
2. ഒരു ഫങ്ഷൻ ഹെഡറിന്റെ മൂന്ന് ഘടകങ്ങൾ പട്ടികപ്പെടുത്തുക.
3. ഫങ്ഷൻ പ്രോട്ടോടൈപ്പ് എന്നാൽ എന്ത്?
4. വിളിക്കുന്ന ഫങ്ഷനിൽ നിന്നും വിളിച്ച ഫങ്ഷനിലേക്ക് ഡാറ്റാ അയക്കുന്നതിന് ഏത് ഘടകമാണ് ഉപയോഗിക്കുന്നത്.
5. C++ ൽ ഉപയോഗിക്കുന്ന രണ്ട് പരാമീറ്റർ കൈമാറ്റരീതികൾ ഏതൊക്കെയാണ്?
6. ഫോർമൽ ആർഗ്യുമെന്റുകൾക്ക് ഒപ്പം '&' എന്ന ചിഹ്നം ഉപയോഗിക്കുന്ന ഫങ്ഷൻ വിളി (function call) യുടെ പേര്

3.5 വേരിയബിളുകളുടേയും ഫങ്ഷനുകളുടേയും വ്യാപ്തിയും ജീവനവും (Scope and life of variables and functions)

ഒന്നിലധികം ഫങ്ഷനുകൾ അടങ്ങിയ C++ പ്രോഗ്രാം നാം ചർച്ച ചെയ്തു. മുൻ നിർവചിത ഫങ്ഷനുകൾ അതിന് അനുബന്ധമായ ഹെഡർ ഫയലുകൾ ഉൾപ്പെടുത്തിയാണ് പ്രോഗ്രാമിൽ ഉപയോഗിക്കുന്നത്. ഉപയോക്തൃ നിർമ്മിത ഫങ്ഷനുകൾ main() ഫങ്ഷന് മുമ്പോ ശേഷമോ നിർവ്വചിക്കപ്പെടുന്നു. ഫങ്ഷനുകൾ നിർവ്വചിക്കുമ്പോൾ ഫങ്ഷൻ പ്രോട്ടോടൈപ്പിനുള്ള പ്രസക്തി നാം കണ്ടു കഴിഞ്ഞു. വേരിയബിളുകൾ ഫങ്ഷൻ ചട്ടക്കൂടിലും ആർഗ്യുമെന്റുകളായും നാം ഉപയോഗിച്ചു. ഇനി നമുക്ക് വേരിയബിളുകളുടേയും ഫങ്ഷനുകളുടേയും പ്രോഗ്രാമിലുള്ള ലഭ്യതയും പ്രാപ്യതയും ചർച്ച ചെയ്യാം. ഒരു പ്രോഗ്രാമിൽ ലോക്കൽ വേരിയബിളുകളുടെ പ്രാപ്യത പ്രോഗ്രാം 3.10 വിവരിക്കുന്നു.

പ്രോഗ്രാം 3.10. വേരിയബിളുകളുടെ വ്യാപ്തിയും ജീവനവും വിവരിക്കുന്നതിന്.

```
#include <iostream>
using namespace std;
int cube(int n)
{
    int cb;
    cout<< "The value of x passed to n is " << x;
    cb = n * n * n;
    return cb;
}
int main()
```

ഇത് തെറ്റ് ആകുന്നു.
എന്തുകൊണ്ടെന്നാൽ വേരിയബിൾ x, main() ഫങ്ഷനിലാണ് പ്രഖ്യാപിച്ചത്. അതുകൊണ്ട് മറ്റ് ഫങ്ഷനിൽ അത് ഉപയോഗിക്കാൻ കഴിയില്ല.

```
{
    int x, result;
    cout << "Enter a number : ";
    cin >> x;
    result = cube(x);
    cout << "Cube = " << result;
    cout << "\nCube = "<<cb;
}
```

ഇത് തെറ്റ് ആകുന്നു എന്തുകൊണ്ടെന്നാൽ cb എന്ന വേരിയബിൾ cube() എന്ന ഫങ്ഷനിലാണ് പ്രഖ്യാപിച്ചത്. അതുകൊണ്ട് മറ്റ് ഫങ്ഷനിൽ അത് ഉപയോഗിക്കാൻ കഴിയില്ല.

നാം പ്രോഗ്രാം കമ്പയിൽ ചെയ്യുമ്പോൾ, കോളൗട്ടിൽ കാണിച്ചിരിക്കുന്ന കാരണങ്ങളാൽ രണ്ട് തെറ്റുകൾ ഉണ്ടാകും. വേരിയബിളുകളുടേയും ഫങ്ഷനുകളുടേയും ലഭ്യത, പ്രാപ്യത എന്നീ ആശയത്തെ വ്യാപ്തി, ജീവനം, എന്നീ പദങ്ങൾ കൊണ്ട് സൂചിപ്പിക്കുന്നു. പ്രോഗ്രാമിന്റെ ഏത് ഭാഗത്താണോ ഒരു വേരിയബിൾ ഉപയോഗിക്കുന്നത് അതാണ് അതിന്റെ വ്യാപ്തി (Scope) മുകളിലത്തെ പ്രോഗ്രാമിൽ വേരിയബിൾ cb യുടെ വ്യാപ്തി ഫങ്ഷൻ cube() ൽ ആകുന്നു എന്തുകൊണ്ടെന്നാൽ അത് ആ ഫങ്ഷനിലാണ് പ്രഖ്യാപിച്ചത്. അതിനാൽ ഈ വേരിയബിൾ ആ ഫങ്ഷന് പുറത്ത് ഉപയോഗിക്കാൻ കഴിയില്ല. ഈ വ്യാപ്തി **ലോക്കൽ വ്യാപ്തി** എന്ന് അറിയപ്പെടുന്നു. ഒരു ഫങ്ഷന്റെ പ്രവർത്തനം പൂർത്തീകരിക്കുമ്പോൾ ആ ഫങ്ഷന്റെ ഉള്ളിലെ എല്ലാ വേരിയബിളുകൾക്കും അനുവദിച്ച മെമ്മറി സ്വതന്ത്രമാക്കുന്നു. മറ്റൊരു തരത്തിൽ പറയുമ്പോൾ ഒരു ഫങ്ഷന് ഉള്ളിൽ പ്രഖ്യാപിച്ച വേരിയബിളുകളുടെ സമയം ആ ഫങ്ഷനിലെ അവസാനത്തെ നിർദേശം പ്രവർത്തിക്കുന്നതോട് കൂടി അവസാനിക്കുന്നു. അതുകൊണ്ട് main() ഫങ്ഷനിൽ n എന്ന വേരിയബിൾ ഉപയോഗിക്കുമ്പോൾ അതിൽ നിന്നും വിളിക്കുന്ന ഫങ്ഷനിലെ n എന്ന ആർഗ്യുമെന്റിൽ നിന്നും വിളിക്കപ്പെട്ട ഫങ്ഷനിലെ വേരിയബിൾ nൽ നിന്നും അത് വ്യത്യസ്തമായിരിക്കും. ഒരു ഫങ്ഷന് ഉള്ളിൽ പ്രഖ്യാപിച്ച വേരിയബിളുകൾക്കും യഥാക്രമ പരാമീറ്ററുകൾക്കും ലോക്കൽ വ്യാപ്തി ഉണ്ടായിരിക്കും.

വേരിയബിളിനെ പോലെ തന്നെ ഫങ്ഷനുകൾക്കും വ്യാപ്തി ഉണ്ട്. എവിടെ ആണോ ഒരു ഫങ്ഷൻ പ്രഖ്യാപിച്ചിരിക്കുന്നത് അവിടെയാണ് ആ ഫങ്ഷൻ ഉപയോഗിക്കാൻ കഴിയുക. അതായത് ഫങ്ഷന് ലോക്കൽ വ്യാപ്തി ഉണ്ടെന്ന് പറയാം. അത് main() ഫങ്ഷന് മുമ്പേ മറ്റ് ഫങ്ഷനുകൾക്ക് പുറമെയോ പ്രഖ്യാപിച്ചാൽ ആ ഫങ്ഷന്റെ വ്യാപ്തി പ്രോഗ്രാം മുഴുവൻ ആയിരിക്കും. അതായത് പ്രോഗ്രാമിലെ ഏത് സ്ഥലത്തും ഫങ്ഷൻ ഉപയോഗിക്കാൻ കഴിയും. ഈ വ്യാപ്തി **ഗ്ലോബൽ വ്യാപ്തി** എന്ന് അറിയപ്പെടുന്നു. വേരിയബിളുകളും ഇപ്രകാരം ഗ്ലോബൽ വ്യാപ്തിയിൽ പ്രഖ്യാപിക്കാൻ കഴിയും. അങ്ങനെയുള്ള പ്രഖ്യാപനങ്ങൾ പ്രോഗ്രാമിലെ എല്ലാ ഫങ്ഷനുകൾക്കും പുറത്ത് ആയിരിക്കും. ഒരു പ്രോഗ്രാമിലെ വേരിയബിളുകളുടേയും ഫങ്ഷനുകളുടേയും വ്യാപ്തിയും, ജീവനവും സംബന്ധിച്ച് കൂടുതൽ വ്യക്തത കിട്ടുവാൻ പ്രോഗ്രാം 3.11 നോക്കുക.

പ്രോഗ്രാം 3.11 വേരിയബിളുകളുടേയും ഫങ്ഷനുകളുടേയും വ്യാപ്തിയും ജീവനവും വിവരിക്കുന്നതിന്

```
#include <iostream>
using namespace std;
int cb;          //global variable
void test()//global function since defined above other functions
{
    int cube(int n); //It is a local function
```

```
    cb=cube(x); //Invalid call. x is local to main()
    cout<<cb;
}
int main() // beginning of main() function
{
    int x=5; //local variable
    test();  //valid call since test() is a global function
    cb=cube(x); //Invalid call. cube() is local to test()
    cout<<cb;
}
int cube(int n)//Argument n is local variable
{
    int val= n*n*n; //val is local variable
    return val;
}
```

തന്നിരിക്കുന്ന കാൾ ഔട്ടുകൾ ഫങ്ഷനുകളുടെ വ്യാപ്തിയും ജീവനവും വിശദമാക്കുന്നു. ഒരു ഫങ്ഷൻ മറ്റൊരു ഫങ്ഷന്റെ ചട്ടക്കൂടിനുള്ളിൽ പ്രഖ്യാപിക്കുകയാണെങ്കിൽ അതിനെ ലോക്കൽ ഫങ്ഷൻ എന്ന് വിളിക്കുന്നു. അത് ആ ഫങ്ഷന് ഉള്ളിൽ മാത്രമേ ഉപയോഗിക്കാൻ കഴിയൂ. ഒരു ഫങ്ഷൻ മറ്റെല്ലാ ഫങ്ഷനുകളുടേയും ചട്ടക്കൂടിന് പുറമെ പ്രഖ്യാപിച്ചാൽ അതിനെ ഗ്ലോബൽ ഫങ്ഷൻ എന്ന് വിളിക്കുന്നു. ഒരു ഗ്ലോബൽ ഫങ്ഷൻ പ്രോഗ്രാമിൽ ഉടനീളം ഉപയോഗിക്കാൻ കഴിയും. മറ്റൊരു തരത്തിൽ പറഞ്ഞാൽ ഒരു ഗ്ലോബൽ ഫങ്ഷന്റെ വ്യാപ്തി പ്രോഗ്രാം മുഴുവനാകുമ്പോൾ ലോക്കൽ ഫങ്ഷന്റെ വ്യാപ്തി അത് പ്രഖ്യാപിച്ചിരിക്കുന്ന ഫങ്ഷനിൽ മാത്രം ആയിരിക്കും. വേരിയബിളുകളുടേയും ഫങ്ഷനുകളുടേയും വ്യാപ്തിയും ജീവനവും ടേബിൾ 3.5 ൽ സംഗ്രഹിച്ചിരിക്കുന്നു.

വ്യാപ്തിയും ജീവനും	ലോക്കൽ	ഗ്ലോബൽ
വേരിയബിളുകൾ	• ഒരു ഫങ്ഷൻ അല്ലെങ്കിൽ സ്റ്റേറ്റ്മെന്റുകളുടെ കൂട്ടത്തിന് ഉള്ളിൽ പ്രഖ്യാപിക്കുന്നു. • ആ ഫങ്ഷൻ അല്ലെങ്കിൽ ബ്ലോക്കിൽ മാത്രമേ ലഭ്യമാകൂ. • ഫങ്ഷൻ അല്ലെങ്കിൽ ബ്ലോക്ക് പ്രവർത്തിക്കുമ്പോൾ മെമ്മറി അനുവദിക്കുകയും ഫങ്ഷന്റെയോ ബ്ലോക്കിന്റെയോ പ്രവർത്തനം പൂർത്തിയാകുമ്പോൾ അവ സ്വതന്ത്രമാക്കുകയും ചെയ്യുന്നു.	• എല്ലാ ഫങ്ഷന്റെയും പുറമെ പ്രഖ്യാപിക്കുന്നു. • പ്രോഗ്രാമിലെ എല്ലാ ഫങ്ഷനുകൾക്കും ലഭ്യമാണ്. • പ്രോഗ്രാമിന്റെ പ്രവർത്തനം തുടങ്ങുന്നതിന് തൊട്ട് മുൻപേ മെമ്മറി അനുവദിക്കുകയും പ്രോഗ്രാം അവസാനിക്കുമ്പോൾ അവ സ്വതന്ത്രമാവുകയും ചെയ്യുന്നു.
ഫങ്ഷനുകൾ	• ഒരു കൂട്ടം സ്റ്റേറ്റ്മെന്റിന് ഉള്ളിലോ ഫങ്ഷന് ഉള്ളിലോ പ്രഖ്യാപിക്കുകയും വിളിക്കുന്ന ഫങ്ഷന് ശേഷം നിർവ്വചിക്കുകയും ചെയ്യുന്നു. • ആ ഫങ്ഷൻ അല്ലെങ്കിൽ ബ്ലോക്കിന് ഉള്ളിൽ മാത്രമേ പ്രാപ്യമാകൂ.	• മറ്റ് എല്ലാ ഫങ്ഷനുകളുടേയും വെളിയിൽ പ്രഖ്യാപിക്കുകയോ നിർവ്വചിക്കുകയോ ചെയ്യുന്നു. • പ്രോഗ്രാമിലെ എല്ലാ ഫങ്ഷനുകൾക്കും പ്രാപ്യമാണ്.

പട്ടിക 3.5 : വേരിയബിളുകളുടേയും ഫങ്ഷനുകളുടേയും വ്യാപ്തിയും ജീവനവും

നമുക്ക് സംഗ്രഹിക്കാം

പ്രോഗ്രാമിങ്ങ് എളുപ്പത്തിൽ ആക്കുന്ന ഒരു സമീപനമാണ് മോഡുലാർ പ്രോഗ്രാമിങ്ങ്. C++ ഫങ്ഷനിലൂടെ മോഡുലറൈസേഷന് സൗകര്യം ഒരുക്കുന്നു. ഒരു പ്രത്യേക ഉദ്യമം നടത്തുന്നതിന് പ്രോഗ്രാമിൽ ഉൾക്കൊള്ളിച്ചിരിക്കുന്ന പേരോടു കൂടിയ ഒരു ഘടകമാണ് ഫങ്ഷൻ. C++ൽ മുൻ നിർവചിത ഉപയോക്തൃ നിർവചിത എന്നീ രണ്ട് തരം ഫങ്ഷനുകൾ ഉണ്ട്. മുൻ നിർവചിത ഫങ്ഷനുകൾ ഉപയോഗിക്കണമെങ്കിൽ അതുമായി ബന്ധപ്പെട്ട ഹെഡർ ഫയൽ നാം പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തണം. വിളിക്കുന്ന ഫങ്ഷന് ശേഷമാണ് നിർവ്വചിച്ചിരിക്കുന്നതെങ്കിൽ അത്തരം ഫങ്ഷൻ പ്രഖ്യാപിക്കേണ്ടത് ആവശ്യമാണ്. ഫങ്ഷൻ വിളിക്കുമ്പോൾ വിളിക്കുന്ന ഫങ്ഷനിൽ നിന്നും വിളിച്ച ഫങ്ഷനിലേക്ക് ഡാറ്റ ആർഗ്യുമെന്റിലൂടെ അയച്ചേക്കാം. ആർഗ്യൂമെന്റുകളെ യഥാക്രമ (ഫോർമൽ) പരാമീറ്റർ, ആച്വൽ (യഥാർത്ഥ) പരാമീറ്റർ എന്നിങ്ങനെ രണ്ടായി തരംതിരിക്കാം. ഫങ്ഷനിലേക്ക് പരാമീറ്റർ അയയ്ക്കുന്നതിന് കാൾ-ബൈ-വാല്യൂ രീതിയോ അല്ലെങ്കിൽ കാൾ ബൈ റഫറൻസ് രീതിയോ ഉപയോഗിക്കാം. ഒരു പ്രോഗ്രാമിലെ വേരിയബിളുകൾക്കും ഫങ്ഷനുകൾക്കും അവ പ്രഖ്യാപിച്ചിരിക്കുന്ന സ്ഥലത്തിനനുസരിച്ച് വ്യാപ്തിയും ജീവനവും ഉണ്ട്.

നമുക്ക് വിലയിരുത്താം

1. ഒരു സംഖ്യ സ്വീകരിച്ച് അത് അഭിഭാജ്യം ആണെങ്കിൽ 1 ഉം അല്ലെങ്കിൽ 0 ഉം തിരിച്ച് നൽകുന്നതിനുള്ള ഒരു ഫങ്ഷൻ നിർവ്വചിക്കുക. ഈ ഫങ്ഷൻ ഉപയോഗിച്ച് 100 നും 200 നും ഇടക്കുള്ള എല്ലാ അഭാജ്യ സംഖ്യകളും പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം എഴുതുക.
2. ഒരു ഉപയോക്തൃ നിർവചിത ഫങ്ഷന്റെ സഹായത്താൽ മൂന്ന് ഇൻഡിജർ നമ്പറുകളുടെ ഗുണനഫലം കാണുക. കാൾ ബൈ വാല്യൂ രീതിയിലും കാൾ ബൈ റഫറൻസ് രീതിയിലും ഫങ്ഷൻ വിളിക്കുക. ഉത്തരം പരിശോധിക്കുക.
3. ഒരു ഫങ്ഷൻ ഉപയോഗിച്ച് തന്നിരിക്കുന്ന മൂന്ന് അല്ലെങ്കിൽ രണ്ട് സംഖ്യകളിൽ ഏറ്റവും ചെറിയ സംഖ്യ കണ്ടുപിടിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം എഴുതുക (തനത് ആർഗ്യൂമെന്റുകളുടെ ആശയം ഉപയോഗിക്കുക).
4. ഒരു ഉപയോക്തൃ നിർവ്വചിത ഫങ്ഷന്റെ സഹായത്തോടെ ഒരു സംഖ്യയിലെ അക്കങ്ങളുടെ തുക കണ്ടുപിടിക്കുക. (അതായത് സംഖ്യ 3245 ആണെങ്കിൽ, ഉത്തരം 3+2+4+5 = 14 ആയിരിക്കണം).
5. ഒരു ഫങ്ഷൻ ഉപയോഗിച്ച് തന്നിരിക്കുന്ന രണ്ട് സംഖ്യകളുടെ LCM കണ്ടുപിടിക്കുന്നതിനുള്ള ഒരു പ്രോഗ്രാം എഴുതുക.
6. തന്നിരിക്കുന്ന ഒരു സംഖ്യ പോസിറ്റീവ്, നെഗറ്റീവ്, പൂജ്യം എന്നിവ ആണോയെന്ന് പരിശോധിക്കുന്നതിനുള്ള പ്രോഗ്രാം എഴുതുക. ഒരു ഉപയോക്തൃ നിർവചിത ഫങ്ഷൻ ഉപയോഗിച്ച് പരിശോധിക്കുക.

നമുക്ക് വിലയിരുത്താം

1. C++ ലെ ഒരു ഫങ്ഷൻ എന്താണ്?
2. ഒരു സ്ട്രിങ്ങിന്റെ വലിപ്പം കണ്ടുപിടിക്കുന്ന മുൻനിർവചിത ഫങ്ഷനാണ്.........................
3. C++ പ്രോഗ്രാമുകളിൽ ഹെഡർ ഫയലുകളുടെ കടമ എന്താണെന്ന് എഴുതുക?
4. ഫങ്ഷൻ നിർവചനത്തിനു വേണ്ടി void ഡാറ്റാതരം ഉപയോഗിക്കുന്നത് ഏത് സന്ദർഭത്തിലായിരിക്കും.
5. യഥാർഥ ആർഗ്യുമെന്റുകൾ (actual parameters) യഥാക്രമം ആർഗ്യുമെന്റുകൾ (formal parameters) എന്നിവയുടെ വ്യത്യാസങ്ങൾ കണ്ടെത്തുക.
6. താഴെ കൊടുത്തിരിക്കുന്ന ഫങ്ഷനുകൾക്ക് വേണ്ട പ്രോട്ടോ ടൈപ്പുകൾ നിർമിക്കുക.
   a. Total() – രണ്ട് ഡബിൾ ഡാറ്റ ഇനത്തിലുള്ള ആർഗ്യുമെന്റുകൾ സ്വീകരിക്കുകയും ഒരു ഡബിൾ ഡാറ്റ ഇനം തിരിച്ചു നൽകുന്നു.
   b. Math() – ഒരു ആർഗ്യുമെന്റും സ്വീകരിക്കുകയോ തിരിച്ചു നൽകുകയോ ചെയ്യുന്നില്ല.
7. ഗ്ലോബൽ, ലോക്കൽ വേരിയബലുകളുടെ വ്യാപ്തി (scop) ഉദാഹരണ സഹിതം ചർച്ച ചെയ്യുക.
8. ഫങ്ഷൻ വിളികൾക്ക് വേണ്ടി ഉപയോഗിക്കുന്ന കാൾ ബൈ വാല്യു, കാൾ ബൈ റഫറൻസ് എന്നീ രീതികൾ താരതമ്യം ചെയ്യുക.
9. C++ ൽ എല്ലാ ആർഗ്യുമെന്റുകളും എടുത്തു പറയാതെ ഫങ്ഷനെ വിളിക്കാൻ സാധിക്കും. എങ്ങനെ?
10. ഒരു ഗ്ലോബൽ ഫങ്ഷനിൽ നിന്നും ഒരു ലോക്കൽ ഫങ്ഷൻ എങ്ങനെ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു?
11. താഴെ കൊടുത്തിരിക്കുന്നവയ്ക്ക് ആവശ്യമായ മുൻനിർവചിത ഫങ്ഷനുകൾ തിരിച്ചറിയുക.
    a. 'c' എന്ന അക്ഷരം 'C' ആക്കി മാറ്റുന്നതിന്
    b. തന്നിരിക്കുന്ന ഒരു ക്യാരക്ടർ അക്ഷരമാണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നു.
    c. സ്ട്രിങ്ങുകളായ "comp" ഉം "uter" ഉം കൂട്ടിച്ചേർത്ത് "computer" എന്ന സ്ട്രിങ് ഉണ്ടാക്കുന്നതിന്.
    d. 25 ന്റെ വർഗമൂല്യം കണ്ടുപിടിക്കുന്നതിന്
    e. -10 എന്ന സംഖ്യയിൽ നിന്നും 10 തിരിച്ച് നൽകുന്നതിന്.
12. താഴെ കൊടുത്തിരിക്കുന്ന ഫങ്ഷൻ നോക്കുക

```
int sum(int a,int b=o,int(=o)
{
        relarn (a+b+c);
 }
```

a പരാമീറ്റർ ലിസ്റ്റിനെ സംബന്ധിച്ച് ഫങ്ഷന്റെ പ്രത്യേകതകൾ എന്താണ്?

b താഴെകൊടുത്തിരിക്കുന്ന ഫങ്ഷനുകളുടെ ഔട്ട്പുട്ട് എഴുതി അതിന്റെ പ്രവർത്തനം വിശദമാക്കുക ഫങ്ഷൻ കാൾ തെറ്റാണങ്കിൽ അതിന്റെ കാരണം എഴുതുക.

```
(i) cont<<sum (1,2,3):   (ii)cont <<sum (5,2);
(iii) cout<<sun();        (1v) cout <<sum(o);
```

13. `int fun (in, inl);` എന്നത് ഒരു ഫങ്ഷന്റെ പ്രോട്ടോടൈപ്പ് ആണ്. താഴെ കൊടുത്തിരിക്കുന്ന ഫങ്ഷൻ വിളികൾ അസാധുവാണ് ഓരോന്നിന്റെയും കാരണം എഴുതുക.

```
a)fun(2,4);        b) cout<<fun();  c) val=fun(2.5,3.3);
 d)cin>>fun(a,b);  e) 2=fun(3);
```

14. താഴെ കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാം പരിഗണിച്ച് ആര (radius)ത്തിന്റെ വില 5 ആണെങ്കിലുള്ള ഔട്ട്പുട്ട് കണ്ടെത്തുകയും അതിന്റെ കാരണം വിശദമാക്കുകയും ചെയ്യുക.

```
#include<iostream>
using namespace std;
float area(float &);
int main()
{
   float r, ans;
   cout<<"Enter radius :";
   ans = area(r);
   cout<<area;
   cout<<r;
   return 0;
}
float area (float &p)
{
   float q;
   q = 3.14 * p * P;
   p++;
   return q;
}
```

15. ചോദ്യം പതിനാലിൽ കാൾ ബൈ വാല്യൂ രീതി ഉപയോഗിച്ച് ഫങ്ഷൻ വിളിക്ക് മാറ്റം വരുത്തി ഔട്ട്പുട്ടിലുള്ള സാധ്യമായ വ്യത്യാസം എഴുതുക

J1Q7B1

4 വെബ് സാങ്കേതികവിദ്യ

പ്രധാന പഠനനേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പഠനം പൂർത്തിയാകുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ

- സുരക്ഷിത ആശയവിനിമയം വെബ്പേജുകളിലൂടെ സാധ്യമാകുന്നതെങ്ങനെ എന്ന് വിശദീകരിക്കുന്നു.
- വെബ്സെർവറിന്റെ ഉപയോഗവും വെബ്ഹോസ്റ്റിങ്ങിനെക്കുറിച്ചുള്ള അറിവും വിവരിക്കുന്നു.
- സ്റ്റാറ്റിക്, ഡൈനാമിക് എന്നീ വെബ് പേജുകൾ താരതമ്യം ചെയ്യുന്നു.
- പ്രോഗ്രാമിങ് ഭാഷകളും സ്ക്രിപ്റ്റുകളും തമ്മിലുള്ള വ്യത്യാസം തിരിച്ചറിയുന്നു.
- വിവിധ സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ താരതമ്യം ചെയ്യുന്നു.
- കാസ്കേഡിംഗ് സ്റ്റൈൽ ഷീറ്റിന്റെ ആവശ്യകത വിശദീകരിക്കുന്നു.
- ലളിതമായ HTML എലമെന്റുകൾ ഉപയോഗിച്ചുള്ള വെബ്പേജ് നിർമാണം മനസ്സിലാക്കുന്നു.
- അടിസ്ഥാന HTML ടാഗുകളും അവയുടെ പ്രധാന ആട്രിബ്യൂട്ടുകളും പട്ടികപ്പെടുത്തുന്നു.
- HTML ടാഗുകളെ തരം തിരിക്കുന്നു.
- HTML പേജുകൾ ആകർഷകമാക്കുന്നതിന് ഫോർമാറ്റിംഗ് ടാഗുകൾ HTML ഡോക്യുമെന്റുകളിൽ ഉപയോഗിക്കുന്നത് എങ്ങനെയെന്ന് മനസ്സിലാക്കുന്നു.
- ഫോർമാറ്റിംഗ് ടാഗുകൾ തമ്മിലുള്ള സാമ്യങ്ങളും വ്യത്യാസങ്ങളും തിരിച്ചറിയുന്നു.
- <PRE>, <DIV> എന്നീ ടാഗുകളുടെ ഉപയോഗം നിരീക്ഷിക്കുന്നു.
- ഒരു ഡോക്യുമെന്റിലെ ഉള്ളടക്കമോ മറ്റ് ഒബ്ജക്ടുകളോ (വസ്തുക്കളോ) ചലിപ്പിക്കുന്നതിനുള്ള ടാഗ് തിരിച്ചറിയുന്നു.
- <FONT> ടാഗ് ഉപയോഗിച്ച് ടെക്സ്റ്റ് ഉള്ളടക്കം ഫലപ്രദമായി രൂപപ്പെടുത്തുന്നു.
- HTML ഡോക്യുമെന്റുകളിൽ കമന്റുകൾ ഉപയോഗിക്കുന്നു.
- <IMG> ടാഗ് ഉപയോഗിച്ച് HTML ഡോക്യുമെന്റുകളിൽ ചിത്രങ്ങൾ ഉൾക്കൊള്ളിക്കുന്നു.

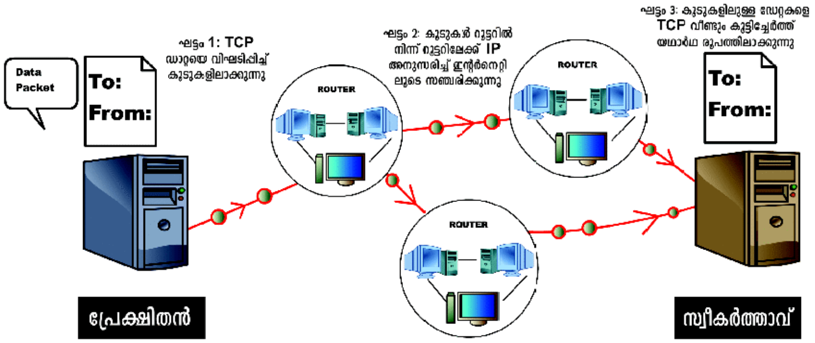
ഇന്റർനെറ്റ് യുഗത്തിലാണ് നാമെല്ലാം ജീവിക്കുന്നത്. വിവരങ്ങൾ ആവശ്യമായി വരുമ്പോൾ നാം ആദ്യം തിരയുന്നത് ഇന്റർനെറ്റിലായിരിക്കും. 11-ാം ക്ലാസിലെ പരീക്ഷാഫലം വെബ് സൈറ്റിൽ നിങ്ങൾ തിരഞ്ഞ് അറിഞ്ഞിട്ടുണ്ട്. ഓൺ ലൈൻ അപേക്ഷ സമർപ്പണം, വെബ് സൈറ്റിലുള്ള കാര്യങ്ങൾ പരിശോധിക്കൽ, സിനിമ കാണൽ, ധനവ്യാപാരസ്ഥാപനങ്ങളിലെ ക്രയവിക്രയം, ഓൺ ലൈൻ ക്രയവിക്രയത്തിലൂടെ സാധനങ്ങൾ വാങ്ങൽ തുടങ്ങി വിവിധ ആവശ്യങ്ങൾക്കായി ധാരാളം വെബ് സൈറ്റുകളെ നാം ആശ്രയിക്കാറുണ്ട്. വെബ് പേജുകളുടെ ഒരു കൂട്ടമാണ് വെബ് സൈറ്റ് എന്ന് നമുക്കറിയാം. ചിഹ്നങ്ങൾ (അക്ഷരങ്ങളോ അല്ലാത്തവയോ ആയത്), ചിത്രങ്ങൾ, ഗ്രാഫുകൾ, ചലനാത്മക ചിത്രങ്ങൾ, ശബ്ദ വീഡിയോകൾ തുടങ്ങിയവ വെബ് പേജുകളിലുണ്ടാവാം. സ്ഥാപനങ്ങൾ, ഉൽപന്നങ്ങൾ, സേവനങ്ങൾ തുടങ്ങിയവയെക്കുറിച്ചുള്ള വിവരം പ്രദാനം ചെയ്യുന്നതിനാണ് ധാരാളം പേജുകളുള്ള ഒരു വെബ്സൈറ്റ് രൂപകൽപന ചെയ്യുന്നത്. ഈ വെബ്സൈറ്റ് ഇന്റർനെറ്റിൽ നിന്നും നമുക്ക് എങ്ങനെ ലഭിക്കുന്നു? ഇന്റർനെറ്റുമായി ബന്ധിപ്പിച്ചിട്ടുള്ള വെബ് സെർവറുകളിൽ ഈ പേജുകൾ മറ്റുള്ളവർക്ക് ലഭിക്കത്തക്കവിധം സംഭരിച്ച് വച്ചിരിക്കുന്നു. ഈ അധ്യായത്തിൽ ഇന്റർനെറ്റ് വഴിയുള്ള ആശയവിനിമയവും വെബ് സെർവറുകൾക്ക് അതിലുള്ള പ്രാധാന്യവും അവലോകനം ചെയ്യുന്നു. വെബ് സൈറ്റുകൾ വികസിപ്പിക്കുന്നതിനാവശ്യമായ വിവിധ ടൂളുകളെക്കുറിച്ചും, സാങ്കേതികതകളെക്കുറിച്ചും ഇവിടെ പ്രതിപാദിക്കുന്നുണ്ട്. പരിവർത്തനാത്മക

(dynamic) വെബ് പേജുകളുടെ ആശയവും അവ പരിവർത്തനാത്മകം ആക്കുന്നതിന് ഉപയോഗിക്കുന്ന സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ അവലോകനവും ഇവിടെ ചർച്ചചെയ്യുന്നു. ഹൈപ്പർ ടെക്സ്റ്റ് മാർക്കപ്പ് ഭാഷ (HTML) ഉപയോഗിച്ചാണ് വെബ് പേജുകൾ നിർമിക്കുന്നത്. ഇന്റർനെറ്റിന്റെ ഭാഷയായി HTML അറിയപ്പെടുന്നു. ഒരു വെബ് പേജിന്റെ ഉള്ളടക്കം ബ്രൗസർ വിൻഡോയിൽ എങ്ങനെ പ്രദർശിപ്പിക്കണമെന്നാണ് HTML ബ്രൗസറിനോട് പറയുന്നത്. HTML ഉപയോഗിച്ചുള്ള വെബ് പേജ് നിർമിതിയുടെ അടിസ്ഥാന തത്വങ്ങൾ ഈ അധ്യായത്തിൽ നമുക്ക് പരിചയപ്പെടാം.

4.1 വെബ്ബിലൂടെയുള്ള ആശയ വിനിമയം (Communication on the web)

വെബ് ബ്രൗസർ (ക്ലയന്റ്) ഇന്റർനെറ്റിലുള്ള ഒരു വെബ് സൈറ്റ് പരിശോധിക്കുമ്പോൾ സംഭവിക്കുന്ന കാര്യങ്ങൾ നടക്കുന്നുവെന്ന് 11-ാം ക്ലാസ്സിലെ അധ്യായം 12 -ൽ നാം പഠിച്ചതാണ്. അനുബന്ധ ഇന്റർനെറ്റ് പ്രോട്ടോകോൾ (IP) മേൽവിലാസത്തിനായി URL (യൂണിഫോം റിസോഴ്സ് ലൊക്കേറ്റർ)-നെ DNS (ഡൊമൈൻ നെയിം സിസ്റ്റം) സെർവറിലേയ്ക്ക് അയയ്ക്കുകയും തുടർന്ന് ലഭിക്കുന്ന IP മേൽവിലാസം ഉപയോഗിച്ച് ബ്രൗസർ സെർവറുമായി ബന്ധം സ്ഥാപിക്കുകയും ചെയ്യുന്നു. സെർവർ കമ്പ്യൂട്ടറിലുള്ള സെർവർ സോഫ്റ്റ്വെയർ ഈ ആവശ്യം പ്രോസസ് ചെയ്ത് പ്രദർശിപ്പിക്കേണ്ട വെബ് പേജിനെ ക്ലയന്റിന് അയച്ച് കൊടുക്കുന്നു. ക്ലയന്റ് ബ്രൗസർ വെബ് പേജ് ക്രമമായി കാണിക്കുന്നു.

വെബ് ഉപയോഗിച്ചുള്ള ആശയവിനിമയത്തിന്, കമ്പ്യൂട്ടറുകൾ അല്ലെങ്കിൽ ഉപകരണങ്ങൾ പരസ്പരം തിരിച്ചറിയേണ്ടതുണ്ട്. ട്രാൻസ്മിഷൻ കൺട്രോൾ പ്രോട്ടോകോൾ/ ഇന്റർനെറ്റ് പ്രോട്ടോകോൾ (TCP/IP) എന്ന ഒറ്റ പ്രോട്ടോകോൾ തന്നെ എല്ലാ ഉപകരണങ്ങളും അനുസരിക്കുന്നത് കൊണ്ടാണ് ഇത് സാധ്യമാകുന്നത്. TCP/IP പ്രോട്ടോകോളുകളെക്കുറിച്ചും അവയുടെ പ്രവർത്തനത്തെക്കുറിച്ചും 11-ാം ക്ലാസിലെ കമ്പ്യൂട്ടർ ശൃംഖലകൾ (കമ്പ്യൂട്ടർ നെറ്റ്വർക്ക്സ്) എന്ന അധ്യായത്തിൽ നാം വിശദമായി ചർച്ച ചെയ്തിട്ടുണ്ട്. അയയ്ക്കേണ്ട ഡാറ്റയെ, TCP/IP പ്രോട്ടോകോൾ വിഘടിപ്പിച്ച് ചെറിയ കൂടുകളിലാക്കി സ്വീകർത്താവിന്റെ കമ്പ്യൂട്ടർ വിലാസത്തോടൊപ്പം ചേർക്കുന്നു. ഇന്റർനെറ്റ് പ്രോട്ടോകോളിന്റെ സഹായത്താൽ, റൂട്ടറുകൾ (വഴികാട്ടുന്ന ഉപകരണങ്ങൾ) ഈ ഡാറ്റ ഉൾക്കൊള്ളുന്ന കൂടുകളെ അവ എത്തേണ്ട കമ്പ്യൂട്ടറിലെത്തിക്കുന്നു. അയക്കുന്നയാൾ (പ്രേക്ഷിതൻ) ഡാറ്റയുടെ ഒരു കൂട് സ്വീകർത്താവിൽ എത്തിക്കുന്നതെങ്ങനെ എന്ന് ചിത്രം 4.1 ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 4.1: ഡാറ്റ ഉൾക്കൊള്ളുന്ന കൂടുകൾ പ്രേക്ഷിതനിൽ നിന്ന് സ്വീകർത്താവിലേക്ക് വഴികാട്ടി അയക്കുന്നു

വെബ്സൈറ്റ് പരിശോധിക്കൽ, ഇ-മെയിൽ അയക്കൽ തുടങ്ങി പലതരത്തിലുള്ള ആശയവിനിമയങ്ങൾ ഇന്റർനെറ്റിൽ നടക്കുന്നു. വെബ്സൈറ്റുകൾക്ക് വേണ്ടിയുള്ള തിരയൽ HTTP (ഹൈപ്പർ ടെക്സ്റ്റ് ട്രാൻസ്മിഷൻ പ്രോട്ടോകോൾ) ഉപയോഗിച്ചും ഇ-മെയിൽ വഴിയുള്ള ആശയവിനിമയം SMTP (സിമ്പിൾ മെയിൽ ട്രാൻസ്ഫർ പ്രോട്ടോകോൾ) ഉപയോഗിച്ചുമാണ് നടക്കുന്നതെന്ന് നാം പഠിച്ചിട്ടുണ്ട്. താഴെ തലത്തിലുള്ള ഇന്റർനെറ്റ് പ്രോട്ടോകോളിന്റെ ഏറ്റവും മുകളിലായാണ് ഈ രണ്ട് പ്രോട്ടോകോളുകളും പ്രവർത്തിക്കുന്നത്. ഇന്റർനെറ്റ് വഴിയുള്ള ആശയവിനിമയം ലളിതമാക്കുന്നത് ഇന്റർനെറ്റ് പ്രോട്ടോകോളാണ്. ഈ പ്രോട്ടോകോൾ തന്നെ ഉപയോഗിക്കുന്നതുകൊണ്ട് ധാരാളം നേട്ടങ്ങളുണ്ട്. വിവിധതരം ഡാറ്റ കൈകാര്യം ചെയ്യുന്നതിനായി റൂട്ടറുകൾ പ്രത്യേകം പ്രോഗ്രാം ചെയ്യേണ്ടിവരുന്നില്ല. കൊണ്ടുപോകേണ്ട ഡാറ്റ ശ്രദ്ധിക്കാതെ അവ ഉൾക്കൊള്ളുന്ന കൂടുകൾ എത്തിക്കേണ്ട മേൽവിലാസത്തെക്കുറിച്ച് മാത്രം ശ്രദ്ധിച്ചാൽ മതിയാകും. കൂടുകളിലെ ഡാറ്റ ഉൾക്കൊള്ളുന്ന ഭാഗം സുതാര്യമായതിനാൽ, പുതിയ പ്രോട്ടോകോളുകൾ രൂപപ്പെടുത്തി ഇന്റർനെറ്റിൽ ഉപയോഗിക്കുവാനുള്ള സ്വാതന്ത്ര്യം നമുക്ക് ലഭിക്കുന്നു. TCP/IP യുടെ ഈ സുതാര്യതയും വഴക്കവുമാണ്, സന്ദേശങ്ങൾ കൈകാര്യം ചെയ്യുന്നതിനുള്ള സമൂഹമാധ്യമ സൈറ്റുകൾ, ശബ്ദം, വീഡിയോ തുടങ്ങിയവ ഉള്ളടക്കങ്ങളായി വരുന്ന സൈറ്റുകൾ, സുരക്ഷിതപണമിടപാടുകൾക്കായി ധനകാര്യസ്ഥാപനങ്ങൾ ഉപയോഗിക്കുന്ന സൈറ്റുകൾ തുടങ്ങിയവയ്ക്കായി പ്രോട്ടോകോളുകൾ നിർമിക്കുന്നതിന് സഹായിച്ചത്. ഇന്റർനെറ്റിന്റെ സാമ്പത്തിക വിജയത്തിന് കാരണം ഇതായിരുന്നു.

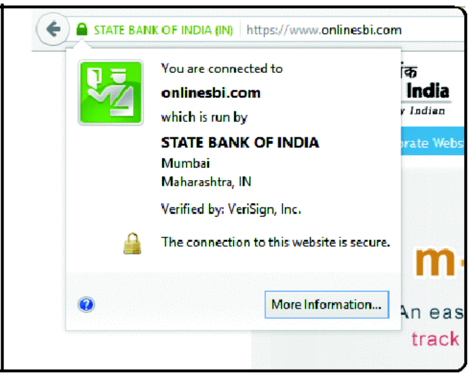
വെബ് വഴിയുള്ള ആശയവിനിമയം ക്ലയന്റ് ബ്രൗസറിൽ നിന്ന് വെബ് സെർവറിലേക്കെന്നും, വെബ് സെർവറിൽനിന്ന് വെബ് സെർവറിലേക്കെന്നും, രണ്ടായി തരംതിരിച്ചിരിക്കുന്നു. ഇന്റർനെറ്റ് വഴിയുള്ള ആശയ വിനിമയത്തിന് ആധികാരികതയും സുരക്ഷിതത്വവും അത്യാവശ്യമാണ്. വെബിലുള്ള ഒരു കമ്പ്യൂട്ടർ/സെർവർ യഥാർഥ അവകാശിയായ കമ്പ്യൂട്ടറാണെന്ന് ഉറപ്പ് വരുത്തുന്ന പ്രക്രിയയാണ് ആധികാരികത ഉറപ്പിക്കൽ (authentication). ഇന്റർനെറ്റ് വഴിയുള്ള ആശയവിനിമയത്തിന് സുരക്ഷിതത്വം ലഭിക്കുന്നതിനാൽ, കമ്പ്യൂട്ടറിലെ വിവരങ്ങൾ ചോർത്തുന്നവർക്ക് (ഹാക്കേഴ്സിന്) സന്ദേശങ്ങൾ പിടിച്ചെടുക്കുവാനോ, രൂപമാറ്റം വരുത്തുവാനോ സാധിക്കുകയില്ല.

4.1.1 ക്ലയന്റിൽ നിന്നും വെബ് സെർവറിലേക്കുള്ള ആശയവിനിമയം (Client to web server communication)

ക്ലയന്റിൽ നിന്ന് വെബ് സെർവറിലേക്കുള്ള ആശയവിനിമയത്തിന് സാധാരണഗതിയിൽ ആധികാരികത ഉറപ്പിക്കേണ്ടതില്ല. എന്നാൽ, ധനകാര്യസ്ഥാപനങ്ങളുടെ ഉപയോഗത്തിനോ, ഇ-മെയിൽ സേവനത്തിനോ ആണെങ്കിൽ യൂസർ നാമവും (യൂസർ നെയിം) പാസ്‌വേഡും സെർവറിലേക്ക് അയക്കേണ്ടതുണ്ട്. സുരക്ഷിതത്വകാരണങ്ങളാൽ, ഈ വിവരങ്ങൾ സാധാരണ ചിഹ്നങ്ങളായി/അക്ഷരങ്ങളായി സെർവറിലേയ്ക്ക് അയയ്ക്കാറില്ല. അഥവാ അങ്ങനെ ചെയ്താൽ നുഴഞ്ഞ് കയറ്റക്കാർക്ക് (ഹാക്കേഴ്സിന്) യൂസർ നാമവും പാസ്‌വേഡും മോഷ്ടിക്കാനാകും. അത്തരം സാഹചര്യങ്ങളിൽ യൂസർ നാമ

വും പാസ്‌വേഡും രൂപമാറ്റം വരുത്തി (എൻക്രിപ്റ്റ് ചെയ്ത്) സെർവറിലേക്ക് അയക്കാനായി നാം HTTPS (ഹൈപ്പർ ടെക്സ്റ്റ് ട്രാൻസ്ഫർ പ്രോട്ടോകോൾ സെക്യൂർ) എന്ന സാങ്കേതികവിദ്യ ഉപയോഗിക്കുന്നു. ഇന്റർനെറ്റിലുള്ള കമ്പ്യൂട്ടറുകൾ തമ്മിൽ ഇത്തരത്തിൽ ബന്ധം സ്ഥാപിക്കാൻ ഉതകുന്ന സുരക്ഷിത സാങ്കേതികവിദ്യയായ SSL (സെക്യൂർഡ് സോക്കറ്റ് ലെയർ) ഉപയോഗിച്ച് ആണ് HTTPS പ്രവർത്തിക്കുന്നത്. HTTP യ്ക്ക് സുരക്ഷിതത്വം പ്രദാനം ചെയ്യുന്നത് HTTPS ആണ്. SSL പ്രോട്ടോകോൾ സ്വകാര്യത ഉറപ്പ് വരുത്തുന്നതോടൊപ്പം മറ്റൊരു വെബ്സൈറ്റിനും ഉപയോക്താവിന്റെ ലോഗിൻ അക്കൗണ്ടിൽ ആൾമാറാട്ടം നടത്താനോ അയച്ച വിവരം മാറ്റം വരുത്താനോ സാധിക്കാത്തവിധം സുരക്ഷ ഉറപ്പ് വരുത്തുകയും ചെയ്യുന്നു.

ചിത്രം 4.2(a): ക്ലയന്റ് - സെർവർ ആധികാരികത ഉറപ്പിക്കൽ പ്രവർത്തനം

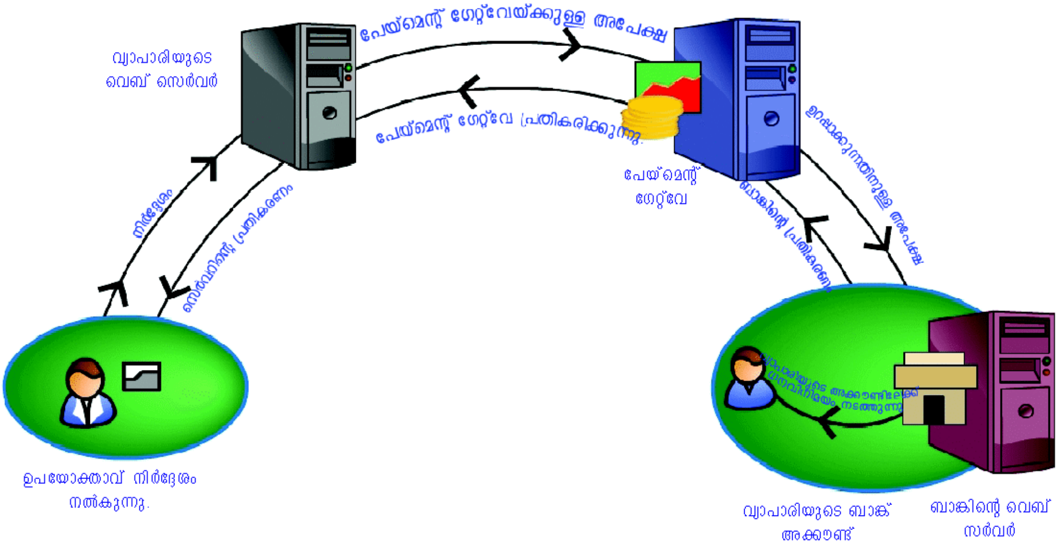


ചിത്രം 4.2(b): SBI വെബ് സൈറ്റിന്റെ ആധികാരികത ഉറപ്പിക്കൽ

ബ്രൗസർ ഒരു സുരക്ഷിത പേജ് ആവശ്യപ്പെടുമ്പോൾ, സെർവർ ആദ്യം അതിന്റെ SSL യോഗ്യതാപത്രം തിരിച്ച് നൽക്കുന്നു. തുടർന്ന് ഈ സാക്ഷ്യപത്രം ബന്ധപ്പെട്ട സാക്ഷ്യപത്ര അധികാരികളുമായി ഒത്തുനോക്കി സാധുത ഉറപ്പ് വരുത്തുന്നു. സാക്ഷ്യപത്ര അധികാരിക്കു നൽകിയ യോഗ്യതാപത്രം സാധുത ഉള്ളതാണെങ്കിൽ, അധികാരി സാക്ഷ്യപ്പെടുത്തുന്നു. ഒരു പ്രത്യേക വെബ്സൈറ്റ് അത് അവകാശപ്പെടുന്ന സ്ഥാപനത്തിന്റേത് തന്നെ ആണെന്ന് ഉറപ്പിക്കലാണ് ഇത്. വെരിസൈൻ ഇൻകോർപ്പറേഷൻ ഒരു സാക്ഷ്യപത്ര അധികാരിയാണ്. ഒരു SSL സാക്ഷ്യപത്രം സാധുത ഉള്ളതായാൽ, ബ്രൗസർ രൂപമാറ്റം വരുത്തിയുള്ള (എൻക്രിപ്റ്റ് ചെയ്ത) ആശയ വിനിമയത്തിന് വേണ്ടി ഒരു നിശ്ചിത സമയം തുടങ്ങിവയ്ക്കുന്നു. ഈ പ്രവർത്തനം ചിത്രം 4.2 (a)-യിൽ ചിത്രീകരിച്ചിരിക്കുന്നു. ഇന്ത്യയിൽ ധനകാര്യസ്ഥാപനങ്ങളുമായുള്ള ഇടപാടുകൾക്ക് ഉപയോഗിക്കുന്ന വെബ്സൈറ്റുകൾ, ക്ലയന്റുകളിൽ നിന്നും രഹസ്യവിവരങ്ങൾ സ്വീകരിക്കുവാൻ HTTPS ഉപയോഗിക്കണമെന്ന് വിവരസാങ്കേതികവിദ്യ നിയമം നിഷ്കർഷിച്ചിട്ടുണ്ട്. നിങ്ങൾ അഡ്രസ് ബാറിലുള്ള ലോക്കിൽ ക്ലിക്ക് ചെയ്താൽ ചിത്രം 4.2 (b)-യിൽ തന്നിട്ടുള്ള സാക്ഷ്യപത്രം കാണാവുന്നതാണ്.

4.1.2 വെബ് സെർവറിൽ നിന്നും വെബ് സെർവറിലേക്കുള്ള ആശയവിനിമയം (Web server to web server communication)

ചില വെബ് ആപ്ലിക്കേഷനുകളുടെ പ്രവർത്തനത്തിന് വെബ് സെർവറിൽനിന്ന് വെബ് സെർവറിലേക്കുള്ള ആശയവിനിമയം ആവശ്യമാണ്. ഉദാഹരണമായി, ഇന്റർനെറ്റ് വഴി സാധനങ്ങൾ വാങ്ങാൻ ഉപയോഗിക്കുന്ന ഒരു വെബ്സൈറ്റ് സൂക്ഷിച്ചിരിക്കുന്ന വെബ് സെർവറിന് (ഇന്റർനെറ്റിലെ വ്യാപാരിക്ക്/വിൽപ്പനക്കാരന്), ധനകാര്യസ്ഥാപനത്തിന്റെ വെബ് സെർവറിലേക്കും അതുപോലെ തിരിച്ചും രഹസ്യ വിവരങ്ങൾ അയയ്ക്കേണ്ടി വരുന്നു. അത്തരം സന്ദർഭങ്ങളിൽ വ്യാപാരിയുടെയും ധനകാര്യ സ്ഥാപനത്തിന്റെയും വെബ്സർവറുകളുടെ ആധികാരികത ഉറപ്പിക്കേണ്ടതുണ്ട്. സ്വീകരിക്കപ്പെട്ട ഡാറ്റ യഥാർഥ സെർവറുകളിൽ നിന്നുള്ളവയാണോ എന്നുറപ്പിക്കാൻ ഡിജിറ്റൽ സാക്ഷ്യപത്ര



ചിത്രം 4.3: ഒരറ്റം മുതൽ മറ്റേ അറ്റം വരെയുള്ള ഓൺലൈൻ വ്യവഹാരം

ങ്ങൾ സഹായിക്കുന്നു. ഒരിക്കൽ സെർവറുകളുടെ ആധികാരികത ഉറപ്പിച്ചു കഴിഞ്ഞാൽ, അവ രൂപമാറ്റം വരുത്തിയ (എൻക്രിപ്റ്റ്) ഡാറ്റ ഉപയോഗിച്ച് ആശയവിനിമയം നടത്തുന്നു. ഇന്റർനെറ്റ് വഴി പണം അടയ്ക്കുകയോ/കൈമാറുകയോ ചെയ്യുമ്പോൾ, വ്യാപാരിയുടെ സെർവറിനും ധനകാര്യസ്ഥാപനത്തിന്റെ സെർവറിനും ഇടയ്ക്ക് ഒരു പേയ്മെന്റ് ഗേറ്റ്വേ സെർവർ പ്രവർത്തിച്ച് രൂപമാറ്റം (എൻക്രിപ്റ്റ് ചെയ്ത) വരുത്തി ധനവിനിമയം നടത്തുന്നു. ഈ പ്രവർത്തനം ചിത്രം 4.3-ൽ കാണിച്ചിരിക്കുന്നു.

4.2 വെബ് സെർവർ സാങ്കേതികവിദ്യകൾ (Web server technologies)

ഹയർസെക്കന്ററി വിദ്യാഭ്യാസ ഡയറക്ടറേറ്റിന്റെ (DHSE-ഡയറക്ടറേറ്റ് ഓഫ് ഹയർ സെക്കന്ററി എഡ്യൂക്കേഷൻ) ഔദ്യോഗിക വെബ്സൈറ്റായ www.dhsekerala.gov.in സന്ദർശിക്കുമ്പോൾ സംഭവിക്കുന്നത് എന്തൊക്കെയെന്ന് നമുക്ക് നോക്കാം. ആദ്യമായി

വെബ് സൈറ്റിന്റെ ഹോം പേജ് (ഒരു വെബ് സൈറ്റ് പരിശോധിക്കുമ്പോൾ കാണിക്കുന്ന ആദ്യത്തെ പേജ്), DHSE യുടെ സെർവറിൽ നിന്നും നമ്മുടെ (ക്ലയന്റിന്റെ) കമ്പ്യൂട്ടറിലേക്ക് കൈമാറ്റം ചെയ്യുന്നു. സാധാരണയായി, വെബ് പേജിന് അഭ്യർഥന നടത്തുന്ന ഒരു കമ്പ്യൂട്ടറോ മൊബൈൽ ഉപകരണമോ ആയിരിക്കും ക്ലയന്റ്. വെബ് സെർവറിൽ ശേഖരിച്ചു വച്ചിരിക്കുന്ന വെബ് സൈറ്റിലെ പേജുകൾ, ക്ലയന്റിന്റെ അപേക്ഷയ്ക്കനുസരിച്ച്, സെർവർ സോഫ്റ്റ്വെയർ വിട്ടു നൽകുന്നു. വെബ് സെർവറിന്റെ സവിശേഷതകളെക്കുറിച്ചും അവ ക്രമീകരിക്കുവാൻ ഉപയോഗിക്കുന്ന വസ്തുക്കളെക്കുറിച്ചും നമുക്ക് ചർച്ച ചെയ്യാം.

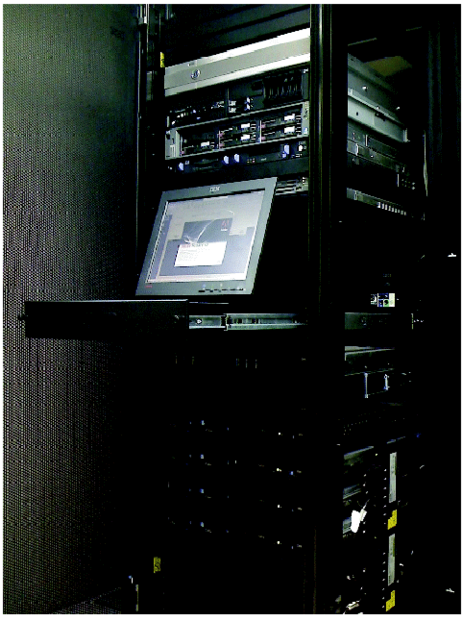
4.2.1 വെബ് സെർവർ (Web server)

വെബ്സൈറ്റുകൾ സംഭരിച്ചിരിക്കുന്ന സെർവർ കമ്പ്യൂട്ടറുകളെ സൂചിപ്പിക്കുന്നതിനാണ് പലപ്പോഴും വെബ്സെർവർ എന്ന പദം ഉപയോഗിക്കുന്നത്. ഒരു സെർവർ കമ്പ്യൂട്ടറിനെ വെബ്സെർവറാക്കാനായി അതിൽ സ്ഥാപിച്ചിട്ടുള്ള വെബ് സെർവർ സോഫ്റ്റ്വെയറിനെയും ഈ പദം ഉപയോഗിച്ച് സൂചിപ്പിക്കാറുണ്ട്. വെബ്സെർവർ കമ്പ്യൂട്ടറിന്റെയും വെബ് സെർവർ സോഫ്റ്റ്വെയറിന്റെയും സവിശേഷതകൾ എന്തൊക്കെയാണെന്ന് നമുക്ക് നോക്കാം.

ചിത്രം 4.4: ഒരു ഡാറ്റാ സെന്റർ

ഇന്റർനെറ്റ് ഉപയോക്താക്കൾക്ക് ഇ-മെയിൽ, ബ്ലോഗ് തുടങ്ങിയ സേവനങ്ങളോ വെബ് പേജുകളോ വിട്ട് നൽകാൻ വെബ്സെർവർ സഹായിക്കുന്നു. www, ഇ-മെയിൽ തുടങ്ങിയ സേവനങ്ങൾ ഇന്റർനെറ്റിലൂടെ നൽകുന്നതിനു വേണ്ടി ഒരു സെർവർ ഓപ്പറേറ്റിംഗ് സിസ്റ്റവും സെർവർ സോഫ്റ്റ്വെയറും പ്രവർത്തനക്ഷമമാക്കിയിട്ടുള്ള ഒരു സെർവർ കമ്പ്യൂട്ടറാണ് വെബ്സെർവർ. ഇത് വളരെ കാര്യക്ഷമതയുള്ളതും എപ്പോഴും പ്രവർത്തിക്കുന്നതും ഉയർന്ന ബാന്റ്വിഡ്ത്തോടുകൂടിയ ഇന്റർനെറ്റ് ബന്ധമുള്ളതും ആയിരിക്കും. ലോകമെമ്പാടുമുള്ള ഇന്റർനെറ്റ് ഉപയോക്താക്കൾക്ക് ഇതിലുള്ള വെബ് സൈറ്റുകളും സേവനങ്ങളും ഉപയോഗിക്കാനാകും. ഒരു വെബ് സെർവറിൽ ഒന്നോ അതിലധികമോ പ്രൊസസ്സറുകളും, ഉയർന്ന വേഗതയുള്ള റാമും, വർധിച്ച കാര്യക്ഷമതയുള്ള ഹാർഡ് ഡിസ്കും, ദ്രുതഗതിയിലുള്ള ആശയവിനിമയത്തിന് ഉപകരിക്കുന്ന ഈഥർനെറ്റ് കാർഡുമൊക്കെയുണ്ടാകും. വേഗതയുള്ള ഇന്റർനെറ്റ് ബന്ധം ഉറപ്പാക്കാനും, ആവശ്യാനുസരണം വൈദ്യുതി എത്തിക്കുന്നതിനും മറ്റുമായി ഒരു വെബ് സെർവർ സാധാരണ ഗതിയിൽ ഡാറ്റ സെന്ററുകളിലാണ് സ്ഥാപിക്കാറുള്ളത്. സ്ഥാപനങ്ങളുടെ സെർവറുകളും അനുബന്ധ കമ്പ്യൂട്ടറുകളും സ്ഥാപിക്കുന്നതിന് മാത്രമായി വിനിയോഗിച്ചിട്ടുള്ള ഭൗതികസ്ഥലങ്ങളാണ് ഡാറ്റസെന്ററുകൾ. ക്ലയന്റിന് വേണ്ടി, നിയുക്ത പ്രവൃത്തിയുമായി ബന്ധപ്പെട്ട നിർണായക ഡാറ്റ സംഭരിക്കുന്നതിനും, പ്രോസസ്സ് ചെയ്യുന്നതിനും, പ്രയോജനപ്പെടുത്തുന്നതിനും വേണ്ടിയാണ് ഡാറ്റസെന്ററുകൾ ഉപയോഗിക്കുന്നത്. വൈദ്യുതവിത

രണത്തിന് തടസം വരാതിരിക്കാനുള്ള സംവിധാനം, ശീതീകരണ സംവിധാനം, വേഗമേറിയ നെറ്റ്‌വർക്ക്, സുരക്ഷിതത്വക്രമീകരണങ്ങൾ തുടങ്ങിയ സൗകര്യങ്ങൾ ഡാറ്റസെന്ററിന് ആവശ്യമാണ്. ഇത്തരത്തിലുള്ള ഒരു ഡാറ്റസെന്റർ ചിത്രം 4.4-ൽ കാണിച്ചിരിക്കുന്നു. ചിത്രം 4.5-ൽ കാണുന്ന തരത്തിലുള്ള എളുപ്പത്തിൽ പരിപാലിക്കാവുന്നതും വാതായനങ്ങളോട് കൂടിയതുമായ പ്രത്യേകതരം ചട്ടക്കൂടുകളിൽ ഉറപ്പിച്ച ധാരാളം സെർവറുകൾ ഡാറ്റ സെന്ററുകളിലുണ്ടാവും.


ചിത്രം 4.5: പ്രത്യേകതരം ചട്ടക്കൂടുകളിലുറപ്പിച്ചിട്ടുള്ള സെർവറുകൾ

വിവിധ ലിനക്സ് പതിപ്പുകൾ (റെഡ്ഹാറ്റ്, ഓപ്പൺസൂസ്, ഡിബിയൻ, ഉബുണ്ടു തുടങ്ങിയവ), മൈക്രോ സോഫ്റ്റ് വിൻഡോസ് സെർവർ, ഫ്രീ ബി.എസ്.ഡി, ഒറാക്കിൾ സൊളാരീസ് എന്നിവ പ്രചാരത്തിലുള്ള സെർവർ ഓപ്പറേറ്റിംഗ് സിസ്റ്റങ്ങളാണ്. ഒരു സെർവർ സ്ഥാപിച്ച് അതിൽ ഒരു വെബ് സെർവർ സോഫ്റ്റ്‌വെയർ ഉൾക്കൊള്ളിച്ച് ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന് അനുയോജ്യമായ രീതിയിൽ അതിനെ രൂപപ്പെടുത്തിയെടുക്കേണ്ടതുണ്ട്. സംഭരിച്ച് വച്ചിട്ടുള്ള ഫയലുകൾ ഉപയോക്താക്കൾക്ക് സമയബന്ധിതമായി നൽകുന്നതിനുവേണ്ടി, ഹൈപ്പർ ടെക്സ്റ്റ് ട്രാൻസ്ഫർ പ്രോട്ടോകോളിന്റെ സഹായത്താൽ ക്ലയന്റ്-സെർവർ മോഡലിൽ പ്രവൃത്തിക്കുന്ന ഒരു പ്രോഗ്രാമാണ് വെബ് സെർവർ സോഫ്റ്റ്‌വെയർ. ഉപയോക്താക്കൾ നൽകുന്ന അപേക്ഷയ്ക്കനുസരിച്ച്, വെബ് സെർവർ സോഫ്റ്റ്‌വെയറിന്റെ സഹായത്തോടെ ഈ ഫയലുകൾ വെബ് പേജുകളാക്കി ക്ലൈയന്റിന് അയയ്ക്കുന്നു. തുടർന്ന് ക്ലയന്റ് ബ്രൗസറിൽ ഇവ കാണുന്നു. അപ്പാഷെ സെർവർ, മൈക്രോസോഫ്റ്റ് ഇന്റർനെറ്റ് ഇൻഫർമേഷൻ സർവീസസ്, ഗൂഗിൾ വെബ്സെർവർ (GWS), എൻജിൻ-എക്സ് തുടങ്ങിയവ പ്രചാരത്തിലുള്ള വെബ്സെർവർ പാക്കേജുകളാണ്.

സോഫ്റ്റ്‌വെയർ സ്ഥാപിച്ച് പ്രവർത്തന യോഗ്യമാക്കിയശേഷം അതോടൊപ്പം സോഫ്റ്റ്‌വെയർ പാക്കേജുകളായ FTP (ഫയൽ ട്രാൻസ്ഫർ പ്രോട്ടോകോൾ), ഇ-മെയിൽ, DNS, ഡാറ്റ ബേസ് തുടങ്ങിയവ കൂടി വെബ്സെർവറിൽ ഉൾപ്പെടുത്താവുന്നതാണ്. വെബ് സൈറ്റിന്റെ സുരക്ഷിതത്വത്തിന് രൂപംകൊടുക്കുക, ഒന്നിലധികം വെബ് സൈറ്റുകളുടെ നിർമാണം സാധ്യമാക്കുക എന്നിങ്ങനെയുള്ള വ്യവസ്ഥകൾ വെബ് സെർവർ സോഫ്റ്റ്‌വെയർ തന്നെ കൈകാര്യം ചെയ്യും.

4.2.2 സോഫ്റ്റ്‌വെയർ പോർട്ടുകൾ (Software ports)

കമ്പ്യൂട്ടർ സിസ്റ്റത്തിലെ ഹാർഡ്‌വെയർ പോർട്ടുകളെക്കുറിച്ച് 11-ാം ക്ലാസിൽ നാം പഠിച്ചിട്ടുണ്ട്. പുറമേ നിന്നുള്ള ഉപകരണങ്ങൾ കമ്പ്യൂട്ടറുമായി ബന്ധിപ്പിക്കുവാൻ ഹാർഡ്‌വെ

യർ പോർട്ടുകൾ ഉപയോഗിക്കുന്നു. ഈ പോർട്ടുകൾ ഉപയോഗിച്ചാണ് ഇത്തരം ഉപകരണങ്ങൾ കമ്പ്യൂട്ടറുമായി ആശയവിനിമയം നടത്തുന്നത്. ഉദാഹരണമായി VGA പോർട്ട് മോണിട്ടറുമായി ബന്ധിപ്പിക്കുന്നതിനും, PS/2 പോർട്ട് കീബോർഡ്/മൗസ് എന്നിവയുമായി ബന്ധിപ്പിക്കുന്നതിനും ഉപയോഗിക്കുന്നു. അതുപോലെ, ഒരു സോഫ്റ്റ്‌വെയർ പോർട്ട് ഉപയോഗിച്ചുകൊണ്ട് ക്ലയന്റ് കമ്പ്യൂട്ടറിനെ സെർവറുമായി ബന്ധിപ്പിച്ച്, അതിലെ സേവനങ്ങളായ HTTP, FTP, SMTP തുടങ്ങിയവ ഉപയോഗിക്കാനാകും. പോർട്ടുകളെ തമ്മിൽ തിരിച്ചറിയുന്നതിനായി അവയ്ക്ക് അതിവിശിഷ്ടമായ സംഖ്യാനാമം നൽകുന്നു. ഇ-മെയിൽ, ഫയൽ കൈമാറ്റം തുടങ്ങിയ ഒരേ സെർവർ കമ്പ്യൂട്ടറിൽ തന്നെ നടക്കുന്ന വ്യത്യസ്ത സേവനങ്ങളെ തിരിച്ചറിയുക എന്നതാണ് സോഫ്റ്റ്‌വെയർ പോർട്ടിന്റെ ഉദ്ദേശ്യം. വ്യത്യസ്ത പോർട്ട് സംഖ്യാനാമങ്ങളുപയോഗിച്ച്, സെർവറിൽ ലഭ്യമായ ഓരോ സേവനവും തുടങ്ങാനും, ഉപയോഗിക്കാനും/എടുക്കാനും സാധിക്കുന്നു. പോർട്ട് സംഖ്യാനാമം എന്നത് ഒരു 16-ബിറ്റ് സംഖ്യയാണ്. അത് കമ്പ്യൂട്ടറിന്റെ IP മേൽവിലാസത്തോടൊപ്പം (URL-ന് ഒപ്പം) ചേർത്തുപയോഗിച്ച് സെർവറിൽ ലഭ്യമായ ഒരു പ്രത്യേക സേവനവുമായി ആശയവിനിമയം സാധ്യമാക്കുന്നു. ഒരു വെബ്സൈറ്റിലുള്ള സേവനങ്ങൾ താഴെ തന്നിരിക്കുന്ന വിധം ഉപയോഗിക്കാൻ സാധിക്കും.

http://google.co.in:80

ഇവിടെ http എന്നത് പ്രോട്ടോകോളും, google.co.in എന്നത് ഡൊമൈൻ നാമവും 80 എന്നത് പോർട്ടിന്റെ സംഖ്യാനാമവുമാണ്. പട്ടിക 4.1-ൽ അറിയപ്പെടുന്ന ചില പോർട്ട് സംഖ്യാനാമങ്ങളും അനുബന്ധ സേവനങ്ങളും കൊടുത്തിരിക്കുന്നു.

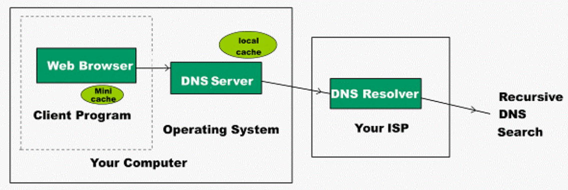
തനത് പോർട്ട് സംഖ്യ	സേവനം
20 & 21	ഫയൽ കൈമാറുന്നതിനുള്ള ഫയൽ ട്രാൻസ്ഫർ പ്രോട്ടോകോൾ (FTP)
22	സെക്യൂർ ഷെൽ (SSH)
25	ലഘു സന്ദേശങ്ങൾ കൈമാറുന്നതിനുള്ള സിമ്പിൾ മെയിൽ ട്രാൻസ്ഫർ പ്രോട്ടോകോൾ (SMTP)
53	ഡൊമൈൻ നെയിം സിസ്റ്റം (DNS)
80	ഹൈപ്പർ ടെക്സ്റ്റ് ട്രാൻസ്ഫർ പ്രോട്ടോകോൾ (HTTP)
110	പോസ്റ്റ് ഓഫീസ് പ്രോട്ടോകോൾ (POP3)
443	HTTP സെക്യൂർ (HTTPS)

പട്ടിക 4.1: പോർട്ടുകളും അവയുടെ സേവനങ്ങളും

4.2.3 DNS സെർവറുകൾ (DNS servers)

ചിത്രം 4.6: ലോകമെമ്പാടുമുള്ള റൂട്ട് സെർവറുകൾ

11-ാം ക്ലാസിൽ കമ്പ്യൂട്ടർ ശൃംഖലകൾ എന്ന അധ്യായത്തിൽ ഡൊമൈൻ നാമകരണ രീതിയെക്കുറിച്ച് നാം പഠിച്ചിട്ടുണ്ട്. ഡൊമൈൻ നാമങ്ങളുടെ ഡാറ്റബേസും അനുബന്ധ IP മേൽവിലാസങ്ങളും ഉള്ള ഒരു പ്രത്യേക നെറ്റ്‌വർക്കിങ് സോഫ്റ്റ്‌വെയർ ഓരോ DNS സെർവറിലും പ്രവൃത്തിക്കുന്നുണ്ടാവും. ഡൊമൈൻ നാമകരണ സമ്പ്രദായം ഒരു DNS സെർവറിൽ പ്രവർത്തിക്കുകയും ക്ലയന്റ് കമ്പ്യൂട്ടർ നൽകുന്ന അപേക്ഷയ്ക്കനുസരിച്ച് ഡൊമൈൻ നാമത്തിന്റെ IP മേൽവിലാസം തിരിച്ച് നൽകുകയും ചെയ്യുന്നു.



ചിത്രം 4.7 : DNS തിരയൽ

ഒരു ഡൊമൈൻ നാമകരണ സമ്പ്രദായത്തിൽ ധാരാളം DNS സെർവറുകളുണ്ടായിരിക്കും. ഇവ സ്ഥാനക്രമമനുസരിച്ച് ക്രമീകരിച്ചിരിക്കും. .com, .org തുടങ്ങിയ ഉയർന്നതും മുകൾ തട്ടിലുള്ളതുമായ ഡൊമൈനുകളോ, രാജ്യങ്ങളെ സൂചിപ്പിക്കുന്ന ഡൊമൈൻ നാമങ്ങളായ .in അല്ലെങ്കിൽ .uk തുടങ്ങിയവയോ ഉൾപ്പെടുന്ന 13 റൂട്ട് സെർവറുകളായിരിക്കും ഏറ്റവും മുകളിലുള്ളത്. ചിത്രം 4.6-ൽ കാണിച്ചിരിക്കുന്നത് പോലെ ലോകമെമ്പാടുമായി പലസ്ഥലങ്ങളിൽ റൂട്ട് സെർവറുകളുടെ ധാരാളം പകർപ്പുകൾ സ്ഥാപിച്ചിട്ടുണ്ടാകും. മറ്റെല്ലാ DNS സെർവറുകളും, സ്ഥാനക്രമത്തിന്റെ താഴെത്തലങ്ങളിൽ പ്രതിഷ്ഠിച്ചിരിക്കും. DNS എങ്ങനെയാണ് ഡൊമൈൻ നാമത്തിന്റെ IP മേൽവിലാസം തിരഞ്ഞെടുക്കുന്നതെന്ന് നമുക്ക് നോക്കാം.

കേരളത്തിലെ പോലീസ് വകുപ്പിന്റെ വെബ് സൈറ്റ് നാം സന്ദർശിക്കുന്നു എന്ന് കരുതുക. www.keralapolice.org എന്ന പേരിലുള്ള പോലീസ് വകുപ്പിന്റെ ഡൊമൈൻ നാമം നമുക്ക് ബ്രൗസറിൽ ടൈപ്പ് ചെയ്യാം. DNS എങ്ങനെ IP മേൽവിലാസം കണ്ടെത്തുന്നു. എന്ന് താഴെ തന്നിട്ടുള്ള ഘട്ടങ്ങൾ വിശദീകരിക്കുന്നു. ചിത്രം 4.7 കാണുക.

1. എല്ലാ ബ്രൗസറുകളും അതിന്റെ ക്യാഷ് മെമ്മറിയിൽ സമീപകാലത്ത് സന്ദർശിച്ച IP മേൽവിലാസങ്ങൾ സൂക്ഷിക്കുന്നുണ്ടാകും. അതിനാൽ, ബ്രൗസർ ആദ്യം പ്രാദേശിക മെമ്മറി (മിനി ക്യാഷ്) പരിശോധിച്ച് IP മേൽവിലാസമുണ്ടോ എന്ന് ഉറപ്പ് വരുത്തുന്നു. ഉണ്ടെങ്കിൽ അതുപയോഗിക്കുന്നു.

2. ബ്രൗസറിന്റെ ക്യാഷ് മെമ്മറിയിൽ IP മേൽവിലാസം ഇല്ലെങ്കിൽ അത് ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ പ്രാദേശിക ക്യാഷ് മെമ്മറി പരിശോധിക്കുന്നു.

3. അവിടെയുമില്ലെങ്കിൽ ISP-യുടെ പ്രാദേശിക DNS സെർവറിൽ പരിശോധിക്കുന്നു.

4. ISP യുടെ DNS സെർവറിലും ഡൊമൈൻ നാമമില്ല എങ്കിൽ ISP യുടെ DNS സെർവർ, റൂട്ട് സെർവറിൽ തുടങ്ങി ആവർത്തിച്ചുള്ള തിരച്ചിലിലൂടെ അത് കണ്ടെത്തുന്നു.

5. ISP-യുടെ DNS സെർവർ, IP മേൽവിലാസം ബ്രൗസറിന് തിരിച്ച് നൽകുന്നു.

6. www.keralapolice.gov യുടെ IP മേൽവിലാസം ഉപയോഗിച്ചുകൊണ്ട് ബ്രൗസർ, വെബ് സെർവറുമായി ബന്ധപ്പെടുകയും, ബ്രൗസർ വിൻഡോയിൽ വെബ് പേജ് കാണിക്കുകയും ചെയ്യുന്നു. IP മേൽവിലാസം കിട്ടിയില്ലായെങ്കിൽ 'സെർവർ കണ്ടെത്തിയില്ല' എന്ന സന്ദേശം ബ്രൗസർ വിൻഡോയിലെത്തിക്കുന്നു.

ഡൊമൈൻ നാമങ്ങളുടെയും അനുബന്ധ മേൽവിലാസങ്ങളുടെയും മൊത്തമായുള്ള ഡാറ്റാബേസ് ലോകമെമ്പാടുമുള്ള 13 റൂട്ട് സെർവറുകളാണ് കൈകാര്യം ചെയ്യുന്നത്. ഓരോ റൂട്ട് സെർവറും പലരാജ്യങ്ങളിൽ സ്ഥാപിച്ചിട്ടുള്ള സെർവറുകളുടെ ഒരു ശൃംഖലയിലാണുള്ളത്. അവയ്ക്ക് A, B, C, D, E, F, G, H, I, J, K, L, M എന്നിങ്ങനെ നാമകരണം നൽകിയിരിക്കുന്നു. ICANN, NASA (നാഷണൽ എറോനോട്ടിക്സ് ആന്റ് സ്പേസ് അഡ്മിനിസ്ട്രേഷൻ), മേരിലാന്റ് സർവകലാശാല, വെരിസൈൻ ഇൻകോർപറേഷൻ തുടങ്ങിയ സ്ഥാപനങ്ങളാണ് ഈ സെർവറുകളെ പരിപാലിക്കുന്നത്. റൂട്ട് സെർവറുകളെ പരിപാലിക്കുന്ന സ്ഥാപനങ്ങൾ ചേർന്നാണ് ICANN (ഇന്റർനെറ്റ് കോർപറേഷൻ ഫോർ അസൈന്റ് നെയിംസ് ആന്റ് നമ്പേഴ്സ്) റൂട്ട് സെർവർ സമ്പ്രദായ ഉപദേശക സമിതി ഉണ്ടായിരിക്കുന്നത്. ഇവരാണ് ഇന്റർനെറ്റിലെ റൂട്ട് സെർവർ സമ്പ്രദായത്തിന്റെ പ്രവർത്തനം, കാര്യനിർവഹണം, സുരക്ഷിതത്വം, സമ്പൂർണത ഉറപ്പാക്കൽ തുടങ്ങിയവയുമായി ബന്ധപ്പെട്ട ഉപദേശങ്ങൾ ICANN ന് നൽകുന്നത്. ഇന്ത്യയിൽ NIXI (നാഷണൽ ഇന്റർനെറ്റ് എക്സ്ചേഞ്ച് ഓഫ് ഇന്ത്യ) യാണ് മുംബൈ (I റൂട്ട്), ഡൽഹി (K റൂട്ട്), ചെന്നൈ (F റൂട്ട്) എന്നിവിടങ്ങളിൽ റൂട്ട് സെർവറുകൾ നിർമിച്ച് നൽകിയിട്ടുള്ളത്.

നൂറ് കണക്കിന് കമ്പ്യൂട്ടറുകളും ഉപകരണങ്ങളും ഇന്റർനെറ്റുമായി ബന്ധിപ്പിച്ചിട്ടുള്ള വിദ്യാഭ്യാസസ്ഥാപനങ്ങൾ, സർക്കാർ വകുപ്പുകൾ, സോഫ്റ്റ്‌വെയർ നിർമാണശാലകൾ തുടങ്ങിയവർ അവരുടെ ഇൻട്രാനെറ്റിൽ സ്വന്തം DNS സെർവർ സ്ഥാപിച്ചിട്ടുണ്ടാകും. ഉപയോക്താക്കൾ സ്ഥിരമായി ഉപയോഗിക്കുന്ന ഡൊമൈൻനാമങ്ങളും അവയുടെ IP മേൽവിലാസങ്ങളും ഉൾപ്പെടുന്ന ഒരു പട്ടിക ഈ പ്രാദേശിക DNS സെർവറിലുണ്ടാകും. പുതിയ ഡൊമൈൻ നാമങ്ങൾ ഉൾപ്പെടുത്തിക്കൊണ്ട് ഈ പട്ടിക സമയാസമയം നവീകരിക്കാറുമുണ്ട്. ഇന്റർനെറ്റിലെ കമ്പ്യൂട്ടർ ഒരു വെബ്സൈറ്റ് ഉപയോഗിക്കാൻ ശ്രമിക്കുമ്പോൾ പ്രസ്തുത വെബ് സൈറ്റിന്റെ ഡൊമൈൻ നാമവുമായി ബന്ധപ്പെട്ട IP മേൽവിലാസത്തിനായി ആദ്യം പ്രാദേശിക DNS സെർവറിൽ അന്വേഷിച്ച് കണ്ടെത്തുന്നു. ഇത് സ്ഥാപനത്തിന്റെ ഇന്റർനെറ്റ് വേഗത വർധിപ്പിക്കുന്നു. ഈ രീതിയിൽ IP മേൽവിലാസം കണ്ടെത്താനായില്ലായെങ്കിൽ ISP യുടെ DNS സെർവറിൽ അന്വേഷിക്കുന്നു.

നിലവിലുള്ള നമ്മുടെ DNS ദാതാവിന് പകരം ഉപയോഗിക്കാവുന്ന സ്വതന്ത്ര DNS സേവനമാണ് ഗൂഗിളിന്റെ പൊതു DNS. ഇതിന്റെ IP മേൽവിലാസങ്ങളാണ് 8.8.8.8, 8.8.4.4 എന്നിവ. ഈ പൊതു DNS ന്റെ IP മേൽവിലാസങ്ങളിൽ ഏതിലേക്കെങ്കിലും നമ്മുടെ കമ്പ്യൂട്ടറിനെ നയിക്കുന്നരീതിയിൽ നമ്മുടെ ശൃംഖലയെ ക്രമീകരിക്കാവുന്നതാണ്.

നിങ്ങളുടെ പുരോഗതി അറിയുക

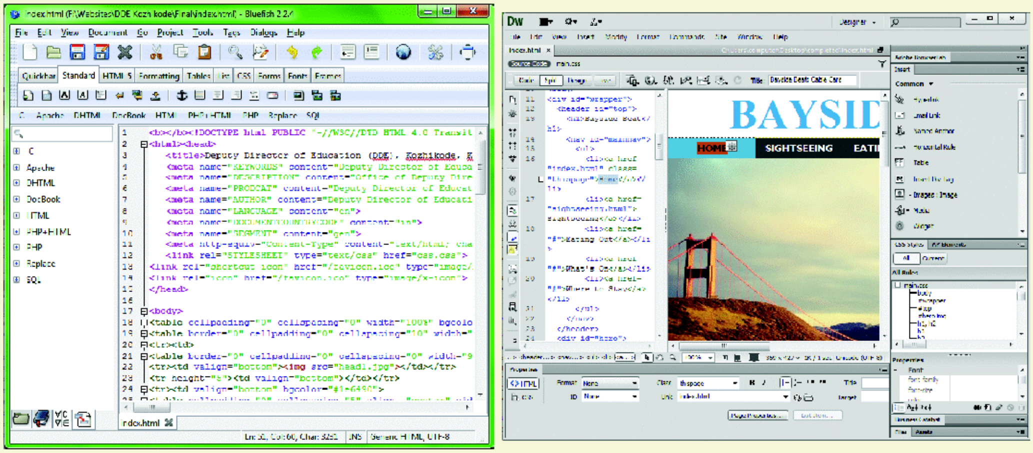
1. ഇന്റർനെറ്റ് പ്രോട്ടോകോളിന്റെ (IP) ഉയർന്ന തലത്തിൽ പ്രവർത്തിക്കുന്ന ഒരു പ്രോട്ടോകോളിന്റെ പേരെഴുതുക.
2. HTTPS എന്നതിന്റെ പൂർണ രൂപം എഴുതുക.
3. ഡാറ്റ സെന്ററുകളിൽ വെബ്സെർവറുകൾ സ്ഥാപിക്കുന്നതിന്റെ നേട്ടങ്ങൾ എന്തെല്ലാം?
4. ശരിയോ തെറ്റോ എന്ന് പ്രസ്താവിക്കുക.
   a. ക്ലയന്റ്-സെർവർ മാതൃക അനുസരിച്ചാണ് വെബ് സെർവർ സോഫ്റ്റ്വെയർ പ്രവർത്തിക്കുന്നത്.
   b. സെർവർ ഓപ്പറേറ്റിംഗ് സിസ്റ്റവും വെബ് സെർവർ സോഫ്റ്റ്വെയറും ചേർന്നതാണ് വെബ് സെർവർ.
5. ഒരു സോഫ്റ്റ്വെയർ പോർട്ട് സംഖ്യാനാമസമ്പ്രദായത്തിലെ ബിറ്റുകളുടെ എണ്ണം ________ ആണ്.
   a. 8 b. 16 c. 32 d. 64
6. ഡൊമൈൻ നാമകരണസമ്പ്രദായം ഡൊമൈൻ നാമത്തിന്റെ ____________ തിരികെ നൽകുന്നു.

4.3 വെബ് രൂപകൽപ്പന ചെയ്യാം (Web designing)

വെബ് സൈറ്റ് നിർമാണത്തിന്റെ ആദ്യഘട്ടം വെബ് പേജുകളുടെ ആസൂത്രണമാണ്. നമ്മുടെ വിദ്യാലയത്തിനായി ഒരു വെബ്സൈറ്റ് വികസിപ്പിക്കുന്നു എന്ന് കരുതുക. വെബ് പേജുകളും അവ തമ്മിലുള്ള ബന്ധങ്ങളും തീരുമാനിച്ച് കഴിഞ്ഞാൽ അവയുടെ രൂപകൽപ്പനയിലേക്ക് കടക്കാം. ഇതിനായി ഏതെങ്കിലും ഒരു ടെക്സ്റ്റ് എഡിറ്റർ ഉപയോഗിക്കാവുന്നതാണ്. ഒരു ഹോം പേജും, സ്കൂളിൽ ലഭ്യമാകുന്ന കോഴ്സുകൾ, സൗകര്യങ്ങൾ, ബന്ധപ്പെടേണ്ട മേൽവിലാസം തുടങ്ങിയവ ദൃശ്യമാക്കുന്ന മറ്റൊരു പേജും നിർമിച്ച് അവ തമ്മിൽ മെനു ഉപയോഗിച്ച് ബന്ധപ്പെടുത്തുന്ന രീതിയിൽ മനോഹരമായ ഒരു വെബ് സൈറ്റ് നമുക്ക് നിർമിക്കാം.

വെബ് രൂപകൽപ്പന ചെയ്യുന്നതിന് നിരവധി സോഫ്റ്റ്വെയറുകൾ ലഭ്യമാണ്. വേഡ് പ്രൊസ്സസർ ഉപയോഗിച്ച് അനായാസമായി ഒരു ഡോക്യുമെന്റ് തയ്യാറാക്കുന്നത് പോലെ, വെബ് നിർമാണ ടൂളുകളിൽ ലഭ്യമായിട്ടുള്ള സങ്കേതങ്ങളുടെ സഹായത്തോടെ വെബ് പേജുകൾ നിർമിക്കാവുന്നതാണ്. വെബ് പേജുകൾ രൂപകൽപന ചെയ്യുന്നതിനും അവയെ പരസ്പരം ബന്ധിപ്പിച്ച് കൊണ്ട് വെബ്സൈറ്റ് രൂപീകരിക്കുന്നതിനുമുള്ള സംവിധാനങ്ങൾ ഈ സോഫ്റ്റ്വെയറുകൾ നൽകുന്നു. FTP പ്രോട്ടോകോളുകൾ ഉപയോഗിച്ച് ഫയലുകൾ സെർവറുകളിലേക്ക് കൈമാറ്റം ചെയ്യുന്നതിനുള്ള സ്വാഭാവിക സൗകര്യം ഇത്തരം സോഫ്റ്റ്വെയറുകളിൽ ഉണ്ടാകും. ബ്ലൂഫിഷ്, ബൂട്ട് സ്ട്രാപ്പ്, അഡോബ് ഡ്രീംവീവർ, മൈക്രോസോഫ്റ്റ് എക്സ്പ്രഷൻ വെബ് തുടങ്ങിയവ വെബ് നിർമാണത്തിന് ഉപയോഗിക്കുന്ന പ്രമുഖ സോഫ്റ്റ്വെയറുകളാണ്, ചിത്രം 4.8-ൽ വെബ് നിർമാണത്തിന് ഉപയോഗിക്കുന്ന

പ്രശസ്തമായ ചില സോഫ്റ്റ്‌വെയറുകളുടെ IDE (ഇന്റഗ്രേറ്റഡ് ഡവലപ്മെന്റ് എൻവയോൺമെന്റ്) കാണിച്ചിരിക്കുന്നു.



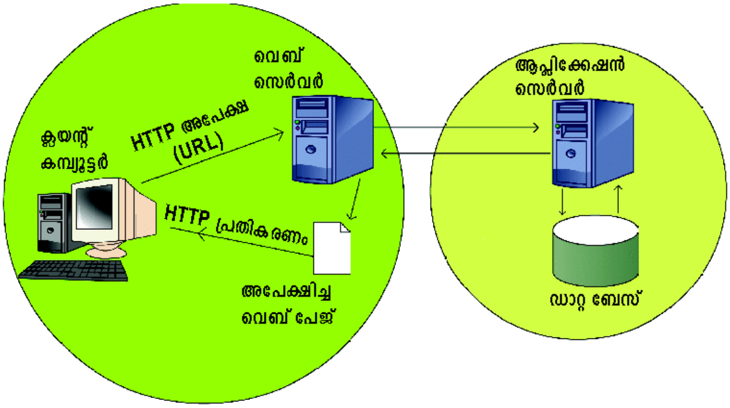
ബ്ലൂഫിഷ്   ഡ്രീം വീവർ

ചിത്രം 4.8 : വെബ് രൂപകൽപനക്കുപയോഗിക്കുന്ന സോഫ്റ്റ്‌വെയറുകളുടെ IDE

വെബ് സൈറ്റുകളുടെ നിർമാണത്തെക്കുറിച്ച് ഹൈസ്കൂൾ ക്ലാസുകളിൽ നിങ്ങൾ പഠിച്ചിട്ടുണ്ട്. HTML-ൽ ഉള്ള ടാഗുകളും അവയുടെ ആട്രിബ്യൂട്ടുകളും ഉപയോഗിച്ച് വെബ് പേജുകൾ രൂപകൽപ്പന ചെയ്യാം.

4.4 സ്റ്റാറ്റിക്, ഡൈനാമിക് വെബ് പേജുകൾ (Static and dynamic web pages)

ചെറുകിട വ്യവസായങ്ങൾ, വിദ്യാലയങ്ങൾ തുടങ്ങിയവയുടെ വെബ് സൈറ്റുകൾ എപ്പോൾ സന്ദർശിച്ചാലും, പേജുകളിലെ ഉള്ളടക്കം സ്ഥിരമായി (സ്റ്റാറ്റിക് ആയി) നിൽക്കുന്നത് ശ്രദ്ധിച്ചു കാണുമല്ലോ? ഇത്തരം വെബ് സൈറ്റുകളെ സ്റ്റാറ്റിക് വെബ്സൈറ്റുകൾ എന്ന് വിളിക്കുന്നു. എസ്.എസ്.എൽ.സി/ഹയർസെക്കന്ററി പരീക്ഷ എഴുതിയ വിദ്യാർഥികൾ തങ്ങളുടെ രജിസ്റ്റർ നമ്പർ നൽകുമ്പോൾ, അവരവരുടെ പരീക്ഷാഫലം ദൃശ്യമാകുംവിധം ഉള്ളടക്കം മാറിവരുന്ന വെബ് സൈറ്റുകളും നിങ്ങൾ കണ്ടിട്ടുണ്ടാകും. ഇവ ഡൈനാമിക് പേജുകൾ എന്നറിയപ്പെടുന്നു. പ്രോഗ്രാം കോഡിൽ മാറ്റം വരുത്താത്തിടത്തോളം ഉള്ളടക്കം സ്ഥിരമായിത്തന്നെ നിലനിൽക്കുന്ന പേജുകളാണ് സ്റ്റാറ്റിക് വെബ് പേജുകൾ. ആദ്യകാലത്ത് HTML മാത്രം ഉപയോഗിച്ചാണ് ഈ സ്റ്റാറ്റിക് വെബ് പേജുകൾ നിർമിച്ചിരുന്നത്. പിന്നീട് ജാവ സ്ക്രിപ്റ്റ്, വിബി സ്ക്രിപ്റ്റ് തുടങ്ങിയ സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ വരവോടെ വെബ് പേജുകൾക്ക് ചലനാത്മകത കൈവന്നു. വെബ് പേജിന്റെ ചില ഭാഗങ്ങളിൽ മൗസ് പോയിന്റർ വയ്ക്കുമ്പോൾ ആ ഭാഗത്ത് നിറവ്യത്യാസവും രൂപമാറ്റവും സംഭവിക്കുന്നതും, ചിത്രങ്ങൾ ഒന്നിന് പിറകേ ഒന്നായി ക്രമീകരിക്കപ്പെടുന്നതുമെല്ലാം സ്ക്രിപ്റ്റിംഗ് ഭാഷ ഉപയോഗിക്കുന്നതു കൊണ്ടാണ്. ഇത്തരം പ്രത്യേകതകളുള്ള വെബ് പേജുകളും സ്റ്റാറ്റിക് വെബ് പേജുകളായിട്ടാണ് കണക്കാക്കിയിട്ടുള്ളത്.



സ്റ്റാറ്റിക് വെബ് പേജിനായുള്ള നിർദേശം   ഡൈനാമിക് വെബ് പേജിനായുള്ള നിർദേശം

ചിത്രം 4.9 : സ്റ്റാറ്റിക്, ഡൈനാമിക് എന്നീ വെബ് പേജുകൾക്കായുള്ള നിർദേശങ്ങൾ

ഓരോ തവണ വെബ് പേജ് എടുക്കുമ്പോഴും സെർവർ ഭാഗത്ത് പേജിലുള്ള കോഡ് അത് നിർമിക്കുന്നു എങ്കിൽ അത്തരം പേജിനെ ഡൈനാമിക് വെബ് പേജ് എന്ന് വിളിക്കുന്നു. വെബ് സെർവറിൽ സ്ഥാപിച്ചിട്ടുള്ള ആപ്ലിക്കേഷൻ പ്രോഗ്രാം ഉപയോഗിച്ച് ഡൈനാമിക് പേജുകൾ പ്രവർത്തിപ്പിക്കുന്നു. വെബ് പേജിലെ സ്ക്രിപ്റ്റ്, വെബ് സെർവറിൽ പ്രവർത്തിക്കുമ്പോൾ ഉണ്ടാകുന്ന HTML പേജുകൾ ക്ലയന്റ് ബ്രൗസറിലേക്കയക്കപ്പെടുന്നു. മിക്കവാറും സന്ദർഭങ്ങളിൽ ഇത്തരം വെബ് പേജുകൾ നിർമിക്കാൻ ഡാറ്റബേസിലുള്ള ഡാറ്റയാണ് ഉപയോഗിക്കുന്നത്. SSLC/HSE പരീക്ഷാഫലങ്ങൾ കാണിക്കുന്ന വെബ് സൈറ്റുകൾ, റോഡ്, റെയിൽ, വ്യോമ യാത്രകൾക്കുള്ള ടിക്കറ്റ് മുൻകൂട്ടി ഉറപ്പിക്കാനാകുന്ന സൈറ്റുകൾ, പണമിടപാടുകൾക്കുള്ള വെബ് സൈറ്റുകൾ തുടങ്ങിയവയുടെ പേജുകളിൽ പരിവർത്തനാത്മക ഉള്ളടക്കമുണ്ടാവുകയും അവ ഡൈനാമിക് പേജുകളായി കണക്കാക്കുകയും ചെയ്യുന്നു. PHP, ASP, JSP തുടങ്ങിയവ ഉപയോഗിച്ച് ഡൈനാമിക് വെബ് പേജുകൾ നിർമിക്കുന്നു. സ്റ്റാറ്റിക്, ഡൈനാമിക് എന്നീ വെബ് പേജുകൾ തമ്മിലുള്ള ഒരു താരതമ്യം പട്ടിക 4.2-ൽ തന്നിട്ടുണ്ട്.

സ്റ്റാറ്റിക് വെബ് പേജുകൾ	ഡൈനാമിക് വെബ്പേജുകൾ
വെബ് പേജിന്റെ ഉള്ളടക്കവും രൂപരേഖയും സ്ഥിരമായിരിക്കും.	ഉള്ളടക്കവും രൂപരേഖയും പ്രോഗ്രാം പ്രവർത്തിക്കുമ്പോൾ വ്യത്യാസപ്പെടുന്നു.
സ്റ്റാറ്റിക് വെബ് പേജുകൾ ഒരിക്കലും ഡാറ്റാബേസ് ഉപയോഗിക്കുന്നില്ല.	ഡാറ്റാബേസ് ഉപയോഗിച്ച് ക്വറികളുടെ സഹായത്താൽ പരിവർത്തനാത്മക ഉള്ളടക്കം സൃഷ്ടിക്കുന്നു.
സ്റ്റാറ്റിക് വെബ് പേജുകൾ ബ്രൗസറിൽ നേരിട്ട് പ്രവർത്തിക്കുന്നതിനാൽ സെർവർ ഭാഗത്തുള്ള ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകളുടെ ആവശ്യമില്ല.	ഡൈനാമിക് വെബ് പേജുകൾ സെർവർ ഭാഗത്തുള്ള ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകളുടെ സഹായത്താൽ പ്രവർത്തിച്ച് ഫലങ്ങൾ പ്രദർശിപ്പിക്കുന്നു.
സ്റ്റാറ്റിക് വെബ് പേജുകൾ എളുപ്പത്തിൽ നിർമിക്കാനാകും	ഡൈനാമിക് വെബ് പേജുകളുടെ നിർമാണത്തിന് പ്രോഗ്രാമിങ് വൈദഗ്ധ്യം ആവശ്യമാണ്.

പട്ടിക 4.2: സ്റ്റാറ്റിക്, ഡൈനാമിക് എന്നീ വെബ് പേജുകളുടെ താരതമ്യം

4.5 സ്ക്രിപ്റ്റുകൾ (Scripts)

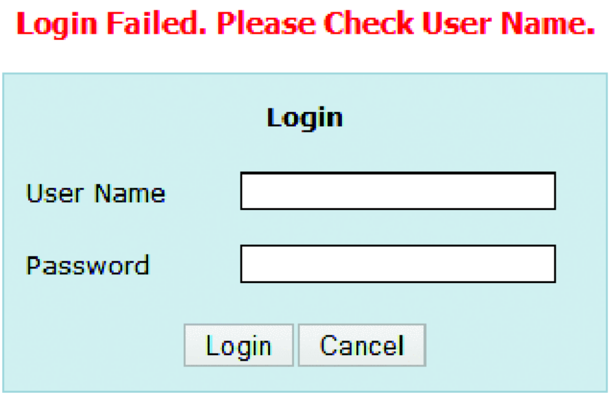
HTML പേജുകളിൽ ഉൾക്കൊള്ളിക്കുന്ന പ്രോഗ്രാം കോഡുകളാണ് സ്ക്രിപ്റ്റുകൾ. നോട്ട് പാഡ് പോലുള്ള ഒരു ടെക്സ്റ്റ് എഡിറ്റർ ഉപയോഗിച്ചാണ് അവ തയാറാക്കുന്നത്. ഡൈനാമിക് വെബ് പേജുകൾ നിർമ്മിക്കുന്നതിനായി ജാവ സ്ക്രിപ്റ്റ്, വി.ബി. സ്ക്രിപ്റ്റ്, പി.എച്ച്.പി, പേൾ തുടങ്ങിയ സ്ക്രിപ്റ്റിങ് ഭാഷകൾ ഉപയോഗിക്കുന്നു. ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന്റെ സഹായത്തോടെ കമ്പ്യൂട്ടർ ഹാർഡ്‌വെയറിനെക്കൊണ്ട് നിർദേശങ്ങൾ നടപ്പിലാക്കിക്കുന്നവയാണ് സാമ്പ്രദായിക പ്രോഗ്രാമിങ് ഭാഷകൾ. എന്നാൽ വെബ് സെർവറുകളാൽ/വെബ് ബ്രൗസറുകളാൽ വ്യാഖ്യാനിക്കപ്പെടുന്നവയാണ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ. ഇന്ന്, ഭൂരിഭാഗം ഏക കമ്പ്യൂട്ടർ അധിഷ്ഠിത പ്രോഗ്രാമുകളും വെബ് അധിഷ്ഠിത പ്രോഗ്രാമുകളാൽ പുനസ്ഥാപിക്കപ്പെട്ടിട്ടുണ്ട്. ധനകാര്യസ്ഥാപനങ്ങളിലും മറ്റും ഉപയോഗിച്ചിരുന്ന സോഫ്റ്റ്‌വെയറുകൾ അതാത് ശാഖകളിൽത്തന്നെ സ്ഥാപിച്ചിരുന്നവയാണ്. എന്നാൽ ഇന്ന് ഈ സ്ഥാപനങ്ങൾ തങ്ങളുടെ സോഫ്റ്റ്‌വെയർ വെബ് സെർവറിൽ ലഭ്യമാക്കുകയും അത് ഇന്റർനെറ്റിന്റെ സഹായത്താൽ ജീവനക്കാർ ഉപയോഗിക്കുകയും ചെയ്യുന്നു. ധനവിനിമയ സ്ഥാപനത്തിന് ആവശ്യമായ സോഫ്റ്റ്‌വെയറുകൾ, ഹയർസെക്കന്ററി പ്രവേശനത്തിനുള്ള സോഫ്റ്റ്‌വെയർ തുടങ്ങിയ വെബ് അധിഷ്ഠിത സോഫ്റ്റ്‌വെയറുകളുടെ നിർമാണത്തിന് സ്ക്രിപ്റ്റിങ് ഭാഷകൾ ഉപയോഗിക്കുന്നു. ഒരു HTML പേജിൽ **<SCRIPT>**, **</SCRIPT>** എന്നീ ടാഗുകൾക്ക് ഉള്ളിലായാണ് സ്ക്രിപ്റ്റ് എഴുതുന്നത്. ഒരു HTML ഡോക്യുമെന്റിൽ സ്ക്രിപ്റ്റ് ഉൾപ്പെടുത്തുന്നതിനോ, സൂചിപ്പിക്കുന്നതിനോ ആണ് `<SCRIPT>` ടാഗ് ഉപയോഗിക്കുന്നത്. ഇതിന് **Type**, **Src** എന്നീ ആട്രിബ്യൂട്ടുകളുണ്ട്. സ്ക്രിപ്റ്റ് ടാഗിന് ഉള്ളിലുള്ള സ്ക്രിപ്റ്റിങ് ഭാഷ തിരിച്ചറിയാനായാണ് `Type` ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. സ്ക്രിപ്റ്റ് കോഡ് ഉൾക്കൊള്ളുന്ന ഫയലിന്റെ പേര് നൽകാനാണ് SRC ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. HTML ടാഗുകളെയും ആട്രിബ്യൂട്ടുകളെയും കുറിച്ച് നമുക്ക് തുടർന്നുള്ള ഭാഗങ്ങളിൽ മനസിലാക്കാം.

ഒരു HTML കോഡിനുള്ളിൽ ജാവ സ്ക്രിപ്റ്റ് ഉൾപ്പെടുത്താൻ `<SCRIPT Type="text/javascript">` ഉപയോഗിക്കുന്നു.

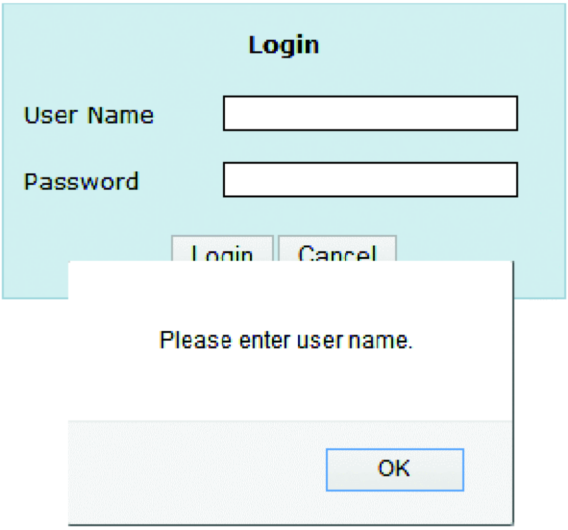
4.5.1 വിവിധതരം സ്ക്രിപ്റ്റിങ് ഭാഷകൾ (Types of scripting languages)

ഇന്റർനെറ്റിലെ ഒരു വെബ്സൈറ്റിന്റെ ലോഗിൻ പേജ് പരിഗണിക്കുക (സാധാരണയായി ഉപയോക്താവ് യൂസർ നാമവും, പാസ്‌വേഡും നൽകാനുപയോഗിക്കുന്ന പേജാണിത്). ഉപയോക്താവ് യൂസർ നാമവും പാസ്‌വേഡും നൽകിയശേഷം ലോഗിൻ ബട്ടണിൽ ക്ലിക്ക് ചെയ്യും. എന്നാൽ ഇവ നൽകാതെ ലോഗിൻ ബട്ടണിൽ ക്ലിക്ക് ചെയ്താൽ എന്ത് സംഭവിക്കും? സ്വാഭാവികമായും ഇവ നൽകിയിട്ടില്ലെന്ന് കമ്പ്യൂട്ടർ ഉപയോക്താവിനോട് പറയും. രണ്ട് കമ്പ്യൂട്ടറുകളാണ് ഈ വെബ് പേജ് നിയന്ത്രിക്കുന്നത് വെബ് പേജ് പ്രദർശിപ്പിക്കുന്ന ക്ലയന്റ് കമ്പ്യൂട്ടറും വെബ് പേജ് നൽകുന്ന സെർവർ കമ്പ്യൂട്ടറും. ഉപയോക്താവ് യൂസർ നാമം നൽകിയിട്ടുണ്ടോ എന്ന് എവിടെയാണ് പരിശോധിക്കുന്നത്. ക്ലയന്റ് കമ്പ്യൂട്ടറിലോ അതോ സെർവർ കമ്പ്യൂട്ടറിലോ? ധാരാളം ആളുകൾ ഒരേ വെബ്സൈറ്റ് സന്ദർശിക്കുന്നതിനാലും സെർവർ തന്നെ ഇത്തരം അപേക്ഷകളെല്ലാം കൈകാര്യം ചെയ്യേ

ണ്ടതിനാലും ക്ലയന്റ് കമ്പ്യൂട്ടറിനേക്കാൾ വളരെയധികം തിരക്കിലായിരിക്കും സെർവർ കമ്പ്യൂട്ടർ. ആയതിനാൽ ക്ലയന്റ് ഭാഗത്ത് ചെയ്യാവുന്ന പ്രവൃത്തികളെല്ലാം അവിടെത്തന്നെ ചെയ്യുന്നതാണ് ഉചിതം. അങ്ങനെ സെർവറിന്റെ ജോലിഭാരം കുറയ്ക്കാനാകും. ഈ പരിശോധന സെർവറിലാണ് നടക്കുന്നതെങ്കിൽ, ഉപയോക്താവ് നൽകിയ അപേക്ഷ ക്ലയന്റിൽ നിന്നും ഇന്റർനെറ്റിലൂടെ സെർവറിൽ അയ്ക്കേണ്ടിവരും. ചിലപ്പോഴെങ്കിലും സെർവറിലെത്താൻ ഇതിന് ഇന്റർനെറ്റിലൂടെ വളരെ ദൂരം സഞ്ചരിക്കേണ്ടിയും വരും. വളരെ അധികം ക്ലയന്റുകളിൽ നിന്ന് ലഭിക്കുന്ന ഇത്തരം അപേക്ഷകൾ വന്ന ക്രമത്തിൽ സെർവർ ശേഖരിക്കുന്നു. സെർവർ പ്രൊസസ്സ് ചെയ്യുന്നതുവരെ അയച്ച അപേക്ഷ നിരയിൽ കാത്ത് നിൽക്കുന്നു. ക്ലയന്റ് ഏതെങ്കിലും യൂസർ നാമമോ പാസ്‌വേഡോ നൽകിയിട്ടുണ്ടോയെന്ന് ക്രമം അനുസരിച്ച് സെർവർ പരിശോധിക്കുന്നു. നൽകിയിട്ടില്ലെങ്കിൽ, ചിത്രം 4.10-ൽ കാണിച്ചിട്ടുള്ളതുപോലെ ഉചിതമായ ഒരു സന്ദേശം സെർവർ ക്ലയന്റിലേക്ക് അയയ്ക്കുന്നു. ഈ സന്ദേശവും ഇന്റർനെറ്റിലൂടെ വളരെ ദൂരം സഞ്ചരിച്ച് ക്ലയന്റിലെത്തുന്നു. ചുരുക്കത്തിൽ യൂസർ നാമം നൽകാതെ ഉപയോക്താവ് സബ്മിറ്റ് ബട്ടണിൽ ക്ലിക്ക് ചെയ്താൽ യൂസർ നാമം നൽകിയിട്ടില്ല എന്ന സന്ദേശം ക്ലയന്റ് കമ്പ്യൂട്ടറിൽ ലഭിക്കുന്നതുവരെ അയാൾ കാത്തിരിക്കേണ്ടിവരുന്നു. കൂടാതെ ക്ലയന്റിൽ നിന്ന് സെർവറിലേക്കും തിരിച്ചും ഡാറ്റ സഞ്ചരിക്കുന്നതിനാൽ ശൃംഖലയുടെ ഗതാഗത തിരക്ക് വർധിക്കുകയും ചെയ്യുന്നു.



ചിത്രം 4.10 : സെർവറിന്റെ പ്രതികരണം



ചിത്രം 4.11 : ക്ലയന്റ് ബ്രൗസറിന്റെ പ്രതികരണം

ഈ പരിശോധന ക്ലയന്റ് ഭാഗത്തു തന്നെ നടക്കുകയാണെങ്കിൽ, ഉപയോക്താവ് സബ്മിറ്റ് ബട്ടൺ അമർത്തുമ്പോൾ, യൂസർ നാമമോ, പാസ്‌വേഡോ നൽകിയിട്ടുണ്ടോ എന്ന് ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റ് കോഡിന് തന്നെ പരിശോധിക്കാവുന്നതാണ്. അല്ലാത്തപക്ഷം ഉചിതമായ ഒരു സന്ദേശം ദൃശ്യമാക്കാവുന്നതുമാണ്. ഈ പ്രക്രിയയിൽ, ഡാറ്റ ഇന്റർനെറ്റിലൂടെ സെർവറിലേക്ക് സഞ്ചരിക്കുകയോ, ലളിതമായ ഈ ദൗത്യത്തിന് സെർവറിനെ

ശല്യപ്പെടുത്തുകയോ ചെയ്യുന്നില്ല. ഉപയോക്താവ് സബ്മിറ്റ് ബട്ടൺ അമർത്തുമ്പോൾ യൂസർനാമവും/പാസ്‌വേഡും നൽകിയിട്ടില്ല എന്ന സന്ദേശം നിമിഷനേരം കൊണ്ട് ചിത്രം 4.11-ൽ കാണുന്നത് പോലെ ഉപയോക്താവിനു ലഭിക്കും. ഇതിനായി ശൃംഖലയിലെ വിഭവങ്ങൾ ഉപയോഗിക്കേണ്ടിവരുന്നുമില്ല.

ഇനി നമുക്ക് മറ്റൊരു സാഹചര്യം പരിചയപ്പെടാം. ഒരു ഉപയോക്താവ് തെറ്റായ യൂസർ നാമവും പാസ്‌വേഡും നൽകിയെന്ന് കരുതുക. യൂസർ നാമവും പാസ്‌വേഡും നൽകി യിട്ടുണ്ടോ എന്ന് ക്ലയന്റ് കമ്പ്യൂട്ടറിന് പരിശോധിക്കുവാൻ കഴിഞ്ഞേക്കാം. എന്നാൽ ഇത് ശരിയാണോ എന്ന് പരിശോധിക്കുവാൻ കഴിയില്ല. യൂസർ നാമങ്ങളുടെയും അനുബന്ധ പാസ്‌വേഡുകളുടേയും വിശദാംശങ്ങൾ സെർവർ കമ്പ്യൂട്ടറിലാണ് സൂക്ഷിക്കുന്നത് എന്ന താണ് ഇതിന് കാരണം. ഇവയുടെ പരിശോധന സെർവർ ഭാഗത്ത് മാത്രമേ ചെയ്യാൻ സാധിക്കുകയുള്ളൂ. ഡാറ്റയുടെ സാധുത ഉറപ്പ് വരുത്തുന്നതിന് സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്ന സന്ദർഭമാണിത്. സെർവറിലുള്ള വിവരങ്ങൾ ഉപയോഗപ്പെടുത്തിക്കൊണ്ട് ഡാറ്റയുടെ സാധുത ഉറപ്പ് വരുത്തേണ്ടിവന്നാൽ അത് സെർവർ ഭാഗത്ത് തന്നെ ചെയ്യേ ണ്ടതുണ്ട്.

സ്ക്രിപ്റ്റിങ് ഭാഷകളെ ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ എന്നും സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ എന്നും തരംതിരിച്ചിരിക്കുന്നു എന്ന് ഇതിൽ നിന്നും വ്യക്തമാണല്ലോ? ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിങ് ഉപയോഗിക്കുന്നത് ക്ലയന്റ് ഭാഗത്ത് ചെയ്യേണ്ട ജോലി ബ്രൗസറിൽ പ്രവർത്തിപ്പിച്ച് ചെയ്യിക്കുന്നതിനാണ്. സെർവറിൽ പ്രവർത്തിക്കുന്ന സ്ക്രിപ്റ്റുകളെ സെർവർ ഭാഗത്തുള്ള സ്ക്രിപ്റ്റുകൾ എന്നു പറയുന്നു. സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ പ്രവർത്തിക്കുമ്പോൾ ലഭിക്കുന്ന ഔട്ട്പുട്ട് HTML പേജുകളുടെ രൂപത്തിൽ ഉള്ളതും ക്ലയന്റിലേക്ക് അയയ്ക്കാവുന്നതും ആയിരിക്കും.

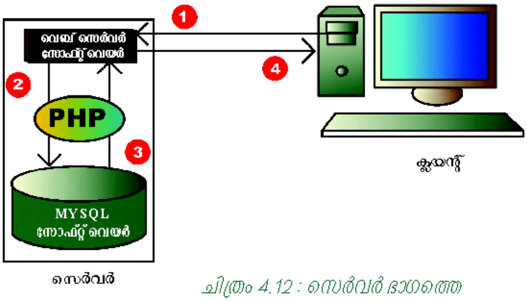
A. ക്ലയന്റ്ഭാഗത്തെ സ്ക്രിപ്റ്റിങ് (Client side scripting)

ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിങ്ങിൽ ഡാറ്റയുടെ സാധുത ഉറപ്പാക്കുന്ന സ്ക്രിപ്റ്റ് കോഡ് HTML കോഡിനൊപ്പം ബ്രൗസറിലേക്ക് ഡൗൺലോഡ് ചെയ്യപ്പെടുന്നു. ഡാറ്റ അസാധു ആണെങ്കിൽ ഉചിതമായ സന്ദേശം ദൃശ്യമാകുന്നു. ശരിയായ സാധുത ഉറപ്പാക്കിയശേഷം മാത്രമേ ഡാറ്റയെ വെബ് സെർവറിലേക്ക് അയയ്ക്കുകയുള്ളൂ.

സ്ക്രിപ്റ്റ്, ക്ലയന്റ് ബ്രൗസറിൽ പ്രവർത്തിക്കുന്നതിനാൽ അത് ഉപയോക്താക്കൾക്ക് ഉടൻ തന്നെ പ്രതികരണം നൽകുന്നു. വെബ് ആപ്ലിക്കേഷൻ പ്രവർത്തിക്കുമ്പോൾ, ചില ചുമ തലകൾ ക്ലയന്റ് ബ്രൗസറുമായി പങ്കുവയ്ക്കുവാൻ ഇത്തരം സ്ക്രിപ്റ്റിങ്ങിലൂടെ സാധി ക്കുന്നു. സ്ക്രിപ്റ്റുകളെ പിന്തുണയ്ക്കാത്ത ബ്രൗസറുകൾ ഉള്ളതാണ് ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിങ്ങിന്റെ പരിമിതി. ചില അവസരങ്ങളിൽ ബ്രൗസറിൽ സ്ക്രിപ്റ്റുകൾ പ്രവർത്തി ക്കുന്നതിനുള്ള അനുമതി ഉപഭോക്താക്കൾ നിഷേധിച്ചിട്ടുമുണ്ടാകും. അത്തരം സാഹച ര്യങ്ങളിൽ ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റ് പ്രവർത്തിക്കുകയില്ല. ജാവാസ്ക്രിപ്റ്റും വി.ബി. സ്ക്രിപ്റ്റും പ്രമുഖ സ്ക്രിപ്റ്റിങ് സാങ്കേതിക വിദ്യകളാണ്. ഡാറ്റ സെർവറിലേക്ക് അയയ്ക്കുന്നതിനു മുൻപ്, സാധുത പരിശോധനയും ലളിതമായ കണക്കുകൂട്ടലുകളും ചെയ്യുന്നതിനാണ് ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിങ് സാധാരണയായി ഉപയോഗിക്കുന്നത്.

B. സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിങ് (Server side scripting)

ഡൈനാമിക് വെബ് പേജുകളെക്കുറിച്ച് കഴിഞ്ഞ ഭാഗത്ത് നാം ചർച്ച ചെയ്തിട്ടുള്ളതാണ്. ഇത്തരം പേജുകൾ ഉണ്ടാക്കാൻ സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിങ് ഉപയോഗിക്കുന്നു. സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിങ്ങിനുള്ള മറ്റൊരു ഉദാഹരണം നോക്കാം. SSLC പരീക്ഷാഫലം പ്രസിദ്ധീകരിക്കുന്ന വെബ് സൈറ്റ് പരിഗണിക്കുക. നാം ഒരു കുട്ടിയുടെ രജിസ്റ്റർ നമ്പർ നൽകുമ്പോൾ, വെബ് സൈറ്റിൽ ആ കുട്ടിയുടെ മാർക്ക് ലിസ്റ്റ് കാണാം. ഓരോ കുട്ടിയുടെ കാര്യവും ഇതുപോലെ ആയിരിക്കും. SSLC പരീക്ഷയെഴുതിയ ലക്ഷക്കണക്കിന് കുട്ടികൾ ഓരോരുത്തർക്കും വേണ്ടി ഓരോ വെബ് പേജ് തയാറാക്കുകയെന്നത് പ്രായോഗികമല്ല. അങ്ങനെയെങ്കിൽ ഇത് സാധിക്കുന്നതെങ്ങനെ? ലക്ഷക്കണക്കിന് കുട്ടികളുടെ പരീക്ഷാഫലം വെബ് സെർവറിലുള്ള ഒരു ഡാറ്റാബേസിൽ സംഭരിക്കുന്നു. ഉപയോക്താവ് രജിസ്റ്റർ നമ്പർ നൽകുമ്പോൾ ആ വിദ്യാർഥിയുടെ പരീക്ഷാഫലം ഡാറ്റബേസിൽ നിന്നും എടുക്കാൻ സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ ഉപയോഗിക്കുന്നു. പിന്നീട് ഈ പരീക്ഷാഫലം ഉപയോഗിച്ച് സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ ഒരു വെബ് പേജ് നിർമിക്കുന്നു. തുടർന്ന് ഈ വെബ് പേജ് ക്ലയന്റ് ബ്രൗസറിലേയ്ക്ക് അയക്കുകയും ബ്രൗസർ അത് പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു. ഇത്തരത്തിൽ സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റ്, SSLC പരീക്ഷയെഴുതിയ ഓരോ കുട്ടിക്കുമായി ഓരോ വെബ് പേജ് നിർമിക്കുന്നു. വെബ് അടിസ്ഥാനമാക്കിയുള്ള ആപ്ലിക്കേഷനുകളുടെ ത്വരിത ഗതിയിലുള്ള വളർച്ച സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകളുടെ ഉപയോഗം കൂട്ടിയിട്ടുണ്ട്.



ചിത്രം 4.12 : സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകളുടെ പ്രവർത്തനം

ഉപയോക്താവ് നിർദേശിക്കുന്നതും, സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ ഉൾപ്പെട്ടിട്ടുള്ള വെബ് പേജ് സെർവറിൽ പ്രവർത്തിക്കുകയും, തൽഫലമായി ലഭിക്കുന്ന HTML കോഡ് ക്ലയന്റ് ബ്രൗസറിലേക്ക് അയയ്ക്കുകയും ചെയ്യുന്ന സാങ്കേതികവിദ്യയാണ് സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിങ്. സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ ഉള്ളടക്കം മാറിക്കൊണ്ടിരിക്കുന്ന വെബ് പേജുകൾ നിർമിക്കുന്നു. സ്ക്രിപ്റ്റുകൾ സെർവറിൽ പ്രവർത്തിക്കുന്നതിനാൽ ക്ലയന്റ് കമ്പ്യൂട്ടറിലുള്ള ബ്രൗസറിന്റെയോ ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിന്റെയോ പതിപ്പും തരവും ഈ പ്രവർത്തനത്തെ ബാധിക്കുന്നില്ല. സെർവറിലുള്ള വിഭവങ്ങൾ ഉപയോഗിച്ചുകൊണ്ടാണ് ഇത് പ്രവർത്തിക്കുന്നത്. സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റ് എഴുതുന്നതിന് ഉപയോഗിക്കുന്ന പ്രമുഖമായ സ്ക്രിപ്റ്റിങ് ഭാഷകളാണ് PERL, PHP, ASP, JSP തുടങ്ങിയവ.

സ്ക്രിപ്റ്റിങ് ഭാഷകളുടെ തരംതിരിക്കലിനെക്കുറിച്ചുള്ള ഒരു താരതമ്യപഠനം പട്ടിക 4.3-ൽ തന്നിട്ടുണ്ട്.

ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിങ്	സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിങ്
സ്ക്രിപ്റ്റ് ക്ലയന്റ് ബ്രൗസറിലേയ്ക്ക് പകർത്തുന്നു.	സ്ക്രിപ്റ്റ് വെബ് സെർവറിൽ നിലനിർത്തുന്നു.
സ്ക്രിപ്റ്റ് ക്ലയന്റ് ബ്രൗസറിൽ പ്രവർത്തിക്കുന്നു.	സ്ക്രിപ്റ്റ് വെബ് സെർവറിൽ പ്രവർത്തിക്കുമ്പോൾ ലഭിക്കുന്ന വെബ് പേജ് ക്ലയന്റ് ബ്രൗസറിലേക്കയയ്ക്കുന്നു.
ക്ലയന്റ് ഭാഗത്തെ ഡാറ്റയുടെ സാധുത പരിശോധനയ്ക്കാണ് പ്രധാനമായും സ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്നത്.	ഡാറ്റാബേസുമായി ബന്ധപ്പെടുന്നതിനും വെബ് സെർവറിൽ നിന്നും ഡാറ്റ ബ്രൗസറിലേയ്ക്ക് അയക്കുന്നതിനുമാണ് സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ ഉപയോഗിക്കുന്നത്.
ഉപയോക്താവിന് ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിന്റെ പ്രവർത്തനം തടയാൻ സാധിക്കും.	സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകളുടെ പ്രവർത്തനം ഉപയോക്താവിന് തടയുവാൻ സാധിക്കില്ല.
ബ്രൗസറിന്റെ തരവും പതിപ്പും ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിന്റെ പ്രവർത്തനത്തെ ബാധിക്കുന്നു.	വെബ് ബ്രൗസറിന്റെ സവിശേഷതകൾ സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിന്റെ പ്രവർത്തനത്തെ ബാധിക്കുന്നില്ല.

പട്ടിക 4.3: ക്ലയന്റ് ഭാഗത്തെയും സെർവർ ഭാഗത്തെയും സ്ക്രിപ്റ്റുകൾ തമ്മിലുള്ള താരതമ്യം

ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ ഡാറ്റയുടെ സാധുത ഉപയോക്താവിന്റെ ബ്രൗസറിൽ വച്ചുതന്നെ പരിശോധിക്കുന്നതിനാണ് ഉപയോഗിക്കുന്നത്. മാത്രമല്ല അത് സെർവറിന്റെ ജോലി ഭാരവും ശൃംഖലയിലെ ഗതാഗതക്കുരുക്കും കുറയ്ക്കുകയും ചെയ്യുന്നു. ആയതിനാൽ, സാധുത പരിശോധിക്കുന്നതിനായുള്ള സ്ക്രിപ്റ്റുകൾ പൂർണമായും ക്ലയന്റ് ഭാഗത്തേക്ക് മാറ്റിയിട്ടുണ്ട്. അതുകൊണ്ട്, സെർവറിലേക്ക് അയയ്ക്കുന്ന ഡാറ്റ പൂർണമായും തെറ്റുകളില്ലാത്തതും നേരിട്ട് ഡാറ്റാബേസിൽ സംഭരിക്കാവുന്നതുമാണ്. എന്നാൽ ക്ലയന്റ് ബ്രൗസർ, സ്ക്രിപ്റ്റുകൾ മനസ്സിലാക്കാതിരിക്കുകയോ, ഉപയോക്താവ് സുരക്ഷിത കാരണങ്ങളാൽ സ്ക്രിപ്റ്റിന്റെ ബ്രൗസറിലുള്ള പ്രവർത്തനം തടയുകയോ ചെയ്താൽ, സാധുത പരിശോധന നടത്താതെ ഡാറ്റ സെർവറിലേക്ക് അയയ്ക്കപ്പെടും. ഇക്കാരണത്താൽ അസാധു ആയ ഡാറ്റ ഡാറ്റബേസിൽ സംഭരിക്കപ്പെടും. ഡാറ്റയുടെ സാധുത സെർവർ ഭാഗത്ത് കൂടി ഉറപ്പാക്കാനായാൽ ഡാറ്റാബേസിൽ സംഭരിക്കുന്ന ഡാറ്റയുടെ സാധുത സംരക്ഷിക്കപ്പെടും.

4.5.2 സ്ക്രിപ്റ്റിങ് ഭാഷകൾ (Scripting languages)

രണ്ടു തരം സ്ക്രിപ്റ്റിങ് ഭാഷകൾ നാം പരിചയപ്പെട്ടു കഴിഞ്ഞു. പ്രചാരത്തിലുള്ള ചില സ്ക്രിപ്റ്റിങ് ഭാഷകളുടെ സവിശേഷതകൾ നമുക്ക് ചർച്ച ചെയ്യാം.

A. ജാവ സ്ക്രിപ്റ്റ് (Java Script)

ഉപയോക്താവുമായി സംവദിക്കുന്ന വെബ് പേജുകളുടെ നിർമാണത്തിന് ഉപയോഗിക്കുന്ന സ്ക്രിപ്റ്റിങ് ഭാഷയാണ് ജാവാസ്ക്രിപ്റ്റ്. നെറ്റ്സ്കേപ്പ് കമ്മ്യൂണിക്കേഷൻ കോർപ്പറേഷനിൽ പ്രവർത്തിച്ചിരുന്ന കാലത്ത് ബ്രണ്ടൻ ഇച്ച് (Brenden Eich - ചിത്രം 4.13) നിർമിച്ചതാണ് ജാവ സ്ക്രിപ്റ്റ്. ക്ലയന്റ് ഭാഗത്തെ വ്യാഖ്യാനിക്കാവുന്ന ഭാഷയാണ് ജാവ സ്ക്രിപ്റ്റ് . ജാവ സ്ക്രിപ്റ്റ് കോഡ് എഴുതുവാൻ ജിനി IDE യോ നോട്ട്പാഡോ പോലുള്ള ടെക്സ്റ്റ് എഡിറ്ററുകൾ ഉപയോഗിക്കാവുന്നതാണ്. ക്ലയന്റ് ഭാഗത്തെ ഒരു പ്രമുഖ സ്ക്രിപ്റ്റിങ് ടൂൾ ആയ ഇത് മിക്കവാറും ബ്രൗസറുകളിൽ പ്രവർത്തിക്കുന്നു. ജാവസ്ക്രിപ്റ്റ് HTML

ചിത്രം 4.13 ബ്രണ്ടൻ ഇച്ച്
(1961   )

കോഡിൽ ഉൾപ്പെടുത്തിയോ പ്രത്യേക ഫയലിലെഴുതി HTML ഫയലിനോട് ലിങ്ക് ചെയ്തോ ഉപയോഗിക്കാം. പ്രത്യേക ഫയലിൽ എഴുതിയ ജാവ സ്ക്രിപ്റ്റ് .js എക്സ്റ്റൻഷനോടുകൂടി സൂക്ഷിക്കണം. ജാവാസ്ക്രിപ്റ്റ് ഫയലുകളെ തിരിച്ചറിയാൻ ഇത് സഹായിക്കുന്നു. ക്ലയന്റ് ഭാഗത്തെ ഫോമുകളിലെ ഡാറ്റയുടെ സാധുത ഉറപ്പിക്കാൻ ഉപയോഗിക്കുന്ന ടൂൾ എന്ന നിലയിൽ ജാവാ സ്ക്രിപ്റ്റ് പ്രശസ്തമാണ്. ലളിതമായ കണക്ക് കൂട്ടലുകളും ചലിക്കുന്ന കാർട്ടൂൺ ചിത്രങ്ങളും വെബ് പേജുകളിൽ ഉൾപ്പെടുത്തുന്നതിന് ഇത് ഉപയോഗിക്കുന്നു. അധ്യായം 6-ൽ ജാവ സ്ക്രിപ്റ്റിനെക്കുറിച്ച് വിശദമായി ചർച്ച ചെയ്യാം.

ജാവ സ്ക്രിപ്റ്റിന്റെ പ്രചാരം ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ തയാറാക്കുന്ന പ്രവർത്തനത്തിന്റെ പുരോഗതിക്ക് കാരണമായി. ഒന്നാംവർഷ ഹയർസെക്കന്ററി പ്രവേശനത്തിന് അപേക്ഷ സമർപ്പിക്കുമ്പോൾ, SSLC രജിസ്റ്റർ നമ്പർ നൽകിയാലുടൻ, നിങ്ങളുടെ പേരും, ജനനത്തീയതിയും, മറ്റ് വിവരങ്ങളും താഴെയുള്ള ടെക്സ്റ്റ് ബോക്സുകളിൽ പ്രത്യക്ഷപ്പെടുന്നത് ശ്രദ്ധിച്ചിരിക്കുമല്ലോ. ബ്രൗസറിൽ പുനർപ്രവേശിക്കാതെയാണ് ഇത് ചെയ്യുന്നത്. വെബ് പേജ് പുനർനിർമിക്കാതെ (റിഫ്രഷ് ചെയ്യാതെ) സെർവറിൽ നിന്ന് ഡാറ്റ എടുത്ത് ടെക്സ്റ്റ് ബോക്സുകളിൽ നിറയ്ക്കുന്നു. അജാക്സ് (Ajax) എന്ന സാങ്കേതികവിദ്യയാണ് ഇവിടെ ഉപയോഗിക്കുന്നത്. ബ്രൗസറുകളുടെ വിനിമയശേഷി വർധിപ്പിക്കുന്നതിനാണ് അജാക്സ് ഉപയോഗിക്കുന്നത്. അജാക്സ് എന്നാൽ അസിങ്ക്രോണസ് ജാവ സ്ക്രിപ്റ്റ്, എക്സ്റ്റൻസിബിൾ മാർക്കപ്പ് ലാംഗ്വേജ് (XML) എന്നിവ ചേർന്നതാണ്. ഉപയോക്താക്കൾക്ക് പുതിയ ടാഗുകൾ നിർമിക്കാൻ ഉതകുന്ന മാർക്കപ്പ് ഭാഷയാണ് XML. അജാക്സ് വെബ്സൈറ്റിൽ പ്രവർത്തനക്ഷമമായ ശേഷം പരിവർത്തനാത്മക ഉള്ളടക്കം പ്രദർശിപ്പിക്കാനായി വെബ് പേജ് പൂർണമായും വീണ്ടും കൊണ്ടുവരേണ്ടതില്ല. ക്ലയന്റും സെർവറും തമ്മിൽ പിന്നണിയിൽ ഡാറ്റ കൈമാറ്റം ചെയ്തുകൊണ്ട് വെബ് പേജിനെ നവീകരിക്കാൻ അജാക്സ് സഹായിക്കുന്നു. ബ്രൗസറിന് ജാവ സ്ക്രിപ്റ്റ് കൈകാര്യം ചെയ്യാനാകാതെ വരുകയോ, ബ്രൗസറിലുള്ള ജാവ സ്ക്രിപ്റ്റിന്റെ പ്രവർത്തനം ഉപയോക്താവ് തടയുകയോ ചെയ്താൽ അജാക്സ് ആപ്ലിക്കേഷൻ പ്രവർത്തിക്കുകയില്ല.

B. വിബി സ്ക്രിപ്റ്റ് (VB Script)

വിഷ്വൽ ബേസിക് എന്ന പ്രോഗ്രാമിങ് ഭാഷ അടിസ്ഥാനമാക്കി മൈക്രോസോഫ്റ്റ് കോർപ്പറേഷൻ വികസിപ്പിച്ച സ്ക്രിപ്റ്റിങ് ഭാഷയാണ് വിബി സ്ക്രിപ്റ്റ്. മൈക്രോസോഫ്റ്റ് ഇന്റർനെറ്റ് എക്സ്പ്ലോററിനായി ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിങ് ഭാഷയായും മൈക്രോസോഫ്റ്റ് ഇന്റർനെറ്റ് ഇൻഫർമേഷൻ സർവീസസിനായി സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിങ് ഭാഷയായും വി.ബി. സ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്നു. നിർഭാഗ്യവശാൽ, മറ്റ് ബ്രൗസറുകൾക്ക് വിബി സ്ക്രിപ്റ്റ് കോഡ് കൃത്യമായി മനസിലാക്കാനോ പ്രദർശിപ്പിക്കാനോ സാധിക്കില്ല. അതുകൊണ്ടു തന്നെ, ക്ലയന്റ് ഭാഗത്തെ പ്രചാരം കുറഞ്ഞ ഒരു സ്ക്രിപ്റ്റിങ് ടൂൾ ആണ് ഇത്. സെർവർ ഭാഗത്ത് വിൻഡോസ് ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിന് പ്രചാരമേറെയുള്ളതിനാൽ സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റെഴുതുന്നതിന് വിബി സ്ക്രിപ്റ്റ് കൂടുതലായി ഉപയോഗിക്കുന്നു.

.NET ഫ്രെയിംവർക്കിന്റെ - ഉപയോഗയോഗ്യമായ പ്രോഗ്രാം കോഡിന്റെ ഒരു ശേഖരം, ആവിർഭാവത്തോടെ വി.ബി. സ്ക്രിപ്റ്റ് .NET ഫ്രെയിംവർക്കിലെ ASP.NET ന്റെ ഭാഗമായി മാറി.

C. PHP

ചിത്രം 4.14 റാസ്മസ് ലെർഡോർഫ് (1968)

PHP എന്നത് ഹൈപ്പർ ടെക്സ്റ്റ് പ്രീപ്രൊസസറിനെ സൂചിപ്പിക്കുന്നു. വെബ് വികസനത്തിന് അനുയോജ്യമായതും, HTML- ൽ ഉൾക്കൊള്ളിക്കാവുന്നതുമായ ഇത് പൊതു ആവശ്യങ്ങൾക്കായുള്ള ഒരു ഓപ്പൺ സോഴ്സ് സ്ക്രിപ്റ്റിങ് ഭാഷയാണ്. സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റ് എഴുതുന്നതിനുള്ള ടൂൾ ആയ ഇതിന്റെ കോഡ് ജാവ, C, PERL എന്നിവയ്ക്ക് സമാനമാണ്. വളരെ എളുപ്പത്തിൽ പരിവർത്തനാത്മക വെബ് പേജുകൾ നിർമിക്കാൻ ഇത് ഉപയോഗിക്കാം. 1994-ൽ റാസ്മസ് ലെർഡോർഫ് (Rasmus Lerdorf-ചിത്രം 4.14) ആണ് PHP വികസിപ്പിച്ചത്, എങ്കിലും PHP സമൂഹമാണ് തുടർന്ന് അത് പരിപോഷിപ്പിച്ച് കൊണ്ടിരിക്കുന്നത്. PHP കോഡുള്ള വെബ് പേജുകളുടെ ഫയൽനാമത്തിന്റെ എക്സ്റ്റൻഷൻ .php എന്നായിരിക്കും. ഉപയോക്താവ് PHP വെബ് പേജിനായി അപേക്ഷിക്കുമ്പോൾ, വെബ് സെർവർ HTML കോഡിൽ ഉൾപ്പെടുത്തിയിട്ടുള്ള PHP കോഡ് വ്യാഖ്യാനിക്കുകയും പ്രവർത്തിപ്പിക്കുകയും ചെയ്യുന്നു. PHP കോഡ് വെബ് സെർവറിൽ പ്രവർത്തിക്കണമെങ്കിൽ ഒരു PHP ഇന്റർപ്രെട്ടർ വെബ് സെർവറിൽ ഉണ്ടായിരിക്കണം. വെബ് സെർവറിൽ PHP കോഡ് പ്രവർത്തിക്കുമ്പോൾ നിർമിക്കപ്പെടുന്ന HTML പേജ് ക്ലയന്റ്ബ്രൗസറിലേക്ക് അയയ്ക്കുന്നു. ഡാറ്റാബേസ് പ്രോഗ്രാമിങ്ങിന് സഹായിക്കുന്നു എന്നതാണ് PHP യുടെ പ്രധാനപ്പെട്ട സവിശേഷത. PHP യോടൊപ്പം സാധാരണ ഉപയോഗിക്കുന്ന ഡാറ്റാബേസാണ് MYSQL. എല്ലാ ഓപ്പറേറ്റിങ് സിസ്റ്റങ്ങൾക്കും PHP ഇന്റർപ്രെട്ടർ ലഭ്യമാണ്. സൗജന്യമായി ഇന്റർനെറ്റിൽ നിന്നും പകർത്താവുന്ന LAMP (Linux, Apache, MYSQL, PHP) സെർവർ സോഫ്റ്റ്വെയറാണ് ലിനക്സ് പ്ലാറ്റ്ഫോമുകളിൽ ഉപയോഗിക്കുന്നത്. ഇവിടെ, ലിനക്സ് എന്നത് സെർവർ ഓപ്പറേറ്റിങ് സിസ്റ്റവും, അപ്പാഷെ എന്നത് വെബ് സെർവറും, MYSQL എന്നത് ഡാറ്റാബേസും, PHP എന്നത് സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിംഗ് ഭാഷയും ആയി പ്രവർത്തിക്കുന്നു. ഇന്റർനെറ്റിൽ നിന്നും സൗജന്യമായി പകർത്താവുന്ന WAMP (WINDOWS, Apache, MYSQL, PHP) സെർവർ സോഫ്റ്റ്വെയറാണ് വിൻഡോസ് ഓപ്പറേറ്റിങ് സിസ്റ്റം ഉപയോഗിക്കുന്നത്. 10-ാമത്തെ അധ്യായത്തിൽ PHP-യെക്കുറിച്ച് കൂടുതലായി ചർച്ച ചെയ്യാം.

D. ആക്റ്റീവ് സെർവർ പേജുകൾ (Active Server Pages)

പരസ്പരവിനിമയം നടത്തുന്ന വെബ് ആപ്ലിക്കേഷനുകൾ നിർമിക്കുന്നതിനും പ്രവൃത്തിപ്പിക്കുന്നതിനും വേണ്ടി സെർവർ ഭാഗത്ത് ഉപയോഗിക്കുന്ന സ്ക്രിപ്റ്റിങ് സംവിധാനമാണ് മൈക്രോസോഫ്റ്റിന്റെ ആക്റ്റീവ് സെർവർ പേജുകൾ (ASP). ASP -യിൽ HTML-ഉം, സ്ക്രിപ്റ്റിങ് ഭാഷയിലെ കോഡുണ്ടായിരിക്കും. സ്ക്രിപ്റ്റിങ് ഭാഷ വിബി സ്ക്രിപ്റ്റോ ജാവ സ്ക്രിപ്റ്റോ ആയിരിക്കും. ASPഫയലുകളുടെ എക്സ്റ്റൻഷൻ .asp എന്നായിരിക്കും. മൈക്രോ സോഫ്റ്റിന്റെ സെർവർ സോഫ്റ്റ്വെയറായ ഇന്റർനെറ്റ് ഇൻഫർമേഷൻ

സർവീസസിൽ (IIS-ൽ) തനതായി നിർമിക്കപ്പെട്ടിട്ടുള്ള സവിശേഷത ഉപയോഗിച്ച് ഈ ഫയലുകൾ കംപൈൽ ചെയ്യുന്നു. ASP ഫയലുകൾ വിൻഡോസ് ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിൽ മാത്രം പ്രവർത്തിക്കുന്നവയാണ്. സെർവറിൽ എക്സിക്യൂട്ട് ചെയ്തശേഷം ലഭിക്കുന്ന HTML വെബ് പേജ് ക്ലയന്റ് ഭാഗത്തെ ബ്രൗസറിലേക്കയക്കുന്നു. വിവിധതരത്തിലുള്ള ഡാറ്റബേസുകളെ പിന്തുണയ്ക്കുന്ന, വളരെ പ്രമുഖമായ ഒരു സ്ക്രിപ്റ്റിങ് ഭാഷയാണിത്. ASP.NET -ന്റെ വരവോടെ, ASP -യുടെ തുടർപതിപ്പുകൾ പുറത്തിറക്കുന്നത് മൈക്രോസോഫ്റ്റ് നിർത്തലാക്കി. കോഡുകളുടെ ചുരുക്കെഴുത്തും, വ്യത്യസ്തങ്ങളായ ബട്ടണുകൾ, ടെക്സ്റ്റ് ബോക്സുകൾ തുടങ്ങിയവയുടെ ലഭ്യതയും വെബ് ആപ്ലിക്കേഷനുകൾ നിർമിക്കാനായി ASP.NET-ലുള്ള സവിശേഷതകളാണ്.

E. ജാവ സെർവർ പേജുകൾ (Java Server Pages)

പരിവർത്തനാത്മക വെബ് പേജ് നിർമിക്കുന്നതിനുള്ള എളുപ്പവും വേഗമേറിയതുമായ സാങ്കേതിക വിദ്യ JSP പ്രദാനം ചെയ്യുന്നു. 1999-ൽ സൺ മൈക്രോസിസ്റ്റംസ് നിർമിച്ച സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിങ് ഭാഷയാണ് ഇത്. PHP-ക്ക് സമാനമാണ് JSP, എങ്കിലും പ്രോഗ്രാമിങ് ഭാഷയായി ജാവയാണ് ഉപയോഗിക്കുന്നത്. JSP ഫയലുകളുടെ ഫയൽ എക്സ്റ്റൻഷൻ .jsp എന്നാണ്. JSP പ്രവർത്തിപ്പിക്കുന്നതിന് അപ്പാഷെ ടോം ക്യാറ്റ് വെബ് സെർവർ ആവശ്യമാണ്. JSP കോഡിലുള്ള HTML കോഡും ജാവകോഡും വെബ് സെർവറിൽ പ്രവർത്തിക്കുന്നതിന്റെ (എക്സിക്യൂട്ട് ചെയ്ത്) ഫലമായി ലഭിക്കുന്ന HTML കോഡിനെ ബ്രൗസറിലേയ്ക്ക് അയക്കുന്നു. വൻതോതിലുള്ള വെബ് അധിഷ്ഠിത സോഫ്റ്റ്വെയറുകൾ നിർമിക്കുന്നതിനും പ്രവർത്തിപ്പിക്കുന്നതിനും ഉപയോഗിക്കുന്ന ജാവ 2 പ്ലാറ്റ്ഫോം എന്റർപ്രൈസ് എഡിഷന്റെ (J2EE) അവിഭാജ്യ ഘടകമാണ് JSP.

4.6 കാസ്കേഡിങ് സ്റ്റൈൽ ഷീറ്റ് (Cascading Style Sheet)

HTML ഭാഷയിൽ എഴുതിയിട്ടുള്ള ഒരു ലിഖിതത്തിന്റെ ഘടന വിവരിക്കാൻ ഉപയോഗിക്കുന്ന സ്റ്റൈൽ ഷീറ്റ് ഭാഷയാണ് കാസ്കേഡിങ് സ്റ്റൈൽ ഷീറ്റ് (CSS). ഒരു വെബ് പേജിലുള്ള ചിഹ്നങ്ങളുടെ നിറം, അക്ഷരങ്ങളുടെ രീതി, ഖണ്ഡികകൾക്ക് ഇടയിലുള്ള സ്ഥലം, നിരകളുടെ വലുപ്പം പ്രദർശിപ്പിക്കുന്ന രീതി, അരികുകളും അവയുടെ നിറങ്ങളും, പശ്ചാത്തല ചിത്രങ്ങൾ/നിറങ്ങൾ എങ്ങനെയാണ് ഉപയോഗിക്കുന്നത് തുടങ്ങിയ വിവിധതരത്തിലുള്ള പ്രതീതികൾ നിയന്ത്രിക്കുവാൻ CSS ഉപയോഗിക്കുന്നു. HTML ഉള്ളടക്കത്തെ അതിന്റെ രൂപഭംഗിയിൽ നിന്നും വേർതിരിച്ച് നിർത്താൻ CSS നമ്മെ അനുവദിക്കുന്നു. മൂന്ന് വ്യത്യസ്ത രീതികളിൽ CSS നമുക്ക് നടപ്പിലാക്കാം - ഇൻലൈൻ (ഒപ്പംതന്നെ), എംബഡഡ് (ഉൾക്കൊള്ളിച്ച്), ലിങ്ക്ഡ് (ബന്ധിപ്പിച്ച്) എന്നിവയാണവ.

- ഒപ്പമെഴുതുന്ന രീതിയിൽ, വെബ് പേജിന്റെ ബോഡി ഭാഗത്ത്, സ്റ്റൈൽ ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് കൊണ്ട്, CSS ശൈലി ഓരോ ടാഗിനും പ്രത്യേകമായി പ്രയോഗിക്കാവുന്നതാണ്.
- ഉൾക്കൊള്ളിക്കുന്ന CSS കോഡുകൾ, വെബ്പേജിന്റെ ഹെഡ് ഭാഗത്ത് സ്ഥാപിക്കുന്നു.
- CSS കോഡ് മാത്രമുൾക്കൊള്ളുന്ന ഒരു ബാഹ്യ ഫയലിനെ വെബ് പേജുമായി ബന്ധിപ്പിച്ച് കൊണ്ടാണ് CSS നിർവഹണം ലിങ്ക്ഡ് രീതിയിൽ നടപ്പിലാക്കുന്നത്.

എല്ലാ പേജുകൾക്കും ഒരേ കോഡ് പുനരുപയോഗിക്കാമെന്നതാണ് CSS -ന്റെ നേട്ടം. ഒരു വെബ്സൈറ്റിനുവേണ്ടിയുള്ള CSS സ്റ്റൈൽ ബാഹ്യഫയലായി ലിങ്ക്ചെയ്താണ് നടപ്പിലാക്കുന്നതെങ്കിൽ, ഒരു സ്റ്റൈൽ വ്യത്യാസപ്പെടുത്തുന്നതിലൂടെ, വെബ് സൈറ്റിലെ എല്ലാ പേജുകളിലെയും ടാഗുകളുടെ അവതരണ രീതി വ്യത്യാസപ്പെടുത്താനാകും. CSS സ്റ്റൈൽ പൊതുവായി ഒരു സ്ഥലത്ത് എഴുതുന്നതിനാൽ HTML-ഉം, CSS വേർതിരിച്ച് നിർത്താൻ സാധിക്കുകയും അങ്ങനെ പരിപാലനം എളുപ്പമാവുകയും ചെയ്യുന്നു. കൂടാതെ വെബ് പേജുകളിലുള്ള ടാഗുകളുടെ സ്റ്റൈൽ വിവരണവുമായി നല്ല രീതിയിൽ ബന്ധപ്പെടുത്തി ക്രമീകരിച്ചിരിക്കുന്നതുകൊണ്ട് എളുപ്പത്തിൽ അത് മനസ്സിലാക്കാനാവുന്നു. ഇത് വെബ് പേജുകളുടെ ബൈറ്റുകളിലുള്ള വലുപ്പം കുറയ്ക്കുകയും അങ്ങനെ വേഗത്തിൽ വെബ് പേജുകൾ ബ്രൗസറിൽ ദൃശ്യമാക്കുകയും ചെയ്യുന്നു.

വ്യത്യസ്ത സ്ക്രീൻ വലുപ്പമുള്ള ഉപകരണങ്ങളായ ഡെസ്ക്ടോപ്പ് മോണിട്ടറുകൾ, ടാബ് കമ്പ്യൂട്ടറുകൾ, മൊബൈൽ ഫോണുകൾ തുടങ്ങിയവയിൽ വെബ് പേജുകൾ പ്രദർശിപ്പിക്കാൻ അനുയോജ്യമാക്കുന്നത് CSS ആണ്. ചലനാത്മചിത്രങ്ങൾ, ആനിമേഷനുകൾ, പരസ്പരസംവേദനം എന്നിവ കൊണ്ടുവരാനാകുന്ന HTML-ന്റെ അടുത്ത പതിപ്പായ HTML5-ൽ ഉപയോഗിച്ചിരിക്കുന്നത് ജാവ സ്ക്രിപ്റ്റിനോടൊപ്പമുള്ള CSS ആണ്.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. കോഡ് വ്യത്യാസപ്പെടുത്തുന്നത് വരെ ഉള്ളടക്കം മാറാതെ നിൽക്കുന്ന വെബ് പേജുകളെ ________ എന്ന് വിളിക്കുന്നു.
2. ഡൈനാമിക് വെബ് പേജുകൾ നിർമിക്കുന്നതിന് ഉപയോഗിക്കുന്ന രണ്ട് സാങ്കേതിക വിദ്യകളുടെ പേരെഴുതുക.
3. സ്ക്രിപ്റ്റ് ഉൾക്കൊള്ളിക്കാൻ ഉപയോഗിക്കുന്ന ടാഗാണ് ________.
4. ക്ലയന്റ് ഭാഗത്ത് സ്ക്രിപ്റ്റ് എഴുതുന്നതിന്റെ ഏതെങ്കിലും ഒരു ഉപയോഗമെഴുതുക.
5. ഒരു ജാവ സ്ക്രിപ്റ്റ് ഫയലിന്റെ എക്സ്റ്റൻഷൻ ________ ആണ്.
6. അജാക്സ് ഉപയോഗിക്കുന്നത് കൊണ്ടുള്ള നേട്ടങ്ങൾ എന്തൊക്കെയാണ്?
7. താഴെതന്നിട്ടുള്ള സ്ക്രിപ്റ്റിങ് ഭാഷകളെ ക്ലയന്റ് ഭാഗത്തേതെന്നും സെർവർ ഭാഗത്തേതെന്നുമായി തരംതിരിക്കുക.

   ജാവ സ്ക്രിപ്റ്റ്, PHP, ASP, വി.ബി. സ്ക്രിപ്റ്റ്
8. .asp ഫയലുകൾ ________ എന്ന വെബ് സെർവർ സോഫ്റ്റ്വെയർ ഉപയോഗിച്ചാണ് കംപൈൽ ചെയ്യുന്നത്.
9. CSS പ്രാവർത്തികമാക്കുന്നതിനുള്ള വ്യത്യസ്ത മാർഗങ്ങൾ എഴുതുക.

4.7 HTML ഡോക്യുമെന്റുകളുടെ അടിസ്ഥാന ആശയം (Basic concepts of HTML documents)

വെബ് പേജുകളുടെ നിർമാണത്തിന് വ്യാപകമായി ഉപയോഗിക്കുന്ന ഭാഷയാണ് HTML. ഓരോ വെബ് പേജും യഥാർഥത്തിൽ ഒരു HTML ഫയലാണ്. ഹൈപ്പർ ടെക്സ്റ്റ് ഡോക്യുമെന്റുകൾ നിർമിക്കുന്നതിനുള്ള ഒരു കൂട്ടം നിർദേശങ്ങൾ അടങ്ങിയ ലളിതമായ വാചകങ്ങളാണ് (ടെക്സ്റ്റുകളാണ്) ഓരോ HTML ഫയലും. ഈ നിർദേശങ്ങൾ HTML ടാഗുകൾ എന്ന് അറിയപ്പെടുന്നു. ഈ ടാഗുകൾ ഉപയോഗിക്കുമ്പോൾ നിർദേശങ്ങൾ കൂടുതൽ വ്യക്തതയുള്ളതാക്കാനായി ചില കീ-വേഡുകൾ അവയോടൊപ്പം ചേർക്കുന്നു. ഈ വാക്കുകൾ ആട്രിബ്യൂട്ടുകൾ എന്നറിയപ്പെടുന്നു. ഒരു വെബ് പേജിലെ ഉള്ളടക്കം ബ്രൗസറിൽ എങ്ങനെ പ്രദർശിപ്പിക്കണമെന്ന് തീരുമാനിക്കുന്നത് HTML ഡോക്യുമെന്റിലുള്ള ടാഗുകളും ആട്രിബ്യൂട്ടുകളും ഒരുമിച്ച് പ്രവർത്തിച്ചുകൊണ്ടാണ്. HTML ടാഗുകളെയും അവയുടെ ആട്രിബ്യൂട്ടുകളെയും കുറിച്ചുള്ള പഠനമാണ് യഥാർഥത്തിൽ HTML പഠനം. ടാഗുകളുടെയും ആട്രിബ്യൂട്ടുകളുടെയും വിശദാംശങ്ങളിലേക്ക് കടക്കുന്നതിന് മുമ്പ് ഒരു HTML ഡോക്യുമെന്റിന്റെ അടിസ്ഥാന ഘടന നമുക്ക് പരിശോധിക്കാം.

4.7.1 ഒരു HTML ഡോക്യുമെന്റിന്റെ അടിസ്ഥാനഘടന (Basic structure of an HTML document)

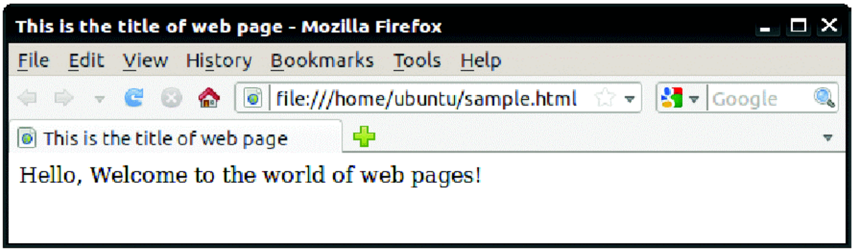
ഉദാഹരണം 4.1-ൽ ഒരു HTML ഡോക്യുമെന്റിന്റെ അടിസ്ഥാനഘടന കാണിച്ചിരിക്കുന്നു.

ഉദാഹരണം 4.1: HTML ഡോക്യുമെന്റിന്റെ മാതൃക

```
<HTML>
   <HEAD>
     <TITLE> This is the title of web page </TITLE>
   </HEAD>
   <BODY>
     Hello, Welcome to the world of web pages!
   </BODY>
</HTML>
```

ഒരു ജോഡി ആംഗിൾ ബ്രാക്കറ്റുകൾക്കുള്ളിലായി (<, >) വലിയ അക്ഷരങ്ങളിലുള്ള ചില വാക്കുകൾ നിങ്ങൾക്ക് കാണാം. ഇവ HTML ടാഗുകളാണ്. HTML ടാഗുകൾ വലിയ അക്ഷരങ്ങളിലെഴുതണമെന്ന് നിർബന്ധമില്ല. HTML-ൽ വലിയ അക്ഷരവും ചെറിയ അക്ഷരവും തമ്മിൽ വ്യത്യാസമില്ല (കേയ്സ് സെൻസിറ്റീവ് അല്ല). നമുക്ക് വലിയ അക്ഷരത്തിലോ, ചെറിയ അക്ഷരത്തിലോ അവ കൂട്ടികലർത്തിയോ ഉപയോഗിക്കാവുന്നതാണ്. മറ്റ് വാക്കുകളിൽനിന്നും അല്ലെങ്കിൽ ചിഹ്നങ്ങളിൽനിന്നും വേർതിരിച്ചറിയാനായി, HTML ടാഗുകളും അവയുടെ ആട്രിബ്യൂട്ടുകളുടെ ആദ്യക്ഷരവും വലിയ അക്ഷരങ്ങളായും മറ്റുള്ളവ ചെറിയ അക്ഷരങ്ങളായുമാണ് സൂചിപ്പിക്കുന്നത്.

ഉദാഹരണം 4.1-ൽ കാണിച്ചിരിക്കുന്നത് പോലെ, എല്ലാ HTML പേജുകളും തുടങ്ങുന്നത് **<HTML>** ടാഗിലും അവസാനിക്കുന്നത് **</HTML>** ടാഗിലുമായിരിക്കും. ഒരു HTML ലിഖിതത്തെ ഹെഡ് ഭാഗമെന്നും ബോഡി ഭാഗമെന്നും രണ്ടായി തിരിച്ചിരിക്കുന്നു. ഹെഡ് വിശദീകരിക്കുന്നതിന് **<HEAD>** ടാഗുപയോഗിക്കുന്നു. വെബ് പേജിന്റെ ശീർഷകം ഉൾപ്പെടെ ലിഖിതത്തെക്കുറിച്ചുള്ള വിവരങ്ങൾ ഹെഡ് ഭാഗത്തുണ്ടാകും. ബ്രൗസർ വിൻഡോയിലെ ടൈറ്റിൽ ബാറിൽ കാണിക്കുന്ന വെബ് പേജിന്റെ ശീർഷകം നിർണയിക്കുന്നത് **<TITLE>** ടാഗ് ഉപയോഗിച്ചുകൊണ്ടാണ്. ബോഡി ഭാഗം നിർവചിക്കുന്നതിന് **<BODY>** ടാഗ് ഉപയോഗിക്കുന്നു. ബോഡി ഭാഗത്തുള്ളത് വെബ് പേജിന്റെ ഉള്ളടക്കമായിരിക്കും. മുകളിൽ തന്നിട്ടുള്ള HTML ലിഖിതം ബ്രൗസർ വിൻഡോയിൽ കണ്ടാൽ ചിത്രം 4.15-ൽ കാണുന്നത് പോലുണ്ടാകും.



ചിത്രം 4.15 : വെബ് ബ്രൗസർ ഉപയോഗിച്ച് തുറന്ന ഒരു വെബ് പേജ്

വിവിധ HTML പതിപ്പുകൾ: 1991 ന്റെ അവസാനം ടിം ബെർണേഴ്സ് ലീ (Tim Berners Lee) ആണ് HTML നിർമിച്ചത്. എന്നാൽ, നിലവാരമുള്ള ആദ്യത്തെ HTML പതിപ്പായ HTML 2.0 പ്രസിദ്ധീകരിച്ചത് 1995 - ലാണ്. പ്രമുഖ HTML പതിപ്പായ HTML 4.01 പ്രസിദ്ധീകരിച്ചത് 1999-കളുടെ അവസാനമാണ്. 2012 - ലാണ് ഏറ്റവും പുതിയ പതിപ്പായ HTML 5 പുറത്തിറക്കിയത്. ഇത് കൂടുതൽ കൂടുതൽ ദൃശ്യ - ശ്രാവ്യ സൗകര്യങ്ങൾ ഉൾപ്പെടുത്തിക്കൊണ്ട് ഇപ്പോഴും പരിഷ്കരിച്ചുകൊണ്ടിരിക്കുന്നു.

4.7.2 HTML ഡോക്യുമെന്റിലെ ടാഗുകൾ (Tags in HTML document)

മുമ്പ് സൂചിപ്പിച്ചതു പോലെ, വെബ് പേജുകളുടെ ഉള്ളടക്കം ഏതു രീതിയിൽ ക്രമീകരിക്കണമെന്നും പ്രദർശിപ്പിക്കണമെന്നും ബ്രൗസറിനോട് പറയുന്ന നിർദേശങ്ങളാണ് HTML ടാഗുകൾ. എല്ലാ ടാഗുകൾക്കും <, > എന്നീ ആങ്കിൾ ബ്രായ്ക്കറ്റുകളിലടക്കം ചെയ്ത ടാഗ് നാമമുണ്ടായിരിക്കും.

4.7.3 കണ്ടെയ്നർ ടാഗുകളും എംപ്റ്റി ടാഗുകളും (Container tags and empty tags)

ഭൂരിഭാഗം ടാഗുകളും ജോഡികളായാണ് ഉപയോഗിക്കുന്നത്. പ്രാരംഭ (ഓപ്പണിംഗ്) ടാഗും അവസാനിപ്പിക്കുന്ന (ക്ലോസിങ്) ടാഗും ഉണ്ടാകും. ഉദാഹരണമായി, <HTML> എന്നത് തുടങ്ങുന്ന ടാഗും </HTML> എന്നത് അവസാനിപ്പിക്കുന്ന ടാഗുമാണ്. അവസാനിപ്പിക്കുന്ന ടാഗിനും, തുടങ്ങുന്ന ടാഗിനും ഒരേ പേരാണുള്ളത്. എന്നാൽ ആദ്യത്തെ ആങ്കിൾ ബ്രാക്കറ്റിന് ശേഷം ഒരു ഫോർവേഡ് സ്ലാഷ് (/) ചിഹ്നം അവസാനിപ്പിക്കുന്ന ടാഗിന്

അധികമായുണ്ടായിരിക്കും. തുടങ്ങുന്ന ടാഗും അവസാനിപ്പിക്കുന്ന ടാഗും ആവശ്യമുള്ള ടാഗുകളാണ് കണ്ടെയ്നർ ടാഗുകൾ. ഒരു സെക്ഷനിലേക്ക് ഉപയോഗിക്കാവുന്നവയാണ് കണ്ടെയ്നർ ടാഗുകൾ. സെക്ഷന്റെ തുടക്കത്തിൽ ഓപ്പണിംഗ് ടാഗും അവസാനം ക്ലോസിങ് ടാഗും നൽകുന്നു. ഉദാഹരണമായി, ഒരു HTML ലിഖിതത്തിന്റെ തുടക്കവും ഒടുക്കവും ക്രമപ്പെടുത്തുന്ന ടാഗ് ജോഡികളാണ് <HTML>, </HTML> എന്നിവ.

ചില ടാഗുകൾ ഈ രീതിയിലല്ല ഉപയോഗിക്കുന്നത്. അത്തരം ടാഗുകൾക്ക് ക്ലോസിങ് ടാഗ് കാണുകയില്ല. അവ ശൂന്യ ടാഗുകൾ അഥവാ എംപ്റ്റി ടാഗുകൾ എന്നറിയപ്പെടുന്നു. ഈ ടാഗുകൾ ഒരു സെക്ഷനിലേക്കുള്ളവയല്ല.
, <HR>, <IMG> തുടങ്ങിയവ ശൂന്യ ടാഗുകൾക്ക് ഉദാഹരണങ്ങളാണ്. ഈ അധ്യായത്തിലെ തുടർന്നുള്ള ഭാഗങ്ങളിൽ ഇത്തരം ടാഗുകൾ നമുക്ക് കാണാനാകും.

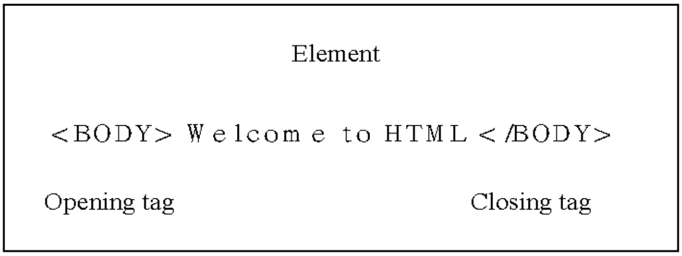
4.7.4 ടാഗുകളുടെ ആട്രിബ്യൂട്ടുകൾ (Attributes of tags)

നിറം, അളവ്, സ്ഥാനം, വിന്യാസം, അല്ലെങ്കിൽ മറ്റ് ആകാരങ്ങൾ തുടങ്ങിയ അധികവിവരങ്ങൾ വെബ് ബ്രൗസറിന് നൽകാനായി ചില ഘടകങ്ങൾ ഓപ്പണിങ് ടാഗിനുള്ളിൽ നൽകുന്നത് പതിവാണ്. ഈ ഘടകങ്ങളെ ആട്രിബ്യൂട്ടുകൾ എന്ന് വിളിക്കുന്നു. ഭൂരിഭാഗം ആട്രിബ്യൂട്ടുകൾക്കും ഒരു വില നൽകേണ്ടത് ആവശ്യമാണ്. HTML-ൽ ഏക ഉദ്ധരണികൾക്കുള്ളിലോ ജോഡികളായ ഉദ്ധരണികൾക്കുള്ളിലോ ആണ് വില നൽകുന്നത് (അതായത്, ആട്രിബ്യൂട്ട്='വില' അല്ലെങ്കിൽ ആട്രിബ്യൂട്ട് ="വില"). ആവശ്യാനുസരണം ഉപയോഗിക്കാവുന്ന ഒരു കൂട്ടം ആട്രിബ്യൂട്ടുകൾ ഓരോ ടാഗിനോടൊപ്പവും ഉണ്ടായിരിക്കും. ഒരു ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുകയാണെങ്കിൽ, സാധാരണയായി ടാഗ് നാമത്തിനുശേഷം ഒരു സ്പെയ്സ് കഴിഞ്ഞാകും അത് നൽകുന്നത്. ഒന്നിലധികം ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിക്കുമ്പോൾ, അവയുടെ ക്രമത്തിന് പ്രാധാന്യം നൽകേണ്ടതില്ല.

ഉദാഹരണമായി, ഒരു വെബ് പേജിലെ പശ്ചാത്തലനിറം മഞ്ഞയാക്കി മാറ്റാൻ `<BODY Bgcolor = "Yellow">`. ഇവിടെ Bgcolor എന്നത് <BODY> ടാഗിന്റെ ആട്രിബ്യൂട്ടും Yellow എന്നത് അതിന്റെ വിലയുമാണ്. ടാഗുകളും അവയുടെ ആട്രിബ്യൂട്ടുകളും തുടർന്ന് വരുന്ന ഭാഗങ്ങളിൽ നാം ചർച്ച ചെയ്യുന്നതാണ്.

