ગુજરાત રાજ્યના શિક્ષણવિભાગના પત્ર-ક્રમાંક
મશબ/1214/15/છ, તા. 29-1-2014થી–મંજૂર

કમ્પ્યૂટર-અધ્યયન

ધોરણ 12

પ્રતિજ્ઞાપત્ર

ભારત મારો દેશ છે.
બધાં ભારતીયો મારાં ભાઈબહેન છે.
હું મારા દેશને ચાહું છું અને તેના સમૃદ્ધ અને
વૈવિધ્યપૂર્ણ વારસાનો મને ગર્વ છે.
હું સદાય તેને લાયક બનવા પ્રયત્ન કરીશ.
હું મારાં માતાપિતા, શિક્ષકો અને વડીલો પ્રત્યે આદર રાખીશ
અને દરેક જણ સાથે સભ્યતાથી વર્તીશ.
હું મારા દેશ અને દેશબાંધવોને મારી નિષ્ઠા અર્પું છું.
તેમનાં કલ્યાણ અને સમૃદ્ધિમાં જ મારું સુખ રહ્યું છે.

કિંમત : ₹ 105.00

ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળ
'વિદ્યાયન', સેક્ટર 10-એ, ગાંધીનગર-382010

© ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળ, ગાંધીનગર
આ પાઠ્યપુસ્તકના સર્વ હક ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળને હસ્તક છે.
આ પાઠ્યપુસ્તકનો કોઈ પણ ભાગ કોઈ પણ રૂપમાં ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળના નિયામકની લેખિત પરવાનગી વગર પ્રકાશિત કરી શકાશે નહિ.

વિષય-સલાહકાર
પ્રો. આર. પી. સોની

લેખન-સંપાદન
ડૉ. હર્ષલ આરોલકર (કન્વીનર)
શ્રી જીજ્ઞેશભાઈ સ્માર્ત
શ્રી જ્યોતિકાબહેન દોશી
શ્રી તૃપ્તિબહેન ડોડિયા
શ્રી હાર્દિકભાઈ જોશી

અનુવાદ
શ્રી રોહિતભાઈ દોશી
શ્રી ગિરીશ બ્રહ્મભટ્ટ
શ્રી સાકેત દવે
શ્રી રજનીકાન્ત પંડ્યા

સમીક્ષા
શ્રી બીમલભાઈ રાવલ
શ્રી પંકજકુમાર શુક્લ
શ્રી રાજેશ્રીબહેન પડિયા
શ્રી નિશીતાબહેન ગાંધી
શ્રી સેજલબહેન ત્રિવેદી
ડૉ. વિમલભાઈ પંડ્યા
શ્રી હિમાંશુભાઈ મણિયાર

ભાષાશુદ્ધિ
શ્રી ઓમપ્રકાશ દવે

સંયોજન
શ્રી આશિષ એચ. બોરીસાગર
(વિષય-સંયોજક : ગણિત)

નિર્માણ-આયોજન
શ્રી સી. ડી. પંડ્યા
(નાયબ નિયામક : શૈક્ષણિક)

મુદ્રણ-આયોજન
શ્રી હરેશ એસ. લીમ્બાચીયા
(નાયબ નિયામક : ઉત્પાદન)

પ્રસ્તાવના

માધ્યમિક અને ઉચ્ચતર માધ્યમિક શિક્ષણ બોર્ડે ઓપન સોર્સ ઑપરેટિંગ સિસ્ટમ અને તેને સુસંગત વિવિધ મુદ્દાઓ માટેના કમ્પ્યૂટર-અધ્યનયને લગતા ઓપન સોર્સ સૉફ્ટવેર આધારિત નવો અભ્યાસક્રમ તૈયાર કર્યો છે. આ અભ્યાસક્રમ ગુજરાત સરકાર દ્વારા મંજૂર કરવામાં આવ્યો છે.

ગુજરાત સરકાર દ્વારા મંજૂર થયેલા **ધોરણ 12**ના **કમ્પ્યૂટર-અધ્યયન** વિષયના નવા અભ્યાસક્રમ અનુસાર તૈયાર કરવામાં આવેલું આ પાઠ્યપુસ્તક વિદ્યાર્થીઓ સમક્ષ મૂકતાં મંડળ આનંદ અનુભવે છે.

આ વિષયનું અંગ્રેજી માધ્યમનું પાઠ્યપુસ્તક પ્રસિદ્ધ કરતાં પહેલાં એની હસ્તપ્રતની આ સ્તરે શિક્ષણકાર્ય કરતા શિક્ષકો અને તજ્જ્ઞો દ્વારા સર્વાંગી સમીક્ષા કરાવવામાં આવી છે અને તેમનાં સૂચનો અનુસાર હસ્તપ્રતમાં યોગ્ય સુધારા-વધારા કર્યા પછી આ પાઠ્યપુસ્તક પ્રસિદ્ધ કરવામાં આવ્યું છે. અંગ્રેજી માધ્યમના પાઠ્યપુસ્તકનો આ ગુજરાતી અનુવાદ છે.

પ્રસ્તુત પાઠ્યપુસ્તકને વિષયવસ્તુલક્ષી, રસપ્રદ અને ક્ષતિરહિત બનાવવા માટે મંડળે પૂરતી કાળજી લીધી છે, તેમ છતાં શિક્ષણમાં રસ ધરાવનાર વ્યક્તિઓ પાસેથી પુસ્તકની ગુણવત્તા વધારે તેવાં સૂચનો આવકાર્ય છે.

ડૉ. ભરત પંડિત
નિયામક
તા. 7-10-2014

ડૉ. નીતિન પેથાણી
કાર્યવાહક પ્રમુખ
ગાંધીનગર

પ્રથમ આવૃત્તિ : 2014, પુનઃમુદ્રણ : 2014

પ્રકાશક : ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળ, 'વિદ્યાયન', સેક્ટર 10-એ, ગાંધીનગર વતી ભરત પંડિત, નિયામક

મુદ્રક :

મૂળભૂત ફરજો

ભારતના દરેક નાગરિકની ફરજ નીચે મુજબ રહેશે :*

(ક) સંવિધાનને વફાદાર રહેવાની અને તેના આદર્શો અને સંસ્થાઓનો, રાષ્ટ્રધ્વજનો અને રાષ્ટ્રગીતનો આદર કરવાની;

(ખ) આઝાદી માટેની આપણી રાષ્ટ્રીય લડતને પ્રેરણા આપનારા ઉમદા આદર્શોને હૃદયમાં પ્રતિષ્ઠિત કરવાની અને અનુસરવાની;

(ગ) ભારતનાં સાર્વભૌમત્વ, એકતા અને અખંડિતતાનું સમર્થન કરવાની અને તેમનું રક્ષણ કરવાની;

(ઘ) દેશનું રક્ષણ કરવાની અને રાષ્ટ્રીય સેવા બજાવવાની હાકલ થતાં, તેમ કરવાની;

(ચ) ધાર્મિક, ભાષાકીય, પ્રાદેશિક અથવા સાંપ્રદાયિક ભેદોથી પર રહીને, ભારતના તમામ લોકોમાં સુમેળ અને સમાન બંધુત્વની ભાવનાની વૃદ્ધિ કરવાની, સ્ત્રીઓના ગૌરવને અપમાનિત કરે, તેવા વ્યવહારો ત્યજી દેવાની;

(છ) આપણી સમન્વિત સંસ્કૃતિના સમૃદ્ધ વારસાનું મૂલ્ય સમજી તે જાળવી રાખવાની;

(જ) જંગલો, તળાવો, નદીઓ અને વન્ય પશુપક્ષીઓ સહિત કુદરતી પર્યાવરણનું જતન કરવાની અને તેની સુધારણા કરવાની અને જીવો પ્રત્યે અનુકંપા રાખવાની;

(ઝ) વૈજ્ઞાનિક માનસ, માનવતાવાદ અને જિજ્ઞાસા તથા સુધારણાની ભાવના કેળવવાની;

(ટ) જાહેર મિલકતનું રક્ષણ કરવાની અને હિંસાનો ત્યાગ કરવાની;

(ઠ) રાષ્ટ્ર પુરુષાર્થ અને સિદ્ધિનાં વધુ ને વધુ ઉન્નત સોપાનો ભણી સતત પ્રગતિ કરતું રહે એ માટે, વૈયક્તિક અને સામૂહિક પ્રવૃત્તિનાં તમામ ક્ષેત્રે શ્રેષ્ઠતા હાંસલ કરવાનો પ્રયત્ન કરવાની.

(ડ) માતા-પિતાએ અથવા વાલીએ 6 વર્ષથી 14 વર્ષ સુધીની વયના પોતાના બાળક અથવા પાલ્યને શિક્ષણની તકો પૂરી પાડવાની.

*ભારતનું સંવિધાન : કલમ 51-क

અનુક્રમણિકા

1. કમ્પોઝરનો ઉપયોગ કરી HTML ફૉર્મની રચના 1
2. કેસ્કેડિંગ સ્ટાઇલશીટ અને જાવાસ્ક્રિપ્ટ 25
3. કમ્પોઝરનો ઉપયોગ કરી સરળ વેબસાઇટની રચના 52
4. ઇ-કૉમર્સનો પરિચય 72
5. એમ-કૉમર્સનો પરિચય 90
6. ઑબ્જેક્ટ આધારિત ખ્યાલો 116
7. જાવાની મૂળભૂત બાબતો 127
8. જાવામાં ક્લાસ અને ઑબ્જેક્ટ 158
9. એરે અને સ્ટ્રિંગનો ઉપયોગ 186
10. જાવામાં અપવાદરૂપ પરિસ્થિતિનું વ્યવસ્થાપન 203
11. ફાઇલ-વ્યવસ્થાપન 220
12. લેટેક્સની મદદથી દસ્તાવેજનું પ્રકાશન 241
13. અન્ય ઉપયોગી નિઃશુલ્ક ટૂલ્સ અને સેવાઓ 260

આ પાઠ્યપુસ્તક વિશે...

પ્રિય શિક્ષકો,

કમ્પ્યૂટર-સાક્ષરતાને ખૂબ જ ઝડપથી વિસ્તારવાના ધ્યેય સાથે, ગુજરાત સરકારે ICT@School કાર્યક્રમ અંતર્ગત રાજ્યની 6000 કરતાં વધુ અનુદાનિત શાળાઓને અદ્યતન કમ્પ્યૂટર-સંસાધનો પૂરાં પાડ્યાં છે. નવી નીતિની પહેલરૂપે બધી શાળાઓને ઉબુન્ટુ (લિનક્સનું ભિન્ન સ્વરૂપ) ઑપરેટિંગ સિસ્ટમ અને અન્ય મુક્તપણે ઉપલબ્ધ નિઃશુલ્ક (ઓપનસૉર્સ) સૉફ્ટવેર પણ પૂરાં પાડવામાં આવ્યાં છે, જેથી લાઇસન્સની ચિંતા વગર મુક્ત રીતે સૉફ્ટવેર વાપરી શકે તેમજ તેની આપ-લે પણ કરી શકે. અગાઉનાં પાઠ્યપુસ્તકો મોટે ભાગે માલિકીહક ધરાવતા (પ્રોપ્રાઇટરી) સૉફ્ટવેર માટે લખાયાં હતાં, તેથી નવા અભ્યાસક્રમને આધારે નવેસરથી પાઠ્યપુસ્તકો તૈયાર કરવાની જરૂરિયાત હતી. 8મું ધોરણ પ્રાથમિક વિભાગમાં તબદીલ થવાને કારણે પણ આ ખૂબ જ જરૂરી હતું. આથી, ધોરણ 9થી 12 માટે કમ્પ્યૂટર-અધ્યયન માટેના વિવિધ વિષયો માટે મુક્તપણે ઉપલબ્ધ ઑપરેટિંગ સિસ્ટમ અને તેને અનુરૂપ સૉફ્ટવેર ટૂલ્સ પર આધારિત નવો અભ્યાસક્રમ તબક્કાવાર રીતે પૂરો પાડવામાં આવ્યો.

વિદ્યાર્થીઓ હવે ઑપરેટિંગ સિસ્ટમ, ઓપન ઑફિસના ઘટકો, HTML, સી પ્રોગ્રામિંગ, ઇન્ટરનેટ અને વેબસર્ફિંગ જેવા મૂળભૂત કમ્પ્યૂટર સૉફ્ટવેરથી માહિતગાર છે. 'કમ્પ્યૂટર-અધ્યયન' વિષયની શ્રેણીમાં 12મા ધોરણનું આ ચોથું પાઠ્યપુસ્તક છે. આ પુસ્તકનો હેતુ HTML ફૉર્મની રચના, જાવાસ્ક્રિપ્ટ, કમ્પોઝરનો ઉપયોગ કરી સરળ વેબસાઇટની રચના અને ઇ-કૉમર્સ તથા એમ-કૉમર્સનો પરિચય આપવાનો છે. આ ઉપરાંત, ઑબ્જેક્ટ ઓરિએન્ટેડ અભિગમના પરિચય બાદ મૂળભૂત જાવા ભાષા સમજાવવામાં આવી છે, જે વેબ-વિનિયોગો માટે વ્યાપકપણે ઉપયોગમાં લેવાતી ભાષા હોવાને કારણે ઘણી પ્રચલિત છે. અંતમાં, વિદ્યાર્થીઓની જાણકારી માટે લેટેક્સ જેવા પ્રચલિત ઓપનસૉર્સ પૅકેજ અને અન્ય પ્રકીર્ણ ટૂલ્સનો પરિચય આપવામાં આવ્યો છે.

અમે આશા રાખીએ છીએ કે, આવરી લીધેલો અભ્યાસક્રમ વિદ્યાર્થીઓને કમ્પ્યૂટર ઍપ્લિકેશન્સ અંગે સ્પષ્ટ સમજ મેળવવામાં ઉપયોગી બનશે અને આપ ઓપનસૉર્સ સૉફ્ટવેર ટૂલ્સનો ઉપયોગ કરી શિક્ષણ આપવામાં અને પ્રાયોગિક કાર્યમાં આનંદ અનુભવશો.

પ્રિય વિદ્યાર્થીઓ,

ધોરણ 9 થી 11માં તમે ઉબુન્ટુ ઓપરેટિંગ સિસ્ટમ, રાઇટર, કેલ્સી, ઇમ્પ્રેસ જેવા ઓપન ઑફિસ ઘટકો, બેઝ જેવા ડેટાબેઝ મેનેજમેન્ટ ટૂલ તથા HTMLમાં વેબપેજની રજૂઆત વિશે અભ્યાસ કર્યો. અલ્ગોરિધમની રચના અને સાદા સી પ્રોગ્રામિંગ વિશે પણ તમે જાણ્યું, જેને પ્રોસિજર આધારિત પ્રોગ્રામિંગ ભાષા કહેવામાં આવે છે. આ ઉપરાંત તમારી સમક્ષ એનિમેશન માટેના મલ્ટીમીડિયા ટૂલ્સને પણ રજૂ કરવામાં આવ્યું. ધોરણ 12ના આ પાઠ્યપુસ્તકમાં કમ્પોઝરનો ઉપયોગ કરી વેબપેજની રચના, ઇ-કૉમર્સ, ઑબ્જેક્ટ ઓરિએન્ટેડ અભિગમના પાયાના સિદ્ધાંતો, જાવા પ્રોગ્રામિંગ ભાષા તથા અન્ય ઉપયોગી ટૂલ્સ રજૂ કરવામાં આવ્યાં છે.

પ્રકરણ 1થી 3ની મદદથી તમે સરળ છતાં કાર્યક્ષમ વેબસાઇટની રચના કરવાનો પ્રયત્ન કરી શકશો. આ પ્રકરણોમાં HTML ફૉર્મ, સ્ટાઇલ શીટ, જાવાસ્ક્રિપ્ટ અને વેબસાઇટની રચના માટે કમ્પોઝર નામના ટૂલની ચર્ચા કરવામાં આવી છે. પ્રકરણ 4 અને 5માં ઇ-કૉમર્સ, એમ-કૉમર્સ અને એલ-કૉમર્સ વિશે જાણકારી આપવામાં આવી છે. આજકાલ સૉફ્ટવેરના વિકાસમાં મહદંશે ઑબ્જેક્ટ ઓરિએન્ટેડ પદ્ધતિનો ઉપયોગ કરવામાં આવે છે. ઑબ્જેક્ટ ઓરિએન્ટેડ અભિગમનો અભ્યાસ તમે પ્રકરણ 6માં કરશો. વેબ-વિકાસમાં અગ્રસર ઑબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામિંગ ભાષાનો પરિચય પ્રકરણ 7માં આપવામાં આવ્યો છે. ક્લાસ અને ઑબ્જેક્ટ, ઍરે અને સ્ટ્રિંગ, અપવાદરૂપ પરિસ્થિતિ અને ફાઇલનું વ્યવસ્થાપન જેવી સુવિધાઓ પ્રકરણ 8 થી 11માં સમાવવામાં આવી છે. એક અન્ય ઓપનસૉર્સ શબ્દપ્રક્રિયક LaTex કે જે પુસ્તકના લેખકો, પ્રકાશકો અને વૈજ્ઞાનિકોમાં સવિશેષ પ્રચલિત છે તેનો પરિચય પ્રકરણ 12માં આપવામાં આવ્યો છે. અંતિમ પ્રકરણમાં કમ્પ્યૂટર સાથે કાર્ય કરી શકાય તેવાં કેટલાંક અન્ય ઉપયોગી ઓપનસૉર્સ ટૂલ્સ જેવાં કે, આર્કાઇવ મેનેજર, VLC મીડિયા પ્લેયર, ગૂગલ મેપ્સ, આંકડાશાસ્ત્રીય ટૂલ R, સ્કાઇપ અને રેશનલ પ્લાન વગેરેને તેમનાં સામાના કાર્યોની ઉપયોગિતા સાથે સંક્ષેપમાં ચર્ચવામાં આવ્યાં છે.

તમે આ પુસ્તકનો કાળજીપૂર્વક અભ્યાસ કરી પ્રાયોગિક કાર્યમાં પ્રવૃત્ત થાઓ તથા વેબસાઇટની અને જાવાના પ્રોગ્રામની રચનામાં સંપૂર્ણ આત્મવિશ્વાસ કેળવી શકો એ અપેક્ષિત છે.

•

કમ્પોઝરનો ઉપયોગ કરી HTML ફૉર્મની રચના 1

ઈન્ટરનેટનો ઉપયોગ વધવાથી અનેક પ્રવૃત્તિઓ ઑનલાઇન બની છે. આપણે પોતાની કે કોઈ વસ્તુની માહિતી રજૂ કરવા માટે વેબપેજનો ઉપયોગ કરતા હોઈએ છીએ. વેબસાઇટનો મુલાકાતી પોતાની વિગતો દાખલ કરવા માટે HTML ફૉર્મની મદદ લે છે. ફૉર્મ મુલાકાતીની વિગતો દાખલ કરવા વિશે વધુ સંવાદન (interactivity) અને નિયંત્રણ પૂરાં પાડે છે. ઉદાહરણ તરીકે, જો તમે કોઈ વેબસાઇટ પર મેઇલ-એકાઉન્ટ ખોલવા કે નોંધણી કરાવવા ઇચ્છતા હો, તો તે માટે તમારે ફૉર્મમાં તમારી અંગત વિગતો ભરવાની જરૂર પડે છે. ઉપયોગકર્તાના મેઇલ-એકાઉન્ટને ગોઠવવા માટે આ વિગતોનો ઉપયોગ કરવામાં આવે છે. ત્યાર પછી વિનિયોગ (application) દ્વારા આ વિગતોનો સંગ્રહ કરવામાં આવે છે અને વેબસાઇટ પર નોંધાયેલા ઉપયોગકર્તાઓ વિશેની વિગતો મેળવવા માટે તેનો ઉપયોગ કરવામાં આવે છે.

ફૉર્મ એક સંગ્રાહક (container) છે, જેનો ઉપયોગ ઉપયોગકર્તા પાસેથી જુદા-જુદા પ્રકારની વિગતો મેળવવા માટે કરવામાં આવે છે. HTML ફૉર્મમાં લેબલ (label), ચેકબૉક્સ (checkbox), લખાણ ઉમેરવા માટેનું ક્ષેત્ર (text input field), રેડિયો-બટન (radio button), સબમિટ બટન (submit button), રિસેટ બટન (reset button) અને આવા અનેક ઘટકો ઉમેરવામાં આવે છે. આ ઘટકોનો ઉપયોગ ફૉર્મમાં વિગતો દાખલ કરવા તથા વિગતોની યથાર્થતા ચકાસવા માટે પણ કરવામાં આવે છે. HTML ટેગનો ઉપયોગ કરી આપણે એક સરળ ફૉર્મની રચના કરીશું, પરંતુ તે પહેલાં HTML ફૉર્મની રચના કરવા માટે ઉપયોગી એવા ઘટકોની ચર્ચા કરીએ. આ ઘટકો નીચેના વિભાગમાં વર્ણવ્યા છે :

- ફૉર્મ (Form)
- ઇનપુટ (Input)
- ટેક્સ્ટ-એરિયા (Text Area)
- સિલેક્ટ અને ઑપ્શન (Select and Option)

ફૉર્મના ઘટકો (Form Elements)

HTML ફૉર્મની રચના કરવા માટે form ઘટકનો ઉપયોગ કરવામાં આવે છે. ફૉર્મમાં સમાવિષ્ટ તમામ ઘટકોના સંગ્રાહક તરીકે તે કાર્ય કરે છે. આ ઘટકને અમલમાં મૂકવા માટે <form>... </form> ટેગનો ઉપયોગ કરવામાં આવે છે. નીચે form ઘટકનું ઉદાહરણ દર્શાવ્યું છે.

```
<form action="register.html" method="post">
.
.
નિવેશ-ઘટકો (input elements)
.
.
</form>
```

form ઘટક action અને method નામની બે લાક્ષણિકતાઓ (attributes)નો ઉપયોગ કરે છે. જ્યારે ફૉર્મ સબમિટ કરવામાં આવે, ત્યારે તેની વિગતો ક્યા સ્થાને મોકલવી છે, તે સ્પષ્ટ કરવા માટે action લાક્ષણિકતાનો ઉપયોગ કરવામાં આવે છે. કિંમત તરીકે તે ફાઇલનું નામ સ્વીકારે છે. ફૉર્મમાં વિગતો ઉમેર્યા પછી જ્યારે ઉપયોગકર્તા સબમિટ બટન પર ક્લિક કરે છે, ત્યારે આ ફાઇલ ખોલવામાં આવે છે.

વિગતો મોકલતી વખતે method લાક્ષણિક્તાના ઉપયોગ દ્વારા HTTP પદ્ધતિ સ્પષ્ટ કરવામાં આવે છે. તે બે કિંમત સ્વીકારી શકે છે : GET અને POST. GET પદ્ધતિ ફૉર્મમાંથી વિગતો મેળવી, તેને URLના અંતમાં ઉમેરી સર્વરને મોકલી આપે છે. આ પદ્ધતિ એક સમયે માત્ર મર્યાદિત માહિતી મોકલવાની અનુમતિ આપે છે. POST પદ્ધતિમાં વિગતોને HTTP ટ્રાન્ઝેક્શન દ્વારા બ્લૉક સ્વરૂપે મોકલવામાં આવે છે. વિગતોને વિનંતીમાં સમાવીને મોકલવામાં આવે છે. આ પદ્ધતિમાં વિગતોની લંબાઈ બાબતે કોઈ મર્યાદા નથી. method લાક્ષણિક્તાની પૂર્વનિર્ધારિત કિંમત GET છે.

નિવેશ-ઘટક (Input Element)

ફૉર્મમાં રેડિયો-બટન, ટૅક્સ્ટબૉક્સ અને ચેકબૉક્સ જેવાં વિવિધ ક્ષેત્ર (field) ઉમેરવા માટે નિવેશ-ઘટકોનો ઉપયોગ કરવામાં આવે છે. <input> ....</input> અથવા માત્ર <input> ટૅગનો ઉપયોગ કરી આ ઘટક અમલમાં મૂકી શકાય છે. <input> ટૅગનો ઉપયોગ type, name અને value જેવી લાક્ષણિક્તાઓ સાથે કરવામાં આવે છે.

ફૉર્મમાં ઉમેરવામાં આવનાર ફિલ્ડની સ્પષ્ટતા input ઘટક સાથે type લાક્ષણિક્તા દ્વારા કરવામાં આવે છે. ફૉર્મમાં આવેલા ફિલ્ડને નામ આપવા માટે name લાક્ષણિક્તાનો ઉપયોગ કરવામાં આવે છે. ફૉર્મમાં ફિલ્ડને પૂર્વનિર્ધારિત કિંમત સાથે દર્શાવવા માટે value લાક્ષણિક્તાનો ઉપયોગ કરવામાં આવે છે. કોષ્ટક 1.1માં આ લાક્ષણિક્તાઓને તેના ઉપયોગ સાથે દર્શાવી છે.

Type	સમજૂતી	Example
Radio	ફૉર્મમાં રેડિયો-બટનની રચના કરે છે. રેડિયો-બટનના જૂથમાંથી એક સમયે માત્ર કોઈ પણ એક રેડિયો-બટન પસંદ કરી શકાય છે. સામાન્ય રીતે વિકલ્પોનાં જૂથમાંથી એક વિકલ્પ પસંદ કરવા માટે રેડિયો-બટનનો ઉપયોગ કરવામાં આવે છે.	<INPUT TYPE = "radio" NAME = "var" VALUE = "txt">
Checkbox	ફૉર્મમાં ચેકબૉક્સની રચના કરે છે. એકસાથે એકથી વધુ ચેકબૉક્સની પસંદગી શક્ય છે. આપેલ વિકલ્પોનાં જૂથમાં એકથી વધુ વિકલ્પ પસંદ કરવા માટે ચેકબૉક્સનો ઉપયોગ કરવામાં આવે છે.	<INPUT TYPE="checkbox" NAME = "var" VALUE ="txt" >
Text	ફૉર્મમાં લખાણ ઉમેરવા માટે ટૅક્સ્ટ-ફિલ્ડની રચના કરે છે. ટૅક્સ્ટ ફિલ્ડમાં ઉપયોગકર્તા ઇચ્છિત કોઈ પણ વિગતો ઉમેરી શકે છે.	<INPUT TYPE="text" NAME = "var" VALUE = "txt" >
Password	ફૉર્મમાં પાસવર્ડ ફિલ્ડની રચના કરે છે. આ ફિલ્ડ ટૅક્સ્ટ-ફિલ્ડ સમાન છે, પરંતુ તેમાં ઉમેરવામાં આવેલ અક્ષરો ઉપયોગકર્તા સમક્ષ દર્શાવવામાં આવતા નથી. તેને બદલે અક્ષરોનું અવાચ્ય સ્વરૂપમાં રૂપાંતરણ કરવામાં આવે છે.	<INPUT TYPE="password" NAME = "var" >
Submit	ફૉર્મમાં સબમિટ બટનની રચના કરે છે. સબમિટ બટન પર ક્લિક કરવામાં આવે, ત્યારે ફૉર્મમાં દાખલ કરવામાં આવેલ વિગતો ફૉર્મ ઘટક સાથે ઉપયોગમાં લેવામાં આવેલ action લાક્ષણિક્તાને આપવામાં આવેલ ફાઈલ તરફ મોકલવામાં આવે છે.	<INPUT TYPE="submit" VALUE = "label" >
Reset	ફૉર્મમાં રિસેટ બટનની રચના કરે છે. રિસેટ બટન પર ક્લિક કરવાથી ફૉર્મમાં દાખલ કરવામાં આવેલ વિગતો દૂર કરી ફૉર્મને પૂર્વનિર્ધારિત કિંમતો સાથે દર્શાવવામાં આવે છે.	<INPUT TYPE = "reset" VALUE = "label" >

કોષ્ટક 1.1: input ટૅગ સાથે ઉપયોગમાં લેવામાં આવતા type લાક્ષણિક્તાની કિંમતો

ટેક્સ્ટ-એરિયા ઘટક (Textarea Element)

ટેક્સ્ટ-એરિયા ઘટક એકથી વધુ લીટીનું લખાણ નિવેશ કરવાની સુવિધા પૂરી પાડે છે. આ ઘટકનો અમલ કરવા માટે <textarea>...</textarea> ટેગનો ઉપયોગ કરવામાં આવે છે. તેમાં અમર્યાદિત અક્ષરો ઉમેરી શકાય છે. નોંધ (Comment), અહેવાલ (report) કે લાંબી સમજૂતી (long description) ઉમેરવા માટે તેનો ઉપયોગ કરી શકાય છે. ટેક્સ્ટ-એરિયા ઘટકનું કદ rows અને cols લાક્ષણિકતાઓ દ્વારા બદલી શકાય છે. ટેક્સ્ટ-એરિયામાં ઉપર કે નીચે ગયા (scroll) વિના જોઈ શકાય તેવી હરોળોની સંખ્યા નક્કી કરવા માટે rows લાક્ષણિકતાનો ઉપયોગ કરવામાં આવે છે. ટેક્સ્ટ-એરિયામાં ડાબી કે જમણી બાજુ ગયા (scroll) વિના જોઈ શકાય તેવા સ્તંભની સંખ્યા નક્કી કરવા માટે cols લાક્ષણિકતાનો ઉપયોગ કરવામાં આવે છે. ફૉર્મમાં ટેક્સ્ટ-એરિયા ઉમેરવાની રીત નીચેનાં ઉદાહરણમાં દર્શાવી છે.

```
<form method="post" action="comment.html">

Input your comments: <br /> <textarea name="comments" rows="4" cols="20">

...Your comments here...

</textarea>

</form>
```

સિલેક્ટ અને ઑપ્શન ઘટક (Select and Option Element)

ફૉર્મમાં ડ્રૉપડાઉન યાદી (dropdown list) કે મેનુની રચના કરવા માટે select ઘટકનો ઉપયોગ કરવામાં આવે છે. મેનુમાં દર્શાવવામાં આવનાર કિંમતો ઉમેરવા માટે option ઘટકનો ઉપયોગ કરવામાં આવે છે. ડ્રૉપડાઉન મેનુની રચના કરવા માટે <select>....</select> ટેગનો ઉપયોગ કરવામાં આવે છે. મેનુના ઘટકોની રચના કરવા માટે <option>... </option> ટેગનો ઉપયોગ કરવામાં આવે છે. નીચેનું ઉદાહરણ select અને option ઘટકનો ઉપયોગ દર્શાવે છે.

```
<select>
        <option value="Ahmedabad" >Ahmedabad</option>
        <option value="Rajkot" >Rajkot</option>
        <option value="Surat" >Surat </option>
</select>
```

હવે આપણે, અત્યાર સુધીમાં શીખેલા ઘટકોનો ઉપયોગ કરી નોંધણી માટેના એક નમૂનારૂપ ફૉર્મની રચના કરીએ. કોડ-લિસ્ટિંગ 1.1માં ફૉર્મ બનાવવા માટેનો HTML કોડ દર્શાવ્યો છે. કોડનું પરિણામ આકૃતિ 1.1માં દર્શાવવામાં આવ્યું છે.

```
<HTML>
<HEAD>
        <TITLE>Registration Form</TITLE>
</HEAD>
<BODY bgcolor="lightblue">
<h1> <center>Registration Form</center></h1>
<FORM name="frmRegistration" action="form.html">
        <center>
        <TABLE BORDER="0">
<TR>
        <TD width="12%">First Name</TD>
        <TD width="1%"> </TD>
```

```
        <TD> <INPUT type="textbox" name="txtFirstName"></TD>
</TR>
<TR>
        <TD>Middle Name</TD>
        <TD> </TD>
        <TD><INPUT type="text box" name="txtMiddleName"></TD>
</TR>
<TR>
        <TD>Last Name</TD>
        <TD> </TD>
        <TD> <INPUT type="text box" name="txtLastName"></TD>
</TR>
<TR>
        <TD>Gender</TD>
        <TD> </TD>
        <TD>
        <INPUT type="radio" name="Gender" value="male" CHECKED>Male
        <INPUT type="radio" name="Gender" value="female" >Female
        </TD>
</TR>
<TR>
        <TD>Hobby</TD>
        <TD> </TD>
        <TD>
        <INPUT type="checkbox" name="chkSinging" value="Sing" CHECKED>Singing
        <INPUT type="checkbox" name="chkDancing" value="Dance">Dancing
        <INPUT type="checkbox" name="chkReading" value="Read">Reading
        </TD>
</TR>
<TR>
        <TD>Address</TD>
        <TD> </TD>
        <TD>
        <Textarea name="txtAddress" rows="5" cols="70">Insert Address Here</Textarea>
        </TD>
</TR>
<TR>
        <TD>City</TD>
        <TD> </TD>
        <TD>
```

```
				<Select Name="cmbCity">
				<Option >Ahmedabad</Option>
				<Option >Baroda</Option>
				<Option selected>Rajkot</Option>
				<Option >Surat</Option>
				</Select>
				</TD>
		</TR>
		<TR>
				<TD> </TD>
				<TD> </TD>
				<TD> <INPUT type="submit" name="cmdSubmit" value="Submit">
				<INPUT  type="reset"  name="cmdReset"  value="Reset">
				</TD>
		</TR>
		</TABLE>
		</center>
</FORM>
</BODY>
</HTML>
```

કોડલિસ્ટિંગ 1.1 : નોંધણી માટેનું નમૂનારૂપ ફૉર્મ બનાવવા માટેનો HTML કોડ

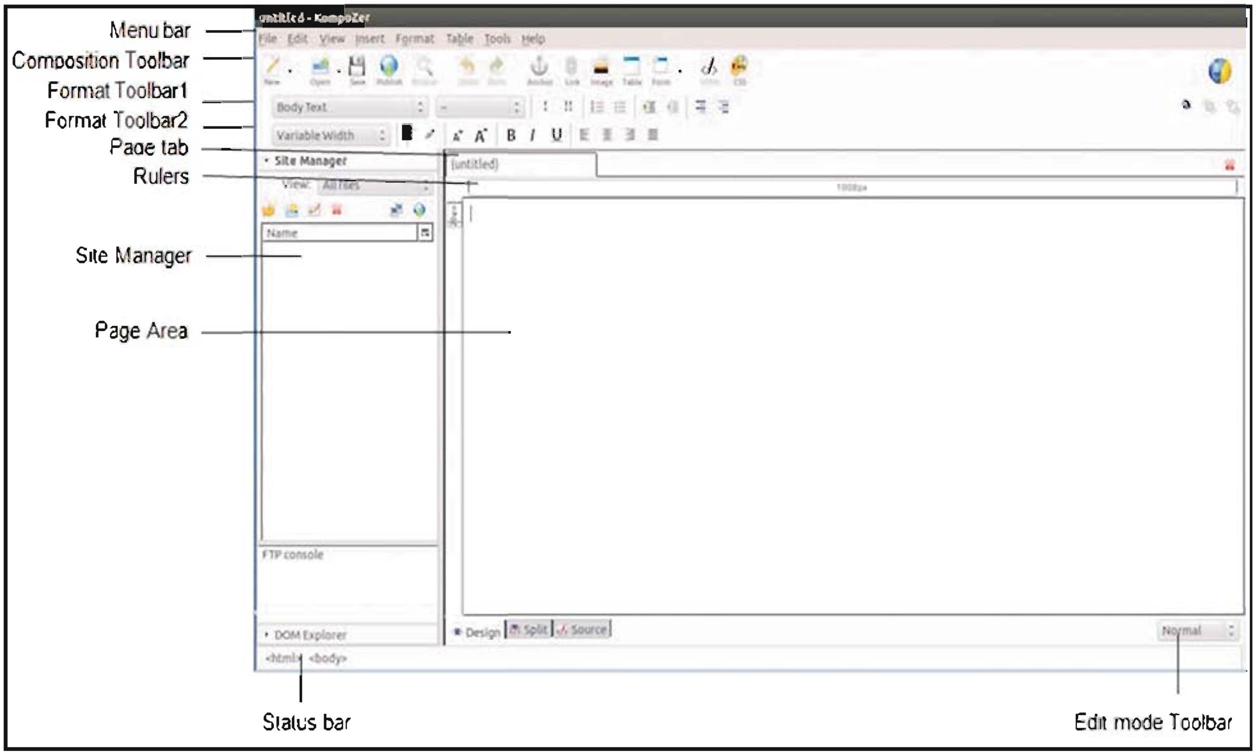
આકૃતિ 1.1 : વેબબ્રાઉઝરમાં દર્શાવેલું નોંધણી માટેનું ફૉર્મ

તમે જોઈ શકો છો કે HTML ટૅગનો ઉપયોગ કરી ફૉર્મની રચના કરવી તે એક કંટાળાજનક પ્રક્રિયા છે. IDE (Integrated Development Environment)નો ઉપયોગ કરવો તે આ માટેની એક સરળ રીત છે. IDE એક એવો સૉફ્ટવેર વિનિયોગ છે, જે પ્રોગ્રામરને સૉફ્ટવેરનો વિકાસ કરવા માટેની સંપૂર્ણ સુવિધા પૂરી પાડે છે. તે GUI (Graphical User Interface), ટૅક્સ્ટ કે કોડ-એડિટર (text/code editor), કંપાઈલર કે ઇન્ટરપ્રિટર (compiler/interpreter) અને ડીબગર (debugger) પૂરા પાડે છે. કમ્પોઝર (KompoZer); એક્લિપ્સ (Eclipse), જેબિલ્ડર (JBuilder) અને નેટબિન્સ (Netbeans) એ કેટલાંક ઓપનસોર્સ IDEનાં ઉદાહરણ છે. હવે, વેબપેજની રચના કરવા કમ્પોઝરના ઉપયોગની ચર્ચા કરીએ.

કમ્પોઝરનો પરિચય (Introduction to KompoZer)

કમ્પોઝર એ વેબવિકાસ માટેનું એક નિઃશુલ્ક અને ઑપનસૉર્સ IDE છે. તે http://www.KompoZer.net પરથી ડાઉનલોડ કરી શકાય છે. તે WYSIWYG "What You See Is What You Get" તરીકે ઓળખાતાં માધ્યમ સાથે વેબપેજ એડિટર પૂરું પાડે છે. તે એક સંપૂર્ણ વેબ-ઑથરિંગ સિસ્ટમ (Web Authoring System) છે, જે વેબપેજના વિકાસ અને વેબફાઇલના વહીવટનું સંચાલન કરે છે. કમ્પોઝર દ્વારા વેબપેજની રચના કરવી એ ઝડપી અને સરળ કાર્ય છે. તદુપરાંત, ઉપયોગકર્તા સૉર્સકોડનો ઉપયોગ કરીને તથા તેમાં સુધારા કરીને વેબપેજમાં ફેરફારો પણ કરી શકે છે. કમ્પોઝરમાં સાઇટ-મૅનેજર (Site-Manager)નો સમાવેશ કરવામાં આવ્યો છે, જે સ્થાનિક (local) તથા દૂરસ્થિત (remote) સર્વર એમ બંને સ્થાને રહેલ ફાઇલોનો ઝડપી ઉપયોગ પૂરો પાડે છે. કમ્પોઝરનો ઉપયોગ કરી વેબપેજ તથા તેને સંલગ્ન ફાઇલોને દૂરસ્થિત સર્વર પર અપલોડ કરી શકાય છે. તે કેસ્કેડિંગ સ્ટાઇલશીટ (Cascading Style Sheet) દ્વારા 'સ્ટાઇલ' (Style)ને પણ સમર્થન પૂરું પાડે છે. CSS વિશે આપણે હવે પછીના પ્રકરણમાં અભ્યાસ કરીશું.

કમ્પોઝરની મદદથી ફૉર્મની રચના વિશેનો અભ્યાસ કરતાં પહેલાં, સૌપ્રથમ કમ્પોઝરના ઇન્ટરફેસને સમજીએ. કમ્પોઝરને તેના આઇકનની મદદથી શરૂ કરો. જુદાં જુદાં ટૂલબાર અને સ્ટેટસબાર (જો દેખાતાં ન હોય તો) દર્શાવવા માટે **View → Show/Hide** વિકલ્પ પર ક્લિક કરો. યાદીમાં આપેલ Composition Toolbar, Format Toolbar1, Format Toolbar2, Edit Mode Toolbar અને Status bar એ તમામ વિકલ્પોને પસંદ કરો. Site-Manager અને Rulers વિકલ્પને પણ પસંદ કરો. ટૂલબાર પસંદ કર્યા પછીની વિન્ડો આકૃતિ 1.2માં દર્શાવી છે.



આકૃતિ 1.2 : કમ્પોઝર ઇન્ટરફેસ

આકૃતિ 1.2માં File, Edit, View, Insert, Format, Table, Tools અને Help જેવા વિકલ્પો ધરાવતો મેનુબાર વિન્ડોની ઉપરના ભાગમાં જોઈ શકાય છે. મેનુબારની નીચે ત્રણ ટૂલબાર આવેલા છે : Composition, Format Toolbar1 અને Format Toolbar2.

નવી ફાઇલની રચના કરવા, ફાઇલ ખોલવા, ફાઇલનો સંગ્રહ કરવા તથા વેબસાઇટને પ્રકાશિત (publish) કરવા માટે કમ્પોઝિશન ટૂલબારનો ઉપયોગ કરવામાં આવે છે. લખાણની ગોઠવણી કરવા, નિશાની (bullet) અને અનુક્રમ (Numbering) ઉમેરવા તથા ગોઠવણીને લગતાં અન્ય કાર્ય કરવા માટે ફૉર્મેટ ટૂલબાર-1 અને ફૉર્મેટ ટૂલબાર-2નો ઉપયોગ કરવામાં આવે છે.

વિન્ડોની મધ્યમાં બે વિભાગ જોઈ શકાય છે : સાઈટ-મૅનેજર અને ખાલી વેબપેજ. સાઈટ-મૅનેજર એ સાઈટમાં અથવા એકથી વધુ સાઈટ વચ્ચેના નેવિગેશન માટે ઉપયોગમાં લઈ શકાય તેવું એક સક્ષમ સાધન છે. Close બટન પર ક્લિક કરીને અથવા તો F9 કી દબાવીને સાઈટ-મૅનેજરના વિભાગને બંધ કરી શકાય છે. Pagepane એક ખાલી અને નામ વગરનું (Untitled) વેબપેજ દર્શાવે છે. વિન્ડોની નીચે આવેલા જમણી બાજુના ભાગમાં ત્રણ મોડ સાથેનો Edit mode ટૂલબાર આપવામાં આવે છે : Normal, HTML Tags અને Preview. આ ત્રણે મોડ સુધારા માટેની સુવિધા પૂરી પાડે છે.

પ્રિવ્યૂ મોડ બ્રાઉઝરમાં જોઈ શકાય તે પ્રકારનો પાનાનો દેખાવ દર્શાવે છે. તફાવત એ છે કે પ્રિવ્યૂ મોડમાં સ્ક્રિપ્ટનો અમલ કરવામાં આવતો નથી અને તેથી તેની અસર દર્શાવવામાં આવતી નથી. પ્રિવ્યૂ મોડમાં લિંકનો અમલ પણ કરી શકાતો નથી.

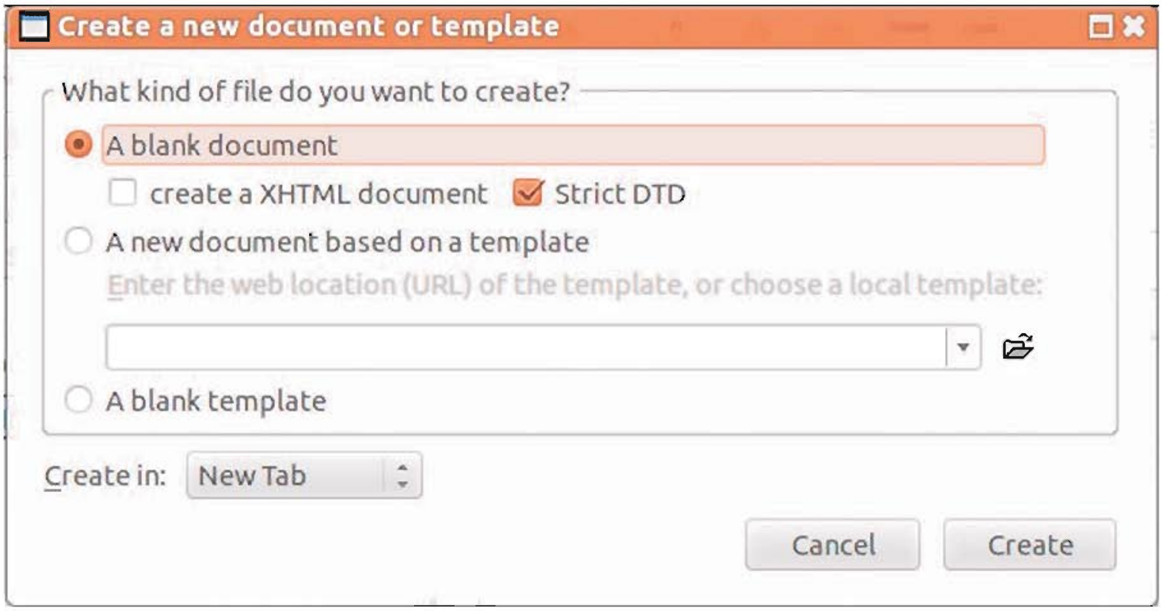
નૉર્મલ વ્યૂ એ પ્રિવ્યૂ મોડ જેવો જ છે. આ મોડમાં કોષ્ટકની સીમારેખા દર્શાવવામાં આવે છે. જેઓ HTMLથી પરિચિત છે તેવા ઉપયોગકર્તાને HTML ટેગ વ્યૂ મદદરૂપ બને છે. તમામ ટેગની શરૂઆત દર્શાવવા માટે પીળા રંગના નિશાનનો ઉપયોગ કરવામાં આવે છે. આ નિશાન પર ક્લિક કરવાથી ઘટકના સમગ્ર વિભાગને પસંદ કરી હાઈલાઈટ (Highlight) કરી શકાય છે.

પેજપેનની ડાબી બાજુ Design, Split અને Source વિભાગો (tabs) દર્શાવવામાં આવ્યા છે. વેબપેજની રૂપરેખા તૈયાર કરવા માટે Design વિભાગનો ઉપયોગ કરવામાં આવે છે. વર્તમાન ઘટકના HTML સૉર્સને Split વિભાગમાં દર્શાવવામાં આવે છે. HTML કોડની તમામ વિગતો Source વિભાગમાં દર્શાવવામાં આવે છે જે સૉર્સકોડ સુધારવામાં મદદરૂપ બને છે.

વિન્ડોના નીચેના ભાગમાં સ્ટેટસબાર જોઈ શકાય છે. જ્યારે પાનાંમાં કોઈ પણ વસ્તુ પર ક્લિક કરવામાં આવે છે, ત્યારે તેનું માળખું સ્ટેટસબારમાં દર્શાવવામાં આવે છે. જો આપણે કોઈ પણ ટૂલબારની ગોઠવણી બદલવા માંગતા હોઈએ, તો તે ટૂલબાર પર રાઈટ ક્લિક કરી Customize Toolbar વિકલ્પ પર ક્લિક કરી શકાય. ત્યાર પછી તે ટૂલબારને આપણી પસંદગી પ્રમાણે ગોઠવી શકાશે.

નવી ફાઈલ બનાવવી (Create a New File)

નવી ફાઈલ બનાવવા માટે કમ્પોઝર શરૂ કરો. મેનુબારમાં **File → New** વિકલ્પ પર ક્લિક કરો. આમ કરવાથી આકૃતિ 1.3માં દર્શાવ્યા મુજબ "Create a new document or template" શીર્ષક સાથે એક ડાયલોગબૉક્સ ખૂલશે. ડાયલોગબૉક્સમાં આપેલા વિકલ્પો પૈકી "A blank document" વિકલ્પ પસંદ કરો. ડાયલોગબૉક્સના નીચેના ભાગમાં "Create in" લેબલ જોઈ શકાશે. તેની બાજુમાં આવેલા ડ્રૉપડાઉન મેનુમાંથી New Tab વિકલ્પ પસંદ કરો. તે નવા વિભાગ (Tab)માં વેબપેજની રચના કરશે. Create બટન પર ક્લિક કરો.