4.7.5 HTML എലമെന്റുകൾ (HTML Elements)

ഒരു ജോഡി ടാഗുകളും അവയ്ക്കിടയിൽ ഉൾക്കൊള്ളിച്ചിട്ടുള്ള ഉള്ളടക്കവും ചേർന്നതാണ് എലമെന്റ്. ചിത്രം 4.16-ൽ കാണിച്ചിരിക്കുന്നത് പോലെ, ബോഡി എലമെന്റിൽ ഓപ്പണിങ് ടാഗായ <BODY> യും ക്ലോസിങ് ടാഗായ </BODY> യും അവയ്ക്കിടയിലുള്ള ഉള്ളടക്കവും ഉണ്ടാകും.



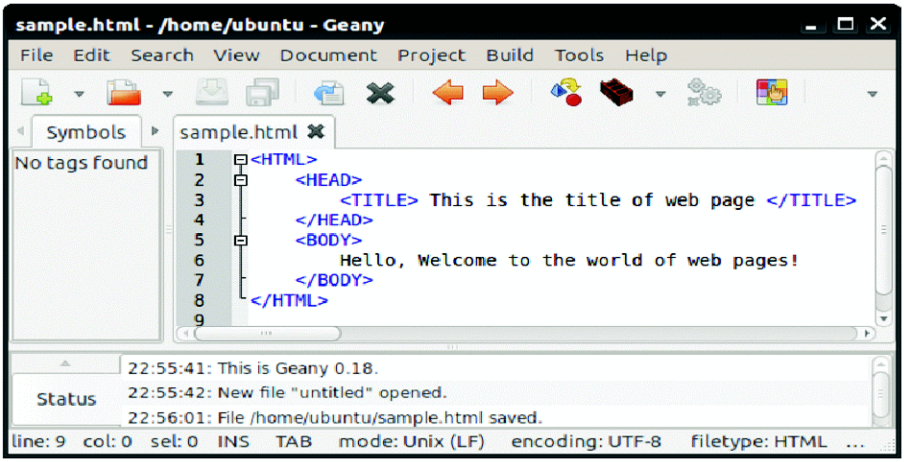
ചിത്രം 4.16 : ഒരു HTML ഘടകം

ഒരു HTML ലിഖിതത്തിന്റെ അടിസ്ഥാനഘടനയിൽ നാല് ജോഡി ടാഗുകളുണ്ടാകും. അവയാണ് :

```
<HTML>    </HTML>
<HEAD>    </HEAD>
<TITLE>   </TITLE>
<BODY>    </BODY>
```

4.8 ഒരു HTML ഡോക്യുമെന്റ് നിർമിക്കുന്നു (Creating an HTML document)

ഉദാഹരണം 4.17-ൽ തന്നിട്ടുള്ള HTML കോഡ് ഉപയോഗിച്ച് ലളിതമായ ഒരു വെബ് പേജ് നമുക്ക് നിർമിക്കാം. ജീനി (Geany), ജിഎഡിറ്റ് (Gedit), ടെക്സ്റ്റ്പാഡ് (TextPad), നോട്ട്പാഡ് (NotePad), നോട്ട്പാഡ്++ (NotePad++) തുടങ്ങിയ ടെക്സ്റ്റ് എഡിറ്ററുകളിൽ ഏതെങ്കിലും ഉപയോഗിച്ച് HTML ലിഖിതം നിർമിക്കാം. ഒരു ഫയൽ നാമവും എക്സ്റ്റെൻഷൻ ആയി .htm അല്ലെങ്കിൽ .html എന്നും നൽകി ഫയൽ സേവ് ചെയ്യണം (ഉദാഹരണമായി, sample.html). ജീനി എഡിറ്ററിൽ Sample.html എന്ന പേരിൽ സേവ് ചെയ്തിട്ടുള്ള HTML കോഡ് നമുക്ക് ചിത്രം 4.17-ൽ കാണാം.



ചിത്രം 4.17 : ജീനി എഡിറ്ററിലുള്ള HTML കോഡ്

ഒരു HTML ഡോക്യുമെന്റ് ബ്രൗസറിലൂടെ കാണുന്നു (Viewing an HTML document in a Browser)

HTML ലിഖിതം തയാറാക്കിക്കഴിഞ്ഞാൽ, അത് ബ്രൗസർ ഉപയോഗിച്ച് കാണാവുന്നതാണ്. മോസില്ല ഫയർഫോക്സ് (Mozilla Firefox), ഗൂഗിൾ ക്രോം (Google Chrome), ഇന്റർനെറ്റ് എക്സ്പ്ലൊറർ (Internet Explorer), നെറ്റ്സ്കേപ്പ് നാവിഗേറ്റർ (Netscape Navigator) തുടങ്ങിയ ധാരാളം ബ്രൗസറുകളുണ്ട്. മോസില്ല ഫയർഫോക്സിൽ വെബ് പേജ് തുറന്നപ്പോൾ കിട്ടിയ സ്ക്രീൻ ചിത്രങ്ങളാണ് ഈ പുസ്തകത്തിൽ തന്നിട്ടുള്ളത്. ജീനിയുടെ ടൂൾബാറിലുള്ള Execute (എക്സിക്യൂട്ട്) ബട്ടൺ ക്ലിക്ക് ചെയ്യുമ്പോഴും, HTML കോഡിന്റെ ഔട്ട്പുട്ട് ലഭിക്കും.

4.9 അനിവാര്യമായ HTML ടാഗുകൾ (Essential HTML tags)

ഒരു വെബ് പേജ് നിർമിക്കുന്നതിന് അവശ്യം വേണ്ട ടാഗുകളെക്കുറിച്ച് നമുക്ക് മനസ്സിലാക്കാം. ടാഗുകൾ, അവയുടെ ഉപയോഗം, ബന്ധപ്പെട്ടുകിടക്കുന്ന ആട്രിബ്യൂട്ടുകളും അവയുടെ വിലകളും, ബ്രൗസർ വിൻഡോയിൽ അവയുടെ രൂപം തുടങ്ങിയവ ഇവിടെ വിശദീകരിക്കുന്നു.

4.9.1 <HTML>- ഒരു HTML പേജ് തുടങ്ങുന്നു (<HTML> - Starting an HTML page)

HTML ഡോക്യുമെന്റ് പൂർണമായി ഒരു ജോഡി <HTML>, </HTML> എന്നീ ടാഗ് ജോഡികൾക്കുള്ളിൽ വലയം ചെയ്തിരിക്കുന്നു. ഒരു HTML ഡോക്യുമെന്റ് തിരിച്ചറിയാനാണ് <HTML> ടാഗ് ഉപയോഗിക്കുന്നത്. പൊതുവായിപ്പറഞ്ഞാൽ ഒരു HTML പേജിൽ <HTML> ആദ്യ ടാഗും </HTML> അവസാന ടാഗുമായിരിക്കും. വെബ് പേജിലെ മറ്റെല്ലാം ഈ രണ്ട് ടാഗുകൾക്ക് ഇടയിലായിരിക്കും. അതായത് ഹെഡ് ഭാഗവും ബോഡി ഭാഗവും <HTML>, </HTML> എന്നീ ടാഗുകൾക്കിടയിലായിരിക്കും ഉൾക്കൊള്ളിച്ചിരിക്കുന്നത്. ഇത് ഒരു ജോഡി കണ്ടെയ്നർ ടാഗുകളാണ്. **Dir**, **Lang** എന്നിവയാണ് <HTML> ടാഗിന്റെ പ്രധാന ആട്രിബ്യൂട്ടുകൾ.

Dir

വെബ് പേജിൽ പ്രദർശിപ്പിക്കുന്ന (കാണിക്കുന്ന) അക്ഷരങ്ങളുടെ ദിശ സൂചിപ്പിക്കുന്നതിനാണ് <HTML> ടാഗിലെ Dir ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. **ltr** (ലെഫ്റ്റ് ടു റൈറ്റ് - ഇടത്ത് നിന്ന് വലത്തേക്ക്) **rtl** (റൈറ്റ് ടു ലെഫ്റ്റ് - വലത്ത് നിന്ന് ഇടത്തേക്ക്) എന്നീ വിലകളാണ് ഈ ആട്രിബ്യൂട്ടിനുള്ളത്. ഈ ആട്രിബ്യൂട്ടിന്റെ തനതുവില വില ltr ആയിരിക്കും. ഉള്ളടക്കം അവതരിപ്പിക്കുന്നതിന് ഹീബ്രു, അറബിക്, ചൈനീസ് തുടങ്ങിയ ഭാഷകൾ ഉപയോഗിക്കുമ്പോൾ rtl എന്ന വില നൽകണം. ഉദാഹരണമായി <HTML Dir = "rtl"> എന്ന കോഡ് ഡോക്യുമെന്റ് വലത് വശത്ത് നിന്നും ഇടത് വശത്തേക്ക് വായിക്കണമെന്ന് സൂചിപ്പിക്കുന്നു.

Sl. No.	Code	Language
1	En	English
2	Fr	French
3	De	German
4	It	Italian
5	El	Greak
6	Es	Spanish
7	Ar	Arabic
8	Ja	Japanese
9	Hi	Hindi
10	Ru	Russian

പട്ടിക 4.4: പ്രചാരത്തിലുള്ള ചില ഭാഷകളും അവയുടെ കോഡുകളും

Lang

ലിഖിതത്തിനുള്ളിൽ പൊതുവായി ഉപയോഗിക്കുന്ന ഭാഷയെ സൂചിപ്പിക്കുന്നതിന് <HTML> ടാഗിന്റെ **Lang** ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. "en" എന്നത് ഇംഗ്ലീഷ് ഭാഷയേയും "it" എന്നത് ഇറ്റാലിയൻ ഭാഷയേയും സൂചിപ്പിക്കുന്നു.

ഉദാഹരണമായി <HTML Lang = "ar"> എന്ന കോഡ് അറബിക് ഭാഷയിലാണ് HTML ഡോക്യുമെന്റെഴുതിയിട്ടുള്ളതെന്ന് സൂചിപ്പിക്കുന്നു. പട്ടിക 4.3-ൽ Lang ആട്രിബ്യൂട്ടിന്റെ വിലയായി ഉപയോഗിക്കാവുന്ന ചില ഭാഷാകോഡുകൾ കൊടുത്തിട്ടുണ്ട്.

4.9.2 <HEAD> ഹെഡ് ഭാഗം നിർമിക്കുന്നു (<HEAD> - Creating head)

ശീർഷകം, ഉപയോഗിച്ചിട്ടുള്ള സ്ക്രിപ്റ്റുകൾ, ശൈലി നിർവചനങ്ങൾ തുടങ്ങിയ ഡോക്യുമെന്റിനെ കുറിച്ചുള്ള വിവരങ്ങൾ HTML ലിഖിതത്തിന്റെ ഹെഡ്ഭാഗത്ത് ഉൾപ്പെടുത്തുന്നു. ഹെഡ് ഭാഗം പ്രസ്താവിക്കുന്ന ടാഗ് ജോഡികളാണ് <HEAD>, </HEAD> എന്നിവ. ഇതും ഒരു കണ്ടെയ്നർ ടാഗ് ജോഡിയാണ്.

4.9.3 <TITLE> ശീർഷകം നിർമിക്കുന്നു (<TITLE> - Creating a title)

വെബ് ബ്രൗസറിന്റെ ടൈറ്റിൽ ബാറിൽ പ്രത്യക്ഷപ്പെടുന്ന വെബ് പേജിന്റെ ശീർഷകം അടങ്ങിയിട്ടുള്ള ഒരു കണ്ടെയ്നർ ടാഗാണിത്. ശീർഷകം ഉപയോഗിച്ചാണ് സെർച്ച് എൻജിൻ പേജ് തിരിച്ചറിയുന്നത്. <HEAD>, </HEAD> എന്നീ ടാഗ് ജോഡികൾക്കുള്ളിൽ <TITLE>, </TITLE> എന്നീ ടാഗ് ജോഡികൾ ഉപയോഗിച്ച് വെബ് പേജിന് ശീർഷകം നൽകുന്നു.

4.9.4 <BODY>-വെബ് പേജിന് ഒരു ബോഡി നിർമിക്കുന്നു (<BODY> - Creating a body)

ലിഖിതത്തിന്റെ ബോഡി ഭാഗം വിവരിക്കുന്നതിന് ഉപയോഗിക്കുന്ന ബോഡി ടാഗ് ജോഡികളാണ് <BODY>, </BODY> എന്നിവ. ബ്രൗസർ വിൻഡോയിൽ പ്രദർശിപ്പിക്കേണ്ട ഉള്ളടക്കം ഈ ഭാഗത്ത് ഉൾക്കൊള്ളിക്കുന്നു. ഡോക്യുമെന്റിന്റെ ഉള്ളടക്കം നിർവചിക്കുന്ന മറ്റെല്ലാ ടാഗുകളും ബോഡി ഭാഗത്താണ് ഉൾപ്പെടുത്തുന്നത്. ഈ ടാഗുകളെക്കുറിച്ച് വിശദീകരിക്കുന്നതിന് മുമ്പ്, <BODY> ടാഗിന്റെ ആട്രിബ്യൂട്ടുകൾ ഏതെല്ലാമെന്ന് നോക്കാം.

Background

വെബ് പേജിന് ഒരു പശ്ചാത്തലചിത്രം നൽകുന്നത് ഈ ആട്രിബ്യൂട്ടാണ്. <BODY> ടാഗിന്റെ ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് പേജിനെ കൂടുതൽ ആകർഷകമാക്കുന്നു. അതിന്റെ പൊതു ഘടനയാണ്:

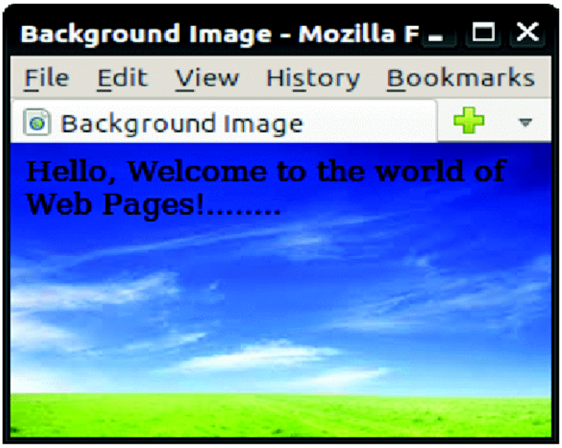
```
<BODY Background = "URL of the picture">
```

ആകാശത്തിന്റെ ചിത്രം ഒരു വെബ് പേജിന് പശ്ചാത്തലമായി നൽകുന്നതിനുള്ള HTML കോഡ് ഉദാഹരണം 4.2-ൽ കാണിച്ചിരിക്കുന്നു.

ഉദാഹരണം 4.2: ഒരു വെബ് പേജിന് പശ്ചാത്തലമായി ചിത്രം നൽകുന്നു

```
<HTML>
<HEAD>
   <TITLE> Background Image </TITLE>
</HEAD >
<BODY Background = "Sky.jpg">
   Hello, Welcome to the world of Web Pages!........
</BODY>
</HTML>
```

ഉദാഹരണം 4.1-ൽ ഉള്ള HTML കോഡിലെ <BODY> ടാഗിന് ഒരു Background ആട്രിബ്യൂട്ടും അതിന് വിലയായി "Sky.jpg" എന്നും നൽകിക്കൊണ്ട് <BODY Background= "Sky.jpg"> ആയി വ്യത്യാസപ്പെടുത്തിയിരിക്കുന്നു. വെബ് പേജ് തുറക്കുന്നതിന് മുമ്പ്, ചിത്രത്തിന്റെ ഫയൽ ഇപ്പോൾ ഉപയോഗിച്ചുകൊണ്ടിരിക്കുന്ന ഫോൾഡറിൽ/ഡയറക്ടറിയിൽ ഉൾപ്പെടുത്തേണ്ടതാണ്. ചിത്രം 4.18-ൽ കാണിച്ചിരിക്കുന്നത് പോലെ വെബ് പേജ് പ്രദർശിപ്പിക്കാവുന്നതാണ്.



ചിത്രം 4.18 : ഒരു പശ്ചാത്തല ചിത്രം

Bgcolor

ഡോക്യുമെന്റിന്റെ ബോഡി ഭാഗത്തെ പശ്ചാത്തലത്തിന് ഒരു നിറം നിർദേശിക്കുന്ന ആട്രിബ്യൂട്ടാണിത്. ഉദാഹരണമായി, <BODY Bgcolor = "grey"> എന്ന കോഡ് പശ്ചാത്തലത്തിന് ചാരനിറം നൽകുന്നു.

Bgcolor ആട്രിബ്യൂട്ടിന് രണ്ട് രീതിയിൽ വില നൽകാനാകും.

- *Color_name* - പശ്ചാത്തലത്തിന് നിറനാമം നിർദ്ദേശിക്കുന്നു ("red", "grey" തുടങ്ങിയവ പോലെ).
- *Hex_number* - പശ്ചാത്തലത്തിന് ഹെക്സാഡെസിമൽ കോഡിലുള്ള നിറം നിർദേശിക്കുന്നു. ("#ff6080", "#303030" തുടങ്ങിയവ പോലെ). ഓരോ പതിനാറക്ക കോഡും ഹാഷ് ചിഹ്നത്തിലാണ് ആരംഭിക്കുന്നത്.

Colour	Colour Name	Colour HEX
	Black	#000000
	Red	#FF0000
	Green	#00FF00
	Blue	#0000FF
	Yellow	#FFFF00
	Aqua	#00FFFF
	Grey	#C0C0C0
	White	#FFFFFF

പട്ടിക 4.4 : നിറങ്ങളുടെയും, അവയുടെ പേരിന്റെയും, 16 അക്ക കോഡ് വിലയുടെയും പട്ടിക

ആറക്ക സംഖ്യയും അക്ഷരങ്ങളും ചേർന്ന് നിറങ്ങൾക്ക് RGB (ചുവപ്പ്, പച്ച, നീല) വിലകൾ നൽകി പ്രദർശിപ്പിക്കുന്നു. ആറ് അക്കങ്ങളിൽ ആദ്യ രണ്ടെണ്ണം ചുവപ്പിന്റെ അളവിനെയും, അടുത്ത രണ്ട് അക്കങ്ങൾ പച്ചയുടെ തോതിനെയും, അവസാനത്തെ രണ്ടക്കങ്ങൾ നീലയുടെ അളവിനെയും 00 മുതൽ FF വരെയുള്ള വിലകൾ കൊണ്ട് സൂചിപ്പിക്കുന്നു. ഉദാഹരണമായി, #000000 എന്നത് കറുപ്പും, #FF0000 എന്നത് തിളക്കമുള്ള ചുവപ്പും, #00FF00 എന്നത് തിളങ്ങുന്ന പച്ചയും, #FFFFFF എന്നത് വെള്ളയുമായിരിക്കും (മൂന്ന് നിറങ്ങളും പൂർണമായും ഉൾക്കൊള്ളിച്ചിരിക്കുന്നു). ഹെക്സാ

ഡെസിമൽ കോഡിലെ വിവിധ നിറക്കൂട്ടുകൾ നമുക്ക് പരീക്ഷിക്കാവുന്നതാണ്. ചില നിറങ്ങളും, അവയുടെ പേരുകളും, പതിനാറക്ക കോഡ് വിലകളും പട്ടിക 4.5-ൽ കാണിച്ചിരിക്കുന്നു.

Text

പേജിന്റെ ഉള്ളടക്കത്തിലുള്ള വാക്യങ്ങളുടെ നിറം സൂചിപ്പിക്കുന്ന ആട്രിബ്യൂട്ടാണ് ഇത്. തനത് രൂപത്തിൽ വെള്ള/ചാര നിറമുള്ള പശ്ചാത്തലവും വാക്യങ്ങൾക്ക് കറുപ്പ് നിറവുമായി ബ്രൗസർ കാണിക്കുന്നു. Bgcolor ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് പശ്ചാത്തലത്തിന്റെ നിറം എങ്ങനെ വ്യത്യാസപ്പെടുത്താമെന്ന് നാം ചർച്ച ചെയ്ത് കഴിഞ്ഞു. അതുപോലെ, Text ആട്രിബ്യൂട്ടുപയോഗിച്ചുകൊണ്ട് വാക്യങ്ങളുടെ നിറം വ്യത്യാസപ്പെടുത്താനാകും. ഉദാഹരണമായി, <BODY Text = "yellow"> എന്നത് മഞ്ഞ നിറത്തിലുള്ള വാക്യത്തെ പ്രദർശിപ്പിക്കുന്നു. Bgcolor-ന് വില നൽകിയത് പോലെ നിറത്തിന്റെ പേരോ പതിനാറക്ക കോഡോ ഉപയോഗിച്ച് Text ആട്രിബ്യൂട്ടിന് വില നൽകാവുന്നതാണ്. ഉദാഹരണമായി Text = "Blue" അല്ലെങ്കിൽ Text = "#00FFDD" തുടങ്ങിയവ.

Link, Alink and Vlink

ക്ലിക്ക് ചെയ്യുമ്പോൾ, ഒരു ഡോക്യുമെന്റിൽ നിന്ന് മറ്റൊന്നിലേക്കോ, അല്ലെങ്കിൽ അതേ ഡോക്യുമെന്റിന്റെ വേറൊരു ഭാഗത്തേക്കോ നീങ്ങാൻ സഹായിക്കുന്ന ഒരു ഘടകം/ഒരു വാക്യം/ഒരു ചിത്രമാണ് ഹൈപ്പർലിങ്ക്. ഒരു ലിഖിതത്തെ മൊത്തമായോ അല്ലെങ്കിൽ ഡോക്യുമെന്റിലുള്ള പ്രത്യേക ഭാഗത്തെയോ ആണ് ഹൈപ്പർ ലിങ്ക് സൂചിപ്പിക്കുന്നത്. ഹൈപ്പർ ലിങ്കിനെക്കുറിച്ച് വിശദമായി പിന്നീട് നമുക്ക് ഈ അധ്യായത്തിൽ ചർച്ച ചെയ്യാം.

Link: കാണുന്നയാൾ സന്ദർശിക്കാത്ത ഹൈപ്പർ ലിങ്കിന്റെ നിറം സൂചിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്ന ആട്രിബ്യൂട്ടാണ് ഇത്. Link ആട്രിബ്യൂട്ടിന്റെ തനത് നിറം നീലയായിരിക്കും.

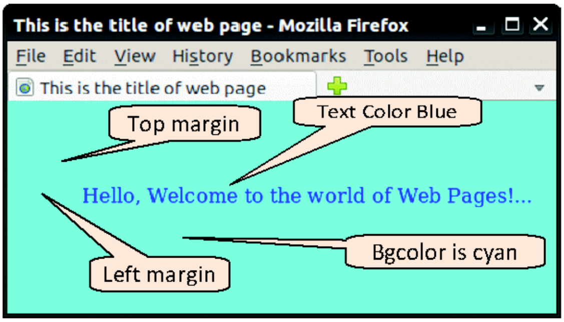
Alink: സജീവമായ ഹൈപ്പർ ലിങ്കിന്റെ നിറമാണ് ഇത് സൂചിപ്പിക്കുന്നത്. മൗസ് അതിനുമേൽ ക്ലിക്ക് ചെയ്യുന്ന നിമിഷം മാത്രമാണ് ആ ലിങ്ക് സജീവമാകുന്നത്. ആയതിനാൽ, തിരഞ്ഞെടുക്കുമ്പോൾ ഹൈപ്പർ ലിങ്കിന്റെ നിറം Alink ന്റെ വില സൂചിപ്പിക്കും. ഇവിടെ ഹൈപ്പർ ലിങ്കിന്റെ തനത് നിറം പച്ചയായിരിക്കും.

Vlink: കാണുന്നയാൾ സന്ദർശിച്ച ഹൈപ്പർ ലിങ്കിന്റെ നിറം സൂചിപ്പിക്കാനാണ് ഇത് നൽകുന്നത്. Vlink ന്റെ തനത് നിറം ധൂമനിറം (പർപ്പിൾ) ആയിരിക്കും.

Leftmargin, Topmargin

പേജിന്റെ അരികിൽ നിന്നുള്ള ശൂന്യസ്ഥലത്തെയാണ് മാർജിൻ സൂചിപ്പിക്കുന്നത്. ബ്രൗസർ ജാലകത്തിൽ ഡോക്യുമെന്റിന്റെ ഇടത് ഭാഗത്ത് കുറച്ച് ശൂന്യപ്രദേശം ഒഴിച്ചിടാനായി Leftmargin ആട്രിബ്യൂട്ടും മുകളറ്റത്ത് കുറച്ച് ശൂന്യ സ്ഥലം നൽകാനായി Topmargin ആട്രിബ്യൂട്ടും ഉപയോഗിക്കുന്നു. വിലനൽകുന്നത് പിക്സൽ പോയിന്റുകളുടെ എണ്ണമായാണ്.

ഉദാഹരണമായി, <BODY Leftmargin = "60" Topmargin = "70"> എന്നത്, പ്രദർശിപ്പിക്കേണ്ട വാചകം വെബ് പേജിന്റെ ഇടത്തേ അറ്റത്ത് നിന്ന് 60 പിക്സൽ പോയിന്റ് അകലെയും, മുകളറ്റത്ത് നിന്ന് 70 പിക്സൽ പോയിന്റ് അകലെയും ആണെന്ന് കാണിക്കുന്നു. <BODY> ടാഗിന്റെ Bgcolor, Text, Top margin, Leftmargin എന്നീ ആട്രിബ്യൂട്ടുകൾ ഉദാഹരണം 4.3-ലെ കോഡിലൂടെ വിശദീകരിച്ചിരിക്കുന്നു. ചിത്രം 4.19-ൽ അനുബന്ധ വെബ് പേജ് കാണാവുന്നതാണ്.



ചിത്രം 4.19 : BODY ടാഗിനോടൊപ്പം ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിച്ചിരിക്കുന്നു

ഉദാഹരണം 4.3: ഒരു വെബ്പേജിന്റെ പശ്ചാത്തലത്തിന് നിറം കൊടുക്കുന്നു

```
<HTML>
<HEAD>
     <TITLE> This is the title of web page </TITLE>
</HEAD>
<BODY Bgcolor= "cyan" Text= " Blue"
      Topmargin= "70" Leftmargin= "60">
      Hello, Welcome to the world of Web Pages!........
</BODY>
</HTML>
```

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. HTML പ്രതിനിധീകരിക്കുന്നത് ________ നെയാണ്.
2. എന്താണ് കണ്ടെയ്നർ ടാഗ്?
3. അവസാനിപ്പിക്കുന്ന ടാഗ് ഇല്ലാതെ തുടക്ക ടാഗ് മാത്രം വേണ്ട ഇനം ടാഗാണ് ________.
4. ശരിയോ തെറ്റോ എന്ന് പ്രസ്താവിക്കുക.
    a. ടാഗുകൾ കേയ്സ് സെൻസിറ്റീവ് ആണ്.
    b. <BODY> ടാഗിന്റെ ഒരു ആട്രിബ്യൂട്ടാണ് Bgcolor.
    c. ഒരു എംപ്റ്റി ടാഗാണ് <TITLE>.
    d. <HEAD> ടാഗിന്റെ ഒരു ആട്രിബ്യൂട്ടാണ് Dir.
5. <HTML> ടാഗിന്റെ ആട്രിബ്യൂട്ടുകളുടെ പേരെഴുതുക.
6. ഒരു ടാഗിൽ ആട്രിബ്യൂട്ടുകളുടെ ഉപയോഗമെന്താണ്?
7. <BODY> ടാഗിന്റെ വിവിധ ആട്രിബ്യൂട്ടുകൾ എഴുതുക.

നമുക്കു ചെയ്യാം

നിങ്ങളുടെ വിദ്യാലയത്തിന്റെ പേരും അതോടൊപ്പം പശ്ചാത്തലത്തിൽ വിദ്യാലയത്തിന്റെ ചിത്രവും പ്രദർശിപ്പിക്കുന്ന ഒരു HTML ലിഖിതം നിർമിക്കുക. തുടർന്ന് വാക്യത്തിന്റെ നിറവും പശ്ചാത്തലവും വ്യത്യാസപ്പെടുത്തി പേജ് പരിഷ്കരിക്കുക.

4.10 പൊതുവായ ചില ടാഗുകൾ (Some common tags)

HTML ഡോക്യുമെന്റിനു വേണ്ട അടിസ്ഥാന ടാഗുകളെയും അവയുടെ ആട്രിബ്യൂട്ടുകളെയും കുറിച്ച് നാം ചർച്ച ചെയ്തു കഴിഞ്ഞു. വെബ് പേജിലെ ഉള്ളടക്കം കൂടുതൽ ആകർഷകമാക്കാൻ ഉപയോഗിക്കുന്ന മറ്റു ടാഗുകളുമുണ്ട്. ഇക്കൂട്ടത്തിൽ, HTML ലിഖിതത്തിന്റെ ബോഡി ഭാഗത്തെ ഉള്ളടക്കത്തിലുള്ള വാക്യങ്ങളുടെ ഘടന രൂപപ്പെടുത്താൻ ഉപയോഗിക്കുന്ന ടാഗുകളെ ഫോർമാറ്റിങ് ടാഗുകൾ എന്ന് വിളിക്കുന്നു. ബോഡി ഭാഗത്തെ ഉള്ളടക്കത്തിന്റെ രൂപഘടനയ്ക്ക് ആവശ്യമായ ടാഗുകളിൽ ചിലത് നമുക്ക് പരിചയപ്പെടാം.

4.10.1 <H1>, <H2>, <H3>, <H4>, <H5>, <H6> ശീർഷക ടാഗുകൾ (Heading tags)

ഒരു ഖണ്ഡികയുടെ തുടക്കത്തിൽ അത് എന്താണെന്ന് വിശദീകരിക്കുന്നതിനായി കൊടുക്കുന്ന ഒരു വാക്കോ, വാക്യമോ, വാചകമോ ആണ് ശീർഷകം. ബോഡി ഭാഗത്തെ സാധാരണ വാക്യങ്ങളെക്കാൾ വലുതോ കടുപ്പിച്ചതോ ആയ സവിശേഷതയുള്ള അക്ഷരങ്ങൾ കൊണ്ടാണ് ശീർഷകം പ്രദർശിപ്പിക്കുന്നത്. **<H1>** മുതൽ **<H6>** വരെ ആറ് തലങ്ങളിലുള്ള ശീർഷകങ്ങൾ HTML-ൽ ഉണ്ട്. ഇവിടെ **<H1>** ഏറ്റവും വലുതും **<H6>** ഏറ്റവും ചെറുമായ ശീർഷകങ്ങളാണ് നിർമിക്കുന്നത്. ശീർഷകം പ്രദർശിപ്പിക്കുന്നതിന് മുന്നിലും പിന്നിലുമായി ബ്രൗസർ ഓരോവരി ചേർക്കുന്നു. ഈ ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്രിബ്യൂട്ട് **Align** ആണ്. അതിന് നൽകാവുന്ന വിലകൾ താഴെ തന്നിരിക്കുന്നു.

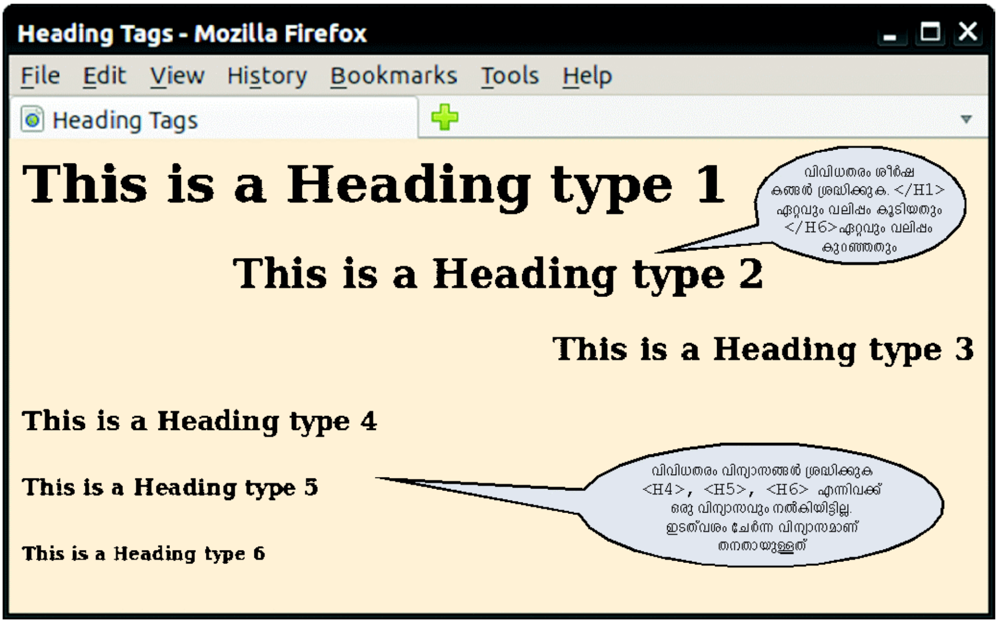
Left : വാക്യം ഇടത് അരികിനോട് (മാർജിനുമായി) പൊരുത്തപ്പെട്ട് നിൽക്കും.
Right : വാക്യം വലത് അരികിനോട് (മാർജിനുമായി) പൊരുത്തപ്പെട്ട് നിൽക്കും.
Center : വാക്യം പേജിന്റെ മാധ്യഭാഗത്ത് പ്രത്യക്ഷപ്പെടും.

വിവിധങ്ങളായ ശീർഷകങ്ങളും, പൊരുത്തപ്പെടലുകളും ഉദാഹരണം 4.4 - ൽ കാണിച്ചിരിക്കുന്നതോടൊപ്പം അനുബന്ധ വെബ് പേജ് ചിത്രം 4.20-ൽ പ്രദർശിപ്പിച്ചിരിക്കുന്നു.

ഉദാഹരണം 4.4: വിവിധ ശീർഷക ശൈലികൾ വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
   <TITLE> Heading Tags </TITLE>
</HEAD >
<BODY Bgcolor= "#FFEFD5">
   <H1 Align= "left"> This is a Heading type 1 </H1>
   <H2 Align= "center"> This is a Heading type 2 </H2>
     <H3 Align= "right"> This is a Heading type 3 </H3>
     <H4> This is a Heading type 4 </H4>
```

```
    <H5> This is a Heading type 5 </H5>
    <H6> This is a Heading type 6 </H6>
</BODY>
</HTML>
```



ചിത്രം 4.20 : വ്യത്യസ്ത ശീർഷക ടാഗുകളോടൊപ്പമുള്ള വിവിധ പൊരുത്തപ്പെടലുകൾ

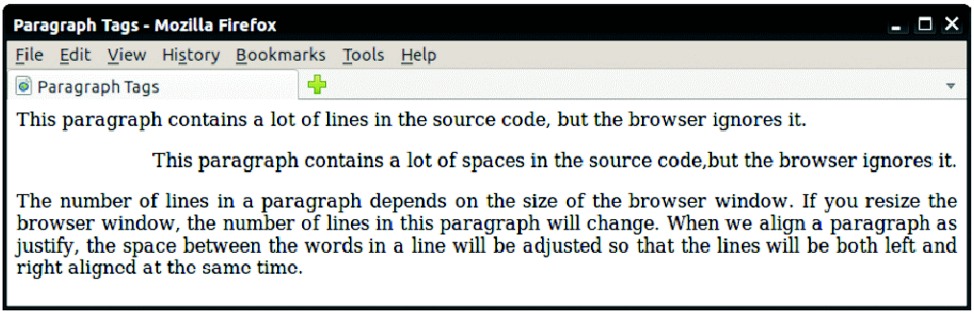
4.10.2 <P> ടാഗ് – ഖണ്ഡികകൾ നിർമിക്കുന്നു (Creating Paragraphs)

<BODY> ടാഗിലെ വാകൃങ്ങളെ വിവിധ ഖണ്ഡികകളിൽ ആക്കുവാൻ നമ്മെ സഹായിക്കുന്നത് **<P>** ടാഗാണ്. ഇത് ഒരു പുതിയ ഖണ്ഡികയെ സൂചിപ്പിക്കുകയും ഒരു ശൂന്യവരി ഖണ്ഡികയ്ക്ക് മുന്നിൽ ചേർക്കാൻ ബ്രൗസറിനോട് നിർദേശിക്കുകയും ചെയ്യുന്നു. മറ്റ് ഏതൊരു വേർഡ് പ്രൊസ്സസറിലെയും ഖണ്ഡികകൾ പോലെ തന്നെയാണ് HTML-ലെ ഖണ്ഡികയും. ഒരു ഖണ്ഡിക <P> ടാഗിൽ തുടങ്ങി </P> ടാഗിൽ അവസാനിക്കുന്നു. Left, right, center, justify എന്നീ വിലകൾ നൽകി ഖണ്ഡികയിലെ വാക്യങ്ങളുടെ വിന്യാസം ഉറപ്പിക്കുന്നത് **Align** ആട്രിബ്യൂട്ട് ഉപയോഗിച്ചു കൊണ്ടാണ്. വ്യത്യസ്ത വിന്യാസം നൽകി ഖണ്ഡികകൾ എങ്ങനെ രൂപപ്പെടുത്തുന്നുവെന്ന് ഉദാഹരണം 4.5-ലുള്ള കോഡിൽ തന്നിരിക്കുന്നു. അങ്ങനെ ലഭിക്കുന്ന വെബ് പേജ് ചിത്രം 4.21-ൽ കാണിച്ചിരിക്കുന്നു

ഉദാഹരണം 4.5: വിവിധ വിന്യാസത്തോട് കൂടിയ ഖണ്ഡികകൾ രൂപകൽപന ചെയ്യുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
    <TITLE> Paragraph Tags </TITLE>
```

```
</HEAD>
<BODY>
     <P>
     This paragraph contains
     a lot of lines in the source code,
     but the browser ignores it.
     </P>
     <P Align= "right">
     This paragraph contains a lot of spaces
     in the source code,but the browser ignores it.
     </P>
     <P Align= "justify">
     The number of lines in a paragraph depends on
     the size of the browser window.
     If you resize the browser window, the number of lines
     in this paragraph will change.
     When we align a paragraph as justify, the space between
     the words in a line will be adjusted
     so that the lines will be both left and right aligned
     at the same time.
   </P>
</BODY>
</HTML>
```



ചിത്രം 4.21: Align ആട്രിബ്യൂട്ട് ഉൾക്കൊള്ളിച്ചിട്ടുള്ള <P> ടാഗിന്റെ ഉപയോഗം

ചിത്രം 4.5-ലെ സോഴ്സ് കോഡിൽ അധിക ശൂന്യസ്ഥലങ്ങളോട് കൂടിയ മൂന്ന് വരികൾ ഒന്നാം ഖണ്ഡികയിലും, രണ്ട് വരികൾ രണ്ടാം ഖണ്ഡികയിലും ഉൾക്കൊള്ളിച്ചിട്ടുള്ളത് നമുക്ക് കാണാനാകും. എന്നാൽ ചിത്രം 4.21-ൽ കാണിച്ചിട്ടുള്ള ഈ കോഡിന്റെ വെബ് പേജിൽ ആദ്യത്തെ രണ്ട് ഖണ്ഡികകൾക്ക് ഓരോ വരിയാണുള്ളത്. അതായത്, പേജ് പ്രദർശിപ്പിക്കുമ്പോൾ, അധികമുള്ള വരികളും ശൂന്യസ്ഥലങ്ങളും ബ്രൗസർ നീക്കം ചെയ്യുന്നു. രണ്ടാം ഖണ്ഡിക വലതുവശത്ത് പൊരുത്തപ്പെട്ട് നിൽക്കുന്നതും മൂന്നാം ഖണ്ഡിക ഇരുവശങ്ങളിലും പൊരുത്തപ്പെട്ട് നിൽക്കുന്നതും ശ്രദ്ധിക്കുക. ഒന്നിലധികം

ശൂന്യസ്ഥലങ്ങളും വരികളും ഒരൊറ്റ ശൂന്യസ്ഥലമായാണ് കരുതുന്നത്. അതുകൊണ്ട്, HTML കോഡിൽ അധികമായി ശൂന്യസ്ഥലങ്ങളോ, വരികളോ ചേർക്കുന്നതുകൊണ്ട് ഔട്ട്പുട്ട് വ്യത്യാസപ്പെടുത്താനാകില്ല. എന്നാൽ ഇത് **
** ടാഗ് ഉപയോഗിച്ച് സാധ്യമാകും. ഉള്ളടക്ക വാക്യങ്ങളിൽ **
** ടാഗ് ഉപയോഗിച്ചുകൊണ്ട് അധികവരി ചേർക്കാം. വലുതോ ചെറുതോ ആയ സ്ക്രീനുകളിലും, വലുപ്പം വ്യത്യാസപ്പെടുത്തിയ വിൻഡോകളിലും ലഭിക്കുന്ന വെബ് പേജുകൾ, ചിത്രം 4.21-ൽ തന്നിട്ടുള്ളതിന് സമാനമായിരിക്കുകയില്ല എന്നത് ശ്രദ്ധിക്കുക.

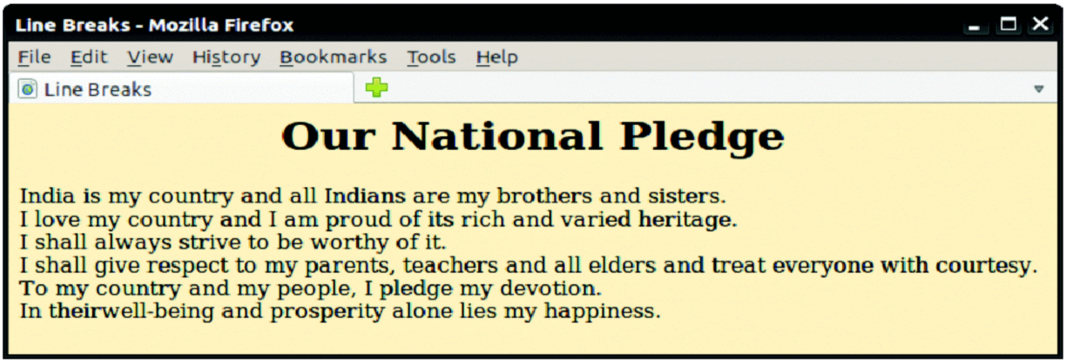
4.10.3
 ടാഗ് – വരി വിഘടിപ്പിക്കുന്നു (Inserting line break)

ഒരു വെബ് പേജിലെ വരി മാറ്റിയെഴുതുക എന്നതാണ് **
** ടാഗിന്റെ ഉദ്ദേശം. നിലവിലുള്ള വരിയിലെ വാക്യം മുറിച്ച് അടുത്ത വരിയുടെ തുടക്കത്തിൽ നിന്നാരംഭിക്കാനാണ്
 ടാഗുപയോഗിക്കുന്നത്. ഇതൊരു എംറ്റി ടാഗാണ്.

ഉദാഹരണം 4.6-ൽ തന്നിട്ടുള്ള HTML കോഡ് നമ്മുടെ രാജ്യത്തിന്റെ പ്രതിജ്ഞ പ്രദർശിപ്പിക്കുന്നു. ചിത്രം 4.22-ൽ കാണിച്ചിട്ടുള്ള പരിണത ഫലമായി ലഭിക്കുന്ന പേജിൽ
 ടാഗിന്റെ ഉപയോഗവും
, <P> എന്നീ ടാഗുകൾ തമ്മിലുള്ള വ്യത്യാസവും നമുക്ക് കാണാം.

ഉദാഹരണം 4.6: ദേശീയ പ്രതിജ്ഞയിൽ വരികൾ മുറിക്കുന്നത് (Line break) കാണിക്കുവാൻ വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
   <TITLE> Line Breaks </TITLE>
</HEAD >
<BODY Bgcolor = "#FFEFC5">
   <H1 Align = "center"> Our National Pledge </H1>
   <P>India is my country and all Indians
     are my brothers and sisters.<BR>
     I love my country and I am proud of its
     rich and varied heritage.<BR>
     I shall always strive to be worthy of it.<BR>
     I shall give respect to my parents, teachers and all
     elders and treat everyone with courtesy.<BR>
     To my country and my people, I pledge my
     devotion. <BR>In theirwell-being and prosperity alone
     lies my happiness.
   </P>
</BODY>
</HTML>
```



*ചിത്രം 4.22 :
 ടാഗിന്റെ ഉപയോഗം കാണിക്കുന്ന ഔട്ട്പുട്ട്*

നമുക്കു ചെയ്യാം

*ഉദാഹരണം 4.6, ചിത്രം 4.24 എന്നിവ നൽകുന്ന സൂചനയ്ക്ക് അനുസൃതമായി, <P> ടാഗും
 ടാഗും തമ്മിൽ വേർതിരിക്കുന്ന ആശയങ്ങൾകൊണ്ട് പട്ടിക 4.5 പൂരിപ്പിക്കുക.*

<P> tag	 tag
	Breaks the current line and continues to the next line.
Container tag	

*ചിത്രം 4.3 : <P> ടാഗും
 ടാഗും*

4.10.4 <HR> ടാഗ് – തിരശ്ചീനമായ വര സൃഷ്ടിക്കുന്നു (Creating horizontal line)

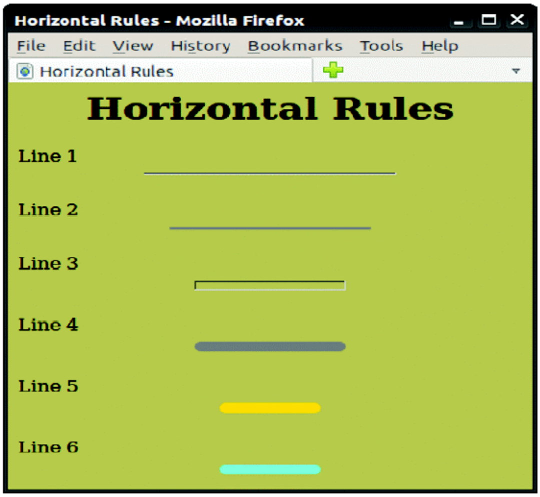
ബ്രൗസർ വിൻഡോയുടെ വീതിക്ക് കുറുകെ തിരശ്ചീനമായ ഒരു വര നിർമിക്കുന്നത് **<HR>** ടാഗാണ്. അതിന്റെ ആട്രിബ്യൂട്ടുകളായ **Size, Width** എന്നിവ ഉപയോഗിച്ച് നമുക്ക് വരയുടെ **വീതിയും കനവും** വ്യത്യാസപ്പെടുത്താം. കനത്തിന് പിക്സലുകളായും വീതിക്ക് പിക്സലുകളായോ മൊത്തം വീതിയുടെ ശതമാനമായോ വില നൽകുന്നു. <HR> ടാഗിന്റെ മറ്റ് രണ്ട് ആട്രിബ്യൂട്ടുകളാണ് Noshade, Color എന്നിവ. Noshade ആട്രിബ്യൂട്ടിന് പ്രത്യേക വില കാണുകയില്ല. വരയ്ക്ക് നിറം കൊടുക്കുന്നത് Color ആട്രിബ്യൂട്ടാണ്. left, center, right എന്നിങ്ങനെ വരയുടെ വിന്യാസത്തിന് ഉപയോഗിക്കുന്ന Align ആട്രിബ്യൂട്ടാണ് മറ്റൊന്ന്. <HR> ഒരു എംറ്റി ടാഗാണ് എന്ന് ഓർക്കുക.

<HR> ടാഗിന്റെയും അതിന്റെ ആട്രിബ്യൂട്ടുകളുടെയും പ്രവർത്തനം വ്യക്തമാക്കുന്ന കോഡ് ഉദാഹരണം 4.7-ൽ തന്നിട്ടുണ്ട്. അതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 4.23-ൽ പ്രദർശിപ്പിച്ചിരിക്കുന്നു.

ഉദാഹരണം 4.7: വിവിധ തരത്തിലുള്ള രേഖകൾ വരയ്ക്കുന്നതിന്

```
<HTML>
<HEAD>
```

```
   <TITLE> Horizontal Rules </TITLE>
</HEAD >
<BODY Bgcolor= "#BDB76B">
<H1 Align= "center"> Horizontal Rules </H1>
Line 1<HR Width= "50%" Align= "center"> <BR>
Line 2<HR Width= "40%" Align= "center" Noshade> <BR>
Line 3<HR Size= "10" Width= "30%" Align= "center"> <BR>
Line 4<HR Size= "10" Width= "30%" Align= "center" Noshade><BR>
Line 5<HR Size= "10" Width= "20%" Align= "center" Noshade
          Color= "gold"> <BR>
Line 6<HR Size="10" Width="20%" Align="center" Color="Aqua">
</BODY>
</HTML>
```



ചിത്രം 4.23 : പലതരത്തിലുള്ള തിരശ്ചീനമായ രേഖകൾ

ഈ HTML ലിഖിതവും അതിന്റെ ഔട്ട്പുട്ടും പരിശോധിച്ചശേഷം, വരകളുടെ ആകൃതികളിലുള്ള സമാനതകളും വ്യത്യാസങ്ങളും നിങ്ങൾക്ക് തിരിച്ചറിയാനാകുമോ? ഇവിടെ, ആദ്യത്തെ രേഖ Size, Noshade എന്നീ ആട്രിബ്യൂട്ടുകളില്ലാതെയും, രണ്ടാമത്തെ രേഖ Size ആട്രിബ്യൂട്ടില്ലാതെയുമാണ് വരച്ചിരിക്കുന്നത്. രേഖ മൂന്നിനും, നാലിനും വലുപ്പം 10 ആണ്. എന്നാൽ മൂന്നാമത്തേതിന് Noshade ആട്രിബ്യൂട്ട് ഇല്ല. അവസാനത്തെ രണ്ട് രേഖകൾക്ക് Color ആട്രിബ്യൂട്ട് അധികമായുണ്ട്. ഓരോ അവസ്ഥയിലുമുള്ള ഈ രേഖകളുടെ ആകൃതികൾ ശ്രദ്ധിക്കുക. നിങ്ങളുടെ കമ്പ്യൂട്ടർ ലാബിലെ പ്രവർത്തനത്തിൽ സാധ്യമായ എല്ലാവിലകളും വ്യത്യസ്തങ്ങളായ ആട്രിബ്യൂട്ടുകൾക്ക് നൽകിക്കൊണ്ട് നിങ്ങൾക്ക് കോഡ് മാറ്റാവുന്നതും വെബ് പേജിൽ അപ്പോഴുണ്ടാകുന്ന മാറ്റം ദർശിക്കാവുന്നതുമാണ്.

4.10.5 <CENTER> ടാഗ് – ഉള്ളടക്കം മധ്യഭാഗത്താക്കുന്നു (Centering the content)

വെബ് പേജിലെ ഉള്ളടക്കം തിരശ്ചീനമായി മധ്യഭാഗത്തേക്ക് കൊണ്ടു വരുന്നത് **<CENTER>** ടാഗാണ്. ഉള്ളടക്കം സാധാരണയായി വാക്യങ്ങൾ, ചിത്രങ്ങൾ, പട്ടികകൾ തുടങ്ങിയവയായിരിക്കും. ഇത് ഒരു കണ്ടെയ്നർ ടാഗ് ആയതുകൊണ്ട് <CENTER>, </CENTER> എന്നീ ടാഗ് ജോഡികൾക്ക് ഇടയ്ക്ക് നൽകുന്ന ഉള്ളടക്കം ബ്രൗസർ ജാലകത്തിന്റെ മധ്യഭാഗത്ത് നിർത്തുന്നു. ഈ ടാഗിന് ആട്രിബ്യൂട്ടുകൾ ഇല്ല.

4.10.6 വാക്യങ്ങൾക്ക് രൂപഘടന നൽകുന്ന ടാഗുകൾ (Text formating tags)

വേഡ് പ്രോസസിംഗ് പ്രോഗ്രാമുകൾ ചെയ്യുന്നതു പോലെ, നമുക്ക് വെബ് പേജിലെ വാക്യങ്ങൾക്ക് രൂപഘടന നൽകാനാകും. വാക്യങ്ങളുടെ പ്രാധാന്യത്തെ സൂചിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്ന സവിശേഷതകളാണ് ബോൾഡ്, ഇറ്റാലിക്, അണ്ടർലൈൻ തുടങ്ങിയവ. ഇതിനായുള്ള HTML ടാഗുകളെക്കുറിച്ച് നമുക്ക് ചർച്ച ചെയ്യാം.

<B> - ടെക്സ്റ്റ് ബോൾഡ് ആക്കുന്നു

ഈ ടാഗ് ടെക്സ്റ്റിന്റെ ശൈലി ബോൾഡ് ആക്കുന്നു. ഉള്ളടക്കത്തിലെ വാക്യങ്ങളുടെ ബാഹ്യരൂപം കൂടുതൽ കനത്തിൽ പ്രദർശിപ്പിക്കുന്നത് <B> ടാഗാണ്.

<I> - ടെക്സ്റ്റ് ഇറ്റാലിക്സിലാക്കുന്നു

ഇത് വാക്യങ്ങൾക്ക് ചരിവ് നൽകി രൂപപ്പെടുത്തുന്നു. <I> ടാഗിനും </I> ടാഗിനുമിടയ്ക്കുള്ള ഉള്ളടക്കത്തിന് ചരിവ് നൽകുന്നു.

<U> - ടെക്സ്റ്റിന് അടിവര ഇടുന്നു

വെബ് പേജിലെ വാക്യത്തിന് അടിവര നൽകാനാണ് <U> ടാഗുപയോഗിക്കുന്നത്. <U> ടാഗിനും </U> ടാഗിനുമിടയ്ക്ക് ഉൾക്കൊള്ളിച്ചിട്ടുള്ള ഉള്ളടക്കം അടിവര ഉള്ളതായി മാറുന്നു. രൂപഘടന നൽകുന്ന ടാഗുകളായ <U>, <B>, <I> എന്നിവ ഒന്നിച്ച് ഉപയോഗിക്കാനാകും എന്നതിനാൽ, ഉള്ളടക്കം കനം കൂടിയതും, ചരിഞ്ഞതും, അടിവരയുള്ളതുമാകും.

<S>, <STRIKE> - ടെക്സ്റ്റിന് കുറുകെ വരയ്ക്കുന്നു

<S>, <STRIKE> എന്നീ ടാഗുകൾ ഉപയോഗിക്കുന്നത് ഒരേ പ്രക്രിയയ്ക്കാണ്. വാക്യങ്ങൾ കുറുകെ വെട്ടിയ രൂപത്തിൽ പ്രദർശിപ്പിക്കുന്നു. ഉദാഹരണമായി, കുറുകേ വെട്ടിയ വാക്യമാണ് ~~Thank you all~~.

<BIG> - ടെക്സ്റ്റ് വലുപ്പം ഉള്ളതാക്കുന്നു

സാധാരണ വലുപ്പത്തിലുള്ള ടെക്സ്റ്റിനേക്കാൾ വലുപ്പം ഉള്ളടക്കത്തിന് നൽകാനാണ് <BIG> ടാഗ് ഉപയോഗിക്കുന്നത്. ഡോക്യുമെന്റിലെ വാക്കുകൾക്ക് അല്ലെങ്കിൽ വരികൾക്ക് പ്രത്യേക പ്രാധാന്യം നൽകാനാണ് ഇത് ഉപയോഗിക്കുന്നത്. സാധാരണ ഗതിയിൽ <BIG> ടാഗിലെ അക്ഷരങ്ങളുടെ വലുപ്പം നിലവിലുള്ള വലുപ്പത്തേക്കാൾ ഒരുപടി കൂടുതലായിരിക്കും.

<SMALL>- ടെക്സ്റ്റ് ചെറുതാക്കുന്നു

ടെക്സ്റ്റുകളുടെ നിലവിലുള്ള വലുപ്പത്തേക്കാൾ ചെറുതാക്കാനാണ് <SMALL> ടാഗ് ഉപയോഗിക്കുന്നത്. സാധാരണയായി, <SMALL> ടാഗിലെ അക്ഷരങ്ങളുടെ വലുപ്പം, നിലവിലുള്ള അക്ഷരങ്ങളുടെ വലുപ്പത്തേക്കാൾ ഒന്ന് കുറവായിരിക്കും.

<STRONG> - ടെക്സ്റ്റ് കൂടുതൽ കനമുള്ളതാക്കുന്നു

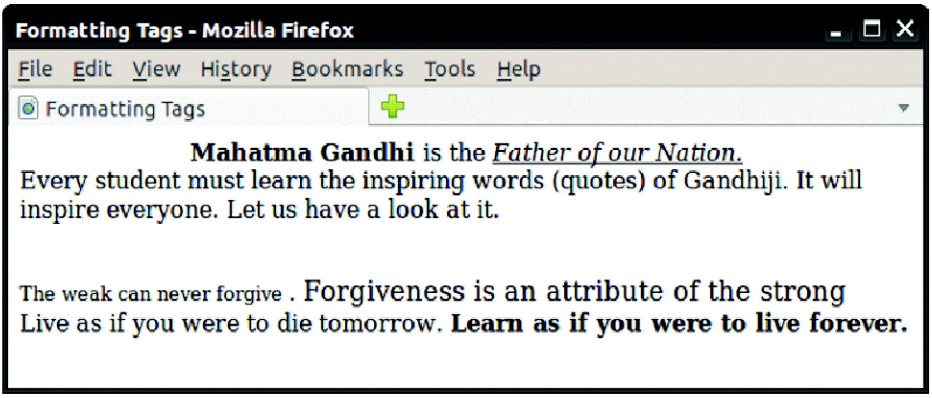
ഒരു ഉപവാക്യ ടാഗാണ് <STRONG> ഇത് പ്രാധാന്യമുള്ള വാക്യത്തെ നിർവചിക്കുന്നു. <STRONG> വാക്യം സാധാരണയായി കൂടുതൽ കനമുള്ളതാക്കിയ ബാഹ്യാകൃതിയോട്

കൂടിയാണ് ആവിഷ്കരിക്കുന്നത്. <B> ടാഗിനെപ്പോലെയാണിത്. വാക്യരൂപത്തിലുള്ള ഉള്ളടക്കത്തിലെ ഒരു ഉപവാക്യത്തിന് കൂടുതൽ പ്രാധാന്യം നൽകാനാണ് <STRONG> ടാഗ് ഉപയോഗിക്കുന്നത്.

ഉദാഹരണം 4.8-ൽ തന്നിട്ടുള്ള കോഡ് ഈ ടാഗുകളുടെ പ്രായോഗികത വിവരിക്കുന്നു. സന്ദർഭോചിതമായി, നമുക്ക് മഹാത്മാഗാന്ധിയുടെ വാക്കുകൾ ഉദ്ധരിക്കാം. ഈ കോഡ് ആവിഷ്കരിക്കുന്നതിന്റെ ഫലമായുണ്ടാകുന്ന വെബ് പേജ് ചിത്രം 4.24-ൽ കാണിച്ചിരിക്കുന്നു.

ഉദാഹരണം 4.8: വാക്യങ്ങൾ രൂപപ്പെടുത്തുന്ന ടാഗുകളെക്കുറിച്ച് വിവരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
   <TITLE> Formatting Tags </TITLE>
</HEAD>
<BODY>
   <P>
   <CENTER><B>Mahatma Gandhi </B>is the <I> <U> Father of
             our Nation.</U> </I> </CENTER>
   Every student must learn the inspiring words (quotes)
   of Gandhiji. It will inspire everyone. Let us have a
   look at it.
   </P><BR>
   <SMALL> The weak can never forgive </SMALL>.
   <BIG> Forgiveness is an attribute of the strong</BIG>
   <BR> Live as if you were to die tomorrow.
   <STRONG> Learn as if you were to live forever.</STRONG>
</BODY>
</HTML>
```



ചിത്രം 4.24 : വാക്യഘടന രൂപപ്പെടുത്തുന്ന ടാഗുകളെക്കുറിച്ചുള്ള വിവരണം

വാക്യങ്ങൾക്ക് ചില പ്രത്യേക രൂപം നൽകുന്ന ടാഗുകൾ നമുക്ക് പരിചയപ്പെടാം.

<EM> - വാക്യങ്ങൾക്ക് പ്രത്യേക പ്രാധാന്യം നൽകുന്നു

വാക്യങ്ങൾക്ക് പ്രത്യേക പ്രാധാന്യം നൽകാൻ ഉപയോഗിക്കുന്ന ടാഗാണ് <EM>. പ്രയോഗത്തിൽ വരുമ്പോൾ, ആ ഭാഗം സാധാരണ ഇറ്റാലിക്സിലായിരിക്കും ആവിഷ്കരിക്കുന്നത്. <I> ടാഗുപയോഗിക്കുമ്പോഴുള്ള ഗുണം തന്നെയാകും <EM> ടാഗും തരുന്നത്.

<SUB>, <SUP> എന്നീ ടാഗുകൾ - സബ് സ്ക്രിപ്റ്റും സൂപ്പർ സ്ക്രിപ്റ്റും നിർമ്മിക്കുന്നു

ജലം, സൽഫ്യൂരിക് അമ്ലം തുടങ്ങിയവയുടെ രാസനാമം വാക്യം ഹൈസ്കൂൾ ക്ലാസ്സുകളിൽ നാം പഠിച്ചിട്ടുള്ളതാണ്. അവ യഥാക്രമം H_2O, H_2SO_4 എന്നിങ്ങനെ ആയിരിക്കും. HTML-ൽ അത്തരം പ്രതീകങ്ങൾ നമുക്ക് എങ്ങനെ ചിത്രീകരിക്കാനാകും? അക്കങ്ങൾ സബ്സ്ക്രിപ്റ്റ് രൂപത്തിലാണ് എഴുതിയിട്ടുള്ളതെന്ന് നമുക്ക് കാണാം. ഒരു വെബ് പേജിൽ സബ്സ്ക്രിപ്റ്റ് നിർമിക്കാനാണ് <SUB> ടാഗുപയോഗിക്കുന്നത്. H₂O എന്ന കോഡുപയോഗിച്ച് കൊണ്ട് H_2O എന്ന വാക്യം നമുക്ക് പ്രദർശിപ്പിക്കാനാകും.

അതുപോലെ, $(a+b)^2 = a^2 + 2ab+ b^2$ പോലുള്ള ബീജഗണിത ആവിഷ്കാര രീതിയിലെ സൂപ്പർ സ്ക്രിപ്റ്റുകൾ <SUP> ടാഗ് ഉപയോഗിച്ച് നമുക്ക് സൂചിപ്പിക്കാനാകും. ഈ സൂത്രവാക്യം നമുക്ക് ഇങ്ങനെ എഴുതാം.

```
(a+b)<SUP>2</SUP> = a<SUP>2</SUP> + 2ab + b<SUP>2</SUP>
```

<BLOCKQUOTE>, <Q> എന്നീ ടാഗുകൾ - ഒരു ഉദ്ധരണി പ്രദർശിപ്പിക്കുന്നു

<BLOCKQUOTE> ടാഗ് അവയ്ക്കിടയിൽ ഉൾക്കൊള്ളിച്ചിട്ടുള്ള ഉള്ളടക്കം മാറ്റി പ്രദർശിപ്പിക്കാൻ ഉപയോഗിക്കുന്നു. വാക്യങ്ങളെ ഇരട്ട ഉദ്ധരണികൾക്കുള്ളിൽ മാറ്റി പ്രദർശിപ്പിക്കുന്നത് സൂചിപ്പിക്കാനാണ് <Q> ടാഗ് ഉപയോഗിക്കുന്നത്. ഖണ്ഡിക മുറിക്കാതെയുള്ള ചുരുങ്ങിയ ഉദ്ധരണികൾക്ക് വേണ്ടിയുള്ളതാണ് ഈ ടാഗ്. എന്നാൽ <BLOCKQUOTE> വലിയ ഉദ്ധരണികൾക്ക് ഉപയോഗിക്കാൻ വേണ്ടിയുള്ളതാണ്.

ഉദാഹരണം 4.9-ൽ മുകളിൽ സൂചിപ്പിച്ചിട്ടുള്ള ടാഗുകളുടെ വിശദീകരണമുണ്ട്. ലോക പരിസ്ഥിതി ദിനം ആഘോഷിക്കുന്നത് ജൂൺ 5-ന് ആണ് എന്നത് നമുക്ക് അനുസ്മരിക്കാം. ഈ ദിവസത്തേക്കുള്ള ചിന്തകൾ നാം സൃഷ്ടിക്കുകയും ഇത്തരം ടാഗുകളുപയോഗിച്ച് ഒരു വെബ് പേജ് നിർമിക്കുകയും ചെയ്യാം. ചിത്രം 4.25-ൽ അനുബന്ധമായ വെബ് പേജ് കാണിച്ചിട്ടുണ്ട്.

ഉദാഹരണം 4.9: <SUP>, <BLOCKQUOTE>, <Q> എന്നീ ടാഗുകൾ വിശദീകരിക്കുന്നതിന്

```
<HTML>
<HEAD>
   <TITLE> BlockQuote and Q tags  </TITLE>
```

```
</HEAD>
<BODY Bgcolor= "#98FB98" Text= "#008000">
   Every year we celebrate World Environment Day on 5<SUP>
   th</SUP> June. Let us have a message to all on this
   occassion.
   <BLOCKQUOTE> <B>June 5<SUP>th</SUP> is World Environment
     Day. </B>
     Mother nature too needs care and protection. Show her
     your care by caring for her trees. Love trees and love
     nature. And work for a greener environment because
     generations have to come... The future depends on us...
   </BLOCKQUOTE>
   <Q>Keep your world clean and green. Save trees, Save the
      environment!!
   </Q>
</BODY>
</HTML>
```

ആദ്യ ഖണ്ഡിക ആരംഭിക്കുന്നത് ബ്രൗസർ ജാലകത്തിന്റെ ഇടത് മൂലയിൽ നിന്നാണെന്ന് ചിത്രം 4.25-ൽ നിന്നും നമുക്ക് മനസിലാക്കാം. അതിൽ സൂപ്പർ സ്ക്രിപ്റ്റ് പ്രയോഗിച്ചിട്ടുണ്ട്. <BLOCKQUOTE> ടാഗ് ഉപയോഗിച്ചതിനാൽ രണ്ടാമത്തെ ഖണ്ഡിക മാറ്റി പ്രദർശിപ്പിച്ചു കൊണ്ടാണ് തുടങ്ങിയിരിക്കുന്നത്. അവസാനമായി, മൂന്നാമത്തെ ഖണ്ഡിക ഇരട്ട ഉദ്ധരണികൾക്ക് ഉള്ളിലാക്കിയിരിക്കുന്നു. അതിന്റെ ഉള്ളടക്കം <Q>, </Q> ടാഗ് ജോഡികൾക്ക് ഉള്ളിൽ ഉൾപ്പെടുത്തിയിരിക്കുന്നു. ഈ ഖണ്ഡികകളുടെ വിന്യാസം കൂടി നിരീക്ഷിക്കുക. <BLOCK QUOTE> ടാഗിൽ ഒന്നിലധികം ഖണ്ഡികകൾ ഉൾപ്പെടുത്തിയിരിക്കുന്നത് ശ്രദ്ധിക്കുക.

ചിത്രം 4.25 : ഉദാഹരണം 4.9 ലെ കോഡിന് അനുബന്ധമായ വെബ് പേജ്

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. വാക്യങ്ങൾക്ക് ഘടന നൽകുന്ന ചില ടാഗുകളുടെ പേരെഴുതുക.
2. <HR> ടാഗിന്റെ വ്യത്യസ്ത ആട്രിബ്യൂട്ടുകളുടെ നാമാവലി തയാറാക്കുക.
3. എത്ര തലങ്ങളിലുള്ള ശീർഷക ടാഗുകൾ HTML - ൽ ലഭ്യമാണ്?
4. x^3+y^3 എന്ന് പ്രദർശിപ്പിക്കുന്നതിനുള്ള HTML കോഡ് ശകലം എഴുതുക.
5. ശരിയോ തെറ്റോ എന്ന് പ്രസ്താവിക്കുക.
   a.
 ടാഗ് ഒരു എംപ്റ്റി ടാഗാണ്.
   b. HTML ഡോക്യുമെന്റിൽ <EM> ടാഗിനും <I> ടാഗിനും ഒരേ ഉപയോഗമാണുള്ളത്.
   c. <U> ടാഗിന്റെയും <I> ടാഗിന്റെയും ഒരുമിച്ചുള്ള ഉപയോഗം അനുവദനീയമല്ല.
6. <STRONG> ടാഗിന്റെ ഉപയോഗമെന്താണ്?
7. <STRONG> ടാഗിന്റേതിന് സമാനമായ കൃത്യം നിർവഹിക്കുന്ന ടാഗ് ഏതാണ്?
8. താഴെതന്നിട്ടുള്ളവയിൽ നിന്നും ഒറ്റപ്പെട്ടത് പുറത്തെടുക്കുക.
   a. HTML   b. ALIGN   c. HEAD   d. CENTER

4.10.7 <PRE>-പൂർവ രൂപീകൃത ടെക്സ്റ്റുകൾ പ്രദർശിപ്പിക്കുന്നു (Displaying Preformatted text)

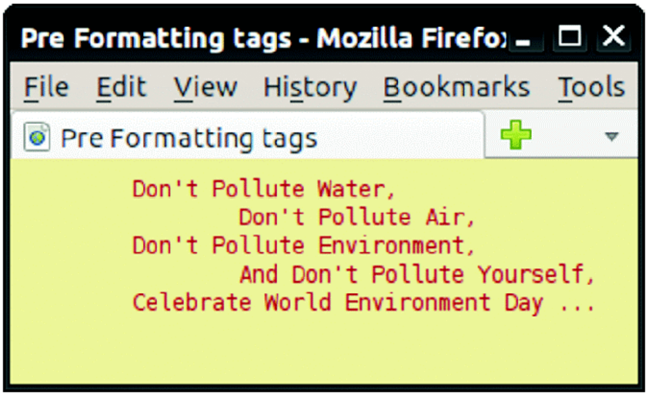
ടെക്സ്റ്റ് എഡിറ്ററിൽ നൽകിയ ഉള്ളടക്കം നമുക്ക് അതേ രൂപത്തിൽ തന്നെ പ്രദർശിപ്പിക്കണം എന്നിരിക്കട്ടെ. ഈ ആവശ്യം **<PRE>** ടാഗിന് നിർവഹിക്കാനാകും. സാധാരണഗതിയിൽ വൈറ്റ് സ്പേസുകൾ, ന്യൂ ലൈൻ ക്യാരക്ടറുകൾ, ടാബ് സ്പെയ്സുകൾ തുടങ്ങിയവയ്ക്ക് ബ്രൗസർ പരിധി നിശ്ചയിച്ചിട്ടുണ്ട്. അതുകൊണ്ട്, ബ്രൗസറിന്റെ സ്വയം പ്രേരിത ഘടനാരൂപീകരണ സ്വഭാവം <PRE> ടാഗുപയോഗിച്ച് നമുക്ക് പ്രവർത്തന രഹിതമാക്കാനാകും. ഈ ടാഗ്, അതിലുൾപ്പെടുത്തിയിട്ടുള്ള വാക്യത്തിന്റെ ഘടന രൂപപ്പെടുത്തിക്കഴിഞ്ഞതാണ് എന്ന് ബ്രൗസറിനോട് പറയുകയും, ആയതിനാൽ വീണ്ടും ഘടന പുനർനിർണയിക്കേണ്ടതില്ല എന്ന് ഓർമിപ്പിക്കുകയും ചെയ്യുന്നു. അതായത്, വാക്യം അതേപടി തന്നെ പ്രദർശിപ്പിക്കുവാൻ ബ്രൗസറിനോട് പറയുന്നു.

ഉദാഹരണം 4.10, ചിത്രം 4.26 എന്നിവ ഈ ടാഗിനെക്കുറിച്ച് ഒരു ധാരണ നമുക്ക് നൽകുന്നു. ലോക പരിസ്ഥിതിദിനത്തോടനുബന്ധിച്ച് കുറച്ച് വാക്യങ്ങളുള്ള ഒരു വെബ് പേജ് നമുക്ക് നിർമിക്കാം.

ഉദാഹരണം 4.10: <PRE> ടാഗ് വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
   <TITLE> Pre Formatting tags </TITLE>
</HEAD>
<BODY Bgcolor = "#eee8aa" Text = "#b22222">
   <PRE>
     Don't Pollute Water,
            Don't Pollute Air,
     Don't Pollute Environment,
            And Don't Pollute Yourself,
     Celebrate World Environment Day ...
   </PRE>
</BODY>
</HTML>
```

<PRE>, </PRE> എന്നീ ടാഗുകൾക്ക് ഇടയിലുള്ളതെന്തും HTML ലിഖിതത്തിലുള്ളത് പോലെ തന്നെ പ്രദർശിപ്പിക്കും എന്ന് ചിത്രം 4.26-ൽ കാണിച്ചിട്ടുള്ള വെബ് പേജിൽ നിന്നും മനസിലാക്കാം.



ചിത്രം 4.26 : <PRE> ടാഗിനെക്കുറിച്ചുള്ള വിശദീകരണം

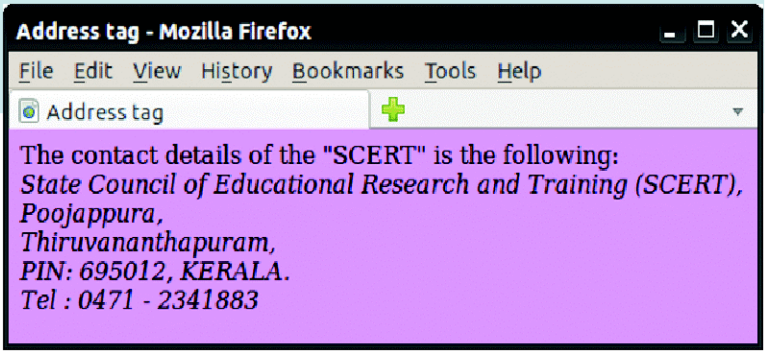
4.10.8 <ADDRESS> - മേൽവിലാസം പ്രദർശിപ്പിക്കുന്നു (Displaying the address)

ഒരു പ്രമാണത്തിന്റെ അല്ലെങ്കിൽ പ്രബന്ധത്തിന്റെ രചയിതാവിനെ/ഉടമസ്ഥനെ ബന്ധപ്പെടാനുള്ള വിവരം പോലുള്ളവ നിർവചിക്കുവാനുള്ള ടാഗാണ് **<ADDRESS>** ടാഗ്. പേര്, ടെലഫോൺ നമ്പർ, പിൻ നമ്പർ, ഇ-മെയിൽ മേൽവിലാസങ്ങൾ തുടങ്ങിയവയാണ് ഈ ടാഗിന്റെ ഉള്ളടക്കമായി ഉൾക്കൊള്ളിക്കുന്നത്. വാക്യങ്ങളെ ഇറ്റാലിക്സ് രീതിയിലാണ് മിക്കവാറും ബ്രൗസറുകൾ പ്രദർശിപ്പിക്കുന്നത്.

ഉദാഹരണം 4.11-ൽ തന്നിട്ടുള്ള കോഡ് <ADDRESS> ടാഗിനെക്കുറിച്ച് വിശദീകരിക്കുന്നു. പേജിന്റെ രൂപം ചിത്രം 4.27-ൽ കാണിച്ചിരിക്കുന്നു.

ഉദാഹരണം 4.11: <ADDRESS> ടാഗ് വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
   <TITLE> Address tag </TITLE>
</HEAD>
   <BODY Bgcolor= "#DDA0DD">
     The contact details of the "SCERT" is the following:
   <ADDRESS>
     State Council of Educational Research and Training
     (SCERT),<BR>
     Poojappura,<BR>
     Thiruvananthapuram,<BR>
     PIN: 695012, KERALA.<BR>
     Tel : 0471 - 2341883
   </ADDRESS>
</BODY>
</HTML>
```



ചിത്രം 4.27 : <ADDRESS> ടാഗ് ഉൾപ്പെടുത്തിയിട്ടുള്ള ഒരു വെബ് പേജ്

ബന്ധപ്പെടാനുള്ള വിവരത്തിന്റെ ഭാഗമെന്ന വരുമ്പോൾ, <ADDRESS> ടാഗ് സാധാരണ ഗതിയിൽ തപാൽ മേൽവിലാസം സൂചിപ്പിക്കാനാണ് ഉപയോഗിക്കുന്നത്. എന്നിരുന്നാലും, തനത് ശൈലിയിൽ <EM>, <I> എന്നിവ പ്രദർശിപ്പിക്കുന്ന ശൈലിക്ക് സമാനമായാണ് <ADDRESS> ടാഗും വാക്യം പ്രദർശിപ്പിക്കുന്നത്. ആയതിനാൽ, ബന്ധപ്പെടാനുള്ള വിവരം കൈകാര്യം ചെയ്യാൻ അനുയോജ്യമായ രീതിയിൽ അത് ഉപയോഗിക്കാവുന്നതാണ്. എഴുത്തുകാരന്റെ പേര്, പകർപ്പവകാശം തുടങ്ങിയ വിവരങ്ങൾ ഉൾക്കൊള്ളുന്ന അടിക്കുറിപ്പിനുള്ളിലാണ് <ADDRESS> ടാഗ് എലമെന്റ് മാതൃകാപരമായി നൽകാറുള്ളത്.

4.10.9 <MARQUEE> മാർക്വ്യൂ വാക്യം പ്രദർശിപ്പിക്കുന്നു (Displaying text in a scrolling Marquee)

നാം ഇതുവരെ ചർച്ച ചെയ്ത HTML ടാഗുകളെല്ലാം ഉള്ളടക്കങ്ങൾ ബ്രൗസറിൽ പ്രദർശിപ്പിക്കുക മാത്രമാണ് ചെയ്യുന്നത്. എന്നാൽ ഒരു വാക്യശകലം അല്ലെങ്കിൽ ചിത്രം വെബ് പേജിൽ തിരശ്ചീനമായോ ലംബമായോ ഒഴുകി നീങ്ങുന്നതായോ കാണിക്കുന്നതിന്, **<MARQUEE>** ടാഗ് ഉപയോഗിക്കുന്നു.

<MARQUEE> ടാഗിനൊപ്പം ഉപയോഗിക്കുന്ന പ്രധാനപ്പെട്ട ആട്രിബ്യൂട്ടുകളുടെ നാമാവലി താഴെ തന്നിട്ടുണ്ട്.

- **Height**: പിക്സലായോ ബ്രൗസർ ജാലകത്തിലെ ഉയരത്തിന്റെ ശതമാനമായോ മാർക്യൂവിന്റെ ഉയരം നിശ്ചയിക്കുന്നു.
- **Width**: പിക്സലായോ, ബ്രൗസർ ജാലകത്തിലെ വീതിയുടെ ശതമാനമായോ വീതി സൂചിപ്പിക്കുന്നു.
- **Direction**: ഏത് ദിശയിലാണ് ടെക്സ്റ്റ് ചലിക്കേണ്ടതെന്ന് സൂചിപ്പിക്കുന്നു. up, down, left, right എന്നീ വിലകളാണ് ഇതിനുള്ളത്.
- **Behaviour**: മാർക്യൂവിന്റെ സ്ക്രോളിങ് ഏത് വിധത്തിലാകണമെന്ന് സൂചിപ്പിക്കുന്നു. scroll, slide, alternate എന്നീ വിലകളാണ് ഇതിനുള്ളത്.
- **Scrolldelay**: ഓരോ സ്ക്രോളിനും ഇടയ്ക്കുള്ള സമയദൈർഘ്യം സൂചിപ്പിക്കുന്നു. 10, 15 തുടങ്ങിയവ പോലെ സെക്കന്റിലുള്ള വിലകളാണ് ഇതിനുള്ളത്.
- **Scrollamount**: മാർക്യു ടെക്സ്റ്റിന്റെ വേഗത സൂചിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്നു.
- **Loop**: മാർക്യൂ ടെക്സ്റ്റ് എത്രപ്രാവശ്യം സ്ക്രീനിൽ സ്ക്രോൾ ചെയ്യണമെന്ന് സൂചിപ്പിക്കുന്നു. ഇതിന്റെ തനത് വില Infinte ആയിരിക്കും, അങ്ങനെ ആയാൽ മാർക്യൂ അവസാനമില്ലാതെ സ്ക്രോൾ ആകും.
- **Bgcolor**: നിറമോ, നിറത്തെ സൂചിപ്പിക്കുന്ന ഹെക്സാഡെസിമൽ സംഖ്യയോ ഉപയോഗിച്ച് പശ്ചാത്തലനിറം സൂചിപ്പിക്കുന്നു.
- **Hspace**: ഇത് മാർക്യൂവിന് ചുറ്റുമുള്ള തിരശ്ചീനമായ സ്ഥലം സൂചിപ്പിക്കുന്നു. ഇതിന്റെ വില പിക്സൽ ആയോ ശതമാനവില ആയോ നൽകുന്നു.
- **Vspace**: മാർക്യൂവിനു ചുറ്റുമുള്ള ലംബമായ ശൂന്യസ്ഥലത്തെ സൂചിപ്പിക്കുന്നു. ഇതിന്റെ വിലയും പിക്സൽ ആയോ ശതമാനവില ആയോ ആണ് നൽകുന്നത്.

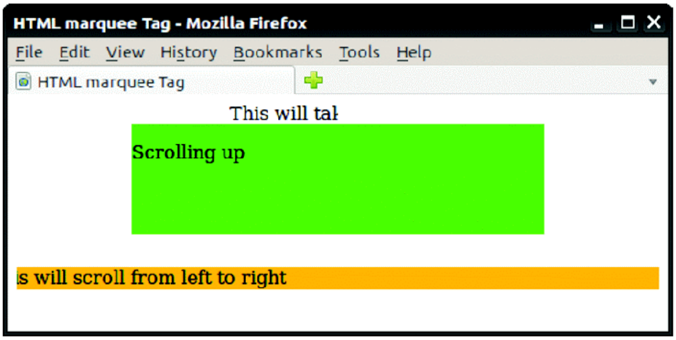
ഉദാഹരണം 4.12-ൽ തന്നിട്ടുള്ള കോഡ് <MARQUEE> ടാഗിന്റെ ഉപയോഗത്തേയും ചിത്രം 4.28 അനുബന്ധ വെബ് പേജിനേയും കാണിക്കുന്നു.

ഉദാഹരണം 4.12: <MARQUEE> ടാഗ് വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
   <TITLE> HTML marquee Tag </TITLE>
</HEAD>
<BODY>
   <MARQUEE Width= "50%"> This  will take only 50% width of
     Browser Window</MARQUEE>
   <MARQUEE Height= "100" Hspace= "100" Bgcolor= "#44BB22"
          Direction= "up"> Scrolling up </MARQUEE>
   <MARQUEE Height= "20" Vspace= "30" Bgcolor= "#FFBB00"
     Direction= "right"> This will scroll from left to right
   </MARQUEE>
</BODY>
</HTML>
```

വീതി 50% ആയതിനാൽ ആദ്യത്തെ മാർക്യൂ മധ്യഭാഗത്ത് നിന്നും തുടങ്ങി വിൻഡോയുടെ ഇടത് വശത്തേക്ക് സ്ക്രോൾ ചെയ്യുന്നു. രണ്ടാമത്തെ മാർക്യൂ പച്ചനിറത്തോടുകൂടിയതും 100 പിക്സൽ ഉയരം സ്ക്രോൾ ചെയ്യാൻ ഉപയോഗിച്ചിട്ടുള്ളതുമാണ്.

ഇടത് മാർജിനിൽ നിന്നും തിരശ്ചീനമായി 100 പിക്സൽ അകലെയാണ് സ്ക്രോൾ മേഖല ഏരിയ സജ്ജീകരിച്ചിട്ടുള്ളത്. മൂന്നാമത്തെ മാർക്യൂ "#FFBB00" നിറം കൊണ്ട് പശ്ചാത്തലം ഒരുക്കിയിട്ടുള്ളതും മുന്നിലുള്ള മാർക്യൂവിൽ നിന്നും ലംബമായി 30 പിക്സൽ താഴേക്ക് സ്ഥാപിച്ചിട്ടുള്ളതുമാണ്. സ്ക്രോൾ ചെയ്യുന്ന ജാലകം 40 പിക്സൽ ഉയരത്തിലുള്ളതും ഇടത്ത് നിന്ന് വലത്തേക്ക് നീങ്ങുന്നതുമാണ്. വാക്കുകളോ, വാക്യങ്ങളോ മാത്രമല്ല ചിത്രങ്ങളും മാർക്യൂവിന്റെ ഉള്ളടക്കമായി നമുക്ക് ഉപയോഗിക്കാം.



ചിത്രം 4.28 : Marquee ടാഗും അതിന്റെ ആട്രിബ്യൂട്ടുകളും

4.10.10 <DIV> - വാക്യങ്ങളുടെ ഒരു ഭാഗം ചിട്ടപ്പെടുത്തുന്നു (Formatting a block of text)

ഡോക്യുമെന്റിന്റെ ഒരു ഭാഗം അല്ലെങ്കിൽ ഒരു നിര ചിട്ടപ്പെടുത്താനാണ് <DIV> ടാഗ് ഉപയോഗിക്കുന്നത്. HTML ലിഖിതത്തിലുള്ള വലിയ ഭാഗങ്ങൾ ഒരുമിച്ച് ഒരു കൂട്ടമാക്കി ചിട്ടപ്പെടുത്താനും **<DIV>** ടാഗ് കൊണ്ട് സാധിക്കുന്നു. ഖണ്ഡികകൾ, പട്ടികകൾ തുടങ്ങിയവ ഈ ഭാഗത്തുണ്ടാകും. <DIV> ടാഗിന് മുൻപും പിൻപും മിക്കവാറും ബ്രൗസറുകൾ ഒരു വരി (ശൂന്യവരി) ചേർത്തിട്ടുണ്ടാകും. <DIV> ടാഗിന്റെ ആട്രിബ്യൂട്ടുകൾ താഴെ തന്നിട്ടുള്ളവയാണ്.

Align : left, right, center, justify എന്നീ വിലകൾ കൊണ്ട് തിരശ്ചീനമായ വിന്യാസം നിജപ്പെടുത്തുന്നു.

Id : ടാഗിനെ തിരിച്ചറിയാനായി സവിശേഷമായ ഒരു സൂചകം നൽകുന്നു.

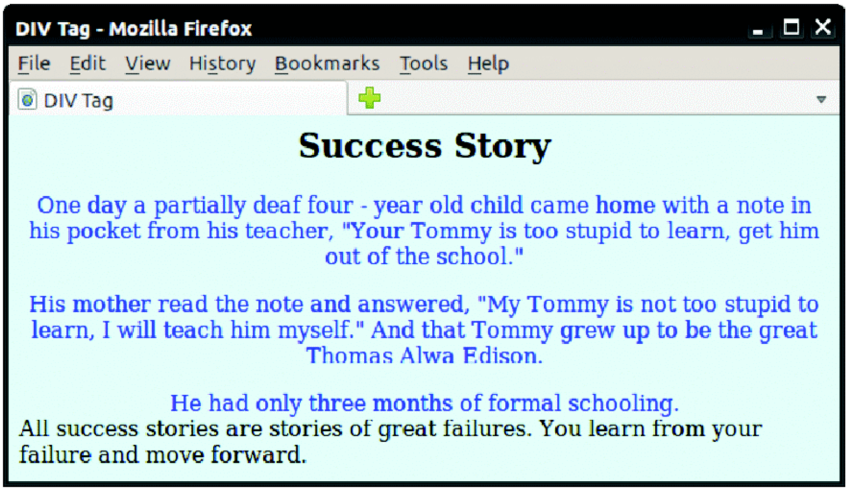
Style : ഉള്ളടക്കത്തിലെ അക്ഷരം, നിറം തുടങ്ങിയവ എങ്ങനെ ആവിഷ്കരിക്കണമെന്ന് നിർദേശിക്കുന്നു.

<DIV> ടാഗിന്റെ Align, Style എന്നീ ആട്രിബ്യൂട്ടുകളുടെ ഉപയോഗം ഉദാഹരണം 4.13-ലും, തൽഫലമായി ലഭിക്കുന്ന വെബ്പേജ് ചിത്രം 4.29-ലും കാണാവുന്നതാണ്.

ഉദാഹരണം 4.13: <DIV> ടാഗ് വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
   <TITLE> DIV Tag </TITLE>
</HEAD >
```

```
<BODY Bgcolor= "#ddffff">
   <H2 Align= "center"> Success Story </H2>
   <DIV Align= "Center" Style= "Color:#0000FF"> One day a
   partially deaf four - year old child came home with a
   note in his pocket from his teacher, "Your Tommy is too
   stupid to learn, get him out of the school."
   <P>His mother read the note and answered, "My Tommy is
      not too stupid to learn, I will teach him myself." And
      that Tommy grew up to be the great Thomas Alwa Edison.
   </P> He had only three months of formal schooling.
   </DIV>
   All success stories are stories of great failures.
   You learn from your failure and move forward.
</BODY>
</HTML>
```



ചിത്രം 4.29 : <DIV> ടാഗിന്റെ ഉപയോഗം

4.10.11 <FONT> ലിപികളുടെ സ്വഭാവസവിശേഷതകൾ നിർദേശിക്കുന്നു (Specifying font characteristics)

<FONT>, </FONT> എന്നീ ടാഗുകൾക്ക് ഇടയിൽ ഉൾക്കൊള്ളിച്ചിട്ടുള്ള വാക്യങ്ങളുടെ ശൈലി, വലുപ്പം, നിറം എന്നിവ വ്യത്യാസപ്പെടുത്തുവാൻ **<FONT>** ടാഗ് നമ്മെ സഹായിക്കുന്നു. ഡോക്യുമെന്റിലെ ഒരു ചെറിയ ഭാഗത്തിന്റെ രൂപഘടന മാറ്റുന്നതിനാണ് ഇത് സാധാരണയായി ഉപയോഗിക്കുന്നത്. <FONT> ടാഗിന്റെ ആട്രിബ്യൂട്ടുകൾ താഴെ തന്നിട്ടുള്ളവയാണ്.

Color : നിറനാമമായോ, ഹെക്സാഡെസിമൽ വിലയായോ ടെക്സ്റ്റിന് നിറം നൽകുന്നു.

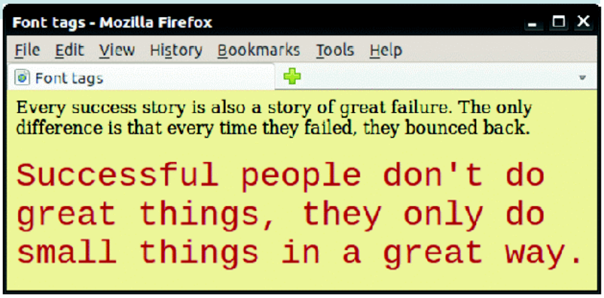
Face : അക്ഷരങ്ങളുടെ ശൈലി സൂചിപ്പിക്കുന്ന ആട്രിബ്യൂട്ടാണ് ഇത്. വില സൂചിപ്പിച്ചിട്ടില്ല എങ്കിൽ ബ്രൗസർ അനുകൂലിക്കുന്ന ശൈലികളിൽ ആദ്യത്തേത് തനത് (Default) ശൈലിയായി ഡോക്യുമെന്റിലെ വാക്യങ്ങൾക്ക് നൽകുന്നു.

Size : അക്ഷരങ്ങളുടെ വലുപ്പം 1 മുതൽ 7 വരെയുള്ള വിലകൾ കൊണ്ട് സൂചിപ്പിക്കാൻ ഉപയോഗിക്കുന്ന ആട്രിബ്യൂട്ടാണ് ഇത്. തനത് വില 3 ആയിരിക്കും.

<FONT> ടാഗിന്റെ ഉപയോഗം ഉദാഹരണം 4.14-ൽ തന്നിട്ടുള്ള കോഡ് വിശദീകരിക്കുകയും അനുബന്ധ വെബ് പേജ് ചിത്രം 4.30 കാണിക്കുകയും ചെയ്യുന്നു.

ഉദാഹരണം 4.14: <FONT> ടാഗ് വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD> <TITLE> Font tags </TITLE> </HEAD>
<BODY Bgcolor= "#eee8aa">
   Every success story is also a story of great failure.
   The only difference is that every time they failed,
   they bounced back. <BR><BR>
   <FONT Size="6" Face="Courier New" Color="#B22222">
   Successful people don&apos;t do great things,
   they only do small things in a great way.
   </FONT>
</BODY>
</HTML>
```



ചിത്രം 4.30 : <FONT> ടാഗിന്റെ ഉപയോഗം

മുകളിൽ തന്നിട്ടുള്ള കോഡിൽ ഒരു ഭാഗം (') നാം പച്ചനിറത്തിൽ രേഖപ്പെടുത്തിയിട്ടുണ്ട്. ഇത് അർഥമാക്കുന്നത് എന്താണ്? **don't** എന്ന വാക്കിലെ ഏക ഉദ്ധരണി ചിഹ്നം കൊണ്ട് വരുന്നതിന് വേണ്ടിയാണിതെന്ന് ലഭിക്കുന്ന വെബ് പേജിൽ നിന്നും നമുക്ക് മനസിലാക്കാനാകും. ഇതുപോലെ മറ്റ് പല ചിഹ്നങ്ങളുമുണ്ട്. തുടർന്ന് വരുന്ന ഭാഗങ്ങളിൽ അവ ചർച്ച ചെയ്യാം.

4.11 സാർഥക ചിഹ്നങ്ങൾക്ക് വേണ്ടി HTML – ൽ ഉപയോഗിക്കുന്ന സവിശേഷ വസ്തുക്കൾ (HTML entities for reserved characters)

<, >, & തുടങ്ങിയ ചിഹ്നങ്ങൾക്ക് പ്രത്യേകമായ അർത്ഥമാണ് HTML-ൽ ഉള്ളത്. മാത്രമല്ല, HTML ഡോക്യുമെന്റിന്റെ ഉള്ളടക്കത്തിലെ വാക്യങ്ങളിൽ ഇവ ഉൾപ്പെടുത്താനും സാധിക്കില്ല. HTML ടാഗുകളോടും ആട്രിബ്യൂട്ടുകളോടും ഒപ്പം കാണപ്പെടുന്ന വിരാമ ചിഹ്നങ്ങളെപ്പോലെ ഈ ചിഹ്നങ്ങളെ ബ്രൗസർ കണക്കാക്കും.

ഉദാഹരണമായി, ആങ്കിൾ ബ്രാക്കറ്റുകളായ <, > എന്നിവ ഉപയോഗിക്കുന്നത് ടാഗുകളെ സൂചിപ്പിക്കാനാണ്. വെബ് പേജിലെ വാക്യങ്ങളിൽ ഈ അടയാളങ്ങൾ നമുക്ക് എപ്പോഴെങ്കിലും പ്രദർശിപ്പിക്കേണ്ടി വന്നാൽ, HTML-ലെ സവിശേഷ വസ്തുക്കൾ നമുക്ക് ഉപയോഗിക്കേണ്ടിവരും. പട്ടിക 4.7-ൽ സാർത്ഥക ചിഹ്നങ്ങളുടെയും അനുബന്ധ HTML വസ്തുക്കളുടെയും നാമാവലി നൽകിയിരിക്കുന്നു.

ഒരു വെബ് പേജിൽ A < B & A > C എന്ന വാക്യം പ്രദർശിപ്പിക്കുന്നതിന്, **A < B & A > C** എന്ന് HTML കോഡിൽ നൽകണം.

Character	Entity	Description
		Non Breaking Space
"	"	Double quotation mark
'	'	Single quotation mark
&	&	Ampersand
<	<	Less than
>	>	Greater than
©	©	Copyright Symbol
™	™	Trademark Symbol
®	®	Registered Symbol

പട്ടിക 4.6 : എന്റിറ്റികളുടെ പട്ടികയും അവയുടെ വിശദീകരണവും

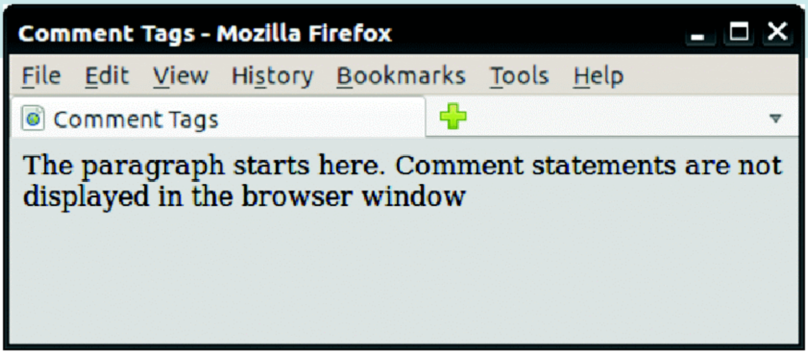
4.12 HTML ഡോക്യുമെന്റിൽ കമന്റുകൾ ഉൾപ്പെടുത്തുന്നു (Adding comments in HTML document)

HTML ഡോക്യുമെന്റിൽ, പ്രത്യേകിച്ച് സങ്കീർണമായ ഡോക്യുമെന്റിൽ കമന്റ് ഉൾപ്പെടുത്തുന്നത് ഒരു നല്ല ശീലമാണ്. കോഡ് മനസ്സിലാക്കാൻ കമന്റുകൾ നമ്മെ സഹായിക്കുന്നു. കൂടാതെ കോഡിന്റെ വായനക്ഷമതയും അത് വർധിപ്പിക്കുന്നു. സോഴ്സ് കോഡിൽ വിവരണങ്ങൾ ഉൾപ്പെടുത്താൻ HTML-ൽ കമന്റ് ടാഗ്കളുണ്ട്. വിവരണങ്ങൾ ബ്രൗസർ ജാലകത്തിൽ കാണിക്കുകയില്ല. HTML വിവരണങ്ങൾ <!-- , --> എന്നീ ടാഗുകൾക്ക് ഇടയ്ക്കാണ് നൽകാറുള്ളത്. അപ്രകാരം നൽകുന്ന ഏത് ഉള്ളടക്കവും കമന്റായി കണക്കാക്കുകയും ബ്രൗസർ അത് പൂർണമായും അവഗണിക്കുകയും ചെയ്യുന്നു. ജിനി എഡിറ്ററിൽ കമന്റുകൾ ചുവപ്പ് നിറത്തിലാണ് പ്രദർശിപ്പിക്കാറുള്ളത്. ഉദാഹരണം 4.15-ലെ കോഡ് കമന്റുകളുടെ ഉപയോഗത്തെക്കുറിച്ച് വിശദീകരിക്കുകയും ചിത്രം 4.31-ലെ വെബ് പേജ് അങ്ങനെ ലഭിക്കുന്ന സന്ദേശം കാണിക്കുകയും ചെയ്യുന്നു.

ഉദാഹരണം 4.15: കമന്റ് ടാഗിന്റെ ഉപയോഗം വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD> <!-- Document Header Starts -->
   <TITLE> Comment Tags </TITLE>
</HEAD>
<BODY Bgcolor= "#D8D8D8">
   <!-- This is a comment -->
```

```
    <p>The paragraph starts here. Comment statements are
    not displayed in the browser window</p>
    <!-- Comments are not displayed in the browser -->
</BODY>
</HTML>
```



ചിത്രം 4.31 : HTML കമന്റുകൾ

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. സാർത്ഥക ചിഹ്നങ്ങൾ എങ്ങനെയാണ് HTML-ൽ സൂചിപ്പിക്കുന്നത്?
2. ________ ടാഗിനോട് ഒപ്പമാണ് Face ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്.
3. <FONT> ടാഗിന്റെ ആട്രിബ്യൂട്ടുകളുടെ നാമാവലി തയാറാക്കുക.
4. <PRE> ടാഗിന്റെ ഉപയോഗമെന്താണ്?
5. ഒരു ടെക്സ്റ്റ് സ്ക്രോൾ ചെയ്യാനായി ________ ടാഗ് ഉപയോഗിക്കുന്നു.
6. <MARQUEE> ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്രിബ്യൂട്ടുകൾ ഏതെല്ലാമാണ്?
7. <ADDRESS> ടാഗിന്റെ ഉപയോഗമെന്താണ്?
8. <FONT> ടാഗിലെ അക്ഷരങ്ങളുടെ തനത് വലുപ്പം എത്രയായിരിക്കും?
9. <DIV>ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്രിബ്യൂട്ടുകളുടെ പേരെഴുതുക.

4.13 ചിത്രങ്ങൾ ഉൾക്കൊള്ളിക്കുന്നു (Inserting images)

ഉള്ളടക്ക വിവരണം കൂടുതൽ ആകർഷകവും വിനിമയാത്മകവും ആക്കുന്നത് ചിത്രങ്ങളാണ്. ഇക്കാലത്ത് വെബ്സൈറ്റുകൾ അധികവും ചിത്രങ്ങളാൽ സമ്പന്നമാണ്. വെബ് വികസിപ്പിക്കുന്നതിന് വേണ്ട ധാരാളം സവിശേഷതകളുള്ള പുതിയ HTML പതിപ്പുകളിൽ പോലും ചിത്രങ്ങൾ ഉൾപ്പെടുത്തുന്നതിനുള്ള കോഡ് വളരെ ലളിതമാണ്. HTML പേജുകളിൽ ചിത്രങ്ങൾ ഉൾക്കൊള്ളിക്കുന്നതിന് **<IMG>** ടാഗ് ഉപയോഗിക്കുന്നു. വളരെ ലളിതമായി ഈ ടാഗ് ഉപയോഗിക്കുന്ന വിധം താഴെ കൊടുത്തിരിക്കുന്നു.

```
<IMG Src = "picture1.jpg">
```

<IMG> ടാഗ് ഒരു എംപ്റ്റി ടാഗാണ്. മാത്രമല്ല, അതിന് ധാരാളം ആട്രിബ്യൂട്ടുകളും ഉണ്ട്. പേജിൽ ഉൾക്കൊള്ളിക്കേണ്ട ചിത്രത്തിന്റെ ഫയൽനാമം സൂചിപ്പിക്കുന്ന Src ആട്രിബ്യൂട്ടാണ് ഏറ്റവും പ്രധാനപ്പെട്ടത്. നമ്മുടെ ആവശ്യത്തിന് അനുസരിച്ച് JPEG, PNG, GIF എന്നീ ചിത്രഫയലുകൾ ഉപയോഗിക്കാനാകുമെങ്കിലും, ശരിയായ ഫയൽ നാമവും അതോടൊപ്പമുള്ള എക്സ്റ്റൻഷനും Src ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് സൂചിപ്പിച്ചിട്ടുണ്ടെന്ന് ഉറപ്പ് വരുത്തണം. ചിത്രഫയൽ ഇപ്പോൾ ഉപയോഗത്തിലുള്ള ഫോൾഡറിൽ ഇല്ലെങ്കിൽ, അത് എവിടെയാണോ ലഭ്യമാകുന്നത് അവിടേക്കുള്ള വഴി (പാത്ത്)/URL നാം സൂചിപ്പിക്കണം.

ചിത്രത്തിന് വേണ്ടി സ്ഥലം ഒരുക്കുന്നു (Setting space for the image)

Width, **Height** എന്നീ ആട്രിബ്യൂട്ടുകൾക്ക് വിലകൾ നൽകിക്കൊണ്ട് വെബ് പേജിൽ ചിത്രത്തിനുവേണ്ടിയുള്ള ഇടം നമുക്ക് ഒരുക്കാം. പിക്സലിന്റെ എണ്ണമായോ അല്ലെങ്കിൽ യഥാർഥ വലുപ്പത്തിന്റെ ശതമാനമായോ വിലകൾ നൽകാവുന്നതാണ്. ഈ ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിച്ചിട്ടില്ലെങ്കിൽ ചിത്രത്തിന്റെ യഥാർഥ വലുപ്പത്തിൽ ബ്രൗസർ അത് പ്രദർശിപ്പിക്കും.

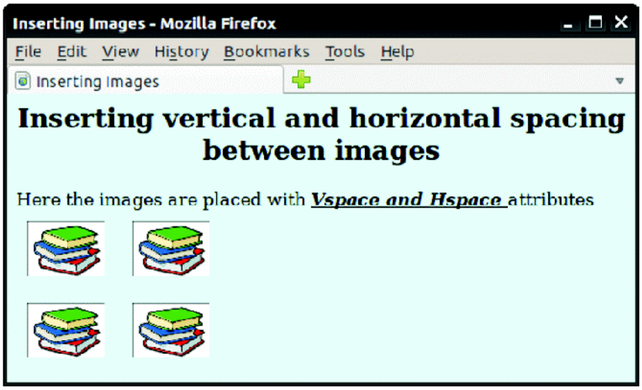
ചിത്രങ്ങൾക്ക് ഇടയ്ക്കുള്ള ശൂന്യസ്ഥലങ്ങൾ എങ്ങനെ ക്രമപ്പെടുത്താമെന്ന് നമുക്ക് ചർച്ച ചെയ്യാം. ഒരു ജാലകത്തിൽ ചിത്രങ്ങൾ സ്ഥാപിക്കുമ്പോൾ രണ്ട് രീതിയിലാണ് ശൂന്യസ്ഥലങ്ങൾ അവയ്ക്കിടയിൽ വരാറുള്ളത്-തിരശ്ചീനമായും, ലംബമായും. വെബ് പേജിൽ തിരശ്ചീനമായും ലംബമായും ചിത്രങ്ങൾക്കിടയ്ക്ക് സ്ഥലം നൽകുന്നതിന് HTML-ൽ ലഭ്യമാകുന്ന രണ്ട് ആട്രിബ്യൂട്ടുകളാണ് **Hspace**, **Vspace** എന്നിവ.

രണ്ട് രീതിയിൽ ചിത്രങ്ങൾക്ക് ഇടയ്ക്ക് ശൂന്യസ്ഥലം നൽകുന്നത് ഉദാഹരണം 4.16-ൽ തന്നിട്ടുള്ള HTML കോഡ് വിശദീകരിക്കുന്നു. അനുബന്ധ വെബ് പേജ് ചിത്രം 4.31-ൽ കാണിച്ചിരിക്കുന്നു.

ഉദാഹരണം 4.16: ചിത്രങ്ങൾക്കിടയിൽ പലതരത്തിലുള്ള ശൂന്യസ്ഥലം നൽകുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
   <TITLE> Inserting Images </TITLE>
</HEAD>
<BODY Bgcolor="#E0FFFF">
   <H2 Align="center">Inserting vertical and horizontal
                  spacing between images</H2>
   Here the images are placed with <B><I><U> Vspace and
   Hspace </U></I></B> attributes <BR>
   <IMG Src= "book3.jpg" Height= "50" Width= "70"
```

```
        Vspace= "10" Hspace= "10">
   <IMG Src= "book3.jpg" Height= "50" Width= "70"
        Vspace= "10" Hspace= "10"> <BR>
   <IMG Src= "book3.jpg" Height= "50" Width= "70"
        Vspace= "10" Hspace= "10">
   <IMG Src= "book3.jpg" Height= "50" Width= "70"
        Vspace= "10" Hspace= "10"> <BR>
</BODY>
</HTML>
```



ചിത്രം 4.32 : നിർദ്ദേശിച്ച വലുപ്പങ്ങളിലുള്ളതും, ഇടയ്ക്ക് തിരശ്ചീനവും ലംബവുമായി ശൂന്യസ്ഥലങ്ങൾ ഉൾക്കൊള്ളിച്ചിട്ടുള്ളതുമായ ചിത്രങ്ങൾ

നിർദിഷ്ട വീതിയിലും ഉയരത്തിലുമാണ് ചിത്രങ്ങൾ വിന്യസിച്ചിട്ടുള്ളതെന്നും തിരശ്ചീനമായും ലംബമായും നൽകിയ ശൂന്യസ്ഥലം സൂചിപ്പിച്ചിട്ടുള്ളത് പോലെയാണ് ചിത്രങ്ങൾക്കിടയിലെ അകലമെന്നും ചിത്രം 4.32-ൽ കാണാൻ സാധിക്കും.

വാക്യങ്ങൾ ആരംഭിക്കുന്ന വരി അടിസ്ഥാനമാക്കി ചിത്രങ്ങൾ വിന്യസിക്കുന്നതിന് ഉപയോഗിക്കുന്ന <IMG> ടാഗിന്റെ Align ആട്രിബ്യൂട്ടിനെക്കുറിച്ച് നമുക്ക് ചർച്ച ചെയ്യാം. ഈ ആട്രിബ്യൂട്ടിന് നൽകാനാകുന്ന വിലകൾ താഴെകൊടുത്തിരിക്കുന്നു.

Bottom : ചിത്രത്തിന്റെ കീഴ്ഭാഗം വാക്യങ്ങൾ ആരംഭിക്കുന്ന വരിയുമായി വിന്യസിക്കുന്നു.

Middle : ചിത്രത്തിന്റെ മധ്യഭാഗം (ലംബമായി) വാക്യങ്ങൾ ആരംഭിക്കുന്ന വരിയുമായി വിന്യസിക്കുന്നു.

Top : വാക്യങ്ങളുടെ മുകൾഭാഗവുമായി ചിത്രം വിന്യസിക്കുന്നു.

Align ആട്രിബ്യൂട്ടിന് വിലകൾ നൽകുമ്പോൾ ഉണ്ടാകുന്ന പ്രതീതി നമുക്ക് കാണാം. ഉദാഹരണം 4.17, ചിത്രം 4.33 എന്നിവ ഇത് വിശദീകരിക്കുന്നു.

ഉദാഹരണം 4.17: ഒരു ചിത്രത്തിന് വ്യത്യസ്ത വിന്യാസങ്ങൾ നൽകുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
   <TITLE> Alignment of Images </TITLE>
</HEAD>
```

```
<BODY Bgcolor= "#E0FFFF">
   <H2 Align= "center">Alignment of Images</H2>
   This Image is <I><U>aligned at the bottom </U></I>
   <IMG Src= "book3.jpg" Height= "40" Width= "50"
        Align= "Bottom"> <BR> <BR>
       This Image is <I><U>aligned at the Middle </U></I>
   <IMG Src= "book3.jpg" Height= "40" Width= "50"
        Align= "Middle"> <BR> <BR>
        This Image is <I><U>aligned at the Top </U></I>
   <IMG Src= "book3.jpg" Height= "40" Width= "50" Align="Top">
</BODY>
</HTML>
```

<IMG> ടാഗിന്റെ Align ആട്രിബ്യൂട്ടിന് മറ്റ് ചില വിലകൾ കൂടിയുണ്ട്. അവ ചിത്രത്തെ ബ്രൗസർ ജാലകത്തിന്റെ ഇടത് ഭാഗത്തേക്കും വലതുഭാഗത്തേക്കും യഥാക്രമം വിന്യസിക്കുന്ന left, right - എന്നിവയാണ്.

ചിത്രം 4.33 : വാക്യവുമായുള്ള ചിത്രത്തിന്റെ വിവിധ പൊരുത്തപ്പെടലുകൾ

ചിത്രം 4.34 : ബ്രൗസർ വിൻഡോയ്ക്ക് ഇടതും വലതുമായി വിന്യസിക്കപ്പെട്ട ചിത്രങ്ങൾ

നമുക്കു ചെയ്യാം

ബ്രൗസർ ജാലകത്തിൽ ചിത്രത്തിന്റെ ഇടത്തോട്ടും വലത്തോട്ടുമുള്ള വിന്യാസം ചിത്രം 4.34 -ൽ കാണിച്ചിട്ടുള്ളത് നിരീക്ഷിക്കുക. ഈ പേജിന് വേണ്ട HTML ലിഖിതം തയാറാക്കുക.

ഒരു ചിത്രത്തിന് ബോർഡർ നൽകുന്നു (Setting border around an image)

ഒരു വെബ് പേജിൽ ഉൾപ്പെടുത്തിയിട്ടുള്ള ചിത്രത്തിന് നമുക്ക് ബോർഡർ നൽകണമെന്ന് വിചാരിക്കുക. <IMG> ടാഗിന്റെ Border ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് ഇത് സാധ്യമാക്കാം. ഈ ആട്രിബ്യൂട്ടിന് അനുയോജ്യമായ വില നൽകികൊണ്ട് ബോർഡറിന് കനം നൽകാം. ഉദാഹരണം 4.18-ലെ HTML കോഡും ചിത്രം 4.35-ലെ അനുബന്ധ വെബ് പേജും Border ആട്രിബ്യൂട്ടിന്റെ ഉപയോഗം നമുക്ക് കാണിച്ചു തരുന്നു.

ഉദാഹരണം 4.18: ചിത്രത്തിന് ബോർഡർ നൽകുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
   <TITLE> Inserting Images </TITLE>
</HEAD>
<BODY Bgcolor= "#E0FFFF">
   <H2 Align= "center">Inserting Border to Images</H2>
   Here is an image <B><I><U> with Border </U></I></B>
   attribute
   <IMG Src= "book3.jpg" Height= "50" Width= "70" Border="5">
   <BR>Here is an image<B><I><U> without Border</U></I></B>
   attribute
   <IMG Src= "book3.jpg" Height= "50" Width= "70">
</BODY>
</HTML>
```

ചിത്രം 4.35 : ബോർഡറുകൾ ഉള്ളതും ഇല്ലാത്തതുമായ ചിത്രങ്ങൾ

<IMG> ടാഗിന്റെ വിവിധ ആട്രിബ്യൂട്ടുകളെക്കുറിച്ചും അവ ചിത്രത്തിലുണ്ടാക്കുന്ന വ്യതിയാനങ്ങളെക്കുറിച്ചും നാം പഠിച്ച് കഴിഞ്ഞു. Src ആട്രിബ്യൂട്ടിൽ നിർദ്ദേശിച്ച ചിത്രഫയൽ കണ്ടെത്താനായില്ലെങ്കിൽ വെബ് പേജ് കാഴ്ചയിൽ എങ്ങനെയുണ്ടാകും? വെബ് പേജിൽ ചിത്രത്തിനായി നിർദ്ദേശിച്ച സ്ഥലം ശൂന്യമായിരിക്കും. മറ്റ് ചില കാരണങ്ങളാലും ബ്രൗസറിന് ചിത്രം പ്രദർശിപ്പിക്കാനാകാതെ വരാറുണ്ട്. അത്തരം സന്ദർഭങ്ങളിൽ, ചിത്രത്തിനു പകരം ഒരു വാക്യം ബ്രൗസറിൽ നമുക്ക് ദൃശ്യമാക്കാനാകും. ബ്രൗസറിന് ചിത്രം പ്രദർശിപ്പിക്കാനാകാതെ വന്നാൽ പകരം ഒരു വാക്യം സൂചിപ്പിക്കാൻ HTML നൽകുന്ന ആട്രിബ്യൂട്ടാണ് **Alt**. ഉദാഹരണം 4.19-ലെ കോഡ് Alt ആട്രിബ്യൂട്ടിന്റെ ഉപയോഗം വിശദീകരിക്കുന്നു. അങ്ങനെ ലഭിക്കുന്ന വെബ് പേജ് ചിത്രം 4.36-ൽ കാണിച്ചിരിക്കുന്നു.

ഉദാഹരണം 4.19: <IMG> ടാഗിലെ Alt ആട്രിബ്യൂട്ടിന്റെ ഉപയോഗം വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD> <TITLE> Inserting Images </TITLE> </HEAD >
<BODY Bgcolor= "#E0FFFF">
   <H2 Align= "center">Inserting Images</H2>
   If the browser cannot display the image, then the text
   entered in the <B>Alt</B> attribute will be displayed.<BR>
```

```
    <IMG Src= "book5.jpg" Height= "20%" Width= "20%"
         Alt= "Image of an opened book"> <BR>
</BODY>
</HTML>
```

ചിത്രം 4.36 : <IMG> ടാഗിലെ Alt ആട്രിബ്യൂട്ടിന്റെ ഉപയോഗം

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. ഒരു HTML ലിഖിതത്തിൽ ചിത്രങ്ങൾ ഉൾക്കൊള്ളിക്കുന്നതിനായി __________ ടാഗ് ഉപയോഗിക്കുന്നു.
2. <IMG> ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്രിബ്യൂട്ടാണ് __________.
3. <IMG> ടാഗിൽ Alt ആട്രിബ്യൂട്ടിന്റെ ഉപയോഗമെന്ത്?
4. ഒരു ചിത്രം ഒരു പ്രത്യേക വലുപ്പത്തിൽ പ്രദർശിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്ന ആട്രിബ്യൂട്ടുകളുടെ പേരെഴുതുക.
5. രണ്ട് ചിത്രങ്ങൾക്കിടയിൽ തിരശ്ചീനവും ലംബവുമായ സ്ഥലങ്ങൾ ലഭ്യമാക്കുന്ന ആട്രിബ്യൂട്ടുകൾ ഏതൊക്കെയാണ്?

നമുക്ക് സംഗ്രഹിക്കാം

ഇന്റർനെറ്റിന്റെ സഫലത നിർണയിക്കുന്ന ഘടകമാണ് സുരക്ഷിതമായ ആശയവിനിമയം. ഇന്റർനെറ്റിലൂടെയുള്ള വ്യവഹാരങ്ങളുടെ സുരക്ഷിതത്വം നിർവഹിക്കുന്നത് HTTPS, ഡിജിറ്റൽ സാക്ഷ്യപത്രങ്ങൾ എന്നിവ ഉപയോഗിച്ചാണ്. ഡാറ്റ സംഭരിക്കുന്നതിനും കൈമാറ്റം (വിനിമയം) നടത്തുന്നതിനുമായി ഉപയോഗിക്കുന്ന വെബ് സെർവർ, സോഫ്റ്റ്‌വെയർ പോർട്ടുകൾ, DNS സെർവറുകൾ തുടങ്ങിയ അടിസ്ഥാന സൗകര്യങ്ങൾ ഇന്റർനെറ്റിന് ആവശ്യമാണ്. ഒന്നോ അതിലധികമോ വെബ് പേജുകൾ ചേർന്നാണ് വെബ്സൈറ്റ് ഉണ്ടായിരിക്കുന്നത്. HTML കോഡ് എഴുതിയോ അല്ലെങ്കിൽ വെബ് രൂപകൽപന ചെയ്യാനുപയോഗിക്കുന്ന സോഫ്റ്റ്‌വെയറുകളുടെ സഹായത്താലോ നമുക്ക് വെബ് പേജുകൾ നിർമിക്കാം. വെബ് പേജുകൾ പരിവർത്തനാത്മകമെന്നും സ്ഥിരമായതെന്നും തരംതിരിച്ചിട്ടുണ്ട്. സ്ക്രിപ്റ്റുകൾ ഉപയോഗിച്ചുകൊണ്ട് പരി

വർത്തനാത്മക പേജുകൾ വികസിപ്പിക്കാം. ക്ലയന്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിങ് ഭാഷകളായ ജാവ സ്ക്രിപ്റ്റ്, VB സ്ക്രിപ്റ്റ് തുടങ്ങിയവ ഉപയോഗിച്ച് ക്ലയന്റ് ഭാഗത്ത് ഡാറ്റയുടെ സാധുത ഉറപ്പിക്കാനാവും. പരിവർത്തനാത്മക വെബ് പേജുകൾ സെർവറിൽ നിർമിക്കുന്നതിനായി സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകളായ PHP, ASP തുടങ്ങിയവ ഉപയോഗിക്കുന്നു. വെബ് സൈറ്റിന് ആകമാനം ഏകീകൃത ശൈലി ലഭ്യമാക്കാൻ കാസ്കേഡിങ് സ്റ്റൈൽ ഷീറ്റ് (CSS) ഉപയോഗിക്കുന്നു. HTML ഭാഷയെയും വെബ് പേജ് രൂപപ്പെടുത്തലിനെയും കുറിച്ചുള്ള അടിസ്ഥാന ആശയങ്ങൾ നാം ചർച്ച ചെയ്തു. വ്യത്യസ്തങ്ങളായ ടാഗുകളെയും അവയുടെ പ്രധാനപ്പെട്ട ആട്രിബ്യൂട്ടുകളേയും നാം പരിചയപ്പെട്ടു. ചില ടാഗുകൾ ജോഡികളായും മറ്റ് ചിലത് ജോഡികളല്ലാതെയും ഉപയോഗിക്കുന്നു. വിവിധ ടാഗുകൾ കൊണ്ട് വാക്യങ്ങൾ ഭംഗിയായി വിന്യസിച്ച് നമുക്കൊരു വെബ് പേജ് നിർമിക്കാം. മാർക്യൂകൾ, ചിത്രങ്ങൾ, ശബ്ദങ്ങൾ, ചലിക്കുന്ന ചിത്രങ്ങൾ എന്നിവ ഉൾപ്പെടുത്തി വെബ് പേജുകളുടെ ഭംഗി വർധിപ്പിക്കാം. ഈ അധ്യായത്തിൽ ചർച്ച ചെയ്ത ആശയങ്ങളെക്കുറിച്ച് വ്യക്തമായ ധാരണയും HTML ഡോക്യുമെന്റുകൾ നിർമിക്കുന്നതിന് വേണ്ട പ്രായോഗിക പരിചയവും നമുക്കുള്ളതുകൊണ്ട് തുടർന്ന് വരുന്ന അധ്യായങ്ങളിലെ ആശയങ്ങൾ എളുപ്പത്തിൽ ഗ്രഹിക്കാവുന്നതാണ്. കൂടാതെ, മനോഹരമായ വെബ് സൈറ്റുകളുടെ മാതൃകയുണ്ടാക്കുന്നതിനും, വെബ് ആപ്ലിക്കേഷനുകൾ സ്വയം നിർമിക്കുന്നതിനും നാം പ്രാപ്തരാവുകയും ചെയ്യും.

നമുക്ക് പരിശീലിക്കാം

1. താഴെ കൊടുത്തിരിക്കുന്ന സവിശേഷതകളോടുകൂടിയ കേരളത്തിന്റെ ഒരു വെബ് പേജ് നിർമിക്കുന്നതിന് വേണ്ട HTML കോഡ് എഴുതുക.
   - വാക്യങ്ങൾക്ക് രൂപഘടന നൽകുന്ന ടാഗുകളും ആട്രിബ്യൂട്ടുകളും ഉപയോഗിച്ച് ഒരു തലക്കെട്ടും അഞ്ച് വാചകങ്ങളുള്ള ഒരു ഖണ്ഡികയും തയാറാക്കുക.
   - വെബ് പേജിന്റെ പശ്ചാത്തല ചിത്രമായി ഒരു പ്രകൃതി ദൃശ്യം വേണം.
2. ചുവടെ കൊടുത്തിരിക്കുന്ന വിശദാംശങ്ങളും സവിശേഷതകളുമുള്ള നിങ്ങളുടെ സ്കൂൾ വെബ് പേജിന് ആവശ്യമായ HTML കോഡെഴുതുക.
   - പേജിന്റെ പശ്ചാത്തലത്തിന് ഒരു നിറം നൽകുക.
   - സ്കൂളിന്റെ ഒരു ചിത്രം നൽകുക.
3. ചുവടെ കൊടുത്തിരിക്കുന്ന വിശദാംശങ്ങളും സവിശേഷതകളുമുള്ള നിങ്ങളുടെ സ്കൂൾ വെബ് പേജിനാവശ്യമായ HTML കോഡെഴുതുക.
   - വാക്യങ്ങൾ രൂപപ്പെടുത്തുന്ന ടാഗുകളും ആട്രിബ്യൂട്ടുകളും ഉപയോഗിച്ചുകൊണ്ട് ഓരോ ജില്ലയ്ക്കും ഒരു തലക്കെട്ടും തുടർന്ന് മൂന്ന് വാചകങ്ങളുള്ള ഒരു ഖണ്ഡികയും തയാറാക്കുക.
   - സ്കൂളിന്റെ തപാൽ മേൽവിലാസം ഉൾപ്പെടുത്തുക.
   - "പുതിയ അധ്യയന വർഷത്തേക്കുള്ള പ്രവേശനം മെയ് മാസം 10-ാം തീയതി ആരംഭിക്കുന്നു" എന്ന മാർക്യൂ ഉൾപ്പെടുത്തുക.

4. ചുവടെ കൊടുത്തിരിക്കുന്ന സവിശേഷതകൾ ഉൾപ്പെടുത്തി നമ്മുടെ ദേശീയ ഗാനത്തിന്റെ വരികൾ പ്രദർശിപ്പിക്കുന്ന ഒരു വെബ് പേജിന് ആവശ്യമായ HTML കോഡ് എഴുതുക.
   - വ്യത്യസ്ത സവിശേഷതകൾ ഉള്ള ഫോണ്ട് ഉപയോഗിച്ച് ഒരു തലക്കെട്ട് നിർമിക്കണം.
   - നമ്മുടെ ദേശിയ പതാകയുടെ ഒരു ചിത്രം ഉൾപ്പെടുത്തണം.

നമുക്ക് വിലയിരുത്താം

1. ഇന്റർനെറ്റിലൂടെയുള്ള ഡാറ്റയുടെ സഞ്ചാരത്തിൽ റൂട്ടറുകളുടെ പങ്ക് എന്ത്?
2. സമൂഹ മാധ്യമ വെബ് സൈറ്റുകൾ ആശയവിനിമയത്തിന് അവരുടെ സ്വന്തം പ്രോട്ടോകോളാണ് വികസിപ്പിച്ചിരിക്കുന്നത്. ഇന്റർനെറ്റിൽ TCP/IP പ്രോട്ടോകോൾ ഉപയോഗിക്കുന്നതു കൊണ്ട് ഇതെങ്ങനെ സാധ്യമാക്കും?
3. ഇ-മെയിൽ അക്കൗണ്ടിലേക്ക് യൂസർ നാമവും പാസ്‌വേഡും ഇന്റർനെറ്റിലൂടെ സുരക്ഷിതമായാണ് അയയ്ക്കേണ്ടത്.
   a. സെർവറിലേക്ക് ഡാറ്റ അയക്കാൻ ഉപയോഗിക്കുന്ന സുരക്ഷിത സാങ്കേതികവിദ്യയുടെ പേരെഴുതുക.
   b. സുരക്ഷിത ഡാറ്റാവിനിമയത്തിന് ഈ സാങ്കേതികവിദ്യ എങ്ങനെ സഹായിക്കുന്നു?
4. ഓൺ-ലൈനിലൂടെയുള്ള വ്യവഹാരത്തിൽ പേമെന്റ് ഗേറ്റ്‌വേയുടെ ധർമമെന്താണ്?
5. ABC എഞ്ചിനിയറിങ് കോളേജിന്റെ കാമ്പസിൽ ഏകദേശം 1000 കമ്പ്യൂട്ടറുകൾ ഇന്റർനെറ്റുമായി ബന്ധിപ്പിച്ചിട്ടുണ്ട്. ഒരു പ്രാദേശിക സെർവർ ഈ സാങ്കേതികവിദ്യാഭ്യാസ സ്ഥാപനത്തിൽ ഉണ്ടാകുന്നതിന്റെ നേട്ടമെന്താണ്?
6. ഒരു വെബ് സെർവർ ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിനും വെബ് സെർവർ പാക്കേജിനും ഉദാഹരണമെഴുതുക.
7. വെബ് സെർവറിൽ സോഫ്റ്റ്‌വെയർ പോർട്ടിന്റെ ഉപയോഗമെന്ത്?
8. HTTP -ക്ക് ഉപയോഗിച്ചിട്ടുള്ള പോർട്ടാണ് __________.
9. നിങ്ങൾ www.prdkerala.org എന്ന വെബ്സൈറ്റ് പരിശോധിക്കുകയാണെന്ന് കരുതുക. DNS എങ്ങനെയാണ് IP മേൽവിലാസം വേർതിരിച്ചെടുക്കുന്നതെന്ന് വിശദീകരിക്കുക.
10. സ്ക്രിപ്റ്റുകൾ എന്നാൽ എന്ത്? സ്ക്രിപ്റ്റ് എഴുതാൻ ഉപയോഗിക്കുന്ന വിവിധ ഭാഷകൾ വിശദീകരിക്കുക.
11. നിങ്ങളുടെ സ്കൂൾ വെബ് സൈറ്റിലെ ഹോം പേജും 11-ാം ക്ലാസിലെ പരീക്ഷാഫലങ്ങൾ കാണിക്കുന്ന വെബ് പേജും പരിഗണിക്കുക.
    a. ഈ രണ്ട് പേജുകളും തമ്മിൽ നിർമാണത്തിലുള്ള വ്യത്യാസങ്ങൾ താരതമ്യം ചെയ്യുക.
    b. ഈ വെബ് പേജുകൾ വികസിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്ന സാങ്കേതികവിദ്യകളെക്കുറിച്ച് എഴുതുക.

12. a. ജാവ സ്ക്രിപ്റ്റ് ഫയലിന്റെ ഫയൽ എക്സ്റ്റൻഷൻ ___________ ആണ്.
    b. ഒരു വെബ് പേജിൽ ജാവ സ്ക്രിപ്റ്റിന്റെ പ്രധാനപ്പെട്ട രണ്ട് ഉപയോഗങ്ങളെഴുതുക.
13. എന്താണ് അജാക്സ്? അതിന്റെ ഉപയോഗമെന്ത്?
14. നിങ്ങളുടെ ക്ലാസിലെ കുട്ടികൾക്ക് ഓരോ പരീക്ഷയ്ക്കും ലഭിച്ച മാർക്ക് പ്രദർശിപ്പിക്കുന്നതിനായി രവി ഒരു വെബ് പേജ് നിർമിക്കാൻ ആഗ്രഹിക്കുന്നു.

    a. അത് നിർവഹിക്കുന്നതിന് ഒരു സാങ്കേതിക വിദ്യ നിർദേശിക്കുക.

    b. നിങ്ങളുടെ നിർദേശം ന്യായീകരിക്കുക.
15. MYSQL-ലെ ഡാറ്റബേസ് ഉപയോഗിച്ചുകൊണ്ട് PHP-യിൽ മനോജ് ഒരു വെബ് സൈറ്റ് വികസിപ്പിക്കാൻ ആലോചിക്കുന്നു. ലിനക്സ് വെബ് ഹോസ്റ്റിംഗ് സംവിധാനമാണ് അയാൾ ഉപയോഗിക്കുന്നത് എങ്കിൽ, അതിന്റെ നിർവ്വഹണത്തിന് ആവശ്യമായവ എന്തെല്ലാമാണ്?
16. “മിക്കവാറും എല്ലാ വെബ് സൈറ്റുകളും വികസിപ്പിക്കുമ്പോൾ CSS ഉപയോഗിക്കാറുണ്ട്. വെബ് സൈറ്റുകളിൽ CSS ഉപയോഗിക്കുന്നത് കൊണ്ടുള്ള നേട്ടങ്ങൾ എന്തൊക്കെയാണ്?
17. HTML വികസിപ്പിച്ചത് ആരാണ്?
18. HTML-ൽ പ്രധാനമായും രണ്ട് ഭാഗങ്ങളാണുള്ളത്. അവയുടെ പേരുകൾ എഴുതുക.
19. നിങ്ങൾ വെബ്സൈറ്റുകൾ വിശകലനം ചെയ്യുകയാണെങ്കിൽ ലിങ്കുകൾ, സന്ദർശിച്ച ലിങ്കുകൾ, പശ്ചാത്തലം തുടങ്ങിയവയ്ക്ക് വിവിധ നിറങ്ങൾ കാണാനാകും. HTML-ൽ ഇത് എങ്ങനെ ചെയ്യുന്നുവെന്ന് ഉദാഹരണസഹിതം വിശദമാക്കുക.
20. HTML-ലെ ജോഡിയായ ടാഗുകളും ജോഡിയല്ലാത്ത ടാഗുകളും ഉദാഹരണസഹിതം താരതമ്യം ചെയ്യുക.
21. Alink ആട്രിബ്യൂട്ടിന്റെ തനതായ നിറം ഏതാണ്?
22. Vlink ആട്രിബ്യൂട്ടിന്റെ തനതായ നിറം ഏതാണ്?
23. താഴെ തന്നിട്ടുള്ള HTML വാക്കുകളെ തരംതിരിക്കുക.

    BR, IMG, ALIGN, FONT, FACE
24. Noshade ആട്രിബ്യൂട്ട് ഉള്ള ടാഗിന്റെ പേരെഴുതുക.
25. ഒരു വെബ് പേജിൽ ചിത്രഫയലുകൾ ഉൾക്കൊള്ളിക്കുന്നതിന് ഉപയോഗിക്കുന്ന <IMG> ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്രിബ്യൂട്ട് ഏത്?
26. <IMG> ടാഗിൽ Alt ആട്രിബ്യൂട്ടിന്റെ ആവശ്യകത പ്രസ്താവിക്കുക.
27. <IMG> ടാഗ് ഉപയോഗിക്കുമ്പോൾ ലഭിക്കുന്ന ഒരു ചിത്രത്തിന്റെ തനത് വിന്യാസം ___________ ആകുന്നു.
28. <FONT> ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്രിബ്യൂട്ടുകളുടെ നാമാവലി തയാറാക്കുക.

J3T9P7

5 HTML ഉപയോഗിച്ചു കൊണ്ടുള്ള വെബ് രൂപകൽപ്പന

പ്രധാന പഠനനേട്ടങ്ങൾ

പഠനം പൂർത്തിയാകുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ

- വെബ് പേജുകളിലെ വിഭവങ്ങൾ ഫലപ്രദമായി അവതരിപ്പിക്കാൻ വിവിധ ഇനം ലിസ്റ്റുകൾ ഉപയോഗിക്കുന്നു.
- ഹൈപ്പർ ലിങ്കിന്റെ പ്രസക്തി തിരിച്ചറിഞ്ഞ് വിവിധ തരം ലിങ്കിങ്ങിനായി <A> ടാഗ് ഉപയോഗിക്കുന്നു.
- <EMBED> ടാഗ് ഉപയോഗിച്ച് ഓഡിയോ, വീഡിയോ എന്നിവ വെബ് പേജിൽ ഉൾപ്പെടുത്തുന്നു.
- വെബ് പേജിനുള്ളിൽ പശ്ചാത്തലസംഗീതവും വീഡിയോയും ഉൾപ്പെടുത്തുന്നു.
- ടേബിൾ നിർമിക്കാൻ ആവശ്യമായ ടാഗുകളുടെയും അവയുടെ ആട്രിബ്യൂട്ടുകളുടെയും പട്ടിക തയാറാക്കി വിശദീകരിക്കുന്നു.
- വിവിധ സവിശേഷതകളുള്ള ടേബിളിന്റെ രൂപകൽപനയ്ക്കായി <TABLE> ടാഗുമായി ബന്ധപ്പെട്ട ടാഗുകൾ ഉപയോഗിക്കുന്നു.
- വ്യത്യസ്തങ്ങളായ ടേബിൾ രൂപകൽപനക്കായി വേണ്ട ടാഗുകളും അവയുടെ ആട്രിബ്യൂട്ടുകളും മനസിലാക്കുന്നു.
- വെബ് പേജിനുള്ളിൽ ഫ്രെയിമുകളുടെ ആവശ്യകത തിരിച്ചറിയുന്നു.
- ഫ്രെയിം ഉപയോഗിച്ച് ഒരേ ബ്രൗസർ ജാലകത്തിൽ തന്നെ ഒന്നിലധികം വെബ് പേജുകൾ പ്രദർശിപ്പിക്കുന്നതിന് വേണ്ട ടാഗുകൾ ഉപയോഗിക്കുന്നു.
- വെബ് പേജിനുള്ളിൽ ഫോം എന്ന ആശയം തിരിച്ചറിയുന്നു.
- ഒരു ഫോമിന്റെ വിവിധ ഭാഗങ്ങളും അവ നിർമിക്കുന്നതിനാവശ്യമായ ടാഗുകളും അവയുടെ ആട്രിബ്യൂട്ടുകളും പരിചയപ്പെടുന്നു.
- ടേബിളുകൾ, ഫ്രെയിമുകൾ, ഫോം എന്നിവ ഉപയോഗിച്ച് വെബ് പേജുകൾ രൂപകൽപന ചെയ്യുന്നു.

HTML -ലെ അടിസ്ഥാന ടാഗുകൾ കഴിഞ്ഞ അധ്യായത്തിലൂടെ നാം പഠിച്ചു. അത്തരം ടാഗുകളും അവയുടെ ആട്രിബ്യൂട്ടുകളും ഉപയോഗിച്ച് ലളിതമായ വെബ് പേജുകൾ രൂപകൽപന ചെയ്യുന്നതും നാം പഠിച്ചു. എന്നാൽ കൂടുതൽ സൗകര്യങ്ങളും അവശ്യ സേവനങ്ങളും നൽകുന്ന മറ്റ് ചില വെബ് സൈറ്റുകളും നമുക്ക് പരിചിതമാണല്ലോ? വിവിധതരം ലിസ്റ്റുകൾ അടങ്ങുന്ന വെബ്സൈറ്റുകളുണ്ട്. എങ്ങനെയാണ് വിവരങ്ങൾ പട്ടികാരൂപത്തിൽ രേഖപ്പെടുത്തുന്നത്? വെബ് പേജുകളുടെ തമ്മിലുള്ള പരസ്പര ബന്ധമാണ് വേൾഡ് വൈഡ് വെബിന്റെ നട്ടെല്ല്. വിവിധതരം ലിങ്കിങ്ങിനെ (Linking) കുറിച്ച് ഈ അധ്യായത്തിൽ നമ്മൾ ചർച്ച ചെയ്യുന്നുണ്ട്. ചില വിവരങ്ങൾ പട്ടികാരൂപത്തിൽ കാണപ്പെടാറുണ്ട്. ചില സമയത്ത് ഒരേ ബ്രൗസർ ജാലകത്തിൽ ഒന്നിലധികം വെബ്പേജുകൾ നമ്മൾ കാണാറുണ്ട്. അതുപോലെതന്നെ പരീക്ഷയുടെ മാർക്ക് ലിസ്റ്റ് ലഭിക്കുവാനായി രജിസ്റ്റർ നമ്പർ നൽകുന്നതിനും സ്കൂൾ, കോളേജ് പ്രവേശനം, സ്കോളർഷിപ്പ് എന്നിവയ്ക്കുള്ള അപേക്ഷകൾ സമർപ്പിക്കുന്നതിനും, വൈദ്യുതി, വെള്ളം എന്നിവയ്ക്കുള്ള ബിൽ അടയ്ക്കുന്നതിനുമുള്ള വെബ്സൈറ്റുകൾ നമുക്ക് പരിചിതങ്ങളാണ്. ഒരേ ബ്രൗസർ ജാലകത്തിൽ ഒന്നിലധികം വെബ് പേജുകൾ ഉൾക്കൊള്ളിക്കാനാകുമോ? ഉൾക്കൊള്ളിക്കാൻ ആകുമെങ്കിൽ അത് എങ്ങനെ? ഉപയോക്താവിൽ നിന്നും ഡാറ്റ സ്വീകരിച്ച് കൂടുതൽ വിവരങ്ങൾ നൽകുന്ന വെബ് പേജുകൾ ഉണ്ടാക്കുന്നതെങ്ങനെ? വെബ് പേജ് നിർമാതാക്കൾക്ക് ഇത്തരത്തിലുള്ള എല്ലാ സൗകര്യങ്ങളും HTML-ൽ ലഭ്യമാണ്. ഈ ചോദ്യങ്ങൾക്കുള്ള ഉത്തരങ്ങൾ ഈ അധ്യായത്തിലൂടെ ചർച്ച ചെയ്യുന്നു.

5.1 HTML-ലെ ലിസ്റ്റുകൾ (Lists in HTML)

വിവരങ്ങൾ അവതരിപ്പിക്കുമ്പോൾ ലിസ്റ്റുകൾ ആശയവിനിമയ സാധ്യത വർധിപ്പിക്കുന്നു. വിവിധതരം ലിസ്റ്റുകളുണ്ട്. അക്കമിട്ട ലിസ്റ്റുകളും ബുള്ളറ്റിട്ട ലിസ്റ്റുകളും നമുക്ക് പരിചിതമാണല്ലോ. ഇത്തരം ലിസ്റ്റുകൾ വെബ്പേജിൽ ഉൾപ്പെടുത്താൻ HTML സൗകര്യം ചെയ്യുന്നു. എല്ലാ ലിസ്റ്റുകളിലും ഒന്നോ അതിലധികമോ ഘടകങ്ങൾ ഉണ്ടായിരിക്കണം. HTML-ലെ ലിസ്റ്റുകളെ ക്രമരഹിത ലിസ്റ്റ്, ക്രമലിസ്റ്റ്, ഡെഫനിഷൻ ലിസ്റ്റ് എന്നിങ്ങനെ മൂന്നായി തരംതിരിച്ചിരിക്കുന്നു.

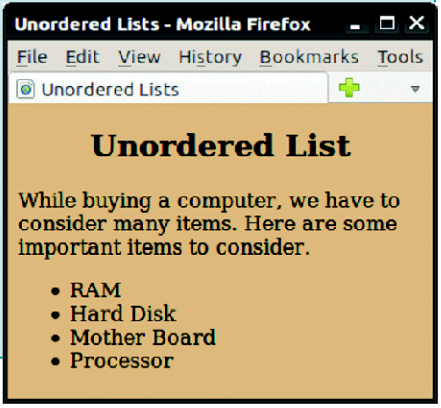
5.1.1 ക്രമരഹിത ലിസ്റ്റുകൾ (Unordered lists)

ക്രമരഹിത ലിസ്റ്റ് അഥവാ ബുള്ളറ്റഡ് ലിസ്റ്റുകളിൽ ഓരോ ഇനത്തിന്റെയും മുന്നിൽ ഒരു ബുള്ളറ്റോ അല്ലെങ്കിൽ മറ്റേതെങ്കിലും ഗ്രാഫിക് ചിഹ്നങ്ങളോ പ്രദർശിപ്പിക്കുന്നു. **<UL>**, **</UL>** എന്നീ ടാഗ് ജോഡികൾ ഉപയോഗിച്ച് ക്രമരഹിത ലിസ്റ്റുകൾ നിർമിക്കുവാൻ നമുക്ക് സാധിക്കും. ലിസ്റ്റിലെ ഒരോ ഇനവും **<LI>**, **</LI>** ടാഗ് ജോഡികൾ ഉപയോഗിച്ച് ചേർക്കപ്പെടുന്നു. ലിസ്റ്റിലെ ഇനങ്ങൾ യാതൊരു ക്രമവും ഇല്ലാതെ പ്രദർശിപ്പിക്കുന്നതിനാണ് ക്രമരഹിത (Unordered) ലിസ്റ്റുകൾ ഉപയോഗിക്കുന്നത്.

ഉദാഹരണം 5.1-ൽ ഒരു കമ്പ്യൂട്ടറിന്റെ ഏതാനും ഹാർഡ്‌വെയർ ഭാഗങ്ങൾ ബുള്ളറ്റഡ് ലിസ്റ്റായി പ്രദർശിപ്പിക്കുന്നതിനുള്ള HTML കോഡ് നൽകുന്നു. ഇതിനനുസൃതമായ വെബ്പേജ് ചിത്രം 5.1 ൽ കാണിച്ചിരിക്കുന്നു.

ഉദാഹരണം 5.1: ക്രമരഹിത ലിസ്റ്റ് നിർമിക്കാനാവശ്യമായ HTML കോഡ്

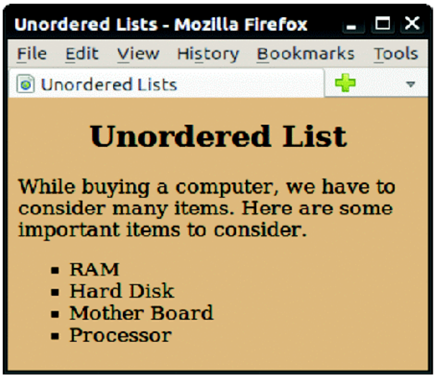
```
<HTML>
<HEAD>
   <TITLE> Unordered Lists </TITLE>
</HEAD>
<BODY Bgcolor= "#DEB887">
   <CENTER> <H2> Unordered List </H2> </CENTER>
   While buying a computer, we have to consider many items.
   Here are some important items
   to consider.
   <UL>
     <LI> RAM </LI>
     <LI> Hard Disk </LI>
     <LI> Mother Board </LI>
     <LI> Processor </LI>
   </UL>
</BODY>
</HTML>
```



ചിത്രം 5.1: ക്രമരഹിത ലിസ്റ്റ് ഉപയോഗിച്ച വെബ് പേജ്

<UL> ടാഗിലെ Type ആട്രിബ്യൂട്ടിന് Disc (തന തുവില), Square, Circle എന്നീ മൂന്ന് വിലകൾ നൽകാവുന്നതാണ്. ഇങ്ങനെ ചെയ്യുമ്പോൾ ലിസ്റ്റി ലെ ഓരോ ഇനത്തിന്റെയും മുന്നിൽ അതിനനുസൃ തമായ ബുള്ളറ്റ് ചിത്രീകരിക്കുന്നു. താഴെ നൽകി യിട്ടുള്ള HTML കോഡ് ചിത്രം 5.2 ൽ കാണിച്ചിരി ക്കുന്ന ലിസ്റ്റ് അടങ്ങിയ വെബ് പേജ് പ്രദർശിപ്പി ക്കുന്നു.

```
<UL Type= "Square">
    <LI> RAM </LI>
    <LI> Hard Disk </LI>
    <LI> Mother Board </LI>
    <LI> Processor </LI>
</UL>
```



ചിത്രം 5.2: ബുള്ളറ്റഡ് ക്രമരഹിത ലിസ്റ്റ്

> <UL> ടാഗിന്റെ Type ആട്രിബ്യൂട്ടിന് None എന്ന വില നൽകുകയാണെങ്കിൽ ക്രമരഹിത ലിസ്റ്റിലെ ഓരോ ഇനത്തിന്റെയും മുമ്പിൽ ബുള്ളറ്റുകളൊന്നും പ്രദർശിപ്പിക്കുകയില്ല. മുക ളിൽ കൊടുത്തിരിക്കുന്ന HTML കോഡിൽ <UL Type = "None"> എന്ന മാറ്റം വരുത്തി പരിശോധിക്കാം.

5.1.2 ക്രമലിസ്റ്റ് (Ordered lists)

ലിസ്റ്റിലെ ഇനങ്ങളുടെ മുന്നിൽ സംഖ്യകളോ അക്ഷരങ്ങളോ ചേർത്ത് ക്രമമായി പ്രദർശി പ്പിക്കുന്നതിന് ക്രമലിസ്റ്റ് ഉപയോഗിക്കുന്നു. **<OL>**, **</OL>** എന്നീ ടാഗ് ജോഡികൾ ഉപ യോഗിച്ചാണ് ഇത്തരം ലിസ്റ്റ് തയാറാക്കുന്നത്. ക്രമലിസ്റ്റിലെ ഓരോ ഇനവും പ്രദർശിപ്പി ക്കുന്നതിന് **<LI>**, **</LI>** എന്നീ ടാഗ് ജോഡികൾ ഉപയോഗിക്കുന്നു. ഇതിനെ സംഖ്യാ നുസൃത ലിസ്റ്റ് എന്നും പറയാറുണ്ട്.

ഉദാഹരണം 5.2 ൽ ക്രമലിസ്റ്റ് നിർമിക്കാനാവശ്യമായ കോഡ് കൊടുത്തിരിക്കുന്നു. ഇതിനനുസൃതമായ വെബ്പേജ് ചിത്രം 5.3 ൽ പ്രദർശിപ്പിക്കുന്നു.

ഉദാഹരണം 5.2: ക്രമലിസ്റ്റ് നിർമിക്കാനാവശ്യമായ HTML കോഡ്

```
<HTML>
<HEAD>
    <TITLE> Ordered Lists </TITLE>
</HEAD>
    <BODY Bgcolor= "#DDA0DD">
    <H2 Align= "center"> Ordered List </H2>
    Consider the memory devices of a computer.
    Then according to the speed of data processing,
```

```
we can arrange the memory devices as follows.
    <OL>
      <LI> Registers </LI>
      <LI> Cache </LI>
      <LI> RAM </LI>
      <LI> Hard Disk </LI>
    </OL>
</BODY>
</HTML>
```

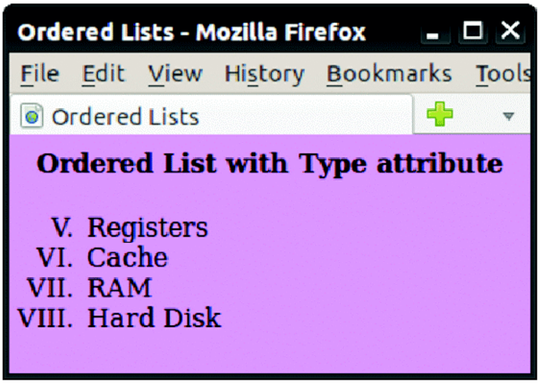
ചിത്രം 5.3: ക്രമലിസ്റ്റ് അടങ്ങിയ വെബ് പേജ്

ചിത്രം 5.3 ൽ 1 മുതൽ 4 വരെയുള്ള ക്രമനമ്പർ ഉൾക്കൊള്ളുന്ന ലിസ്റ്റ് പ്രദർശിപ്പിക്കുന്നു. നമുക്ക് മറ്റ് പല സംഖ്യാരീതികളും ഉപയോഗിച്ച് ലിസ്റ്റിലെ ഇനങ്ങൾ പ്രദർശിപ്പിക്കാം. **Type** ആട്രിബ്യൂട്ടിന് ചുവടെ കൊടുത്തിരിക്കുന്ന വിലകൾ നൽകി അനുയോജ്യമായ വിധത്തിൽ ലിസ്റ്റിലെ ഇനങ്ങളുടെ സംഖ്യാരീതിയിൽ നമുക്ക് മാറ്റം വരുത്താം.

1	തനതായ സംഖ്യാരീതി (1, 2, 3, ...)
A	ഇംഗ്ലീഷ് ഭാഷയിലെ വലിയ അക്ഷരങ്ങൾ (A, B, C, ...)
a	ഇംഗ്ലീഷ് ഭാഷയിലെ ചെറിയ അക്ഷരങ്ങൾ (a, b, c, ...)
I	റോമൻ ലിപിയിലെ വലിയ സംഖ്യകൾ (I, II, III, ...)
i	റോമൻ ലിപിയിലെ ചെറിയ സംഖ്യകൾ (i, ii, iii, ...)

ഒരു ക്രമലിസ്റ്റിലെ ഇനങ്ങളുടെ ക്രമനമ്പറിന്റെ തുടക്കം സ്വാഭാവികമായും ആ സംഖ്യാരീതിയിലെ ആദ്യ നമ്പറിലായിരിക്കും. അതായത് ക്രമനമ്പർ തുടങ്ങുന്നത് 1, A, a, I, i എന്നിവയിൽ ഏതെങ്കിലും ഒന്നിലായിരിക്കും. ഇത് നമ്മുടെ ഇഷ്ടാനുസരണം ഏത് സംഖ്യയിലോ അക്ഷരത്തിലോ ആരംഭിക്കാം. ഇതിനായി സ്റ്റാർട്ട് (**Start**) ആട്രിബ്യൂട്ട് ഉപയോഗിക്കാം. ഒരു ക്രമലിസ്റ്റിന്റെ ഇനങ്ങളുടെ ക്രമനമ്പർ 5 ൽ തുടങ്ങണമെങ്കിൽ **<OL Start= "5">** എന്നതിലൂടെ സാധ്യമാകും. അതിനാൽ 5 ൽ തുടങ്ങിയ ക്രമമ്പർ 6, 7, 8, ... എന്നിങ്ങനെ തുടരുന്നതാണ്.

ഏത് സംഖ്യാരീതി ഉപയോഗിക്കുമ്പോഴും **Start** ആട്രിബ്യൂട്ടിന് നൽകുന്ന വില പൂർണസംഖ്യ ആയിരിക്കും. ഉദാഹരണത്തിന് താഴെ കൊടുത്തിരിക്കുന്ന ക്രമലിസ്റ്റിന്റെ തുടക്കനമ്പർ V ആണ്, പിന്നീട് VI, VII, VIII, ... എന്നിങ്ങനെ തുടരുന്നു. ഇതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 5.4-ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 5.4: സംഖ്യാരീതിയും തുടക്കവിലയും വ്യത്യസ്തമായ ക്രമലിസ്റ്റ്

```
    <BODY Bgcolor ="#DDA0DD">
    <H4 Align="center">Ordered List with Type attribute</H4>
    <OL Type= "I" Start= "5">
      <LI> Registers </LI>
      <LI> Cache </LI>
      <LI> RAM </LI>
      <LI> Hard Disk </LI>
    </OL>
</BODY>
```

ഒരു ക്രമലിസ്റ്റിലെ ഇനങ്ങളുടെ ക്രമനമ്പർ അവരോഹണ ക്രമത്തിൽ പ്രദർശിപ്പിക്കാം. ഇതിനായി <OL> ടാഗിന്റെ ആട്രിബ്യൂട്ടായ Reversed ഉപയോഗിക്കാം. ഉദാഹരണം 5.2 ൽ നൽകിയിരിക്കുന്ന HTML കോഡിൽ <OL Reversed> എന്ന മാറ്റം വരുത്തി നമുക്ക് പരിശോധിക്കാം. Reversed ഒരു ബൂളിയൻ ആട്രിബ്യൂട്ട് ആയതിനാൽ ഇതിന് വില നൽകേണ്ടതില്ല.

ഒരു ക്രമലിസ്റ്റിലെ ഇനങ്ങളുടെ ക്രമനമ്പർ നമുക്ക് അനുയോജ്യമായ വില നൽകി പ്രദർശിപ്പിക്കുവാൻ <LI> ടാഗിന്റെ ആട്രിബ്യൂട്ടായ Value ഉപയോഗിക്കാം. മുകളിൽ കൊടുത്തിരിക്കുന്ന HTML കോഡിൽ <LI Value = "10"> RAM </LI> എന്നാണ് നൽകുന്നതെങ്കിൽ, 10 RAM എന്നായിരിക്കും വെബ് പേജിൽ പ്രദർശിപ്പിക്കുക. ഇതേ രീതിയിൽ <OL> ടാഗിനുള്ളിലെ ഓരോ <LI> ഇനങ്ങൾക്കും Value ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് ഓരോരോ വിലകൾ നൽകാം.

5.1.3 ഡെഫനിഷൻ ലിസ്റ്റ് (Definition lists)

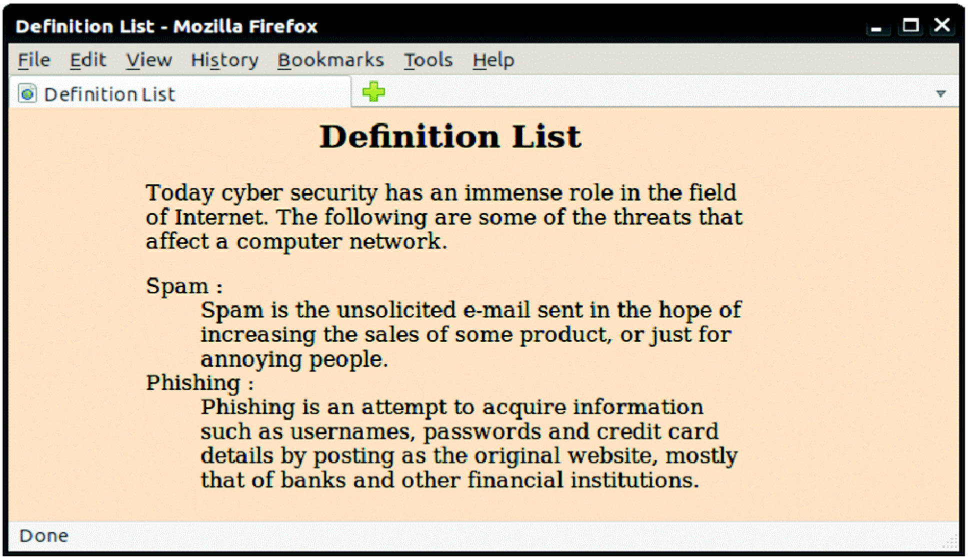
കുറച്ച് പദങ്ങളും അവയുടെ നിർവചനങ്ങളും നൽകുന്നതിന് ഡെഫനിഷൻ ലിസ്റ്റ് ഉപയോഗിക്കുന്നു. തൊട്ടടുത്ത വരിയിൽ പദങ്ങളിൽ നിന്നും തിരശ്ചീനമായി നിശ്ചിത അകലം കഴിഞ്ഞതിന് ശേഷമാണ് നിർവചനങ്ങൾ പ്രദർശിപ്പിക്കുന്നത്. **<DL>**, **</DL>** എന്നീ ടാഗ് ജോഡികൾ ഉപയോഗിച്ച് ഡെഫനിഷൻ ലിസ്റ്റ് നിർമിക്കുവാൻ നമുക്ക് സാധിക്കും. ഇവിടെ ബുള്ളറ്റുകളോ സംഖ്യകളോ ഇനങ്ങൾക്ക് മുൻപിൽ പ്രദർശിപ്പിക്കാറില്ല. **<DT>** ടാഗ് പദങ്ങൾ നിർമിക്കുന്നതിനും **<DD>** ടാഗ് അതിന്റെ നിർവചനങ്ങൾ ചേർക്കുന്നതിനും ഉപയോഗിക്കുന്നു.

ഇന്റർനെറ്റ് സുരക്ഷയുമായി ബന്ധപ്പെട്ട പദങ്ങളും അവയുടെ നിർവചനങ്ങളും പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ് പേജിന്റെ HTML കോഡ് ഉദാഹരണം 5.3-ൽ നൽകിയിരിക്കുന്നു. ഇതിന്റെ ഔട്ട് പുട്ട് ചിത്രം 5.5-ൽ കാണാം.

ഉദാഹരണം 5.3: ഡെഫനിഷൻ ലിസ്റ്റ് നിർമാണം

```
<HTML>
<HEAD> <TITLE> Definition List </TITLE> </HEAD >
<BODY Bgcolor= "#FFE4C4" Leftmargin= "100" Rightmargin= "150">
```

```
    <H2 Align= "center"> Definition List </H2>
      Today cyber security has an immense role in the
      field of Internet. The following are some of
      the threats that affect a computer network.
    <DL>
      <DT>Spam :</DT>
      <DD> Spam is the unsolicited e-mail sent in the
           hope of increasing the sales of some product, or
           just for annoying people.</DD>
      <DT>Phishing :</DT>
      <DD> Phishing is an attempt to acquire information
           such as usernames, passwords and credit card details
           by posting as the original website, mostly that
           of banks and other financial institutions. </DD>
    </DL>
</BODY>
</HTML>
```



ചിത്രം 5.5: ഡെഫനിഷൻ ലിസ്റ്റ് ഉൾക്കൊള്ളുന്ന വെബ് പേജ്

ചിത്രം 5.5-ലെ നിർവചനങ്ങളുടെ ഇൻഡന്റേഷനും, ഇടതും വലതും ഭാഗങ്ങളിലുള്ള മാർജിനുകളും ശ്രദ്ധിക്കുക.

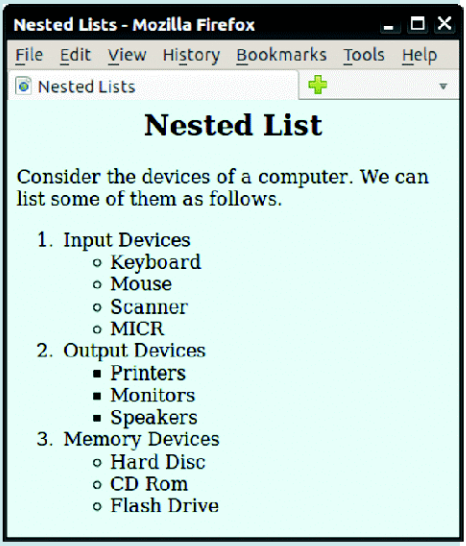
5.1.4 നെസ്റ്റഡ് ലിസ്റ്റുകൾ (Nested lists)

ഒരു ലിസ്റ്റിലെ ഏതെങ്കിലും ഇനത്തിന്റെ താഴെ മറ്റൊരു ലിസ്റ്റ് നൽകുവാനാകും. ഇത്തരം ലിസ്റ്റുകളെ നെസ്റ്റഡ് ലിസ്റ്റ് എന്നറിയപ്പെടുന്നു. ഇത് വിവിധ തരത്തിൽ തയാറാക്കാം.

ഉദാഹരണമായി ഒരു ക്രമലിസ്റ്റിനുള്ളിൽ മറ്റൊരു ക്രമലിസ്റ്റോ, ക്രമരഹിതലിസ്റ്റോ നൽകുവാനാകും. അതുപോലെ ഒരു ക്രമരഹിത ലിസ്റ്റിനുള്ളിൽ മറ്റൊരു ക്രമലിസ്റ്റോ ക്രമരഹിതലിസ്റ്റോ നൽകാനുമാകും. ഉദാഹരണം 5.4 ൽ നൽകുന്ന HTML കോഡ് നെസ്റ്റഡ് ലിസ്റ്റിന്റെ ആശയം വിശദമാക്കുന്നു.

ഉദാഹരണം 5.4: നെസ്റ്റഡ് ലിസ്റ്റ് നിർമാണം

```
<HTML>
<HEAD>
   <TITLE> Nested Lists </TITLE>
</HEAD>
<BODY Bgcolor= "#E0FFFF">
   <H2 Align= "center"> Nested List </H2>
   Consider the devices of a computer.
   We can list some of them as follows.
   <OL>
      <LI> Input Devices </LI>
      <UL>
         <LI>Keyboard</LI>
         <LI>Mouse</LI>
         <LI>Scanner</LI>
         <LI>MICR</LI>
      </UL>
      <LI> Output Devices </LI>
      <UL Type= "Square">
         <LI>Printers</LI>
         <LI>Monitors</LI>
         <LI>Speakers</LI>
      </UL>
      <LI> Memory Devices </LI>
      <UL Type= "Circle">
         <LI>Hard Disc</LI>
         <LI>CD Rom</LI>
         <LI>Flash Drive</LI>
      </UL>
   </OL>
</BODY>
</HTML>
```



ചിത്രം 5.6: നെസ്റ്റഡ് ലിസ്റ്റ് ഉപയോഗിച്ച് ഉപകരണങ്ങളുടെ തരംതിരിവ് പ്രദർശിപ്പിക്കുന്നു

ഉദാഹരണം 5.4 ൽ മൂന്ന് ക്രമരഹിത ലിസ്റ്റുകൾ ഒരു ക്രമലിസ്റ്റിലേക്ക് നെസ്റ്റ് ചെയ്തിരിക്കുന്നു.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. HTML ലെ വിവിധതരം ലിസ്റ്റുകൾ ഏതൊക്കെയാണ്?
2. നിങ്ങളുടെ ക്ലാസിലെ കുട്ടികളുടെ ലിസ്റ്റ് തയാറാക്കുവാൻ അധ്യാപകൻ ആവശ്യപ്പെട്ടു എന്നു കരുതുക. ഏതുതരം ലിസ്റ്റായിരിക്കും നിങ്ങൾ തിരഞ്ഞെടുക്കുന്നത്? എന്തുകൊണ്ട്?
3. <UL>, <OL> എന്നീ ടാഗുകളുടെ പൊതുവായ ആട്രിബ്യൂട്ടുകൾ എന്തൊക്കെയാണ്?
4. <UL>, <OL> എന്നീ ടാഗുകൾ തമ്മിലുള്ള വ്യത്യാസങ്ങൾ എഴുതുക.
5. ഡെഫനിഷൻ ലിസ്റ്റിൽ ഉപയോഗിക്കുന്ന ടാഗുകളുടെ പേര് എഴുതുക.

5.2 ലിങ്കുകളുടെ നിർമാണം (Creating links)

ഒരു വെബ് പേജിനെ മറ്റൊരു വെബ് പേജുമായോ ഒരു വെബ് പേജിന്റെ ഏതെങ്കിലും ഭാഗത്തെ അതേ പേജിലെ മറ്റൊരു ഭാഗവുമായോ ബന്ധിപ്പിക്കുന്നതിന് സഹായിക്കുന്ന ഘടകമാണ് ഹൈപ്പർലിങ്ക്. ഹൈപ്പർലിങ്കിനെ സാധാരണയായി ലിങ്കുകൾ എന്നാണ് വിളിക്കുന്നത്. HTML-ൽ **<A>** ടാഗിനും **</A>** ടാഗിനും ഇടയിൽ എന്ത് നൽകിയാലും അത് ലിങ്കിന്റെ ഭാഗമാകും. ഉപയോക്താവിന് അവിടെ ക്ലിക്ക് ചെയ്തുകൊണ്ട് ലിങ്ക് ചെയ്തിരിക്കുന്ന ഡോക്യുമെന്റിലേക്ക് എത്താനാകും. <A> ടാഗിന്റെ മുഖ്യ ആട്രിബ്യൂട്ടാണ് ഹൈപ്പർ റഫറൻസ് എന്ന് അറിയപ്പെടുന്ന **Href**. ഇതിന്റെ വിലയായി സാധാരണ URL ആണ് നൽകുന്നത് (ഹൈപ്പർലിങ്ക് ബന്ധിപ്പിക്കേണ്ട വെബ് പേജിന്റെ വിലാസം). ഉദാഹരണത്തിന് താഴെകൊടുത്തിരിക്കുന്ന കോഡ് ഭാഗം ശ്രദ്ധിക്കുക.

```
<A Href= "http://www.dhsekerala.gov.in">Higher Secondary</A>
```

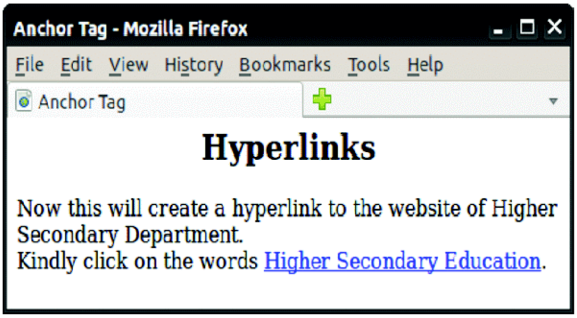
ഇത് "http:// www.dhsekerala.gov.in." എന്ന വെബ്സൈറ്റിനെ ലക്ഷ്യം വയ്ക്കുന്ന ഹൈപ്പർലിങ്ക് സൃഷ്ടിക്കുന്നു. ഉപയോക്താവ് ഈ ലിങ്കിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ ഈ URL-ന്റെ ഹോംപേജ് ബ്രൗസറിൽ ലഭ്യമാകും. <A>, </A> എന്നിവയ്ക്കിടയിൽ നൽകുന്ന വാക്യങ്ങൾ വ്യത്യസ്ത നിറത്തിലും അടിവരയോടുകൂടിയും കാണപ്പെടും.

ഹൈപ്പർലിങ്ക് സൃഷ്ടിക്കുന്ന HTML കോഡ് ഉദാഹരണം 5.5ലും അതിന്റെ ചിത്രം 5.7ലും കാണിച്ചിരിക്കുന്നു.

ഉദാഹരണം 5.5: ഒരു വെബ് പേജിൽ ഹൈപ്പർലിങ്കിന്റെ നിർമാണം

```
<HTML>
<HEAD>
   <TITLE> Anchor Tag </TITLE>
</HEAD>
<BODY Bgcolor= "#FFFFFF">
   <H2 Align= "center"> Hyperlinks </H2>
   <P>Now this will create a hyperlink to the website of
      Higher Secondary Department.<BR>
```

```
        Kindly click on the words
    <A Href= "http://www.dhsekerala.gov.in">Higher Secondary
        Education</A>.
</BODY>
</HTML>
```



ചിത്രം 5.7: ഹൈപ്പർ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ് പേജ്

ചിത്രം 5.7 ൽ കാണിച്ചിരിക്കുന്നതു പോലെ ഹൈപ്പർ ലിങ്ക് ചെയ്ത വാക്യം (Higher Secondary Education) അടിവരയിട്ട രീതിയിലും വ്യത്യസ്തമായ നിറത്തിലും ആയിരിക്കും.

ഹൈപ്പർ ലിങ്കുകളുടെ ലക്ഷ്യസ്ഥാനത്തിനനുസരിച്ച് അവയെ ഇന്റേണൽ അല്ലെങ്കിൽ എക്സ്റ്റേണൽ എന്ന് തരംതിരിച്ചിരിക്കുന്നു.

5.2.1 ഇന്റേണൽ ലിങ്കിങ് (Internal linking)

ഒരു വെബ് പേജിനെ അതിന്റെ തന്നെ മറ്റൊരു ഭാഗവുമായി ബന്ധിപ്പിക്കുന്നത് ഇന്റേണൽ ലിങ്കിങ് എന്നറിയപ്പെടുന്നു. ഇതിനായി **Name** ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് ലിങ്ക് ചെയ്യേണ്ട ഭാഗം തിരിച്ചറിയുവാൻ പ്രത്യേകം നാമകരണം ചെയ്യേണ്ടതുണ്ട്. ഈ പേര് <A> ടാഗിന്റെ ആട്രിബ്യൂട്ടായ Href-ന്റെ വിലയായി നൽകി ലിങ്കിങ് സാധ്യമാകും. ഉദാഹരണത്തിന് "Environment Pollution" എന്ന വിഷയത്തെ സംബന്ധിക്കുന്ന ഒരു വെബ് പേജ് നിർമിക്കണം എന്ന് കരുതുക. ഇതിൽ മൂന്ന് ഖണ്ഡികകളിലായി "Introduction", "Air Pollution", "Water Pollution" എന്നിവ നൽകിയിട്ടുണ്ട്. ഈ ഖണ്ഡികകളെ സൂചിപ്പിക്കുന്നതിനായി <A> ടാഗിന്റെ Name ആട്രിബ്യൂട്ടിന്റെ വിലകളായി Introduction, Air, Water എന്നീ പേരുകൾ നൽകിയിരിക്കുന്നു.

```
<A Name= "Introduction"> INTRODUCTION </A>
<A Name= "Air"> Air Pollution </A>
<A Name= "Water"> Water Pollution </A>
```

വെബ് പേജിന്റെ ഈ ഭാഗങ്ങളിലേക്ക് ബന്ധിപ്പിക്കുന്നതിനായി <A> ടാഗിന്റെ Href ആട്രിബ്യൂട്ടിന് യഥാക്രമം #Introduction, #Air, #Water (# ചിഹ്നം നിർബന്ധമാണ്) എന്നീ വിലകൾ നൽകണം. ഇതിനുള്ള HTML കോഡ് ഭാഗം താഴെക്കൊടുക്കുന്നു.

```
<A Href = "#Introduction"> Go to Introduction </A>
<A Href = "#Air"> Air pollution </A>
<A Href = "#Water"> Water pollution </A>
```

ഇന്റേണൽ ലിങ്കിന്റെ ആശയം ഉൾപ്പെടുന്ന വെബ് പേജ് നമുക്ക് നിർമിക്കാം. ഇതിനുള്ള HTML കോഡ് ഉദാഹരണം 5.6 ലും അതിനനുസൃതമായ വെബ് പേജ് ചിത്രം 5.8 ലും നൽകിയിരിക്കുന്നു.

ഉദാഹരണം 5.6: ഇന്റേണൽ ലിങ്ക് ഉൾപ്പെടുന്ന വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE> Internal Linking </TITLE> </HEAD>
<BODY Bgcolor= "f8f8f8">
   <H2 Align= "center">ENVIRONMENTAL POLLUTION</H2>
   <A Name= "Introduction"><B>INTRODUCTION</B></A>
   <P><FONT Size= "15">E</FONT>nvironment pollution is a
   wide-reaching problem and it is likely to affect the
   health of human population.Here we discuss the environment
   pollution in the perspective of <A HREF= "#Air">air
   pollution </A>,
   <A Href= "#Water"> water pollution </A>and land/soil
   waste pollution. Studies find that these kinds of
   pollutions are not only seriously affecting humans
   but also animals and plants.
   </P>
   <A Name= "Air"><B> Air Pollution</B></A>
   <P>The air we breathe is an essential ingredient for our
   health and wellbeing. Unfortunately polluted air is common
   throughout the world, especially in developed countries.
   </P>
   <A Name= "Water"><B> Water pollution</B></A>
   <P>The water we drink is an essential ingredient for our
   health and wellbeing. Unfortunately polluted water and
   air are common throughout the world.Water pollution is
   caused by the discharge of industrial effluents, sewage
   water and agricultural or household waste.
   </P>
   <A Href= "#Introduction">Go to Introduction </A>
</BODY>
</HTML>
```

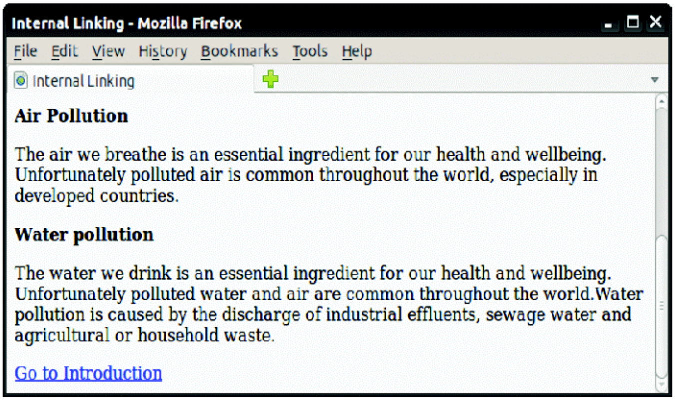
ചിത്രം 5.8 ൽ ഈ വെബ് പേജിന്റെ ഒരു ഭാഗം കാണിച്ചിരിക്കുന്നു. ഉദാഹരണം 5.8 ൽ കാണിച്ചിരിക്കുന്ന air pollution, water pollution എന്നീ ലിങ്കുകളിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ

ചിത്രം 5.8: ഇന്റേണൽ ഹൈപ്പർ ലിങ്കുകൾ ഉൾപ്പെടുന്ന വെബ് പേജ്

ചിത്രം 5.9 ൽ കാണിച്ചിരിക്കുന്ന വെബ് പേജിന്റെ അനുയോജ്യമായ ഭാഗം ദൃശ്യമാകും. ഇതുപോലെ ചിത്രം 5.9 ലെ വെബ് പേജിന്റെ താഴെ ഭാഗത്തുള്ള Go to Introduction എന്ന ലിങ്കിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ Indroduction എന്ന ഭാഗം ദൃശ്യമാകും.

5.2.2 എക്സ്റ്റേണൽ ലിങ്കിങ് (External linking)

ഒരു വെബ് പേജിനെ മറ്റൊരു വെബ് പേജുമായി ബന്ധിപ്പിക്കുന്ന പ്രവർത്തനത്തെ എക്സ്റ്റേണൽ ലിങ്കിങ് എന്ന് പറയുന്നു. നിലവിൽ ലഭ്യമായ വെബ് പേജിലെ <A> ടാഗിന്റെ Href ആട്രിബ്യൂട്ടിന്റെ വിലയായി ബാഹ്യ ഫയലിന്റെ URL നൽകി ഇതു സാധ്യമാക്കാം. ഇതിന്റെ പ്രവർത്തനങ്ങൾ നാം ഈ അധ്യായത്തിലെ ഭാഗം 5.2-ൽ ചർച്ച ചെയ്തതാണ്.



ചിത്രം 5.9: വെബ് പേജിന്റെ ഹൈപ്പർലിങ്ക് ചെയ്യപ്പെട്ട ഭാഗങ്ങൾ

5.2.3 ഡഞതഘ എന്ന ആശയം (Concept of URL)

URL-ന്റെ പൂർണ്ണരൂപം Uniform Resource Locator (യൂണിഫോം റിസോഴ്സ് ലൊക്കേറ്റർ) എന്നാണ്. വെബ് അഡ്രസ് എന്നാണിതിന്റെ അർഥം. റിലേറ്റീവ്, അബ്സല്യൂട്ട് എന്നീ രണ്ടു തരം URL കൾ നിലവിലുണ്ട്.

`<A Href= "http://www.scertkerala.gov.in">` എന്നത് ഒരു അബ്സല്യൂട്ട് URL ആണ്. കാരണം ഇവിടെ വെബ് അഡ്രസ് പൂർണമായും നൽകിയിരിക്കുന്നു. എന്നാൽ ലിങ്ക് `<A Href = "image.html">` എന്നാണ് എഴുതിയതെങ്കിൽ അത് റിലേറ്റീവ് ലിങ്ക് ആയി പരിഗണിക്കും. ഇവിടെ നമ്മൾ നൽകിയ "*image.html*" ഒരു റിലേറ്റീവ് URL ആണ്. എന്തുകൊണ്ടെന്നാൽ *image.html* ഉം വെബ് പേജും ഒരേ ഡയറക്ടറിയിൽ ശേഖരിച്ചിരിക്കുന്നു. നിലവിൽ ലഭ്യമായ വെബ് പേജ് */home/kite/HTML/hyperlink.html* എന്നാണ് ശേഖരിച്ചിരിക്കുന്നതെങ്കിൽ `<A Href = "image.html">` എന്ന ഹൈപ്പർ ലിങ്ക് സൂചിപ്പിക്കുന്നത് */home/kite/HTML* എന്ന ഫോൾഡറിലെ *image.html* എന്ന ഫയലിനെ ആയിരിക്കും.

5.2.4 ഗ്രാഫിക്കൽ ഹൈപ്പർ ലിങ്ക് നിർമാണം (Creating graphical hyperlinks)

ടെക്സ്റ്റുകളെ ഹൈപ്പർ ലിങ്കായി ഉപയോഗിക്കുന്ന രീതി നാം മനസിലാക്കിക്കഴിഞ്ഞു. അതുപോലെ ഇമേജുകളും ഹൈപ്പർ ലിങ്കാക്കി ഉപയോഗിക്കാം. അതിനായി <A>, </A> എന്നീ ടാഗുകളുടെ ഇടയിൽ <IMG> ടാഗ് നൽകിയാൽ മതിയാകും. അതിന്റെ HTML കോഡ് ഉദാഹരണം 5.7ൽ കാണിച്ചിരിക്കുന്നു. ഇതു സംബന്ധിച്ച വിവരങ്ങൾ ചിത്രം 5.10 ലും നൽകിയിരിക്കുന്നു.

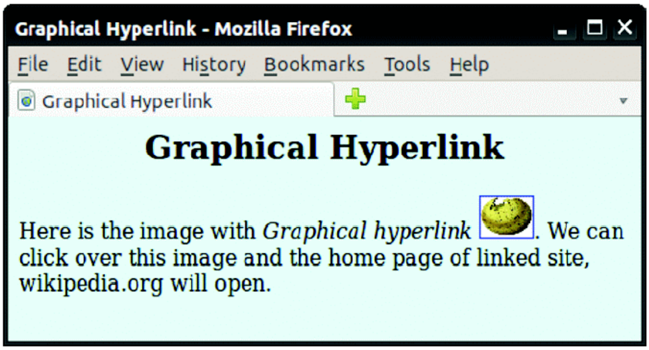
ഉദാഹരണം 5.7: ഗ്രാഫിക്കൽ ഹൈപ്പർ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE> Graphical Hyperlink </TITLE> </HEAD>
<BODY Bgcolor = "#E0FFFF">
   <H2 Align= "center">Graphical Hyperlink</H2>
   Here is the image with <I>Graphical hyperlink </I>
   <A Href= "https://www.wikipedia.org">
   <IMG Src= "wiki.jpg" Alt= "Image of Wiki"
       Height= "30" Width= "40" Border= "1"> </A>.
       We can click over this image and the home page of
       linked site, wikipedia.org will open.
</BODY>
</HTML>
```

നിങ്ങൾ മൗസ് പോയിന്റർ വിക്കിപീഡിയ ലോഗോയുടെ മുകളിൽ കൊണ്ടുവരുമ്പോൾ അത് ചൂണ്ടു വിരൽ ഉയർത്തിയ കൈ അടയാളമായി മാറുന്നു. ഇത് സൂചിപ്പിക്കുന്നത് ഈ ലോഗോ ഒരു ഗ്രാഫിക്കൽ ഹൈപ്പർ ലിങ്ക് ആണെന്നാണ്. അവിടെ ക്ലിക്ക് ചെയ്യുമ്പോൾ ബ്രൗസറിൽ www.wikipedia.org എന്ന വെബ്സൈറ്റ് ലഭ്യമാകും.

5.2.5 ഇ-മെയിൽ ലിങ്ക് നിർമിക്കുന്ന വിധം (Creating e-mail linking)

ഹൈപ്പർലിങ്ക് പ്രോട്ടോക്കോളായ **mailto** ഉപയോഗിച്ച് നമുക്ക് ഇ-മെയിൽ ഹൈപ്പർലിങ്ക് നിർമിക്കുവാനാകും. ഇത് വിശദമാക്കുന്ന HTML കോഡ് ഉദാഹരണം 5.8 ലും വെബ് പേജിന്റെ ചിത്രം 5.11 ലും നൽകിയിട്ടുണ്ട്.



ചിത്രം 5.10: ഗ്രാഫിക്കൽ ഹൈപ്പർ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ് പേജ്

ഉദാഹരണം 5.8: ഇ-മെയിൽ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE> e-mail Linking </TITLE> </HEAD >
<BODY Bgcolor= "#E0FFFF">
   <H2 Align= "center">e-mail linking</H2>
   Now we can create an <B><I>e-mail hyperlink </I></B> to
   SCERT in the following way. Kindly click on the word
   <A Href= mailto: "scertkerala@gmail.com"> SCERT</A> Kerala.
</BODY>
</HTML>
```

ചിത്രം 5.11- ൽ കാണിച്ചിരിക്കുന്ന വെബ് പേജിൽ SCERT എന്ന ലിങ്ക് നൽകിയിട്ടുണ്ട്. ഇതിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ scertkerala @gmail.com എന്ന് അഡ്രസ് ചെയ്യപ്പെട്ട ഒഴിഞ്ഞ മെസ്സേജ്ബോക്സോടുകൂടിയ ഇ-മെയിൽ പ്രോഗ്രാം തുറക്കുന്നു.

ചിത്രം 5.11:ഇ - മെയിൽ ഹൈപ്പർ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ് പേജ്

ഒരു വെബ് പേജിൽ നിന്ന് ഇ-മെയിൽ ലിങ്ക് നൽകി ഇ-മെയിൽ പ്രോഗ്രാം തുറക്കുമ്പോൾ, ഇ-മെയിൽ പ്രോഗ്രാമിലെ എല്ലാ ഭാഗങ്ങൾക്കും തനതു വിലകൾ നൽകാവുന്നതാണ്. ഉദാഹരണമായി "Thank you sir" എന്ന സന്ദേശം "Director SCERT" എന്ന സബ്ജക്ടോടുകൂടി scertkerala@gmail.com എന്ന വിലാസത്തിലേക്കും, കാർബൺ കോപ്പിയായി (CC) scertkerala1@gmail.com എന്ന വിലാസത്തിലേക്കും, bcc ആയി scertkerala2@gmail.com എന്ന വിലാസത്തിലേക്കും ഒരു ഇ-മെയിൽ ലിങ്ക് വഴി അയക്കണമെങ്കിൽ മുകളിൽ കൊടുത്തിരിക്കുന്ന HTML കോഡിലെ <A> ടാഗ് താഴെ പറയുന്ന വിധം മാറ്റി എഴുതിയാൽ മതിയാകും.

```
<A Href="mailto:scertkerala@gmail.com?
cc=scertkerala1@gmail.com&
bcc=scertkerala2@gmail.com
&samp;subject=Director%20SCERT
&samp;bodt=Thank%20You%20Sir">
send E-mail with cc bcc subject and body </A>
```

<A> ടാഗിലെ ? ചിഹ്നം ആദ്യത്തെ വിഭാഗത്തെയും & മറ്റ് വിഭാഗങ്ങളെയും വേർതിരിക്കാൻ ഉപയോഗിക്കുന്നു.

5.3 സംഗീതം, വീഡിയോ എന്നിവ ഉൾപ്പെടുത്തൽ (Inserting music and video)

ഇപ്പോൾ ലഭ്യമാകുന്ന മിക്കവാറും വെബ് പേജുകളിലും ടെക്സ്റ്റുകൾ, ചിത്രങ്ങൾ എന്നിവയ്ക്ക് പുറമെ സംഗീത, വീഡിയോ ശകലങ്ങളും ഉൾപ്പെടുത്തി സമ്പുഷ്ടമാക്കിയിരിക്കുന്നു. എങ്ങനെയാണ് ഒരു വെബ് പേജിൽ മ്യൂസിക്, വീഡിയോ തുടങ്ങിയവ ഉൾപ്പെടുത്തുന്നത്? മൾട്ടിമീഡിയ വിഭവങ്ങൾ ഉൾപ്പെടുത്താൻ ഇൻലൈൻ, എക്സ്റ്റേണൽ എന്നീ രണ്ടു തരം രീതികൾ ഉപയോഗിക്കുന്നു. ഇൻലൈൻ രീതിയിൽ ഫയലുകളും അതിനാവശ്യമായ വിവരങ്ങളും വെബ് പേജിന്റെ ഭാഗമായാണ് കൈകാര്യം ചെയ്യുന്നത്. ബ്രൗസർ ജാലകം തുറക്കുമ്പോൾത്തന്നെ ഇതിലെ സംഗീതവും വീഡിയോ ദൃശ്യവും ഉപയോക്താവിന് കേൾക്കുവാനും കാണുവാനും സാധിക്കും. എന്നാൽ വെബ് പേജിൽ ലഭ്യമാകുന്ന മൾട്ടിമീഡിയ വിഭവങ്ങൾ ആ കമ്പ്യൂട്ടറിൽ ലഭ്യമായ സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് തുറക്കുന്ന രീതിയാണ് എക്സ്റ്റേണൽ രൂപത്തിൽ അവലംബിച്ചിരിക്കുന്നത്. നമുക്ക് .jpg,

.gif, .avi, .png, .tiff, .mp3, .mp4, തുടങ്ങിയ എക്സ്റ്റൻഷനിൽ കാണപ്പെടുന്ന മൾട്ടിമീഡിയ ഫയലുകൾ ഒരു വെബ് പേജിലേക്ക് ലിങ്ക് ചെയ്യുവാനാകും.

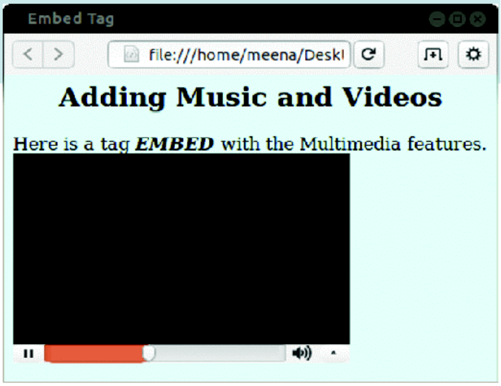
മൾട്ടിമീഡിയ വിഭവങ്ങളായ സംഗീതവും വീഡിയോ ദൃശ്യങ്ങളും ഉൾപ്പെടുത്താൻ HTML ലെ **<EMBED>** എന്ന ടാഗാണ് ഉപയോഗിക്കുന്നത്. വെബ് പേജിൽ ഈ ടാഗ് ഉൾപ്പെടുത്തുമ്പോൾ മൾട്ടിമീഡിയ മാർഗങ്ങളുടെ നിയന്ത്രണത്തിനുവേണ്ട ഉപാധികളും ലഭ്യമാകുന്നു. എന്നാൽ ഈ ടാഗിന്റെ പ്രവർത്തനം പിന്തുണയ്ക്കാത്ത ബ്രൗസറുകളിൽ **<NOEMBED>** എന്ന ടാഗ് ഉപയോഗിച്ച് ബദൽ സംവിധാനം സജ്ജമാക്കാവുന്നതാണ്.

<EMBED> ടാഗിന്റെ പ്രധാന ആട്രിബ്യൂട്ട് **Src** ആണ്. ഇതിന്റെ വിലയായി വെബ് പേജിൽ ഉൾപ്പെടുത്തേണ്ട സംഗീതം/വീഡിയോയുടെ URL നൽകണം. മറ്റ് ആട്രിബ്യൂട്ടുകൾ **Height**, **Width**, **Align**, **Alt**, തുടങ്ങിയവ ആണ്. ഇതിന്റെ വിലകൾ നമുക്ക് പരിചിതമാണല്ലോ? ഇതുകൂടാതെ **Hidden** എന്നൊരു ആട്രിബ്യൂട്ടും ഈ ടാഗിനുണ്ട്. നമ്മൾ വെബ് പേജുകളിൽ ഉൾപ്പെടുത്തുന്ന സംഗീത-വീഡിയോ ശകലങ്ങൾ ഉപയോക്താവിന് കാണണമെങ്കിൽ **Hidden** ആട്രിബ്യൂട്ടിന് തനത് വിലയായ False എന്നും കാണേണ്ട എങ്കിൽ True എന്നും നൽകണം.

ഒരു വെബ് പേജിൽ ഒരു ഓഡിയോ ഉൾപ്പെടുത്തിയതെങ്ങനെയെന്ന് ഉദാഹരണം 5.9 ലും വെബ് പേജിന്റെ ചിത്രം 5.12 ലും നൽകുന്നു.

ഉദാഹരണം 5.9: ഓഡിയോ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD>
     <TITLE> Embed Tag </TITLE>
</HEAD>
<BODY Bgcolor = "#DDFFFF">
   <H2 Align= "center"> Adding Music and Videos </H2>
   Here is a tag <B><I>EMBED </I></B> with the Multimedia
   features.<BR>
   <EMBED Src= "song1.mp3" Width= "300" Height= "60">
   </EMBED>
</BODY>
</HTML>
```



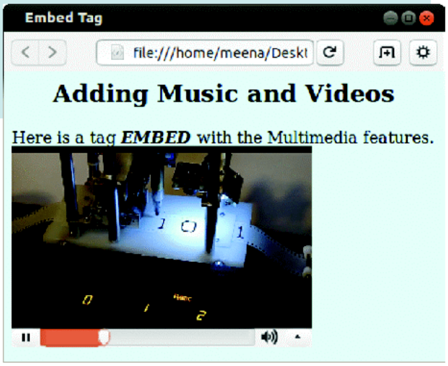
ചിത്രം 5.12: ഓഡിയോ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ് പേജ്

ഉദാഹരണം 5.9 ൽ കൊടുത്തിരിക്കുന്ന HTML പേജ് ബ്രൗസർ വഴി തുറക്കുമ്പോൾ അതിൽ ഉള്ളടക്കം ചെയ്യപ്പെട്ട സംഗീതം കേൾക്കാൻ സാധിക്കുന്നു. ഇതിന്റെ നിയന്ത്രണങ്ങളായ പ്ലേ, പോസ്, ശബ്ദ വ്യതിയാനം എന്നിവയും വെബ് പേജിൽ ലഭ്യമാകും. ഇതുപോലെ വീഡിയോ വെബ് പേജിൽ ലഭ്യമാക്കുന്നതിന് <EMBED> ടാഗും അതിന്റെ ആട്രിബ്യൂട്ടായി **Src** യും ഉപയോഗിക്കാം.

ഉദാഹരണം 5.10 ൽ കൊടുത്തിരിക്കുന്ന HTML കോഡിന് അനുസൃതമായ വെബ് പേജ് (ചിത്രം 5.13) ഈ ലിങ്കിങ് ചിത്രീകരിക്കുന്നു.

ഉദാഹരണം 5.10: ഒരു വെബ് പേജിൽ വീഡിയോ ലഭ്യമാക്കുന്ന HTML കോഡ്

```
<HTML>
<HEAD>
    <TITLE> Embed Tag </TITLE>
</HEAD >
<BODY Bgcolor = "#DDFFFF">
    <H2 Align="center">Adding Music and Videos</H2>
    Here is a tag <B><I>EMBED </I></B>with the Multimedia
    features.
    <EMBED Src= "alan.mp4" Width= "300" Height= "150">
    </EMBED>
    <NOEMBED><IMG Src= "book2.jpg"
          Alt= "Alternative Media">
    </NOEMBED>
</BODY>
</HTML>
```



ചിത്രം 5.13: വീഡിയോ ഉൾക്കൊള്ളുന്ന വെബ് പേജ്

നമ്മൾ ഉദാഹരണം 5.9 ൽ നൽകിയിരിക്കുന്ന HTML പേജ് തുറക്കുമ്പോൾ ഉള്ളടക്കം ചെയ്ത വീഡിയോ പ്ലേ ചെയ്ത് തുടങ്ങും. ഇവിടെയും ശബ്ദം, വീഡിയോ എന്നിവ നിയന്ത്രിക്കാനാവശ്യമായ വോളിയം, പോസ്, ഫുൾ സ്ക്രീൻ മോഡ് എന്നീ കൺട്രോളുകൾ ലഭ്യമാകും. മറ്റ് വെബ് പേജിലുള്ള ഓഡിയോയും വീഡിയോയും നമ്മുടെ വെബ് പേജിലേക്ക് ലിങ്ക് ചെയ്ത് ഉൾപ്പെടുത്താം.

വെബ് പേജിന്റെ പശ്ചാത്തലത്തിൽ ഓഡിയോ ഇൻലൈൻ ആയി മ്യൂസിക് പ്ലേ ചെയ്യുന്നതിന് വേണ്ടി **<BGSOUND>** എന്ന ടാഗ് ഉപയോഗിക്കുന്നു. ഇതിന്റെ ഉദാഹരണം 5.11 ൽ കൊടുത്തിരിക്കുന്നു.

ഉദാഹരണം 5.11: <BGSOUND> ടാഗിന്റെ ഉപയോഗം വ്യക്തമാക്കുന്ന HTML കോഡ്

```
<HTML>
<HEAD> <TITLE> Background Music </TITLE> </HEAD >
    <BODY Bgcolor = "#DDFFFF">
    <H2 Align= "center">Adding Background Music </H2>
    Here is a tag <B><I> BGSOUND </I></B> which helps
    us to play background music in our web page.
    <BGSOUND Src= "Song2.mp3" Loop= "Infinite">
</BODY>
</HTML>
```

ഓഡിയോയുടെ ശബ്ദനിയന്ത്രണത്തിനായി നമുക്ക് <BGSOUND> ടാഗിന്റെ Volume ആട്രിബ്യൂട്ട് ഉപയോഗിക്കാം. HTML പേജിൽ എക്സ്റ്റേണൽ ഫയൽ ഉൾപ്പെടുത്തുന്നതിനായി Src, Href ആട്രിബ്യൂട്ടുകളിൽ അവയുടെ കൃത്യമായ വഴി നൽകിയിരിക്കണം. ഉദാഹരണം 5.10-ൽ ഈ ആട്രിബ്യൂട്ടുകളുടെ വിലകളിൽ മാറ്റം വരുത്തി പരീക്ഷിച്ചു നോക്കാവുന്നതാണ്. Loop ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് മ്യൂസിക് എത്ര തവണ ആവർത്തിക്കണം എന്നുള്ളത് തീരുമാനിക്കാം. ഇതിന്റെ വിലയായി Infinite നൽകിയാൽ വെബ് പേജ് ബ്രൗസറിൽ ലഭ്യമാകുന്നിടത്തോളം തുടർച്ചയായി മ്യൂസിക് പ്ലേ ചെയ്യപ്പെടുന്നതാണ്.

5.4 വെബ് പേജിൽ പട്ടിക ഉൾപ്പെടുത്തുന്ന വിധം (Creating tables in a web page)

ചിലസമയങ്ങളിൽ ധാരാളം വിവരങ്ങൾ നമുക്ക് ചിട്ടയായ രീതിയിൽ പ്രദർശിപ്പിക്കേണ്ടതായിവരും. ഉദാഹരണത്തിന് ഒരു ആശുപത്രിയിൽ 2012-2014 കാലയളവിൽ പുതിയതായി ക്യാൻസർ കണ്ടെത്തിയ രോഗികളിൽ പുകവലി, പാൻ, മദ്യം, ഇതല്ലാതെ മറ്റുകാരണങ്ങളാൽ ക്യാൻസർ ബാധിച്ചവരുടെ എണ്ണം എന്നിവ പട്ടിക രൂപത്തിൽ പട്ടിക 5.1 ൽ നൽകുന്നു.

ഒരു ആശുപത്രിയിൽ 2012 – 2014 കാലയളവിൽ പുതിയതായി ക്യാൻസർ കണ്ടെത്തിയ രോഗികൾ			
Year	2012	2013	2014
Smokers	129	140	143
Pan users	54	56	49
Alcohol users	74	68	77
Other cases	95	93	92

പട്ടിക 5.1: ഒരു ആശുപത്രിയിൽ ക്യാൻസർ കണ്ടെത്തിയ രോഗികൾ

വരിയും നിരയുമായി സെല്ലുകൾ ചിട്ടയായി അടുക്കിവച്ചിരിക്കുന്നതാണ് പട്ടിക എന്നത് നമുക്കെല്ലാം അറിയാം. ഈ അറകളിൽ (cells) ചിത്രങ്ങളോ വാക്യങ്ങളോ മറ്റൊരു ടേബിളോ ഉൾക്കൊള്ളിക്കാം. HTML-ൽ **<TABLE>** എന്ന ടാഗ് ഉപയോഗിച്ചാണ് പട്ടിക നിർമിക്കുന്നത്. ഈ ടാഗിനോടൊപ്പം **<TR>**, **<TH>**, **<TD>** എന്നീ ടാഗുകളും പട്ടിക നിർമിക്കാൻ ഉപയോഗിക്കുന്നു. ഈ ടാഗുകളെക്കുറിച്ച് തുടർന്നുള്ള ഭാഗങ്ങളിൽ ചർച്ച ചെയ്യാം.

5.4.1 <TABLE> ടാഗ്

പട്ടിക ടാഗ് ഒരു കണ്ടെയിനർ ടാഗാണ്. പട്ടിക നിർമാണത്തിനാവശ്യമായ എല്ലാ ഉള്ളടക്കങ്ങളും ടാഗ് ജോഡികളായ **<TABLE>**, **</TABLE>** എന്നിവയുടെ ഇടയിൽ നൽകണം. പട്ടികയുടെ പൊതുവായ രൂപരേഖ തയാറാക്കുന്നതിന് വിവിധ ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിക്കുന്നു. ഇവയിൽ മുഖ്യമായവ താഴെ കൊടുത്തിരിക്കുന്നു.

1. **Border**: പട്ടികയ്ക്ക് ചുറ്റുമുള്ള വരകളുടെ കനമാണ് ഈ ആട്രിബ്യൂട്ടിലൂടെ വ്യക്തമാക്കുന്നത്. ഈ ആട്രിബ്യൂട്ടിന് പൂജ്യമല്ലാത്ത വില (പിക്സലായാണ് നൽകേണ്ടത്) നൽകിയാൽ പട്ടികയുടെ ബോർഡർ നിർമിക്കാം. ഇതിന് പൂജ്യം എന്ന വില നൽകുമ്പോൾ ടേബിളിന് ചുറ്റുമുള്ള വരകൾ അപ്രത്യക്ഷമാകും.
2. **Bordercolor**: പട്ടികയുടെ അതിരുകൾക്ക് നിറം നൽകുന്നു.
3. **Align**: ബ്രൗസർ ജാലകത്തിൽ പട്ടികയുടെ സ്ഥാനം നിർണയിക്കുന്നു. ഇതിന്റെ സാധ്യമായ വിലകൾ left (തനതുവില) right, center എന്നിവയാകുന്നു.
4. **Bgcolor**: പട്ടികക്ക് പശ്ചാത്തലനിറം നൽകുന്നതിന് ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു.
5. **Background:** പട്ടികയുടെ പശ്ചാത്തലത്തിൽ ഒരു ചിത്രം പ്രദർശിപ്പിക്കുന്നതിന് ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഇതിനായി ചിത്രം സൂക്ഷിച്ചിരിക്കുന്ന സ്ഥലം (path) Background ആട്രിബ്യൂട്ടിന്റെ വിലയായി നൽകണം. ഉദാഹരണത്തിന് <TABLE Background = "images/flower.gif"> *images* എന്ന ഫോൾഡറിൽ സൂക്ഷിച്ചിരിക്കുന്ന *flower.jpg* എന്ന ചിത്രം ടേബിളിന്റെ ഉള്ളിൽ പ്രദർശിപ്പിക്കുന്നു. <TABLE > ടാഗിന്റെ ആട്രിബ്യൂട്ടുകളായി Bgcolor, Background എന്നിവ രണ്ടും നൽകിയാൽ Background ആട്രിബ്യൂട്ടിന്റെ വിലയായ ചിത്രമായിരിക്കും പ്രദർശിപ്പിക്കുക.
6. **Cellspacing**: പട്ടികയിലെ അറകൾക്ക് (Cell) ഇടയിൽ സ്ഥലം ഉണ്ട്. ഈ സ്ഥലം കൂട്ടുവാനോ കുറയ്ക്കുവാനോ കഴിയും. Cellspacing എന്ന ആട്രിബ്യൂട്ട് അറകൾക്ക് ഇടയിലുള്ള സ്ഥലം നിർണയിക്കുന്നു. ഇതിന്റെ വിലയായി പിക്സലുകളുടെ എണ്ണമാണ് നൽകുന്നത്.
7. **Cellpadding**: പട്ടികക്കുള്ളിലെ അറകളിൽ അടങ്ങിയിരിക്കുന്ന വസ്തുവും അറകളുടെ അതിർത്തിയും തമ്മിലുള്ള സ്ഥാനവ്യത്യാസം നിർണയിക്കുന്നതിനായി ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. പിക്സലിലാണ് ഇതിന്റെ വിലയും നൽകേണ്ടത്.
8. **Width** and **Height**: പട്ടികയുടെ വീതിയും ഉയരവും ഈ ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിച്ച് നൽകാം. ഇതിന്റെ വിലകൾ പിക്സലിലോ ബ്രൗസർ ജാലകത്തിന്റെ വീതിയുടെയും ഉയരത്തിന്റെയും ശതമാനമായോ നൽകാം.
9. **Frame**: പട്ടികയുടെ അതിരുകൾ (Border) ഏത് രീതിയിൽ പ്രദർശിപ്പിക്കണം എന്നതിനായി ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കാം. ഈ ആട്രിബ്യൂട്ടിന്റെ വിലകൾ പട്ടികാരൂപത്തിൽ പട്ടിക 5.2 ൽ നൽകുന്നു.

വില	വിവരണം
Void	പട്ടികയുടെ അതിരുകൾ പ്രദർശിപ്പിക്കാതിരിക്കാൻ.
Above	മുകൾഭാഗത്തെ മാത്രം അതിര് പ്രദർശിപ്പിക്കുന്നതിന്.
Below	പട്ടികയുടെ താഴ്ഭാഗത്ത് മാത്രം അതിര് പ്രദർശിപ്പിക്കുന്നതിന്.

വില	വിവരണം
Hsides	പട്ടികക്ക് തിരശ്ചീനമായി (മുകളിലും താഴെയും) മാത്രം ബോർഡർ പ്രദർശിപ്പിക്കുന്നതിന്.
lhs or rhs	പട്ടികയുടെ ഇടത് അല്ലെങ്കിൽ വലത് ഭാഗത്ത് മാത്രം ബോർഡർ പ്രദർശിപ്പിക്കുന്നതിന്.
Vsides	പട്ടികക്ക് ലംബമായി ഇടതും വലതും ഭാഗത്ത് മാത്രം ബോർഡർ പ്രദർശിപ്പിക്കുന്നതിന്.
box or border	പട്ടികയുടെ എല്ലാ ഭാഗത്തും ബോർഡർ പ്രദർശിപ്പിക്കുന്നതിന് (തനതു വില)

പട്ടിക 5.2: *Frame* ആട്രിബ്യൂട്ടിന്റെ വിലകൾ

10. **Rules**: അറകളുടെ ചുറ്റുമുള്ള വരകൾ വരയ്ക്കുന്നതിനായി ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഇതിന്റെ വിലകൾ പട്ടിക 5.3-ൽ കാണിച്ചിരിക്കുന്നു.

വില	വിവരണം
none	അറകൾക്ക് (Cell) ചുറ്റും വരകൾ പ്രദർശിപ്പിക്കുന്നില്ല. (border)
cols	നിരകൾക്ക് (Columns) മാത്രം വരകൾ പ്രദർശിപ്പിക്കുന്നതിന്
rows	വരികൾക്ക് (rows) മാത്രം വരകൾ പ്രദർശിപ്പിക്കുന്നതിന്
groups	വരികളുടെ കൂട്ടങ്ങൾക്കും നിരകളുടെ കൂട്ടങ്ങൾക്കും വരകൾ പ്രദർശിപ്പിക്കുന്നതിന്
all	എല്ലാ വരികൾക്കും നിരകൾക്കും ഇടയിൽ വരകൾ പ്രദർശിപ്പിക്കുന്നതിന്

പട്ടിക 5.3: Rules ആട്രിബ്യൂട്ടിന്റെ വിലകൾ

ഇനി <TABLE> ടാഗുമായി ബന്ധപ്പെട്ട മറ്റു ടാഗുകളെക്കുറിച്ച് ചർച്ച ചെയ്യാം.

5.4.2 <TR> ടാഗ്

ഒരു പട്ടിക ടാഗിനുള്ളിൽ വരികൾ (rows) നിർമിക്കുന്നതിനായി <TR> ടാഗ് ഉപയോഗിക്കുന്നു. ഇത് ഒരു കണ്ടയ്നർ ടാഗ് ആണ്. ഓരോ വരിയും <TR> </TR> ടാഗ് ജോഡികൾക്കിടയിൽ ഉൾപ്പെടുത്തുന്നു. <TR> ടാഗ് എല്ലായ്പ്പോഴും <TABLE> ടാഗിനുള്ളിൽ മാത്രമേ ഉപയോഗിക്കാൻ കഴിയൂ. ഓരോ വരിയിലും ഒരു കൂട്ടം അറകൾ (Cells) അടങ്ങിയിരിക്കുന്നു. ഒരു പട്ടികയിലെ ഏറ്റവും ചെറിയഘടകമാണ് ഒരു അറ (Cell). പട്ടികക്കുള്ളിലെ അറകളെ (Cells) ഹെഡിങ്, ഡാറ്റ എന്നിങ്ങനെ രണ്ടായി തരം തിരിക്കാം. പട്ടിക 5.1 ൽ ചുവന്ന നിറത്തിൽ കാണിച്ചിരിക്കുന്ന വിലകൾ ഹെഡിങ് സെല്ലുകളും, നീല നിറത്തിൽ കാണിച്ചിരിക്കുന്നത് ഡാറ്റാസെല്ലുകളും ആണ്.

5.4.3 <TH> ടാഗ്

ഹെഡിങ് സെല്ലുകളെ (Heading cells) നിർവചിക്കുന്നതിനാണ് **<TH>** ടാഗ് ഉപയോഗിക്കുന്നത്. ഇതും ഒരു കണ്ടയ്നർ ടാഗ് ആണ്. **<TH> </TH>** എന്നീ ടാഗുകൾക്കിടയിലാണ് ഹെഡിങ് നൽകുന്നത്. ഈ ഹെഡിങ് കനം കൂട്ടി (Bold face) സെല്ലിന്റെ

മധ്യത്തിൽ (Center) പ്രദർശിപ്പിക്കുന്നു. എല്ലായ്പ്പോഴും <TR> ടാഗിനുള്ളിലാണ് <TH> ടാഗ് നൽകുന്നത്.

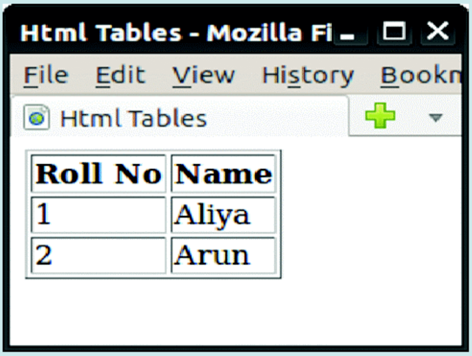
5.4.4 <TD> ടാഗ്

<TH> ടാഗിന് ഏറെക്കുറെ സമാനമായ **<TD>** ടാഗ് ഡാറ്റാസെല്ലുകൾ പ്രദർശിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്നു. ഇത് ഒരു കണ്ടെയ്നർ ടാഗ് ആണ്. <TD> </TD> ടാഗ്ജോഡികൾക്ക് ഇടയിൽ പ്രദർശിപ്പിക്കേണ്ട ഡാറ്റ നൽകുന്നു. ഈ ടാഗും <TH> ടാഗിനെപോലെ <TR> ടാഗിന് ഉള്ളിലായാണ് നൽകുന്നത്.

ഉദാഹരണം 5.12 നൽകിയിരിക്കുന്ന HTML Code ഒരു പട്ടിക നിർമിക്കാൻ ഉപയോഗിക്കുന്നു. ഇതിന്റെ വെബ് പേജ് ചിത്രം 5.14-ൽ കാണിച്ചിരിക്കുന്നു.

ഉദാഹരണം 5.12: ഒരു സാധാരണ പട്ടിക ഉൾക്കൊള്ളുന്ന വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE> Html Tables </TITLE>
</HEAD>
<BODY>
   <TABLE Border="1">
      <TR>
         <TH>Roll No</TH>
         <TH>Name</TH>
      </TR>
      <TR>
         <TD>1</TD>
         <TD>Aliya</TD>
      </TR>
      <TR>
         <TD>2</TD>
         <TD>Arun</TD>
      </TR>
   </TABLE>
</BODY>
</HTML>
```



ചിത്രം 5.14: രണ്ട് നിരകൾ ഉള്ള ഒരു സാധാരണ പട്ടിക ഉൾക്കൊള്ളുന്ന വെബ് പേജ്

മുകളിൽ പ്രതിപാദിച്ച ഏതാനും ചില ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിച്ച് നമുക്ക് താഴെ പറയുന്ന പട്ടിക നിർമിക്കാം.

ഉദാഹരണം 5.13 - ൽ നൽകിയിരിക്കുന്ന HTML Code ഉപയോഗിച്ച് ചിത്രം 5.15 കാണുന്നതുപോലുള്ള വെബ് പേജ് നിർമിക്കാം.

Year	2012 - 14
Smokers	412
Pan users	159
Alcohol users	219
Other cases	280

ഉദാഹരണം 5.13: പട്ടികക്ക് ചുറ്റും ബോർഡറും നിറങ്ങൾ നൽകിയതുമായ പട്ടിക ഉൾക്കൊള്ളുന്ന വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE> Hospital Table </TITLE> </HEAD>
<BODY>
   <TABLE Border= "3" Bordercolor= "RED" Bgcolor= "#4EB0AF"
         Align= "left" Cellspacing= "16" Cellpadding= "5"
         Width= "50%">
      <TR>
         <TH> Year </TH>
         <TH> 2012-14 </TH>
      </TR>
      <TR>
         <TH> Smokers </TH>
         <TD> 412 </TD>
      </TR>
      <TR>
         <TH> Pan users </TH>
         <TD> 159 </TD>
      </TR>
      <TR>
         <TH> Alcohol users </TH>
         <TD> 219 </TD>
      </TR>
      <TR>
         <TH> Other cases </TH>
         <TD> 280 </TD>
      </TR>
   <TABLE>
<BODY>
<HTML>
```

ചിത്രം 5.15: Cellspacing ഉം Cellpadding ഉം ഉപയോഗിച്ചുള്ള പട്ടിക

ഉദാഹരണം 5.13-ൽ <TR>, </TR> ടാഗ് ജോഡികൾക്കിടയിൽ <TH> ടാഗ് ഉപയോഗിക്കുമ്പോൾ ചിത്രം 5.15 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ പട്ടികയുടെ ആദ്യത്തെ നിര ഹെഡർ കോളത്തിന് തുല്യമായിരിക്കും.

<TR> ടാഗിന്റെ ആട്രിബ്യൂട്ടുകൾ

ഒരു വരിയുടെ സ്വഭാവസവിശേഷതകൾ മാറ്റുന്നതിനായി <TR> ടാഗിന്റെ ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിക്കുന്നു.

1. **Align**: അതായത് വരിയിലെ അറകളിലെ വാക്യങ്ങൾ തിരശ്ചീനമായി സ്ഥാപിക്കുന്നതിന് ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. Left, right or center എന്നീ വില

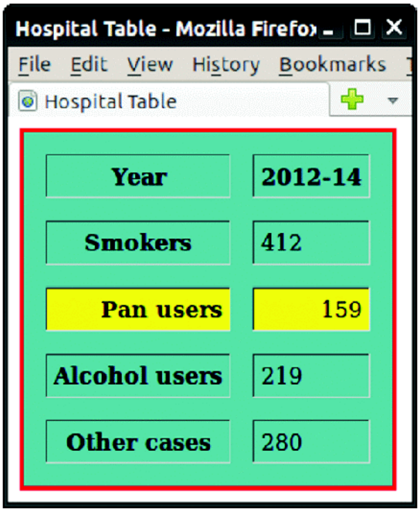
കൾ ഇതിന് നൽകാം. ഈ ആട്രിബ്യൂട്ടിന്റെ തനത് വില ഡാറ്റായെ സംബന്ധിച്ചിടത്തോളം left ഉം ഹെഡിങ്ങിന് center ഉം ആയിരിക്കും (ചിത്രം 5.15 കാണുക).

2. **Valign**: ഏതെങ്കിലും ഒരു വരിയിലെ (row) അറയിലെ ഉള്ളടക്കം ലംബമായി സ്ഥാപിക്കുന്നതിന് Valign ആട്രിബ്യൂട്ട് ഉപയോഗിക്കാം. ഇതിന്റെ വിലയായി top, middle, bottom or baseline എന്നിവ നൽകാം. Baseline ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുമ്പോൾ അറകളിലെ വാക്യങ്ങളെയും ഉള്ളടക്കത്തെയും അതിന്റെ ബേസ് ലൈനോടുകൂടി സ്ഥാപിക്കുന്നു.

3. **Bgcolor:** ഒരു പ്രത്യേക വരിക്ക് പശ്ചാത്തലനിറം നൽകുന്നതിനായി Bgcolor ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഒരു വരിയെ പ്രത്യേകമായി എടുത്ത് കാണിക്കുന്നതിന് ഇത് സഹായിക്കുന്നു.

ഉദാഹരണം 5.13-ൽ നൽകിയ HTML കോഡിന്റെ പരിഷ്കരിച്ച കോഡ് ഭാഗം ചുവടെ ചേർക്കുന്നു. ഇത് ചിത്രം 5.16 ൽ കാണുന്നതു പോലെ പട്ടികയിലെ മൂന്നാമത്തെ വരിയുടെ പശ്ചാത്തല നിറത്തിലും തിരശ്ചീനവും ലംബവുമായ വിന്യാസങ്ങളിലും ഭേദഗതി വരുത്തിയിരിക്കുന്നു.

```
<TR Bgcolor= "yellow" Align= "right" Valign= "middle">
   <TH> Pan users </TH>
   <TD> 159 </TD>
</TR>
```



ചിത്രം 5.16: Bgcolor, Align, Valign എന്നിവ ഉപയോഗിച്ച് പരിഷ്കരിച്ച വരി ഉൾപ്പെടുന്ന പട്ടിക

പരിഷ്കരിച്ച വെബ് പേജ് ചിത്രം 5.16 ൽ കാണിച്ചിരിക്കുന്നു.

<TH>, <TD> ടാഗുകളുടെ ആട്രിബ്യൂട്ടുകൾ

<TH>, <TD> ടാഗുകൾ പട്ടികയിൽ അറകൾ നിർമ്മിക്കുവാൻ ഉപയോഗിക്കുന്നതിനാൽ ഇതിന്റെ മിക്കവാറും ആട്രിബ്യൂട്ടുകൾ ഒരു പോലെയാണ്. ഇതിൽ പ്രധാനപ്പെട്ട ചില ആട്രിബ്യൂട്ടുകൾ നമുക്ക് ചർച്ച ചെയ്യാം.

1. **Align**: അറയിലെ ഉള്ളടക്കത്തിന്റെ തിരശ്ചീനമായ വിന്യാസം സൂചിപ്പിക്കുന്നതിന് വേണ്ടിയാണ് ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. ഈ ആട്രിബ്യൂട്ടിന്റെ വിലയായി left, right, center ഇവയിൽ ഏതെങ്കിലും നൽകാം.

2. **Valign**: അറകളിലെ ഉള്ളടക്കത്തിന്റെ ലംബമായ വിന്യാസം സൂചിപ്പിക്കുന്നതിന് ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഇതിന്റെ വിലയായി top, bottom, middle, baseline എന്നിവ നൽകാം.

3. **Bgcolor**: ഏത് അറയ്ക്കും (cell) ഒരു പ്രത്യേക പശ്ചാത്തലനിറം നൽകുന്നതിനാണ് ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. <TABLE>, <TR>, <TD>/<TH> എന്നിവയ്ക്കെല്ലാം Bgcolor ആട്രിബ്യൂട്ടുകൾ നൽകിയാലും <TH>/<TD> ടാഗിന്റെ

Bgcolor ആട്രിബ്യൂട്ടിന് നൽകുന്ന വിലയായിരിക്കും അറയുടെ പശ്ചാത്തലനിറമായി പ്രദർശിപ്പിക്കുക.

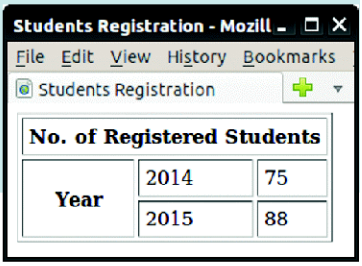
4. **Colspan**: സാധാരണയായി ഒരു അറ ഒരു നിരയിൽ മാത്രം ഒതുങ്ങുന്നതായിരിക്കും. എന്നാൽ ചില അറകളെ ഒന്നിൽ കൂടുതൽ നിരകളിലേക്ക് ചിലപ്പോൾ വ്യാപിപ്പിക്കേണ്ടതായി വന്നേക്കാം. ഒരു അറ എത്ര നിരകളിലായി വ്യാപിച്ചിരിക്കണം എന്നതനുസരിച്ച് Colspan ആട്രിബ്യൂട്ടിന്റെ വിലനൽകാം. ഉദാഹരണം <TH Colspan="3"> എന്നത് ഒരു അറയെ (Cell) മൂന്ന് നിരകളിലായി വ്യാപിപ്പിക്കുന്നു.

5. **Rowspan:** Colspan-നെ പോലെ തന്നെ എത്ര വരികളിലായി (row) ഒരു അറ (cell) വ്യാപിച്ചിരിക്കുന്നു എന്നതിനാണ് ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. ഉദാഹരണത്തിന് <TD Rowspan="4"> എന്ന് നൽകുമ്പോൾ 4 വരികളിലായി ഒരു അറ വ്യാപിക്കുന്നു.

മുകളിൽ പറഞ്ഞ ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിച്ച് ഒരു പട്ടിക നിർമിക്കുന്നതിനുള്ള HTML കോഡ് ഉദാഹരണം 5.14 ൽ നൽകിയിരിക്കുന്നു. വെബ് പേജിന്റെ ചിത്രം 5.17 ൽ നൽകുന്നു.

ഉദാഹരണം 5.14: Colspan, Rowspan എന്നീ ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിച്ച് പട്ടിക നിർമിക്കുന്നു

```
<HTML>
<HEAD> <TITLE> Students Registration </TITLE>
</HEAD>
<BODY>
   <TABLE Border= "1" Cellspacing= "3" Cellpadding= "5">
      <TR>
         <TH Colspan= "3"> No. of Registered Students </TH>
      </TR>
      <TR>
         <TH Rowspan= "2"> Year </TH>
         <TD> 2014 </TD> <TD> 75 </TD>
      </TR>
      <TR>
         <TD> 2015 </TD> <TD> 88 </TD>
   </TR>
   </TABLE>
<BODY>
<HTML>
```



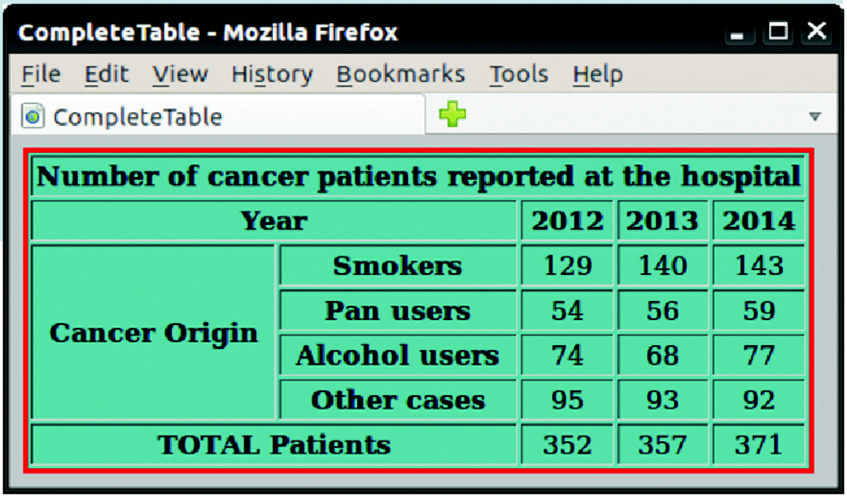
ചിത്രം 5.17: Rowspan, colspan എന്നീ ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിച്ച് പട്ടിക നിർമിക്കുന്നു

പട്ടിക 5.1-ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ക്യാൻസർ രോഗികളുടെ വിശദാംശങ്ങൾ ഉൾക്കൊള്ളുന്ന വെബ് പേജ് നിർമിക്കാം. ഇതിന്റെ HTML കോഡ്, ഉദാഹരണം 5.15 ലും ടേബിളിന്റെ ചിത്രം 5.18 ലും നൽകിയിരിക്കുന്നു.

ഉദാഹരണം 5.15: ക്യാൻസർ രോഗികളുടെ വിവരങ്ങൾ അടങ്ങുന്ന വെബ് പേജ്

```
<HTML>
<HEAD> <TITLE> CompleteTable </TITLE> </HEAD>
<BODY Bgcolor= "silver">
   <TABLE Border= "3" Bordercolor= "red" Bgcolor= "#4EB0AF"
         Align= "left" Cellspacing= "2" Cellpadding= "2"
         Width= "50%">
      <TR>
         <TH Colspan= "5"> Number of cancer patients reported
                           at the hospital </TH>
      </TR>
      <TR Align= "center">
         <TH Colspan= "2"> Year </TH>
         <TH> 2012 </TH>
         <TH> 2013 </TH>
         <TH>  2014 </TH>
      </TR>
      <TR Align= "center">
         <TH Rowspan= "4"> Cancer Origin </TH>
         <TH> Smokers </TH>
         <TD> 129 </TD>
         <TD> 140 </TD>
         <TD> 143 </TD>
      </TR>
      <TR Align= "center">
         <TH> Pan users </TH>
         <TD> 54 </TD>
         <TD> 56 </TD>
         <TD> 59 </TD>
      </TR>
      <TR Align= "center">
         <TH> Alcohol users </TH>
         <TD> 74 </TD>
         <TD> 68 </TD>
         <TD> 77 </TD>
      </TR>
      <TR Align= "center">
```

```
        <TH> Other cases </TH>
        <TD> 95 </TD>
        <TD> 93 </TD>
        <TD> 92 </TD>
     </TR>
     <TR Align= "center">
        <TH Colspan= "2"> TOTAL Patients </TH>
        <TD> 352 </TD>
        <TD> 357 </TD>
        <TD> 371 </TD>
     </TR>
   </TABLE>
</BODY>
</HTML>
```



Number of cancer patients reported at the hospital				
Year		2012	2013	2014
Cancer Origin	Smokers	129	140	143
	Pan users	54	56	59
	Alcohol users	74	68	77
	Other cases	95	93	92
TOTAL Patients		352	357	371

ചിത്രം 5.18: Bgcolor, Rowspan, Colspan, Align, Valign എന്നിവ ഉപയോഗിച്ച് ക്യാൻസർ രോഗികളുടെ വിവരങ്ങൾ പട്ടികപ്പെടുത്തിയ വെബ് പേജ്

നമുക്കു ചെയ്യാം

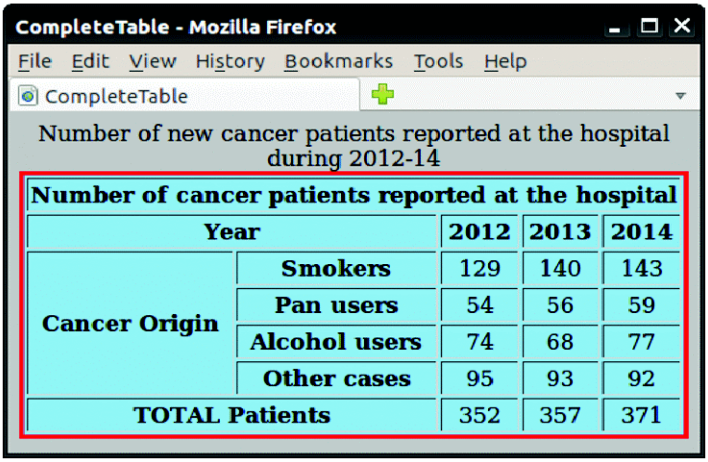
നിങ്ങളുടെ സ്കൂളിലെ വിവിധ രണ്ടാം ഭാഷകൾ പഠിക്കുന്ന കുട്ടികളുടെ എണ്ണം പ്രദർശിപ്പിക്കുന്ന ടേബിൾ നിർമിക്കുക. ഈ ടേബിളിൽ ഹയർ സെക്കന്ററി വിഭാഗത്തിലെ ഓരോ ക്ലാസിലെയും വിവരങ്ങൾ ഉണ്ടായിരിക്കണം.

5.4.5 <CAPTION> ടാഗ് ഉപയോഗിച്ച് ടേബിളിന്റെ ശീർഷകം

<CAPTION> ടാഗ് ഉപയോഗിച്ച് ഒരു പട്ടികയുടെ ശീർഷകം നൽകാം. ഈ ടാഗ് ഉപയോഗിച്ച് ഒരു പട്ടികയെക്കുറിച്ചുള്ള വിവരണം അതിന്റെ ശീർഷകമായി എളുപ്പത്തിൽ നൽകാം. നമുക്ക് ചിത്രം 5.18 ൽ കാണിച്ചിരിക്കുന്ന പട്ടികയ്ക്ക് ഒരു ശീർഷകം നൽകി പരിഷ്കരിക്കണമെന്ന് കരുതുക. ഉദാഹരണം 5.15 ലെ HTML കോഡിലെ **<TR>** ടാഗിന് മുൻപുള്ള ഭാഗത്തിന് താഴെ കൊടുക്കുന്നതുപോലെ മാറ്റം വരുത്താം.

```
<TABLE Border= "3" Bordercolor= "red" Bgcolor= "skyblue"
       Align= "left" Cellspacing= "2" Cellpadding= "2"
       Width= "50%">
<CAPTION>  Number of new cancer patients reported at the
           hospital during 2012-14
</CAPTION>
```

ഇങ്ങനെ ഉണ്ടാക്കിയ പട്ടിക ചിത്രം 5.19 ൽ കാണിച്ചിരിക്കുന്നു.



CompleteTable - Mozilla Firefox

File Edit View History Bookmarks Tools Help

CompleteTable

Number of new cancer patients reported at the hospital during 2012-14

Number of cancer patients reported at the hospital				
Year		2012	2013	2014
Cancer Origin	Smokers	129	140	143
	Pan users	54	56	59
	Alcohol users	74	68	77
	Other cases	95	93	92
TOTAL Patients		352	357	371

ചിത്രം 5.19: ശീർഷകം നൽകി പുതുക്കിയ പട്ടിക

നിങ്ങളുടെ പുരോഗതി അറിയാം

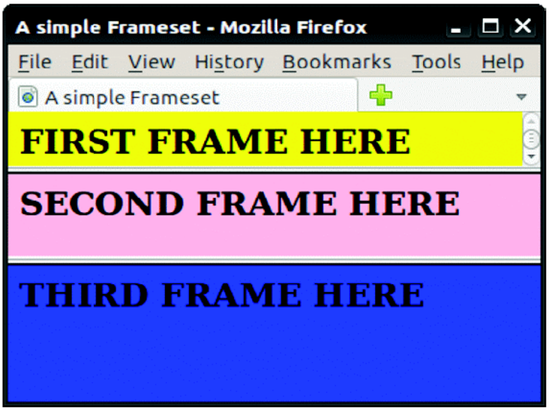
1. <TABLE> ടാഗുമായി ബന്ധപ്പെട്ട രണ്ട് ടാഗുകളുടെ പേരെഴുതുക.
2. താഴെകൊടുത്തിരിക്കുന്നവയിൽ വ്യത്യസ്തമായത് തിരഞ്ഞെടുക്കുക.

   a. TABLE b. TR c. TH d. COLSPAN
3. <TD>, <TH> എന്നീ ടാഗുകളുടെ വ്യത്യാസങ്ങൾ എഴുതുക.
4. ഒരു empty ടാഗാണ് <TABLE> ഈ പ്രസ്താവന ശരിയോ തെറ്റോ എന്ന് എഴുതുക.
5. <TR> ടാഗിന്റെ ഏതെങ്കിലും രണ്ട് ആട്രിബ്യൂട്ടുകൾ എഴുതുക.

5.5 ബ്രൗസർ ജാലകത്തിന്റെ വിഭജനം (Dividing the browser window)

നമുക്ക് ചിലപ്പോൾ ഒരു വെബ് ബ്രൗസർ ജാലകത്തിൽ ഒന്നിൽ കൂടുതൽ വെബ് പേജുകൾ ഉൾപ്പെടുത്തേണ്ടിവരും. ബ്രൗസർ ജാലകത്തെ രണ്ടോ അതിലധികമോ ഭാഗങ്ങളാക്കി തിരിച്ച് ഒരേ സമയം ഓരോ ഭാഗത്തും വ്യത്യസ്തങ്ങളായ വെബ് പേജുകൾ ഉൾക്കൊള്ളിക്കാം. HTML ൽ ബ്രൗസർ ജാലകത്തെ വിവിധ ഭാഗങ്ങളായി വിഭജിക്കുവാൻ **<FRAMESET>** എന്ന ടാഗ് സൗകര്യം നൽകുന്നു. ഇങ്ങനെയുള്ള ഓരോ ഭാഗത്തിലും ഓരോ വെബ് പേജുകൾ ഉൾക്കൊള്ളിക്കാനാവും. ഒരു **<FRAMESET>** ഉപയോ

ഗിച്ച് നിർമിക്കുന്ന ഓരോ വിഭാഗത്തിനേയും ഫ്രെയിം എന്നു വിളിക്കുന്നു. മൂന്ന് ഫ്രെയിമുകൾ ഉൾക്കൊള്ളുന്ന ലളിതമായ ഫ്രെയിം സെറ്റിന്റെ ചിത്രം 5.20-ൽ ചിത്രീകരിച്ചിരിക്കുന്നു. നമുക്ക് **<FRAMESET>**, **<FRAME>** എന്നീ ടാഗുകൾ ഉപയോഗിച്ച് ഒരു ഫ്രെയിം സെറ്റ് നിർമിക്കാം.



ചിത്രം 5.20: മൂന്ന് ഫ്രെയിമുകൾ അടങ്ങിയ ഒരു ഫ്രെയിംസെറ്റ്

5.5.1 <FRAMESET> ടാഗ്

ഒരു കണ്ടെയ്നർ ടാഗായ **<FRAMESET>** ബ്രൗസർ ജാലകത്തെ വിവിധ ഫ്രെയിം ഭാഗങ്ങളായി വിഭജിക്കുന്നു. <FRAMESET> </FRAMESET> എന്നീ ടാഗുകളുടെ ഇടയിലാണ് ഫ്രെയിമുകൾ നിർവചിക്കുന്നത്. ഈ ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്രിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കുന്നു.

1. **Cols**: ഈ ആട്രിബ്യൂട്ട് ഫ്രെയിംസെറ്റ് പേജിലെ ലംബമായ ഫ്രെയിമുകളുടെ എണ്ണവും അവയുടെ അളവുകളും നിർണയിക്കുന്നു. ഫ്രെയിമിന്റെ വീതി മൊത്തം വീതിയുടെ ശതമാനത്തിലോ പിക്സലിന്റെ എണ്ണത്തിലോ നൽകാം. ഉദാഹരണമായി `<FRAMESET Cols = "30%,500,*">` എന്നത് മൂന്ന് ലംബമായ ഫ്രെയിമുകൾ സൃഷ്ടിക്കുന്നു. ഇതിൽ ആദ്യത്തെ ഫ്രെയിം വിൻഡോയുടെ മൊത്തം വീതിയുടെ 30 ശതമാനവും അടുത്തതിന് 500 പിക്സൽ സ്ഥലവും മൂന്നാമത്തെ ഫ്രെയിമിന്റെ അളവ് * ചിഹ്നത്തിനാൽ രേഖപ്പെടുത്തിയിരിക്കുന്നതിനാൽ ആകെ വീതിയുടെ ബാക്കിവരുന്ന സ്ഥലവും ആയിരിക്കും.

2. **Rows**: ഈ ആട്രിബ്യൂട്ട് Cols നെ പോലെ തിരശ്ചീനമായ ഫ്രെയിമുകളുടെ എണ്ണവും അളവും നിർവചിക്കുന്നു.

3. **Border:** ഈ ആട്രിബ്യൂട്ട് ഫ്രെയിമുകളുടെ അതിരിന്റെ (Border) കനം വ്യക്തമാക്കുന്നു. ഇതിന്റെ വില പിക്സലിന്റെ എണ്ണമായിട്ടാണ് നൽകുന്നത്.

4. **Bordercolor:** ഈ ആട്രിബ്യൂട്ട് ഫ്രെയിമുകളുടെ അതിരിന്റെ (border) നിറം നൽകുവാൻ ഉപയോഗിക്കുന്നു.

5.5.2 <FRAME> ടാഗ്

<FRAMESET> ടാഗിനുള്ളിൽ ഫ്രെയിമുകൾ നിർവചിക്കുന്ന എംപ്റ്റി ടാഗാണിത്. <FRAMESET> ടാഗിന് ഉള്ളിലെ ഓരോ ഭാഗത്തിനും തുല്യമായി **<FRAME>** ടാഗ് നൽകണം. ഫ്രെയിമിനകത്ത് HTML പേജ് പ്രദർശിപ്പിക്കുവാൻ ഉപയോഗിക്കുന്ന **Src** ആട്രിബ്യൂട്ട് എപ്പോഴും <FRAME> ടാഗിനൊപ്പം ഉപയോഗിച്ചിരിക്കണം. <FRAME> ടാഗുമായി ബന്ധപ്പെട്ട പ്രധാന ആട്രിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കുന്നു.

1. **Src**: നമ്മൾ മുമ്പ് ചർച്ചചെയ്തതുപോലെ, ഫ്രെയിമിൽ പ്രദർശിപ്പിക്കേണ്ട വെബ് പേജിന്റെ URL സൂചിപ്പിക്കുന്നതിനാണ് Src ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. ഉദാ

ഹരണമായി <FRAME Src = "school.html"> എന്നത് *school.html* എന്ന പേജിനെ ഫ്രെയിമിൽ പ്രദർശിപ്പിക്കുന്നു.

2. **Scrolling**: ഈ ആട്രിബ്യൂട്ട് ഫ്രെയിമിലെ HTML പേജിൽ ലംബവും തിരശ്ചീനവുമായ സ്ക്രോൾബാറുകൾ പ്രദർശിപ്പിക്കണമോ വേണ്ടയോ എന്ന് തീരുമാനിക്കുന്നു. Scrolling ആട്രിബ്യൂട്ടിന് Yes, No, Auto എന്നീ വിലകൾ നൽകാം. Yes എന്ന വില നൽകുമ്പോൾ സ്ക്രോൾ ബാറുകൾ സ്ഥിരമായി പ്രത്യക്ഷപ്പെടും. No എന്ന വില നൽകുമ്പോൾ സ്ക്രോൾ ബാർ ഒന്നും പ്രത്യക്ഷപ്പെടുന്നില്ല. ഇതിന്റെ തനതുവിലയായ Auto നൽകിയാൽ ഫ്രെയിമിന്റെ ഉള്ളടക്കം അതിന്റെ വലുപ്പത്തെക്കാൾ അധികമാകുമ്പോൾ സ്ക്രോൾബാറുകൾ പ്രത്യക്ഷപ്പെടും.
3. **Noresize**: ഉപയോക്താവ് ഒരു പ്രത്യേക ഫ്രെയിമിന്റെ ബോർഡറുകൾ വലിച്ച് നീട്ടാതിരിക്കാൻ ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഈ ആട്രിബ്യൂട്ടിന് വിലകൾ ആവശ്യമില്ല. ഉദാഹരണമായി <FRAME Src= "school.html" Noresize>.
4. **Marginwidth and Marginheight**: ഒരു ഫ്രെയിമിലെ ലംബവും തിരശ്ചീനവുമായ മാർജിനുകൾ (margins) പിക്സലിന്റെ (pixels) എണ്ണത്തിൽ നൽകുന്നതിന് യഥാക്രമം Marginwidth, Marginheight എന്നീ ആട്രിബ്യൂട്ടുകൾ ഉപയോഗിക്കുന്നു.
5. **Name**: ഫ്രെയിമിന് പേര് നൽകുന്നതിന് Name ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. തുടർന്ന് വരുന്ന HTML കോഡിൽ ഫ്രെയിമിനെ സൂചിപ്പിക്കുന്നതിനായി ഈ പേര് ഉപയോഗിക്കാം.

ചിത്രം 5.20 ൽ കാണിച്ചിരിക്കുന്ന ഫ്രെയിം സെറ്റ് നിർമിക്കാൻ ചുവടെ ചേർക്കുന്ന HTML കോഡ് ഉപയോഗിക്കാം. മൂന്ന് HTML പേജുകളായ *sampleframe1.html*, *sampleframe2.html*, *sampleframe3.html* എന്നിവ ഇതിന് മുമ്പേ നിർമിച്ചിട്ടുണ്ട് എന്ന് കരുതുക.

```
<HTML>
<HEAD> <TITLE> A simple Frameset </TITLE> </HEAD>
   <FRAMESET Rows= "20%, 30%, 50%">
      <FRAME Src= "sampleframe1.html">
      <FRAME Src= "sampleframe2.html">
      <FRAME Src= "sampleframe3.html">
   </FRAMESET>
</HTML>
```

5.5.3 ഫ്രെയിമിനെ ടാർഗറ്റ് ചെയ്യുന്നു (Targeting frames)

ഒരു ഫ്രെയിംസെറ്റിലുള്ള ഏതെങ്കിലുമൊരു ഫ്രെയിമിലെ പേജിൽ നൽകുന്ന ഹൈപ്പർ ലിങ്കിലൂടെ ബന്ധിപ്പിക്കപ്പെട്ട മറ്റൊരു പേജ് നാം ഉദ്ദേശിക്കുന്ന ഫ്രെയിമിൽ ദൃശ്യമാക്കാൻ സാധിക്കും. നമ്മൾ ഫ്രെയിമിലെ ലിങ്കിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ ബന്ധപ്പെട്ട പേജ് മറ്റൊരു ഫ്രെയിമിൽ തുറക്കും. ഇതിനുവേണ്ടി ആദ്യം നമ്മൾ ഉദ്ദിഷ്ട ഫ്രെയിമിന് Name ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് പേര് നൽകണം. അതിനുശേഷം ലിങ്കിന്റെ <A> ടാഗിന്റെ Target ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് പേര് നൽകിയ ഫ്രെയിമിനെ സൂചിപ്പിക്കാം.

നമുക്കു ചെയ്യാം

മൂന്ന് HTML ഫയലുകളായ *bio.html,poem.html,fiction.html* എന്നിവ നിർമിച്ച് അതിൽ ആത്മകഥ, കവിത, നോവൽ എന്നീ ഗണത്തിലുള്ള പുസ്തകങ്ങളുടെ ലിസ്റ്റ് തയാറാക്കുക.

ഇപ്പോൾ രണ്ട് ഫ്രെയിമുകൾ ഉൾപ്പെടുന്ന ഒരു വെബ് പേജ് നിർമിക്കാം. ഇതിൽ ഒന്നാമത്തേത് ഈ ഫയലുകളിലേക്കുള്ള ലിങ്കും രണ്ടാമത്തേത് ലിങ്കുമായി ബന്ധപ്പെട്ട വെബ് പേജ് തുറക്കുന്നതിനും ഉപയോഗിക്കുന്നു. ഒരു ഉപയോക്താവ് ഒന്നാമത്തെ ഫ്രെയിമിൽ സൂചിപ്പിച്ച ഏതെങ്കിലും ഒരു ലിങ്കിൽ ക്ലിക്കു ചെയ്യുമ്പോൾ അതിന് അനുബന്ധമായ ഫയൽ രണ്ടാമത്തെ ഫ്രെയിമിൽ തുറക്കുന്നു.
മുകളിൽ പ്രതിപാദിച്ചിരിക്കുന്ന മൂന്ന് ഫയലുകൾ നിർമിച്ചതിനുശേഷം ഉദാഹരണം 5.16 ൽ കൊടുത്തിരിക്കുന്ന HTML കോഡ് ഉപയോഗിച്ചാൽ Target ആട്രിബ്യൂട്ടിന്റെ പ്രവർത്തനത്തെ കുറിച്ച് മനസിലാക്കാം.

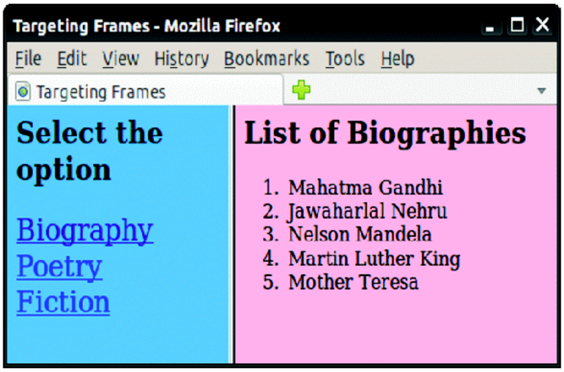
ഉദാഹരണം 5.16: ഫ്രെയിമുകൾ ലക്ഷ്യം വയ്ക്കുന്നതിന്റെ ഉദാഹരണം

താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് ***main.html*** എന്ന ഫയലിൽ സേവ് ചെയ്യുക.

```
<HTML>
<HEAD> <TITLE> Left Frame </TITLE> </HEAD>
<BODY Bgcolor= "#00AFFF" Text= "#282D2F">
   <H2> Select the option </H2>
   <FONT Size= "5">
   <A Href= "bio.html" Target="right_frame">Biography</A><BR>
   <A Href= "poem.html" Target="right_frame">Poetry</A><BR>
   <A Href= "fiction.html" Target="right_frame">Fiction</A>
   </FONT>
</BODY>
</HTML>
```

ഇനി താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് ഒരു ഫയലിൽ സേവ് ചെയ്യാം.

```
<HTML>
<HEAD> <TITLE> Targeting Frames </TITLE> </HEAD>
<FRAMESET Cols= "200, *">
   <FRAME              Src=              "main.html"
          Name= "left_frame">
   <FRAME Name= "right_frame">
</FRAMESET>
</HTML>
```



ചിത്രം 5.21: ഫ്രെയിം ലക്ഷ്യം വയ്ക്കുന്ന വെബ് പേജ്

മുകളിലെ ഡോക്യുമെന്റ് ബ്രൗസറിന് നൽകിയാൽ രണ്ട് ഫ്രെയിമുകൾ ഉൾക്കൊള്ളുന്ന വെബ് പേജ് തുറക്കും. ബ്രൗസർ ജാലകത്തിലെ ആദ്യത്തെ നിരയിൽ *main.html* കാണിക്കുകയും രണ്ടാമത്തെ നിര ഒഴിഞ്ഞുകിടക്കുകയും ചെയ്യും. ആ ഫ്രെയിമിന് വേണ്ടി **Src** ആട്രിബ്യൂട്ട് സൂചിപ്പിച്ചിട്ടില്ല. എന്നാൽ `right_frame` എന്ന പേര് നൽകിയിട്ടുണ്ട് എന്ന കാര്യം ശ്രദ്ധിക്കുക. ഇതിൽ ആദ്യത്തെ ലിങ്കായ Biography തിരഞ്ഞെടുത്താൽ ചിത്രം 5.21 ൽ കാണുന്നതു പോലെ *bio.html* എന്ന ഫയലിന്റെ ഉള്ളടക്കം രണ്ടാമത്തെ ഫ്രെയിമിൽ തുറന്നു വരും. വെബ് പേജിലെ രണ്ടാമത്തെ ഫ്രെയിമിൽ കാണുന്ന *bio.html* എന്ന ഫയലിന്റെ ഉള്ളടക്കം ചിത്രത്തിൽ കാണിച്ചിരിക്കുന്നതുപോലെ ആണെന്ന് കരുതുക. ചിത്രത്തിലെ ലിങ്കുകളുടെ നിറവ്യത്യാസവും ശ്രദ്ധിക്കുക.

main.html എന്ന HTML പേജിൽ മൂന്ന് ജോഡി <A> ടാഗുകൾ മൂന്ന് HTML ഫയലുകളിലേക്ക് ലിങ്ക് ചെയ്യുവാൻ ഉൾപ്പെടുത്തിയിരിക്കുന്നു. ടാഗിന്റെ ആട്രിബ്യൂട്ടായ **Target** ഉപയോഗിച്ച് ലിങ്ക് ചെയ്ത ഫയൽ തുറക്കേണ്ട ഫ്രെയിമിനെ സൂചിപ്പിക്കുന്നു. ഫ്രെയിമിന് പേര് കൊടുക്കേണ്ടതിന്റെ ആവശ്യകത ഇവിടെ സ്പഷ്ടമാകുന്നു.

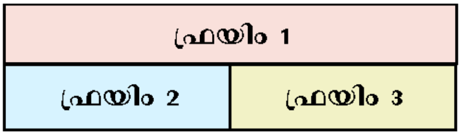
HTML ന്റെ 5-ാം പതിപ്പ് <FRAMESET> നെ പിന്തുണയ്ക്കുന്നില്ല.

ഫ്രെയിംസെറ്റിന്റെ ഉപയോഗത്തിന് എതിരായി ധാരാളം വാദങ്ങൾ ഉയർന്നു വരുന്നു. ഇതിൽ ചിലത് ചുവടെ ചേർക്കുന്നു.

- ഫ്രെയിംസെറ്റിനുള്ളിൽ സാധാരണയായി ബ്രൗസറിന്റെ ബാക്ക്ബട്ടൺ പ്രവർത്തനക്ഷമമല്ല.
- ഒരു ഫ്രെയിം സെറ്റിനുള്ളിൽ ഒരു പ്രത്യേക ഡോക്യുമെന്റ് തുറക്കുവാൻ ബുദ്ധിമുട്ടാണ്.
- ഒരു പ്രത്യേക ഫ്രെയിം റീലോഡ് ചെയ്യാൻ ശ്രമിക്കുമ്പോൾ മൊത്തത്തിലുള്ള ഫ്രെയിംസെറ്റ് റീലോഡ് ചെയ്തേക്കാം. ഇതിന്റെ ഫലമായി ഫ്രെയിമിന്റെ ഉള്ളടക്കങ്ങളെ അതിന്റെ തനതായ ഉറവിടത്തിലേക്ക് റീലോഡ് ചെയ്യും.
- ഫ്രെയിമിലെ ഡോക്യുമെന്റുകളിലൂടെ നാവിഗേറ്റ് ചെയ്യാൻ സർച്ച് എൻജിനുകൾക്ക് ബുദ്ധിമുട്ടാണ്.
- ഒരു ഫ്രെയിംസെറ്റിന്റെ ഉള്ളടക്കം പ്രിന്റ് ചെയ്യുവാൻ ബുദ്ധിമുട്ടാണ്.
- ഒരു ഫ്രെയിം സെറ്റിന്റെ ഉള്ളടക്കം ബുക്ക് മാർക്ക് ചെയ്യുവാൻ ബുദ്ധിമുട്ടാണ്.

5.5.4 ഫ്രെയിംസെറ്റുകളുടെ നെസ്റ്റിങ് (Nesting of framesets)

ചിത്രം 5.20 ലും 5.21 ലും ബ്രൗസർ ജാലകത്തിലെ ഫ്രെയിമുകൾ നമുക്ക് കാണാൻ സാധിക്കും. ആദ്യത്തെ ചിത്രത്തിൽ ബ്രൗസർ ജാലകം തിരശ്ചീനമായ മൂന്ന് ഫ്രെയിമുകളായി തിരിക്കുന്നു. ചിത്രം 5.22-ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ബ്രൗസർ ജാലകത്തെ ലംബമായ രണ്ട് ഫ്രെയിമുകളായി വിഭജിക്കണമെന്ന് കരുതുക. ഇത് കൂട്ടിയിണക്കിയ ഫ്രെയിംസെറ്റുകൾ ഉപയോഗിച്ച് സാധ്യമാകും. ഒരു ഫ്രെയിം



ചിത്രം 5.22: ഫ്രെയിംസെറ്റിന്റെ നെസ്റ്റിങ്

സെറ്റിനുള്ളിൽ മറ്റൊരു ഫ്രെയിം സെറ്റ് കൂട്ടിച്ചേർക്കുന്ന പ്രവർത്തനത്തെ നെസ്റ്റിങ് ഓഫ് ഫ്രെയിംസെറ്റ് എന്നു വിളിക്കുന്നു.

ചിത്രം 5.22-ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഫ്രെയിമുകൾ കിട്ടാനാവശ്യമായ ഫ്രെയിം സെറ്റ് നിർമിക്കുന്നതിനുള്ള വിവിധ ഘട്ടങ്ങൾ ചുവടെ ചേർക്കുന്നു.

1. രണ്ട് വരികളായി വിഭജിക്കുന്ന ആദ്യത്തെ ഫ്രെയിം സെറ്റ് നിർമിക്കുക.

```
<FRAMESET Rows= "85, *">
    <FRAME Src= "sampleframe1.html">
</FRAMESET>
```

ബ്രൗസർ ജാലകത്തെ തിരശ്ചീനമായ രണ്ട് ഫ്രെയിമുകളാക്കി വിഭജിക്കുകയും ആദ്യത്തെ വരി *sampleframe1.html* എന്നതിന് മാറ്റി വയ്ക്കുകയും ചെയ്യുക.

2. രണ്ടാമത്തെ വരി സ്വതന്ത്രമായി ഇരിക്കുകയാണല്ലോ? ഇപ്പോൾ നമ്മൾ ഇതിനെ ലംബമായ രണ്ട് ഫ്രെയിമുകളാക്കി വിഭജിക്കുന്നു. അതിനായി രണ്ടാമത്തെ <FRAME> ടാഗിന് പകരം പുതിയൊരു <FRAMESET> ടാഗ് തുറക്കുന്നു.

```
<FRAMESET Rows= "85, *">
    <FRAME Src= "sampleframe1.html">
    <FRAMESET Cols= "220, *">
    </FRAMESET>
</FRAMESET>
```

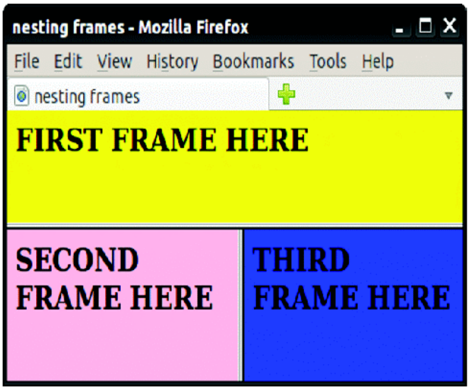
ഈ കോഡ് രണ്ടാമത്തെ വരിയെ രണ്ട് നിരയായി ഭാഗിക്കുന്നു. ഇപ്പോൾ അകത്തെ ഫ്രെയിം സെറ്റിനുള്ളിൽ നമുക്ക് രണ്ട് <FRAME> ടാഗുകൾ കൂട്ടിച്ചേർത്ത് HTML കോഡ് പൂർത്തിയാക്കാം. ഇതിന്റെ ഉദാഹരണം 5.17 ൽ കാണിച്ചിരിക്കുന്നു. ഇതിന്റെ വെബ് പേജിന്റെ ചിത്രം 5.23 ൽ കാണിക്കുന്നു. ഇതിൽ ഉപയോഗിച്ചിരിക്കുന്ന മൂന്ന് HTML പേജുകൾ ചിത്രം 5.17 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ നേരത്തേ തന്നെ നിർമിച്ചിട്ടുണ്ട് എന്ന് കരുതുക.

ഉദാഹരണം 5.17: നെസ്റ്റഡ് ഫ്രെയിം സെറ്റിന്റെ ആശയം വിശദമാക്കുന്നു

```
<HTML>
    <HEAD>
        <TITLE> nesting frames </TITLE>
    </HEAD>
    <FRAMESET Rows= "85, *">
        <FRAME Src= "sampleframe1.html">
        <FRAMESET Cols= "200, *">
            <FRAME Src= "sampleframe2.html">
            <FRAME Src= "sampleframe3.html">
        </FRAMESET>
    </FRAMESET>
</HTML>
```

5.5.5 <NOFRAMES> ടാഗ്

പഴയ ചില ബ്രൗസറുകൾ ഫ്രെയിമുകളെ പിന്തുണയ്ക്കുന്നില്ല. ഈ സാഹചര്യം ബ്രൗസറിന് ഉപയോക്താവിനെ അറിയിക്കേണ്ടതുണ്ട്. ഇതിനായി **<NOFRAMES> </NOFRAMES>** ടാഗ് ജോഡികൾ ഉപയോഗിച്ച് ഫ്രെയിംസ് പിന്തുണയ്ക്കാത്ത ബ്രൗസർ ജാലകത്തിൽ ചില വിവരങ്ങൾ പ്രദർശിപ്പിക്കാനാവും. താഴെകൊടുത്തിരിക്കുന്ന കോഡ് <NOFRAMES> ടാഗിന്റെ ഉപയോഗം വ്യക്തമാക്കുന്നു.



ചിത്രം 5.23: നെസ്റ്റഡ് ഫ്രെയിം സെറ്റ്

```
<HTML>
    <HEAD> <TITLE> A simple Frameset </TITLE>
    </HEAD>
    <FRAMESET Rows= "20%, 30%, 50%">
        <FRAME Src= "sampleframe1.html">
        <FRAME Src= "sampleframe2.html">
        <FRAME Src= "sampleframe3.html">
    </FRAMESET>
    <NOFRAMES>
        <P> Your browser doesnot support frames.<BR>
            Click <A Href="index.htm">here... </A></P>
    </NOFRAMES>
</HTML>
```

ബ്രൗസർ <FRAMESET> ടാഗ് പിന്തുണയ്ക്കുന്നില്ലെങ്കിൽ, പകരമായി "Click here..." എന്ന സന്ദേശത്തോട് കൂടി *index.html* ലേക്കുള്ള ലിങ്ക് ലഭ്യമാകും.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. <FRAMESET Rows="100,*"> എന്നത് ബ്രൗസർ ജാലകത്തെ .... ഭാഗങ്ങളാക്കി വിഭജിക്കുന്നു.
2. <FRAME> ടാഗിന്റെ ഏതെങ്കിലും മൂന്ന് ആട്രിബ്യൂട്ടുകൾ എഴുതുക.
3. നെസ്റ്റഡ് ഫ്രെയിംസെറ്റ് എന്നാൽ എന്ത്?
4. <NOFRAME> ടാഗിന്റെ ഉപയോഗം എന്താണ്?
5. ഫ്രെയിംസെറ്റ് പേജിന് <BODY> ഭാഗത്തിന്റെ ആവശ്യകത ഇല്ല. ശരിയോ തെറ്റോ?

5.6 വെബ് പേജുകളിലെ ഫോമുകൾ (Forms in web pages)

ഉപയോക്താവിൽ നിന്ന് ഡാറ്റ ശേഖരിച്ച് അവയിൽ വിവിധ പ്രവർത്തനങ്ങൾ നടത്തുവാൻ HTML ഫോമുകൾ ആവശ്യമാണ്. ഉദാഹരണമായി, കേരളത്തിൽ ഹയർ സെക്കന്ററി വിദ്യാഭ്യാസത്തിനായി XI-ാം ക്ലാസിലെ പ്രവേശനത്തിന് www.hscap.kerala.gov.in എന്ന വെബ്സൈറ്റിൽ നിങ്ങളുടെ പേര്, SSLC രജിസ്റ്റർ നമ്പർ, ഗ്രേഡുകൾ, ആവശ്യമുള്ള വിഷയങ്ങൾ, സ്കൂൾ തുടങ്ങിയ വിവരങ്ങൾ നൽകേണ്ടതുണ്ട്. ഇതിനായി നമുക്ക് HTML ഫോം ഉപയോഗിക്കാം.

ഉപയോക്താവിൽ നിന്ന് വിവരങ്ങൾ ഒരു HTML ഫോമിലൂടെ സ്വീകരിച്ച് ബാക്ക് -എൻഡ് ആപ്ലിക്കേഷനുകളായ കോമൺഗേറ്റ്വേ ഇന്റർഫെസ് (CGI) ആക്ടീവ് സെർവർ പേജുകൾ (ASP), PHP തുടങ്ങിയവയിലേക്ക് ഡാറ്റാപ്രോസസിങ് നടത്തുവാനായി അയയ്ക്കുന്നു. 7-ാമത്തെ അധ്യായത്തിൽ ഈ സാങ്കേതിക വിദ്യയെക്കുറിച്ച് കൂടുതലായി പഠിക്കാം. ഒരു HTML ഫോമിന് രണ്ട് ഘടകങ്ങൾ ഉണ്ട്: **<FORM>** കണ്ടെയ്നറും, അവയ്ക്കുള്ളിൽ സ്ഥാപിക്കാവുന്ന വിവിധ കൺട്രോളുകളും. ടെക്സ്റ്റ് ബോക്സ്, ലിസ്റ്റ്, ടെക്സ്റ്റ് ഏരിയ, ഡ്രോപ്-ഡൗൺ മെനുകൾ, റേഡിയോ ബട്ടണുകൾ, ചെക്ക് ബോക്സ് തുടങ്ങിയവ വിവിധ തരത്തിലുള്ള ഫോം കൺട്രോളുകളാണ്.

5.6.1 <FORM> ടാഗ്

ഫോം നിർമിക്കുന്നതിനുള്ള കണ്ടെയ്നർ ആയി **<FORM>** ടാഗ് പ്രവർത്തിക്കുന്നു. **<FORM>** ൽ ആരംഭിക്കുന്നതും **</FORM>** ൽ അവസാനിക്കുന്നതുമായ ടാഗ് ജോഡി ഉപയോഗിച്ചാണ് ഫോം നിർമിക്കുന്നത്. ഒരു വെബ് ബ്രൗസറിന് ഫോമിൽ കൂടി മാത്രമേ വിവരങ്ങൾ ശേഖരിക്കുവാൻ കഴിയൂ. ശേഖരിച്ച ഡാറ്റ കൈകാര്യം ചെയ്യാൻ ചില ബാക്ക്-എൻഡ് ആപ്ലിക്കേഷനുകൾ ഉപയോഗിക്കണം. ഇതിനായി ഫോം ഹാൻഡ്ലറുകളായ (Form handler) CGI, JavaScript, PHP എന്നിവ ഉപയോഗിക്കാം. ഒരു ഫോമിൽ നിന്ന് ശേഖരിച്ച എല്ലാ ഇൻപുട്ടുകളും ഒരേ ഫോം ഹാൻഡ്ലറിൽ കൂടി മാത്രമേ പ്രോസസ് ചെയ്യാൻ കഴിയൂ. ഒരു ഫോമിൽക്കൂടി അയച്ച ഡാറ്റ നിയന്ത്രിക്കുന്ന വെബ് സർവറിലെ ഒരു പ്രോഗ്രാമാണ് ഫോം ഹാൻഡ്ലർ. <FORM> ടാഗിലെ Action ആട്രിബ്യൂട്ടിന്റെ വിലയായി ഫോം ഹാൻഡ്ലെറിന്റെ പേര് നൽകുന്നു. സർവർ, അനുബന്ധസാങ്കേതിക വിദ്യകൾ എന്നിവയുടെ ആശയം 7-ാമത്തെ അധ്യായത്തിൽ ചർച്ചചെയ്യാം.

<FORM> ടാഗിൽ സാധാരണയായി ഉപയോഗിക്കുന്ന ആട്രിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കുന്നു.

1. **Action**: ഈ ആട്രിബ്യൂട്ടിന്റെ വിലയായി ഡാറ്റ സ്വീകരിക്കുവാനും പ്രോസസ് ചെയ്യുവാനും കഴിയുന്ന ഫോം ഹാൻഡ്ലറിന്റെ URL നൽകുന്നു.
2. **Method**: ഈ ആട്രിബ്യൂട്ടിന്റെ വിലയായി ഡാറ്റ അപ്ലോഡ് (upload) ചെയ്യുന്ന രീതി സൂചിപ്പിക്കുന്നു. സാധാരണയായി `Get` ഉം `Post` ഉം രീതികളാണ് അവലംബിക്കുന്നത്.
3. **Target**: ഈ ആട്രിബ്യൂട്ട് സ്ക്രിപ്റ്റിന്റെ ഫലം പ്രദർശിപ്പിക്കേണ്ട വിൻഡോയോ ഫ്രെയിമോ സൂചിപ്പിക്കുന്നു. ഇതിന്റെ വിലകൾ പട്ടിക 5.4-ൽ നൽകുന്നു `_blank`, `_self`, `_parent`.

വില	വിവരണം
_blank	ലിങ്ക് ചെയ്യപ്പെട്ട ഡോക്യുമെന്റ് പുതിയ ജാലകത്തിൽ തുറക്കുന്നു.
_self	ലിങ്ക് ചെയ്യപ്പെട്ട ഡോക്യുമെന്റ് ലിങ്കുള്ള അതേ ഫ്രെയിമിൽ തുറക്കുന്നു.
_parent	ലിങ്ക് ചെയ്യപ്പെട്ട ഡോക്യുമെന്റ് പേരന്റ് ഫ്രെയിംസെറ്റിൽ തുറക്കുന്നു.
_top	മുഖ്യ ബ്രൗസർ ജാലകത്തിൽ നിലവിലുള്ള ഫ്രെയിമിനെ മാറ്റി ലിങ്ക് ചെയ്യപ്പെട്ട ഡോക്യുമെന്റ് തുറക്കുന്നു.
name	പേര് ഉപയോഗിച്ച് സൂചിപ്പിച്ചിരിക്കുന്ന ജാലകത്തിൽ ലിങ്ക് ചെയ്യപ്പെട്ട ഡോക്യുമെന്റ് തുറക്കുന്നു.

പട്ടിക 5.4: Target ആട്രിബ്യൂട്ടിന്റെ വിലകൾ

ബ്രൗസറിൽ നിന്ന് വെബ്സെർവറിലെ ഫോം ഹാൻഡ്ലറിലേക്ക് ഫോം സമർപ്പിക്കുന്നതിന് മുമ്പ് ഉപയോക്താവ് ഫോമിൽ എല്ലാ അവശ്യവിവരങ്ങളും നൽകിയിട്ടുണ്ടോ എന്നും അല്ലെങ്കിൽ സ്വീകരിച്ച വിവരങ്ങൾ നിലനിൽക്കുന്നതാണോ അല്ലയോ എന്നും പരിശോധിക്കുന്നതിനാണ് ജാവാ സ്ക്രിപ്റ്റ് അല്ലെങ്കിൽ മറ്റേതെങ്കിലും ക്ലയന്റ്സൈഡ് സ്ക്രിപ്റ്റിങ് ഭാഷ സാധാരണയായി ഉപയോഗിക്കുന്നത്.

ഫോം കൺട്രോളുകൾ (Form controls)

HTML ഫോമിൽ നിന്ന് ഡാറ്റ ശേഖരിക്കാൻ വിവിധതരം ഫോം കൺട്രോളുകൾ ഉപയോഗിക്കാം. ഇവയിൽ ടെക്സ്റ്റ് ബോക്സ്, പാസ്‌വേർഡ്, ചെക്ക് ബോക്സ്, റേഡിയോ ബട്ടൺ, ടെക്സ്റ്റ് ഏരിയ, സെലക്ട് ബോക്സ്, സബ്മിറ്റ്, റീസെറ്റ് ബട്ടൺ തുടങ്ങിയവ ഉൾപ്പെടുന്നു. ഇതിലെ മിക്ക കൺട്രോളുകളും **<INPUT>** ടാഗ് വഴി ഫോമിൽ ഉൾപ്പെടുത്താം.

5.6.2 <INPUT> ടാഗ്

ഉപയോക്താക്കളിൽ നിന്ന് ഡാറ്റ സ്വീകരിക്കാൻ ഫോമിൽ വിവിധതരം കൺട്രോളുകൾ പ്രദർശിപ്പിക്കും. ഇൻപുട്ടിന്റെ തരം അനുസരിച്ച് വിവിധതരത്തിലുള്ള കൺട്രോളുകളായ ടെക്സ്റ്റ് ബോക്സ്, റേഡിയോ ബട്ടൺ, സബ്മിറ്റ് ബട്ടൺ തുടങ്ങിയവ **<INPUT>** ടാഗ് ഉപയോഗിച്ച് നിർമിക്കാം. ഇതിന്റെ **Type** ആട്രിബ്യൂട്ട് കൺട്രോളിന്റെ തരം തീരുമാനിക്കുന്നു.

<INPUT> ടാഗിന്റെ ആട്രിബ്യൂട്ടുകൾ

1. **Type**: <INPUT> ടാഗ് നിർമിക്കുന്ന കൺട്രോളിന്റെ തരം നിർണയിക്കുന്നത് ഈ ആട്രിബ്യൂട്ടാണ്. ഇതിന്റെ പ്രധാനപ്പെട്ട വിലകൾ പട്ടിക 5.5ൽ കൊടുത്തിരിക്കുന്നു.

വില	വിവരണം
Text	ടെക്സ്റ്റ് ബോക്സ് നിർമിക്കുന്നു.
Password	ടെക്സ്റ്റ് ബോക്സ് തന്നെയാണ് ഇതും നിർമിക്കുന്നത് എന്നാൽ ഇതിൽ നൽകുന്ന അക്ഷരങ്ങൾ എല്ലാം അസ്റ്ററിസ്ക് (*) പോലുള്ള ചിഹ്നങ്ങളാണ് പ്രദർശിപ്പിക്കുക.
Checkbox	അതെ, അല്ല എന്നീ വിലകൾ നൽകുന്നതിന് നിർമിച്ചിരിക്കുന്നു. ഒരു കൂട്ടം വിലകളിൽ ഒന്നോ അതിലധികമോ തിരഞ്ഞെടുക്കാൻ ഉപയോഗിക്കുന്നു.
Radio	ഒരു കൂട്ടം വിലകളിൽ നിന്ന് ഒരു വില തിരഞ്ഞെടുക്കുവാൻ ഇത് ഉപയോഗിക്കുന്നു. Name ആട്രിബ്യൂട്ടിന് ഒരേ വിലയുള്ള ഒന്നിലധികം റേഡിയോ ബട്ടണിൽ നിന്ന് ഒന്നുമാത്രമേ ഒരുസമയത്ത് തിരഞ്ഞെടുക്കുവാൻ കഴിയൂ. ഉപയോക്താവ് സെലക്ഷൻ മാറ്റുമ്പോൾ നേരത്തേ തിരഞ്ഞെടുത്തത് ഡിസെലക്ട് ചെയ്യപ്പെടും.
Reset	ഒരു ഫോമിലെ രേഖപ്പെടുത്തലുകളെല്ലാം ഒഴിവാക്കി ഫോമിനെ പ്രാരംഭഘട്ടത്തിലേതുപോലെ ആക്കുന്നതിന് ഉപയോഗിക്കുന്ന പ്രത്യേക ബട്ടൺ ആണിത്.
Submit	ഫോമിൽ നൽകിയ ഡാറ്റാ സെർവറിലേക്ക് സമർപ്പിക്കുന്നതിനായി ഉപയോഗിക്കുന്ന പ്രത്യേക ബട്ടണാണിത്.
Button	ഫോമിൽ ഒരു ഗ്രാഫിക്കൽ ബട്ടൺ നിർമിക്കുവാൻ ഉപയോഗിക്കുന്നു. ഈ ബട്ടണിൽ ക്ലിക്ക് ചെയ്ത് ഫങ്ഷനുകളെ വിളിക്കാം.

പട്ടിക 5.5: Type ആട്രിബ്യൂട്ടിന്റെ വിവിധ വിലകൾ

2. **Name**: ഇൻപുട്ട് കൺട്രോളുകളുടെ പേര് നൽകുന്നു. ഫോം സമർപ്പിക്കുമ്പോൾ ഡാറ്റായും അത് ഉൾക്കൊള്ളുന്ന കൺട്രോളിന്റെ പേരും സർവറിലേക്ക് അയയ്ക്കുന്നു.

3. **Value**: ഒരു കൺട്രോളിന് തനത് വില നൽകുന്നതിന് ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു.

4. **Size**: ഇൻപുട്ട് ടൈപ്പായ ടെക്സ്റ്റ് ബോക്സ്, പാസ്‌വേർഡ് ടെക്സ്റ്റ് എന്നിവയുടെ വലുപ്പം നിശ്ചയിക്കുന്നതിന് ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു ഇൻപുട്ടുകളായ ടെക്സ്റ്റ്, പാസ്‌വേർഡ് എന്നിവയ്ക്ക് മാത്രം ഇത് ബാധകമാകും.

5. **Maxlength**: ടെക്സ്റ്റ് ബോക്സ്, പാസ്‌വേർഡ് എന്നിവയിൽ ഉപയോക്താവിന് ഇൻപുട്ട് ചെയ്യുവാൻ സാധിക്കുന്ന പരമാവധി അക്ഷരങ്ങളുടെ എണ്ണം നിശ്ചയിക്കുവാൻ ഈ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഈ ആട്രിബ്യൂട്ട് ടെക്സ്റ്റ്ബോക്സ്, പാസ്‌വേർഡ് എന്നിവയുടെ കൂടെ മാത്രമേ ഉപയോഗിക്കൂ.

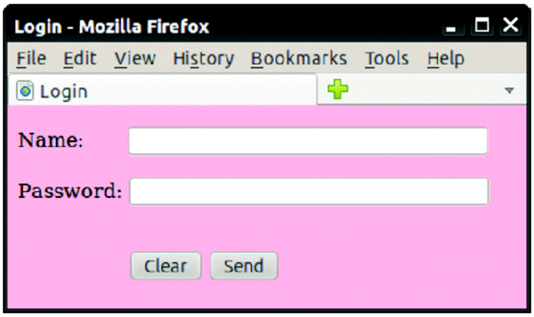
പാസ്‌വേർഡ് ഫീൽഡിലെ ഉള്ളടക്കങ്ങൾ ഉപയോക്താവിന്റെ കാഴ്ചയിൽ നിന്ന് മറയ്ക്കുന്നു. ഇതിൽ നമ്മൾ നൽകുന്ന വിവരങ്ങൾ രഹസ്യകോഡ് ആക്കിമാറ്റുകയോ കൂട്ടിക്കലർത്തുകയോ ചെയ്യുന്നില്ല.

ഒരു HTML ഫോമിൽക്കൂടി പേര്, പാസ്‌വേഡ് എന്നിവ നൽകുന്നതിനുള്ള HTML കോഡ് ഉദാഹരണം 5.18 ലും വെബ് പേജിന്റെ ചിത്രം 5.24 ലും കാണിക്കുന്നു.

ഉദാഹരണം 5.18 : HTML ഫോം നിർമാണം

```
<HTML>
<HEAD> <TITLE> Login </TITLE> </HEAD>
<BODY Bgcolor= "Pink">
   <FORM Action= "login.php" Method= "post">
      <P> Name:    
      <INPUT Type= "text" Name= "firstname" Size= "30"
            Maxlength= "25"> </P>
      <P>Password:
      <INPUT Type= "password" Name= "Psswd" Size= "30"
            Maxlength= "25"> </P> <BR>
                     
       
      <INPUT Type= "reset" Value= "Clear">
      <INPUT Type= "submit" Value= "Send">
   </FORM>
</BODY>
</HTML>
```

ഉദാഹരണം 5.18 ൽ <FORM> ടാഗിലെ Action ആട്രിബ്യൂട്ടിന്റെ വിലയായി *login. php* നൽകുന്നു. ഇതിനർഥം ഫോമിൽക്കൂടി നൽകപ്പെടുന്ന ഡാറ്റ *login.php* എന്ന ഫയലിൽ ശേഖരിച്ചിരിക്കുന്ന ആപ്ലിക്കേഷനിലൂടെ പ്രോസസ് ചെയ്യപ്പെടുന്നു എന്നാണ്. ` ` എന്ന HTML അംഗത്തിന്റെ ഉപയോഗം ശ്രദ്ധിക്കുക. ഇത് അഭാജ്യമായ സ്ഥലം നൽകുവാനാണ് ഉപയോഗിക്കുന്നത്.



ചിത്രം 5.24: ടെക്സ്റ്റ്/പാസ്‌വേർഡ് എന്നിവ ഉൾപ്പെട്ട HTML ഫോം

ഒരു റേഡിയോ ബട്ടൺ (ഓപ്ഷൻ ബട്ടൺ), ചെക്ക് ബോക്സ് എന്നിവയിലൂടെ ഡാറ്റ ഇൻപുട്ട് ചെയ്യുന്ന HTML ഫോം രൂപകൽപന ചെയ്യാം. ഒന്നിൽ കൂടുതൽ വിലകളിൽ നിന്ന് ഒരു വില മാത്രം തിരഞ്ഞെടുക്കാൻ റേഡിയോ ബട്ടണും. ഒന്നിലധികം വിലകൾ തിരഞ്ഞെടുക്കുവാൻ ചെക്ക് ബോക്സും ഉപയോഗിക്കുന്നു. സാധാരണയായി ഒരു ലിസ്റ്റിലെ ഒന്നോ അതിലധികമോ ഇനങ്ങളെ മാർക്ക് ചെയ്യുന്ന രീതിയിലാണ് ചെക്ക് ബോക്സ് ഉപയോഗിക്കുന്നത്.

ഉദാഹരണം 5.19 ൽ നൽകിയിരിക്കുന്ന HTML കോഡ് ഒരു വ്യക്തിയുടെ ലിംഗഭേദവും വിനോദങ്ങളും യഥാക്രമം റേഡിയോ ബട്ടൺ, ചെക്ക് ബോക്സ് എന്നിവയിലൂടെ സൂചിപ്പിക്കുന്നു. ഇതിനനുസൃതമായ വെബ് പേജ് ചിത്രം 5.25 ൽ കാണിക്കുന്നു.

ഉദാഹരണം 5.19 : റേഡിയോ ബട്ടൺ ചെക്ക് ബോക്സ് എന്നിവ ഉൾപ്പെടുത്തിയ ഒരു HTML ഫോം നിർമാണം

```
<HTML>
<HEAD> <TITLE>Checkbox Radio Button Control</TITLE> </HEAD>
<BODY Bgcolor= "#E9BEE5">
    <FORM> <BR>Sex:       
        <INPUT Type= "radio" Name= "sex" Value= "male"> Male
        <INPUT Type= "radio" Name= "sex" Value= "female"> Female
        <BR> <BR>Hobbies:
        <INPUT Type= "checkbox" Name= "Hobby" Value= "Games">
        Playing Games
        <INPUT Type= "checkbox" Name= "Hobby"
                Value="WatchingTV"> Watching TV
        <INPUT Type= "checkbox" Name= "Hobby" Value= "Reading">
        Reading
    </FORM>
</BODY>
</HTML>
```

ചിത്രം 5.25: ചെക്ക് ബോക്സും റേഡിയോ ബട്ടണും ഉൾപ്പെടുത്തിയ വെബ് പേജ്.

5.6.3 <TEXTAREA> ടാഗ്

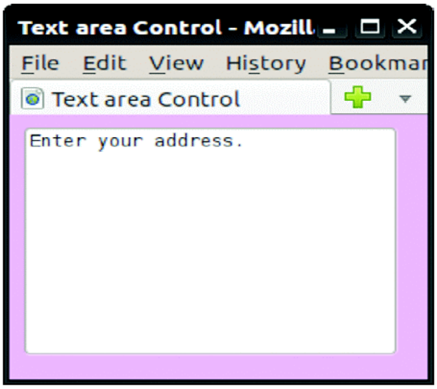
ഒരു ഫോമിൽ `<INPUT Type = "text">` എന്ന പ്രസ്താവന ഉപയോഗിച്ച് ഒറ്റ വരി വാക്യം നൽകുന്ന രീതി നാം കണ്ടു കഴിഞ്ഞു. (ഉദാഹരണം 5.18 ഉം ചിത്രം 5.24 ഉം) നോക്കുക. എന്നാൽ ചില അവസരങ്ങളിൽ ഒന്നിലധികം വരികളുള്ള വാചകങ്ങൾ ഫോമിൽ നൽകേണ്ടത് ആവശ്യമായി വരും. ഒരു വ്യക്തിയുടെ തപാൽ വിലാസം ഇതിന് ഉദാഹരണമാണ്. ഇതിനായി **<TEXTAREA>** എന്ന കണ്ടെയ്നർ ടാഗ് ഉപയോഗിക്കാം. ടാഗ് ജോഡികൾക്ക് ഇടയിലുള്ള ഒന്നിലധികം വരികളുള്ള വാക്യങ്ങൾ പ്രദർശിപ്പിക്കാ

നുള്ള ടെക്സ്റ്റ് ഏരിയായുടെ വിസ്തൃതി നിർണയിക്കുന്നത് ഈ ടാഗിന്റെ ആട്രിബ്യൂട്ടു കളുടെ വിലയെ ആശ്രയിച്ചിരിക്കും. <TEXTAREA> ടാഗിന്റെ പ്രധാന ആട്രിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കുന്നു.

1. **Name**: ഇത് കൺട്രോളിന് പേര് നൽകുന്നു.
2. **Rows**: ഇത് വരികളുടെ എണ്ണം നിർണയിക്കുന്നു.
3. **Cols**: ഇത് നിരകളുടെ എണ്ണം നിർണയിക്കുന്നു. അതായത് ഒരു വരിയിലെ അക്ഷ രങ്ങളുടെ എണ്ണം.

താഴെ നൽകിയിരിക്കുന്ന കോഡ് ഭാഗം പരിഗണിച്ച് <TEXTAREA>ടാഗിന്റെ ഉപയോഗം തിരിച്ചറിയുക.

```
<FORM Action= "guestbook.php" Method= "post">
   <TEXTAREA Rows= "10" Cols= "30" Name= "address">
       Enter your address.
   </TEXTAREA>
</FORM>
```



ചിത്രം 5.26: ടെക്സ്റ്റ് ഏരിയ ഉൾക്കൊള്ളുന്നു.

ഈ കോഡിനെ ബ്രൗസറിൽ തുറക്കുമ്പോൾ ലഭ്യ മാകുന്ന വെബ് പേജ് ചിത്രം 5.26 ൽ നൽകിയിരി ക്കുന്നു.<TEXTAREA>, </TEXTAREA> എന്നീ ടാഗ് ജോഡികൾക്കിടയിൽ നൽകുന്ന വാക്യങ്ങൾ കൺ ട്രോളിൽ പ്രദർശിപ്പിക്കുന്നു. ഉപയോക്താവ് ഏതെ ങ്കിലും ഡാറ്റ കൺട്രോളിൽ നൽകുമ്പോൾ ഈ വാക്യത്തിന് പകരമായി അത് പ്രദർശിപ്പിക്കുന്നു.

5.6.4 <SELECT> ടാഗ്

ഉപയോക്താവിന് ഒന്നോ അതിലധികമോ ഓപ്ഷനുകൾ തിരഞ്ഞെടുക്കുവാൻ കഴിയുന്ന കൺട്രോൾ ആണ് സെലക്ട് ബോക്സ്. ഇത് വിവിധ ഓപ്ഷനുകൾ താഴേക്കു വീഴുന്ന ഒരു ലിസ്റ്റ് രൂപത്തിൽ നൽകുന്നു. അതിനാൽ ഇത് ഡ്രോപ്ഡൗൺ ലിസ്റ്റ് ബോക്സ് എന്ന പേരിലും അറിയപ്പെടുന്നു. പരിമിതമായ സ്ഥലത്ത് കൂടുതൽ ഓപ്ഷനുകൾ പ്രദർശി പ്പിക്കുവാൻ സെലക്ട് ബ്ലോക്സ് സഹായിക്കുന്നു. **<SELECT>, </SELECT>** ടാഗിന കത്ത് നൽകുന്ന **<OPTION>** ടാഗ് ഉപയോഗിച്ചാണ് ലിസ്റ്റിൽ പ്രദർശിപ്പിക്കേണ്ട ഇന ങ്ങൾ നൽകുന്നത്.

കണ്ടെയ്നർ ടാഗായ <SELECT> ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്രിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കു ന്നു.

1. **Name**: ഈ ആട്രിബ്യൂട്ട് കൺട്രോളിന് പേര് നൽകുന്നു. ഈ പേര് ഇതിലെ ഡാറ്റ യോടൊപ്പം സെർവറിലേക്ക് അയയ്ക്കുന്നു. സെർവർ ഈ പേര് ഉപയോഗിച്ച് കൺട്രോളിന്റെ പേരിനൊപ്പമുള്ള വില തിരിച്ചറിയുകയും സ്വീകരിക്കുകയും ചെയ്യുന്നു.

2. **Size**: ഒരു സ്ക്രോളിങ് ലിസ്റ്റ് അവതരിപ്പിക്കുന്നതിന് ഉപയോഗിക്കാം. ഇതിന്റെ വിലയുടെ അടിസ്ഥാനത്തിൽ ഇത് ഒരു ലിസ്റ്റ് ബോക്സാണോ അല്ലെങ്കിൽ ഒരു ഡ്രോപ്ഡൗൺ ലിസ്റ്റ്ബോക്സാണോ എന്ന് തീരുമാനിക്കുന്നു. ഇതിന്റെ വില 1 ആണെങ്കിൽ ഡ്രോപ്ഡൗൺ ലിസ്റ്റ് ബോക്സായിരിക്കും.
3. **Multiple**: ഉപയോക്താവിന് ഒന്നിൽകൂടുതൽ ഇനങ്ങൾ തിരഞ്ഞെടുക്കുവാൻ ഈ ആട്രിബ്യൂട്ട് അനുവദിക്കുന്നു.

ഇനി നമുക്ക് <OPTION> ടാഗിനെക്കുറിച്ച് ചർച്ച ചെയ്യാം. <SELECT> </SELECT> എന്നീ ടാഗ് ജോഡികൾക്കിടയിൽ നൽകുന്ന എംപ്റ്റി ടാഗ് ആണിത്. <OPTION> ടാഗ് സെലക്ട് ബോക്സിനകത്ത് നൽകിയിരിക്കുന്ന ഓപ്ഷനുകൾ ലിസ്റ്റ് ചെയ്യുന്നു. ഈ ടാഗിന്റെ വിവിധ ആട്രിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കുന്നു.

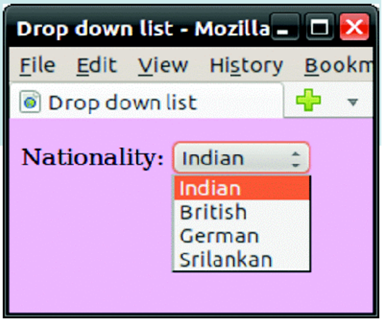
1. **Selected**: തനത് സെലക്ഷനെ സൂചിപ്പിക്കാൻ ഉപയോഗിക്കുന്നു.
2. **Value**: <OPTION> ടാഗിന് അകത്തുള്ള വിലകളിൽ നിന്നും ഉള്ളടക്കത്തേക്കാൾ വിഭിന്നമായ വില നൽകുവാൻ ഇത് ഉപയോഗിക്കുന്നു. ഈ വില ലഭ്യമല്ലെങ്കിൽ ഇതിന്റെ ഉള്ളടക്കം വിലയായി സ്വീകരിക്കും.

ഉദാഹരണം 5.20 ൽ നൽകിയിരിക്കുന്ന HTML കോഡ് ഉപയോഗിച്ച് ഒരു ഡ്രോപ്ഡൗൺ ലിസ്റ്റ് ബോക്സ് ഉൾപ്പെടുന്ന വെബ് പേജ് നിർമിക്കാം. <SELECT> ടാഗിന്റെ Size ആട്രിബ്യൂട്ടിന് ഒന്ന് (1) എന്ന വില ആണെന്നും കൂടാതെ സെലക്ട് ബോക്സിന് 4 ഓപ്ഷനുകൾ ഉണ്ടെന്നും പ്രത്യേകം ശ്രദ്ധിക്കുക.

ഉദാഹരണം 5.20: ഒരു ഡ്രോപ്ഡൗൺ ലിസ്റ്റ് ഉൾപ്പെട്ട HTML ഫോം നിർമാണം

```
<HTML>
<HEAD> <TITLE> Drop down list </TITLE> </HEAD>
<BODY Bgcolor= "#E9BEE5">
   <FORM Action= "guestbook.php" Method= "post">
      <P> Nationality:
      <SELECT Name= "Nationality" Size= "1">
          <OPTION Value= "Indian" selected> Indian
          <OPTION Value= "British"> British
          <OPTION Value= "German"> German
          <OPTION Value= "Srilankan"> Srilankan
      </SELECT>
   </FORM>
</BODY>
</HTML>
```

കോമ്പോ ബട്ടണിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ ലഭ്യമായ ഡ്രോപ്ഡൗൺ ലിസ്റ്റിൽ ചിത്രം 5.27ൽ കാണിക്കുന്നു.



ചിത്രം 5.27: സെലക്ട് ബോക്സ് ഉള്ള ഫോം

നമുക്കു ചെയ്യാം

ഉദാഹരണം 5.20 ലെ പ്രോഗ്രാം കോഡിൽ Size ആട്രിബ്യൂട്ടിനെ വ്യത്യാസപ്പെടുത്തിക്കൊണ്ട് സെലക്ട് ബോക്സിൽ ഉണ്ടാകുന്ന മാറ്റം നിരീക്ഷിക്കുക.

5.6.5 <FIELDSET> ടാഗ് ഉപയോഗിച്ച് ഫോം ഡാറ്റയെ ഒന്നിപ്പിക്കുന്നു (Grouping form data with <FIELDSET>tag)

ചിലപ്പോൾ പരസ്പരബന്ധമുള്ള ഡാറ്റ ഉൾക്കൊള്ളുന്ന കൺട്രോളുകളെ ഒരുമിച്ച് ചേർക്കേണ്ടത് ആവശ്യമായി വന്നേക്കാം. ഇതിനായി **<FIELDSET>** ടാഗ് ഉപയോഗിക്കാം. ഈ ടാഗ് ഉപയോഗിച്ച് ഫോമിനകത്ത് പരസ്പര ബന്ധമുള്ള കൺട്രോളുകൾ അടങ്ങിയ ഉപ വിഭാഗങ്ങൾ ഉൾപ്പെടുത്തുവാൻ സാധിക്കും. ഓരോ ഉപവിഭാഗത്തെയും തിരിച്ചറിയുവാൻ **<LEGEND>** ടാഗ് ഉപയോഗിക്കാം. <FIELDSET> ഇനങ്ങൾക്ക് ശീർഷകം നൽകുന്നതിന് <LEGEND> ടാഗ് ഉപയോഗിക്കാം. ഇത് ഒരു കണ്ടെയ്നർ ടാഗ് ആണ്.

നമ്മൾ ഇതുവരെ ചർച്ച ചെയ്ത എല്ലാ കൺട്രോളുകളും ഉൾപ്പെടുത്തി പേര്, വയസ്, അഡ്രസ്, വിനോദങ്ങൾ എന്നീ വിവരങ്ങൾ സമർപ്പിക്കുന്നതിനുള്ള HTML കോഡ് ഉദാഹരണം 5.21 ലും അതിനനുസൃതമായ വെബ് പേജ് ചിത്രം 5.28 ലും കാണിച്ചിരിക്കുന്നു.

ഉദാഹരണം 5.21: ഒരു വിദ്യാർഥിയുടെ വിവരങ്ങൾ സമർപ്പിക്കുന്ന HTML കോഡ്

```
<HTML>
<HEAD> <TITLE> FormResume </TITLE> </HEAD>
<BODY Bgcolor= "#E9BEE5">
   <CENTER ><H3>Enter your details</H3></CENTER>
   <FORM Action= "guestbook.php" Method= "get">
      Name: 
      <INPUT Type= "text" Name= "first_name" Size= "20"
            Maxlength= "20" Value= "First Name Here"><BR><BR>
      Age:
      <INPUT Type="text" Name="age" Size="3" Maxlength="3"><BR>
      Sex:    
      <INPUT Type="radio" Name="sex" Value="male"> Male
      <INPUT Type="radio" Name="sex" Value="female"> Female
      <FIELDSET>
         <LEGEND>Nationality</LEGEND>
         <SELECT Name= "Nationality" Size= "4">
            <OPTION Value= "Indian" Selected> Indian
            <OPTION Value= "British"> British
            <OPTION Value= "German"> German
            <OPTION Value= "Srilankan"> Srilankan
         </SELECT><BR><BR>
```

```
            Nativity:
            <INPUT Type= "text" Name="State" Size="15"><BR><BR>
            District: 
            <INPUT Type= "text" Name= "District" Size= "15">
        </FIELDSET><BR>
        Hobbies:
        <INPUT Type= "checkbox" Name= "Hobby" Value= "games">
                Playing Games
        <INPUT Type= "checkbox" Name= "Hobby"
                Value= "WatchingTV"> Watching TV
        <INPUT Type= "checkbox" Name= "Hobby" Value= "Reading">
                Reading<BR><BR>
        <TEXTAREA Rows= "5" Cols= "25" Name= "address">Address
        </TEXTAREA><BR><BR>
        <INPUT Type= "submit" Value= "submit">
        <INPUT Type= "Reset" Value= "reset">
    </FORM>
</BODY>
</HTML>
```

ചിത്രം 5.28: ഒരു വിദ്യാർഥിയുടെ വിവരങ്ങൾ സമർപ്പിക്കുന്നതിനുള്ള ഓൺലൈൻ ഫോം

5.6.6 ഫോം സമർപ്പണം (Form submission)

നാം ഫോമിലെ submit ബട്ടണിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ ഫോമിൽ നൽകിയ ഡാറ്റ സെർവറിലേക്ക് അയയ്ക്കപ്പെടുന്നു. ഉപയോക്താവിന്റെ (ക്ലയന്റ്) കമ്പ്യൂട്ടറിൽ നിന്നും വരുന്ന വിവരങ്ങൾ വെബ്സെർവർ സ്വീകരിച്ച് അതിൽ ലഭ്യമായ സെർവർസൈഡ് പ്രോഗ്രാം ഉപയോഗിച്ച് പ്രോസസ് ചെയ്യുന്നു.

ഫോം സമർപ്പിക്കുമ്പോൾ ചെയ്യേണ്ട പ്രവർത്തനങ്ങളെ നിർവചിക്കുന്ന ആട്രിബ്യൂട്ടാണ് `Action`. സാധാരണയായി സെർവർസൈഡ് പ്രോഗ്രാമിന്റെ URL ആണ് ഇതിന്റെ വിലയായി നൽകുന്നത്. ഒരു ഫോം സർവറിലേക്ക് അയയ്ക്കുന്നതിനുള്ള പൊതുവായ മാർഗം **submit** ബട്ടണിൽകൂടി ആണ്. സമർപ്പിക്കുന്ന ഫോമിലെ വിവരങ്ങൾ കൈകാര്യം ചെയ്യാൻ *guestbook.php* എന്ന സെർവർ സൈഡ് സ്ക്രിപ്റ്റ് ഉപയോഗിച്ചിരിക്കുന്നതായി ഉദാഹരണം 5.21 ൽ കാണാവുന്നതാണ്.

<FORM> ടാഗിലെ `Method` ആട്രിബ്യൂട്ട് ഫോം സമർപ്പിക്കുന്നതിനുള്ള HTTP മാർഗം (`get` or `post`) സൂചിപ്പിക്കുന്നു.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. വെബ് പേജിലൂടെ ഇൻപുട്ട് ഡാറ്റ നൽകുവാൻ HTML ൽ ________ ആണ് ഉപയോഗിക്കുന്നത്.
2. ഫോമിനകത്ത് ടെക്സ്റ്റ് ഡാറ്റ ഇൻപുട്ട് ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്ന രണ്ട് ടാഗുകളുടെ പേരെഴുതുക.
3. റേഡിയോ ബട്ടൺ, ചെക്ക്ബോക്സ് എന്നിവ തമ്മിലുള്ള വ്യത്യാസങ്ങൾ എഴുതുക.
4. ഫോമിനുള്ളിൽ ഡാറ്റ ഒരുമിപ്പിക്കുവാൻ ഉപയോഗിക്കുന്ന ടാഗ് ഏതാണ്?
5. <FORM> ടാഗിനുള്ളിൽ ഡാറ്റ ഇൻപുട്ട് ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്ന ടാഗിന്റെ പേര് എഴുതുക.

5.7. HTML 5 ന്റെ പൊതു അവലോകനം (Overview of HTML 5)

HTML ന്റെ പുതുക്കിയ പതിപ്പാണ് HTML 5. HTML 4.01 ന് ശേഷം വേൾഡ് വൈഡ് വെബ് കൺസോർഷ്യവും (W3 C) വെബ് ഹൈപ്പർ ടെക്സ്റ്റ് ആപ്ലിക്കേഷൻ ടെക്നോളജി വർക്കിങ് ഗ്രൂപ്പും (WHATWG) കൂട്ടായി വികസിപ്പിച്ചെടുത്തതാണ് HTML5. ഇതിൽ വീഡിയോ പ്ലേ ബാക്ക്, ഡ്രാഗ് ആൻഡ് ഡ്രോപ്പ് തുടങ്ങിയവ കൂട്ടിച്ചേർത്തിട്ടുണ്ട്.

പ്രമുഖ ബ്രൗസറുകളായ ഗൂഗിൾ ക്രോം, മോസില്ല ഫയർഫോക്സ്, ആപ്പിൾ സഫാരി ഓപ്പേറ, ഇന്റർനെറ്റ് എക്സ്പ്ലോറർ എന്നിവയുടെ ഏറ്റവും പുതിയ പതിപ്പുകൾ HTML5 നെ പിന്തുണയ്ക്കുന്നു. ഐഫോണുകൾ, ഐപാഡുകൾ, ആൻഡ്രോയിഡ് ഫോണുകൾ തുടങ്ങിയവയിലുള്ള മൊബൈൽ ബ്രൗസറുകളും HTML 5 നെ പിന്തുണയ്ക്കുന്നു. ഇപ്പോഴുള്ള ബ്രൗസറുകളിൽ പഴയതിനെയും പ്രവർത്തിപ്പിക്കാവുന്ന രീതിയിൽ ഉൾക്കൊള്ളിച്ചാണ് HTML 5 രൂപകൽപന ചെയ്തിരിക്കുന്നത്. HTML5 ന്റെ ലോഗോ ചിത്രം 5.29-ൽ നൽകിയിരിക്കുന്നു.

HTML

ചിത്രം 5.29: HTML 5 ന്റെ ലോഗോ.

HTML5 ൽ പരിചയപ്പെടുത്തുന്ന ഏതാനും പുതിയ ടാഗുകൾ താഴെ നൽകുന്നു.

1. **<VIDEO>, <AUDIO>:** HTML ഡോക്കുമെന്റുകളിൽ ദൃശ്യ ശ്രാവ്യ മാധ്യമങ്ങൾ വളരെ എളുപ്പം ഉൾക്കൊള്ളിക്കുവാൻ ഈ ടാഗുകൾ സൗകര്യമൊരുക്കുന്നു.

2. **<CANVAS>:** ചിത്രം ഗ്രാഫിക്സ് എന്നിവ വരയ്ക്കുക. ചാർട്ടുകളും ഗ്രാഫുകളും തയാറാക്കുക, നമുക്കാവശ്യമുള്ള രീതിയിൽ ഗ്രാഫിക്സ് മാറ്റിയെടുക്കുക തുടങ്ങിയവയ്ക്കുള്ള ലളിതവും ശക്തവുമായ മാർഗമാണ് ഈ ടാഗ് നൽകുന്നത്.

3. **<HEADER>, <FOOTER>:** ഒരു ഡോക്യുമെന്റിന്റെയോ സെക്ഷന്റെയോ ഹെഡർ വ്യക്തമാകുവാൻ <HEADER> ടാഗും, ഫൂട്ടർ (അടിക്കുറുപ്പ്) വ്യക്തമാക്കാൻ <FOOTER> ടാഗും ഉപയോഗിക്കുന്നു. ഒരു ഫൂട്ടറിനുള്ളിൽ ഡോക്കുമെന്റിന്റെ രചയിതാവ് പകർപ്പവകാശത്തെക്കുറിച്ചുള്ള വിവരങ്ങൾ, ബന്ധപ്പെടുവാനുള്ള വിലാസം തുടങ്ങിയവ സൂചിപ്പിക്കുന്നു. നമുക്ക് ഒരേ ഡോക്കുമെന്റിൽ തന്നെ ധാരാളം <HEADER>, <FOOTER> ഇനങ്ങൾ ഉൾക്കൊള്ളിക്കാനാകും. ഒരു വെബ് പേജിനുള്ളിൽ ആർട്ടിക്കിളുകളും സെക്ഷനുകളും നിർമിക്കുവാൻ ഈ ടാഗുകൾ ഉപയോഗിക്കുന്നു.

4. **<ARTICLE>, <SECTION>:** ആർട്ടിക്കിളുകൾ സ്വതന്ത്രവും വ്യത്യസ്തവുമായ വാർത്താക്കുറിപ്പ് അല്ലെങ്കിൽ ബ്ലോഗ് പോസ്റ്റ് ആണ്. ഒരു പേജിനകത്ത് വ്യത്യസ്തങ്ങളായ വിഷയ മേഖലകളാക്കുകയോ ആർട്ടിക്കിളിനെ പലഭാഗങ്ങളാക്കുകയോ ചെയ്യുന്നതിന് സെക്ഷൻ ഉപയോഗിക്കുന്നു. <ARTICLE>, <SECTION> എന്നീ ടാഗുകൾ ശരിയായ രീതിയിൽ ഉപയോഗിച്ചാൽ അത് സെർച്ച് എൻജിനിൽ ഈ പേജ് ലഭ്യതയ്ക്കുള്ള സാധ്യത വർധിക്കും.

5. **<OUTPUT>:** ഒരു സ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് നടത്തുന്ന കണക്കുകൂട്ടലുകളുടെ ഫലം ലഭ്യമാക്കുന്നതിന് ഈ ടാഗ് ഉപയോഗിക്കുന്നു.

6. **<DETAILS>:** ഹെഡറിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ എക്സ്പാൻഡ് ചെയ്യുന്നതിനും കൊളാപ്സ് ചെയ്യുന്നതിനും വേണ്ട വാക്യങ്ങളുടെ ഭാഗം നിർണയിക്കുവാൻ ഈ ടാഗ് ഉപയോഗിക്കുന്നു.

7. **<FIGURE> and <FIGCAPTION>:** ചിത്രങ്ങൾ ഉൾക്കൊള്ളിക്കുവാൻ <FIGURE> ടാഗും, അതിന്റെ ശീർഷകം നൽകാൻ <FIGCAPTION> ടാഗും ഉപയോഗിക്കുന്നു. <FIGCAPTION> ടാഗ് <FIGURE> ടാഗിനുള്ളിലാണ് നൽകുന്നത്.

8. **<PROGRESS>, <METER>:** <PROGRESS>, <METER> എന്നീ ടാഗുകൾ ഒരേ പോലെയാണ്. ഒരു ജോലിയുടെ പുരോഗതിയെ സൂചിപ്പിക്കാൻ <PROGRESS> ടാഗ് ഉപയോഗിക്കുന്നു. ഇത് ഫയൽ അപ്‌ലോഡ് പോലുള്ള പ്രവർത്തനങ്ങളുടെ പുരോഗതി കാണിക്കുന്നതിന് ഉപയോഗിക്കുന്നു. എന്നാൽ <METER> ടാഗ് ഒരു പരിധിയിലുള്ള അളവ് മാത്രം നൽകുന്നതിന് ഉപയോഗിക്കുന്നു (ഉദാഹരണത്തിന് ഹാർഡ് ഡിസ്കിന്റെ ഉപയോഗം നൽകുന്നതിന്).

നമുക്കു സംഗ്രഹിക്കാം

ഈ അധ്യായത്തിലൂടെ HTML ന്റെ ചില മികച്ച സവിശേഷതകൾ നാം കണ്ടു കഴിഞ്ഞു. ഉള്ളടക്കവാക്യങ്ങൾ ആകർഷകമാക്കുന്നതിനായി വിവിധയിനം ലിസ്റ്റുകൾ നാം ചർച്ച ചെയ്തു. ഒരു വെബ് പേജിൽ ഓഡിയോ, വീഡിയോ, ചിത്രങ്ങൾ തുടങ്ങിയവ ഉൾപ്പെടുത്തി അതിന്റെ മനോഹാരിത വർധിപ്പിക്കാം. കൂടാതെ ഒരു വെബ് പേജിൽ ഉൾപ്പെടുത്തുന്ന വിവിധതരം ഹൈപ്പർ ലിങ്കുകളെയും അവയുടെ

പ്രാധാന്യത്തെയും കുറിച്ച് നാം മനസിലാക്കി. പല വിവരങ്ങളും വ്യക്തമായും എളുപ്പത്തിലും മനസിലാക്കുവാൻ കഴിയുന്ന വിധത്തിൽ രേഖപ്പെടുത്തുവാൻ പട്ടിക ഉപയോഗിച്ചു. പട്ടിക നിർമാണത്തിനാവശ്യമായ <TABLE>, <TR>, <TH>, <TD> എന്നീ ടാഗുകളും അവയുടെ ആട്രിബ്യൂട്ടുകളും നാം ചർച്ച ചെയ്തു. പട്ടികയിലെ വരികളും നിരകളും ആവശ്യാനുസരണം ഒന്നിച്ച് ചേർക്കുന്ന Rowspan, Colspan എന്നീ ആട്രിബ്യൂട്ടുകളെയും നാം മനസിലാക്കി. ഒരേ സമയം വിവിധ വെബ് പേജുകളെ തുറന്നു കാണുവാൻ ബ്രൗസർ ജാലകത്തെ പല ഭാഗങ്ങളായി വിഭജിക്കാമെന്ന് നാം കണ്ടു. നെസ്റ്റഡ് ഫ്രെയിം സെറ്റ് എന്ന ആശയമുപയോഗിച്ച് ബ്രൗസർ ജാലകത്തെ വിവിധരീതികളിൽ വിഭജിക്കാമെന്നും നാം മനസ്സിലാക്കി. ബ്രൗസർ ജാലകത്തിലെ വിവിധ ഫ്രെയിമുകളെ ബന്ധിപ്പിക്കാൻ Target ആട്രിബ്യൂട്ടും നാം ചർച്ച ചെയ്തു. ഒരു വെബ് ഡോക്യുമെന്റിൽ ഉപയോക്താവിൽ നിന്നും വിലകൾ സ്വീകരിച്ച് സെർവറിലേക്ക് അയയ്ക്കുവാൻവേണ്ട ഫോം നിർമാണം നാം പഠിച്ചു. വിവിധ തരത്തിൽ ഡാറ്റ സ്വീകരിക്കുവാൻ വേണ്ടി ടെക്സ്റ്റ് ബോക്സ്, പാസ്‌വേർഡ്, റേഡിയോ ബട്ടൺ, ടെക്സ്റ്റ് ഏരിയ, സെലക്റ്റ് ബോക്സ് തുടങ്ങിയ വിവിധ കൺട്രോളുകൾ നാം പരിചയപ്പെട്ടു. ക്ലയന്റ് സൈഡ് സ്ക്രിപ്റ്റ്, സർവർസൈഡ് സ്ക്രിപ്റ്റ് എന്നിവയെ കുറിച്ച് ചില കാര്യങ്ങൾ സൂചിപ്പിച്ചു. ഈ വിഷയങ്ങൾ ചർച്ച ചെയ്യുന്ന അടുത്ത അധ്യായങ്ങളിലേക്കുള്ള ചവിട്ടുപടിയായി ഈ അധ്യായത്തെ കരുതാം.

നമുക്കു പരിശീലിക്കാം

1. താഴെ കൊടുത്തിരിക്കുന്ന വിശദാംശങ്ങളും സവിശേഷതകളും ഉൾക്കൊള്ളിച്ച് കേരളത്തിലെ ഏതെങ്കിലും ഒരു ജില്ലയുടെ വെബ് പേജ് തയാറാക്കാനുള്ള HTML കോഡ് എഴുതുക.

    - ടെക്സ്റ്റ് ഫോർമാറ്റിങ് ടാഗുകളും അവയുടെ ആട്രിബ്യൂട്ടുകളും ഉപയോഗിച്ച് ഒരു ശീർഷകവും അതിനു താഴെ മൂന്ന് വാചകത്തിൽ കുറയാതെയുള്ള ഒരു ഖണ്ഡികയിൽ ജില്ലയെക്കുറിച്ചുള്ള വിവരണവും നൽകുക.
    - ജില്ലയിലെ പ്രധാനപ്പെട്ട വിനോദസഞ്ചാര കേന്ദ്രങ്ങളുടെ ലിസ്റ്റ്.

2. നിങ്ങളുടെ സ്കൂളിന്റെ വെബ് പേജ് രൂപകൽപന ചെയ്യാൻ ആവശ്യമായ HTML കോഡ് താഴെപ്പറയുന്ന വിവരങ്ങൾ/സവിശേഷതകൾ ഉൾപ്പെടുത്തി എഴുതുക.

    - ടെക്സ്റ്റ് ഫോർമാറ്റിങ് ടാഗുകളും അവയുടെ ആട്രിബ്യൂട്ടുകളും ഉപയോഗിച്ച്

      ഒരു ശീർഷകവും അതിനനുബന്ധമായി മൂന്ന് വാചകങ്ങളിൽ കുറയാതെ നിങ്ങളുടെ സ്കൂളിന്റെ പ്രത്യേകതകൾ എടുത്തുപറയുന്ന ഖണ്ഡികയും.
    - നിങ്ങളുടെ സ്കൂളിലെ ഏതെങ്കിലും പഠനത്തോടൊപ്പമുള്ള പ്രവർത്തനങ്ങളുടെ ലിസ്റ്റ്. ഉദാഹരണം : സ്കൗട്ട്, ഗൈഡ്, എൻ.എസ്.എസ്.

3. താഴെപ്പറയുന്ന വിവരങ്ങൾ വ്യക്തമാക്കുന്ന വെബ് പേജിന്റെ HTML കോഡ് എഴുതുക.

> **Components of a Computer**
>
> - **Hardware**
>   1. I/O Devices
>   2. RAM
>   3. Hard Disk & DVD Drive
> - **Software**
>   1. Operating System
>   2. Application Programs

4. കേരളത്തിന്റെ ദേശീയോത്സവമായ ഓണത്തിന്റെ എടുത്തുപറയത്തക്ക പ്രത്യേകതകൾ വർണിക്കുന്ന വെബ് പേജിനുവേണ്ടി താഴെപ്പറയുന്ന സവിശേഷതകൾ ഉൾപ്പെടുത്താനുള്ള HTML കോഡ് എഴുതുക.

- ആകർഷകമായ ഫോണ്ടിലുള്ള ശീർഷകം.
- വള്ളംകളിയുടെ ചിത്രം പേജിന്റെ പശ്ചാത്തലമാക്കുക.
- ഓണത്തോട് അനുബന്ധിച്ച് നടത്തപ്പെടുന്ന പരമ്പരാഗത ആചാരങ്ങളായ അത്തപ്പൂക്കളം, തുമ്പിതുള്ളൽ, വള്ളംകളി, കുമ്മാട്ടിക്കളി, പുലികളി എന്നിവ ഏതെങ്കിലും രണ്ടിലേക്ക് ഇന്റേണൽ ലിങ്ക് നൽകുക.

5. നിങ്ങളുടെ സ്കൂളിന്റെ ഹൈസ്കൂൾ, ഹയർസെക്കന്ററി വിഭാഗങ്ങളുടെ വിവരങ്ങൾ അടങ്ങുന്ന രണ്ട് വെബ് പേജുകൾ നിർമിക്കുക. അടുത്ത വെബ് പേജ് നിർമിച്ച് തിരശ്ചീനമായ രണ്ട് ഭാഗങ്ങളാക്കി തിരിക്കുന്നു. ഒന്നാമത്തെ ഫ്രെയിമിൽ സ്കൂളിനെക്കുറിച്ചുള്ള ആമുഖവും രണ്ട് ലിങ്കുകളുമാണ് (ഹൈസ്കൂൾ, ഹയർസെക്കന്ററി) നൽകേണ്ടത്. ഉപയോക്താവ് ലിങ്കിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ യഥാക്രമം അവയുടെ വെബ് പേജുകൾ രണ്ടാമത്തെ ഫ്രെയിമിൽ ലഭ്യമാക്കുന്നതിന് വേണ്ട HTML കോഡ് തയാറാക്കുക.

6. ചുവടെ ചേർക്കുന്ന ടേബിളും അതിന് താഴെ കൊടുത്തിരിക്കുന്ന കേരള പോലീസിന്റെ വെബ്സൈറ്റിലേക്കുള്ള എക്സ്റ്റേണൽ ലിങ്ക് എന്നിവ പ്രദർശിപ്പിക്കുന്ന വെബ് പേജിന് ആവശ്യമായ HTML കോഡും തയാറാക്കുക.

Road Accidents in Kerala during 2012 - 2014

Year	Total Number of		
	Cases	Persons Killed	Persons Injured
2012	36174	4286	41915
2013	35215	4258	40346
2014	36282	4049	41096

Data Source: www.keralapolice.org

7. ചുവടെ ചേർക്കുന്ന രീതിയിലുള്ള ഒരു അപേക്ഷ ഫോം തയാറാക്കുന്നതിനാവശ്യമായ HTML കോഡ് എഴുതുക.

APPLICATION FOR THE BEST STUDENT AWARD

Name: [ ] Sex: Male ○ Female ○

Class & Division: [Select ▼]

Total Grade Point in Class XI: [ ]

Average Grade Point in Termly Exams in Class XII: [ ]

Cocurricular Activities:

NCC ☐ NSS ☐ Sports ☐ Arts ☐ Literary ☐

Other Achievements: [ ]

നമുക്കു വിലയിരുത്താം

1. ചുവടെ ചേർക്കുന്നവയിൽ തെറ്റ് കണ്ടുപിടിച്ച് തിരുത്തി എഴുതുക.

a.
```
<UL Type = "i" Start = 3>
```
b.
```
<IMG Src = "Myschool.jpg" Size  = "50" >
```
c.
```
<HTML>
     <HEAD><TITLE><HEAD></TITLE>
     <BODY> this is the body of the HTML
       document</BODY>
</HTML>
```

2. രോഹിത് HTML ഉപയോഗിച്ച് നിർമിച്ച ടേബിളിൽ ബോർഡർ കാണുവാൻ സാധിക്കുന്നില്ല. ഇതിന്റെ കാരണം എന്തായിരിക്കും?

3. ലിങ്ക് ചെയ്ത പേജ് ഒരു ഫ്രെയിമിനുള്ളിൽ പ്രദർശിപ്പിക്കുവാൻ <A> ടാഗിന്റെ __________ ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നു.

4. നിങ്ങളുടെ ആത്മാർഥ സുഹൃത്തുക്കളുടെ ഒരു ലിസ്റ്റ് തയാറാക്കുവാൻ നിങ്ങളുടെ കമ്പ്യൂട്ടർ അധ്യാപകൻ/അധ്യാപിക ആവശ്യപ്പെടുന്നു എന്ന് കരുതുക. നിങ്ങൾ ഏത് ടാഗിനാണ് പരിഗണന നൽകുക. ഇതിനുവേണ്ട HTML കോഡ് ഭാഗം തയാറാക്കുക.

5. <OL> ടാഗിലെ type ആട്രിബ്യൂട്ടിന് നൽകാവുന്ന വിലകളുടെ പേര് എഴുതുക.

6. <UL> ടാഗിന്റെ ആട്രിബ്യൂട്ടുകൾ എഴുതുക.

7. ഇംഗ്ലീഷ് ഭാഷയിലെ വലിയ അക്ഷരങ്ങൾ ഉപയോഗിച്ച് ക്രമനമ്പരോട് കൂടിയ ലിസ്റ്റ് നിങ്ങൾ എങ്ങനെ നിർമിക്കും?
8. ഇംഗ്ലീഷ് ഭാഷയിലെ ചെറിയ അക്ഷരങ്ങൾ ഒരു ലിസ്റ്റിലെ ഇനങ്ങളുടെ ക്രമനമ്പർ ആയി പ്രദർശിപ്പിക്കണം എന്ന് കരുതുക. എങ്ങനെ ഇത് സാധ്യമാക്കാം?
9. ക്രമനമ്പർ 6 മുതൽ ആരംഭിക്കുന്ന ഒരു ലിസ്റ്റ് നമുക്ക് എങ്ങനെ നിർമിക്കാം?
10. താഴെ കൊടുത്തിരിക്കുന്ന വിവരങ്ങൾ ഉൾക്കൊള്ളിച്ചു വെബ് പേജിനു വേണ്ട <HTML>കോഡ് എഴുതുക.

ABC Pvt. Ltd. Kerala
1. Health Care 2. Baby Products • Toys • Dress 3. Ladies Wear • Kurthas • Jeans

11. വരുൺ ഒരു വെബ് പേജ് നിർമിക്കുന്നു. അതിൽ 'sample' എന്ന വാക്ക് ഹൈപ്പർ ലിങ്കായി പ്രദർശിപ്പിച്ച് /home/scert/Desktop-ൽ സംഭരിക്കുന്ന *sample.html* എന്ന ഫയലിലേക്ക് ലിങ്ക് നൽകുവാനുള്ള HTML കോഡ് ഭാഗം എഴുതുക.
12. `Noshade` എന്ന ആട്രിബ്യൂട്ടുള്ള ടാഗിന്റെ പേര് എഴുതുക.
13. സുനിലിന് അയാളുടെ സ്വന്തം ആവശ്യങ്ങൾക്കായി ഒരു ഇ-മെയിൽ ലിങ്ക് ഉൾപ്പെടുന്ന വെബ് പേജ് നിർമിക്കണം. നിങ്ങൾക്ക് ഇ-മെയിൽ ലിങ്കിനുവേണ്ട പ്രോട്ടോകോൾ നിർദ്ദേശിക്കാമോ?
14. സഹീറിന് അയാളുടെ വെബ് പേജിൽ നിന്നും www.gmail.com എന്ന വെബ്സൈറ്റിലേക്ക് ബന്ധിപ്പിക്കണം. ഇതിനാവശ്യമായ ടാഗും അവയുടെ ആട്രിബ്യൂട്ടുകളും എഴുതുക.
15. HTML ൽ ലഭ്യമായ രണ്ട് തരം ഹൈപ്പർ ലിങ്കുകളുടെ സ്വഭാവ സവിശേഷതകൾ പ്രതിപാദിക്കുക.
16. `Cellspacing`, `Cellpadding` എന്നിവയുടെ വ്യത്യാസങ്ങൾ എഴുതുക.
17. `Text`, `Textarea` എന്നീ കൺട്രോളുകളുടെ വ്യത്യാസങ്ങൾ എഴുതുക.
18. `<FORM>` ടാഗിന്റെ പ്രധാന ആട്രിബ്യൂട്ടുകൾ `Action` നും .... ഉം ആകുന്നു.
19. തെറ്റോ ശരിയോ എന്ന് പറയുക.
    a. `<TABLE>` ടാഗിലെ `Align` ആട്രിബ്യൂട്ടിന്റെ തനത് വില `center` ആകുന്നു.
    b. `<FRAME>` ഒരു കണ്ടെയ്നർ ടാഗ് ആകുന്നു.
    c. ഉപയോക്താവ് ഒരു നിശ്ചിത ഫ്രെയിമിലെ ബോർഡറിന്റെ വലുപ്പ വ്യത്യാസം വരുത്തുന്നത് തടയുവാൻ സ്ക്രോളിങ് ഉപയോഗിക്കുന്നു.

20. ഏത് ടാഗാണ് ഒരു ബ്രൗസർ ജാലകത്തെ വിവിധ ഭാഗങ്ങളാക്കി വിഭജിക്കുന്നത്?

21. <FRAME> ടാഗിന്റെ പ്രധാന ആട്രിബ്യൂട്ടുകൾ എഴുതുക.

22. ഒരു വെബ് പേജിനെ ലംബമായ രണ്ട് ഭാഗങ്ങളാക്കി തിരിക്കുന്ന ഫ്രെയിം സെറ്റ് നിർമ്മിക്കുക. നിങ്ങൾക്ക് ഇഷ്ടപ്പെട്ട ഫുട്ബോൾ കളിക്കാരുടെ പേര് ഇടതുഭാഗത്തും, സെലക്ട് ചെയ്യുന്ന കളിക്കാരനെക്കുറിച്ചുള്ള ലഘു വിവരണം വലതുഭാഗത്തും പ്രദർശിപ്പിക്കണം. ഇതിനാവശ്യമായ കോഡ് എഴുതുക.

23. ചുവടെ ചേർക്കുന്ന HTML കോഡ് പരിശോധിച്ചതിന് ശേഷം അതിന്റെ ഔട്ട്പുട്ട് എന്താകും എന്ന് എഴുതുക.

```
<HTML>
<HEAD><TITLE>A simple table</TITLE></HEAD>
<BODY>
<TABLE border="1" Cellspacing= "1" Cellpadding= "10">
  <TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD></TR>
  <TR><TD> 4 </TD><TD> 5 </TD><TD> 6 </TD></TR>
  <TR><TD> 7 </TD><TD> 8 </TD><TD> 9 </TD></TR>
</TABLE>
</BODY>
</HTML>
```

<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td colspan="3">7</td></tr>
</table>

24. താഴെ കൊടുത്തിരിക്കുന്ന പട്ടിക പ്രദർശിപ്പിക്കുവാൻ വേണ്ട HTML കോഡ് എഴുതുക.

25. നിങ്ങളുടെ ഇമെയിൽ, വിലാസം, ഫോൺ നമ്പർ, പാസ്‌വേഡ് തുടങ്ങിയവ സ്വീകരിക്കുവാൻ വേണ്ട ഫോം നിർമ്മിക്കുന്നതിനാവശ്യമായ HTML കോഡ് എഴുതുക.

26. നിങ്ങൾക്ക് ഇഷ്ടപ്പെട്ട മൂന്ന് ബൈക്കുകളുടെ പേര് അടങ്ങുന്ന ലിസ്റ്റ് ഒരു ഫ്രെയിമിലും, അതിന്റെ ചിത്രവും, വിവരണവും അടുത്ത ഫ്രെയിമിൽ ലിങ്ക് ചെയ്തും പ്രദർശിപ്പിക്കുന്നതിനുള്ള HTML കോഡ് എഴുതുക.

27. ചുവടെ ചേർക്കുന്ന HTML കോഡ് ഉപയോഗിച്ച് രാജ് നിർമിച്ച വെബ് പേജിലെ പട്ടിക ശരിയായ രൂപത്തിൽ കാണുവാൻ സാധിക്കുന്നില്ല. ഇതിന്റെ കാരണം കണ്ടെത്തി ശരിയായ രീതിയിൽ എഴുതുക.

```
<HTML>
<HEAD><TITLE> My Page </TITLE></HEAD>
<BODY>
<TABLE><TR><TH>Roll No. </TH><TH> Name </TH></TR>
<TR><TD>1 </TD><TD> Huda </TD></TR>
<TR><TD>2 </TD><TD>Bincy</TD></TR>
</TABLE>
</BODY>
</HTML>
```

G4N2I8

ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ചുള്ള ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിങ്

പ്രധാന പഠനനേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പഠനം പൂർത്തിയാകുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ

- ക്ലൈന്റ് വശത്തെയും സെർവർ വശത്തെയും സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ ഉപയോഗം തിരിച്ചറിയുന്നു.
- ക്ലൈന്റ് സൈസ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ ആവശ്യകത വിശദീകരിക്കുന്നു.
- ക്ലൈന്റ് സൈസ് സ്ക്രിപ്റ്റിംഗ് ഭാഷയായ ജാവാസ്ക്രിപ്റ്റിന്റെ പ്രാധാന്യം തിരിച്ചറിയുന്നു.
- വെബ്പേജിൽ ജാവാസ്ക്രിപ്റ്റ് ഫംഗ്ഷൻ ഉപയോഗിക്കുന്നു
- ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ ഡാറ്റ ഇനങ്ങൾ വിശദീകരിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിൽ ശരിയായ വേരിയബിളുകൾ ഉപയോഗിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിൽ അനുയോജ്യമായ കൺട്രോൾ സ്ട്രക്ച്ചറുകൾ ഉപയോഗിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിൽ അനുയോജ്യമായ അന്തർനിർമിത ഫംഗ്ഷനുകൾ ഉപയോഗിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിലെ ഡോക്യുമെന്റ് ഘടകങ്ങൾ ഉപയോഗിക്കുന്ന രീതി വിവരിക്കുന്നു.
- ടെക്സ്റ്റ് ബോക്സുകളിലേയും കോംബോ ബോക്സുകളിലേയും വിലകൾ കൈകാര്യം ചെയ്യുന്നതിനുള്ള ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷനുകൾ നിർമിക്കുന്നു.

മുൻ അധ്യായങ്ങളിൽ ടെക്സ്റ്റുകളും ഗ്രാഫിക്സുകളും ഉൾപ്പെടുന്ന വിവിധ തരം വെബ് പേജുകൾ നിർമിക്കുവാൻ നാം പഠിച്ചു. ഈ ഇന്റെർനെറ്റ് യുഗത്തിൽ നമ്മളിൽ പലരും വിവിധ ആവശ്യങ്ങൾക്കായി അനവധി വെബ്സൈറ്റുകൾ സന്ദർശിക്കാറുണ്ട്. നമുക്ക് പരിചിതമല്ലാത്ത അനേകം സവിശേഷതകൾ ഇത്തരം വെബ് പേജുകളിൽ അടങ്ങിയിരിക്കുന്നു. അതുകൊണ്ടുതന്നെ ഇത്തരം വെബ് പേജുകൾ നിർമിക്കുന്നതിനായി സ്ക്രിപ്റ്റിംഗ് ഭാഷകളിലുള്ള അറിവ് ആവശ്യമാണ്. ക്ലൈന്റ് സൈഡിലും സെർവർ സൈഡിലും വിവിധയിനം സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ ഉപയോഗിക്കുന്നുണ്ട്. ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകളായ ജാവാസ്ക്രിപ്റ്റ്, വിബി സ്ക്രിപ്റ്റ് എന്നിവയിൽ ജാവാസ്ക്രിപ്റ്റാണ് കൂടുതലായി ഉപയോഗിക്കുന്നത്. കാരണം വിബി സ്ക്രിപ്റ്റിനെ അപേക്ഷിച്ച് ജാവാസ്ക്രിപ്റ്റിനെയാണ് കൂടുതൽ ബ്രൗസറുകളും പിന്തുണയ്ക്കുന്നത്. ഇന്റർനെറ്റിൽ അനേകം ആളുകൾ വ്യത്യസ്ത ബ്രൗസറുകൾ ഉപയോഗിക്കുന്നതിനാൽ വെബ് പേജുകൾ കഴിയുന്നത്ര എല്ലാ ബ്രൗസറുകൾക്കും അനുയോജ്യമായ രീതിയിൽ നിർമിക്കേണ്ടതുണ്ട്. ഒരു വെബ് പേജിൽ ജാവാസ്ക്രിപ്റ്റ് എങ്ങനെ ഉപയോഗിക്കാമെന്ന് ഈ അധ്യായത്തിൽ മനസ്സിലാക്കാം. നമുക്ക് C++ പരിചയമുള്ളതിനാൽ ജാവാസ്ക്രിപ്റ്റ് മനസ്സിലാക്കുവാൻ വളരെ എളുപ്പമാണ്. കാരണം C++ ന്റെ അതേ വാക്യഘടനയാണ് ജാവാസ്ക്രിപ്റ്റ് പിന്തുടരുന്നത്.

നെറ്റ്സ്കേപ് ബ്രൗസറിനുവേണ്ടി 'ബ്രെൻഡൻ ഇച്ച്' ആണ് ജാവാസ്ക്രിപ്റ്റ് വികസിപ്പിച്ചത്. ജാവാസ്ക്രിപ്റ്റിന്റെ ആദ്യനാമം 'മോച്ച' എന്നായിരുന്നു. 1995-ൽ നെറ്റ് സ്കേപ് ബ്രൗസർ പതിപ്പ് 2.0 ത്തിൽ ഉൾപ്പെടുത്തിയപ്പോൾ ഇതിന്റെ പേര് 'ജാവാസ്ക്രിപ്റ്റ്' എന്നാക്കിമാറ്റി. ആദ്യകാലങ്ങളിൽ, നെറ്റ് സ്കേപ് ബ്രൗസർ മാത്രമേ ജാവാസ്ക്രിപ്റ്റിന് പിന്തുണ നൽകിയിരുന്നുള്ളൂ. എന്നാൽ ജാവാസ്ക്രിപ്റ്റിന്റെ വിപുലമായ പ്രചാരം കാരണം, ഇന്റർനെറ്റ് എക്സ്പ്ലോറർ 1996-ൽ ജാവാസ്ക്രിപ്റ്റിന് പിന്തുണ നൽകി. ഇപ്പോൾ, ലോകത്തിലെ മിക്ക ബ്രൗസറുകളും ജാവാ സ്ക്രിപ്റ്റിനെ പിന്തുണയ്ക്കുന്നു.

ബ്രെൻഡൻ ഇച്ച്

6.1 ജാവസ്ക്രിപ്റ്റ് ആരംഭിക്കാം (Getting started with JavaSript)

നാലാമത്തെ അധ്യായമായ വെബ് ടെക്നോളജിയിൽ ക്ലൈന്റ് സൈഡിലും സെർവർ സൈഡിലും ഉള്ള സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ ഉപയോഗം നാം ചർച്ചചെയ്തു. ക്ലൈന്റ് സൈഡിലെ ഡാറ്റയുടെ സാധുത ഉറപ്പുവരുത്തുന്നതിനായി ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ ഉപയോഗിക്കുന്നു. ഇത് സെർവറിലെ നെറ്റ് വർക്ക് ട്രാഫിക്കും ജോലി ഭാരവും കുറയ്ക്കുന്നു. സെർവർ സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ സെർവറിൽ പ്രവർത്തിക്കുന്നു. ഇതിലൂടെ നിർമിക്കപ്പെട്ട വെബ് പേജ് ക്ലൈന്റ് ബ്രൗസറിലേക്ക് തിരിച്ചയക്കുന്നു. ഡാറ്റാ ബേസിന്റെ രൂപത്തിൽ വലിയ അളവ് ഡാറ്റ സെർവറിൽ സംഭരിക്കുന്നു. അതിനാൽ സെർവർ സൈഡിലെ സ്ക്രിപ്റ്റിംഗ് ഭാഷ വ്യത്യസ്ത ആവശ്യങ്ങൾക്ക് വേണ്ടി ഈ ഡാറ്റബേസുമായി സംവദിക്കേണ്ടതായി വരും. എന്നാൽ ഒരു ക്ലൈൻറ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷ സെർവർ ഡാറ്റബേസുമായി നേരിട്ട് ബന്ധപ്പെടുന്നില്ല. അതുകൊണ്ടുതന്നെ ക്ലൈന്റ് സൈഡിലും സെർവർ സൈഡിലും ഉപയോഗിക്കുന്ന ഭാഷകളും കമാൻഡുകളും വ്യത്യസ്തങ്ങളാണ്.

ഈ അധ്യായത്തിൽ ക്ലൈന്റ് വശം സ്ക്രിപ്റ്റിംഗ് ഭാഷ എന്ന നിലയിൽ നമുക്ക് ജാവാ സ്ക്രിപ്റ്റിന്റെ അടിസ്ഥാന വിവരങ്ങൾ മനസിലാക്കാം. ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് HTML പേജിലെ വിവിധ ഭാഗങ്ങളിൽ പ്രോഗ്രാമുകൾ ഉൾപ്പെടുത്താൻ കഴിയും. ഇതിനായി <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നു.

<SCRIPT> Tag

ഒരു HTML പേജിൽ സ്ക്രിപ്റ്റിംഗ് കോഡ് ഉൾപ്പെടുത്താൻ <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നു. <SCRIPT> ടാഗിലെ 'language' ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നത് സ്ക്രിപ്റ്റിംഗ് ഭാഷയുടെ പേര് വ്യക്തമാക്കുന്നതിനാണ്. ഇവിടെ ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷയായി ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്നതിനുവേണ്ടി <SCRIPT> ടാഗിലെ 'language' ആട്രിബ്യൂട്ടിന്റെ വില 'JavaScript' എന്ന് നൽകണം.

ഒരു HTML പേജിൽ <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നത് താഴെ കാണുന്നത് പോലെയാണ്.

The <SCRIPT> tag can be used in an HTML page as follows.

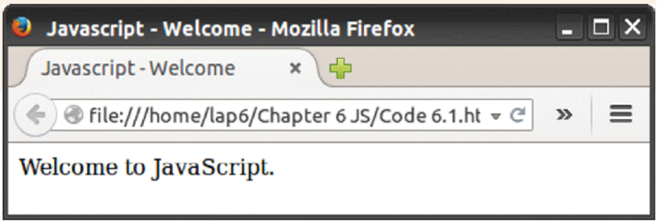
```
<SCRIPT Language= "JavaScript">
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
</SCRIPT>
```

ഇനി നമുക്ക് താഴെകൊടുത്തിരിക്കുന്ന HTML ഫയൽ പരിഗണിക്കാം

ജാവാസ്ക്രിപ്റ്റിലെ ഐഡന്റിഫയറുകൾ കേസ് സെൻസിറ്റീവ് ആണ്. ജാവാസ്ക്രിപ്റ്റിൽ സാധാരണയായി ഐഡന്റിഫയറുകൾക്ക് കാമൽകേസ് (camelCase) പേരുകളാണ് ഉപയോഗിക്കുക. ഉദാഹരണത്തിന് firstName, checkData തുടങ്ങിയവ. ഒന്നിൽകൂടുതൽ വാക്കുകൾ ഒറ്റവാക്കായി ഉപയോഗിക്കുന്ന ഒരു നാമകരണ രീതിയാണ് കാമൽകേസ്. ഓരോ വാക്കിന്റെയും ആദ്യ അക്ഷരം വലിയ അക്ഷരം ഉപയോഗിക്കുന്നതിനെ 'അപ്പർകാമൽകേസ്' എന്ന് വിളിക്കുന്നു. ആദ്യത്തെ വാക്കൊഴികെ മറ്റെല്ലാ വാക്കിന്റെയും ആദ്യ അക്ഷരം വലിയ അക്ഷരം ഉപയോഗിക്കുന്നതിനെ 'ലോവർകാമൽകേസ്' എന്നു വിളിക്കുന്നു. കാമൽകേസ് രീതി വാക്കുകൾ വായിക്കുവാൻ എളുപ്പമുള്ളതാക്കുന്നു.

ഉദാഹരണം 6.1 : ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് ഒരു വെബ് പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Welcome</TITLE> </HEAD>
<BODY>
    <SCRIPT Language= "JavaScript">
        document.write("Welcome to JavaScript.");
    </SCRIPT>
</BODY>
</HTML>
```



ചിത്രം 6.1: ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്ന വെബ് പേജ്

മുകളിലുള്ള കോഡ് ഏതെങ്കിലും ടെക്സ്റ്റ് എഡിറ്ററിൽ ടൈപ്പ് ചെയ്യാം. മുൻ അധ്യായങ്ങളിൽ HTML പേജുകൾ സൃഷ്ടിക്കുന്നതിന് നിങ്ങൾ ഉപയോഗിച്ചിരിക്കുന്ന ജീനി എഡിറ്ററിലും ഇത് ടൈപ്പ് ചെയ്യാവുന്നതാണ്. code6.html എന്ന പേരിൽ ഫയൽ സൂക്ഷിക്കുക. HTML പേജിൽ ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ചിട്ടുണ്ടെങ്കിലും അത് സേവ് ചെയ്തിരിക്കുന്നത്. html എക്സ്റ്റൻഷനിലാണ് എന്നത് പ്രത്യേകം ശ്രദ്ധിക്കേണ്ടതാണ്. ഇനി ഏതെങ്കിലും ഒരു ബ്രൗസറിൽ നമ്മൾ സേവ് ചെയ്ത HTML ഫയൽ തുറക്കുക. ചിത്രം 6.1 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഒരു വെബ് പേജ് ലഭിക്കും. 'document.write' എന്ന പ്രസ്താവന ചെറിയ അക്ഷരങ്ങളിലാണ് എഴുതിയിരിക്കുന്നത് എന്നതും ശ്രദ്ധിക്കുക, കാരണം ജാവാസ്ക്രിപ്റ്റ് കേസ് സെൻസിറ്റീവ് സ്ക്രിപ്റ്റിംഗ് ഭാഷയാണ്. സ്ക്രിപ്റ്റിലെ കീവേഡുകൾ എല്ലാം ചെറിയ അക്ഷരത്തിലാണ് എഴുതേണ്ടത്.

മുകളിൽ കൊടുത്തിരിക്കുന്ന HTML കോഡിൽ, document.write() എന്നത് ഒരു ജാവാസ്ക്രിപ്റ്റ് നിർദേശം ആണ്. അത് HTML പേജിന്റെ ബോഡിക്കകത്ത് ഒരു ടെക്സ്റ്റ് ഉൾക്കൊള്ളിക്കുവാൻ ഉപയോഗിക്കുന്നു. അതായത്, മുകളിൽ നൽകിയിരിക്കുന്ന HTML

കോഡ് നിർമിക്കുന്ന അതേ വെബ്പേജ് താഴെ നൽകിയിരിക്കുന്ന ജാവാ സ്ക്രിപ്റ്റ് ഇല്ലാത്ത HTML കോഡും നിർമ്മിക്കുന്നു.

ഉദാഹരണം 6.2 : HTML ഉപയോഗിച്ച് ഒരു വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Welcome</TITLE> </HEAD>
<BODY>
   Welcome to JavaScript.
</BODY>
</HTML>
```

മേൽ വിവരിച്ച ഉദാഹരണങ്ങൾ 6.1 ഉം 6.2 ഉം താരതമ്യം ചെയ്യാം. രണ്ടാമത്തെ HTML കോഡിൽ, "Welcome to JavaScript" എന്നത് ബോഡി ഭാഗത്തിൽ നേരിട്ട് എഴുതിയിരിക്കുന്നു. ആദ്യത്തെ കോഡിൽ "Welcome to JavaScript" എന്നത് ജാവാസ്ക്രിപ്റ്റ് രീതിയായ document.write( ) ഉപയോഗിച്ച് ബോഡി ഭാഗത്ത് ഉൾപ്പെടുത്തിയിരിക്കുന്നു.

യഥാർഥത്തിൽ 'document' എന്നത് വെബ് പേജിന്റെ ബോഡി വിഭാഗത്തെ പ്രതിനിധാനം ചെയ്യുന്നു. അതിനാൽ document.write() എന്ന ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷൻ വെബ് പേജിലെ ബോഡി ഭാഗത്തിൽ ഒരു ടെക്സ്റ്റ് ഉൾപ്പെടുത്തും. C++ പോലെ സ്ക്രിപ്റ്റിലെ ഓരോ പ്രസ്താവനയും ഒരു അർദ്ധവിരാമത്തിൽ (;) അവസാനിക്കുന്നു എന്നത് ശ്രദ്ധിക്കുക.