આકૃતિ 1.3 : નવી ફાઈલની રચના

હયાત ફાઈલ ખોલવી (Open an Existing File)

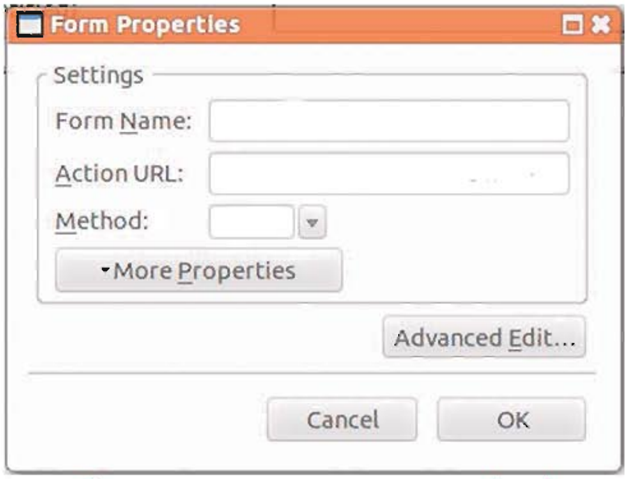
હયાત ફાઈલ ખોલવા માટે કમ્પોઝિશન ટૂલબાર પર આવેલા

આઇકન પર ક્લિક કરો. વૈકલ્પિક રીતે, **File → Open** વિકલ્પ પણ પસંદ કરી શકાય. જો હાલમાં જ ખોલવામાં આવી હોય, તો તેવી ફાઇલને **File → Recent Pages** વિકલ્પની મદદથી પણ ખોલી શકાય છે.

હવે, કમ્પોઝરની મદદથી ફૉર્મની રચના કરતા શીખીએ. આપણે એક સરળ ફૉર્મ બનાવીએ; જેમાં બે ઇનપુટ ફિલ્ડ હોય : નામ અને ઇ-મેઇલ સરનામું તથા સબમિટ બટન. ફૉર્મની રચના કરવા માટે નીચેનાં પગલાંને અનુસરો :

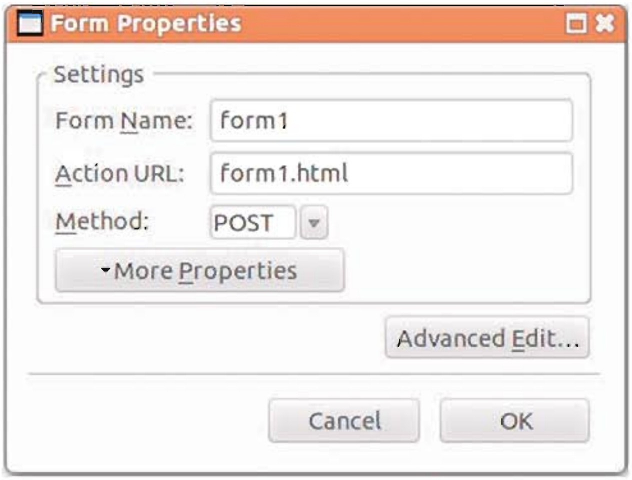
- કમ્પોઝર શરૂ કરો. નવી ફાઇલ બનાવો.
- મેનુબારમાંથી **Insert → Form → Define Form** વિકલ્પ પસંદ કરો. વૈકલ્પિક રીતે, કમ્પોઝિશન ટૂલબારમાં આવેલ

બટન પર ક્લિક કરી શકાય. આમ કરવાથી આકૃતિ 1.4માં આવેલ Form Properties ડાયલોગ-બૉક્સ રજૂ કરવામાં આવશે. More Properties પર ક્લિક કરી અતિરિક્ત વિકલ્પો દર્શાવી શકાશે.



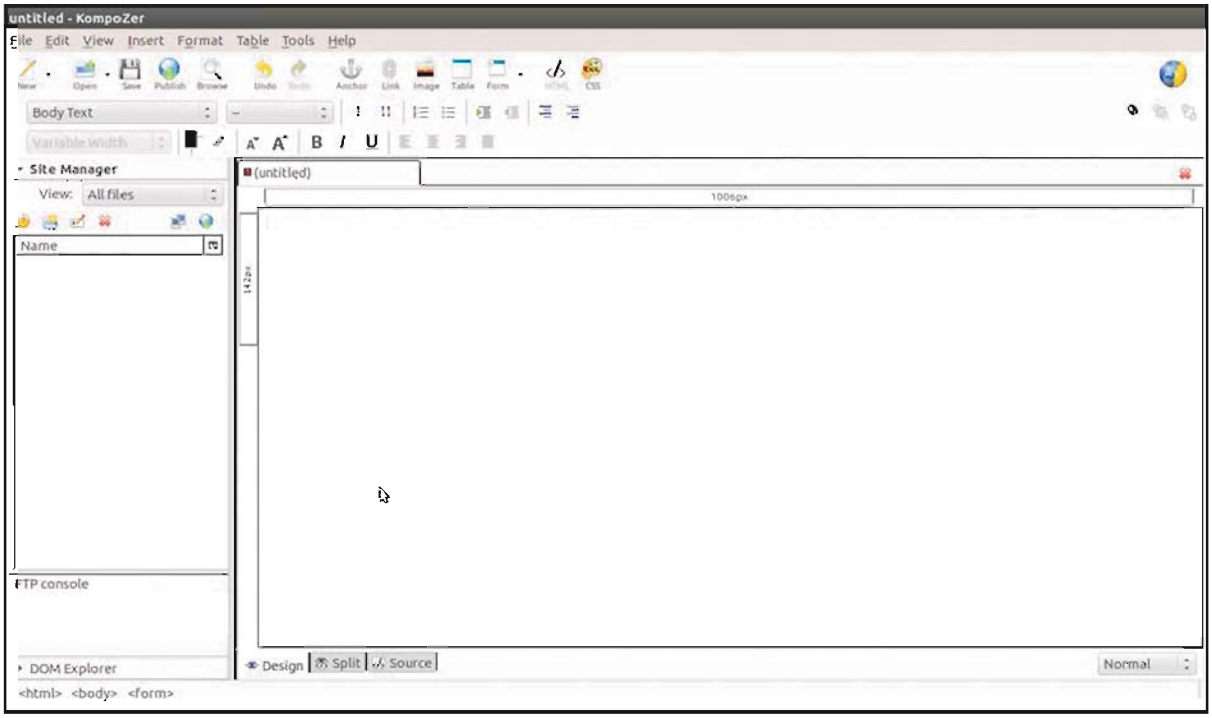
આકૃતિ 1.4 : Form Properties ડાયલોગબૉક્સ

- ફૉર્મ માટે યોગ્ય નામ ઉમેરો. ફૉર્મની વિગતો મોકલવાની હોય તે ફાઇલનું નામ Action URL વિકલ્પની કિંમત સ્વરૂપે ઉમેરો. Method ડ્રૉપડાઉન મેનુમાંથી POST પદ્ધતિ પસંદ કરો. અને OK બટન પર ક્લિક કરો. Form Properties ડાયલોગબૉક્સમાં ઉમેરવામાં આવેલી વિગતો આકૃતિ 1.5માં દર્શાવી છે.



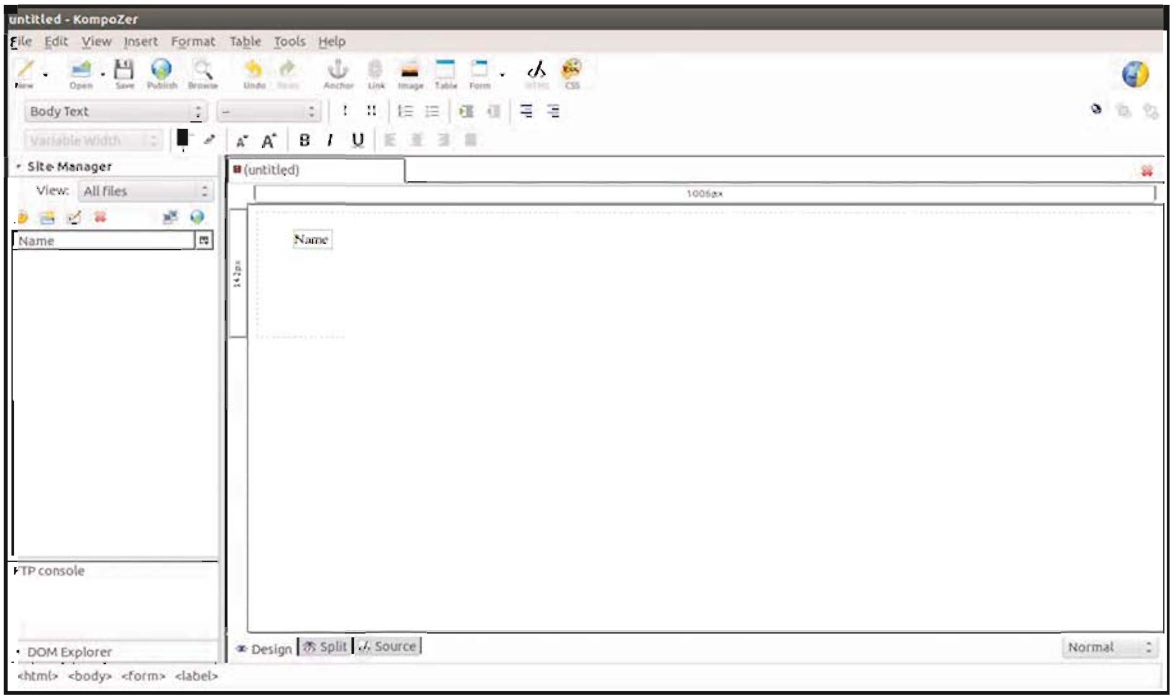
આકૃતિ 1.5 : Form Properties ડાયલોગબૉક્સમાં ઉમેરેલી વિગતો

- આકૃતિ 1.6માં દર્શાવ્યા મુજબ ફૉર્મ શીર્ષકરહિત (untitled) પાનામાં આછા ભૂરા રંગની સીમારેખા સાથે ઉમેરવામાં આવશે. સામાન્ય દેખાવમાં ફૉર્મની આસપાસ ટપકાંવાળું ભૂરું ચોકઠું દર્શાવવામાં આવશે. ટેક્સ્ટબૉક્સ, રેડિયો-બટન, ચેકબૉક્સ અને ડ્રૉપડાઉન બૉક્સ જેવા ફૉર્મના તમામ ઘટકો આ બૉક્સમાં મૂકવામાં આવશે. એકથી વધુ વાર એન્ટર કી દબાવી ફૉર્મમાં કાર્ય કરવા માટેની થોડી જગ્યા બનાવો.



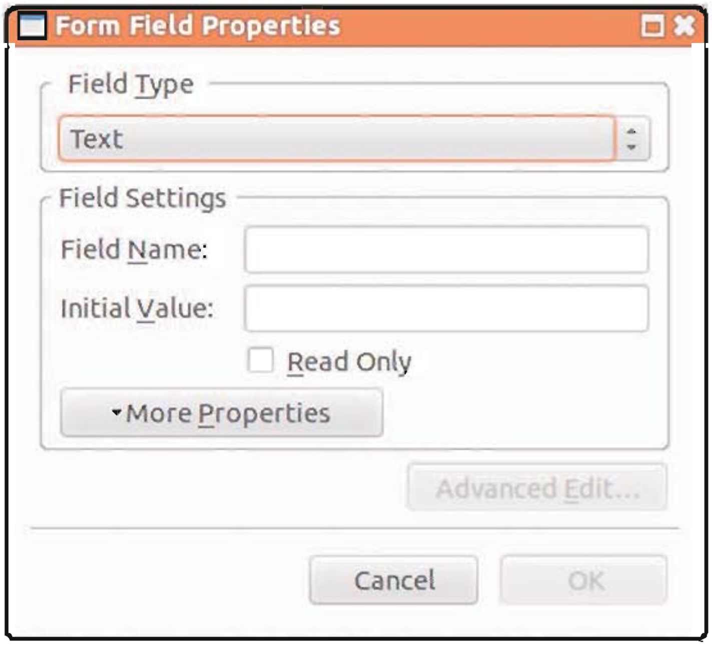
આકૃતિ 1.6 : ફૉર્મની આછા ભૂરા રંગની સીમારેખા

- સૌપ્રથમ આપણે name ક્ષેત્ર માટે લેબલ ઉમેરીશું. **Insert → Form → Define label** પસંદ કરો. ફૉર્મમાં જે સ્થાને લેબલ ઉમેરવું હોય, ત્યાં કર્સર ગોઠવો. આકૃતિ 1.7માં દર્શાવ્યા મુજબ લેબલમાં "Name" લખાણ ટાઇપ કરો. લેબલફિલ્ડમાંથી બહાર આવવા ફિલ્ડ સિવાયના અન્ય કોઈ પણ સ્થાને ક્લિક કરો.



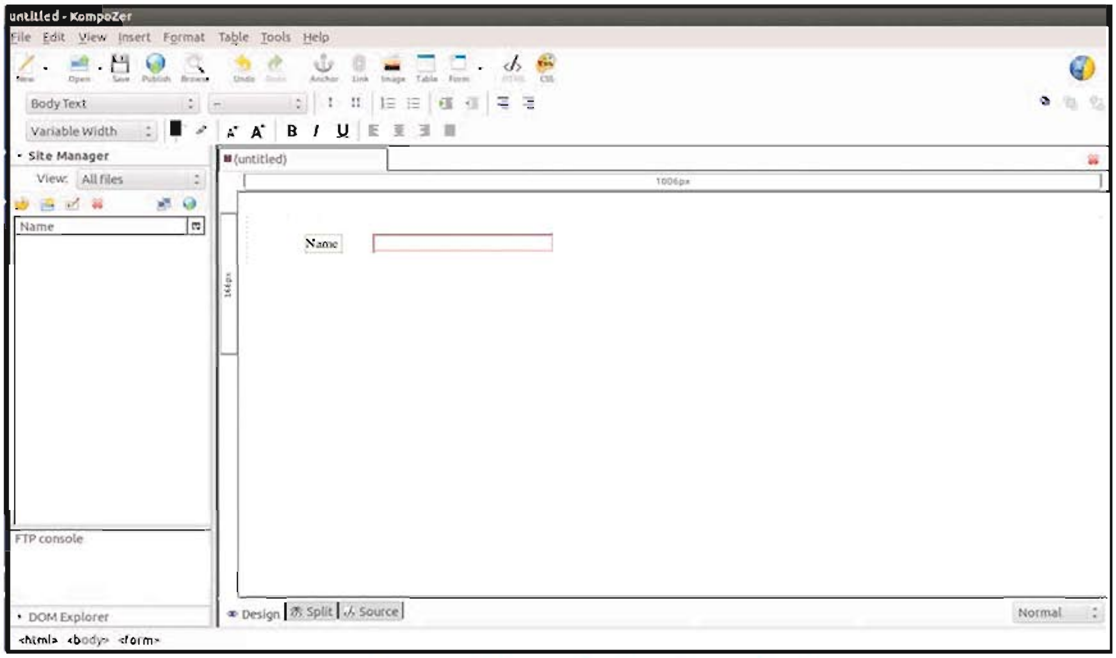
આકૃતિ 1.7 : ફૉર્મમાં ઉમેરવામાં આવેલ લેબલફિલ્ડ

- ફોર્મમાં ઇનપુટ ટેક્સ્ટફિલ્ડ ઉમેરવા માટે **Insert → Form → Form Field** વિકલ્પ પસંદ કરો. આકૃતિ 1.8 Form Field Properties ડાયલોગબૉક્સ દર્શાવે છે. ડ્રૉપડાઉન મેનુ વિવિધ ઇનપુટ ફિલ્ડના પ્રકાર દર્શાવે છે, જેની ચર્ચા આપણે અગાઉ કરી ચૂક્યા છીએ. More Properties બટન પર ક્લિક કરો, તે ફિલ્ડના કદ અને મહત્તમ લંબાઈ જેવા કેટલાક અન્ય ગુણધર્મો દર્શાવે છે.



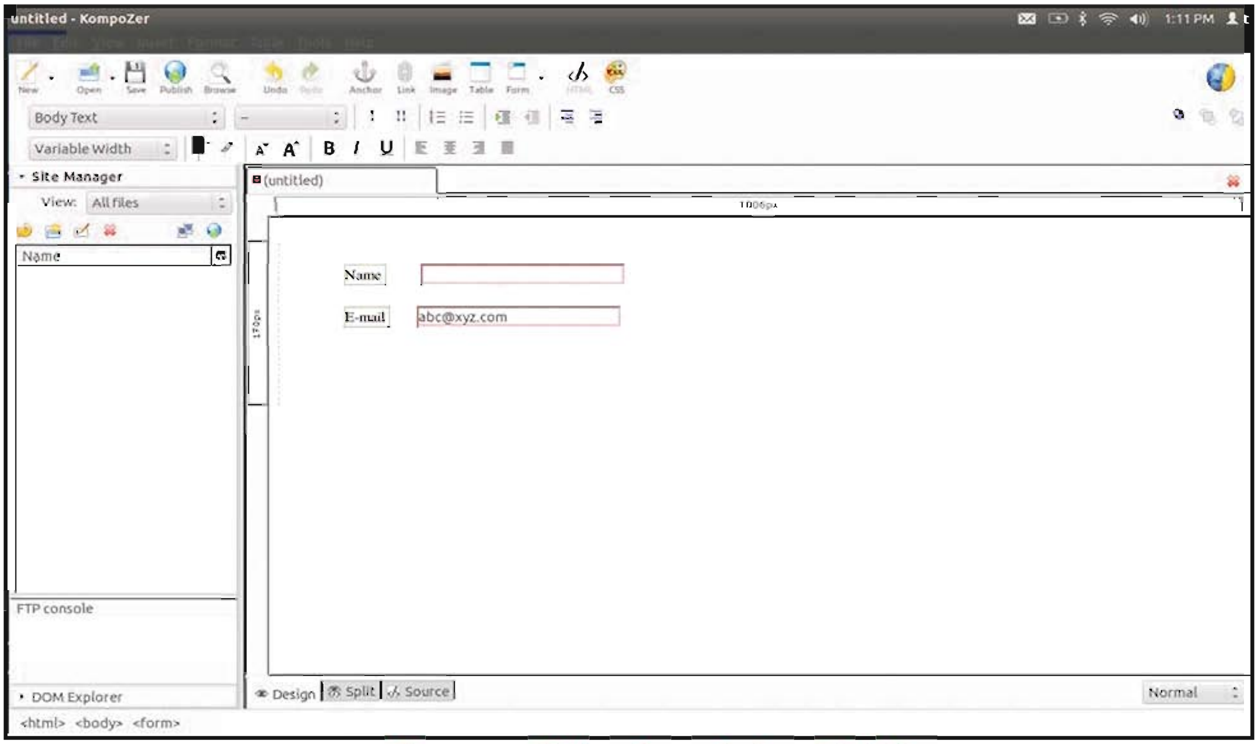
આકૃતિ 1.8 : ફોર્મમાં ફિલ્ડ ઉમેરવું

- ડ્રૉપડાઉન મેનુમાંથી Text વિકલ્પ પસંદ કરો. Field Settings વિભાગમાં આવેલ Field Name ટેક્સ્ટબૉક્સમાં ફિલ્ડનું નામ ઉમેરો. અહીં આપણે ફિલ્ડનાં નામ તરીકે name ઉમેર્યું છે. જરૂરી વિગતો ઉમેરતાં પહેલાં ફિલ્ડમાં અન્ય કોઈ વિગત દર્શાવવા માટે Initial Value ટેક્સ્ટબૉક્સમાં થોડું લખાણ ઉમેરો. અહીં આપણે આ બૉક્સ ખાલી રાખ્યું છે. OK બટન પર ક્લિક કરો. ટેક્સ્ટ ઇનપુટ ફિલ્ડ ઉમેર્યા પછીનું ફૉર્મ આકૃતિ 1.9માં દર્શાવ્યું છે.



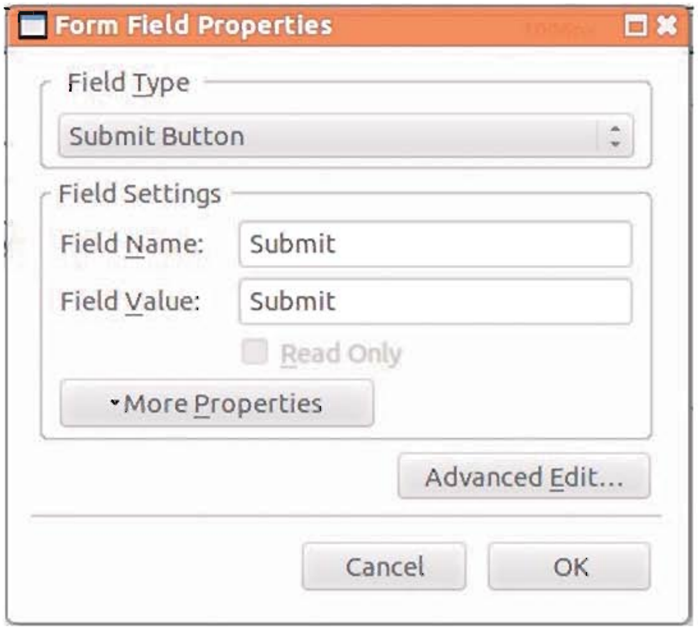
આકૃતિ 1.9 : લેબલ અને ટેક્સ્ટફિલ્ડ ઉમેરવાં

* હવે, આવી જ રીતે Name લેબલ ફિલ્ડની નીચે અન્ય લેબલ ફિલ્ડ E-mail ઉમેરો. Name ફિલ્ડ ઉમેર્યું તે જ રીતે E-mail માટે પણ ઇનપુટ ટેક્સ્ટફિલ્ડ ઉમેરો. અહીં Initial Value ટેક્સ્ટબૉક્સમાં abc@xyz.com લખાણ ઉમેરો. તે ઉપયોગકર્તાને ઇ-મેઇલ સરનામાંનું સ્વરૂપ સમજવામાં મદદરૂપ બનશે. બંને ટેક્સ્ટફિલ્ડ ઉમેર્યા પછીનું ફૉર્મ આકૃતિ 1.10માં દર્શાવ્યું છે.



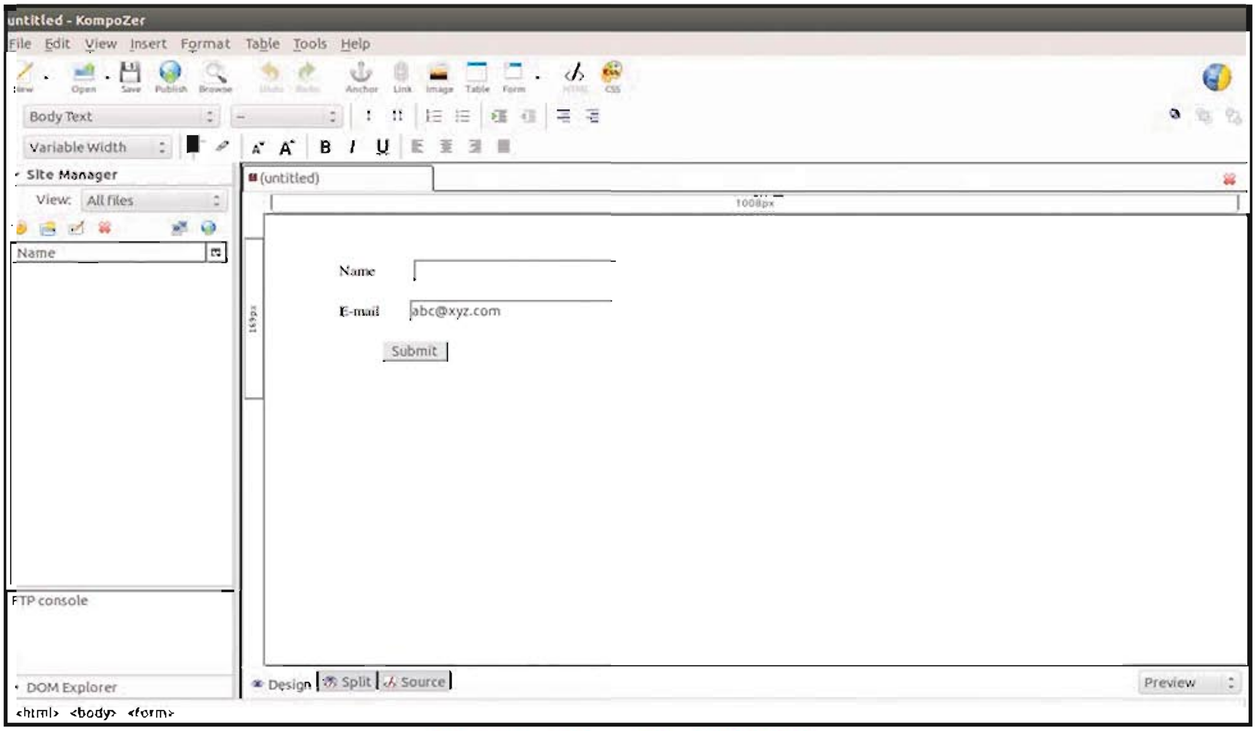
આકૃતિ 1.10 : બંને ટેક્સ્ટફિલ્ડ ઉમેર્યા પછીનું ફૉર્મ

* અંતમાં આપણે ફૉર્મમાં સબમિટ બટન ઉમેરીશું. **Insert → Form → Form Field** પર ક્લિક કરો. ડ્રોપડાઉન મેનુમાંથી Submit Button વિકલ્પ પસંદ કરો. Field Name અને Field Value બંને ટેક્સ્ટબૉક્સમાં Submit લખાણ ટાઇપ કરો અને OK બટન પર ક્લિક કરો. સબમિટ બટન માટેના Form Field Properties ડાયલૉગબૉક્સનો દેખાવ આકૃતિ 1.11માં દર્શાવ્યો છે.