`<SCRIPT Language= "JavaScript">` എന്ന HTML നിർദേശം സൂചിപ്പിക്കുന്നത് തുടർന്നുവരുന്നത് ജാവാസ്ക്രിപ്റ്റ് കോഡ് ആണെന്നുള്ളതാണ്. ഇനി നമുക്ക് ജാവാ സ്ക്രിപ്റ്റിനെ ബ്രൗസർ എങ്ങനെ കൈകാര്യം ചെയ്യുന്നുവെന്ന് കാണാം. എല്ലാ ബ്രൗസറിലും ജാവാസ്ക്രിപ്റ്റ് എൻജിൻ ഉണ്ട്. പ്രവർത്തിപ്പിക്കുന്ന സമയത്ത് സ്ക്രിപ്റ്റ് കോഡിനെ ജാവാസ്ക്രിപ്റ്റ് എൻജിൻ വ്യാഖ്യാനിക്കും. ജാവാസ്ക്രിപ്റ്റ് കോഡ് നിർവഹണത്തിനുള്ള വിർച്യുൽ മെഷിനാണ് (virtual machine) ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിൻ. ബ്രൗസർ, ജാവാസ്ക്രിപ്റ്റ് കോഡ് കാണുമ്പോൾ അതിനെ സ്ക്രിപ്റ്റ് എൻജിനിലൂടെ കടത്തിവിടുന്നു. ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിൻ കോഡിനെ പ്രവർത്തിപ്പിക്കുന്നു. ഒരു HTML പേജിൽ ജാവാ സ്ക്രിപ്റ്റ് കോഡ് ഇല്ലെങ്കിൽ ബൗസറിന് ഒറ്റയ്ക്ക് തന്നെ HTML പേജ് പ്രദർശിപ്പിക്കാൻ കഴിയും. ജാവാസ്ക്രിപ്റ്റ് കോഡ് ഉണ്ടെങ്കിൽ ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിന്റെ സഹായത്തോടു കൂടി മാത്രമേ HTML പേജ് പ്രദർശിപ്പിക്കുവാൻ കഴിയൂ. ഇപ്രകാരം ജാവാസ്ക്രിപ്റ്റ് ഇല്ലാത്ത HTML ഫയലുകൾ എപ്പോഴും ജാവാസ്ക്രിപ്റ്റ് ഉള്ളവയെക്കാൾ വേഗത്തിൽ പ്രദർശിപ്പിക്കാൻ കഴിയും.

ഉദാഹരണം 6.3 ൽ ജാവാസ്ക്രിപ്റ്റ് കോഡും HTML ടാഗുകളും ഇടകലർത്തി നൽകിയിരിക്കുന്നു. ഇതിന്റെ ഔട്ട്പുട്ട് വെബ്പേജ് ചിത്രം 6.2 ൽ നൽകിയിരിക്കുന്നു.

ഉദാഹരണം 6.3: ഹെഡ്ഡിങ് ടാഗുകൾ ഉൾക്കൊള്ളുന്ന ഒരു വെബ് പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Welcome</TITLE> </HEAD>
<BODY>
   <H1>
   <SCRIPT Language= "JavaScript">
      document.write("This is in H1 Head");
   </SCRIPT>
   </H1>
   <BR>
   <H2>
   <SCRIPT Language= "JavaScript">
      document.write("This is in H2 Head");
   </SCRIPT>
   </H2>
</BODY>
</HTML>
```

ചിത്രം 6.2: ഹെഡ്ഡിംഗ് ടാഗുകൾ അടങ്ങുന്ന വെബ് പേജ്

മുകളിൽ നൽകിയിരിക്കുന്ന HTML ടാഗുകൾക്കിടയിൽ ഒന്നിലധികം തവണ സ്ക്രിപ്റ്റിംഗ് കോഡുകൾ ഉപയോഗിച്ചിട്ടുണ്ട്. അതുപോലെ നിങ്ങൾക്ക് HTML ടാഗുകൾക്കിടയിൽ എത്ര തവണ വേണമെങ്കിലും സ്ക്രിപ്റ്റ് കോഡുകൾ ഉപയോഗിക്കാം. സ്ക്രിപ്റ്റ് കോഡുകൾ ഉപയോഗിക്കുന്നത് എവിടെയായിരുന്നാലും <SCRIPT>, </SCRIPT> ടാഗുകൾക്കിടയിൽ അവ എഴുതിയിരിക്കണം.

ഒരു വെബ് പേജിൽ ജാവാസ്ക്രിപ്റ്റിന്റെ പ്രവർത്തനം അനുവദിക്കുന്നതിനോ നിരാകരിക്കുന്നതിനോ ഉള്ള സൗകര്യം എല്ലാ വെബ് ബ്രൗസറുകളും ഉപയോക്താക്കൾക്ക് നൽകുന്നുണ്ട്. ജാവാസ്ക്രിപ്റ്റ് നിരാകരിക്കുന്നതിലൂടെ, ആ ബ്രൗസറിലെ ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിന്റെ പ്രവർത്തനമാണ് നിരാകരിക്കുന്നത്. മോസില്ലയിൽ, Tools -> Options -> Content -> Enabled Java Script തിരഞ്ഞെടുത്ത് ജാവാസ്ക്രിപ്റ്റ് പ്രവർത്തനക്ഷമമാക്കുകയോ പ്രവർത്തനക്ഷമമല്ലാതാക്കുകയോ ചെയ്യാം. Google Chrome Â, Tools -> Settings -> Show Advanced Settings -> Content Settings -> Do not allow any site to run JavaScript രീതി ഉപയോഗിക്കാം. ഒരു വെബ് ബ്രൗസറിൽ ജാവാസ്ക്രിപ്റ്റ് പ്രവർത്തനക്ഷമമല്ലാതാക്കിയിട്ടുണ്ടെങ്കിൽ ആ ബ്രൗസറിൽ സ്ക്രിപ്റ്റ് പ്രവർത്തിക്കുകയില്ല അതായത്, ബ്രൗസർ <SCRIPT> ... </ SCRIPT> ടാഗുകൾക്കിടയിലുള്ള ഉള്ളടക്കം അവഗണിക്കും. ഒരു ബ്രൗസറിന്റെ പ്രവർത്തനം പ്രധാനമായും അതിന്റെ സ്ക്രിപ്റ്റ് എഞ്ചിന്റെ പ്രകടനത്തെ ആശ്രയിച്ചിരിക്കുന്നു. ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കാത്ത ഒരു ഡൈനാമിക് വെബ് പേജ് കാണാൻ പ്രയാസമാണ്. മികച്ചതും, വേഗതയേറിയതും ശക്തവുമായ ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിനുകളുടെ നിർമാണത്തിനായി എല്ലാ ബ്രൗസർ നിർമ്മാതാക്കളും പരസ്പരം മത്സരിക്കുന്നു.

നിങ്ങളുടെ പുരോഗതി അറിയാം

1. <SCRIPT> ടാഗിന്റെ ഒരു ആട്രിബ്യൂട്ടിന്റെ പേര് എഴുതുക.
2. ജാവാസ്ക്രിപ്റ്റിൽ ഒരു HTML പേജിന്റെ ബോഡി ഭാഗത്തിൽ ഒരു ടെക്സ്റ്റ് പ്രിന്റ് ചെയ്യാൻ ഉപയോഗിക്കുന്ന ഫങ്ഷൻ __________ ആണ്
3. ജാവാസ്ക്രിപ്റ്റ് കോഡ് ആണ് എഴുതിയിരിക്കുന്നത് എന്ന് വ്യക്തമാക്കുന്നത് <SCRIPT> ടാഗിന്റെ language ആട്രിബ്യൂട്ടിന് നൽകിയ __________ എന്ന മൂല്യമാണ്.
4. ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിന്റെ ഉപയോഗം എന്താണ്?
5. താഴെ പറയുന്ന പ്രസ്താവനകൾ ശരിയോ തെറ്റോ എന്ന് എഴുതുക.

a. ജാവ സ്ക്രിപ്റ്റ് മാത്രമാണ് ക്ലൈന്റ് വശം സ്ക്രിപ്റ്റിംഗ് ഭാഷ.

b. ഒരു HTML പേജിൽ ക്ലയന്റ് വശം സ്ക്രിപ്റ്റിംഗ് ഭാഷ ഉൾപ്പെടുത്താൻ <SCRIPT> ടാഗാണ് ഉപയോഗിക്കുന്നത്.

c. ഒരു HTML ഫയലിൽ ഒരു <SCRIPT> ടാഗ് മാത്രമേ അടങ്ങിയിരിക്കാവൂ.

d. നമുക്ക് HTML കോഡിൽ ജാവാസ്ക്രിപ്റ്റ് കോഡ് ഇടകലർത്തുവാൻ കഴിയും.

e. ജാവാസ്ക്രിപ്റ്റ് കോഡ് എല്ലായ്പ്പോഴും <SCRIPT>, </ SCRIPT> എന്നതിന് അകത്തതായിരിക്കണം.

f. ഓരോ ജാവാസ്ക്രിപ്റ്റ് പ്രസ്താവനകളും അർദ്ധവിരാമം(;) കൊണ്ട് അവസാനിക്കുന്നു.

നമുക്കു ചെയ്യാം

പേജിന്റെ ബോഡി വിഭാഗത്തിനുള്ളിൽ മാത്രം ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് താഴെ പറയുന്ന വെബ് പേജ് സൃഷ്ടിക്കാൻ ഒരു HTML കോഡ് എഴുതുക. അതായത്, ബോഡി വിഭാഗം താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ ആയിരിക്കണം.

```
<BODY>
<SCRIPT Language= "JavaScript">
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
</SCRIPT>
</BODY>
```

6.2 ജാവാസ്ക്രിപ്റ്റിലെ ഫങ്ഷനുകളുടെ നിർമാണം (Creating functions in JavaScript)

C++ ലെ ഫങ്ഷനുകളെക്കുറിച്ച് നാം നേരത്തെ മനസ്സിലാക്കിയിട്ടുണ്ട്. ജാവാസ്ക്രിപ്റ്റിലും പ്രവർത്തനങ്ങൾ നിർവചിക്കപ്പെട്ടിട്ടുള്ളതും വിളിക്കപ്പെടുന്നതും C++ ലെ അതേ രീതിയിൽ തന്നെയാണ്. ഒരു പേരുള്ള നിർദ്ദേശങ്ങളുടെ കൂട്ടമാണ് ഫങ്ഷൻ. വ്യത്യസ്ത

ആവശ്യങ്ങൾക്ക് ഉപയോഗിക്കാൻ കഴിയുന്ന ധാരാളം അന്തർനിർമിത ഫങ്ഷനുകൾ ജാവാസ്ക്രിപ്റ്റിൽ ഉണ്ട്. ഇവയിൽ ചില ഫങ്ഷനുകൾ ഈ അധ്യായത്തിൽ പിന്നീട് ചർച്ചചെയ്യും. ഈ അന്തർനിർമിത ഫങ്ഷനുകൾക്ക് പുറമെ നമുക്കും ഫങ്ഷനുകൾ നിർവ്വചിക്കാം. വെബ് പ്രോഗ്രാമിന്റെ ഒരു ഭാഗം ഒന്നിൽ കൂടുതൽ തവണ പ്രവർത്തിപ്പിക്കാം എന്നതാണ് ഒരു ഫങ്ഷൻ ഉപയോഗിക്കുന്നതുകൊണ്ടുള്ള ഏറ്റവും വലിയ മെച്ചം. ഈ ഭാഗം ഒരു പ്രാവശ്യം ഫങ്ഷനായി എഴുതിയാൽ മതി. താഴെ പറയുന്ന കോഡ് നോക്കുക.

```
function print()
{
    document.write("Welcome to JavaScript.");
}
```

ഇവിടെ, ഒരു ഫങ്ഷൻ നിർവചിക്കുവാനുള്ള കീവേഡ് 'function' ഉം ഫങ്ഷന്റെ പേര് print ഉം ആണ്. ഫങ്ഷന്റെ പേര് സാധുതയുള്ള ഏതൊരു ഐഡന്റിഫയറും ആകാം. C++ ൽ ഐഡന്റിഫയറുകൾ നാമകരണം ചെയ്യുന്ന അതേ നിയമങ്ങൾ ജാവാസ്ക്രിപ്റ്റിലും ഉപയോഗിക്കുന്നു. ഇവിടെ print( ) ഫങ്ഷനിൽ ഒരു പ്രസ്താവന മാത്രമേ ഉള്ളൂ, ആവശ്യമെങ്കിൽ ഒന്നിലധികം പ്രസ്താവനകൾ ഒരു ഫങ്ഷനിൽ ഉൾപ്പെടുത്താവുന്നതാണ്. ഒരു ഫങ്ഷൻ നിർവചിക്കുന്നത് കൊണ്ട് അത് സ്വയം പ്രവർത്തിക്കണം എന്നില്ല, അതിനെ വിളിക്കുമ്പോൾ മാത്രമേ അത് പ്രവർത്തിക്കൂ. ഇതിനർത്ഥം ഒരു വെബ് പേജിലെ ഒരു ഫങ്ഷൻ വിളിക്കാതിരിക്കുകയാണെങ്കിൽ ആ ഫങ്ഷൻ ഒരിക്കലും പ്രവർത്തിക്കുകയില്ല. ഫങ്ഷന്റെ പേര് ഉപയോഗിച്ച് താഴെ പറയുന്ന ഫങ്ഷനെ വിളിക്കാം.

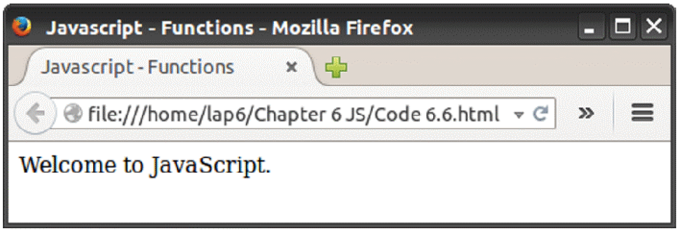
```
print();
```

ഫങ്ഷന്റെ പേരിന് ശേഷമുള്ള അർധവിരാമം ശ്രദ്ധിക്കുക. നമുക്ക് മുകളിലെ print( ) ഫങ്ഷൻ ഒരു HTML പേജിൽ ഉപയോഗിക്കുന്നത് എങ്ങനെ എന്ന് നോക്കാം.

ഉദാഹരണം 6.4: print() ഫംഗ്ഷൻ അടങ്ങിയ വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Functions</TITLE>
   <SCRIPT Language= "JavaScript">
      function print()
      {
         document.write("Welcome to JavaScript.");
      }
   </SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

HTML പേജിന്റെ ബോഡി ഭാഗത്തിൽ ഒന്നും എഴുതിയിട്ടില്ലെന്നത് ശ്രദ്ധിക്കുക. ഈ കോഡ് ബ്രൌസർ വിൻഡോയിലെ 'Welcome to JavaScript' പ്രദർശിപ്പിക്കില്ല. കാരണം ഫങ്ഷൻ നിർവചിച്ചിട്ടുണ്ടെങ്കിലും, അത് പേജിലെ ഏതെങ്കിലും സ്ഥലത്തുനിന്ന് വിളിച്ചിട്ടില്ല. അതിനാൽ, ഫങ്ഷൻ ഒരിക്കലും പ്രവർത്തിക്കുകയില്ല, സ്ക്രീനിൽ ഒന്നും പ്രദർശിപ്പിക്കുകയുമില്ല. ബ്രൌസർ ജാലകത്തിൽ "Welcome to JavaScript" എന്ന് ഔട്ട്പുട്ട് ലഭിക്കുന്നതിന് HTML പേജി ന്റെ ബോഡി ഭാഗം താഴെ കാണുന്ന രീതിയിൽ പരിഷ്കരിക്കേണ്ടതുണ്ട്.



ചിത്രം 6.3: ഫങ്ഷൻ ഉപയോഗിച്ചിരിക്കുന്ന വെബ് പേജ്

```
<BODY>
<SCRIPT Language= "JavaScript">
    print();
</SCRIPT>
</BODY>
```

ഇനി നമുക്ക് ജാവാ സ്ക്രിപ്റ്റ് ഫങ്ഷന്റെ വാക്യഘടന പരിചയപ്പെടാം

```
function function_name()
{
    statements;
}
```

ഇവിടെ function_name () എന്ന വരിയിൽ ഫങ്ഷൻ ഹെഡറും, ബ്രാക്കറ്റിലുള്ള കോഡ് ഫങ്ഷൻ ബോഡി എന്നു വിളിക്കുന്നു. C++ ൽ ഫങ്ഷന് റിട്ടേൺ ടൈപ്പ് ഉണ്ടെങ്കിലും ജാവാസ്ക്രിപ്റ്റിൽ ഫങ്ഷന് റിട്ടേൺ ടൈപ്പില്ല. ജാവാസ്ക്രിപ്റ്റിലും C++ ലേതുപോലെ നിങ്ങൾക്ക് ഫങ്ഷനിൽ നിന്ന് ചില മൂല്യങ്ങൾ തിരികെ നൽകാം. ഈ അധ്യായത്തിൽ ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗത്തെക്കുറിച്ചുള്ള അടിസ്ഥാന ആശയങ്ങൾ മാത്രം നൽകാൻ ഉദ്ദേശിച്ചിട്ടുള്ളതിനാൽ, ഇത്തരം കാര്യങ്ങൾ ഇവിടെ ചർച്ച ചെയ്യുന്നില്ല. മറ്റൊരു വ്യത്യാസം C++ൽ ഒരു ഫങ്ഷൻ നിർവചിക്കാൻ function എന്ന കീവേഡ് ഉപയോഗിക്കില്ല, എന്നാൽ ജാവാസ്ക്രിപ്റ്റിൽ function എന്ന കീവേഡ് ഉപയോഗിക്കുന്നു. HTML പേജിന്റെ ഹെഡ് ഭാഗത്തിനുള്ളിൽ ഫങ്ഷൻ നിർവചിക്കപ്പെട്ടിരിക്കുന്നതായി നിങ്ങൾ ശ്രദ്ധിച്ചിരിക്കാം. ഹെഡ് ഭാഗത്തിനുള്ളിൽ തന്നെ ഫങ്ഷൻ നിർവചിക്കണം എന്ന് നിർബന്ധമില്ല, ബോഡി ഭാഗത്തും ഫങ്ഷൻ നിർവചിക്കാം.

```
<BODY>
    <SCRIPT Language= "JavaScript">
        function print()
```

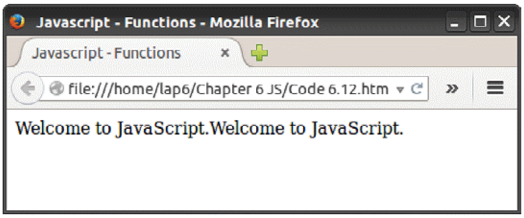
```
        {
            document.write("Welcome to JavaScript.");
        }
        print();
    </SCRIPT>
</BODY>
```

മുകളിൽ പറഞ്ഞിരിക്കുന്ന കോഡ് ഇതേ ഔട്ട്പുട്ട് നൽകുന്നത് ചിത്രം 6.3 ൽ കൊടുത്തിരിക്കുന്നു.

ഫങ്ഷൻ ബോഡി ഭാഗത്തിനകത്താണ് നിർവചിക്കപ്പെട്ടിരിക്കുന്നതെങ്കിൽപ്പോലും അതിന്റെ പ്രവർത്തിപ്പിക്കുന്നതിനുവേണ്ടി അതിനെ വിളിക്കണം എന്നത് ശ്രദ്ധിക്കുക. ഉദാഹരണത്തിന് താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് സ്ക്രീനിൽ ഒന്നും പ്രദർശിപ്പിക്കുന്നില്ല.

```
<BODY>
<SCRIPT Language= "JavaScript">
function print()
{
    document.write("Welcome to JavaScript.");
}
</SCRIPT>
</BODY>
```

ഒരു HTML പേജിൽ എവിടെയും ഒരു ഫങ്ഷൻ നിർവചിക്കപ്പെടാമെങ്കിലും, ഹെഡ് ഭാഗത്തിൽ ഫങ്ഷൻ നിർവചനം ഉൾപ്പെടുത്തുന്നതാണ് നല്ലത്.



ചിത്രം 6.4: ബ്രേക്ക് ഇല്ലാതെ രണ്ട് പ്രിന്റ് ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്ന വെബ് പേജ്

print() ഫങ്ഷൻ രണ്ടു തവണ വിളിച്ചിരിക്കുന്ന താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് പരിഗണിക്കുക, ഇത് ചിത്രം 6.4ൽ കാണുന്ന രീതിയിലുള്ള ഔട്ട്പുട്ട് നൽകുന്നു.

```
<BODY>
    <SCRIPT Language= "JavaScript">
        print();
        print();
    </SCRIPT>
</BODY>
```

ഔട്ട്പുട്ട് ചിത്രം 6.4 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ആയിരിക്കും. എന്തുകൊണ്ട്?

വാസ്തവത്തിൽ, നമ്മൾ രണ്ടുതവണ ഫങ്ഷനെ വിളിക്കുമ്പോൾ, അത് 'Welcome to JavaScript' എന്ന് രണ്ടുതവണ ബോഡി ഭാഗത്ത് എഴുതും. അതിനാൽ മുകളിലുള്ള കോഡിന്റെ ബോഡി ഭാഗം താഴെ എഴുതിയിരിക്കുന്നതിനു തുല്യമായിരിക്കും.

```
<BODY>
    Welcome to JavaScript.Welcome to JavaScript.
</BODY>
```

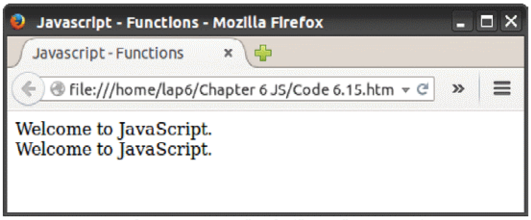
സന്ദേശം രണ്ട് വ്യത്യസ്ത വരികളിൽ പ്രദർശിപ്പിക്കുന്നതിന് നിർബന്ധമായും
 ടാഗ് ഉപയോഗിക്കണം.

```
<BODY>
    Welcome to JavaScript.<BR>Welcome to JavaScript.
</BODY>
```

ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് ഇതേ റിസൾട്ട് ലഭിക്കുന്നതിന് താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ മാറ്റം വരുത്താം.

```
<SCRIPT Language= "JavaScript">
    function print()
    {
        document.write("Welcome to JavaScript.<BR>");
    }
</SCRIPT>
```

മുകളിൽ പറഞ്ഞിരിക്കുന്ന HTML കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.5 ൽ കൊടുത്തിരിക്കുന്നു.



ചിത്രം 6.5: രണ്ട് പ്രിന്റ് ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്ന വെബ് പേജ്

നിങ്ങളുടെ പുരോഗതി അറിയാം

1. താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന ശരിയോ തെറ്റോ എന്ന് പറയുക.
   a. ഒരു വെബ് പേജ് ബ്രൗസറിൽ തുറക്കുമ്പോൾ ഫങ്ഷൻ സ്വമേധയാ പ്രവർത്തിക്കും.
   b. ഒരു ഫംഗ്ഷൻ സാധാരണയായി HTML പേജിന്റെ ഹെഡ്ഡ് ഭാഗത്തിൽ ഉൾപ്പെടുത്തുന്നു.
   c. ഒരു ഫംഗ്ഷൻ എത്ര തവണ വേണമെങ്കിലും വിളിക്കാം.
   d. ബോഡി ഭാഗത്തിനകത്ത് ഒരു ഫങ്ഷൻ നിർവചിച്ചിട്ടുണ്ടെങ്കിലും അത് വിളിച്ചിട്ടില്ലെങ്കിൽ പ്രവർത്തിക്കില്ല.
2. ജാവാസ്ക്രിപ്റ്റിൽ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന്റെ ഗുണദോഷങ്ങൾ എഴുതുക.

നമുക്കു ചെയ്യാം

1. താഴെക്കൊടുത്തിരിക്കുന്ന രണ്ട് HTML കോഡുകൾ (കോഡ് A ഉം കോഡ് B ഉം) പരിഗണിച്ച് ഔട്ട്പുട്ട് എഴുതുക.

കോഡ് *A*

```
<HTML>
<HEAD>
<SCRIPT Language= "JavaScript">
    function print()
    {
        document.write("Welcome to JavaScript");
    }
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    print();
    document.write("<BR>");
    print();
</SCRIPT>
</BODY>
</HTML>
```

കോഡ് *B*

```
<HTML>
<HEAD>
<SCRIPT Language= "JavaScript">
function print()
{
      document.write("Welcome to JavaScript");
}
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
      print();
</SCRIPT>
<BR>
<SCRIPT Language= "JavaScript">
      document.write("<BR>");
      print();
</SCRIPT>
</BODY>
</HTML>
```

2. ചിത്രം 6.6 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഔട്ട്പുട്ട് ലഭിക്കുന്നതിന് ഒരു HTML കോഡ് താഴെ കൊടുക്കുന്നു. കോഡിൽ `startGreen()`, `stopGreen()`, `startRed()` and `stopRed()` എന്നിങ്ങനെ നാലു ഫങ്ഷനുകൾ ഉണ്ട്. ഈ ഫങ്ഷനുകൾ HTML പേജിന്റെ ബോഡി ഭാഗത്തിൽ വ്യത്യസ്ത സ്ഥലങ്ങളിൽ നിന്നും വിളിക്കുന്നു. `stopGreen()` എന്ന ഫങ്ഷൻ ഒഴികെയുള്ള എല്ലാ ഫങ്ഷൻ നിർവചനങ്ങളുടെയും ബോഡി ശൂന്യമാണ്. നിങ്ങൾക്ക് ചിത്രത്തിൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഔട്ട്പുട്ട് ലഭിക്കുന്നതിന് മറ്റു ഫങ്ഷനുകളുടെ നിർവചനം പൂർത്തിയാക്കുക.

```
<HTML>
<HEAD>
<SCRIPT Language= "JavaScript">
      function startGreen()
      {
            .............
            .............
      }
      function stopGreen()
      {
            document.write("</FONT>");
      }
      function startRed()
      {
            .............
            .............
      }
      function stopRed()
      {
            .............
            .............
      }
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
      startGreen();
```

```
</SCRIPT>
This is in Green colour with size 5
<SCRIPT Language= "JavaScript">
     stopGreen();
</SCRIPT>
<SCRIPT Language= "JavaScript">
     startRed();
</SCRIPT>
This is in Red colour with size 3
<SCRIPT Language= "JavaScript">
     stopRed();
</SCRIPT>
<SCRIPT Language= "JavaScript">
     startGreen();
</SCRIPT>
This is in Green colour with size 5
<SCRIPT Language= "JavaScript">
     stopGreen();
</SCRIPT>
</BODY>
</HTML>
```

6.3 ജാവാസ്ക്രിപ്റ്റിലെ ഡാറ്റ ഇനങ്ങൾ (Data types in JavaScript)

മെമ്മറി ഫലപ്രദമായി ഉപയോഗപ്പെടുത്തുന്നതിന് പ്രോഗ്രാമിങ് ഭാഷകൾ ഡാറ്റയെ വ്യത്യസ്ത വിഭാഗങ്ങളായി തരം തിരിച്ചിരിക്കുന്നു. C++ ലെ അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ `int`, `char`, `float`, `double`, `void` തുടങ്ങിയവയാണെന്ന് നമ്മൾ മനസ്സിലാക്കിയിട്ടുണ്ട്. ഈ ഡാറ്റ ഇനങ്ങളൊടൊപ്പം, ടൈപ്പ് മോഡിഫയറുകളും C++ ൽ ഉണ്ടായിരുന്നു. അടിസ്ഥാന ഡാറ്റ ഇനങ്ങളുടെ എണ്ണം 3 ആയി പരിമിതപ്പെടുത്തിയിരിക്കുന്നതിലൂടെ ജാവാസ്ക്രിപ്റ്റ് ഇതിന്റെ സങ്കീർണത കുറയ്ക്കുന്നു. താഴെ പറയുന്നവയാണ് ജാവാസ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന മൂന്ന് അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ.

നമ്പർ (Number)

എല്ലാ നമ്പറുകളും ഈ വിഭാഗത്തിൽ വരുന്നു. പോസിറ്റിവ് സംഖ്യകളും, നെഗറ്റീവ് സംഖ്യകളും, എല്ലാ ഫ്ളോട്ട് വിലകളും (ഭിന്നസംഖ്യകൾ), പൂർണ സംഖ്യകളും നമ്പർ (number) ഡാറ്റ ഇനം ആയി പരിഗണിക്കപ്പെടും. അതിനാൽ, 27, -300, 1.89, -0.0082 എന്നിവ ജാവാസ്ക്രിപ്റ്റിലെ നമ്പർ ഡാറ്റ ഇനത്തിനുള്ള ഉദാഹരണങ്ങളാണ്.

സ്ട്രിംഗ് (String)

ഇരട്ട ഉദ്ധരണികൾക്കുള്ളിൽ ഉള്ള പ്രതീകങ്ങൾ, അക്കങ്ങൾ അല്ലെങ്കിൽ മറ്റേതെങ്കിലും ചിഹ്നങ്ങളുടെ കൂട്ടത്തെ ജാവാസ്ക്രിപ്റ്റിൽ ഒരു സ്ട്രിംഗായി പരിഗണിക്കും. ഉദാഹരണം "Kerala", "Welcome", "SCHOOL", "1234", "Mark20", "abc$" and "sanil@123".

ബൂളിയൻ (Boolean)

ഈ വിഭാഗത്തിൽ രണ്ട് മൂല്യങ്ങൾ മാത്രമേ ഉള്ളൂ, അവ true, false എന്നിവയാണ്. ഈ മൂല്യങ്ങൾ ഇരട്ട ഉദ്ധരണികളിലല്ല. ഇരട്ട ഉദ്ധരണികൾക്കുള്ളിൽ ആണെങ്കിൽ അവ സ്ട്രിംഗ് ആയിരിക്കും. ജാവാസ്ക്രിപ്റ്റ് കേസ് സെൻസിറ്റീവ് ആയതുകൊണ്ട് ബൂളിയൻ വിലകളെ പ്രതിനിധാനം ചെയ്യുന്നതിന് TRUE, FALSE എന്നിങ്ങനെ ഉപയോഗിക്കുവാൻ പാടില്ല.

6.4 ജാവാ സ്ക്രിപ്റ്റിലെ വേരിയബിളുകൾ (Variables in JavaSript)

വിലകൾ സംഭരിക്കുന്നതിന് വേരിയബിളുകൾ ഉപയോഗിക്കാമെന്ന് നമുക്കറിയാം. ജാവാ സ്ക്രിപ്റ്റിൽ **var** എന്ന കീവേഡ് ഉപയോഗിച്ചാണ് എല്ലാത്തരം വേരിയബിളുകളും പ്രഖ്യാപിക്കുന്നത്.

```
var  x;
```

ഇവിടെ 'x' എന്നത് വേരിയബിളിന്റെ പേരാണ്, ഒരു വേരിയബിളിന് C++ ലേതുപോലെ ഏതു പേരും ഉപയോഗിക്കാം. അത് ഒരു സാധുതയുള്ള ഐഡന്റിഫയർ ആയിരിക്കണമെന്നുമാത്രം. C++ ൽ വ്യത്യസ്ത തരം വേരിയബിളുകൾ പ്രഖ്യാപിക്കാൻ കീവേഡുകൾ `int`, `float`, `char` തുടങ്ങിയവ ഉപയോഗിച്ചു. പക്ഷേ, ജാവാസ്ക്രിപ്റ്റിൽ **var** എന്ന കീവേഡ്' മാത്രമാണ് എല്ലാത്തരം വേരിയബിളുകൾ പ്രഖ്യാപിക്കുന്നതിനും ഉപയോഗിക്കുന്നത്. ജാവാസ്ക്രിപ്റ്റിൽ വേരിയബിളിന് വില നൽകുമ്പോൾ മാത്രം നിർവചനം പൂർത്തിയാകുന്നു. ഒരു വേരിയബിളിന് മൂല്യം നൽകുമ്പോൾ മാത്രമാണ് വേരിയബിളിന്റെ തരം ജാവാസ്ക്രിപ്റ്റ് മനസ്സിലാക്കുന്നത്.

താഴെകൊടുത്തിരിക്കുന്ന പ്രസ്താവന പരിഗണിക്കുക.

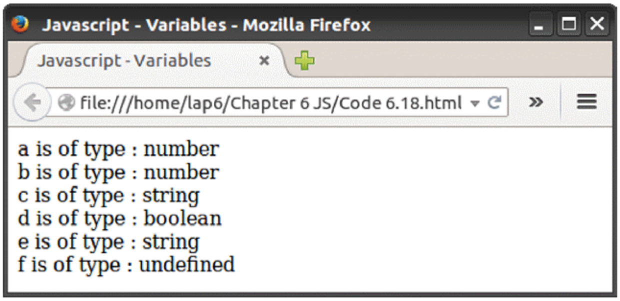
```
var x, y;

x = 25;

y = "INDIA";
```

മുകളിൽ പറഞ്ഞിരിക്കുന്ന ഉദാഹരണത്തിൽ, x എന്നത് നമ്പർ തരത്തിലും y സ്ട്രിംഗ് തരത്തിലുംപെട്ട വേരിയബിളുകൾ ആണ്. എന്നാൽ കീവേഡ് number, string എന്നൊന്നും ഇവിടെ ഉപയോഗിച്ചിട്ടില്ല. താഴെ കൊടുത്തിരിക്കുന്ന ഉദാഹരണത്തിൽ ജാവാ സ്ക്രിപ്റ്റ് എങ്ങനെയാണ് വേരിയബിളുകൾ നിർവചിക്കുന്നത് എന്ന് വിവരിക്കുന്നു.

ഉദാഹരണം 6.5 : വേരിയബിളുകളുടെ ഉപയോഗം വിവരിക്കുന്നു.

```
<HTML>
<HEAD> <TITLE>Javascript - Variables</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
   var a, b, c, d, e, f;
   a = 25;
   b = 18.5;
   c = "INDIA";
   d = true;
   e = "true";
   document.write("a is of type : ");
   document.write(typeof(a));
   document.write("<BR>b is of type : ");
   document.write(typeof(b));
   document.write("<BR>c is of type : ");
   document.write(typeof(c));
   document.write("<BR>d is of type : ");
   document.write(typeof(d));
   document.write("<BR>e is of type : ");
   document.write(typeof(e));
   document.write("<BR>f is of type : ");
   document.write(typeof(f));
</SCRIPT>
</BODY>
</HTML>
```

ഇവിടെ, `typeof()` എന്നൊരു ഫങ്ഷൻ പുതുതായി ഉപയോഗിച്ചിരിക്കുന്നു. പേര് സൂചിപ്പിക്കുന്നതുപോലെ ഒരു വേരിയബിളിന്റെ തരം കണ്ടുപിടിക്കുവാൻ ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഈ ഫങ്ഷനെ പിന്നീട് വിശദമായി പഠിക്കാം. മുകളിലുള്ള ഉദാഹരണത്തിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.6 ൽ കൊടുത്തിരിക്കുന്നു.



ചിത്രം 6.6: വേരിയബിളുകളുടെ തരം ഉപയോഗം പ്രദർശിപ്പിക്കുന്ന വെബ് പേജ്.

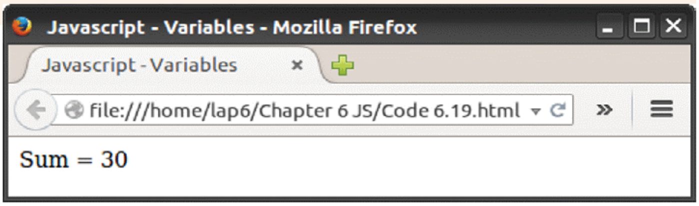
വേരിയബിൾ f പ്രഖ്യാപിച്ചിട്ടുണ്ടെങ്കിലും അതിന് ഒരു വില നൽകിയിട്ടില്ല. അതുകൊണ്ട് സ്ക്രിപ്റ്റ് എഞ്ചിന് അതിന്റെ തരം മനസ്സിലാക്കുവാനോ പ്രദർശിപ്പിക്കുവാനോ സാധിക്കു

ന്നില്ല. നിർവചിക്കാത്ത വേരിയബിളിനെ പ്രതിനിധാനം ചെയ്യുന്നതിനായി സ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന ഒരു പ്രത്യേക ഡാറ്റ ഇനമാണ് undefined. ഒരൊറ്റ var കീവേഡ് ഉപയോഗിച്ച് എത്ര വേരിയബിളുകൾ വേണമെങ്കിലും പ്രഖ്യാപിക്കാനാവും (ഡിക്ലയർ ചെയ്യാനാവും). വേരിയബിളുകൾ കോമ (,) ഉപയോഗിച്ച് വേർതിരിക്കേണ്ടതാണ്. വേരിയബിളുകൾ ഉപയോഗിക്കുന്ന ഒരു ജാവാസ്ക്രിപ്റ്റിന്റെ പ്രവർത്തനം നമുക്ക് പരിഗണിക്കാം.

ഉദാഹരണം 6.6: രണ്ട് സംഖ്യകളുടെ തുക കണ്ടു പിടിക്കുന്നതിനുള്ള വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD><TITLE>Javascript - Variables</TITLE>
<SCRIPT Language= "JavaScript">
function add()
{
    var m, n, sum;
    m = 20;
    n = 10;
    sum = m + n;
    document.write("Sum = ");
    document.write(sum);
}
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    add();
</SCRIPT>
</BODY>
</HTML>
```

മുകളിൽ പറഞ്ഞ ഉദാഹരണത്തിൽ, ബോഡി ഭാഗത്തിൽ ഒരു ജാവാസ്ക്രിപ്റ്റ് പ്രസ്താവന മാത്രമേ അടങ്ങിയിട്ടുള്ളൂ, അത് add() എന്ന ഫങ്ഷനെ വിളിക്കുന്നു. ഫങ്ഷൻ പ്രവർത്തനത്തിന്റെ ഫലം ചിത്രം 6.7ൽ കാണിച്ചിരിയ്ക്കുന്നു.



ചിത്രം 6.7: രണ്ടു സംഖ്യകളുടെ തുക കണ്ടുപിടിക്കുന്നതിനുള്ള വെബ്പേജ്

6.5 ജാവാസ്ക്രിപ്റ്റിലെ ഓപ്പറേറ്ററുകൾ (Operators in JavaScript)

ജാവാ സ്ക്രിപ്റ്റിലെ മിക്കവാറും എല്ലാ ഓപ്പറേറ്ററുകളും C++ ലേതിന് സമാനമാണ്. അവ ഏതെല്ലാമാണെന്ന് നോക്കാം

6.5.1 അരിത്തമാറ്റിക് ഓപ്പറേറ്ററുകൾ (Arithmetic Operatiors)

ജാവാസ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന അരിത്തമാറ്റിക് ഓപ്പറേറ്ററുകൾ പട്ടിക 6.1 കാണിക്കുന്നു.

ഓപ്പറേഷൻ	വിവരണം	ഉദാഹരണം	y യുടെ മൂല്യം	ഉത്തരം (x)
+	സങ്കലനം	x = y + 10	15	25
–	വ്യവകലനം	x = y - 10	15	5
*	ഗുണനം	x = y * 3	15	45
/	ഹരണം	x = y / 2	15	7.5
%	മോഡുലസ് ഹരണത്തിലെ ശിഷ്ടം	x = y % 2	15	1
++	വർദ്ധനവ് Increment	x = ++y x = y++	15 15	16 15
--	കുറവ് Decrement	x = --y x = y--	15 15	14 15

പട്ടിക 6.1: അരിത്തമാറ്റിക് ഓപ്പറേറ്ററുകൾ

മുകളിലുള്ള പട്ടികയിലെ എല്ലാ അരിത്തമാറ്റിക് ഓപ്പറേററുകളും C++ ൽ പ്രവർത്തിക്കുന്നപോലെതന്നെ പ്രവർത്തിക്കുന്നുവെന്ന് കാണാം.

6.5.2 വിലനൽകൽ ഓപ്പറേറ്ററുകൾ (Assignment Operators)

ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ വിലനൽകൽ ഓപ്പറേറ്ററുകളുടെ ഉപയോഗരീതി പട്ടിക 6.2ൽ കാണിച്ചിരിക്കുന്നു.

ഓപ്പറേറ്റർ	വിവരണം	ഉദാഹരണം	a യുടെ വില	b യുടെ വില	ഉത്തരം (a)
=	വില നൽകുന്നു	a = b	10	3	3
+=	സങ്കലനത്തിനുശേഷം വില നൽകുന്നു	a+=b	10	3	13
-=	വ്യവകലനത്തിനുശേഷം വില നൽകുന്നു	a-=b	10	3	7
*=	ഗുണനത്തിനുശേഷം വില നൽകുന്നു	a*=b	10	3	30
/=	ഹരണത്തിനുശേഷം വില നൽകുന്നു	a/=b	10	3	3.33
%=	മോഡുലസ് പ്രവർത്തനത്തിനുശേഷം വില നൽകുന്നു	a%=b	10	3	1

പട്ടിക 6.2: ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ വിലനൽകൽ ഓപ്പറേറ്ററുകൾ

മുകളിലുള്ള പട്ടികയിൽ നിന്ന് ഓരോ ഓപ്പറേറ്ററുകളുടെയും പ്രവർത്തനം മനസ്സിലാക്കുക വളരെ എളുപ്പമാണ്. a യുടെയും b യുടെയും വിലകളുപയോഗിച്ച് 'ഉദാഹരണം ' എന്ന

കോളത്തിലെ പ്രസ്താവന പ്രവർത്തിപ്പിച്ച ശേഷമുള്ള ഫലം 'ഉത്തരം' എന്ന കോളത്തിൽ കാണിച്ചിരിക്കുന്നു.

6.5.3 റിലേഷണൽ ഓപ്പറേറ്റർ (താരതമ്യ ഓപ്പറേറ്റർ) (Relational/Comparison operator)

ജാവാസ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന വിവിധ തരം താരതമ്യ ഓപ്പറേറ്ററുകൾ പട്ടിക 6.3 ൽ കാണിച്ചിരിക്കുന്നു.

ഓപ്പറേറ്റർ	വിവരണം	ഉദാഹരണം	a യുടെ വില	b യുടെ വില	ഉത്തരം (a)
==	തുല്യം	a==b	10	3	false
!=	തുല്യമല്ല	a!=b	10	3	true
<	കുറവ്	a<b	10	3	false
<=	കുറവോ അല്ലെങ്കിൽ തുല്യമോ	a<=b	10	3	false
>	കൂടുതൽ	a>b	10	3	true
>=	കൂടുതലോ അല്ലെങ്കിൽ തുല്യമോ	a>=b	10	3	true

പട്ടിക 6.3: റിലേഷണൽ ഓപ്പറേറ്റർ

ഒരു താരതമ്യ/റിലേഷണൽ ഓപ്പറേഷന്റെ ഫലം ശരിയോ തെറ്റോ എന്ന് പട്ടികയിൽ നിന്ന് വ്യക്തമാണ്. ഈ ഓപ്പറേറ്ററുകൾ രണ്ട് വശത്തും ഉള്ള മൂല്യങ്ങൾ താരതമ്യം ചെയ്യുകയും അതിനനുസരിച്ച് ഫലം നൽകുകയും ചെയ്യുന്നു.

6.5.4 ലോജിക്കൽ ഓപ്പറേറ്റർ (Logical Operators)

ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ ലോജിക്കൽ ഓപ്പറേറ്ററുകളുടെ പട്ടിക 6.4 ൽ ഉദാഹരണത്തോടൊപ്പം കാണിച്ചിരിക്കുന്നു.

ഓപ്പറേറ്റർ	വിവരണം	ഉദാഹരണം	a യുടെ വില	b യുടെ വില	ഉത്തരം (a)
&&	ആന്റ്	a && b	true	false	false
||	ഓർ	a || b	true	false	true
!	നോട്ട്	!a	true		false

പട്ടിക 6.4: ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ

മുകളിൽ കൊടുത്തിരിക്കുന്ന പട്ടികയിൽ നിന്ന്, ഈ ഓപ്പറേറ്ററുകൾ എല്ലാം തന്നെ C++ ലേതിന് സമാനമാണെന്നത് വളരെ വ്യക്തമാണ്. വിവിധ ആവശ്യങ്ങൾക്കായി മറ്റ് ധാരാളം ഓപ്പറേറ്ററുകൾ ജാവാസ്ക്രിപ്റ്റ് നൽകുന്നു. ജാവാസ്ക്രിപ്റ്റിനെക്കുറിച്ചുള്ള നമ്മുടെ ചർച്ച ഈ അധ്യായത്തിൽ മാത്രം ഒതുങ്ങിയിരിക്കുന്നതിനാൽ അത്തരം

ഓപ്പറേറ്ററുകളെക്കുറിച്ചുള്ള ചർച്ച ഈ പുസ്തകത്തിന്റെ പരിധിക്കു പുറത്താണ്. കൂടാതെ, മുകളിൽ വിവരിച്ച ഓപ്പറേറ്ററുകൾ ഒരു തുടക്കക്കാരനു വേണ്ട എല്ലാ പ്രവർത്തനങ്ങളും നടത്തുന്നതിന് പര്യാപ്തമാണ്. എന്നിരുന്നാലും, താഴെ പറയുന്ന സ്ട്രിംഗ് ഓപ്പറേറ്റർ വിവിധ സാഹചര്യങ്ങളിൽ നിങ്ങൾക്ക് ഉപയോഗപ്രദമാകും. ഈ ഓപ്പറേറ്റർ C++ ൽ ലഭ്യമല്ല.

6.5.5 സ്ട്രിംഗ് അഡീഷൻ ഓപറേറ്റർ (+) (String addition operator (+))

രണ്ട് സംഖ്യകൾ കൂട്ടുന്നതിന് + ഓപ്പറേറ്റർ ഉപയോഗിക്കാമെന്ന് നമുക്ക് അറിയാം. അതേ ഓപ്പറേറ്ററായ '+' രണ്ട് സ്ട്രിംഗുകൾ കൂട്ടിച്ചേർക്കുന്നതിനും ഉപയോഗിക്കാം. കൺകാറ്റ് നേഷൻ (concatenation) എന്നതിനർഥം രണ്ട് സ്ട്രിംഗുകൾ കൂട്ടിചേർക്കുക എന്നാണ്.

ഉദാഹരണം

```
var x, y;
x = "A good beginning ";
y = "makes a good ending.";
z = x + y;
```

+ ഓപ്പറേറ്റർ രണ്ട് സ്ട്രിംഗുകൾ കൂട്ടിച്ചേർക്കും, അങ്ങനെ വേരിയബിൾ z ന്റെ മൂല്യം 'A good beginning makes a good ending' എന്നാകും. ഒരേ ഓപ്പറേറ്റർ വ്യത്യസ്ത ഓപ്പറന്റുകളുടെ തരം മനസ്സിലാക്കിയാണ് പ്രവർത്തിക്കുന്നത്. ഓപ്പറന്റുകൾ സംഖ്യ ആണെങ്കിൽ, അവയുടെ തുക കണ്ടുപിടിക്കും. മറിച്ച് സ്ട്രിംഗുകൾ ആണെങ്കിൽ അവ കൂട്ടിച്ചേർക്കും. ഇനി, താഴെ പറഞ്ഞിരിക്കുന്ന കോഡിൽ z ന്റെ മൂല്യം പ്രവചിക്കുക?

```
var  x, y;
x = "23";
y = 5;
z = x + y;
```

ഇതിന്റെ ഉത്തരം 235 ആണ്. ഓപ്പറന്റുകളിൽ ഏതെങ്കിലും ഒരെണ്ണം സ്ട്രിങ്ങാണെങ്കിൽ + ഓപ്പറേറ്റർ എല്ലാ ഓപ്പറന്റുകളെയും സ്ട്രിങ്ങായി കരുതി കൂട്ടിച്ചേർക്കുന്നു. അങ്ങനെ z ന്റെ വിലയായി 235 ലഭിക്കുന്നു. x ന്റെയും y യുടെയും തുക സംഖ്യാരൂപത്തിൽ കാണണമെന്നുകരുതുക. ഇതിനായി താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിൽ പ്രസ്താവന മാറ്റിയെഴുതാം.

```
z = Number(x) + y;
```

മുകളിൽ പറഞ്ഞിരിക്കുന്ന പ്രസ്താവന പ്രവർത്തിച്ച ശേഷം വേരിയബിൾ z ന്റെ വില 28 ആയിരിക്കും. Number() എന്നത് ജാവാ സ്ക്രിപ്റ്റിലെ ഒരു ഫംഗ്ഷൻ ആണ്, അത് സ്ട്രിംഗ് തരം ഡാറ്റയെ നമ്പറായി മാറ്റുന്നു. ഈ അധ്യായത്തിലെ ചില ഉദാഹരണങ്ങളിൽ ഈ ഫംഗ്ഷൻ ഉപയോഗിക്കുന്നുണ്ട്.

നിങ്ങളുടെ പുരോഗതി അറിയാം

1. നമ്പർ, സ്ട്രിംഗ്, _______ എന്നിവയാണ് ജാവാസ്ക്രിപ്റ്റിലെ അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ.
2. 'true' ഒരു _______ ഇനം ഡാറ്റയാണ്.
3. 'false' ഒരു _______ ഇനം ഡാറ്റയാണ്.
4. ജാവാസ്ക്രിപ്റ്റിൽ ഒരു വേരിയബിൾ പ്രഖ്യാപിക്കാൻ വേണ്ട കീവേഡ് _______ ആണ്.
5. ജാവാസ്ക്രിപ്റ്റിൽ ഡാറ്റ ഇനം അറിയാൻ ഉപയോഗിക്കുന്നതിനുള്ള ഫങ്ഷൻ _______ ആണ്.
6. % ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നത് എന്തിനാണ്?
7. ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ ഏവ?

6.6 ജാവാസ്ക്രിപ്റ്റിലെ കൺട്രോൾ സ്ട്രക്ചറുകൾ (Control structures in JavaScript)

ഒരു പ്രോഗ്രാമിന്റെ പ്രവർത്തനത്തിന്റെ നിയന്ത്രണത്തിന് കൺട്രോൾ സ്ട്രക്ചറുകൾ ഉപയോഗിക്കുന്നു. C++ ൽ പഠിച്ചിട്ടുള്ള എല്ലാ കൺട്രോൾ സ്ട്രക്ചറുകളും ജാവാസ്ക്രിപ്റ്റിലും വ്യത്യാസമില്ലാതെ ഉപയോഗിക്കാവുന്നതാണ്. ജാവാ സ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന വിവിധ കൺട്രോൾ സ്ട്രക്ചറുകൾ ഉദാഹരണസഹിതം നമുക്ക് പരിശോധിക്കാം.

6.6.1 if

എല്ലാ പ്രോഗ്രാമിങ് ഭാഷയിലും ഏറ്റവുമധികം ഉപയോഗിക്കപ്പെടുന്ന കൺട്രോൾ സ്ട്രക്ചറാണിത്. ചില വ്യവസ്ഥകൾ അടിസ്ഥാനമാക്കി ഒരു പ്രസ്താവന അല്ലെങ്കിൽ ഒരു കൂട്ടം പ്രസ്താവനകൾ പ്രവർത്തിപ്പിക്കാൻ ഇത് ഉപയോഗിക്കുന്നു. പട്ടിക 6.5 ൽ നൽകിയിരിക്കുന്നതുപോലെ രണ്ട് രീതിയിൽ ഇത് ഉപയോഗിക്കാം

if ന്റെ വാക്യഘടന	if ...... ന്റെ വാക്യഘടന

if (test_expression)
{
    statements;
}
	if (test_expression)
	{
	    statements;
	}
	else
	{
	    statements;
	}

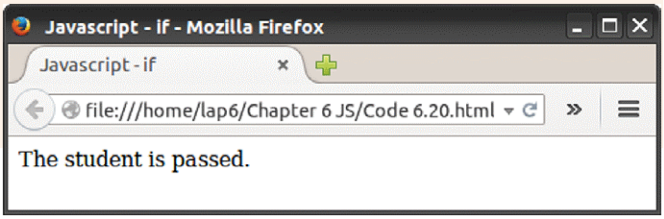
പട്ടിക 6.5: if, else ഓടുകൂടിയ if പ്രസ്താവനയുടെ വാക്യഘടന

if, if .... else പ്രസ്താവനയുടെ ഉപയോഗം കാണിക്കുന്ന ഒരു ഉദാഹരണം നോക്കാം.

ഉദാഹരണം 6.12: ഒരു വിദ്യാർഥി ജയിച്ചോ ഇല്ലയോ എന്നു പരിശോധിക്കുന്നതിനുള്ള ഒരു വെബ് പേജ് നിർമാണം.

ഉദാഹരണം 6.7: രണ്ട് സംഖ്യകളുടെ തുക കണ്ടു പിടിക്കുന്നതിനുള്ള വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - if</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    var score;
    score = 35;
    if (score < 30)
    {
        document.write("The student is failed.");
    }
    else
    {
        document.write("The student is passed.");
    }
</SCRIPT>
</BODY>
</HTML>
```



ചിത്രം 6.8: if പ്രസ്താവന വ്യക്തമാക്കുന്ന വെബ് പേജ്

മുകളിൽ കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാം കോഡ് if - else പ്രസ്താവന ഉപയോഗിക്കുന്ന വിധം കാണിക്കുന്നു. പ്രോഗ്രാമിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.8 ൽ നൽകിയിരിക്കുന്നു. നിങ്ങൾ 30 ൽ താഴെയുള്ള സംഖ്യ score ന്റെ വിലയായി നൽകി ഫലത്തിൽ വരുന്ന മാറ്റം ശ്രദ്ധിക്കുക.

6.6.2 സ്വിച്ച് (switch)

സ്വിച്ച് ഒരു 'മൾട്ടി ബ്രാഞ്ചിങ്' പ്രസ്താവനയാണ്. ഇതുപയോഗിച്ച് ഒരു എക്സ്പ്രഷന്റെ വില അടിസ്ഥാനമാക്കി വ്യത്യസ്ത പ്രോഗ്രാം കോഡുകൾ പ്രവർത്തിപ്പിക്കുന്നതിനായി തിരഞ്ഞെടുക്കാം. സ്വിച്ചിന്റെ വാക്യഘടന

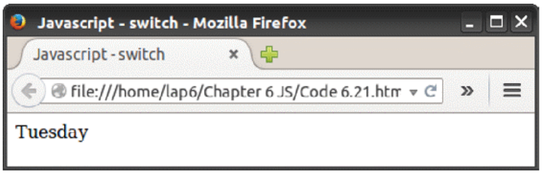
```
    switch (പരിശോധന പ്രയോഗങ്ങൾ)
    {
        case value1:
                    പ്രസ്താവനകൾ;
                    break;
        case value2:
                    പ്രസ്താവനകൾ;
                    break;
        .................
        .................
        default:
                    പ്രസ്താവനകൾ;
    }
```

എക്സ്പ്രഷന്റെ വില അടിസ്ഥാനമാക്കി ഉചിതമായ കേസ് ഭാഗം പ്രവർത്തിക്കുന്നു. ഇവിടെ എക്സ്പ്രഷൻ ഒരു വേരിയബിളിന്റെ പേര് ആവാം. നൽകുന്ന നമ്പറിന് അനുയോജ്യമായ ദിവസം പ്രിന്റ് ചെയ്യുന്ന ഒരു HTML കോഡ് താഴെ കൊടുത്തിരിക്കുന്നു.

ഉദാഹരണം 6.8: നൽകുന്ന നമ്പറിന് അനുയോജ്യമായ ദിവസം പ്രിന്റ് ചെയ്യുന്ന വെബ് പേജ് നിർമിക്കുന്നതിനുള്ള HTML കോഡ്

```
<HTML>
<HEAD> <TITLE>Javascript - switch</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    var d;
    d = 3;
    switch(d)
    {
        case 1:
            document.write("Sunday");
            break;
        case 2:
            document.write("Monday");
            break;
        case 3:
            document.write("Tuesday");
            break;
        case 4:
            document.write("Wednesday");
            break;
        case 5:
            document.write("Thursday");
            break;
        case 6:
            document.write("Friday");
            break;
        case 7:
            document.write("Saturday");
            break;
        default:
            document.write("Invalid Day");
    }
</SCRIPT>
</BODY>
</HTML>
```

ഈ കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.9 ൽ നൽകിയിരിക്കുന്നു.



ചിത്രം 6.9: സ്വിച്ച് പ്രസ്താവനയുടെ ഉപയോഗം വ്യക്തമാക്കുന്ന വെബ് പേജ്

6.6.3 for ലൂപ്പ് (for loop)

ഒരു കൂട്ടം നിർദ്ദേശങ്ങൾ ആവർത്തിച്ച് നടപ്പാക്കാൻ for ലൂപ്പ് ഉപയോഗിക്കുന്നു. for ലൂപ്പിന്റെ ആവർത്തനം നിയന്ത്രിക്കുന്നതിനായി ഒരു ലൂപ്പ് വേരിയബിൾ ഉപയോഗിയ്ക്കുന്നു.

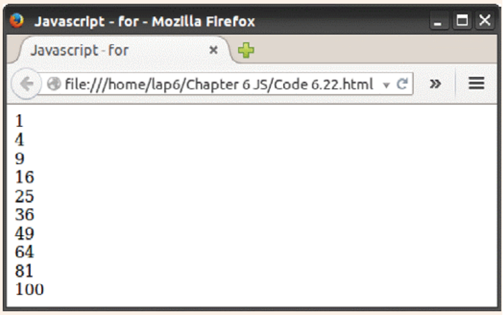
for ലൂപ്പിന്റെ വാക്യഘടന

```
for(initialisation; test_expression; update_statement)
{
    പ്രസ്താവനകൾ;
}
```

initialisation ലൂപ്പ് വേരിയബിളിന് ആദ്യ വില നൽകുവാൻ ഉപയോഗിക്കുന്നു. test_expression ലൂപ്പ് തുടരണോ വേണ്ടയോ എന്ന് പരിശോധിക്കാനുള്ള വ്യവസ്ഥയും update_statement ലൂപ്പ് വേരിയബിളിന്റെ വില വർധിപ്പിക്കുകയോ കുറയ്ക്കുകയോ ചെയ്യുന്നതിനുമുള്ള പ്രസ്താവനയുമാണ്. താഴെ പറയുന്ന ഉദാഹരണം ലൂപ്പിന്റെ പ്രവർത്തനങ്ങൾ വിശദീകരിക്കുന്നു.

ഉദാഹരണം 6.9: ആദ്യത്തെ 10 നമ്പറുകളുടെ വർഗങ്ങൾ പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - for</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    var i, s;
    for (i=1; i<=10; i++)
    {
        s = i*i;
        document.write(s);
        document.write("<BR>");
    }
</SCRIPT>
</BODY>
</HTML>
```

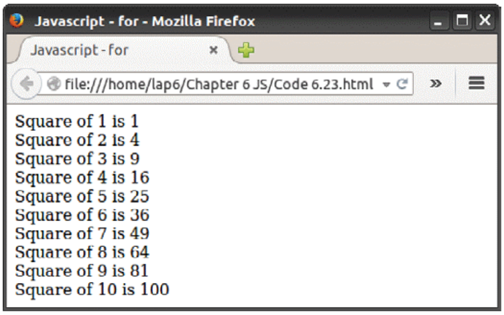


ചിത്രം 6.10: വേരിയബിളുകളുടെ തരം ഉപയോഗം പ്രദർശിപ്പിക്കുന്ന വെബ് പേജ്.

ഇതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.10 ൽ കൊടുത്തിരിക്കുന്നു.

ചിത്രം 6.11: ഔട്ട്പുട്ട് 'Square of 1 is 1' എന്ന് ലഭിക്കുന്നതിന് മുകളിലുള്ള കോഡ് നമുക്ക് പരിഷ്കരിക്കാം.

```
    for (i=1; i<=10; i++)
    {
        s = i*i;
        document.write("Square
of " + i + " is " + s);
        document.write("<BR>");
    }
```



ചിത്രം 6.11: നമ്പറുകളുടെ വർഗ്ഗം പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ് പേജ്

ഔട്ട്പുട്ട് നിർമ്മിക്കുന്നതിന് സ്ട്രിംഗ് കൂട്ടി ച്ചേർക്കുന്നതിനുള്ള (സ്ട്രിംഗ് അഡിഷൻ) ഓപ്പറേറ്റർ ആയി + ഉപയോഗിക്കുന്നു.

6.6.4 while ലൂപ്പ്

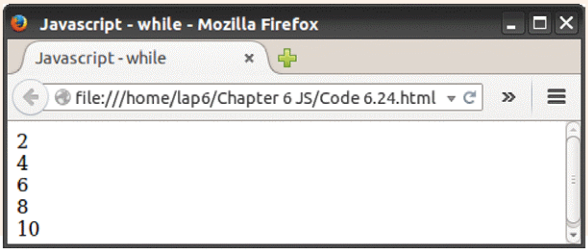
ഒരു വ്യവസ്ഥയുടെ അടിസ്ഥാനത്തിൽ ഒരു കൂട്ടം പ്രസ്താവനകൾ ആവർത്തിച്ച് പ്രവർത്തിപ്പിക്കുന്നതിനാണ് ലൂപ്പ് എന്ന്പറയുന്നത്.

```
    while (test_expression)
    {
        statements;
    }
```

ഇവിടെ `test_expression` എന്നത് ഒരു വ്യവസ്ഥയാണ്. വ്യവസ്ഥ ശരിയായിരിക്കുന്നിടത്തോളം ലൂപ്പിനുള്ളിലെ പ്രസ്താവനകൾ തുടർച്ചയായി പ്രവർത്തിച്ചുകൊണ്ടിരിക്കും. താഴെക്കാണുന്ന ഉദാഹരണം 10 വരെയുള്ള ഇരട്ട സംഖ്യകൾ പ്രദർശിപ്പിക്കുന്നു.

ഉദാഹരണം 6.10: പത്തു വരെയുള്ള ഇരട്ട സംഖ്യകൾ പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ്പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - while</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    var i;
    i = 2;
    while (i<=10)
    {
        document.write(i);
        document.write("<BR>");
        i += 2;
    }
</SCRIPT>
</BODY>
</HTML>
```



ചിത്രം 6.12: while ലൂപ്പിന്റെ ഉപയോഗത്തെ വ്യക്തമാക്കുന്ന വെബ് പേജ്

തന്നിരിക്കുന്ന കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.12 ൽ നൽകിയിരിക്കുന്നു.

ജാവാസ്ക്രിപ്റ്റിൽ C++ ലേതിന് സമാനമായ do while ലൂപ്പ് ലഭ്യമാണ്. മിക്കവാറും എല്ലാ ജോലികളും for ലൂപ്പിലും while ലൂപ്പിലും ചെയ്യാമെന്നതിനാൽ ഈ അധ്യായത്തിൽ നമ്മൾ മറ്റ് ലൂപ്പുകൾ ചർച്ച ചെയ്യുന്നില്ല.

ഉദാഹരണം 6.10 ൽ document.write("<BR>"); എന്ന പ്രസ്താവന ഒഴിവാക്കുകയാണെങ്കിൽ എന്ത് ഔട്ട്പുട്ട് ലഭിക്കും;

നമുക്കു ചെയ്യാം

നിങ്ങളുടെ പുരോഗതി അറിയാം

1. ഒരു കൂട്ടം പ്രസ്താവനകൾ തിരഞ്ഞെടുത്ത് പ്രവർത്തിപ്പിക്കുന്നതിന് ________ ഉം ________ ഉം കൺട്രോൾ സ്ട്രക്ചറുകൾ ഉപയോഗിക്കുന്നു.
2. ജാവാസ്ക്രിപ്റ്റിലെ ലൂപ്പിങ് പ്രസ്താവനകൾക്ക് ഉദാഹരണങ്ങൾ എഴുതുക.
3. ____________ ഒരു മൾട്ടി ബ്രാഞ്ചിങ് പ്രസ്താവനയാണ്.
4. ശരിയോ തെറ്റോ എന്നെഴുതുക.

   a. break പ്രസ്താവന switch ബ്ലോക്കിനുള്ളിൽ ഉപയോഗിക്കുന്നു.

   b. switch, പ്രസ്താവന ഉപയോഗിക്കുകയാണെങ്കിൽ, ചുരുങ്ങിയത് ഒരു തവണയെങ്കിലും അതിൽ break പ്രസ്താവന ഉപയോഗിച്ചിരിക്കണം.

   c. if-else ഉപയോഗിച്ച് എഴുതിയ പ്രോഗ്രാമുകൾ എല്ലാം സ്വിച്ച് പ്രസ്താവന ഉപയോഗിച്ച് മാറ്റി എഴുതാം.

   d. switch പ്രസ്താവന ഉപയോഗിച്ച് എഴുതിയ പ്രോഗ്രാമുകൾ എല്ലാം if-else ഉപയോഗിച്ച് മാറ്റി എഴുതാം.
5. for ലൂപ്പും while ലൂപ്പും തമ്മിലുള്ള വ്യത്യാസം എന്താണ്?

6.7 അന്തർ നിർമിത ഫങ്ഷനുകൾ (Built-in function)

ജാവാസ്ക്രിപ്റ്റിൽ ധാരാളം അന്തർനിർമിത ഫങ്ഷനുകൾ ലഭ്യമാണ്. ഫങ്ഷനുകളെ മെത്തേഡ് എന്നും വിളിക്കുന്നു. സാധാരണയായി ഉപയോഗിക്കുന്ന ഏതാനും ഫങ്ഷനുകളെക്കുറിച്ച് ചർച്ച ചെയ്യാം.

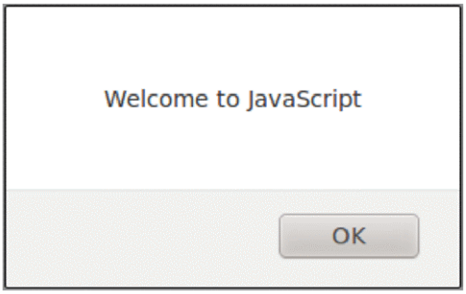
a. alert() fuction

സ്ക്രീനിൽ ഒരു സന്ദേശം പ്രദർശിപ്പിക്കാൻ ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു.

ഉദാഹരണത്തിന് :

```
alert("Welcome to JavaScript");
```

എന്ന പ്രസ്താവന ഉപയോഗിച്ച് ചിത്രം 6.13 ൽ കാണിച്ചിരിക്കുന്ന സന്ദേശ ജാലകം ബ്രൗസറിൽ കാണിക്കാം. ഡാറ്റയുടെ സാധുത ഉറപ്പുവരുത്തുന്ന സമയത്ത് ഉപയോക്താവിന് സന്ദേശം നൽകുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു.



ചിത്രം 6.13: സന്ദേശജാലകം

b. `isNaN()` fuction

തന്നിരിക്കുന്ന വില ഒരു സംഖ്യയാണോ അല്ലയോ എന്നു പരിശോധിക്കുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഈ ഫങ്ഷനിൽ, `NaN` എന്നത് നോട്ട് എ നമ്പർ എന്ന് സൂചിപ്പിക്കുന്നു. നൽകിയിരിക്കുന്ന വില ഒരു സംഖ്യയല്ലെങ്കിൽ ഫങ്ഷൻ `true` എന്ന് തിരികെ നൽകുന്നു. ഉദാഹരണത്തിന്,

1. `isNaN("welcome");`
2. `isNaN("A123");`
3. `isNaN("Score50");`
4. `isNaN("A");`

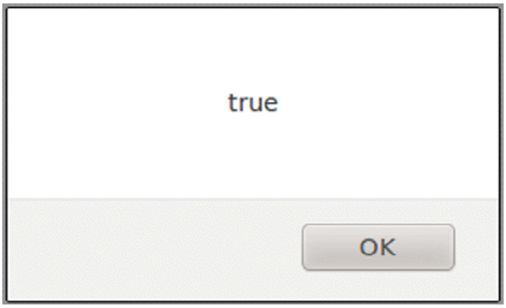
താഴെ പറയുന്ന പ്രസ്താവനകൾ `false` എന്ന വില തിരികെ നൽകുന്നു.

1. `isNaN("13");`
2. `isNaN(13);`
3. `isNaN("13.5");`
4. `isNaN("0.123");`

താഴെ പറയുന്ന പ്രസ്താവന ചിത്രം 6.14 ൽ കാണിച്ചിരിക്കുന്ന സന്ദേശ ജാലകം ബ്രൗസറിൽ പ്രദർശിപ്പിക്കുന്നു.

```
alert(isNaN("A"));
```

ഡാറ്റയുടെ സാധൂകരണത്തിനായി ഈ ഫങ്ഷൻ വളരെ ഉപയോഗപ്രദമാണ്. ഉദാഹരണത്തിന്, ഒരു വിദ്യാർഥിയുടെ പ്രായം നൽകുന്നതിന് വെബ് പേജിൽ ഒരു ടെക്സ്റ്റ് ബോക്സ് ഉണ്ട് എന്ന് കരുതുക, അബദ്ധത്തിൽ അതിൽ സംഖ്യക്ക് പകരം ഒരു ക്യാരക്ടർ നൽകിയേക്കാം. അപ്പോൾ ഈ ഫങ്ഷൻ, ഇൻപുട്ട് നൽകിയത് ഒരു നമ്പർ ആണോയെന്ന് പരിശോധിക്കുകയും ഇത് ഒരു സംഖ്യയല്ലെങ്കിൽ, `alert()` ഫങ്ഷൻ ഉപയോഗിച്ച് ഒരു സന്ദേശം പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു.



ചിത്രം 6.14: `isNaN()` ന്റെ ഔട്ട്പുട്ട്

c. toUpperCase() fuction

ഈ ഫങ്ഷൻ തന്നിരിക്കുന്ന സ്ട്രിംഗിനെ അപ്പർ കേസാക്കി മാറ്റുന്നു. താഴെ തന്നിരിക്കുന്ന ഉദാഹരണം നോക്കുക.

```
var x, y;
x = "JavaScript";
y = x.toUpperCase();
alert(y);
```

ഈ കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.15 ൽ കാണിച്ചിരിക്കുന്നു. toUpperCase() എന്ന ഫങ്ഷൻ സ്ട്രിംഗ് വേരിയബിൾ x ന്റെ പേരിനോട് ചേർത്ത് വിളിക്കുന്നു. അതായത് toUpperCase() എന്നത് സ്ട്രിംഗിനെ വലിയ അക്ഷരമാക്കി തിരികെ നൽകുന്നു. ജാവസ്ക്രിപ്റ്റ് കേസ് സെൻസിറ്റീവ് ഭാഷയാണ്. അതിനാൽ കോഡിൽ ഫങ്ഷൻ ഏതു തരം അക്ഷരത്തിലാണോ അതേ രീതിയിൽ തന്നെ ഉപയോഗിക്കണം.

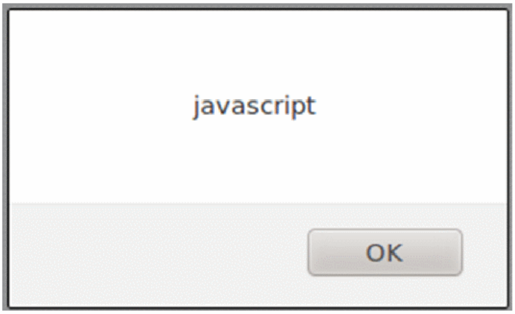
ചിത്രം 6.15: LowerCase() *ന്റെ ഔട്ട്പുട്ട്*

d. toLowerCase() fuction

ഈ ഫങ്ഷൻ തന്നിരിക്കുന്ന സ്ട്രിംഗിനെ ചെറിയ അക്ഷരമാക്കി തിരികെ നൽകുന്നു. താഴെ നൽകിയിരിക്കുന്ന ഉദാഹരണം ശ്രദ്ധിക്കുക

```
var x, y;
x = "JavaScript";
y = x.toLowerCase();
alert(y);
```

മുകളിലെ കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.16ൽ കാണിച്ചിരിക്കുന്നു. സ്ട്രിംഗിൽ എല്ലാ ക്യാരക്ടറുകളും ഇപ്പോൾ ചെറിയ അക്ഷരത്തിലാണ് അതിനാൽ toLowerCase() അതേ സ്ട്രിംഗ് തിരികെ നൽകുന്നു.

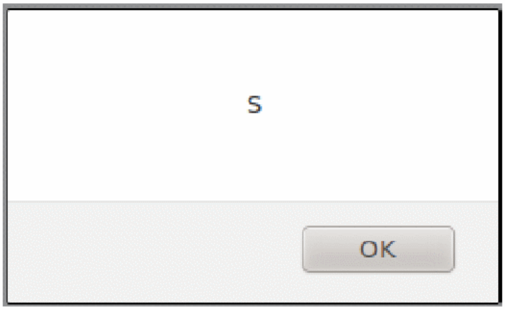


ചിത്രം 6.16: toLowerCase() *ന്റെ ഔട്ട്പുട്ട്*

e. charAt() fuction

ഈ ഫങ്ഷൻ ഒരു പ്രത്യേക സ്ഥാനത്തുള്ള ക്യാരക്ടർ തിരികെ നൽകുന്നു. charAt(0) എന്നത് സ്ട്രിംഗിലെ ആദ്യത്തെ അക്ഷരം തിരികെ നൽകുന്നു. charAt(1) സ്ട്രിംഗിലെ രണ്ടാമത്തെ അക്ഷരം തിരികെ നൽകുന്നു. ഉദാഹരണം നോക്കുക.

```
var x;
x = "JavaScript";
```



ചിത്രം 6.17: charAt() *ന്റെ റിസൾട്ട്*

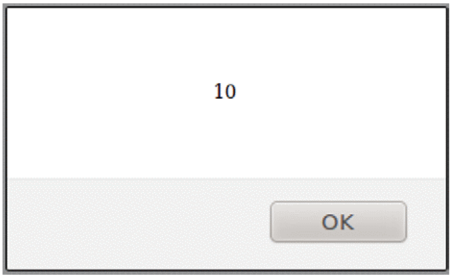
```
y = x.charAt(4);
alert(y);
```

ഇതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.16 ൽ നൽകിയിരിക്കുന്നു. വേരിയബിൾ x ലെ സ്ട്രിംഗിന്റെ അഞ്ചാമത്തെ അക്ഷരം 'S' ആയതിനാൽ ബ്രൗസർ ജാലകത്തിൽ B അക്ഷരം കാണിക്കുന്നു.

f. length പ്രോപ്പർട്ടി

ഫങ്ഷനുകൾ കൂടാതെ, പ്രോഗ്രാമർക്ക് ഉപയോഗപ്രദമായ ചില പ്രോപ്പർട്ടികളും സ്ട്രിംഗ് വേരിയബിളിനോടൊപ്പം ഉപയോഗിക്കാം. length പ്രോപ്പർട്ടി സ്ട്രിംഗിന്റെ നീളം തിരികെ നൽകുന്നു. സ്ട്രിംഗിലെ ക്യാരക്ടറുകളുടെ എണ്ണമാണ് നീളം എന്നതു കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. ഉദാഹരണത്തിന്

```
var x, n;
x = "JavaScript";
n = x.length;
alert(n);
```



ചിത്രം 6.18: length പ്രോപ്പർട്ടിയുടെ റിസൾട്ട്

ഇവിടെ length പ്രോപ്പർട്ടി എങ്ങനെ വിളിക്കുന്നു എന്ന് നോക്കാം. x.length എന്ന രൂപത്തിൽ x എന്ന വേരിയബിളിന്റെ പേര് ചേർത്തതാണ് ഈ പ്രോപ്പർട്ടി വിളിക്കുന്നത്. മുകളിലെ കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.18 ൽ കാണിച്ചിരിക്കുന്നു. ഫങ്ഷനും പ്രോപ്പർട്ടിയും തമ്മിലുള്ള പ്രധാന വ്യത്യാസം, ഫങ്ഷനോടൊപ്പം () ബ്രാക്കറ്റ് ഉപയോഗിക്കുന്നു (ചില സന്ദർഭങ്ങളിൽ ബ്രാക്കറ്റിൽ പരാമീറ്ററുകളും) എന്നാൽ പ്രോപ്പർട്ടിയോടൊപ്പം () ബ്രാക്കറ്റിന്റെ ആവശ്യം ഇല്ല എന്നതാണ്.

നിങ്ങളുടെ പുരോഗതി അറിയാം

താഴെപ്പറയുന്നവയിൽ വേരിയബിൾ y യുടെ വില എഴുതുക.

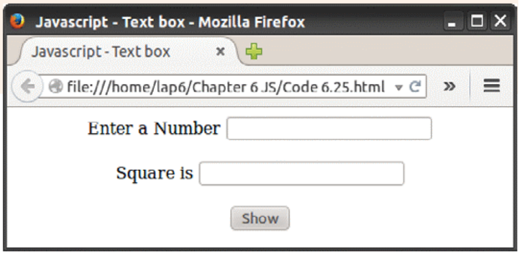
1.
```
x = "welcome";
y = x.length;
```
2.
```
x = "WELCOME";
y = x.toLowerCase();
```
3.
```
x = "Welcome";
y = x.toUpperCase();
```
4.
```
x = "welcome";
y = x.toLowerCase();
```
5.
```
x = "welcome";
y = isNaN(x);
```
6.
```
x = "welcome";
y = charAt(3);
```

6.8 ജാവാസ്ക്രിപ്റ്റിലൂടെ ടെക്സ്റ്റ് ബോക്സിലെ വിലകൾ സ്വീകരിക്കുന്നവിധം (Accessing values in a textbox using JavaScript)

മുൻ അധ്യായത്തിൽ ഒരു വെബ് പേജിൽ ടെക്സ്റ്റ് ബോക്സ്, ചെക്ക്ബോക്സ്, റേഡിയോ ബട്ടൺ, സബ്മിറ്റ് ബട്ടൺ തുടങ്ങിയ വിവിധ കൺട്രോളുകൾ (controls) എങ്ങനെ ഉപയോഗിക്കാം എന്ന് പഠിച്ചു. ഇത്തരം കൺട്രോളുകളെ ജാവാസ്ക്രിപ്റ്റിന്റെ സഹായത്താൽ എങ്ങിനെ ഉപയോഗിക്കാം എന്നാണ് ഇവിടെ ചർച്ചചെയ്യുന്നത്. ഈ അധ്യായത്തിൽ നമ്മൾ ചർച്ച ചെയ്ത ഒരു പ്രോഗ്രാമിലും ഉപയോക്താവിൽ നിന്ന് യാതൊരുവിധ ഇൻപുട്ടും സ്വീകരിച്ചിട്ടില്ല. പ്രോസസ്സിംഗിന് ആവശ്യമായ ഡാറ്റ നേരിട്ട് പ്രോഗ്രാം കോഡിൽ തന്നെ നൽകുകയാണ് ചെയ്തത്. ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് ഒരു വെബ് പേജ് ഘടകങ്ങളെ എങ്ങനെ സ്വീകരിക്കുന്നു എന്ന് നോക്കാം. ഈ പാഠഭാഗത്തിനുശേഷം നമുക്ക് ശരിയായ ഒരു ഇന്ററാക്ടീവ് വെബ് പേജ് രൂപപ്പെടുത്തുവാൻ കഴിയും. അതായത്, ഉപയോക്താവ് ടെക്സ്റ്റ് ബോക്സിൽ വില നൽകാനും അതിന്മേൽ ചില പ്രോസസിംങ് നടത്തി ഫലം മറ്റൊരു ടെക്സ്റ്റ് ബോക്സിൽ പ്രദർശിപ്പിക്കാനും കഴിയും. ഈ കാര്യങ്ങൾ മനസ്സിലാക്കുന്നതിന് താഴെ കൊടുത്തിരിക്കുന്ന HTML കോഡ് ശ്രദ്ധാപൂർവം വായിക്കുക.

ഉദാഹരണം 6.11: ഒരു വെബ് ഫോം (web form) പ്രദർശിപ്പിക്കുന്ന ഒരു വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Text box</TITLE> </HEAD>
<BODY>
   <FORM Name= "frmSquare">
      <CENTER>
         Enter a Number
         <INPUT Type= "text" Name= "txtNum">
         <BR><BR>
         Square is
         <INPUT Type= "text" Name= "txtSqr">
         <BR><BR>
         <INPUT Type= "button" Value= "Show">
      </CENTER>
   </FORM>
</BODY>
</HTML>
```



ചിത്രം 6.19: വെബ് ഫോം പ്രദർശിപ്പിക്കുന്ന ഒരു വെബ് പേജ്

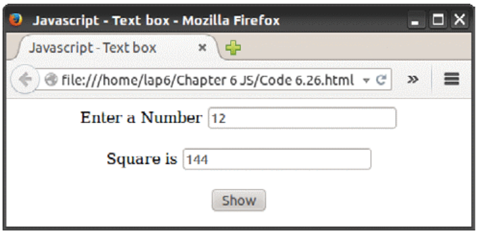
ഈ കോഡിന്റെ പ്രവർത്തന ഫലം ചിത്രം 6.20 ൽ കൊടുത്തിരിക്കുന്നു. ഇവിടെ ഫോമിന് frmSquare എന്നും, ടെക്സ്റ്റ് ബോക്സുകൾക്ക് txtNum, txtSquare എന്നും പേരുകൾ നൽകിയിരിക്കുന്നു. ജാവസ്ക്രി

പ്റ്റിൽ ഇത്തരം ഘടകങ്ങൾക്ക് ഉപയോഗിക്കുന്നതിന് അവയ്ക്ക് പേര് നൽകേണ്ടത് വളരെ പ്രധാനമാണ്. ഒരു വെബ് പേജ് ഘടകത്തിന് നമ്മൾ പേര് നൽകുന്നില്ലെങ്കിൽ, ജാവസ്ക്രിപ്റ്റിന് അവയെ ഉപയോഗിക്കാൻ കഴിയില്ല. Show എന്ന് എഴുതിയിരിക്കുന്ന സബ്മിറ്റ് ബട്ടന് പേര് നൽകിയിട്ടില്ലെന്നത് ശ്രദ്ധിക്കുക. കാരണം, ഈ ബട്ടൺ ജാവസ്ക്രിപ്റ്റിൽ നിന്ന് പരാമർശിക്കാൻ കഴിയില്ല.

ഇനി നമുക്ക് ഈ പ്രോഗ്രാം കോഡിൽ കുറച്ചു മാറ്റം വരുത്തി നോക്കാം.

ഉദാഹരണം 6.12: നൽകുന്ന സംഖ്യയുടെ വർഗം (സ്ക്വയർ) പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു വെബ് പേജ് നിർമാണം.

```
<HTML>
<HEAD> <TITLE>Javascript - Text box</TITLE>
   <SCRIPT Language= "JavaScript">
      function showSquare()
      {
         var num, ans;
         num = document.frmSquare.txtNum.value;
         ans = num * num;
         document.frmSquare.txtSqr.value = ans;
      }
   </SCRIPT>
</HEAD>
<BODY>
   <FORM Name= "frmSquare">
   <CENTER>
      Enter a Number
      <INPUT Type= "text" Name= "txtNum">
      <BR><BR>
      Square is
      <INPUT Type= "text" Name= "txtSqr">
      <BR><BR>
      <INPUT Type= "button" Value= "Show"
                        onClick= "showSquare()">
   </CENTER>
   </FORM>
</BODY>
</HTML>
```



ചിത്രം 6.20: ഒരു സംഖ്യയുടെ വർഗം പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ് പേജ്

മുകളിലുള്ള കോഡ് ശ്രദ്ധാപൂർവം പരിശോധിച്ച് കൂട്ടിച്ചേർത്തിട്ടുള്ളവ ശ്രദ്ധിക്കുക (ഉദാഹരണം 6.17). വെബ് പേജിന്റെ ഹെഡ് ഭാഗത്ത് showSquare() എന്ന ഫങ്ഷൻ നിർവചിച്ചിരിക്കുന്നു. താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിൽ ഈ ഫങ്ഷനെ വിളിക്കുന്നു.

```
<INPUT Type= "button" Value=
"Show" onClick= "showSquare()">
onclick= ""
```

എന്നത് ഇൻപുട്ട് ടാഗിന്റെ ഉള്ളിലാണ് എഴുതിയിട്ടുള്ളത്. ഉപയോക്താവ് ഈ ബട്ടൺ ക്ലിക്കുചെയ്യുമ്പോൾ, showSquare() എന്ന ഫങ്ഷൻ പ്രവർത്തിക്കുന്നു. ബട്ടനിൽ ക്ലിക്കുചെയ്തുകൊണ്ട് ഒരു ഫങ്ഷൻ എങ്ങനെ വിളിക്കാം എന്ന് മനസ്സിലായായല്ലോ.

ഫങ്ഷൻ നിർവചനത്തിലെ താഴെപ്പറയുന്ന വരി ശ്രദ്ധിക്കുക

```
num = document.frmSquare.txtNum.value;
```

document.frmSquare.txtNum.value നോക്കുക. ഇവിടെ ‘document’ എന്നത് വെബ് പേജിന്റെ ബോഡി ഭാഗത്തെ സൂചിപ്പിക്കുന്നു. frmSquare എന്നത് body ഭാഗത്ത് നൽകിയിട്ടുള്ള ഫോമിന്റെ പേരാണ്. ‘txtNum’ എന്നത് 'frmSquare'- ൽ ഉള്ള ടെക്സ്റ്റ് ബോക്സിന്റെ പേരാണ്, value എന്നത് ടെക്സ്റ്റ് ബോക്സിലെ ഉള്ളടക്കത്തെ പരാമർശിക്കുന്നു. അതായത് document.frmSquare.txtNum.value എന്നത് ഡോക്യുമെന്റിലെ frmSquare എന്ന ഫോമിലെ txtNum ന്റെ വിലയാണ്. മുകളിൽ പറഞ്ഞ വരി ടെക്സ്റ്റ് ബോക്സിലെ വില വേരിയബിൾ num-ൽ ലഭ്യമാക്കും.

ഇനി പറയുന്ന വരിയുടെ അർഥം മനസ്സിലാക്കാൻ ശ്രമിക്കുക.

```
document.frmSquare.txtSqr.value = ans;
```

മുകളിലുള്ള വരി രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ans എന്ന വേരിയബിളിന്റെ വില കാണിക്കുന്നു. ബട്ടനിൽ ക്ലിക്കുചെയ്യുമ്പോൾ വെബ്പേജിലെ ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ നൽകിയിരിക്കുന്ന സംഖ്യയുടെ വർഗം രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ കാണിക്കുന്നു. ഉപയോക്താവിന് ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഏത് സംഖ്യയും ടൈപ്പു ചെയ്യാം, അതിന്റെ വർഗ്ഗം കാണുന്നതിന് show ബട്ടൺ ക്ലിക്കുചെയ്യുക. പ്രവർത്തന സമയത്തുള്ള വെബ് പേജിന്റെ സ്ക്രീൻ ഷോട്ട് ചിത്രം 6.20 ൽ കൊടുത്തിരിക്കുന്നു

```
<INPUT Type= "button" Value= "Show" onClick= "showSquare()">
```

എന്നത്

```
<INPUT Type= "button" Value= "Show"
                    onMouseEnter= "showSquare()">
```

എന്നാക്കിയാൽ എന്ത് സംഭവിക്കുമെന്ന് നോക്കാം. ബട്ടനു മുകളിലൂടെ മൗസ് പോയിന്റർ നീക്കുമ്പോൾ ഫങ്ഷൻ പ്രവർത്തിപ്പിക്കുന്നതിനായി വിളിക്കുന്നു. ഫങ്ഷൻ പ്രവർത്തനത്തിനുവേണ്ടി നിങ്ങൾ ബട്ടണിൽ ക്ലിക്ക് ചെയ്യേണ്ടതില്ല. ഒരു വെബ് പേജിലെ ചില

ബട്ടനു മുകളിലൂടെ മൗസ് പോയിന്റർ നീക്കുമ്പോൾ, ബട്ടന്റെ നിറം മാറുന്നത് നിങ്ങൾ ശ്രദ്ധിച്ചിരിക്കാം. ബട്ടന്റെ നിറം മാറുന്നതിനുള്ള ഒരു ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷൻ എഴുതി അത് onMouseEnter എന്ന ഇവന്റിലൂടെ വിളിച്ച് ഇത് സാധ്യമാക്കാം.

<INPUT Type= "button" Value= "Show"onClick= "showSquare()">

onMouseEnter, onClick, onMouseEnter, onMouseLeave, onKeyDown, onKeyUp തുടങ്ങിയവ സാധാരണയായി ഉപയോഗിക്കുന്ന ചില ഇവന്റുകളാണ്.

സാധാരണ ഉപയോഗിക്കുന്ന ചില ജാവ സ്ക്രിപ്റ്റ് ഇവന്റുകളും അവയുടെ വിവരണങ്ങളും പട്ടിക 6.6 ൽ ചേർത്തിരിക്കുന്നു.

ഇവന്റ്	വിവരണം
onClick	ഉപയോക്താവ് ഒരു ഒബ്ജക്റ്റിൽ ക്ലിക്കുചെയ്യുമ്പോൾ സംഭവിക്കുന്നു
onMouseEnter	മൗസ് പോയിന്റർ ഒരു ഒബ്ജക്റ്റിലേക്ക് നീങ്ങുമ്പോൾ സംഭവിക്കുന്നു
onMouseLeave	മൗസ് പോയിന്റർ ഒരു ഒബ്ജക്റ്റിൽ നിന്നും മാറുകയാണെങ്കിൽ സംഭവിക്കുന്നു
onKeyDown	ഉപയോക്താവ് കീബോർഡിലെ ഒരു കീ അമർത്തുമ്പോൾ സംഭവിക്കുന്നു
onKeyUp	ഉപയോക്താവ് കീബോർഡിലെ ഒരു കീയിൽ നിന്ന് വിടുമ്പോൾ സംഭവിക്കുന്നു

പട്ടിക 6.6: സാധാരണ ഉപയോഗിക്കുന്ന ജാവാസ്ക്രിപ്റ്റ് ഇവന്റുകൾ

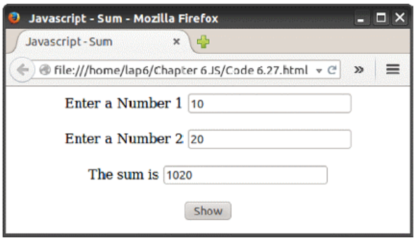
ഒരു വെബ്പേജിൽ രണ്ട് വ്യത്യസ്ത ടെക്സ്റ്റ് ബോക്സുകളിൽ ഉപയോക്താവിന് രണ്ട് സംഖ്യ നൽകാനും, ഒരു ബട്ടണിൽ അമർത്തുമ്പോൾ ഈ സംഖ്യകളുടെ തുക മൂന്നാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ കാണിക്കുന്നതിനുമുള്ള ഒരു വെബ്പേജ് നിർമ്മിക്കാം. കോഡ്, ഔട്ട്പുട്ട് എന്നിവ ചിത്രം 6.21 ൽ കൊടുത്തിരിക്കുന്നു.

ഉദാഹരണം 6.13: രണ്ട് സംഖ്യകളുടെ തുക കാണിക്കുന്ന വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Sum</TITLE>
   <SCRIPT Language= "JavaScript">
      function showSum()
      {
         var num1, num2, ans;
         num1 = document.frmSum.txtNum1.value;
         num2 = document.frmSum.txtNum2.value;
         ans = num1 + num2;
         document.frmSum.txtSum.value = ans;
      }
   </SCRIPT>
```

```
</HEAD>
<BODY>
<FORM Name= "frmSum">
    <CENTER>
    Enter a Number 1
    <INPUT Type= "text" Name= "txtNum1">
    <BR><BR>
    Enter a Number 2
    <INPUT Type= "text" Name= "txtNum2">
    <BR><BR>
    The sum is
    <INPUT Type= "text" Name= "txtSum">
    <BR><BR>
    <INPUT Type= "button" Value= "Show" onClick= "showSum()">
    </CENTER>
</FORM>
</BODY>
</HTML>
```

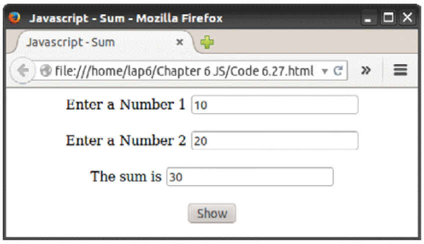
ഈ പ്രോഗ്രാമിന് ഇൻപുട്ടായി 10, 20 എന്നീ സംഖ്യകൾ നൽകിയപ്പോൾ റിസൾട്ടായി 1020 എന്നാണ് കിട്ടിയത് (ചിത്രം 6.21 ൽ കാണിച്ചിരിക്കുന്നു). സ്ട്രിങ്ങുകൾ കൂട്ടിച്ചേർക്കുന്നതിന് + ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നു എന്നതിനാലാണിത്. നമ്മൾ രണ്ട് സ്ട്രിങ്ങുകൾ "10"+"20" ചേർക്കുമ്പോൾ ഉത്തരം '1020' ആണല്ലോ. ഒരു, ടെക്സ്റ്റ് ബോക്സിന്റെ തനതുവില എല്ലായ്പ്പോഴും സ്ട്രിങ്ങായിരിക്കും അതിനാൽ ടെക്സ്റ്റ് ബോക്സിലെ ഉള്ളടക്കം ഒരു നമ്പർ ആയിരുന്നാലും ഒരു വേരിയബിളിലേക്ക് നൽകുമ്പോൾ അതിനെ സ്ട്രിംഗ് ആയി മാത്രമേ പരിഗണിക്കൂ. രണ്ട് സംഖ്യകളുടെ തുക കിട്ടുന്നതിന് ഫങ്ഷൻ showSum() താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിൽ പരിഷ്കരിക്കാം.



ചിത്രം 6.21 ഓപ്പറേറ്റർ ഉപയോഗത്തെ ചിത്രീകരിക്കുന്നതിന് വെബ് പേജ്

```
    function showSum()
    {
        var num1, num2, ans;
        num1 = Number(document.frmSum.txtNum1.value);
        num2 = Number(document.frmSum.txtNum2.value);
        ans = num1 + num2;
document.frmSum.txtSum.value = ans;
    }
```

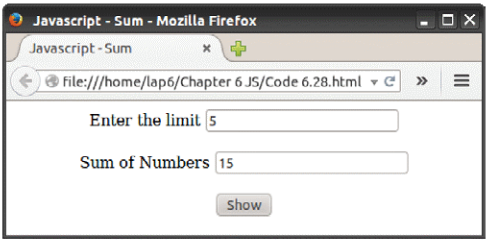
ഇവിടെ Number() എന്ന ഫങ്ഷൻ, ഡാറ്റയെ സംഖ്യയായി മാറ്റി ആ സംഖ്യ num1 എന്ന വേരിയബിളിലേക്ക് നൽകുന്നു. ഇവിടെ num1, num2 എന്നിവയെ നമ്പർ തരം ഡാറ്റയായി കണക്കാക്കുകയും അതുവഴി ശരിയായ തുക കിട്ടുകയും ചെയ്യും. ഉദാഹരണം 6.13 ലെ showSum() എന്ന ഫങ്ഷനിൽ മാറ്റം വരുത്തിയാൽ ചിത്രം 6.22 ൽ കാണിച്ചിരിക്കുന്ന വെബ് പേജ് ഔട്ട്പുട്ടായി ലഭിക്കും. തന്നിരിക്കുന്ന പരിധി വരെയുള്ള സംഖ്യകളുടെ തുക പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു വെബ് പേജ് നിർമിക്കാം. ഇതിൽ പരിധി ഉപയോക്താവിന് ടെക്സ്റ്റ് ബോക്സിലൂടെ നൽകാൻ കഴിയണം.



ചിത്രം 6.22: രണ്ട് സംഖ്യകളുടെ തുക കാണിക്കുന്ന വെബ് പേജ്

ഉദാഹരണം 6.14: തന്നിരിക്കുന്ന പരിധിവരെയുള്ള സംഖ്യകളുടെ തുക കാണിക്കുന്നതിനുള്ള വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Sum</TITLE>
   <SCRIPT Language= "JavaScript">
      function sumLimit()
      {
         var sum = 0, i, limit;
         limit = Number(document.frmSum.txtLimit.value);
         for(i = 1; i <= limit; i++)
            sum += i;
         document.frmSum.txtSum.value = sum;
      }
   </SCRIPT>
</HEAD>
<BODY>
   <FORM Name= "frmSum">
      <CENTER>
         Enter the limit
         <INPUT Type= "text" Name= "txtLimit">
         <BR><BR>
         Sum of Numbers
         <INPUT Type= "text" Name= "txtSum">
         <BR><BR>
         <INPUT Type= "button" Value= "Show"
                                 onClick= "sumLimit()">
      </CENTER>
   </FORM>
</BODY>
</HTML>
```



ചിത്രം 6.23: നൽകിയ പരിധിവരെയുള്ള സംഖ്യകളുടെ തുക കാണിക്കുന്ന വെബ് പേജ്

മുകളിലെ കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.23 ൽ നൽകിയിരിക്കുന്നു. ഈ വെബ് പേജിൽ ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഉപയോക്താവ് പരിധി നൽകണം. അതിനു ശേഷം **show** ബട്ടണിൽ ക്ലിക്കുചെയ്യുമ്പോൾ നൽകിയ പരിധി വരെയുള്ള സംഖ്യകളുടെ തുക രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ കാണിക്കുന്നു. ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഉപയോക്താവ് പരിധി നൽകാതെ **show** ബട്ടൺ ക്ലിക്കുചെയ്താൽ അത് ഒരു സന്ദേശവും പ്രദർശിപ്പിക്കുന്നില്ല. പരിധി ഇല്ലെങ്കിൽ തുക പൂജ്യമായിരിക്കും കാരണം ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സ് ശൂന്യമാകുമ്പോൾ `Number()` ഫങ്ഷൻ ശൂന്യ വിലയെ പൂജ്യമായി മാറ്റുന്നു. അതായത് `document.frmSum.txtLimit.value` ന്റെ വില പൂജ്യമായിരിക്കും. അങ്ങനെ ലൂപ്പ് പ്രവർത്തിക്കാതിരിക്കുകയും `sum` എന്ന വേരിയബിളിന്റെ ആദ്യ വിലയായ പൂജ്യം രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു. ജാവാസ്ക്രിപ്റ്റ് ഒരിക്കലും ഒരു തെറ്റ് ഉണ്ടെന്ന് കാണിച്ച് തരില്ല. സ്ക്രിപ്റ്റ് എഞ്ചിന് എന്തെങ്കിലും നിർദേശം പ്രാവർത്തികമാക്കാൻ കഴിയാത്തപ്പോൾ, അത് ഫങ്ഷനിലെ ആ വരിയും ബാക്കി ഭാഗങ്ങളും അവഗണിക്കും.

താഴെ കൊടുത്തിരിക്കുന്ന `sumLimit()` ഫങ്ഷൻ പരിശോധിക്കാം.

```
function sumLimit()
{
    var sum = 0, i, limit;
    if (document.frmSum.txtLimit.value == "")
    {
        alert("Please enter the limit!");
        return;
    }
    limit = Number(document.frmSum.txtLimit.value);
    for(i=1; i<=limit; i++)
        sum += i;
    document.frmSum.txtSum.value = sum;
}
```

പരിധി നൽകാതെ നിങ്ങൾ show ബട്ടണിൽ ക്ലിക്കുചെയ്താൽ, ഈ കോഡ് പരിധി നൽകാൻ ഓർമ്മപ്പെടുത്തുന്ന ഒരു സന്ദേശം കാണിക്കും. ബാക്കി നിർദേശങ്ങൾ ഒഴിവാക്കിക്കൊണ്ട് ഫങ്ഷനിൽ നിന്ന് പുറത്തുകടക്കുവാൻ `return` പ്രസ്താവന ഉപയോഗിക്കുന്നു. ഇത് C++ ൽ ഉപയോഗിക്കുന്ന റിട്ടേൺ പ്രസ്താവനയ്ക്ക് സമാനമാണ്. ഇനി നമുക്ക് ടെക്സ്റ്റ് ബോക്സിൽ നൽകിയ പരിധി അക്ഷരമാണോ സംഖ്യയാണോ എന്ന് പരിശോധിക്കുന്നതിനായി `isNaN()` എന്ന ഫങ്ഷൻ ഉപയോഗിക്കാം. ഉപയോക്താവ് ശരിയായ ഡാറ്റ നൽകിയിട്ടുണ്ടോ എന്ന് പരിശോധിക്കുന്നതിന് ഇത് സഹായിക്കുന്നു.

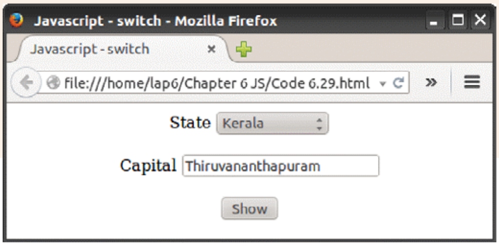
```
function sumLimit()
{
    var sum = 0, i, limit;
    if (document.frmSum.txtLimit.value == "")
    {
        alert("Please enter the limit!");
        return;
    }
    if (isNaN(document.frmSum.txtLimit.value))
    {
        alert("Please enter a number as the limit!");
        return;
    }
    limit = Number(document.frmSum.txtLimit.value);
    for(i = 1; i <= limit; i++)
        sum += i;
    document.frmSum.txtSum.value = sum;
}
```

ക്ലൈന്റ് ഭാഗത്തെ ഡാറ്റയുടെ സാധുത വിലയിരുത്തുന്നതിനായി ജാവ സ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്നു. ആവശ്യമുള്ള എല്ലാ ഡാറ്റയും നൽകിയിട്ടുണ്ടോ, നൽകിയ ഡാറ്റ ശരിയായ ഘടനയിലാണോ തുടങ്ങിയവ പരിശോധിക്കാനായി ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കാം. ഏതെങ്കിലും പോരായ്മ ഉണ്ടെങ്കിൽ അത് ഓർമ്മിപ്പിക്കുന്ന ഒരു സന്ദേശം കാണിക്കുവാൻ കഴിയും. താഴെക്കൊടുത്തിരിക്കുന്ന ഉദാഹരണത്തിൽ ജാവാസ്ക്രിപ്റ്റിലെ ഒരു ഡ്രോപ്പ് ഡൗൺ ലിസ്റ്റ് ഉപയോഗിക്കുന്നു ഈ ഡ്രോപ്പ് ഡൌൺ ലിസ്റ്റ് ഉപയോഗിച്ച് ഒരു സംസ്ഥാനം തെരഞ്ഞെടുക്കാനും show ബട്ടണിൽ ക്ലിക്കുചെയ്യുമ്പോൾ, തെരഞ്ഞെടുത്ത സംസ്ഥാനത്തിന്റെ തലസ്ഥാനം ഒരു ടെക്സ്റ്റ് ബോക്സിൽ പ്രദർശിപ്പിക്കാനുമുള്ള ഒരു വെബ്പേജ് നൽകിയിരിക്കുന്നു. ഇതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.25 ൽ കൊടുത്തിരിക്കുന്നു.

ഉദാഹരണം 6.15: ഒരു സംസ്ഥാനത്തിന്റെ തലസ്ഥാനം പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ്പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - switch</TITLE>
<SCRIPT Language= "JavaScript">
function capital()
{
    var n, answer;
    n = document.frmCapital.cboState.selectedIndex;
    switch (n)
    {
```

```
    case 0:
        answer = "Thiruvananthapuram";
        break;
    case 1:
        answer = "Bengaluru";
        break;
    case 2:
        answer = "Chennai";
        break;
    case 3:
        answer = "Mumbai";
        break;
    }
    document.frmCapital.txtCapital.value = answer;
}
</SCRIPT>
</HEAD>
<BODY>
    <FORM Name= "frmCapital">
    <CENTER> State
        <SELECT Size= 1 Name= "cboState">
            <OPTION>Kerala</OPTION>
            <OPTION>Karnataka</OPTION>
            <OPTION>Tamilnadu</OPTION>
            <OPTION>Maharashtra</OPTION>
        </SELECT>
        <BR><BR>
        Capital
        <INPUT Type= "text" Name= "txtCapital">
        <BR><BR>
        <INPUT Type= "button" Value= "Show" onClick= "capital()">
    </CENTER>
    </FORM>
</BODY>
</HTML>
```



ചിത്രം 6.24: സംസ്ഥാനത്തിന്റെ തലസ്ഥാനം പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ് പേജ്

ഈ പ്രോഗ്രാമിലെ താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന പരിഗണിക്കുക.

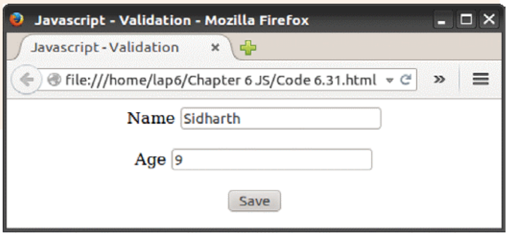
```
document.frmCapital.cboState.selectedIndex;
```

ഇവിടെ 'cboState' എന്നത് ഡ്രോപ്ഡൌൺ ലിസ്റ്റിന്റെ പേരാണ്. 'selectedIndex' എന്നത് ഡ്രോപ് ഡൌൺ ലിസ്റ്റിന്റെ സൂചിക കാണിക്കുന്നു. ഒന്നാമത്തെ ഇനം തിരഞ്ഞെടുത്താൽ സൂചിക 0-ഉം രണ്ടാമത്തെ ഇനം തിരഞ്ഞെടുത്താൽ, സൂചിക 1 ഉം ആയി കണക്കാക്കുന്നു. മുകളിലെ വരി തിരഞ്ഞെടുത്ത വേരിയബിളിന്റെ ഇൻഡക്സ് വേരിയബിൾ n ൽ നൽകുന്നു. വിദ്യാർഥിയുടെ പേരും പ്രായവും നൽകാൻ ഉപയോക്താവിനെ അനുവദിക്കുന്ന വെബ് പേജാണ് താഴെ കൊടുത്തിരിക്കുന്നത്. പേരിൽ കുറഞ്ഞത് 5 അക്ഷരങ്ങൾ ഉണ്ടായിരിക്കണം. പ്രായം 15 മുതൽ 20 വരെ ഉള്ള ഒരു സംഖ്യയായിരിക്കണം.

ഉദാഹരണം 6.16: പേരിന്റെയും പ്രായത്തിന്റെയും സാധുത പരിശോധിക്കുന്ന വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Validation</TITLE>
<SCRIPT Language= "JavaScript">
function checkData()
{
    var T_name, T_age, N_age;
    T_name = document.frmValid.txtName.value;
    if (T_name == "")
    {
        alert("Please enter name!");
        return;
    }
    if (T_name.length < 5)
    {
        alert("Name must contain at least 5 characters!");
        return;
    }
    T_age = document.frmValid.txtAge.value;
    if (T_age == "")
    {
        alert("Please enter age!");
        return;
    }
    if (isNaN(T_age))
    {
        alert("Please enter a number as the age!");
        return;
    }
    N_age = Number(T_age);
    if (N_age < 15 || N_age > 20)
```

```
    {
        alert("The age must be between 15 and 20!");
        return;
    }
}
</SCRIPT>
</HEAD>
<BODY>
    <FORM Name= "frmValid">
        <CENTER>Name
            <INPUT Type= "text" Name= "txtName">
            <BR><BR>
            Age
            <INPUT Type= "text" Name= "txtAge">
            <BR><BR>
            <INPUT Type= "button" Value= "Save"
                                    onClick= "checkData()">
        </CENTER>
    </FORM>
</BODY>
</HTML>
```



ചിത്രം 6.25: ലൂപ്പിന്റെ ഉപയോഗം വിശദീകരിക്കുന്ന വെബ് പേജ്

HTML കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.25 ൽ കൊടുത്തിരിക്കുന്നു. ഡാറ്റയുടെ എല്ലാ സാധുതകളും ഇത് പരിശോധിക്കുന്നു. ആദ്യം, പേരിന്റെ ഫീൽഡിൽ ഒരു വില ഉണ്ടോ എന്ന് പരിശോധിക്കുന്നു. പിന്നീട് പേരിന്റെ ദൈർഘ്യം കുറഞ്ഞത് 5 അക്ഷരങ്ങൾ ഉണ്ടോ എന്ന് പരിശോധിക്കുന്നു. ശേഷം വയസ്സ് നൽകിയിട്ടുണ്ടോ എന്ന് പരിശോധിക്കുന്നു. പിന്നീട് നൽകിയ ഡാറ്റ സംഖ്യ ആണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നു. അന്തിമമായി പ്രായപരിധി 15 മുതൽ 20 വരെ ആണോ എന്ന് പരിശോധിക്കുന്നു.

6.9 സ്ക്രിപ്റ്റുകൾ വെബ്പേജിൽ ചേർക്കുന്നതിനുള്ള വഴികൾ (Ways to add scripts to a webpage)

പല രീതിയിൽ HTML കോഡുകൾക്കുള്ളിൽ സ്ക്രിപ്റ്റുകൾ ഉൾപ്പെടുത്താവുന്നതാണ്. മുൻ ഉദാഹരണങ്ങളിൽ നമ്മൾ ജാവാസ്ക്രിപ്റ്റ് വെബ് പേജിന്റെ ഹെഡ് ഭാഗത്താണ് ഉൾപ്പെടുത്തിയിരുന്നത്. ഇതിന് പുറമേ, സ്ക്രിപ്റ്റുകൾ <BODY> ടാഗിലോ അല്ലെങ്കിൽ ഒരു ബാഹ്യ ഫയലിലോ നൽകാവുന്നതാണ്.

ഇനി വെബ്പേജുകളിൽ സ്ക്രിപ്റ്റുകൾ കൂട്ടിച്ചേർക്കുന്നതിനുള്ള വ്യത്യസ്ത രീതികൾ പരിചയപ്പെടാം.

6.9.1 <BODY> ടാഗിന് ഉള്ളിൽ (Inside <BODY>)

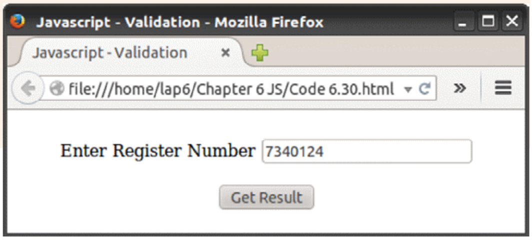
ഈ അധ്യായത്തിന്റെ തുടക്കത്തിൽ <BODY> ടാഗിന് ഉള്ളിൽ സ്ക്രിപ്റ്റുകൾ സ്ഥാപിക്കുന്നത് നാം ചർച്ചചെയ്തു. വെബ്പേജിന്റെ ഉള്ളടക്കം ബ്രൗസറിൽ പ്രദർശിപ്പിക്കുമ്പോൾ സ്ക്രിപ്റ്റുകളും പ്രവർത്തിക്കും. വെബ് പേജ് ഡോക്യുമെന്റിന്റെ ആരംഭം മുതൽക്കാണ് പ്രദർശിപ്പിക്കുക. ഒരു സ്ക്രിപ്റ്റ് കോഡ് കാണുമ്പോൾ അത് പ്രവർത്തിക്കുകയും തുടർന്ന് വെബ് പേജിന്റെ ബാക്കി ഭാഗം പ്രവർത്തിക്കുകയും ചെയ്യുന്നു.

ഒരു ഉദാഹരണം ഉപയോഗിച്ച് ഈ രീതി മനസ്സിലാക്കാം. ഒരു വിദ്യാർഥിയുടെ റിസൾട്ട് ലഭിക്കുന്നതിനുള്ള ഒരു വെബ് പേജാണ് താഴെ കൊടുത്തിരിക്കുന്നത്. ടെക്സ്റ്റ് ബോക്സിൽ ഉപയോക്താവ് ഒരു രജിസ്റ്റർ നമ്പർ നൽകണം. **Get Result** ബട്ടനിൽ ക്ലിക്കു ചെയ്യുമ്പോൾ രജിസ്റ്റർ നമ്പർ ബോക്സിൽ ഏതെങ്കിലും ഡാറ്റ ഉണ്ടോ എന്ന് പരിശോധിക്കാനുള്ള ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷൻ പ്രവർത്തിക്കണം. ടെക്സ്റ്റ് ബോക്സിൽ വില ഉണ്ടെങ്കിൽ അത് ഏഴ് അക്കമുള്ള ഒരു സംഖ്യ തന്നെ ആയിരിക്കണം. ഈ കാര്യങ്ങൾ ഫങ്ഷൻ പരിശോധിക്കണം. വെബ് പേജിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.26 ൽ നൽകിയിരിക്കുന്നു.

ഉദാഹരണം 6.17: സാധുതാപരിശോധനയ്ക്കു ശേഷം രജിസ്റ്റർ നമ്പർ സ്വീകരിക്കുന്ന വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Validation</TITLE> </HEAD>
<BODY>
   <FORM Name= "frmValid">
      <SCRIPT Language= "JavaScript">
         function checkData()
         {
            var rno;
            rno = document.frmValid.txtRegno.value;
            if (rno == "")
            {
               alert("Please enter Register No.");
               return;
            }
            if (isNaN(rno))
            {
               alert("Invalid Register No.");
               return;
            }
            if (rno.length < 7)
```

```
            {
                alert("The Register No. must have 7 digits");
                return;
            }
        }
    </SCRIPT>
    <CENTER>
        <BR>Enter Register Number
        <INPUT Type= "text" Name= "txtRegno">
        <BR><BR>
        <INPUT Type= "button" Value= "Get Result"
                                    onClick= "checkData()">
    </CENTER>
  </FORM>
</BODY>
</HTML>
```



ചിത്രം 6.26: രജിസ്റ്റർ നമ്പർ സ്വീകരിക്കാനുള്ള ഫോം പ്രദർശിപ്പിക്കുന്ന വെബ് പേജ്

സ്ക്രിപ്റ്റ് <BODY> ടാഗിനുള്ളിൽ അവസാനഭാഗത്തും ആകാം. <BODY> ടാഗിൽ അല്ലെങ്കിൽ <HEAD> ടാഗിൽ സ്ക്രിപ്റ്റു I Ä F gpX rb rxps- ¦ nÂ A h HTML കോഡിനൊപ്പം ബ്രൗസറിലെത്തുകയും ചെയ്യും. വെബ്പേജ് പ്രദർശിപ്പിക്കുന്നത് വൈകുവാൻ ഇത് കാരണമാകുന്നു. സ്ക്രിപ്റ്റുകൾ </BODY> ടാഗിന് മുമ്പായി കൊടുക്കുകയാണെങ്കിൽ വെബ്പേജിലെ ഉള്ളടക്കങ്ങൾ അക്ഷരങ്ങൾ, ചിത്രങ്ങൾ പോലുള്ളവ സ്ക്രീനിൽ വേഗത്തിൽ ദൃശ്യമാകും. എന്നാൽ വെബ് പേജ് ലഭ്യമാക്കുന്നതിനിടയിൽ സ്ക്രിപ്റ്റുകൾ പ്രവർത്തിക്കണമെന്നില്ല.

6.9.2 <HEAD> ടാഗിന് ഉള്ളിൽ (Inside <HEAD>)

സാധാരണ രീതിയിൽ വെബ് പേജിന്റെ ഹെഡ് ഭാഗത്താണ് സ്ക്രിപ്റ്റുകൾ ഉൾപ്പെടുത്തുന്നത്. നാം ചർച്ച ചെയ്ത എല്ലാ ഉദാഹരണങ്ങളിലും അങ്ങനെ തന്നെയാണ് ചെയ്തിരിക്കുന്നത്. ഇതിന് പ്രധാന കാരണം മിക്ക HTML പേജുകളുടെയും ബോഡി ഭാഗത്തിനുള്ളിൽ ഉള്ളടക്കത്തെ സൂചിപ്പിക്കുന്ന ഒരു വലിയ അളവ് ടെക്സ്റ്റ് ദൃശ്യമാക്കേണ്ടതായിട്ടുണ്ട്. ഫങ്ഷൻ നിർവചനം കൂടി ഇവിടെ ചേർക്കുകയാണെങ്കിൽ വെബ്പേജിൽ മാറ്റം വരുത്തുമ്പോൾ ഇത് രൂപകല്പനചെയ്യുന്നയാൾക്ക് ആശയക്കുഴപ്പം സൃഷ്ടിക്കും. ബോഡി ഭാഗത്തിന് മുൻപായി ഒരു വെബ് പേജിന്റെ ഹെഡ് ഭാഗം ലോഡ് ചെയ്യുന്നു എന്നത് ഇതിന്റെ നേട്ടമാണ്. അതിനാൽ, ബോഡി ഭാഗത്ത് വിളിക്കപ്പെടുന്ന ഫങ്ഷനുകളുടെ നിർവചനം നേരത്തെ തന്നെ മെമ്മറിയിൽ ലഭ്യമാക്കിയിരിക്കും. ഉദാഹരണം 6.17 ൽ <SCRIPT> നും </SCRIPT> നും ഇടയിലുള്ള കോഡ് HTML കോഡിന്റെ ഹെഡ് ഭാഗത്തേക്ക് മാറ്റി ഉപയോഗിക്കാവുന്നതാണ്.

6.9.3 ബാഹ്യ ജാവാസ്ക്രിപ്റ്റ് ഫയൽ (External JavaSript file)

HTML ഡോക്യുമെന്റിലെ എല്ലാ സ്ക്രിപ്റ്റുകളും ഒരു ബാഹ്യ ഫയലിലേക്ക് ശേഖരിക്കുകയും ഡോക്യുമെന്റിൽ നിന്ന് ആ ഫയലിലേക്ക് ഒരു ലിങ്ക് സ്ഥാപിക്കുകയും ചെയ്യാം. ഈ ഫയൽ '.js' എന്ന എക്സ്റ്റൻഷനോടുകൂടി സൂക്ഷിക്കുക. ബാഹ്യ ഫയലുകളിൽ ജാവാസ്ക്രിപ്റ്റ് സ്ഥാപിക്കുന്നത് കൊണ്ട് ചില ഗുണങ്ങളുണ്ട്. ഒരേ സ്ക്രിപ്റ്റ് ഒന്നിലധികം HTML പേജുകളിലോ ഒരു മുഴുവൻ വെബ് സൈറ്റിലോ ആവർത്തിക്കപ്പെടുമ്പോൾ ഇത് ഉപയോഗപ്രദമാണ്. ഇത് HTML ഉം ജാവാസ്ക്രിപ്റ്റും വേർതിരിച്ച് കാണാൻ സഹായിക്കുന്നു. ഇത് HTML കോഡിനെയും ജാവാസ്ക്രിപ്റ്റിനെയും വേർതിരിക്കുവാനും ഇവ രണ്ടിനെയും എളുപ്പത്തിൽ വായിക്കാനും നിലനിർത്താനും സഹായിക്കുകയും ചെയ്യുന്നു. ജാവാസ്ക്രിപ്റ്റ് ബാഹ്യ ഫയലുകളിൽ സംഭരിക്കുന്നത് പേജ് വേഗത്തിൽ ലഭ്യമാക്കാൻ സഹായിക്കും.

മുകളിൽ പറഞ്ഞിരിക്കുന്ന ഉദാഹരണം 6.17 ൽ ജാവാസ്ക്രിപ്റ്റ് കോഡ് "check.js" എന്ന പേരിൽ ഒരു പ്രത്യേക ഫയൽ ആയി സൂക്ഷിക്കുന്നു. ഈ ഫയലിന്റെ ഉള്ളടക്കം താഴെ കാണിച്ച രീതിയിലാണ്.

```
function checkData()
{
    var rno;
    rno = document.frmValid.txtRegno.value;
    if (rno == "")
    {
        alert("Please enter Register No.");
        return;
    }
    if (isNaN(rno))
    {
        alert("Invalid Register No.");
        return;
    }
    if (rno.length < 7)
    {
        alert("The Register No. must have 7 digits");
        return;
    }
}
```

ഈ ഫയലിൽ ജാവാസ്ക്രിപ്റ്റ് കോഡ് മാത്രമേ ഉള്ളൂ എന്നും <SCRIPT> ടാഗ് ഉപയോഗിച്ചിട്ടില്ല എന്നും ശ്രദ്ധിക്കുക. HTML ഫയലിനുള്ളിൽ മാത്രമാണ് <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നത്. <SCRIPT> ടാഗ് ഉപയോഗിച്ച് ജാവാസ്ക്രിപ്റ്റ് ഫയലിനെ HTML ഫയലുമായി ലിങ്ക് ചെയ്യാവുന്നതാണ്. Type ആട്രിബ്യൂട്ട് സൂചിപ്പിക്കുന്നത് ലിങ്കു

ചെയ്തിരിക്കുന്ന ഫയൽ ഒരു ജാവാസ്ക്രിപ്റ്റ് ഫയൽ ആണെന്നതാണ്. Src ആട്രിബ്യൂട്ട് ബാഹ്യ ജാവാസ്ക്രിപ്റ്റ് ഫയലിന്റെ സ്ഥാനവും ഫയലിന്റെ പേരും വ്യക്തമാക്കുന്നു. പരിഷ്കരിച്ച HTML കോഡ് താഴെ കൊടുത്തിരിക്കുന്നു.

```
<HTML>
<HEAD><TITLE>Javascript - Validation</TITLE>
   <SCRIPT Type= "text/JavaScript" Src= "checkdata.js">
   </SCRIPT>
</HEAD>
<BODY>
   <FORM Name= "frmValid">
      <CENTER>
         <BR>Enter Register Number
         <INPUT Type= "text" Name= "txtRegno">
         <BR><BR>
         <INPUT Type= "button" Value= "Get Result"
                             onClick= "checkData()">
      </CENTER>
   </FORM>
</BODY>
</HTML>
```

<SCRIPT> ടാഗിൽ Src ആട്രിബ്യൂട്ട് ഉണ്ടെങ്കിൽ അതിലെ ഉള്ളടക്കം അവഗണിക്കപ്പെടും. അതായത്, നിങ്ങൾക്ക് ഒറ്റ <SCRIPT> ടാഗുപയോഗിച്ച് ബാഹ്യ ഫയൽ ചേർക്കാനും ഒരു കോഡ് നടപ്പാക്കാനും ഒരുമിച്ച് സാധ്യമല്ല. ഇതിനായി രണ്ട് വ്യത്യസ്ത <SCRIPT> ടാഗുകൾ ആവശ്യമാണ്. ഇവയിൽ ഒന്ന് Src ഉപയോഗിച്ച് ഒരു ബാഹ്യ ഫയൽ ചേർക്കാനും മറ്റൊന്ന് കോഡ് ചേർക്കാനും ഉപയോഗിക്കാം

നിങ്ങളുടെ പുരോഗതി അറിയാം

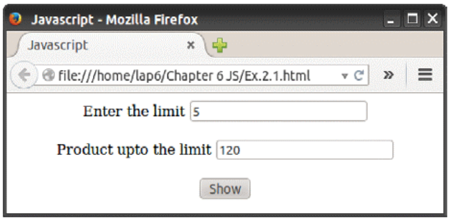
1. ജാവാസ്ക്രിപ്റ്റിൽ onMouseEnter ഇവന്റ് എപ്പോഴാണ് പ്രവർത്തിക്കുക?
2. ഡ്രോപ്ഡൗൺ ലിസ്റ്റിൽ <SELECT> നിന്നും തിരഞ്ഞെടുത്ത ഇനത്തിന്റെ സൂചിക ലഭിക്കാനുള്ള പ്രോപ്പർട്ടി ________________ ആണ്.
3. ജാവാസ്ക്രിപ്റ്റിലെ onKeyDown, onKeyUp എന്നിവ താരതമ്യം ചെയ്യുക.
4. Number() ഫങ്ഷൻ ഉപയോഗിക്കുന്നത് എന്തിനാണ്?
5. ജാവാസ്ക്രിപ്റ്റ് കോഡ് </BODY> ടാഗിന് മുൻപായി ഉപയോഗിക്കുന്നതിന്റെ ഗുണം എന്ത്?

നമുക്കു സംഗ്രഹിക്കാം

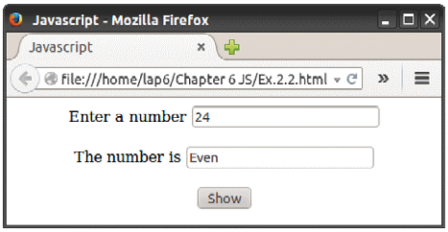
ഡാറ്റയുടെ സാധൂകരണത്തിനായി ഉപയോഗിക്കുന്ന ഒരു ക്ലൈന്റ് ഭാഗം സ്ക്രിപ്റ്റിംഗ് ഭാഷയായിട്ടാണ് ജാവാസ്ക്രിപ്റ്റിനെ ഈ അധ്യായത്തിൽ അവതരിപ്പിക്കുന്നത്. ജാവാ സ്ക്രിപ്റ്റ് കോഡ് HTML ൽ ഉൾപ്പെടുത്തുന്നതിന് ഉപയോഗിക്കുന്ന ടാഗുകളും ജാവാ സ്ക്രിപ്റ്റിലെ പ്രധാന ഫങ്ഷനുകളും ഇവിടെ വിശദീകരിച്ചിരിക്കുന്നു. ജാവാസ്ക്രിപ്റ്റിലെ ഡാറ്റാതരങ്ങളും വേരിയബിളിന്റെ ഉപയോഗവും വിശദമായി ചർച്ചചെയ്യുന്നു. ഓപ്പറേറ്റ റുകളുടെയും കൺട്രോൾ സ്ട്രക്ചറുകളുടെയും ഉപയോഗം C++ലേതിന് സമാനമാണ്. വിവിധ അന്തർ നിർമ്മിത ഫങ്ഷനുകളും ജാവാസ്ക്രിപ്റ്റിലെ ഇവന്റുകളും ഉദാഹരണ സഹിതം വിശദീകരിച്ചിരിക്കുന്നു. HTML പേജിൽ ജാവാസ്ക്രിപ്റ്റ് കോഡ് ഉൾപ്പെടുത്തുന്ന വ്യത്യസ്ത രീതികൾ വിശദമായി പ്രതിപാദിച്ചിരിക്കുന്നു.

നമുക്കു പരിശീലിക്കാം

1. താഴെ കാണുന്ന സ്ക്രീൻ പ്രദർശിപ്പിക്കുന്നതിന് ഒരു വെബ് പേജ് നിർമിക്കുക. ഉപയോക്താവിന് ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഒരു നമ്പർ നൽകാം. Show ബട്ടൺ ക്ലിക്കുചെയ്യുമ്പോൾ രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ 1 മുതൽ നൽകിയിട്ടുള്ള പരിധി വരെയുള്ള എല്ലാ സംഖ്യകളുടെയും ഗുണനഫലം പ്രദർശിപ്പിക്കണം.



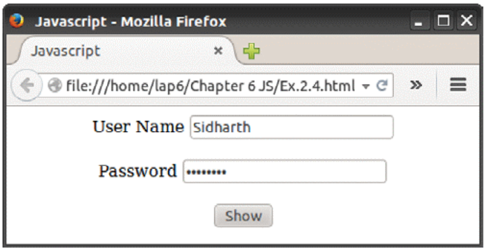
2. താഴെ കാണുന്ന സ്ക്രീൻ പ്രദർശിപ്പിക്കുന്നതിന് ഒരു വെബ് പേജ് നിർമിക്കുക. ഉപയോക്താവിന് ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഒരു നമ്പർ നൽകാം. Show ബട്ടണിൽ ക്ലിക്കുചെയ്യുമ്പോൾ രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഒറ്റ സംഖ്യയോ ഇരട്ട സംഖ്യയോ എന്നതിന് അനുസൃതമായി Odd അല്ലെങ്കിൽ Even എന്ന് പ്രദർശിപ്പിക്കണം.



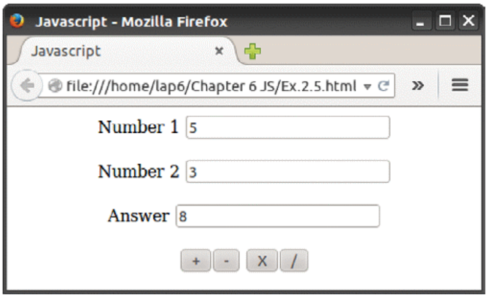
3. താഴെക്കൊടുത്തിരിക്കുന്ന സ്ക്രീൻ പ്രദർശിപ്പിക്കുന്നതിന് ഒരു വെബ് പേജ് തയാറാക്കുക. ഉപയോക്താവിന് ടെക്സ്റ്റ് ബോക്സിൽ ഒരു പ്രായം നൽകാം. ടെക്സ്റ്റ് ബോക്സിൽ ഒരു സംഖ്യക്ക് പകരം ഉപയോക്താവ് അക്ഷരമാണ് നൽകുന്നതെങ്കിൽ, Show ബട്ടണിൽ ക്ലിക്കുചെയ്യുമ്പോൾ 'Invalid Age" എന്ന ഒരു

സന്ദേശം പ്രദർശിപ്പിക്കണം അല്ലെങ്കിൽ 'Correct Data" എന്ന സന്ദേശം പ്രദർശിപ്പിക്കണം.

4. ചിത്രത്തിൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഒരു ലോഗിൻ പേജ് നിർമിക്കുക. പാസ് വേഡ് നൽകുന്നതിനായി പാസ് വേഡ് ടെക്സ്റ്റ് ബോക്സും യൂസർനെയിം നൽകാനായി ഒരു ടെക്സ്റ്റ് ബോക്സും ഉണ്ടായിരിക്കണം. യൂസർനെയിമിൽ കുറഞ്ഞത് 4 ഉം പാസ് വേഡിൽ കുറഞ്ഞത് 6 ഉം ക്യാരക്ടറുകൾ ഉണ്ടായിരിക്കണം. പാസ്‌വേഡിലെ ആദ്യത്തെ രണ്ട് ക്യാരക്ടറുകൾ അക്കങ്ങൾ ആയിരിക്കണം. show ബട്ടണിൽ ക്ലിക്കു ചെയ്യുമ്പോൾ സാധുവായ ഡാറ്റയാണ് ബോക്സുകളിൽ നൽകിയിട്ടുള്ളതെങ്കിൽ 'Correct Data" എന്നും അല്ലെങ്കിൽ 'Wrong Data" എന്നുമുള്ള സന്ദേശം പ്രദർശിപ്പിക്കണം.



5. താഴെ കാണുന്ന ചിത്രത്തിലെ പോലെ ലളിതമായ ഒരു കാൽക്കുലേറ്റർ അടങ്ങിയ ഒരു വെബ് പേജ് നിർമിക്കുക. രണ്ട് സംഖ്യകൾ നൽകാൻ രണ്ടു ടെക്സ്റ്റ് ബോക്സുകൾ ഉണ്ടായിരിക്കണം. സങ്കലനം, വ്യവകലനം, ഗുണനം, ഹരണം എന്നിവയ്ക്കായി 4 ബട്ടണുകൾ ഉണ്ടായിരിക്കണം. ബട്ടനിൽ ക്ലിക്കുചെയ്യുമ്പോൾ മൂന്നാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഉത്തരം കാണിക്കണം.



നമുക്കു വിലയിരുത്താം

1. താഴെപ്പറയുന്നവയിൽ വേരിയബിൾ z എന്ന വേരിയബിളിന്റെ വില എഴുതുക

a.
```
var x, y, z;
x = 5;
y = 3;
z = ++x - y--;
```

b.
```
var x, y, z;
x = "12";
y = 13;
z = x + y;
```

c.
```
var x, y, z;
x = 20;
y = 8;
x %= y;
z = x++;
```

d.
```
var x, y, z;
x = 1;
y = 4;
z = !(x < y);
```

e.
```
var x, y, z;
x = 5;
y = 6 ;
z = (x > y) || (y % 2 == 0);
```

2. താഴെകൊടുത്തിരിക്കുന്നവയുടെ ഔട്ട്പുട്ട് കണ്ടെത്തുക.

a.
```
<HTML>
<BODY>
<SCRIPT Language= "JavaScript">
var i;
for (i = 10; i >= 1; i--)
      document.write(i + "<BR>");
</SCRIPT>
</BODY>
</HTML>
```

b.
```
<HTML>
<BODY>
<SCRIPT Language= "JavaScript">
var i, s = 0;
for (i = 1; i <= 100; i += 2)
      s += i;
document.write("Sum = " + s);
</SCRIPT>
</BODY>
</HTML>
```

c.
```
<HTML>
<BODY>
<SCRIPT Language= "JavaScript">
var n, s = 0;
n = 0;
while (n <= 50)
{
      s = s + n;
      n = n + 5;
}
```

```
document.write("Sum = " + s);
</SCRIPT>
</BODY>
</HTML>
```

d.
```
<HTML>
<BODY>
<SCRIPT Language= "JavaScript">
var n, f = 1;
n = 5;
while ( n > 0)
{
      f = f * n;
      n--;
}
document.write("Product  = " + f);
</SCRIPT>
  </BODY>
  </HTML>
```

3. താഴെ കൊടുത്തിരിക്കുന്ന html കോഡ് ശ്രദ്ധിക്കുക

```
<FORM Name= "frmStud">
<INPUT Type= "text" Name= "studentName">
</FORM>
```

ടെക്സ്റ്റ് ബോക്സിൻറെ വില n എന്ന വേരിയബിളിൽ സംഭരിക്കുന്നതിനായി വിട്ടഭാഗം പൂരിപ്പിക്കുക.

```
var n;
n = ..........................;
```

4. നിങ്ങൾ checkData () എന്ന് പേരിൽ ഒരു ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷൻ എഴുതിയിട്ടുണ്ടെന്ന് കരുതുക. മൗസ് പോയിന്റർ ബട്ടനുമുകളിലൂടെ നീക്കുമ്പോൾ മാത്രമേ ഫങ്ഷൻ പ്രവർത്തിക്കേണ്ടതുള്ളൂ. അതിനുവേണ്ടി താഴെക്കൊടുത്തിരിക്കുന്ന കോഡ് എങ്ങനെ പൂർത്തീകരിക്കാം.

```
<INPUT Type= "button" .......... = "checkData()">
```

5. <SCRIPT> ടാഗും അതിന്റെ ആട്രിബ്യൂട്ടുകളും വിശദമാക്കുക.

6. ജാവാസ്ക്രിപ്റ്റിൽ ഒരു അന്തർനിർമിത ഫങ്ഷൻ നിർവചിക്കുന്നതിനുള്ള വാക്യഘടന എഴുതുക.

7. താഴെക്കൊടുത്തിരിക്കുന്ന വിലകൾ ജാവാസ്ക്രിപ്റ്റിലെ അനുയോജ്യമായ ഡാറ്റാ ഇനങ്ങൾ ഉപയോഗിച്ച് വേർതിരിക്കുക.

   "Welcome", "123", "true", 67.4, .98, false, "hello"

8. ജാവാസ്ക്രിപ്റ്റിൽ undefined ഡാറ്റ ഇനം എന്നതുകൊണ്ട് എന്താണ് അർഥമാക്കുന്നത്?

9. ജാവാസ്ക്രിപ്റ്റിലെ ഓപ്പറേറ്ററുകൾ വിശദീകരിക്കുക.

10. താഴെപ്പറയുന്നവ ചെയ്യുന്നതിനുള്ള ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷനുകളുടെ പേര് എഴുതുക
    a. N എന്ന വേരിയബിളിൽ സംഖ്യ ആണോ എന്ന് പരിശോധിക്കാൻ.
    b. "scert" എന്ന സ്ട്രിംഗ് വലിയ അക്ഷരങ്ങളായി മാറ്റാൻ.
    c. "HTML" എന്ന സ്ട്രിംഗ് ചെറിയ അക്ഷരങ്ങളായി മാറ്റാൻ.
    d. "Welcome to functions" എന്ന സന്ദേശം പ്രദർശിപ്പിക്കാൻ.
    e. "Computer" എന്ന സ്ട്രിംഗിലെ മൂന്നാമത്തെ ക്യാരക്ടർ പ്രദർശിപ്പിക്കാൻ.

11. 'Computer" എന്ന സ്ട്രിങിന്റെ നീളം പ്രദർശിപ്പിക്കുന്നതിനുള്ള ജാവാസ്ക്രിപ്റ്റ് കോഡ് എഴുതുക.

12. ഒരു വെബ് പേജിൽ ഒരു ബട്ടൺ ഉണ്ട്. താഴെപ്പറയുന്ന ഇവന്റുകൾക്കനുസരിച്ച് Message( ) എന്ന ഫങ്ഷൻ പ്രവർത്തിപ്പിക്കാനുള്ള HTML കോഡ് എഴുതുക.
    a. ഉപയോക്താവ് ബട്ടണിൽ മൗസ് ക്ലിക്ക് ചെയ്യുമ്പോൾ.
    b. ഉപയോക്താവ് ബട്ടണിനുമേൽ മൗസ് പോയിന്റർ നീക്കുമ്പോൾ.

13. ഒരു HTML പേജിൽ ഹെഡ് ഭാഗത്ത് ജാവാസ്ക്രിപ്റ്റ് കോഡ് എഴുതുന്നതിന്റെ ഗുണങ്ങൾ ഏവ?

14. ഒരു വിഷയത്തിന്റെ മാർക്ക് നൽകുവാൻ ഒരു ടെക്സ്റ്റ് ബോക്സ് അടങ്ങിയിരിക്കുന്ന ഒരു വെബ് പേജ് രൂപകൽപന ചെയ്യുക.
    a. ഈ വെബ് പേജിനുവേണ്ടിയുള്ള HTML കോഡ് എഴുതുക.
    b. ഈ ടെക്സ്റ്റ് ബോക്സിലെ വിലയുടെ സാധൂകരണത്തിന് ഒരു ബാഹ്യ ജാവാസ്ക്രിപ്റ്റ് ഫയൽ നിർമിച്ച് അത് HTML ഡോക്യുമെന്റുമായി ബന്ധിപ്പിക്കുക. വിലയിരുത്തുന്നതിനുള്ള സൂചകങ്ങൾ
       (i) അത് ശൂന്യമായിരിക്കരുത്
       (ii) അത് ഒരു സംഖ്യയായിരിക്കണം
       (iii) 0-നും 60-നും ഇടയിലായിരിക്കണം മാർക്ക്.
    c. സ്ക്രിപ്റ്റ് ഒരു ബാഹ്യ ഫയൽ ആയി എഴുതുന്നതിന്റെ ഗുണഫലങ്ങൾ എഴുതുക.

7 വെബ് ഹോസ്റ്റിങ്

A2P9H8

പ്രധാന പഠനനേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പഠനം പൂർത്തിയാകുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ

- വെബ്സെർവറിന്റെ ഉപയോഗം വെബ്ഹോസ്റ്റിങ് ആശയം എന്നിവ വിവരിക്കുന്നു.
- വ്യത്യസ്ത തരത്തിലുള്ള ഹോസ്റ്റിങ് രീതി മനസിലാക്കുന്നു.
- ഹോസ്റ്റിങ് സ്ഥലം വാങ്ങുന്നതിനുള്ള മാർഗ്ഗം വിശദമാക്കുന്നു.
- ഡൊമൈൻ നെയിം രജിസ്റ്റർ ചെയ്യും എഫ്.റ്റി.പി. ക്ലയന്റ് സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് ഹോസ്റ്റ് ചെയ്യുന്നു.
- സൗജന്യ ഹോസ്റ്റിംഗി ന്റെ സവിശേഷതകൾ മനസ്സിലാക്കുന്നു.
- കണ്ടന്റ് മാനേജ്മെന്റ് സംവിധാനത്തിന്റെ ഉപയോഗം തിരിച്ചറിയുന്നു.
- റെസ്പോൺസീവ് വെബ് രൂപകൽപനയുടെ ആവശ്യകത തിരിച്ചറിയുന്നു.

വെബ് പേജുകൾ എങ്ങനെ നിർമ്മിക്കാമെന്ന് മുൻ അധ്യായങ്ങളിൽ നാം പഠിച്ചുവല്ലോ? ഒട്ടേറെ വെബ്പേജുകൾ ചേർത്തു നിർമിക്കപ്പെടുന്ന വെബ്സൈറ്റ് ഒരു സ്ഥാപനത്തിന്റെയോ ഉല്പന്നത്തിന്റേയോ സേവനത്തിന്റേയോ വിവരങ്ങൾ നൽകുന്നവയായിരിക്കും. നമ്മുടെ സ്കൂളിന്റെ ഒരു വെബ്സൈറ്റ് HTML ഉപയോഗിച്ച് നിർമിച്ചു എന്നിരിക്കട്ടെ. ഈ വെബ്സൈറ്റ് ഇന്റർനെറ്റിൽ എങ്ങനെ ലഭ്യമാക്കാം? അങ്ങനെ ലഭ്യമാക്കണമെങ്കിൽ ഈ വെബ്പേജുകളെ ഇന്റർനെറ്റുമായി ബന്ധമുള്ള ഒരു വെബ്സെർവറിൽ സൂക്ഷിക്കണം. വെബ് ഹോസ്റ്റിംഗിനെപ്പറ്റിയുള്ള അവലോകനം, വിവിധ തരം വെബ്ഹോസ്റ്റിങ്ങുകൾ അതിന്റെ പ്രത്യേകതകൾ തുടങ്ങിയവ ഈ അധ്യായത്തിൽ പ്രതിപാദിക്കുന്നു. കൂടാതെ ഒരു വെബ്സൈറ്റിനെ തിരിച്ചറിയാനുള്ള ഡൊമൈൻനാമം (ഡൊമൈൻ നെയിം) എങ്ങനെ തെരഞ്ഞെടുക്കാം എന്നും അത് എങ്ങനെ രജിസ്റ്റർ ചെയ്യാം എന്നും ഇവിടെ ചർച്ചചെയ്യുന്നു. വിവിധ FTP ക്ലയന്റ് സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് നമ്മുടെ കമ്പ്യൂട്ടറിൽ നിന്ന് സെർവറിലേയ്ക്ക് ഫയലുകൾ (വെബ്പേജുകൾ, ചിത്രങ്ങൾ മുതലായവ) എങ്ങനെ മാറ്റാമെന്നും ഇതിൽ വിശദമാക്കുന്നു. ഈ അധ്യായം പഠിച്ചുകഴിയുമ്പോൾ ഒരു ഡൊമൈൻ നാമം രജിസ്റ്റർ ചെയ്യാനും വെബ്സൈറ്റ് ഹോസ്റ്റ് ചെയ്യാനും പഠിതാവിന് കഴിയും.

7.1 വെബ് ഹോസ്റ്റിംഗ് (Web hosting)

നമ്മുടെ സ്കൂളിനായി ഒട്ടേറെ പേജുകളുള്ള ഒരു വെബ്സൈറ്റ് എങ്ങനെ നിർമിക്കുമെന്ന് മുൻ അധ്യായങ്ങളിൽ നാം പഠിച്ചിട്ടുണ്ട്. ഏതെങ്കിലും ഒരു ടെക്സ്റ്റ് എഡിറ്റർ അല്ലെങ്കിൽ ഒരു വെബ്ഡിസൈൻ ടൂൾ ഉപയോഗിച്ച് നമ്മുടെ വെബ്സൈറ്റ് നിർമിക്കാം. ഹോം പേജിൽ സ്കൂളിലെ സൗകര്യങ്ങൾ, കോഴ്സുകൾ മേൽവിലാസം മറ്റ് വെബ് പേജുകളിലേയ്ക്കു ള്ള ലിങ്കുകൾ എന്നിവ ഉൾപ്പെടുത്തി ആകർഷണീയമാക്കാം.

നമ്മുടെ കമ്പ്യൂട്ടറിൽ സ്കൂളിന്റെ വെബ്സൈറ്റ് നിർമാണം പൂർത്തിയാക്കിയ ശേഷം അത് ഇന്റർനെറ്റിൽ ലഭ്യമാക്കണം. അതിനായി നമ്മുടെ കൈവശമുള്ള വെബ്സൈറ്റ് ഏതെങ്കിലും വെബ്സെർവറിലേയ്ക്ക് അപ്ലോഡ് ചെയ്തെങ്കിൽ മാത്രമേ ലോകത്തു ള്ള എല്ലാവർക്കും അത് വീക്ഷിക്കാനാവൂ. വെബ്സൈറ്റ് സൂക്ഷിക്കുവാനായി വെബ്സെർവ റിലെ കുറച്ച് സ്ഥലം വാടകയ്ക്ക് എടുക്കുകയോ അല്ലെങ്കിൽ നമ്മുടേതായ ഒരു വെബ് സെർവർ ക്രമീകരിക്കുകയോ ചെയ്യാം. ഒരു വെബ്സെർവർ ക്രമീകരിക്കുക എന്നത് വളരെ ചെലവേറിയ ഒരു പ്രക്രിയയാണ്. ആയതിനാൽ നിലവിലുള്ള ഒരു വെബ്സെർവറിന്റെ സ്ഥലം വാടകയ്ക്കെടുക്കുന്നതായിരിക്കും ഉചിതം.

വെബ്ഹോസ്റ്റിംഗ് എന്നാൽ ഇന്റർനെറ്റിൽ ലഭ്യമാക്കേണ്ട വെബ്സൈറ്റിലെ ഫയലുകളെ വെബ്സെർവറിൽ സൂക്ഷിക്കുകയും അതിനാവശ്യമായ സേവനം നൽകുകയും ചെയ്യുന്ന പ്രക്രിയയാണ്. വെബ്ഹോസ്റ്റിങ് സേവനങ്ങൾ നൽകുന്ന കമ്പനികളെ വെബ്ഹോസ്റ്റ് എന്നുവിളിക്കും. വെബ് സെർവറുകൾ വെബ്ഹോസ്റ്റുകളുടെ ഉടമസ്ഥതയിലും നിയന്ത്ര ണത്തിലും ഉള്ളതായിരിക്കും. തടസമില്ലാതെ ഇന്റർനെറ്റ് ബന്ധം നൽകുവാനും PHP, JAVA, ASP.NET തുടങ്ങിയ പ്രോഗ്രാമിങ് ഭാഷകൾക്ക് ഡാറ്റ ബേസ് പിൻബലം ആവ ശ്യമുള്ളപ്പോൾ നൽകാനും ഇത്തരം സെർവറുകൾക്ക് കഴിയും.

7.1.1 വിവിധ തരം വെബ് ഹോസ്റ്റിംഗുകൾ (Types of web hosting)

HTML, ചിത്രങ്ങൾ എന്നിവ അടങ്ങിയ നമ്മുടെ സ്കൂൾ വെബ്സൈറ്റിന് 4 എം.ബി. സ്ഥലമാണ് വെബ്സെർവറിൽ വേണ്ടത് എന്ന് കരുതുക. എന്നാൽ വെബ്ഹോസ്റ്റുകൾ 10 എം.ബി., 20 എം.ബി തുടങ്ങിയ പാക്കേജുകളായാണ് ഇന്ന് സാധാരണയായി ലഭ്യമാക്കു ന്നത്. അപ്പോൾ നമ്മുടെ ആവശ്യത്തിനനുസരിച്ച് ഏറ്റവും അനുയോജ്യമായ ഒരു പാക്കേജ് തിരഞ്ഞെടുക്കേണ്ടതുണ്ട്. നമ്മുടെ വെബ്സൈറ്റിൽ എത്ര ആളുകൾ സന്ദർശിക്കുമെന്നത് ഈ തിരഞ്ഞെടുപ്പിന് ഒരു മാനദണ്ഡമാണ്. നമ്മുടെ വെബ്സൈറ്റിൽ ഡാറ്റബേസും സ്ക്രിപ്റ്റുകളും ഉണ്ടെങ്കിൽ അതിനാവശ്യമായ സേവനം ലഭ്യമാകുന്ന വെബ്ഹോസ്റ്റുക ളെയാണ് തിരഞ്ഞെടുക്കേണ്ടത്. വെബ്ഹോസ്റ്റുകൾ വിവിധതരത്തിലുള്ള ഹോസ്റ്റിങ് പാക്കേജുകൾ ലഭ്യമാക്കുന്നു. ഷെയേർഡ് ഹോസ്റ്റിങ്, വെർച്ചൽ പ്രൈവറ്റ് ഹോസ്റ്റിങ്, ഡെഡിക്കേറ്റഡ് ഹോസ്റ്റിങ് എന്നിങ്ങനെയാണ് അവയെ തരം തിരിച്ചിരിക്കുന്നത്.

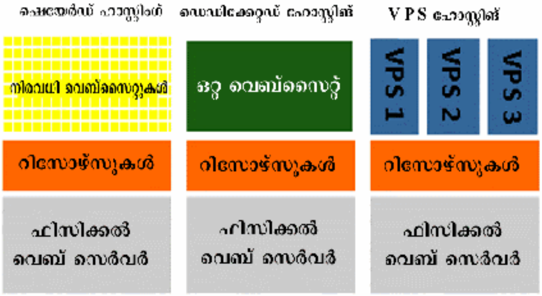
a. ഷെയേർഡ് ഹോസ്റ്റിങ് (Shared hosting): വെബ് ഹോസ്റ്റിങിലെ സർവസാധാ രണമായ ഹോസ്റ്റിങ് രീതിയാണ് ഷെയേർഡ് വെബ്ഹോസ്റ്റിങ്. പല വെബ്സൈറ്റുകൾ ഒരു വെബ്സെർവറിൽ സൂക്ഷിക്കുകയും സെർവറിന്റെ RAM, CPU എന്നിവ പങ്കിടു കയും ചെയ്യുന്നതുകൊണ്ടാണ് ഇതിനെ ഷെയേർഡ് ഹോസ്റ്റിങ് എന്നു വിളിക്കുന്നത്. ഇത് ഉയർന്ന ബാൻഡ് വിഡ്ത്തും ഉയർന്ന സംഭരണ സ്ഥലവും ആവശ്യമുള്ള വെബ്സൈ

റ്റുകൾക്ക് അനുയോജ്യമല്ല. സന്ദർശകർ കുറവുള്ള വെബ്സൈറ്റുകൾക്കാണ് ഷെയേർഡ് ഹോസ്റ്റിംഗ് കൂടുതൽ ഇണങ്ങുന്നത്. ഇത് വളരെ ചെലവ് കുറഞ്ഞതും എളുപ്പത്തിൽ ഉപയോഗിക്കാവുന്നതുമാണ്. അതുകൊണ്ട് തന്നെ ഇതുവളരെ വ്യാപകമായി ഉപയോഗിക്കുന്നു. സെർവർ നവീകരണവും സോഫ്റ്റ്വെയർ ഇൻസ്റ്റാൾ ചെയ്യുമ്പോൾ ഉണ്ടാകുന്ന സുരക്ഷാപ്രശ്നങ്ങളും ഹോസ്റ്റിങ് കമ്പനി തന്നെ പരിഹരിക്കും. നിരവധി വെബ്സൈറ്റുകൾ ഇതിന്റെ ബാൻഡ് വിഡ്ത്ത് പങ്കിടുന്നതുകൊണ്ട് ഏതെങ്കിലും ഒരു വെബ്സൈറ്റിന് വലിയ അളവിലുള്ള ട്രാഫിക്ക് ഉണ്ടാവുകയാണെങ്കിൽ ഈ വെബ്സെർവറിലെ എല്ലാ വെബ്സൈറ്റുകളുടെയും പ്രവർത്തനം മന്ദഗതിയിലാകും എന്നതാണ് ഇതിന്റെ പ്രധാനപ്പെട്ട ന്യൂനത.

b. ഡെഡിക്കേറ്റഡ് ഹോസ്റ്റിങ് (Dedicated hosting): ഉപഭോക്താവ് ഒരു വെബ്സെർവറും അനുബന്ധ സൗകര്യങ്ങളും പൂർണ്ണമായും വാടകയ്ക്ക് എടുക്കുന്നതിനെ ഡെഡിക്കേറ്റഡ് ഹോസ്റ്റിങ് എന്ന് വിളിക്കുന്നു. ഈ വെബ് സെർവർ മറ്റ് വെബ്സൈറ്റുകളുമായി വിഭവങ്ങൾ പങ്കിടുന്നില്ല. വലിയ സ്ഥാപനങ്ങളും സർക്കാർ വകുപ്പുകളും ധാരാളം സന്ദർശകർ ഉള്ളതുകൊണ്ട് ഡെഡിക്കേറ്റഡ് വെബ്ഹോസ്റ്റിങാണ് തിരഞ്ഞെടുക്കുന്നത്. ഇവിടെ ഹാർഡ്വെയറും സോഫ്റ്റ്വെയറും തിരഞ്ഞെടുക്കുന്നതിനുള്ള സ്വാതന്ത്ര്യവും വെബ്സെർവറിന്റെ പൂർണ്ണ നിയന്ത്രണവും ഉപയോക്താവിനാണ്. ഡെഡിക്കേറ്റഡ് സെർവറിന്റെ കാര്യക്ഷമത എപ്പോഴും ഉയർന്ന നിലവാരത്തിലുള്ളതും, എന്നാൽ വളരെ ചെലവേറിയതുമാണ്. ഡെഡിക്കേറ്റഡ് സെർവറുകൾ സാധാരണ ഗതിയിൽ ഡാറ്റ സെന്ററുകളിൽ സ്ഥാപിച്ചിരിക്കുന്നതിനാൽ ഇന്റർനെറ്റ് കണക്ഷനും വൈദ്യുതിയും വിദഗ്ധ സേവനവും എപ്പോഴും ലഭ്യമാകും. ഇതുമൂലം ഇന്റർനെറ്റ് കണക്ഷനും വൈദ്യുതി സംബന്ധമായുള്ള സൗകര്യങ്ങളും ക്രമീകരിക്കുന്നതിനേക്കാൾ കുറവായിരിക്കും. ബാന്റ് വിഡ്ത്ത് പങ്കുവയ്ക്കപ്പെടാത്തതിനാൽ വളരെ വേഗത്തിൽ വെബ്സൈറ്റുകൾ ഉപയോഗിക്കുവാൻ ഉപയോക്താവിന് സാധ്യമാകും. സേവനദാതാവിന്റെ ഭൗതിക സൗകര്യത്തിനുള്ളിൽ ഉപയോക്താവ് സ്വന്തമായി വാങ്ങിയ വെബ് സെർവർ പ്രവർത്തിക്കുവാൻ ക്രമീകരണം ഒരുക്കുകയാണെങ്കിൽ അതിനെ കോ ലോക്കേഷൻ എന്നുവിളിക്കും.

c. വെർച്വൽ പ്രൈവറ്റ് സെർവർ (Virtual Private Server): വെർച്വൽ പ്രൈവറ്റ് സെർവർ എന്നാൽ വെർച്വൽ ടെക്നോളജി ഉപയോഗിച്ച് പലതായി വിഭജിക്കപ്പെട്ട ഒരു വെബ്സെർവറാണ്. ഓരോ VPS ഉം ഡെഡിക്കേറ്റഡ് സെർവർ പോലെ പ്രവർത്തിക്കും. അതിനായി പ്രത്യേക ഓപ്പറേറ്റിങ് സിസ്റ്റം വെബ്സെർവർ സോഫ്റ്റ് വെയർ പാക്കേജുകളായ ഇ-മെയിൽ, ഡാറ്റാബേസ് തുടങ്ങിയവയും അതിൽ സറാപിച്ചിരിക്കും. ഷെയേർഡ് ഹോസ്റ്റിങ് പോലെ ഇവയെ കരുതേണ്ടതില്ല. കാരണം VPS-ൽ ഒരു നിശ്ചിതശതമാനം RAM ഓരോ വെർച്വൽ വെബ്സെർവറിനുമായി മാറ്റിവച്ചിരിക്കുന്നു. ഓരോ VPS-ഉം പൂർണ്ണ സ്വാതന്ത്ര്യമുള്ള വെബ്സെർവറായി പ്രവർത്തിക്കും. അതായത് ഒരു പ്രത്യേക ഭൗതിക സെർവറിൽ പ്രവർത്തിക്കുന്നതുപോലെ VPS-ന്റെ ഉപഭോക്താവിന് സോഫ്റ്റ് വെയർ ഇൻസ്റ്റാൾ ചെയ്യാനും അവ ക്രമീകരിക്കാനും പൂർണ്ണ അവകാശം ഉണ്ടായിരിക്കും. മറ്റ് വെർച്വൽ സെർവറുകളെ ബാധിക്കാതെ പുനരാരംഭിക്കാൻ ഓരോ VPS-നും അവകാശമുണ്ട്. VPS-ഹോസ്റ്റിങ് സെർവറിലെ ഓരോ വെബ്സൈറ്റിനും തനതായ ബാന്റ് വിഡ്ത്ത് ലഭ്യമാക്കും. ഇതുമൂലം സെർവർ ഷെയർ ചെയ്യപ്പെട്ടാലും ഡെഡിക്കേറ്റഡ്

ഹോസ്റ്റിങിന്റെ നേട്ടം കൈവരിക്കാനാകും. ഷെയേർഡ് ഹോസ്റ്റിങിനെക്കാളും സവിശേഷതകൾ ആവശ്യമുള്ള വെബ്സൈറ്റുകൾക്കാണ് ഇത്തരം ഹോസ്റ്റിങ് യോജിക്കുന്നത്. പക്ഷേ ഡെഡിക്കേറ്റഡ് ഹോസ്റ്റിങിന്റെ എല്ലാ പ്രത്യേകതകളും ഇവിടെ ലഭ്യമല്ല. ഡെഡിക്കേറ്റഡ് ഹോസ്റ്റിങിന്റെ ഒട്ടുമിക്ക സർവ്വീസുകളും മിതമായ നിരക്കിൽ നൽകുവാൻ VPS-ഹോസ്റ്റിങിന് കഴിയും. അറിയപ്പെടുന്ന ചില സെർവർ വെർച്വലൈസേഷൻ സോഫ്റ്റ് വെയറുകളാണ് VM വെയർ, വെർച്വൽബോക്സ്, ഫ്രീ VPS, യൂസർ മോഡ് ലിനക്സ്, മൈക്രോസോഫ്റ്റ് ഹൈപ്പർ വി തുടങ്ങിയവ.

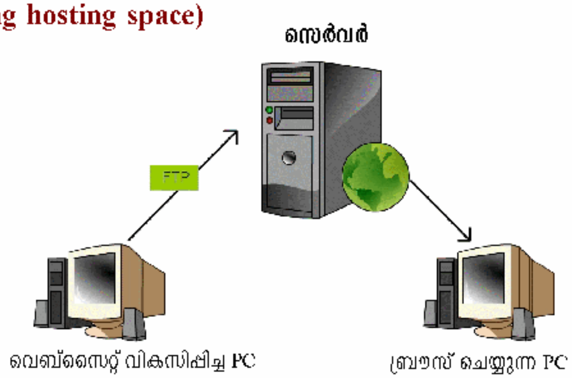


ചിത്രം 7.1: വെബ് ഡിസൈനിംഗ് തരങ്ങൾ

വിവിധതരത്തിലുള്ള ഹോസ്റ്റിംഗ് പാക്കേജുകളെപ്പറ്റിയുള്ള പ്രതീകാത്മക വിവരണം ചിത്രം 7.1 ൽ നൽകിയിരിക്കുന്നു.

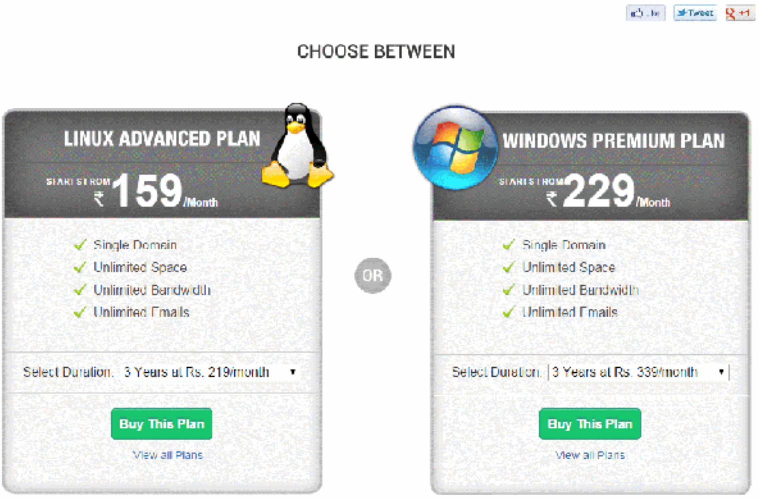
7.1.2 ഹോസ്റ്റിങിന് സ്ഥലം വാങ്ങുക (Buying hosting space)

സ്കൂളിന് വേണ്ടി നിർമ്മിച്ച വെബ്സൈറ്റ് ഇപ്പോൾ നമ്മുടെ കമ്പ്യൂട്ടറിലെ ഒരു ഫോൾഡറിൽ സൂക്ഷിച്ചിരിക്കുകയാണ്. ഇത് ഇന്റർനെറ്റിൽ ലഭ്യമാകാൻ ഈ ഫയലുകളെ വെബ്സെർവറിലേയ്ക്ക് പകർത്തണം. ചിത്രം 7.2 ൽ ഇത് കാണിച്ചിരിക്കുന്നു.



ചിത്രം 7.2 : ഇന്റർനെറ്റിൽ ഒരു വെബ്സൈറ്റ് ലഭ്യമാക്കുന്നു

ഇതിനായി അനുയോജ്യമായ ഒരു വെബ്ഹോസ്റ്റിങ് രീതി നാം തിരഞ്ഞെടുക്കേണ്ടതുണ്ട്. ഷെയേർഡ് ഹോസ്റ്റിങ് നമ്മുടെ വെബ്സൈറ്റിന് അനുയോജ്യമായതിനാലും, ചെലവ് കുറവായതിനാലും അത് തിരഞ്ഞെടുക്കുന്നതാവും കൂടുതൽ ഉത്തമം. ഏതു തരം ഹോസ്റ്റിങ് സെർവറാണെന്ന് തരം തീരുമാനിച്ചു കഴിഞ്ഞാൽ നമ്മുടെ വെബ്സൈറ്റിലെ ഫയലുകൾ സൂക്ഷിക്കാനാവശ്യമായ



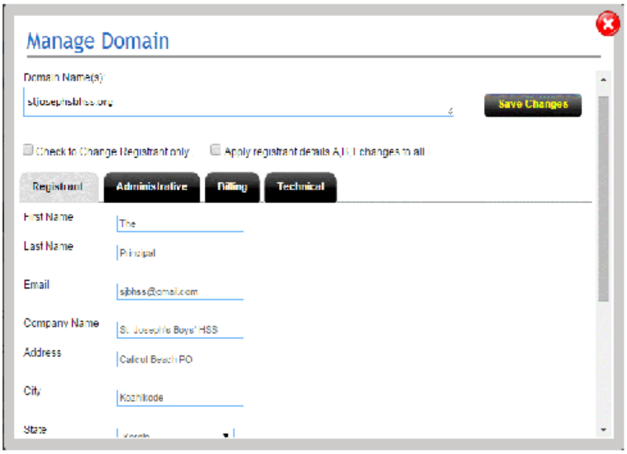
ചിത്രം 7.3 : ഹോസ്റ്റിംഗ് ഓപ്ഷനുകൾ

സെർവർ സ്ഥലം സേവന ദാതാവിൽ നിന്ന് വാങ്ങണം. ഹോസ്റ്റിങ് സ്ഥലം വാങ്ങുമ്പോൾ നിരവധി കാര്യങ്ങൾ പരിഗണിക്കേണ്ടതായുണ്ട്. ഒന്നാമതായി നമ്മുടെ വെബ്സൈറ്റിന് എത്രസ്ഥലം ആവശ്യമുണ്ടെന്ന് കണ്ടെത്തണം. വെബ്പേജിൽ പ്രോഗ്രാമിങ് ഉള്ളടക്കം ഉണ്ടെങ്കിൽ വെബ്സെർവറിൽ ഈ സേവനം നൽകുന്ന സാങ്കേതിക വിദ്യ ഉണ്ടായിരിക്കണം. വെബ്പേജ് നിർമ്മാണത്തിനായി ഉപയോഗിച്ചിരിക്കുന്ന പ്രോഗ്രാം, വിന്റോസ് ഹോസ്റ്റിങിനാണോ ലിനെക്സ് ഹോസ്റ്റിങിനാണോ അനുയോജ്യമെന്ന് കണ്ടെത്തണം. അതിനുശേഷം വിൻഡോസ് സെർവറോ ലിനെക്സ് സെർവറോ തിരഞ്ഞെടുക്കണം. ചിത്രം 7.3 ഇത് വ്യക്തമാക്കുന്നു. വെബ്സൈറ്റിൽ HTML കോഡ് മാത്രമാണ് ഉപയോഗിച്ചിരിക്കുന്നതെങ്കിൽ ഏത് തരം സെർവറും ഉപയോഗിക്കാം. എന്നാൽ ഡാറ്റാ ബേസ്, ഇ-മെയിൽ എന്നിവയ്ക്ക് ആവശ്യമായ സാങ്കേതിക സൗകര്യം ആവശ്യമുണ്ടെങ്കിൽ അത് കൂടി പരിഗണിച്ച് വേണം വെബ്ഹോസ്റ്റ് തിരഞ്ഞെടുക്കാൻ.

7.1.3 ഡൊമൈൻ നെയിം രജിസ്റ്റർ ചെയ്യുക (Domain name registration)

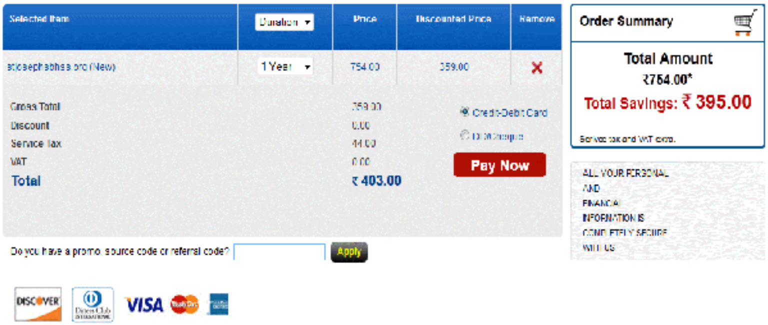
സ്കൂൾ വെബ്സൈറ്റിന് ആവശ്യമായ ഹോസ്റ്റിങ് സ്ഥലം നാം വാങ്ങിക്കഴിഞ്ഞു. ഇന്റർനെറ്റിൽ ഓരോ വെബ്സൈറ്റും തിരിച്ചറിയുന്നതിനായി പ്രത്യേകം URL ആവശ്യമാണ്. ഇതിനായി അനുയോജ്യമായ ഒരു ഡൊമൈൻ നാമം സ്കൂൾ വെബ്സൈറ്റിനായി നാം തിരഞ്ഞെടുക്കണം. ഇന്റർനെറ്റിൽ ഒരു വെബ്സൈറ്റ് തിരിച്ചറിയാൻ ഡൊമൈൻ സഹായിക്കുന്നതാണ്. മിക്ക വെബ്ഹോസ്റ്റിങ് കമ്പനികളും ഡൊമൈൻ നാമം രജിസ്റ്റർ ചെയ്യുവാനുള്ള സേവനം നൽകുന്നുണ്ട്. നമ്മുടെ വെബ്സൈറ്റിന് അനുയോജ്യമായ ഒരു ഡൊമൈൻ നാമം തിരഞ്ഞെടുത്തശേഷം ഇവ ലഭ്യമാണോ എന്നും മറ്റാരെങ്കിലും ഇതേ പേരിൽ രജിസ്റ്റർ ചെയ്തിട്ടുണ്ടോ എന്നും പരിശോധിക്കണം. www.whois.net പോലെയുള്ള സൈറ്റുകളും വെബ്ഹോസ്റ്റിങ് കമ്പനികളുടെ വെബ്സൈറ്റുകളും ഇത്തരം പരിശോധനയ്ക്ക് നമ്മെ സഹായിക്കുന്നു. രജിസ്റ്റർ ചെയ്യപ്പെട്ട എല്ലാ ഡൊമൈൻ നാമങ്ങളുടെയും ഡാറ്റാബേസായ ICANN ഡാറ്റാബേസ് പരിശോധിച്ച് മറുപടി നൽകുന്നു. ചിത്രം 7.4 ഇത് സൂചിപ്പിക്കുന്നു. ഈ ഡൊമൈൻ നാമം ലഭ്യമാണെങ്കിൽ ICANN ന്റെ whois.net ഡാറ്റാബേസിൽ പേര്,

ചിത്രം 7.4 : ഡൊമൈൻ നാമ രജിസ്ട്രേഷന്റെ തിരയൽ ഫലം



ചിത്രം 7.5 : WHOIS വിവരങ്ങൾ

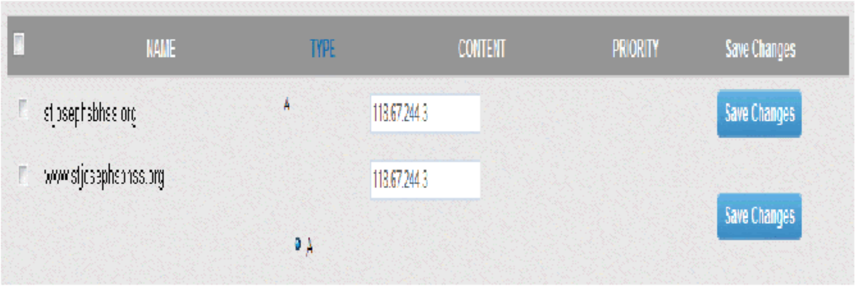
മേൽവിലാസം, ഫോൺ നമ്പർ, ഇ-മെയിൽ എന്നിവ ചിത്രം 7.5-ൽ കാണിച്ചിട്ടുള്ളതു പോലെ രേഖപ്പെടുത്തണം. ഈ വിവരങ്ങൾ രജിസ്റ്റർ ചെയ്യുന്ന ആളിന്റെ ഇഷ്ടാനുസരണം പൊതുവായോ വ്യക്തിപരമായോ സൂക്ഷിക്കുന്നതാണ്. വാർഷിക വരിസംഖ്യ ഓൺലൈനായി അടച്ച് നമ്മുടെ പേരിൽ ഡൊമൈൻനാമം രജിസ്റ്റർ ചെയ്യാവുന്നതാണ്. വാങ്ങുന്നതിന്റെ ഷോപ്പിങ് കാർട്ട് ചിത്രം 7.6 ൽ പ്രദർശിപ്പിച്ചിരിക്കുന്നു.



ചിത്രം 7.6 : ഡൊമൈൻ വാങ്ങൽ

ഇപ്പോൾ നമ്മൾ വെബ്സെർവറിൽ സ്കൂൾ വെബ്സൈറ്റിനായുള്ള സ്ഥലം വാങ്ങി ഡൊമൈൻ നാമം രജിസ്റ്റർ ചെയ്തു. ഏതെങ്കിലും ഉപയോക്താവ് നമ്മുടെ ഡൊമൈൻ നാമം (stjosephhss.org) ബ്രൗസർ വിൻഡോയിൽ രേഖപ്പെടുത്തിയാൽ നാം വെബ്സെർവറിൽ സൂക്ഷിച്ചിരിക്കുന്ന വെബ്പേജ് ദൃശ്യമാകും. DNS നമ്മുടെ സെർവറിന്റെ IP മേൽവിലാസം ബ്രൗസറിൽ ലഭ്യമാക്കുകയും ആ IP മേൽവിലാസം ഉപയോഗിച്ച് നമ്മുടെ വെബ്പേജ് സൂക്ഷിച്ചിരിക്കുന്ന വെബ് സെർവറുമായി ബന്ധമുണ്ടാക്കുകയും ചെയ്യുന്നതുമൂലമാണ് ഇത് സാധ്യമാകുന്നത്. വെബ് സെർവറുമായി ബന്ധിപ്പിച്ചിട്ടുള്ള ഡൊമൈൻ നാമത്തിന്റെ IP മേൽവിലാസം സൂക്ഷിച്ചുവയ്ക്കുന്നതിനാണ് A.റിക്കോർഡ് ഉപയോഗിക്കുന്നത്. ഡൊമൈനിന്റെ കൺട്രോൾ പാനലിൽ ലോഗ് ഇൻ ചെയ്തതിനുശേഷം A. റിക്കോർഡ് പുതുക്കാവുന്നതാണ്.

വെബ്സെർവറിന്റെ IP മേൽവിലാസം സൂചിപ്പിക്കുന്നതിനായി ചിത്രം 7.7 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ A റിക്കോർഡ് ക്രമീകരിക്കാം. ഇതിന് ശേഷം DNS, സെർവറുകൾക്ക് നാം നൽകിയ വെബ്സൈറ്റ് സൂക്ഷിച്ചിരിക്കുന്ന വെബ്സെർവറിലേയ്ക്ക് ബ്രൗസർ വഴി ബന്ധപ്പെടാനുള്ള IP മേൽവിലാസം ലഭ്യമാകും. അതു വഴി നമ്മുടെ വെബ്സെർവറിലേയ്ക്ക് വളരെ വേഗം ബന്ധപ്പെടാനാവും.



ചിത്രം 7.7 : ഡൊമൈനിന്റെ A റിക്കോർഡ് മാറ്റുന്ന ജാലകം

ഒരു ഡൊമൈനെകുറിച്ചുള്ള വിവരങ്ങൾ **WHO IS search** -ൽ ലഭ്യമാകും. ഡൊമൈൻ ഉടമസ്ഥത എവിടെ എപ്പോൾ രജിസ്റ്റർ ചെയ്തു, കാലാവധി തുടങ്ങിയ വിവരങ്ങൾ ഇതിൽ ഉൾപ്പെടും. ഒരു പ്രത്യേക ഡൊമൈൻ പേര് ലഭ്യമാണോ എന്ന് കണ്ടെത്തുന്നതിന് ഇത് ഉപയോഗിക്കും. **www.kerala.gov.in** എന്നത് **WHOIS** ൽ നോക്കുമ്പോൾ അത് താഴെ പറയുന്ന വിവരങ്ങൾ ലഭ്യമാക്കും.