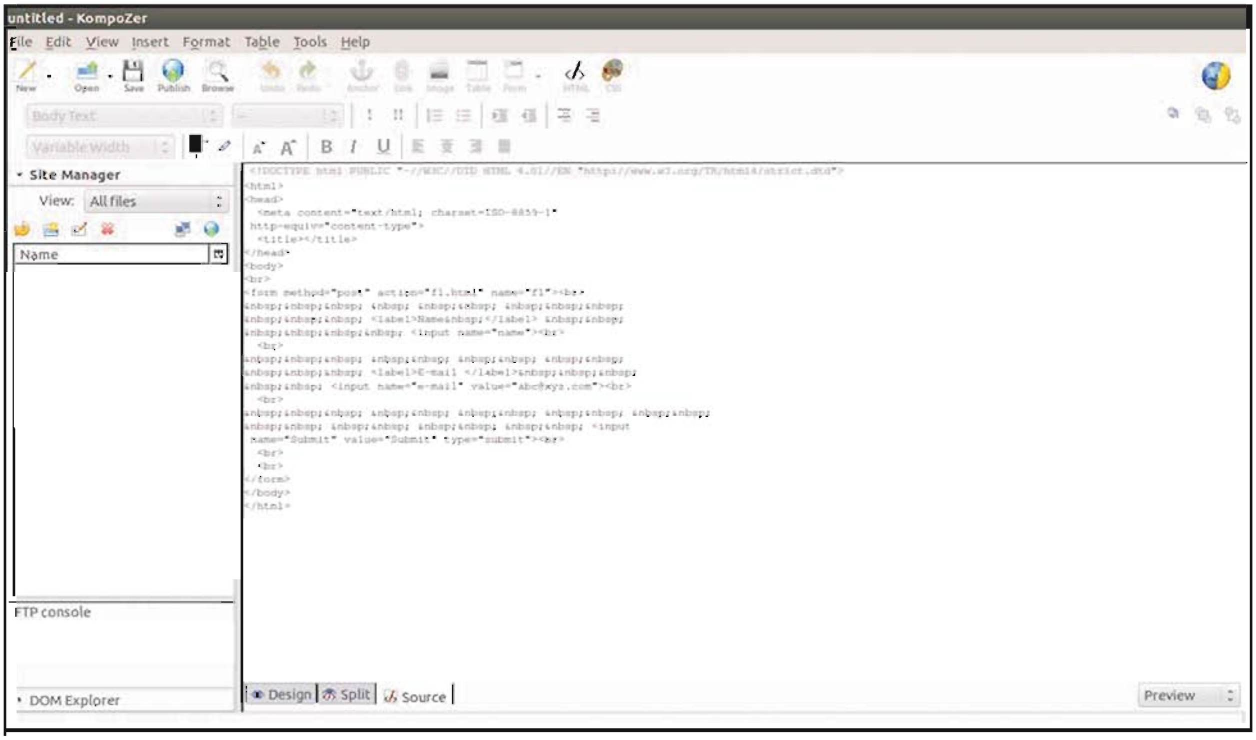
આકૃતિ 1.11 : ઇનપુટ ફિલ્ડ સબમિટ બટન

- હાલમાં ફૉર્મ સામાન્ય દેખાવ (Normal View) સ્વરૂપે છે. ફૉર્મના પૂર્વદર્શન (preview) માટે એડિટમોડ ટૂલબાર પર આવેલાં **Preview** ટૂલબટન પર ક્લિક કરો. આકૃતિ 1.12માં ફૉર્મ પ્રિવ્યૂ મોડમાં દર્શાવ્યું છે.



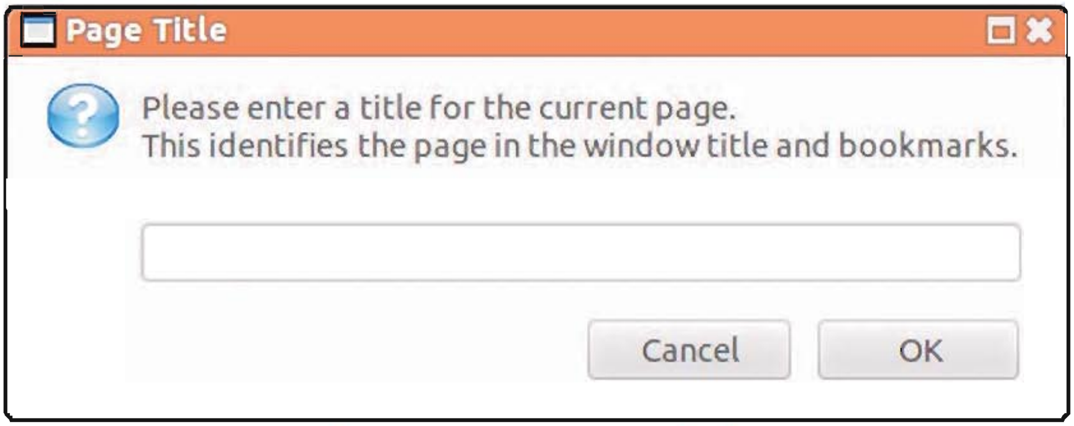
આકૃતિ 1.12 : પ્રિવ્યૂ મોડમાં ફૉર્મનો દેખાવ

આમ, કમ્પોઝરનો ઉપયોગ કરી આપણે પ્રથમ ફૉર્મની રચના કરી. તમે જોઈ શકો છો કે ટૂંકા સમયમાં ફૉર્મની રચના કરવા માટે કમ્પોઝર કેવી રીતે મદદરૂપ બને છે તથા તે લાંબા સમય સુધી સૉર્સકોડ લખવાના કંટાળાજનક કાર્યમાંથી પણ આપણને મુક્ત કરે છે. Source વિભાગ પર ક્લિક આપી હાલમાં બનાવેલ ફૉર્મના સૉર્સકોડને પણ જોઈ શકાય છે. જુઓ આકૃતિ 1.13.



આકૃતિ 1.13 : ફૉર્મના સૉર્સકોડનો દેખાવ

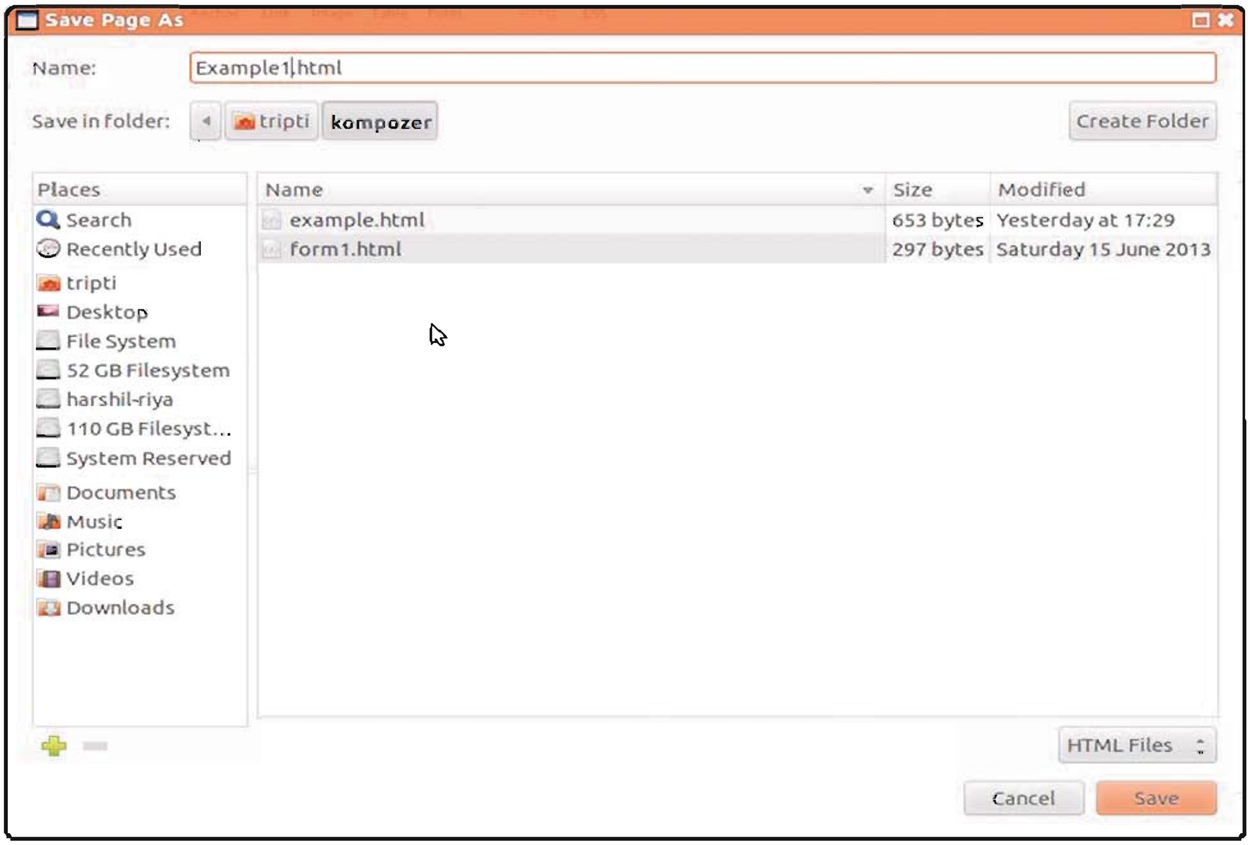
- તૈયાર થયેલ ફાઇલનો હવે સંગ્રહ કરીએ. **File → Save** વિકલ્પ પર હવે ક્લિક કરો. વૈકલ્પિક રીતે કમ્પોઝિશન ટૂલબાર પર આવેલા બટન પર પણ ક્લિક કરી શકાય. આમ કરવાથી આકૃતિ 1.14માં દર્શાવેલ Page Title ડાયલોગબૉક્સ ખૂલશે. અહીં વેબપેજને યોગ્ય શીર્ષક આપી શકાશે. આપણે તેને example1 નામ આપ્યું છે. હવે, OK બટન પર ક્લિક કરો.



આકૃતિ 1.14 : Page Title ડાયલોગબૉક્સ

વેબપેજને બ્રાઉઝરમાં દર્શાવવામાં આવે ત્યારે બ્રાઉઝર વિન્ડોના ટાઇટલબારમાં પાનાંનું શીર્ષક જોઈ શકાય છે. જો આપણે એકથી વધુ વેબપેજની રચના કરી હોય, તો તેવા કિસ્સામાં પાનાંનાં શીર્ષક તરીકે વેબસાઇટનું નામ આપવું જોઈએ. આપણે ફૉર્મ સાથેના એક જ વેબપેજની રચના કરી હોવાથી આ ઉદાહરણમાં શીર્ષક પાનાંને example1 નામ આપ્યું છે.

- OK બટન પર ક્લિક કર્યા બાદ આકૃતિ 1.15માં દર્શાવેલ Save Page As ડાયલોગબૉક્સ ખૂલશે, જે ફાઇલનું નામ અને ફાઇલનો સંગ્રહ કરવા માટેના સ્થાન વિશેની માહિતી પૂછશે.



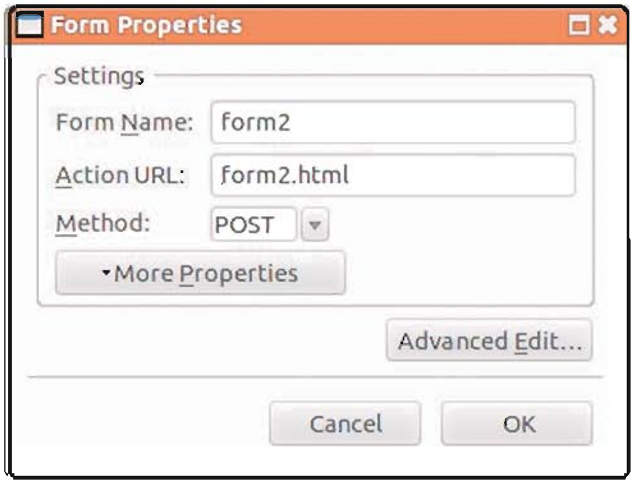
આકૃતિ 1.15 : Save Page As ડાયલોગબૉક્સ

ફાઇલને ***html*** કે ***htm*** અનુલંબન (extension) આપીને સંગ્રહ કરવાનું યાદ રાખો. Save બટન પર ક્લિક કરો. આમ કરવાથી ફરી મુખ્ય વિન્ડો પર જઈ શકાશે.

નોંધ : વેબસાઇટની રચના કરતી વખતે જો કોઈ પાનું હોમપેજ હોય કે જે વેબસાઇટનું URL ટાઇપ કરવાથી ખૂલે તેમ રાખવાનું હોય, તો તેનો index.html નામ સાથે સંગ્રહ કરો.

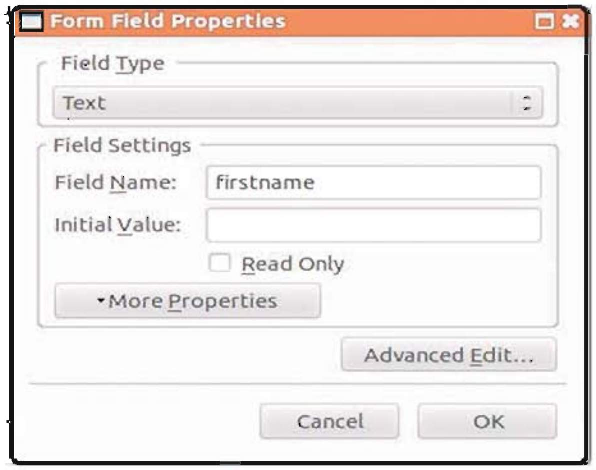
કમ્પોઝરમાં સરળ ફૉર્મની રચના કરતાં, તેને ખોલતાં તથા સંગ્રહ કરતાં શીખ્યા પછી હવે આપણે HTML ટૅગનો ઉપયોગ કરી પહેલાં બનાવ્યું હતું તેવું જ એક નોંધણીફૉર્મ ફરી બનાવીએ. નોંધણી માટેના ફૉર્મની રચના કરવા માટે નીચેનાં પગલાંને અનુસરો :

- નવી ફાઇલની રચના કરો.
- મેનુબારમાંથી **Insert → Form → Define Form** વિકલ્પ પસંદ કરો. Form Properties ડાયલોગબૉક્સમાં આકૃતિ 1.16માં દર્શાવ્યા મુજબની વિગતો ઉમેરો.



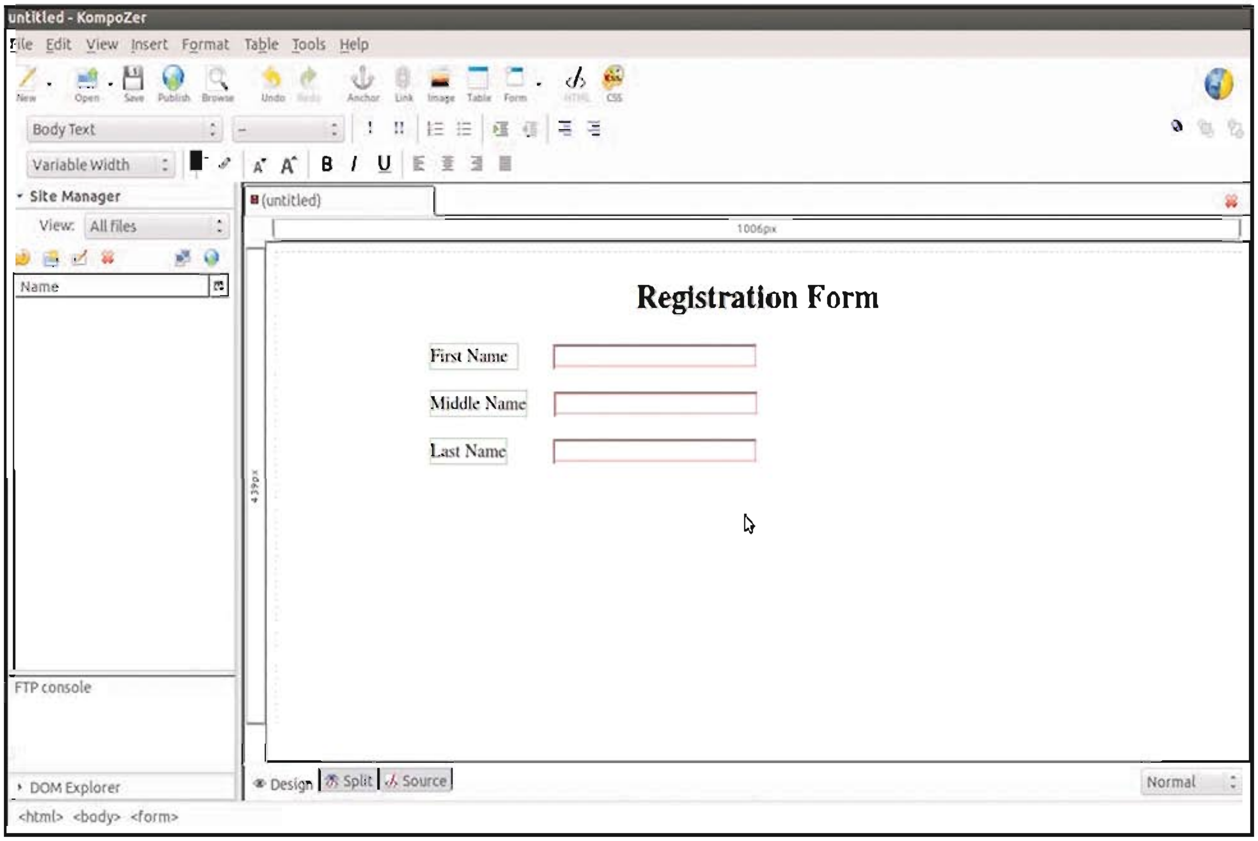
આકૃતિ 1.16 : Form Properties ડાયલોગબૉક્સ

- OK બટન પર ક્લિક કરો. આછા ભૂરા રંગની સીમારેખા ધરાવતું ફૉર્મ દર્શાવવામાં આવશે. ફૉર્મમાં જગ્યા ઉમેરવા માટે એન્ટર કી દબાવો.
- ફૉર્મને શીર્ષક આપવા માટે ફૉર્મેટ ટૂલબાર-1માંથી Heading-1 પસંદ કરો. ફૉર્મેટ ટૂલબાર-2માંથી Centre Align આઇકોન પસંદ કરો. "Registration Form" લખાણ ઉમેરો.
- લેબલ ઉમેરવા માટે **Form → Define Label** વિકલ્પ પસંદ કરો. Field Name ટૅક્સ્ટબૉક્સમાં "First Name" ટાઇપ કરો. ત્યાર બાદ "First name" લેબલ માટેનું ઇનપુટ ફિલ્ડ ઉમેરવા માટે **Form → Form Field** વિકલ્પ પર ક્લિક કરો. Field Type મેનુમાંથી ટૅક્સ્ટ વિકલ્પ પસંદ કરો. આકૃતિ 1.17 Form Field Properties ડાયલોગબૉક્સ દર્શાવે છે. આપણે ફિલ્ડનાં નામ તરીકે "firstname" લખાણનો ઉપયોગ કર્યો છે, તે નોંધો. OK બટન પર ક્લિક કરો.



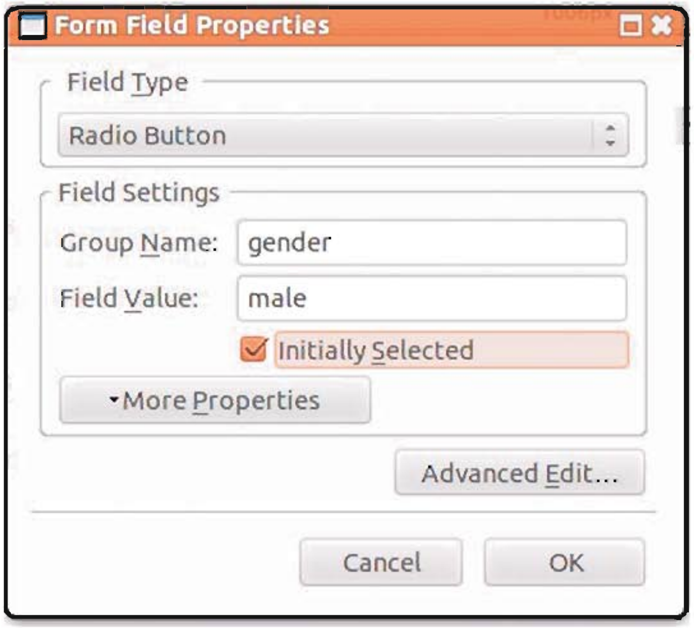
આકૃતિ 1.17 : First name માટે Form Field Properties ડાયલોગબૉક્સ

- આવી જ રીતે, ફૉર્મમાં "Middle Name" અને "Last Name" લેબલ ઉમેરો. ફિલ્ડ ઉમેર્યા પછી ફૉર્મનો દેખાવ આકૃતિ 1.18માં દર્શાવ્યા મુજબ દેખાશે.



આકૃતિ 1.18 : ફિલ્ડ ઉમેર્યા પછી ફૉર્મનો દેખાવ

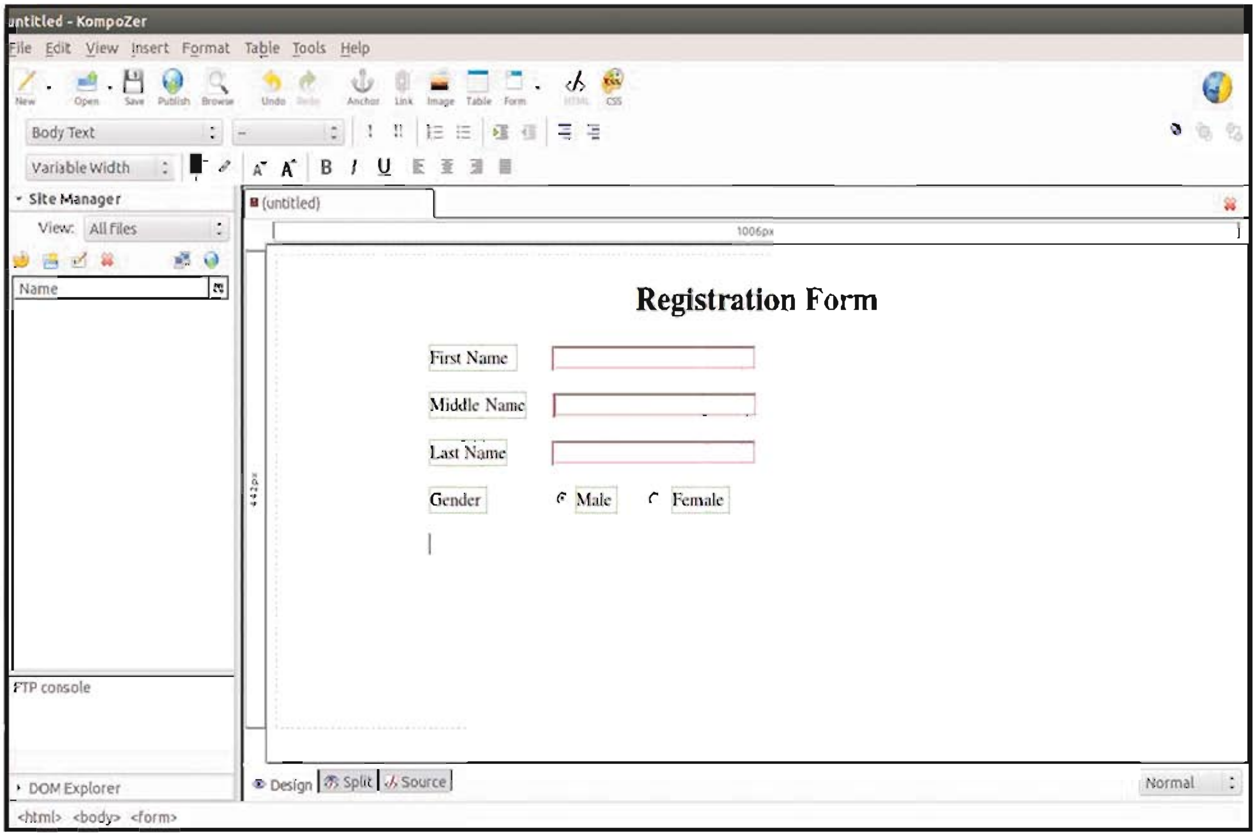
- હવે, Gender ફિલ્ડ માટે રેડિયો-બટન ઉમેરવાની જરૂર છે. પ્રથમ, "Gender" નામના લેબલની રચના કરો.
- રેડિયો-બટન ઉમેરવા માટે **Form → Form Field** વિકલ્પ પસંદ કરો અને આકૃતિ 1.19માં દર્શાવ્યા મુજબ ડ્રૉપડાઉન મેનુમાંથી Field Typeના વિકલ્પ તરીકે Radio Button પસંદ કરો.



આકૃતિ 1.19 : રેડિયો-બટન માટેની Form Field Properties

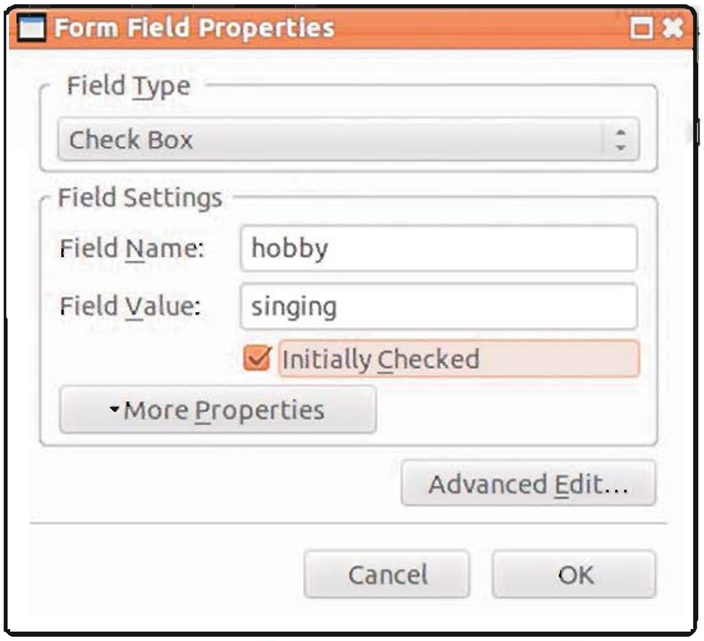
Group-Name બૉક્સમાં નામ ટાઈપ કરો. (નામમાં જગ્યા (space) સમાવી શકાતી નથી તેની નોંધ કરો.) અહીં આપણે Group Name તરીકે "gender" લખાણ ઉમેર્યું છે. આ જ રીતે Field Value ટેક્સ્ટબૉક્સમાં "male" ટાઈપ કરો. જ્યારે ફૉર્મ ખૂલે, ત્યારે જો આપણે male વિકલ્પને પૂર્વનિર્ધારિત રીતે પસંદ કરેલો રાખવા ઇચ્છતા હોઈએ, તો "Initially Selected" લખાણ આગળ આવેલું ચેકબૉક્સ પસંદ કરી OK બટન પર ક્લિક કરો.

- આ રેડિયો-બટનની પાસે "Male" શીર્ષક સાથે એક લેબલ ઉમેરો.
- આ જ રીતે, "Female" નામના અન્ય રેડિયો-બટનની રચના કરો. યાદ રાખો કે, જ્યારે આપણે રેડિયો-બટનની રચના જૂથમાં કરતા હોઈએ, ત્યારે તમામ શક્ય જવાબો માટે જૂથનું નામ (Group Name) એકસમાન હોવું જોઈએ. માટે Group Name તરીકે "gender" ઉમેરો. Field Value ટેક્સ્ટબૉક્સમાં "female" ટાઈપ કરો. OK બટન પર ક્લિક કરો.
- આ રેડિયો-બટનની પાસે "Female" લેબલ ઉમેરો. રેડિયો-બટન અને તેના લેબલ ઉમેર્યા પછી ફૉર્મ આકૃતિ 1.20માં દર્શાવ્યા મુજબ દેખાશે.



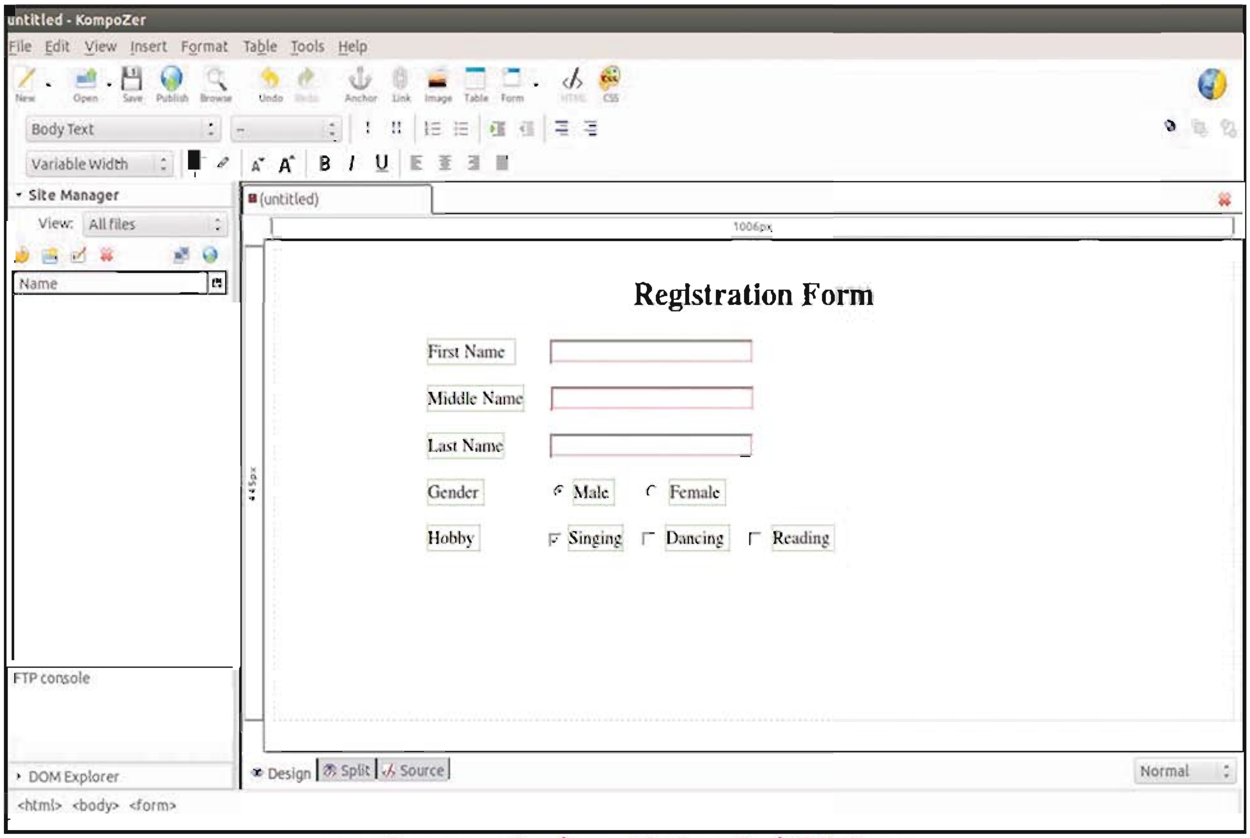
આકૃતિ 1.20 : રેડિયો-બટન ઉમેર્યા પછી ફૉર્મનો દેખાવ

- હવે આપણને "Hobby" ફિલ્ડની જરૂર છે. વ્યક્તિને એકથી વધુ શોખ હોઈ શકે છે, માટે Hobby ફિલ્ડ માટે એકથી વધુ વિકલ્પોની પસંદગી શક્ય હોવી જોઈએ. આમ, Hobby ફિલ્ડ માટે આપણને ચેકબૉક્સ બનાવવાની જરૂર છે. Hobby માટે લેબલની રચના કરો.
- હવે, **Form → Form Field** વિકલ્પ પસંદ કરી ડ્રૉપડાઉન મેનુમાંથી ફિલ્ડના પ્રકાર તરીકે CheckBox પસંદ કરો. આકૃતિ 1.21માં દર્શાવ્યા પ્રમાણે, Field Name બૉક્સમાં નામ અને Field Value બૉક્સમાં કિંમત ઉમેરો. ફૉર્મ ખૂલે ત્યારે આ વિકલ્પ પૂર્વનિર્ધારિત રીતે પસંદ થયેલો રાખવા માટે "Initially Selected" લખાણ આગળ આવેલા ચેકબૉક્સને પસંદ કરો. OK બટન પર ક્લિક કરો.



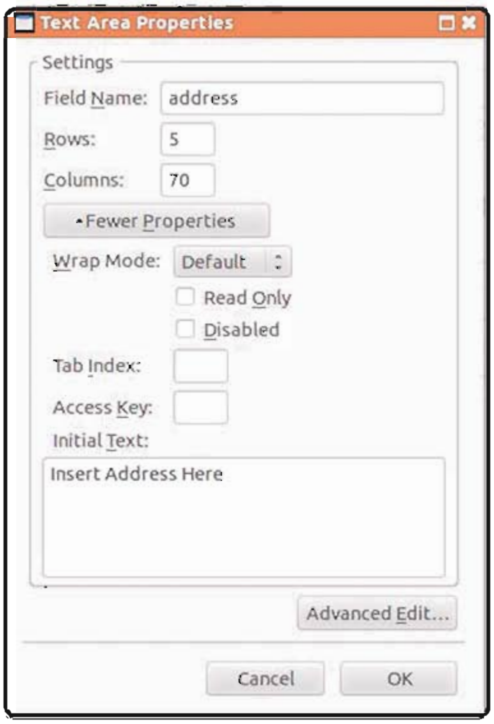
આકૃતિ 1.21 : ચેકબૉક્સ પ્રકારના ફિલ્ડની ગોઠવણ

- આ ચેકબૉક્સની બાજુમાં "Singing" લેબલ ઉમેરો.
- આ જ રીતે, Field Value તરીકે "dancing" અને "reading" ઉમેરી અન્ય બે ચેકબૉક્સની રચના કરો. ચેકબૉક્સના તમામ વિકલ્પોનાં Field Name સમાન હોય તે યાદ રાખો.
- ચેકબૉક્સની બાજુમાં "Dancing" અને "Reading" લેબલ ઉમેરો. ચેકબૉક્સ અને તેનાં લેબલ ઉમેર્યા પછી ફૉર્મ આકૃતિ 1.22માં દર્શાવ્યા મુજબ દેખાશે.



આકૃતિ 1.22 : ચેકબૉક્સ ઉમેર્યા પછી ફૉર્મનો દેખાવ

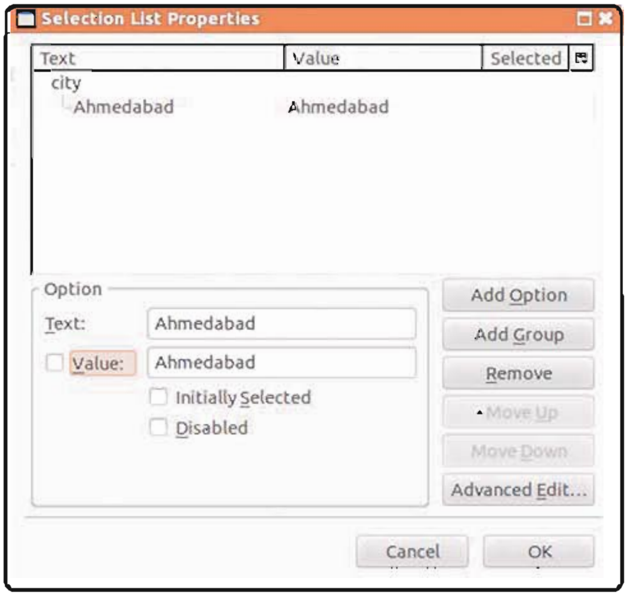
- ત્યાર બાદ આપણને "Address" ફિલ્ડની જરૂર છે. ઉપયોગકર્તા સરનામાંનાં ફિલ્ડમાં વધુ લીટીનું લખાણ ઉમેરી શકે છે. માટે તેના ફિલ્ડનો પ્રકાર ટેક્સ્ટ-એરિયા રાખીશું. પ્રથમ "Address" નામના લેબલની રચના કરો. હવે, **Form → Text Area** વિકલ્પ પસંદ કરો. આકૃતિ 1.23માં દર્શાવેલ Text Area Properties ડાયલોગબૉક્સ ખૂલશે.



આકૃતિ 1.23 : Text Area Properties ડાયલોગબૉક્સ

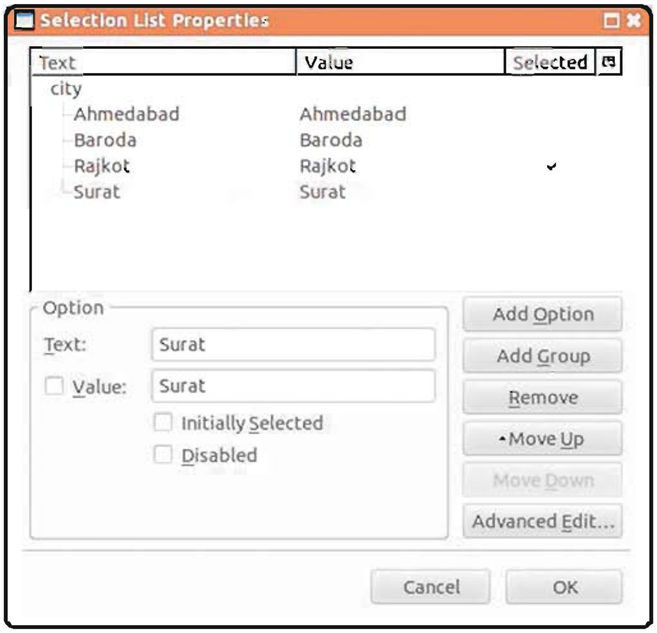
Field Name બૉક્સમાં નામ ઉમેરો. ટેક્સ્ટ-એરિયા માટે જરૂરી હરોળ અને સ્તંભની સંખ્યા ઉમેરો. અહીં આપણે હરોળની સંખ્યા 5 અને સ્તંભની સંખ્યા 70 રાખી છે. ફૉર્મ ખૂલે ત્યારે ફિલ્ડમાં પૂર્વનિર્ધારિત રીતે દર્શાવવામાં આવનાર યોગ્ય લખાણને Initial Text ફિલ્ડમાં ઉમેરો. OK બટન પર ક્લિક કરો.

- ત્યાર બાદ આપણે "City" ફિલ્ડ ઉમેરીશું. શહેર માટેનાં ફિલ્ડનું લેબલ ઉમેરો. ઉપયોગકર્તાને ડ્રૉપડાઉન મેનુમાંથી શહેરની પસંદગી અંગે પૂછવામાં આવશે. તેથી શહેરના ફિલ્ડ માટે પસંદગીયાદી (Selection List)નો ઉપયોગ કરવાની જરૂર છે. **Form → Selection List** વિકલ્પ પસંદ કરો. આકૃતિ 1.24માં દર્શાવ્યા મુજબ Selection List Properties ડાયલોગબૉક્સ ખૂલશે.



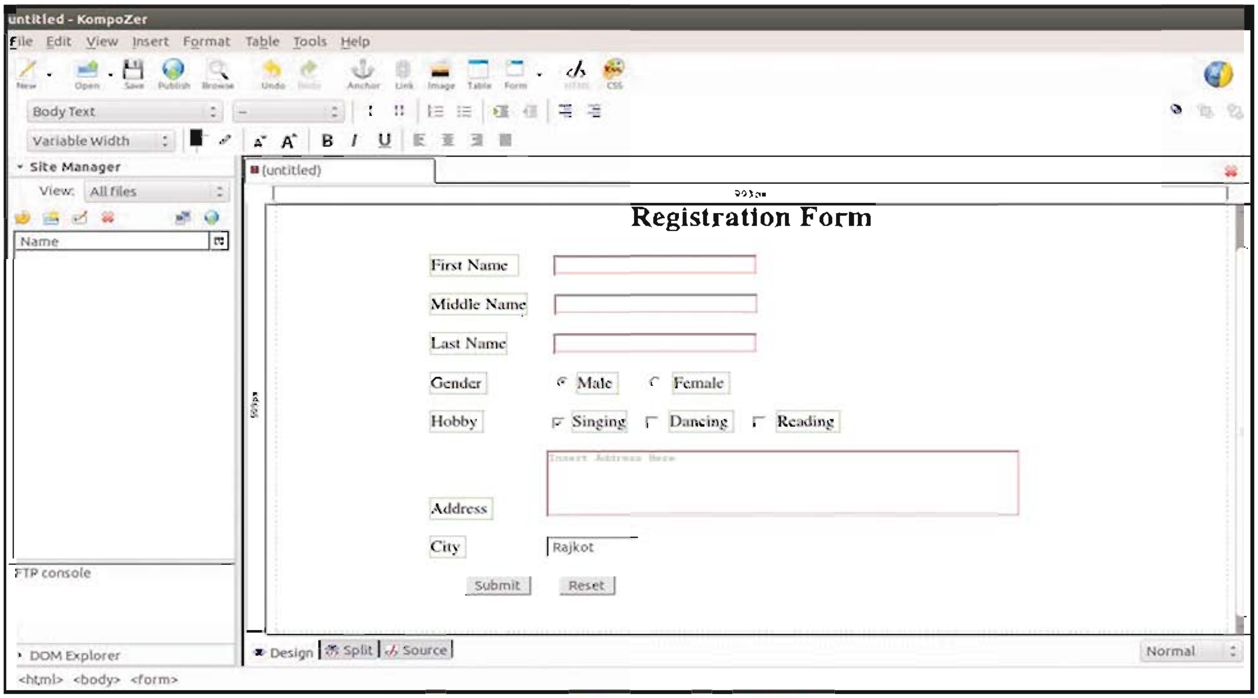
આકૃતિ 1.24 : Selection List Properties ડાયલોગબૉક્સ

List Name બૉક્સમાં "city" ટાઇપ કરી, Add Option બટન પર ક્લિક કરો. ત્યાર પછી, Textfieldમાં "Ahmedabad" ટાઇપ કરો. Add Option વિકલ્પ પર ફરી ક્લિક કરી શહેરનું નામ "Baroda" ઉમેરો. આ જ પ્રમાણે, "Rajkot" અને "Surat" શહેર પણ ઉમેરો. રાજકોટ શહેર ઉમેરતી વખતે "Initially Selected" વિકલ્પ પસંદ કરવાનું યાદ રાખો. OK બટન પર ક્લિક કરો. શહેરનું નામ ઉમેર્યા બાદ Selection List Properties ડાયલૉગબૉક્સનો દેખાવ આકૃતિ 1.25માં દર્શાવ્યો છે.



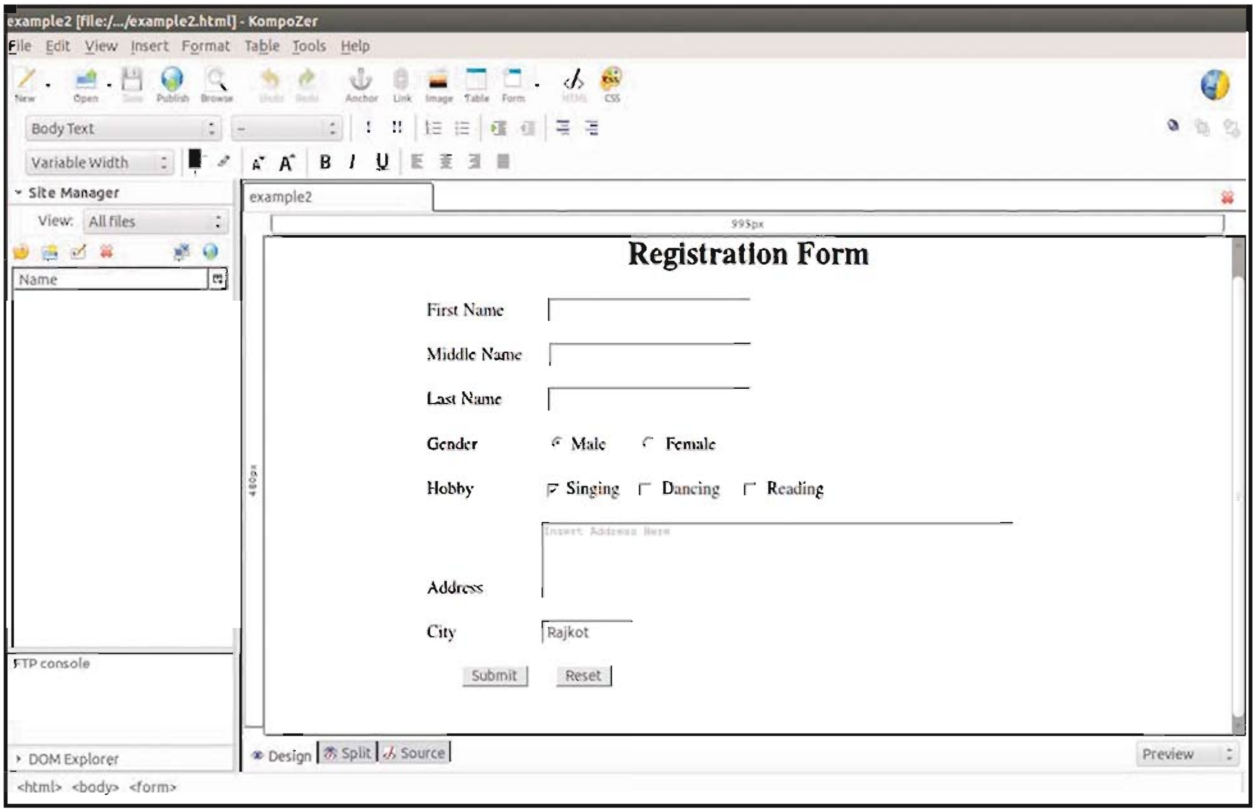
આકૃતિ 1.25 : વિવિધ શહેરો માટેનું Selection List Properties ડાયલૉગબૉક્સ

- ત્યાર બાદ, આપણે "Submit" બટન ઉમેરીશું. Form → Form Field વિકલ્પ પસંદ કરો. ડ્રૉપડાઉન મેનુમાંથી Submit Button વિકલ્પ પસંદ કરો. Field Name અને Field Value ટેક્સ્ટબૉક્સમાં Submit લખાણ ઉમેરો. OK બટન પર ક્લિક કરો.
- આ જ રીતે, આપણે "Reset" બટન ઉમેરીશું. ડ્રૉપડાઉન મેનુમાંથી Reset Button વિકલ્પ પસંદ કરો. Field Name અને Field Value ટેક્સ્ટબૉક્સમાં Reset લખાણ ઉમેરો. OK બટન પર ક્લિક કરો. આકૃતિ 1.26માં ફૉર્મનો અંતિમ દેખાવ નોર્મલ વ્યૂ દ્વારા દર્શાવ્યો છે.