```
Domain ID:D8944-AFIN
Domain Name:KERALA.GOV.IN
Created On:31-Dec-2003 05:00:00 UTC
Last Updated On:16-Jul-2014 11:37:59 UTC
Expiration Date:31-Dec-2016 05:00:00 UTC
Sponsoring Registrar:National Informatics Centre (R12-AFIN)
Status:OK
Registrant ID:R-R03120114034
Registrant Name:Government of Kerala
Registrant Organization:
Registrant Street1:Chief Minister's Office
Registrant Street2:
Registrant Street3:
Registrant City:Government Secretariate, Trivandrum, 69500
Registrant State/Province:Kerala
Registrant Postal Code:695001
Registrant Country:IN
```

നമുക്കു ചെയ്യാം

ജനപ്രീതിയാർജ്ജിച്ച വെബ്ഹോസ്റ്റുകളുടെയും അവയുടെ ഏറ്റവും കുറഞ്ഞ അളവിലുള്ള ഹോസ്റ്റിംഗ് സ്ഥലത്തിന് ഒരു വർഷത്തേക്ക് നൽകേണ്ട വിലകളും താരതമ്യം ചെയ്യുന്ന ടേബിൾ തയ്യാറാക്കുക. വിൻഡോസിനും, ലിനക്സ് വെബ്ഹോസ്റ്റിങ്ങിനും ഉള്ള വിലകൾ പ്രത്യേകമായി തയ്യാറാക്കുക.

പ്രശസ്തരായ വെബ്ഹോസ്റ്റ് സേവനദാതാക്കളുടെ .org, .com ഡൊമൈൻ രജിസ്ട്രേഷനുകൾക്ക് ഒരു വർഷത്തെ വിലവിവരപ്പട്ടിക താരതമ്യം ചെയ്യുന്ന ടേബിൾ തയ്യാറാക്കുക.

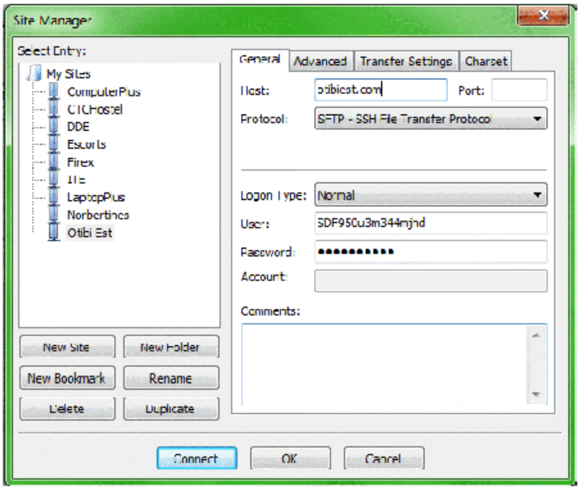
www.scert.kerala.gov.in, www.dhsekerala.gov.in എന്നീ വെബ്സൈറ്റുകളുടെ WHOIS വിവരങ്ങൾ www.whois.net ൽ തിരഞ്ഞശേഷം അവയുടെ ചാർട്ട് തയ്യാറാക്കുക.

നിങ്ങളുടെ പുരോഗതി അറിയാം

1. വെബ്ഹോസ്റ്റിങ് സേവനം നൽകുന്ന കമ്പനികളെ ____________ എന്നു പറയുന്നു.
2. വെബ്ഹോസ്റ്റിങ് തരം നിശ്ചയിക്കുന്ന ഘടകങ്ങൾ തയ്യാറാക്കുക.
3. വി.പി.എസ്. എന്നത്
   a. Virtual Premium Service   b. Virtual Private Service
   c. Virtual Premium Server   e. Virtual Private Server
4. കോ-ലൊക്കേഷൻ എന്നാലെന്ത്?
5. WHOIS വിവരങ്ങൾ ഉൾക്കൊള്ളുന്നതെന്തൊക്കെ?
6. ഒരു ഡൊമൈൻ നെയിമിന് A റിക്കോർഡ് പ്രാധാന്യമുള്ളതാകുന്നതെന്തുകൊണ്ട്?

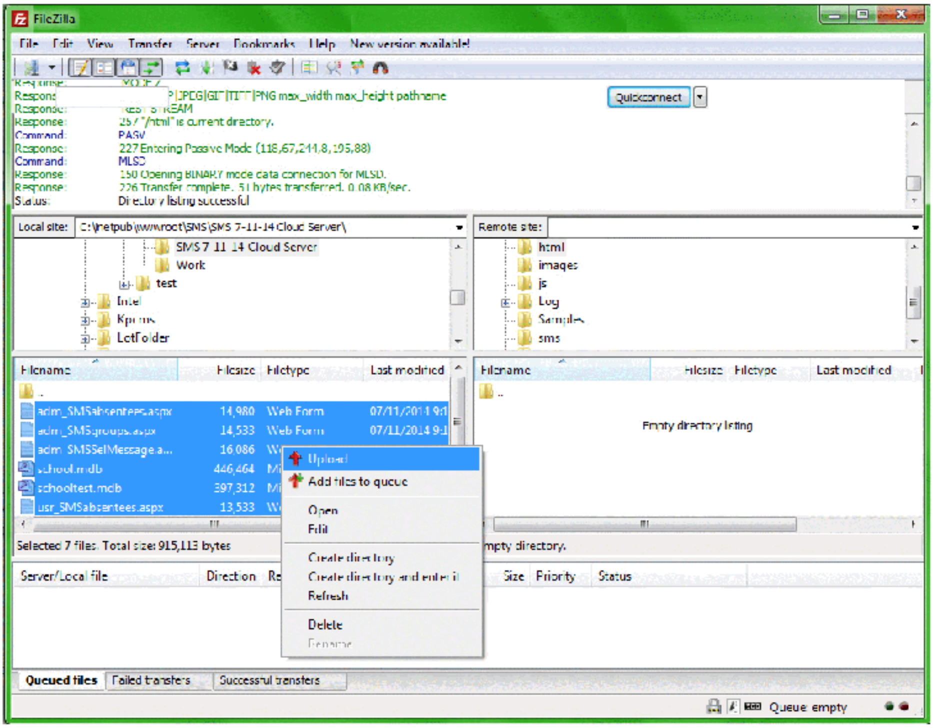
7.1.4 എഫ്.റ്റി.പി. ക്ലയന്റ്സോഫ്റ്റ്വെയർ (FTP client software)

ഹോസ്റ്റിങിനായി സെർവർ സ്ഥലവും ഡൊമൈൻ നാമവും വാങ്ങിയശേഷം നമ്മുടെ കമ്പ്യൂട്ടറിൽ സൂക്ഷിച്ചിരിക്കുന്ന വെബ്സൈറ്റ് ഫയലുകളെ വെബ്സെർവറിലേയ്ക്ക് പകർത്തേണ്ടതായുണ്ട്. ഇതിനായി എഫ്.റ്റി.പി. ക്ലയന്റ് സോഫ്റ്റ്വെയർ ഉപയോഗിക്കുന്നു.



ചിത്രം 7.8 : FTP സോഫ്റ്റ് വെയറിന്റെ ലോഗിൻ പേജ്

FTP യെ കുറിച്ച് കമ്പ്യൂട്ടർ ശൃംഖലകൾ എന്ന അധ്യായത്തിൽ 11-ാം ക്ലാസിൽ നാം ചർച്ച ചെയ്തു. ഇന്റർനെറ്റുമായി ബന്ധിപ്പിച്ചിരിക്കുന്ന ഒരു കമ്പ്യൂട്ടറിൽ നിന്ന് മറ്റൊരു കമ്പ്യൂട്ടറിലേയ്ക്ക് ഫയൽ നീക്കുന്നതിന് FTP ഉപയോഗിക്കുന്നു. FTP ക്ലയന്റ് സോഫ്റ്റ്വെയർ വിദൂര സെർവറുമായി ബന്ധപ്പെടുകയും അതിനുശേഷം കമ്പ്യൂട്ടറിലെ ഫയലുകൾ സെർവർ കമ്പ്യൂട്ടറിലേയ്ക്ക് മാറ്റുകയും ചെയ്യുന്നു. FTP സെർവറുമായി ബന്ധം ഉണ്ടാക്കുവാൻ FTP ക്ലയന്റ് സോഫ്റ്റ്വെയറിന് യൂസർനെയിം, പാസ്‌വേർഡ്, ഡൊമൈൻ നാമം എന്നിവ

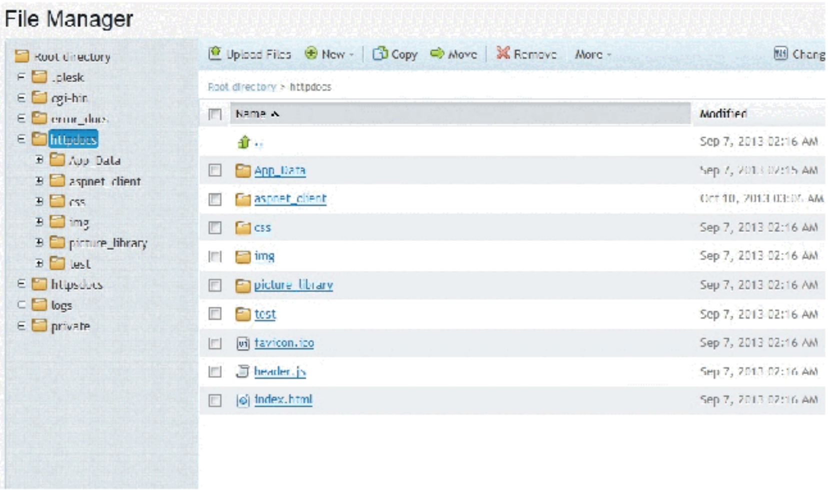


ചിത്രം 7.9 : FTP ക്ലൈന്റ് സോഫ്റ്റ്വയറിന്റെIDE

ആവശ്യമാണ്. ചിത്രം 7.8 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ സൈറ്റ് മാനേജർ ഡയലോഗ് ബോക്സിൽ ഇവ ലഭ്യമാക്കണം. സുരക്ഷിതമല്ലാത്ത ഒരു ലളിതമായ ടെക്സ്റ്റിന്റെ രൂപത്തിൽ ഈ യൂസർനാമവും പാസ്‌വേർഡും സെർവറിലേയ്ക്ക് FTP അയയ്ക്കും. എന്നാൽ ഇപ്പോൾ SSH FTP പ്രോട്ടോക്കോൾ ഈ വിവരങ്ങൾ മറ്റൊരു രൂപത്തിലാക്കി വിവർത്തനം ചെയ്ത് (ENCRYPT) വെബ്സെർവറിലേയ്ക്ക് FTP യുടെ സഹായത്തോടെ അയയ്ക്കുന്നു. സെക്യൂർഷെൽ പ്രോട്ടോക്കോൾ (SSH)ഉപയോഗിച്ച് സുരക്ഷിതമായ ഫയൽനീക്കത്തിന് SFTP (SSH FTP) സഹായിക്കുകയും ചെയ്യും.

FTP ക്ലയന്റ് സോഫ്റ്റ് വെയറിന്റെ ആധികാരികത ഉറപ്പിച്ച് കഴിഞ്ഞാൽ അതിന്റെ IDE-ചിത്രം 7.9 -ലേതുപോലെ പ്രത്യക്ഷമാകും. ഈ ചിത്രത്തിൽ ഇടതുവശത്ത് പ്രദർശിപ്പിച്ചിരിക്കുന്ന ഫോൾഡർ നമ്മുടെ കമ്പ്യൂട്ടറിലേതും വലതുവശത്തേത് വെബ്സെർവർ കമ്പ്യൂട്ടറിലേതും ആകുന്നു. മെനു ഉപയോഗിച്ചോ ഡ്രാഗ് ചെയ്തോ ഇടതുവശത്തുള്ള ഫയലിനെ വലതുവശത്തുള്ള ജാലകത്തിലേക്കാക്കി വെബ്സെർവറിലേയ്ക്ക് അപ്‌ലോഡ് ചെയ്യാം. ഇങ്ങനെ നമ്മുടെ കമ്പ്യൂട്ടറിലെ ഫയലുകൾ വെബ്സെർവറിലേക്ക് പകർത്താം. പ്രമുഖമായ ചില FTP സോഫ്റ്റ് വെയറുകളാണ് ഫയൽസില്ല, ക്യൂട്ട് FTP., സ്മാർട്ട് FTP. തുടങ്ങിയവ.

ചില വെബ്ഹോസ്റ്റിങ് കമ്പനികൾ ഫയലുകൾ അപ്‌ലോഡ് ചെയ്യുന്നതിന് അവരുടേതായ കൺട്രോൾ പാനലുകൾ സജ്ജീകരിച്ചിട്ടുണ്ട്. ഇവ പുറത്ത് നിന്നുള്ള FTP ക്ലൈന്റ് സോഫ്റ്റ് വെയർ വഴി ഫയൽ അപ്‌ലോഡ് ചെയ്യുവാൻ അനുവദിക്കുകയില്ല. ചിത്രം 7.10 ഇത്തരത്തിലുള്ള ഒരു കൺട്രോൾ പാനൽ പ്രദർശിപ്പിച്ചിരിക്കുന്നു.



ചിത്രം 7.10 : ഹോസ്റ്റിംഗ് സ്ഥാപനങ്ങൾ ലഭ്യമാക്കുന്ന FTP കൺട്രോൾ പാനൽ

7.2 സൗജന്യ ഹോസ്റ്റിങ് (Free hosting)

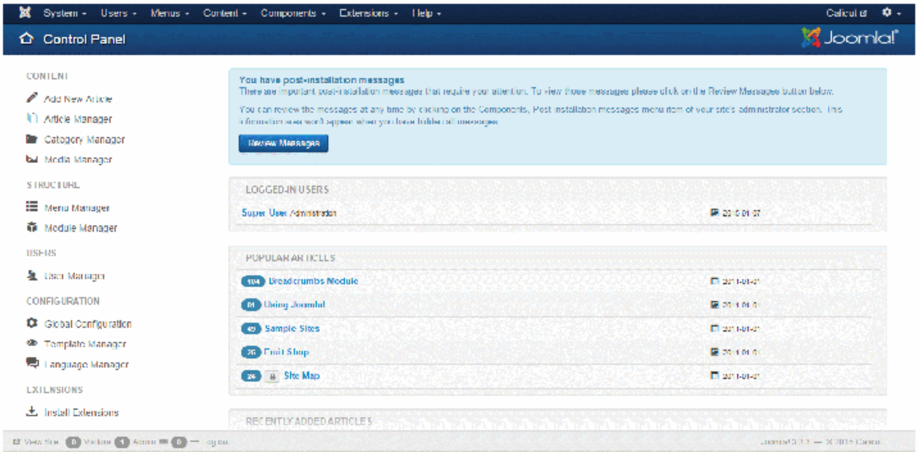
വെബ് ഹോസ്റ്റിങ് സൗകര്യം സൗജന്യമായി നൽകുന്നതാണ് സൗജന്യ ഹോസ്റ്റിംഗ്. ഇവിടെ സേവന ദാതാവ് ഈ വെബ്സൈറ്റിൽ ചില പരസ്യങ്ങൾ നൽകി ഇതിന്റെ ചെലവ് കണ്ടെത്തുന്നു. സൗജന്യവെബ്ഹോസ്റ്റിംഗ് സൈറ്റുകൾ നമ്മുടെ വെബ്സൈറ്റിലെ ഫയലുകളെ സെർവറിൽ സൂക്ഷിക്കുന്നു. എന്നാലിതിന് ചില നിയന്ത്രണങ്ങളുണ്ട്. ഫയലിന്റെ വലുപ്പം നിയന്ത്രണവിധേയമായിരിക്കും. (5 MBവരെ) ശബ്ദവും വീഡിയോയും ചിലതിൽ അനുവദനീയമല്ല. ചില വെബ്സൈറ്റുകൾ അതിന്റെ ഒരു ടെംപ്ലേറ്റ്

അടിസ്ഥാനമാക്കി വെബ്സൈറ്റ് രൂപകൽപന ചെയ്യുവാൻ മാത്രമേ അനുവദിക്കുകയുള്ളൂ. മറ്റ് ഫയലുകൾ വെബ്സെർവറിൽ അപ്‌ലോഡ് ചെയ്യുവാനും അനുവദിക്കില്ല. ചില സൗജന്യ വെബ്ഹോസ്റ്റിങ് കമ്പനികൾ ഡൊമൈൻ നെയിം രജിസ്ട്രേഷനും അനുവദിക്കുന്നു. ഈ വെബ്സൈറ്റുകൾ ഫയലുകളെ അപ്‌ലോഡ് ചെയ്യുന്നതിനുള്ള കൺട്രോൾ പാനലുകൾ ചിത്രം 7.10 -ലേതുപോലെ നൽകുന്നു. ചില വെബ്സൈറ്റുകൾ മുൻ നിശ്ചയിച്ച ഘടനയിൽ മാത്രമേ വെബ്സൈറ്റ് നിർമിക്കുവാൻ അനുവദിക്കുകയുള്ളൂ. സൗജന്യ വെബ്ഹോസ്റ്റിംഗ് സൗകര്യം അവരുടേതായ ഒരു സബ് ഡൊമൈനോ അല്ലെങ്കിൽ ഒരു ഡയറക്ടറി സർവ്വീസോ ഉപയോഗിച്ചാണ് നമ്മുടെ വെബ്സൈറ്റിനെ ലഭ്യമാക്കുവാൻ സഹായിക്കുന്നത്. ഒരേ ആശയങ്ങളുള്ളവരുമായി വിവരങ്ങൾ പങ്കിടാനും, ലാഭേച്ഛയില്ലാത്ത പ്രസ്ഥാനങ്ങൾക്കും അധിക ചെലവ് വഹിക്കാനാകാത്തവർക്കും സൗജന്യ വെബ്ഹോസ്റ്റിങ് സഹായകരമാണ്. വളരെ ചെലവ് കുറഞ്ഞ രീതിയിൽ വെബ്ഹോസ്റ്റിങ് സൗകര്യം നിലവിൽ വന്നതോടുകൂടി സൗജന്യ വെബ്ഹോസ്റ്റിങ് കുറഞ്ഞുവരുന്നു. sites.google.com, yola.com എന്നിവ സൗജന്യ വെബ്ഹോസ്റ്റിങ് സർവ്വീസുകളാണ്. ഹയർസെക്കന്ററി അധ്യാപകർക്കും വിദ്യാർഥികൾക്കും ഏറെ പ്രയോജനകരമായ വിവരങ്ങൾ നൽകുന്ന hsslive.in എന്ന വെബ്സൈറ്റിന് സൗജന്യ ഹോസ്റ്റിങ് ആണ് ഉപയോഗിച്ചിരിക്കുന്നത്. ചിത്രം 7.11 ൽ ഇവ പ്രദർശിപ്പിച്ചിരിക്കുന്നു.

ചിത്രം 7.11 : സൗജന്യ വെബ്ഹോസ്റ്റിംഗ്

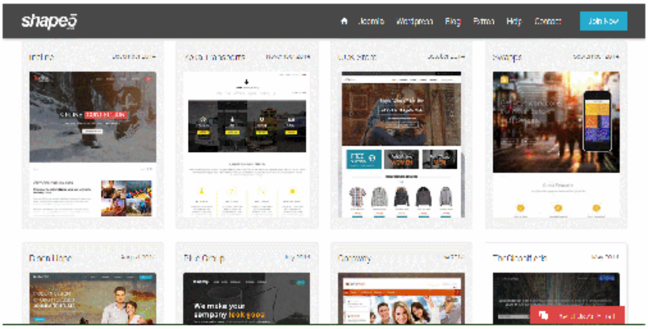
7.3 കണ്ടന്റ് മാനേജ്മെന്റ് സിസ്റ്റം (Content Management System)

വെബ്സൈറ്റുകൾ നിർമ്മിക്കുന്നതിനും നിയന്ത്രിക്കുന്നതിനും അവയിലേയ്ക്ക് പുതിയ വിവരങ്ങൾ ഉൾപ്പെടുത്തുന്നതിനും സാധ്യമായ ഒരു ഇന്റർനെറ്റ് അധിഷ്ഠിതമായ സോഫ്റ്റ് വെയർ സംവിധാനമാണ് കണ്ടന്റ് മാനേജ്മെന്റ് സിസ്റ്റം. മിക്ക CMS കളും അവയുടെ ഔദ്യോഗിക വെബ്സൈറ്റിൽ നിന്ന് സൗജന്യമായി ഡൗൺലോഡ് ചെയ്യാം. ഈ ഫയലുകളെ നമ്മുടെ വെബ്സൈറ്റിലെ ഹോസ്റ്റിങ് സ്ഥലത്തേക്ക് കോപ്പി ചെയ്യാം. അതിനു ശേഷം ചിത്രം 7.12 ലെപോലെ അതിനെ പ്രവർത്തനസജ്ജമാക്കുകയും ചെയ്യാം.



ചിത്രം 7.12 : ജൂംല ഉപയോഗിച്ച് വെബ്സൈറ്റ് പ്രവർത്തന സജ്ജമാക്കുന്നു

വെബ്സൈറ്റുകളിൽ CMS കളായി വേണ്ടി മുൻപ് രൂപകല്പന ചെയ്ത് വച്ചിട്ടുള്ള ചില ടെംപ്ലേറ്റുകൾ ലഭ്യമാകും. ചിത്രം 7.13-ൽ ഇവയുടെ ഉദാഹരണം നൽകിയിരിക്കുന്നു. ഉപയോക്താവിന് ഈ ടെംപ്ലേറ്റുകളിൽ ഒന്ന് തെരഞ്ഞെടുത്ത് നമ്മുടെ വെബ്സൈറ്റിൽ അപ്‌ലോഡ് ചെയ്യാം. അതിനുശേഷം അതിന്റെ ശീർഷകങ്ങളും ചിത്രങ്ങളും മറ്റ് വിവരങ്ങളും നമ്മുടെ ഇഷ്ടപ്രകാരം മാറ്റുകയും ചെയ്യാം.



ചിത്രം 7.13 : ജൂംലയിൽ ലഭിക്കുന്ന ടെംപ്ലേറ്റുകൾ

ഇത് വേഡ് പ്രോസസ്സറിൽ ഒരു ഫയൽ നിർമിക്കുന്നത്ര ലളിതമാണ്. CMS-കളിൽ അടിസ്ഥാന നിയന്ത്രണ സംവിധാനങ്ങൾ അതിന്റെ ഡിസൈൻ സമയത്ത് ഉൾപ്പെടുത്തിയിട്ടുണ്ട്, ഇത് സാങ്കേതിക അറിവ് ഇല്ലാത്തവർക്ക് പോലും ഒരു വെബ്സൈറ്റ് നിർമിക്കാൻ സഹായകമാകും. CMS -ലെ ടെംപ്ലേറ്റുകൾ കോഡുകളുടെ ആവർത്തനം ഒഴിവാക്കുകയും ശീർഷകം രൂപകല്പന ചെയ്യുന്നതിനും എല്ലാപേജിലും മെനു ദൃശ്യമാക്കുന്നതിനും ചില പ്രത്യേക സംവിധാനങ്ങൾ ഒരുക്കിയിട്ടുണ്ട്. പ്രമുഖമായ ചില CMS വെബ്സൈറ്റുകളിൽ ഇവയുടെ ടെംപ്ലേറ്റുകൾ സൗജന്യമായോ മിതമായ തുകയ്ക്കോ ലഭ്യമാണ്. CMS-നെ നമ്മുടെ വെബ്സൈറ്റിന് അനുയോജ്യമാം

ചിത്രം 7.14 : ജൂംലയിൽ ഇങ്ങടെ ഉപയോഗിച്ച് വികസിപ്പിച്ച ഒരു വെബ്പേജ്

വിധം ക്രമപ്പെടുത്തി നൽകുന്ന വ്യക്തികളും സ്ഥാപനങ്ങളും ഇന്നുണ്ട്. ഒരു ഷെയേർഡ് വെബ്സെർവറിലാണ് വെബ്സൈറ്റ് ഹോസ്റ്റ് ചെയ്യുന്നതെങ്കിൽ അതിൽ CMS ഫയലുകൾ ഉൾപ്പെടുത്താനാകുമോ എന്ന് ശ്രദ്ധിക്കണം.

സാമ്പത്തികലാഭം ഉള്ളതുകൊണ്ട് ധാരാളം സംഘടനകളും ബ്ലോഗറുകളും അവരുടെ വെബ്സൈറ്റിനായി സി.എം.എസ്. ഉപയോഗിക്കുന്നു. പ്രമുഖമായ ചില C M S സോഫ്റ്റുവെയറുകളാണ് വേഡ്പ്രസ്, ദ്രുപാൽ, ജൂംല എന്നിവ ചിത്രം 1.4-ൽ ജൂംല ഉപയോഗിച്ച് നിർമ്മിച്ച മോട്ടോർ വാഹന വകുപ്പിന്റെ വെബ്സൈറ്റ് കാണാനാകും.

7.4 റെസ്പോൺസീവ് വെബ്രൂപകൽപന (Responsive webdesign)

ഇന്ന് ഇന്റർനെറ്റുമായി ബന്ധപ്പെടുവാൻ കമ്പ്യൂട്ടർ, ലാപ്ടോപ്പ്, ടാബ്ലെറ്റ്, മൊബൈൽ ഫോൺ തുടങ്ങിയ ധാരാളം ഉപകരണങ്ങൾ ഉണ്ട്. പല ഉപകരണങ്ങൾക്കും വ്യത്യസ്ത സ്ക്രീൻവലുപ്പമാണുള്ളത്. പണ്ട് നിർമിച്ചിട്ടുള്ള വെബ്പേജുകൾ ഡെസ്ക് ടോപ്പിനും ലാപ്ടോപ്പിനും അനുയോജ്യമാംവിധം ഉള്ളതാണ്. ഈ വെബ്സൈറ്റുകൾ ടാബ്ലെറ്റും, മൊബൈൽഫോണും വഴി കാണുവാൻ പ്രയാസമാണ്. അല്ലെങ്കിൽ ഉപഭോക്താവ് ഈ വെബ്പേജിന്റെ ഓരോ ഭാഗവും കാണുവാൻ സ്ക്രോൾബാർ വശങ്ങളിലേയ്ക്ക് നീക്കി നോക്കണം. മുൻകാലത്ത് മൊബൈൽവഴി വെബ്സൈറ്റ് കാണുവാൻ പ്രത്യേക വെബ്സൈറ്റുകൾ നിർമിച്ചിരുന്നു. ഇവ മൊബൈലിന്റെ സ്ക്രീൻ വലുപ്പത്തിന് അനുയോജ്യമാംവിധമാണ് നിർമിച്ചിരുന്നത്. എന്നാൽ ഒരു സ്ഥാപനത്തിന് രണ്ട് വെബ്സൈറ്റുകൾ കൈകാര്യം ചെയ്യേണ്ടിവരുന്നത് പ്രയാസമാണ്. അതുകൊണ്ട് വെബ്പേജിന്റെ വലുപ്പം സ്ക്രീനിന്റെ വലുപ്പത്തിന് അനുസരിച്ച് മാറുന്ന രീതിയിൽ രൂപപ്പെടുത്തിയാൽ അത് കൂടുതൽ പ്രയോജനകരമാകും. ഇത്തരത്തിലുള്ള വെബ്പേജ് രൂപകൽപനയെ റെസ്പോൺസീവ് വെബ്രൂപകൽപന എന്നു പറയും. വിവിധ ഉപകരണങ്ങളിൽ ഒരു റെസ്പോൺസീവ് വെബ്സൈറ്റ് എങ്ങനെ ദൃശ്യമാകുമെന്ന് ചിത്രം 7.15 ൽ കാണിച്ചിരിക്കുന്നു.

എല്ലാ ഉപകരണങ്ങളിലും പ്രവർത്തിക്കത്തക്ക വിധമാണ് റെസ്പോൺസീവ് വെബ്ഡിസൈനിംഗിൽ വെബ്സൈറ്റ് രൂപപ്പെടുത്തുന്നത്. അതുകൊണ്ടുതന്നെ സ്ക്രീൻ വലുപ്പമോ ഉപകരണമേതെന്നോ ഇവിടെ പരിഗണിക്കപ്പെടുന്നില്ല. റെസ്പോൺസീവ് വെബ്ഡിസൈനിങ് എന്ന ആശയം ഒരു സ്വതന്ത്ര ഡിസൈനറായ ഈഥാൻ മർക്കോട്ടിന്റേതാണ്. സ്ഥിരമായി മാറിക്കൊണ്ടിരിക്കുന്ന ഒരു വെബ്രൂപകൽപന രീതിയെപ്പറ്റി അദ്ദേഹം വിവരിക്കുന്നു. ഈ വെബ്രൂപകൽപന രീതി ഫ്ളക്സിബിൾ ഗ്രിഡ് ലേഔട്ടും ഫ്ളക്സിബിൾ ഇമേജും മീഡിയകൊറിയും ഉപയോഗിച്ച് പ്രാവർത്തികമാക്കിയിരിക്കുന്നു. ഫ്ളക്സിബിൾ ഗ്രിഡ് ലേഔട്ട് സൗകര്യം ഉപയോഗിച്ച് ഉപകരണത്തിന്റെ പ്രദർശനതലത്തിന്റെ വലുപ്പം പരിഗണിച്ച് വെബ്പേജിനെ അതിന് അനുസൃതമായി സജ്ജീകരിക്കുകയും ചെയ്യും. മീഡിയകൊറി ഓരോ ഉപകരണത്തിനും അനുസരിച്ചുള്ള വ്യത്യസ്ത ശൈലിയിൽ ലഭ്യമാക്കും. ഒരു വെബ് പേജിലെ തിരശ്ചീന മെനു മൊബൈലിൽ ഡ്രോപ്ഡൗൺ മെനു ആയി

ചിത്രം 7.15 : റെസ്പോൺസീവ് വെബ്രൂപകൽപന

പരിവർത്തനം ചെയ്യപ്പെടും. CSS ഫയലിനുള്ളിലെ മീഡിയകൊറിയാണ് ഇതിന് സഹായിക്കുന്നത്. ഇവിടെ പ്രദർശനതലത്തിന്റെ വലുപ്പം മാറുന്നതിന് അനുസരിച്ച് വെബ്സൈറ്റിന്റെ വലുപ്പവും ഭാവവും മാറിക്കൊണ്ടിരിക്കും.

നമുക്കു ചെയ്യാം

ഇന്ത്യൻ ക്രിക്കറ്റ് ടീം അംഗങ്ങളുടെ പേര്, വയസ്സ്, നേടിയ റണ്ണുകളുടെ എണ്ണം, എടുത്ത വിക്കറ്റുകളുടെ എണ്ണം എന്നിവ ഉൾക്കൊളളുന്ന ഒരു വെബ്സൈറ്റ് രൂപകൽപന ചെയ്ത് ഫ്രീ ഹോസ്റ്റിങ് സേവനങ്ങൾ ഉപയോഗിച്ച് ഇതിനെ ഹോസ്റ്റ് ചെയ്യുക.

ജനകീയമായ CMS ദാതാക്കളുടെ ഒരു ലിസ്റ്റ് തയ്യാറാക്കുന്നതിനോടൊപ്പം അവരുടെ സവിശേഷതകളും തയ്യാറാക്കുക.

നിങ്ങൾ തയാറാക്കിയ സ്കൂൾ വെബ്സൈറ്റുകളുടെ ഫയലുകൾ ഏതെങ്കിലും FTP സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് ലഭ്യമായ വെബ് ഹോസ്റ്റിങ് സ്ഥലത്ത് അപ്‌ലോഡ് ചെയ്യുക.

നമുക്കു സംഗ്രഹിക്കാം

ഒരു വെബ്സൈറ്റ് നിർമ്മിച്ചതിനുശേഷം അനുയോജ്യമായ ഒരു ഹോസ്റ്റിങ് രീതി ഉപയോഗിച്ച് ഇന്റർനെറ്റിൽ ലഭ്യമാക്കാം. ചെറിയ സ്ഥാപനങ്ങൾ തങ്ങളുടെ വെബ്സൈറ്റിന് ഷെയേർഡ് ഹോസ്റ്റിങ് ഉപയോഗിക്കും. ഉയർന്ന ട്രാഫിക്കുള്ളതും ഉയർന്ന സുരക്ഷ വേണ്ടതുമായ വെബ്സൈറ്റുകൾക്ക് വി.പി.എസ്. ഹോസ്റ്റിങ് ഉപയോഗിക്കും. എന്നാൽ വലിയ സ്ഥാപനങ്ങളുടെ വെബ്സൈറ്റുകളിൽ നിരന്തരം ഉയർന്ന ട്രാഫിക്കുകൾ ഉണ്ടാകുന്നതുകൊണ്ട് ഡെഡിക്കേറ്റഡ് ഹോസ്റ്റിങ് ഉപയോഗിക്കും. വെബ്ഹോസ്റ്റിങ്ങിന് ആവശ്യമായ സ്ഥലം വാങ്ങിയതിനുശേഷം FTP സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് നമ്മുടെ വെബ്സൈറ്റ് ഫയലുകളെ വെബ്സെർവറിലേയ്ക്ക് മാറ്റാം. അതിനുമുമ്പ് ഡൊമൈൻ നാമം സേവന ദാതാവ് വഴി രജിസ്റ്റർ ചെയ്യണം. വെബ്സെർവറിലെ നമ്മുടെ ഡൊമൈൻ സൂചിപ്പിക്കുവാൻ A റിക്കോർഡ് സജ്ജമാക്കണം. സൗജന്യനിരക്കിൽ വെബ്സൈറ്റിന് ലഭ്യമാക്കുന്ന വെബ്സെർവറുകൾ ലഭ്യമാണ്. കണ്ടന്റ് മാനേജ്മെന്റ് സംവിധാനം ചില ഉപകരണങ്ങളും ചില അടിസ്ഥാന സുരക്ഷിത സൗകര്യങ്ങളും ഒരുക്കി സാങ്കേതികവിദ്യകളിൽ അറിവ് കുറഞ്ഞ ആളുകൾക്കുപോലും വെബ്സൈറ്റുകൾ നിർമ്മിക്കുവാനുളള സാധ്യത നൽകുന്നു. വ്യത്യസ്ത ഉപകരണങ്ങൾ ഉപയോഗിച്ച് ബ്രൗസ് ചെയ്യുന്നതുകൊണ്ട് സ്ക്രീനിന്റെ വലുപ്പമനുസരിച്ച് വെബ്സൈറ്റുകൾ ലഭ്യമാക്കുക എന്നത് പ്രാധാന്യത്തോടെ കാണേണ്ടിയിരിക്കുന്നു. ഇതിനായി റസ്പോൺസീവ് വെബ്രൂപകൽപന ഉപയോഗിക്കുന്നു.

നമുക്കു വിലയിരുത്താം

1. വെബ്ഹോസ്റ്റിങ് എന്നതുകൊണ്ട് എന്താണ് അർഥമാക്കുന്നത്? പലതരത്തിലുള്ള വെബ്ഹോസ്റ്റിംഗിനെ കുറിച്ച് വിശദീകരിക്കുക.

2. നഗരത്തിലുള്ള ഒരു സൂപ്പർമാർക്കറ്റ് സ്ഥാപനം അവരുടെ വ്യാപാരം ഓൺ ലൈൻ ആക്കുവാൻ ഉദ്ദേശിക്കുന്നു. വെബ്സൈറ്റ് വഴി സാധനങ്ങൾ വിൽക്കാനും പണമിട പാടുകൾ ഓൺലൈൻ വഴി നടത്തുവാനും അവർ ആസൂത്രണം ചെയ്യുന്നുവെങ്കിൽ

   a. ഏത് തരത്തിലുള്ള ഹോസ്റ്റിങ് ആണ് ഈ വെബ്പേജിന് അനുയോജ്യമാകുക?

   b. അത് തിരഞ്ഞെടുക്കുവാനുള്ള കാരണവും വിശദമാക്കുക.

3. എമിൽ സ്വന്തം മരുന്നു കടയുടെ വെബ്സെർവറിൽ വെബ് ഹോസ്റ്റിംഗ് സ്ഥലം വാങ്ങുവാൻ ആഗ്രഹിക്കുന്നു. വെബ്സെർവറിൽ ഹോസ്റ്റിംഗിനായി സ്ഥലം വാങ്ങു മ്പോൾ പരിഗണിക്കേണ്ട പ്രത്യേകതകൾ പട്ടികപ്പെടുത്തുക.

4. ഒരു ഡൊമൈൻ നാമവും വെബ്സർവറിൽ സംഭരിച്ചിട്ടുള്ള വെബ്സൈറ്റും തമ്മിൽ എങ്ങനെയാണ് പരസ്പരം ബന്ധിപ്പിക്കുന്നത്?

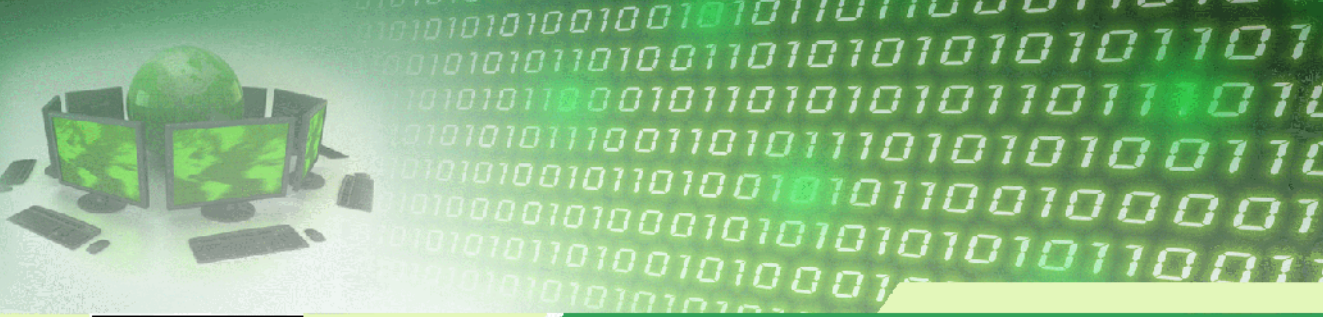
5. FTP സോഫ്റ്റ്‌വെയറിൽ SFTP (പ്രോട്ടോകോൾ) ഉപയോഗിക്കുന്നത് കൊണ്ടുള്ള മേന്മയെന്താണ്?

6. സ്വന്തം കുടുംബത്തിനായി ഒരു വെബ്സൈറ്റ് ഹോസ്റ്റ് ചെയ്യുവാൻ രാജു ആഗ്രഹി ക്കുന്നു. സൗജന്യ വെബ്സൈറ്റ് ഹോസ്റ്റിംഗ് കമ്പനികൾ നൽകുന്ന സംവിധാന ങ്ങൾ എന്തൊക്കെയാണ്?

7. CMS എന്നാൽ എന്താണ്? CMS -ന്റെ പ്രത്യേകതകൾ എന്തൊക്കെ? ഉദാഹരണ ങ്ങൾ നൽകുക.

8. ഇന്നത്തെ കാലത്ത് വെബ്സൈറ്റിന്റെ നിർമ്മാണത്തിൽ റെസ്പോൺസീവ് വെബ് സൈറ്റ് രൂപകൽപനയുടെ ആവശ്യകത വിശദീകരിക്കുക.

9. എങ്ങനെയാണ് റെസ്പോൺസീവ് വെബ്പേജ് രൂപകൽപന നടപ്പിലാക്കുന്നത്?



Z9X1U1

8 ഡാറ്റാബേസ് നിർവഹണ സംവിധാനം

പ്രധാന പഠനനേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പഠനം പൂർത്തിയാകുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ

- ഫയലുകളുടെ ആവശ്യം മനസ്സിലാക്കുന്നു.
- പരമ്പരാഗത ഫയൽ നിർവഹണ സംവിധാനത്തിന്റെ പ്രധാന പരിമിതികൾ തിരിച്ചറിയുന്നു.
- ഡാറ്റാബേസ് നിർവഹണ സംവിധാനത്തിന്റെ (DBMS) വിവിധ ഗുണങ്ങൾ വിശദീകരിക്കാനറിയുന്നു.
- DBMS ന്റെ വിവിധ ഘടകങ്ങളും അവയുടെ ഉദ്ദേശ്യവും വിശദീകരിക്കുന്നു.
- DBMS ലെ വിവിധതരം ഉപയോക്താക്കളെയും അവരുടെ കടമകളും തിരിച്ചറിയുന്നു.
- DBMS ലെ ഡാറ്റ സംഗ്രഹത്തിന്റെയും (Abstraction) ഡാറ്റ സ്വാതന്ത്ര്യത്തിന്റെയും (Independence) വിവിധ തലങ്ങൾ വിശദീകരിക്കുന്നു.
- ഉദാഹരണങ്ങൾ ഉദ്ധരിച്ചുകൊണ്ട് റിലേഷണൽ ഡാറ്റ മോഡൽ വിശദീകരിക്കുന്നു.
- RDBMS ലെ വ്യത്യസ്ത പദങ്ങൾ അനുയോജ്യമായി ഉപയോഗിക്കുന്നു.
- റിലേഷണൽ ബീജഗണിതത്തിലെ വിവിധ പ്രവർത്തനങ്ങൾ പ്രയോഗിക്കുകയും വിലയിരുത്തുകയും ചെയ്യുന്നു.

ഇത് അറിവിന്റെയും വിവര സാങ്കേതിക വിദ്യയുടെയും കാലമാണ്. മത്സരയോട്ടങ്ങളുടെ ഈ ലോകത്തിൽ സ്ഥാപനങ്ങളുടെ നിലനിൽപ്പ് ഉയർന്ന കൃത്യതയോടും വേഗതയോടും ആവശ്യമുള്ള വിവരങ്ങൾ ലഭിക്കുന്നതിനെ ആശ്രയിച്ചിരിക്കും. വിവരങ്ങൾ ലഭ്യമാകുന്നത് ഡാറ്റ സംസ്ക്കരണത്തിലൂടെയാണെന്ന് നമുക്ക് അറിയാം. വിവരങ്ങൾ രൂപപ്പെടുത്തുന്നതിനായി വളരെ അധികം ഡാറ്റ സ്വീകരിക്കുകയും സംഭരിക്കുകയും പ്രക്രിയകൾ നടത്തുകയും ചെയ്യുന്നു.

ബാങ്കുകൾ, ബിസിനസ്സ് സ്ഥാപനങ്ങൾ, സ്കൂളുകൾ മുതലായവയ്ക്ക് വിവരങ്ങൾ ആവശ്യമാണെന്ന് നിങ്ങൾക്ക് അറിയാമല്ലോ. പരമ്പരാഗതമായി ഈ സ്ഥാപനങ്ങൾ വളരെയധികം ഡാറ്റ കൈകാര്യം ചെയ്തത് എങ്ങനെയാണെന്ന് നിങ്ങൾക്ക് ഊഹിക്കാനാകുമോ? മുൻകാലങ്ങളിൽ ബുക്ക് കീപ്പിംഗ് മാതൃകയിൽ, അതായത് വിവരങ്ങൾ പുസ്തകങ്ങളിൽ കരകൃതമായി (manual processing) എഴുതി കൈകാര്യം ചെയ്യുന്ന രീതി അവലംബിച്ചിരുന്നു. ഈ രീതിയിൽ എഴുതിയ പുസ്തകങ്ങൾ സൂക്ഷിക്കുന്നതിന് ധാരാളം സ്ഥലം ആവശ്യമാണെന്നും ഡാറ്റ സംസ്കരണം വളരെ പ്രയാസകരമാണെന്നതും വ്യക്തമാണ്. കമ്പ്യൂട്ടറുകളുടെ ആവിർഭാവത്തോടെ ഡാറ്റ ഫലപ്രദമായി സംഭരിക്കുവാൻ സാധിക്കുന്നു, എന്നാൽ പകർപ്പ്, പൊരുത്തക്കേട്, അസാധാരണത്വം മുതലായവയ്ക്കുള്ള സാധ്യത നിലനിൽക്കുന്നു. ഈ പരിമിതികൾ മറികടക്കാനുള്ള ഫലപ്രദ

മായ റെക്കോർഡ് കീപ്പിങ് സംവിധാനമെന്ന നിലയിൽ 'ഡാറ്റാബേസ് മാനേജ്മെന്റ് സിസ്റ്റം' (DBMS) എന്ന ആശയം ഈ അധ്യായത്തിൽ അവതരിപ്പിച്ചിരിക്കുന്നു. ഡാറ്റാബേസിൽ നിന്ന് ആവശ്യമുള്ളതും പ്രസക്തവുമായ വിവരങ്ങൾ വീണ്ടെടുക്കാനുള്ള വിവിധ പ്രവർത്തനങ്ങൾ ചർച്ച ചെയ്യുന്നുമുണ്ട്.

8.1 ഡാറ്റാബേസ് എന്ന ആശയം (Concept of database)

ഹയർ സെക്കൻഡറി വകുപ്പിന്റെ സ്കൂൾ പ്രവേശന ഏകജാലക സംവിധാനം പരിഗണിക്കുക. പതിനൊന്നാം ക്ലാസ്സ് പ്രവേശനത്തിനായി ഓരോ വർഷവും വിദ്യാർഥികൾ, കോഴ്സുകൾ, സ്കൂളുകൾ, ഗ്രേഡുകൾ എന്നീ ഡാറ്റയുടെ ഒരു വലിയ ശേഖരം (ഏകദേശം 5,00,000 അപേക്ഷകൾ അല്ലെങ്കിൽ 19 GB ഡാറ്റ) പരിപാലിക്കുന്നു. ഈ ഡാറ്റ നിരവധി വിദ്യാലയങ്ങളും വിദ്യാർഥികളും ഒരേ സമയം ഉപയോഗിക്കുന്നു. വിദ്യാർഥികളുടെയും സ്കൂളുകളുടെയും അലോട്ട്മെന്റിനെക്കുറിച്ചുള്ള ചോദ്യങ്ങൾക്ക് വേഗത്തിൽ ഉത്തരം നൽകണം. വ്യത്യസ്ത സ്കൂളുകളുടെ ഡാറ്റയിൽ വരുത്തുന്ന മാറ്റങ്ങൾ സ്ഥിരമായി ഉപയോഗിക്കണം. കൂടാതെ ഡാറ്റയുടെ ചില ഭാഗങ്ങൾ (ഉദാ. ഗ്രേഡുകൾ അല്ലെങ്കിൽ WGPA) ഉപയോഗിക്കുന്നത് നിയന്ത്രിക്കേണ്ടതുമുണ്ട്. പരമ്പരാഗത ഫയൽ നിർവഹണ സംവിധാനത്തിൽ ഡാറ്റ സൂക്ഷിച്ചുകൊണ്ട് നമുക്ക് അതിനെ നിയന്ത്രിക്കാനാവും. എന്നാൽ ഈ രീതിക്ക് പല ന്യൂനതകളുണ്ട്.

- വ്യത്യസ്ത ആപ്ലിക്കേഷനുകൾക്കായി ഒരേ ഡാറ്റയുടെ കൂടുതൽ പകർപ്പുകൾ സൂക്ഷിക്കേണ്ടതുണ്ട്. ഇത്തരം സംഭരണം ഡാറ്റ ആവർത്തിക്കപ്പെടുന്നതിന് ഇടയാക്കുന്നു.
- വ്യത്യസ്ത ഉപയോക്താക്കൾ ഒരേ സമയം ഉണ്ടാക്കുന്ന മാറ്റങ്ങളിൽ നിന്ന് ഡാറ്റയെ പരിരക്ഷിക്കുന്നതിനുള്ള ഒരു സംവിധാനവും ഇല്ല.
- മാറ്റങ്ങൾ വരുത്തുമ്പോൾ സിസ്റ്റം തകരാറിലായാൽ സ്ഥിരമായ ഒരു അവസ്ഥയിലേക്ക് ഡാറ്റ പുനഃസ്ഥാപിക്കുമെന്ന് ഉറപ്പുവരുത്താൻ ഒരു വഴിയും ഇല്ല.
- സുരക്ഷയ്ക്കായി ഓപ്പറേറ്റിംഗ് സംവിധാനങ്ങൾ ഒരു രഹസ്യകോഡ് സംവിധാനം മാത്രമാണ് നൽകുന്നത്. ഡാറ്റയിൽ സുരക്ഷാ നയങ്ങൾ നടപ്പിലാക്കാൻ ഇത് പര്യാപ്തമല്ല.
- ഡാറ്റയിൽ നിലവാരമൊന്നും ഇല്ല.

8.1.1 ഡാറ്റാബേസിന്റെ ആവശ്യകത (Need of database)

മുകളിൽ പറഞ്ഞിരിക്കുന്ന പിഴവുകൾ ഡാറ്റാബേസ് ഉപയോഗിച്ച് തരണം ചെയ്യാവുന്നതാണ്. വലിയ അളവിൽ ഡാറ്റ കൈകാര്യം ചെയ്യാൻ പരമ്പരാഗത ഫയൽ നിർവഹണ സംവിധാനം മതിയാകില്ല. ഇത്തരം സന്ദർഭങ്ങളിൽ ഡാറ്റാബേസ് ഉപയോഗിക്കുന്നു. ഒന്നിലധികം ആപ്ലിക്കേഷനുകൾക്ക് ഉപയോഗിക്കാൻ കഴിയുന്നവിധത്തിൽ ആവർത്തനം അധികമില്ലാതെ സംഭരിച്ച പരസ്പരബന്ധിതമായ വിവരങ്ങളുടെ ഒരു സംഘടിത ശേഖരമാണ് ഡാറ്റാബേസ്. ഹയർ സെക്കണ്ടറി സ്കൂൾ പ്രവേശനത്തിന് ഉപയോഗിക്കുനന്ന ഏകജാലക സംവിധാനം ഡാറ്റയെ ഒരു ഡാറ്റാബേസ് നിർവഹണ സംവിധാനത്തിൽ സൂക്ഷിച്ചു കൊണ്ട് കൈകാര്യ ചെയ്യുന്നതിന് ഒരു ഉദാഹരണമാണ്. അടിസ്ഥാനപരമായി ഒരു ഡാറ്റാബേസ് നിർവഹണ സംവിധാനം എന്നത് ഡാറ്റാബേസിന്റെ സംഭരണം, വീണ്ടെടുക്കൽ, നിർവഹണം മുതലായവയ്ക്ക് സൗകര്യമൊരുക്കുന്ന ഒരു കൂട്ടം പ്രോഗ്രാമുകളാണ്. ഡാറ്റാബേസ് സംഭരിക്കുന്നതിനും വീണ്ടെടുക്കുന്നതിനും സൗകര്യപ്രദവും കാര്യക്ഷമവുമായ ഒരു അന്തരീക്ഷം പ്രദാനം ചെയ്യുക എന്നതാണ് DBMS ന്റെ പ്രാഥമിക ലക്ഷ്യം.

8.1.2 DBMS ന്റെ നേട്ടങ്ങൾ (Advantages of database)

ഡാറ്റാബേസ് സംവിധാനങ്ങൾ വലിയ അളവിലുള്ള ഡാറ്റ കൈകാര്യം ചെയ്യാൻ രൂപകല്പന ചെയ്തിരിയ്ക്കുന്നു. ഡാറ്റ സംഭരണത്തിനായുള്ള ഡാറ്റാബേസ് ഘടനകളുടെ നിർവ ചനവും, ഡാറ്റ കൈകാര്യം ചെയ്യുന്നതിനുള്ള സംവിധാനങ്ങളും DBMSൽ ഉൾപ്പെട്ടിരി ക്കുന്നു. കൂടാതെ, സംഭരിച്ച ഡാറ്റയുടെ അനധികൃത ഉപയോഗം, സിസ്റ്റം പരാജയം മൂലം ഡാറ്റ നഷ്ടമാവുന്ന അവസ്ഥ തുടങ്ങിയ പ്രശ്നങ്ങളിൽ നിന്നുള്ള സുരക്ഷ DBMS ഉറപ്പുവരുത്തുന്നു. പല ഉപയോക്താക്കൾക്കിടയിൽ ഡാറ്റ പങ്കുവയ്ക്കേണ്ടതുണ്ടെങ്കിൽ അതുമൂലമുള്ള അപാകതയുള്ള ഫലങ്ങൾ സിസ്റ്റം ഒഴിവാക്കണം. പരമ്പരാഗത ഫയൽ നിർവഹണ സംവിധാനത്തെക്കാളും ഡാറ്റാബേസ് നിർവഹണ സംവിധാനത്തിന് അനേ കം ഗുണങ്ങളുണ്ട്, അവ താഴെ ചർച്ചചെയ്യാം.

- **ഡാറ്റ റിഡൻഡൻസി നിയന്ത്രിക്കുന്നു:** പരമ്പരാഗത ഫയൽ നിർവഹണ സംവിധാ നത്തിൽ, ഡാറ്റ നിരവധി ഫയലുകളിലായി സംഭരിക്കപ്പെടാം. ഒന്നിലധികം സ്ഥാന ങ്ങളിൽ ഒരേ ഡാറ്റ സംഭരിക്കുക (ഒരേ ഫയലിൽ അല്ലെങ്കിൽ വ്യത്യസ്ത ഫയലുകളിൽ ആയിരിക്കാം) അല്ലെങ്കിൽ ഡാറ്റയുടെ തനിപ്പകർപ്പിന് ഡാറ്റ റിഡൻഡൻസി എന്ന് പറ യുന്നു. റിഡൻഡൻസി മൂലം ഡാറ്റ സംഭരണത്തിനും ഡാറ്റ ഉപയോഗത്തിനും ഉയർന്ന ചെലവ് വരുന്നു. ഡാറ്റാബേസ് സംവിധാനത്തിൽ ഡാറ്റകളുടെ പകർപ്പ് സൂക്ഷിക്കുന്നില്ല, പകരം എല്ലാ ഡാറ്റയും കേന്ദ്രീകൃത രീതിയിൽ ഒരിടത്ത് സൂക്ഷിച്ചിരിക്കുന്നു. ഡാറ്റ ആവശ്യമുള്ള എല്ലാ ആപ്ലിക്കേഷനുകളും ഉപയോക്താക്കളും കേന്ദ്രീകൃതമായി പരിപാ ലിക്കപ്പെട്ട ഡാറ്റാബേസ് ഉപയോഗിക്കുന്നു. സാങ്കേതികമോ ബിസിനസ് പരമോ ആയ കാരണങ്ങൾകൊണ്ട് ചില സന്ദർഭങ്ങളിൽ ഒരേ ഡാറ്റയുടെ നിരവധി പകർപ്പുകൾ സൂക്ഷിക്കാറുണ്ട്. എന്നിരുന്നാലും, ഏത് സാഹചര്യത്തിലും ഡാറ്റയുടെ ആവർത്ത നത്തെ ശ്രദ്ധാപൂർവം നിയന്ത്രിക്കുന്നു.

- **ഡാറ്റയുടെ പൊരുത്തം:** ഡാറ്റ റിഡൻഡൻസി ഡാറ്റയുടെ പൊരുത്തമില്ലായ്മയിലേക്ക് നയിച്ചേക്കാം. അതായത്, ഒരേ ഡാറ്റയുടെ വിവിധ കോപ്പികളിൽ വ്യത്യസ്ത വിലകൾ കാണിക്കാം. നിങ്ങളുടെ ക്ലാസ്സിലെ വിദ്യാർഥികളുടെ വിലാസം നിങ്ങളുടെ ക്ലാസ് അധ്യാപകനും പ്രിൻസിപ്പാളും പ്രത്യേകമായി പരിപാലിക്കുന്നുവെന്ന് കരുതുക. വിലാ സത്തിൽ മാറ്റങ്ങൾ വരുത്തുന്നതിന് ചില വിദ്യാർഥികൾ ക്ലാസ് അധ്യാപകനെയും മറ്റു ചിലർ പ്രിൻസിപ്പാളിനെയും സമീപിക്കുന്നു. ഒരു നിശ്ചിത കാലയളവിനു ശേഷം, രണ്ടു പട്ടികകളും അപ്രസക്തവും പൊരുത്തപ്പെടാത്തതുമാകും. ഡാറ്റ റിഡൻഡൻസി നിയന്ത്രക്കുന്നതിലൂടെ ഡാറ്റയുടെ പൊരുത്തം ഉറപ്പാക്കാം. ഡാറ്റ ഒരിടത്ത് മാത്രം സൂക്ഷിക്കുകയാണെങ്കിൽ, അതിന്റെ വിലകളിലേക്കുള്ള ഏത് മാറ്റവും ആ സ്ഥലത്ത് മാത്രമെ ചെയ്യാനാകൂ, ഒപ്പം മാറ്റം വരുത്തിയ വിലകൾ എല്ലാ ഉപയോക്താക്കൾക്കും ഉടൻ ലഭ്യമാകുകയും ചെയ്യും.

- **കാര്യക്ഷമമായ ഡാറ്റ ഉപയോഗം:** ഡാറ്റ കാര്യക്ഷമമായി സംഭരിക്കാനും വീണ്ടെടു ക്കാനും നിരവധി സാങ്കേതികവിദ്യകൾ DBMS ഉപയോഗപ്പെടുത്തുന്നു.

- **ഡാറ്റസമഗ്രത:** ഡാറ്റസമഗ്രത ഡാറ്റാബേസിലെ ഡാറ്റയുടെ പൂർണത, കൃത്യത, സ്ഥിരത എന്നിവയെ സൂചിപ്പിക്കുന്നു. ഒരു ഡാറ്റ റെക്കോർഡിന്റെ രണ്ട് മാറ്റങ്ങൾക്കിടയിലുള്ള ഡാറ്റയിൽ എന്തെങ്കിലും വ്യതിയാനമുണ്ടെങ്കിൽ അത് സൂചിപ്പിക്കാൻ ഇതിന് കഴിയും. ഡാറ്റാബേസ് രൂപകല്പന ചെയ്യുന്ന ഘട്ടത്തിൽ ശരിയായ നിയമങ്ങളും നടപടി

ക്രമങ്ങളും ഉപയോഗിച്ച് ഡാറ്റസമഗ്രത നടപ്പിൽ വരുത്തുന്നു. തെറ്റുകളുടെ പരിശോധനയും സാധൂകരണ പരിപാടികളും ഉപയോഗിച്ച് ഡാറ്റസമഗ്രത നിലനിർത്താൻ കഴിയും.

- **ഡാറ്റ സുരക്ഷ:** ഡാറ്റാബേസിലുള്ള വിവരങ്ങൾ ഏതൊരു കമ്പനിക്കും സ്ഥാപനത്തിനും മൂല്യമേറിയതാണ്. അതിനാൽ ഇത് സുരക്ഷിതമായും സ്വകാര്യവുമായും സൂക്ഷിക്കണം. അംഗീകൃതമല്ലാത്ത വ്യക്തികൾ നടത്തുന്ന അവിചാരിതമോ കരുതിക്കൂട്ടിയുള്ളതോ ആയ വെളിപ്പെടുത്തലുകൾ, അനധികൃത പരിഷ്കരണം, നശീകരണം എന്നീ പ്രശ്നങ്ങളിൽ നിന്നുള്ള സംരക്ഷണമാണ് ഡാറ്റ സുരക്ഷ കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. വിവിധ പ്രോഗ്രാമുകളും ഉപയോക്താക്കളും പൊതുവായി ഡാറ്റ പങ്കുവയ്ക്കാം. എന്നാൽ ഉപയോഗിക്കുവാനുള്ള അവകാശങ്ങൾ ക്രമീകരിച്ചുകൊണ്ട് തിരഞ്ഞെടുത്ത ഉപയോക്താക്കൾക്ക് നിർദ്ദിഷ്ട വിവരങ്ങൾ മാത്രം ഉപയോഗിക്കുവാൻ കഴിയുന്ന രീതിയിൽ പരിമിതപ്പെടുത്താവുന്നതാണ്. പാസ്സ്‌വേഡിന്റെ സഹായത്താൽ, ഒരു ഡാറ്റാബേസിലെ വിവരങ്ങൾ അംഗീകൃത വ്യക്തികൾക്ക് മാത്രമായി ലഭ്യമാക്കാം.
- **ഡാറ്റ പങ്കുവയ്ക്കൽ:** ഡാറ്റാബേസിൽ സംഭരിച്ചിരിക്കുന്ന ഡാറ്റ നിരവധി ഉപയോക്താക്കൾക്കും പ്രോഗ്രാമുകൾക്കും ഒരേസമയം പങ്കിടാൻ കഴിയും. കൂടാതെ ഓരോരുത്തർക്കും വ്യത്യസ്ത ഉദ്ദേശ്യങ്ങൾക്കായും ഉപയോഗിക്കാം.
- **മാനദണ്ഡങ്ങളുടെ നടപ്പാക്കൽ:** ഡാറ്റാബേസിന്റെ കേന്ദ്ര നിയന്ത്രണം ഉപയോഗിച്ച്, ഒരു ഡാറ്റാബേസ് അഡ്മിനിസ്ട്രേറ്റർ (DBA) ആവശ്യമായ മാനദണ്ഡങ്ങൾ നിർവചിക്കുകയും നടപ്പിലാക്കുകയും ചെയ്യുന്നു. കംപ്യൂട്ടറുകൾ തമ്മിൽ ഡാറ്റ വിനിമയം നടത്തുന്നതിനുള്ള ഡാറ്റ ഫോർമാറ്റുകൾക്ക് മാനദണ്ഡങ്ങൾ നിർവചിക്കാം. നാമകരണ സമ്പ്രദായങ്ങൾ, പ്രദർശന ഫോർമാറ്റുകൾ, റിപ്പോർട്ട് ഘടനകൾ, പദാവലി, ഡോക്യുമെന്റേഷൻ നിലവാരങ്ങൾ, അപ്ഡേറ്റ് നടപടികൾ, വിനിയോഗ നിയമങ്ങൾ തുടങ്ങിയവ പ്രായോഗികമായ മാനദണ്ഡങ്ങളിൽ ഉൾപ്പെടുന്നു. വിവിധ വകുപ്പുകളുടെയും പ്രൊജക്റ്റുകളുടെയും ഉപയോക്താക്കളുടെയും ഇടയിൽ ആശയവിനിമയവും സഹകരണവും ഉണ്ടാക്കുന്നതിന് ഇത് സഹായിക്കുന്നു.
- **തകർച്ചയിൽ നിന്ന് വീണ്ടെടുക്കൽ:** ഒരു സിസ്റ്റം തകരുമ്പോൾ, മുഴുവൻ ഡാറ്റയും അല്ലെങ്കിൽ ഡാറ്റയുടെ ഒരു ഭാഗം ഉപയോഗശൂന്യമാകും. തകർച്ചയിൽ നിന്നും ഡാറ്റ വീണ്ടെടുക്കുന്നതിന് DBMS ചില സംവിധാനങ്ങൾ നൽകുന്നു.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. ഒരേ ഡാറ്റ വിവിധ സ്ഥലങ്ങളിൽ സൂക്ഷിക്കുന്നതിനെ ________ എന്നു പറയുന്നു.
2. ഒരു വിദ്യാർഥിയുടെ വിലാസം സ്കൂൾ റെക്കോർഡിൽ രണ്ടു രീതിയിൽ സൂക്ഷിച്ചിരിക്കുന്നു; ഈ സാഹചര്യം ________ എന്നറിയപ്പെടുന്നു.
3. ഡാറ്റയുടെ അംഗീകൃതമല്ലാത്ത ഉപയോഗം ________ ഉപയോഗിച്ചു കൊണ്ട് തടയുന്നു.
4. ഡാറ്റ പങ്കുവയ്ക്കുന്നതു ഡാറ്റ കൊണ്ട് റിഡൻഡൻസി കുറയ്ക്കും. ഈ പ്രസ്താവന ശരിയോ തെറ്റോ എന്ന് എഴുതുക.
5. ഡാറ്റ റിഡൻഡൻസി ഡാറ്റയുടെ സ്ഥിരത കുറയ്ക്കും. ശരിയോ തെറ്റോ എന്ന് എഴുതുക.

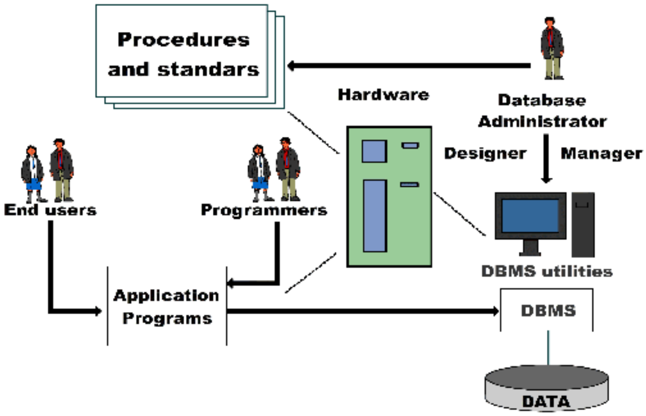
8.2 DBMS ഘടകങ്ങൾ (Components of the DBMS environment)

DBMS ന് ചുവടെ കൊടുത്തിരിക്കുന്ന വിധത്തിൽ നിരവധി ഘടകങ്ങൾ ഉണ്ട്. ഓരോന്നും അതിന്റെ പരിതസ്ഥിതിയിൽ വളരെ പ്രധാനമായ ചുമതലകൾ നിർവഹിക്കുന്നു.

- ഹാർഡ്‌വെയർ
- സോഫ്റ്റ്‌വെയർ
- ഡാറ്റ
- ഉപയോക്താക്കൾ
- നടപടിക്രമങ്ങൾ

ഹാർഡ്‌വെയർ: ഡാറ്റാബേസിന്റെ സംഭരണത്തിനും വീണ്ടെടുക്കലിനും ഉപയോഗിക്കുന്ന കമ്പ്യൂട്ടർ സിസ്റ്റം ആണ് ഹാർഡ് വെയർ. കമ്പ്യൂട്ടറുകൾ (പിസികൾ, വർക്ക്സ്റ്റേഷനുകൾ, സർവറുകൾ, സൂപ്പർകമ്പ്യൂട്ടറുകൾ), സംഭരണ ഉപകരണങ്ങൾ (ഹാർഡ് ഡിസ്കുകൾ, മാഗ്നെറ്റിക് ടേപ്പുകൾ), നെറ്റ്‌വർക്ക് ഉപകരണങ്ങൾ (ഹബ്, സ്വിച്ച്, റൗട്ടറുകൾ, ഫൈബർ ഓപ്റ്റിക്സ്), ഡാറ്റ സൂക്ഷിക്കുവാനുള്ള മറ്റു അനുബന്ധ ഉപകരണങ്ങൾ എന്നിവ ഇതിൽ ഉൾപ്പെടുന്നു.

സോഫ്റ്റ്‌വെയർ: DBMS, ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകൾ, യൂട്ടിലിറ്റികൾ എന്നിവ ഇതിൽ പെടും. ഉപയോക്താവിനും ഡാറ്റാബേസിനും ഇടയിലുള്ള ഒരു പാലമായി DBMS പ്രവർത്തിക്കുന്നു. മറ്റൊരുവിധത്തിൽ പറഞ്ഞാൽ, ഉപയോക്താക്കളുമായും ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകളുമായും ഡാറ്റാബേസുകളുമായും സംവദിക്കുന്ന സോഫ്റ്റ്‌വെയർ ആണ് DBMS. ഡാറ്റാബേസ് ഉപയോഗിക്കുന്ന സമയത്ത് ഉപയോക്താക്കളിൽ നിന്നുള്ള എല്ലാ അഭ്യർഥനകളും DBMS കൈകാര്യം ചെയ്യുന്നു. ഡാറ്റാനിർവചനം (data definition), ഡാറ്റ കൈകാര്യം ചെയ്യൽ (data manipulation), ഡാറ്റാ സുരക്ഷിതത്വം (data securtiy), ഡാറ്റ വിശ്വാസ്യത (data integrtiy), ഡാറ്റ വീണ്ടെടുക്കൽ (data recovery), പ്രകടനം മെച്ചപ്പെടുത്തൽ (performance optimization) തുടങ്ങിയ നിരവധി ചുമതലകൾ കൈകാര്യം ചെയ്യുന്ന നിരവധി സോഫ്റ്റ്‌വെയർ ഘടകങ്ങൾ DBMS സോഫ്റ്റ്‌വെയറിൽ ഉൾപ്പെടുന്നു. സങ്കീർണമായ ഹാർഡ്‌വെയർ വിശദാംശങ്ങളിൽ നിന്ന് ഡാറ്റാബേസിന് കവചം തീർക്കുന്ന ഒരു പൊതു സംവിധാനമാണ് DBMS. ഇത് ഡാറ്റയുടെ ഉപയോഗം നിയന്ത്രിക്കുകയും ഡാറ്റയുടെ സറിരത നിലനിർത്താൻ സഹായിക്കുകയും ചെയ്യുന്നു. റിപ്പോർട്ടുകൾ തയാറാക്കുക, തീരുമാനങ്ങൾ എടുക്കാൻ ആവശ്യമായ വിവരങ്ങൾ നൽകുക, ഡാറ്റാസമാഹരണം

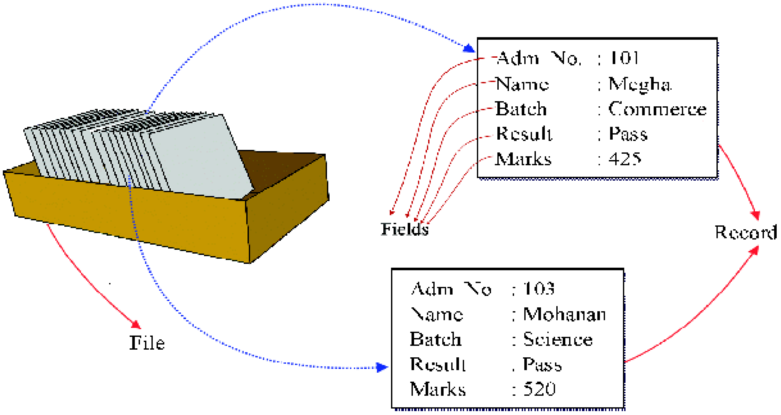


ചിത്രം 8.1: ഡാറ്റാബേസ് സിസ്റ്റം

നടത്തുക എന്നിവയ്ക്കായി ഡാറ്റാബേസിൽ ലഭ്യമായ ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകൾ ഉപയോഗിക്കുന്നു. ഡാറ്റാബേസ് സംവിധാനം കൈകാര്യം ചെയ്യുവാൻ സഹായിക്കുന്ന സോഫ്റ്റ്‌വെയറുകളാണ് യൂട്ടിലിറ്റികൾ. ഉദാഹരണത്തിന്, ഡാറ്റാബേസ് ഘടനകൾ നിർമ്മിക്കുക, ഡാറ്റാബേസ് ഉപയോഗം നിയന്ത്രിക്കുക, ഡാറ്റാബേസ് പ്രവർത്തനങ്ങൾ നിരീക്ഷിക്കുക തുടങ്ങിയവയ്ക്കായി എല്ലാ പ്രധാന DBMS കളും ഗ്രാഫിക്കൽ യൂസർ ഇന്റർഫേസുകൾ (GUIs) ഉപയോഗിക്കുന്നു.

ഡാറ്റ: അവസാനഘട്ട ഉപയോക്താക്കളുടെ കാഴ്ചപ്പാടിൽ DBMS പരിസ്ഥിതിയിലെ/ ചട്ടക്കൂടിലെ ഏറ്റവും പ്രധാനപ്പെട്ട ഘടകം ഡാറ്റ തന്നെയാണ്. ഡാറ്റാബേസിൽ പ്രവർത്തനഡാറ്റയും മെറ്റാഡാറ്റയും (ഡാറ്റയെക്കുറിച്ചുള്ള ഡാറ്റ) അടങ്ങിയിരിക്കുന്നു. ഒരു സ്ഥാപനത്തിന് ആവശ്യമായ എല്ലാ ഡാറ്റയും ഡാറ്റാബേസിൽ ഉണ്ടായിരിക്കണം. യഥാർഥ ഡാറ്റയും ഡാറ്റ ഉപയോഗിക്കുന്ന പ്രോഗ്രാമുകളും പരസ്പരം വേർതിരിക്കപ്പെടുന്നു എന്നത് ഡേറ്റാബേസുകളുടെ പ്രധാന സവിശേഷതയാണ്. വിവരങ്ങൾ ഫലപ്രദമായി സംഭരിക്കുന്നതിനും വീണ്ടെടുക്കുന്നതിനും ഡാറ്റയെ ഫീൽഡുകൾ, റെക്കോർഡുകൾ, ഫയലുകൾ എന്നിങ്ങനെ ക്രമീകരിച്ചിരിക്കുന്നു.

ഒരു ക്ലാസിലെ വിദ്യാർഥികളുടെ അഡ്മിഷൻ നമ്പർ, പേര്, ബാച്ച്, പരീക്ഷാഫലം, മാർക്കുകൾ എന്നിവ ശേഖരിക്കുന്നതിനുള്ള കാർഡുകളുടെ ഒരു ശേഖരം സങ്കൽപ്പിക്കുക. ഓരോ കാർഡിനും സമാനഘടന ഉണ്ടായിരിക്കും, എന്നാൽ അവയിൽ രേഖപ്പെടുത്തിയിരിക്കുന്ന ഡാറ്റ ചിത്രം 8.2 ലെ പോലെ വ്യത്യസ്തമായിരിക്കും.



ചിത്രം 8.2: ഡാറ്റ സംസ്ഥാപന ആശയം

ഫീൽഡുകൾ: ശേഖരിച്ച ഡാറ്റയുടെ ഏറ്റവും ചെറിയ ഘടകമാണ് ഒരു ഫീൽഡ്. ഓരോ ഫീൽഡിലും ഒരു പ്രത്യേക തരം ഡാറ്റ അടങ്ങിയിരിക്കുന്നു. ചിത്രം 8.2 AdmNo, Name, Batch, Result, Marks എന്നിവ ഫീൽഡുകളാണ്.

റെക്കോർഡ്: അനുബന്ധ ഫീൽഡുകളുടെ ശേഖരമാണ് ഒരു റെക്കോർഡ്. മുകളിലുള്ള ചിത്രത്തിൽ ബോക്സിലെ ഓരോ കാർഡിലും ബന്ധപ്പെട്ട എല്ലാ ഫീൽഡുകളും ഉൾക്കൊള്ളുന്നു. ഉദാഹരണത്തിന്, 103, മോഹനൻ, സയൻസ്, പാസ്, 520 എന്നീ ഫീൽഡുകൾ ഉൾക്കൊള്ളുന്നതാണ് ഒരു റെക്കോർഡ്.

ഫയൽ: ഒരു ഫയൽ എന്നത് ഒരേ തരത്തിലുള്ള റെക്കോർഡുകളുടെ ഒരു ശേഖരമാണ്. ചിത്രം 8.2ൽ കാണിച്ചിരിക്കുന്ന ബോക്സ് ഒരു ഫയൽ ആയി കണക്കാക്കാം.

ഉപയോക്താക്കൾ: ഡാറ്റാബേസിലെ ഡാറ്റ ഉപയോഗിക്കുന്ന നിരവധി ഉപയോക്താക്കൾ ഉണ്ട്. DBMS ഉപയോഗിക്കുന്നവരുടെ ആശയവിനിമയത്തിന്റെ രീതി അനുസരിച്ച് ഒരു ഡാറ്റാബേസ് സംവിധാനത്തിന്റെ ഉപയോക്താക്കളെ തരം തിരിക്കാം. ഡാറ്റാബേസ് അഡ്മിനിസ്ട്രേറ്റർ (DBA), ആപ്ലിക്കേഷൻ പ്രോഗ്രാമർമാർ, സാങ്കേതിക പരിജ്ഞാനമുള്ള ഉപയോക്താക്കൾ (Sophisticated users), സാധാരണ ഉപയോക്താക്കൾ (Naive Users) എന്നിവയാണ് വിവിധ വിഭാഗങ്ങളിലെ ഉപയോക്താക്കൾ.

നടപടിക്രമങ്ങൾ: ഡാറ്റാബേസിന്റെ രൂപകല്പനയും ഉപയോഗവും നിയന്ത്രിക്കുന്ന നിർദേശങ്ങളും ചട്ടങ്ങളുമാണ് നടപടിക്രമങ്ങൾ. സംവിധാനത്തിന്റെ ഉപയോക്താക്കൾക്കും ഡാറ്റാബേസ് കൈകാര്യം ചെയ്യുന്ന വ്യക്തികൾക്കും സിസ്റ്റം എങ്ങനെ ഉപയോഗിക്കണം അല്ലെങ്കിൽ പ്രവർത്തിപ്പിക്കണമെന്ന് രേഖപ്പെടുത്തിയ പ്രമാണങ്ങൾ ആവശ്യമാണ്. ചുവടെ കൊടുത്തിരിക്കുന്ന പ്രവർത്തനങ്ങൾക്കാവശ്യമായ നിർദേശങ്ങൾ ഇത്തരം രേഖകളിൽ ഉണ്ടായിരിക്കും.

i) DBMS ലേക്ക് ലോഗിൻ ചെയ്യുക.

ii) ഒരു നിശ്ചിത DBMS സംവിധാനം അല്ലെങ്കിൽ ആപ്ലിക്കേഷൻ പ്രോഗ്രാം ഉപയോഗിക്കുക.

iii) DBMS പ്രവർത്തനം ആരംഭിക്കുകയും അവസാനിപ്പിക്കുകയും ചെയ്യുക.

iv) ഡാറ്റാബേസിന്റെ പകർപ്പ് എടുക്കുകയും ഹാർഡ്‌വെയറിന്റെയോ സോഫ്റ്റ്‌വെയറിന്റെയോ തകരാറുകൾ പരിഹരിക്കുകയും ചെയ്യുക.

v) വിവിധ ഡിസ്കുകളിലായി വ്യാപിച്ചിരിക്കുന്ന ഡാറ്റാബേസ് തിരിച്ചറിയുകയും പ്രവർത്തനം മെച്ചപ്പെടുത്തുകയും ദ്വിതീയ സംഭരണിയിലേക്ക് ചരിത്രരേഖയായി ഡാറ്റ സൂക്ഷിക്കുകയും ചെയ്യുക.

   രണ്ടാം സംഭരണിയിലേക്ക് ആർക്കൈവ് ഡാറ്റയായി (Archive Daa) സൂക്ഷിക്കുകയും ചെയ്യുക.

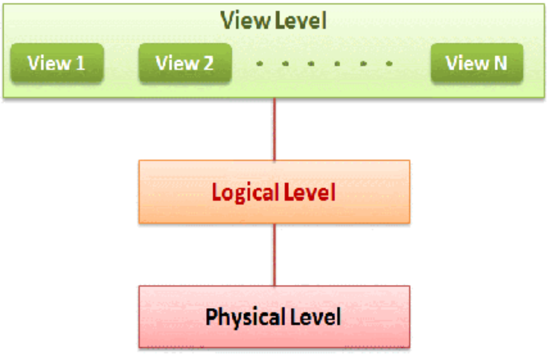
നിങ്ങളുടെ പുരോഗതി അറിയുക

1. ഡാറ്റയെ കുറിച്ചുള്ള ഡാറ്റ __________ എന്നറിയപ്പെടുന്നു.
2. ഡാറ്റാബേസ് സംവിധാനത്തിന്റെ പ്രധാന ഘടകങ്ങളുടെ പേര് എഴുതുക?
3. താഴെ പറയുന്ന DBMS ന്റെ ഘടകങ്ങളെ തരം തിരിക്കുക:

   ഹാർഡ് ഡിസ്ക്, സ്വിച്ച് , ഡിബിഎ, പേറോൾ സംവിധാനം, സാധാരണ ഉപയോക്താവ്, റെയിൽവേ റിസർവേഷൻ സംവിധാനം.

8.3 ഡാറ്റ സംഗ്രഹവും ഡാറ്റ സ്വാതന്ത്ര്യവും
(Data abstraction and data independence)

ഒരു സിസ്റ്റം ഉപയോഗയോഗ്യമാക്കുവാൻ ഡാറ്റ കാര്യക്ഷമമായി വീണ്ടെടുക്കണം. കാര്യക്ഷമതയോടെ ഡാറ്റയെ ഡാറ്റാബേസിൽ പ്രതിനിധാനം ചെയ്യുന്നതിനായി ഡിസൈനർ (designer) സങ്കീർണമായ ഡാറ്റാഘടനകൾ ഉപയോഗപ്പെടുത്തുന്നു. ഡാറ്റാബേസ് ഉപയോക്താക്കളിലധികവും കമ്പ്യൂട്ടർ പരിജ്ഞാനം ഇല്ലാത്തവരായതിനാൽ, കമ്പ്യൂട്ടർ പ്രോഗ്രാമുകൾ വികസിപ്പിക്കുന്നവർ ഡാറ്റാബേസ് സങ്കീർണത പലതലത്തിലുള്ള ഡാറ്റ സംഗ്രഹം വഴി ഉപയോക്താക്കളിൽ നിന്ന് മറയ്ക്കുന്നു.



ചിത്രം 8.3: ഡാറ്റാ സംഗ്രഹത്തിന്റെ തലങ്ങൾ

ചിത്രം 8.3 ൽ വിവരിച്ചിരിക്കുന്നതുപോലെ, ഒരു DBMS ലെ ഡാറ്റസംഗ്രഹം മൂന്നുതലങ്ങളിലായി വിവരിച്ചിരിക്കുന്നു ഭൗതിക തലം, ലോജിക്കൽ തലം, വ്യൂ തലം.

a. ഭൗതിക തലം (Physical level)

ഡിസ്കുകൾ, ടേപ്പുകൾ മുതലായ രണ്ടാംതര സംഭരണ ഉപകരണങ്ങളിൽ എങ്ങനെയാണ് ഡാറ്റ സൂക്ഷിച്ചിരിക്കുന്നത് എന്ന് ഏറ്റവും താഴ്ന്ന തലത്തിലുള്ള ഡാറ്റ സംഗ്രഹം സൂചിപ്പിക്കുന്നു. ഫിസിക്കൽ തലം താഴ്ന്ന തലത്തിലുള്ള സങ്കീർണ ഡാറ്റാഘടന വിശദീകരിക്കുന്നു. റിലേഷനുകൾ (പട്ടിക രൂപത്തിലുള്ള ഡാറ്റ) സൂക്ഷിക്കുന്നതിനും വേഗത്തിൽ ഡാറ്റ വീണ്ടെടുക്കുന്നതിനായി ഇൻഡക്സുകൾ എന്നറിയപ്പെടുന്ന സഹായക ഡാറ്റ ഘടന നിർമ്മിക്കുന്നതിനും ഏത് ഫയൽ വ്യവസ്ഥ ഉപയോഗിക്കണമെന്ന് നാം തീരുമാനിക്കേണ്ടതുണ്ട്. SWS (ഏകജാലക സംവിധാനം) ഡാറ്റാബേസിലെ ഭൗതിക ഡാറ്റ ഘടനയുടെ ഒരു മാതൃക താഴെക്കൊടുക്കുന്നു.

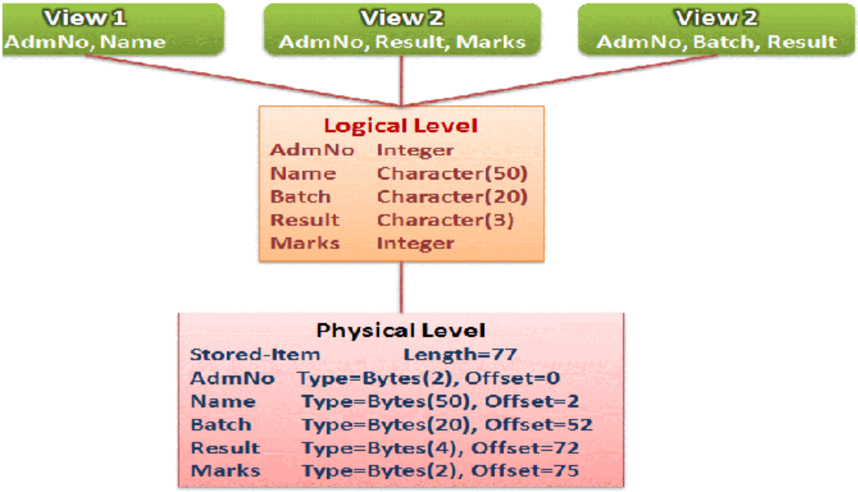
- എല്ലാ റിലേഷനുകളും ക്രമമല്ലാത്ത റെക്കോർഡുകൾ ഉള്ള ഫയലുകളായി സംഭരിക്കുക. (DBMSലെ ഒരു ഫയൽ ഡാറ്റയുടെയോ റെക്കോർഡുകളുടെയോ ശേഖരമാണ്):
- സ്റ്റുഡന്റ്, സ്കൂൾ, കോഴ്സ് എന്നീ ഫയലുകളിൽ ആദ്യ കോളത്തിനെ അടിസ്ഥാനമാക്കി ഇൻഡക്സുകൾ സൃഷ്ടിക്കുക.

b. ലോജിക്കൽ തലം (Logical level)

ഡാറ്റ സംഗ്രഹത്തിന്റെ അടുത്ത ഉയർന്ന തലമായ ലോജിക്കൽ തലം ഡാറ്റാബേസിൽ എന്ത് ഡാറ്റയാണ് സംഭരിക്കുന്നതെന്നും, ഡാറ്റകൾ തമ്മിൽ എന്ത് ബന്ധമാണ് നിലനിൽക്കുന്നതെന്നും വിശദമാക്കുന്നു. താരതമ്യേന വളരെ ലളിതമായ ഘടനകളെ അടിസ്ഥാനമാക്കിയാണ് ഡാറ്റാബേസിലെ ലോജിക്കൽ തലം വിശദീകരിക്കുന്നത്. ലോജിക്കൽ തലത്തിൽ ലളിതമായ ഘടനകൾ നടപ്പിലാക്കുന്നതിന്, ഭൗതിക തലത്തിൽ സങ്കീർണമായ ഘടനകൾ ഉപയോഗിക്കുന്നു. ലോജിക്കൽതലത്തിലുള്ള ഉപയോക്താവിന് ഈ സങ്കീർണത അറിഞ്ഞിരിക്കേണ്ട ആവശ്യമില്ല. ഡാറ്റാബേസിൽ എന്ത് വിവരമാണ് സൂക്ഷിക്കേണ്ടത് എന്ന് തീരുമാനിക്കുന്ന ഡാറ്റാബേസ് അഡ്മിനിസ്ട്രേറ്റർമാർ ഡാറ്റ സംഗ്രഹത്തിന്റെ ലോജിക്കൽ തലമാണ് ഉപയോഗിക്കുന്നത്. ലോജിക്കൽ ഡാറ്റാതലത്തെ കൺസെപ്ചൽ ഡാറ്റാതലം എന്നും പറയുന്നു.

c. വ്യൂ തലം (View level)

ഡാറ്റാബേസ് സംഗ്രഹത്തിന്റെ ഏറ്റവും ഉയർന്ന തലമായ വ്യൂ തലം, ഉപയോക്താക്കളോട് ഏറ്റവും അടുത്ത് നിൽക്കുന്നു. ഓരോ ഉപയോക്താക്കളും ഡാറ്റ കാണുന്ന രീതിയാണ് വ്യൂതലം. ഈ തലത്തിൽ ഡാറ്റാബേസിന്റെ ഒരു ഭാഗം മാത്രമേ വിവരിക്കുന്നുള്ളൂ. ഡാറ്റാബേസിലെ മിക്ക ഉപയോക്താക്കൾക്കും ഡാറ്റാബേസിൽ അടങ്ങിയിരിക്കുന്ന എല്ലാ വിവരങ്ങളും ആവശ്യമില്ല. പകരം അവർക്ക് ഡാറ്റാബേസിന്റെ അവശ്യമായ ഭാഗം മാത്രമേ ഉപയോഗിക്കേണ്ടതുള്ളൂ. ഇത് DBMSമായുള്ള അവരുടെ ഇടപെടൽ ലളിതമാക്കുന്നു. ഒരു ഡാറ്റാബേസിനായി നിരവധി വ്യൂ തലങ്ങൾ ഉണ്ടായേക്കാം. ചിത്രം 8.4ൽ, AdmNo, Name, Batch, Result, Marks എന്നീ ഫീൽഡുകളുള്ള STUDENT ഫയലിന്റെ ഡാറ്റാബേസ് സംഗ്രഹത്തിന്റെ മൂന്നു തലങ്ങൾ കാണിക്കുന്നു.



ചിത്രം 8.4: ഡാറ്റ സംഗ്രഹത്തിന്റെ തലങ്ങളുടെ ഉദാഹരണം

8.3.1 ഡാറ്റാസ്വാതന്ത്ര്യം (Data independence)

ഒരു ഡാറ്റാബേസിനെ അതിന്റെ മൂന്ന്തലത്തിലുള്ള ഡാറ്റാസംഗ്രഹം വഴി വീക്ഷിക്കുമ്പോൾ, ഒരു നിശ്ചിത തലത്തിലെ ഡാറ്റാബേസ് ഘടനയിൽ എന്തെങ്കിലും മാറ്റമുണ്ടാകുന്നത് മറ്റ് തലങ്ങളുടെ ഘടനയെ ബാധിച്ചേക്കാം. ഡാറ്റാബേസിൽ നടത്തുന്ന നിരന്തരമായ മാറ്റങ്ങൾ ഡാറ്റാബേസിന്റെ പുനക്രമീകരണത്തിന് കാരണമാകരുത്. തൊട്ടടുത്ത തലത്തിലെ ഡാറ്റാഘടനയുടെ നിർവചനത്തെ ബാധിക്കാതെ ഒരു തലത്തിലെ ഡാറ്റാഘടനയിൽ മാറ്റം വരുത്തുവാനുള്ള കഴിവിനെ ഡാറ്റാസ്വാതന്ത്ര്യം എന്ന് പറയുന്നു. ഡാറ്റാസ്വാതന്ത്ര്യത്തിന് രണ്ടു തലങ്ങളുണ്ട് - ഭൗതിക ഡാറ്റാസ്വാതന്ത്ര്യവും (physical data independence) ലോജിക്കൽ ഡാറ്റാസ്വാതന്ത്ര്യവും (logical data independence).

a. ഭൗതിക ഡാറ്റാസ്വാതന്ത്ര്യം (Physical data independence)

ലോജിക്കൽ തലത്തിലെ ഡാറ്റാഘടനയെ ബാധിക്കാതെ ഭൗതികതലത്തിലെ ഡാറ്റാഘടനയിൽ മാറ്റം വരുത്തുവാനുള്ള കഴിവിനെ ഭൗതിക ഡാറ്റാസ്വാതന്ത്ര്യം എന്നു പറയുന്നു. അതായത് ഭൗതിക തലത്തിലെ ഡാറ്റാഘടനയിൽ മാറ്റം വരുത്തിയാലും ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകൾ മാറ്റമില്ലാതെ ഒരേപോലെ തന്നെ തുടരും.

b. ലോജിക്കൽ ഡാറ്റാസ്വാതന്ത്ര്യം (Logical data independence)

ബാഹ്യതലത്തിലെ (വ്യൂ തലം) ഡാറ്റാഘടനയെ ബാധിക്കാതെ ലോജിക്കൽ തലത്തിലെ ഡാറ്റാഘടനയിൽ മാറ്റം വരുത്താനുള്ള കഴിവിനെ ലോജിക്കൽ ഡാറ്റ സ്വാതന്ത്ര്യം എന്നു പറയുന്നു. ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകളുടെ തൽസ്ഥിതി ഉറപ്പാക്കുന്നത് ലോജിക്കൽ ഡാറ്റാസ്വാതന്ത്ര്യമാണ്. ആപ്ളിക്കേഷൻ പ്രോഗ്രാമുകൾ ഡാറ്റാബേസിന്റെ ലോജിക്കൽ ഘടനയെ കൂടുതൽ ആശ്രയിച്ചിരിക്കുന്നതു കൊണ്ട്, ഭൗതിക ഡാറ്റാസ്വാതന്ത്ര്യത്തെക്കാൾ ലോജിക്കൽ ഡാറ്റാസ്വാതന്ത്ര്യം കൈവരിക്കാൻ വളരെ പ്രയാസമാണ്.

8.4 ഡാറ്റാബേസിന്റെ ഉപയോക്താക്കൾ (Users of database)

വൈദഗ്ധ്യത്തിന്റെ തോതനുസരിച്ച് അല്ലെങ്കിൽ DBMS കൈകാര്യം ചെയ്യുന്ന രീതിയുടെ അടിസ്ഥാനത്തിൽ ഒരു ഡാറ്റാബേസ് സംവിധാനത്തിന്റെ ഉപയോക്താക്കളെ ചുവടെ പ്പറയുന്ന ഗ്രൂപ്പുകളിൽ ഉൾപ്പെടുത്താം:

- ഡാറ്റാബേസ് അഡ്മിനിസ്ട്രേറ്റർ (DBA)
- ആപ്ലിക്കേഷൻ പ്രോഗ്രാമർമാർ
- സാങ്കേതിക പരിജ്ഞാനമുള്ള ഉപയോക്താക്കൾ
- സാധാരണ ഉപയോക്താക്കൾ

8.4.1 ഡാറ്റാബേസ് അഡ്മിനിസ്ട്രേറ്റർ (DBA)

കേന്ദ്രീകൃതവും പങ്കുവയ്ക്കപ്പെട്ടതുമായ ഡാറ്റാബേസിന്റെ നിയന്ത്രണച്ചുമതലയുള്ള വ്യക്തിയാണ് ഡാറ്റാബേസ് അഡ്മിനിസ്ട്രേറ്റർ (DBA). DBA പല സുപ്രധാന ജോലികളും നിർവഹിക്കേണ്ടതാണ്.

കൺസെപ്ചൽ, ഫിസിക്കൽ സ്കീമകളുടെ രൂപകല്പന: DBMS ൽ സംഭരിക്കേണ്ട ഡാറ്റ എന്താണെന്നും അത് എങ്ങനെ ഉപയോഗപ്പെടുത്താം എന്നും മനസ്സിലാക്കാൻ സംവിധാനത്തിന്റെ ഉപയോക്താക്കളുമായി ആശയവിനിമയം നടത്തുന്നത് DBA യാണ്. ഈ അറിവ് അടിസ്ഥാനമാക്കി, DBA കൺസെപ്ചൽ സ്കീമയും ഫിസിക്കൽ സ്കീമയും രൂപപ്പെടുത്തണം.

സുരക്ഷയും അംഗീകാരവും: ഡാറ്റയുടെ ശരിയായ ഉപയോഗം ഉറപ്പാക്കുന്നതിനുള്ള ഉത്തരവാദിത്വം DBAൽ നിക്ഷിപ്തമാണ്. ഉദാഹരണമായി, ഒരു സ്കൂളിൽ, അധ്യാപകർ വിദ്യാർഥികൾക്ക് പഠന മേഖലകൾ പരിചയപ്പെടാനും, വിദ്യാർഥിയുടെ പരീക്ഷാ ഫലങ്ങൾ അറിയാനും, ഒരു വിഷയം പഠിപ്പിക്കുന്ന അധ്യാപകരുടെ വിശദാംശങ്ങൾ കണ്ടെത്തുവാനും അനുവാദം നൽകുന്നു. അതേസമയം വിദ്യാർഥികൾക്ക് അധ്യാപകരുടെ ശമ്പളവിവരങ്ങളോ അല്ലെങ്കിൽ മറ്റ് വിദ്യാർഥികളുടെ ഗ്രേഡ് വിവരങ്ങളോ കാണാൻ അനുവാദമില്ല. പഠനമേഖലകളെ കുറിച്ചുള്ള ഒരു വ്യൂ മാത്രം ഉപയോഗിക്കാൻ വിദ്യാർഥികൾക്ക് അനുമതി നൽകിക്കൊണ്ട് DBA യ്ക്ക് ഈ നയം നടപ്പിലാക്കാൻ കഴിയും.

ഡാറ്റാലഭ്യതയും നഷ്ടത്തിൽ നിന്നുള്ള വീണ്ടെടുക്കലും: ഒരു ഇടപാട് പൂർത്തിയാക്കുന്നതിനോ തകരാർ പരിഹരിക്കുന്നതിനോ സംവിധാനം പരാജയപ്പെടുമ്പോൾ ഡാറ്റ ഒരു സ്ഥിരാവസ്ഥയിലേക്ക് പുനഃസ്ഥാപിക്കുന്നതിനായി DBA നടപടികൾ എടുക്കേണ്ടതാണ്. ഈ പ്രവർത്തനങ്ങൾക്കുള്ള സോഫ്റ്റ്വെയർ പിന്തുണ DBMS നൽകുന്നുണ്ട്, പക്ഷേ, അതാതു സമയം ഡാറ്റ ബാക്കപ്പ് ചെയ്യുവാനും സിസ്റ്റം പ്രവർത്തനത്തിന്റെ ലോഗുകൾ

(ഡാറ്റാബേസിലെ പ്രവർത്തനങ്ങളായ ഡാറ്റ കൂട്ടിച്ചേർക്കൽ, ഒഴിവാക്കൽ, പരിഷ്കരിക്കൽ, തുടങ്ങിയ പ്രക്രിയകൾ സൂക്ഷിക്കുന്ന ഫയലുകൾ) പരിപാലിക്കാനും ആവശ്യമായ നടപടിക്രമങ്ങൾ സ്വീകരിക്കേണ്ടത് DBA ആണ്.

8.4.2 ആപ്ലിക്കേഷൻ പ്രോഗ്രാമർമാർ (Application programmers)

ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകൾ വഴി DBMS മായി സംവദിക്കുന്ന കമ്പ്യൂട്ടർ പ്രൊഫഷണലുകളാണ് ആപ്ലിക്കേഷൻ പ്രോഗ്രാമർമാർ. വിഷ്വൽ ബേസിക്, സി, സി++, ജാവാ പോലുള്ള കമ്പ്യൂട്ടർ ഭാഷകൾ ഉപയോഗിച്ച് എഴുതുകയും ഡാറ്റ കൈകാര്യം ചെയ്യുന്ന ഭാഷ (DML) യിലൂടെ DBMS മായി സംവദിക്കുകയും ചെയ്യുന്ന പ്രോഗ്രാമുകളാണ് ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകൾ. ബാഹ്യസ്കീമയിലൂടെ ഡാറ്റയെ മികച്ച രീതിയിൽ ആപ്ലിക്കേഷൻ പ്രോഗ്രാമുകൾ ഉപയോഗിക്കുന്നു

8.4.3 സാങ്കേതിക പരിജ്ഞാനമുള്ള ഉപയോക്താക്കൾ (Sophisticated users)

എൻജിനീയർമാർ, ശാസ്ത്രജ്ഞർ, ബിസിനസ്സ് വിശകലന വിദഗ്ധർ, DBMS സൗകര്യങ്ങൾ നന്നായി പരിചയമുള്ളവർ എന്നിവരാണ് ഈ വിഭാഗത്തിൽപ്പെടുന്നത്. അവരുടെ സങ്കീർണമായ ആവശ്യകതകൾ നിറവേറ്റുന്നതിനായി സ്വന്തം ക്വറികളിലൂടെ (ഡാറ്റാബേസ് അഭ്യർഥന) അവർ DBMS-മായി സംവദിക്കുന്നു.

8.4.4 സാധാരണ ഉപയോക്താക്കൾ (Naive users)

മുമ്പേ എഴുതി തയാറാക്കിയ പ്രോഗ്രാമുകളിൽ കൂടി DBMS സംവിധാനവുമായി സംവദിക്കുന്നവരാണ് സാധാരണ ഉപയോക്താക്കൾ. DBMSന്റെ വിശദാംശങ്ങൾ അവർ അറിയുകയോ ശ്രദ്ധിക്കുകയോ ഇല്ല. സാധാരണ ഉപയോക്താക്കൾ ഉയർന്നതലത്തിലുള്ള ഡാറ്റ സംഗ്രഹം മാത്രം കൈകാര്യം ചെയ്യുന്നു. വെബിലൂടെ ഡാറ്റ ഉപയോഗിക്കുന്ന ആളുകൾ, ഒരു കാര്യാലയത്തിലെ ക്ളറിക്കൽ ജീവനക്കാർ, ഒരു സൂപ്പർമാർക്കറ്റിലെ അല്ലെങ്കിൽ ഹോട്ടലുകളിലെ ബില്ലിങ് ക്ലർക്ക്, ബാങ്ക് ക്ലാർക്ക് മുതലായവർ സാധാരണ ഉപയോക്താക്കളാണ്.

നിങ്ങളുടെ പുരോഗതി അറിയുക

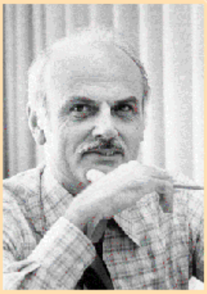
1. ചോദ്യോത്തര രൂപത്തിൽ ഡാറ്റാബേസുമായി സംവദിക്കുന്ന ഉപയോക്താവ് __________ എന്നറിയപ്പെടുന്നു.
2. ഒരു സൂപ്പർമാർക്കറ്റിലെ ബില്ലിംഗ് ക്ലാർക്ക് ഏതുതരം ഉപയോക്താവാണ്?
3. ഒരു ഡാറ്റാബേസിന് ഡാറ്റാസുരക്ഷ നൽകുന്ന ഉപയോക്താവാരാണ്?
4. ഒരു ഡാറ്റാബേസിന്റെ ഘടനയെ മാറ്റുന്ന ഉപയോക്താവാരാണ് ?
5. ആപ്ളിക്കേൻ പ്രോഗ്രാം വഴി ഡാറ്റാബേസുമായി സംവദിക്കുന്ന ഉപയോക്താവ് __________ എന്നറിയപ്പെടുന്നു.

8.5 റിലേഷണൽ ഡാറ്റാമോഡൽ (Relational data model)

വ്യത്യസ്ത പേരുകളാൽ തിരിച്ചറിയാവുന്നതും റിലേഷനുകൾ എന്നറിയപ്പെടുന്നതുമായ ഒരു കൂട്ടം പട്ടികകളുടെ ശേഖരമായി ഡാറ്റാ ബേസിനെ പ്രതിനിധാനം ചെയ്യുന്ന

മാതൃകയാണ് റിലേഷണൽ ഡാറ്റാമോഡൽ റിലേഷണൽ ഡാറ്റാമോഡലിൽ ഡാറ്റയും അവ തമ്മിലുള്ള ബന്ധങ്ങളും പട്ടിക രൂപത്തിൽ അവതരിപ്പിക്കുവാൻ സാധിക്കും. ഈ അവതരണരീതി ഒരു ഡാറ്റാബേസിന്റെ ആശയങ്ങൾ എളുപ്പത്തിൽ മനസ്സിലാക്കാൻ സഹായിക്കുന്നു.

എഡ്ഗാർ ഫ്രാങ്ക് കോഡ് (19 ഓഗസ്റ്റ് 1923 - 18 ഏപ്രിൽ 2003) എന്ന കമ്പ്യൂട്ടർ ശാസ്ത്രജ്ഞനാണ് ഡാറ്റാ മാനേജ്മെന്റിനായുള്ള റിലേഷണൽ മോഡൽ കണ്ടുപിടിച്ചത്. ഇംഗ്ലണ്ടിലെ പോർട്ട്ലാൻഡ് ദ്വീപിൽ അദ്ദേഹം ജനിച്ചു. രണ്ടാം ലോകമഹായുദ്ധകാലത്ത് റോയൽ എയർ ഫോഴ്സിൽ പൈലറ്റ് ആയി സേവനമനുഷ്ഠിച്ചു. 1948 ൽ അദ്ദേഹം ഐ.ബി.എം.ൽ ചേർന്നു. 1981 ൽ അദ്ദേഹം ടൂറിംഗ് അവാർഡ് കരസ്ഥമാക്കി. 2003 ഏപ്രിൽ 18ന് 79-ാം വയസ്സിൽ ഫ്ളോറിഡയിലുള്ള വില്യംസ് ദ്വീപിലെ തന്റെ ഭവനത്തിൽ വച്ച് ഹൃദയാഘാതത്തെ തുടർന്ന് അദ്ദേഹം മരണമടഞ്ഞു.



ഇന്ന്, ഡാറ്റാബേസ് ഉൽപ്പന്നങ്ങളിൽ ഭൂരിഭാഗവും റിലേഷണൽ മോഡൽ അടിസ്ഥാനമാക്കിയുള്ളതാണ്. അവ റിലേഷണൽ ഡാറ്റാബേസ് നിർവഹണ സംവിധാനം (RDBMS) എന്നാണ് അറിയപ്പെടുന്നത്. ഡാറ്റാപ്രാതിനിധ്യവും സങ്കീർണമായ അന്വേഷണങ്ങളും ലളിതമായി അവതരിപ്പിക്കാൻ കഴിയുന്നു എന്നതാണ് മറ്റ് ഡാറ്റാമോഡലുകളിൽ നിന്ന് റിലേഷണൽ മോഡലിനെ വ്യത്യസ്തമാക്കുന്നത്. ഒറാക്കിൾ, മൈക്രോസോഫ്റ്റ് SQL സെർവർ, MySQL, DB2, ഇൻഫോമിക്സ്, ഇൻഗ്രസ് എന്നിവയാണ് ജനപ്രിയ RDBMS പാക്കേജുകൾ.

വാണിജ്യപരമായി ഏറ്റവുമധികം ഉപയോഗിക്കുന്ന റിലേഷണൽ മോഡൽ ഡാറ്റാബേസ് സംവിധാനങ്ങളിൽ സ്ട്രക്ചേർഡ് ക്വറി ലാംഗ്വേജ് (SQL), ക്വറി ബൈ എക്സാംബിൾ (QBE) അല്ലെങ്കിൽ ഡാറ്റലോഗ് എന്നിവ ഉൾപ്പെടുന്ന ഒരു ചോദ്യഭാഷ അടങ്ങിയിരിക്കുന്നു. വളരെ വ്യാപകമായി ഉപയോഗിക്കുന്ന SQL അടുത്ത അധ്യായത്തിൽ നമ്മൾ പഠിക്കും.

8.6 RDBMS ലെ പദാവലികൾ (Terminologies in RDBMS)

റിലേഷണൽ ഡാറ്റാബേസുകളിലെ പ്രവർത്തനങ്ങൾ ചർച്ച ചെയ്യുന്നതിനു മുൻപ് RDBMS മായി ബന്ധപ്പെട്ട ചില പദാവലികൾ പരിചയപ്പെടാം.

a). എന്റിറ്റി

മറ്റുള്ളവയിൽ നിന്ന് വേർതിരിച്ചറിയാവുന്ന ഒരു വ്യക്തിയെ അല്ലെങ്കിൽ ഒരു വസ്തുവിനെ എന്റിറ്റി എന്ന് പറയുന്നു. ഉദാഹരണത്തിന്, ഓരോ സ്കൂളും ഓരോ വിദ്യാർഥിയും ഓരോ വ്യത്യസ്ത എന്റിറ്റിയാണ്.

b) റിലേഷൻ

ഡാറ്റാശേഖരത്തെ വരികളും നിരകളുമായി ക്രമീകരിക്കുന്നതാണ് റിലേഷൻ. പട്ടിക എന്നും ഇത് അറിയപ്പെടുന്നു. STUDENT എന്ന് പേരുള്ള ഒരു റിലേഷൻ, പട്ടിക 8.1 ൽ കാണിച്ചിരിക്കുന്നു.

STUDENT relation

AdmNo	Roll	Name	Batch	Marks	Result
101	24	Sachin	Science	480	EHS
102	14	Rahul	Commerce	410	EHS
103	4	Fathima	Humanities	200	NHS
104	12	Mahesh	Commerce	180	NHS
105	24	Nelson	Humanities	385	EHS
106	8	Joseph	Commerce	350	EHS
107	24	Shaji	Humanities	205	NHS
108	2	Bincy	Science	300	EHS

പട്ടിക 8.1 STUDENT റിലേഷൻ

c) ടൂപിൾ

ഒരു റിലേഷനിലെ വരികളെ (റെക്കോർഡുകൾ) സാധാരണയായി ടൂപിൾ എന്ന് പറയുന്നു. ഒരു പ്രത്യേക എന്റിറ്റിയെ പ്രതിനിധീകരിക്കാൻ ഉപയോഗിക്കുന്ന മൂല്യങ്ങളുടെ ഒരു പൂർണഗണം ഒരു വരി ഉൾക്കൊള്ളുന്നു. പട്ടിക 8.1 ൽ, STUDENT റിലേഷനിലെ ഓരോ വരിയും ഓരോ വിദ്യാർഥിയുടെ പരീക്ഷാഫലത്തെ പ്രതിനിധീകരിക്കുന്നു.

d) ആട്രിബ്യൂട്ട്

ഒരു റിലേഷനിലെ നിരകളെ ആട്രിബ്യൂട്ടുകൾ എന്ന് വിളിക്കുന്നു. AdmNo, Roll, Name, Batch, Marks, Result എന്നിവ STUDENT റിലേഷന്റെ ആട്രിബ്യൂട്ടുകളാണ്. ഓരോ ആട്രിബ്യൂട്ടിന്റെയും മൂല്യങ്ങൾ ഡൊമെയ്ൻ എന്ന മൂല്യങ്ങളുടെ സാധുത പട്ടികയിൽ നിന്ന് എടുക്കുന്നു.

e) ഡിഗ്രി

ഒരു റിലേഷനിലെ നിരകളുടെ എണ്ണം ആ റിലേഷന്റെ ഡിഗ്രി നിർണയിക്കുന്നു. STUDENT എന്ന റിലേഷന് ആറ് നിരകൾ അല്ലെങ്കിൽ ആട്രിബ്യൂട്ടുകൾ ഉണ്ട്, അതിനാൽ STUDENT റിലേഷന്റെ ഡിഗ്രി ആറ് ആകുന്നു.

f) കാർഡിനാലിറ്റി

ഒരു റിലേഷനിലെ വരികളുടെ അല്ലെങ്കിൽ ടൂപിളുകളുടെ എണ്ണം കാർഡിനാലിറ്റി എന്നറിയപ്പെടുന്നു. STUDENT റിലേഷനിൽ എട്ടു വരികൾ ഉണ്ട്, അതിനാൽ STUDENT റിലേഷന്റെ കാർഡിനാലിറ്റി 8 ആണ്.

g) ഡൊമെയ്ൻ

ഒരു നിരയെ പ്രതിനിധാനം ചെയ്യുന്ന മൂല്യങ്ങളുടെ ഗണമാണ് ഒരു ഡൊമെയ്ൻ. ഉദാഹരണത്തിന്, പട്ടിക 8.1 ൽ കാണിച്ചിരിക്കുന്ന STUDENT ബന്ധത്തിൽ Batch എന്ന നിരയുടെ ഡൊമെയ്ൻ {Science, Humanities, Commerce} എന്ന മൂല്യങ്ങളുടെ ഗണമാണ്. അതായത്, ഈ ഗണത്തിൽ നിന്നുള്ള ഏതെങ്കിലുമൊരു മൂല്യം Batch നിരയിൽ ദൃശ്യമാകാം. അതുപോലെ തന്നെ, Result എന്ന നിരയുടെ ഡൊമെയ്ൻ ആണ് {EHS, NHS}.

h) സ്കീമ

ഒരു ഡാറ്റാബേസ് രൂപകല്പന ചെയ്യുന്ന സന്ദർഭത്തിൽ വ്യക്തമാക്കപ്പെടുന്ന ഡാറ്റാബേസിന്റെ ഘടനയെയോ വിവരണത്തെയോ ഡാറ്റാബേസ് സ്കീമ എന്നുവിളിക്കുന്നു. റിലേഷണൽ മോഡലിൽ, ഒരു റിലേഷന്റെ സ്കീമ അതിന്റെ പേര്, ഓരോ നിരയുടെയും പേര്, ഓരോ നിരയുടെയും ഡാറ്റ തരം എന്നിവ വ്യക്തമാക്കുന്നു. ഉദാഹരണമായി, ഒരു സ്കൂൾ ഡാറ്റാബേസിലെ വിദ്യാർഥികളുടെ വിവരങ്ങൾ ഇനിപ്പറയുന്ന ഘടനയുമായി ബന്ധപ്പെടുത്തി സൂക്ഷിക്കാം:

STUDENT (Admno : integer,
Roll : integer,
Name : character(50),
Batch : character(20),
Marks : decimal,
Result : character(4))

i) ഇൻസ്റ്റൻസ്

റിലേഷണൽ സ്കീമയിൽ നിർവചിച്ചിരിക്കും പ്രകാരം തുല്യയെണ്ണം നിരകളുള്ള ടുപ്പിളുകളുടെ ഒരു കൂട്ടത്തെയാണ് റിലേഷന്റെ ഇൻസ്റ്റൻസ് എന്നു പറയുന്നത്. നിരകളുടെ പേരും ഡാറ്റ തരവും സൂചിപ്പിക്കുന്ന 6 നിരകൾ STUDENT റിലേഷനിലെ ഓരോ വരിക്കും ഉണ്ടെന്ന് മേൽക്കാണുന്ന സ്കീമ വ്യക്തമാക്കുന്നു. STUDENT ബന്ധത്തിന്റെ ഉദാഹരണമാണ് പട്ടിക 8.1 ൽ കാണിച്ചിരിക്കുന്നത്.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. വരികളും നിരകളുമായി ഡാറ്റയെ ക്രമീകരിച്ചിരിക്കുന്നതിനെ ________ എന്ന് പറയുന്നു.
2. ________ ഒരു പട്ടികയിലെ ഒരു പ്രത്യേക എന്റിറ്റിയുടെ പൂർണമായ ഡാറ്റ നൽകുന്നു.
3. ഒരു റിലേഷനിലെ വരികളുടെ എണ്ണം ________ ആണ്.
4. ഒരു റിലേഷനിലെ ________ എണ്ണം റിലേഷന്റെ ഡിഗ്രി എന്നറിയപ്പെടുന്നു
5. റിലേഷണൽ മോഡലുകളിൽ ഡാറ്റ ക്രമീകരിച്ചിരിക്കുന്നത് ________ രീതിയിലാണ്.

8.6.1 കീകൾ (Keys)

ഒരു റിലേഷനെ ടൂപ്പിളുകളുടെ ഒരു ഗണമായി നിർവചിക്കുന്നു. അതായത് എല്ലാ നിരകളുടെയും മൂല്യങ്ങളുടെ സംയോജനം ഒന്നു തന്നെയാകുന്ന രണ്ടു ടൂപ്പിളുകൾ പാടില്ല. അതിനാൽ, ഒരു റിലേഷനിലെ ഓരോ ടൂപ്പിളും തിരിച്ചറിയുവാൻ ഒരു മാർഗം ഉണ്ടായിരിക്കണം. ഇത്തരം വ്യത്യാസങ്ങൾ സൃഷ്ടിക്കാൻ ഒരു കീയുടെ ആശയം നമ്മെ അനുവദിക്കുന്നു. ഒരു റിലേഷനിലെ ഓരോ ടൂപ്പിളും മറ്റ് ടൂപ്പിളുകളിൽ നിന്ന് അനന്യമായി വേർതിരിക്കുന്നതിനുള്ള ഒരു ആട്രിബ്യൂട്ടിന്റെയോ അല്ലെങ്കിൽ ഒരു കൂട്ടം ആട്രിബ്യൂട്ടുകളുടെയോ ശേഖരമാണ് കീ. ഒന്നിലധികം ആട്രിബ്യൂട്ടുകൾ ചേർന്നാണ് ഒരു കീ ഉണ്ടാകുന്നതെങ്കിൽ അതിനെ കോംപോസിറ്റ് (സംയുക്ത) കീ എന്നു പറയുന്നു. ഒരു റിലേഷനിലെ ഓരോ ടൂപ്പിളും വ്യത്യസ്തമായതിനാൽ അതിന്റെ മുഴുവൻ ആട്രിബ്യൂട്ടുകളും പരിഗണിച്ചാൽ അത് ഒരു കീ ആയിരിക്കും. എന്നിരുന്നാലും ചെറിയ കീകൾ ഉണ്ടെങ്കിൽ, പ്രായോഗിക കാരണങ്ങളാൽ, അവയെയാണ് പരിഗണിക്കുന്നത്.

a. കാൻഡിഡേറ്റ് കീ

ഒരു റിലേഷനിലെ ഒരു ട്യൂപ്പിളുകളെ അനന്യമായി തിരിച്ചറിയുന്ന ആട്രിബ്യൂട്ടുകളുടെ ചുരുങ്ങിയ ഗണമാണ് ഒരു കാൻഡിഡേറ്റ് കീ. പട്ടിക 8.1 ലെ STUDENT റിലേഷനിൽ, AdmNo ന് ഓരോ ട്യൂപ്പിളുകളും തിരിച്ചറിയുവാൻ കഴിയും. അതിനാൽ ഇത് ഒരു കാൻഡിഡേറ്റ് കീയായി കണക്കാക്കാം. ഒരു റിലേഷനിൽ ഒന്നിലധികം കാൻഡിഡേറ്റ് കീകൾ ഉണ്ടായിരിക്കാം. കൂടാതെ, ഒരു കാൻഡിഡേറ്റ് കീ ഒരു ആട്രിബ്യൂട്ട് മാത്രം ആകണമെന്നില്ല. അത് ഒരു സംയുക്ത കീയും ആകാം. ഉദാഹരണമായി, Roll, Batch, Year എന്നിവയുടെ സംയോജനവും ഒരു പ്രത്യേക വിദ്യാർഥിയെ തിരിച്ചറിയുവാൻ ഉപയോഗിക്കാം. അതിനാൽ, Roll + Batch + Year എന്നത് STUDENT റിലേഷന്റെ മറ്റൊരു കാൻഡിഡേറ്റ് കീയായി പരിഗണിക്കപ്പെടാം.

b. പ്രൈമറി കീ

ഡാറ്റാബേസ് സൃഷ്ടാവ് പട്ടികയിലെ ഒരു ട്യൂപ്പിൾ തിരിച്ചറിയാനായി തിരഞ്ഞെടുത്തിരിക്കുന്ന കാൻഡിഡേറ്റ് കീകളിൽ ഒന്നാണ് പ്രൈമറി കീ. ഒരു റിലേഷനിലെ വരികളെ തനതായി തിരിച്ചറിയാൻ കഴിയുന്ന ഒന്നോ അതിലധികമോ നിരകളുടെ ഒരു കൂട്ടമാണ് പ്രൈമറി കീ. ഓരോ എൻറിറ്റിയേയും തനതായി തിരിച്ചറിയുന്നതിനാൽ അത് ശൂന്യമോ, പകർപ്പോ ആകാൻ പാടില്ല.

പ്രൈമറി കീ സ്ഥാനത്തേക്കുള്ള സ്ഥാനാർഥികളായി കാൻഡിഡേറ്റ് കീകളെ കണക്കാക്കുന്നു. കാൻഡിഡേറ്റ് കീകളിൽ നിന്ന് ആട്രിബ്യൂട്ടുകളുടെ എണ്ണം ഏറ്റവും കുറവുള്ള സംയുക്തത്തെ പ്രൈമറി കീയായി തിരഞ്ഞെടുക്കാം. പട്ടിക 8.1 ൽ (STUDENT റിലേഷനിൽ) AdmNo പ്രൈമറി കീയായി ഉപയോഗിക്കാം. അതായത്, STUDENT റിലേഷനിലെ രണ്ട് വിദ്യാർഥികൾക്ക് ഒരേ AdmNo ഉണ്ടായിരിക്കില്ല. പട്ടിക 8.1 ൽ, Name ആട്രിബ്യൂട്ടിൽ തനതായ മൂല്യങ്ങൾ നമുക്ക് കാണാൻ സാധിക്കും. എന്നാൽ യഥാർഥ സാഹചര്യത്തിൽ വിദ്യാർഥികളിൽ കൂടുതൽ പേർക്ക് ഒരേ പേര് ഉണ്ടാകും.

c. ആൾട്ടർനേറ്റ് കീ

പ്രൈമറി കീ അല്ലാത്ത ഒരു കാൻഡിഡേറ്റ് കീയെ ആൾട്ടർനേറ്റ് കീ എന്നു വിളിക്കുന്നു. രണ്ടോ അതിലധികമോ കാൻഡിഡേറ്റ് കീകൾ ഉണ്ടെങ്കിൽ, അവയിലൊന്നിനു മാത്രമേ പ്രൈമറി കീയായി സജ്ജമാകാൻ കഴിയൂ. അവശേഷിക്കുന്ന ബാക്കി കീകളാണ് ആൾട്ടർനേറ്റ് കീ. തന്നിരിക്കുന്ന ഉദാഹരണത്തിൽ AdmNo പ്രൈമറി കീ ആയി എടുത്തിരിക്കുന്നതിനാൽ Roll + Batch + Year എന്ന സംയോജനമാണ് ആൾട്ടർനേറ്റ് കീ.

d. ഫോറിൻ കീ

ഒരു റിലേഷനിൽ മറ്റൊരു റിലേഷനിലെ പ്രൈമറി കീയുണ്ടെങ്കിൽ ആ കീയെ ഫോറിൻ കീ എന്നു വിളിക്കാം. രണ്ടോ അതിലധികമോ ടേബിളുകളുമായി ബന്ധപ്പെടുത്തുന്ന തിനായി ഒരു ഫോറിൻ കീ ഉപയോഗിക്കുമ്പോൾ ഇത് റഫറൻസ് കീ എന്നും അറിയ പ്പെടുന്നു. പട്ടിക 8.2 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ Batch name നു പകരം Batch code ഉപയോഗിച്ചിട്ടുണ്ടെന്ന് കരുതുക. പട്ടിക 8.3ൽ കാണിച്ചിരിക്കുന്നതുപോലെ നമുക്ക് BATCH എന്ന ഒരു റിലേഷനുമുണ്ട്. BatchCode എന്നത് BATCH റിലേഷനിലെ പ്രൈമറി കീ ആയിരിക്കുമെന്നത് വ്യക്തമാണ്. എന്നാൽ ഇത് STUDENT പട്ടികയിൽ കീ അല്ലാത്ത നിരയായി ഉപയോഗിക്കുന്നു. അങ്ങനെ, BatchCode എന്നത് STUDENT റിലേഷന്റെ കാര്യത്തിൽ ഒരു ഫോറിൻ കീ ആയി പരാമർശിക്കപ്പെടുന്നു.

STUDENT relation

AdmNo	Roll	Name	BatchCode	Marks	Result
101	24	Sachin	S2	480	EHS
102	14	Rahul	C2	410	EHS
103	4	Fathima	H2	200	NHS
104	12	Mahesh	C2	180	NHS
105	24	Nelson	H2	385	EHS
106	8	Joseph	C2	350	EHS
107	24	Shaji	H2	205	NHS
108	2	Bincy	S2	300	EHS

പട്ടിക 8.2: പരിഷ്കരിച്ച STUDENT റിലേഷൻ

BATCH relation

BatchCode	BatchName	Strength
S1	Science	150
S2	Science	150
C1	Commerce	100
C2	Commerce	100
H1	Humanities	100
H2	Humanities	100

പട്ടിക 8.3: BATCH റിലേഷൻ

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. ഒരു റിലേഷനിൽ ഒരു വരി അദ്വിതീയമായി തിരിച്ചറിയാൻ കഴിയുന്ന ആട്രിബ്യൂട്ട് ________ എന്നറിയപ്പെടുന്നു.
2. ഒരു റിലേഷനിൽ എത്ര പ്രൈമറി കീകൾ സാധ്യമാണ് ?
3. Employee എന്ന പട്ടികയിലെ കാൻഡിഡേറ്റ് കീകളാണ് Emp code, Pan no. Emp code നെ പ്രൈമറി കീയായി നിശ്ചയിച്ചാൽ, Pan no ________ ആയിരിക്കും
4. ഒരു കീയിൽ ഒന്നിൽ കൂടുതൽ ആട്രിബ്യൂട്ടുകളുണ്ടെങ്കിൽ അതിനെ ________ കീയെന്നുപറയുന്നു.
5. രണ്ട് റിലേഷനുകളെ ബന്ധിപ്പിക്കുന്നതിന് ________ കീ ഉപയോഗിക്കുന്നു.

8.7 റിലേഷണൽ ബീജഗണിതം (Relational algebra)

ഒരു ഡാറ്റാബേസിന്റെ ഘടനയും രൂപകല്പനയും നിർണയിക്കുന്നതിന് റിലേഷണൽ മോഡൽ പ്രദാനം ചെയ്യുന്ന ആശയങ്ങളെകുറിച്ച് നമ്മൾ ചർച്ചചെയ്തു. ഡാറ്റാബേസ് രൂപകല്പന ചെയ്ത് ഡാറ്റ സൂക്ഷിച്ചു കഴിഞ്ഞാൽ, ആവശ്യത്തിനനുസരിച്ച് വിവരം തിരിച്ചെടുക്കേണ്ടതാണ്. ഇതിനായി RDBMS വൈവിധ്യമാർന്ന പ്രവർത്തനങ്ങൾ കാഴ്ച വയ്ക്കുന്നു. ഒരു ഡാറ്റാബേസിലെ മുഴുവൻ റിലേഷനുകളെയും കൈകാര്യം ചെയ്യുന്ന പ്രവർത്തനങ്ങളുടെ ശേഖരം റിലേഷണൽ ബീജഗണിതം എന്നറിയപ്പെടുന്നു. ഈ പ്രവർത്തനങ്ങൾ റിലേഷണൽ മോഡലുമായി ബന്ധപ്പെട്ട ഒരു പ്രത്യേകതരം ഭാഷയായ അന്വേഷണ ഭാഷയുടെ (query language) സഹായത്തോടെ നടത്തപ്പെടുന്നു. ഈ പ്രവർത്തനങ്ങൾ നടത്തുന്നതിനുവേണ്ടി അടുത്ത അധ്യായത്തിൽ ആ ഭാഷ നമ്മൾ പഠിക്കും. റിലേഷണൽ ബീജഗണിതത്തിലെ വിവിധ പ്രവർത്തനങ്ങളിൽ ഒന്നോ രണ്ടോ റിലേഷനുകൾ ഇൻപുട്ട് ആയി സ്വീകരിച്ച് പുതിയ ഒരു റിലേഷൻ ഉണ്ടാക്കുന്നു. സെലക്ട് (SELECT), പ്രൊജക്റ്റ് (PROJECT), യൂണിയൻ (UNION), ഇന്റർസെക്ഷൻ (INTERSECTION), സെറ്റ് ഡിഫറൻസ് (SET DIFFERENCE), കാർട്ടീഷ്യൻ പ്രൊഡക്റ്റ് (CARTESIAN PRODUCT) തുടങ്ങിയവയാണ് റിലേഷണൽ ബീജഗണിതത്തിലെ അടിസ്ഥാനപരമായ പ്രവർത്തനങ്ങൾ. SELECT, PROJECT എന്നിവ ഒരു റിലേഷനിൽ മാത്രം പ്രവർത്തിക്കുന്നതിനാൽ അത് യൂനറി പ്രവർത്തനമാണ്. ബാക്കിയുള്ള പ്രവർത്തനങ്ങൾ രണ്ട് റിലേഷനുകളിൽ പ്രവർത്തിക്കുന്നതിനാൽ ബൈനറി പ്രവർത്തനങ്ങളാണ്.

8.7.1 സെലക്ട് പ്രവർത്തനം (SELECT operation)

നൽകിയിരിക്കുന്ന ഒരു ഉപാധി അനുസരിക്കുന്ന വരികൾ ഒരു റിലേഷനിൽ നിന്ന് തിരഞ്ഞെടുക്കുന്നതിന് സെലക്ട് പ്രവർത്തനം ഉപയോഗിക്കുന്നു. ഉപയോക്താവിന്റെ ഇഷ്ടാനുസരണം നിശ്ചിത വരികൾ തിരഞ്ഞെടുക്കാനുള്ള ഒരു ഉപയോക്തൃ നിർദിഷ്ട വ്യവസ്ഥയാണ് ഉപാധി (predicate). സിഗ്മ (σ) എന്ന അക്ഷരം ഉപയോഗിച്ച് ഈ പ്രവർത്തനം സൂചിപ്പിക്കുന്നു. സെലക്ടിന്റെ പൊതുവായ മാതൃക താഴെ കൊടുക്കുന്നു.

$$\sigma_{\text{condition}}(\text{Relation})$$

SELECT പ്രവർത്തനത്തിന്റെ ഫലമായുണ്ടാകുന്ന റിലേഷനിൽ തന്നിട്ടുള്ള വ്യവസ്ഥകൾ പാലിക്കപ്പെടുന്ന എല്ലാ വരികളും അടങ്ങിയിരിക്കുന്നു. റിലേഷണൽ ബീജഗണിതത്തിൽ താഴെപ്പറയുന്ന താരതമ്യ ഓപ്പറേറ്ററുകൾ ഉപയോഗിക്കുന്നു. < (കുറവ്), <= (കുറവോ അല്ലെങ്കിൽ തുല്യമോ), > (കൂടുതൽ), >= (കൂടുതലോ അല്ലെങ്കിൽ തുല്യമോ), = (തുല്യമാണ്), <> (തുല്യമല്ല) എന്നീ താരതമ്യ ഓപ്പറേറ്ററുകൾ ലളിതമായ വ്യവസ്ഥകൾ സജ്ജമാക്കുന്നതിന് ഉപയോഗിക്കുന്നു. അതുപോലെ ലോജിക്കൽ ഓപ്പറേറ്ററുകളായ ∨ (OR), ^ (AND), ! (NOT) എന്നിവ സമ്മിശ്ര വ്യവസ്ഥകൾ നിർമ്മിക്കാനും ഉപയോഗിക്കുന്നു.

SELECT പ്രവർത്തനം വിശദീകരിക്കുന്നതിന്, പട്ടിക 8.1 ൽ നൽകിയിട്ടുള്ള STUDENT റിലേഷൻ പരിഗണിക്കുക. എങ്ങനെയാണ് SELECT പ്രവർത്തനം റിലേഷണൽ

ബീജഗണിതത്തിൽ ഉപയോഗിക്കുന്നതെന്നും അത് എന്ത് ഫലമാണ് ഉണ്ടാക്കുന്നതെന്നും താഴെ കൊടുത്തിരിക്കുന്ന ഉദാഹരണങ്ങൾ വ്യക്തമാക്കുന്നു

ഉദാഹരണം 8.1: ഉന്നതവിദ്യാഭ്യാസത്തിന് യോഗ്യതയുള്ള എല്ലാ വിദ്യാർഥികളെയും തിരഞ്ഞെടുക്കുക.

$$\sigma_{\text{Result="EHS"}}(\text{STUDENT})$$

പട്ടിക 8.4 ൽ കാണിച്ചിരിക്കുന്നതുപോലെയുള്ള ഒരു റിലേഷനാണ് ഈ പ്രവർത്തനത്തിന്റെ ഫലം.

AdmNo	Roll	Name	Batch	Marks	Result
101	24	Sachin	Science	480	EHS
102	14	Rahul	Commerce	410	EHS
105	24	Nelson	Humanities	385	EHS
106	8	Joseph	Commerce	350	EHS
108	2	Bincy	Science	300	EHS

പട്ടിക 8.4: ഉദാഹരണം 8.1 ന്റെ ഔട്ട്പുട്ട്

ഉദാഹരണം 8.2: കോമേഴ്സ് ബാച്ചിലെ പരാജയപ്പെട്ട വിദ്യാർഥികളെ തിരഞ്ഞെടുക്കുക

$$\sigma_{\text{Result="NHS"} \wedge \text{Batch="Commerce"}}(\text{STUDENT})$$

പട്ടിക 8.5 ൽ കാണിച്ചിരിക്കുന്നതു പോലെയുള്ള ഒരു റിലേഷനാണ് ഈ പ്രവർത്തനത്തിന്റെ ഫലം.

AdmNo	Roll	Name	Batch	Marks	Result
104	12	Mahesh	Commerce	180	NHS

പട്ടിക 8.5: ഉദാഹരണം 8.2 ന്റെ ഔട്ട്പുട്ട്

ഉദാഹരണം 8.3: സയൻസ് അല്ലെങ്കിൽ കൊമേഴ്സിലുള്ള എല്ലാ വിദ്യാർഥികളെയും തിരഞ്ഞെടുക്കുക.

$$\sigma_{\text{Batch="Science"} \vee \text{Batch="Commerce"}}(\text{STUDENT})$$

പട്ടിക 8.6 ലെ ഔട്ട്പുട്ട് ഈ ഓപ്പറേഷൻ കാണിക്കുന്നു

AdmNo	Roll	Name	Batch	Marks	Result
101	24	Sachin	Science	480	EHS
102	14	Rahul	Commerce	410	EHS
104	12	Mahesh	Commerce	180	NHS
106	8	Joseph	Commerce	350	EHS
108	2	Bincy	Science	300	EHS

പട്ടിക 8.6: ഉദാഹരണം 8.3 ന്റെ ഔട്ട്പുട്ട്

8.7.2 പ്രൊജക്റ്റ് പ്രവർത്തനം (PROJECT operation)

PROJECT പ്രവർത്തനം റിലേഷനിൽ നിന്ന് ചില നിരകൾ തിരഞ്ഞെടുക്കുകയും ഒരു പുതിയ റിലേഷൻ രൂപപ്പെടുത്തുകയും ചെയ്യുന്നു. ഏതെങ്കിലും പ്രവർത്തനത്തിന് ഒരു റിലേഷനിലെ ഏതാനും നിരകൾ മാത്രം തിരഞ്ഞെടുക്കണമെങ്കിൽ PROJECT പ്രവർത്തനം ഉപയോഗിക്കാം. പൈ (π) എന്ന അക്ഷരം ഉപയോഗിച്ച് ഈ പ്രവർത്തനം സൂചിപ്പിക്കുന്നു. PROJECT പ്രവർത്തനത്തിന്റെ പൊതുവായ മാതൃക താഴെ പറയുന്നു.

$$\pi_{A1, A2, \ldots, An}\ (\text{Relation})$$

ഇവിടെ A1, A2,. . . ., എന്നത് തന്നിരിക്കുന്ന റിലേഷനുകളിലെ നിരകളാകുന്നു.

ഉദാഹരണം 8.4: STUDENT റിലേഷനിലെ പേര്, റിസൽട്ട്, മാർക്ക് എന്നീ നിരകൾ തിരഞ്ഞെടുക്കുക.

$$\pi_{\text{Name, Marks, Result}}\ (\text{STUDENT})$$

ഈ പ്രക്രിയയുടെ ഫലം പട്ടിക 8.7 ൽ നൽകിയിരിക്കുന്നു

SELECT, PROJECT എന്നീ പ്രവർത്തനങ്ങളെ ഒരൊറ്റ പ്രസ്താവനയിൽ സംയോജിപ്പിക്കുവാൻ സാധിക്കും. ഇതിന്റെ വിശദീകരണം ഉദാഹരണം 8.5 ലും 8.6 ലും കൊടുത്തിട്ടുണ്ട്

Name	Marks	Result
Sachin	480	EHS
Rahul	410	EHS
Fathima	200	NHS
Mahesh	180	NHS
Nelson	385	EHS
Joseph	350	EHS
Shaji	205	NHS
Bincy	300	EHS

പട്ടിക 8.7: ഉദാഹരണം 8.4 ന്റെ ഔട്ട്പുട്ട്

ഉദാഹരണം 8.5: ഉന്നത പഠനത്തിനു യോഗ്യരായ വിദ്യാർഥികളുടെ അഡ്മിഷൻ നമ്പരും പേരും തിരഞ്ഞെടുക്കുക.

$$\pi_{\text{AdmNo, Name}}\ (\sigma_{\text{result="EHS"}}\ (\text{STUDENT}))$$

ഈ പ്രക്രിയയുടെ ഫലമായുണ്ടായ റിലേഷൻ പട്ടിക 8.8 ൽ കൊടുത്തിട്ടുണ്ട്. ഫലത്തിന്റെ കൃത്യത പരിശോധിച്ചുറപ്പിക്കുന്നതിനു പട്ടിക 8.4 ഉപയോഗിച്ച് താരതമ്യം ചെയ്യുക.

AdmNo	Name
101	Sachin
102	Rahul
105	Nelson
106	Joseph
108	Bincy

പട്ടിക 8.8: ഉദാഹരണം 8.5 ന്റെ ഔട്ട്പുട്ട്

ഉദാഹരണം 8.6: ഉന്നതവിദ്യാഭ്യാസത്തിന് യോഗ്യരല്ലാത്ത ഹ്യുമാനിറ്റീസ് ബാച്ചിലെ വിദ്യാർഥികളുടെ പേരും മാർക്കുകളും തിരഞ്ഞെടുക്കുക.

$$\pi_{\text{Name, Marks}}\ (\sigma_{\text{result="NHS"} \wedge \text{Batch="Humanities"}}\ (\text{STUDENT}))$$

കൂട്ടിയിണക്കിയ ഈ പ്രവർത്തനങ്ങളുടെ ഫലം പട്ടിക 8.9 ൽ കാണിച്ചിരിക്കുന്നു.

Name	Marks
Fathima	200
Shaji	205

പട്ടിക 8.9: ഉദാഹരണം 8.6 ന്റെ ഔട്ട്പുട്ട്

8.7.3 യൂണിയൻ പ്രവർത്തനം (UNION operation)

UNION പ്രവർത്തനം ഒരു ബൈനറി ഓപ്പറേഷനാണ്. അത് രണ്ട് നിർദിഷ്ട റിലേഷനുകളിലെ എല്ലാ വരികളും അടങ്ങുന്ന ഒരു പുതിയ റിലേഷൻ തരുന്നു. '∪' എന്ന ചിഹ്നം കൊണ്ട് UNION പ്രവർത്തനം സൂചിപ്പിക്കുന്നു. രണ്ടു റിലേഷനുകളും യൂണിയൻ അനുരൂപമായിരിക്കുകയും ഫലത്തിന്റെ സ്കീമ ആദ്യ റിലേഷന്റെ സ്കീമയ്ക്ക് സമാനമായിരിക്കുകയും ചെയ്യും. രണ്ട് റിലേഷനുകൾ യൂണിയൻ അനുരൂപമാണെങ്കിൽ, അവയ്ക്ക് ഒരേ എണ്ണം നിരകളും, സമാനമായ നിരകൾക്ക് സമാന ഡൊമെയ്നുമായിരിക്കണം. യൂണിയൻ അനുരൂപത നിർവചിക്കുന്നതിന് നിരകളുടെ പേരുകൾ പരിഗണിക്കാറില്ല എന്നത് ശ്രദ്ധിക്കുക.

പട്ടിക 8.10 ലും 8.11 ലും നൽകിയിരിക്കുന്ന ARTS, SPORTS എന്നീ രണ്ട് റിലേഷനുകൾ പരിഗണിക്കുക. ഇവയിൽ യഥാക്രമം കലാമേളയിൽ പങ്കെടുക്കുന്ന വിദ്യാർഥികളുടെ വിശദാംശങ്ങളും കായികമേളയിൽ പങ്കെടുക്കുന്ന വിദ്യാർഥികളുടെ വിശദാംശങ്ങളും അടങ്ങിയിരിക്കുന്നു. AdmNo, Name, BatchCode എന്നിവ ARTS, SPORTS എന്നീ റിലേഷനുകളിലെ നിരകളാണ്. ഈ രണ്ട് റിലേഷനുകളും യൂണിയൻ അനുരൂപമാണെന്നത് വ്യക്തമാണ്. അതായത്, ഈ രണ്ട് റിലേഷനുകളിലും ഒരേ എണ്ണം നിരകൾ ഉണ്ട്, സദൃശമായ നിരകളുടെ തരവും ഒരുപോലെയാണ്.

ARTS relation

AdmNo	Name	BatchCode
101	Sachin	S2
103	Fathima	H2
106	Joseph	C2
110	Nikitha	S1
132	Vivek	C1
154	Nevin	C1

പട്ടിക 8.10: ARTS

SPORTS relation

AdmNo	Name	BatchCode
102	Rahul	C2
103	Fathima	H2
105	Nelson	H2
106	Joseph	C2
108	Bincy	S2
132	Vivek	C1
164	Rachana	S1

പട്ടിക 8.11: SPORTS

ARTS ∪ SPORTS എന്ന റിലേഷൻ കലയിലോ, സ്പോർട്സിലോ അല്ലെങ്കിൽ രണ്ടിലുമോ പങ്കെടുക്കുന്ന വിദ്യാർഥികളുടെ വിശദാംശങ്ങൾ നൽകുന്നു. അതായത്, ARTS ∪ SPORTS പട്ടിക 8.12 ൽ കാണിച്ചിരിക്കുന്നതുപോലെയുള്ള ഒരു റിലേഷൻ നൽകുന്നു. ഈ ടേബിളിൽ ARTS അല്ലെങ്കിൽ SPORTS അല്ലെങ്കിൽ രണ്ടിലുമുള്ള റെക്കോർഡുകൾ ഉൾപ്പെടുന്നു. അതേ സമയം തനിപ്പകർപ്പ് ഒഴിവാക്കുകയും ചെയ്യുന്നു. പട്ടിക 8.12 ൽ 103, 106 എന്നീ അഡ്മിഷൻ നമ്പറുകളിലുള്ള വിദ്യാർഥികളുടെ രേഖകൾ ഒരിക്കൽ മാത്രമേ ലഭ്യമാകൂ.

AdmNo	Name	BatchCode
101	Sachin	S2
103	Fathima	H2
106	Joseph	C2
110	Nikitha	S1
132	Vivek	C1
154	Nevin	C1
102	Rahul	C2
105	Nelson	H2
108	Bincy	S2
164	Rachana	S1

പട്ടിക 8.12: ARTS ∪ SPORTS

8.7.4 ഇന്റർസെക്ഷൻ പ്രവർത്തനം (INTERSECTION operation)

INTERSECTION പ്രവർത്തനവും ഒരു ബൈനറി പ്രവർത്തനമാണ്. അത്, രണ്ട് നിർദിഷ്ട റിലേഷനുകളിലെ പൊതുവായ എല്ലാ വരികളും അടങ്ങുന്ന ഒരു പുതിയ റിലേഷൻ തരുന്നു. ∩ എന്നചിഹ്നം കൊണ്ട് INTERSECTION പ്രവർത്തനം സൂചിപ്പിക്കുന്നു. രണ്ടു റിലേഷനുകളും യൂണിയൻ അനുരൂപമായിരിക്കുകയും ഫലത്തിന്റെ സ്കീമ ആദ്യ റിലേഷന്റെ സ്കീമയ്ക്ക് സമാനമായിരിക്കുകയും ചെയ്യും. രണ്ട് റിലേഷനുകൾ യൂണിയൻ അനുരൂപമാണെങ്കിൽ, അവയ്ക്ക് ഒരേ എണ്ണം നിരകളും, സമാനമായ നിരകൾക്ക് സമാന ഡൊമെയ്നുമായിരിക്കണം. യൂണിയൻ അനുരൂപത നിർവചിക്കുന്നതിന് നിരകളുടെ പേരുകൾ പരിഗണിക്കാറില്ല എന്നത് ശ്രദ്ധിക്കുക.

നാം 8.10, 8.11 എന്നീ റിലേഷനുകളിൽ INTERSECT പ്രവർത്തനം നടത്തുകയാണെങ്കിൽ, ആർട്സിലും സ്പോർട്സിലും പൊതുവായി പങ്കെടുക്കുന്ന വിദ്യാർഥികളുടെ വിശദാംശങ്ങൾ ലഭിക്കും. അതായത് ARTS ∩ SPORTS റിലേഷനിൽ ARTS, SPORTS എന്നിവ രണ്ടിലുമുള്ള വരികൾ മാത്രമേ ഉണ്ടാകു (പട്ടിക 8.13)

AdmNo	Name	BatchCode
103	Fathima	H2
106	Joseph	C2
132	Vivek	C1

പട്ടിക 8.13: ARTS ∩ SPORTS

8.7.5 സെറ്റ് ഡിഫറൻസ് പ്രവർത്തനം (SET DIFFERENCE operation)

SET DIFFERENCE പ്രവർത്തനവും ഒരു ബൈനറി പ്രവർത്തനമാണ്. അത്, ആദ്യ റിലേഷനിൽ ഉള്ളതും രണ്ടാമത്തെ റിലേഷനിൽ ഇല്ലാത്തതുമായ എല്ലാ വരികളും അടങ്ങുന്ന ഒരു പുതിയ റിലേഷൻ തരുന്നു. '–' എന്നചിഹ്നം കൊണ്ട് SET DIFFERENCE, പ്രവർത്തനം സൂചിപ്പിക്കുന്നു. ഇവിടെയും രണ്ടു റിലേഷനുകളും യൂണിയൻ അനുരൂപമായിരിക്കണം.

AdmNo	Name	BatchCode
101	Sachin	S2
110	Nikitha	S1
154	Nevin	C1

പട്ടിക 8.14: ARTS - SPORTS

AdmNo	Name	BatchCode
102	Rahul	C2
105	Nelson	H2
108	Bincy	S2
164	Rachana	S1

പട്ടിക 8.15: SPORTS - ARTS

ARTS – SPORTS റിലേഷനിൽ, ആർട്സിൽ പങ്കെടുക്കുകയും എന്നാൽ സ്പോർട്സിൽ പങ്കെടുക്കാത്തതുമായ വിദ്യാർഥികളുടെ വിശദാംശങ്ങൾ ലഭിക്കും. അതായത് ARTS – SPORTS റിലേഷനിൽ പട്ടിക 8.14 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ARTS ൽ ഉള്ളതും, എന്നാൽ SPORTS ൽ ഇല്ലാത്തതുമായ വരികൾ ഉണ്ടാകും. അതുപോലെ SPORTS – ARTS റിലേഷനിൽ പട്ടിക 8.15 ൽ കാണിച്ചിരിക്കുന്നതു പോലെ SPORTS ൽ ഉള്ളതും, എന്നാൽ ARTS ൽ ഇല്ലാത്തതുമായ വരികളും ഉണ്ടാകും.

8.7.6 കാർട്ടീഷ്യൻ പ്രൊഡക്ട് പ്രവർത്തനം (CARTESIAN PRODUCT operation)

രണ്ട് റിലേഷനുകളിൽ നിന്നുള്ള വരികളുടെ സാധ്യമായ എല്ലാ കൂട്ടിച്ചേരലുകളും ഉൾക്കൊള്ളുന്ന ഒരു റിലേഷൻ കാർട്ടീഷ്യൻ പ്രൊഡക്ട് നൽകുന്നു. പുതിയ റിലേഷന്റെ ഡിഗ്രി

(ആട്രിബ്യൂട്ടുകളുടെ എണ്ണം) രണ്ട് റിലേഷനുകളുടെയും ആകെ ഡിഗ്രിക്ക് തുല്യമാണ്. ഇതിന്റെ കാർഡിനാലിറ്റി (വരികളുടെ എണ്ണം) രണ്ട് റിലേഷനിലുമുള്ള വരികളുടെ എണ്ണത്തിന്റെ ഗുണനഫലമാണ്. കാർട്ടീഷ്യൻ പ്രൊഡക്ട് സൂചിപ്പിക്കുന്നതിന് '×' (ക്രോസ്) ചിഹ്നം ഉപയോഗിക്കുന്നു. ഇത് ക്രോസ് പ്രൊഡക്ട് എന്നും അറിയപ്പെടുന്നു. ആദ്യത്തെ റിലേഷനിലെ എല്ലാ വരികളും രണ്ടാമത്തെ റിലേഷന്റെ വരികളുമായി കൂടിച്ചേർന്ന് പുതിയ ബന്ധത്തിന്റെ വരികൾ രൂപപ്പെടുത്തുന്നു.

TEACHER relation

TeacherId	Name	Dept
1001	Viswesaran	English
1002	Meenakshi	Computer

പട്ടിക 8.16: TEACHER റിലേഷൻ

പട്ടിക 8.16ൽ കാണിച്ചിരിക്കുന്ന TEACHER റിലേഷൻ നമുക്ക് പരിഗണിക്കാം. പട്ടിക 8.2 ലെ STUDENT റിലേഷനുമായുള്ള ഈ റിലേഷന്റെ കാർട്ടീഷ്യൻ പ്രൊഡക്ട് പ്രവർത്തനത്തിന്റെ ഫലം പട്ടിക 8.19 ൽ കാണിച്ചിരിക്കുന്നു. STUDENT റിലേഷനിലെ ഓരോ വരിയും TEACHER റിലേഷനിലെ വരികളുമായി കൂടിച്ചേർന്നതായി ഈ പട്ടിക കാണിക്കുന്നു.

Adm No	Roll	Name	Batch Code	Marks	Result	TeacherId	Name	Dept
101	24	Sachin	S2	480	EHS	1001	Viswesaran	English
101	24	Sachin	S2	480	EHS	1002	Meenakshi	Computer
102	14	Rahul	C2	410	EHS	1001	Viswesaran	English
102	14	Rahul	C2	410	EHS	1002	Meenakshi	Computer
103	4	Fathima	H2	200	NHS	1001	Viswesaran	English
103	4	Fathima	H2	200	NHS	1002	Meenakshi	Computer
104	12	Mahesh	C2	180	NHS	1001	Viswesaran	English
104	12	Mahesh	C2	180	NHS	1002	Meenakshi	Computer
105	24	Nelson	H2	385	EHS	1001	Viswesaran	English
105	24	Nelson	H2	385	EHS	1002	Meenakshi	Computer
106	8	Joseph	C2	350	EHS	1001	Viswesaran	English
106	8	Joseph	C2	350	EHS	1002	Meenakshi	Computer
107	24	Shaji	H2	205	NHS	1001	Viswesaran	English
107	24	Shaji	H2	205	NHS	1002	Meenakshi	Computer
108	2	Bincy	S2	300	EHS	1001	Viswesaran	English
108	2	Bincy	S2	300	EHS	1002	Meenakshi	Computer

പട്ടിക 8.19: STUDENT X TEACHER

ഒരു ഡാറ്റാബേസ് മോഡൽ ഡാറ്റയുടെ ലോജിക്കൽ രൂപരേഖ നിർവചിക്കുന്നു. ഡാറ്റയുടെ വിവിധ ഘടകങ്ങൾ തമ്മിലുള്ള ബന്ധത്തെ ഈ മോഡൽ വിശദീകരിക്കുന്നു. ഹൈറാർക്കിയൽ മോഡൽ, നെറ്റ്‌വർക്ക് മോഡൽ, റിലേഷണൽ മോഡൽ, ഒബ്ജക്റ്റ് ഓറിയന്റഡ് മോഡൽ എന്നീ വിവിധങ്ങളായ മോഡലുകൾ ഉപയോഗിച്ചു വരുന്നു. പഴയകാല മെയിൻഫ്രെയിം ഡാറ്റാബേസ് നിർവഹണ സംവിധാനമായ IBM ന്റെ ഇൻഫർമേഷൻ മാനേജ്മെന്റ് സിസ്റ്റം (IMS) പോലെയുള്ള ഹൈറാർക്കിയൽ മോഡൽ ഉപയോഗിച്ചിരുന്നു. സാധാരണയായി ഉപയോഗിച്ചിരുന്ന നെറ്റ്‌വർക്ക് മോഡലാണ് സിൽക്കോം സിസ്റ്റത്തിന്റെ ടോട്ടലും കുള്ളിനെറ്റിന്റെ IDMS ഉം.

നമുക്ക് സംഗ്രഹിക്കാം

DBMSലെ അടിസ്ഥാന ആശയങ്ങളെക്കുറിച്ചും അതിന്റെ ഘടകങ്ങളെക്കുറിച്ചും നമ്മൾ ചർച്ച ചെയ്തു. പരമ്പരാഗത ഫയൽ സംവിധാനത്തെ അപേക്ഷിച്ച് DBMSനുള്ള മേന്മകളും മനസ്സിലാക്കി. റിലേഷണൽ ഡാറ്റാ മോഡലിന്റെ പശ്ചാത്തലത്തിൽ ഡാറ്റാബേസുമായി ബന്ധപ്പെട്ട വിവിധ പദാവലികളെ കുറിച്ചുള്ള ഒരു സംക്ഷിപ്ത രൂപം പ്രതിപാദിച്ചു. ഡാറ്റാബേസിൽ ഒരിക്കൽ ഡാറ്റ വ്യവസ്ഥാപിതമായി ക്രമീകരിച്ചു കഴിഞ്ഞാൽ, ആവശ്യമുള്ള വിവരങ്ങൾ നിർമ്മിക്കുന്നതിനായി റിലേഷണൽ ബീജഗണിതത്തിലെ പ്രവർത്തനങ്ങൾ എങ്ങനെ ഉപയോഗിക്കാമെന്ന് ഉദാഹരണത്തിന്റെ സഹായത്തോടെ നമ്മൾ പഠിച്ചു.

ഈ അധ്യായത്തിൽ അവതരിപ്പിച്ച ആശയങ്ങളെക്കുറിച്ചുള്ള നല്ല ഒരു ധാരണ അടുത്ത അധ്യായം ഫലപ്രദമായി പഠിക്കുന്നതിന് അത്യാവശ്യമാണ്. ക്വറി ഭാഷ ഉപയോഗിച്ച് ഡാറ്റാബേസുകൾ നിർമ്മിക്കുന്നതും ശേഖരിച്ച വിവരങ്ങൾ കൈകാര്യം ചെയ്യുന്ന രീതിയും അടുത്ത അധ്യായത്തിൽ നമ്മൾ ചർച്ച ചെയ്യും.

നമുക്ക് വിലയിരുത്താം

1. ഡാറ്റാബേസുമായി ബന്ധപ്പെട്ട പ്രവർത്തനങ്ങൾ കൈകാര്യം ചെയ്യുന്നതും നിയന്ത്രിക്കുന്നതും ആരാണ്?
   a. ഡാറ്റാബേസ് അഡ്മിനിസ്ട്രേറ്റർ b. പ്രോഗ്രാമർ
   c. സാങ്കേതിക പരിജ്ഞാനമുള്ള ഉപയോക്താവ് d. സാധാരണ ഉപയോക്താവ്
2. റിലേഷണൽ മോഡലിൽ, കാർഡിനാലിറ്റി എന്നു കൊണ്ട് ഉദ്ദേശിക്കുന്നത് എന്താണ്?
   a. ടൂപ്പിളുകളുടെ എണ്ണം b. ആട്രിബ്യൂട്ടുകളുടെ എണ്ണം
   c. ടേബിളുകളുടെ എണ്ണം d. കൺസ്ട്രയ്ന്റുകളുടെ എണ്ണം
3. റിലേഷണൽ ബീജഗണിതത്തിലെ കാർട്ടീഷ്യൻ പ്രൊഡക്ട് എന്നത്,
   a. ഒരു യൂനറി ഓപ്പറേറ്റർ b. ഒരു ബൈനറി ഓപ്പറേറ്റർ
   c. ഒരു ടർണറി ഓപ്പറേറ്റർ d. നിർവചിക്കപ്പെട്ടിട്ടില്ല
4. ഡാറ്റാബേസ് സംഗ്രഹം എത്ര തലങ്ങളാണ്?
   a. ഒരു തലം b. രണ്ട് തലങ്ങൾ
   b. മൂന്നു തലങ്ങൾ d. നാലു തലങ്ങൾ
5. ഒരു റിലേഷണൽ മോഡലിൽ, റിലേഷൻ എന്ന് പറയുന്നത്
   a. ടൂപ്പിളുകൾ b. ആട്രിബ്യൂട്ടുകൾ
   c. ടേബിളുകൾ d. വരികൾ

6. ഒരു ഡാറ്റാബേസ് സംഗ്രഹത്തിന്റെ ബാഹ്യതലം എന്നത്

a. ഭൗതിക തലം
b. ലോജിക്കൽ തലം
c. കൺസെപ്ചൽ തലം
d. വ്യൂ തലം

7. ഒരു ഡാറ്റാബേസിലെ ബന്ധപ്പെട്ട ഫീൽഡുകളുടെ കൂട്ടം അറിയപ്പെടുന്നത്.

a. ഡാറ്റ ഫയൽ
b. ഡാറ്റ റെക്കോർഡ്
c. മെനു
d. ബാങ്ക്

8. ഒരു ഡാറ്റാബേസ് ഡെവലപ്പർ ഒരു റെക്കോർഡ് എന്നത് കൊണ്ട് ഉദ്ദേശിക്കുന്നത്.

a. മാനദണ്ഡം
b. റിലേഷൻ
c. ടൂപിൾ
d. ആട്രിബ്യൂട്ട്

9. ഡാറ്റാബേസ് മാനേജ്മെന്റ് സമീപനത്തിന്റെ ഒരു ഗുണം

a. ഡാറ്റ പ്രോഗ്രാമുകളെ ആശ്രയിച്ചിരിക്കുന്നു
b. ഡാറ്റാ റിഡൻഡൻസി വർധിക്കുന്നു
c. ഡാറ്റ സംയോജിപ്പിച്ച് ഒന്നിലധികം പ്രോഗ്രാമുകൾക്ക് ഉപയോഗിക്കുവാൻ കഴിയും
d. ഇവ ഒന്നുമല്ല

10. ഡാറ്റ സ്വാതന്ത്ര്യം എന്നാൽ എന്താണ് ?

a. ഡാറ്റ വെവ്വേറെ നിർവചിക്കുകയും പ്രോഗ്രാമുകളിൽ ഉൾപ്പെടുത്താതിരിക്കുകയും ചെയ്യുക.
b. പ്രോഗ്രാമുകൾ ഡാറ്റയുടെ ഭൗതിക ആട്രിബ്യൂട്ടുകളെ ആശ്രയിക്കുന്നില്ല.
c. പ്രോഗ്രാമുകൾ ഡാറ്റയുടെ ലോജിക്കൽ ആട്രിബ്യൂട്ടുകളെ ആശ്രയിക്കുന്നില്ല.
d. (ബി) യും (സി) യും

11. റിലേഷനുകൾ തമ്മിലുള്ള ബന്ധം പ്രതിനിധാനം ചെയ്യുന്ന കീയെ എന്ത് വിളിക്കുന്നു?

a. പ്രൈമറി കീ
b. കാൻഡിഡേറ്റ് കീ
c. ഫോറിൻ കീ
d. ആൾട്ടർനേറ്റ് കീ

12. ടേബിളുകളിലെ ചില നിരകളിൽ മാത്രം മതിയെങ്കിൽ താഴെ പറയുന്ന ഏത് പ്രവർത്തനമാണ് ഉപയോഗിക്കുന്നത്?

a. PROJECTION
b. SELECTION
c. UNION
d. SELECT

13. താഴെ പറയുന്ന ഏത് പ്രവർത്തനത്തിനാണ് റിലേഷനുകൾ യൂണിയൻ അനുരൂപമാകേണ്ടത്.

a. UNION
b. INTERSECTION
c. SET DIFFERENCE
d. മുകളിൽ പറഞ്ഞവയെല്ലാം

14. ഏത് ഡാറ്റാബേസ് തലമാണ് ഉപയോക്താക്കൾക്ക് ഏറ്റവും അടുത്തുള്ളത്?

a. ബാഹ്യതലം
b. ആന്തരികതലം
c. ഭൗതികതലം
d. ലോജിക്കൽതലം

15. R1, R2 എന്നീ റിലേഷനുകളുടെ UNION ഓപ്പറേഷൻ നടത്തിയാൽ, അതിൽ

a. R1 ന്റെ എല്ലാ ടൂപ്പിളുകളും ഉൾപ്പെടും.

b. R2 ന്റെ എല്ലാ ടൂപ്പിളുകളും ഉൾപ്പെടും.

c. R1, R2 എന്നിവയിലുള്ള എല്ലാ ടൂപ്പിളുകളും ഉൾപ്പെടും.

d. R1, R2 എന്നിവയിലെ പൊതുവായ എല്ലാ ടൂപ്പിളുകളും ഉൾപ്പെടും.

16. ഒരു ഫയലിൽ നിന്ന് ചില റെക്കോർഡുകൾ ശേഖരിക്കുന്നതിനുള്ള ഒരു ഫയൽ മാനിപ്പുലേഷൻ പ്രവർത്തനം.

a. SELECT
b. PROJECT
c. JOIN
d. PRODUCT

17. R (A, B, C) എന്ന റിലേഷണൽ ഘടനയിൽ A യുടെ മൂല്യത്തിൽ NULL മൂല്യവും ഉൾപ്പെടുന്നു. താഴെ പറയുന്ന പ്രസ്താവനയിൽ ശരിയായിട്ടുള്ളത് ഏത്?

a. A ഒരു കാൻഡിഡേറ്റ് കീയാണ്
b. A ഒരു കാൻഡിഡേറ്റ് കീ അല്ല
c. A ഒരു പ്രൈമറി കീയാണ്
d. (a) യും (c) യും

18. കാർഡിനാലിറ്റി 22 ഉള്ള ഒരു റിലേഷനിൽ എത്ര ടൂപ്പിളുകൾ ഉണ്ട്?

a. 22
b. 11
c. 1
d. ഇവയൊന്നുമല്ല

19. ഒരു റിലേഷനിലെ നിരയുടെ സാധ്യമായ മൂല്യങ്ങളെ എന്ത് വിളിക്കുന്നു?

a. ആട്രിബ്യൂട്ട്
b. ഡിഗ്രി
c. ടൂപ്പിൾ
d. ഡൊമെയ്ൻ

20. പരമ്പരാഗത ഫയലുകളിൽ ഡാറ്റ സംഭരിക്കുന്നതിനുപകരം ഒരു ഡാറ്റാബേസ് സംവിധാനം നിങ്ങൾ തിരഞ്ഞെടുക്കുന്നത് എന്തുകൊണ്ട്?

21. DBMS ലെ വിവിധ തലത്തിലുള്ള ഡാറ്റാസംഗ്രഹങ്ങൾ വിശദീകരിക്കുക?

22. ലോജിക്കൽ, ഭൗതികം എന്നീ ഡാറ്റാസ്വാതന്ത്ര്യത്തിന്റെ ആശയങ്ങളുമായി സ്കീമ യുടെ വിവിധ തലങ്ങൾ എങ്ങനെ ബന്ധപ്പെട്ടിരിക്കുന്നു?

23. താഴെ തന്നിരിക്കുന്ന EMPLOYEE റിലേഷന്റെ ആട്രിബ്യൂട്ടുകൾ, ഡിഗ്രി, കാർഡിനാലിറ്റി, Name ന്റെ ഡൊമെയ്ൻ, Emp_Code ന്റെ ഡൊമെയ്ൻ എന്നിവ എഴുതുക.

Emp_Code	Name	Department	Designation	Salary
1000	Sudheesh	Purchase	Manager	25000
1001	Dhanya	Sales	Manager	25000
1002	Fathima	Marketing	Clerk	12000
1003	Shajan	Sales	Clerk	13000

24. ചോദ്യം 23ൽ തന്നിരിക്കുന്ന EMPLOYEE റിലേഷന്റെ പ്രൈമറി കീ, കാൻഡിഡേറ്റ് കീകൾ, ആൾട്ടർനേറ്റ് കീകൾ എന്നിവ എഴുതുക.

25. താഴെ കൊടുത്തിരിക്കുന്ന STUDENT റിലേഷന്റെ പ്രൈമറി കീ Reg_no ആയാൽ.

a. STUDENT റിലേഷന്റെ കാൻഡിഡേറ്റ് കീകളും ആൾട്ടർനേറ്റ് കീകളും കണ്ടെത്തുക.

b. പ്രൈമറി കീയും കാൻഡിഡേറ്റ് കീയും തമ്മിലുള്ള ബന്ധം എന്താണ്?

Reg_no	Name	Batch	Result	Marks
101	Sachin	Science	Pass	480
103	Fathima	Humanities	Fail	200
106	Joseph	Commerce	Pass	350
108	Bincy	Science	Pass	300

26. ഒരു ഡാറ്റാബേസ് എന്നാൽ എന്ത്? DBMS ഉപയോഗിക്കുന്നതിന്റെ ഗുണങ്ങളും ദോഷങ്ങളും വിവരിക്കുക.

27. ഡാറ്റാസ്വാതന്ത്ര്യം എന്നാൽ എന്താണ്? ഭൗതിക ഡാറ്റാസ്വാതന്ത്ര്യവും ലോജിക്കൽ ഡാറ്റാസ്വാതന്ത്ര്യവും തമ്മിലുള്ള വ്യത്യാസം എഴുതുക.

28. ഡാറ്റാസ്റ്റാൻഡേർഡ് നിർവചിക്കുക എന്നത് ഒരു DBMS ന്റെ പ്രധാന സവിശേഷതയാണ്. ഒരു ഡാറ്റാബേസിൽ ഈ മാനദണ്ഡങ്ങൾ എങ്ങനെയാണ് ബാധകമാകുന്നത്?

29. യൂണിയൻ അനുരൂപമായ T1, T2 എന്നീ റിലേഷനുകളിൽ T1 ന്റെ കാർഡിനാലിറ്റി 10 ഉം T2 ന്റെ കാർഡിനാലിറ്റി 8 ഉം ആണ്. T1 ∪ T2 ന്റെ കാർഡിനാലിറ്റി 13 ആണെങ്കിൽ, T1 ∩ T2 ന്റെ കാർഡിനാലിറ്റി എന്തായിരിക്കും? നിങ്ങളുടെ ഉത്തരം സമർഥിക്കുക.

30. യൂണിയൻ അനുരൂപമായ T1, T2 എന്നീ റിലേഷനുകളിൽ T1 ന്റെ കാർഡിനാലിറ്റി 10 ഉം T2 ന്റെ കാർഡിനാലിറ്റി 8 ഉം ആണ്.

a. T1 ∪ T2 ന്റെ സാധ്യമായ കാർഡിനാലിറ്റി എന്തായിരിക്കും?

b. T1 ∪ T2 ന്റെ ഏറ്റവും കുറഞ്ഞ കാർഡിനാലിറ്റി എന്തായിരിക്കും?

31. Ctiy (ctiy_name, state), Hotel (name, address, ctiy_name) എന്നീ റിലേഷനുകൾ പരിഗണിച്ച് താഴെപ്പറയുന്നവയ്ക്കുള്ള റിലേഷണൽ ബീജഗണിത പ്രസ്താവനകൾ എഴുതുക.

a. കൊച്ചി നഗരത്തിലെ ഹോട്ടലുകളുടെ പേരും വിലാസവും കണ്ടെത്തുക.

b. കേരളത്തിലെ നഗരങ്ങളുടെ വിശദാംശങ്ങൾ.

c. തൃശൂരിലെ ഹോട്ടലുകളുടെ പേരുകൾ കണ്ടെത്തുക.

d. വിവിധ ഹോട്ടലുകളുടെ പേരുകൾ കണ്ടെത്തുക.

e. കോഴിക്കോട് അല്ലെങ്കിൽ മൂന്നാറിലെ ഹോട്ടലുകളുടെ പേരുകൾ കണ്ടെത്തുക.

32. ചോദ്യം 23 ൽ കാണിച്ചിരിക്കുന്ന EMPLOYEE ബന്ധം ഉപയോഗിച്ചുകൊണ്ട്, താഴെ പറയുന്ന റിലേഷണൽ ബീജഗണിത എക്സ്പ്രഷനുകളുടെ ഫലം എഴുതുക.

a. $\sigma_{\text{Department="Sales"}}$ (EMPLOYEE).

b. $\sigma_{\text{salary>20000} \wedge \text{Department="Sales"}}$ (EMPLOYEE).

c. $\sigma_{\text{salary>20000} \vee \text{Department="Sales"}}$ (EMPLOYEE).

d. $\pi_{\text{name, salary}}$ (EMPLOYEE).

e. $\pi_{\text{name, salary}}(\sigma_{\text{Designation="Manager"}}$ (EMPLOYEE)).

f. $\pi_{\text{name, Department}}(\sigma_{\text{Designation="Clerk"} \wedge \text{salary > 20000}}$ (EMPLOYEE)).

33. ഒരു ബാങ്കിലെ ഉപഭോക്താക്കളുടെ വിവരങ്ങൾ സൂക്ഷിച്ചിരിക്കുന്ന BORROWER, DEPOSITOR എന്നീ റിലേഷനുകൾ ഉപയോഗിച്ചുകൊണ്ട്, താഴെ പറയുന്ന റിലേഷണൽ ബീജഗണിത പ്രസ്താവനകൾ എഴുതുക.

a. നിക്ഷേപകരുടെയും വായ്പ എടുത്തവരുടെയും വിശദാംശങ്ങൾ പ്രദർശിപ്പിക്കുക.

b. നിക്ഷേപകരും വായ്പ എടുത്തവരുമായ ഉപഭോക്താക്കളുടെ പേര് പ്രദർശിപ്പിക്കുക.

c. വായ്പ എടുത്തിട്ടില്ലാത്ത നിക്ഷേപകരുടെ പേര് വിവരം പ്രദർശിപ്പിക്കുക.

d. നിക്ഷേപമില്ലാതെ വായ്പ എടുത്തവരുടെ പേരും വായ്പാ തുകയും പ്രദർശിപ്പിക്കുക.

BORROWER

Acc_No	Name	Amount
AC123	Albin	50000
AC103	Rasheeda	25000
AC106	Vishnu	25000
AC108	Aiswarya	30000

DEPOSITOR

Acc_No	Name	Amount
AC123	Albin	500
AC105	Shabana	25000
AC116	Vishnu	125000
AC108	Aiswarya	3000

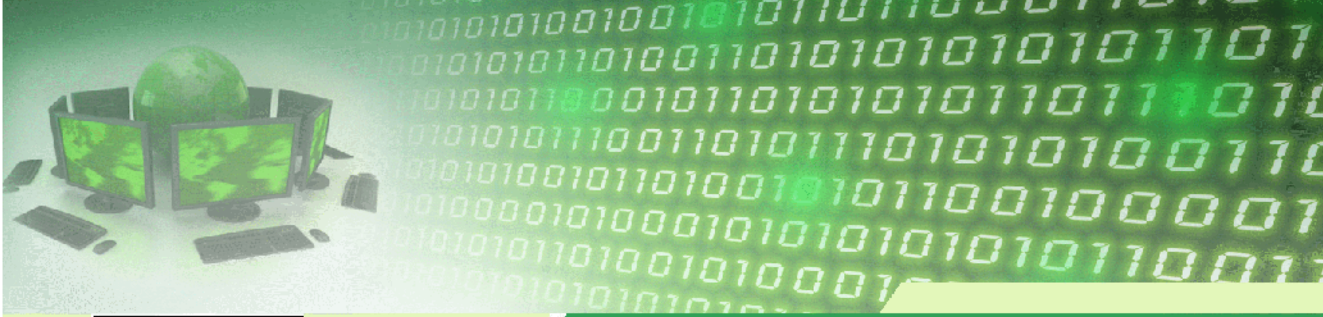
34. താഴെകൊടുത്തിരിക്കുന്ന CUSTOMER, BRANCH എന്നീ റിലേഷനുകളുടെ കാർട്ടീഷ്യൻ പ്രൊഡക്ട് എഴുതുക.

CUSTOMER

Acc_No	Name	Branch_ID	Amount
AC123	Albin	B1001	50000
AC103	Rasheeda	B1001	25000
AC106	Vishnu	B1001	25000
AC108	Aiswarya	B1077	30000

BRANCH

Branch_ID	Name
B1001	Kochi
B1002	Guruvayur
B1077	Idukki



P4G5H1

9 സ്ട്രക്ചേർഡ് ക്വറി ലാംഗ്വേജ് (SQL)

പ്രധാന പഠനനേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പഠനം പൂർത്തിയാകുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ

- സ്ട്രക്ചേർഡ് ക്വറി ലാംഗ്വേജ് (SQL) എന്ന പ്രോഗ്രാമിങ് ഭാഷയുടെ പ്രാധാന്യവും വിശേഷഗുണങ്ങളും തിരിച്ചറിയുന്നു.
- SQL ന്റെ ഘടകങ്ങളെക്കുറിച്ച് വിശദമാക്കുന്നു.
- DDL, DML, DCL കമാൻഡുകളുടെ വ്യത്യാസം കണ്ടെത്തുന്നു.
- MySQL ന്റെ പ്രത്യേകതകൾ തിരിച്ചറിയുന്നു.
- വിവിധ ഡാറ്റ ഇനങ്ങളും അവയുടെ സവിശേഷതകളും പട്ടികപ്പെടുത്തുന്നു.
- SQL ലെ വിവിധ കൺസ്ട്രയിന്റുകളുടെ സ്വാധീനം വിശദമാക്കുന്നു.
- DDL കമാൻഡുകളായ CREATE, ALTER, DROP എന്നിവ ഉപയോഗിച്ച് വിവിധ പ്രവർത്തനങ്ങൾ ചെയ്യുന്നു.
- ഡാറ്റ കൈകാര്യം ചെയ്യുന്നതിനായി SELECT, INSERT, UPDATE, DELETE എന്നിവ പോലെയുള്ള DML കമാൻഡുകൾ ഉപയോഗിക്കുന്നു.
- SQL കമാൻഡുകളുടെ വിവിധ ഉപവാക്യങ്ങളും അവയുടെ ഉപയോഗവും മനസ്സിലാക്കുന്നു.
- വിവിധ നിബന്ധനകൾ ഉണ്ടാക്കുന്നതിന് ഓപ്പറേറ്ററുകൾ ഉപയോഗിക്കുന്നു.
- സംഗ്രഹ ഫങ്ഷനുകൾ (Aggregate Functions) പട്ടികപ്പെടുത്തുകയും അവയുടെ ഉപയോഗം വിവരിക്കുകയും ചെയ്യുന്നു.
- വിവരം ലഭ്യമാക്കുന്നതിന് നെസ്റ്റഡ് ക്വറി നിർമ്മിക്കുന്നു.

കഴിഞ്ഞ അധ്യായത്തിൽ നാം റിലേഷണൽ ഡാറ്റാബേസ് മാനേജ്മെന്റ് സിസ്റ്റത്തെക്കുറിച്ച് (RDBMS) ചർച്ച ചെയ്തിരുന്നു. റിലേഷണൽ ഡാറ്റാബേസ് എന്നത് പട്ടികകളിൽ (Tables) ശേഖരിച്ചുവച്ചിരിക്കുന്ന അനുബന്ധ വിവരങ്ങളുടെ ശേഖരണമാണെന്ന് നമുക്കറിയാം. ഇവയെ നമുക്ക് റിലേഷൻ (Relation) എന്നുവിളിക്കാം. വിവിധതരം പ്രവർത്തനങ്ങൾ റിലേഷനുകളിൽ ചെയ്യുന്നതിന് വേണ്ട റിലേഷണൽ ആൾജിബ്രയെക്കുറിച്ച് അടിസ്ഥാനാശയം നമുക്കുണ്ട്. പട്ടിക നിർമ്മിക്കുക, അതിലേക്ക് ഡാറ്റ ചേർക്കുക, അതിലെ ഡാറ്റ കൈകാര്യം ചെയ്യുക, ഡാറ്റ പുതുക്കുക, ഡാറ്റ നീക്കം ചെയ്യുക എന്നിവയെക്കുറിച്ച് ഇനി നമുക്ക് ഒരു വ്യക്തത വരേണ്ടത് ആവശ്യമാണ്. ഈ അധ്യായത്തിൽ സ്ട്രക്ചേർഡ് ക്വറി ലാംഗ്വേജിനെക്കുറിച്ചും (SQL) അതിന്റെ പ്രവർത്തനങ്ങളെക്കുറിച്ചും വിവരിക്കുന്നു. റിലേഷണൽ ഡാറ്റാബേസ് മാനേജ്മെന്റ് സിസ്റ്റം (RDBMS) പ്രോഗ്രാമുകളായ MySQL, ഒറാക്കിൾ, സൈബേസ്, ഇൻഫോമിക്സ്, പോസ്റ്റ്ഗ്രെസ്സ്, SQL സെർവർ, MS ആക്സസ് തുടങ്ങിയവയുടെ അടിസ്ഥാന ഡാറ്റാബേസ് SQL ആണ്. SQL പ്രാവർത്തികമാക്കുന്നതിന് ജനപ്രിയ സ്വതന്ത്ര RDBMS ആയ MySQL നാം ഉപയോഗിക്കുന്നു.

9.1 സ്ട്രക്ചേർഡ് ക്വറി ലാംഗ്വേജ് (SQL) (Structured Query Language)

റിലേഷണൽ ഡാറ്റാബേസ് മാനേജ്മെന്റ് സിസ്റ്റത്തിലെ ഡാറ്റ കൈകാര്യം ചെയ്യാൻ SQL ഉപയോഗിക്കുന്നു. റിലേഷണൽ ഡാറ്റാബേസുമായി എളുപ്പത്തിലും കാര്യക്ഷമമായും സമ്പർക്കം പുലർത്താനുള്ള ഒരു വഴിയാണ് SQL. SQLന് ധാരാളം പതിപ്പുകൾ ഉണ്ട്. IBMന്റെ സാൻജോസ് ലബോറട്ടറിയിലെ (ഇപ്പോൾ അൽമാൻഡൻ റിസർച്ച് സെന്റർ) ഡൊണാൾഡ് ഡി ചാംബെർളിനും, റെയ്മണ്ട്.എഫ്.ബോയ്സും ചേർന്ന് 1970 ലാണ് ഇതിന്റെ ആദ്യ പതിപ്പ് വികസിപ്പിച്ചത്. ഈ ഭാഷയെ യഥാർഥത്തിൽ സ്ട്രക്ചർഡ് ഇംഗ്ലീഷ് ക്വറി ലാംഗ്വേജ് (sequel) എന്നാണ് വിളിച്ചിരുന്നത്. പിന്നീട് SQL എന്ന പേരിലേക്ക് മാറ്റി. 1986ൽ അമേരിക്കൻ നാഷണൽ സ്റ്റാൻഡേർഡ് ഇൻസ്റ്റിറ്റ്യൂട്ട് (ANSI) SQL സ്റ്റാൻഡേർഡ് പ്രസിദ്ധീകരിച്ചു.

റിലേഷണൽ ഡാറ്റാബേസ് സിസ്റ്റം എന്നത് പട്ടികകളുടെ (റിലേഷൻ) ഘടനാപരമായ ശേഖരമാണ്. ഡാറ്റ ഈ പട്ടികകളിലാണ് ശേഖരിക്കുന്നത്. ടേബിളുകൾ സവിശേഷമായി തിരിച്ചറിയുന്നത് പേരുകൾ ഉപയോഗിച്ചാണ്. അത് വരികളും നിരകളും ഉൾകൊള്ളുന്നു. ഒരു പട്ടികയിൽ ഒരു വരി പ്രതിനിധീകരിക്കുന്നത് ബന്ധപ്പെട്ട ഡാറ്റകളുടെ ശേഖരമാണ്. വരിയെ ടൂപ്പ്ൾ (Tuple) എന്നും നിരയെ ആട്രിബ്യൂട്ട് (Attribute) എന്നും വിളിക്കുന്നുവെന്ന് നമുക്കറിയാം.

നമുക്കു ചെയ്യാം

student എന്ന പട്ടിക പരിശോധിക്കുക (പട്ടിക 9.1) തുടർന്ന് താഴെ കൊടുത്തിരിക്കുന്ന ചോദ്യങ്ങൾക്ക് ഡാറ്റാബേസുമായി ബന്ധപ്പെട്ട പദാവലിയുടെ അടിസ്ഥാനത്തിൽ ഉത്തരമെഴുതുക.

Adm_no	Name	Gender	Date_Birth	Income	Course
1001	Alok	M	02/10/1998	24000	Science
1002	Nike	M	26/01/1998	35000	Science
1003	Bharath	M	01/01/1999	45000	Commerce
1004	Virat	M	05/12/1998	22000	Science
1005	Meera	F	15/08/1998		Science
1006	Divakar	M	21/02/1998		Humanities

പട്ടിക 9.1: സ്റ്റുഡന്റ് പട്ടിക

i) ഈ പട്ടികയുടെ കാർഡിനാലിറ്റി _____________ ആണ്.

ii) ഈ പട്ടികയുടെ ഡിഗ്രി _____________ ആണ്.

iii) പട്ടികയിലെ വിവിധ വരികൾ (ടൂപ്പ്ൾ) എഴുതുക.

iv) പട്ടികയിലെ വിവിധ ആട്രിബ്യൂട്ടുകൾ എഴുതുക.

v) 'Course' എന്ന ആട്രിബ്യൂട്ടിന്റെ ഡൊമൈനിലെ വിലകൾ ഏതെല്ലാം?

RDBMS പ്രാവർത്തികമാക്കാനുള്ള ശക്തമായ ഒരു ഉപകരണമാണ് SQL. അത് പട്ടിക നിർമ്മിക്കുക, ഡാറ്റ ചേർക്കുക, ഡാറ്റ കൈകാര്യം ചെയ്യുക, ഡാറ്റ നീക്കം ചെയ്യുക, പട്ടികയുടെ ഘടന മാറ്റുക, പട്ടിക നീക്കം ചെയ്യുക തുടങ്ങിയവയ്ക്കുള്ള സൗകര്യങ്ങൾ നൽകുന്നു.

9.1.1 SQL ന്റെ സവിശേഷതകൾ (Features of SQL)

സ്ട്രക്ചേർഡ് ക്വറി ലാംഗ്വേജ് എന്നത് ANSI/ISO സ്റ്റാൻഡേർഡ് പ്രകാരമുള്ള, ഡാറ്റാബേസ് നിർമ്മിക്കുന്നതിനുള്ള ഒരു ഭാഷയാണ്. ക്വറി എന്നത് ഡാറ്റാബേസിനോടുള്ള അഭ്യർഥനയാണ്. മുൻപ് പറഞ്ഞിട്ടുള്ള എല്ലാ റിലേഷണൽ പ്രവൃത്തികളും അതിലൂടെ നമുക്ക് ചെയ്യാൻ സാധിക്കും. താഴെ പറയുന്ന സവിശേഷതകൾ കാരണം ക്വറികൾ നിർമ്മിക്കുന്നതിന് SQL ഫലപ്രദമാണ്.

- SQL എന്നത് ഒരു റിലേഷണൽ ഡാറ്റാബേസ് ഭാഷയാണ്. C, C++ എന്നിവ പോലെയുള്ള പ്രോഗ്രാമിങ് ഭാഷ അല്ല.
- അത് ലളിതവും, വഴക്കമുള്ളതും, സുശക്തവുമാണ്.
- അതിൽ പട്ടിക നിർമ്മിക്കുന്നതിനും, മാറ്റം വരുത്തുന്നതിനും, പട്ടികയിൽ ഡാറ്റ ചേർക്കുന്നതിനും, ഡാറ്റ കൈകാര്യം ചെയ്യുന്നതിനും ഉള്ള കമാൻഡുകൾ ഉണ്ട്.
- അത് ജനപ്രിയ RDBMS സോഫ്റ്റ് വെയറുകളായ Oracle, SQLServer, MySQL, MS Access, Sybase, Infomix തുടങ്ങിയവയ്ക്ക് മാർഗനിർദേശങ്ങൾ നൽകുന്നു.
- SQL ഒരു നോൺപ്രോസിഡറൽ ഭാഷയാണ്. ഏത് ഡാറ്റ തിരിച്ചെടുക്കണം, നീക്കം ചെയ്യണം, കൂട്ടിച്ചേർക്കണം അതിനുപരിയായി എങ്ങനെ ഈ പ്രവർത്തനങ്ങൾ ചെയ്യണം എന്ന് അത് വിവരിക്കുന്നു.
- ഡാറ്റയുടെ സുരക്ഷ ഉറപ്പാക്കുന്നതിന്റെ ഭാഗമായി ഡാറ്റാബേസിൽ നിന്നോ അല്ലെങ്കിൽ പട്ടികകളിലോ മാറ്റം വരുത്തുന്നതിന് വിവിധ ഉപയോക്താക്കൾക്ക് (Users) അനുമതി നൽകുന്നതിനും അത് റദ്ദ് ചെയ്യുന്നതിനും ഉള്ള സൗകര്യം SQL നൽകുന്നു.
- വ്യൂ എന്ന ആശയം ലഭ്യമാക്കുന്നു (ആശയം പിന്നീട് ഈ അധ്യായത്തിൽ ചർച്ച ചെയ്യും).

9.1.2 SQL ന്റെ ഘടകങ്ങൾ (Components of SQL)

SQL ന് മൂന്ന് ഘടകങ്ങളാണ് ഉള്ളത്. ഇത് ഡാറ്റ നിർവചന ഭാഷ (DDL), ഡാറ്റ കൈകാര്യം ചെയ്യുന്ന ഭാഷ (DML), ഡാറ്റ നിയന്ത്രണ ഭാഷ (DCL) എന്നിവയാണ്. ഇനി നമുക്ക് മൂന്ന് ഘടകങ്ങളെക്കുറിച്ചും RDBMS വികസിപ്പിക്കുന്നതിൽ ഇവയ്ക്കുള്ള പങ്ക് എന്താണെന്നും ചർച്ച ചെയ്യാം.

ഡാറ്റ നിർവചന ഭാഷ (DDL) (Data Definition Language)

പട്ടിക 9.1 പരിഗണിക്കുക (student പട്ടിക). ഇതുപോലെ ഒരു പട്ടിക നമുക്ക് എങ്ങനെ നിർമ്മിക്കാം? ഈ പട്ടികയിൽ പുതിയ നിര എങ്ങനെ കൂട്ടിച്ചേർക്കും? ഒരു ഡാറ്റാബേസിൽ നിന്ന് ഒരു പട്ടിക എങ്ങനെ ഒഴിവാക്കും? ഈ ചോദ്യങ്ങൾക്കെല്ലാം DDL പരിഹാരം നൽകും.

DDL എന്നത് RDBMSന്റെ സ്കീമാ (ഘടന) നിർവഹണം കൈകാര്യം ചെയ്യുന്നതിനുള്ള കമാൻഡുകൾ നൽകുന്ന SQL ന്റെ ഘടകമാണ്. DDL കമാൻഡുകൾ ഉപയോഗിക്കുന്നത് ഡാറ്റാബേസ് ഒബ്ജക്ടുകളായ പട്ടികകൾ, വ്യൂകൾ, കീകൾ എന്നിവ നിർമ്മിക്കാനും, പുതുക്കുവാനും, ഒഴിവാക്കുവാനും വേണ്ടിയാണ്. സാധാരണ DDL കമാൻഡുകൾ ആണ് CREATE, ALTER, DROP എന്നിവ.

ഡാറ്റ കൈകാര്യം ചെയ്യുന്ന ഭാഷ (DML) (Data Manipulation Language)

പട്ടിക 9.1 ൽ നമുക്ക് വിവിധതരം ടൂപ്പിളുകൾ (വരികൾ അല്ലെങ്കിൽ റെക്കോർഡുകൾ) കാണാൻ കഴിയും. ഈ ടൂപ്പിളുകൾ എങ്ങനെയാണ് ഒരു പട്ടികയിൽ ചേർക്കുന്നത്? ഒരു വിദ്യാർഥിയുടെ കുടുംബമാസവരുമാനം പുതുക്കണമെന്ന് വിചാരിക്കുക. ഇത് സാധ്യ മാണോ? പട്ടികയിൽനിന്ന് ഒരു കുട്ടിയുടെ റെക്കോർഡ് എങ്ങിനെ നീക്കം ചെയ്യും? ഡാറ്റ കൈകാര്യം ചെയ്യുന്ന ഭാഷ (DML) ഇത്തരം പ്രവർത്തനങ്ങൾക്കുള്ള കമാൻഡുകൾ നൽകുന്നു.

ഒരു കൂട്ടം കമാൻഡുകൾ നൽകുന്നതിലൂടെ ഡാറ്റാബേസ് സിസ്റ്റവുമായി കാര്യക്ഷമമായ ഉപയോക്തൃ ഇടപെടൽ നടത്തുന്നതിനുള്ള SQLന്റെ ഘടകമാണ് DML. പട്ടികകളിലേക്ക് ഡാറ്റ ചേർക്കാനും നിലവിലുള്ള ഡാറ്റ വീണ്ടെടുക്കാനും, പട്ടികകളിൽ നിന്നുള്ള ഡാറ്റ ഒഴിവാക്കുവാനും ശേഖരിച്ച ഡാറ്റ പരിഷ്ക്കരിക്കാനും ഉപയോക്താക്കളെ DML അനുവ ദിക്കുന്നു. സാധാരണമായി ഉപയോഗിക്കുന്ന DML കമാൻഡുകളാണ് SELECT, INSERT, UPDATE, DELETE എന്നിവ.

ഡാറ്റ നിയന്ത്രണ ഭാഷ (DCL) (Data Control Language)

ഡാറ്റ നിയന്ത്രണഭാഷ ഉപയോഗിക്കുന്നത് ഡാറ്റാബേസ് വിനിയോഗം നിയന്ത്രിക്കുന്നതി നാണ്. കാരണം ഡാറ്റാബേസിന്റെ സുരക്ഷിതത്വം വളരെ പ്രധാനപ്പെട്ടതാണ്. ഡാറ്റാകമ്മി റ്റിങ്ങും അഡ്മിനിസ്ട്രേറ്റിങ്ങ് പ്രിവിലേജും അടക്കമുള്ള ഡാറ്റ നിയന്ത്രണ കമാൻഡുകൾ അടങ്ങുന്നതാണ് DCL. GRANT, REVOKE എന്നിവ DCL ന്റെ ഭാഗമായി ഉപയോഗി ക്കുന്നു.

GRANT : ഒരു ഡാറ്റാബേസിൽ ഉപയോക്താവിന് (User) പ്രത്യേക അവകാശങ്ങൾ നൽകുന്നു.

REVOKE : ഉപയോക്താവിന് GRANTഉപയോഗിച്ചു നൽകിയ പ്രത്യേക അവകാശ ങ്ങൾ പിൻവലിക്കുന്നു.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. SQLന്റെ പൂർണരൂപം____________________ ആണ്
2. SQLന്റെ മൂന്നു ഘടകങ്ങൾ ഏതെല്ലാം?
3. SQL ഉപയോഗിക്കുന്നത്
   a. ഡാറ്റാബേസിന്റെ ഘടന ഉണ്ടാക്കാൻ.
   b. ഡാറ്റാബേസ് ക്വറി മാത്രം നിർമ്മിക്കാൻ.
   c. ഡാറ്റാബേസിലെ ഡാറ്റയിൽ മാറ്റം വരുത്താൻ.
   d. മുകളിൽ പറഞ്ഞവയെല്ലാം
4. SQL എന്നത്
   a. ഒരു പ്രോഗ്രാമിംഗ് ഭാഷയാണ് b. ഒരു ഓപ്പറേറ്റിംഗ് സിസ്റ്റമാണ്
   c. ഡാറ്റ കൈകാര്യം ചെയ്യാനുള്ള ഭാഷയാണ് d. ഒരു DBMS ആണ്
5. താഴെപ്പറയുന്നവയിൽ RDBMS പാക്കേജ് അല്ലാത്തത് ഏത് ?
   a. ORACLE b. SQL SERVER c. MySQL d. HTML

9.2 Working on MySQL

അമേരിക്കൻ നാഷണൽ സ്റ്റാൻഡേർഡ് ഇൻസ്റ്റിറ്റ്യൂട്ട് (ANSI) 1986 ലും ഇന്റർനാഷണൽ ഓർഗനൈസേഷൻ ഫോർ സ്റ്റാൻഡേർഡൈസേഷൻ (ISO) 1987 ലും SQL ന്റെ നിലവാരം നിശ്ചയിച്ചു. 1986 മുതൽ ഘടനയിൽ കുറച്ചു മാറ്റങ്ങൾ വരുത്തി SQL സ്റ്റാൻഡേർഡ് പരിഷ്ക്കരിച്ചു. പിന്നീട് ഈ സ്റ്റാൻഡേർഡ് പല പ്രാവശ്യമായി പരിഷ്ക്കരിക്കുകയും പുതിയ പതിപ്പുകൾ ഇറക്കുകയും ചെയ്തു. SQL ഡാറ്റാബേസ് ക്വറി ഭാഷയുടെ ISO ANSI സ്റ്റാൻഡേർഡിന്റെ ഏഴാമത്തെ പതിപ്പാണ് SQL:2011. ഡിസംബർ 2011 ലാണ് ഇത് ഔദ്യോഗികമായി അംഗീകരിക്കപ്പെട്ടത്. ഇത്തരം മാനദണ്ഡങ്ങൾ നിലനിൽക്കുന്നുണ്ടെങ്കിലും വ്യത്യസ്ത ഡാറ്റാബേസ് സോഫ്റ്റ്‌വെയർ പാക്കേജുകൾ ANSI SQL സ്റ്റാൻഡേർഡ് അനുസരിച്ചു അവരുടേതായ പതിപ്പുകൾ ഇറക്കുന്നുണ്ട്. അതിനാൽ മിക്ക SQL കോഡുകളും വ്യത്യസ്ത ഡാറ്റാബേസ് സോഫ്റ്റ്‌വെയറുകളിൽ ചെറിയ മാറ്റങ്ങളോടെയല്ലാതെ പ്രവർത്തിക്കുകയില്ല. ഈ അധ്യായത്തിൽ SQL പ്രവർത്തിക്കുന്നത് ഓപ്പൺ സോഴ്സ് സോഫ്റ്റ്‌വെയറായ MySQL ഉപയോഗിച്ചാണ്.

MySQL സൗജന്യവും, വളരെ വേഗം പ്രവർത്തിക്കുന്നതും, എളുപ്പത്തിൽ ഉപയോഗിക്കാൻ സാധിക്കുന്നതുമായ ഒരു RDBMS ആണ്

- അത് ഡാറ്റാബേസിന് ശക്തമായ സുരക്ഷ നൽകുന്നു.
- ഇത് എളുപ്പത്തിൽ ഉപയോഗിക്കാവുന്നതും പല ഓപ്പറേറ്റിംഗ് സിസ്റ്റങ്ങളിലും PHP, PERL, C, C++, JAVA തുടങ്ങിയ ഭാഷകളിൽ പ്രവർത്തിക്കുന്നതുമാണ്.

- MySQL വലിയ അളവിലുള്ള ഡാറ്റകളിൽ വേഗത്തിലും ഫലപ്രദമായും പ്രവർത്തിക്കുന്നു.
- ഇത് വെബ് ഡെവലപ്മെന്റിന് ഉപയോഗിക്കുന്ന ജനപ്രിയ പ്രോഗ്രാമിംഗ് ഭാഷകളിൽ ഒന്നായ PHP ക്ക് ഏറ്റവും അനുയോജ്യമാണ്.

1995 ൽ Michael "Monty" Widenius and David Axmark ചേർന്നാണ് MySQL വികസിപ്പിച്ചത്. യഥാർഥത്തിൽ ഇത് MySQL AB എന്ന സ്വീഡിഷ് കമ്പനിയുടെ ഉടമസ്ഥതയിലുള്ളതാണ്. പിന്നീട് 2008 ൽ സൺ മൈക്രോ സിസ്റ്റം ഇത് ഏറ്റെടുത്തു. സൺ മൈക്രോ സിസ്റ്റത്തെ 2010 ൽ ഒറാക്കിൾ കോർപ്പറേഷൻ ഏറ്റെടുത്തു. MySQL സാധാരണയായി Linux-Apache-MySQL-PHP, (LAMP), Windows-Apache-MySQL-PHP (WAMP), Mac-Apache-MySQL-PHP (MAMP) എന്ന രീതിയിലാണ് വിന്യസിച്ചിരിക്കുന്നത്. ഓപ്പറേറ്റിംഗ് സിസ്റ്റം ഉൾപ്പെടെ LAMP ലെ എല്ലാ ഘടകങ്ങളും സൗജന്യവും ഓപ്പൺ സോഴ്സുമാണ്. MySQL ന്റെ ഔദ്യോഗിക വെബ് സൈറ്റാണ് www.mysql.com. റഫറൻസിനു MySQL ന്റെ സഹായ ഗ്രന്ഥം (Refence Manual) http://dev.mysq.co/doc ൽ ലഭ്യമാണ്.

9.2.1 MySQL തുറക്കുന്നു (Opening SQL)

mysql> പ്രോംപ്റ്റിൽ കമാൻഡ് നൽകി നമുക്ക് MySQL പ്രവർത്തിപ്പിക്കാൻ സാധിക്കും. ഉബുണ്ടു ലിനക്സിൽ, ഈ പ്രോംപ്റ്റ് ലഭിക്കുന്നതിന് താഴെപറയുന്ന കമാൻഡ് ഉപയോഗിച്ച് ടെർമിനൽ ജാലകം തുറക്കണം.

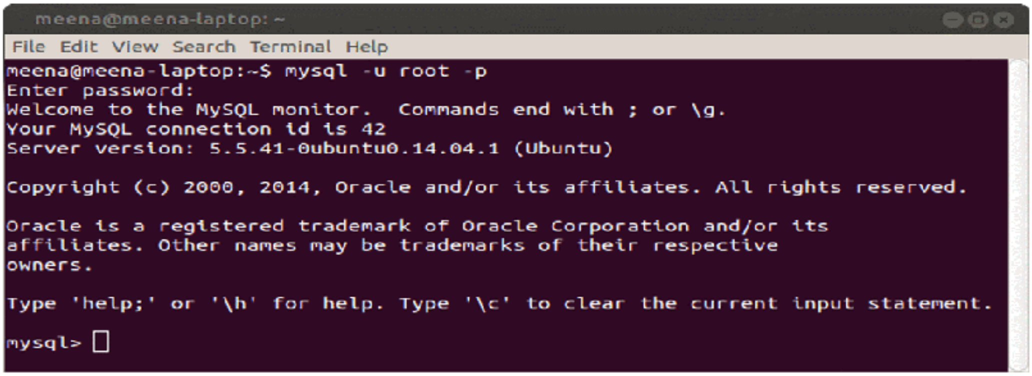
```
Applications -> Accessories -> Terminal
```

MySQL തുടങ്ങുന്നതിന് ടെർമിനൽ ജാലകത്തിൽ താഴെപറയുന്ന കമാൻഡ് നൽകണം