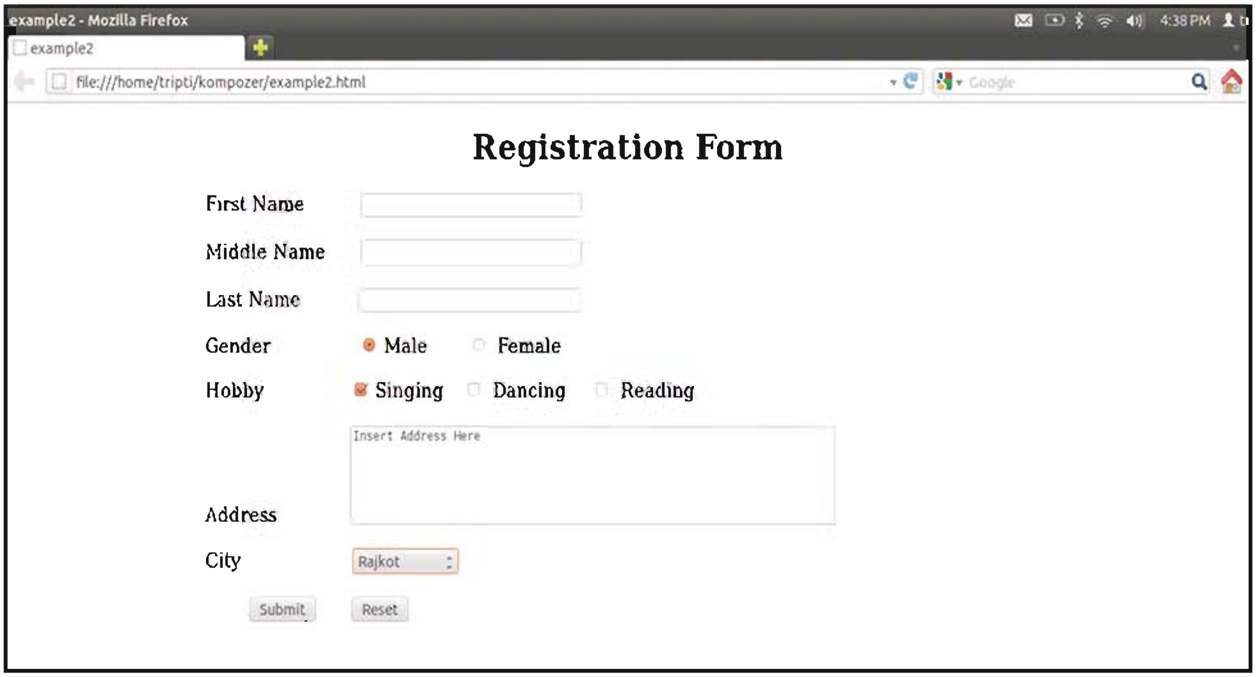
આકૃતિ 1.26 : નોર્મલ વ્યૂમાં ફૉર્મનો દેખાવ

ફાઇલનો "example2" નામથી સંગ્રહ કરો. ફૉર્મનો પ્રિવ્યૂ મોડ આકૃતિ 1.27માં દર્શાવ્યો છે.



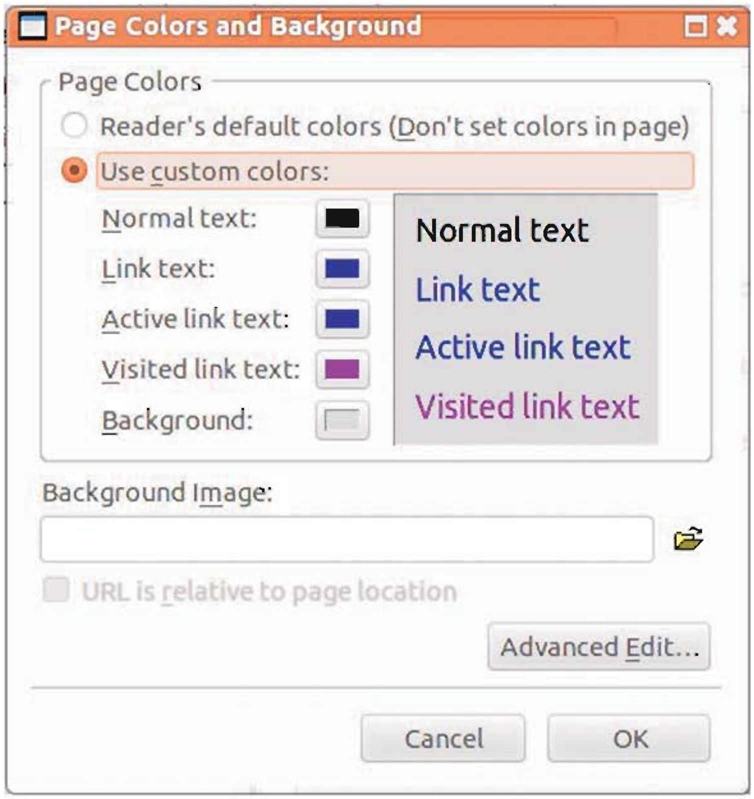
આકૃતિ 1.27 : પ્રિવ્યૂ મોડમાં ફૉર્મની રજૂઆત

આકૃતિ 1.28 નોંધણી માટેના ફૉર્મનો દેખાવ બ્રાઉઝરમાં દર્શાવે છે.



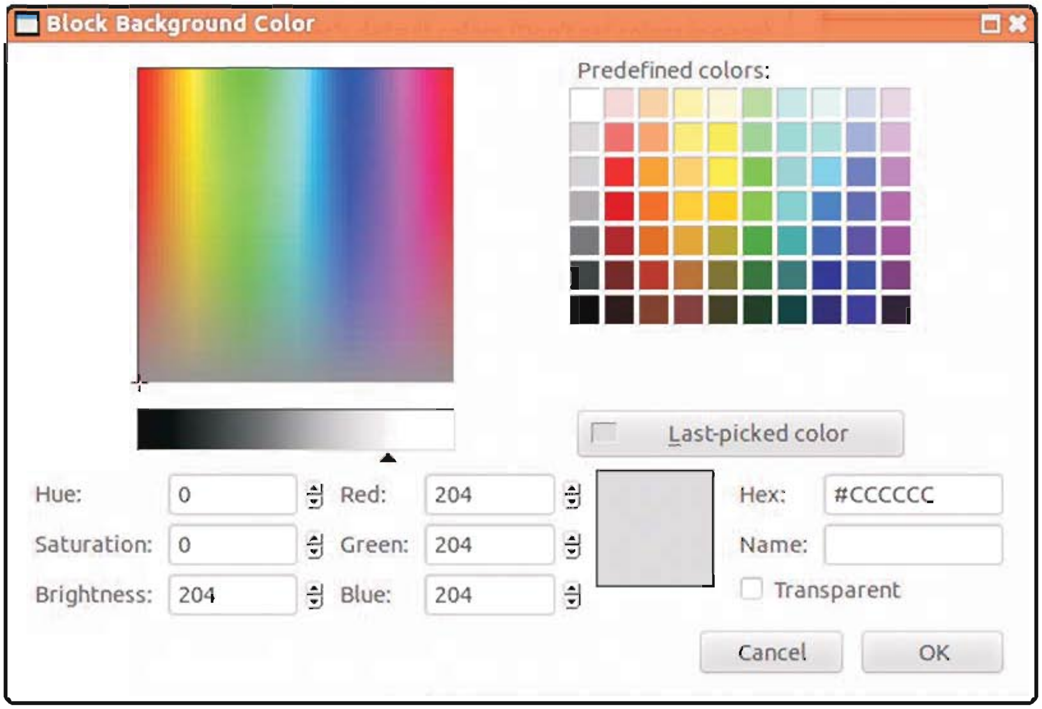
આકૃતિ 1.28 : બ્રાઉઝરમાં ફૉર્મનો દેખાવ

ફૉર્મના બૅકગ્રાઉન્ડનો રંગ સફેદ છે, તેની નોંધ લો. જો ફૉર્મને અન્ય રંગ બૅકગ્રાઉન્ડમાં આપવો હોય, તો **Format → Page Colors and Background** વિકલ્પ પસંદ કરો. આમ કરવાથી આકૃતિ 1.29માં દર્શાવ્યા મુજબનું ડાયલોગબૉક્સ ખૂલશે.



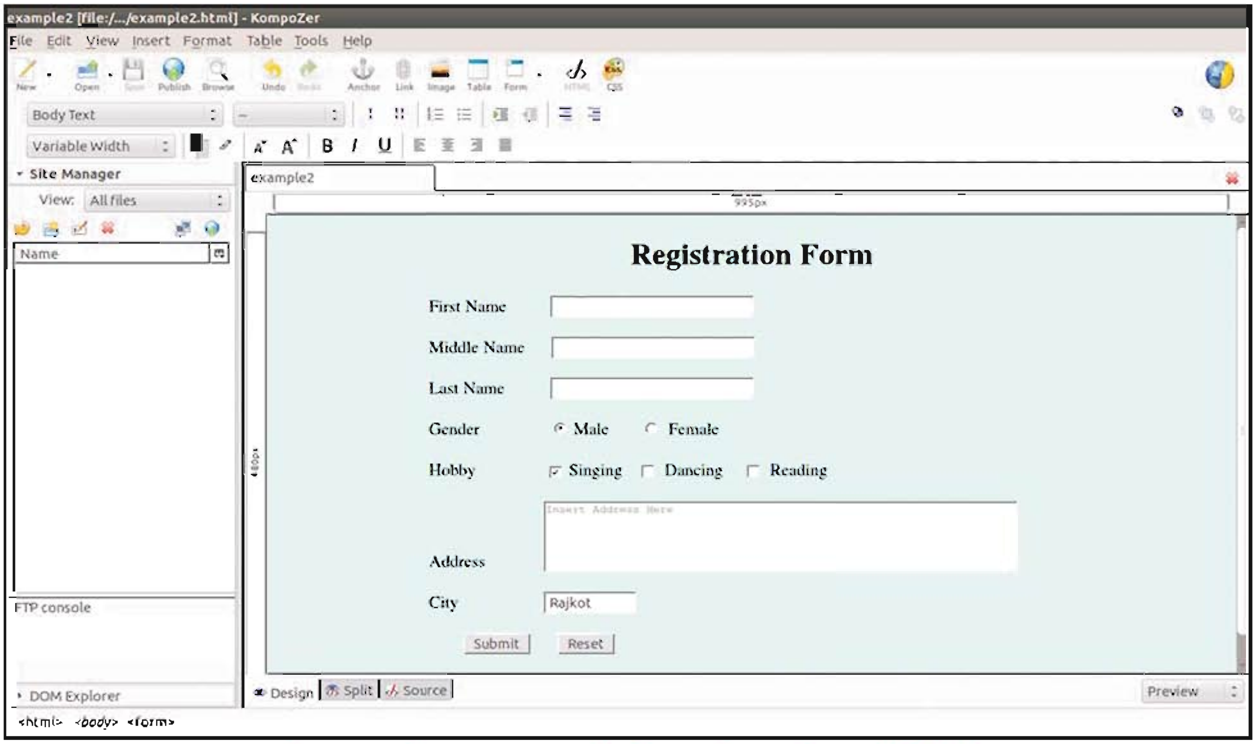
આકૃતિ 1.29 : Page Colors and Background ડાયલોગબૉક્સ

"Use Custom Colors" વિકલ્પ પસંદ કરો. Background વિકલ્પ પર ક્લિક કરી આકૃતિ 1.30માં દર્શાવ્યા મુજબ Block Background Color ડાયલોગબૉક્સમાંથી તમારી પસંદગીનો રંગ પસંદ કરો. OK બટન પર ક્લિક કરો, આથી ફરી આકૃતિ 1.29માં આપવામાં આવેલું ડાયલોગબૉક્સ રજૂ કરવામાં આવશે. ફરી OK બટન પર ક્લિક કરો.



આકૃતિ 1.30 : બૅકગ્રાઉન્ડ રંગની પસંદગી

રંગની પસંદગી કર્યા પછી ફૉર્મનો દેખાવ આકૃતિ 1.31માં દર્શાવ્યા મુજબ દેખાશે.



આકૃતિ 1.31 : બૅકગ્રાઉન્ડ રંગ ઉમેર્યા બાદ પ્રિવ્યૂ મોડમાં ફૉર્મનો દેખાવ

બ્રાઉઝરનો ઉપયોગ કરી ફૉર્મ જુઓ અને બૅકગ્રાઉન્ડ રંગમાં થયેલું પરિવર્તન ધ્યાનમાં લો.

સારાંશ

વેબ પર વિગતો સ્વીકારવા માટે ફૉર્મનો ઉપયોગ કરવામાં આવે છે. HTMLમાં ફૉર્મ એક સંગ્રાહક છે. ઉપયોગકર્તા પાસેથી જુદા-જુદા પ્રકારના નિવેશ એકત્રિત કરવા માટે ફૉર્મનો ઉપયોગ કરવામાં આવે છે. ઉપયોગકર્તા પોતાની અંગત માહિતી, કોઈ ઉત્પાદન અંગેનો પ્રતિસાદ, મોજણી કે પરિવહનની વિગતો, ક્રેડિટકાર્ડની વિગતો વગેરે ફૉર્મમાં ઉમેરે છે. કમ્પોઝર વેબવિકાસ માટેનું એક નિ:શુલ્ક અને ઑપનસૉર્સ IDE છે, જેના ઉપયોગથી વેબસાઇટની રચના કરી શકાય છે. તે WYSIWYG - "What You See Is What You Get" નામે ઓળખાતા એક સરળ ગ્રાફિકલ ઇન્ટરફેસ ધરાવતું વેબપેજ એડિટર પૂરું પાડે છે. કમ્પોઝરની મદદથી ફૉર્મની રચના સરળતાથી અને ઝડપથી કરી શકાય છે.

સ્વાધ્યાય

1. ફૉર્મ શું છે ? HTMLમાં ફૉર્મની રચના કરવા માટેના ઘટકોની યાદી બનાવો.
2. HTML ફૉર્મમાં નિવેશ માટેના ઘટકોનો ઉપયોગ જણાવો. ઇનપુટ ટૅગની વિવિધ લાક્ષણિક્તા વિશે લખો.
3. HTML ફૉર્મમાં ટૅક્સ્ટ-એરિયા ઘટકનો હેતુ જણાવો.
4. સિલેક્ટ અને ઑપ્શન વિકલ્પ વિશે જણાવો.
5. કમ્પોઝર વિન્ડોમાં દર્શાવાતા જુદા-જુદા ટૂલબારની યાદી બનાવો.
6. ફૉર્મની બે લાક્ષણિક્તાઓ કઈ છે ? સમજાવો.

7. આપેલ વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરો :

(1) ઉપયોગકર્તા પાસેથી જુદા-જુદા પ્રકારના નિવેશ એકત્રિત કરવા માટે નીચેનામાંથી ક્યા સંગ્રાહક એકમનો ઉપયોગ કરવામાં આવે છે ?

(a) Form (b) Webpage

(c) Text (d) Input

(2) HTML ફૉર્મની રચના કરવા માટે નીચેનામાંથી ક્યા ઘટકનો ઉપયોગ કરવામાં આવે છે ?

(a) Textarea (b) Form

(c) Select અને Option (d) Input

(3) ફૉર્મ ઘટકનો અમલ કરવા માટે નીચેનામાંથી ક્યા ટેગનો ઉપયોગ કરવામાં આવે છે ?

(a) <form>... </form> (b) <form>... <form>

(c) </form>... </form> (d) <frm>... </frm>

(4) ફૉર્મ સબમિટ કરતી વખતે ફૉર્મની વિગતો ક્યા સ્થાને મોકલવાની છે, તેની સ્પષ્ટતા માટે નીચેનામાંથી કઈ લાક્ષણિક્તાનો ઉપયોગ કરવામાં આવે છે ?

(a) method (b) action

(c) submit (d) input

(5) ફૉર્મની વિગતો મોકલતી વખતે HTTP પદ્ધતિ સ્પષ્ટ કરવા માટે નીચેનામાંથી ફૉર્મની કઈ લાક્ષણિક્તાનો ઉપયોગ કરવામાં આવે છે ?

(a) submit (b) action

(c) method (d) input

(6) method લાક્ષણિક્તા સાથે કઈ કિંમતોનો ઉપયોગ કરી શકાય છે ?

(a) GET અને POST (b) GET અને SET

(c) GET અને PUT (d) SET અને POST

(7) નીચેનામાંથી કઈ પદ્ધતિ એક સમયે માત્ર મર્યાદિત માહિતી મોકલવાની અનુમતિ પૂરી પાડે છે ?

(a) GET (b) POST

(c) SET (d) PUT

(8) નીચેનામાંથી કઈ પદ્ધતિ HTTP વ્યવહાર દ્વારા વિગતોને બ્લૉક સ્વરૂપે મોકલવાની સુવિધા પૂરી પાડે છે ?

(a) GET (b) SET

(c) PUT (d) POST

(9) નીચેનામાંથી કઈ લાક્ષણિક્તા દ્વારા ફૉર્મમાં નિર્માણ કરવાના ફિલ્ડના પ્રકારનો સ્પષ્ટ નિર્દેશ કરવામાં આવે છે ?

(a) Input (b) Type

(c) Name (d) Value

(10) નીચેનામાંથી ક્યો ઘટક એકથી વધુ લીટીના નિવેશ માટે અનુમતિ આપે છે ?

(a) Textarea (b) Input

(c) Select અને Option (d) Form

(11) નીચેનામાંથી ક્યા ઘટકનો ઉપયોગ ફૉર્મમાં ડ્રૉપડાઉન યાદી અથવા મેનુની રચના કરવા માટે જરૂરી છે ?

(a) Input (b) Textarea

(c) Select (d) Form

(12) નીચેનાંમાંથી ક્યું નિઃશુલ્ક, ઑપનસૉર્સ વેબવિકાસ માટેનું IDE છે ?

(a) HTML (b) KompoZer

(c) Scite (d) Base

(13) WYSIWYG એટલે શું ?

(a) When You See Is When You Get (b) What You See Is When You Get

(c) What You See Is What You Get (d) When You See Is What You Get

પ્રાયોગિક સ્વાધ્યાય

1. વિદ્યાર્થીઓની અંગત વિગતો મેળવવા માટેના ફૉર્મની રચના કરો.
2. તમારી શાળાના મુલાકાતીને પ્રતિસાદ (feedback) આપવા માટેના ફૉર્મની રચના કરો.
3. તમે તમારાં માતા-પિતા સાથે રજાઓમાં ફરવા ગયા હતા; પ્રવાસ-આયોજકે તમારી પાસે પ્રવાસને લગતી સમીક્ષા (review)માંગી છે. આ માટેના ફૉર્મની રચના કરો.

કેસ્કેડિંગ સ્ટાઇલશીટ અને જાવાસ્ક્રિપ્ટ 2

કેસ્કેડિંગ સ્ટાઇલશીટ (Cascading Style- Sheet) અથવા CSS વેબસાઇટમાં આવેલા દર્શનીય ઘટકોની શૈલી (Style) સ્પષ્ટ કરવા માટેની સુવિધા પૂરી પાડે છે. દસ્તાવેજની માહિતીને તેને દર્શાવવાની વિગતોથી અલગ રાખવા તે મદદરૂપ બને છે. દસ્તાવેજને કેવી રીતે દર્શાવવો તેની વિગતોને સ્ટાઇલ (Style) તરીકે ઓળખવામાં આવે છે. સ્ટાઇલ જે-તે ઘટકનો દેખાવ સ્ક્રીન પર નક્કી કરે છે. સ્ટાઇલને અન્ય માહિતીથી અલગ રાખવાથી નીચે જણાવેલ લાભ મેળવી શકાય :

- કોડિંગમાં નકલ (duplication) ટાળી શકાય.
- એક જ વિગત માટે જુદા-જુદા ઉદ્દેશોને અનુલક્ષીને જુદી-જુદી સ્ટાઇલનો ઉપયોગ કરી શકાય.
- કોડની સરળ જાળવણી કરી શકાય.

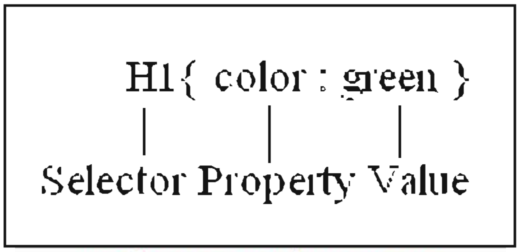
ઉદાહરણ તરીકે, HTMLમાં આપણે ઇચ્છતા હોઈએ કે <p> ટૅગથી ઉમેરેલા કેટલાક ફકરાનું લખાણ ઘાટા (bold) અક્ષરોમાં આવવું જોઈએ, તો તે માટે સૉર્સકોડમાં આવેલ તમામ ફકરાના <p> ટૅગ માટે અક્ષરો ઘાટા કરવા માટેનો <b> ટૅગ ઉમેરવો પડે. જો વેબસાઇટ વિશાળ હોય તો ટૅગનું આ પુનરાવર્તન કંટાળાજનક અને સમયનો વ્યય કરનારું બની રહે. પરંતુ CSSનો ઉપયોગ કરી આપણે તમામ ઘટકોનો પ્રકાર આપણી પસંદગી પ્રમાણેની સ્ટાઇલ સાથે દર્શાવી શકીએ. માટે, વેબસાઇટમાં વધુ સંખ્યામાં <p> ટૅગ આવેલા હોય તોપણ, CSSનો ઉપયોગ કરી, તમામ <p> ટૅગ માટે એકસાથે સ્ટાઇલ ગોઠવી શકીએ. આમ, એ કહી શકાય કે, HTMLનો ઉપયોગ માહિતીના વર્ણન માટે કરવામાં આવે છે, શૈલી માટે નહીં, જ્યારે CSS દસ્તાવેજમાં ઘટકોની શૈલીનું વર્ણન કરે છે, તેની વિગતોનું નહીં. વેબસાઇટમાં CSSનો ઉપયોગ કરી ફોન્ટના પ્રકાર, ફોન્ટ અને ઘટકોના રંગ, આજુબાજુની જગ્યા (Pad Space), હાંસિયા અને ઘટકોની સ્થિતિનું નિયંત્રણ કરી શકાય છે.

CSSની વાક્યરચના (Syntax of CSS)

CSSની વાક્યરચનામાં rules નામે ઓળખાતાં વિશિષ્ટ ચિહ્નોનો ઉપયોગ કરવામાં આવે છે. CSS rulesના બે મુખ્ય વિભાગ છે : પસંદગીકાર (selector) અને એક કે વધુ ઘોષણાઓ (declarations). સિલેક્ટર એ HTML ઘટક છે, જેના પર આપણે સ્ટાઇલને લાગુ કરવા માંગીએ છીએ. સિલેક્ટર તરીકે ઉપયોગમાં લેવામાં આવેલ HTML ઘટક સાથે સંકળાયેલ ગુણધર્મ અને તેને અનુરૂપ કિંમતોનો સમાવેશ ડિક્લેરેશન વિભાગમાં કરવામાં આવે છે. CSSની સામાન્ય વાક્યરચના નીચે મુજબ વ્યાખ્યાયિત કરી શકાય :

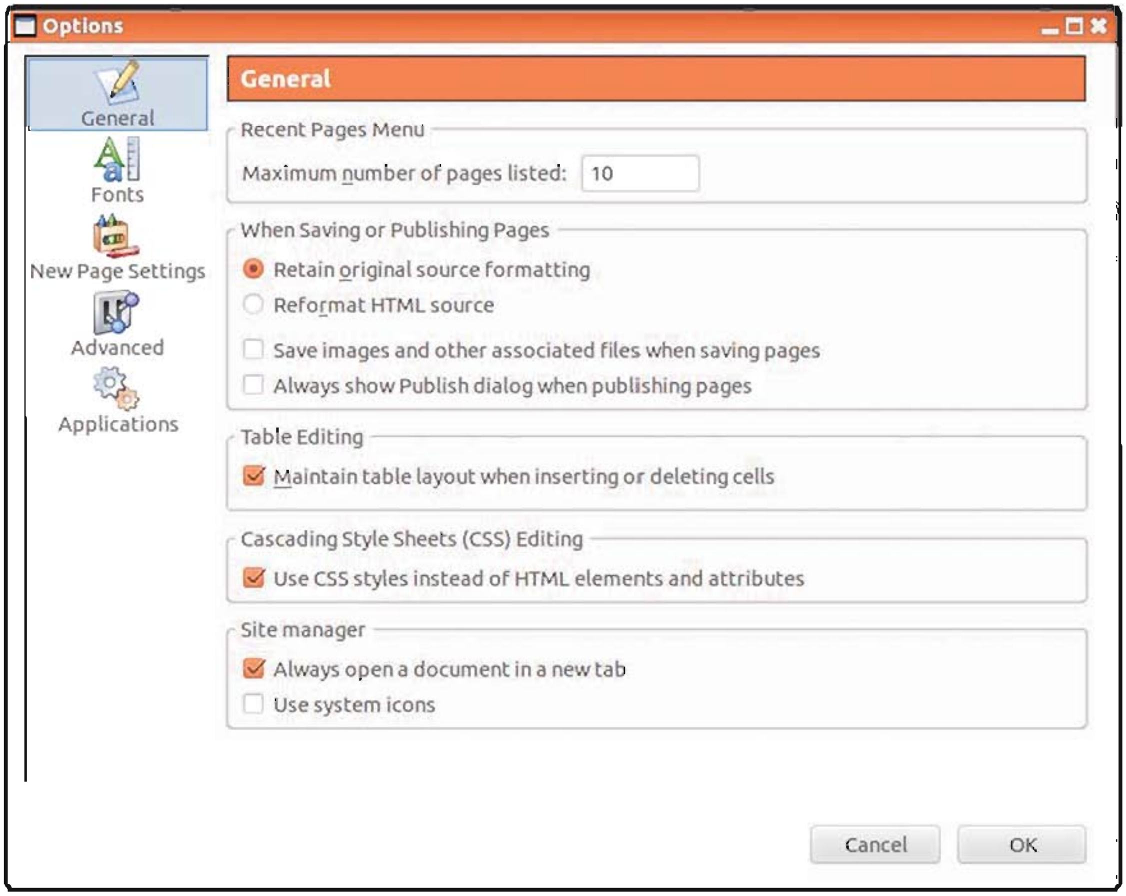
selector {property : value}

આ વાક્યરચનાનું એક ઉદાહરણ આકૃતિ 2.1માં આપ્યું છે.