```
mysql -u root -p
```

വിൻഡോസ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിൽ MySQL തുറക്കുന്നത് താഴെപ്പറയുന്ന രീതിയിലാണ്

Start -> Programs -> MySQL -> MySQL Server(Version Number) -> MySQL Commandline client.



ചിത്രം 9.1: ഉബുണ്ടു ലിനക്സ് ടെർമിനൽ ജാലകത്തിൽ MySQL പ്രോംപ്റ്റ്

MySQL തുറക്കുമ്പോൾ അത് പരിശോധനയ്ക്കായി പാസ്‌വേർഡ് ആവശ്യപ്പെടും. ഇൻസ്റ്റലേഷന്റെ സമയത്ത് ഉപയോഗിച്ച അതേ പാസ്‌വേർഡ് തന്നെ ആയിരിക്കണം ഇവിടെ നാം ഉപയോഗിക്കേണ്ടത്. പാസ്‌വേർഡ് പരിശോധനയ്ക്ക് ശേഷം MySQL ന്റെ പ്രോംപ്റ്റ് ചിത്രം 9.1 ൽ കാണുന്ന രീതിയിൽ ലഭിക്കുന്നു.

SQL കേസ് സെൻസിറ്റീവ് അല്ല. അതായത് ഇംഗ്ലീഷിലെ വലിയ അക്ഷരങ്ങൾ ഉപയോഗിച്ചോ, ചെറിയ അക്ഷരങ്ങൾ ഉപയോഗിച്ചോ, രണ്ടും കൂടിച്ചേർന്നോ കമാൻഡുകൾ നൽകാൻ സാധിക്കും. എന്നാൽ ഇതിനു ശേഷം മറ്റ് ടെക്സ്റ്റുകളിൽ നിന്ന് SQL കമാൻഡുകളെയും, കീ വേർഡുകളെയും വേർതിരിച്ചറിയുന്നതിനായി നാം ചില രീതികൾ ഉപയോഗിക്കും. കമാൻഡുകൾക്കും കീവേർഡുകൾക്കും വലിയ അക്ഷരങ്ങൾ (Upper Case Letters) ഉപയോഗിക്കുന്നു. എന്നാൽ ഉപയോക്താവ് നിഷ്കർഷിച്ച വാക്കുകളായ (User defined) ടേബിളിന്റെ പേര്, നിരകളുടെ പേര് തുടങ്ങിയവ ചെറിയ അക്ഷരത്തിൽ (Lower Case Letters) സൂചിപ്പിക്കുന്നു. കമാൻഡുകളും ഔട്ട്പുട്ടുകളും (പ്രതികരണങ്ങളും) ഒരു ടെക്സ്റ്റ് ഫയലിനകത്ത് tee എന്ന കമാൻഡ് നൽകി സൂക്ഷിക്കാവുന്നതാണ്. ഉദാഹരണത്തിന്, tee E:\output.txt എന്ന നിർദേശത്തിന്റെ പ്രവർത്തനത്തിന് ശേഷം സ്ക്രീനിൽ പ്രത്യക്ഷപെടുന്നതെല്ലാം output.txt എന്ന ഫയൽ നിർമ്മിച്ച് E:ഡ്രൈവിൽ സംഭരിക്കുന്നു. ഈ അധ്യായത്തിൽ ഈ ഫയലിൽ സൂക്ഷിച്ചിട്ടുള്ള ഔട്ട്പുട്ടുകൾ ചിത്രരൂപേണ കൊടുത്തിരിക്കുന്നു.

ഉപയോക്താവിൽ നിന്ന് ഏതു തരം ക്വറിയും സ്വീകരിക്കാൻ MySQL തയാറാണ് എന്ന സന്ദേശമാണ് പ്രോംപ്റ്റ് നമുക്ക് നൽകുന്നത്. ഈ പ്രോംപ്റ്റിൽ നമുക്ക് ക്വറികൾ നൽകാവുന്നതാണ്.

MySQLൽ നിന്ന് പുറത്തുകടക്കാൻ QUIT അല്ലെങ്കിൽ EXIT എന്ന് കമാൻഡ് പ്രോംപ്റ്റിൽ നൽകണം.

```
mysql> EXIT;
```

9.2.2 MySQL ൽ ഒരു ഡാറ്റാബേസ് നിർമ്മിക്കുന്നു (Creating a database)

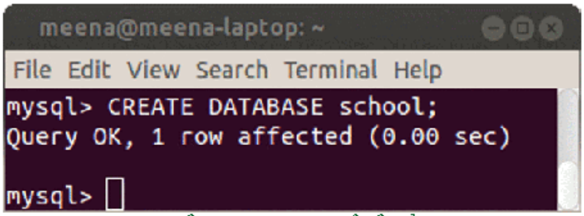
ഡാറ്റയിൽ പ്രവർത്തിക്കുന്നതിന് മുൻപ് നാം ഒരു ഡാറ്റാബേസ് സൃഷ്ടിക്കേണ്ടതുണ്ട്. ഡാറ്റാബേസ് എന്നത് പട്ടിക (Table) സംഭരിക്കുന്നതിനുള്ള ഒരു പെട്ടി പോലെയാണ്. MySQL ൽ ഡാറ്റാബേസ് ഉണ്ടാക്കുന്നതിന് CREATE DATABASE എന്ന കമാൻഡ് ഉപയോഗിക്കുന്നു. അതിന്റെ ഘടന താഴെപറയുന്ന രീതിയിലാണ്:

```
CREATE DATABASE <database_name>;
```

ഡാറ്റാബേസ് ഉണ്ടാക്കുമ്പോൾ താഴെ പറയുന്ന വസ്തുതകൾ ശ്രദ്ധിക്കേണ്ടതാണ്

- വാക്യഘടനയിൽ കൊടുത്ത <database_name> എന്നത് നിർമ്മിക്കേണ്ട ഡാറ്റാബേസിന്റെ പേരാണ്.
- ഡാറ്റാബേസിന്റെ പേര് സാധിക്കുന്നിടത്തോളം അർഥപൂർണവും വിവരണാത്മകവും ആയിരിക്കണം.
- <database_name> അനന്യമായിരിക്കണം. ഒരേ പേരിലുള്ള രണ്ട് ഡാറ്റാബേസുകൾ MySQL സെർവറിൽ ഉണ്ടാക്കാൻ സാധിക്കില്ല.

school എന്ന പുതിയ ഡാറ്റാബേസ് ഉണ്ടാക്കി നമ്മളുടെ ഡാറ്റാബേസ് പ്രവർത്തനങ്ങൾ ആരംഭിക്കാം. ചിത്രം 9.2ൽ കാണിച്ചിരിക്കുന്നത് ഈ കമാൻഡിന്റെ പ്രവർത്തനത്തിനുശേഷമുള്ള സ്ക്രീനാണ്. MySQL കമാൻഡ് പ്രോംപ്റ്റും, കമാൻഡും ആദ്യത്തെ വരിയിൽ കാണാൻ കഴിയും. ഈ കമാൻഡിനെ പ്രവർത്തിപ്പിച്ചശേഷം നൽകുന്ന സന്ദേശമാണ് *ചിത്രം 9.2ൽ കാണുന്നത്. (ഇനി മുതൽ മുകളിൽ കാണുന്ന രീതിയിലുള്ള സ്ക്രീൻ ഷോട്ടുകൾ ഒഴിവാക്കി പകരം നിർദിഷ്ട ഓപ്പറേഷനായുള്ള നിർദേശം പ്രത്യേക അക്ഷരങ്ങളിൽ (Seperate fonts) അവതരിപ്പിക്കുന്നതാണ്)*



ചിത്രം 9.2: ഒരു കമാൻഡിന്റെ പ്രവർത്തനത്തിനുശേഷംMySQL ജാലകം

9.2.3 ഡാറ്റാബേസ് തുറക്കുന്നു (Opening database)

ഒരു ഡാറ്റാബേസിൽ പ്രവർത്തനം നടത്തുന്നതിന് നാം അത് തുറക്കേണ്ടതുണ്ട്. നാം ഒരു ഡാറ്റാബേസ് തുറക്കുമ്പോൾ അത് MySQL സെർവറിലെ സജീവ ഡാറ്റാബേസ് *(Active database)* ആക്കി മാറ്റണം. ഇങ്ങനെ ഒരു ഡാറ്റാബേസ് തുറക്കുന്നതിന് USE എന്ന കമാൻഡാണ് MySQL ൽ നൽകേണ്ടത്. അതിന്റെ വാക്യഘടന താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിലാണ്:

```
USE <database_name>;
```

ഇനി നമുക്ക് school എന്ന ഡാറ്റാബേസ് തുറക്കുന്നതിന് ചുവടെ കൊടുത്തിരിക്കുന്ന കമാൻഡ് നൽകാം.

```
USE school;
```

ഈ കമാൻഡിന്റെ പ്രവർത്തനത്തിന് ശേഷമുള്ള പ്രതികരണം താഴെ കൊടുത്തിരിക്കുന്നു:

```
Database changed
```

ഇപ്പോൾ school എന്ന ഡാറ്റാബേസാണ് സജീവ ഡാറ്റാബേസ്. ഇത് അർഥമാക്കുന്നത് വിവിധ DDL, DML, DCL കമാൻഡുകൾ ഇനി മുതൽ പ്രവർത്തിപ്പിക്കുന്നത് school ഡാറ്റാബേസിനെ അടിസ്ഥാനമാക്കിയാണ്. ഡാറ്റാബേസിന്റെ നിലനിൽപ്പ് ഇനി നമുക്ക് പരിശോധിക്കാം. ഒരു ഡാറ്റാബേസ് ഉണ്ടോ ഇല്ലയോ എന്നറിയാൻ `SHOW DATA BASES` എന്ന കമാൻഡ് ഉപയോഗിക്കാം. അത് കംപ്യൂട്ടറിലെ എല്ലാ ഡാറ്റാബേസുകളും പട്ടികപ്പെടുത്തുന്നു.

```
SHOW DATABASES;
```

ഈ കമാൻഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 9.3 ൽ കൊടുത്തിരിക്കുന്നു.

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| school             |
| test               |
+--------------------+
4 rows in set (0.00 sec)
```

ചിത്രം 9.3: SHOW DATABASES കമാൻഡിന്റെ ഔട്ട്പുട്ട്

9.2.4 SQL ലെ ഡാറ്റ ഇനങ്ങൾ (Data types in SQL)

ഒരു പട്ടികയിലെ നിരയിൽ നൽകുന്ന വിലയുടെ ഇനത്തെ ഡാറ്റ ഇനം നിർവചിക്കുന്നു. ഡാറ്റ ഇനം ഒരു ഡാറ്റയുടെ കൃത്യത ഉറപ്പുവരുത്തുന്നു. ഡാറ്റാബേസ് ഡിസൈനിങ് സമയത്ത് നിരകൾക്ക് കൃത്യമായ ഡാറ്റ ഇനങ്ങൾ

നൽകുവാൻ ശ്രദ്ധിക്കേണ്ടതാണ്. ഉദാഹരണത്തിന് 2 എന്ന സംഖ്യയെ ടെക്സ്റ്റ് ഡാറ്റ ഇനമായി (സ്ട്രിംഗ്) നൽകിയാൽ അതിനെ ഗണിത പ്രക്രിയയിൽ ഉപയോഗിക്കാൻ കഴിയില്ല. നേരെമറിച്ച് അതേ സംഖ്യ പൂർണസഖ്യ നിരയിൽ നൽകിയാൽ അതിനെ ഗണിതത്തിന് ഉപയോഗിക്കാം. അതുകൊണ്ട് SQL ഡാറ്റ ഇനങ്ങൾ എന്ന ആശയത്തെ അവ പ്രതിനിധാനം ചെയ്യുന്ന ഡാറ്റ ഇനങ്ങൾ, അവ ഓരോന്നും ഉൾക്കൊള്ളുന്ന വിലകളുടെ പരിധി മുതലായ ഘടകങ്ങളുടെ അടിസ്ഥാനത്തിൽ നമുക്ക് മനസ്സിലാക്കാം. SQL ന്റെ വ്യത്യസ്ത പതിപ്പുകളിൽ ഡാറ്റ ഇനങ്ങൾ വ്യത്യസ്തമായിരിക്കും. ഡാറ്റ ഇനം SQL ന്റെ വ്യത്യസ്ത പതിപ്പുകളിൽ വ്യത്യാസമുണ്ട്.

ഡാറ്റ ഇനത്തെ മൂന്നായി തരംതിരിക്കാം. സംഖ്യാഡാറ്റ ഇനം, സ്ട്രിംഗ്ഡാറ്റ ഇനം, തീയതി സമയം എന്നിവ സൂചിപ്പിക്കുന്ന ഡാറ്റ ഇനം എന്നിവയാണവ. സംഖ്യാപരമായ മൂല്യങ്ങളായ 7,100,234,-456,0 തുടങ്ങിയ സംഖ്യകളെ സംഖ്യാ ഡാറ്റ ഇനം ഉപയോഗിച്ച് പ്രതിനിധാനം ചെയ്യാം. "Aleena" (ഒരു വിദ്യാർഥിയുടെ പേര്), "Kerala" (ഒരു സംസ്ഥാനത്തിന്റെ പേര്), "F" (ലിംഗഭേദത്തെ സൂചിപ്പിക്കാൻ) എന്നിവ സ്ട്രിംഗ് ഡാറ്റ ഇനം ആണ്. '01-01-2020', '23:34:3' തുടങ്ങിയവ തീയതിയും സമയവും സൂചിപ്പിക്കാനുള്ള ഡാറ്റ ഇനം ഉപയോഗിച്ച് പ്രതിനിധാനം ചെയ്യാം.

a. സംഖ്യാഡാറ്റ ഇനങ്ങൾ

സംഖ്യാഡാറ്റ ഇനത്തിന്റെ മൂല്യങ്ങളെ ഏതെങ്കിലും സാധാരണ സംഖ്യ പോലെ ഉപയോഗിക്കാം. അവയിൽ ഗണിതക്രിയകളായ സങ്കലനം, വ്യവകലനം, ഗുണനം, ഹരണം എന്നിവ ചെയ്യാം. MySQL ൽ സാധാരണയായി ഉപയോഗിക്കുന്ന സംഖ്യാഡാറ്റ ഇനങ്ങളാണ് `INT` അല്ലെങ്കിൽ `INTEGER` ഉം, `DEC` അല്ലെങ്കിൽ `DECIMAL` ഉം.

(i) `INT` അല്ലെങ്കിൽ `INTEGER`

നമുക്കറിയാവുന്നതുപോലെ പൂർണസംഖ്യകൾ എന്നാൽ ദശാംശ ഭാഗം ഇല്ലാത്ത സംഖ്യകളാണ്. അവ അധിസംഖ്യയോ, പൂജ്യമോ, ന്യൂനസംഖ്യയോ ആവാം. MySQLൽ ഒരു പൂർണസംഖ്യയെ പ്രതിനിധാനം ചെയ്യാൻ `INT` അല്ലെങ്കിൽ `INTEGER` എന്ന ഡാറ്റ ഇനം ഉപയോഗിക്കാം. ഉദാഹരണത്തിന് 69, 0, 112 തുടങ്ങിയ ഡാറ്റകൾ INT ഡാറ്റ ഇനങ്ങളാണ്.

(ii) `DEC` അല്ലെങ്കിൽ `DECIMAL`

ദശാംശ ഭാഗത്തോടുകൂടിയ സംഖ്യകളെ `DEC` അല്ലെങ്കിൽ `DECIMAL` എന്ന ഡാറ്റ ഇനം കൊണ്ട് പ്രതിനിധീകരിക്കാം. ഈ ഡാറ്റ ഇനത്തിന്റെ മാതൃകാ രൂപം `DECIMAL(size,D)` അല്ലെങ്കിൽ `DEC(size,D)` എന്നാണ്. ഇവിടെ `size` എന്ന ഘടകം, ദശാംശ ഭാഗം ഉൾപ്പെടുന്ന മൂല്യത്തിന്റെ മൊത്തം എണ്ണം സൂചിപ്പിക്കുന്നു. `D` എന്നത് സൂചിപ്പിക്കുന്നത് പൂർണ്ണ സംഖ്യകൾക്കു ശേഷമുള്ള അക്കങ്ങളുടെ എണ്ണമാണ്. ഉദാഹരണത്തിന് `DEC (5,2)` അല്ലെങ്കിൽ `DECIMAL(5,2)` എന്നതിൽ 5 സംഖ്യയുടെ കൃത്യതയും (precision), 2 അതിന്റെ അളവും (Scale) സൂചിപ്പിക്കുന്നു. ഈ സവിശേഷതയോടുകൂടി ഒരു നിരയിൽ ഏറ്റവും കൂടിയത് അഞ്ച് അക്കം സൂക്ഷിക്കാം അതിൽ 2 അക്കങ്ങൾ ദശാംശ ബിന്ദുവിന് ശേഷമായിരിക്കും. ഈ ഡാറ്റ ഇനം ഉപയോഗിച്ച് ഒരു നിരയിൽ നൽകാൻ സാധിക്കുന്ന അക്കങ്ങളുടെ പരിധി -999.99 മുതൽ 999.99 വരെയാണ്. MySQL ലിലെ സംഖ്യാഡാറ്റ ഇനങ്ങളുടെ ഒരു അവലോകനമാണ് പട്ടിക 9.2 ൽ നൽകിയിരിക്കുന്നത്. ഇതിൽ നൽകിയിരിക്കുന്ന മൂല്യങ്ങൾ ഓരോ SQL പതിപ്പിനെയും ആശ്രയിച്ചിരിക്കും എന്ന് ഓർമിക്കുക.

ഡാറ്റാ തരം	ഉപയോഗം	ചിഹ്നത്തോടു കൂടിയത്	ചിഹ്നം ഇല്ലാതെ	സംഭരണ ശേഷി (ബൈറ്റിൽ)
TINY INT	വളരെ ചെറിയ പൂർണ സംഖ്യ	-128 മുതൽ 127	0 മുതൽ 255	1
SMALL INT	ചെറിയ പൂർണ സംഖ്യ	- 32768 മുതൽ 32767	0 മുതൽ 65535	2
MEDIUM INT	ഇടത്തരം വലുപ്പമുള്ള പൂർണസംഖ്യ	-8388608 മുതൽ 8388607	0 മുതൽ 1677215	3
INT	സാധാരണ പൂർണ സംഖ്യാമൂല്യം	- 2147483648 മുതൽ 2147483647	0 മുതൽ 4294967295	4
BIG INT	വലിയ പൂർണ സംഖ്യാമൂല്യം	19 അക്കങ്ങൾ വരെയുള്ള മൂല്യം	264 അക്കങ്ങൾ വരെയുള്ള മൂല്യം	8
FLOAT(M,D)	ദശാംശ സംഖ്യ	പൂർണസംഖ്യ ഭാഗം 24 ആകുന്നതുവരെ		4
DOUBLE(M,D)	ഇരട്ട കൃത്യത യുള്ള ദശാംശ സംഖ്യാമൂല്യം	പൂർണസംഖ്യ ഭാഗം 53 ആകുന്നതുവരെ		8
DECIMAL(M,D)	സൂക്ഷ്മമായ കൃത്യതയുള്ള ദശാംശ സംഖ്യാമൂല്യം	സംഖ്യാഡാറ്റ ഇനത്തിനു ഏറ്റവും കൂടിയത് 65 അക്കങ്ങൾ വരെ, അതിൽ 30 അക്കങ്ങൾ ദശാംശ ബിന്ദുവിന് ശേഷം		8

പട്ടിക 9.2: MySQL ലെ സംഖ്യ ഡാറ്റ ഇനങ്ങളും അവയുടെ പ്രത്യേകതകളും

b. സ്ട്രിംഗ് ഡാറ്റ ഇനങ്ങൾ

സ്ട്രിംഗ് എന്നാൽ അക്ഷരങ്ങളുടെ കൂട്ടമാണ് MySQLലിൽ ഏറ്റവും സാധാരണയായി ഉപയോഗിക്കുന്ന സ്ട്രിംഗ്ഡാറ്റ ഇനങ്ങളാണ് CHARACTER അല്ലെങ്കിൽ CHAR ഉം, VARCHAR ഉം.

(i) CHARACTER അല്ലെങ്കിൽ CHAR

അക്ഷരങ്ങൾ, ലിപികൾ, സംഖ്യകൾ, പ്രത്യേക ചിഹ്നങ്ങൾ മുതലായവ ഇതിൽ ഉൾപ്പെടുന്നു. CHAR ന് ഒരു നിശ്ചിത വലുപ്പമാണ് ഉണ്ടായിരിക്കുക. ഈ ഡാറ്റ ഇനത്തിന്റെ ഘടന CHAR(X) എന്നാണ്. X എന്നത് ഡാറ്റ നിർമ്മിക്കാൻ ഉപയോഗിക്കുന്ന അക്ഷരങ്ങളുടെ

പരമാവധി എണ്ണം സൂചിപ്പിക്കുന്നു X ന്റെ വില 0 നും 255 നും ഇടയിലായിരിക്കും. ഒരു നിരയിലെ നിശ്ചിത വലുപ്പമുള്ളതും, ചെറുതുമായ ഡാറ്റകൾക്കാണ് CHAR പ്രധാനമായും ഉപയോഗിക്കുന്നത്. ഉദാഹരണത്തിന് ഒരു പട്ടികയിലെ Gender എന്ന നിരയിൽ പുരുഷനെ സൂചിപ്പിക്കുന്നതിന് 'M' എന്നും സ്ത്രീയെ സൂചിപ്പിക്കുന്നതിന് 'F' എന്നും നമുക്ക് സംഭരിക്കണമെങ്കിൽ ഈ നിരയെ CHAR ഡാറ്റ ഇനമായി പ്രഖ്യാപിക്കുന്നതാണ് നല്ലത്. സംഭരിക്കേണ്ട ഡാറ്റയ്ക്ക് ആവശ്യമില്ലെങ്കിലും നിർദേശിക്കപ്പെട്ട അത്രയും സ്ഥലം എപ്പോഴും ഉപയോഗിക്കുന്നു. ഡാറ്റയിലുള്ള അക്ഷരങ്ങൾ നിരയുടെ പ്രഖ്യാപിത വലുപ്പത്തെക്കാൾ കുറവാണെങ്കിൽ ബാക്കിയുള്ള ഭാഗം ശൂന്യസ്ഥലം (spacebar character) ഉപയോഗിച്ചു പൂർണമാക്കുന്നതാണ്. എന്നാൽ പട്ടികയിൽ നിന്ന് ഈ മൂല്യം വീണ്ടെടുക്കുമ്പോൾ, ബാക്കിയുള്ള എല്ലാസ്ഥലങ്ങളും ഒഴിവാക്കപ്പെടുന്നു. CHAR തരത്തിലുള്ള നിരയുടെ വലുപ്പം 1 ആണെങ്കിൽ, വലുപ്പം സൂചിപ്പിക്കേണ്ടതില്ല, കാരണം CHAR ന്റെ തനത് വലുപ്പം 1 ആണ്.

(ii) VARCHAR (size)

VARCHAR എന്നത് വിവിധ നീളത്തിലുള്ള സ്ട്രിങ്ങുകളെ സൂചിപ്പിക്കുന്നു. അത് CHARന് സമാനമാണ്. എന്നാൽ ഡാറ്റയ്ക്ക് അനുവദിച്ച സ്ഥലം അതിന്റെ യഥാർഥ വലുപ്പത്തെ ആശ്രയിച്ചിരിക്കും അല്ലാതെ പ്രഖ്യാപിക്കപ്പെട്ട വലുപ്പം അനുസരിച്ചല്ല. ഉദാഹരണമായി ഒരു പട്ടികയിലെ Name എന്ന നിരയിൽ ഡാറ്റ സംഭരിക്കണമെങ്കിൽ ആ നിരയെ VARCHAR തരമായി പ്രഖ്യാപിക്കുന്നതാണ് നല്ലത്. കാരണം ആ നിരയിലെ ഡാറ്റയിൽ വ്യത്യസ്ത എണ്ണം അക്ഷരങ്ങൾ അടങ്ങിയിരിക്കാം. ഒരു സ്ട്രിങ്ങിന്റെ നീളം 0 മുതൽ 65535 വരെ ആകാം. VARCHAR, മെമ്മറി സ്ഥലം മിതമായി ഉപയോഗിക്കുന്നു. എന്തുകൊണ്ടെന്നാൽ, വിലകൾ സംഭരിക്കുമ്പോൾ അതിനോടുകൂടി ശൂന്യസ്ഥലങ്ങൾ ചേർക്കുന്നില്ല. മേൽ വിലാസം, പേരുകൾ തുടങ്ങിയവ ഇത്തരം ഡാറ്റകൾക്ക് ഉദാഹരണങ്ങളാണ്

c. തീയതിക്കും സമയത്തിനുമുള്ള ഡാറ്റ ഇനങ്ങൾ

MySQL ലിൽ തീയതിയും, സമയവും ശേഖരിക്കുന്നതിന് ഡാറ്റ ഇനങ്ങളുണ്ട്. തീയതി സംഭരിക്കുന്നതിന് MySQL ൽ ഉപയോഗിക്കുന്ന ഡാറ്റ ഇനമാണ് DATE. അതുപോലെ സമയം സംഭരിക്കുന്നതിന് TIME ഡാറ്റാ ഇനം ഉപയോഗിക്കുന്നു.

i) DATE

DATE ഡാറ്റ ഇനം ഉപയോഗിക്കുന്നത് തീയതികൾ സംഭരിക്കുന്നതിനാണ്. YYYY-MM-DD എന്ന രൂപത്തിലാണ് MySQL തീയതിയെ സംഭരിക്കുന്നത്. ഇതുപയോഗിച്ച് 1000-01-01 മുതൽ 9999-12-31 വരെ രേഖപ്പെടുത്താവുന്നതാണ്. MySQL ൽ ഒരു രീതിയിൽ തീയതികൾ പ്രദർശിക്കപ്പെടുന്നു എന്നാൽ SQL പ്രസ്താവനകളിൽ നമുക്ക് വിവിധ തരം DATE രൂപഘടനകൾ ഉപയോഗിക്കാം. YYYY-MM-DD എന്നതാണ് അടിസ്ഥാന മാതൃക. എന്നാൽ നമുക്ക് ഏതെങ്കിലും വിരാമചിഹ്നം (Punctuation) തീയതികൾക്കിടയിൽ ഉപയോഗിക്കാൻ സാധിക്കും. ഉദാഹരണത്തിന് '2011-01-24', '2011/01/25', '20111026' തുടങ്ങിയവ സാധുവായ തീയതികളാണ്. മുകളിലുള്ള ഏതെങ്കിലും മാതൃകയിൽ DATE ഡാറ്റ തരത്തിലുള്ള നിരയിലേക്ക് വിലകൾ നമുക്ക് ചേർക്കാവുന്നതാണ്. MySQL ൽ DATE

എല്ലായ്പ്പോഴും year-month-day എന്ന ക്രമത്തിൽ തന്നെ കൊടുത്തിരിക്കണം (ഉദാഹരണത്തിന് '98-09-04')

ii) `TIME`

ഒരു നിരയിൽ സമയ ഡാറ്റ സംഭരിക്കുന്നതിനാണ് MySQL ൽ `TIME` ഡാറ്റ ഇനം ഉപയോഗിക്കുന്നത്. HH:MM:SS അംഗീകൃത മാതൃകയിലാണ് അത് വിലകൾ പ്രദർശിപ്പിക്കുന്നത്. `TIME` ഡാറ്റ ഇനം ഉപയോഗിച്ച് ഒരു പ്രത്യേക സമയം (10 മണിക്കൂർ 5 മിനിറ്റ് 2 സെക്കന്റ് എന്ന പോലെ) സംഭരിക്കുന്നതിനോടൊപ്പം, രണ്ടു സമയത്തിനിടയിലുള്ള ഇടവേള (23 മണിക്കൂറിനെക്കാൾ അധികമുള്ളതും) സംഭരിക്കുന്നതിനും അത് ഉപയോഗിക്കാം.

നമുക്കു ചെയ്യാം

പട്ടിക 9.3 ലെ ഡാറ്റ ഇനം എന്ന നിര അനുയോജ്യമായ ഡാറ്റ ഇനം ഉപയോഗിച്ച് പൂരിപ്പിക്കുക.

മൂല്യം	ഡാറ്റ ഇനം
325.678	
'A'	
'Computer'	
'2016-01-01'	
450	
22:32:45	
456787	

പട്ടിക 9.3: വിലകളുടെ ഡാറ്റ ഇനം

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. __________ കമാൻഡാണ് ഒരു ഡാറ്റാബേസിനെ സജീവമാക്കാൻ ഉപയോഗിക്കുന്നത്
2. കമ്പ്യൂട്ടറിൽ ഉള്ള ഡാറ്റാബേസിന്റെ പേര് എങ്ങനെ കാണാൻ കഴിയും?
3. `CHAR`, `VARCHAR` എന്നീ ഡാറ്റ ഇനങ്ങൾ തമ്മിലുള്ള വ്യത്യാസം എന്ത്?
4. `DATE` ഡാറ്റ ഇനം എങ്ങനെയാണ് മൂല്യങ്ങൾ സംഭരിക്കുന്നത്?
5. `CHAR (5)` എന്ന ഡാറ്റ ഇനത്തിൽ പ്രഖ്യാപിച്ചിട്ടുള്ള ഒരു നിരയിൽ 234 എന്ന സംഖ്യ നമുക്ക് സംഭരിക്കാൻ സാധിക്കുമോ?

9.3 SQL കമാൻഡുകൾ (SQL commands)

ഡാറ്റാബേസിൽ വിവിധ തരം പ്രവർത്തങ്ങൾ ചെയ്യുന്നതിനുള്ള കമാൻഡുകൾ SQLൽ ലഭ്യമാക്കിയിരിക്കുന്നു. നാം മുൻപ് സൂചിപ്പിച്ചതു പോലെ കമാൻഡുകളെ DDL, DML, DCL കമാൻഡുകളെന്നും തരംതിരിച്ചിട്ടുണ്ട് ഏറ്റവും സാധാരണയായി ഉപയോഗിക്കുന്ന

DDL, DML കമാൻഡുകളെക്കുറിച്ച് നമുക്ക് ചർച്ച ചെയ്യാം. ഡാറ്റാബേസിന്റെ ഘടനാപരമായ കാര്യങ്ങൾ ചെയ്യുന്നതിനാണ് DDL കമാൻഡുകൾ ഉപയോഗിക്കുന്നത്. ഒരു പട്ടിക നിർമ്മിക്കുക, അത് പുതുക്കുക, പട്ടിക നീക്കം ചെയ്യുക തുടങ്ങിയ പ്രവർത്തനങ്ങൾ ഇതിൽ ഉൾപ്പെടുന്നു.

ഒരു പട്ടികയുടെ ഉള്ളടക്കവുമായി ബന്ധപ്പെട്ട പ്രവർത്തനങ്ങൾക്കാണ് DML കമാൻഡുകൾ ഉപയോഗിക്കുന്നത്. ഒരു പട്ടികയിൽ റെക്കോർഡുകൾ ചേർക്കുക, വീണ്ടെടുക്കുക, പരിഷ്ക്കരിക്കുക, ഒഴിവാക്കുക എന്നീ പ്രവർത്തനങ്ങൾ ഇതിൽ ഉൾപ്പെടുന്നു. ഒരു പ്രത്യേക വസ്തുവിനെ സംബന്ധിച്ച ഡാറ്റ സംഭരിക്കുന്നതിന് പട്ടിക നിർമ്മിക്കുക, ആവശ്യമായ വിവരം വീണ്ടെടുക്കുക, ആവശ്യമില്ലാത്തവ ഒഴിവാക്കുക എന്നീ പ്രവർത്തനങ്ങൾ ചെയ്യുന്നതിന് ഈ കമാൻഡുകൾ പരിചയപ്പെടുത്തുന്നു.

9.4 പട്ടികയുടെ നിർമ്മാണം (Creating tables)

ഒരു റിലേഷണൽ ഡാറ്റാബേസിന്റെ ഏറ്റവും പ്രധാനപ്പെട്ട ഘടകമാണ് പട്ടികകൾ. ഏതൊരു ഡാറ്റാബേസിന്റെയും പ്രാഥമികമായ ആവശ്യം പട്ടികകളിൽ സംഭരിച്ചിരിക്കുന്ന ഡാറ്റ പരിപാലിക്കുക എന്നതാണ്. ഇപ്പോൾ നമുക്ക് പട്ടിക 9.1 ലുള്ള student എന്ന പട്ടിക പരിഗണിക്കാം. ഒരു കൂട്ടം നിരകളുള്ള ഒരു പട്ടിക നാം എങ്ങനെയാണ് നിർമ്മിക്കുന്നത്? `CREATE TABLE` എന്ന DDL കമാൻഡ് ഉപയോഗിച്ചാണ് പട്ടിക നിർവചിക്കുന്നത്. ഇതിന് പട്ടികയുടെ പേര്, നിരയുടെ പേര്, ഡാറ്റ ഇനം, അവയുടെ വലുപ്പം മറ്റെന്തെങ്കിലും നിബന്ധനകൾ ഉണ്ടെങ്കിൽ അവയും ഉൾപ്പെടുത്തി പട്ടിക നിർവചിക്കേണ്ടതുണ്ട്. ഒരു പട്ടികയിൽ കുറഞ്ഞത് ഒരു നിരയെങ്കിലും ഉണ്ടായിരിക്കണം. `CREATE TABLE` കമാൻഡിന്റെ വാക്യഘടന താഴെക്കൊടുക്കുന്നു.

```
CREATE TABLE <table_name>
(<column_name> <data_type> [<constraint>]
[, <column_name> <data_type> [<constraint>,]
  ..............................
  ..............................    );
```

ഇവിടെ <table_name> എന്നത് പട്ടികയുടെ പേര് സൂചിപ്പിക്കുന്നു. <column_name> എന്നത് പട്ടികയിലെ ഒരു നിരയുടെ പേര് സൂചിപ്പിക്കുന്നു. <constraint> എന്നത് ഒരു നിരയിലെ വിലകളിൽ ഉപയോഗിക്കാവുന്ന നിയമങ്ങൾ ഏതെന്ന് വ്യക്തമാക്കുന്നു. എല്ലാ നിരകളും ഒരു ജോഡി ആവരണചിഹ്നങ്ങൾക്ക് (Parenthesis) അകത്താണ് എഴുതേണ്ടത്. ഓരോന്നും കോമ ഉപയോഗിച്ച് വേർതിരിക്കണം. നമുക്ക് എല്ലാ നിരകളും ഒരു വരിയിൽ എഴുതുവാൻ കഴിയും.

9.4.1 പട്ടികകളുടെയും, നിരകളുടെയും പേര് നൽകുന്നതിനുള്ള നിയമങ്ങൾ (Rules for naming tables and columns)

പട്ടികകയ്ക്കും, നിരയ്ക്കും പേരു നൽകുമ്പോൾ താഴെ പറയുന്ന കാര്യങ്ങൾ ശ്രദ്ധിക്കേണ്ടതാണ്.

- പേരിൽ അക്ഷരങ്ങൾ (A-Z, a-z), സംഖ്യകൾ(0-9), അണ്ടർ സ്കോർ (_), ഡോളർ($) തുടങ്ങിയ ചിഹ്നങ്ങൾ ഉപയോഗിക്കാവുന്നതാണ്.
- പേരിനകത്ത് ഒരു അക്ഷരമെങ്കിലും ഉണ്ടായിരിക്കണം.
- പേരിൽ ശൂന്യസ്ഥലം, പ്രത്യേക ചിഹ്നങ്ങൾ എന്നിവ ഉപയോഗിക്കാൻ പാടില്ല.
- SQL ലിനകത്തെ കീവേർഡുകൾ ഉപയോഗിക്കാൻ പാടില്ല.
- ഒരു പട്ടികയുടെ പേര് മറ്റൊരു പട്ടികയ്ക്ക് നൽകാൻ പാടില്ല.

ചില MySQL പതിപ്പുകളിൽ പട്ടികയുടെ പേര് വിരാമ ചിഹ്നങ്ങൾ (" ' ") ഉപയോഗിച്ച് എഴുതാറുണ്ട്. പട്ടികയുടെ പേര് ഉദ്ധരണികൾക്കുള്ളിൽ ഉപയോഗിക്കാം. ഏത് പ്രത്യേക ചിഹ്നവും പട്ടികയുടെ പേരിൽ ഉൾപ്പെടുത്താം.

സീരിയൽ നമ്പർ	ആട്രിബ്യൂട്ടുകൾ	വിശദീകരണം
1	Admission number	പൂർണ സംഖ്യ
2	Name	20 അക്ഷരങ്ങളുള്ള സ്ട്രിംഗ്
3	Gender	ഒരക്ഷരം
4	Date of birth	തീയതി
5	Course	15 അക്ഷരങ്ങളുള്ള ഒരു സ്ട്രിംഗ്
6	Family income	പൂർണ സംഖ്യ

പട്ടിക 9.4: student പട്ടികയുടെ ആട്രിബ്യൂട്ടുകൾ

പട്ടിക 9.4 ൽ കൊടുത്തിരിക്കുന്ന വിശദാംശങ്ങൾക്ക് അനുസരിച്ച് ഒരു പട്ടിക നിർമ്മിക്കാനാവശ്യമായ പ്രസ്താവന താഴെ കൊടുക്കുന്നു.

ക്വറി 9.1

```
CREATE TABLE student
(adm_no INT,
name VARCHAR(20),
gender CHAR,
dob DATE,
course VARCHAR(15),
f_income INT);
```

ഇവിടെ CREATE TABLE എന്ന കമാൻഡ് adm_no, name, gender, dob, course, f_income എന്നീ ആറു നിരകളോടു കൂടിയ ഒരു പട്ടിക നിർമ്മിക്കുന്നു. adm_no എന്ന നിരയിൽ പൂർണ സംഖ്യകളും name എന്ന നിരയിൽ പരമാവധി 20 അക്ഷരങ്ങളും വിദ്യാർഥി ആൺകുട്ടിയാണോ പെൺകുട്ടിയാണോ എന്ന് സൂചിപ്പിക്കുന്നതിനുള്ള ഒരു

അക്ഷരം gender ലും സൂചിപ്പിക്കുന്നു. course എന്ന നിരയിൽ വിദ്യാർഥിയുടെ കോഴ്സും, f_income എന്നത് കുടുംബത്തിന്റെ വരുമാനം സംഭരിക്കുന്നതിനുള്ള നിരയും ആണ്. name, course എന്നീ നിരകൾ സ്ട്രിംഗ് സംഭരിക്കുന്നതിനുള്ള സ്ഥലം കരുതി വയ്ക്കും. ഒരു പട്ടിക നിർമ്മിച്ച് അതിലേക്ക് റെക്കോർഡുകൾ ചേർക്കുമ്പോൾ ചില നിരകളിൽ പ്രത്യേക നിബന്ധനകൾ പാലിക്കേണ്ടത് ആവശ്യമാണ്. ചില നിരകളിൽ ശൂന്യമാകാതിരിക്കുക, മറ്റു ചില നിരകളിൽ ആവർത്തനങ്ങൾ ഒഴിവാക്കുക തുടങ്ങിയവയാകാം ഈ നിബന്ധനകൾ. പട്ടികയുടെ നിർമ്മാണ വേളയിൽത്തന്നെ അത് സംബന്ധിച്ച അറിവുണ്ടായിരിക്കണം. Constraint എന്നു വിളിക്കുന്ന ചില കീവേർഡുകൾ ഈ ആവശ്യത്തിനായി MySQL ൽ ലഭ്യമാക്കിയിട്ടുണ്ട്.

9.4.2 കൺസ്ട്രെയിന്റുകൾ (Constraints)

ഒരു പട്ടികയിലെ നിരകളിൽ ചേർക്കേണ്ട ഡാറ്റകളിൽ നടപ്പിൽ വരുത്തുന്ന നിയമങ്ങളാണ് കൺസ്ട്രെയിന്റുകൾ. ഒരു പട്ടിക നിർമ്മിക്കുമ്പോൾ ഫീൽഡുകളിലേക്ക് നാം നൽകുന്ന വിലകളിന്മേൽ കൺസ്ട്രയിന്റുകൾ നമുക്ക് പ്രയോഗിക്കാം. ഇത് ഒരു നിരയുടെ നിർവചനത്തിൽ നൽകിയാൽ ഈ നിബന്ധന ലംഘിക്കുന്ന ഒരു വിലയും MySQL സ്വീകരിക്കില്ല. ഇത് ഡാറ്റാബേസിലെ ഡാറ്റയുടെ കൃത്യതയും, വിശ്വാസ്യതയും ഉറപ്പുവരുത്തുന്നു. ഈ നിബന്ധനകൾ ഡാറ്റാബേസിന്റെ സമ്പൂർണത ഉറപ്പുവരുത്തുന്നതിനാൽ അവയെ ഡാറ്റാബേസ് സമ്പൂർണതാ കൺസ്ട്രെയിന്റുകൾ (Integrity Constraints) എന്നു വിളിക്കുന്നു. കൺസ്ട്രെയിന്റുകൾ നിരയുടെ തലത്തിലോ പട്ടികയുടെ തലത്തിലോ ഉണ്ടാക്കാവുന്നതാണ്.

a. നിരകളിൻമേലുള്ള കൺസ്ട്രെയിന്റുകൾ

നിരകളിൻമേലുള്ള കൺസ്ട്രെയിന്റുകൾ ഓരോ നിരയിലും പ്രത്യേകമായി ഉപയോഗിക്കുന്നു. അവ ഒരു നിരയുടെ ഡാറ്റ ഇനം നൽകിയതിനുശേഷമാണ് എഴുതുക. താഴെ പറയുന്നവയാണ് നിരകളുടെ തലത്തിലുള്ള കൺസ്ട്രെയിന്റുകൾ.

i) NOT NULL

ഒരു നിരയിൽ മൂല്യം ശൂന്യമാക്കരുത് എന്നാണ് ഇത് പ്രസ്താവിക്കുന്നത്. NULL എന്നത് SQL ലെ ശൂന്യ സ്ഥലത്തെ സൂചിപ്പിക്കുന്ന കീവേർഡാണ്. അത് വിട്ടഭാഗമോ, പൂജ്യമോ അല്ല എന്ന് പ്രധാനമായും ഓർമ്മിക്കേണ്ടതാണ്. അത് തികച്ചും വ്യത്യസ്തമായ ഒന്നാണ്. വിട്ടഭാഗം മറ്റൊരു വിട്ടഭാഗത്തിനും, പൂജ്യം മറ്റൊരു പൂജ്യത്തിനും തുല്യമാണെങ്കിൽ NULL ഒരിക്കലും മറ്റൊരു NULL ന് തുല്യമല്ല. രണ്ടു NULL മൂല്യങ്ങളെ കൂട്ടാനോ, കുറയ്ക്കാനോ, താരതമ്യം ചെയ്യാനോ സാധ്യമല്ല.

ii) AUTO_INCREMENT

MySQL ലെ കീവേഡായ AUTO_INCREMENT ഉപയോഗിക്കുന്നത് സ്വയം വർധന നടപ്പിൽ വരുത്തുന്നതിനാണ്. AUTO_INCREMENT എന്ന കൺസ്ട്രയിന്റ് ഉള്ള ഒരു നിരയിൽ മൂല്യം നൽകിയിട്ടില്ലെങ്കിൽ MySQL സീരിയൽ നമ്പറുകൾ സ്വയമായി ക്രമീകരിക്കുകയും പുതിയ നിരയുടെ അനുബന്ധ റെക്കോർഡിൽ പുതുതായി നിശ്ചയിച്ചിട്ടുള്ള മൂല്യങ്ങൾ

ചേർക്കുകയും ചെയ്യുന്നു. AUTO_INCREMENT ന്റെ തനത് മൂല്യം 1 ആണ്. അതിനാൽ ഓരോ റെക്കോർഡിനും 1 വച്ച് മൂല്യം വർധിച്ചുകൊണ്ടേയിരിക്കും. ഓരോ നിരയിലേക്കും NULL മൂല്യം കൊടുക്കുകയാണെങ്കിൽ ഈ സവിശേഷ സ്വഭാവം അത് കാണിക്കും. AUTO_INCREMENT സവിശേഷത ഓരോ പുതിയ വരിക്കും ഒരു അനന്യമായ ID (Unique ID) നിശ്ചയിക്കുന്നത് എളുപ്പമാക്കുന്നു. കാരണം MySQL നമുക്കായി മൂല്യങ്ങൾ ഉണ്ടാക്കുന്നു. AUTO_INCREMENT നൽകിയിട്ടുള്ള നിര പട്ടികയുടെ പ്രാഥമിക കീ ആയി നിർവചിക്കേണ്ടതുണ്ട്. ഒരു പട്ടികയിലെ ഒരു നിരയിൽ മാത്രമേ AUTO_INCREMENT അനുവദിക്കൂ.

iii) UNIQUE

ഈ കൺസ്ട്രയിന്റ് ഉപയോഗിക്കുന്ന നിരയിലെ രണ്ടു വരികളിൽ ഒരേ മൂല്യം ഉണ്ടായിരിക്കുകയില്ല.

ഈ കൺസ്ട്രയിന്റ് പട്ടികയുടെ പ്രൈമറി കീയായി ഒരു നിരയെ മാറ്റുന്നു. ഈ കൺസ്ട്രയിന്റ് UNIQUE ന് സമാനമാണ്. ഒരു നിരയിൽ അല്ലെങ്കിൽ നിരകളുടെ സംയോജനത്തിൽ കൺസ്ട്രയിന്റ് ഉപയോഗിക്കാൻ സാധിക്കും. പ്രൈമറി കീയിൽ NULL മൂല്യങ്ങൾ ഉണ്ടായിരിക്കരുത്. മറ്റൊരു തരത്തിൽ പറഞ്ഞാൽ ഇത് UNIQUE, NOT NULL എന്നീ കൺസ്ട്രെയിന്റുകളുടെ സംയോജനമായി കണക്കാക്കാം. പ്രൈമറി കീ നടപ്പിൽ വരുത്തുന്നതിന് ആ നിര അനന്യമായതും, ശൂന്യമല്ലാത്തതും ആയിരിക്കണം.

v) DEFAULT

ഈ കൺസ്ട്രയിന്റ് ഉപയോഗിച്ച് ഒരു നിരയ്ക്ക് തനത് മൂല്യം നൽകാവുന്നതാണ്. ഉപയോക്താവ് ഒരു നിരയ്ക്ക് മൂല്യം നൽകുന്നില്ലെങ്കിൽ ഈ കൺസ്ട്രയിന്റ് ഉപയോഗിച്ച് തനതു മൂല്യം കൊടുക്കുന്നതാണ്.

ഇനി നമുക്ക് student പട്ടികയ്ക്ക് ചില കൺസ്ട്രയിന്റുകൾ നൽകാം. ഇതിന് താഴെ പറയുന്ന രീതിയിൽ ക്വറി 9.1 പരിഷ്കരിക്കാം (ക്വറി 9.2 കാണുക).

ക്വറി 9.2

```
CREATE TABLE student
(adm_no INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(20) NOT NULL,
gender CHAR DEFAULT 'M',
dob DATE,
course VARCHAR(15)
f_income INT);
```

ക്വറി 9.2 ൽ PRIMARY KEY, AUTO_INCREMENT എന്നീ കൺസ്ട്രയിന്റുകൾ adm_no എന്ന നിരയിൽ കൊടുത്തിട്ടുണ്ട്. അതിനാൽ ഈ നിരയിൽ ആവർത്തിച്ചുവരുന്ന ഡാറ്റകൾ

സ്വീകരിക്കപ്പെടുകയില്ല. ഈ നിരയിലേക്ക് നാം മൂല്യങ്ങളൊന്നും നൽകിയില്ലെങ്കിൽ MySQL തന്നെ ഡാറ്റ യാന്ത്രികമായി സൃഷ്ടിക്കും. NOT NULL കൺസ്ട്രയിന്റ് name എന്ന നിരയിൽ നൽകിയിരിക്കുന്നതിനാൽ ആ നിര ശൂന്യമായിരിക്കാൻ അനുവദിക്കുന്നതല്ല. അതായത് ആ നിരയിൽ ഒരു ഡാറ്റ നിർബന്ധമാണ് അതുപോലെ gender എന്ന നിരയിൽ 'M' എന്ന് നാം നൽകിയില്ലെങ്കിൽ തനതു വിലയായി 'M' സംഭരിക്കപ്പെടും.

b. പട്ടികയുടെ തലത്തിലുള്ള കൺസ്ട്രയിന്റുകൾ

പട്ടികയുടെ തലത്തിലുള്ള കൺസ്ട്രയിന്റുകൾ നിരയുടെ തലത്തിലുള്ള കൺസ്ട്രയിന്റുകൾക്ക് സമാനമാണ്. പട്ടികയുടെ തലത്തിലുള്ള കൺസ്ട്രയിന്റുകൾ ഒന്നിലധികം നിരകളിൽ ഉപയോഗിക്കാൻ സാധിക്കും എന്നതാണ് പ്രധാന വ്യത്യാസം. ഒരു കൺസ്ട്രയിന്റ് ഏതാനും നിരകളിൽ പ്രയോഗിച്ചാൽ അതിനെ പട്ടികയുടെ തലത്തിലുള്ള കൺസ്ട്രയിന്റ് എന്നു പറയുന്നു. പട്ടികയുടെ തലത്തിലുള്ള കൺസ്ട്രയിന്റ് ഒരു പട്ടികയുടെ നിർവചനത്തിൽ അവസാനമാണ് കൊടുക്കേണ്ടത്. ഉദാഹരണത്തിന് ക്വറി 9.3, stock എന്ന പട്ടിക നിർമിക്കുന്നു. icode, iname എന്നീ നിരകൾക്ക് UNIQUE എന്ന് സംയുക്തമായി നൽകിയിരിക്കുന്നു.

ക്വറി 9.3

```
CREATE TABLE stock
(icode CHAR(2) PRIMARY KEY AUTO_INCREMENT,
iname VARCHAR(30) NOT NULL,
dt_purchase DATE,
rate DECIMAL(10,2),
qty INT,
UNIQUE (icode, iname));
```

ക്വറി 9.3 icode, iname എന്നീ നിരകളിൽ ഒരേ ഡാറ്റ വന്നിട്ടില്ല എന്ന് ഉറപ്പു വരുത്തുന്നു.

9.4.3 ഒരു പട്ടികയുടെ ഘടന മനസ്സിലാക്കുന്നു (Viewing the structure of a table)

student, stock എന്നീ രണ്ടു പട്ടികകൾ നാം നിർമ്മിച്ചല്ലോ. ഒരു പട്ടിക നിർമ്മിച്ച ശേഷം അതിന്റെ ഘടന എങ്ങനെയാണ് കാണുന്നത്? DESCRIBE കമാൻഡ് ഉപയോഗിച്ച് ഒരു പട്ടികയുടെ ഘടന മനസ്സിലാക്കാൻ സാധിക്കും. ഇതിന്റെ വാക്യഘടന താഴെ കൊടുത്തിരിക്കുന്നു.

```
DESCRIBE <table_name>;
```

OR

```
DESC <table_name>;
```

student എന്ന പട്ടികയുടെ ഘടന കാണുന്നതിനുള്ള കമാൻഡാണ്

```
DESC student;
```

ചിത്രം 9.4 ഈ കമാൻഡിന്റെ ഔട്ട്പുട്ട് കാണിച്ചു തരുന്നു.

```
+-----------+---------------+------+-----+---------+----------------+
| Field     | Type          | Null | Key | Default | Extra          |
+-----------+---------------+------+-----+---------+----------------+
| adm_no    | int(11)       | NO   | PRI | NULL    | auto_increment |
| name      | varchar(20)   | NO   |     | NULL    |                |
| gender    | char(1)       | YES  |     | M       |                |
| dob       | date          | YES  |     | NULL    |                |
| course    | varchar(15)   | YES  |     | NULL    |                |
| f_income  | int(11)       | YES  |     | NULL    |                |
+-----------+---------------+------+-----+---------+----------------+
6 rows in set (0.02 sec)
```

ചിത്രം 9.4: student പട്ടികയുടെ ഘടന

```
+--------------------+
| Tables_in_school   |
+--------------------+
| student            |
+--------------------+
1 row in set (0.00 sec)
```

ചിത്രം 9.5: SHOW TABLES ന്റെ ഔട്ട്പുട്ട്

ക്വറി 9.2 ൽ adm_no, f_income എന്നീ രണ്ടു നിരകളുടെ വലുപ്പം പറഞ്ഞിട്ടില്ല എന്നത് ശ്രദ്ധിക്കുക. എന്നാൽ MySQL ഈ രണ്ടു നിരകളുടെ വലുപ്പം 11 ആയി സ്വീകരിക്കും adm_no എന്ന നിര PRIMARY KEY, AUTO_INCREMENT ആയി പ്രസ്താവിച്ചിട്ടുണ്ട്. അതിനാൽ ഈ നിരകളിൽ ശൂന്യമോ ആവർത്തിച്ചുള്ള മൂല്യങ്ങളോ വരുവാൻ അനുവദിക്കില്ല. ഈ നിരയിലേക്ക് മൂല്യങ്ങൾ നൽകാതിരുന്നാൽ ബന്ധപ്പെട്ട നിരയുടെ മൂല്യം 1 വർധിപ്പിച്ചു കൊണ്ട് പുതിയ മൂല്യം നിർമ്മിക്കും. ആദ്യ റെക്കോർഡിൽ ഈ നിരയുടെ മൂല്യം ഇല്ലാതിരുന്നാൽ MySQL തനതുമൂല്യമായി 1 നൽകുന്നു. gender എന്ന നിരയിലേക്ക് ഒരു സ്ഥിര മൂല്യം 'M' എന്ന് നൽകിയിരിക്കുന്നതായി നമുക്ക് കാണാൻ സാധിക്കും. ചിത്രം 9.4 ഈ സവിശേഷത വിവരിക്കുന്നു. നിലവിലുള്ള ഡാറ്റാബേസിലെ പട്ടികകൾ കാണുന്നതിന് SHOW TABLES എന്ന കമാൻഡ് ഉപയോഗിക്കാം (ചിത്രം 9.5).

നമുക്കു ചെയ്യാം

ചിത്രം 9.4 ഉപയോഗിച്ച് stock എന്ന പട്ടികയുടെ ഘടന എഴുതുക. നിങ്ങളുടെ ഉത്തരം DESC stock; എന്ന കമാൻഡ് ഉപയോഗിച്ച് പരിശോധിക്കുക.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. ഒരു പട്ടികയുടെ ഘടന കാണുന്നതിനു വേണ്ടി താഴെ പറയുന്ന ഏതു കമാൻഡാണ് ഉപയോഗിക്കേണ്ടത് ?

   a. LIST b. SHOW c. DESCRIBE d. STRUCT

2. CREATE TABLE കമാൻഡിന്റെ വാക്യഘടന എഴുതുക.
3. നിരകളിൽ ഉപയോഗിക്കുന്ന കൺസ്ട്രയിന്റുകളുടെ പേരെഴുതുക.
4. PRIMARY KEY, UNIQUE എന്നീ കൺസ്ട്രയിന്റുകൾ തമ്മിലുള്ള വ്യത്യാസം എഴുതുക.

5. AUTO_INCREMENT എന്ന കൺസ്ട്രയിന്റിന്റെ സവിശേഷതകൾ എന്തൊക്കെ?

6. ഒരു പട്ടികയ്ക്ക് പേര് നൽകുന്നതിനുള്ള നിയമങ്ങൾ എഴുതുക.

7. ഒരു പട്ടികയുടെ PRIMARY KEY യായി എത്ര നിരകൾ പ്രസ്താവിക്കാം?

9.5 ഒരു പട്ടികയിൽ ഡാറ്റ ചേർക്കുന്ന രീതി (Inserting data into tables)

ഒരു ഡാറ്റാബേസും അതിന്റെ പട്ടികകളും നാം നിർമ്മിച്ചു കഴിഞ്ഞു. ഈ പട്ടികയിൽ ഏതാനും റെക്കോർഡുകൾ ചേർക്കുന്നത് എങ്ങനെയെന്ന് നമുക്ക് ചർച്ച ചെയ്യാം. പട്ടികയിൽ ഡാറ്റ ചേർക്കുന്നതിന് INSERT INTO എന്ന DML കമാൻഡ് ഉപയോഗിക്കുന്നു. അതിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
INSERT INTO <table_name> [<column1>,<column2>,...,<columnN>]
VALUES(<value1>,<value2>,...,<valueN>);
```

ഇവിടെ <table_name> എന്നത് വരികൾ ഉൾപ്പെടുത്തേണ്ട പട്ടികയുടെ പേരാണ്. <column1>,<column2>,.........,<column N> എന്നത് മൂല്യങ്ങൾ ചേർക്കേണ്ട നിരകളാണ്. <value1>,<value2>,....<value N> എന്നത് നിർദേശിച്ചിട്ടുള്ള നിരകളിൽ ചേർക്കേണ്ട വിലകളാണ്.

ഉദാഹരണത്തിന് *student* എന്ന പട്ടികയുടെ ഡാറ്റകളായ *1001, 'Alok', 'M', 1998/10/2, 'Science', 24000* എന്നിവ പുതിയ റെക്കോർഡായി adm_no, name, gender, dob, course, f_income എന്നീ നിരകളിലേക്ക് യഥാക്രമം ചേർക്കാം. (ക്വറി 9.4 ഉപയോഗിക്കുക)

ക്വറി 9.4

```
INSERT INTO student
VALUES (1001,'Alok','M','1998/10/2', 'Science',
  24000);
```

കമ്പ്യൂട്ടറിന്റെ പ്രതികരണം ഇതായിരിക്കും.

```
Query OK, 1 row affected (0.05 sec)
```

ഓരോ നിരയിലും ഓരോ മൂല്യങ്ങൾ നൽകി INSERT INTO കമാൻഡ് student എന്ന പട്ടികയിൽ ഒരു പുതിയ വരി ചേർക്കുന്നു. പട്ടികയിൽ ഒരു വരി ചേർക്കുമ്പോൾ എല്ലാ നിരയിലും മൂല്യങ്ങൾ ഉണ്ടെങ്കിൽ ക്വറിയിൽ നിരയുടെ പേര് പരാമർശിക്കേണ്ട ആവശ്യമില്ല. പക്ഷേ, മൂല്യങ്ങളുടെ ക്രമം പട്ടികയിലെ നിരകളുടെ ക്രമത്തിനനുസരിച്ചാണ് എന്ന് ഉറപ്പ് വരുത്തേണ്ടതാണ്. ഇനി ക്വറി 9.5ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ചില മാറ്റങ്ങൾ വരുത്തി മറ്റൊരു വരി ചേർക്കാം.

ക്വറി 9.5

```
INSERT INTO student (name, dob, course, f_income)
VALUES ('Nike','1998/11/26','Science',35000);
```

ഈ പ്രസ്താവനയുടെ പ്രതികരണം ഇതായിരിക്കും.

```
Query OK, 1 row affected (0.01 sec)
```

ക്വറി 9.5ൽ adm_no, gender എന്നീ നിരകൾക്ക് വില നൽകിയിട്ടില്ല. ആദ്യത്തെ കുട്ടിയുടെ അഡ്മിഷൻ നമ്പർ 1001 ആയിരിക്കെ അടുത്ത കുട്ടിയുടെ അഡ്മിഷൻ നമ്പർ 1002 ആയിരിക്കും. gender ന് തനതു മൂല്യം 'M' എന്ന് നിശ്ചയിച്ചിട്ടുണ്ട്.

പട്ടികയിൽ ഡാറ്റ ചേർക്കുമ്പോൾ താഴെ പറയുന്ന കാര്യങ്ങൾ ഓർമ്മിക്കണം:

- ഒരു പുതിയ വരി ചേർക്കുമ്പോൾ ആ നിരയുടെ ഡാറ്റ ഇനം, ആ നിരയിലെ ഡാറ്റയുമായി യോജിക്കുന്നുണ്ടോ എന്ന് ഉറപ്പ് വരുത്തുക.
- ഒരു പട്ടികയ്ക്ക് നിശ്ചയിച്ച സമ്പൂർണതാ കൺസ്ട്രയിന്റ് ആ പട്ടിക പിന്തുടരുന്നുന്നുണ്ടോ എന്ന് ഉറപ്പ് വരുത്തുക.
- CHAR അല്ലെങ്കിൽ VARCHAR എന്നീ തരത്തിലുള്ള ഡാറ്റകൾ ഒറ്റ ഉദ്ധരണികൾക്കുള്ളിലോ ഇരട്ട ഉദ്ധരണികൾക്കുള്ളിലോ എഴുതണം.
- DATE തരത്തിലുള്ള നിരകൾക്ക് വില നൽകേണ്ടത് ഒറ്റ ഉദ്ധരണികൾക്കുള്ളിലാണ്. ഈ സ്ട്രിംഗ് ആന്തരികമായി DATE എന്ന ഡാറ്റ ഇനത്തിലേക്ക് മാറ്റം ചെയ്യപ്പെടുന്നു.
- NULL മൂല്യങ്ങളെ NULL ആയി പ്രസ്താവിക്കണം (ഉദ്ധരണികൾ ഇല്ലാതെ).
- എല്ലാ നിരകൾക്കും ഡാറ്റ ലഭ്യമല്ലെങ്കിൽ പട്ടികയുടെ പേരിന് ശേഷം നിരകളുടെ ലിസ്റ്റ് ഉൾപ്പെടുത്തണം.

ഒരൊറ്റ INSERT INTO കമാൻഡ് ഉപയോഗിച്ച് പല വരികൾ ഒരു പട്ടികയിൽ ചേർക്കാൻ MySQL അനുവദിക്കുന്നു. പൊതുവായ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
INSERT INTO <table-name> VALUES(...), (...), ... ;
```

പട്ടിക 9.1 ൽ രണ്ട് റെക്കോർഡുകൾ കൂടി ക്വറി 9.6 ഉപയോഗിച്ച് നമുക്ക് ചേർക്കാം

ക്വറി 9.6

```
INSERT INTO student (name, dob, course, f_income)
VALUES('Bharath','1999/01/01','Commerce',45000),
      ('Virat','1998/12/05','Science',22000);
```

കമ്പ്യൂട്ടറിന്റെ പ്രതികരണം ഇതായിരിക്കും.

```
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

ഇവിടെ രണ്ട് റെക്കോർഡുകൾ ആവരണചിഹ്നത്തിനകത്തായി വെവ്വേറെ നൽകിയിട്ടുണ്ടെന്ന് നമുക്ക് കാണാം. രണ്ടു റെക്കോർഡുകൾ വിജയകരമായി ചേർത്തിട്ടുണ്ടെന്ന് പ്രതികരണം സൂചിപ്പിക്കുന്നു. ഒരു വിദ്യാർഥിയുടെ മാസവരുമാനത്തിന്റെ ഡാറ്റ നമുക്ക് ഇല്ല എന്ന് കരുതുക. ഈ റെക്കോർഡ് എങ്ങനെ ചേർക്കും? ക്വറി 9.7 ഇത് വിവരിക്കുന്നു.

ക്വറി 9.7

```
INSERT INTO student(name, dob, gender, course)
VALUES('Meera','1998/08/15','F','Science');
```

ഈ ക്വറിക്ക് മറുപടിയായി adm_no ന്റെ മൂല്യം കമ്പ്യൂട്ടർ തന്നെ ചേർക്കുന്നു. എന്നാൽ f_income എന്ന നിരയുടെ വില NULL ആയിരിക്കും. നിരയുടെ ക്രമവും ഈ ക്വറിയിൽ

മാറിയിട്ടുണ്ട്. ഒരു വരിയിലെ മൂല്യങ്ങളുടെ അഭാവത്തെ ക്വറി 9.8 ൽ കാണിച്ചിരിക്കുന്നതു പോലെ മറ്റൊരു തരത്തിൽ കൈകാര്യം ചെയ്യാം.

ക്വറി 9.8

```
INSERT INTO student(name, dob, gender, course, f_income)
VALUES('Divakar','1998/02/21','M' Science',NULL);
```

VALUES ഉപവാക്യത്തിൽ f_income ത്തിന് NULL കൊടുത്തിരിക്കുന്നത് ശ്രദ്ധിക്കുക.

നമുക്കു ചെയ്യാം

student എന്ന പട്ടികയിൽ ഏതാനും റെക്കോർഡുകൾ കൂടി നമുക്ക് ചേർക്കാം. പട്ടിക 9.5 കാണിച്ചിരിക്കുന്നതു പോലെ വിദ്യാർഥികളുടെ വിശദാംശങ്ങൾ ചേർക്കുന്നതിന് ആവശ്യമായ ക്വറികൾ എഴുതുക.

adm_no	name	gender	dob	course	f_income
1025	Kaushi	M	1998/10/2	Commerce	17000
1026	Niveditha	F	1999/03/04	Humanities	52000
1027	Sreekumar	M	1998/06/06	Science	15000
1057	Chaithanya	F	1999/06/03	Science	

പട്ടിക 9.5: student പട്ടികയിൽ കൂടുതൽ റെക്കോർഡുകൾ

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. ഒരു പട്ടികയിൽ ഒരു വരി ചേർക്കുന്നതിന് താഴെ കൊടുത്തിരിക്കുന്നതിൽ ഏത് കമാൻഡ് ഉപയോഗിക്കുന്നു ?

   a. ADD   b. CREATE   c. INSERT   d. MAKE

2. ഒരു പട്ടികയിൽ പുതിയ ഡാറ്റ ചേർക്കുന്നതിനുള്ള പ്രസ്താവന ഏത്?

   a. ADD RECORD   b. INSERT RECORD

   c. INSERT INTO   d. INSERT ROW

3. INSERT കമാൻഡിന്റെ കൂടെ ഉപയോഗിക്കുന്ന കീവേർഡുകൾ എഴുതുക.

നമുക്കു ചെയ്യാം

ക്വറി 9.3 ഉപയോഗിച്ച് ഉണ്ടാക്കിയ stock എന്ന പട്ടികയിൽ കൂടുതൽ റെക്കോർഡുകൾ ചേർക്കുന്നതിനുള്ള SQL കമാൻഡുകൾ എഴുതുക. നിരയിൽ വിലകൾ നൽകുമ്പോൾ AUTO_INCREMENT, UNIQUE എന്നീ കൺസ്ട്രയിന്റുകളുടെ സൗകര്യം ഉപയോഗിക്കുക.

9.6 പട്ടികയിൽ നിന്ന് വിവരങ്ങൾ വീണ്ടെടുക്കുന്ന രീതി (Retrieving information from tables)

school എന്ന ഡാറ്റാബേസും student എന്ന പട്ടികയും നിർമ്മിച്ച് അതിൽ 10 റെക്കോർഡുകൾ നാം ചേർത്തുവല്ലോ. ഈ പട്ടികകളിൽ ശേഖരിച്ച ഡാറ്റ എങ്ങനെ തിരിച്ചെടുക്കാമെന്ന് നമുക്ക് പഠിക്കാം. അത് ഒരു ഡാറ്റ കൈകാര്യം ചെയ്യൽ പ്രക്രിയയാണ്. ഈ ആവശ്യത്തിനായി SQL ലിൽ SELECT കമാൻഡ് ലഭ്യമാക്കിയിട്ടുണ്ട്. ഒരു പട്ടികയിലെ നിർദിഷ്ട നിരയിൽ നിന്ന് വിവരങ്ങൾ വീണ്ടെടുക്കാൻ ഇത് ഉപയോഗിക്കുന്നു. SELECT കമാൻഡിന് നിരവധി രൂപങ്ങൾ ഉണ്ട്. SELECT കമാൻഡിന്റെ ലളിതമായ രൂപമാണ് താഴെകൊടുത്തിരിക്കുന്നത്.

```
SELECT <column_name>[,<column_name>,<column_name>, ...]
FROM <table_name>;
```

ഇവിടെ <column_name> എന്നത്, ഏത് നിരയിൽ നിന്നാണ് ഡാറ്റ വീണ്ടെടുക്കേണ്ടത് എന്നും <table_name> എന്നത്, ഏത് പട്ടികയിൽ നിന്നാണ് വിവരങ്ങൾ വീണ്ടെടുക്കേണ്ടത് എന്നുമാണ് സൂചിപ്പിക്കുന്നത്. പട്ടികയുടെ പേര് നൽകുന്നത് FROM എന്ന കീവേർഡ് ഉപയോഗിച്ചാണ്. ഇത് SELECT കമാൻഡിന്റെ കൂടെ നിർബന്ധമായും ഉപയോഗിക്കേണ്ട ഉപവാക്യമാണ് (Clause). SELECT കമാൻഡിൽ കൊടുത്തിരിക്കുന്ന ക്രമത്തിൽ തന്നെ നിരകളായി ഡാറ്റ പ്രദർശിപ്പിക്കപ്പെടും.

ഇപ്പോൾ നമുക്ക് SELECT കമാൻഡിന്റെ നിർവഹണം വിവിധ ക്വറികൾ വഴി പരിചയപ്പെടാം. ക്വറി 9.9 പ്രവർത്തിക്കുമ്പോൾ ചിത്രം 9.6 ൽ കാണിച്ചിരിക്കുന്നതു പോലെ student പട്ടികയിലെ വിദ്യാർഥികളുടെ name, course എന്നിവ നമുക്ക് ലഭിക്കുന്നു.

ക്വറി 9.9

```
SELECT name, course
FROM student;
```

```
+------------+-------------+
| name       | course      |
+------------+-------------+
| Alok       | Science     |
| Nike       | Science     |
| Bharath    | Commerce    |
| Virat      | Science     |
| Meera      | Science     |
| Divakar    | Science     |
| Kaushi     | Commerce    |
| Niveditha  | Humanities  |
| Sreekumar  | Science     |
| Chaithanya | Science     |
+------------+-------------+
10 rows in set (0.00 sec)
```

ചിത്രം 9.6: Query 9.9 ന്റെ ഔട്ട്പുട്ട്

പട്ടിക മുഴുവനും നമുക്ക് പ്രദർശിപ്പിക്കണമെങ്കിൽ ആ റിലേഷനിലെ നിരകളുടെ ലിസ്റ്റ് മുഴുവൻ നൽകേണ്ടതില്ല, പകരം നിരകളുടെ പൂർണമായ ലിസ്റ്റ് ഉൾപ്പെടുത്തുന്നതിന് നക്ഷത്ര ചിഹ്നം (*) ഉപയോഗിക്കാം (ക്വറി 9.10). ഇതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 9.7 ൽ കൊടുത്തിരിക്കുന്നു.

ക്വറി 9.10

```
SELECT * FROM student;
```

```
+--------+-------------+--------+------------+-------------+----------+
| adm_no | name        | gender | dob        | course      | f_income |
+--------+-------------+--------+------------+-------------+----------+
|   1001 | Alok        | M      | 1998-10-02 | Science     |    24000 |
|   1002 | Nike        | M      | 1998-11-26 | Science     |    35000 |
|   1003 | Bharath     | M      | 1999-01-01 | Commerce    |    45000 |
|   1004 | Virat       | M      | 1998-12-05 | Science     |    22000 |
|   1005 | Meera       | F      | 1998-08-15 | Science     |     NULL |
|   1006 | Divakar     | M      | 1998-02-21 | Science     |     NULL |
|   1025 | Kaushi      | M      | 1998-10-02 | Commerce    |    17000 |
|   1026 | Niveditha   | F      | 1999-03-04 | Humanities  |    52000 |
|   1027 | Sreekumar   | M      | 1998-06-06 | Science     |    15000 |
|   1057 | Chaithanya  | F      | 1999-06-03 | Science     |     NULL |
+--------+-------------+--------+------------+-------------+----------+
10 rows in set (0.00 sec)
```

ചിത്രം 9.7: student പട്ടികയുടെ മുഴുവൻ ഉള്ളടക്കം

ചിത്രം 9.7 ൽ f_income എന്ന നിരയിലെ ചില വരികളിൽ NULL മൂല്യങ്ങൾ നമുക്ക് കാണാൻ സാധിക്കും. വരിയിലേക്ക് മൂല്യങ്ങൾ ചേർക്കുമ്പോൾ വിലകൾ വിട്ടുപോയതാണ് ഇതിന് കാരണം. (ക്വറി 9.7, 9.8 പട്ടിക 9.1, 9.5 പരിശോധിക്കുക).

9.6.1 DISTINCT ഉപയോഗിച്ച് നിരകളിലെ ആവർത്തനങ്ങൾ ഉള്ള മൂല്യങ്ങൾ നീക്കം ചെയ്യുന്ന രീതി (Eliminating duplicate values in columns using DISTINCT)

student എന്ന പട്ടികയിലെ വിവിധ കോഴ്സുകളുടെ പേരുകൾ നമുക്ക് അറിയാമെന്ന് കരുതുക, SELECT course FROM student; എന്ന ക്വറി നിർമ്മിക്കുകയാണെങ്കിൽ അത് course എന്ന നിരയിലെ എല്ലാ മൂല്യങ്ങളെയും ചിത്രം 9.7 ൽ കാണിച്ചിരിക്കുന്ന പോലെ പ്രദർശിപ്പിക്കും. റിലേഷനിലെ എല്ലാ വരികളിൽ നിന്നുമുള്ള ഡാറ്റ ഒന്നിൽ കൂടുതൽ തവണ ആവർത്തിക്കുന്നുണ്ടെങ്കിലും അവ പ്രദർശിപ്പിക്കും. ഈ ആവർത്തനം DISTINCT എന്ന കീവേർഡ് ഉപയോഗിച്ച് നമുക്ക് ഒഴിവാക്കാൻ സാധിക്കും (ക്വറി 9.11). ചിത്രം 9.8 ൽ ഇതിന്റെ ഔട്ട്പുട്ട് കൊടുത്തിരിക്കുന്നു.

ക്വറി 9.11

```
SELECT DISTINCT course
FROM student;
```

```
+------------+
| course     |
+------------+
| Science    |
| Commerce   |
| Humanities |
+------------+
3 rows in set (0.25 sec)
```

ചിത്രം . 9.8: DISTINCT ന്റെ ഉപയോഗം

ഔട്ട്പുട്ടിൽ ആവർത്തനം ഇല്ല. കീവേർഡിനോടു കൂടി ഉപയോഗിച്ചിരിക്കുന്ന നിരയിൽ ഒന്നിലധികം NULL വില ഉണ്ടെങ്കിൽ അതിൽ ഒരു വില മാത്രമെ ഫലത്തിൽ കാണിക്കുകയുള്ളു. DISTINCT നു പകരം ALL എന്ന കീവേർഡ് ഉപയോഗിച്ചുവെന്നിരിക്കട്ടെ, ആ നിരയിലെ എല്ലാ ആവർത്തന മൂല്യങ്ങളും ഫലത്തിൽ അടങ്ങിയിരിക്കും. അതായത് DISTINCT, ALL എന്നിവ ഉപയോഗിക്കാത്ത അവസ്ഥയിലുള്ള അതേ ഫലം ലഭിക്കുന്നു.

9.6.2 പ്രത്യേക വരികൾ WHERE ഉപവാക്യം ഉപയോഗിച്ച് തിരഞ്ഞെടുക്കുന്ന രീതി (Selecting specific rows using WHERE clasue)

ചില സന്ദർഭങ്ങളിൽ നമുക്ക് പട്ടികയിലെ ഒരു ഉപഗണം മാത്രം തിരഞ്ഞെടുക്കേണ്ടതായി വരും. ഉദാഹരണമായി പെൺകുട്ടികളെക്കുറിച്ചുള്ള വിശദാംശങ്ങൾ മാത്രമോ, അല്ലെങ്കിൽ

കുടുംബമാസവരുമാനം Rs 25000/- ന് താഴെയുള്ള വിദ്യാർഥികളുടെ വിശദാംശങ്ങളോ ആവശ്യമായി വന്നേക്കാം. ഇവിടെ റെക്കോർഡുകളുടെ വീണ്ടെടുക്കലിൽ ഒരു തിരഞ്ഞെടുക്കൽ ഉണ്ട്. തീർച്ചയായും ഇത് ഒരു നിബന്ധനയുടെ അടിസ്ഥാനത്തിലായിരിക്കും. SELECT കമാൻഡിന്റെ WHERE ഉപവാക്യം ഉപയോഗിച്ച് റെക്കോർഡുകൾ വീണ്ടെടുക്കുന്നതിനുള്ള ചില മാനദണ്ഡങ്ങൾ ഉണ്ടാക്കാൻ SQL നമ്മെ സഹായിക്കുന്നു. WHERE ഉപവാക്യത്തോടുകൂടിയ SELECT കമാൻഡിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
SELECT <column_name>[,<column_name>,<column_name>, ...]
FROM <table_name>
WHERE <condition>;
```

WHERE ഉണ്ടെങ്കിൽ SELECT കമാൻഡ് പട്ടികയിലെ ഓരോ വരിയിൽക്കൂടിയും കടന്നുപോവുകയും നിബന്ധനകൾ പാലിക്കുന്നുണ്ടോ എന്ന് പരിശോധിക്കുകയും ചെയ്യും. ഒരു വരി നിബന്ധന പാലിക്കുന്നുണ്ടെങ്കിൽ ആ വരി ഔട്ട്പുട്ടിൽ പ്രദർശിപ്പിക്കുന്നു. ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ, റിലേഷണൽ ഓപ്പറേറ്ററുകൾ എന്നിവ ഉപയോഗിച്ച് നിബന്ധനകൾ നിർമ്മിക്കാവുന്നതാണ്. MySQL ൽ ഇതിന് വ്യത്യസ്ത ഓപ്പറേറ്ററുകൾ ഉപയോഗിക്കുന്നു. ഇവ പട്ടിക 9.6 ൽ കൊടുത്തിരിക്കുന്നു.

ഓപ്പറേറ്റർ		അർഥം/ ഫലം
റിലേഷണൽ ഓപ്പറേറ്റർ	=	തുല്യമായത്
	<> or !=	തുല്യമല്ലാത്തത്
	>	വലുത്
	<	ചെറുത്
	>=	വലുതോ തുല്യമായതോ
	<=	ചെറുതോ തുല്യമായതോ
ലോജിക്കൽ ഓപ്പറേറ്റർ	NOT	നിബന്ധന തെറ്റാണെങ്കിൽ ഔട്ട്പുട്ട് ശരി
	AND	രണ്ട് നിബന്ധനയും ശരിയാണെങ്കിൽ ഔട്ട്പുട്ട് ശരി
	OR	നിബന്ധനകളിൽ ഏതെങ്കിലും ശരിയാണെങ്കിൽ ഔട്ട്പുട്ട് ശരി

പട്ടിക 9.6: വ്യവസ്ഥകൾ ക്രമപ്പെടുത്തുന്നതിന് ഉപയോഗിക്കുന്ന ഓപ്പറേറ്ററുകൾ

ഒരു പട്ടികയിലെ പെൺകുട്ടികളുടെ വിശദാംശങ്ങൾ നൽകുന്നതിനുള്ള SQL പ്രസ്താവന ഇനി എഴുതാം. ക്വറി 9.12 ൽ നിബന്ധന ക്രമപ്പെടുത്തുന്നതിനുള്ള WHERE ഉപവാക്യവും, " = " ഓപ്പറേറ്ററും അടങ്ങിയിരിക്കുന്നു. അതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 9.9 കാണിച്ചിരിക്കുന്നു.

ക്വറി 9.12

```
SELECT * FROM student
WHERE gender='F';
```

```
+--------+-------------+--------+------------+-------------+----------+
| adm_no | name        | gender | dob        | course      | f_income |
+--------+-------------+--------+------------+-------------+----------+
|   1005 | Meera       | F      | 1998-08-15 | Science     |     NULL |
|   1026 | Niveditha   | F      | 1999-03-04 | Humanities  |    52000 |
|   1057 | Chaithanya  | F      | 1999-06-03 | Science     |     NULL |
+--------+-------------+--------+------------+-------------+----------+
3 rows in set (0.06 sec)
```

ചിത്രം 9. 9: വിദ്യാർഥിനികളുടെ ലിസ്റ്റ് (ക്വറി 9.12 ന്റെ ഔട്ട്പുട്ട്)

നമുക്ക് വരുമാനം 25000 രൂപയിൽ താഴെയുള്ള സയൻസ് ഗ്രൂപ്പിലെ വിദ്യാർഥികളുടെ പേര്, കോഴ്സ്, മാസ വരുമാനം എന്നിവ കാണണം എന്നു വിചാരിക്കുക. ഇവിടെ നമുക്ക് രണ്ട് നിബന്ധനകൾ നിർമ്മിക്കേണ്ടതായി വരും. ഒന്ന് വരുമാനം പരിശോധിക്കുന്നതിനും, മറ്റൊന്ന് കോഴ്സ് പരിശോധിക്കുന്നതിനുമാണ്. ഇനി മുകളിൽ പറഞ്ഞ രണ്ടു നിബന്ധനകളും പട്ടികയിലെ റെക്കോർഡുകളുമായി യോജിക്കുന്നുണ്ടെങ്കിൽ name, course, f_income എന്നീ നിരകൾ പ്രദർശിപ്പിക്കുന്നതിന് വീണ്ടെടുക്കുന്നു (ക്വറി 9.13). ഇതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 9.10 ൽ കാണിച്ചിരിക്കുന്നു. രണ്ടു നിബന്ധനകൾ യോജിപ്പിക്കുന്നതിനു വേണ്ടി AND ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നു.

ക്വറി 9.13

```
SELECT name, course, f_income FROM student
WHERE course='Science' AND f_income<25000;
```

സയൻസ് ഗ്രൂപ്പിൽ ഉൾപ്പെടാത്ത (കോമേഴ്സ്, ഹ്യൂമാനിറ്റീസ് ഗ്രൂപ്പുകൾ) വിദ്യാർഥികളുടെ name, course, f_income എന്നിവ പ്രദർശിപ്പിക്കാനാവശ്യമായ ക്വറി എഴുതുക.

```
+-----------+---------+----------+
| name      | course  | f_income |
+-----------+---------+----------+
| Alok      | Science |    24000 |
| Virat     | Science |    22000 |
| Sreekumar | Science |    15000 |
+-----------+---------+----------+
3 rows in set (0.00 sec)
```

ചിത്രം 9.10: AND ഓപ്പറേറ്ററിന്റെ ഉപയോഗം

ഈ ക്വറി എഴുതുമ്പോൾ നിബന്ധന തയാറാക്കുന്നതിനായി OR ഓപ്പറേറ്റർ നിങ്ങൾ ഉപയോഗിച്ചേക്കാം. ഇതേ വിവരങ്ങൾ NOT ഓപ്പറേറ്റർ ഉപയോഗിച്ചാലും ലഭിക്കും. ക്വറി 9.14 ഇത് വിവരിക്കുന്നു. ചിത്രം 9.11 ൽ അതിന്റെ ഔട്ട്പുട്ട് പ്രദർശിപ്പിച്ചിരിക്കുന്നു

ക്വറി 9.14

```
SELECT name, course, f_income FROM student
WHERE NOT course='Science';
```

stock എന്ന പട്ടികയുമായി ബന്ധപ്പെട്ട് നമുക്ക് ചില നിബന്ധനകൾ ഉണ്ടാക്കാം, അനുയോജ്യമായ ഓപ്പറേറ്ററുകൾ ഉപയോഗിച്ച് ഇതിനുള്ള ക്വറികൾ തയാറാക്കാൻ ശ്രമിക്കാം.

```
+-----------+------------+----------+
| name      | course     | f_income |
+-----------+------------+----------+
| Bharath   | Commerce   |    45000 |
| Kaushi    | Commerce   |    17000 |
| Niveditha | Humanities |    52000 |
+-----------+------------+----------+
3 rows in set (0.00 sec)
```

ചിത്രം 9.11: NOT ഓപ്പറേറ്ററിന്റെ ഉപയോഗം

SQL ൽ നിബന്ധനകൾക്കായി പ്രത്യേകം ഓപ്പറേറ്ററുകൾ ഉണ്ട്. BETWEEN....AND, IN, LIKE, IS മുതലായവ

ഇതിൽ ഉൾപ്പെടുന്നു. നിബന്ധനകൾ തയാറാക്കുന്നതിന് ഈ ഓപ്പറേറ്ററുകൾ സഹായിക്കുന്നത് എങ്ങനെയെന്ന് നമുക്ക് ചർച്ച ചെയ്യാം.

a. വിലകളുടെ പരിധികൾ അടിസ്ഥാനമാക്കിയുള്ള നിബന്ധനകൾ

ഒരു പരിധിയിലുള്ള മൂല്യങ്ങളെ നിബന്ധനകളായി നൽകാം. പരിധി നിർദേശിക്കുന്നതിന് SQL ൽ BETWEEN..AND എന്ന ഓപ്പറേറ്റർ ഉപയോഗിക്കാം. കുടുംബ വരുമാനം Rs 25000 മുതൽ Rs 45000 വരെയുള്ള വിദ്യാർഥികളുടെ ലിസ്റ്റ് നമുക്ക് ആവശ്യമുണ്ടെന്ന് വിചാരിക്കുക. ക്വറി 9.15 ൽ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന ഉപയോഗിച്ച് ഇത് ലഭ്യമാക്കാം. ചിത്രം 9.12 ൽ ഇതിന്റെ ഔട്ട്പുട്ട് കാണിച്ചിരിക്കുന്നു.

ക്വറി 9.15

```
SELECT name, f_income FROM student
WHERE f_income>=25000 AND f_income<=45000;
```

ഇതേ ഔട്ട്പുട്ട് ക്വറി 9.16 ൽ ഉള്ള പ്രസ്താവന ഉപയോഗിച്ച് ലഭിക്കുന്നതാണ്. ഈ പ്രസ്താവനയിൽ നിബന്ധന ഉണ്ടാക്കാൻ BETWEEN... AND ഓപ്പറേറ്റർ ഉപയോഗിച്ചിട്ടുണ്ട്. പരിധിയിൽ കൊടുത്തിരിക്കുന്ന കുറഞ്ഞ വിലയും കൂടിയ വിലയും ഔട്ട്പുട്ടിൽ ഉൾപ്പെട്ടിരിക്കുന്നു

ക്വറി 9.16

```
SELECT name, f_income FROM student

WHERE f_income BETWEEN 25000 AND 45000;
```

സംഖ്യാതരത്തിലുള്ള ഡാറ്റയുടെ പരിധി BETWEEN ... AND ഓപ്പറേറ്റർ അനുവദിക്കുന്നുവെന്ന് ക്വറി 9.16 ൽ നിന്നും ചിത്രം 9.12 ൽ നിന്നും നമുക്ക് അനുമാനിക്കാൻ കഴിയും.

```
+---------+----------+
| name    | f_income |
+---------+----------+
| Nike    |    35000 |
| Bharath |    45000 |
+---------+----------+
2 rows in set (0.00 sec)
```

ചിത്രം 9.12: BETWEEN..AND ഓപ്പറേറ്ററിന്റെ ഉപയോഗം

b. വിലകളുടെ ലിസ്റ്റ് അടിസ്ഥാനമാക്കിയുള്ള നിബന്ധനകൾ

റെക്കോർഡുകൾ വീണ്ടെടുക്കുന്നതിനു വേണ്ടിയുള്ള നിബന്ധനകൾ തയാറാക്കുമ്പോൾ വിലകളുടെ ലിസ്റ്റ് ഉപയോഗിക്കാവുന്നതാണ്. വിലകൾ ഏതു ഡാറ്റ ഇനമായാലും നിർദിഷ്ട നിരയിലെ നിബന്ധനയുമായി യോജിക്കുന്നതായിരിക്കണം. ഈ അവസരത്തിൽ ലിസ്റ്റിനോടുകൂടി IN എന്ന ഓപ്പറേറ്റർ ഉപയോഗിക്കാം. സയൻസിലെയും ഹ്യൂമാനിറ്റീസിലെയും വിദ്യാർഥികളുടെ വിശദാംശങ്ങൾ നമുക്ക് വീണ്ടെടുക്കണമെന്നിരിക്കട്ടെ, ക്വറി 9.17 ലെ പ്രസ്താവനയ്ക്ക് ചിത്രം 9.13 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ വിശദാംശങ്ങൾ നിർമ്മിക്കാൻ കഴിയും.

ക്വറി 9.17

```
SELECT * FROM student
WHERE course='Commerce' OR course='Humanities';
```

```
+--------+------------+--------+------------+------------+----------+
| adm_no | name       | gender | dob        | course     | f_income |
+--------+------------+--------+------------+------------+----------+
|   1003 | Bharath    | M      | 1999-01-01 | Commerce   |    45000 |
|   1025 | Kaushi     | M      | 1998-10-02 | Commerce   |    17000 |
|   1026 | Niveditha  | F      | 1999-03-04 | Humanities |    52000 |
+--------+------------+--------+------------+------------+----------+
3 rows in set (0.00 sec)
```

ചിത്രം 9.13: ക്വറി 9.18 ന്റെ ഔട്ട്പുട്ട്

ഇതേ ഔട്ട്പുട്ട് ക്വറി 9.18 ൽ കാണുന്ന രീതിയിൽ IN ഓപ്പറേറ്റർ ഉപയോഗിച്ച് താഴെക്കാണുന്ന രീതിയിൽ ലഭ്യമാക്കാം.

ക്വറി 9.18

```
SELECT * FROM student
WHERE course IN('Commerce', 'Humanities');
```

ക്വറി 9.18 ൽ നാം കാണുന്നതുപോലെ IN ഓപ്പറേറ്റർ ഒരു റെക്കോർഡിലെ നിർദിഷ്ട നിരയിലെ (ഇവിടെ course) വില തന്നിരിക്കുന്ന ലിസ്റ്റിലെ ഏതെങ്കിലും വിലയോട് തുല്യത ഉണ്ടോ എന്ന് പരിശോധിക്കുന്നു. റെക്കോർഡുകളിലൂടെ കടന്നുപോകുമ്പോൾ തുല്യ തുക കണ്ടെത്തിയാൽ അവ പ്രദർശിപ്പിക്കുന്നു. ഈ പട്ടികയിൽ മൂന്ന് കോഴ്സ് മാത്രമേ ഉള്ളൂ. ക്വറി 9.19 ഉപയോഗിക്കുമ്പോൾ ചിത്രം 9.13 ൽ കാണിച്ചിരിക്കുന്ന അതേ ഫലം കാണിക്കുന്നതുമാണ്.

ക്വറി 9.19

```
SELECT * FROM student
WHERE course NOT IN('Science');
```

c. പാറ്റേൺ ചേർച്ച അടിസ്ഥാനമാക്കിയുള്ള നിബന്ധനകൾ

ചില മാതൃകകളെ അടിസ്ഥാനമാക്കി ഡാറ്റകൾ വീണ്ടെടുക്കേണ്ടതായി വരാം. അതായത് സ്ട്രിംഗ് ഡാറ്റയിലെ അക്ഷരങ്ങളിലുള്ള സാദൃശ്യമായിരിക്കും റെക്കോർഡുകൾ വീണ്ടെടുക്കുന്നതിനുള്ള മാനദണ്ഡം. SQL ൽ ഈ ആവശ്യത്തിനായി LIKE എന്ന പാറ്റേൺ മാച്ചിങ് ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നു. പാറ്റേണുകളെ രണ്ടു വൈൽഡ് കാർഡ് അക്ഷരങ്ങളായ % ഉം _ (അണ്ടർ സ്കോർ) ഉം ഉപയോഗിച്ച് നിർദേശിക്കാം. % (പെർസെന്റ്) സബ് സ്ട്രിങ്ങിന്റെ തുല്യത പരിശോധിക്കുന്നു. _ (അണ്ടർ സ്കോർ) ഒരു അക്ഷരത്തിന്റെ തുല്യത പരിശോധിക്കുന്നു. പാറ്റേണുകൾ കേസ് സെൻസിറ്റീവ് ആണ്, അതായത് വലിയ അക്ഷരങ്ങൾ ചെറിയ അക്ഷരങ്ങളുമായി തുല്യത പ്രാപിക്കില്ല. താഴെ പറയുന്ന ഉദാഹരണങ്ങൾ പരിഗണിക്കുക.

- "Ab%" എന്നത് Ab എന്ന അക്ഷരങ്ങളിൽ തുടങ്ങുന്ന ഏതു സ്ട്രിങ്ങിനും തുല്യമാണ്.
- "%cat%" എന്നത് "cat" എന്ന വാക്ക് ഉപസ്ട്രിങ്ങുമായി വരുന്ന ഏതു സ്ട്രിങ്ങുമായും തുല്യത പാലിക്കുന്നു. ഉദാഹരണത്തിന് "education", "indication", "catering" തുടങ്ങിയവ.
- "_ _ _ _" എന്നത് ഇടയിൽ ശൂന്യ സ്ഥലം ഇല്ലാത്ത നാല് അക്ഷരങ്ങൾ മാത്രമുള്ള ഏത് സ്ട്രിങ്ങിനും തുല്യമാണ്.

- "___%" എന്നത് ഏറ്റവും കുറഞ്ഞത് മൂന്ന് അക്ഷരങ്ങൾ ഉള്ള ഏത് സ്ട്രിങ്ങുമായും തുല്യത പാലിക്കും.

ക്വറി 9.20 `LIKE` ഓപ്പറേറ്ററിന്റെ ഉപയോഗം ക്വറി 9.20 വിവരിക്കുന്നു. പട്ടികയിലെ 'ar' എന്ന് അവസാനിക്കുന്ന പേരുകളുടെ ഒരു ലിസ്റ്റ് അത് നൽകുന്നു ചിത്രം 9.14 ൽ ഈ ക്വറിയുടെ ഔട്ട്പുട്ട് കാണിച്ചിരിക്കുന്നു

ക്വറി 9.20

```
SELECT name FROM student
WHERE name LIKE '%ar';
```

```
+-----------+
| name      |
+-----------+
| Divakar   |
| Sreekumar |
+-----------+
2 rows in set (0.00 sec)
```

ചിത്രം 9.14: പാറ്റേൺ ചേർച്ചയുടെ ഉപയോഗം

ക്വറി 9.21 'Divakar' എന്ന ഔട്ട്പുട്ട് തരുന്നു. കാരണം രണ്ട് _ (അണ്ടർ സ്കോർ) അക്ഷരങ്ങളാണ് പാറ്റേണിന്റെ തുല്യത പരിശോധിക്കുന്നത്.

ക്വറി 9.21

```
SELECT name FROM student
WHERE name LIKE 'Div__ar';
```

d. NULL വില പരിശോധന അടിസ്ഥാനമാക്കിയുള്ള നിബന്ധനകൾ

റെക്കോർഡുകളിൽ ചില ഫീൽഡുകളിൽ `NULL` വില ഉണ്ടായിരിക്കാമെന്നു നാം കണ്ടു കഴിഞ്ഞു. ഇങ്ങനെയുള്ള റെക്കോർഡുകളെ `IS` ഓപ്പറേറ്ററുകളുടെ സഹായത്തോടെ വീണ്ടെടുക്കാം. റെക്കോർഡിലെ നിർദിഷ്ട നിരയിൽ `NULL` മൂല്യങ്ങൾ വന്നാൽ നിബന്ധനകൾ ശരിയാകും. `student` എന്ന പട്ടികയിൽ `f_income` എന്ന നിരയിൽ `NULL` മൂല്യം മൂന്ന് വരികളിൽ ഉണ്ട്. (ചിത്രം 9.7 പരിശോധിക്കുക) ക്വറി 9.22 ഈ റെക്കോർഡുകൾ വീണ്ടെടുക്കുന്നു. ചിത്രം 9.15 ഇതിന്റെ ഔട്ട്പുട്ട് പ്രദർശിപ്പിക്കുന്നു.

```
+------------+---------+
| name       | course  |
+------------+---------+
| Meera      | Science |
| Divakar    | Science |
| Chaithanya | Science |
+------------+---------+
3 rows in set (0.00 sec)
```

ചിത്രം 9.15: NULL വില പരിശോധന

ക്വറി 9.22

```
SELECT name,course FROM student
WHERE f_income IS NULL;
```

`f_income` നിരയിൽ `NULL` മൂല്യങ്ങൾ ഇല്ലാത്ത റെക്കോർഡുകൾ വീണ്ടെടുക്കണമെന്നിരിക്കട്ടെ, ക്വറി 9.23 അത് നിർവഹിക്കുന്നു.

ക്വറി 9.23

```
SELECT name,course FROM student
WHERE f_income IS NOT NULL ;
```

നമുക്കു ചെയ്യാം

stock എന്ന പട്ടികയെ അടിസ്ഥാനപ്പെടുത്തി SQL ഓപ്പറേറ്ററുകൾ ഉപയോഗിക്കാനുള്ള ക്വറികൾ നിർമ്മിക്കുക.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. ഒരു നിരയിൽ ആവർത്തനങ്ങൾ ഒഴിവാക്കാൻ SELECT കമാൻഡിന്റെ കൂടെ ഉപയോഗിക്കുന്ന കീവേർഡിന്റെ പേരെഴുതുക.
2. SELECT ക്വറിയുടെ കൂടെ ഉപയോഗിക്കുന്ന പ്രധാന ഉപവാക്യം ഏത്?
3. ഒരു നിരയിൽ NULL മൂല്യങ്ങൾ ഉണ്ടോ എന്ന് പരിശോധിക്കുന്നതിനുള്ള ഓപ്പറേറ്റർ ഏത്?

   a. IN   b. LIKE   c. IS   d. NOT
4. പാറ്റേൺ ചേർച്ചയ്ക്ക് ഉപയോഗിക്കുന്ന ഓപ്പറേറ്ററാണ് __________.
5. താഴെപ്പറയുന്ന പ്രസ്താവനയിലെ തെറ്റ് എന്ത്?

   SELECT * FROM emp WHERE grade = NULL;
6. ഒരു പട്ടികയിൽ നിന്ന് റെക്കോർഡുകൾ തിരഞ്ഞെടുക്കുന്നതിനുള്ള കമാൻഡാണ് ______________.

9.6.3 ORDER BY ഉപവാക്യം ഉപയോഗിച്ച് ഉത്തരങ്ങളുടെ ക്രമീകരണം (Sorting results using ORDER BY clause)

ഇതുവരെ ചർച്ച ചെയ്ത SELECT ക്വറിയുടെ ഫലങ്ങളിലെല്ലാം പട്ടികയിലുള്ള അതെ ക്രമത്തിലാണ് റെക്കോർഡുകൾ എപ്പോഴും ലഭിച്ചിട്ടുള്ളതെന്ന് നാം കണ്ടു. അവയെ ഏതെങ്കിലും രീതിയിൽ ക്രമീകരിക്കാൻ സാധിക്കുമോ? ഒരു ക്വറിയുടെ ഉത്തരത്തെ ആരോഹണക്രമത്തിലോ അവരോഹണക്രമത്തിലോ ക്രമീകരിക്കാൻ കഴിയും. ഇതിനായി ORDER BY എന്ന ഉപവവാക്യം ഉപയോഗിക്കാം. ക്രമീകരണത്തിന് ASC (ആരോഹണക്രമം) അല്ലെങ്കിൽ DESC (അവരോഹണക്രമം) എന്ന കീവേർഡ് ഉപയോഗിക്കാം. തനതു രീതിയിൽ പ്രദർശിപ്പിക്കുന്നത് ആരോഹണക്രമത്തിലായിരിക്കും. പട്ടികയിലുള്ള വരികളുടെ ക്രമത്തിൽ മാറ്റം വരുന്നില്ല. പകരം സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നവ മാത്രമാണ് ഈ രീതിയിൽ ക്രമീകരിക്കുന്നത്. ക്വറി 9.24 ഉപയോഗിച്ച് വിദ്യാർഥികളുടെ വിശദാംശങ്ങൾ അവരുടെ പേരിന്റെ അക്ഷരമാലാ ക്രമത്തിലാക്കാം. ചിത്രം 9.16 ഇതിന്റെ ഔട്ട്പുട്ട് കാണിക്കുന്നു.

ക്വറി 9.24

```
SELECT * FROM student ORDER BY name;
```

```
+--------+------------+--------+------------+------------+----------+
| adm_no | name       | gender | dob        | course     | f_income |
+--------+------------+--------+------------+------------+----------+
|   1001 | Alok       | M      | 1998-10-02 | Science    |    24000 |
|   1003 | Bharath    | M      | 1999-01-01 | Commerce   |    45000 |
|   1057 | Chaithanya | F      | 1999-06-03 | Science    |     NULL |
|   1006 | Divakar    | M      | 1998-02-21 | Science    |     NULL |
|   1025 | Kaushi     | M      | 1998-10-02 | Commerce   |    17000 |
|   1005 | Meera      | F      | 1998-08-15 | Science    |     NULL |
|   1002 | Nike       | M      | 1998-11-26 | Science    |    35000 |
|   1026 | Niveditha  | F      | 1999-03-04 | Humanities |    52000 |
|   1027 | Sreekumar  | M      | 1998-06-06 | Science    |    15000 |
|   1004 | Virat      | M      | 1998-12-05 | Science    |    22000 |
+--------+------------+--------+------------+------------+----------+
10 rows in set (0.00 sec)
```

ചിത്രം 9.16: വിദ്യാർഥികളുടെ വിശദാംശങ്ങൾ അവരുടെ പേരിന്റെ അക്ഷരമാലാ ക്രമത്തിൽ

കുടുംബത്തിന്റെ മാസവരുമാനത്തിനനുസരിച്ച് വിദ്യാർഥികളുടെ വിശദാംശങ്ങൾ നമുക്ക് ലഭിക്കണമെങ്കിൽ ക്വറി 9.25 ഉപയോഗിക്കാം. ചിത്രം 9.17 ൽ ഔട്ട്പുട്ട് കാണിച്ചിരിക്കുന്നു.

ക്വറി 9.25

```
SELECT * FROM student
ORDER BY f_income DESC;
```

```
+--------+------------+--------+------------+------------+----------+
| adm_no | name       | gender | dob        | course     | f_income |
+--------+------------+--------+------------+------------+----------+
|   1026 | Niveditha  | F      | 1999-03-04 | Humanities |    52000 |
|   1003 | Bharath    | M      | 1999-01-01 | Commerce   |    45000 |
|   1002 | Nike       | M      | 1998-11-26 | Science    |    35000 |
|   1001 | Alok       | M      | 1998-10-02 | Science    |    24000 |
|   1004 | Virat      | M      | 1998-12-05 | Science    |    22000 |
|   1025 | Kaushi     | M      | 1998-10-02 | Commerce   |    17000 |
|   1027 | Sreekumar  | M      | 1998-06-06 | Science    |    15000 |
|   1006 | Divakar    | M      | 1998-02-21 | Science    |     NULL |
|   1005 | Meera      | F      | 1998-08-15 | Science    |     NULL |
|   1057 | Chaithanya | F      | 1999-06-03 | Science    |     NULL |
+--------+------------+--------+------------+------------+----------+
10 rows in set (0.00 sec)
```

ചിത്രം 9.17: വിദ്യാർഥികളുടെ വരുമാനത്തിന്റെ അടിസ്ഥാനത്തിലുള്ള ലിസ്റ്റ് (കൂടിയത് മുതൽ കുറഞ്ഞതു വരെ)

ചിത്രം 9.17 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ORDER BY ഉപവാക്യത്തോടൊപ്പം ഉപയോഗിച്ചിരിക്കുന്ന നിരയിൽ NULL ഉണ്ടെങ്കിൽ അവ പട്ടികയുടെ അവസാനം പ്രദർശിപ്പിക്കും. ഒന്നിലധികം ക്രമീകരണങ്ങൾ ORDER BY ഉപവാക്യം ഉപയോഗിച്ച് നടത്താൻ സാധിക്കും. ഉദാഹരണത്തിന് വിദ്യാർഥികളെ അവരുടെ കോഴ്സ് അനുസരിച്ച്, പേരിന്റെ അക്ഷരമാലാ ക്രമത്തിലാക്കണം എന്ന് വിചാരിക്കുക. ക്വറി 9.26 ൽ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന ഇതിനു വേണ്ടി ഉപയോഗിക്കാം. ചിത്രം 9.18 ൽ ഇതിന്റെ ഔട്ട്പുട്ട് കാണിച്ചിരിക്കുന്നു.

ക്വറി 9.26

```
SELECT name,course FROM student
ORDER BY course,
name;
```

ചിത്രം 9.18 ലുള്ള ഔട്ട്പുട്ട് നോക്കുക. ആദ്യം course എന്ന നിരയിലെ വിലകളും, പിന്നീട് ഓരോ കോഴ്സിലെ പേരുകളും അക്ഷരമാലാ ക്രമത്തിലാക്കിയിരിക്കുന്നു.

നേരത്തെ നാം WHERE ഉപവാക്യം ഉപയോഗിച്ച് നിബന്ധനകളുടെ അടിസ്ഥാനത്തിൽ വരികൾ വീണ്ടെടുക്കുന്നതെങ്ങനെ എന്ന് മനസ്സിലാക്കി. ഈ വരികൾ ഒരു പ്രത്യേക രീതിയിൽ ORDER BY ഉപവാക്യം ഉപയോഗിച്ച് ക്രമത്തിൽ പ്രദർശിപ്പിക്കാവുന്നതാണ്. ഇനി സയൻസ് ഗ്രൂപ്പിലെ

```
+------------+------------+
| name       | course     |
+------------+------------+
| Bharath    | Commerce   |
| Kaushi     | Commerce   |
| Niveditha  | Humanities |
| Alok       | Science    |
| Chaithanya | Science    |
| Divakar    | Science    |
| Meera      | Science    |
| Nike       | Science    |
| Sreekumar  | Science    |
| Virat      | Science    |
+------------+------------+
10 rows in set (0.00 sec)
```

ചിത്രം 9.18: പല ഘടകങ്ങൾ ചേർന്ന ക്രമീകരണം

കുട്ടികളുടെ പേര്, കുടുംബ വരുമാനം എന്നിവ വരുമാനത്തിന്റെ അവരോഹണ ക്രമത്തിൽ നമുക്ക് പ്രദർശിപ്പിക്കണമെന്നിരിക്കട്ടെ. ഈ ഔട്ട്പുട്ട് ലഭിക്കുന്നതിന് ക്വറി 9.27 ൽ കാണിച്ചിരിക്കുന്നതു പോലെ രണ്ട് ഉപവാക്യങ്ങൾ സംയോജിപ്പിക്കാവുന്നതാണ്. ഔട്ട്പുട്ട് ചിത്രം 9.19 കാണിച്ചിരിക്കുന്നു.

ക്വറി 9.27

```
SELECT name,course,f_income FROM student
WHERE course= 'Science'
ORDER BY f_income DESC;
```

ചിത്രം 9.19 കാണിച്ചിരിക്കുന്നതുപോലെ ക്രമീകരണ നിരയിൽ NULL വിലയുള്ള റെക്കോർഡുകൾ അവസാനം കാണപ്പെടും. മറ്റൊരു പ്രധാനപ്പെട്ട കാര്യം ORDER BY ഉപവാക്യം WHERE ഉപവാക്യത്തിനു ശേഷമാണ് SELECT ൽ ഉപയോഗിക്കേണ്ടത്. കാരണം റെക്കോർഡുകൾ ആദ്യം തിരഞ്ഞെടുത്തതിനു ശേഷമാണ് അവ ക്രമീകരിക്കുന്നത്.

name	course	f_income
Nike	Science	35000
Alok	Science	24000
Virat	Science	22000
Sreekumar	Science	15000
Meera	Science	NULL
Divakar	Science	NULL
Chaithanya	Science	NULL

7 rows in set (0.00 sec)

ചിത്രം 9.19: നിബന്ധന അടിസ്ഥാനമാക്കിയുള്ള തിരഞ്ഞെടുക്കലും ക്രമീകരിക്കലും

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. ORDER BY ഉപവാക്യത്തിന്റെ അർഥമെന്ത്?
2. ഡാറ്റ അവരോഹണക്രമത്തിൽ ക്രമീകരിക്കുന്നതിന് MySQL ൽ ഉപയോഗിക്കുന്ന കീവേർഡ് ഏത്?

   a. DESC b. ASC c. SORT d. MODIFY
3. ORDER BY ഉപവാക്യത്തിന്റെ അഭാവത്തിൽ വരികൾ ഏതു ക്രമത്തിലാണ് പ്രദർശിപ്പിക്കപ്പെടുന്നത്?
4. SQLൽ ORDER BY ഉപവാക്യത്തിന്റെ തനതു ക്രമീകരണ രീതി എങ്ങനെയായിരിക്കും?

നമുക്കു ചെയ്യാം

stock എന്ന പട്ടിക അടിസ്ഥാനമാക്കി വിവിധ നിബന്ധനകൾ ഉപയോഗിച്ച് റെക്കോർഡുകളെ ക്രമീകരിക്കാനുമുള്ള ക്വറികൾ നിർമിക്കുക.

9.6.4 സംഗ്രഹ ഫങ്ഷനുകൾ (Aggregate functions)

ഒരു പട്ടികയിൽ മുഴുവനായോ WHERE ഉപയോഗിച്ച് പട്ടികയിൽ നിന്ന് വേർതിരിച്ചെടുത്ത ഒരു ഉപഗണത്തിനോ ഉപയോഗിക്കാവുന്ന ധാരാളം ബിൽട്ട് ഇൻ ഫങ്ഷനുകൾ MySQLൽ ഉണ്ട്. ഈ ഫങ്ഷനുകളെ സംഗ്രഹ ഫങ്ഷനുകൾ എന്നു പറയുന്നു. കാരണം ഇവ വരികളുടെ സംഗ്രഹത്തിനായി പ്രവർത്തിക്കുന്നു. ഈ ഫങ്ഷനുകളുടെ ഫലം ഒരൊറ്റ വിലയായിരിക്കും. സാധാരണയായി ഉപയോഗിക്കുന്ന സംഗ്രഹ ഫങ്ഷനുകൾ പട്ടിക 9.7 ൽ കൊടുത്തിരിക്കുന്നു.

ഫങ്ഷൻ	തിരികെത്തരുന്ന വില
SUM()	ആർഗ്യുമെന്റായി കൊടുത്തിരിക്കുന്ന നിരയിലുള്ള വിലകളുടെ തുക
AVG()	ആർഗ്യുമെന്റായി കൊടുത്തിരിക്കുന്ന നിരയിലുള്ള വിലകളുടെ ശരാശരി
MIN()	ആർഗ്യുമെന്റായി കൊടുത്തിരിക്കുന്ന നിരയിലെ ഏറ്റവും ചെറിയ വില
MAX()	ആർഗ്യുമെന്റായി കൊടുത്തിരിക്കുന്ന നിരയിലെ ഏറ്റവും വലിയ വില.
COUNT()	ആർഗ്യുമെന്റായി കൊടുത്തിരിക്കുന്ന നിരയിലെ NULL അല്ലാത്ത വിലകളുടെ എണ്ണം

പട്ടിക 9.7: MySQL ലെ ചില ബിൽട്ട് ഇൻ ഫങ്ഷനുകൾ

ക്വറി 9.28 വിദ്യാർഥിയുടെ കുടുംബ വരുമാനത്തിന്റെ ഏറ്റവും കൂടിയത്, കുറഞ്ഞത്, ശരാശരി എന്നിവ നൽകുന്നു. ചിത്രം 9.20 ഇതിന്റെ ഫലം കാണിച്ചിരിക്കുന്നു.

ക്വറി 9.28

```
SELECT MAX(f_income),MIN(f_income),AVG(f_income )
FROM student;
```

ഈ വിലകൾ കണക്കാക്കുമ്പോൾ സംഗ്രഹ നിരയിലെ NULL വിലകൾ പരിഗണിക്കുകയില്ല.

```
+---------------+---------------+---------------+
| MAX(f_income) | MIN(f_income) | AVG(f_income) |
+---------------+---------------+---------------+
|         52000 |         15000 |    30000.0000 |
+---------------+---------------+---------------+
1 row in set (0.03 sec)
```

ചിത്രം 9.20: സംഗ്രഹ ഫങ്ഷനുകളുടെ ഉപയോഗം

ക്വറി 9.29 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ചില തിരഞ്ഞെടുക്കൽ മാനദണ്ഡത്തിന്റെ അടിസ്ഥാനത്തിൽ രൂപീകൃതമായ പട്ടികയിലെ ഒരു ഉപഗണത്തിൽ ഈ ഫങ്ഷനുകൾ ഉപയോഗിക്കാവുന്നതാണ്. ഇത് സയൻസ് ഗ്രൂപ്പിലെ വിദ്യാർഥികളുടെ എണ്ണം തരുന്നു. ചിത്രം 9.19 ൽ അതിന്റെ ഔട്ട്പുട്ട് കാണിച്ചിരിക്കുന്നു.

ക്വറി 9.29

```
SELECT  COUNT(*), COUNT(f_income)
FROM student WHERE course = 'Science';
```

ചിത്രം 9.19 നാം പരിശോധിച്ചാൽ കൃത്യമായ ഉത്തരം 7 എന്ന് കാണാൻ കഴിയും. പിന്നെ എന്തുകൊണ്ടാണ് ചിത്രം 9.21 ലെ വിലകൾ തമ്മിൽ തുല്യമല്ലാത്തത്? ആദ്യത്തെ നിര COUNT (*) ആണ്. അത് course എന്ന നിരയിൽ സയൻസ് എന്ന വില ഉള്ള റെക്കോർഡു

കളുടെ എണ്ണം തരുന്നു. നക്ഷത്ര ചിഹ്നം (*) പട്ടികയിലെ എല്ലാ നിരകളെയും സൂചിപ്പിക്കുന്നു. അതുകൊണ്ട് ഒരു റെക്കോർഡിൽ ഏറ്റവും കുറഞ്ഞത് ഒരു ഫീൽഡെങ്കിലും ഉണ്ടെങ്കിൽ COUNT (*) ആ റെക്കോർഡ് പരിഗണിക്കും. പക്ഷെ ക്വറി 9.29ലെ COUNT (f_income), സയൻസിലെ f_income എന്ന നിരയിലെ NULL വിലകൾ കണക്കിലെടുക്കുന്നില്ല. അതുകൊണ്ടാണ് ചിത്രം 9.21 ലെ രണ്ടാമത്തെ നിരയിൽ 4 എന്ന വില കാണാൻ കഴിയുന്നത്. ചിത്രം 9.17 പരിശോധിക്കുക അതിൽ f_income എന്ന നിരയിലെ NULL ന്റെ എണ്ണം തിരിച്ചറിയുക.

```
+----------+------------------+
| COUNT(*) | COUNT(f_income)  |
+----------+------------------+
|        7 |                4 |
+----------+------------------+
1 row in set (0.03 sec)
```

ചിത്രം 9.21: COUNT() ഫങ്ഷന്റെ ഉപയോഗം

9.6.5 GROUP BY ഉപവാക്യം ഉപയോഗിച്ച് തരം തിരിക്കൽ (Grouping of records using GROUP BY clause)

ചിലപ്പോൾ മൂല്യങ്ങളുടെ അടിസ്ഥാനത്തിൽ പട്ടികയിലെ വരികൾ വിവിധ ഗണങ്ങളാക്കി നമുക്ക് വിഭജിക്കേണ്ടി വരും. പൊതുവിലയുടെ അടിസ്ഥാനത്തിൽ GROUP BY ഉപയോഗിച്ച് ഒരു പട്ടികയിലെ വരികൾ തരംതിരിക്കാം. GROUP BY ഉപവാക്യത്തിൽ കൊടുത്തിരിക്കുന്ന ആട്രിബ്യൂട്ട് ഉപയോഗിച്ച് ഗ്രൂപ്പുകൾ രൂപീകരിക്കാം. GROUP BY ഉപവാക്യത്തിൽ കൊടുത്തിരിക്കുന്ന ആട്രിബ്യൂട്ടിൽ ഒരേ വിലയുള്ള വരികൾ ഒരു ഗ്രൂപ്പിൽ കാണപ്പെടും. നിരയിലെ വ്യത്യസ്ത വിലകളുടെ അടിസ്ഥാനത്തിലാണ് ഗ്രൂപ്പുകൾ രൂപീകൃതമാകുന്നത്. അതിനാൽ ഈ പ്രക്രിയ റെക്കോർഡുകളുടെ തരംതിരിക്കലായി കണക്കാക്കാം. ക്വറി 9.29 ഉപയോഗിച്ച് സയൻസ് ഗ്രൂപ്പിലെ കുട്ടികളുടെ എണ്ണം ലഭിച്ചു. ഇനി ഓരോ ഗ്രൂപ്പിലെയും കുട്ടികളുടെ എണ്ണവും അതോടൊപ്പം കുടുംബത്തിന്റെ ശരാശരി മാസവരുമാനവും അറിയണമെന്ന് വിചാരിക്കുക. ക്വറി 9.30 ഉപയോഗിക്കാം. ഔട്ട്പുട്ട് ചിത്രം 9.22ൽ കൊടുത്തിരിക്കുന്നു.

ക്വറി 9.30

```
SELECT course, COUNT(*), AVG(f_income)
FROM student GROUP BY course ;
```

```
+------------+----------+---------------+
| course     | COUNT(*) | AVG(f_income) |
+------------+----------+---------------+
| Commerce   |        2 |    31000.0000 |
| Humanities |        1 |    52000.0000 |
| Science    |        7 |    24000.0000 |
+------------+----------+---------------+
3 rows in set (0.00 sec)
```

ചിത്രം 9.22: GROUP BY ഉപയോഗിച്ചു കൊണ്ടുള്ള വരികളുടെ തരം തിരിക്കൽ

ഇവിടെ COURSE എന്ന നിരയിലെ വിവിധ വിലകളുടെ അടിസ്ഥാനത്തിൽ വിവിധ ഗ്രൂപ്പുകൾ രൂപീകരിച്ചിരിക്കുന്നു (കോമേഴ്സ്, ഹ്യൂമാനിറ്റീസ്, സയൻസ്). അതിനു ശേഷം COUNT (*), AVG(f_income) എന്നീ ഫങ്ഷനുകൾ ഇതിലെ ഓരോ ഗ്രൂപ്പിലും ഉപയോഗിച്ചിരിക്കുന്നു.

9.6.6 HAVING ഉപവാക്യത്തിലൂടെ നിബന്ധനകൾ പ്രയോഗിച്ച് ഗ്രൂപ്പുകൾ രൂപീകരിക്കുന്നു (Applying conditions to form groups using HAVING clause)

ഈ ഉപവാക്യം GROUP BY ഉപവാക്യത്തോടൊപ്പം ഉപയോഗിക്കുന്നു. ഓരോ വരിയിലും നിബന്ധനകൾ ഉപയോഗിക്കുന്ന WHERE ഉപവാക്യത്തെക്കുറിച്ചു ഇതിനോടകം പഠിച്ചു കഴിഞ്ഞു. HAVING ഉപവാക്യത്തിന്റെ സഹായത്തോടുകൂടി നിബന്ധനകൾ പ്രയോഗിച്ച് നമുക്ക് ഗ്രൂപ്പുകൾ രൂപീകരിക്കാവുന്നതാണ്. HAVING ഉപവാക്യത്തിലെ നിബന്ധന ഒരു കൂട്ടം റെക്കോർഡുകൾ രൂപീകരിക്കുന്നതിനാണ് ഉപയോഗിക്കുന്നത്, അല്ലാതെ വരികളിൽ ഒറ്റയ്ക്കല്ല. ക്വറി 9.31 ഗ്രൂപ്പുണ്ടാക്കുന്നതിനു വേണ്ട നിബന്ധനകൾ നൽകുന്നു. HAVING ഉപവാക്യത്തിന്റെ കൂടെ നിബന്ധനകൾ നൽകുമ്പോൾ മാത്രമാണ് ഇവിടെ ഗ്രൂപ്പുകൾ രൂപീകൃതമാകുന്നത്.

ക്വറി 9.31

```
SELECT course, COUNT(*)
GROUP BY course
HAVING COUNT(*)>3;
```

ചിത്രം 9.22 ഉം, 9.23 ഉം പരിശോധിച്ചാൽ നമുക്ക് സയൻസ് ഗ്രൂപ്പിൽ മാത്രമെ 3 വിദ്യാർഥികളിൽ കൂടുതൽ ഉള്ളൂ എന്ന് കാണാൻ കഴിയും. കൊമേഴ്സ്, ഹ്യുമാനിറ്റീസ് ഗ്രൂപ്പുകൾ ക്വറി 9.31 ന്റെ ഔട്ട്പുട്ടിൽ വന്നിട്ടില്ല. കാരണം, ഇവ ഓരോന്നിന്റെയും റെക്കോർഡുകളുടെ എണ്ണം യഥാക്രമം 2, 1 എന്നിങ്ങനെയാണ്. അതിനാലാണ് ഈ ഗ്രൂപ്പുകൾ ക്വറി 9.31 ന്റെ ഉത്തരത്തിൽ കാണാൻ സാധിക്കാത്തത്.

```
+---------+----------+
| course  | COUNT(*) |
+---------+----------+
| Science |        7 |
+---------+----------+
1 row in set (0.00 sec)
```

ചിത്രം 9.23: നിബന്ധനകളുടെ അടിസ്ഥാനത്തിൽ ഗ്രൂപ്പുകൾ തിരിക്കുന്നു

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. SQL ലെ സംഗ്രഹണത്തിനുപയോഗിക്കുന്ന ഫങ്ഷനുകൾ പട്ടികപ്പെടുത്തുക?
2. COUNT(*), COUNT(column_name) എന്നിവ എങ്ങനെ വ്യത്യാസപ്പെട്ടിരിക്കുന്നു?
3. WHERE, HAVING ഉപവാക്യങ്ങൾ തമ്മിലുള്ള വ്യത്യാസം എന്ത്?
4. താഴെ പറയുന്ന ക്വറിയുടെ ഉത്തരം എന്ത്?

```
SELECT COUNT (DISTINCT course) FROM student;
```

നമുക്കു ചെയ്യാം

stock എന്ന പട്ടികയുടെ അടിസ്ഥാനത്തിൽ സംഗ്രഹ ഫങ്ഷനുകളും ORDER BY ഉപവാക്യവും ഉപയോഗിച്ച് ക്വറികൾ തയാറാക്കുക.

9.7 പട്ടികയിലെ ഡാറ്റയുടെ പരിഷ്കരണം (Modifying data in tables)

ചില സന്ദർഭങ്ങളിൽ ഒരു പട്ടികയിലെ നിരകളുടെ വിലകളിൽ മാറ്റം വരുത്തേണ്ടതായി വരാറുണ്ട്. ഉദാഹരണത്തിന് ശമ്പളം അല്ലെങ്കിൽ കൂലി പുതുക്കുമ്പോൾ വിദ്യാർഥിയുടെ കുടുംബത്തിന്റെ മാസവരുമാനത്തിൽ മാറ്റം വരാറുണ്ട്. അതുപോലെ ചില വിദ്യാർഥികളുടെ മാസവരുമാനം വിട്ടുപോയത് (NULL) പിന്നീട് സാധുവായ ഡാറ്റ നൽകി മാറ്റം വരുത്താവുന്നതാണ്. ഇങ്ങനെയുള്ള മാറ്റങ്ങൾ DML കമാൻഡായ UPDATE ഉപയോഗിച്ച് നിർവഹിക്കാം. അത് ഒന്നോ അതിലധികമോ നിർദിഷ്ട നിരകളിലെ വിലകളിൽ മാറ്റം വരുത്തുന്നതിന് ഉപയോഗിക്കാം. ഈ മാറ്റം പട്ടികയിലെ തിരഞ്ഞെടുത്ത വരികളിൽ മാത്രമെ പ്രതിഫലിക്കൂ. ഒരു വരിയിലെ നിരയിൽ പുതിയ ഡാറ്റ കൊണ്ടുവരുന്നതിന് SET കീവേർഡ് ഉപയോഗിക്കുന്നു. ഇത് UPDATE കമാൻഡിന്റെ കൂടെ നിർബന്ധമായും ഉപയോഗിക്കേണ്ട ഒരു ഉപവാക്യമാണ്. പുതിയ ഡാറ്റ ഒരു സ്ഥിര സംഖ്യയോ, പ്രയോഗമോ (Expression) മറ്റൊരു പട്ടികയിലെ ഡാറ്റയോ ആകാം. UPDATE കമാൻഡിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
UPDATE <table_name> SET <column_name> = <value>
[,<column_name> = <value>,. . . ]
[WHERE <condition>] ;
```

'Kaushi' എന്ന വിദ്യാർഥിയുടെ കുടുംബത്തിന്റെ മാസവരുമാനം Rs 27000 ലേക്ക് മാറ്റണമെന്ന് വിചാരിക്കുക. ക്വറി 9.32 ഇതിനായി ഉപയോഗിക്കുന്നു.

ക്വറി 9.32

```
UPDATE student
SET f_income = 27000
WHERE name = 'Kaushi' ;
```

ഈ ക്വറിയുടെ പ്രവർത്തനത്തിനു ശേഷം MySQL ൽ നിന്ന് താഴെക്കാണുന്ന രീതിയിലുള്ള പ്രതികരണം ലഭിക്കുന്നു.

```
Query O.K, 1 row affected (0.08 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

റെക്കോഡിൽ വന്ന മാറ്റം കാണുന്നതിനു വേണ്ടി താഴെ പറയുന്ന ക്വറി ഉപയോഗിക്കാം

```
SELECT * FROM student WHERE name = 'Kaushi'
```

ചിത്രം 9.24 ൽ ഈ ക്വറിയുടെ ഔട്ട്പുട്ട് കാണിച്ചിരിക്കുന്നു. ഈ ക്വറിയുടെ ഔട്ട്പുട്ട് ചിത്രം 9.17 മായി താരതമ്യം ചെയ്ത് മാറ്റങ്ങൾ മനസ്സിലാക്കുക.

```
+--------+--------+--------+------------+----------+----------+
| adm_no | name   | gender | dob        | course   | f_income |
+--------+--------+--------+------------+----------+----------+
|   1025 | Kaushi | M      | 1998-10-02 | Commerce |    27000 |
+--------+--------+--------+------------+----------+----------+
1 row in set (0.00 sec)
```

ചിത്രം 9.24: f_income നിരയിൽ പരിഷ്കരിച്ച ഡാറ്റയോടുകൂടിയ Kaushi യുടെ വിശദാംശങ്ങൾ

റെക്കോർഡിലെ നിരകൾക്ക് വില നൽകാൻ നമുക്ക് എക്സ്പ്രഷൻ ഉപയോഗിക്കാം. ക്വറി 9.33 ഈ ആശയം വിവരിക്കുന്നു.

ക്വറി 9.33

```
UPDATE student
SET f_income =  f_income + 1000
WHERE  f_income < 25000;
```

ഈ ക്വറിയോടുള്ള പ്രതികരണം താഴെ കൊടുക്കുന്നു.

```
Query O.K, 3 rows affected (0.06 sec)
Rows matched: 3 Changed: 3 Warnings: 0
```

3 വരികളാണ് തന്നിരിക്കുന്ന നിബന്ധനകൾ പാലിക്കുന്നത്. അതിനാൽ f_ income എന്ന നിരയിലെ ഈ വരികളെ 1000 വച്ച് വർധിപ്പിക്കുന്നു (ചിത്രം 9.17, 9.25 എന്നിവ പരിശോധിക്കുക)

NULL മൂല്യമുള്ള നിരകളിൽ മാറ്റം വരുത്തുന്നതിന് UPDATE കമാൻഡുകൾ ഉപയോഗിക്കാം. ഇപ്പോൾ NULL വിലയുള്ള നിരയിൽ മാസവരുമാനം 20000 രൂപ നമുക്ക് ചേർക്കണമെന്നിരിക്കട്ടെ. ക്വറി 9.34 അത് സാധ്യമാക്കുന്നു.

ക്വറി 9.34

```
UPDATE student
SET f_income =  20000
WHERE  f_income IS NULL;
```

ക്വറി 9.33 ലും 9.34 ലും വരുത്തിയ മാറ്റങ്ങൾ ചിത്രം 9.25 ൽ നമുക്ക് കാണാൻ കഴിയും. അതിന് SELECT *FROM student; എന്ന ക്വറി ഉപയോഗിക്കുക

adm_no	name	gender	dob	course	f_income
1001	Alok	M	1998-10-02	Science	25000
1002	Nike	M	1998-11-26	Science	35000
1003	Bharath	M	1999-01-01	Commerce	45000
1004	Virat	M	1998-12-05	Science	23000
1005	Meera	F	1998-08-15	Science	20000
1006	Divakar	M	1998-02-21	Science	20000
1025	Kaushi	M	1998-10-02	Commerce	27000
1026	Niveditha	F	1999-03-04	Humanities	52000
1027	Sreekumar	M	1998-06-06	Science	16000
1057	Chaithanya	F	1999-06-03	Science	20000

10 rows in set (0.00 sec)

ചിത്രം 9.25: f income നിരയിലെ തിരുത്തലുകൾ

നമുക്ക് ചെയ്യാം

stock എന്ന പട്ടിക ഉപയോഗിച്ച് ചില നിരകളിലെ വിലകളിൽ മാറ്റങ്ങൾ വരുത്താനാവശ്യമായ SQL പ്രസ്താവനകൾ എഴുതുക.

9.8 ഒരു പട്ടികയുടെ രൂപഘടന മാറ്റുന്നു (Changing the structure of a table)

ചില സാഹചര്യങ്ങളിൽ നമുക്ക് പട്ടികയുടെ രൂപഘടനയിൽ മാറ്റങ്ങൾ വരുത്തേണ്ടതായി വരാം. ഈ പ്രവർത്തനം രൂപരേഖ (schema) യുമായി ബന്ധപ്പെട്ടതാണ്. അതിന് **DDL** കമാൻഡായ `ALTER TABLE` ഉപയോഗിക്കാം. നിരകൾ കൂട്ടിച്ചേർക്കൽ, ഒഴിവാക്കൽ, നിലവിലുള്ള നിരയുടെ ഡാറ്റ ഇനത്തിലും വലുപ്പത്തിലും മാറ്റം വരുത്തുക, പട്ടികയ്ക്ക് പുതിയ പേര് നൽകുക തുടങ്ങിയ പ്രവർത്തനങ്ങൾക്ക് ഈ നിർദേശം ഉപയോഗിക്കാം. ഇനി ഈ മാറ്റങ്ങൾ എങ്ങനെയാണ് വരുത്തുന്നത് എന്ന് നമുക്ക് നോക്കാം.

9.8.1 പുതിയ നിര കൂട്ടിച്ചേർക്കുന്നു (Adding a new column)

ഒന്നോ അതിലധികമോ നിരകൾ ഒരു പട്ടികയിൽ എവിടെ വേണമെങ്കിലും കൂട്ടിച്ചേർക്കാവുന്നതാണ്. പുതിയ നിര കൂട്ടിച്ചേർക്കുന്നതിനുള്ള (`ALTER TABLE`ന്റെ) വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
ALTER TABLE <table_name>
ADD <column_name> <data_type> [<size>] [<constraint>]
[FIRST | AFTER <column_name> ] ;
```

ഇവിടെ <table_name> എന്നത് ഏത് പട്ടികയിലാണോ മാറ്റം വരുത്തേണ്ടത് അതിന്റെ പേരാണ്. `ADD` എന്നത് ഒരു പട്ടികയിൽ പുതിയ നിരകൾ കൂട്ടിച്ചേർക്കാനുള്ള കീ വേഡാണ്. <column_name> <data_type> [<size>] എന്നിവ പുതിയ നിരയുടെ വിശദാംശങ്ങളാണ്. `FIRST | AFTER` എന്നത് പുതുതായി ചേർക്കുന്ന നിരയുടെ സ്ഥാനം സൂചിപ്പിക്കുന്നു.

പുതിയ നിര ആദ്യമാണ് ചേർക്കേണ്ടത് എങ്കിൽ നമുക്ക് `FIRST` എന്ന ഉപവാക്യം ഉപയോഗിക്കാം. പുതിയ നിര പ്രത്യേകസ്ഥാനത്താണ് ചേർക്കേണ്ടതെങ്കിൽ AFTER <column_name> എന്ന ഉപവാക്യം ഉപയോഗിക്കണം. പുതിയ നിരയുടെ സ്ഥാനം നാം വ്യക്തമാക്കിയിട്ടില്ലെങ്കിൽ അത് പട്ടികയുടെ അവസാനനിരയായി കൂട്ടിച്ചേർക്കപ്പെടും. പുതുതായി കൂട്ടിച്ചേർക്കുന്ന നിരകളിൽ `NULL` മൂല്യങ്ങളാണ് ഉണ്ടാവുക എന്നത് ശ്രദ്ധിക്കുക.

ഇനി നമുക്ക് `gr_mks, reg_no` എന്നീ രണ്ട് നിരകൾ student പട്ടികയിൽ കൂട്ടിച്ചേർക്കാം. ഇവ യഥാക്രമം ഗ്രേസ്മാർക്കും വിദ്യാർഥിയുടെ ഹയർ സെക്കന്ററി പരീക്ഷയിലെ രജിസ്റ്റർ നമ്പറും സംഭരിക്കുന്നതിനുള്ളതാണ്. ക്വറി 9.35 ന് പട്ടികയുടെ രൂപ ഘടനയിൽ ഈ മാറ്റങ്ങൾ വരുത്താൻ കഴിയും.

ക്വറി 9.35

```
ALTER TABLE student
ADD gr_mks INTEGER
AFTER dob, ADD reg_no INTEGER;
```

`MySQL` ന്റെ പ്രതികരണം താഴെ കൊടുക്കുന്നു.

```
Query O.K, 10 rows affected (0.25 sec)
Rows matched: 10 Changed: 10 Warnings: 0
```

9.8.2 ഒരു നിരയുടെ നിർവചനത്തിൽ മാറ്റം വരുത്തുന്നു (Chaning the definition of a column)

ഒരു നിരയുടെ സവിശേഷതകളായ ഡാറ്റ ഇനം, വലുപ്പം അല്ലെങ്കിൽ കൺസ്ട്രയിന്റ് എന്നിവയ്ക്ക് മാറ്റം വരുത്തുന്നതിന് ALTER TABLE ന്റെ കൂടെ MODIFY എന്ന ഉപവാക്യം ഉപയോഗിക്കാം. ഇതിനുള്ള വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
ALTER TABLE <table_name>
MODIFY <column_name> <data_type> [<size>] [<constraint>];
```

ഇനി പുതുതായി കൂട്ടിച്ചേർത്ത reg_no എന്ന നിരയിൽ UNIQUE എന്ന ഉപവാക്യം നൽകി രണ്ടു കുട്ടികൾക്ക് ഒരേ രജിസ്റ്റർ നമ്പർ ഉണ്ടായിരിക്കില്ല എന്ന് ഉറപ്പുവരുത്തുവാൻ സാധിക്കുന്നതാണ്. ക്വറി 9.36 ഈ മാറ്റം വരുത്തുന്നു.

ക്വറി 9.36

```
ALTER TABLE student
MODIFY reg_no INTEGER UNIQUE;
```

MySQL ന്റെ പ്രതികരണവും ക്വറി 9.35 ൽ നമുക്ക് ലഭിച്ച പ്രതികരണവും ഒന്നു തന്നെയായിരിക്കും. നമുക്ക് ഈ മാറ്റങ്ങൾ DESC student; എന്ന കമാൻഡ് ഉപയോഗിച്ച് മനസ്സിലാക്കാവുന്നതാണ്. ക്വറി 9.35 ലും 9.36 ലും വരുത്തിയിട്ടുള്ള മാറ്റങ്ങൾ നമുക്ക് വീക്ഷിക്കാവുന്നതാണ് ചിത്രം 9.26 ൽ ഔട്ട്പുട്ട് കാണിച്ചിട്ടുണ്ട്.

```
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| adm_no    | int(11)      | NO   | PRI | NULL    | auto_increment |
| name      | varchar(20)  | NO   |     | NULL    |                |
| gender    | char(1)      | YES  |     | M       |                |
| dob       | date         | YES  |     | NULL    |                |
| gr_mks    | int(11)      | YES  |     | NULL    |                |
| course    | varchar(15)  | YES  |     | NULL    |                |
| f_income  | int(11)      | YES  |     | NULL    |                |
| reg_no    | int(11)      | YES  | UNI | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

ചിത്രം 9.26: student പട്ടികയിൽ മാറ്റം വരുത്തിയതിനുശേഷമുള്ള രൂപഘടന

gr_mks എന്ന നിര dob ന് ശേഷവും reg_no എന്ന നിര അവസാന നിരയായും കൂട്ടിച്ചേർത്തതായി ചിത്രം 9.26 ൽ നമുക്ക് കാണാൻ സാധിക്കും. reg_no ന് നേരെ Key നിരയിൽ UNIQUE കൺസ്ട്രയിന്റ് ഉൾപ്പെട്ടിരിക്കുന്നതു കൂടി ശ്രദ്ധിക്കുക. SELECT *FROM student എന്ന കമാൻഡ് ഉപയോഗിച്ചാൽ ചിത്രം 9.27 ൽ കാണുന്നതുപോലെ പുതിയ നിരകളുടെ ഉള്ളടക്കത്തിൽ NULL വിലകൾ കാണാൻ കഴിയും. പുതുതായി നിർമ്മിച്ച നിരകളിൽ ഇതുവരെ വിലകൾ കൊടുക്കാത്തതു കൊണ്ടാണ് ഇങ്ങനെ സംഭവിച്ചത്.

adm_no	name	gender	dob	gr_mks	course	f_income	reg_no
1001	Alok	M	1998-10-02	NULL	Science	25000	NULL
1002	Nike	M	1998-11-26	NULL	Science	35000	NULL
1003	Bharath	M	1999-01-01	NULL	Commerce	45000	NULL
1004	Virat	M	1998-12-05	NULL	Science	23000	NULL
1005	Meera	F	1998-08-15	NULL	Science	20000	NULL
1006	Divakar	M	1998-02-21	NULL	Science	20000	NULL
1025	Kaushi	M	1998-10-02	NULL	Commerce	27000	NULL
1026	Niveditha	F	1999-03-04	NULL	Humanities	52000	NULL
1027	Sreekumar	M	1998-06-06	NULL	Science	16000	NULL
1057	Chaithanya	F	1999-06-03	NULL	Science	20000	NULL

10 rows in set (0.00 sec)

ചിത്രം 9.27:പട്ടികയുടെ രൂപഘടന പരിഷ്കരിച്ചതിനു ശേഷമുള്ള പുതിയ നിരകളുടെ ഉള്ളടക്കം

നമുക്കു ചെയ്യാം

പുതുതായി കൂട്ടിച്ചേർത്ത നിരകളിൽ ഡാറ്റ ചേർക്കുന്നതിനും അത് പ്രദർശിപ്പിക്കുന്നതിനുമുള്ള SQL പ്രസ്താവനകൾ എഴുതുക. stock എന്ന പട്ടികയ്ക്ക് ഘടനാപരമായ മാറ്റങ്ങൾ വരുത്തി അതിൽ പുതിയ വിലകൾ ചേർക്കുക.

9.8.3 പട്ടികയിൽ നിന്ന് ഒരു നിര ഒഴിവാക്കുന്നു (Removing column from a table)

പട്ടികയിൽ നിന്ന് ഒരു നിര ഒഴിവാക്കണമെങ്കിൽ ALTER TABLE കമാൻഡിനോടൊപ്പം DROP എന്ന ഉപവാക്യം ഉപയോഗിക്കണം ഇതിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
ALTER TABLE <table_name>
DROP <column_name>;
```

ഉദാഹരണമായി student എന്ന പട്ടികയിൽ നിന്ന് gr_mks എന്ന നിര ഒഴിവാക്കണമെങ്കിൽ ക്വറി 9.37 ഉപയോഗിച്ചാൽ മതിയാകും.

ക്വറി 9.37

```
ALTER TABLE student
DROP gr_mks;
```

മുൻപ് ALTER TABLE പ്രസ്താവനകളിൽ നിന്ന് നമുക്ക് ലഭിച്ച അതേ പ്രതികരണമായിരിക്കും MySQL ൽ നിന്നും ലഭിക്കുക. മാറ്റങ്ങൾ DESC student; എന്ന കമാൻഡ് ഉപയോഗിച്ച് നിരീക്ഷിക്കാവുന്നതാണ്.

9.8.4 പട്ടിക പുനർനാമകരണം ചെയ്യുന്നു (Renaming a table)

ഡാറ്റാബേസിലെ ഒരു പട്ടിക പുനർനാമകരണം ചെയ്യുന്നതിന് RENAME TO എന്ന ഉപവാക്യം ALTER TABLE കമാൻഡിനോടൊപ്പം ഉപയോഗിച്ചാൽ മതി. പട്ടികയിൽ വരികളുണ്ടെങ്കിലും അതിനെ നമുക്ക് പുനർനാമകരണം ചെയ്യാം. പട്ടിക പുനർനാമകരണം ചെയ്യുന്നതിനുള്ള കമാൻഡിന്റെ വാക്യ ഘടന താഴെ കൊടുക്കുന്നു

```
ALTER TABLE <table_name>
RENAME TO <new_table_name>;
```

ഉദാഹരണത്തിന് student എന്ന പട്ടികയ്ക്ക് student2015 എന്ന പേര് മാറ്റിനൽകണമെങ്കിൽ ക്വറി 9.38 ഉപയോഗിക്കാം.

ക്വറി 9.38

```
ALTER TABLE student
RENAME TO student2015;
```

```
+-------------------+
| Tables_in_school  |
+-------------------+
| student2015       |
+-------------------+
1 row in set (0.00 sec)
```

ചിത്രം 9.28: പുനർനാമകരണം ചെയ്ത പട്ടിക

ഈ ക്വറിയുടെ പ്രതികരണം താഴെ കൊടുക്കുന്നു

Query O.K, 0 rows affected (0.06 sec)

പട്ടികയുടെ പേര് മാറിയെങ്കിലും അതിലെ ഉള്ളടക്കത്തിൽ മാറ്റം വന്നിട്ടില്ല എന്ന് പ്രതികരണം സൂചിപ്പിക്കുന്നു.

മാറ്റങ്ങൾ കാണുന്നതിന് SHOW TABLES എന്ന ക്വറി ഉപയോഗിക്കാം. ചിത്രം 9.28 ൽ ഔട്ട്പുട്ട് കാണിച്ചിരിക്കുന്നു.

9.9 പട്ടികയിൽ നിന്ന് വരികൾ ഒഴിവാക്കുന്നു (Deleting rows from a table)

ചിലപ്പോൾ ഒരു പട്ടികയിൽ നിന്ന് ഒന്നോ അതിൽക്കൂടുതലോ റെക്കോർഡുകൾ നമുക്ക് ഒഴിവാക്കേണ്ടതായി വരും. ഇതിനായി DML കമാൻഡായ DELETE ഉപയോഗിക്കുന്നു. ഒഴിവാക്കേണ്ട വരികൾ WHERE ഉപവാക്യം ഉപയോഗിച്ച് തിരഞ്ഞെടുക്കുന്നു. WHERE ഉപയോഗിച്ചിട്ടില്ലെങ്കിൽ പട്ടികയിലെ എല്ലാ റെക്കോർഡുകളും നഷ്ടപ്പെടും. DELETE കമാൻഡ് ഒഴിവാക്കുന്നത് റെക്കോർഡുകളെയാണ്. ഇതിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
DELETE FROM <table_name> [WHERE <condition>] ;
```

ഉദാഹരണത്തിന് അഡ്മിഷൻ നമ്പർ 1027 ആയ Sreekumar എന്ന വിദ്യാർഥിയുടെ റെക്കോർഡ് പട്ടികയിൽ നിന്ന് ഒഴിവാക്കണമെങ്കിൽ ക്വറി 9.39 ഉപയോഗിക്കാം.

ക്വറി 9.39

```
DELETE FROM student2015 WHERE adm_no = 1027 ;
```

ഈ ക്വറിയുടെ ഔട്ട്പുട്ട് താഴെ കൊടുക്കുന്നു:

```
Query O.K, 1 row affected (0.08 sec)
```

ഇവിടെ പട്ടികയുടെ പേര് കൊടുത്തിരിക്കുന്നത് student2015 എന്നാണ്. പട്ടികയിലെ മാറ്റം ചിത്രം 9.29 ൽ നിരീക്ഷിക്കാവുന്നതാണ്. അത് ലഭിക്കുന്നതിന് താഴെ കൊടുക്കുന്ന ക്വറി ഉപയോഗിക്കുക.

```
SELECT * FROM student2015;
```

```
+---------+-------------+--------+------------+------------+----------+--------+
| adm_no  | name        | gender | dob        | course     | f_income | reg_no |
+---------+-------------+--------+------------+------------+----------+--------+
|    1001 | Alok        | M      | 1998-10-02 | Science    |    25000 |   NULL |
|    1002 | Nike        | M      | 1998-11-26 | Science    |    35000 |   NULL |
|    1003 | Bharath     | M      | 1999-01-01 | Commerce   |    45000 |   NULL |
|    1004 | Virat       | M      | 1998-12-05 | Science    |    23000 |   NULL |
|    1005 | Meera       | F      | 1998-08-15 | Science    |    20000 |   NULL |
|    1006 | Divakar     | M      | 1998-02-21 | Science    |    20000 |   NULL |
|    1025 | Kaushi      | M      | 1998-10-02 | Commerce   |    27000 |   NULL |
|    1026 | Niveditha   | F      | 1999-03-04 | Humanities |    52000 |   NULL |
|    1057 | Chaithanya  | F      | 1999-06-03 | Science    |    20000 |   NULL |
+---------+-------------+--------+------------+------------+----------+--------+
9 rows in set (0.02 sec)
```

ചിത്രം 9.29: ഒരു റെക്കോർഡ് ഒഴിവാക്കിയതിനു ശേഷം പുനർനാമകരണം ചെയ്ത പട്ടികയിലെ ഉള്ളടക്കം

ചിത്രം 9.29 ൽ ഏഴു നിരകൾ മാത്രമെ കാണാൻ കഴിയൂ. കാരണം ക്വറി 9.37 ഉപയോഗിച്ച് `gr_mks` എന്ന നിര നേരത്തെ ഒഴിവാക്കിയിട്ടുണ്ട്.

ഒരു നിരയിൽ നിന്ന് മൂല്യങ്ങൾ ഒഴിവാക്കാൻ സാധിക്കുകയില്ല. ഇത് ഫലത്തിൽ കൊണ്ടുവരാൻ താഴെ കാണും വിധം UPDATE കമാൻഡ് ഉപയോഗിച്ച് നിരകളിൽ NULL മൂല്യങ്ങൾ ചേർക്കുകയാണ് വേണ്ടത്.

```
UPDATE <table_name>
      SET <column_name> = NULL
      [WHERE <condition>] ;
```

9.10 ഡാറ്റാബേസിൽ നിന്ന് പട്ടിക ഒഴിവാക്കുന്നു (Removing table from a database)

ഡാറ്റാബേസിൽ നിന്ന് ഒരു പട്ടിക ഒഴിവാക്കണമെങ്കിൽ DROP TABLE എന്ന കമാൻഡ് ഉപയോഗിച്ചാൽ മതി. ഈ DDL കമാൻഡ് പട്ടികയിൽ റെക്കോർഡുകൾ ഉണ്ടെങ്കിലും അതിനെ ഡാറ്റാബേസിൽ നിന്ന് സ്ഥിരമായി നീക്കം ചെയ്യുന്നു. ഈ കമാൻഡ് ഉപയോഗിക്കുമ്പോൾ വളരെ ശ്രദ്ധിക്കണം. കാരണം, ഒരിക്കൽ പട്ടിക നീക്കം ചെയ്താൽ അത് തിരിച്ചെടുക്കാൻ സാധിക്കുകയില്ല. ഈ കമാൻഡിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
DROP TABLE <table_name>;
```

ഉദാഹരണത്തിന് school ഡാറ്റാബേസിൽ നിന്ന് student2015 എന്ന പട്ടിക ഒഴിവാക്കണമെങ്കിൽ ക്വറി 9.40 ഉപയോഗിക്കാം.

ക്വറി 9.40

```
DROP TABLE student2015;
```

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. ഒരു പട്ടികയുടെ ഘടനയിൽ മാറ്റം വരുത്താനുപയോഗിക്കുന്ന കമാൻഡാണ് _______.
2. ഒരു നിരയുടെ ഘടനയിൽ മാറ്റം വരുത്തിയാൽ അതിലെ വിലകളെ അത് ബാധിക്കുന്നു. ഈ പ്രസ്താവന ശരിയോ തെറ്റോ?
3. ഒരു പട്ടികയിൽ നിന്ന് എങ്ങനെയാണ് ഒരു നിര ഒഴിവാക്കുന്നത്?
4. ഒരു പട്ടികയിൽ നിന്ന് വരി ഒഴിവാക്കാനുള്ള കമാൻഡ് ഏത്?
5. DELETE FROM എന്ന കമാൻഡിന്റെ കൂടെ WHERE ഉപവാക്യം ഉപയോഗിക്കാതിരുന്നാൽ എന്ത് സംഭവിക്കും?

9.11 നെസ്റ്റഡ് ക്വറികൾ (Nested queries)

'നെസ്റ്റിങ്' എന്ന ആശയം നമുക്ക് സുപരിചിതമാണല്ലോ. 'നെസ്റ്റഡ് if', 'നെസ്റ്റഡ് ലൂപ്പ്' എന്നിവയിൽ ഇത് നാം ചർച്ച ചെയ്തിരുന്നു. ഇനി നമുക്ക് നെസ്റ്റഡ് ക്വറികളെക്കുറിച്ച് ചർച്ച ചെയ്യാം. നെസ്റ്റഡ് എന്നാൽ 'ഒന്നിനുള്ളിൽ മറ്റൊന്ന്' എന്നാണ് അർഥമാക്കുന്നത്. ഇവിടെ ഒരു ക്വറിയുടെ ഫലം ഓപ്പറേഷൻ സമയത്തു തന്നെ മറ്റൊരു ക്വറിയുടെ നിബന്ധനയിലേക്ക് കൈമാറ്റം ചെയ്യപ്പെടുന്നു. ഒരു MySQL ന്റെ ഉള്ളിലുള്ള ക്വറിയെ ഉപക്വറി (inner query, subquery) എന്നും, ആ ഉപക്വറി അടങ്ങിയിരിക്കുന്ന ക്വറിയെ ബാഹ്യക്വറി (outer query) എന്നും പറയുന്നു. SQL ആദ്യം WHERE ഉപവാക്യം അടങ്ങിയ ഉപക്വറി ആദ്യം പ്രവർത്തിപ്പിക്കുന്നു. അതിന്റെ ഫലം ബാഹ്യക്വറിയുടെ നിബന്ധനയിലേക്ക് കൈമാറുന്നു. റിലേഷണൽ ഓപ്പറേറ്റർ ഉപയോഗിക്കുമ്പോൾ ഉപക്വറി ഒറ്റവരി ഔട്ട്പുട്ട് തിരികെ നൽകുന്നുവെന്ന് ഉറപ്പുവരുത്തേണ്ടതാണ്.

ഏറ്റവും കൂടുതൽ കുടുംബവരുമാനമുള്ള വിദ്യാർഥിയുടെ പേരും കോഴ്സും നമുക്ക് പ്രദർശിപ്പിക്കണമെന്നിരിക്കട്ടെ. ക്വറി 9.41 ഉപയോഗിച്ച് നമുക്ക് ഈ കാര്യം ചെയ്യാവുന്നതാണ്. ചിത്രം 9.30 ൽ ഇതിന്റെ ഔട്ട്പുട്ട് കാണിച്ചിരിക്കുന്നു.

ക്വറി 9.41

```
SELECT name,course FROM student2015
WHERE f_income = (SELECT MAX(f_income)
 FROM student2015);
```

ഈ ഉദാഹരണത്തിൽ ഉപക്വറി ആദ്യം വിലയിരുത്തി. അതിൽ നിന്ന് f_income എന്ന നിരയിലെ ഏറ്റവും വലിയ വില തിരികെ ലഭിക്കുന്നു (ഇവിടെ ഇത് Rs 52000). ഈ മൂല്യം പിന്നീട് പട്ടികയിലെ f_income എന്ന റെക്കോർഡിന്റെ മൂല്യവുമായി താരതമ്യം ചെയ്ത് തുല്യമാണെങ്കിൽ ബാഹ്യക്വറി അത് വീണ്ടെടുക്കുന്നു.

```
+-----------+------------+
| name      | course     |
+-----------+------------+
| Niveditha | Humanities |
+-----------+------------+
1 row in set (0.03 sec)
```

ചിത്രം 9.30: നെസ്റ്റഡ് ക്വറിയുടെ ഫലം

9.12 വ്യൂ എന്ന ആശയം (Concept of views)

MySQL വ്യൂ എന്ന ആശയത്തെ പിന്തുണയ്ക്കുന്നു. ഇത് RDBMS ലെ ഒരു സവിശേഷതയാണ്. വ്യൂ എന്നത് ഒരു സാങ്കൽപ്പിക പട്ടികയാണ് (Virtual table). യഥാർഥത്തിൽ ഇവ ഡാറ്റാബേസിൽ ഇല്ല. പക്ഷേ, ഒന്നോ അതിലധികമോ പട്ടികകളിൽ നിന്നും ഉരുത്തിരിച്ചെടുക്കാവുന്നതാണ്. ഏത് പട്ടികയിൽ നിന്നാണോ വ്യൂ നിർമ്മിക്കുന്നതിനാവശ്യമായ വരികൾ ശേഖരിക്കുന്നത് അതിനെ അടിസ്ഥാനപ്പട്ടിക (Base table) എന്നു പറയുന്നു. ഈ വരികൾ ഭൗതികമായി എവിടെയും സംഭരിക്കുന്നില്ല പകരം വ്യൂവിന്റെ നിർവചനം ഡാറ്റാബേസിനകത്ത് സംഭരിക്കുന്നു. അടിസ്ഥാന പട്ടികയിൽ സംഭരിച്ചിരിക്കുന്ന, നമുക്കാവശ്യമുള്ള വിവരങ്ങൾ കാണാൻ കഴിയുന്ന ഒരു ജാലകം പോലെയാണ് വ്യൂ. ഒരു ക്വറിയുടെ നിബന്ധന അടിസ്ഥാനമാക്കി മറ്റു പട്ടികകളിൽ നിന്നും വ്യൂവിലെ ഉള്ളടക്കം എടുക്കുന്നു. `CREATE VIEW` എന്ന DDL കമാൻഡ് ഉപയോഗിച്ച് വ്യൂ നിർമ്മിക്കാം. അതിന്റെ വാക്യഘടന താഴെ കൊടുക്കുന്നു.

```
CREATE VIEW <view_name>
AS SELECT <column_name 1> [, <column_name2>,...]
FROM <table_name>
[WHERE <condition> ];
```

ഇനി ജനുവരി 1,1999 ന് മുൻപ് ജനിച്ച കുട്ടികളുടെ വിവരങ്ങൾ ഉപയോഗിച്ച് ഒരു വ്യൂ ഉണ്ടാക്കാം. ക്വറി 9.42 student1998 എന്ന വ്യൂ നിർമിക്കുന്നു.

ക്വറി 9.42

```
CREATE VIEW student1998
AS SELECT * FROM student2015
WHERE dob < '1999-1-1';
```

ഈ ക്വറിയുടെ ഔട്ട്പുട്ട് താഴെക്കൊടുക്കുന്നു

```
Query O.K, 0 row affected (0.31 sec)
```

ചിത്രം 9.31 വ്യൂവിന്റെ ഘടന കാണിക്കുന്നു. ചിത്രം 9.32 വ്യൂവിന്റെ ഉള്ളടക്കം പ്രദർശിപ്പിക്കുന്നു. ഇത് ലഭിക്കുന്നതിന് യഥാക്രമം കമാൻഡുകളായ `DESC student1998;` , `SELECT * FROM student1998;` എന്നീ നിർദേശങ്ങൾ നൽകേണ്ടതാണ്.

```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| adm_no   | int(11)     | NO   |     | 0       |       |
| name     | varchar(20) | NO   |     | NULL    |       |
| gender   | char(1)     | YES  |     | M       |       |
| dob      | date        | YES  |     | NULL    |       |
| course   | varchar(15) | YES  |     | NULL    |       |
| f_income | int(11)     | YES  |     | NULL    |       |
| reg_no   | int(11)     | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

ചിത്രം 9.31: student1998 എന്ന വ്യൂവിന്റെ രൂപഘടന

adm_no	name	gender	dob	course	f_income	reg_no
1001	Alok	M	1998-10-02	Science	25000	NULL
1002	Nike	M	1998-11-26	Science	35000	NULL
1004	Virat	M	1998-12-05	Science	23000	NULL
1005	Meera	F	1998-08-15	Science	20000	NULL
1006	Divakar	M	1998-02-21	Science	20000	NULL
1025	Kaushi	M	1998-10-02	Commerce	27000	NULL

6 rows in set (0.00 sec)

ചിത്രം 9.32: student1998 എന്ന വ്യൂവിന്റെ ഉള്ളടക്കം

ചിത്രം 9.31ൽ കീ എന്ന നിരയിൽ `adm_no, reg_no` എന്നിവയുടെ കൺസ്ട്രയിന്റ് ഉൾപ്പെട്ടിട്ടില്ല. ചിത്രം 9.32 കാണിക്കുന്നത് 1998 ൽ ജനിച്ച വിദ്യാർഥികളുടെ വിശദാംശങ്ങളാണ്.

പട്ടികകളോടൊപ്പം DML കമാൻഡ് ഉപയോഗിക്കുന്നതുപോലെ വ്യൂ വിനോടൊപ്പം അവ ഉപയോഗിക്കാം. അങ്ങനെയുള്ള ക്വറിയുടെ ഉത്തരം ചിത്രം 9.32 ൽ കൊടുത്തിരിക്കുന്നു. അടിസ്ഥാന പട്ടികകൾ നാം ഉപയോഗിക്കുന്നത് അതിനോട് ചേർന്നുള്ള വ്യൂവിൽ കൂടിയാണ്. അതിനാൽ വ്യൂവിൽ `UPDATE, DELETE` എന്നീ പ്രവർത്തനങ്ങൾ ചെയ്യുമ്പോൾ അടിസ്ഥാന പട്ടികയിലും ആ മാറ്റങ്ങൾ ഉണ്ടാകുന്നു. ക്വറി 9.43 ഈ ആശയം വിവരിക്കുന്നു.

ക്വറി 9.43

```
UPDATE student1998
SET reg_no = 2201020
WHERE adm_no=1001 ;
```

adm_no	name	gender	dob	course	f_income	reg_no
1001	Alok	M	1998-10-02	Science	25000	2201020
1002	Nike	M	1998-11-26	Science	35000	NULL
1003	Bharath	M	1999-01-01	Commerce	45000	NULL
1004	Virat	M	1998-12-05	Science	23000	NULL
1005	Meera	F	1998-08-15	Science	20000	NULL
1006	Divakar	M	1998-02-21	Science	20000	NULL
1025	Kaushi	M	1998-10-02	Commerce	27000	NULL
1026	Niveditha	F	1999-03-04	Humanities	52000	NULL
1057	Chaithanya	F	1999-06-03	Science	20000	NULL

9 rows in set (0.00 sec)

ചിത്രം 9.33: വ്യൂവിലൂടെയുള്ള അടിസ്ഥാന പട്ടികയുടെ പരിഷ്ക്കരണം.

ഇനി നമുക്ക് ഈ ക്വറി പ്രവർത്തിപ്പിച്ച ശേഷം `student2015` എന്ന പട്ടികയിലെ ഡാറ്റ പരിശോധിക്കാം. `SELECT * FROM student2015;` എന്ന ക്വറിയുടെ ഔട്ട്പുട്ട് ചിത്രം 9.33ൽ കാണിച്ചിരിക്കുന്നു. സംഭരണസ്ഥലം പ്രത്യേകമായി ഉപയോഗിക്കാതെ ഒരേ പട്ടിക പല പട്ടികയായി ഉപയോഗിക്കാൻ സാധിക്കുന്നു എന്ന മേൻമ വ്യൂവിൽ അടങ്ങിയിരിക്കുന്നു. പങ്കുവയ്ക്കലിനോടൊപ്പം, സ്വകാര്യതയും വ്യൂ നടപ്പാക്കുന്നു എന്നുള്ളതാണ് മറ്റൊരു നേട്ടം. പട്ടികയിൽ നിന്ന് റെക്കോർഡുകൾ വീണ്ടെടുക്കുക, പരിഷ്ക്കരിക്കുക ഒഴിവാക്കുക എന്നിവയ്ക്കായി `WHERE` ഉപവാക്യത്തോടൊപ്പമുള്ള നിബന്ധനകളുടെ സങ്കീർണത കുറയ്ക്കുന്നതിന് ഇത് സഹായിക്കുന്നു.

ഒരു വ്യൂ ആവശ്യമില്ലെങ്കിൽ അത് ഡാറ്റാബേസിൽ നിന്ന് ഒഴിവാക്കുന്നതിന് DROP VIEW കമാൻഡ് ഉപയോഗിക്കാം. അത് അടിസ്ഥാന പട്ടികയെ ബാധിക്കുകയില്ല. ഇതിന്റെ വാക്യ ഘടന താഴെ കൊടുക്കുന്നു.

```
DROP VIEW <view_name>;
```

ഉദാഹരണത്തിന് student1998 എന്ന വ്യൂ പട്ടിക ഒഴിവാക്കണമെങ്കിൽ ക്വറി 9.44 ഉപയോഗിക്കാം.

ക്വറി 9.44