આકૃતિ 2.1 CSSની વાક્યરચના

કમ્પોઝરનો ઉપયોગ કરી CSS સરળતાથી ઉમેરી શકાય છે. હવે, કમ્પોઝરમાં વેબપેજ સાથે CSS કેવી રીતે ઉમેરી શકાય, તેનો અભ્યાસ કરીએ. કમ્પોઝર પૂર્વનિર્ધારિત રીતે CSSનો ઉપયોગ કરે છે, તેની ખાતરી કરવા **Tools → Preferences** વિકલ્પ પસંદ કરો. આમ કરવાથી આકૃતિ 2.2માં દર્શાવ્યા મુજબ Option ડાયલૉગબૉક્સ ખૂલશે. વિન્ડોની ડાબી બાજુ આવેલ General વર્ગ પસંદ કરો. તેમાં "Use CSS Style Instead of HTML Elements and Attributes" પસંદ કરેલ ન હોય, તો તેને પસંદ કરો. હવે, કમ્પોઝરમાં HTMLને બદલે લખાણની ગોઠવણી માટે CSS સ્ટાઇલનો ઉપયોગ કરવામાં આવશે.



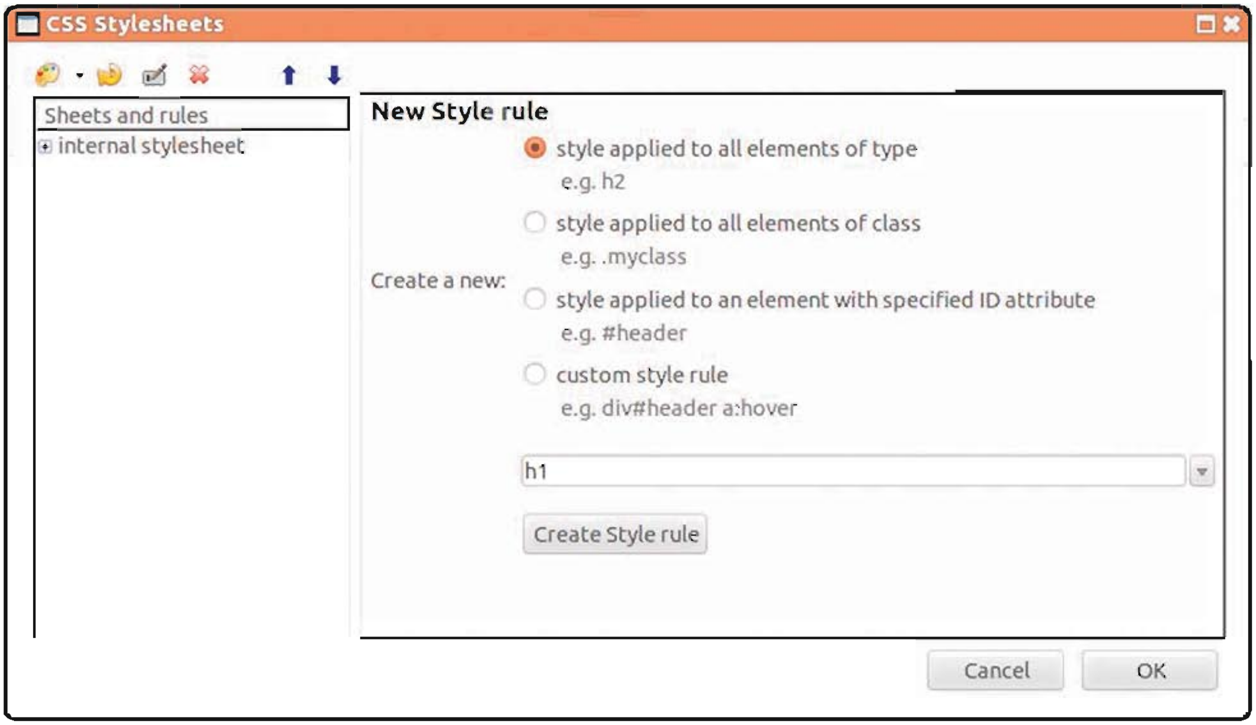
આકૃતિ 2.2 Options ડાયલોગબૉક્સ

ધારો કે, આપણે એક એવી વેબસાઇટની રચના કરવી છે, જેનાં તમામ વેબપેજનાં શીર્ષકો એક નિશ્ચિત શૈલીને અનુસરે. ઉદાહરણ તરીકે, વેબપેજમાં ઉમેરવામાં આવેલાં તમામ Heading1 (h1) નીચે આપેલ શૈલીને અનુસરવાં જોઈએ :

- ➢ Font : Times New Roman
- ➢ Case : Uppercase
- ➢ Alignment : Centre aligned
- ➢ Background color : Light Blue
- ➢ Border : Dotted Border

ઉપરનાં શીર્ષક માટે CSSની રચના કરવા માટે નીચે આપેલ પગલાંને અનુસરો.

- નવી ફાઇલ ખોલો. ફાઇલનો શીર્ષક આપી સંગ્રહ કરો.
- કમ્પોઝિશન ટૂલબાર પર Cascade બટન પર ક્લિક કરો. (નોંધ : જો ફાઇલનો સંગ્રહ કરવામાં ન આવ્યો હોય તો, Page title ડાયલોગબૉક્સ ખૂલશે. પાનાંને શીર્ષક આપ્યા પછી ફાઇલનો સંગ્રહ કરો.) આમ કરવાથી આકૃતિ 2.3માં દર્શાવ્યા મુજબનું ડાયલોગબૉક્સ ખૂલશે. આ ડાયલોગબૉક્સનો ઉપયોગ કરી દરેક ઘટક માટે સ્ટાઇલ વ્યાખ્યાયિત કરી શકાય છે. પ્રથમ રેડિયો-બટન "style applied to all the elements of type" પર ક્લિક કરો.



આકૃતિ 2.3 : CSS Stylesheets ડાયલોગબૉક્સ

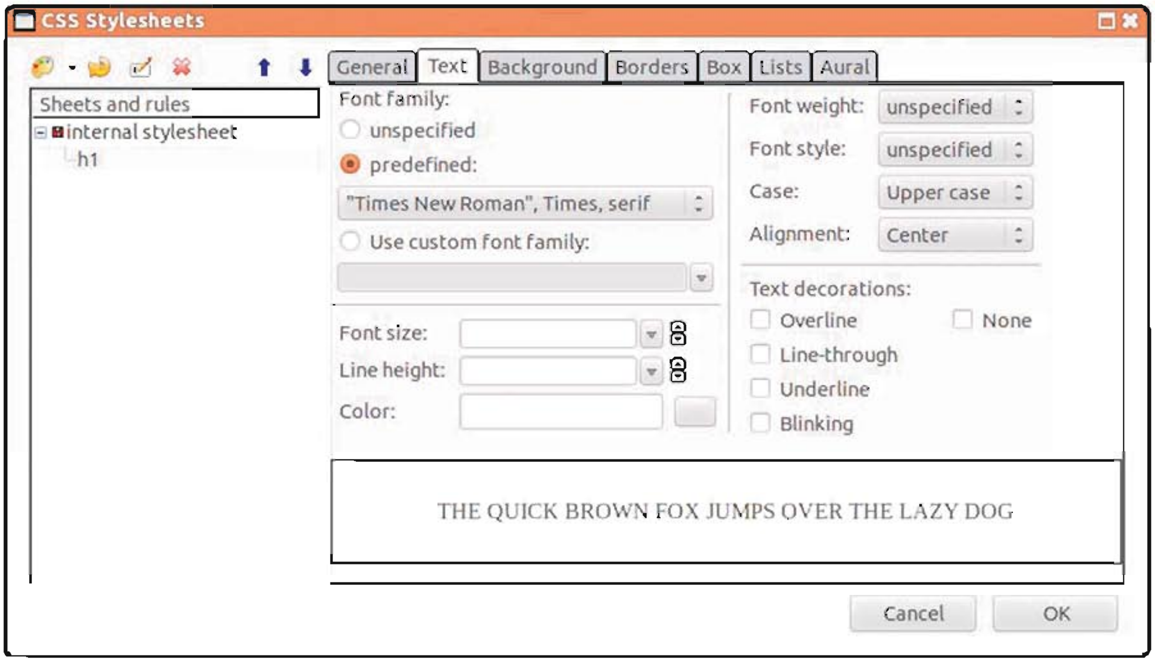
- જેના માટે શૈલીના નિયમો (Style rule) વ્યાખ્યાયિત કરવાના હોય તે ઘટકને ડ્રૉપડાઉન મેનુમાંથી પસંદ કરો. આપણે Heading1 માટે શૈલીની રચના કરવાની હોવાથી આકૃતિ 2.3માં ડ્રૉપડાઉન મેનુમાંથી h1 (Heading1) પસંદ કરવામાં આવ્યું છે.

- Create Style rule બટન પર ક્લિક કરો. જોઈ શકાશે કે CSS Stylesheets ડાયલોગબૉક્સ ખૂલ્લું રહે છે, પરંતુ તેમાંનાં વિકલ્પો બદલાઈ ગયા છે.

- આકૃતિ 2.4માં ડાબી બાજુના વિભાગમાં internal stylesheet શીર્ષકની તરત નીચે h1 ઘટક દર્શાવ્યો છે. જે ઘટક માટે શૈલીના નિયમ વ્યાખ્યાયિત કરવામાં આવે તે દરેકને internal stylesheet શીર્ષક હેઠળ આ યાદીમાં દર્શાવવામાં આવે છે. ડાયલોગબૉક્સમાં જમણી બાજુ General, Text, Background, Borders, Box, Lists અને Aural જેવા વિવિધ વિભાગો (tabs) જોઈ શકાય છે. ઘટકને નિશ્ચિત શૈલી આપવા માટે આ દરેક વિભાગનો ઉપયોગ કરી શકાય છે. પરંતુ જે-તે ઘટક માટે શૈલીના નિયમની રચના કરવામાં આવે, ત્યારે આ બધા જ વિભાગો તેને લાગુ ન પડે એમ પણ બને. હવે, h1 સિલેક્ટર માટે શૈલીના નિયમોની રચના કરીએ.

આકૃતિ 2.4 : h1ના વિકલ્પો સાથે CSS stylesheets ડાયલોગબૉક્સ

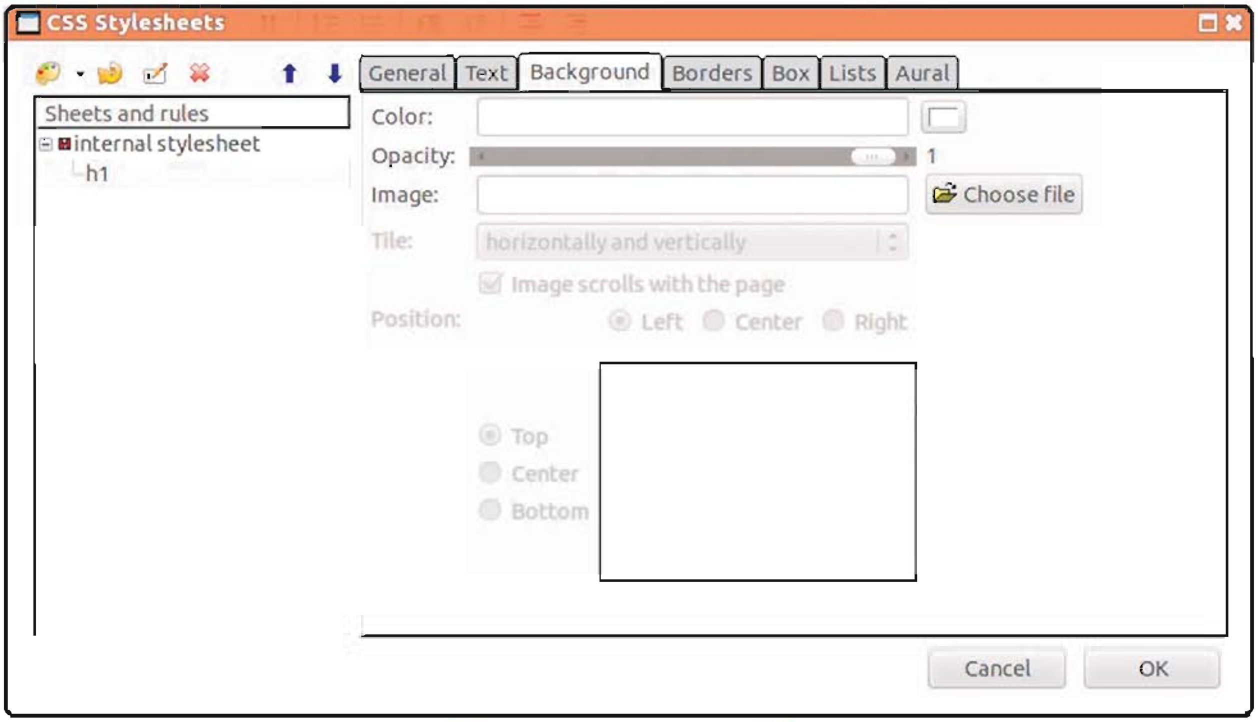
- ડાયલોગબૉક્સની ડાબી બાજુના વિભાગમાં આપેલ h1ને પસંદ કરો. આકૃતિ 2.5માં દર્શાવ્યા મુજબ Text વિભાગ પસંદ કરો. આમ કરવાથી Font family, Font size, Line height, Color, Case, Alignment જેવા વિવિધ વિકલ્પો દર્શાવવામાં આવશે.



આકૃતિ 2.5 : Text વિભાગ

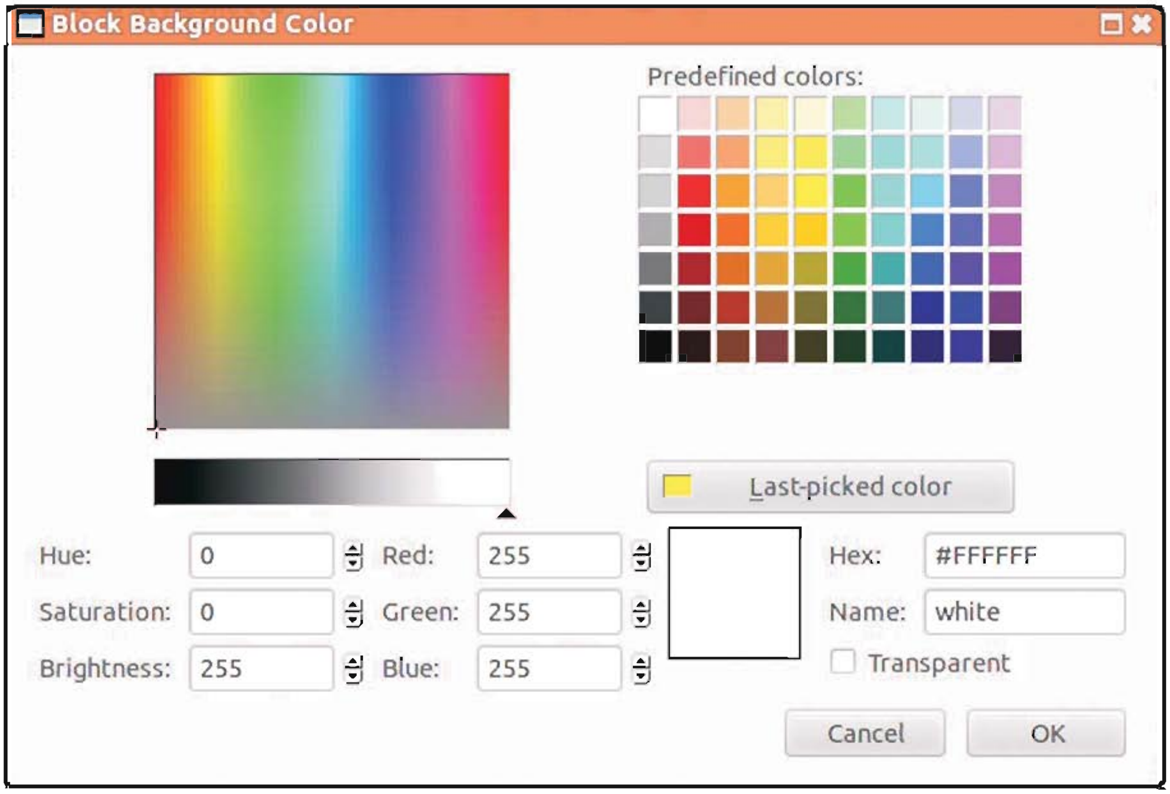
- h1 માટે જરૂરી શૈલી માટેના વિકલ્પોની પસંદગી કરીએ. Font family વિભાગમાં predefined રેડિયો-બટન પસંદ કરી ડ્રૉપડાઉન મેનુમાંથી "Times New Roman" વિકલ્પ પસંદ કરો. Case મેનુમાંથી "Upper Case" અને Alignment મેનુમાંથી "Center" વિકલ્પ પસંદ કરો. પસંદ કરેલા વિકલ્પો આકૃતિ 2.5માં જોઈ શકાય છે. વિકલ્પો બદલવાથી વિન્ડોમાં આવેલું પૂર્વનિર્ધારિત લખાણ પણ શૈલીને અનુરૂપ આપોઆપ બદલાઈ જાય છે તે જોઈ શકાશે.

- બૅકગ્રાઉન્ડ રંગ પસંદ કરવા માટે Background વિભાગ (tab) પસંદ કરો. આમ કરવાથી આકૃતિ 2.6માં દર્શાવ્યા મુજબ બૅકગ્રાઉન્ડ માટેના વિકલ્પો ખૂલશે.



આકૃતિ 2.6 : Background વિભાગ

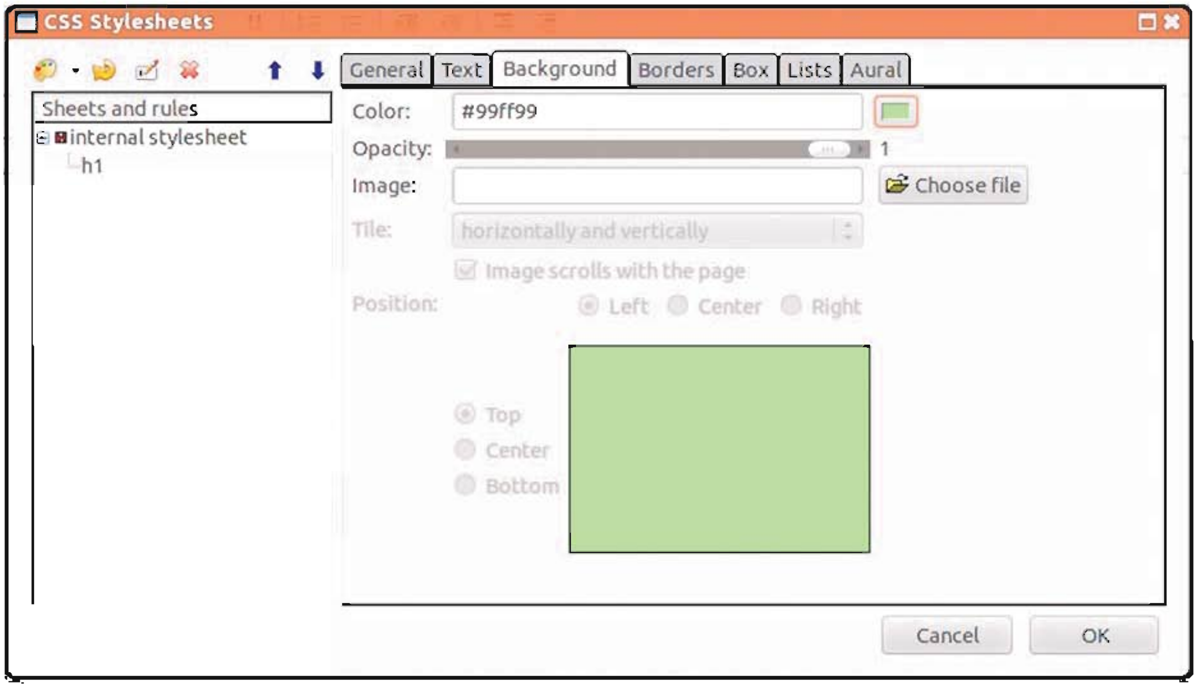
- Color વિકલ્પમાં આવેલ Color Palette બટન પર ક્લિક કરો. આકૃતિ 2.7માં દર્શાવ્યા મુજબ Block Background Color ડાયલોગબૉક્સ રજૂ કરવામાં આવશે. તમારો અપેક્ષિત રંગ પસંદ કરી OK બટન પર ક્લિક કરો.



આકૃતિ 2.7 : Block Background Color ડાયલોગબૉક્સ

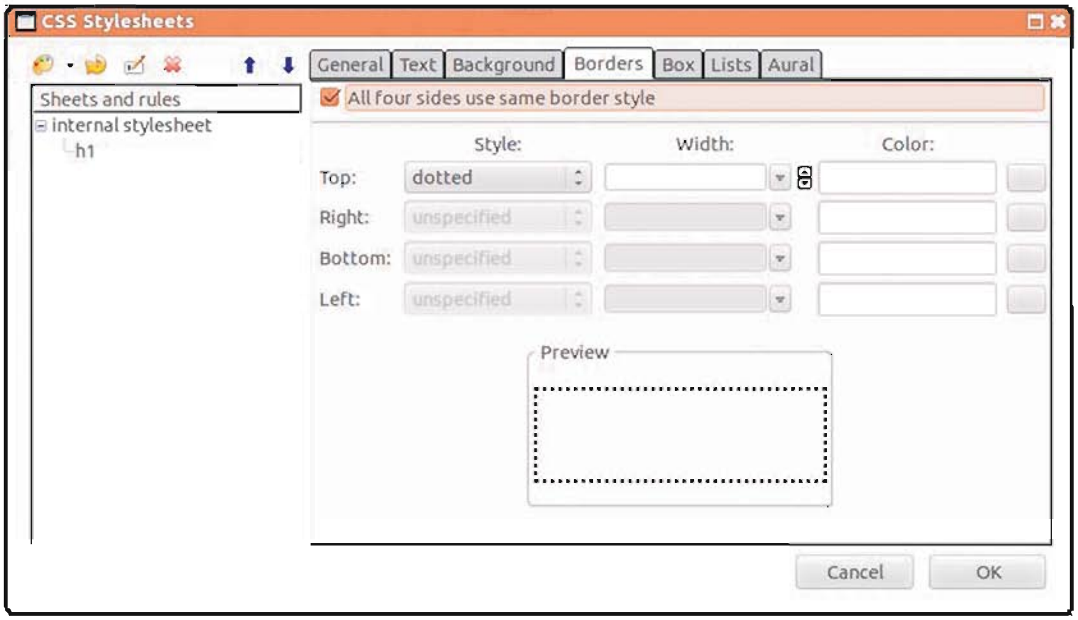
- રંગ પસંદ કર્યા પછી બૅકગ્રાઉન્ડ વિભાગનો દેખાવ આકૃતિ 2.8માં દર્શાવ્યો છે.

નોંધ : જો બૅકગ્રાઉન્ડ તરીકે ચિત્રનો ઉપયોગ કરવા માંગતા હોઈએ, તો Image વિકલ્પમાં "Choose file" પર ક્લિક કરી બૅકગ્રાઉન્ડ માટેની ફાઇલ પસંદ કરવી.