```
DROP VIEW student1998;
```

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. നെസ്റ്റഡ് ക്വറി എന്നാൽ എന്ത്?
2. SQL ലിലെ വ്യൂ എന്നാൽ എന്ത്?
3. വ്യൂ നിർമ്മിക്കുന്നതിന് SELECT കമാൻഡ് ആവശ്യമാണ്. ഈ പ്രസ്താവന ശരിയോ തെറ്റോ?
4. വ്യൂ ഒഴിവാക്കുമ്പോൾ ഡാറ്റാബേസിൽ നിന്ന് ഒരു പട്ടിക ഒഴിവാക്കുന്നു. ഇത് ശരിയോ തെറ്റോ എന്ന് പ്രസ്താവിക്കുക.
5. പട്ടിക പുതുക്കുവാൻ നമുക്ക് അതിന്റെ വ്യൂ ഉപയോഗിക്കാമോ?

നമുക്കു ചെയ്യാം

MySQL ൽ നടപ്പിലാക്കിയിട്ടുള്ള എല്ലാ DDL, DML കമാൻഡുകളും ഉദാഹരണങ്ങളും ഔട്ട്പുട്ട് സഹിതം നാം ചർച്ച ചെയ്തു. ഇനി പട്ടിക 9.8ൽ നമുക്ക് അവ സംഗ്രഹിക്കാം. ആദ്യത്തെ രണ്ടു വരികൾ എഴുതി ചേർത്തിട്ടുണ്ട്. ബാക്കിയുള്ള വരികൾ നിങ്ങൾക്കായി മാറ്റിവച്ചിരിക്കുന്നു.

SQL	കമാൻഡ്	അനിവാര്യമായ കീവേഡ്	കമാൻഡിന്റെ ഉദ്ദേശ്യം
DDL	CREATE TABLE		ഒരു പട്ടിക നിർമിക്കാൻ
	ALTER TABLE	ADD/MODIFY	ഒരു പട്ടികയുടെ ഘടന മാറ്റുന്നതിന്
	DROP TABLE		
	CREATE VIEW		
	DROP VIEW		
DML	INSERT		
	SELECT		
	UPDATE		
	DELETE		

പട്ടിക 9.8: SQL കമാൻഡിന്റെ സംഗ്രഹം

നമുക്ക് സംഗ്രഹിക്കാം

റിലേഷണൽ ഡാറ്റാബേസിൽ പ്രവർത്തനങ്ങൾ ചെയ്യാനുപയോഗിക്കുന്ന ഭാഷയാണ് സ്ട്രക്ചേർഡ് ക്വറി ലാംഗ്വേജ്. MySQL പ്രശസ്തമായ RDBMS പാക്കേജ് ആണ്. അത് ഉപയോഗിച്ച് നമുക്ക് നിർദേശങ്ങൾ നടപ്പിലാക്കാം. ഒരു പട്ടികയുടെ ഘടനയുമായി ബന്ധപ്പെട്ട പ്രവർത്തനങ്ങൾക്ക് വിവിധ DDL കമാൻഡുകൾ ഉണ്ട്. കൺസ്ട്രയിന്റുകൾ ഡാറ്റയുടെ സാധുതയും ഡാറ്റാബേസിന്റെ കെട്ടുറപ്പും ഉറപ്പു വരുത്തുന്നു. ഒരു പട്ടികയിലുള്ള ഡാറ്റയുമായി ബന്ധപ്പെട്ട വിവിധ പ്രവർത്തനങ്ങൾക്കുള്ള DML കമാൻഡുകൾ നാം ചർച്ച ചെയ്തു. ഡാറ്റ കൂട്ടിച്ചേർക്കൽ, വീണ്ടെടുക്കൽ, പുതുക്കൽ, ഒഴിവാക്കൽ തുടങ്ങിയ പ്രവർത്തനങ്ങൾ ഇതിൽ ഉൾപ്പെടും. നെസ്റ്റഡ് ക്വറി, വ്യൂ എന്നീ ആശയങ്ങളും എന്താണെന്ന് നാം പരിചയപ്പെട്ടു. ഈ പാഠഭാഗം നന്നായി മനസ്സിലാക്കേണ്ടത് കമ്പ്യൂട്ടർ മേഖലയിലെ നിങ്ങളുടെ ഉപരിപഠനത്തിന് ആവശ്യമാണ്.

നമുക്ക് വിലയിരുത്താം

1. പരീക്ഷയിൽ വിദ്യാർഥികൾ നേടിയ മാർക്കും വിശദാംശങ്ങളും സംഭരിക്കുന്നതിനുള്ള പട്ടികയുടെ ഘടന താഴെ കൊടുത്തിരിക്കുന്നു.

ഡാറ്റ	ഡാറ്റ ഇനം	വിവരണം
Register number	Numeric	വിദ്യാർഥിയെ തിരിച്ചറിയാൻ ആവശ്യമുള്ളതും ഏകവുമായ ഡാറ്റ
Name	String	പരമാവധി 30 അക്ഷരങ്ങൾ
Course	String	സയൻസ്, കോമേഴ്സ് അല്ലെങ്കിൽ ഹ്യൂമാനിറ്റീസ്
Marks of six subjects	Numeric each	ആറു വ്യത്യസ്ത നിരകൾ ആവശ്യമാണ്

പട്ടിക നിർമ്മിക്കുന്നതിനും താഴെ പറയുന്ന ആവശ്യങ്ങൾക്കുമുള്ള SQL പ്രസ്താവനകൾ എഴുതുക

a) ഡാറ്റ ചേർക്കുക (10 റെക്കോർഡുകൾ).

b) എല്ലാ വിദ്യാർഥികളുടെയും വിശദാംശങ്ങൾ പ്രദർശിപ്പിക്കുക.

c) സയൻസ് ഗ്രൂപ്പിലെ എല്ലാ കുട്ടികളുടെയും വിശദാംശങ്ങൾ പട്ടികപ്പെടുത്തുക.

d) ഓരോ കോഴ്സിലെയും കുട്ടികളുടെ എണ്ണം കണക്കാക്കുക.

e) ആകെ മാർക്ക് കണ്ടുപിടിച്ച് Total എന്ന നിര പുതുക്കുക.

f) ഓരോ വിദ്യാർഥിയുടെയും ആറു മാർക്കുകളുടെ തുക ചേർത്ത് Total എന്ന നിര പുതുക്കുക.

g) ഓരോ ഗ്രൂപ്പിലെയും ഏറ്റവും ഉയർന്ന Total പ്രദർശിപ്പിക്കുക.

h) കോമേഴ്സ് ഗ്രൂപ്പിലെ ആറാമത്തെ വിഷയത്തിലെ ഏറ്റവും കുറഞ്ഞത്, ഏറ്റവും കൂടിയത്, ശരാശരി സ്കോർ എന്നിവ കണ്ടെത്തുക.

i) അക്ഷരമാലാക്രമത്തിൽ ഓരോ കോഴ്സിലെയും പേരുകൾ പ്രദർശിപ്പിക്കുക.

j) ഏറ്റവും കൂടുതൽ Total സ്കോറുള്ള കുട്ടിയുടെ പേര് പ്രദർശിപ്പിക്കുക.

2. ഒരു കമ്പ്യൂട്ടർ ഷോപ്പിലെ ഇനങ്ങളുടെ വിശദാംശങ്ങൾ സംഭരിക്കുന്നതിനുള്ള പട്ടികയുടെ ഘടന താഴെ കൊടുക്കുന്നു.

ഡാറ്റ	ഡാറ്റ ഇനം	വിവരണം
Item number	Numeric	ഒരു വസ്തു തിരിച്ചറിയാൻ തനതായതും ഏകവുമായ ഡാറ്റ
Item name	String	പരമാവധി 30അക്ഷരങ്ങൾ
Date of purchase	Date	ആവർത്തനം അനുവദിക്കുന്നു
Unit price	Fractional Number	ഒരു ഇനത്തിന്റെ വില
Quantity	Numeric	ഇനങ്ങളുടെ എണ്ണം
Manufacturer	String	വിതരണക്കാരന്റെ പേര് (ആവർത്തനം അനുവദിക്കുന്നു)

പട്ടിക നിർമ്മിക്കുന്നതിനും താഴെ പറയുന്ന ആവശ്യങ്ങൾക്കുമുള്ള SQL പ്രസ്താവനകൾ എഴുതുക.

a) ഡാറ്റ ചേർക്കുക (10 റെക്കോർഡുകൾ).

b) എല്ലാ ഇനങ്ങളുടെയും വിശദാംശങ്ങൾ പ്രദർശിപ്പിക്കുക.

c) ഓരോ ഇനത്തിന്റെ പേരും അവയുടെ വിലയും പ്രദർശിപ്പിക്കുക.

d) പട്ടികയിൽ ലഭ്യമായ ഒരു കമ്പനി നിർമ്മിച്ച (കമ്പനിയുടെ പേര് നൽകുക) ഇനങ്ങളുടെ പേര് പട്ടികപ്പെടുത്തുക.

e) ഓരോ നിർമ്മാതാവിന്റെയും ഇനങ്ങളുടെ എണ്ണം എഴുതുക.

f) ഏറ്റവും കൂടുതൽ വിലയുള്ള ഇനത്തിന്റെ വിശദാംശങ്ങൾ പ്രദർശിപ്പിക്കുക.

g) എല്ലാ ഇനങ്ങളുടെയും ശരാശരി വിലയെക്കാൾ കൂടുതൽ വിലയുള്ള ഇനങ്ങളുടെ പേര് പട്ടികപ്പെടുത്തുക.

h) 1-1-2015 നു ശേഷം വാങ്ങിയ ഇനങ്ങളുടെ പേര് പ്രദർശിപ്പിക്കുക.

i) പട്ടികയിലെ രണ്ടോ മൂന്നോ കമ്പനികൾ (കമ്പനിയുടെ പേര് നൽകുക) നിർമ്മിച്ച ഇനങ്ങളുടെ വിശദാംശങ്ങൾ ലഭ്യമാക്കുക.

j) ഒരു കമ്പനിയിലെ 20 എണ്ണത്തിൽ കൂടുതൽ സ്റ്റോക്കുള്ള ഇനങ്ങളുടെ വിശദാംശങ്ങൾ പ്രദർശിപ്പിക്കുക.

3. ഹയർ സെക്കന്ററി സ്കൂളിലെ അധ്യാപകരുടെ വിശദാംശങ്ങൾ സംഭരിക്കുന്നതിനുള്ള പട്ടികയുടെ ഘടന കൊടുക്കുന്നു.

ഡാറ്റ	ഡാറ്റ ഇനം	വിവരണം
Teacher ID	Numeric	ടീച്ചറെ തിരിച്ചറിയാൻ തനതായതും ഏകവുമായ ഡാറ്റ
Name	String	പരമാവധി 30അക്ഷരങ്ങൾ
Gender	Character	ആൺ അല്ലെങ്കിൽ പെൺ
Date of joining	Date	ആവർത്തനം അനുവദിക്കുന്നു
Department	String	സയൻസ്, കോമേഴ്സ്, ഹ്യൂമാനിറ്റീസ് അല്ലെങ്കിൽ ഭാഷ
Basic pay	Numeric	അധ്യാപകരുടെ അടിസ്ഥാന ശമ്പളം

പട്ടിക നിർമ്മിക്കുന്നതിനും താഴെപ്പറയുന്ന ആവശ്യങ്ങൾക്കുമുള്ള SQL പ്രസ്താവനകൾ എഴുതുക.

a) ഡാറ്റ ചേർക്കുക (10 റെക്കോർഡുകൾ).

b) എല്ലാ അധ്യാപികമാരുടെയും വിശദാംശങ്ങൾ പട്ടികപ്പെടുത്തുക.

c) സയൻസ് ഡിപ്പാർട്ട്മെന്റിലെ പുരുഷ അധ്യാപകരുടെ വിശദാംശങ്ങൾ പട്ടികപ്പെടുത്തുക.

d) അടിസ്ഥാന ശമ്പളം Rs 50000 രൂപയോ അതിൽ കൂടുതലോ ആയിട്ടുള്ള ഭാഷാധ്യാപകരുടെ പേരും അടിസ്ഥാന ശമ്പളവും പ്രദർശിപ്പിക്കുക.

e) അധ്യാപകരുടെ പേരും അടിസ്ഥാന ശമ്പളത്തിന്റെ 71% വും പ്രദർശിപ്പിക്കുക.

f) ഓരോ വിഭാഗത്തിലെയും അധ്യാപകരുടെ എണ്ണം കണ്ടുപിടിക്കുക.

g) ഒരു സ്കൂളിലെ അധ്യാപകരുടെ ശരാശരി അടിസ്ഥാന ശമ്പളത്തെക്കാൾ കുറവ് ശമ്പളമുള്ള അധ്യാപകരുടെ വിശദാംശങ്ങൾ പ്രദർശിപ്പിക്കുക.

h) 1-1-2010 ന് മുൻപ് ചേർന്ന പുരുഷ അധ്യാപകരുടെ പട്ടിക തയാറാക്കുക.

i) എല്ലാ അധ്യാപകരുടേയും അടിസ്ഥാന ശമ്പളം 1000 രൂപ വർധിപ്പിക്കുക.

j) ഭാഷാ വിഭാഗത്തിലെ അധ്യാപകരുടെ വിശദാംശങ്ങൾ ഒഴിവാക്കുക.

4. ഒരു ബാങ്കിലെ ഉപഭോക്താക്കളുടെ വിശദാംശങ്ങൾ സംഭരിക്കുന്നതിനുള്ള പട്ടികയുടെ ഘടന കൊടുക്കുന്നു.

ഡാറ്റ	ഡാറ്റ ഇനം	വിവരണം
Account number	Numeric	തിരിച്ചറിയാൻ തനതായതും ഏകവുമായ ഡാറ്റ
Name	String	പരമാവധി 30 അക്ഷരങ്ങൾ
Gender	Character	ആൺ അല്ലെങ്കിൽ പെൺ
Date of joining	Date	ആവർത്തനം അനുവദിക്കുന്നു
Type of account	String	SB അല്ലെങ്കിൽ Current അക്കൗണ്ട്
Balance amount	Numeric	ദശാംശ സംഖ്യകൾ അനുവദിക്കുന്നു

പട്ടിക നിർമ്മിക്കുന്നതിനും താഴെ പറയുന്ന ആവശ്യങ്ങൾക്കുമുള്ള SQL പ്രസ്താവനകൾ എഴുതുക.

a. ഡാറ്റ ചേർക്കുക (10 റെക്കോർഡുകൾ).

b. SB അക്കൗണ്ടുള്ള ഉപഭോക്താവിന്റെ വിശദാംശങ്ങൾ പ്രദർശിപ്പിക്കുക.

c. ബാലൻസ് തുക 5000 രൂപയിൽ കൂടുതൽ ഉള്ള വനിത ഉപഭോക്താക്കളുടെ പേര് പ്രദർശിപ്പിക്കുക.

d. ബാലൻസ് തുക Rs 10000 രൂപയെക്കാൾ കൂടുതൽ ഉള്ള സ്ത്രീകളായ ഉപഭോക്താക്കളുടെ വിശദാംശങ്ങൾ നൽകുക.

e. സ്ത്രീ, പുരുഷ ഉപഭോക്താക്കളുടെ എണ്ണം കണ്ടുപിടിക്കുക.

f. ഏറ്റവും കൂടുതൽ ബാലൻസ് തുക ഉള്ള ഉപഭോക്താക്കളുടെ പേര് പ്രദർശിപ്പിക്കുക.

g. Kumar എന്ന് അവസാനിക്കുന്ന ഉപഭോക്താക്കളുടെ പേര് പ്രദർശിപ്പിക്കുക.

h. 2000 രൂപ നീക്കിയിരിപ്പുള്ള ഉപഭോക്താക്കളുടെ ബാലൻസ് തുക പുതുക്കുക.

i. 20000 രൂപയോ അതിലധികമോ നീക്കിയിരിപ്പുള്ള ഉപഭോക്താക്കളുടെ അക്കൗണ്ടിൽ നിന്നും 2% നികുതി കുറച്ചതിനു ശേഷമുള്ള വിശദാംശങ്ങൾ പ്രദർശിപ്പിക്കുക.

j. കറന്റ് അക്കൗണ്ട് ഉള്ള ഉപഭോക്താക്കളുടെ വിശദാംശങ്ങൾ ഒഴിവാക്കുക.

നമുക്ക് വിലയിരുത്താം

1. CUSTOMER എന്ന പട്ടികയിലെ വരികൾ ഒഴിവാക്കാൻ ഉപയോഗിക്കുന്ന കമാൻഡ് ഏത്?

   a. `REMOVE FROM CUSTOMER`

   b. `DROP TABLE CUSTOMER`

c. DELETE FROM CUSTOMER

d. UPDATE CUSTOMER

2. ഒരു നിരയിലെ ചില വിലകൾ അറിയില്ലെങ്കിൽ ഇതിലേക്ക് ഒരു വരി എങ്ങനെ ചേർക്കാം.
3. CHAR, VARCHAR എന്നീ SQL ഡാറ്റ ഇനങ്ങൾ തമ്മിലുള്ള വ്യത്യാസം കണ്ടെത്തുക.
4. PRIMARY KEY, UNIQUE എന്നീ കൺസ്ട്രെയിന്റുകൾ തമ്മിലുള്ള വ്യത്യാസം എന്ത്?
5. SQL ലെ NULL മൂല്യം എന്നത്കൊണ്ട് അർഥമാക്കുന്നത് എന്ത്?
6. SELECT പ്രസ്താവനയുടെ ഉപവാക്യങ്ങളുടെ കൃത്യമായ ക്രമീകരണം ഏത് ?

   a. SELECT, FROM, WHERE, ORDER BY

   d. SELECT, FROM, ORDER BY, WHERE

   c. SELECT, WHERE, FROM, ORDER BY

   d. SELECT, WHERE, ORDER BY, FROM
7. പാറ്റേൺ തുല്യതയ്ക്ക് ഉപയോഗിക്കുന്ന SQL ഓപ്പറേറ്ററാണ് __________.
8. താഴെ കൊടുത്തിരിക്കുന്ന സ്ട്രിംഗുകൾ വായിക്കുക.

   (i) 'Sree Kumar' (ii) 'Kumaran' (iii) 'Kumar Shanu' (iv) 'Sreekumar'

   SELECT പ്രസ്താവനയിൽ LIKE ഓപ്പറേറ്ററിന്റെ കൂടെ '%kumar' എന്ന പാറ്റേൺ കൃത്യമായി തുല്യത പ്രാപിക്കുന്നത് ഏതിനോടാണ് ?

   a. സ്ട്രിംഗ് (i) ഉം സ്ട്രിംഗ് (ii) ഉം മാത്രം

   b. സ്ട്രിംഗ് (i) ഉം സ്ട്രിംഗ് (iii) ഉം മാത്രം

   c. സ്ട്രിംഗ് (i), (iii), (iv) മാത്രം

   d. എല്ലാ സ്ട്രിംഗുകളും
9. SQL ലെ ഏതെങ്കിലും അഞ്ച് ബിൽട്ട് ഇൻ ഫങ്ഷനുകളും അവ ഓരോന്നും തരുന്ന മൂല്യങ്ങളും പട്ടികപ്പെടുത്തുക.
10. WHERE, HAVING തമ്മിലുള്ള ഏതെങ്കിലും നാല് വ്യത്യാസങ്ങൾ എഴുതുക.
11. SQL ലെ ഏതെങ്കിലും നാല് DML കമാൻഡുകൾ എഴുതുക
12. താഴെ പറയുന്ന ഓരോ SQL കമാൻഡിന്റെയും ഉപവാക്യങ്ങൾ എഴുതുക

    a. INSERT INTO   b. SELECT   c. UPDATE
13. Customer എന്ന പട്ടിക പരിഗണിച്ച് താഴെ കൊടുത്തിരിക്കുന്ന SQL ക്വറികളുടെ ഔട്ട്പുട്ട് എഴുതുക

AccNo	Name	Branch	Amount
1001	Kumar	Calicut	10000
1002	Salim	Trivandrum	20000
1003	Fida	Kottayam	18000
1004	John	Kannur	30000
1005	Raju	Thrissur	5000

a. `SELECT * FROM customer WHERE Amount>25000;`

b. `SELECT Name FROM customer WHERE Branch IN ('Calicut',' Kannur');`

c. `SELECT COUNT(*) FROM customer WHERE Amount< 20000;`

d. `SELECT Name FROM customer WHERE Name LIKE "%m%";`

c. `SELECT *FROM customer ORDER BY Amount DESC;`

14. `COUNT(*)` ഉം `COUNT(column_name)` ഉം തമ്മിലുള്ള വ്യത്യാസം എഴുതുക.

15. ITEM എന്ന പട്ടിക പരിഗണിക്കുക.

Item Code	Name	Category	Unit Price	Sales Price
0001	Pencil	Stationery	5.00	8.00
0002	Pen	Stationery	8.00	10.00
0003	NoteBook	Stationery	10.00	20.00
0004	Chappal	Footwear	50.00	70.00
0005	Apple	Fruits	60.00	90.00
0006	Orange	Fruits	40.00	60.00
0007	Pen	Stationery	10.00	12.00

a. ഈ പട്ടികയ്ക്ക് അനുയോജ്യമായ ഒരു പ്രാഥമിക കീ നിർദേശിക്കുക. ഉത്തരം സാധൂകരിക്കുക.

b. താഴെ പറയുന്ന ആവശ്യങ്ങൾക്കുള്ള SQL പ്രസ്താവനകൾ എഴുതുക.

i) എല്ലാ സ്റ്റേഷനറി ഇനങ്ങളും പട്ടികപ്പെടുത്തുക.

ii) itemcode, name, profit എന്നിവ പട്ടികപ്പെടുത്തുക.

iii) ഓരോ വിഭാഗത്തിലുമുള്ള ഇനങ്ങളുടെ എണ്ണം കണ്ടുപിടിക്കുക.

iv) യൂണിറ്റ് വിലയുടെ അടിസ്ഥാനത്തിൽ എല്ലാ സ്റ്റേഷനറി ഇനങ്ങളും അവരോഹണക്രമത്തിൽ പട്ടികപ്പെടുത്തുക.

v) വിൽപ്പനവില ഏറ്റവും കൂടുതൽ ഉള്ള ഇനങ്ങൾ കണ്ടുപിടിക്കുക.

vi) എല്ലാ സ്റ്റേഷനറി ഇനങ്ങളുടെയും വിശദാംശങ്ങൾ അടങ്ങിയ വ്യൂ ഉണ്ടാക്കുക.

16. ഒരു പട്ടികയുടെ ഘടനയിൽ നമുക്ക് ചെയ്യാൻ സാധിക്കുന്ന വിവിധ പ്രവർത്തനങ്ങൾ എന്തൊക്കെ? ഇതിന് ഏത് കമാൻഡ് ഉപയോഗിക്കണം? ഓരോന്നിന്റെയും മാറ്റത്തിന് ഉപയോഗിക്കുന്ന ഉപവാക്യങ്ങൾ എഴുതുക.

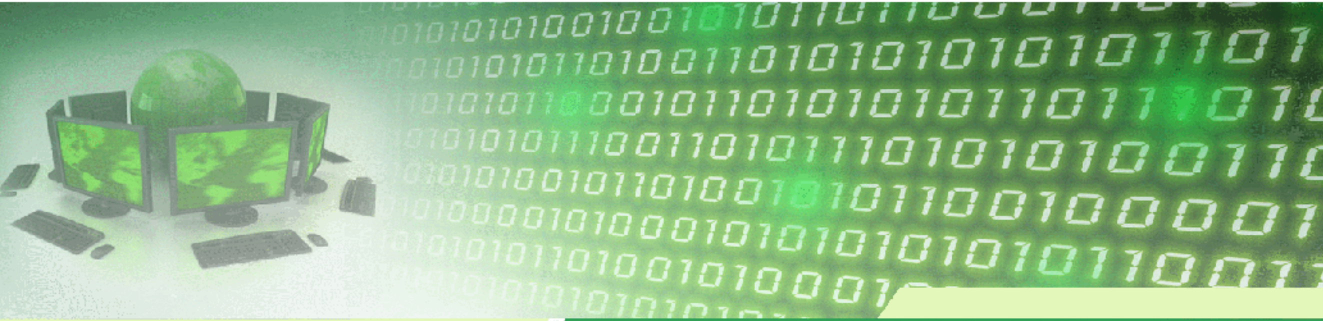
17. SQLൽ 10 റെക്കോർഡുകളോടുകൂടി ഒരു പട്ടിക നിർമ്മിച്ചിട്ടുണ്ടെന്ന് വിചാരിക്കുക. നിർദിഷ്ട വരിയിലെ നിരകളിലുള്ള മൂല്യങ്ങൾക്ക് മാറ്റം വരുത്താൻ ഏത് SQL കമാൻഡാണ് ഉപയോഗിക്കുന്നത്? അതിന്റെ ഘടന എഴുതുക.

18. ഒരു നിരയിൽ ആവർത്തനം ഒഴിവാക്കുന്നതിന് വേണ്ടി SELECT കമാൻഡിന്റെ കൂടെ ഏത് കീ വേഡാണ് ഉപയോഗിക്കുന്നത്? കമാൻഡിന്റെ പേരെഴുതുക.

19. SQL ൽ DISTINCT ഉം UNIQUE ഉം തമ്മിലുള്ള വ്യത്യാസം എഴുതുക.

20. കൂട്ടത്തിൽപ്പെടാത്തത് ഏത്? കാരണം എഴുതുക.

a. CREATE b. SELECT c. UPDATE d. INSERT



10 സംരംഭത്തിലെ വിഭവങ്ങളുടെ ആസൂത്രണം

പ്രധാന പഠനനേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പഠനം പൂർത്തിയാകുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ

- ഒരു സംരംഭത്തിലെ വിഭവങ്ങളുടെ ആസൂത്രണത്തിന്റെ ആവശ്യകത തിരിച്ചറിയുന്നു.
- ഒരു സംരംഭത്തിലെ വിഭവങ്ങളുടെ ആസൂത്രണ (ERP) സമ്പ്രദായത്തിന്റെ വിവിധ പ്രവർത്തന ഘട്ടങ്ങൾ പട്ടികപ്പെടുത്തുന്നു.
- ERP നടപ്പാക്കുന്നതിൽ വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങ്ങിനുള്ള (BPR-നുള്ള) പ്രാധാന്യം വിശദമാക്കുന്നു.
- ഒരു ERP സമ്പ്രദായം നടപ്പാക്കുന്നതിലെ വിവിധ ഘട്ടങ്ങൾ തിരിച്ചറിയുന്നു.
- വിപണിയിലുള്ള ചില പ്രധാനപ്പെട്ട ERP പാക്കേജുകളുടെ പട്ടിക തയ്യാറാക്കുന്നു.
- ഒരു സംരംഭത്തിൽ ERP നടപ്പാക്കുന്നത് കൊണ്ടുള്ള നേട്ടങ്ങളും കോട്ടങ്ങളും വിശദീകരിക്കുന്നു.
- ERP സമ്പ്രദായവുമായി ബന്ധപ്പെട്ടിട്ടുള്ള ചില സാങ്കേതിക വിദ്യകൾ പരിചയപ്പെടുന്നു.

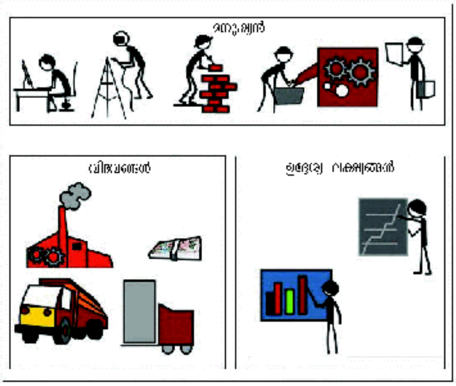
P6J7V7

കിടമൽസരം നിലനിൽക്കുന്ന ഇന്നത്തെ ലോകത്തിൽ, ഒരു സംരംഭത്തിന്റെ ഭാവി വളരെ ബുദ്ധിപരമായി ഒരാൾ കൈകാര്യം ചെയ്യേണ്ടതുണ്ട്. വിവര നിർവ്വഹണമാണ് സംരംഭത്തിന്റെ ഭാവി കൈകാര്യം ചെയ്യുന്നത് കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. ധനപരമായ ഡാറ്റ, ഉപഭോക്താവിനെക്കുറിച്ചുള്ള വിശദാംശങ്ങൾ, വാങ്ങുന്ന സാധനങ്ങളെകുറിച്ചുള്ള വിശദാംശങ്ങൾ, ജീവനക്കാരെ സംബന്ധിച്ച ഡാറ്റ, തുടങ്ങി ധാരാളം കാര്യങ്ങൾ ഒരു വലിയ സംരംഭം ഉൽപ്പാദിപ്പിക്കുന്നു. ഇത്തരം വിവരങ്ങൾ മികച്ച രീതിയിൽ ഉപയോഗപ്പെടുത്താൻ സാധിക്കുന്ന സ്ഥാപനകൾ മാത്രമേ വിജയിക്കുകയുള്ളൂ. വിവര വിസ്ഫോടനത്തിന്റെ ഈ കാലത്ത് വ്യക്തികൾ തന്നെ വലിയ തോതിൽ വിവര നിർവ്വഹണം നടത്തുക എന്നത് അസാധ്യമാണ്. ഒരു സംരംഭത്തിൽ വിഭവ-വിവര ആസൂത്രണം, സംഘാടനം എന്നിവയ്ക്കായി വിവര സാങ്കേതികവിദ്യയും അനുബന്ധ സാങ്കേതിക വിദ്യകളും പ്രയോജനപ്പെടുത്താവുന്നതാണ്. അതുകൊണ്ട് ഭൂരിഭാഗം സംഘടനകളും അവരുടെ വിവര നിയന്ത്രണ പ്രശ്നത്തിന് പരിഹാരമായി എന്റർപ്രൈസ് റിസോഴ്സ് പ്ലാനിങ് (ERP) പാക്കേജുകളിലേക്ക് നീങ്ങിയിട്ടുണ്ട്.

10.1 സംരംഭങ്ങളെക്കുറിച്ച് ഒരു അവലോകനം

ഒരു പൊതുലക്ഷ്യത്തോടെ പ്രവർത്തിക്കുന്ന വ്യക്തികളുടേയും വിഭവങ്ങളുടേയും കൂട്ടമാണ് ഒരു സംരംഭം. ഉല്പാദനം അഥവാ നിർമ്മാണം, ആസൂത്രണം, വിൽപ്പന, വാങ്ങൽ, ധനകാര്യം, വിതരണം തുടങ്ങിയ വിവിധ വകുപ്പുകൾ അല്ലെങ്കിൽ ഭാഗങ്ങൾ അടങ്ങിയതാണ് ഒരു സംരംഭം.

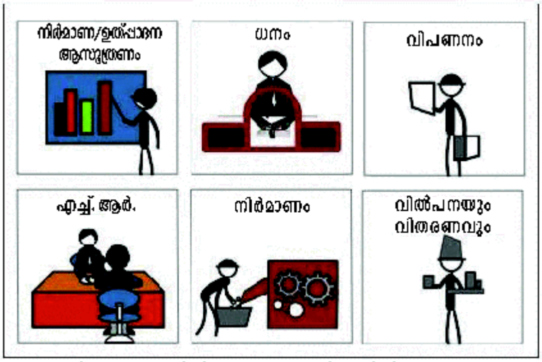
ഓരോ വകുപ്പിനും അവരുടേതായ ഉത്തരവാദിത്വങ്ങളും കർത്തവ്യങ്ങളും ഉണ്ടാവുകയും സംരംഭത്തിന്റെ ഉദ്ദേശപ്രാപ്തിക്കായി അവർ പ്രവർത്തിക്കുകയും ചെയ്യും. പണം, വ്യക്തികൾ, വസ്തുക്കൾ, പ്രവർത്തന രീതികൾ, യന്ത്രസാമഗ്രികൾ തുടങ്ങിയ വിവിധതരം വിഭവങ്ങൾ സംരംഭത്തിന് ആവശ്യമാണ്. വിവിധ വകുപ്പുകൾക്ക് വേണ്ടി വിവര വ്യവസറിതി രൂപപ്പെടുത്തിക്കൊണ്ട് കൃത്യവും സമയ ബന്ധിതവുമായ ഡാറ്റ എത്തേണ്ടവരിലേക്ക് എത്തിക്കുവാൻ ഒരു സംരംഭത്തിന് സാധിക്കും. സംരംഭത്തിലെ വിഭവങ്ങൾ ഒരു പൊതുവായ ലക്ഷ്യത്തോടെ അല്ലെങ്കിൽ ഉദ്ദേശത്തോടെ പ്രവർത്തിക്കുന്നത് ചിത്രം 10.1-ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 10.1: ഒരു സംരംഭത്തിലെ നിർമ്മാണ ഘടകങ്ങൾ

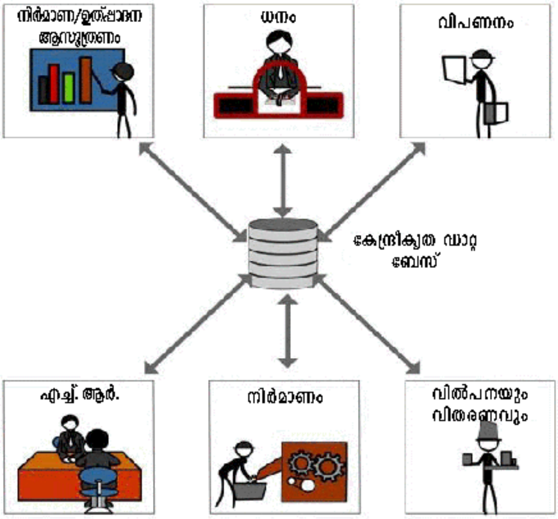
10.2 സംരംഭത്തിലെ വിഭവങ്ങളുടെ ആസൂത്രണം എന്ന ആശയം

ചില സംരംഭങ്ങളിൽ വിവിധ വകുപ്പുകൾ സ്വതന്ത്രമായാണ് പ്രവർത്തിക്കുന്നത്. അതുകൊണ്ട് ഓരോ വകുപ്പിലും ഉല്പാദിപ്പിക്കുന്ന വിവരങ്ങൾ അതേ വകുപ്പിലെ തന്നെ ഉന്നത സ്ഥാനങ്ങളിലുള്ളവർക്ക് മാത്രം ലഭ്യമാവുകയും മറ്റ് വകുപ്പുകൾക്ക് ലഭ്യമാകാതിരിക്കുകയും ചെയ്യാം. ഇവിടെ ഒരു സംരംഭത്തിലെ വിവിധ വകുപ്പുകൾ തമ്മിൽ യാതൊരു തരത്തിലുമുള്ള ആശയവിനിമയവും നടക്കുന്നില്ല. അത്തരം ഒരു സംരംഭത്തിന്റെ ചിത്രം 10.2-ൽ വർണ്ണിച്ചിരിക്കുന്നു.



ചിത്രം.10.2: വിവിധ വകുപ്പുകൾ തമ്മിൽ ഒട്ടുംതന്നെ ആശയവിനിമയം നടക്കാത്ത അല്ലെങ്കിൽ വളരെകുറച്ച് മാത്രം ആശയവിനിമയമുള്ള ഒരു സംരംഭം

ഉയർന്ന കാര്യക്ഷമതയും, നേട്ടവും കൈവരിക്കുന്നതിന് ഓരോ വകുപ്പും മറ്റ് വകുപ്പുകൾ ചെയ്യുന്നത് എന്താണ് എന്ന് അറിഞ്ഞിരിക്കണം. സംരംഭത്തെ ഒരു വ്യവസറയായും അതിലെ എല്ലാ വകുപ്പുകളേയും ഉപവ്യവസ്ഥകളായും കണക്കാക്കാം. സംരംഭത്തിലെ വിവരങ്ങൾ മുഴുവൻ ഒരു ഏകീകൃത ഡാറ്റാബേസിൽ സംഭരിക്കുവാനും അത് എല്ലാ വകുപ്പുകൾക്കും ലഭ്യമാക്കുവാനും സാധിക്കും. അത്തരം ഒരു സംരംഭത്തെക്കുറിച്ച് വർണ്ണന ചിത്രം 10.3-ൽ വർണിച്ചിരിക്കുന്നു.



ചിത്രം 10.3: കേന്ദ്രീകൃത ഡാറ്റബേസോട് കൂടിയ ഒരു സംരംഭം.

ഒരു സംരംഭത്തിന്റെയോ കമ്പനിയുടെയോ എല്ലാ ആവശ്യങ്ങളും ഒറ്റ ഡാറ്റാബേസിൽ പ്രവർത്തിക്കുന്ന ഏകീകൃത സോഫ്റ്റ്‌വെയർ സംവിധാനത്തിലേക്ക് ERP ഉപയോഗിച്ച് ഒരുമിപ്പിക്കാനാകും. അതിനാൽ അവിടെയുള്ള വിവിധ വകുപ്പുകൾക്ക് വിവരം പങ്കുവയ്ക്കുവാനും പരസ്പരം ആശയവിനിമയം നടത്തുവാനും എളുപ്പത്തിൽ സാധിക്കും. ആശയപരമായി, ഒരു സംരംഭത്തിലെ ധനകാര്യം, മാനവശേഷി, ഉൽപ്പാദനം, വിൽപ്പന തുടങ്ങിയ വിവിധ മേഖലകളിൽ സ്വയം പ്രവർത്തിക്കുന്ന കമ്പ്യൂട്ടർ സമ്പ്രദായങ്ങൾക്ക് പകരം ഇത്തരം പ്രവർത്തനങ്ങൾ നടത്തുവാൻ പ്രാപ്തമായ ഒരു സോഫ്റ്റ്‌വെയർ പ്രോഗ്രാം ERP സമ്പ്രദായം ഉപയോഗിക്കുന്നു. അതുകൊണ്ട് ഏത് വകുപ്പിലെ ജീവനക്കാരനും ആ വകുപ്പിന്റെ പ്രവർത്തനങ്ങളുമായി ബന്ധപ്പെട്ട വിവരം ലഭ്യമാകും. അത് കൂടാതെ വകുപ്പുകൾ തമ്മിലും വിവരവിനിമയം നടത്താനാകും. ഉദാഹരണമായി ധനകാര്യ വകുപ്പിലെ ചിലർക്ക് ഏതെങ്കിലും വിൽപ്പന നടപടി (സെയിൽ ഓർഡർ) വിതരണ ശാലയിൽ നിന്നും കയറ്റി വിട്ടിട്ടുണ്ടോ എന്ന് പരിശോധിച്ച് വരുംകാല പ്രവർത്തനത്തിന് ആവശ്യമായ മൂലധനത്തിന്റെ ആസൂത്രണം നിർവ്വഹിക്കാനാകും. പ്രവർത്തനപ്രക്രിയകൾ സംഘടിപ്പിക്കുവാനും ഏകീകരിക്കുവാനും അത് സഹായിക്കുകയും അതുവഴി വിവര വിനിമയത്തിലൂടെ മാനവിക വിഭവങ്ങൾ, അസംസ്കൃത വസ്തുക്കൾ, പണം, യന്ത്രസാമഗ്രികൾ, തുടങ്ങിയവ ഫലപ്രദമായി ഉപയോഗിക്കാൻ സാധിക്കുകയും ചെയ്യും.

പൂർണ്ണമായും ഏകീകരിച്ച വാണിജ്യ നിർവഹണ സമ്പ്രദായമായ ERP ഒരു സംരംഭത്തിലെ ധനകാര്യം, മാനവിക വിഭവം, നിർമാണം, വിൽപ്പന, ചരക്ക് നീക്കം തുടങ്ങിയ പ്രവർത്തന മേഖലകൾക്ക് പര്യാപ്തമാണ്. വിവിധ പദ്ധതി ഘടകങ്ങൾക്ക് വേണ്ട ഡാറ്റ സംഭരിക്കുവാൻ ഒരു ഡാറ്റാബേസ് ഉപയോഗിക്കാനാകും എന്നതാണ് ERP- യ്ക്ക് മൂലാധാരമായിട്ടുള്ളത്. കമ്പ്യൂട്ടറിലെ സോഫ്റ്റ്‌വെയർ ഭാഗങ്ങളും ഹാർഡ്‌വെയർ ഭാഗങ്ങളും ERP സമ്പ്രദായം ഉപയോഗിക്കുന്നു.

10.3 ERP യുടെ പ്രവർത്തന ഘടകങ്ങൾ

ഒരു സംരംഭത്തിലെ വിഭവങ്ങൾ ഫലപ്രദമായി ഉപയോഗിക്കേണ്ടതാവശ്യമാണ്. അതുകൊണ്ട് വിഭവ ആസൂത്രണം നേതൃത്വത്തിന്റെ ഉത്തരവാദിത്വമാണ്. ഉൽപ്പാദനക്ഷമമവും കാര്യക്ഷമവുമായ വിഭവ ആസൂത്രണ പ്രക്രിയയ്ക്ക് ERP സമ്പ്രദായം സംരംഭത്തെ സഹായിക്കുന്നു. ERP പാക്കേജിൽ ധാരാളം ഭാഗങ്ങൾ അഥവാ ഉപഘടകങ്ങൾ അടങ്ങിയിരിക്കുന്നു. സംരംഭത്തിന്റെ സ്വഭാവത്തിനും ERP പാക്കേജിന്റെ തരത്തിനും അനുസരിച്ച് ഘടകങ്ങളുടെ എണ്ണവും പ്രവർത്തനവും വ്യത്യാസപ്പെട്ടിരിക്കും. മിക്കവാറും എല്ലാ പാക്കേജുകളിലും ലഭ്യമാകുന്ന പൊതുവായ ഘടകങ്ങളെക്കുറിച്ച് ചുരുക്കത്തിൽ താഴെ വിവരിച്ചിരിക്കുന്നു.

ധനകാര്യ ഘടകം (ഫിനാൻഷ്യൽ മോഡ്യൂൾ)

ഭൂരിഭാഗം ERP പാക്കേജുകളുടേയും കാതലായ ഭാഗമാണ് ഇത്. വിവിധ വകുപ്പുകളിൽ നിന്നുള്ള ധനകാര്യ ഡാറ്റ സ്വീകരിക്കുവാനും വിലപ്പെട്ട ധനകാര്യ അവലോകനം സൃഷ്ടിക്കുവാനും ഇതിന് സാധിക്കുന്നു. ബാലൻസ് ഷീറ്റ്, പൊതുലഡ്ജർ, ട്രയൽ ബാലൻസ്, ധനകാര്യ വിവരണങ്ങൾ തുടങ്ങിയ ധനപരമായ അവലോകനങ്ങൾ ഇതിൽ ഉണ്ടാകും. ധനകാര്യ അക്കൗണ്ടുകൾ, നിക്ഷേപ നിർവ്വഹണം, സംരംഭത്തിന്റെ നടത്തിപ്പ്, ട്രഷറി തുടങ്ങിയ ഘടകങ്ങളും ഇതിൽ ഉണ്ടാകും.

ഉൽപ്പാദന ഘടകം (മാനുഫാക്ച്വറിങ് മോഡ്യൂൾ)

നിർമാണ പ്രക്രിയ മൊത്തമായി കൈകാര്യം ചെയ്യുന്നതിനുള്ള വാണിജ്യ നിയമങ്ങൾ ഉൽപ്പാദന ഘടകത്തിൽ ഉണ്ടാകും. സാങ്കേതികവിദ്യയും വാണിജ്യ പ്രവർത്തനങ്ങളും ഒരുമിപ്പിച്ച് കൊണ്ട് ഒരു ഏകീകൃത പരിഹാരം സാധ്യമാക്കുന്നതിന് ഈ ERP ഘടകം സംരംഭത്തെ സഹായിക്കുന്നു. നിർമ്മാണത്തിനാവശ്യമായ എല്ലാപ്രവർത്തനങ്ങളും നടപ്പിലാക്കാൻ വേണ്ട വിവരം അത് നൽകുന്നു. നിർമാണ രീതിയും ആസൂത്രണ പദ്ധതിയും ആവശ്യാനുസരണം വ്യത്യാസപ്പെടുത്തുവാനുള്ള സ്വാതന്ത്ര്യം അത് പ്രധാനം ചെയ്യുന്നു.

നിർമ്മാണ ആസൂത്രണ ഘടകം (പ്രൊഡക്ഷൻ പ്ലാനിങ് മോഡ്യൂൾ)

ലഭ്യമായ വിഭവങ്ങൾ പരമാവധി പ്രയോജനപ്പെടുത്തുന്നതിനും നിർമാണ പ്രക്രിയ ആസൂത്രണം ചെയ്യുന്നതിനും ഈ ഘടകം ഉപയോഗിക്കുന്നു. വിഭവങ്ങളുടെ ഫലപ്രദമായ ഉപയോഗമാണ് പരമാവധി പ്രയോജനപ്പെടുത്തുക എന്നത് കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. ഈ ഘടകം ആവശ്യ വസ്തുക്കൾ കണ്ടെത്തുകയും പരമ്പരാഗതമായ നിർമ്മാണ പ്രവർത്തനത്തിന് വേണ്ട വിഭവങ്ങൾ ഫലപ്രദമായി ഉപയോഗിക്കുകയും ചെയ്യുന്നു. വിപണന വിവരങ്ങൾ ഉപയോഗപ്പെടുത്തി ഭാവി വിൽപ്പനയെക്കുറിച്ച് പ്രവചിക്കുന്നു.

HR ഘടകം (ഹ്യൂമൻ റിസോഴ്സ് മോഡ്യൂൾ)

HR ഘടകം മാനവിക വിഭവത്തെ സൂചിപ്പിക്കുന്നു. മാനവിക വിഭവങ്ങളുടേയും മനുഷ്യ ധനത്തിന്റേയും പരിപാലനമാണ് ERP-യുടെ HR ഘടകം ശ്രദ്ധിക്കുന്നത്. ഒരു സംരംഭത്തിലെ എല്ലാ ജീവനക്കാരുടേയും സ്വകാര്യ വിവരങ്ങൾ, ശമ്പള വിവരങ്ങൾ, ഹാജർ, പ്രകടനം, സ്ഥാനക്കയറ്റം തുടങ്ങി പൂർണ്ണവും നവീകരിക്കപ്പെട്ടതുമായ ഡാറ്റാബേസ് HR ഘടകം പരിപാലിക്കുന്നു.

ചരക്ക് പട്ടിക നിയന്ത്രണ ഘടകം (ഇൻവെന്ററി കൺട്രോൾ മോഡ്യൂൾ)

ആവശ്യമായ അളവിൽ സാധനങ്ങൾ സംഭരണ ശാലയിൽ നിലനിർത്തുന്നതിന് വേണ്ട പദ്ധതികൾ ഈ ഘടകത്തിൽ ഉണ്ട്. ചരക്ക് സാധനങ്ങളുടെ ആവശ്യകത, ലക്ഷ്യം നിർണയിക്കൽ, ഇനങ്ങളുടെ ഉപയോഗം നിരീക്ഷിക്കൽ, ചരക്ക് നില അവലോകനം ചെയ്യൽ തുടങ്ങിയ കാര്യങ്ങൾ വ്യക്തമാക്കുകയാണ് ഇതിന്റെ ഉത്തരവാദിത്വം. ചരക്ക് നിയന്ത്രണം, വിൽപ്പന, വാങ്ങൽ, ധനകാര്യം എന്നിവ ഏകീകരിപ്പിക്കുക വഴി ഒരു സംരംഭത്തിന് ബുദ്ധിപരമായ അവലോകനങ്ങൾ നടത്തുവാൻ സാധിക്കുന്നു.

വാങ്ങൽ ഘടകം (പർച്ചേസിങ് മോഡ്യൂൾ)

വാങ്ങൽ ഘടകം അസംസ്കൃത വസ്തുക്കൾ യഥാസമയം കൃത്യമായ വിലയ്ക്ക് വാങ്ങൽ ഘടകം ലഭ്യമാക്കുന്നു. വിതരണക്കാർക്ക് വേണ്ട വാങ്ങൽ ഉടമ്പടി തയ്യാറാക്കുക, അവരെ വിലയിരുത്തുക, വിൽപ്പനച്ചീട്ട് തയ്യാറാക്കുക, തുടങ്ങിയവയ്ക്ക് ആവശ്യമായ ഭാഗങ്ങൾ ഇതിൽ ലഭ്യമാണ്. ചരക്ക് വിവരം, ധനവിവരം, നിർമാണ ആസൂത്രണം തുടങ്ങിയ ഘടകങ്ങളുമായി വാങ്ങൽ ഘടകത്തെ വളരെയധികം ബന്ധപ്പെടുത്തിയിരിക്കുന്നു.

വിപണന ഘടകം (മാർക്കറ്റിങ് മോഡ്യൂൾ)

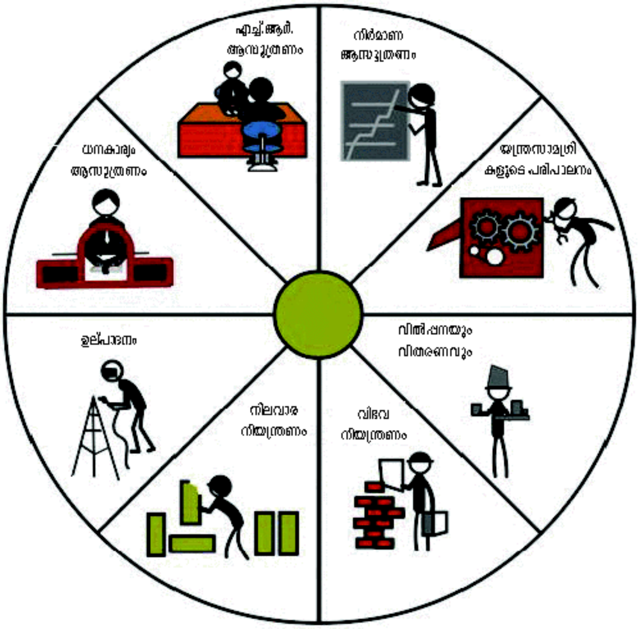
ഉപഭോക്താവിന്റെ ഉടമ്പടി (ഓർഡർ) നിരീക്ഷിക്കുകയും പിൻതുടരുകയും ചെയ്യുക, അവരുടെ സംതൃപ്തി വർദ്ധിപ്പിക്കുക, കടം കൊടുക്കുന്നത് കൊണ്ടുള്ള നഷ്ട സാധ്യത ഒഴിവാക്കുക എന്നീ കാര്യങ്ങൾക്കാണ് വിപണന ഘടകം ഉപയോഗിക്കുന്നത്. കച്ചവടക്കാർക്ക് പദ്ധതി വിശകലനം ചെയ്യുവാനും നടപ്പിലാക്കുവാനും എല്ലാ വിപണന പ്രവർത്തികളും അളക്കുവാനും ഇത് സഹായിക്കുന്നു.

വിൽപനയ്ക്കും വിതരണത്തിനുമുള്ള ഘടകം (സെയിൽസ് ആന്റ് ഡിസ്ട്രിബ്യൂഷൻ മോഡ്യൂൾ)

ഒരു സംരംഭത്തിന്റെ ജീവരക്തമാണ് വിൽപ്പനയിൽ നിന്നുള്ള വരുമാനം. വിപണന ചക്രത്തിന്റെ പ്രധാന ഭാഗങ്ങൾ കൈകാര്യം ചെയ്യുന്നത് വിപണന ഘടകമാണ്. അന്വേഷണങ്ങൾ, ഓർഡർ നൽകൽ, ഓർഡറിനനുസരിച്ചുള്ള പദ്ധതി തയാറാക്കൽ എന്നിവ വിപണന ചക്രത്തിൽ അടങ്ങിയിരിക്കുന്നു. സ്ഥാപനത്തിന്റെ e-കൊമേഴ്സ് വെബ്സൈറ്റുമായി ഈ ഘടകം വളരെ അധികം സമന്വയിപ്പിച്ചിരിക്കുന്നു.

നിലവാരം നിയന്ത്രണ ഘടകം (ക്വാളിറ്റി മാനേജ്മെന്റ് മോഡ്യൂൾ)

ഉൽപ്പന്നത്തിന്റെ നിലവാരം നിയന്ത്രിക്കുവാനാണ് ഈ ഘടകം ഉപയോഗിക്കുന്നത്. നിലവാര നിയന്ത്രണഘടകം താഴെ തന്നിരിക്കുന്ന കർമ്മങ്ങൾ നിർവഹിക്കുന്നു. നിലവാര ആസൂത്രണം, നിലവാര പരിശോധന, നിലവാര നിയന്ത്രണം. സംരംഭത്തിന്റെ വലുപ്പത്തിനും സ്വഭാവത്തിനും അനുസരിച്ച് മുകളിൽ തന്നിട്ടുള്ള ഘടകങ്ങളെ കൂടാതെ മറ്റ് വിവിധ ഘടകങ്ങളും അതിനുണ്ടാകും. ഒരു ERP സമ്പ്രദായം മുകളിൽ പറഞ്ഞിട്ടുള്ള എല്ലാ ഭാഗങ്ങളേയും സമന്വയിപ്പിച്ച് ഒരു സോഫ്റ്റ്വെയർ ആക്കുന്നു. ERP യുടെ വിവിധ പ്രവർത്തന ഘടകങ്ങൾ ചിത്രം 10.4-ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 10.4: ERP യുടെ പ്രവർത്തന ഘടകങ്ങൾ

10.4 വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീ എൻജിനീയറിങ്

ഒരു സംരംഭത്തിൽ ഇടതടവില്ലാതെ നടക്കുന്ന പ്രവർത്തനങ്ങളുടെ വിശകലനം, മാതൃക പുനർനിർമാണം എന്നിവയാണ് വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീ എൻജിനീയറിങ് കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. വാണിജ്യ സാഹചര്യങ്ങളുടെ ഘടനയിലും പ്രവർത്തനത്തിലുമുള്ള മാറ്റം അതിൽ അടങ്ങിയിരിക്കുന്നു. വാണിജ്യ പശ്ചാത്തലങ്ങളിൽ, പ്രത്യേകിച്ച് വിവരസാങ്കേതിക വിദ്യപോലുള്ള മേഖലകളിൽ ധ്രുതഗതിയിൽ ഉണ്ടാകുന്ന മാറ്റം ഉൾക്കൊണ്ട് മെച്ചപ്പെട്ട സേവനവും നേട്ടവും പ്രദാനം ചെയ്യാൻ വാണിജ്യ പ്രവർത്തനങ്ങളുടെ നവമാതൃക സൃഷ്ടിക്കൽ ആവശ്യമായി വരുന്നു. കാര്യാലയങ്ങളിലെ യന്ത്രവൽക്കരണം, ചിലവ് കുറ

യ്ക്കൽ, പ്രയോജനകരമായ വിഭവ ഉപഭോഗം എന്നിവ വാണിജ്യത്തിൽ റീഎൻജിനീയറിങ്ങിന്റെ അനന്തരഫലങ്ങളാണ്. വാണിജ്യ ലോകത്ത് വില, നിലവാരം, തിരഞ്ഞെടുക്കൽ, സേവനം എന്നിവയുടെ അടിസ്ഥാനത്തിലാണ് മത്സരം നടക്കുന്നത്. ഇൻർനെറ്റിന്റേയും, e-കൊമേഴ്സിന്റേയും ആവിർഭാവത്തോടെ ഇന്ന് ദൂരവും സമയവും ഒരു വാണിജ്യത്തിന് തടസ്സമല്ലാതായിത്തീർന്നിരിക്കുന്നു. വ്യവസറിതിയുടെ ഘടനയിലും ആകെ പ്രവർത്തനങ്ങളിലും വ്യത്യാസം വരുത്തിക്കൊണ്ടുള്ള സറാപനത്തിന്റെ പരിവർത്തനം ഇത്തരം മാറ്റങ്ങൾ അനിവാര്യമാക്കുന്നു.

വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങിൽ, റീഎൻജിനീയറിങ്ങും, പ്രവർത്തനങ്ങളും പ്രത്യേകം നിർവചിക്കാവുന്നതാണ്. ഒരു സംരംഭത്തിന് പ്രവർത്തന മികവ് ലഭിക്കാനായി അവിടുത്തെ വാണിജ്യ പ്രവൃത്തികൾ പുനർ വിചിന്തനം നടത്തി നവമാതൃക രൂപപ്പെടുത്തുന്നതാണ് റീഎൻജിനീയറിങ്. സംരംഭത്തിൽ നിർദ്ദിഷ്ഠ ഫലപ്രാപ്തിക്കു വേണ്ടി രൂപപ്പെടുത്തിയിട്ടുള്ള പ്രവൃത്തികളാണ് വാണിജ്യ പ്രവർത്തനങ്ങൾ. മനുഷ്യൻ, വിവരം, മറ്റ് വിഭവങ്ങൾ, എന്നിവ പ്രയോജനപ്പെടുത്തുന്ന അനുബന്ധ നടപടികളാണ് ഓരോ വാണിജ്യ പ്രവൃത്തിയിലും ഉള്ളത്.

ഒരു വാണിജ്യ പ്രവർത്തനത്തിന് മൂന്ന് ഘടകങ്ങൾ ഉണ്ടാകും.

- ഇൻപുട്ട് : അപേക്ഷാപത്രങ്ങൾ, ഉപഭോക്താവിന്റെ അന്വേഷണങ്ങൾ, പ്രവർത്തനത്തിന് ആവശ്യമായ വസ്തുക്കൾ, തുടങ്ങിയ ഡാറ്റ.
- പ്രൊസസ്സിങ് : ഫലപ്രാപ്തിക്ക് വേണ്ടിയുള്ള ഒരു കൂട്ടം പ്രവൃത്തികൾ അല്ലെങ്കിൽ ഘട്ടങ്ങളായുള്ള പ്രവർത്തനങ്ങൾ.
- പരിണതഫലം : പ്രവർത്തന ഘട്ടങ്ങളുടെ ഫലമാണ് ഇത്.

ഇവിടെ, വാണിജ്യ പ്രവർത്തനങ്ങളിൽ പരിഹാരം കാണാൻ ഏറ്റവും പ്രയാസമുള്ളതും സമയം എടുക്കുന്നതുമായ ഘടകമാണ് പ്രൊസസ്സിങ്. അടിസറാനപരമായി ഒരു വാണിജ്യ പ്രവർത്തനം എന്നത് പ്രവൃത്തി ചെയ്യുന്നതിനുള്ള മാർഗ്ഗവും ആ മാർഗ്ഗം മെച്ചപ്പെടുത്തി ലക്ഷ്യം കൈവരിക്കുന്നത് റീഎൻജിനീയറിംഗുമാണ്. അതുകൊണ്ട്, പ്രൊസസ്സിങ് ഭാഗത്തെ പരിമിതികൾ ഒഴിവാക്കി റീ എൻജിനീയറിങ്ങിലൂടെ പ്രവർത്തന മികവ് നേടാൻ BPR ശ്രദ്ധിക്കുന്നു.

ചെലവ്, നിലവാരം, സേവനം, വേഗത എന്നിവയിൽ പ്രവർത്തന മികവ് ലഭിക്കാനായി ഒരു സംരംഭത്തിലെ വാണിജ്യ പ്രവർത്തനങ്ങളെ മാതൃകാപരമായി പുനർവിചിന്തനം നടത്തി നവീകരിക്കുന്നതാണ് വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങ്. BPR-ന്റെ ജീവിത ചക്രത്തിലെ വിവിധ ഘട്ടങ്ങൾ ചിത്രം 10.5-ൽ വിവരിച്ചിരിക്കുന്നു. വാണിജ്യ പ്രവർത്തനങ്ങൾ തിരിച്ചറിയുക, ഇപ്പോഴുള്ളവ വിശകലനം ചെയ്യുക, വാണിജ്യ പ്രവർത്തനങ്ങളുടെ നവമാതൃക സൃഷ്ടിക്കുക, ആ മാതൃക നടപ്പാക്കുക എന്നീ ഘട്ടങ്ങൾ അതിൽ അടങ്ങിയിരിക്കുന്നു.

ചിത്രം 10.5: വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങ് (BPR)

ERP–യും BPR – ഉം തമ്മിലുള്ള ബന്ധം

ഒരു വാണിജ്യത്തിൽ ERP നടപ്പാക്കുന്നതിന് മുമ്പ് അത്തരം ഒരു പുതിയ വ്യവസറിതി ആവശ്യമാണ് എന്ന് ഉറപ്പിക്കേണ്ടതുണ്ട്. ഘടനയിലോ, പ്രവർത്തനത്തിലോ വാണിജ്യപരമായി മെച്ചപ്പെട്ട തരത്തിലുള്ള മാറ്റം കണ്ടെത്താൻ ഒരു സംരംഭത്തെ സഹായിക്കുന്നത് വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങ് ആണ്. അതിനാൽ ERP–ഉം BPR–ഉം ബന്ധപ്പെട്ടിരിക്കുകയും കൈകോർത്ത് പോവുകയും ചെയ്യുന്നു. സംരംഭത്തിലെ വിഭവങ്ങൽ ആസൂത്രണം ചെയ്യുന്നതിന് മുമ്പ് വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങ് നടത്തുന്നു. അത് സോഫ്റ്റ്‌വെയറിലുള്ള അനാവശ്യ ഘടകങ്ങൾ ഒഴിവാക്കുന്നതിന് സംരംഭത്തെ സഹായിക്കും. സോഫ്റ്റ്‌വെയർ പ്രവർത്തന യോഗ്യമാക്കുന്നതിന് മുമ്പ് വാണിജ്യ പ്രവർത്തനങ്ങളുടെ നല്ല വശങ്ങൾ പ്രായോഗികമാക്കുന്നു എന്ന് ആദ്യം തന്നെ ഉറപ്പിക്കാൻ സഹായിക്കുന്നത് BPR ആണ്. കൂടാതെ സോഫ്റ്റ്‌വെയറിന്റെ പ്രവർത്തനം ഇപ്പോഴുള്ള വാണിജ്യ പ്രവൃത്തികളുമായി പൊരുത്തപ്പെടുന്നുവെന്ന് അത് ഉറപ്പാക്കുകയും ചെയ്യുന്നു.

എല്ലാ BPR–ഉം ഒരു പുതിയ ERP സമ്പ്രദായം നടപ്പാക്കുന്നതിലേക്ക് നയിക്കണമെന്നില്ല. ചെലവ്, പ്രയോജനം, മറ്റ് വെല്ലുവിളികൾ എന്നിവ കാരണം ഒരു സംരംഭത്തിന് BPR ആവശ്യമില്ല എന്നും കണ്ടെത്തിയേക്കാം.

മറ്റ് ചില സാഹചര്യങ്ങളിൽ BPR–ഉം ERP- യും ഒരുമിച്ച് ഉപയോഗിച്ച് കൊണ്ട് സംരംഭത്തിന് മെച്ചപ്പെട്ട ഫലപ്രാപ്തിയും നിലവിലുള്ള ERP–യ്ക്ക് പുരോഗതിയും കൈവരിക്കാനാകും.

നിങ്ങളുടെ പുരോഗതി അറിയുക

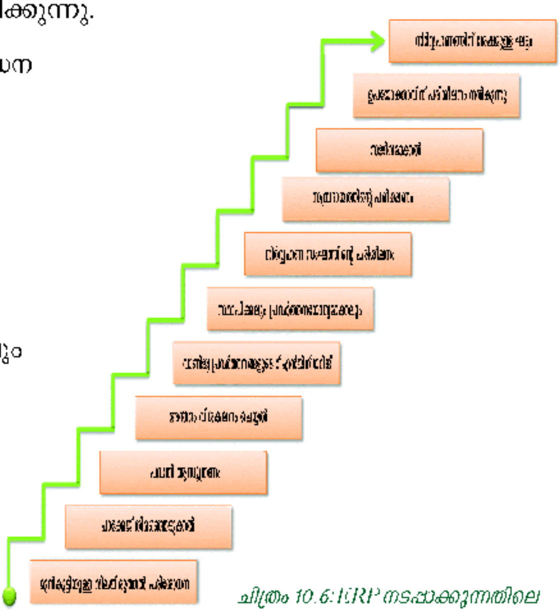
1. സംരംഭം എന്ന പദം നിർവചിക്കുക.
2. .... ഒരു പൂർണ്ണ സംയോജിത വാണിജ്യ നിയന്ത്രണ സംവിധാനമാണ്.
3. ഒരു സംരംഭത്തിലെ ചിട്ടയായ പ്രവർത്തനങ്ങളുടെ വിശകലനവും നവമാതൃക സൃഷ്ടിക്കലുമാണ്...................................
4. ERP - യിൽ പൊതുവായി ഉപയോഗിക്കുന്ന ഏതെങ്കിലും നാല് ഘടകങ്ങളുടെ പട്ടിക തയാറാക്കുക.
5. ഒരു സംരംഭത്തിലെ എല്ലാ വകുപ്പുകളെകുറിച്ചുമുള്ള വിവരങ്ങൾ ERP - യിൽ സംഭരിക്കുന്നത് ....ൽ ആണ്.

10.5. ERP- യുടെ നിർവഹണം (നടപ്പാക്കൽ)

ഇക്കാലത്ത് മിക്കവാറും എല്ലാതരത്തിലുള്ള സ്ഥാപനങ്ങളും അവരുടെ വിവര നിർവ്വഹണ പ്രശ്നത്തിന് പരിഹാരമായി ERP (സംരംഭ പ്രവർത്തനങ്ങളുടെ ആസൂത്രണം) പാക്കേജുകളിലേക്ക് നീങ്ങിക്കൊണ്ടിരിക്കുന്നു. ഒരു ERP പാക്കേജ് ശരിയായ രീതിയിൽ തിരഞ്ഞെടുത്ത് സംരംഭത്തിൽ നടപ്പാക്കുകയാണെങ്കിൽ സ്ഥാപനത്തിന്റെ ഉൽപ്പാദനക്ഷമതയും ലാഭവും നാടകീയമായി വർദ്ധിക്കും. ERP നടപ്പാക്കേണ്ടത് നല്ല രീതിയിലുള്ള ആസൂത്രണത്തോടും ഉൽകൃഷ്ടമായ നിർവഹണത്തോടും കൂടിയാകണം. അതുകൊണ്ട് ERP നടപ്പാക്കുന്ന വിവിധ ഘട്ടങ്ങളെ വേർതിരിക്കുന്നതിന് വ്യക്തമായ അതിരുകൾ ഇല്ല എന്ന് മാത്രമല്ല, മിക്ക സന്ദർഭങ്ങളിലും ഒരു ഘട്ടം ആരംഭിക്കുന്നത് അതിന് തൊട്ട് മുന്നി

ലുള്ള ഘട്ടം അവസാനിക്കുന്നതിന് മുമ്പ് ആയിരിക്കുകയും ചെയ്യും. ERP നടപ്പാക്കുന്നതിനുള്ള വിവിധ ഘട്ടങ്ങൾ താഴെതന്നിരിക്കുന്നു.

- മുൻകൂട്ടിയുള്ള വിലയിരുത്തൽ പരിശോധന
- പാക്കേജ് തിരഞ്ഞെടുക്കൽ
- പദ്ധതി ആസൂത്രണം
- അന്തരം വിശകലനം ചെയ്യൽ
- വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങ്
- സ്ഥാപിക്കലും പ്രവർത്തനയോഗ്യമാക്കലും
- നിർവ്വഹണ സംഘത്തിന്റെ പരിശീലനം
- സമ്പ്രദായത്തിന്റെ പരീക്ഷണം
- സജീവമാക്കൽ
- ഉപയോക്താവിന് പരിശീലനം നൽകുന്നു.
- നിർവ്വഹണത്തിന് ശേഷമുള്ള ഘട്ടം



ചിത്രം 10.6: ERP നടപ്പാക്കുന്നതിലെ വിവിധ ഘട്ടങ്ങൾ

മുൻകൂട്ടിയുള്ള വിലയിരുത്തൽ പരിശോധന

ഒരു സംരംഭത്തിൽ ERP നടപ്പാക്കുന്നതിന് ഉപയോഗിക്കാവുന്ന ധാരാളം പാക്കേജുകൾ വിപണിയിൽ ലഭ്യമാണ്. അനുയോജ്യമായ ഒരു പാക്കേജ് തിരഞ്ഞെടുക്കാൻ എല്ലാ പാക്കേജുകളും വിലയിരുത്തുന്നതിന് പകരം അവയുടെ എണ്ണം നാം പരിമിതപ്പെടുത്തണം. അതുകൊണ്ട് മുൻകൂട്ടിയുള്ള വിലയിരുത്തൽ പരിശോധന ERP പാക്കേജ് നടപ്പാക്കുന്നതിലെ ആദ്യ ഘട്ടമാണ്.

പാക്കേജ് തിരഞ്ഞെടുക്കൽ

പ്രവർത്തനവും കാര്യക്ഷമതയും വിലയിരുത്താനായി ആദ്യഘട്ടത്തിൽ നാം കുറച്ച് പാക്കേജുകൾ തിരഞ്ഞെടുത്തിരുന്നു. പദ്ധതിയുടെ ജയപരാജയങ്ങൾ നിർണ്ണയിക്കുന്നത് നാം തിരഞ്ഞെടുത്ത പാക്കേജ് ആണ്. ഒരു ERP സമ്പ്രദായത്തിന് വലിയ നിക്ഷേപം ആവശ്യമായതിനാൽ ഒരിക്കൽ പാക്കേജ് തിരഞ്ഞെടുത്ത് വാങ്ങിക്കഴിഞ്ഞാൽ മറ്റൊരു പാക്കേജിലേക്ക് എളുപ്പത്തിൽ മാറാനാകില്ല. സംരംഭത്തിന്റെ ആവശ്യങ്ങൾക്ക് ഇണങ്ങുന്നതാകണം തിരഞ്ഞെടുത്ത പാക്കേജ് എന്ന വസ്തുതക്ക് പ്രാധാന്യം നൽകിക്കൊണ്ട് വേണം അത് വിലയിരുത്താൻ.

പ്രോജക്ട് പ്ലാനിംഗ്

ഈ ഘട്ടത്തിൽ നിർവ്വഹണ പ്രക്രിയ ആസൂത്രണം ചെയ്ത് നവമാതൃക സൃഷ്ടിക്കുന്നു. വിവിധ പ്രവർത്തനങ്ങൾക്കുള്ള സമയക്രമവും സമയപരിധിയും നിശ്ചയിക്കുന്നു. ബന്ധപ്പെട്ട ജീവനക്കാരുടെ കർത്തവ്യങ്ങളും ഉത്തരവാദിത്വങ്ങളും കണ്ടെത്തി ഓരോരുത്തർക്കും പങ്കിട്ട് നൽകുന്നു. പദ്ധതി എപ്പോൾ തുടങ്ങണം, എങ്ങനെ ചെയ്യണം, എപ്പോൾ അവസാനിപ്പിക്കണം എന്നീ കാര്യങ്ങൾ തീരുമാനിക്കുന്നത് ഈ ഘട്ടത്തിലാണ്.

ഗ്യാപ് അനാലിസിസ്

ഒരു സംരംഭത്തിന്റെ എല്ലാ ആവശ്യങ്ങളും പൂർണ്ണമായി സഫലീകരിക്കാൻ വിപണിയിൽ ലഭ്യമായ ഒരു ERP പാക്കേജിനും സാധിക്കില്ല. ERP യുടെ ഉടമസ്ഥർ അവരുടെ സോഫ്റ്റ് വെയർ എല്ലാ പ്രശ്നങ്ങൾക്കും പരിഹാരം നൽകുമെന്ന് അവകാശപ്പെടാറുണ്ട് എങ്കിലും, ചില കുറവുകൾ ഉണ്ടാകും.

ഒരു സംരംഭത്തിന്റെ എൻപത് ശതമാനം ആവശ്യങ്ങൾ നിറവേറ്റാൻ മാത്രമേ മികച്ച ഒരു ERP പാക്കേജിന് പോലും സാധിക്കുകയുള്ളു. അതുകൊണ്ട് ERP നടപ്പിലാക്കിയതിന് ശേഷമുള്ള ഘട്ടങ്ങളിൽ ഈ കുറവുകൾ വിശകലനം ചെയ്യുകയും പരിഗണിക്കുകയും വേണം.

വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങ് (BPR)

ചിലവ്, ഗുണനിലവാരം, സേവനം, വേഗത തുടങ്ങിയവയുടെ പ്രകടനം മെച്ചപ്പെടുത്തുവാനായി ഒരു സംരംഭത്തിലെ വാണിജ്യ പ്രവർത്തനങ്ങൾ മാതൃകാപരമായി പുനർവിചിന്തനം നടത്തി നവമാതൃക സൃഷ്ടിക്കുന്നതാണ് വാണിജ്യപ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങ്. ERP സമ്പ്രദായം നടപ്പാക്കുന്നതിൽ BPR-നുള്ള പ്രാധാന്യം മുൻപ് വിശദീകരിച്ചതാണ് (ഭാഗം 10.4 പരിശോധിക്കുക).

സ്ഥാപിക്കലും പ്രവർത്തനയോഗ്യമാക്കലും

ERP നടപ്പാക്കുന്നതിലെ പ്രധാന നിർവ്വഹണ ഘട്ടമാണ് ഇത്. ഒരു പുതിയ ERP പാക്കേജ് സ്ഥാപിക്കുന്നതിന് മുമ്പ് സംരംഭത്തിന്റെ പ്രവർത്തനങ്ങൾ മൊത്തത്തിൽ വിശദമായി സൂക്ഷ്മപരിശോധന നടത്തേണ്ടതാണ്. നിലവിലുള്ള സമ്പ്രദായം മാറ്റി പുതിയ ERP സമ്പ്രദായം കൊണ്ട് വരുന്നതിന് പകരം അനുയോജ്യമായ ഒരു പ്രവർത്തനമാതൃക സൃഷ്ടിക്കുന്നത് ഉത്തമവും ഫലപ്രദവുമായിരിക്കും. നടപ്പിലാക്കാൻ പോകുന്ന യഥാർത്ഥ ERP യുടെ ഒരു ലഘുരൂപമായിരിക്കും ഈ പ്രവർത്തനമാതൃക. ഭാവിയിൽ തുടർച്ചയായ പരീക്ഷണങ്ങൾ നടത്തി ഈ പ്രവർത്തന മാതൃകയുടെ പോരായ്മകൾ കണ്ടെത്തുകയും യഥാർഥ ERP സമ്പ്രദായം കൊണ്ട് വരുമ്പോൾ അത്തരം പോരായ്മകൾ തടയുവാനുള്ള ചുവട്വയ്പുകൾ നടത്തുകയും വേണം. ERP സമ്പ്രദായം പ്രവർത്തനയോഗ്യമാക്കുന്ന ജോലി പരിചയ സമ്പന്നരായ ജീവനക്കാരെ ഏൽപ്പിക്കണം.

നിർവഹണ സംഘത്തിന്റെ പരിശീലനം

സമ്പ്രദായം നടപ്പാക്കുന്നതിനും തുടർന്ന് പ്രവർത്തിപ്പിക്കുന്നതിനുമായി കമ്പനി അതിലെ ജീവനക്കാരെ പരിശീലിപ്പിക്കുന്ന ഘട്ടമാണ് ഇത്. ERP ഉടമസ്ഥരും വേതനാടിസ്ഥാനത്തിലുണ്ടായിരുന്ന ഉപദേശകരും ERP പ്രവർത്തന യോഗ്യമാക്കിയശേഷം സ്ഥാപനം വിട്ട് പോകും. അതുകൊണ്ട് പുതിയ കാര്യങ്ങൾ പഠിക്കുവാനും രൂപാന്തരപ്പെടാനും താൽപ്പര്യപ്പെടുന്ന ശരിയായ മനോഭാവമുള്ള ജീവനക്കാരെ കമ്പനി തിരഞ്ഞെടുക്കണം. അവർക്ക് സാങ്കേതികവിദ്യയെക്കുറിച്ച് ഭയമുണ്ടാവുകയും ചെയ്യരുത്.

പരീക്ഷണം

ഒരു സംരംഭത്തിൽ സാങ്കേതികപരവും നിർവഹണപരവുമായുള്ള മേഖലകളിൽ സോഫ്റ്റ് വെയർ ശരിയായി പ്രവർത്തിക്കുന്നുണ്ടോ എന്ന് പരിശോധിച്ച് ഉറപ്പ് വരുത്തുന്നു. മാതൃകാ

ഡാറ്റ ഉപയോഗിച്ച് കൊണ്ട് ഫലം സാധൂകരിക്കാം. ഏതെങ്കിലും പോരായ്മകൾ ഈ ഘട്ടത്തിൽ കണ്ടെത്തിയാൽ, ERP സമ്പ്രദായം പ്രവർത്തിപ്പിക്കുന്നതിന് മുമ്പ് അത് പരിഹരിക്കണം.

സജീവമാക്കുന്നു

ERP സമ്പ്രദായം സ്ഥാപനത്തിൽ എല്ലായിടത്തും ലഭ്യമാക്കുന്ന ഘട്ടമാണ് ഇത്. ഈ ഘട്ടത്തിനുശേഷം പദ്ധതി ഉപയോകയോഗ്യമാകും. പ്രവർത്തന സജ്ജമാക്കി പരീക്ഷിച്ച ശേഷം, പോരായ്മകൾ പരിഹരിക്കുകയും വിവരണങ്ങൾ ശരിയാണോ എന്ന് പരിശോധിക്കുകയും ചെയ്യുന്നു. തുടർന്ന് സംരംഭവുമായി ബന്ധപ്പെട്ട പ്രവർത്തനങ്ങൾ നടപ്പാക്കാനായി പദ്ധതി സജീവമാക്കുന്നു. നിർബാധമായ പ്രവർത്തിന് ERP ഉടമസ്ഥർ സംരംഭത്തിന് ആവശ്യമായ സഹായ സേവനങ്ങൾ നൽകുന്നു.

ഉപയോക്താക്കൾക്കുള്ള പരിശീലനം

യഥാർത്ഥ ERP ഉപയോക്താക്കളെ പദ്ധതി കൈകാര്യം ചെയ്യാൻ പരിശീലിപ്പിക്കേണ്ടതുണ്ട്. പദ്ധതി സജീവമാക്കുന്നതിന് മുമ്പാണ് ഈ ഘട്ടം തുടങ്ങുന്നത്. പുതിയ പദ്ധതി ഉപയോഗിക്കേണ്ട ജീവനക്കാരെ കണ്ടെത്തുന്നു. അവരുടെ കഴിവുകൾ നിരീക്ഷിക്കുകയും കഴിവിനനുസരിച്ച് അവരെ പല സംഘങ്ങളായി വിഭജിക്കുകയം ചെയ്യുന്നു. തുടർന്ന് ഓരോ സംഘത്തേയും പുതിയ പദ്ധതിയിൽ പരിശീലിപ്പിക്കുന്നു. ERP സമ്പ്രദായത്തിന്റെ വിജയം ഉപയോക്താക്കളുടെ കൈകളിലായതിനാൽ ഈ പരിശീലനം വളരെ പ്രധാനപ്പെട്ടതാണ്.

നടപ്പാക്കലിന് ശേഷമുള്ള ഘട്ടം

പുതിയ ERP സമ്പ്രദായം സ്ഥാപിച്ച് പ്രവർത്തിപ്പിച്ച് തുടങ്ങിയാൽ പരിഷ്ക്കരിക്കലുകളും വിലയിരുത്തലുകളും ആവശ്യമായി വരും. അതുകൊണ്ട് നടപ്പാക്കിയശേഷം, ERP സമ്പ്രദായത്തിന് വേണ്ടി തയ്യാറാക്കിയ ലക്ഷ്യങ്ങളും അവ എത്രമാത്രം നേടിയെന്നതും പരിശോധിക്കണം. ഈ ഘട്ടത്തിൽ പിഴവുകൾ ശരിയാക്കുകയും പ്രവർത്തനക്ഷമത മെച്ചപ്പെടുത്തുവാൻ യുക്തമായ നടപടികൾ സ്വീകരിക്കുകയും വേണം.

10.6 ERP പരിഹാരം ലഭ്യമാക്കുന്നവർ/ERP പാക്കേജുകൾ

പാക്കേജ് തിരഞ്ഞെടുക്കലാണ് ERP സമ്പ്രദായം നടപ്പാക്കുന്നതിൽ വളരെ നിർണായകമായിട്ടുള്ളത്. ഒരു ERP പാക്കേജ് കൃത്യമായി തിരഞ്ഞെടുത്ത്, നീതിപൂർവ്വം നടപ്പാക്കി ഫലപ്രദമായി ഉപയോഗിച്ചാൽ സംരംഭത്തിന്റെ ഉൽപ്പാദനക്ഷമത വർദ്ധിക്കും. ERP നിർമാതാക്കൾ പാക്കേജിന് വേണ്ടിയുള്ള ഗവേണഷണത്തിനും വികസനത്തിനും വേണ്ടി ധാരാളം ധനവും, സമയവും, അദ്ധ്വാനവും ഉപയോഗിക്കുന്നു. ലോകത്ത് ധാരാളം ERP നിർമ്മാതാക്കളുണ്ട്. ഓറക്കിൾ, എസ്.എ.പി, ഉഡു, മൈക്രോസോഫ്റ്റ് ഡൈനാമിക്സ്, ടാലി എന്നിവ പ്രശസ്തമായ ചില ERP പാക്കേജുകളാണ്.

ഓറക്കിൾ

ഓറക്കിൾ കോർപ്പറേഷന്റെ ആസ്ഥാനം USA -യിലെ കാലിഫോർണിയയിലുള്ള റെഡ്വുഡ് ഷോർ ആണ്. ERP സമ്പ്രദായത്തിന്റെ പേരിനേക്കാൾ ഉപരി ഡാറ്റബേസ് സമ്പ്രദായത്തിന്റെ പേരിലാണ് ഓറക്കിൾ അറിയപ്പെടുന്നത്. ഓറക്കിളിന്റെ ERP പാക്കേജ് ധനകാ

ര്യവും വ്യാപാര ഇടപാടുകളും കാര്യക്ഷമമായി കൈകാര്യം ചെയ്യുന്ന ഘടകങ്ങൾ പ്രധാനം ചെയ്യുന്നു. കൂടാതെ വിതരണക്കാരും ഉപഭോക്താവുമായി നല്ലരീതിയിലുള്ള സമ്പർക്കം, ഫലപ്രദമായ നിർമാണ വിശകലനം, കാര്യക്ഷമമായ മാനവിക വിഭവ നിയന്ത്രണം, മെച്ചപ്പെട്ട വില നിർണയം എന്നിവയ്ക്കായുള്ള ഘടകങ്ങളും നൽകുന്നു. കസ്റ്റമർ റിലേഷൻസ് മാനേജ്മെന്റ് (CRM), സ്പ്ലൈ ചെയിൻ മാനേജ്മെന്റ് (SCM) എന്നീ സോഫ്റ്റ്‌വെയറുകളും ഓറക്കിൾ നിർമ്മിച്ചിട്ടുണ്ട്.

പ്രസിദ്ധമായ ERP പാക്കേജ് നിർമ്മാതാക്കളായ JD എഡ്വേർസിനെ 2003-ൽ പീപ്പിൾ സോഫ്റ്റ് കമ്പനിയും, പിന്നീട് 2005-ൽ ഓറക്കിൾ കോർപ്പറേഷനും വാങ്ങി.

SAP

ഡാറ്റാ പ്രൊസസ്സിങ്ങിനായുള്ള സിസ്റ്റം, ആപ്ലിക്കേഷൻസ്, പ്രോഡക്റ്റ്സ് എന്നിവയാണ് SAP കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. 1972-ൽ വാൾഡ്രോപ്പ് ആസ്ഥാനമാക്കി സ്ഥാപിക്കപ്പെട്ട ഒരു ജർമ്മൻ ബഹുരാഷ്ട്ര സോഫ്റ്റ്‌വെയർ കോർപ്പറേഷനാണ് അത്. വാണിജ്യ പരിഹാരത്തിനായുള്ള ഏകീകൃത സോഫ്റ്റ്‌വെയർ വികസിപ്പിച്ച് കൊണ്ടാണ് കമ്പനി തുടങ്ങിയത്. തുടക്കത്തിൽ ബഹുരാഷ്ട്രകമ്പനികളെ ലക്ഷ്യമിട്ടുകൊണ്ടുള്ള സോഫ്റ്റ്‌വെയറുകളാണ് കമ്പനി വികസിപ്പിച്ചിരുന്നത്. മികച്ച സ്വീകാര്യത ലഭിച്ച ശേഷം, ചെറുകിട സംരംഭങ്ങൾക്ക് വേണ്ടിയുള്ള പാക്കേജുകൾ കമ്പനി വികസിപ്പിച്ചു. കസ്റ്റമർ റിലേഷൻസ് മാനേജ്മെന്റ് (CRM), സപ്ലൈ ചെയിൻ മാനേജ്മെന്റ് (SCM), പ്രോഡക്ട് ലൈഫ് സൈക്കിൾ മാനേജ്മെന്റ് (PLM) എന്നീ സോഫ്റ്റ്‌വെയറുകളും SAP വികസിപ്പിച്ചിട്ടുണ്ട്.

ഉഡു

ഒരു സ്വതന്ത്ര ERP ആണ് ഉഡൂ. സ്ഥാപനത്തിന്റെ ആവശ്യത്തിനനുസരിച്ച് സ്വതന്ത്ര ERP -യിലുള്ള സോഴ്സ്കോഡ് അല്ലെങ്കിൽ പ്രോഗ്രാം വ്യത്യാസപ്പെടുത്തുവാൻ സാധിക്കും. മുമ്പ് 2014 മെയ് മാസം വരെ ഉഡൂ അറിയപ്പെട്ടിരുന്നത് സ്വതന്ത്ര ERP എന്ന പേരിലാണ്.

മൈക്രോസോഫ്റ്റ് ഡൈനാമിക്സ്

വാഷിങ്ടണിലെ റെഡ്മണ്ട് ആസ്ഥാനമാക്കി പ്രവർത്തിക്കുന്ന ഒരു അമേരിക്കൻ ബഹുരാഷ്ട്ര കോർപ്പറേഷൻ ആണ് മൈക്രോസോഫ്റ്റ്. മൈക്രോസോഫ്റ്റ് ബിസിനസ് സൊല്യൂഷന്റെ മൈക്രോസോഫ്റ്റ് ഡൈനാമിക്സ് പ്രാഥമികമായി നൽകുന്നത് ഇടത്തരം സംരംഭങ്ങളെ ലക്ഷ്യമിടുന്ന ഒരു കൂട്ടം സംരംഭ വിഭവ ആസൂത്രണ ഉൽപ്പന്നങ്ങളാണ്. എളുപ്പത്തിൽ ഈ പാക്കേജ് സ്ഥാപിച്ച് ഉപയോഗിക്കാം എന്ന് മാത്രമല്ല ഉപയോക്താവിന് മെച്ചപ്പെട്ട സമ്പർക്ക സംവിധാനവും അത് പ്രധാനം ചെയ്യുന്നു. കസ്റ്റമർ റിലേഷൻസ് മാനേജ്മെന്റ് (CRM) സോഫ്റ്റ്‌വെയറും അത് നൽകുന്നു.

ടാലി ERP

ഇന്ത്യയിലെ ബാംഗ്ലൂർ ആസ്ഥാനമായ ഒരു പ്രൈവറ്റ് ലിമിറ്റഡ് സോഫ്റ്റ്‌വെയർ കമ്പനിയാണ് ടാലി സൊല്യൂഷൻസ്. വരവ് ചിലവ് കണക്കുകൾ, ചരക്ക് വിവര പട്ടിക, ശമ്പള വിവര പട്ടിക എന്നീ വാണിജ്യ വ്യാപാര ഇടപാടുകൾക്ക് വേണ്ടിയുള്ള ഒരു സോഫ്റ്റ്‌വെയറാണ് ടാലി ERP. സമീപഭാവിയിൽ പുതിയ ERP നിർമ്മാതാക്കൾ, ERP പാക്കേജുകൾ അവതരിപ്പിക്കുകയോ നിലവിലുള്ളവയ്ക്ക് കൂടുതൽ സൗകര്യങ്ങളും കഴിവുകളും ലഭ്യമാക്കുകയോ ചെയ്യും. ERP പരിഹാരം തിരഞ്ഞെടുക്കുന്നത് ഒരു ഗൗരവമുള്ള ഉദ്യമമായതിനാൽ വളരെ ശ്രദ്ധയോടെ അത് നടപ്പാക്കണം.

മുകളിൽ പറഞ്ഞ പാക്കേജുകളെ കൂടാതെ അറിയപ്പെടുന്ന മറ്റ് അനേകം പാക്കേജുകളും വിപണിയിലുണ്ട്. എപികോർ, ഇൻഫോർ, സേജ്, റീച്ച്, ഓപ്പൺ ബ്രാവോ, അപ്പാഷെ OF ബിസ്സ്, കോമ്പിയർ, വെബ് ERP, ERP5 തുടങ്ങിയവ അവയിൽ ചിലതാണ്.

10.7 ERP – യുടെ നേട്ടങ്ങളും കോട്ടങ്ങളും

ERP പാക്കേജ് ഒരു സംരംഭത്തിൽ സ്ഥാപിക്കുമ്പോൾ അതിന് പ്രത്യക്ഷവും പരോക്ഷവുമായ ഗുണങ്ങളുണ്ടാകും. കൂടാതെ ചില വെല്ലുവിളികളുമുണ്ടാകും. ERP സമ്പ്രദായത്തിന്റെ നേട്ടങ്ങളേയും പരിമിതികളേയും കുറിച്ച് നമുക്ക് ചർച്ച ചെയ്യാം.

10.7.1 ERP സമ്പ്രദായത്തിന്റെ നേട്ടങ്ങൾ

ഒരു സംരംഭത്തിൽ ERP സമ്പ്രദായം നടപ്പാക്കുന്നത് കൊണ്ട് ധാരാളം നേട്ടങ്ങളുണ്ട്. ചില പ്രധാനപ്പെട്ട നേട്ടങ്ങൾ ചുരുക്കത്തിൽ വിശദീകരിക്കുന്നു.

1. മെച്ചപ്പെട്ട വിഭവ ഉപയോഗം:

സോഫ്റ്റ്‌വെയർ സ്ഥാപിക്കുന്നതിലൂടെ മെച്ചപ്പെട്ട വിഭവ ആസൂത്രണവും നിയന്ത്രണവും ഒരു സംരംഭത്തിന് സാധ്യമാകും. അങ്ങനെ വിഭവങ്ങൾ പാഴാക്കുന്നത് കുറയ്ക്കുവാനും വിഭവങ്ങളുടെ മെച്ചപ്പെട്ട ഉപയോഗം ഉറപ്പ് വരുത്തുവാനും സാധിക്കും.

2. മെച്ചപ്പെട്ട ഉപഭോക്തൃസംതൃപ്തി:

സാധനങ്ങൾക്കോ സേവനങ്ങൾക്കോ വേണ്ടിയുള്ള ഉപഭോക്താവിന്റെ ആവശ്യകത പരമാവധി നിർവഹിക്കുന്നതിനെയാണ് ഉപഭോക്തൃസംതൃപ്തി എന്ന് പറയുന്നത്. അധിക ധനമോ സമയമോ നഷ്ടപ്പെടാത്തവിധം ഉപഭോക്താവിന് കൂടുതൽ ശ്രദ്ധയും സേവനവും പ്രധാനം ചെയ്യാൻ ERP സമ്പ്രദായം ഉപയോഗിക്കുന്നത് കൊണ്ട് സാധിക്കും. വെബ് അടിസ്ഥാനമാക്കിയുള്ള ERP ലഭ്യമായതോടെ ഉപഭോക്താവിന് വീട്ടിൽ ഇരുന്നുകൊണ്ട് നിർദേശം (ഓർഡർ) നൽകുവാനും, ഓർഡറിന്റെ തൽസ്ഥിതി അറിയുവാനും, പണം നൽകുവാനും സാധിക്കുന്നു.

3. ശരിയായ വിവരം നൽകുന്നു:

കിടമൽസരം നിലനിൽക്കുന്ന ഈ കാലത്ത്, ഭാവിപരിപാടികൾ ബുദ്ധിപരമായി ആസൂത്രണം ചെയ്ത് നടപ്പാക്കേണ്ടത് ഒരു സംരംഭത്തിന്റെ ആവശ്യമാണ്. അതുകൊണ്ട് മികച്ച നിലവാരമുള്ള ഉചിതവും കൃത്യവുമായ വിവരം ഒരു സംരംഭത്തിന് ആവശ്യമാണ്. സംരംഭത്തിന്റെ ശോഭനമായ ഭാവി പരിപാടികൾ ആസൂത്രണം ചെയ്യുവാൻ ആവശ്യമായ ഇത്തരം വിവരങ്ങൾ ഒരു ERP സമ്പ്രദായം നൽകുന്നു.

4. തീരുമാനങ്ങൾ എടുക്കുന്നതിനുള്ള കഴിവ്:

തീരുമാനങ്ങൾ എടുക്കുന്നവർക്ക് ലഭ്യമാകുന്ന കൃത്യവും പ്രസക്തവുമായ വിവരം ഉപയോഗപ്പെടുത്തി വ്യവസായയുടെ നിർബാധമായ പ്രവർത്തനത്തിന് വേണ്ടി മെച്ചപ്പെട്ട തീരുമാനങ്ങൾ എടുക്കുവാൻ സാധിക്കുന്നു. സംരംഭത്തിന്റെ മികച്ച തീരുമാനം എതിരാളികളേക്കാൾ ഒരുപടി മുന്നിൽ നിൽക്കുവാൻ അവരെ സഹായിക്കുന്നു.

5. സാഹചര്യങ്ങൾക്കനുസരിച്ച് കൂടുതൽ പൊരുത്തപ്പെടുന്നു:

ERP സ്ഥാപനത്തിന്റെ മൊത്തത്തിലുള്ള വഴക്കവും ഉൽപ്പാദന പ്രവർത്തനങ്ങളുടെ വഴക്കവും ERP സമ്പദ്രായം വർധിപ്പിക്കുന്നു. സാഹചര്യങ്ങളിൽ പെട്ടെന്നുണ്ടാകുന്ന മാറ്റം ഉൾക്കൊള്ളുവാൻ സാഹചര്യങ്ങളോട് ഇണക്കമുള്ള ഒരു സ്ഥാപനത്തിന് സാധിക്കുന്നു. വകുപ്പുകളുടെ അതിർ വരമ്പുകൾ ഇല്ലാതെ സംരഭത്തിലെ വിവരം ലഭ്യമാക്കി സ്ഥാപനത്തെ വഴക്കമുള്ളതായി നിലനിർത്താൻ ERP സമ്പ്രദായത്തിന് സാധിക്കുന്നു.

6. വിവര ഏകീകരണം:

വിവധ വകുപ്പുകളുടെ ഏകീകരണം സാധ്യമാകും എന്നതാണ് ERP-യുടെ ഏറ്റവും പ്രധാനപ്പെട്ട നേട്ടം. തന്മൂലം സംരഭത്തെക്കുറിച്ചുള്ള ഏകീകൃത വിവരം നമുക്ക് ലഭ്യമാകും. സംരഭത്തെക്കുറിച്ചുള്ള പൂർണ്ണ വിവരം ഒരു കേന്ദ്രീകൃത ഡാറ്റാബേസിൽ സംഭരിക്കപ്പെടുന്നതിനാൽ സ്ഥാപനത്തിലെ വിവിധ വകുപ്പുകളിൽ നടക്കുന്ന എല്ലാ പ്രവർത്തനങ്ങളുടേയും പൂർണ്ണ ദൃശ്യത കൈവരിക്കാനാകും.

കൂടാതെ, ഉയർന്ന സുരക്ഷിതത്വം, ബുദ്ധിപരമായ വാണിജ്യപ്രവർത്തനം, മൊത്തമായുള്ള അവലോകന സമ്പ്രദായം, വിതരണക്കാരുടെ പ്രവർത്തനശേഷി വർദ്ധിപ്പിക്കൽ, അതിനൂതന സാങ്കേതിക വിദ്യകളുടെ ഉപയോഗം, വളരെ വേഗത്തിലുള്ള ഉൽപ്പന്ന വിതരണം, മികവുറ്റ വിശകലനം, ആസൂത്രണ മികവ് എന്നിവയും ERP പ്രധാനം ചെയ്യുന്നു.

10.7.2 ERP നടപ്പാക്കുന്നത് കൊണ്ടുള്ള നഷ്ടങ്ങൾ

ഒരു സംരഭത്തിൽ ERP പാക്കേജ് നടപ്പാക്കുന്നത് കൊണ്ട് ഉണ്ടാകുന്ന ചില പ്രശ്നങ്ങളും പരിമിതികളും താഴെ തന്നിരിക്കുന്നു.

1. ചെലവ് വർധിക്കുന്നു

ERP സോഫ്റ്റ് വെയർ പ്രവർത്തനയോഗ്യമാക്കി നടപ്പാക്കുന്നതിനുള്ള ചെലവ് വളരെ കൂടുതലാണ്. വലിയ വിലയുള്ള പാക്കേജ്, ബന്ധപ്പെട്ട അനുമതി പത്രത്തിന്റെ വില, മറ്റ് ചെലവുകൾ എന്നിവയാണ് ERP നടപ്പാക്കുന്നതിലെ പ്രധാന പ്രശ്നങ്ങൾ. IT യ്ക്കുവേണ്ടി പുതുതായി തയ്യാറാക്കേണ്ട ഭൗതിക സാഹചര്യം നെറ്റ്വർക്ക് സൗകര്യങ്ങളുടെ നവീക

രണം തുടങ്ങിയ പരോക്ഷ ചെലവുകൾ ERP നടപ്പാക്കുന്നതിന് അധികമായി വരാം. ഒരു ചെറുകിട സംരഭത്തെ സംബന്ധിച്ചിടത്തോളം അത്തരം ചിലവേറിയ ഒരു പാക്കേജ് വാങ്ങി സ്ഥാപിക്കുന്നത് ഉചിതമാകില്ല.

2. കൂടുതൽ സമയം ചെലവഴിക്കേണ്ടി വരുന്നു

ERP നടപ്പാക്കുന്ന പ്രവർത്തനം വിവിധ ഘട്ടങ്ങളിലൂടെ കടന്ന് പോകുന്നതും വളരെ കൂടുതൽ സമയം എടുക്കുന്നതുമാണ്. അത് പൂർത്തീകരിക്കാനും പ്രവർത്തനസജ്ജമാക്കാനും മാസങ്ങളോ അല്ലെങ്കിൽ ചിലപ്പോൾ ഒന്നോ രണ്ടോ വർഷങ്ങളോ വേണ്ടി വരും.

3. പരിശീലനം ലഭിച്ച അധിക ജീവനക്കാരുടെ ആവശ്യകത

ERP സമ്പ്രദായം പ്രവർത്തിപ്പിക്കുന്നതിന് പരിശീലനവും പരിചയവുമുള്ള ജീവനക്കാരെ സംരഭത്തിൽ നിയമിക്കേണ്ടതുണ്ട്. ശരിയായ ERP പാക്കേജ് തിരഞ്ഞെടുത്തതുകൊണ്ട് മാത്രം സംരഭത്തിന്റെ വിജയം ഉറപ്പിക്കാനാകില്ല. കഴിവും പരിശീലനവും സിദ്ധിച്ച ആളുകൾ ERP സമ്പ്രദായം ഉപയോഗിക്കുന്നതും വളരെ പ്രാധാന്യം അർഹിക്കുന്നു.

4. ഉപയോഗിക്കുന്നതും നിലനിർത്തുന്നതുമായി ബന്ധപ്പെട്ട പ്രശ്നങ്ങൾ

സംരഭത്തിലെ ഇപ്പോഴുള്ള പ്രവർത്തനങ്ങളിൽ കാതലായ മാറ്റം ഒരു ERP സമ്പ്രദായം നടപ്പാക്കുന്നതിന് അനിവാര്യമാണ്. മാറ്റങ്ങളെ പ്രതിരോധിക്കുന്ന മനുഷ്യസഹജമായ സ്വഭാവം കാരണം അതുമായി പൊരുത്തപ്പെടാൻ സംരഭത്തിലെ ജീവനക്കാർക്കും നേതൃത്വത്തിനും ചിലപ്പോൾ പ്രയാസം നേരിടാറുണ്ട്. പദ്ധതി ഒരിക്കൽ പൂർത്തീകരിക്കപ്പെടും എന്ന ധാരണയിലാണ് മിക്കവാറും സംരഭങ്ങൾ ERP നടപ്പിലാക്കുന്നത്. എന്നാൽ ERP സമ്പ്രദായം നടപ്പാക്കൽ ഒരിക്കലും അവസാനിക്കാത്ത ജീവിതചക്രമുള്ള തുടർ പ്രവർത്തനങ്ങളടങ്ങിയതാണ്. അതുകൊണ്ട് ERP സമ്പ്രദായം നിലനിർത്തുക എന്നത് വളരെ പ്രയാസകരമാണ്.

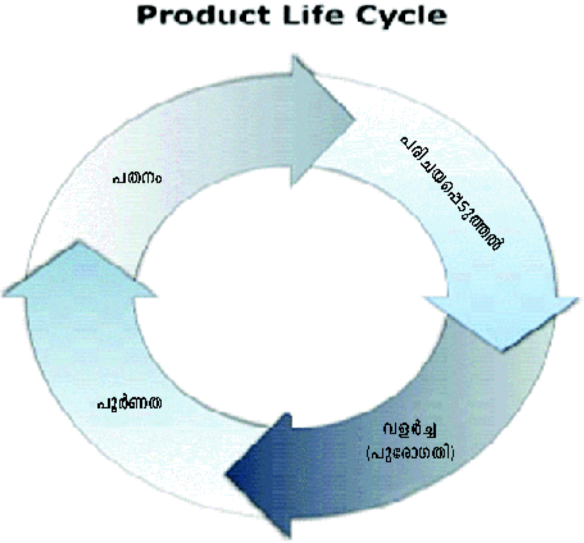
10.8. ERP – യും ബന്ധപ്പെട്ട സാങ്കേതിക വിദ്യകളും

അസംസ്കൃത വസ്തുക്കളുടെ നിയന്ത്രണം, ഉല്പാദന ആസൂത്രണം, വിൽപ്പന, വിതരണം, ധനകാര്യം തുടങ്ങി വേറിട്ട് നിൽക്കുന്ന വാണിജ്യ പ്രവർത്തനങ്ങളെ ഒരു ആപ്ലിക്കേഷൻ ഉപയോഗിച്ച് ERP സമ്പ്രദായം ഏകീകരിക്കുന്നു. ഈ ഭാഗത്ത് വിശദീകരിക്കാൻ പോകുന്ന മറ്റ് ചില സാങ്കേതികവിദ്യകൾ തനത് ERP പാക്കേജിനൊപ്പം ഉപയോഗിച്ചാൽ സംരഭത്തിന്റെ പ്രകടനം ഗണ്യമായി മെച്ചപ്പെടും. ERP പാക്കേജിനൊപ്പം ഉപയോഗിക്കുന്ന ബന്ധപ്പെട്ട ചില സാങ്കേതിക വിദ്യകൾ നമുക്ക് ചർച്ച ചെയ്യാം.

ഉല്പന്നത്തിന്റെ ജീവിത ചക്രനിയന്ത്രണം (പ്രോഡക്റ്റ് ലൈഫ് സൈക്കിൾ മാനേജ്മെന്റ് PLM)

ഒരു ഉല്പന്നത്തിന്റെ ആയുസ് നിർണ്ണയിക്കാനാണ് അതിന്റെ ജീവിതചക്രം ഉപയോഗിക്കുന്നത്. ഒരു ഉല്പന്നത്തിന്റെ ജീവിതചക്രം പൂർണ്ണമായും നിയന്ത്രിക്കുന്ന പ്രവർത്തനമാണ് PLM. പുതിയ ഉൽപ്പന്നം വികസിപ്പിച്ച് പരിചയപ്പെടുത്തുക, വിപണന സാധ്യത വർദ്ധിപ്പിക്കുക, വിപണി സാധ്യതകൾ പൂർണ്ണമാവുക, എതിരാളികളുടെ സമാന ഉല്പന്നങ്ങളുമായി മത്സരിക്കാനാകാതെ വിപണനം കുറയുക എന്നീ നാല് ഘട്ടങ്ങളുള്ള ഉല്പപ്പന്നത്തിന്റെ ജീവിത ചക്ര മാതൃക ചിത്രം 10.7-ൽ കാണിച്ചിരിക്കുന്നു.

ഒരു നവമാതൃക പരിചയപ്പെടുത്തുമ്പോൾ, നിർമ്മാണ പ്രക്രിയയിൽ നൂതന സാങ്കേതികവിദ്യകൾ ഉപയോഗിക്കുന്നു എന്ന് ഉറപ്പിക്കാനും വിപണന സാധ്യതകൾ വർദ്ധിപ്പിക്കാനുമായി PLM നടപ്പാക്കി ഉൽപ്പന്നത്തിന്റെ നിലവാരം മെച്ചപ്പെടുത്തുന്നു. പുതിയ ഉൽപ്പന്നം നിർമ്മിക്കുമ്പോഴും നിലവിലുള്ളത് പരിഷ്ക്കരിക്കുമ്പോഴും കമ്പനി അതിന്റെ ഉപഭോക്താക്കളേയും, വിപണിയേയും എതിരാളികളേയും മനസ്സിലാക്കിയിരിക്കണം. ഉൽപ്പന്നത്തിന്റെ ജീവിത ചക്രത്തിൽ നിന്ന് ലഭിക്കുന്ന വിവരമാണ് അതിന്റെ നിലവാരം അല്ലെങ്കിൽ അവസര മനസ്സിലാക്കാൻ ഒരു സംരഭത്തെ സഹായിക്കുന്നത്.



ചിത്രം 10.7: നാല് ഘട്ടങ്ങളുള്ള ഉൽപ്പന്നത്തിന്റെ ജീവിത ചക്രം കാണിക്കുന്ന പൊതുവായ രേഖാചിത്രം.

ഉപഭോക്തൃ ബന്ധം കൈകാര്യം ചെയ്യുന്നു (കസ്റ്റമർ റിലേഷൻസ് മാനേജ്മെന്റ് – CRM)

സാധനങ്ങളും സേവനങ്ങളും സ്വീകരിക്കുകയോ ഉപയോഗിക്കുകയോ ചെയ്യുന്ന വ്യക്തിയോ സംഘമോ ആണ് ഉപയോക്താവ്. ഉപയോക്താക്കളാണ് ഒരു സംരംഭത്തിന്റെ ഏറ്റവും പ്രധാനപ്പെട്ട ഭാഗം. സംരംഭത്തിന്റെ വിജയം ഉപയോക്താവുമായുള്ള നല്ല ബന്ധത്തെ ആശ്രയിച്ചിരിക്കും. വ്യാപകമായി ഉപയോഗിക്കുന്ന ഒരു പദമായ CRM-ൽ ഉപഭോക്താക്കളും സംരംഭവുമായുള്ള ബന്ധം കൈകാര്യം ചെയ്യുവാൻ നയങ്ങൾ ഉണ്ടാകും. കമ്പനിയുടെ പ്രത്യേക സാഹചര്യം, ഉപഭോക്താക്കളുടെ ആവശ്യങ്ങൾ പ്രതീക്ഷകൾ എന്നിവ CRM-ന്റെ ഉദ്ദേശ്യ ലക്ഷ്യങ്ങളിൽ ഉൾപ്പെടുത്തണം. ഉപഭോക്താവുമായുള്ള ബന്ധം സ്ഥാപിക്കുന്നതിനും, അത് നിലനിർത്തുന്നതിനും, വിപുലീകരിക്കുന്നതിനുംവേണ്ടി വ്യാപകമായി ഉപയോഗിക്കുന്ന ഒരു സമീപനമാണ് CRM. അത് IT സംഘത്തിന്റേയോ, ഉപഭോക്തൃ സേവന വകുപ്പിന്റേയോ മാത്രം ഉത്തരവാദിത്വത്തിൽ പെടുന്ന കാര്യമല്ല. അത് സംരംഭത്തിന്റെ എല്ലാ പ്രധാന ഭാഗങ്ങളെയും സ്പർശിക്കുന്നു. CRM-ൽ ഉൾപ്പെട്ടിട്ടുള്ള പ്രവർത്തനങ്ങൾ ചിത്രം 10.8-ൽ കാണിച്ചിരിക്കുന്നു.

ഉപഭോക്താക്കളെ കുറിച്ചുള്ള വിവരങ്ങൾ ശേഖരിക്കുക, അത് സംഭരിക്കുക, വിശകലനം ചെയ്യുക തുടങ്ങിയവ അതിൽ ഉൾപ്പെടുന്നു. CRM ന്റെ ഭാഗമായി ഡാറ്റ സംഭരിക്കുമ്പോൾ ഉപഭോക്താവിന്റെ സ്വകാര്യതയും ഡാറ്റയുടെ സുരക്ഷിതത്വവും പരിഗണിക്ക

ചിത്രം 10.8: CRM -ൽ ഉള്ള പ്രവർത്തനങ്ങൾ

ണം. അനുവാദമില്ലാതെ മൂന്നാമത് ഒരാളുമായി ഡാറ്റ പങ്കുവയ്ക്കാൻ പാടില്ല എന്നും നിയമപരമായല്ലാതെ മൂന്നാമത് ഒരാൾ അത് ഉപയോഗിക്കാൻ പാടില്ലെന്നും ഉപഭോക്താവിന് ഉറപ്പ് ലഭിക്കണം. തങ്ങളുടെ ഡാറ്റ ഉപയോഗിക്കുന്ന കമ്പനികൾ ഉപഭോക്താക്കൾക്ക് നേട്ടം ഉണ്ടാക്കി നൽകണം എന്നും അവർ ആവശ്യപ്പെടും.

ഉപഭോക്താവിനെക്കുറിച്ചുള്ള മുഴുവൻ വിവരവും മൊത്തമായി സംഭരിക്കാനുള്ള ഒരു ഡാറ്റാബേസും, ഉപഭോക്താവുമായുള്ള വിനിമയത്തിനും, വിശകലനത്തിനും, സഹായത്തിനും വേണ്ട ഒരു സോഫ്റ്റ്‌വെയറും അടങ്ങിയതാണ് CRM സാങ്കേതികവിദ്യ.

മാനേജ്മെന്റ് ഇൻഫർമേഷൻ സിസ്റ്റം – (MIS)

MIS-ൽ മൂന്ന് ഭാഗങ്ങൾ ഉണ്ടെന്ന് നമുക്ക് കാണാനാകും-നേതൃത്വം, വിവരം, സംവിധാനം, നേതൃത്വമാണ് ആത്യന്തികമായ ഉപയോക്താവ്/തീരുമാനങ്ങൾ എടുക്കുന്നവർ, പ്രൊസസ്സ് ചെയ്ത ഡാറ്റയാണ് വിവരം, സംരംഭത്തെ മൊത്തമായി കാണുന്നതും അതിന്റെ ഏകീകരണവുമാണ് സിസ്റ്റം അഥവാ സംവിധാനം. സാങ്കേതിമായി നോക്കുമ്പോൾ വിവരവ്യവസ്ഥിതി ഒരു സ്ഥാപനത്തിന്റെ ചുറ്റുപാടുകളിൽ നിന്നും ആന്തരിക പ്രവർത്തനങ്ങളിൽ നിന്നും, അതിന്റെ പ്രവർത്തനങ്ങളെ സഹായിക്കാനായി വിവരം ശേഖരിക്കുകയും, സംഭരിക്കുകയും വിതരണം നടത്തുകയും ചെയ്യും. ഒരു സംരംഭത്തിൽ തീരുമാനങ്ങൾ എടുക്കുന്നതിനും, ആശയ വിനിമയത്തിനും, ഏകോപനത്തിനും, നിയന്ത്രണത്തിനും, വിശകലനത്തിനും, മാനസിക ആവിഷ്ക്കാരത്തിനും കൂടി അത് ഉപയോഗിക്കുന്നു. വിവരവ്യവസറിതി അസംസ്കൃത ഡാറ്റയെ ഇൻപുട്ട്, പ്രൊസസ്സിങ്, ഔട്ട്പുട്ട് എന്നീ അടിസ്ഥാന പ്രവർത്തനങ്ങളിലൂടെ ഉപയോഗയോഗ്യമായ വിവരമാക്കി രൂപാന്തരപ്പെടുത്തുന്നു.

ഗുമസ്തർ, സഹായികൾ, തലപ്പത്തുള്ളവർ, ഭാരവാഹികൾ, നിർവാഹകർ തുടങ്ങിയ വിവിധ ഭാഗങ്ങളിലുള്ള ജീവനക്കാർ ഒരു സംരംഭത്തിൽ ഉണ്ടാകും. അവർ എല്ലാവരും MIS ന്റെ ഉപയോക്താക്കളാണ്. ഒരു MIS ഉചിതമായ ഡാറ്റയെ സംരംഭത്തിന് ഉള്ളിൽ നിന്നോ പുറത്ത് നിന്നോ ശേഖരിക്കുന്നു. ഈ ഡാറ്റ പ്രൊസസ്സ് ചെയ്ത് ഒരു ഏകീകൃത ഡാറ്റബേസിൽ സംഭരിക്കുകയും ഉപയോക്താക്കൾക്ക് ആവശ്യമുള്ളപ്പോൾ ലഭ്യമാക്കുകയും ചെയ്യുന്നു. ഉപയോക്താവ് ആവശ്യപ്പെടുമ്പോൾ അവലോകനം തയാറാക്കാനുള്ള കഴിവ് MIS-ന് ഉണ്ട്.

അവലോകനങ്ങൾ തയാറാക്കാനാണ് അത് പ്രധാനമായും ഉപയോഗിക്കുന്നത്. അതിന്റെ അടിസ്ഥാനത്തിൽ തീരുമാനങ്ങൾ ഉണ്ടാകുന്നു. ഒരു സ്ഥാപനത്തിൽ MIS -ന്റെ കർത്തവ്യം എന്തെന്ന് ചിത്രം 10.9 കാണിച്ച് തരുന്നു.

ചിത്രം 10.9: ഒരു സംഘടനയിൽ MIS - ന്റെ കർത്തവ്യം

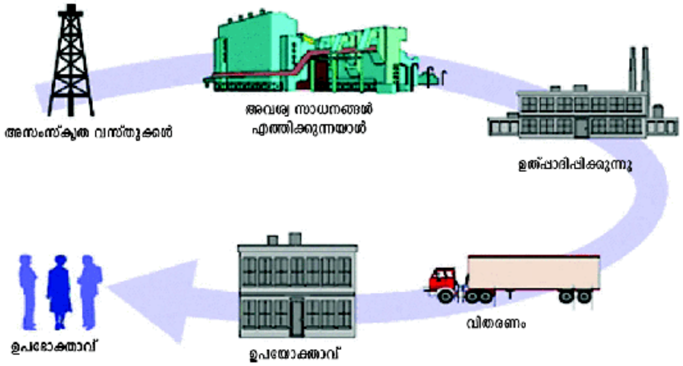
MIS എന്നത് ഒരു സ്ഥാപനത്തിലെ പ്രവർത്തനങ്ങളെ സഹായിക്കുന്നതിനും, നിയന്ത്രിക്കുന്നതിനും തീരുമാനങ്ങൾ എടുക്കുന്നതിനും ആ

വശ്യമായ വിവരം പ്രദാനം ചെയ്യുന്ന, മനുഷ്യനും യന്ത്രവും ചേർന്ന ഒരു ഏകീകൃത സമ്പ്രദായമാണെന്ന് നിർവചിക്കാം.

വിതരണ ശൃഖലയുടെ നിയന്ത്രണം (സപ്ലൈ ചെയിൻ മാനേജ്മെന്റ് – SCM)

വിതരണക്കാരിൽ നിന്നും തുടങ്ങി ഉപഭോക്താവിലേക്ക് വസ്തുക്കൾ സഞ്ചരിക്കുന്നതിനുള്ള എല്ലാ പ്രവർത്തനങ്ങളും അടങ്ങുന്നതാണ് വിതരണ ശൃംഖല. ചിത്രം 10.10-ൽ കാണിച്ചിരിക്കുന്നത് പോലെ അത് അസംസ്കൃത വസ്തുക്കളുടെ ശേഖരണത്തിൽ തുടങ്ങി ഉൽപ്പന്നം ഉപഭോക്താവിന് ലഭ്യമാക്കുന്നതിൽ അവസാനിക്കുന്നു. ഉല്പന്നം ഉപഭോക്താവിന് പെട്ടെന്ന് എത്തിക്കുക എന്നതാണ് കമ്പനികളെ സംബന്ധിച്ചിടത്തോളം വളരെ പ്രാധാന്യം അർഹിക്കുന്നത്. വേഗത്തിലുള്ള ഉല്പന്ന ലഭ്യതയും ഉല്പന്നം എത്തിക്കലും ഉപഭോക്താക്കളുടെ സംതൃപ്തിയും വിൽപ്പനയും വർദ്ധിപ്പിക്കുന്നു. അതുകൊണ്ട് വിതരണ ശൃംഖലയിലെ പ്രവർത്തനങ്ങൾ വേണ്ടവിധം കൈകാര്യം ചെയ്യണം എന്നത് വളരെ പ്രാധാന്യം അർഹിക്കുന്നു. അതിന് വേണ്ട സോഫ്റ്റ്വെയർ പാക്കേജുകൾ വിപണിയിൽ ലഭ്യമാണ്.

വിതരണ ശൃംഖല കൈകാര്യം ചെയ്യുന്നതിനുള്ള ചിട്ടയായ ഒരു സമീപനമാണ് SCM. ചരക്ക് കൈകാര്യം ചെയ്യൽ, ഗതാഗത നിയന്ത്രണം, സംഭരണശാലയുടെ നിയന്ത്രണം, വിതരണ തന്ത്രങ്ങളുടെ ആസൂത്രണം, ഉപഭോക്തൃ സേവന നിലവാരം പരിശോധിക്കൽ, കമ്പ്യൂട്ടർ ഉപയോഗിച്ചുള്ള അനുകരണം തുടങ്ങിയ പ്രവർത്തനങ്ങൾ SCM-ൽ ഉൾപ്പെട്ടിരിക്കുന്നു.



ചിത്രം 10.10:

തീരുമാനങ്ങൾ എടുക്കാൻ സഹായിക്കുന്ന പദ്ധതി (ഡിസിഷൻ സപ്പോർട്ട് സിസ്റ്റം – DSS)

ഉപയോക്താക്കൾക്ക് തീരുമാനം എടുക്കുന്നതിനും, ഉല്പന്നം/സേവനം തിരഞ്ഞെടുക്കുന്നതിനും സഹായകമായ കമ്പ്യൂട്ടർ അധിഷ്ഠിത പരസ്പര വിനിമയ സമ്പ്രദായമാണ് DSS. ഉപയോക്താക്കൾക്ക് വാണിജ്യപരമായ തീരുമാനങ്ങൾ എടുക്കാൻ വളരെ അനായാസം സാധിക്കുന്നവിധത്തിൽ ഡാറ്റയുടെ വിശകലനവും അവതരണവും നൽകുന്ന ഒരു കമ്പ്യൂട്ടർ ആപ്ലിക്കേഷൻ പ്രോഗ്രാം ആണ് ഇത്. വിവിധ തരത്തിലുള്ള വിവരണങ്ങളായി ശരിയായ വിവരം നൽകുന്നതിന് പകരം സാഹചര്യങ്ങൾ വിശകലനം ചെയ്യാൻ സഹായിക്കുന്നതിനാണ് DSS ശ്രദ്ധ നൽകുന്നത്. DSS ഒരു സഹായി മാത്രമാണ്. ഇപ്പോഴും തീരുമാനങ്ങൾ എടുക്കുന്നതിൽ മനുഷ്യൻ അവന്റെ മേൽക്കോയ്മ നിലനിർത്തുന്നു. തീരുമാനങ്ങൾ എടുക്കാൻ സഹായിക്കുന്നതിന് വളരെ ഫലപ്രദമായ ഒരു ഡാറ്റാ ബേസ് നിയന്ത്രണ സംവിധാനം DSS ന് ആവശ്യമാണ്. DSS-ന്റെ പ്രവർത്തനങ്ങൾ ചിത്രം 10.11-ൽ കാണിച്ചിരിക്കുന്നു.

ചിത്രം 10.11: DSS - ന്റെ പ്രവർത്തനങ്ങൾ

നമുക്ക് സംഗ്രഹിക്കാം

വാണിജ്യമേഖലയിൽ വളർന്ന് വരുന്ന പ്രവണതകൾ ഉല്പാദനമികവിനും, സാമ്പത്തിക നേട്ടത്തിനും, ലാഭമുണ്ടാക്കുന്നതിനും സഹായിക്കുന്നത് എങ്ങനെ എന്ന് ഈ അധ്യായത്തിൽ നാം മനസ്സിലാക്കി. സംരംഭത്തിലെ വിഭവങ്ങളുടെ ആസൂത്രണത്തെക്കുറിച്ച് അതിന്റെ പ്രധാന പ്രാവർത്തിക ഘടകങ്ങൾക്ക് അടിസ്ഥാന വിവരം പകർന്ന് നൽകി. ERP നടപ്പാക്കുന്നതിൽ വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങ്ങിന്റെ പ്രാധാന്യം എടുത്ത് പറഞ്ഞു. ERP നടപ്പാക്കുന്നതിലെ വിവിധ ഘട്ടങ്ങൾ ചുരുക്കി പ്രതിപാദിച്ചു. വിപണിയിൽ ലഭ്യമായതും വളരെ സാധാരണയായി ഉപയോഗിക്കാറുള്ളതുമായ ERP പാക്കേജുകൾ പരിചയപ്പെടുത്തി. ERP നടപ്പാക്കുന്നതു കൊണ്ടുള്ള നേട്ടങ്ങളും പരിമിധികളും പട്ടികപ്പെടുത്തി. ERP - യുമായി ബന്ധപ്പെട്ട ചില പ്രധാനപ്പെട്ട സാങ്കേതികവിദ്യകൾ അധ്യായത്തിന്റെ അവസാനഭാഗത്ത് ചുരുക്കി നൽകിയിട്ടുണ്ട്.

നമുക്ക് വിലയിരുത്താം

1. DSS-ന്റെ പൂർണ്ണ രൂപം .... ആണ്.
   a. ഡിസിഷൻ സിഗ്നൽ സിസ്റ്റം b. ഡിസൈൻ സപ്പോർട്ട് സിസ്റ്റം
   c. ഡിസിഷൻ സപ്പോർട്ട് സ്കീം d. ഡിസിഷൻ സപ്പോർട്ട് സിസ്റ്റം
2. ഒറ്റപ്പെട്ടത് കണ്ടെത്തുകയും അതിനുള്ള ന്യായീകരണം നൽകുകയും ചെയ്യുക. SAP, ഓറക്കിൾ, C++, ഉഡു
3. SAP യുടെ പൂർണ്ണ രൂപം .... ആണ്.
4. സംരംഭ വിഭവ ആസൂത്രണം (ERP) നിർവ്വചിക്കുക.
5. ERP പാക്കേജിൽ HR ഘടകത്തിന്റെ പ്രാധാന്യം എന്ത്?
6. വാണിജ്യ പ്രവർത്തനങ്ങളുടെ റീഎൻജിനീയറിങ്ങും (BPR) സംരംഭ വിഭവങ്ങളുടെ ആസൂത്രണവും (ERP) തമ്മിലുള്ള ബന്ധം വിശദമാക്കുക.
7. ഏതെങ്കിലും രണ്ട് ERP പരിഹാരം നൽകുന്നവരെക്കുറിച്ച് ചുരുക്കി എഴുതുക.
8. ഒരു സംരംഭത്തിൽ ERP നടപ്പിലാക്കുന്നതിന്റെ നേട്ടങ്ങൾ പട്ടികപ്പെടുത്തുക.
9. ERP അനുബന്ധ സാങ്കേതികവിദ്യകളിൽ ഏതെങ്കിലും ഒന്നിനെക്കുറിച്ച് ചുരുക്കി എഴുതുക.
10. ERP നടപ്പിലാക്കുന്നതിലെ വിവിധ ഘട്ടങ്ങൾ ചുരുക്കി വിശദീകരിക്കുക.

11 വിവരവിനിമയ സാങ്കേതികവിദ്യയിലെ പ്രവണതകളും പ്രശ്നങ്ങളും

പ്രധാന പഠനനേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പഠനം പൂർത്തിയാകുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ

- വിവിധ മൊബൈൽ കമ്പ്യൂട്ടിങ് സാങ്കേതികപദങ്ങൾ തിരിച്ചറിയുന്നു.
- മൊബൈൽ ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിന്റെ സവിശേഷതകൾ തിരിച്ചറിയുന്നു.
- ആൻഡ്രോയിഡ് ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിന്റെ സവിശേഷതകൾ കണ്ടെത്തുന്നു.
- വിവിധ ബൗദ്ധികസ്വത്തവകാശങ്ങൾ പട്ടികപ്പെടുത്തുകയും വിശദീകരിക്കുകയും ചെയ്യുന്നു.
- സൈബർ ഇടം വിശദീകരിക്കുന്നു.
- വിവിധതരം സൈബർ കുറ്റകൃത്യങ്ങൾ വേർതിരിച്ചറിയുന്നു.
- സൈബർ നിയമങ്ങളും ധാർമികമൂല്യങ്ങളും വിശദമാക്കുന്നു.
- ഐടി നിയമത്തിന്റെ പ്രാധാന്യം തിരിച്ചറിയുന്നു.
- ഇൻഫോമാനിയ തിരിച്ചറിയുന്നു.

Q1C1W7

ഇന്റർനെറ്റിന്റെ ഉപയോഗം കഴിഞ്ഞ കുറെ ദശാബ്ദങ്ങളായി ത്വരിതഗതിയിൽ വർധിച്ചുകൊണ്ടിരിക്കുന്നു. ഇന്റർനെറ്റും അതു നൽകുന്ന സേവനങ്ങളും ഇല്ലാത്ത ഒരു ദിവസത്തെപ്പറ്റി ആധുനിക സമൂഹത്തിന് ചിന്തിക്കാൻ പോലും കഴിയില്ല. കമ്പ്യൂട്ടർ സാങ്കേതികവിദ്യയുടെ വികാസ പരിണാമങ്ങൾ കൂടുതൽ ശക്തവും എന്നാൽ വില കുറഞ്ഞതുമായ മൊബൈൽ ഉപകരണങ്ങളുടെ ആവിർഭാവത്തിലേക്ക് നയിച്ചു. മൊബൈൽ ഫോണുകൾ, ടാബ്‌ലെറ്റുകൾ തുടങ്ങിയ ഇത്തരം ഉപകരണങ്ങൾ ഉപയോഗിച്ച് ആളുകൾ ഇന്റർനെറ്റിലെ സേവനങ്ങൾ ഉപയോഗപ്പെടുത്തുവാൻ തുടങ്ങി.

വിവരസാങ്കേതികവിദ്യയുടെ കുറേക്കൂടി വിശാലമായ അർഥതലമുള്ള ഒരു പദസമുച്ചയമായിട്ടാണ് 'വിവരവിനിമയ സാങ്കേതികവിദ്യ' പലപ്പോഴും ഉപയോഗിക്കുന്നത്. വാർത്താവിനിമയ സംവിധാനങ്ങളേയും കമ്പ്യൂട്ടറുകളേയും സംയോജിപ്പിക്കുന്നതിനാണ് വിവരവിനിമയ സാങ്കേതികവിദ്യ കൂടുതൽ ഊന്നൽ നൽകുന്നത്. മൊബൈൽ ആശയവിനിമയ രംഗത്തെ സേവനങ്ങളും വിവിധ സാങ്കേതികവിദ്യകളുമാണ് ഈ അധ്യായത്തിൽ നാം ചർച്ച ചെയ്യുന്നത്. ആൻഡ്രോയ്ഡ് ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിന് പ്രത്യേക ഊന്നൽ നൽകിക്കൊണ്ട് ഇന്ന് ലഭ്യമായിട്ടുള്ള വിവിധ മൊബൈൽ ഓപ്പറേറ്റിംഗ് സിസ്റ്റങ്ങളെക്കുറിച്ചും നാം ചർച്ച ചെയ്യും.

ഭൂമി, കെട്ടിടം തുടങ്ങിയ സ്ഥാവരജംഗമ വസ്തുക്കളുടെ മേൽ അവകാശമുള്ളതുപോലെ ബൗദ്ധിക ഗുണവിശേഷങ്ങളായ സംഗീതം, സിനിമ, സോഫ്റ്റ്‌വെയർ, രൂപകല്പനകൾ തുടങ്ങിയവയ്ക്കും ഉടമ

സ്വാവകാശം ഉണ്ട്. ബൗദ്ധികസ്വത്തവകാശം എന്നറിയപ്പെടുന്ന അവകാശങ്ങളും അവയുമായി ബന്ധപ്പെട്ട പ്രശ്നങ്ങളും ഈ അധ്യായത്തിൽ നാം ചർച്ച ചെയ്യുന്നു. എല്ലാ സാങ്കേതികവിദ്യകൾക്കും ഒരു ഇരുണ്ട വശം ഉണ്ടെന്നുള്ളതും യാഥാർഥ്യമാണ്. പ്രശ്നങ്ങളുടെയും ഭീഷണികളുടെയും കാര്യത്തിൽ ഇന്റർനെറ്റിനും അതിന്റേതായ ഒരു പങ്കുണ്ട്. ചില ആളുകൾ ഈ മാധ്യമം നിയമ വിരുദ്ധ പ്രവർത്തനങ്ങൾക്കായി ഉപയോഗിക്കാറുണ്ട്. ഇൻറർനെറ്റിലൂടെയുള്ള ഇത്തരം പ്രവർത്തനങ്ങളും, അവ എങ്ങനെ സ്വയം പ്രതിരോധിക്കാം എന്നുള്ളതും നാം ചർച്ച ചെയ്യും.

11.1 മൊബൈൽ കമ്പ്യൂട്ടിങ് (Mobile computing)

കമ്പ്യൂട്ടർ സാങ്കേതിക വിദ്യയിലെ മുന്നേറ്റത്തിലൂടെ ചെറിയ ഉപകരണങ്ങളുടെ കമ്പ്യൂട്ടിങ് ശേഷി പതിൻമടങ്ങ് വർധിച്ചു. വിലക്കുറവും വൈദ്യുതി ഉപഭോഗം കുറവുള്ളതും ഭാരം കുറഞ്ഞതുമായ കമ്പ്യൂട്ടിങ് ഉപകരണങ്ങൾ ഇന്ന് സുലഭമാണ്. ലാപ്ടോപ്പ്, ടാബ്ലെറ്റ്, സ്മാർട്ട് ഫോൺ തുടങ്ങിയ ഉപകരണങ്ങൾ ആളുകളുടെ തൊഴിൽ സംസ്കാരത്തെയും ജീവിത രീതിയെയും അങ്ങേയറ്റം മാറ്റി മറിച്ചു. ഇന്ന് ആളുകൾക്ക് എപ്പോഴും എവിടെ നിന്നും പരസ്പരം ബന്ധപ്പെടുകയും ഫയലുകളോ മറ്റ് വിവരങ്ങളോ സ്വീകരിക്കുകയും അയയ്ക്കുകയും ചെയ്യാവുന്നതിനാൽ കമ്പ്യൂട്ടിങ്ങിന്റെ ആവശ്യകത നാൾക്കുനാൾ വർധിച്ചു.

കമ്പ്യൂട്ടിങ് ശേഷിയും സഞ്ചരിക്കുമ്പോൾ പോലും ഡാറ്റ സംപ്രേഷണവും സ്വീകരണവും സാധിക്കുന്ന സാങ്കേതിക വിദ്യയാണ് മൊബൈൽ കമ്പ്യൂട്ടിങ്ങ്.

മൊബൈൽ കമ്പ്യൂട്ടിങ്ങിന് ലാപ്ടോപ്പ്, ടാബ്ലെറ്റ്, സ്മാർട്ട് ഫോൺ തുടങ്ങിയ കൊണ്ടുനടക്കാൻ പറ്റിയ ഉപകരണങ്ങളും, വയർലെസ് വാർത്താ വിനിമയ ശൃംഖലയും ഇന്റർനെറ്റ് ബന്ധവും ആവശ്യമാണ്. മൊബൈൽ കമ്പ്യൂട്ടിങ്ങിന്റെ ആവശ്യകത മൊബൈൽ വാർത്താവിനിമയ സാങ്കേതികവിദ്യയുടെ വളർച്ചക്കും വികാസത്തിനും തുടക്കമിട്ടു.

11.2 മൊബൈൽ ആശയ വിനിമയം (Mobile communication)

മൊബൈൽ എന്ന പദം പൂർണമായും വിപ്ലവൽക്കരിച്ച് ആശയവിനിമയത്തിലൂടെ വിവരങ്ങൾ കൈമാറുന്നതിനുള്ള നൂതന മാർഗ്ഗങ്ങൾ തുറന്നിട്ടു. ഇന്ന് മൊബൈൽ സാങ്കേതിക വിദ്യ നമ്മുടെ ജീവിത സാഹചര്യങ്ങൾ മെച്ചപ്പെടുത്തി സമൂഹത്തിന്റെ നട്ടെല്ലായി മാറിയിരിക്കുന്നു. മൊബൈൽ ആശയ വിനിമയ ശൃംഖലയ്ക്ക് രണ്ട് ഉപകരണങ്ങൾ തമ്മിൽ ആശയ വിനിമയം നടത്തുവാൻ ഭൗതിക മാധ്യമമാവശ്യമില്ല.

11.2.1 മൊബൈൽ ആശയവിനിമയത്തിലെ തലമുറകൾ (Generations in mobile communication)

1964-ൽ മൊബൈൽ ഫോൺ അവതരിക്കപ്പെട്ടു. ആദ്യ ഘട്ടത്തിൽ മൊബൈൽ ആശയവിനിമയത്തിന്റെ വളർച്ച തുലോം മന്ദഗതിയിലായിരുന്നു. മൊബൈൽ ഉപഭോക്താക്കളുടെ എണ്ണം വർധിക്കുന്നതിനനുസരിച്ച് കുറഞ്ഞ അളവിൽ മാത്രം ലഭ്യമായിരുന്ന ആവൃത്തി സ്പെക്ട്രം (ഫ്രീക്വൻസി സ്പെക്ട്രം) വലിയ പ്രശ്നമായി മാറി. ഇത് പരിഹരിക്കുവാൻ സെല്ലുലാർ ആശയവിനിമയം എന്ന ആശയം ഉരുത്തിരിഞ്ഞു. ഈ ആശയം 1960 കളിൽ ബെൽ ലാബോറട്ടറീസിലാണ് വികസിപ്പിച്ചത്. എന്നിരുന്നാലും 1990 കളുടെ അവസാനം ഇന്ത്യ ഗവൺമെന്റ് മൊബൈൽ ആശയവിനിമയത്തിന് സ്പെക്ട്രം അനുമതി വാഗ്ദാനം ചെയ്തു. ഇതോടെ നമ്മുടെ രാജ്യത്തിൽ സെല്ലുലാർ സാങ്കേതികവിദ്യ സാധാരണ നിലവാരത്തിലായി. മൊബൈൽ ആശയവിനിമയത്തിന്റെ വിവിധ തലമുറകൾ താഴെ കൊടുത്തിരിക്കുന്നു.

a. ഒന്നാം തലമുറ ശൃംഖല (1G)

ഏകദേശം 1980 കളിൽ വികസിപ്പിച്ച വയർലെസ് ടെലിഫോൺ സാങ്കേതികവിദ്യയാണ് 1Gഎന്നത് കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. അനലോഗ് തരംഗങ്ങളെ അടിസ്ഥാനമാക്കിയുള്ള 1G മൊബൈൽ ഫോണുകളിൽ ശബ്ദ സൗകര്യം മാത്രമേ ഉണ്ടായിരുന്നുള്ളൂ.

b. രണ്ടാം തലമുറ ശൃംഖല (2G)

ഇതിൽ ആശയവിനിമയത്തിന് ഡിജിറ്റൽ സംവിധാനം ഉപയോഗിച്ചിരുന്നതിനാൽ ശബ്ദ നിലവാരം വർധിച്ചു. ബൈനറി രൂപത്തിൽ രഹസ്യ കോഡുകൾ ഉപയോഗിച്ചായിരുന്നു 2G ശൃംഖലയിലെ ഫോൺ സംഭാഷണങ്ങൾ. ഇത്തരം ശൃംഖലകളിലെ ഫോണുകൾക്ക് വിനിമയ പരിധി വളരെ കൂടുതലായിരുന്നു. ഇതിലൂടെ ഡാറ്റാ സേവനങ്ങളും ചിത്രങ്ങൾ അടങ്ങിയ സന്ദേശങ്ങളും മൾട്ടി മീഡിയ മെസേജ് സർവീസും (MMS) അവതരിപ്പിച്ചു. GSM ഉം CDMA യുമാണ് 2G ശൃംഖലയിലൂടെ അവതരിപ്പിച്ച രണ്ട് തരം ജനപ്രിയ സ്റ്റാൻഡേർഡുകൾ. ഇവയുടെ വിശദാംശങ്ങൾ ചുവടെ ചേർക്കുന്നു.

(i) ഗ്ലോബൽ സിസ്റ്റം ഫോർ മൊബൈൽ (GSM)

ജി.എസ്.എം എന്നത് ആഗോള തലത്തിൽ സ്വീകരിക്കപ്പെട്ട ഡിജിറ്റൽ സെല്ലുലാർ ആശയവിനിമയത്തിനുള്ള നിലവാരമാണ്. ഒരു റേഡിയോ ഫ്രീക്വൻസിയിൽ ഒരേ സമയം ഒന്നിലധികം സംഭാഷണങ്ങൾ അനുവദിക്കുന്ന ക്ലിപ്ത ബാൻഡ് ടി.ഡി.എം.എ (ടൈം ഡിവിഷൻ മൾട്ടിപ്പിൾ ആക്സസ്) ആണ് ജി.എസ്.എം ൽ ഉപയോഗിക്കുന്നത്. ഇത് ശബ്ദ സംഭാഷണത്തിന് വേണ്ടിയുള്ള ഡിജിറ്റൽ, സർക്യൂട്ട് - സ്വിച്ച്ഡ് ശൃംഖലയാണ്. ഇതിന്റെ ആവൃത്തി ബാന്റ് 900 MHz മുതൽ 1800 MHz വരെയാണ്. ജി.എസ്.എമ്മിന് (GSM) പൊതുവായ അന്താരാഷ്ട്ര നിലവാരമുള്ളതിനാൽ മൊബൈൽ ഫോണുകൾ ലോകത്തെവിടെയും ഉപയോഗിക്കുവാൻ സാധിക്കും. SIM (Subscribe Identity Module) ഉപയോഗിച്ചാണ് ശൃംഖല തിരിച്ചറിയുന്നത്. ഉപയോക്താക്കൾക്ക് സ്വന്തം ഇഷ്ടാനുസരണം ഹാൻഡ് സെറ്റുകൾ തിരഞ്ഞെടുക്കാം. GSM സെല്ലുലാർ നിലവാരത്തിലെ ഏറ്റവും വിജയപ്രദമായ കുടുംബമാകുന്നു.

GPRS ഉം EDGE ഉം

മെച്ചപ്പെട്ട ഡാറ്റ ആശയവിനിമയ സൗകര്യങ്ങൾ ഉൾക്കൊള്ളിക്കുവാൻ GPRS (ജനറൽ പാക്കറ്റ് റേഡിയോ സർവീസ്) ഉം EDGE (എൻഹാൻസ്ഡ് ഡാറ്റ റേറ്റ്സ് ഫോർ ജി.എസ്.എം ഇവല്യൂഷൻ) ഉം 2G ശൃംഖല വികസിപ്പിച്ചു.

GSM ൽ പാക്കറ്റായി ക്രമീകരിക്കപ്പെട്ട മൊബൈൽ ഡാറ്റ സേവനമാണ് GPRS. പരമ്പരാഗത GSM മായി താരതമ്യം ചെയ്യുമ്പോൾ GPRS ഉപയോക്താക്കൾക്ക് കുറഞ്ഞ സമയത്തിനുള്ളിൽ കൂടുതൽ ഡാറ്റ ലഭ്യമായിരുന്നു. ഡാറ്റ കൈമാറ്റത്തിന്റെ അളവിലായിരുന്നു GPRS ൽ ബില്ലുകൾ തയാറാക്കുന്നത്. GPRS ഒരു ഡാറ്റ സാങ്കേതികവിദ്യ മാത്രമായിരുന്നെങ്കിലും ശബ്ദം മെച്ചപ്പെടുത്തുവാനും ഇത് സഹായിച്ചു.

ഒരു ഡിജിറ്റൽ മൊബൈൽ ഫോൺ സാങ്കേതികവിദ്യയായ EDGE ലൂടെ GSM ന് മെച്ചപ്പെട്ട ഡാറ്റ സംപ്രേഷണ നിരക്ക് ലഭ്യമാക്കി. EDGE എന്നത് ജി.പി.ആർ.എസിന്റെ ഒരു അതിഗണമാണ്. ജി.പി.ആർ.എസ്. വിന്യസിച്ചിട്ടുള്ള ഏത് ശൃംഖലയിലും അതിനോട് കൂടി പ്രവർത്തിക്കാൻ

സാധിക്കും. ഇത് GPRS രീതിയെ അപേക്ഷിച്ച് ഏതാണ്ട് മൂന്നിരട്ടി വേഗത നൽകുന്നു. ഇതിനായി ഫോണും ശൃംഖലയും EDGE സാങ്കേതികവിദ്യ പിന്തുണയ്ക്കേണ്ടതുണ്ട്, അല്ലാത്തപക്ഷം ഫോൺ GPRS സംവിധാനത്തിലേക്ക് തിരിച്ചു പോകും.

(ii) കോഡ് ഡിവിഷൻ മൾട്ടിപ്പിൾ ആക്സസ് (CDMA)

കോഡ് ഡിവിഷൻ മൾട്ടിപ്പിൾ ആക്സസ് സിസ്റ്റത്തിൽ ഒരു ആശയവിനിമയ മാധ്യമത്തിലൂടെ ഒരേ സമയം വിവിധ സംപ്രേഷകർക്ക് വിവരങ്ങൾ അയയ്ക്കാം. ഇത് GSM നെ അപേക്ഷിച്ച് കൂടുതൽ സേവനപരിധി നൽകുകയും ദുർബല സിഗ്നൽ പോലും ഭേദപ്പെട്ട നിലയിൽ സ്വീകരിക്കുകയും ചെയ്യുന്നു. സി.ഡി.എം.എയിലെ ശബ്ദ ഗുണനിലവാരം ജി.എസ്.എം.നേക്കാൾ മെച്ചപ്പെട്ടതാണ്. ഇതിലെ സിഗ്നലുകൾക്ക് കൂടുതൽ ബാൻഡ് വിഡ്ത്തും തടസ്സങ്ങൾ പ്രതിരോധിക്കാനുള്ള ശേഷിയുണ്ട്. ഈ സാങ്കേതികവിദ്യ GSM നെ അപേക്ഷിച്ച് മൊബൈൽ ശൃംഖലയ്ക്ക് കൂടുതൽ സുരക്ഷ നൽകുന്നു.

c. മൂന്നാം തലമുറ ശൃംഖല (3G)

കൈയിൽ കൊണ്ടു നടക്കാവുന്ന ഉപകരണങ്ങൾക്ക് ഉയർന്ന ഡാറ്റാ കൈമാറ്റ നിരക്ക് നൽകുന്ന വയർലെസ് സാങ്കേതിക വിദ്യയാണ് 3G. ഉയർന്ന ഡാറ്റ സംപ്രേഷണ നിരക്ക് ലഭ്യമായതിനാൽ ഡാറ്റയും, ശബ്ദവും കൂടിച്ചേർന്ന മൾട്ടി മീഡിയ സേവനങ്ങൾ 3G ശൃംഖല വാഗ്ദാനം ചെയ്യുന്നു. വയർലെസ് ബ്രോഡ്ബാന്റ് എന്നും അറിയപ്പെടുന്ന 3G യിൽ മൊബൈൽ ഫോണിലൂടെ വലിയ അളവിൽ ഡാറ്റ അയയ്ക്കുവാനും സ്വീകരിക്കുവാനുമുള്ള സൗകര്യമുണ്ട്. 3G ശൃംഖലകളിൽ ആക്സസ് ഭാഗത്തിനായി WCDMA (വൈഡ് ബാന്റ് കോഡ് ഡിവിഷൻ മൾട്ടിപ്പിൾ ആക്സസ്) ഉപയോഗിക്കുന്നു. ഇതിന് മൊബൈൽ ടവറുകളും (ബേസ് സ്റ്റേഷൻ) ഫോണുകളും നവീകരിക്കണം. അതോടൊപ്പം മൊബൈൽ ടവറുകൾ വളരെ അടുത്തടുത്തായി സ്ഥാപിക്കുകയും വേണം.

d. നാലാം തലമുറ ശൃംഖല (4G)

ലോങ് ടേം ഇവല്യൂഷൻ (LTE) എന്ന പേരിൽ കൂടി അറിയപ്പെടുന്ന 4G സംവിധാനം മൊബൈൽ ഉപകരണങ്ങൾക്ക് അൾട്രാ ബ്രോഡ്ബാന്റ് ഇന്റർനെറ്റ് സേവനം നൽകുന്നു. ഉയർന്ന ബാന്റ് വിഡ്ത്ത് ആവശ്യമുള്ള വീഡിയോ സ്ട്രീമിങ് പ്രവർത്തനങ്ങൾക്ക് 4G ഉയർന്ന വേഗവും മെച്ചപ്പെട്ട പ്രവർത്തന മികവും നൽകുന്നു. വയർലെസ് ഐ.പി അടിസ്ഥാനമാക്കിയുള്ള പ്രവേശന മാർഗം ഇതിന് അത്യാവശ്യമാണ്. 4G ശൃംഖലകളിൽ ആക്സസ് ഭാഗത്തിനായി OFDMA (ഓർത്തോഗണൽ ഫ്രീക്വൻസി ഡിവിഷൻ മൾട്ടിപ്പിൾ ആക്സസ്) ഉപയോഗിക്കുന്നു. ടെലിവിഷനെക്കാൾ മെച്ചപ്പെട്ട ചിത്രങ്ങളും വീഡിയോകളും 4G നൽകുന്നു.

e. അഞ്ചാം തലമുറ ശൃംഖല (5G)

ഉയർന്ന ബാൻഡ്‌വിഡ്ത്തും, ഡാറ്റ കൈമാറ്റ നിരക്കും അതോടൊപ്പം തന്നെ കുറഞ്ഞ പ്രസരണ കാലതാമസവും (transmission delay) വാഗ്ദാനം ചെയ്യുന്ന അഞ്ചാം തലമുറ ശൃംഖലകൾ (5G) ടെലികമ്മ്യൂണിക്കേഷൻ വ്യവസായത്തിൽ അതിശയകരമായ വിപ്ലവങ്ങൾ സൃഷ്ടിച്ചേക്കാം. അഞ്ചാംതലമുറ ശൃംഖലയിലൂടെ (5G) നൂതന സാങ്കേതികവിദ്യകളായ വീഡിയോ സ്ട്രീമിങ്, ഡ്രൈവർ ഇല്ലാത്ത കാർ, ഇന്റർനെറ്റ് ഓഫ് തിങ്സ് (IoT) എന്നിവ സാധ്യമാകും എന്നാണ് കരുതപ്പെടുന്നത്. നിലവിലുള്ള 4G ശൃംഖലകളെക്കാൾ (ലേറ്റൻസി സമയം 25ms) പതിന്മടങ്ങ് വേഗത നൽകുന്ന അഞ്ചാംതലമുറ ശൃംഖലയിൽ ലേറ്റൻസി സമയം ഒരു മില്ലീ സെക്കന്റായി നിജപ്പെടുത്തിയിട്ടുണ്ട്. ഒരു ഫോർവേർഡിങ് പോയിന്റിൽ നിന്ന് മറ്റൊരു പോയിന്റിലേക്ക് ഒരു പാക്കറ്റ് ഡാറ്റ സഞ്ചരിക്കാനാവശ്യമായ സമയത്തെയാണ് ലേറ്റൻസി സമയം എന്ന് പറയുന്നത്.

11.2.2 മൊബൈൽ വാർത്താ വിനിമയ സേവനങ്ങൾ (Mobile communication services)

മൊബൈൽ വാർത്താവിനിമയ വ്യവസായത്തിലെ വിവിധ സാങ്കേതികവിദ്യകൾക്ക് ധാരാളം സംക്ഷേപങ്ങൾ ഉപയോഗിക്കുന്നു. ഇവിടെ നമുക്ക് ജനപ്രിയമായ കുറച്ച് മൊബൈൽ വാർത്താ വിനിമയ സങ്കേതികവിദ്യകളായ SMS, MMS, GPS, Smart Card എന്നിവയെക്കുറിച്ച് ചർച്ച ചെയ്യാം.

a. ഷോർട്ട് മെസേജ് സർവീസ് (SMS)

മൊബൈൽ വാർത്താവിനിമയ വ്യവസ്ഥയിൽ ഹ്രസ്വ വാചക സന്ദേശങ്ങൾ പരസ്പരം കൈമാറുന്ന സേവനമാണ് ഷോർട്ട് മെസേജ് സർവ്വീസ്. മൊബൈൽ ഫോൺ ഉപയോക്താക്കൾ വൻതോതിൽ ഉപയോഗിക്കുന്ന ഡാറ്റ സേവനമാണ് എസ്.എം.എസ്. പരമാവധി 160 അക്ഷരങ്ങളോ സംഖ്യകളോ അയയ്ക്കുവാനുള്ള സൗകര്യം ജി.എസ്.എം നൽകുന്നു. ഒരു സന്ദേശം അയയ്ക്കുമ്പോൾ അത് ഒരു ഷോർട്ട് മെസ്സേജ് സർവീസ് കേന്ദ്രത്തിൽ (SMSC) എത്തിയതിനു ശേഷം അവിടെ സംഭരിക്കുകയും പിന്നീട് എസ്.എം.എസ്. സ്വീകർത്താവിന് സന്ദേശം അയയ്ക്കുകയും ചെയ്യുന്ന രീതിയാണ് അനുവർത്തിച്ചു വരുന്നത്. ഒരു സ്വീകർത്താവിന് സന്ദേശം അയയ്ക്കാൻ SMSC ശ്രമിക്കുമ്പോൾ സ്വീകർത്താവിനെ ലഭ്യമാകാത്ത സാഹചര്യത്തിൽ കാത്തു നിൽക്കുകയും വീണ്ടും ശ്രമിക്കുകയും ചെയ്യും. എന്നാൽ ചില SMSCകൾ സ്വീകർത്താവിനെ ലഭിക്കാത്ത സാഹചര്യത്തിൽ ഒരു പ്രാവശ്യത്തെ ശ്രമത്തിനു ശേഷം നീക്കം ചെയ്യും. SS7 (സിഗ്നലിങ് സിസ്റ്റം നമ്പർ #7) എന്ന പ്രോട്ടോകോൾ ഉപയോഗിച്ചാണ് SMSC സന്ദേശങ്ങൾ കൈമാറുന്നത്.

ആദ്യത്തെ എസ്.എം.എസ്

ആദ്യ എസ്.എം.എസ് സന്ദേശമായ "മെറി ക്രിസ്തുമസ് " അയച്ചത് 1992 ഡിസംബർ മൂന്നിന് ഒരു പേഴ്സണൽ കമ്പ്യൂട്ടറിൽ നിന്നും ബ്രിട്ടണിലെ വൊഡാഫോണിന്റെ ശൃംഖലയിലെ സെൽ ഫോണിലേക്കാണ്.

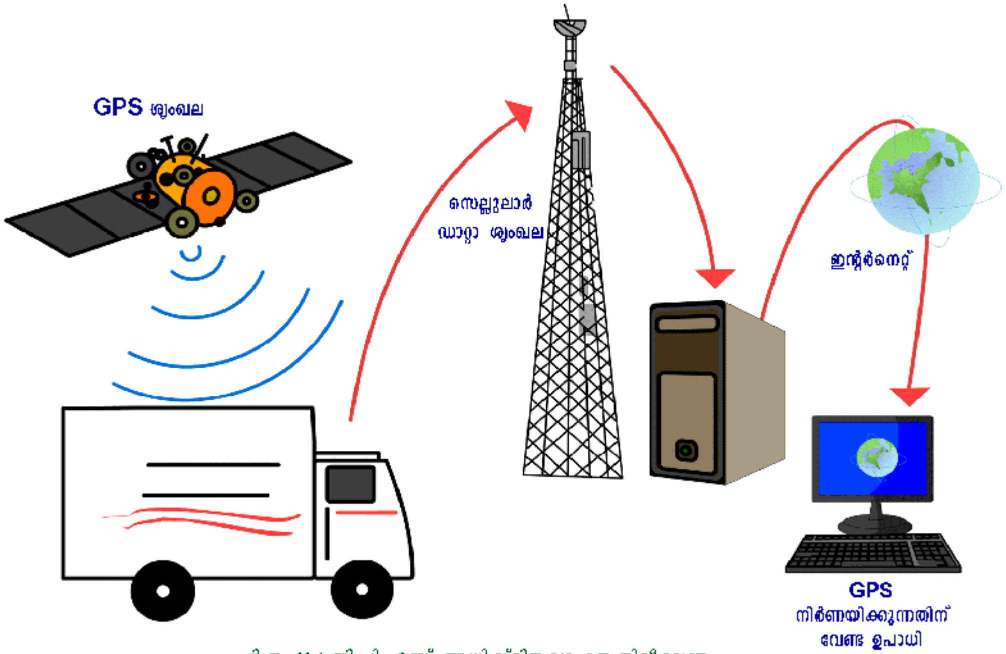
b. മൾട്ടിമീഡിയ മെസേജിങ് സർവീസ് (MMS)

മൾട്ടി മീഡിയ സന്ദേശങ്ങൾ മൊബൈൽ ഫോണിലൂടെ അയയ്ക്കുന്നതിനും സ്വീകരിക്കുന്നതിനുമുള്ള മാർഗമാണ് എം.എം.എസ്. SMS ന്റെ വിപുലീകരിച്ച രൂപമാണ് അത്. എസ്.എം.എസിനെ അപേക്ഷിച്ച് എം.എം.എസ് സന്ദേശങ്ങൾക്ക് വലുപ്പ പരിധി പരാമർശിക്കുന്നില്ല. ഇതിൽ അക്ഷരങ്ങൾ, ഗ്രാഫിക്സ്, പാട്ടുകൾ, വീഡിയോ ശകലങ്ങൾ മുതലായവ ഉൾപ്പെടുത്തുവാൻ സാധിക്കും. വരുന്നതും പോകുന്നതുമായ സന്ദേശങ്ങൾ സംഭരിക്കുന്നതിനും കൈകാര്യം ചെയ്യുന്നതിനും ഒരു MMS സെർവർ ഉണ്ടാകും. വിവിധ സന്ദേശ സംവിധാനങ്ങൾ തമ്മിൽ സന്ദേശങ്ങൾ കൈമാറ്റം ചെയ്യുന്നതിന് ഈ സെർവറുമായി ബന്ധമുള്ള ഒരു MMS പ്രോക്സി റിലേയുമുണ്ടാകും.

c. ഗ്ലോബൽ പൊസിഷനിങ് സിസ്റ്റം (GPS)

ഉപഗ്രഹത്തിന്റെ സഹായത്തോടെ ഭൂമിയിലെ ഏത് സ്ഥലവും രേഖാംശവും അക്ഷാംശവും ഉപയോഗിച്ച് കണ്ടെത്തുന്ന ഗതി നിയന്ത്രണ രീതിയാണ് ജി.പി.എസ്. ഉപഗ്രഹങ്ങൾ, നിരീക്ഷണ നിയന്ത്രണ നിലയങ്ങൾ, സ്വീകർത്താക്കൾ എന്നിവ അടങ്ങുന്ന ജി.പി.എസ് അമേരിക്കൻ പ്രതിരോധ വകുപ്പാണ് രൂപീകരിച്ചതും പ്രവർത്തിപ്പിക്കുന്നതും. ഭൂമിയെ തുടർച്ചയായി വലം വയ്ക്കുന്ന ഉപഗ്രഹക്കൂട്ടങ്ങളാണ് ജി.പി.എസിന്റെ അടിസ്ഥാനം. ഉപഗ്രഹങ്ങളുടെ കൃത്യമായ സ്ഥാനം, സമയം, മറ്റ് വിവരങ്ങൾ എന്നിവ അടങ്ങുന്ന റേഡിയോ വികിരണങ്ങൾ ഇവ സംപ്രേഷണം ചെയ്യുന്നു. നിയന്ത്രണ നിലയങ്ങൾ നിരീക്ഷിക്കുകയും ശരിയാക്കുകയും ചെയ്യുന്ന ഈ

റേഡിയോ വികിരണങ്ങളെ ജി.പി.എസ് റിസീവറുകളിലൂടെ പിടിച്ചെടുക്കുന്നു. ഉപഗ്രഹത്തിൽ നിന്നും ജി.പി.എസ് റിസീവർ വഴി പിടിച്ചെടുക്കുന്ന വിവരങ്ങൾ ഉപയോഗിച്ച് ഉപയോക്താവിന്റെ കൃത്യമായ സ്ഥാനം കണക്കാക്കുന്നു. ജി.പി.എസ് റിസീവർ ഒരു സ്ഥലത്തിന്റെ ദ്വിമാന ചിത്രത്തിനായി മൂന്ന് ഉപഗ്രഹങ്ങളും ത്രിമാനചിത്രങ്ങൾക്കായി നാലോ അതിലധികമോ ഉപഗ്രഹങ്ങളും ഉപയോഗിക്കുന്നു. ത്രിമാന ചിത്രങ്ങൾ ദ്വിമാന ചിത്രങ്ങളെക്കാൾ കൃത്യതയുള്ളതാണ്.



ചിത്രം 11.1: ജി.പി.എസ്. അധിഷ്ഠിത വാഹന നിരീക്ഷണം.

ജി.പി.എസ് ഉപയോഗിച്ച് ചരക്ക് വാഹന കമ്പനികൾക്ക് അവരുടെ വാഹനങ്ങളുടെ നീക്കങ്ങൾ കൃത്യമായി നിർണയിക്കാനാവും. ഇതിന്റെ ചിത്രം 11.1 ൽ ചേർക്കുന്നു. വാഹനത്തിലെ ഗതി നിയന്ത്രണ ഉപകരണങ്ങൾ ഡ്രൈവർക്ക് ഏറ്റവും നല്ല വഴിയെക്കുറിച്ചുള്ള വിവരം നൽകുന്നു. വാണിജ്യാടിസ്ഥാനത്തിൽ വിമാന ഗതി നിയന്ത്രണം, എണ്ണ പര്യവേക്ഷണം, കൃഷി, അന്തരീക്ഷ പഠനം തുടങ്ങിയവയക്ക് ജി.പി.എസ് ഉപയോഗിക്കുന്നു. ഇന്ന് ജി.പി.എസ് റിസീവറുകൾ മൊബൈൽ ഫോണുകളിൽ സംയോജിപ്പിച്ച് വിവിധ ഗതി നിർണ്ണയ പ്രയോഗങ്ങൾക്ക് ഉപയോഗിക്കുന്നു.

d. സ്മാർട്ട് കാർഡുകൾ

ചിത്രം 11.2 ഒരു സ്മാർട്ട് കാർഡിന്റെ മാതൃക

11-ാം ക്ലാസിലെ രണ്ടാം അധ്യായത്തിൽ നാം പഠിച്ചിട്ടുള്ള സ്മാർട്ട് കാർഡുകളും സ്മാർട്ട് കാർഡ് റീഡറുകളും ഒന്നോർത്തു നോക്കൂ. ഒരു കമ്പ്യൂട്ടർ ചിപ്പ് അല്ലെങ്കിൽ വിവരശേഖരണത്തിനും വിനിമയത്തിനുമായി ഉപയോഗിക്കുന്ന മെമ്മറി ഉൾക്കൊള്ളുന്ന ഒരു പ്ലാസ്റ്റിക് കാർഡാണ് സ്മാർട്ട് കാർഡ്. ഇത് സുരക്ഷിതവും (ഡാറ്റ സംരക്ഷിക്കപ്പെടുന്നു) ബുദ്ധിവൈഭവമുള്ളതും (ഡാറ്റ സംഭരിക്കാനും പ്രോസസ്സ് ചെയ്യാനും കഴിയുന്നു) സൗകര്യപ്രദവും (കൊണ്ടു നടക്കാൻ എളുപ്പം) ആണെന്നുള്ളത് ഇതിന്റെ മേന്മകളാണ്. അതുകൊണ്ട് തന്നെയാണ് വ്യാപാര സ്ഥാപനങ്ങളും മറ്റ് സംഘടനകളും തിരിച്ച

റിയലിനായും വിവരസംഭരണത്തിനായും സ്മാർട്ട് കാർഡുകൾ ഉപയോഗിക്കുന്നത്. ഭാരത സർക്കാർ നൽകുന്ന RSBY സ്കീമിന്റെ സ്മാർട്ട് കാർഡിന്റെ മാതൃകാ ചിത്രം 11.2 ൽ കാണിച്ചിരിക്കുന്നു.

ചിത്രം 11.3: GSM SIM കാർഡ്

GSM മൊബൈൽ ആശയവിനിമയത്തിൽ SIM ൽ സ്മാർട്ട് കാർഡ് സാങ്കേതിക വിദ്യ ഉപയോഗിക്കുന്നു. ഒരു ഉപയോക്താവിനെ തിരിച്ചറിയാൻ ചിത്രം 10.3 ലുള്ള സിം കാർഡ് സഹായിക്കുന്നതിനോടൊപ്പം ഇന്റർനെറ്റ് ബ്രൗസിങ്, മൊബൈൽ വാണിജ്യം, മൊബൈൽ ബാങ്കിങ് മുതലായവയ്ക്കും ഇവ ഉപയോഗിക്കുന്നു. ക്രെഡിറ്റ് കാർഡുകൾ, ATM കാർഡുകൾ, ഇന്ധന കാർഡുകൾ, ടെലിവിഷൻ വരിക്കാർക്കുള്ള പ്രവേശനാനുമതിക്കാർഡുകൾ, അതി സുരക്ഷാ തിരിച്ചറിയൽ കാർഡുകൾ എന്നിവയായും സ്മാർട്ട് കാർഡുകൾ ഉപയോഗിച്ചു വരുന്നു.

11.3 മൊബൈൽ ഓപ്പറേറ്റിങ് സിസ്റ്റം (Mobile operating system)

ഡെസ്ക് ടോപ്പ്, ലാപ്പ്ടോപ്പ് എന്നീ കമ്പ്യൂട്ടറുകളിൽ ഓപ്പറേറ്റിങ്ങ് സിസ്റ്റം ഉപയോഗിക്കുന്നത് പോലെ മൊബൈൽ ഉപകരണങ്ങളായ സ്മാർട്ട് ഫോൺ, ടാബ് എന്നിവയിൽ ഉപയോഗിക്കുന്നതാണ് മൊബൈൽ ഓപ്പറേറ്റിങ്ങ് സിസ്റ്റം.

ചിത്രം 11.4: ജനപ്രിയ മൊബൈൽ ഓപ്പറേറ്റിങ് സിസ്റ്റങ്ങളുടെ ലോഗോകൾ.

ഒരു മൊബൈൽ ഉപകരണത്തിലെ ഹാർഡ് വെയർ, മൾട്ടിമീഡിയ പ്രവർത്തനങ്ങൾ, ഇന്റർനെറ്റ് ബന്ധം തുടങ്ങിയവ കൈകാര്യം ചെയ്യുന്നത് മൊബൈൽ ഓപ്പറേറ്റിങ് സിസ്റ്റമാണ്. മൊബൈൽ ഉപകരണങ്ങളിലെ പ്രോഗ്രാമുകളായ ആപ്ലിക്കേഷനുകൾ (Apps) പ്രവർത്തിക്കുന്നത് ഈ സോഫ്റ്റ്‌വെയറിലൂടെയാണ്. ബ്ലാക്ക്ബെറി OS, ഗൂഗിളിന്റെ ആൻഡ്രോയിഡ്, ആപ്പിളിന്റെ ഐ.ഒ.എസ്, മൈക്രോസോഫ്റ്റിന്റെ വിൻഡോസ് ഫോൺ എന്നിവ പ്രചാരത്തിലുള്ള മൊബൈൽ ഓപ്പറേറ്റിങ് സിസ്റ്റങ്ങളാണ് (ചിത്രം 11.4).

ആൻഡ്രോയിഡ് ഓപ്പറേറ്റിങ് സിസ്റ്റം

ടച്ച് സ്ക്രീൻ സംവിധാനമുള്ള മൊബൈൽ ഉപകരണങ്ങളായ സ്മാർട്ട് ഫോൺ, ടാബ്ലെറ്റ് കമ്പ്യൂട്ടർ എന്നിവയ്ക്കായി രൂപകല്പന ചെയ്തിട്ടുള്ളതാണ് ലിനക്സ് അധിഷ്ഠിതമായ ആൻഡ്രോയ്ഡ് ഓപ്പറേറ്റിങ് സിസ്റ്റം. 2003-ൽ കാലിഫോർണിയയിലെ പാലോ ആൾട്ടോയിൽ ആൻഡി റൂബിനും സുഹൃത്തുക്കളും കൂടി സ്ഥാപിച്ച Android Inc ആണ് ആൻഡ്രോയ്ഡ് ഓപ്പറേറ്റിങ് സിസ്റ്റം വികസിപ്പിച്ചത്. 2005 ൽ ഈ സ്ഥാപനത്തെ ഗൂഗിൾ ഏറ്റെടുത്ത് പൂർണ്ണ അനുബന്ധ കമ്പനിയാക്കി മാറ്റി. ഗൂഗിൾ കമ്പനിയിൽ റൂബിൻ നേതൃത്വം നൽകുന്ന സംഘം ലിനക്സ് കേണലിന്റെ പിൻബലത്തിൽ പ്രവർത്തിക്കുന്ന ഒരു മൊബൈൽ ഉപകരണ പ്ലാറ്റ്ഫോമിന് രൂപകല്പന നൽകി. മൊബൈൽ ഉപകരണങ്ങൾക്ക് വേണ്ട ഗുണനിലവാരം വികസിപ്പിക്കുക എന്ന ലക്ഷ്യം മുൻനിർത്തി 2007 ൽ ഗൂഗിൾ, എച്ച്.ടി.സി, ഇന്റൽ, മോട്ടറോള തുടങ്ങിയ കമ്പനികൾ ഓപ്പൺ

ഹാൻഡ്സെറ്റ് അലയൻസ് എന്നൊരു കൂട്ടായ്മ രൂപീകരിച്ചു. ഓപ്പൺ ഹാൻഡ്സെറ്റ് അലയൻസി (OHA) ന്റെ രൂപീകരണത്തോടെ ആൻഡ്രോയ്ഡും പുറത്തിറക്കി.

ആൻഡ്രോയ്ഡിൽ ഉപയോക്താവിനെയും ഉപകരണത്തെയും തമ്മിൽ ബന്ധിപ്പിക്കാൻ ടച്ചിങ്, സ്വൈപ്പിങ്, പിഞ്ചിങ്, റിവേഴ്സ് പിഞ്ചിങ് എന്നിവ പോലെയുള്ള ടച്ച് ഇൻപുട്ടുകളാണ് ഉപയോഗിക്കുന്നത്. മൊബൈലുകളിലെ ആപ്ലിക്കേഷനുകളും വിഡ്ജറ്റുകളും ഉപയോഗിക്കുന്നതിന് വേണ്ട കുറുക്കുവഴികൾ, ഉപയോക്താവിന് സ്വന്തമായി രൂപകല്പന ചെയ്യാം. ആൻഡ്രോയ്ഡ് പുറത്തിറങ്ങിയ 2007 ന് ശേഷം ഇതിന്റെ വിപണി മൂല്യം കുത്തനെ ഉയരുകയും ഇന്ന് ഏറ്റവും കൂടുതലായി ഉപയോഗിക്കുന്ന മൊബൈൽ ഓപ്പറേറ്റിങ് സിസ്റ്റമായി അത് മാറുകയും ചെയ്തു. ആൻഡ്രോയിഡിന്റെ പ്രധാന പതിപ്പുകളെല്ലാം ഇംഗ്ലീഷ് ഭാഷയിലെ അക്ഷരമാലാ ക്രമത്തിലുള്ള കോഡു രൂപത്തിലാണ് പുറത്തിറങ്ങിയത്.

ഇതുവരെ പുറത്തിറങ്ങിയ ആൻഡ്രോയ്ഡ് പതിപ്പുകളുടെ അവരോഹണ ക്രമത്തിലുള്ള പേരുകൾ പട്ടിക 11.1 ൽ ചേർക്കുന്നു.

പതിപ്പ്	പേര്	പുറത്തിറങ്ങിയ തീയതി
8.0 - 8.1	ഓറിയോ	21/08/2017
7.0 - 7.1.2	ന്യൂഗ	22/08/2016
6.0 - 6.1.2	മാഷ്മല്ലോ	05/10/2015
5.0 - 5.1.1	ലോലിപോപ്പ്	12/11/2014
4.4 - 4.4.4	കിറ്റ്കാറ്റ്	31/10/2013
4.1 - 4.3.1	ജെല്ലി ബീൻസ്	09/07/2012
4.0 - 4.0.4	ഐസ്ക്രീം സാൻഡ്വിച്ച്	18/10/2011
3.0 - 3.2.6	ഹണികോംബ്	22/02/2011
2.3 - 2.3.7	ജിഞ്ചർ ബ്രെഡ്	06/12/2010
2.2 - 2.2.3	ഫ്രോയോ	20/05/2010
2.0 - 2.1	എക്ലയർ	26/10/2009
1.6	ഡോനട്ട്	15/09/2009
1.5	കപ്കേക്ക്	27/04/2009
1.1		23/09/2008
1.0		09/12/2009

പട്ടിക 11.1: ആൻഡ്രോയ്ഡ് പതിപ്പുകൾ

ലിനക്സ് കേണലിനെ അടിസ്ഥാനമാക്കിയുള്ള ഒരു കേണലാണ് ആൻഡ്രോയ്ഡ് OS-ൽ അടങ്ങിയിരിക്കുന്നത്. ലിനക്സ് കേണലിന് പ്രബലമായ മെമ്മറിയും പ്രക്രിയ നിർവഹണശേഷിയും (process management), അനുവാദം ആവശ്യമായ സുരക്ഷ ഘടനയും (permission based security structure), സ്വതന്ത്ര സോഫ്റ്റ്വെയർ സ്വാഭാവവും ഉള്ളതു കൊണ്ടാണ് ഇത് ആൻഡ്രോയ്ഡ് ഡെവലപ്പ്മെന്റ് കിറ്റിൽ ഉപയോഗിക്കുന്നത്. ആൻഡ്രോയ്ഡ് ഡെവലപ്മെന്റ് കിറ്റ് ഉപയോഗിച്ച് ആപ്ലിക്കേഷനുകൾ നിർമിക്കാനുള്ള ചട്ടക്കൂട് ആൻഡ്രോയിഡിൽ ലഭ്യമാണ്. ഇത് ഉപയോഗിച്ചാണ് ഗൂഗിൾ മാപ്സ്, ഫേസ് ബുക്ക്, തുടങ്ങിയവ നിർമ്മിച്ചിട്ടുള്ളത്.

അപ്പാഷേ ലൈസൻസ് പ്രകാരം തയാറാക്കപ്പെട്ട ആൻഡ്രോയ്ഡ്, ഹാർഡ്വെയർ നിർമ്മാതാക്കൾക്കും കമ്പ്യൂട്ടർ പ്രോഗ്രാം തയാറാക്കുന്നവർക്കും ആരുടേയും അനുവാദം കൂടാതെ സ്വതന്ത്രമായി പരിഷ്ക്കരിക്കുവാനോ വിതരണം ചെയ്യുവാനോ സാധിക്കും. ആപ്സ് എന്നറിയപ്പെ

ടുന്ന ആപ്ലിക്കേഷനുകൾ തയാറാക്കുവാൻ ഒരു വലിയ സമൂഹമുള്ളതിനാൽ ഇതിന്റെ ഉപയോഗം വർധിക്കുന്നു. ജാവാ പ്രോഗ്രാമിങ് ഭാഷയുടെ ഉപഭോക്തൃ പതിപ്പ് ഉപയോഗിച്ചാണ് ആപ്സ് രചിക്കപ്പെട്ടിട്ടുള്ളത്.

സ്വതന്ത്ര സോഫ്റ്റ്‌വെയറായ ആൻഡ്രോയ്ഡ് ഉപയോഗിച്ച് ധാരാളം ആളുകൾക്ക് ആപ്സ് എന്ന പേരിൽ അറിയപ്പെടുന്ന ആപ്ലിക്കേഷനുകൾ/പ്രോഗ്രാമുകൾ തയാറാക്കുവാൻ സാധിക്കുന്നതിനാൽ ഇതിന്റെ സ്വീകാര്യത വർദ്ധിച്ചു. ഇങ്ങനെ തയാറാക്കുന്ന ആപ്സ് ആൻഡ്രോയ്ഡ് പ്ലേ സ്റ്റോറിൽ നിന്ന് സൗജന്യമായി ഡൗൺലോഡ് ചെയ്യാം. ഇത് ഈ OS ന്റെ ജനപ്രിയത വർധിപ്പിച്ചു. മൊബൈൽ ഹാർഡ്‌വെയർ, സോഫ്റ്റ്‌വെയർ രംഗത്തെ എടുത്തു പറയാവുന്ന മുന്നേറ്റത്തിന് കാരണം ആൻഡ്രോയ്ഡ് മൂലമായതിനാൽ നാളെ ഡെസ്ക്ടോപ്പ് കമ്പ്യൂട്ടറിലെ ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിന് പകരം ഇത് ഉപയോഗിച്ചേക്കാം. ആൻഡ്രോയ്ഡ് ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിലെ പരിഷ്ക്കരിച്ച പതിപ്പുകൾ ഇലക്ട്രോണിക് ഉപകരണങ്ങളായ ടെലിവിഷൻ, വാഷിങ് മെഷീൻ, വാച്ചുകൾ തുടങ്ങിയവയിൽ ഉപയോഗിക്കുന്നു.

നമുക്കു ചെയ്യാം

- *കമ്പോളത്തിൽ ലഭ്യമായ വിവിധതരം മൊബൈൽ ഓപ്പറേറ്റിങ് സിസ്റ്റങ്ങളും അവയുടെ സവിശേഷതകളും പ്രദർശിപ്പിക്കുന്ന ഒരു രേഖാചിത്രം (chart) തയാറാക്കുക.*

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. SIM എന്നാൽ
   (a) Subscriber Identity Module
   (b) Subscriber Identity Mobile
   (c) Subscription Identification Module
   (d) Subscription Identification Mobile
2. എന്താണ് ജി.പി.എസ്. (GPS)?
3. എസ്.എം.എസ്. (SMS) സന്ദേശങ്ങൾ അയയ്ക്കുവാൻ ______ പ്രോട്ടോക്കോൾ ഉപയോഗിക്കുന്നു.
4. മൊബൈൽ ഫോണുകൾ ഉപയോഗിച്ച് ദൃശ്യശ്രവ്യ (Multimedia) മാധ്യമങ്ങൾ എങ്ങനെ അയയ്ക്കുവാൻ സാധിക്കും?
5. മൊബൈൽ ഓപ്പറേറ്റിങ് സിസ്റ്റത്തിന്റെ ധർമ്മങ്ങൾ എന്തൊക്കെയാണ്?

11.4 വ്യാപാരത്തിൽ ഐ.സി.റ്റി. (ICT in business)

പതിനൊന്നാം ക്ലാസ്സിലെ വിവര സാങ്കേതികവിദ്യയുടെ ഉപയോഗം എന്ന പത്താമത്തെ അധ്യായത്തിൽ വിവര സാങ്കേതികവിദ്യ വ്യാപാരത്തിൽ ഉപയോഗിക്കുന്നതിന്റെ മേന്മകൾ നമ്മൾ ചർച്ച ചെയ്തിരുന്നു. ഇവിടെ നമ്മൾ വ്യാപാരം പുരോഗതിയിലേക്ക് കൊണ്ടു പോകുവാൻ സഹായിക്കുന്ന ചില ഉപാധികളെക്കുറിച്ച് ചർച്ച ചെയ്യുന്നു. പല കമ്പനികളുടെയും ഒഴിവാക്കാനാവാത്ത ഒന്നായി ഐ.സി.റ്റി. മാറിയിരിക്കുന്നു. ഉൽപന്നങ്ങളുടെ മേന്മ വർദ്ധിപ്പിക്കുവാനും ഉപഭോക്താക്കൾക്ക് നൽകുന്ന സേവനങ്ങളും ഐ.സി.റ്റി. യുടെ ഉപയോഗത്താൽ വളരെ എളുപ്പം സാധി

ക്കുന്നു. വ്യക്തികൾ സാധനങ്ങൾ വാങ്ങുന്നതും ഉൽപ്പന്നങ്ങൾ ഉപയോക്താക്കളിൽ എത്തിച്ചേരുന്നതും സവിശേഷമായ മാറ്റങ്ങൾക്ക് വിധേയമായിട്ടുണ്ട്. വ്യക്തികളുടെ ഉപഭോക്തൃശീലം ഇന്റർനെറ്റിന്റെ വരവോടു കൂടി മാറ്റത്തിന് വിധേയമായിട്ടുണ്ട്. ഐ.സി.റ്റി.യുടെ വരവോടു കൂടിയാണ് വ്യാപാരത്തിൽ പ്രധാന മാറ്റങ്ങളുണ്ടായത്.

വിപണന തന്ത്രങ്ങളും വ്യാപാര രീതികളും ഒട്ടേറെ മാറ്റങ്ങൾക്ക് വിധേയമായിട്ടുണ്ട്. ഉപഭോക്തൃ താല്പര്യങ്ങൾക്ക് ഇക്കാലത്തെ വ്യാപാരങ്ങളിൽ സുപ്രധാന പങ്കുണ്ട്. പൊതു വിപണനത്തിനൊപ്പം ഓരോ ഉപഭോക്താവിനും സവിശേഷ പ്രധാന്യം ലഭിക്കുന്ന സാഹചര്യമാണ് ഇന്നുള്ളത്.

ഓൺലൈൻ വ്യാപാരം ആരംഭിച്ചതോടെ, ഉപഭോക്താക്കൾ എന്തു വാങ്ങി എന്നു മാത്രമല്ല, അവരെന്താണ് അന്വേഷിക്കുന്നതെന്നും ഏതെല്ലാം സൈറ്റുകൾ സന്ദർശിക്കുന്നുവെന്നും വ്യാപാരികൾക്ക് മനസ്സിലാക്കാൻ കഴിയുന്നു. പരസ്യ പ്രചാരണങ്ങളാൽ ഉപഭോക്താക്കൾ എത്രത്തോളം സ്വാധീനിക്കപ്പെടുന്നുവെന്നും അറിയാൻ കഴിയും.

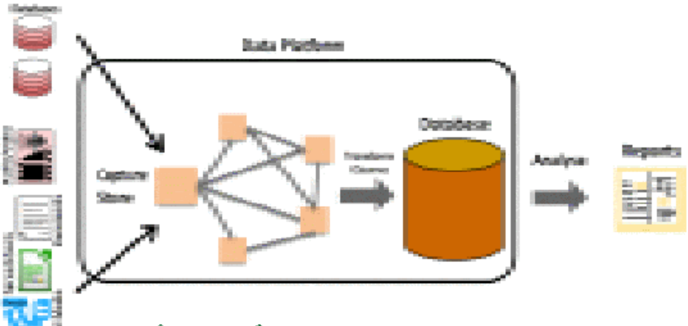
ഓരോ ഉപഭോക്താവിന്റെയും സോഷ്യൽ മീഡിയോ വിശകലനങ്ങളും വ്യാപാരികൾക്ക് ശ്രദ്ധിക്കാൻ കഴിയും.

ഒരാൾ ഒരുല്പന്നം ഓൺലൈനിൽ വാങ്ങുന്നതോടെ ആ ഉല്പന്നം ചുരുങ്ങിയ ചെലവിലും കുറഞ്ഞ സമയത്തിനുള്ളിലും അയാളെ തേടിയെത്തുന്നു.

വിവരവിനിമയ സാങ്കേതികവിദ്യ (ICT) യുടെ സഹായത്തോടെ, ഒരു ഉല്പന്നത്തിന്റെ സഞ്ചാരപഥങ്ങളും കൈമാറ്റ വിശദാംശങ്ങളും ഒരു ഉപഭോക്താവിന് യഥാസമയം പിന്തുടരാൻ കഴിയുന്നു. ഐ.സി.ടി.യുടെ ഉപയോഗത്തോടെ വ്യാപാര രംഗത്തുണ്ടായ സുപ്രധാന വികാസങ്ങൾ നമുക്ക് ചർച്ച ചെയ്യാം.

11.4.1 സാമൂഹിക ശൃംഖലകളും ബൃഹത്തായ ഡാറ്റ അപഗ്രഥനങ്ങളും (Social networks and big data analytics)

ഇക്കാലത്ത് ഒരു ഉല്പന്നം ഓൺലൈൻ ആയോ അല്ലെങ്കിൽ അടുത്തുള്ള കടയിൽ നിന്നോ വാങ്ങുന്നതിനു മുമ്പായി വ്യക്തികൾ ഉൽപന്നങ്ങളുടെ വില, വിലയിരുത്തൽ എന്നിവ വെബ്സൈറ്റിൽ തിരയുകയും, സമാന ഉൽപന്നങ്ങളുമായി താരതമ്യം ചെയ്യുകയും ചെയ്യുന്നു. അവർ അതു കൂടാതെ നിർദ്ദിഷ്ട ഉൽപന്നത്തെക്കുറിച്ച് സാമൂഹിക ശൃംഖലകൾ വഴി തിരയുകയും, സുഹൃത്തുക്കളിൽ നിന്ന് അഭിപ്രായം സ്വീകരിക്കുകയും ചെയ്യുന്നു. അതുകൊണ്ട് വാണിജ്യ സ്ഥാപനങ്ങൾ സാമൂഹിക മാധ്യമങ്ങൾ വഴിയുള്ള ചർച്ചകളും അഭിപ്രായങ്ങളും നിരീക്ഷിക്കുവാൻ തുടങ്ങിയിട്ടുണ്ട്. വാണിജ്യ സ്ഥാപനങ്ങൾ സാമൂഹിക മാധ്യമങ്ങളുടെ ശക്തി ഉപയോഗിച്ചു കൊണ്ട് അവരുടെ വിപണന മേഖലയെക്കുറിച്ച് നല്ലവണ്ണം മനസ്സിലാക്കുന്നു. ഇതിന് ആവശ്യമായ ഡാറ്റ സാമൂഹിക മാധ്യമത്തിൽ നിന്ന് ശേഖരിക്കുകയും അതോടൊപ്പം ഉപയോക്താക്കളുടെ ആസൂത്രണ ഘടനയുടെ രൂപം മനസ്സിലാക്കുകയും ചെയ്യുന്നു. ഇങ്ങനെയുള്ള ഡാറ്റയുടെ വലിപ്പം വളരെ വലുതാണ്. ഈ ഡാറ്റകൾ ആവിർഭവിക്കുന്ന സാമൂഹിക മാധ്യമങ്ങളിലെ കുറിപ്പുകൾ, ഡിജിറ്റൽ ചിത്രങ്ങൾ വീഡിയോകൾ, വിലയ്ക്കു വാങ്ങൽ ഇടപാടുകളുടെ രേഖകൾ, സെൽഫോൺ ഡാറ്റകൾ തുടങ്ങിയവയെല്ലാം തന്നെ പ്രത്യേക വാണിജ്യമേഖലയെ സംബന്ധിച്ചിടത്തോളം വളരെ വലിയ വ്യവസായ ഡാറ്റകളാണ്.



ചിത്രം 11.5: വിപുലമായ ഡാറ്റയുടെ അപഗ്രഥനം

വളരെ വലിയ ഡാറ്റയുടെ വിശകലനം വളരെ പ്രയാസമാണ്. ഈ ഡാറ്റകളുടെ ഉറവിടം എന്നത് ചിലപ്പോൾ ഘടനയോടു കൂടി ഡാറ്റാബേസിൽ നിന്ന് ആകാം. അല്ലെങ്കിൽ ഘടനയില്ലാതെ ശബ്ദത്തിൽ നിന്നോ, ലിഖിതങ്ങളിൽ നിന്നോ, സ്പ്രെഡ് ഷീറ്റിൽ നിന്നോ, സാമൂഹിക മാധ്യമങ്ങളിലെ കുറിപ്പുകളിൽ നിന്നോ ആകാം. വിവിധതരത്തിലുള്ള സോഫ്റ്റ്വെയർ ഉപാധികൾ ഉപയോഗിച്ച് വലിയ ഡാറ്റകളെ വിശകലനം ചെയ്യാം, സൂക്ഷിക്കാം, വ്യാപാരത്തിനാവശ്യമായ തീരുമാനങ്ങൾ എടുക്കാം. മറിഞ്ഞ് കിടക്കുന്ന ഘടനകൾ, വ്യാപാരത്തിലെ പ്രവണതകൾ, ഉപയോക്താക്കളുടെ മുൻഗണന വ്യാപാരത്തിന് സഹായകകരമായ വിവരങ്ങൾ തുടങ്ങിയ വ്യത്യസ്ത ഡാറ്റ ഇനങ്ങളിൽ നിന്ന് വശകലന പ്രക്രിയയ്ക്ക് വിധേയമാക്കി എടുക്കുക എന്നതാണ് ഡാറ്റ അപഗ്രഥനം കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. ഇങ്ങനെ അപഗ്രഥിച്ച് കിട്ടുന്ന കണ്ടെത്തലുകൾ/വിവരങ്ങൾ ഉപയേഗിച്ച് വ്യാപാരം പരിഷ്കരിക്കുവാനുള്ള പ്രചാരണങ്ങൾ, പുതിയ ആദായത്തിനുള്ള അവസരങ്ങൾ, മെച്ചപ്പെട്ട ഉപയോക്തൃ സേവനങ്ങൾ, മെച്ചപ്പെടുത്തിയ പ്രവർത്തന കഴിവുകൾ, എതിർസ്ഥാപനങ്ങളുമായി ഉള്ള മത്സരങ്ങളിലെ മേൽക്കോയ്മ മറ്റ് വ്യാവസായിക മെച്ചങ്ങൾ തുടങ്ങിയവ നേടാം. ഒരു വലിയ ഡാറ്റ അപഗ്രഥനത്തിന്റെ ഒരു മാതൃകാ ഘടന ചിത്രം 11.5 ൽ നൽകിയിരിക്കുന്നു.

മുൻകാലങ്ങളിൽ ഉത്തരം കണ്ടെത്താൻ കഴിയാതിരുന്ന ഒട്ടനേകം ചോദ്യങ്ങൾക്ക് ഉത്തരം കണ്ടെത്താനുള്ള അവസരമാണ് വിപുലമായ ഡാറ്റയുടെ അപഗ്രഥനത്തിലൂടെ വ്യാപാര സ്ഥാപനങ്ങൾക്ക് ലഭിക്കുന്നത്. ഉപഭോക്താക്കളുടെ ആവശ്യങ്ങൾ തിരിച്ചറിഞ്ഞ്, ഫലപ്രദമായ ഇടപെടലുകൾക്ക് അവസരം ഒരുക്കുന്ന ഒരു പുതിയ ലോകത്തിലേക്കുള്ള വാതായനമാണ് തുറക്കപ്പെട്ടത്.

11.4.2 ബിസിനസ് ലോജിസ്റ്റിക് (Business logistics)

ഉപഭോക്താക്കളുടെയോ സ്ഥാപനങ്ങളുടെയോ ആവശ്യങ്ങൾക്കനുസരിച്ച് വസ്തുക്കളോ വിഭവങ്ങളോ അവയുടെ ഉത്ഭവകേന്ദ്രത്തിൽ നിന്നും ഉപഭോഗ സ്ഥാനത്തേക്കുള്ള നീക്കം യുക്തിഭദ്രമായി കൈകാര്യം ചെയ്യുക എന്നതാണ് ബിസിനസ്സ് ലോജിസ്റ്റിക് (Business logistic) കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. ഭക്ഷണ പദാർഥങ്ങൾ, ഉൽപന്നങ്ങൾ, മൃഗങ്ങൾ തുടങ്ങിയ സാധനങ്ങൾ വിന്യാസ ശാസ്ത്രം (logistic) കൈകാര്യം ചെയ്യുന്ന ദ്രവ്യങ്ങളിൽ ഉൾപ്പെടുന്നു.

ശരിയായ വിലയ്ക്ക് യഥാർഥ ഉപഭോക്താവിന് കൃത്യ സമയത്ത് കൃത്യ സ്ഥലത്ത് കൃത്യ അളവിലും നിബന്ധനയിലും യഥാർഥ ഉൽപന്നം ലഭ്യമാക്കുന്നത് ഉറപ്പാക്കുക എന്നതാണ് വ്യാപാര വിന്യാസശാസ്ത്ര (Business logistic) ത്തിന്റെ പ്രഥമ ലക്ഷ്യം.

i. സ്ഥാപനങ്ങൾക്ക് ഉപകരണങ്ങൾ നൽകുന്നവരിൽ നിന്ന് വാങ്ങുക.

ii. സ്ഥാപനങ്ങളുടെ ഉത്പാദന ഇടങ്ങളിലേക്ക് ഉപകരണങ്ങൾ എത്തിക്കുക.

iii. ഉപഭോക്താക്കൾക്ക് പൂർത്തിയാക്കിയ ഉൽപന്നങ്ങൾ സംഭരണശാലകൾ വഴി എത്തിച്ച് നൽകുക.

ഉൽപന്നങ്ങളുടെ വിശ്വാസ്യത, വസ്തുവിവരപ്പട്ടിക, പൊതിഞ്ഞു കെട്ടൽ, സുരക്ഷ തുടങ്ങിയവ ഇതിൽ ഉൾപ്പെടുന്നു. വിന്യാസ ശാസ്ത്രത്തിന്റെ സങ്കീർണാവസ്ഥ ഹാർഡ്‌വെയറും സോഫ്റ്റ്വെയറും ഉപയോഗിച്ച് ലഘൂകരിക്കാം.

റേഡിയോ ഫ്രീക്വൻസി ഐഡന്റിഫിക്കേഷൻ (RFID) എന്ന സാങ്കേതികവിദ്യ ഉപയോഗിച്ച് വിവിധതരത്തിലുള്ള സാധനങ്ങളെ തിരിച്ചറിയാനും, അവ പിന്തുടരാനും, ക്രമപ്പെടുത്താനും കഴിയും. RFID ടാഗും ഒരു റീഡറും ആണ് RFID ഹാർഡ്‌വെയറിൽ ഉൾപ്പെട്ടിരിക്കുന്നത്. RFID ടാഗിൽ സംപ്രേഷണ സാമഗ്രിയും സ്വീകരണ സാമഗ്രിയും ഉൾപ്പെട്ടിട്ടുണ്ട്. RFID ടാഗ് ട്രാൻസ്മിറ്ററിന്റെയും റെസ്പോണ്ടറിന്റെയും ഒരു കൂടിച്ചേരൽ ആണ്. ഈ ടാഗിൽ ഒരു മൈക്രോചിപ്പ് ഡാറ്റ സൂക്ഷിക്കാനും RFID റീഡറിലേക്ക് ഡാറ്റ അയയ്ക്കാനും സ്വീകരിക്കുവാനും ഉള്ള ആന്റിനയും

ഉൾപ്പെടുത്തിയിരിക്കുന്നു. ഈ കനം കുറഞ്ഞ ടാഗ് ഉൽപന്നങ്ങളിലോ അവയുടെ പെട്ടിയിലോ ഉൾക്കൊള്ളിക്കുകയോ പതിപ്പിക്കുകയോ ചെയ്യും. ഓരോ ടാഗിലും ഉൽപന്നങ്ങളെ തിരിച്ചറിയുവാനുള്ള വിവരങ്ങൾ ഉൾക്കൊള്ളിച്ചുണ്ടാകും. റീഡറും ടാഗും തമ്മിലാണ് ആശയവിനിമയം നടക്കുന്നത്. RFID ടാഗും റീഡറും നേർക്ക്നേർ വരണമെന്നില്ലെന്നത് ഈ സംവിധാനത്തിന്റെ വലിയൊരു നേട്ടമാണ്. നിരവധി മീറ്ററുകളിലെ നിന്നു പോലും RFID റീഡറിന് ടാഗിലെ വിവരങ്ങൾ വായിക്കുവാനും ഉല്പന്നത്തെ തിരിച്ചറിയാൻ കഴിയും.

ചിത്രം 11.6: RFID ടാഗ്

ബാറ്ററി ഉപയോഗിച്ച് ഊർജം നൽകി കൊണ്ട് ടാഗിനെ പ്രവർത്തനനിരതവും, അതിന്റെ റീഡർ ഉപയോഗിച്ച് പ്രവർത്തനരഹിതവും ആക്കാം. ലോജിസ്റ്റിക്സിനെ അപേക്ഷിച്ച്, വലിയ സൂപ്പർ മാർക്കറ്റുകളിൽ, ഹൈവേകളിലെ ടോൾ ബൂത്തുകളിൽ, ബാർകോഡിന് പകരം വെയ്ക്കാൻ പറ്റുന്ന രീതിയിൽ RFID പ്രചാരം നേടി വരുന്നു. RFID കടുവ, സിംഹം തുടങ്ങിയ മൃഗങ്ങളിൽ സ്ഥാപിച്ചു കൊണ്ട് ഇവയുടെ കണക്കെടുപ്പ് ലളിതമാക്കാം.

RFID ചരക്കു നീക്കം, ഗതാഗതം, എന്നിവയുടെ ദൃശ്യത വർദ്ധിപ്പിക്കുന്നതിലൂടെ വാണിജ്യ പ്രക്രിയയെ കൂടുതൽ ഫലപ്രദമാക്കുന്നു.

ഒരു ഉല്പന്നത്തിന്റെ തൽക്ഷണ ഡാറ്റ (Real time data) നൽകുന്നതിലൂടെ ആ ഉല്പന്നത്തെക്കുറിച്ചുള്ള നാളിതു വരെയുള്ള (Up to date) വിവരങ്ങൾ ഉപഭോക്താവിന് ലഭ്യമാക്കുന്നതിന്റെ കാര്യക്ഷമത വർദ്ധിപ്പിക്കാനും RFID ക്ക് കഴിയുന്നു. RFID ഉപയോഗിച്ചുള്ള ബിസിനസ് ലോജിസ്റ്റിക് സോഫ്റ്റ്‌വെയറിന്റെ സഹായത്താൽ പ്രവർത്തന ചെലവ് കുറയ്ക്കാനും വിതരണ കേന്ദ്രങ്ങളിലെ ഉല്പാദനക്ഷമത കൂട്ടാനും കഴിയുന്നു. കൃത്യസമയത്ത് ഉല്പന്നം വിതരണം ചെയ്യുന്നതിലൂടെ ഉപഭോക്താക്കൾക്കുള്ള സേവനം മെച്ചപ്പെടുത്താനും അവരുടെ സംതൃപ്തി വർദ്ധിപ്പിക്കാനും സാധിക്കുന്നു.

11.5 വിവരങ്ങളുടെ സുരക്ഷിതത്വം (Information security)

ഇന്നത്തെ കാലത്ത് ആശയവിനിമയം, സാധനങ്ങൾ വാങ്ങിക്കൽ, ബാങ്കിങ് എന്നിങ്ങനെയുള്ള എല്ലാവിധ പ്രവർത്തനങ്ങളും ഇന്റർനെറ്റിലൂടെ നിർവഹിക്കപ്പെടുന്നു. ഇങ്ങനെയുള്ള പ്രവർത്തനങ്ങൾ നിർവഹിക്കപ്പെടുമ്പോൾ കമ്പ്യൂട്ടറുകൾ തമ്മിൽ വിവരങ്ങൾ കൈമാറ്റം ചെയ്യപ്പെടുന്നു. കമ്പ്യൂട്ടറുകൾ തമ്മിൽ കൈമാറ്റം ചെയ്യപ്പെടുന്ന വിവരങ്ങളുടെ സുരക്ഷിതത്വം ആശങ്ക ഉണ്ടാക്കുന്നതാണ്. ഈ ഭാഗത്ത് സൈബർ സുരക്ഷ, പകർപ്പവകാശം, വ്യാപാരമുദ്ര ലംഘനങ്ങൾ, സൈബർ കുറ്റകൃത്യങ്ങൾ എന്നിങ്ങനെ ഇന്റർനെറ്റുപയോഗിക്കുമ്പോൾ ശ്രദ്ധിക്കേണ്ട കാര്യങ്ങളെക്കുറിച്ച് വിശദമായി ചർച്ച ചെയ്യുന്നു.

11.5.1 ബൗദ്ധിക സ്വത്തവകാശം (Intellectual Property Right)

സംഗീതം, സാഹിത്യം, ചിത്രരചന, നൂതനമായ കണ്ടുപിടുത്തങ്ങൾ, ഡിസൈനിംഗ്, സോഫ്റ്റ്‌വെയർ വികസനം എന്നീ മേഖലകളിൽ ധാരാളം പേർ സൃഷ്ടിപരമായ പ്രവർത്തനങ്ങൾ നിർവഹിച്ചു പോരുന്നു. ധാരാളം സമയവും കഠിനാധ്വാനവും ഇവർ അതിനായി ഉപയോഗിക്കുന്നു. ഇങ്ങിനെ ഉണ്ടാകുന്ന അറിവുകളെയും കണ്ടെത്തലുകളെയുമാണ് ബൗദ്ധിക സ്വത്ത് എന്ന് പറയുന്നത്. അനുവാദമില്ലാതെ ഒരാളുടെ ആശയങ്ങൾ മറ്റൊരാൾ എടുക്കുന്നത് നീതിക്കു നിരക്കാത്തതാണ്. ഇങ്ങനെയുള്ള ബൗദ്ധിക അറിവുകൾ വികസിപ്പിക്കുന്നവർക്ക് അതിന്റെ ആനുകൂല്യം ലഭിക്കണം. അതുകൊണ്ട് ഇവ സംരക്ഷിക്കേണ്ട

തുണ്ട്. 1883ലെ പാരീസ് കൺവെൻഷനിലാണ് ബൗദ്ധിക സ്വത്തവകാശത്തിന്റെ പ്രാധാന്യം തിരിച്ചറിഞ്ഞത്.

1886ൽ ഉണ്ടായ ബേൺ (BERNE) കൺവെൻഷനിലും അനുബന്ധമായ നിയമ നിർമ്മാണം നടത്തുകയുണ്ടായി. രണ്ട് ഉടമ്പടികളുടെയും കാര്യനിർവഹണം നടത്തുന്നത് വേൾഡ് ഇന്റലെക്ചൽ പ്രോപ്പർട്ടി ഓർഗനൈസേഷൻ (WIPO) ആണ്. WIPO, 1960 ൽ യുണൈറ്റഡ് നേഷന്റെ (UN) ആഭിമുഖ്യത്തിൽ സ്ഥാപിക്കപ്പെട്ടു. ലോകത്താകമാനമുള്ള ബൗദ്ധിക സ്വത്തുക്കളെയും അതിന്റെ അവകാശികളെയും കണ്ടെത്തി സംരക്ഷിക്കുക എന്നതാണ് WIPO യുടെ ദൗത്യം. WIPOയുടെ അടയാളചിഹ്നം ചിത്രം 11.7 ൽ കൊടുത്തിരിക്കുന്നു.

ബൗദ്ധിക സ്വത്തവകാശം എന്നത് ഭൂമിക്ക് മേലുള്ള അവകാശം, വീടിന് മേലുള്ള അവകാശം എന്നിങ്ങിനെ മറ്റേതൊരു അവകാശവും പോലെ തന്നെയാണ്. ഒരു പ്രത്യേക കാലയളവിലേക്ക് ഒരാൾക്ക് അയാളുടെ കണ്ടെത്തലിന്റെ മുകളിൽ പൂർണമായ അവകാശം നൽകുന്നു. ഒരാൾക്ക് തന്റെ കണ്ടുപിടുത്തത്തിലൂടെയോ സൃഷ്ടിയിലൂടെയോ സാമ്പത്തിക നേട്ടം ഉണ്ടാകുന്നതിനും, അംഗീകാരം ലഭിക്കുന്നതിനും IPR ആളുകളെ പ്രാപ്തരാക്കുന്നു. IPR ഉടമകൾക്ക് തങ്ങളുടെ സൃഷ്ടികൾ പണത്തിനു പകരമായി നൽകാനും സാധ്യമാക്കുന്നു. കമ്പനി അവർക്ക് ലഭിക്കുന്ന അവകാശം വിപണിയിലെത്തിക്കുകയും ചെയ്യുകയും ഈ നവ സൃഷ്ടിയെ സമൂഹത്തിൽ വിൽക്കുകയും ചെയ്യുന്നു. ഇത്തരത്തിൽ ബൗദ്ധിക സ്വത്തവകാശിയും കമ്പനിയും സമൂഹവും സൃഷ്ടിയിൽ നിന്ന് ആനുകൂല്യം കൈപ്പറ്റുന്നു. യുണൈറ്റഡ് നേഷൻസും (UN) മിക്കവാറും എല്ലാ രാജ്യങ്ങളും നവസൃഷ്ടികളെ പ്രോത്സാഹിപ്പിക്കുന്നതിനായി ബൗദ്ധിക സ്വത്തവകാശത്തെയും പ്രോത്സാഹിപ്പിക്കുന്നു.

ചിത്രം 11.7: WIPO ലോഗോ

ഓരോ രാജ്യത്തിനും അതിന്റേതായ രീതിയിലുള്ള ബൗദ്ധിക സ്വത്തവകാശ രജിസ്ട്രേഷൻ സംവിധാനങ്ങളുണ്ട്. വ്യാപാരമുദ്ര, വ്യാവസായികരൂപരേഖ മുതലായവക്ക് WIPO ഒരു അന്താരാഷ്ട്ര രജിസ്ട്രേഷൻ സംവിധാനമായി പ്രവർത്തിക്കുന്നു. WIPO യുടെ നിയമം എല്ലാ അംഗ രാജ്യങ്ങൾക്കും ബാധകമാണ്. ബൗദ്ധിക സ്വത്തവകാശത്തെ രണ്ടായി തരം തിരിക്കുന്നു- വ്യവസായിക സ്വത്തവകാശം, പകർപ്പവകാശം.

a) വ്യാവസായിക സ്വത്ത് (Industrial property)

വ്യവസായം, വാണിജ്യം, കാർഷികോൽപ്പന്നങ്ങൾ എന്നിവയെ സംബന്ധിക്കുന്നതാണ് വ്യാവസായിക സ്വത്ത്. കണ്ടെത്തലുകൾ, വ്യാപാരമുദ്രകൾ, വ്യവസായരൂപരേഖകൾ, ഭൗമശാസ്ത്രപരമായ സൂചകങ്ങൾ എന്നിവ ഈ നിയമ പ്രകാരം സംരക്ഷിക്കപ്പെടുന്നു. ഇന്ത്യയിൽ ബൗദ്ധികസ്വത്തവകാശം കേന്ദ്ര വ്യവസായ വകുപ്പിന്റെ കീഴിലുള്ള കൺട്രോളർ ജനറൽ ഓഫ് പേറ്റന്റ് ഡിസൈൻസ് ആൻഡ് ട്രേഡ്മാർക്സിൽ രജിസ്റ്റർ ചെയ്യാവുന്നതാണ്. ബൗദ്ധിക സ്വത്തവകാശത്തിന്റെ അടയാള ചിഹ്നം ചിത്രം 11.8 ൽ കാണിച്ചിരിക്കുന്നു.

ചിത്രം 11.8: IP India ലോഗോ

കുത്തകാവകാശം (Patents): ഒരു കണ്ടുപിടുത്തത്തിന്റെ മേൽ അത് കണ്ടുപിടിക്കുന്ന വ്യക്തിക്ക് നൽകുന്ന പരിപൂർണമായ അവകാശമാണ് കുത്തകാവകാശം. കണ്ടുപിടുത്തമെന്നാൽ ഒരു പുതിയ ഉൽപ്പന്നമോ, ഒരു കണ്ടെത്തലിനു സഹായിക്കുന്ന പ്രക്രിയയോ

ആകാം. ഒരു നിശ്ചിത കാലയളവിലേക്കാണ് കണ്ടുപിടിക്കുന്ന ആൾക്ക് അയാളുടെ ഉൽപ്പന്നത്തിന്റെ മേൽ നിയമാധികാരമുള്ളത്. കുത്തകാവകാശം ലഭിക്കാൻ താഴെ പറയുന്ന നിബന്ധനകൾ പാലിക്കണം.

- ഒരു പ്രക്രിയയുമായോ ഉൽപ്പന്നവുമായോ ബന്ധപ്പെട്ടിരിക്കണം
- പുതിയതായിരിക്കണം
- കണ്ടെത്തലിനായുള്ള ഒരു ഘട്ടം അടങ്ങിയിരിക്കണം
- വ്യാവസായികോപയോഗത്തിന് അനുയോജ്യമാകണം
- മറ്റുള്ളവർക്ക് ഉപദ്രവകരമായിത്തീരുന്നതൊന്നും വികസിപ്പിക്കരുത്.

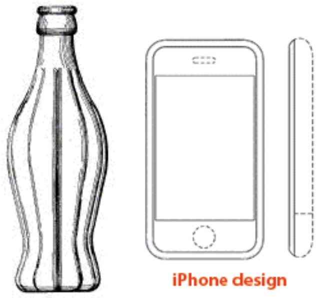
ചിത്രം 11.9: സിപ്പറിന്റെ ആദ്യ പേറ്റന്റ്

കുത്തകാവകാശ സംരക്ഷണമെന്നാൽ ഒരു ഉല്പന്നവും ഉടമസ്ഥന്റെ/യുടെ അനുവാദം കൂടാതെ ഉണ്ടാക്കാനോ ഉപയോഗിക്കാനോ വിതരണം ചെയ്യാനോ വില്പന നടത്താനോ പാടില്ലെന്ന നിയമ വ്യവസ്ഥയാണ്. ഒരു കണ്ടുപിടുത്തം മറ്റുള്ളവർ എങ്ങനെ ഉപയോഗിക്കണമെന്ന് തീരുമാനിക്കാനുള്ള അവകാശം കുത്തകാവകാശിക്ക് ലഭിക്കുന്നു. ഉടമ, തന്റെ കണ്ടുപിടുത്തത്തിന്റെ അവകാശം മറ്റൊരാൾക്ക് നൽകിയാൽ പിന്നീട് അയാൾക്കായിരിക്കും അതിന്റെ കുത്തകാവകാശം. കുത്തകാവകാശത്തിനുള്ള അപേക്ഷ ലഭിക്കുന്നത് മുതൽ 20 വർഷത്തേക്കാണ് ഇന്ത്യയിൽ ഇതിന്റെ കാലാവധി. കുത്തകാവകാശ കാലാവധി കഴിഞ്ഞാൽ സുരക്ഷിതത്വം അവസാനിക്കുകയും എല്ലാവർക്കും സൗജന്യമായി കണ്ടുപിടുത്തം ഉപയോഗിക്കുകയും ചെയ്യാം. ചിത്രം 11.9ൽ കുടുക്കിനുള്ള (zipper) കുത്തകാവകാശം വിശദമായി ചേർത്തിരിക്കുന്നു.

വ്യാപാരമുദ്ര (Trademark): ചില സാധനങ്ങളെ തിരിച്ചറിയുന്നതിനോ, ചില കമ്പനികളോ വ്യക്തികളോ നൽകുന്ന സേവനങ്ങളെ പ്രത്യേകമായി തിരിച്ചറിയുന്നതിനോ ഉപയോഗിക്കുന്ന പ്രത്യേക മുദ്രയാണിത്. ഇത് ഒരു ഉൽപ്പന്നത്തെയോ സേവനത്തെയോ തിരിച്ചറിയാൻ സഹായിക്കുന്ന പേരോ, അടയാള ചിഹ്നമോ, പ്രതീകമോ ആകാം. ഉൽപ്പന്നങ്ങളെ തിരിച്ചറിയുവാൻ വ്യാപാരമുദ്രയുടെ ഉടമയ്ക്ക് അത് ഉപയോഗിക്കാൻ അധികാരമുണ്ട്. ഉപയോക്താക്കൾക്ക് അത് തിരിച്ചറിയുവാനും ഉൽപ്പന്നത്തെയോ സേവനത്തെയോ വിലയ്ക്ക് വാങ്ങുവാനും സാധിക്കുന്നു. വ്യാപാരമുദ്ര രജിസ്റ്റർ ചെയ്തിരിക്കണം. 10 വർഷത്തേക്കാണ് ആദ്യ കാലാവധി. തുടർന്ന് ഇവ പുതുക്കാം. ഒരു വ്യക്തിയോ കമ്പനിയോ വ്യാപാരമുദ്ര ഉപയോഗിക്കുന്നുവോ എന്ന് കൺട്രോളർ ജനറൽ ഓഫ് പേറ്റന്റ് ഡിസൈൻസ് ആൻഡ് ട്രേഡ്മാർക്കിലൂടെ (http://ipindia-online.gov.in) അന്വേഷണം നടത്തി തിരിച്ചറിയാൻ സാധിക്കും. വ്യാപാരമുദ്ര രജിസ്ട്രേഷൻ അതാത് രാജ്യങ്ങൾക്ക് മാത്രം ബാധകമാണ്. ചിത്രം 11.10ൽ ഇന്ത്യയിലെ ചില പ്രശസ്തമായ വ്യാപാരമുദ്രകൾ കാണിച്ചിരിക്കുന്നു.

ചിത്രം 11.10
പ്രശസ്തമായ വ്യാപാര മുദ്രകൾ

വ്യാവസായിക രൂപരേഖ (Industrial designs): ഒരു വസ്തുവിന്റെ ആലങ്കാരികമോ സൗന്ദര്യപരമോ ആയ സ്വഭാവ സവിശേഷതകൾ പ്രതിപാദിക്കുന്നതാണ് വ്യാവസായിക രൂപരേഖ. ഒരു വസ്തുവിന്റെ ത്രിമാന സവിശേഷതകളായ ആകൃതി, ഉപരിതലം എന്നിവയും ദ്വിമാന സവിശേഷതകളായ മാതൃക, വരകൾ, നിറങ്ങൾ എന്നിവയുമൊക്കെ ഈ രൂപകല്പനയിൽ ഉൾപ്പെട്ടേക്കാം. പ്രവർത്തനപരമായ സവിശേഷതകൾക്കപ്പുറം വസ്തുക്കളുടെ ദൃശ്യപരമായ രൂപഘടനയെ വ്യാവസായിക രൂപരേഖ അവകാശം സംരക്ഷിക്കുന്നു. വൈവിധ്യമാർന്ന വ്യവസായ ഉൽപ്പന്നങ്ങളിലും കരകൗശല വസ്തുക്കളിലും, മെഡിക്കൽ ഉപകരണങ്ങൾ, ആഭരണങ്ങൾ, വാച്ചുകൾ, വാഹനങ്ങൾ, വസ്ത്ര രൂപകല്പനകൾ എന്നിവയിലും വ്യാവസായിക രൂപരേഖ ഉപയോഗിക്കപ്പെടുന്നു. കൊക്കക്കോള കുപ്പിയുടെയും, ഐ ഫോണിന്റെയും രജിസ്റ്റർ ചെയ്ത രൂപകല്പന ചിത്രം 11.11 ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 11.11: പ്രശസ്തമായ വ്യാവസായിക രൂപരേഖകൾ

ഭൗമശാസ്ത്രപരമായ സൂചകങ്ങൾ (Geographical indications): ഭൂമിശാസ്ത്രപരമായ ഉത്ഭവം, ഗുണങ്ങൾ അല്ലെങ്കിൽ ആ പ്രദേശത്തിന്റേതായ ഒരു സൽപ്പേര് ഉൽപ്പന്നങ്ങളിൽ ഉപയോഗിക്കുന്നതിനെയാണ് ഭൗമശാസ്ത്രപരമായ സൂചകങ്ങൾ എന്ന് പറയുന്നത്. കാർഷികോൽപ്പന്നങ്ങൾക്ക് അവയുടെ ഉൽപ്പാദന സ്ഥലത്തിന്റെ ഗുണങ്ങളോടൊപ്പം അവിടുത്തെ മണ്ണിന്റെയും കാലാവസ്ഥയുടെയും സ്വാധീനവും ഉണ്ടാകും. ഉത്ഭവ സ്ഥലം ഗ്രാമമോ നഗരമോ ചില മേഖലകളോ രാജ്യമോ ആകാം. ആറന്മുളക്കണ്ണാടിയും പാലക്കാടൻ മട്ട അരിയും കേരളവുമായി ബന്ധപ്പെട്ട ഭൗമശാസ്ത്രപരമായ സൂചകങ്ങൾക്കുള്ള ചില പ്രശസ്തമായ ഉൽപ്പന്നങ്ങൾ ആണ്.

ചിത്രം 11.12: കേരളവുമായി ബന്ധപ്പെട്ട ഭൗമശാസ്ത്രപരമായ സൂചകങ്ങൾ

b) പകർപ്പവകാശം (Copyright)

ഒരു നിശ്ചിത കാലത്തേക്ക് ഒരു സ്രഷ്ടാവിന് തന്റെ സൃഷ്ടിയിന്മേലുള്ള അവകാശമാണ് പകർപ്പവകാശം. പുസ്തകങ്ങൾ, സംഗീതം, ചിത്രങ്ങൾ, ശില്പങ്ങൾ, സിനിമ, പരസ്യം, കമ്പ്യൂട്ടർ സോഫ്റ്റ്‌വെയർ തുടങ്ങിയവയിലുള്ള സൃഷ്ടിപരമോ ബൗദ്ധികമോ കലാപരമോ ആയ പ്രവർത്തനങ്ങളുടെ വൈവിധ്യത്തെ ബാധകമാക്കിയുള്ള അവകാശമാണിത്. സൃഷ്ടികളുടെ പുനരുൽപ്പാദനം, പൊതുജനവുമായുള്ള ആശയവിനിമയം, അനുരൂപീകരണം, പരിഭാഷ എന്നിവയൊക്കെ ഈ അവകാശത്തിന്റെ പരിധിയിൽപ്പെടുന്നു. ഇന്ത്യയിൽ 1958 ജനുവരിയിലാണ് പകർപ്പവകാശനിയമം - 1957 നിലവിൽ വന്നത്. ഈ നിയമത്തിൽ അഞ്ച് പ്രാവശ്യം ഭേദഗതി വരുത്തി. 2012 ൽ പകർപ്പവകാശ സംരക്ഷണ നിയമത്തിൽ ചില സുപ്രധാന ഭേദഗതികൾ വരുത്തി. ഡിജിറ്റൽ പരിസ്ഥിതിയിലേക്കുള്ള ഈ നിയമത്തിന്റെ വ്യാപനം, ഇന്റർനെറ്റ് സേവന ദാതാക്കളുടെ ബാധ്യതകൾ, സംഗീത സംവിധായകർക്ക് റോയൽറ്റി ലഭിക്കുന്നതിനുള്ള അവകാശം ഉറപ്പാക്കൽ, അംഗപരിമിതർ ഏതെങ്കിലും തരത്തിലുള്ള ഉല്പന്നമോ സേവനമോ ഉപയോഗിക്കുന്നത് പകർപ്പവകാശ സംരക്ഷണ നിയമത്തിന്റെ പേരിൽ തടയാതിരിക്കൽ എന്നിവയാണ് ഭേദഗതികൾ.

ഇന്ത്യൻ പകർപ്പവകാശനിയമപ്രകാരം, ഒരു പ്രവൃത്തി അതിന്റെ സൃഷ്ടികർമത്തിൽ ത്തന്നെ പകർപ്പവകാശത്തിൽ സ്വമേധയാ സംരക്ഷിക്കപ്പെടുന്നു. പകർപ്പവകാശത്തിന് എഴുത്തുകാരന്റെ മരണശേഷം 60 വർഷം വരെ നിയമ സാധുത ഉണ്ടായിരിക്കും. പകർപ്പവകാശ രജിസ്ട്രേഷൻ സൃഷ്ടിപരമായ പ്രവർത്തനങ്ങൾക്ക് നിയമപരമായ പരിരക്ഷ നൽകുന്നു.

ചിത്രം 11.13: കോപ്പിറൈറ്റ് ഓഫീസ്, ഇന്ത്യയുടെ ലോഗോ

ഇത് സ്രഷ്ടാവിന് സൃഷ്ടിക്കുമേൽ നിയമപരമായ അവകാശം നൽകിക്കൊണ്ട് അത് ഒരു ബൗദ്ധികസ്വത്താക്കി മാറ്റുന്നു. പകർപ്പവകാശം ലഭിക്കുന്നതിന് രജിസ്റ്റർ ചെയ്യേണ്ട ആവശ്യമില്ല എന്ന വസ്തുത ശ്രദ്ധിക്കേണ്ടതാണ്. ഇന്ത്യയിലെ പകർപ്പവകാശ രജിസ്ട്രേഷൻ മാനവവിഭവവികസന മന്ത്രാലയത്തിന് കീഴിലുള്ള പകർപ്പവകാശ കാര്യാലയമാണ് കൈകാര്യം ചെയ്യുന്നത്. ഇന്ത്യൻ കാര്യാലയത്തിന്റെ ഓഫീസിന്റെ ലോഗോ ചിത്രം 11.13 ൽ നൽകിയിരിക്കുന്നു.

പട്ടിക 11.2 ൽ ബൗദ്ധികസ്വത്തിന്റെ രജിസ്ട്രേഷൻ സ്ഥിതി വ്യക്തമാക്കാൻ ഉപയോഗിക്കുന്ന വിവിധ ചിഹ്നങ്ങൾ കാണിച്ചിരിക്കുന്നു

ബൗദ്ധിക സ്വത്തവകാശം	
രജിസ്റ്റർ ചെയ്ത വ്യാപാരമുദ്ര	®
രജിസ്റ്റർ ചെയ്യാത്ത വ്യാപാരമുദ്ര	TM
പകർപ്പവകാശം	©
പകർപ്പവകാശ റെക്കോർഡിങ് ശബ്ദം	!

പട്ടിക 11.2: ബൗദ്ധിക സ്വത്തവകാശത്തിൽ ഉപയോഗിക്കുന്ന അടയാളങ്ങൾ

ഒരു സൃഷ്ടിയുടെ പകർപ്പവകാശ ഉടമകൾക്ക് ചുവടെ കൊടുത്തിരിക്കുന്ന പ്രവർത്തനങ്ങൾക്ക് അംഗീകാരം നൽകാനോ നിഷേധിക്കാനോ കഴിയും.

- അച്ചടിച്ച രൂപവും ശബ്ദ റെക്കോർഡിങ്ങും ഉൾപ്പടെ എല്ലാ രൂപത്തിലുമുള്ള അതിന്റെ പുനഃസൃഷ്ടി;
- പൊതുവായ കാര്യനിർവഹണവും പൊതുജനങ്ങൾക്കുള്ള ആശയവിനിമയവും;
- അതിന്റെ സംപ്രേഷണം;
- മറ്റു ഭാഷകളിലേക്കുള്ള വിവർത്തനം;
- നോവലിനെ സിനിമയുടെ തിരക്കഥയാക്കി മാറ്റുന്നതു പോലെയുള്ള അനുരൂപീകരണങ്ങൾ.

സ്രഷ്ടാക്കൾ തങ്ങളുടെ സൃഷ്ടികൾ സാമ്പത്തികനേട്ടത്തിനു വേണ്ടി വ്യക്തികൾക്കോ, കമ്പനികൾക്കോ വിൽക്കാറുണ്ട്.

കമ്പ്യൂട്ടർ സോഫ്റ്റ്‌വെറുകൾ (സോഴ്സ് കോഡ്, ഡാറ്റബേസ്, വെബ് സൈറ്റുകൾ എന്നിവ) ഒരു സാഹിത്യ സൃഷ്ടി പോലെ പകർപ്പവകാശത്തിന് വിധേയമാക്കാം. കമ്പ്യൂട്ടർ സോഫ്റ്റ്‌വെയർ പകർപ്പവകാശത്തിനു കീഴിൽ സംരക്ഷിക്കപ്പെട്ടിട്ടുണ്ടെങ്കിലും ഇപ്പോൾ അവ കുത്തകാവകാശം നേടിയെടുക്കുന്നു. സോഫ്റ്റ്‌വെയർ വികസനം ഒരു വ്യവസായമായി കണക്കാക്കപ്പെട്ടതും പകർപ്പവകാശവുമായി താരതമ്യം ചെയ്യുമ്പോൾ കുത്തകാവകാശം മികച്ച സംരക്ഷണം നൽകുന്നതും ഇതിനൊരു കാരണമാണ്. കുത്തകാവകാശ സംരക്ഷണത്തിന് ആവശ്യമായ മാനദണ്ഡങ്ങൾ കർശനമാണ് എന്നത് ശ്രദ്ധിക്കേണ്ടതാണ്. പട്ടിക 11.3 ൽ അവകാശങ്ങളും അനുബന്ധ ബൗദ്ധിക സ്വത്തുക്കളും പ്രദർശിപ്പിച്ചിരിക്കുന്നു. ബൗദ്ധിക സ്വത്തവകാശത്തിന്റെ രജിസ്ട്രേഷനിലെ വ്യത്യാസങ്ങൾ പട്ടിക 11.4 ൽ കൊടുത്തിരിക്കുന്നു.

ബുദ്ധി	വസ്തു	അവകാശം
ആശയം	കണ്ടുപിടുത്തം/പുതുമ	കുത്തകാവകാശം
ആശയം	മേന്മ + സവിശേഷത	വ്യാപാരമുദ്ര
ആശയം	രൂപം	രൂപരേഖ
സാധനം	ഭൂമിശാസ്ത്രപരമായ ഉത്ഭവം	ഭൂമിശാസ്ത്രപരമായ സൂചകങ്ങൾ
ആശയം	ഭാവനകൾ	പകർപ്പവകാശം

പട്ടിക 11.3: ബൗദ്ധിക സ്വത്തവകാശത്തിന്റെ അവകാശങ്ങൾ

	കുത്തകാവകാശം	വ്യാപാരമുദ്ര	പകർപ്പവകാശം
പരാമർശിക്കുന്നു	ഉൽപന്നം, പ്രക്രിയ	പേര്, അടയാള ചിഹ്നം, അടയാളങ്ങൾ	സൃഷ്ടിപരമായ, ബുദ്ധിപരമായ അല്ലെങ്കിൽ കലാപരമായ പ്രവർത്തന രീതികൾ
രജിസ്ട്രേഷൻ	ആവശ്യമാണ്	ആവശ്യമാണ്	യാന്ത്രികമായി രജിസ്റ്റർ ചെയ്യാൻ കഴിയും
കാലാവധി	20 വർഷം	10 വർഷം	ഒടുവിലത്തെ എഴുത്തുകാരന്റെ മരണശേഷം 60 വർഷം വരെ
പുനരുൽപാദനം	ഇല്ല	ഉണ്ട്	

പട്ടിക 11.4: ബൗദ്ധിക സ്വത്തവകാശത്തിന്റെ രജിസ്ട്രേഷനിലെ വ്യത്യാസങ്ങൾ

11.5.2 അവകാശ കയ്യേറ്റം (Infringement)

അനുമതി കൂടാതെ കുത്തകാവകാശമോ, പകർപ്പവകാശമോ, വ്യാപാരമുദ്രയോ ഉപയോഗിക്കുന്നത് ബൗദ്ധിക സ്വത്തിന്മേലുള്ള കയ്യേറ്റമാണ്. ബൗദ്ധിക സ്വത്തിന്റെ സ്വഭാവം, അധികാര പരിധി (രാജ്യങ്ങൾ തമ്മിൽ), പ്രവൃത്തിയുടെ സ്വഭാവം എന്നിവയുടെ അടിസ്ഥാനത്തിൽ ഇത് പൗരാവകാശ നിയമത്തിന്റെയോ ക്രിമിനൽ നിയമത്തിന്റെയോ ലംഘനമായിരിക്കാം.

കുത്തകാവകാശമുള്ള ആളുടെ (patent holder) അനുമതിയില്ലാതെ കുത്തകാവകാശമുള്ള കണ്ടുപിടുത്തം ഉപയോഗിക്കുകയോ വിൽക്കുകയോ ചെയ്യുന്നത് കുത്തകാവകാശ കയ്യേറ്റമാണ്. ഗവേഷണാവശ്യങ്ങൾക്ക് വേണ്ടി കുത്തകാവകാശമുള്ള കണ്ടുപിടുത്തങ്ങൾ ഉപയോഗിക്കുന്നതിന് ധാരാളം രാജ്യങ്ങൾ അനുമതി നൽകുന്നു. മൊബൈൽ ഫോൺ സാങ്കേതികവിദ്യയിൽ ആപ്പിളും സാംസങ്ങും തമ്മിലുള്ള നിയമപരമായ തർക്കം കുത്തകാവകാശ കയ്യേറ്റത്തിന് ഒരുദാഹരണമാണ്.

ഒരാളുടെ വ്യാപാരമുദ്രയോട് സാമ്യമുള്ള വ്യാപാരമുദ്ര മറ്റൊരാൾ ഉപയോഗിക്കലാണ് വ്യാപാരമുദ്ര കയ്യേറ്റം. ഇവിടെ രണ്ടു കക്ഷികളും ഉപയോഗിക്കുന്നത് ഒരേ ഉൽപ്പന്നം അഥവാ സേവനം തന്നെയായിരിക്കും. നിയമപരമായ ആനുകൂല്യം ലഭിക്കാൻ വ്യാപാരമുദ്ര രജിസ്റ്റർ ചെയ്യണം. 'HORLIKS' എന്ന വ്യാപാര നാമം ഉപയോഗിച്ച് ഒരു ചെറിയ കമ്പനി മിഠായികൾ നിർമിച്ചത് 'HORLICKS' എന്ന വ്യാപാര നാമം ഉപയോഗിച്ചിരുന്ന കമ്പനിയുടെ വ്യാപാര മുദ്രാവകാശത്തിൻമേലുള്ള കയ്യേറ്റത്തിനുദാഹരണമാണ്.

പകർപ്പവകാശ കയ്യേറ്റം എന്നത് സൃഷ്ടാവിന്റെ അനുമതിയില്ലാതെ അയാളുടെ കൃതി പുനഃസൃഷ്ടിക്കുകയോ വിതരണം ചെയ്യുകയോ പ്രദർശിപ്പിക്കുകയോ മാറ്റുകയോ പരിഭാഷപ്പെടുത്തുകയോ ആണ്. ഇതിനെ സാഹിത്യ മോഷണം എന്ന് വിളിക്കുന്നു.

സോഫ്റ്റ്‌വെയറുകൾ നിർമ്മാതാക്കളുടെ അനുവാദമോ, അംഗീകാരമോ കൂടാതെ മോഷ്ടിച്ച് മറ്റു കമ്പ്യൂട്ടറുകളിൽ അനധികൃതമായി കോപ്പി ചെയ്യുന്നതിനെ സോഫ്റ്റ്‌വെയർ പൈറസി (മോഷണം) എന്ന് പറയുന്നു. ഒരു സംഗീതശകലം ഗാനരചയിതാവിന്റെയോ, കലാകാരന്റെയോ, ഗാനത്തിന്റെ പകർപ്പവകാശമുള്ള കമ്പനിയുടെയോ അനുവാദമില്ലാതെ കോപ്പി ചെയ്യുകയോ വിതരണം ചെയ്യുകയോ ചെയ്യുന്ന പ്രവർത്തനത്തെ മ്യൂസിക് പൈറസി എന്ന് പറയുന്നു.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. മനസ്സിന്റെ സൃഷ്ടി ________ എന്നറിയപ്പെടുന്നു.
2. W I P O എന്നതിന്റെ പൂർണരൂപം എഴുതുക.
3. ബൗദ്ധിക സ്വത്തവകാശത്തെ ________ എന്നും, ________ എന്നും രണ്ടായി തരംതിരിക്കുന്നു.
4. കുത്തകാവകാശം എന്നത് ________ പ്രത്യേകമായി നൽകിയ അവകാശമാണ്.
5. ________ എന്നത് ഒരു ഉൽപ്പന്നമോ സേവനമോ തിരിച്ചറിയാനുള്ള അടയാളമാണ്
6. വ്യാവസായിക രൂപരേഖ എന്നാൽ എന്താണ്?
7. ഭൗമശാസ്ത്രപരമായ സൂചകങ്ങളുടെ പ്രാധാന്യമെന്ത്?

11.5.3 സൈബർ ഇടം (Cyber space)

ഇന്റർനെറ്റുമായി ബന്ധിപ്പിച്ചിട്ടുള്ള കംപ്യൂട്ടർ സംവിധാനങ്ങൾ സൃഷ്ടിച്ച ഒരു അയഥാർഥ പരിതസ്ഥിതിയാണ് സൈബർ ഇടം. ഒരു പക്ഷേ, ഒരു സ്ഥലത്ത് പല കാര്യങ്ങളും ചെയ്യാൻ കഴിയുന്ന ഒരു സമ്പ്രദായത്തെ സൂചിപ്പിക്കാനാണ് ഈ പദം ഉപയോഗിക്കുന്നത്. സൈബർ ഇടം നമ്മുടെ ജീവിതത്തിൽ സ്വാധീനം ചെലുത്തുന്ന വിവിധ അവസരങ്ങൾ നമുക്ക് ചർച്ച ചെയ്യാം. ആശയവിനിമയത്തിനായി മുൻകാലങ്ങളിൽ നിലവിലിരുന്ന പോസ്റ്റൽ സേവനത്തിന്റെ സ്ഥാനത്ത് ഇന്ന് ഇ-മെയിൽ ഏറെ സ്വീകാര്യതയും നിയമസാധുതയും നേടിയിരിക്കുന്നു. ഹയർ സെക്കന്ററി വിദ്യാഭ്യാസ വകുപ്പിൽ വിദ്യാർഥിപ്രവേശനം, പരീക്ഷ, നാഷണൽ സർവീസ് സ്കീം, ഭരണനിർവഹണം എന്നിവയെല്ലാം വെബ്സൈറ്റിന്റെയും ഇ-മെയിലുകളുടെയും സഹായത്തോടെയാണ് നടക്കുന്നത്. സാധാരണ ജനങ്ങളിൽ ഇന്റർനെറ്റ് പ്രശസ്തമായ ആശയവിനിമയ മാധ്യമമായി മാറിയിരിക്കുന്നു.

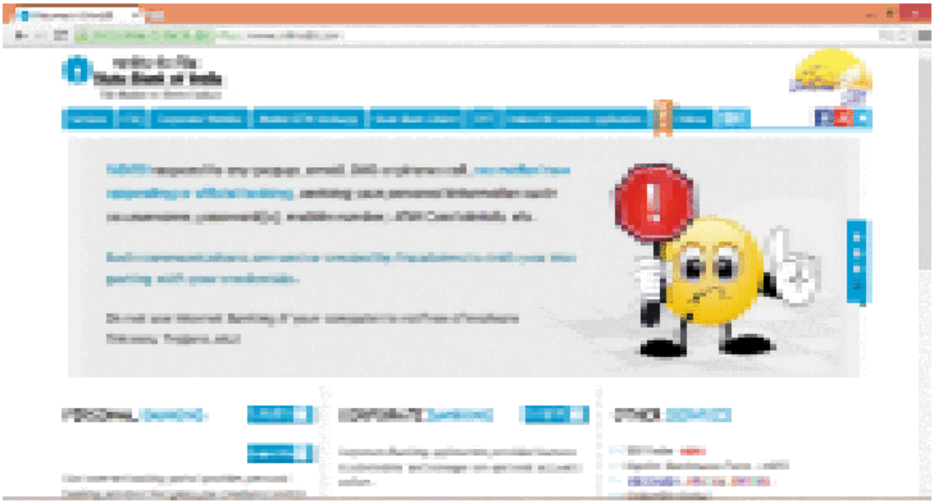
ഇന്ത്യയിലെ ഒരു മെട്രോ നഗരത്തിൽ 2012 ഡിസംബറിൽ ഓടിക്കൊണ്ടിരുന്ന ബസ്സിനുള്ളിൽ വെച്ച് ഒരു പെൺകുട്ടി ശാരീരിക പീഡനത്തിനിരയായി. കുറച്ചു ദിവസങ്ങൾക്കു ശേഷം ആ കുട്ടി മരണപ്പെട്ടു. ജീവിതത്തിന്റെ വിവിധ മേഖലകളിലുള്ളവർ സാമൂഹിക മാധ്യമങ്ങൾ വഴി ഇതിനെതിരെ ശക്തമായി പ്രതികരിച്ചു. ഇതിന്റെ ഫലമായി ലോക്സഭ 2013ൽ ലൈംഗികാതിക്രമങ്ങൾക്ക് എതിരെ The Criminal Law (Amendent) Act 2013 എന്ന ശക്തമായ നിയമം കൊണ്ടുവന്നു. ഇന്റർനെറ്റ് ഉപയോഗിച്ച് സാമൂഹിക മാധ്യമങ്ങൾ വഴി സംഘടിപ്പിച്ച ഈ പ്രതിഷേധത്തിന് സമൂഹത്തെ സ്വാധീനിക്കാനായി.

പണ്ട് കടയിൽ പോയി സാധനങ്ങൾ വാങ്ങിയിരുന്നതിന് പകരം ഇന്റർനെറ്റ് വ്യാപാര സൈറ്റുകൾ വഴി ഇന്ന് സാധനങ്ങൾ വാങ്ങാവുന്നതാണ്. ഉൽപ്പന്നങ്ങളെക്കുറിച്ചുള്ള വിവരങ്ങൾ ഇത്തരം ഇ-വാണിജ്യ സൈറ്റുകൾ നൽകുന്നു. അനായാസമായി സാധനങ്ങൾ

വാങ്ങുന്നതിനുള്ള സൗകര്യം ഇതുവഴി സാധ്യമാകുന്നു. ഇന്റർനെറ്റ് ബാങ്കിങ്, ക്രെഡിറ്റ് കാർഡ്, ഡെബിറ്റ് കാർഡ് എന്നിവ വഴി സാമ്പത്തിക വിനിമയങ്ങളും ലളിതമായി. എല്ലാ സേവനങ്ങളും ഓൺലൈനാക്കി മാറ്റുവാൻ ഇന്റർനെറ്റിന്റെ സഹായത്തോടെ ഇപ്പോൾ സാധിക്കുന്നു. ചിത്രം 11.14 ൽ ചില ഇ-വാണിജ്യ വെബ് സൈറ്റുകൾ കാണിച്ചിരിക്കുന്നു.

ചിത്രം 11.14: ഇ - വാണിജ്യ വെബ്സൈറ്റിന് ഒരു ഉദാഹരണം

ഇന്ന് എല്ലാ ബാങ്കുകളും അവരുടെ ഉപയോക്താക്കൾക്ക് ഇന്റർനെറ്റ് ബാങ്കിങ് സൗകര്യം ഏർപ്പെടുത്തിയിരിക്കുന്നു. സാമ്പത്തിക വിനിമയം നടത്തുവാനും, ടെലിഫോൺ ബിൽ, വൈദ്യുത ബിൽ, ഓൺലൈനായി വ്യാപാരം നടത്തൽ, ട്രെയിൻ ടിക്കറ്റ് ബുക്കിംഗ്, സിനിമ ടിക്കറ്റ് ബുക്കിംഗ് എന്നിവയ്ക്കെല്ലാം ഇന്റർനെറ്റ് ബാങ്കിങ് സൗകര്യം ഉപയോഗപ്പെടുത്തുന്നു. ഈ പ്രവർത്തനങ്ങൾക്ക് ബന്ധപ്പെട്ട ഓഫീസുകൾ സന്ദർശിക്കുന്നതുമായി താരതമ്യം ചെയ്യുമ്പോൾ ഇന്റർനെറ്റ് ബാങ്കിങ്ങിലൂടെ സമയവും അധ്വാനവും ലാഭിക്കാൻ കഴിയുന്നു. അങ്ങനെ ബാങ്ക് ഇടപാടുകളും ഇന്റർനെറ്റിലൂടെ ലഭ്യമാകുന്ന ജനപ്രിയ സേവനമായി മാറി. ഒരു ഇന്റർനെറ്റ് ബാങ്കിങ് വെബ്സൈറ്റ് ചിത്രം 11.15 ൽ കാണിച്ചിരിക്കുന്നു.



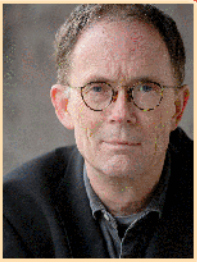
ചിത്രം 11.15: ഇന്റർനെറ്റ് ബാങ്കിംഗ് വെബ്സൈറ്റ്

കാലക്രമത്തിൽ കൂടുതൽ സേവനങ്ങൾ ഓൺലൈനായി മാറിക്കൊണ്ടിരിക്കുന്നു. വെബ്ബിലെ വിവിധ സേവനങ്ങൾ ലഭിക്കുന്നതിനായി ധാരാളം ആളുകൾ ഗണ്യമായ സമയം ഇന്റർനെറ്റിനു മുമ്പിൽ വിനിയോഗിക്കുന്നു. ഇന്റർനെറ്റ് പലപ്പോഴും സൈബർ ഇടം എന്നും

ചിത്രം 11.16: സൈബർ ഇടത്തിന്റെ പ്രതീകാത്മക അവതരണം

അറിയപ്പെടുന്നു. സൈബർ ഇടത്തെ പ്രതീകാത്മകമായി ചിത്രം 11.16 ൽ കാണിച്ചിരിക്കുന്നു.

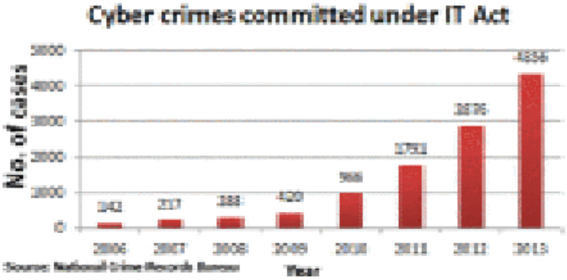
കമ്പ്യൂട്ടർ നെറ്റ്‌വർക്കുകളിലൂടെ ആശയവിനിമയം നടത്തുന്ന ഒരു പ്രതീതി ലോകമാണ് സൈബർ ഇടം. വിവരങ്ങളുടെ പ്രധാന പാതയായ ഇവിടെ വ്യക്തിപരമായ വിവരങ്ങൾ ശേഖരിക്കുകയും, ആശയങ്ങൾ വിനിമയം നടത്തുകയും, പരസ്പരം സംവാദങ്ങളിൽ ഏർപ്പെടുകയും, സാമൂഹിക സഹായം നൽകുകയും, വ്യാപാരം നടത്തുകയും, വിനോദങ്ങളിൽ ഏർപ്പെടുകയും, ചർച്ചകളിൽ മുഴുകുകയും ചെയ്യുന്നു. സാമൂഹിക ഇടപെടലുകൾക്ക് ആധിപത്യമുള്ള ഒരു സ്ഥലമാണ് സൈബർ ഇടം. ഇഷ്ടാനുസരണം ഏതു പ്രവർത്തനത്തിൽ ഏർപ്പെടാനും എന്തും സ്വതന്ത്രമായി പ്രകടിപ്പിക്കാനുമുള്ള ഒരു അനിയന്ത്രിത ഇലക്ട്രോണിക് മാധ്യമമായി ചിലർ സൈബർ ഇടത്തെ കാണുന്നു. അത്തരം പ്രവൃത്തികൾ മറ്റുള്ളവർക്ക് പലപ്പോഴും ബുദ്ധിമുട്ടുകൾ സൃഷ്ടിക്കുകയും തെറ്റിലേക്ക് നയിക്കുകയും ചെയ്യുന്നു. അതിനാൽ ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്ന ഒരു വ്യക്തി മറ്റുള്ളവർക്ക് ഗുണകരമാകുന്ന വിധത്തിൽ നിയമങ്ങളും സദാചാരമൂല്യങ്ങളും പാലിക്കേണ്ടതാണ്. നിയന്ത്രണരഹിതമായ ഈ ഇടം കുറ്റവാളികൾക്കും ലഭ്യമാണ്. സൈബർ ഇടത്തിലെ ആശയവിനിമയം, സാമ്പത്തിക ഇടപാടുകൾ മുതലായവ ദിനംപ്രതി വർദ്ധിക്കുന്നു. അതിനാൽ കുറ്റകൃത്യങ്ങൾക്കുള്ള മാധ്യമമായും ഇത് മാറുന്നു. അതുകൊണ്ട് തന്നെ സൈബർ ഇടത്തിന്റെ സുരക്ഷ ഗൗരവമേറിയ പരിഗണനവിഷയമായി മാറിയിരിക്കുന്നു.



സൈബർ ഇടം എന്ന പദം സൃഷ്ടിച്ചത് കനേഡിയൻ ശാസ്ത്രനോവൽ എഴുത്തുകാരനായ വില്യം ഗിബ്സൺ ആണ്. അദ്ദേഹത്തിന്റെ ഹ്രസ്വ കഥയായ ബേണിങ് ക്രോമിൽ 1982ലാണ് ഇത് പരാമർശിച്ചിരിക്കുന്നത്. യഥാർഥ ലോകത്തിൽ ഇല്ലാത്ത സംഭവങ്ങൾ അല്ലെങ്കിൽ ഇടപാടുകൾ നടക്കുന്ന ഒരു ലോകത്തെ പ്രതിനിധാനം ചെയ്യാൻ അദ്ദേഹം അത് ഉപയോഗിച്ചു. സൈബർ ഇടം എന്നത് പൊതുവായ ഒരു മായാലോകമാണ് എന്ന് അദ്ദേഹം വിശ്വസിച്ചു. മായാദൃശ്യം എന്നത് സൈബർ ഇടത്തിന്റെ ഉള്ളിലുള്ള ജീവന്റെ സ്വഭാവമാണ് എന്ന് ഗിബ്സൺ പറയുന്നു.

11.5.4 സൈബർ കുറ്റകൃത്യങ്ങൾ (Cyber crimes)

വിനോദത്തിനും വ്യാപാരത്തിനും ആശയവിനിമയത്തിനും വിദ്യാഭ്യാസാവശ്യങ്ങൾക്കുമായി ഇന്റർനെറ്റ് പുതിയ വാതായനങ്ങൾ തുറക്കുന്നു എന്ന് നമുക്കറിയാം. അതേസമയം ചിലർ നിയമവിരുദ്ധമായ പ്രവർത്തനങ്ങൾക്കും ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്നു.



ചിത്രം 11.17: ഐ.ടി. നിയമത്തിന്റെ പരിധിയ്ക്കുള്ളിൽപ്പെട്ട സൈബർ കുറ്റകൃത്യങ്ങൾ

ഓൺലൈൻ ബാങ്കിങ്ങിനും മറ്റ് സാമ്പത്തിക ഇടപാടുകൾക്കുമായി സ്മാർട്ട് ഫോണുകളിലൂടെയും ടാബ്‌ലറ്റുകളിലൂടെയുമുള്ള വർധിച്ച ഇന്റർനെറ്റ് ഉപയോഗം അപകട സാധ്യതയും വർധിപ്പിക്കുന്നു. ഉയർന്ന തോതിലുള്ള ഇന്റർനെറ്റ് നുഴഞ്ഞു കയറ്റവും ഓൺലൈൻ

ബാങ്കിങ് സംവിധാനങ്ങളും സൈബർ അക്രമികൾക്ക് ഉപദ്രവകാരികളായ സോഫ്റ്റ്‌വെയർ (മാൽവെയർ) ഉപയോഗിച്ചോ നിയമവിരുദ്ധമായ ഹാക്കിങ് നടത്തിയോ ഓൺലൈൻ സാമ്പത്തിക ഇടപാടുകളെ ഉന്നം വയ്ക്കാൻ അവസരം ഒരുക്കുന്നു.

നാഷണൽ ക്രൈം റെക്കോർഡ്സ് ബ്യൂറോയുടെ കണക്കു പ്രകാരം കഴിഞ്ഞ വർഷങ്ങളിൽ നടന്ന സൈബർകുറ്റകൃത്യങ്ങളുടെ സ്ഥിതിവിവരക്കണക്ക് ചിത്രം 11.17 ലെ ഗ്രാഫിൽ കാണിച്ചിരിക്കുന്നു. ഇന്ത്യയിൽ സൈബർ കുറ്റകൃത്യങ്ങൾ ഗണ്യമായി വർധിക്കുന്നുവെന്ന് ഗ്രാഫിൽ കാണാൻ സാധിക്കും.

കമ്പ്യൂട്ടറുകളും കമ്പ്യൂട്ടർ നെറ്റ്‌വർക്കുകളും ഒരു ടൂളായോ അവയെ ഒരു ലക്ഷ്യമായോ അതുമല്ലെങ്കിൽ അവ ഒരു കുറ്റകൃത്യം നടത്തുന്നതിനുള്ള സ്ഥലമായോ ഉപയോഗിക്കുന്നതിനെയാണ് സൈബർ കുറ്റകൃത്യം എന്ന് നിർവചിച്ചിരിക്കുന്നത്. സൈബർ കുറ്റകൃത്യത്തിന്റെ ഒരു സുപ്രധാന ഘടകം അതിന്റെ പ്രാദേശികമല്ലാത്ത സ്വഭാവമാണ്. വളരെ അകലങ്ങളിലായി വേർതിരിക്കപ്പെട്ടിട്ടുള്ള നിയമവ്യവസ്ഥകളിൽ ഒരു കുറ്റകൃത്യം സംഭവിക്കാം. ഒരു ആക്രമണകാരിക്ക് ഒരു രാജ്യത്ത് നിന്നും മറ്റൊരു രാജ്യത്തെ ലക്ഷ്യമാക്കി ആക്രമണം നടത്തുവാൻ കഴിയും. അതിനാൽ വിവിധ രാജ്യങ്ങളിലെ അന്വേഷണ സംഘവും നീതിന്യായക്കോടതികളും കൈകോർക്കേണ്ടത് ആവശ്യമാണ്. ഇന്റർനെറ്റിന്റെ അജ്ഞാത സ്വഭാവം കാരണം ആളുകൾ വിവിധ കുറ്റകൃത്യങ്ങളിൽ ഏർപ്പെടാൻ സാധ്യതയുണ്ട്. ആളുകൾ അറിഞ്ഞോ അറിയാതെയോ സൈബർ കുറ്റകൃത്യങ്ങളിൽ ഏർപ്പെടുന്നു.

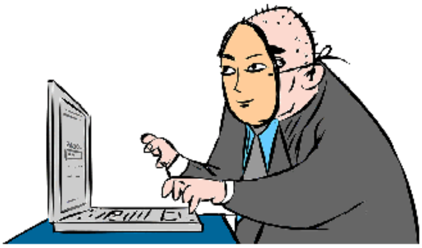
ഫിഷിങ്, ഹാക്കിങ്, സേവന ആക്രമണങ്ങളുടെ നിഷേധം എന്നിവ സൈബർ കുറ്റകൃത്യങ്ങളിൽ ഉൾപ്പെടുന്നു. ഇവ പതിനൊന്നാം ക്ലാസ്സിലെ 9-ാം അധ്യായത്തിൽ നാം പഠിച്ചിട്ടുണ്ട്. അനുവാദമില്ലാതെയുള്ള കമ്പ്യൂട്ടർ ഉപയോഗം, ക്രെഡിറ്റ് കാർഡ് വഞ്ചന, നിയമാനുസൃതമല്ലാതെയുള്ള ഡൗൺലോഡ്, അശ്ലീല ചിത്രങ്ങൾ, സൈബർ ഭീകരവാദം, വൈറസുകൾ, ഉപയോഗശൂന്യമായ ഇ-മെയിലുകൾ (സ്പാമുകൾ) എന്നിവയൊക്കെ കമ്പ്യൂട്ടർ കുറ്റകൃത്യങ്ങളിൽപ്പെടുന്ന പ്രവർത്തനങ്ങളാണ്.

സൈബർ കുറ്റകൃത്യങ്ങളെ പ്രധാനമായും മൂന്നായി തരം തിരിച്ചിരിക്കുന്നു- വ്യക്തികൾക്കെതിരെയുള്ള കുറ്റകൃത്യങ്ങൾ, സ്വത്തിനെതിരെയുള്ള കുറ്റകൃത്യങ്ങൾ, ഭരണകൂടത്തിനെതിരെയുള്ള കുറ്റകൃത്യങ്ങൾ.

a) വ്യക്തികൾക്കെതിരെയുള്ള സൈബർ കുറ്റകൃത്യങ്ങൾ (Cyber crimes against individuals)

ബ്രോഡ്ബാൻഡ് ഇന്റർനെറ്റിന്റെ പ്രചാരം ദൈനംദിന പ്രവത്തനങ്ങളിൽ ഇന്റർനെറ്റിനുള്ള സ്വാധീനം വർധിപ്പിച്ചു. ഇത് ഓൺലൈൻ കുറ്റകൃത്യങ്ങളുടെ അപകട സാധ്യതയും വർധിപ്പിച്ചു. ഒരു വ്യക്തിക്ക് ശാരീരികമോ മാനസികമോ ആയ ബുദ്ധിമുട്ടുകൾ ഉണ്ടാക്കുന്ന മറ്റൊരാളുടെ സൈബർ ഇടത്തിലെ പ്രവൃത്തി സൈബർ കുറ്റകൃത്യമായി കണക്കാക്കുന്നു. ഒരാളുടെ സ്വകാര്യതയിൽ ഇടപെടുക, ആൾമാറാട്ടം നടത്തുക, മറ്റൊരാളുടെ പേരിൽ ഇന്റർനെറ്റ് ഉപയോഗിക്കുക, മറ്റൊരാളെ ശല്യപ്പെടുത്തുക എന്നിവയാണ് വ്യക്തികൾക്കെതിരെയുള്ള സൈബർ കുറ്റകൃത്യങ്ങൾ.

(i) സ്വകാര്യ വിവരങ്ങളുടെ മോഷണം (Identity theft): ഒരു വ്യക്തിയെ തിരിച്ചറിയാനുള്ള പേര്, ക്രെഡിറ്റ് കാർഡ്, ആധാർ നമ്പർ തുടങ്ങിയവ അയാളുടെ സമ്മതം കൂടാതെ മറ്റൊരാൾ തട്ടിപ്പ് നടത്തുന്നതിനും കുറ്റകൃത്യങ്ങൾക്കുമായി ഉപയോഗിക്കുന്നതാണ് സ്വകാര്യ വിവരങ്ങളുടെ മോഷണം എന്നറിയപ്പെടുന്നത്. ഒരാളുടെ ബാങ്ക് അക്കൗണ്ടുകളും



സാമൂഹിക മാധ്യമ അക്കൗണ്ടുകളും ഉപയോഗിക്കുന്നതിനായി അയാളുടെ വ്യക്തിപരമായ തിരിച്ചറിയൽ വിവരങ്ങൾ മോഷ്ടിക്കുന്ന പ്രവൃത്തിയാണിത്. ഇത് അയാളെ അപകീർത്തിപ്പെടുത്തുന്നതിനോ അയാളുടെ അക്കൗണ്ടിൽ നിന്ന് പണം മോഷ്ടിക്കുന്നതിനോ സാധനങ്ങൾക്ക് വില നൽകുന്നതിനോ, ധനകാര്യ സ്ഥാപനങ്ങളിൽനിന്ന് പണം ലഭിക്കുന്നതിനോ മറ്റെന്തെങ്കിലും ആനുകൂല്യങ്ങൾ ലഭിക്കുന്നതിനോ ആകാം.

ബാങ്ക് അക്കൗണ്ടിൽനിന്ന് പണം പിൻവലിക്കുന്നതിനു മുമ്പ് മോഷ്ടാവ് യഥാർഥത്തിലുള്ള ഉടമ പണമിടപാട് വിവരങ്ങൾ അറിയാതിരിക്കുന്നതിനായി മെയിലിങ് മേൽവിലാസം മാറ്റും. അയാളുടെ വ്യക്തിവിവരങ്ങൾ പുതിയ അക്കൗണ്ടുകളോ ക്രെഡിറ്റ് കാർഡ് അക്കൗണ്ടുകൾ തുടങ്ങുന്നതിനോ മൊബൈൽ ഫോൺ കണക്ഷൻ എടുക്കുന്നതിനോ മറ്റോ ഉപയോഗിച്ചേക്കാം.

(ii) ശല്യപ്പെടുത്തൽ (Harassment): വിശിഷ്ട വ്യക്തികളെപ്പറ്റി ചാറ്റ് റൂം, സാമൂഹിക മാധ്യമങ്ങൾ, ഇ-മെയിൽ തുടങ്ങിയവയിൽ അയാളുടെ ലിംഗഭേദം, വംശം, മതം, ദേശീയത എന്നിവയെ ഉന്നം വച്ച് പരിഹാസ്യമായ അഭിപ്രായങ്ങൾ ഇടുന്നത് ഇത്തരം ശല്യപ്പെടുത്തലാണ്. അശ്ലീല പദങ്ങളുപയോഗിക്കുക, അസ്സാന്മാർഗികമോ നിയമപരമല്ലാത്തതോ ആയ പ്രവൃത്തികളിലൂടെ ഭീഷണിപ്പെടുത്തുക എന്നിവയും ശല്യപ്പെടുത്തലിൽ ഉൾപ്പെടുന്നു.
ഇന്റർനെറ്റുപയോഗിച്ച് ഒരാളെ ശല്യപ്പെടുത്തുന്നതിന് സൈബർ സ്റ്റോക്കിങ് (stalking) എന്ന് പറയുന്നു. പ്രശസ്തരായ ചിലർ തങ്ങളുടെ പേരിൽ വന്നിട്ടുള്ള ഫേസ് ബുക്ക് അക്കൗണ്ടുകൾ വ്യാജമാണെന്ന് പറയുന്നത് നാം വായിച്ചിട്ടുണ്ട്. ഇത്തരം അക്കൗണ്ടുകളിൽ അപകീർത്തികരമായ ചിത്രങ്ങളോ വിവരങ്ങളോ വന്നത് കൊണ്ടാണിങ്ങനെ പറയുന്നത്. കുറ്റവാസനയുള്ള ചിലർ ബഹുമാന്യരായ വ്യക്തികളുടെ ഫോട്ടോയും വ്യക്തിവിവരങ്ങളും ഉപയോഗിച്ച് പ്രൊഫൈൽ നിർമ്മിക്കും. ചിലർ അപഹാസ്യമായ ഇ-മെയിലുകൾ, ഫേസ് ബുക്ക് പോസ്റ്റിങ്ങുകൾ എന്നിവ അയച്ച് ആളുകളെ അപകീർത്തിപ്പെടുത്തും. ഇവയെല്ലാം ഇത്തരം ശല്യപ്പെടുത്തലുകളാണ്. ദുരാരോപണങ്ങൾ, ഭീഷണികൾ, നിരീക്ഷിക്കലുകൾ, സ്വകാര്യവിവരങ്ങളുടെ മോഷണം, വിവരങ്ങൾ നശിപ്പിക്കൽ ഇവയെല്ലാം സൈബർ സ്റ്റോക്കിങ്ങിന്റെ പ്രത്യേകതകളാണ്. പ്രായപൂർത്തിയാകാത്ത കുട്ടികളെ ലൈംഗികമായി ചൂഷണം ചെയ്യുന്നതും സൈബർ സ്റ്റോക്കിങ്ങിൽ ഉൾപ്പെടും. ഇത് സൗഹൃദങ്ങളും തൊഴിലുകളും പ്രതിച്ഛായയും ആത്മവിശ്വാസവും നഷ്ടപ്പെടുത്തും.

(iii) ആൾമാറാട്ടവും വഞ്ചനയും (Impersonation and cheating): ഒരു വ്യക്തി മറ്റൊരാളായി അഭിനയിച്ച് ആളുകളെ ദ്രോഹിക്കുന്നതിനെ ആൾമാറാട്ടം എന്ന് പറയുന്നു. ഇന്റർനെറ്റിന്റെ അജ്ഞാതസ്വഭാവം ഉപയോഗിച്ച് ചിലർ ഓൺലൈൻ ആൾമാറാട്ടം നടത്തുന്നു.

വിദൂര രാജ്യങ്ങളിൽ നിന്നും ഇന്ത്യയിലേക്ക് വൻതുകകൾ മാറ്റുന്നതിന് നമ്മുടെ സഹായം ആവശ്യപ്പെട്ടു കൊണ്ടുള്ള ഇ-മെയിലുകൾ നമുക്ക് ലഭിക്കാറുണ്ട്. ഈ പണം ഏതെങ്കിലും വസ്തു, സ്വർണം എന്നിവ വിൽക്കാനുള്ള ആസ്തിയാണെന്ന് അവർ പറയും. നിയമനടപടികൾ പൂർത്തിയാക്കി ഈ ആസ്തി വിൽക്കുന്നതിന് അവർക്ക് കുറച്ചു പണം ആവശ്യമുണ്ടെന്നും അതിൽ ഒരു പങ്ക് നാം വഹിക്കേണ്ടതുണ്ട് എന്നും, പകരം ആസ്തിയുടെ

50% വരെ തരാമെന്നും വാഗ്ദാനം നൽകുന്നു. ഇതിനു വേണ്ടി നാം അയയ്ക്കുന്ന പണം സ്വീകരിച്ച്, ചില പ്രശ്നങ്ങൾ കൂടി തീർക്കാനുണ്ടെന്നു പറഞ്ഞ് കൂടുതൽ പണം ആവശ്യപ്പെടുന്നു. ഇതിന് ഇരയാകുന്നവർക്ക് വൻ തുകകൾ നഷ്ടപ്പെടുന്നു. ഇതുപോലെ വ്യത്യസ്ത കഥകളുമായി ധാരാളം മെയിലുകൾ വന്നു കൊണ്ടിരിക്കുന്നു. ഇത്തരം മെയിലുകൾ ഇന്റർനെറ്റ് വഞ്ചനയ്ക്ക് ഉദാഹരണങ്ങളാണ്.

സമാനമായ കുറ്റകൃത്യങ്ങൾ ഓൺലൈൻ ലേലത്തിലും സംഭവിക്കുന്നു. ചിലപ്പോൾ ഇന്റർനെറ്റ് ലേലസൈറ്റുകളിൽ വിൽപ്പനയ്ക്കുള്ള ഉൽപ്പന്നങ്ങൾ അവതരിപ്പിക്കാറുണ്ട്. അവർ സാധനം തരുന്നതിന് മുമ്പ് തന്നെ പണം വാങ്ങുകയും സാധനം തരാതിരിക്കുകയും ചെയ്യുന്നു.

(iv) സ്വകാര്യത ലംഘനം (Violation of privacy): വ്യക്തമായ കാരണം കൂടാതെ മറ്റൊരാളുടെ വ്യക്തിജീവിതത്തിൽ കടന്നു കൂടുകയോ നുഴഞ്ഞു കയറുകയോ ചെയ്യുന്നതാണ് സ്വകാര്യത ലംഘനം. സ്വകാര്യത നഷ്ടപ്പെട്ട ആളിനോ സ്ഥാപനത്തിനോ നിയമനടപടി എടുക്കുന്നതിന് അവകാശമുണ്ട്. വ്യക്തിഗതവിവരങ്ങൾ, ഫോട്ടോ, തൊഴിലിട ദൃശ്യങ്ങൾ എന്നിവ പ്രചരിപ്പിക്കുന്നത് ഇതിൽ ഉൾപ്പെടുന്നു. സ്ത്രീകളുടെ ചിത്രങ്ങൾ എടുക്കുന്നതിന് പൊതു സ്ഥലങ്ങളിൽ ഒളി ക്യാമറ, മൊബൈൽ ക്യാമറ എന്നിവ ഉപയോഗിക്കുന്നതിനെക്കുറിച്ച് നാം കേട്ടിട്ടുണ്ട്. അനുവാദമില്ലാതെ ഒരാളുടെ ഫോട്ടോ എടുക്കുന്നത് സ്വകാര്യതാ ലംഘനമാണ്. അനുവാദമില്ലാതെ മറ്റുള്ളവരുടെ ദൃശ്യങ്ങൾ പൊതു മാധ്യമങ്ങളിൽ പോസ്റ്റ് ചെയ്യുന്നതും അവ പകർത്തുന്നതും ഇ-മെയിലിൽ പ്രചരിപ്പിയ്ക്കുന്നതും സ്വകാര്യതാ ലംഘനമാണ്.

(v) അശ്ലീല വസ്തുക്കളുടെ വ്യാപനം (Dissemination of obscene material): അശ്ലീലത പോലുള്ള കുറ്റകൃത്യങ്ങൾ വ്യാപിക്കുന്നതിന് ഇന്റർനെറ്റ് ഒരു മാധ്യമമായിരിക്കുന്നു. അശ്ലീല വസ്തുക്കൾ പോസ്റ്റ് ചെയ്യുകയോ വിതരണം ചെയ്യുകയോ ചെയ്യുന്നത് സൈബർ കുറ്റകൃത്യമാണ്. ഇന്റർനെറ്റിലെ അശ്ലീലത പല രൂപത്തിലുണ്ട്. നിരോധിച്ചിട്ടുള്ള സൈറ്റുകൾ എടുക്കുക, കമ്പ്യൂട്ടർ ഉപയോഗിച്ച് ഇവ നിർമ്മിക്കുക, ഇന്റർനെറ്റിലൂടെ ഇവ ഡൗൺലോഡ് ചെയ്യുക എന്നിവയെല്ലാം ഇതിൽ ഉൾപ്പെടുന്നു. ഇത് പലപ്പോഴും പ്രായപൂർത്തിയാകാത്ത കുട്ടികളെ തെറ്റിലേക്ക് നയിക്കുന്നു.

ഇരകളുടെ അറിവില്ലാതെയാണ് മിക്കവാറും സൈബർ കുറ്റങ്ങളും നടക്കുന്നത്. എന്നാൽ എളുപ്പത്തിൽ പണമുണ്ടാക്കുന്നതിന് പലപ്പോഴും ഇരകൾ അറിഞ്ഞുകൊണ്ടുതന്നെ ഇതിൽ പങ്കെടുക്കുന്നുമുണ്ട്. ഇത്തരത്തിലുള്ള കുറ്റകൃത്യങ്ങൾ തടയാൻ വിവിധ സംവിധാനങ്ങൾ ഉണ്ടെന്ന് നാം അറിഞ്ഞിരിക്കണം. ഇ-മെയിൽ സേവനങ്ങൾ നൽകുന്നവർ 'സ്പാം ഫിൽറ്റർ' എന്ന ഒരു സംവിധാനം, ആവശ്യമില്ലാത്ത മെയിലുകൾ തടയാൻ വേണ്ടി ഉപയോഗിക്കുന്നു. ഒരു പരിധിവരെ തട്ടിപ്പുകൾ തടയാൻ ഓൺലൈൻ സാമ്പത്തിക ഇടപാടുകൾക്ക് പല തരത്തിലുള്ള ആധികാരികതാ സംവിധാനങ്ങളുണ്ട്. എന്നിരുന്നാലും ഇന്റർനെറ്റ് ഉപയോഗിക്കുമ്പോൾ നാം അതീവ ശ്രദ്ധ പാലിക്കേണ്ടതുണ്ട്.

b. സ്വത്തിനെതിരായ സൈബർ കുറ്റകൃത്യങ്ങൾ (Cyber crimes against property)

ക്രെഡിറ്റ് കാർഡ്, ബൗദ്ധിക സ്വത്ത് എന്നിങ്ങനെയുള്ള സ്വത്തുക്കൾക്കെതിരെ നടത്തുന്ന സൈബർ കുറ്റകൃത്യങ്ങളെയാണ് സ്വത്തിനെതിരായ സൈബർ കുറ്റകൃത്യങ്ങൾ എന്ന്

പറയുന്നത്. ഹാക്കിങ്, സോഫ്റ്റ്‌വെയർ മോഷണം, വസ്തുവകകൾ മനപ്പൂർവ്വം നശിപ്പിക്കൽ, സൈബർ ഇടത്തിൽ നുഴഞ്ഞു കയറ്റം, കമ്പ്യൂട്ടറിൽ ശേഖരിച്ച് വച്ചിരിക്കുന്ന വിവരങ്ങൾ അനധികൃതമായി കൈവശപ്പെടുത്തൽ എന്നിവയെല്ലാം തന്നെ സ്വത്തിനെതിരായ സൈബർ കുറ്റകൃത്യങ്ങൾ ആണ്.

സ്വത്തിനെതിരായ സൈബർ കുറ്റകൃത്യങ്ങളിൽ ചിലത് താഴെ കൊടുത്തിരിക്കുന്നു.

(i) ക്രെഡിറ്റ് കാർഡ് തട്ടിപ്പ് (Credit card fraud): അനുവാദമില്ലാതെ മറ്റൊരാളുടെ ക്രെഡിറ്റ് കാർഡ് വിവരങ്ങളുപയോഗിച്ച് പണം എടുക്കുക, വാങ്ങിക്കുന്ന സാധനങ്ങൾക്ക് പണം നൽകുക എന്നിവയെല്ലാം ക്രെഡിറ്റ് കാർഡ് തട്ടിപ്പാണ്. ഇന്റർനെറ്റ് മോഷ്ടാക്കൾ വലിയ സ്ഥാപനങ്ങളുടെ വെബ്സൈറ്റ് ഹാക്ക് ചെയ്ത് വളരെയധികം ആളുകളുടെ ക്രെഡിറ്റ് കാർഡ് വിവരങ്ങൾ മോഷ്ടിച്ച സംഭവങ്ങളുണ്ട്. മോഷ്ടാക്കൾ ഈ വിവരങ്ങൾ ആർക്കെങ്കിലും പണം നൽകുന്നതിനുപയോഗിക്കുന്നു, ചിലപ്പോൾ മറ്റ് മോഷ്ടാക്കൾക്ക് കുറഞ്ഞ വിലയ്ക്ക് ഇവ വിൽക്കുകയും ചെയ്യുന്നു.

(ii) ബൗദ്ധിക സ്വത്ത് മോഷണം (Intellectual property theft): ബൗദ്ധിക സ്വത്തവകാശ ലംഘനം ഈ ഇനത്തിൽപ്പെട്ടതാണ്. പകർപ്പവകാശം, കുത്തകാവകാശം, വ്യാപാരമുദ്ര എന്നിവയുടെ ലംഘനങ്ങൾ സ്വത്തിനെതിരായ കടന്നു കയറ്റമാണ്. ഈയിടയ്ക്ക് ഒരു ഇന്ത്യൻ ഐ.ടി കമ്പനി, പ്രോഗ്രാം കോഡിലെ തെറ്റുകൾ തിരുത്താനുള്ള ഒരു സോഫ്റ്റ്‌വെയർ നിർമ്മിച്ചു. ഒരു തൊഴിലാളി ഇത് സി.ഡിയിൽ പകർത്തി കമ്പനിയുടെ എതിരാളികൾക്ക് വലിയ വിലയ്ക്ക് വിൽക്കാൻ ശ്രമിച്ചു. ഇത് കമ്പനിക്ക് വലിയ സാമ്പത്തിക നഷ്ടവും സ്വത്ത് നഷ്ടവും ഉണ്ടാക്കി. ഇതും ബൗദ്ധിക സ്വത്ത് മോഷണമായി പരിഗണിക്കപ്പെടുന്നു. സോഫ്റ്റ്‌വെയർ മോഷണവും സൈബർ നിയമപ്രകാരം കുറ്റകരമാണ്.

ഇക്കാലത്ത് ബൗദ്ധിക സ്വത്ത് മോഷണം സർവ്വസാധാരണമാണ്. ഏതു വിഷയത്തിലും നമുക്ക് സ്വതന്ത്രമായി വിവരങ്ങൾ ഇന്റർനെറ്റിൽ നിന്നും എടുക്കാം. മറ്റൊരാളുടെ ഭാഷ, ചിന്തകൾ, ആശയങ്ങൾ ഇവ നമ്മുടെ സ്വന്തം കൃതികളിൽ ഉപയോഗിക്കുന്നതിന് 'പ്ലേജറിസം' (Plagiarism) അഥവാ സാഹിത്യ ചോരണം എന്ന് പറയുന്നു. ഇന്ത്യയിൽ ബൗദ്ധിക സ്വത്തവകാശം സംരക്ഷിക്കുന്നതിന് ശക്തമായ പകർപ്പവകാശ നിയമങ്ങൾ ലഭ്യമാണ്. ഓൺലൈനിൽ ലഭ്യമായിട്ടുള്ള വിവിധ ടൂളുകൾ ഉപയോഗിച്ച് എളുപ്പത്തിൽ പ്ലേജറിസം കണ്ടുപിടിക്കാൻ സാധിക്കും .

(iii) ഇന്റർനെറ്റ് സമയമോഷണം (Internet time theft): ഇന്ന് ഒട്ടു മിക്ക മോഡത്തിനും റൗട്ടറിനും വയർലെസ് ഇന്റർനെറ്റ് സൗകര്യമുണ്ട്. ഇത് വീടുകൾ, സ്കൂളുകൾ, വ്യാപാര സ്ഥാപനങ്ങൾ എന്നിവിടങ്ങളിലൊക്കെ ഇന്റർനെറ്റ് പങ്കു വയ്ക്കൽ സൗകര്യമൊരുക്കുന്നു. ഇവയൊക്കെ ശരിയായ വിധത്തിൽ പാസ്‌വേഡ് ഉപയോഗിച്ച് സുരക്ഷിതമാക്കിയില്ലെങ്കിൽ മറ്റുള്ളവർ നമ്മുടെ ഇന്റർനെറ്റ് ഉപയോഗിക്കും. ഒരാളുടെ ഇന്റർനെറ്റ് സൗകര്യം മറ്റൊ

രാൾ അനധികൃതമായി ഉപയോഗിക്കുന്നതിനെയാണ് ഇന്റർനെറ്റ് സമയ മോഷണം എന്നു പറയുന്നത്. ഇത് യഥാർഥ ഇന്റർനെറ്റ് ഉപയോക്താവിന് പണവും ഇന്റർനെറ്റ് സമയവും നഷ്ടപ്പെടുത്തുന്നു. ഇതിനും പുറമെ മറ്റുള്ളവർ നമ്മുടെ ഇന്റർനെറ്റ് അക്കൗണ്ടുപയോഗിച്ച് കുറ്റകൃത്യങ്ങൾ ചെയ്യുകയും നാം അതിന് ഉത്തരവാദികൾ ആകുകയും ചെയ്യുന്നു.

വിവിധ തരത്തിലുള്ള ആക്രമണങ്ങളായ വൈറസ്, വേംസ്, മാൻ ഇൻ ദി മിഡിൽ അറ്റാക്ക് എന്നിവ പതിനൊന്നാം ക്ളാസ്സിലെ 9-ാം അധ്യായത്തിൽ നാം പഠിച്ചു കഴിഞ്ഞു. ഇതെല്ലാം ഇത്തരത്തിലുള്ള മോഷണത്തിന് ഉദാഹരണങ്ങളാണ്.

c. ഭരണകൂടത്തിനെതിരായ സൈബർ കുറ്റകൃത്യങ്ങൾ (Cyber crimes against government)

ഇ-ഭരണത്തിന്റെ പ്രശസ്തി ഭരണകൂടങ്ങളെ സൈബർ കുറ്റകൃത്യങ്ങൾക്ക് എളുപ്പത്തിൽ വിധേയരാക്കി. ഇങ്ങനെ വിവിധതരത്തിലുള്ള സർക്കാർ കമ്പ്യൂട്ടർ നെറ്റ്‌വർക്കുകളും, വെബ് സൈറ്റുകളും സൈബർ കുറ്റകൃത്യങ്ങളുടെ അപകട ഭീഷണി നേരിടുന്നു. സൈബർ ഭീകരത, വെബ്സൈറ്റ് വികൃതമാക്കൽ, ഇ- ഭരണ വെബ്സൈറ്റുകൾക്കെതിരെയുള്ള ആക്രമണം എന്നിവ ഇത്തരം കുറ്റകൃത്യങ്ങളുടെ വിവിധ ഭാവങ്ങളാണ്.

(i) സൈബർ ഭീകരത (Cyber terrorism)

ന്യൂക്ലിയർ പവർ പ്ലാന്റുകൾ, വ്യോമഗതാഗതനിയന്ത്രണം, ഗ്യാസ് ലൈൻ നിയന്ത്രണം, ടെലികോം തുടങ്ങിയ സൂക്ഷ്മസംവേദനക്ഷമതയുള്ള നെറ്റ്‌വർക്കുകൾ എന്നിവയിലെ സൈബർ ആക്രമണങ്ങളെ ഈ വിഭാഗത്തിൽപെടുത്താം. ലോക വ്യാപകമായി ഭരണകൂടങ്ങൾക്കെതിരായി ഈ വിധത്തിലുള്ള ആക്രമണങ്ങൾ വർധിച്ചു വരുന്നു. രാജ്യത്തിന്റെ സാമ്പത്തിക സാങ്കേതിക അടിത്തറയെ ബാധിക്കുന്ന വിധത്തിൽ ദേശവിരുദ്ധ ശക്തികൾ ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്നതാണ് സൈബർ ഭീകരത.

2010 ൽ ഇറാന്റെ രഹസ്യ ന്യൂക്ലിയർ പദ്ധതിയെ 'ടക്സ്നെറ്റ്' (tuxnet) എന്ന് പേരുള്ള വൈറസ് ഉപയോഗിച്ച് ആക്രമിച്ചു. യുറേനിയം സമ്പുഷ്ടീകരണ പരിപാടിയെ പ്രവർത്തന രഹിതമാക്കാൻ ഉദ്ദേശിച്ചിട്ടുള്ളതായിരുന്നു ഈ വൈറസ്. ഈ വൈറസ് വളരെയധികം ന്യൂക്ലിയർ കൺട്രോളുകളെ ബാധിച്ച് അവയ്ക്ക് തെറ്റായ നിർദേശങ്ങൾ നൽകി. ന്യൂക്ലിയർ പ്രവർത്തനങ്ങൾ തെറ്റായ രീതിയിൽ നടത്തി സജ്ജീകരണങ്ങൾ കേടു വരുത്തി. സൈബർ ഭീകരതക്കെതിരെ ഒരു രാജ്യത്തിന് വലിയ വില നൽകേണ്ടി വരും. അതിനാൽ സെർവറുകൾക്ക് വളരെ ശക്തിയേറിയ സുരക്ഷാ സംവിധാനങ്ങൾ ഒരുക്കേണ്ടത് സർക്കാരിന്റെ ബാധ്യതയാണ്.

(ii) വെബ്സൈറ്റ് വികൃതമാക്കൽ (Website defacement)

ഇത് ഭരണകൂടത്തിനെതിരെയുള്ള സാധാരണ സൈബർ ആക്രമണമാണ്. ഇതിൽ സർക്കാർ വെബ് സൈറ്റുകൾ ഹാക്ക് ചെയ്യുകയും അവയിൽ സർക്കാരിനെതിരെ ദോഷകരമായ അഭിപ്രായങ്ങൾ ഇടുകയും ചെയ്യുന്നു.

(iii) ഇ-ഭരണ വെബ് സൈറ്റുകൾക്കെതിരെയുള്ള ആക്രമണങ്ങൾ (Attacks against e-governance websites)

ഇത്തരം ആക്രമണങ്ങൾ ഒരു പ്രത്യേകതരം ഓൺലൈൻ സർക്കാർ സേവനം ലഭ്യമല്ലാതാക്കുന്നു. പതിനൊന്നാം ക്ലാസ്സിലെ 9-ാം അധ്യായത്തിൽ നാം പഠിച്ച 'ഡിസ്ട്രിബ്യൂട്ടഡ് ഡിനയൽ ഓഫ് സർവീസ്' (DDOS) ഉപയോഗിച്ചാണ് ഇത് ചെയ്യുന്നത്. ഹാക്കർമാർ വെബ്സൈറ്റ് തട്ടിയെടുത്ത് ഉപയോഗിക്കുന്നത് ഇതിന്റെ മറ്റൊരു തലമാണ്. അവർക്ക് ഉള്ളടക്ക നിയന്ത്രണത്തിലൂടെ (Content Management System) വെബ്സൈറ്റ് നിയന്ത്രണം സാധിക്കുകയും വിവരങ്ങൾ നശിപ്പിക്കാൻ സാധിക്കുകയും ചെയ്യുന്നു. ഇത് മൂലം സർക്കാരിന് വലിയ നഷ്ടം സംഭവിക്കുന്നു.

സൈബർ ഇടത്തിന്റെ അജ്ഞാത സ്വഭാവം കാരണം ആധുനിക ലോകത്ത് സൈബർ ഭീകരത വർധിച്ചിട്ടുണ്ട്. വളരെയധികം ആളുകളെ ബാധിക്കുന്ന വിവിധ ലക്ഷ്യ സ്ഥാനങ്ങളുടെ ലഭ്യതയും, ദൂരെ നിന്ന് ആക്രമണങ്ങൾ നടത്താമെന്നുള്ളതും സൈബർ ഇടത്തിലൂടെയുള്ള ഭീകര പ്രവർത്തനങ്ങൾ വർധിപ്പിക്കുന്നു.

11.5.5 സൈബർ സദാചാരം (Cyber ethics)

ആഗോളതലത്തിൽ സൈബർ കുറ്റകൃത്യങ്ങൾ വർധിച്ചു വരുന്നതായി കണക്കുകൾ സൂചിപ്പിക്കുന്നു. ഇന്റർനെറ്റ് കുറ്റകൃത്യങ്ങൾ പലതരമാണ്. അവ വിവിധ രീതികളിൽ ചെയ്യുന്നു. ആദ്യഘട്ടം നമ്മുടെ കുടുംബത്തെയും വ്യാപാരത്തെയും ഇത്തരം കുറ്റകൃത്യങ്ങളിൽ നിന്ന് സംരക്ഷിക്കുക എന്നതാണ്. അതുപോലെ തന്നെ സൈബർ ഇടത്തിലെ നമ്മുടെ പ്രവർത്തനങ്ങൾ മറ്റുള്ളവർക്ക് ദോഷം വരുത്തില്ല എന്ന് ഉറപ്പു വരുത്തേണ്ടതുണ്ട്. നമ്മുടെ പ്രവർത്തനങ്ങൾ മറ്റു പലരും നിരീക്ഷിക്കുന്നുണ്ടെന്ന് നാം ഓർക്കണം.

ഇന്റർനെറ്റ് ഉപയോഗിക്കുന്നവർക്കുള്ള സൈബർ സദാചാര നിർദേശങ്ങൾ താഴെ കൊടുക്കുന്നു.

- നിങ്ങളുടെ കമ്പ്യൂട്ടറിൽ ആന്റിവൈറസുകൾ, ഫയർവാൾ, സ്പാം ബ്ലോക്കിങ് സോഫ്റ്റ് വെയർ എന്നിവ ഉപയോഗിക്കുക.
- ഓൺലൈൻ പണമിടപാടുകൾ നടത്തുമ്പോൾ സുരക്ഷാ വെബ്സൈറ്റുകൾ ഉണ്ടെന്ന് ഉറപ്പു വരുത്തുക
- അജ്ഞാതർ അയയ്ക്കുന്ന ഇ-മെയിലുകളോട് പ്രതികരിക്കാതിരിക്കുക
- അസാധാരണവും സങ്കീർണവുമായ പാസ്‌വേർഡുകൾ ഉപയോഗിക്കുക. നിശ്ചിത ഇടവേളകളിൽ പാസ്‌വേർഡുകൾ മാറ്റുക. പാസ്‌വേഡിൽ കുറഞ്ഞത് 8 അക്ഷരങ്ങൾ ഉണ്ടായിരിക്കണം. അക്ഷരങ്ങളും, സംഖ്യകളും, ചിഹ്നങ്ങളും ഉപയോഗിക്കണം.
- ഒരു കരാറിന്റെയോ സന്ദേശത്തിന്റെയോ നേരെയുള്ള ചെക്കുബോക്സുകളോ OK ബട്ടണുകളോ അവ വായിച്ചു തീരുന്നതിന് മുമ്പ് ക്ലിക്ക് ചെയ്യരുത്.
- ആധികാരികതയില്ലാത്ത സോഫ്റ്റ്‌വെയർ ഉപയോഗിക്കരുത്.
- നിങ്ങളുടെ തിരിച്ചറിയൽ രേഖ മറച്ചുവച്ച് മറ്റുള്ളവരെ കബളിപ്പിക്കരുത്.

- സാമൂഹിക മാധ്യമങ്ങളിലോ ഇ-മെയിലിലോ മോശമായതോ പരുക്കനായതോ ആയ ഭാഷ ഉപയോഗിക്കരുത്.
- നിങ്ങളുടെ സ്വന്തം കമ്പ്യൂട്ടറുകൾ അല്ലാത്തവ ഉപയോഗിക്കുമ്പോൾ അക്കൗണ്ടിൽ പ്രവേശിക്കുന്നതിന് മുമ്പ് (login), 'Remember me' എന്ന ചെക്ക് മാർക്ക് നീക്കം ചെയ്യുക.

നമുക്ക് ചെയ്യാം

- വിവിധ തരം സൈബർ കുറ്റകൃത്യങ്ങളെപ്പറ്റി ബോധവൽക്കരണ പരിപാടികൾ സംഘടിപ്പിക്കുക.
- സൈബർ ഇടത്തിൽ എങ്ങനെ നമ്മെ സംരക്ഷിക്കാം എന്നതിനെക്കുറിച്ച് പട്ടിക തയാറാക്കുക.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. സൈബർ ഇടം എന്നാലെന്ത്?
2. കമ്പ്യൂട്ടർ, മൊബൈൽ ഫോൺ, ഇന്റർനെറ്റ് എന്നിവ ഉപയോഗിച്ചുള്ള കുറ്റകൃത്യങ്ങളെ ________________ എന്ന് വിളിക്കുന്നു.
3. വ്യക്തികൾക്കെതിരെയുള്ള സൈബർ കുറ്റകൃത്യം എന്നാൽ എന്താണ്?
4. സൈബർ ഭീകരത എന്നത് ________________ ന് എതിരായ ഒരു സൈബർ കുറ്റകൃത്യമാണ്.
5. ഒരാളുടെ യൂസർ നെയിം, പാസ്‌വേർഡ് തുടങ്ങിയവ മോഷ്ടിക്കുന്നത് ________________ ആണ്.

11.5.6 സൈബർ നിയമങ്ങൾ (Cyber laws)

ഇന്റർനെറ്റിന്റെ നിയമപരവും നിയന്ത്രണപരവുമായ കാര്യങ്ങളെ സൂചിപ്പിക്കുന്നതിനാണ് സൈബർ നിയമങ്ങൾ എന്ന പേര് ഉപയോഗിക്കുന്നത്. കമ്പ്യൂട്ടറുകളുടെയും ഇന്റർനെറ്റിന്റെയും നിയന്ത്രണത്തെ സംബന്ധിക്കുന്ന നിയമങ്ങൾ എന്ന് സൈബർ നിയമത്തെ നിർവചിക്കാം.

വിവിധ തരം സൈബർ കുറ്റകൃത്യങ്ങൾ ഉണ്ടെന്ന് നാം കണ്ടു. മോഷണം, വഞ്ചന, കൃത്രിമ രേഖയുണ്ടാക്കൽ, അപകീർത്തിപ്പെടുത്തൽ എന്നിവയെല്ലാം പരമ്പരാഗത സ്വഭാവമുള്ള കുറ്റകൃത്യങ്ങളും ഇന്ത്യൻ ശിക്ഷാ നിയമങ്ങൾക്ക് (Indian Penal Code) വിധേയവുമാണ്. വിവര സാങ്കേതികവിദ്യ നിയമം 2000 വും ഐ.ടി. ആക്ട് ഭേദഗതി ബിൽ 2008 ഉം ഇന്ത്യയിൽ വ്യാപകമായി വരുന്ന സൈബർ കുറ്റങ്ങളെ നേരിടാനുള്ളതാണ്. ഇന്റർനെറ്റ് ഉപയോഗിച്ചുള്ള എല്ലാ പ്രവർത്തനങ്ങളെയും ഇടപാടുകളെയും സ്പർശിക്കുന്നതാകയാൽ സൈബർ നിയമം പ്രാധാന്യമുള്ളതാണ്. നാം മനസ്സിലാക്കിയാലും ഇല്ലെങ്കിലും സൈബർ ഇടത്തിലുള്ള എല്ലാ പ്രവർത്തനത്തിനും പ്രതിപ്രവർത്തനത്തിനും നിയമപരമായ വശങ്ങൾ ഉണ്ട്.

11.5.7 വിവര സാങ്കേതികവിദ്യ നിയമം 2000 (2008 ൽ ഭേദഗതി വരുത്തിയത്) (Information Technology Act 2000 [Amended in 2008])

കമ്പ്യൂട്ടറുകൾ, സെർവറുകൾ, കമ്പ്യൂട്ടർ നെറ്റ്‌വർക്കുകൾ, ഇലക്ട്രോണിക് രൂപത്തിലുള്ള ഡാറ്റയും വിവരങ്ങളും എന്നിവയുടെ ഉപയോഗത്തെ നിയന്ത്രിക്കുന്ന ഭാരത സർക്കാരിന്റെ നിയമമാണ് വിവരസാങ്കേതികവിദ്യ നിയമം 2000. നെറ്റ്‌വർക്ക് സേവനദാതാവിന്റെ ബാധ്യതകൾ, ആധികാരികത വരുത്തൽ, ഡിജിറ്റൽ ഒപ്പ്, സൈബർ കുറ്റകൃത്യങ്ങൾ എന്നിവയുമായി ബന്ധപ്പെട്ട വിവിധ വശങ്ങളെ ഈ നിയമം സ്പർശിക്കുന്നു. ഇലക്ട്രോണിക് ഡാറ്റ കൈമാറ്റത്തിലൂടെയുള്ള ഇടപാടുകൾക്കും മറ്റ് ഇലക്ട്രോണിക് ആശയ വിനിമയ രീതികൾക്കും ഈ നിയമം അംഗീകാരം നൽകുന്നു.

പേപ്പർ ഉപയോഗിച്ചുള്ള ആശയ വിനിമയത്തിന് പകരമുള്ള മാർഗങ്ങൾ നൽകുകയും സർക്കാർ ഏജൻസികളുടെ രേഖകൾ ഇലക്ട്രോണിക് ആയി ഫയൽ ചെയ്യാനും ഐ.ടി നിയമം സഹായിക്കുന്നു. കൂടാതെ ഇലക്ട്രോണിക് ആശയ വിനിമയത്തിന് നിയമ പ്രാബല്യം നൽകുന്നു. ഇത് സൈബർ ഇടത്തിലുള്ള കുറ്റകൃത്യങ്ങൾ, തർക്കങ്ങൾ എന്നിവ നേരിടുന്നതിനും സൈബർ കുറ്റകൃത്യങ്ങളുടെ ഇരകൾക്ക് നീതി ലഭ്യമാക്കുവാൻ സഹായകമാവുകയും ചെയ്യുന്നു. ഇന്ത്യയിലെ ഇ-വാണിജ്യത്തിനുള്ള നിയമപരമായ അടിത്തറ ഒരുക്കുന്നതിന് ഐ.ടി. നിയമം ലക്ഷ്യമിടുന്നു. ഐ.ടി. വ്യവസായത്തെ പ്രോത്സാഹിപ്പിക്കുന്നതിനും ഇ-വാണിജ്യം നിയന്ത്രിക്കുന്നതിനും ഇ-ഭരണത്തിന് സൗകര്യമൊരുക്കുന്നതിനും സൈബർ കുറ്റങ്ങൾ തടയുന്നതിനുമാണ് ഈ നിയമം ഉണ്ടാക്കിയിട്ടുള്ളത്. ആഗോള തലത്തിൽ രാജ്യത്തെ സേവിക്കുന്നതിന് ഇന്ത്യക്കകത്തു തന്നെയുള്ള സുരക്ഷാ ക്രമീകരണ രീതികളെ ഇത് പ്രോത്സാഹിപ്പിക്കുന്നു. ഈ നിയമമനുസരിച്ച് ചില നിയമ ലംഘനങ്ങൾ ഗൗരവമുള്ള കുറ്റങ്ങളായി കണക്കാക്കി കുറ്റവാളികളെ നിയമ വിചാരണയ്ക്ക് വിധേയമാക്കുകയും ചെയ്യുന്നു. അതിനാൽ ഐ.ടി. നിയമം 2000 ന്റെ പ്രധാനപ്പെട്ട ആശയങ്ങൾ മനസ്സിലാക്കേണ്ടതാണ്.

2000 മെയ് മാസത്തിലാണ് ഇന്ത്യൻ പാർലമെന്റിൽ വിവര സാങ്കേതികവിദ്യ ബിൽ പാസ്സാക്കിയത്. ഇത് വിവര സാങ്കേതികവിദ്യ നിയമം 2000 എന്നറിയപ്പെടുന്നു. പിന്നീട് ഈ നിയമം 'ഐ.ടി. ഭേദഗതി ബിൽ 2008' എന്ന പേരിൽ ഭേദഗതി ചെയ്ത് 2008 ഡിസംബറിൽ പാസ്സാക്കി. വിവര സാങ്കേതികവിദ്യ നിയമം 2000 നു ശേഷമുള്ള ഐ.ടി പുരോഗതിയും സുരക്ഷാ കാര്യങ്ങളും ഇതിൽ ഉൾപ്പെടുത്തിയിട്ടുണ്ട്. വൈറസ് കൊണ്ടോ, സേവനം നിഷേധിക്കുന്നത് കൊണ്ടോ, കമ്പ്യൂട്ടറുകൾക്കോ, കമ്പ്യൂട്ടർ നെറ്റ്‌വർക്കുകൾക്കോ ഉണ്ടാകുന്ന കേടുപാടുകൾക്ക് നഷ്ടപരിഹാരം നൽകേണ്ടതാണ്. ഈ നിയമത്തിലെ 65-74 ഭാഗങ്ങൾ സൈബർ കുറ്റങ്ങളെപ്പറ്റി പ്രത്യേകം പ്രതിപാദിക്കുന്നു.

ഐ.ടി നിയമം 2000 ലും ഐ.ടി ഭേദഗതി 2008 ലും സൂചിപ്പിച്ചിട്ടുള്ള ചില സാധാരണ കുറ്റങ്ങളും അവയുടെ ശിക്ഷകളും

ഭാഗം	കുറ്റം	ശിക്ഷ
65	കമ്പ്യൂട്ടറിന്റെ സോഴ്സ് കോഡിൽ നാശം വരുത്തുക	2 ലക്ഷം രൂപ വരെ പിഴയോ, 3 വർഷം തടവോ രണ്ടും കൂടിയോ
66	കമ്പ്യൂട്ടറുമായി ബന്ധപ്പെട്ട കുറ്റങ്ങൾ	5 ലക്ഷം രൂപ പിഴയോ, 3 വർഷം വരെ തടവോ രണ്ടും കൂടിയോ
66B	മോഷ്ടിച്ച കമ്പ്യൂട്ടർ, ആശയവിനിമയ സംവിധാനം ഇവ വാങ്ങുക	1 ലക്ഷം രൂപ പിഴയോ, 3 വർഷം വരെ തടവോ രണ്ടും കൂടിയോ
66C	ഐഡന്റിറ്റി മോഷണം നടത്തുക	1 ലക്ഷം രൂപ പിഴയോ, 3 വർഷം വരെ തടവോ
66D	കമ്പ്യൂട്ടർ വിഭവങ്ങൾ ഉപയോഗിച്ച് ആൾ മാറാട്ടം നടത്തി വഞ്ചിക്കുക	1 ലക്ഷം രൂപ പിഴയോ, 3 വർഷം വരെ തടവോ രണ്ടും കൂടിയോ
66E	സ്വകാര്യത ലംഘനം	2 ലക്ഷം രൂപ പിഴയോ, 3 വർഷം വരെ തടവോ രണ്ടും കൂടിയോ
66F	സൈബർ ഭീകരത	ജീവിതാവസാനം വരെ തടവ്
67	ഇലക്ട്രോണിക് രീതിയിൽ അശ്ലീലം പ്രചരിപ്പിക്കൽ	2 മുതൽ 3 വർഷം വരെ തടവും 5 ലക്ഷം രൂപ വരെ പിഴയും. കുറ്റം ആവർത്തിച്ചാൽ 5 വർഷം വരെ തടവും 10 ലക്ഷം രൂപ വരെ പിഴയും
67A	ഇലക്ട്രോണിക് രീതിയിൽ ലൈംഗിക അതിക്രമം പ്രചരിപ്പിക്കൽ	10 ലക്ഷം രൂപ വരെ പിഴയും, 5 വർഷം വരെ തടവും
67B	ഇലക്ട്രോണിക് രീതിയിൽ കുട്ടികൾക്കെതിരെയുള്ള ലൈംഗിക അതിക്രമം പ്രചരിപ്പിക്കൽ	5 വർഷം വരെ തടവും 10 ലക്ഷം രൂപ വരെ പിഴയും. കുറ്റം ആവർത്തിച്ചാൽ 7 വർഷം വരെ തടവും 10 ലക്ഷം രൂപ വരെ പിഴയും

ഓൺലൈൻ ആശയവിനിമയ സംവിധാനങ്ങളായ ഇ-മെയിൽ, സാമൂഹിക മാധ്യമം, എസ്.എം.എസ്. മുതലായ വയിലൂടെ അശ്ലീല സന്ദേശങ്ങൾ അയച്ചാലുള്ള ശിക്ഷാനടപടികൾ ഇന്ത്യൻ പീനൽ കോഡിലെ (IPC) ബന്ധപ്പെട്ട വിഭാഗങ്ങൾക്കനുസരിച്ചായിരിക്കും.

11.5.8 സൈബർ കുറ്റാന്വേഷണം (Cyber forensics)

സൈബർ കുറ്റങ്ങളുടെ വർധന നിത്യജീവിതത്തെയും രാജ്യ സുരക്ഷയെയും ബാധിക്കുന്നു. വളരെയധികം സൗകര്യങ്ങൾ ഇന്റർനെറ്റ് വഴി ലഭ്യമാണെങ്കിലും അത് കുറ്റവാളികൾക്ക് കൂടുതൽ കുറ്റങ്ങൾ ചെയ്യുന്നതിന് സഹായകമാകുന്നു. സൈബർ കുറ്റങ്ങൾ കണ്ടുപിടിച്ച് അന്വേഷണം നടത്തി ശിക്ഷിക്കുവാൻ പരമ്പരാഗതമായി ഉണ്ടായിരുന്ന നിയമങ്ങൾ ഫലപ്രദമായിരുന്നില്ല.

ശാസ്ത്രീയമായ അറിവുപയോഗിച്ച് കുറ്റകൃത്യങ്ങൾ കണ്ടുപിടിക്കുകയും, അവ തെളിവുകൾ സഹിതം ശേഖരിക്കുകയും, വിശകലനം ചെയ്ത് കോടതിക്ക് സമർപ്പിക്കുകയും

ചെയ്യുന്ന പ്രക്രിയയാണ് ഫോറെൻസിക്സ്. നിയമവും കമ്പ്യൂട്ടർ സയൻസും യോജിപ്പിച്ച് കമ്പ്യൂട്ടർ സിസ്റ്റങ്ങൾ, നെറ്റ്‌വർക്കുകൾ, ആശയവിനിമയ സംവിധാനങ്ങൾ ഇവയിൽ നിന്നുള്ള വിവരങ്ങൾ വിശകലനം നടത്തി കോടതിക്ക് തെളിവ് കൊടുക്കുന്ന ശാസ്ത്ര ശാഖയാണ് സൈബർ ഫോറൻസിക്സ്. ശേഖരിച്ച തെളിവുകളുടെ സത്യസന്ധതയെ സംരക്ഷിക്കുന്ന വിധത്തിൽ ഡാറ്റ വിശകലനം ചെയ്ത് നിയമാനുസരണം ഫലപ്രദമായി ഉപയോഗിക്കലാണ് കമ്പ്യൂട്ടർ ഫോറെൻസിക്സിന്റെ ലക്ഷ്യം.

11.5.9 ഇൻഫോമാനിയ (Infomania)

വിവരങ്ങൾ വിജയത്തിന്റെ താക്കോലാണ്. അവ ശേഖരിച്ച് പ്രൊസസ്സ് ചെയ്തു വേണം ഉപയോഗിക്കേണ്ടത്. എന്നാൽ ഒരാൾക്ക് താങ്ങാവുന്നതിലധികം വിവരങ്ങൾ ലഭിക്കുകയാണെങ്കിൽ അയാൾക്ക് എന്ത് ചെയ്യാനാകും? ഇങ്ങനെ അധികവിവരങ്ങൾ ഉപയോഗിച്ച് പരിക്ഷീണിതനാകുന്ന അവസ്ഥയാണ് ഇൻഫോമാനിയ.

ഇന്റർനെറ്റ്, ഇമെയിൽ, സെൽഫോൺ തുടങ്ങി അനേകം സ്രോതസ്സുകളിൽ കുമിഞ്ഞു കൂടുന്ന വിവരങ്ങൾ പ്രോസസ്സ് ചെയ്യാൻ കഴിയാത്തതു മൂലമാണ് ഇൻഫോമാനിയ ഉണ്ടാകുന്നത്. വിവരങ്ങൾ ശേഖരിക്കുമ്പോൾ അവയുടെ ഗുണമേന്മയും പ്രസക്തിയും കണക്കിലെടുക്കണം. വിവരങ്ങൾ ശേഖരിക്കാനുള്ള അമിതാവേശമാണ് ഇൻഫോമാനിയ. ഇത് മൂലം കുടുംബം, കർത്തവ്യങ്ങൾ തുടങ്ങിയ കൂടുതൽ പ്രധാനപ്പെട്ട കാര്യങ്ങൾ അവഗണിക്കപ്പെടുന്നു. ഭക്ഷണ സമയത്തു പോലും വിവരങ്ങൾ ബ്രൗസ് ചെയ്ത് കൊണ്ടിരിക്കുന്നവരെ നാം കണ്ടിട്ടുണ്ട്. ഇത് ഒരു മാനസിക പ്രശ്നമാണ്. സറിരമായി ഇ-മെയിൽ, സാമൂഹിക നെറ്റ്‌വർക്കുകൾ, ഓൺലൈൻ വാർത്തകൾ എന്നിവ പരതികൊണ്ടിരിക്കുന്നത് ഇൻഫോമാനിയയുടെ ലക്ഷണങ്ങളാണ്. തങ്ങളെ തന്നെ മെച്ചപ്പെടുത്തി സ്വന്തം ഗ്രൂപ്പിൽ നിന്നും പുറത്താകാതിരിക്കുന്നതിനാണ് മിക്കവാറും ഇങ്ങനെ ചെയ്യുന്നത്.

ഇൻഫോമാനിയയ്ക്ക് അടിമപ്പെട്ടിട്ടുള്ളവർക്ക് ശ്രദ്ധയും ഉറക്കവും കുറയുന്നു. സാങ്കേതിക വിദ്യയുടെ അമിതോപയോഗം ബുദ്ധി കുറയ്ക്കുന്നു. ചിലർ ഇ-മെയിലിലും സാമൂഹിക മാധ്യമങ്ങളിലും വരുന്ന സന്ദേശങ്ങൾക്ക് മറുപടി അയയ്ക്കുന്നതിന് മുൻഗണന നൽകുന്നു. ഇതിനാൽ അവർ കുടുംബത്തിലെയും തൊഴിലിലെയും പ്രധാന കാര്യങ്ങൾ മറന്നു പോകുന്നു. ഇതിൽ നിന്ന് രക്ഷ നേടാൻ കുടുംബ കാര്യങ്ങളും തൊഴിലും ചെയ്യുമ്പോൾ സാമൂഹിക മാധ്യമങ്ങൾ ഉപയോഗിക്കുന്നതിൽ നിന്നും, ഓൺലൈൻ വാർത്തകൾ കാണുന്നതിൽ നിന്നും തൽക്കാലത്തേക്കെങ്കിലും മാറി നിൽക്കാൻ അഥവാ ലോഗ് ഓഫ് ചെയ്യാൻ പഠിക്കണം.

നിങ്ങളുടെ പുരോഗതി അറിയുക

1. ഇന്ത്യയിൽ ഏതു വർഷമാണ് ഐ.ടി നിയമം നിലവിൽ വന്നത്?
2. ഐ.ടി നിയമത്തിന്റെ ഉദ്ദേശ്യമെന്താണ്?
3. സൈബർ ഫോറൻസിക്സ് എന്നതുകൊണ്ട് ഉദ്ദേശിക്കുന്നതെന്താണ്?
4. അറിവ് സമ്പാദിക്കാനുള്ള അമിതാവേശമാണ് ________________

നമുക്ക് സംഗ്രഹിക്കാം

മൊബൈൽ ഫോണിന്റെയും കമ്പ്യൂട്ടറിന്റെയും ഏകീകരണം നമ്മുടെ ശ്രദ്ധയെ ഡെസ്ക് ടോപ്പുകളിൽ നിന്നും മൊബൈൽ കമ്പ്യൂട്ടിങ് ഉപകരണങ്ങളിലേക്ക് വഴി തിരിച്ചു വിട്ടിരിക്കുന്നു. ഈ ഉപകരണങ്ങൾ മിക്കവാറും എല്ലായ്പ്പോഴും ഇന്റർനെറ്റുമായി ബന്ധിപ്പിച്ചിരിക്കുന്നതുകൊണ്ടുതന്നെ മൊബൈൽ വിനിമയ സാങ്കേതികവിദ്യ വളരെ പ്രാധാന്യം കൈവരിക്കുകയും ചെയ്തിരിക്കുന്നു. വേഗതയ്ക്ക് പ്രഥമ പരിഗണന നൽകിക്കൊണ്ട് മൊബൈൽ വിനിമയ സാങ്കേതികവിദ്യ 1G മുതൽ 5G വരെ വിവിധ തലമുറകളായി വികാസം പ്രാപിച്ചിരിക്കുന്നു. SMS, MMS, GPS തുടങ്ങിയ സവിശേഷതകളുടെ ആവിർഭാവത്തിലേക്കും ഇത് വഴിതെളിച്ചു. ലിനക്സ് കെർണലിനെ അടിസ്ഥാനമാക്കി ഗൂഗിൾ വികസിപ്പിച്ചെടുത്ത ആൻഡ്രോയിഡ് ഇന്ന് പ്രചാരത്തിലുള്ള മൊബൈൽ ഓപ്പറേറ്റിങ് സിസ്റ്റങ്ങളിൽ ഒന്നാണ്. ഒരു വ്യക്തിക്ക് സൈബർ ഇടങ്ങളിലൂടെ വൈവിധ്യമാർന്ന പ്രവർത്തനങ്ങൾ ചെയ്യുവാൻ സാധിക്കും. സൈബർ ഇടങ്ങളുടെ അജ്ഞാത സ്വഭാവം അതിനെ പലപ്പോഴും കുറ്റവാളികളുടെ താവളങ്ങളാക്കി മാറ്റുന്നു. കുറ്റകൃത്യങ്ങൾ വ്യക്തികൾക്കും വസ്തുവകകൾക്കും സർക്കാരിനും എതിരെയാകാറുണ്ട്. ഇത്തരം അപകടങ്ങളിൽ നിന്ന് സുരക്ഷിതമായിരിക്കാനുള്ള ഏറ്റവും നല്ല മാർഗം ഇൻറർനെറ്റിൽ പതിയിരിക്കുന്ന കെണികളെക്കുറിച്ച് അവബോധമുണ്ടായിരിക്കുക എന്നുള്ളതാണ്. സംഗീതം, സോഫ്റ്റ്‌വെയർ, കലാപരമായ പ്രവർത്തനങ്ങൾ, കണ്ടുപിടുത്തങ്ങൾ തുടങ്ങിയവയുടെമേൽ സൃഷ്ടികർത്താവിന് അല്ലെങ്കിൽ ഉടമസ്ഥന് ഉള്ള അവകാശം ബൗദ്ധിക സ്വത്തവകാശത്തിലൂടെ സംരക്ഷിക്കപ്പെട്ടിരിക്കുന്നു. സൈബർ ഇടങ്ങളിൽ സുരക്ഷിതമായ ഒരു അന്തരീക്ഷം സജ്ജമാക്കുന്നതിന് സൈബർ നിയമങ്ങൾ വ്യക്തമായ ഒരു ചട്ടക്കൂട് പ്രദാനം ചെയ്യുന്നു. കരുത്തുള്ള ഒരു ഐടി ആക്ട് ഇന്ത്യക്ക് ഉണ്ട്. കമ്പ്യൂട്ടറുകൾ, ഇൻറർനെറ്റ്, മൊബൈൽ ഫോണുകൾ എന്നിവ ഉപയോഗിച്ച് നടത്തുന്ന പലതരത്തിലുള്ള കുറ്റകൃത്യങ്ങളെ ഇത് കൈകാര്യം ചെയ്യുന്നു. മിക്കവാറും എല്ലാ സിവിൽ, ക്രിമിനൽ കേസുകളുടെയും നിയമപരമായ പ്രവർത്തനങ്ങളുടെ ഭാഗമായി സൈബർ ഫോറൻസിക് വികസിച്ചു കഴിഞ്ഞു. ഇന്റർനെറ്റ് ഉപയോഗിക്കുമ്പോൾ നമ്മുടെ പ്രവർത്തനങ്ങൾ മറ്റുള്ളവർക്ക് ഹാനികരമാകുന്നില്ല എന്ന് നാം ഉറപ്പാക്കേണ്ടതുണ്ട്. വിവരാധിക്യം മൂലം സംജാതമാകുന്ന പ്രശ്നങ്ങളെ അഭിമുഖീകരിക്കുന്നതിന് ഇൻഫോമാനിയ എന്ന പദം ഉപയോഗിക്കുന്നു.

നമുക്ക് വിലയിരുത്താം

1. മൊബൈൽ കമ്പ്യൂട്ടിങ് എന്നാൽ എന്ത്?
2. മൊബൈൽ ആശയവിനിമയത്തിലെ തലമുറകൾ വിശദീകരിക്കുക?
3. GSM, CDMA എന്നിവ താരതമ്യം ചെയ്യുക.
4. SMS നെ കുറിച്ച് ലഘു കുറിപ്പ് തയാറാക്കുക.
5. GPRS ഉം EDGE ഉം തമ്മിലുള്ള വ്യത്യാസമെന്ത്?
6. മൊബൈൽ ഫോൺ ഉപയോഗിച്ച് മൾട്ടിമീഡിയ സന്ദേശങ്ങൾ അയയ്ക്കുവാനും സ്വീകരിക്കുവാനും ഉള്ള അംഗീകൃത മാർഗ്ഗമാണ്______________
7. GPS ന്റെ പൂർണരൂപം?
8. സ്മാർട്ട് കാർഡ് എന്നാൽ എന്ത്? അത് എങ്ങനെ ഉപകാരപ്രദമാകുന്നു?
9. ആൻഡ്രോയ്ഡ് OS ന്റെ പ്രത്യേകതകൾ വിവരിക്കുക?
10. സൈബർ ഇടം എന്നാലെന്ത്?
11. സൈബർ ഇടത്തെ എന്തുകൊണ്ടാണ് അയഥാർഥ ലോകം എന്നു പറയുന്നത്?
12. ഫിഷിങ് ______________ നു ഉദാഹരണമാണ്.
13. വിവിധ വിഭാഗങ്ങളിലുള്ള സൈബർ കുറ്റകൃത്യങ്ങൾ വിശദമാക്കുക.
14. 'സൈബർ ഇടത്തിൽ നമ്മെ സുരക്ഷിതരാക്കാനുള്ള ഏറ്റവും നല്ല മാർഗം ബോധവൽക്കരണമാണ്'. ആശയം വിശദമാക്കുക.
15. വ്യാപാരമുദ്രയും വ്യാവസായിക രൂപരേഖയും തമ്മിലുള്ള വ്യത്യാസമെന്ത്?
16. പകർപ്പവകാശം എന്നാലെന്ത്? കുത്തകാവകാശത്തിൽ നിന്ന് ഇത് എങ്ങനെ വ്യത്യസ്തമാകുന്നു?
17. ഉടമയ്ക്ക് സ്വത്തിന്മേലുള്ള പ്രത്യേക അവകാശം IPR നൽകുന്നത് എങ്ങനെയാണ്?
18. അവകാശ മോഷണം എന്തെന്ന് വിശദമാക്കുക?
19. അവകാശക്കയ്യേറ്റം എന്നതു കൊണ്ട് ഉദ്ദേശിക്കുന്നതെന്ത്?
20. സൈബർ നിയമങ്ങളുടെ പ്രാധാന്യം എന്ത്?
21. സൈബർ കുറ്റകൃത്യങ്ങളുടെ പ്രാദേശികമല്ലാത്ത സ്വഭാവം കുറ്റാന്വേഷകർക്ക് പ്രശ്നങ്ങൾ സൃഷ്ടിക്കുന്നു. എങ്ങനെയെന്ന് വിശദമാക്കുക.
22. ഇൻഫോമാനിയ ഒരു മാനസിക പ്രശ്നമായി മാറിയിരിക്കുന്നു. നിങ്ങളുടെ അഭിപ്രായമെന്ത്?