આકૃતિ 2.8 : h1 માટે Background વિભાગમાં પસંદ કરવામાં આવેલા વિકલ્પો

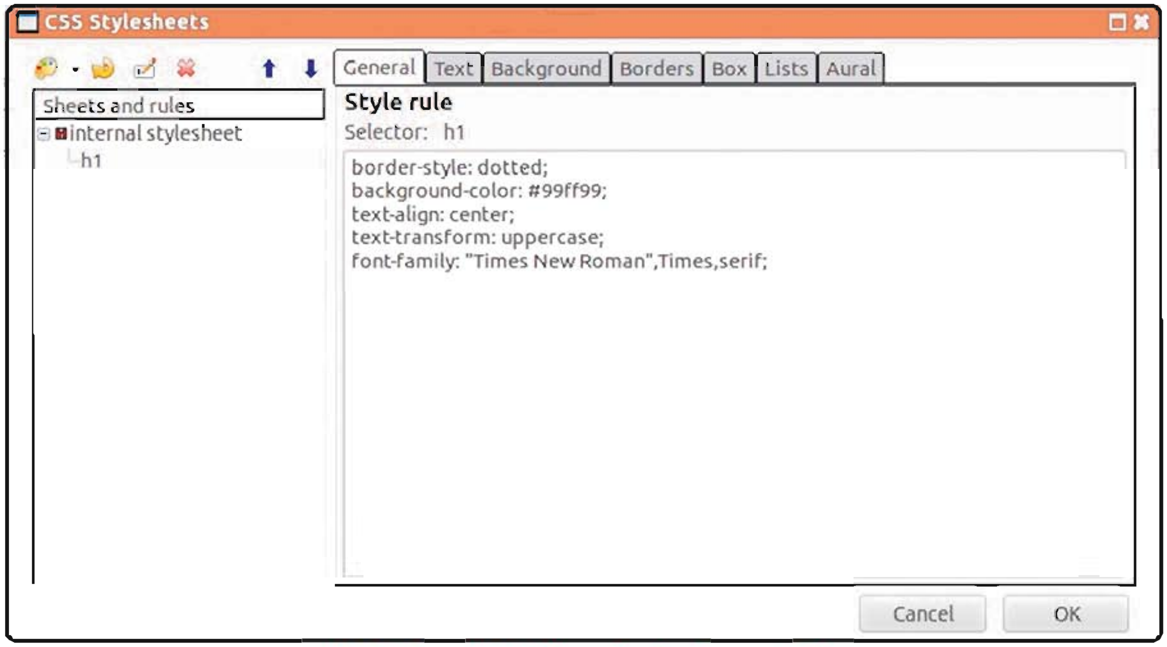
- સીમારેખા ઉમેરવા માટે Border વિભાગ પસંદ કરો. આકૃતિ 2.9માં દર્શાવ્યા મુજબ સીમારેખા માટેના વિકલ્પો દર્શાવવામાં આવશે. જો ચારેબાજુ પર એકસમાન સીમારેખા દર્શાવવી હોય, તો "All four sides use same border style" લખાણની આગળ આવેલું ચેકબોક્સ પસંદ કરો. હવે, માત્ર "Top" વિકલ્પ ઉપલબ્ધ રહેશે, અન્ય તમામ વિકલ્પો અનુપલબ્ધ બનશે. આમ થવાનું કારણ એ છે કે, એક જ બાજુને આપવામાં આવેલી અસર તમામ ચાર બાજુઓ પર લાગુ પાડવામાં આવશે. Style ડ્રોપડાઉન મેનુમાંથી Dotted વિકલ્પ પસંદ કરો. અહીં સીમારેખા માટે અપેક્ષિત જાડાઈ (width) અને રંગ (color) પણ પસંદ કરી શકાય છે. સીમારેખા માટે પસંદ કરેલ શૈલીનું પૂર્વદર્શન (preview) પણ કરી શકાય છે. OK બટન પર ક્લિક કરો.



આકૃતિ 2.9 : h1 માટે Border વિભાગમાં પસંદ કરવામાં આવેલા વિકલ્પ

CSS Stylesheet ડાયલોગબૉક્સમાં અન્ય વિભાગ પસંદ કરવામાં આવે, તો બીજા વિકલ્પ પણ ઉપલબ્ધ છે. જોકે, આપણા ઉદાહરણમાં h1ની શૈલી લાગુ પાડવા માટે માત્ર પસંદગીના વિકલ્પોની જરૂર છે.

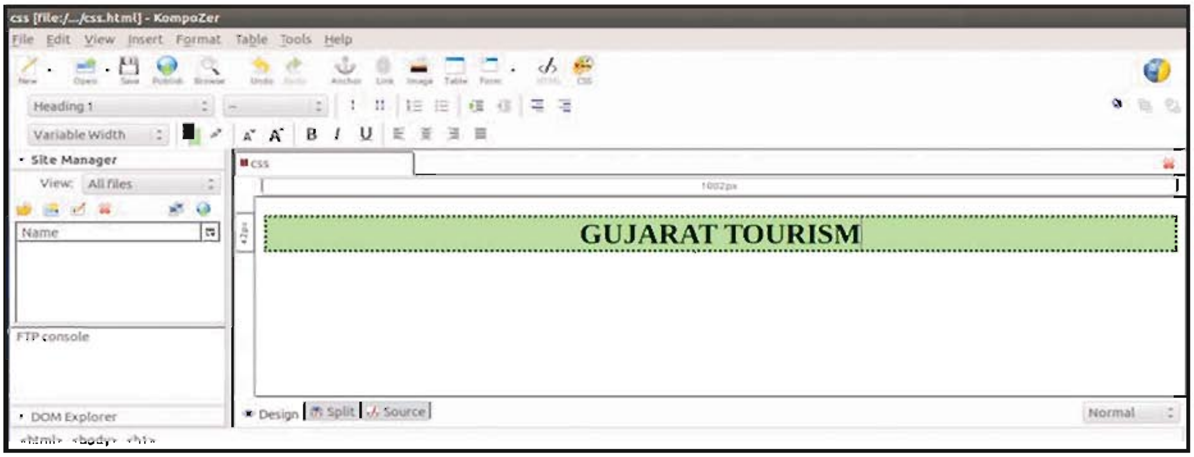
- General વિભાગ પર ક્લિક કરો. આકૃતિ 2.10માં દર્શાવ્યા મુજબ નિર્માણ કરવામાં આવેલ CSS કોડ દર્શાવવામાં આવશે. જો HTMLની જાણકારી હોય, તો આ કોડ સુધારી પણ શકાય છે.



આકૃતિ 2.10 : CSS કોડ દર્શાવતો General વિભાગ

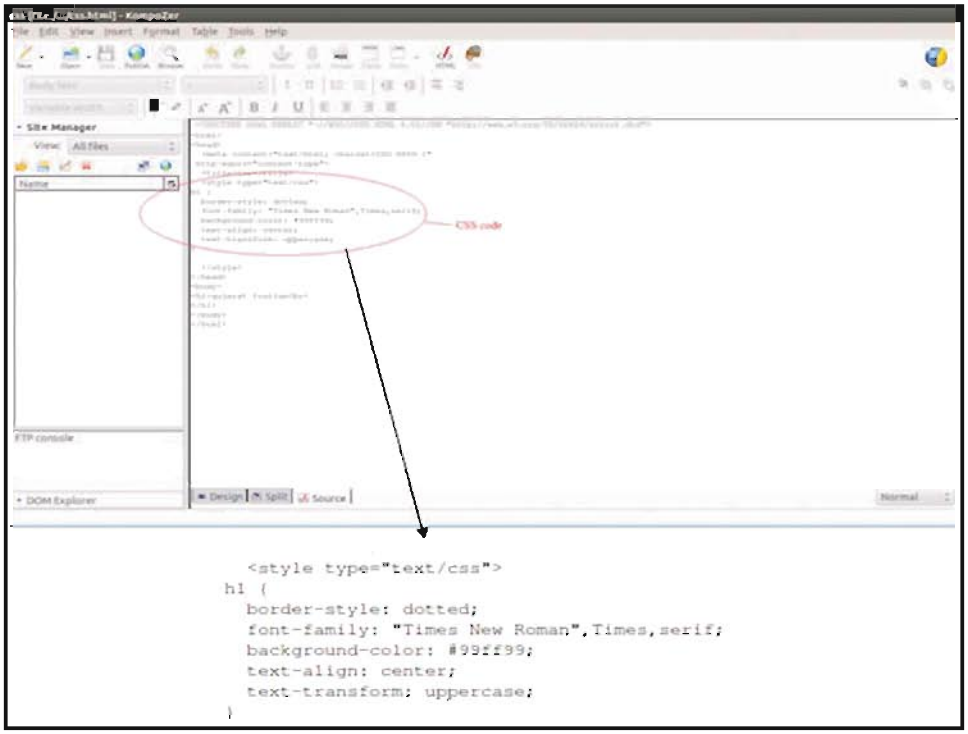
આમ, આપણે h1ને લાગુ પાડી શકાય, તેવી શૈલીની રચના કરી. વેબપેજના અન્ય કોઈ પણ ઘટકની શૈલીની રચના કરવા માટે આ જ પદ્ધતિનો ઉપયોગ કરી શકાય. અન્ય ઘટક ઉમેરવા માટે CSS Stylesheet ડાયલૉગબૉક્સની ઉપર ડાબી બાજુના ખૂણા પર આવેલા CSS બટન ની તીરની નિશાની પર ક્લિક કરો. ડ્રૉપડાઉન મેનુમાંથી "Style-rule" વિકલ્પ પસંદ કરો. શૈલી માટેના નિયમ ઉમેર્યા બાદ OK બટન પર ક્લિક કરો.

હવે, h1 ટૅગ માટે રચવામાં આવેલ CSSનો ઉપયોગ વેબપેજમાં કરીએ. પાનાંમાં શીર્ષક ઉમેરવા માટે ફૉર્મેટ ટૂલબાર-1માંથી Heading1 વિકલ્પ પસંદ કરો. Heading1 વિકલ્પ પસંદ કરવામાં આવે, ત્યારે કર્સર આપોઆપ વેબપેજની મધ્યમાં ગોઠવાઈ જાય છે. (કારણકે આપણે CSSમાં Center ગોઠવણ લાગુ પાડી છે.) થોડું લખાણ ઉમેરો. લખાણ અપરકેસમાં અને Times New Roman ફૉન્ટ સાથે ઉમેરાશે. આકૃતિ 2.11 CSS લાગુ પાડવામાં આવી હોય તેવું Heading1 લખાણ દર્શાવે છે. માટે, હવે પછી કોઈ પણ પાનામાં Heading1 ઉમેરવામાં આવશે, તો તેના માટેની શૈલી દરેક વખતે સમાન રહેશે.



આકૃતિ 2.11 : વેબપેજમાં લાગુ પાડવામાં આવેલ CSS

CSS stylesheet કોડ જોવા માટે વિન્ડોની નીચેના ભાગમાં આવેલ Source વિભાગ પસંદ કરો. આકૃતિ 2.12 પાનાંના સૉર્સકોડમાં આવેલ શીર્ષક વિભાગનો CSS કોડ દર્શાવે છે.



આકૃતિ 2.12 : Source વ્યૂમાં CSS કોડ

CSSના ફાયદા (Advantages of CSS)

ઉપર્યુક્ત ચર્ચા બાદ કહી શકાય કે, CSS તેની નીચે આવેલા સ્તરમાં ફેરફાર કર્યા વગર વેબસાઇટની સંરચના કરવાની સુવિધા પૂરી પાડે છે. વેબસાઇટની ગોઠવણ અને સંરચના માટેના ગુણધર્મોને તેની નીચે આવેલા તેના તાર્કિક બંધારણથી અલગ રાખવામાં આવે, તો વેબસાઇટમાં ફેરફાર કરતી વખતે તેમાં આવેલી વિગતોમાં અકસ્માતે ફેરફાર થવાનો ભય રહેતો નથી.

ઇચ્છિત ઘટકની શૈલીમાં સુધારો કરવા માટે પ્રોગ્રામરે માત્ર CSS ફાઇલમાં ફેરફાર કરવાનો રહે છે, જે કાર્યને સરળ બનાવે છે. દરેક ઘટક માટેની શૈલી માત્ર એક વાર જ ઉમેરવાની હોવાથી, HTMLની સરખામણીએ CSSમાં ઓછો કોડ ઉમેરવો પડે છે. અને તેનાથી વેબપેજ ઝડપથી દર્શાવી શકાય છે. CSSનો ઉપયોગ વેબસાઇટની રચનાને ઝડપી અને કાર્યક્ષમ બનાવે છે.

CSSના ગેરફાયદા (Disadvantages of CSS)

જુદાં-જુદાં બ્રાઉઝર માટે CSSની સુસંગતતા બદલાતી રહે છે. આનો અર્થ એ થયો કે, સ્ટાઇલશીટની કેટલીક સુવિધાઓને દરેક બ્રાઉઝર સમર્થન આપતું નથી અને ઉપયોગકર્તાની અપેક્ષા અનુસાર તે શૈલીને દર્શાવી શકતું નથી. પરંતુ હવે, બ્રાઉઝરની અદ્યતન આવૃત્તિઓ વધુ પ્રમાણભૂત બની છે અને સુસંગતતાનો મુદ્દો ક્ષીણ થયેલો છે.

જાવાસ્ક્રિપ્ટ (JavaScript)

મૂળભૂત રીતે વેબપેજમાં દેખાવ (appearance)નું નિયંત્રણ કરવા માટે HTMLનો ઉપયોગ કરવામાં આવે છે. HTMLમાં બનાવવામાં આવેલા વેબપેજ સ્થિત (Static) છે અને બ્રાઉઝર રેન્ડર કરે પછી તેને બદલી શકાતાં નથી. બીજી તરફ, ઇન્ટરનેટના વિકાસ સાથે લોકો વધુ સારું સંવાદન અને દર્શનીય આલેખનની અપેક્ષા રાખતા થયા. પરંતુ HTML માત્ર અપરિવર્તનશીલ વેબપેજ આપી શકે છે. આમ, વેબ માટેની નવી પ્રોગ્રામિંગ ભાષામાં વધુ સંવાદનની જરૂર ઊભી થઈ. તેથી નેટસ્કેપે જાવાસ્ક્રિપ્ટનો વિકાસ કર્યો. જાવાસ્ક્રિપ્ટ એ વેબપેજમાં પ્રોગ્રામિંગનું પાસું ઉમેરવાની સુવિધા આપતી સ્ક્રિપ્ટિંગ ભાષા છે.

સ્ક્રિપ્ટિંગ ભાષા સરળ અને હળવી પ્રોગ્રામિંગ ભાષા છે, જેમાં 'સી' કે 'જાવા' ભાષા જેવાં અદ્યતન કાર્યોનો સમાવેશ કરવામાં આવ્યો હોતો નથી. વેબસાઇટના આલેખનમાં સુધારો કરવા તથા ફૉર્મની યથાર્થતા ચકાસવા માટે વેબપેજમાં જાવાસ્ક્રિપ્ટનો ઉપયોગ કરવામાં આવે છે. તે HTML પાનામાં સંવાદન ઉમેરે છે. અને તેને HTML કોડમાં પ્રત્યક્ષ રૂપે ઉમેરવામાં આવે છે. હાલમાં મોઝિલા ફાયરફોક્સ, ક્રોમ, સફારી અને ઇન્ટરનેટ એક્સપ્લોરર જેવાં તમામ વેબબ્રાઉઝર જાવાસ્ક્રિપ્ટને સમર્થન આપે છે. જાવાસ્ક્રિપ્ટના ઉપયોગ પછી વેબપેજ અપરિવર્તનશીલ રહેતું નથી, પરંતુ ઉપયોગકર્તા સાથેના સંવાદન, બ્રાઉઝરનું નિયંત્રણ અને HTML વિગતોની ગતિશીલ (dynamic) રજૂઆત માટેના કોડનો સમાવેશ કરી શકે છે.

કમ્પોઝરની મદદથી ફૉર્મ કેવી રીતે બનાવી શકાય તે આપણે જોયું. જ્યારે ઉપયોગકર્તા ફૉર્મમાં વિગતો ઉમેરે ત્યારે, તે કેટલાંક મહત્ત્વનાં ફિલ્ડને ખાલી છોડી દે અથવા ફિલ્ડમાં વિગતો ખોટા સ્વરૂપમાં ઉમેરે તેવું પણ બને. આવા કિસ્સામાં કેટલીક યથાર્થતા પૂરી પાડવી જરૂરી બને છે. યથાર્થતા ઉપયોગકર્તાને ભૂલ કરતા અટકાવે છે. જ્યારે ઉપયોગકર્તા ખોટી વિગત ઉમેરે અથવા ફિલ્ડને ખાલી છોડી દે ત્યારે ભૂલનો સંદેશ દર્શાવવો જોઈએ અને ફૉર્મને સબમિટ થતાં રોકવું જોઈએ.

હાલમાં, જાવાસ્ક્રિપ્ટના સૌથી સામાન્ય વિનિયોગ ક્લાયન્ટ બાજુની સ્ક્રિપ્ટ છે, જે વેબબ્રાઉઝરમાં ચલાવવામાં આવે છે. HTML ફૉર્મમાં ઉમેરેલી વિગતોને સર્વર તરફ મોકલવામાં આવે તે પહેલાં ક્લાયન્ટ બાજુ તેની યથાર્થતા ચકાસવા માટે જાવાસ્ક્રિપ્ટનો ઉપયોગ કરવામાં આવે છે. જાવાસ્ક્રિપ્ટનો ઉપયોગ કરી સામાન્ય રીતે ફૉર્મમાં નીચેની બાબતો તપાસવામાં આવે છે :

- ઉપયોગકર્તાએ કોઈ જરૂરી ફિલ્ડ ખાલી છોડી દીધું છે ?
- ઉપયોગકર્તાએ યોગ્ય ઇ-મેઇલ સરનામું દાખલ કર્યું છે ?

- બે ફિલ્ડની વિગતો સમાન છે કે નહીં ?
- ઉપયોગકર્તાએ યોગ્ય તારીખ દાખલ કરી છે ?
- ઉપયોગકર્તાએ આંકડાકીય ફિલ્ડમાં લખાણ ઉમેર્યું છે ? ઉદાહરણ તરીકે, જથ્થા (quantity) માટેના ફિલ્ડમાં આંકડાકીય વિગતને બદલે ઉપયોગકર્તા શાબ્દિક વિગત દાખલ કરે.

ફૉર્મની યથાર્થતા કેવી રીતે ચકાસવી, તેનો અભ્યાસ કરતાં પહેલાં જાવાસ્ક્રિપ્ટ કોડ કેવી રીતે લખવો તે શીખીએ.

જાવાસ્ક્રિપ્ટ કોડનો અમલ HTML વેબપેજમાં જ કરવામાં આવે છે. આમ, જાવાસ્ક્રિપ્ટ કોડને વેબપેજમાં સીધા જ એક સ્વતંત્ર વિભાગ સ્વરૂપે મૂકી શકાય છે. HTML પાનામાં <script>...</script> ટેગનો ઉપયોગ કરી જાવાસ્ક્રિપ્ટનો કોડ ઉમેરવામાં આવે છે. <script> અને </script> ટેગની વચ્ચે આવેલી લીટીઓ જાવાસ્ક્રિપ્ટનો કોડ ધરાવે છે. યાદ રાખો કે, જાવાસ્ક્રિપ્ટ એ 'કેસ-સેન્સિટિવ' ભાષા છે.

જાવાસ્ક્રિપ્ટને HTMLના <body> અથવા <head> વિભાગમાં સમાવવામાં આવે છે, પરંતુ કોડને <head> ટેગમાં ઉમેરવામાં આવે તે વધુ ઇચ્છનીય છે. <script> ટેગબ્રાઉઝરને તેની વચ્ચે રહેલાં તમામ લખાણોને એક સ્ક્રિપ્ટ તરીકે અર્થઘટન કરવાનો નિર્દેશ કરે છે. કમ્પોઝરના સૉર્સવ્યૂનો ઉપયોગ કરી એક સરળ જાવાસ્ક્રિપ્ટ કોડ લખીએ. આકૃતિ 2.13 જાવાસ્ક્રિપ્ટ સાથેનો સૉર્સવ્યૂ દર્શાવે છે. ફાઇલનો સંગ્રહ કરી તેને વેબબ્રાઉઝરમાં ખોલો.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1"
 http-equiv="content-type">
  <title>js1</title>
  <script>
document.write("HELLO STUDENTS");
  </script>
</head>
<body>
<br>
</body>
</html>
```

આકૃતિ 2.13 : સૉર્સવ્યૂમાં જાવાસ્ક્રિપ્ટ

અહીં જોઈ શકાય છે કે, <script> ટેગમાં માત્ર એક વિધાન "document.write("HELLO STUDENTS")" લખવામાં આવ્યું છે. અહીં write એક પ્રક્રિયા (method) છે, જે document ઑબ્જેક્ટનો એક ભાગ છે. જે લખાણ દર્શાવવાનું હોય, તેને document.write() પ્રક્રિયાના ચલ તરીકે આપવામાં આવે છે.

આકૃતિ 2.13માં વિધાનના અંતમાં આપવામાં આવેલું અર્ધવિરામ ચિહ્ન (semicolon) જાવાસ્ક્રિપ્ટમાં વૈકલ્પિક છે. અર્ધવિરામ ચિહ્નનો ઉપયોગ સમાપ્તિ (termination) ને બદલે અલગતા (separation) માટે થાય છે. તેથી જો દરેક વિધાન નવી લીટીમાં લખવામાં આવે, તો અર્ધવિરામ મૂકવાની જરૂર નથી. આપણા ઉદાહરણમાં document.write વિધાન પછીનું અર્ધવિરામ કાઢી નાખવામાં આવશે તોપણ પરિણામ સમાન રહેશે. આમ છતાં, અર્ધવિરામનો ઉપયોગ એક સારી પ્રોગ્રામિંગ પ્રથા છે. આકૃતિ 2.14માં કોડના પરિણામને બ્રાઉઝરમાં દર્શાવ્યું છે.

આકૃતિ 2.14 : બ્રાઉઝરમાં પરિણામ

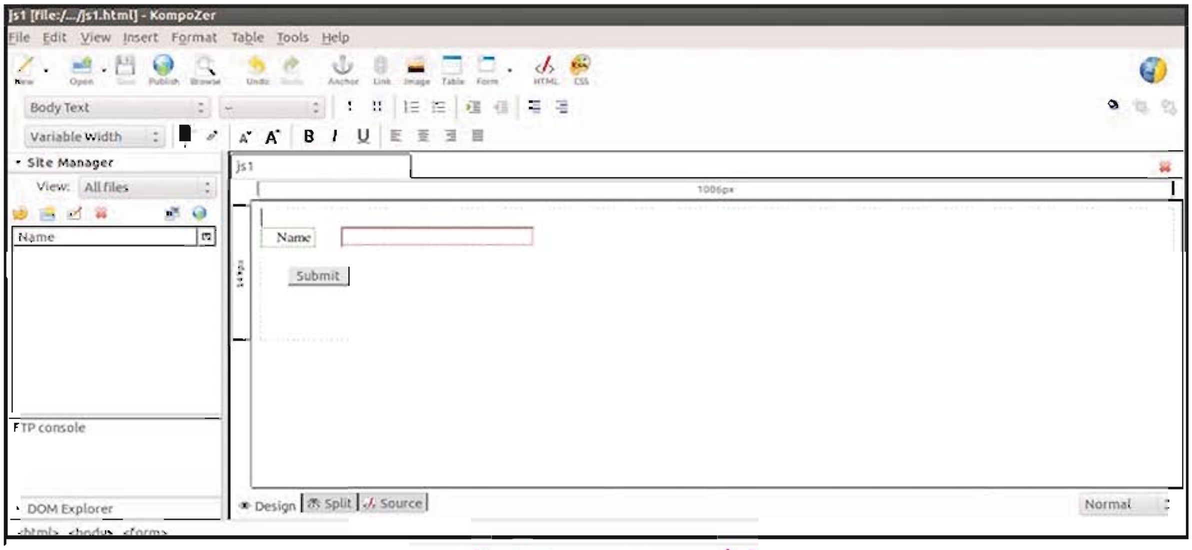
નોંધ : જાવાસ્ક્રિપ્ટ વિધાનોને બ્લૉક સ્વરૂપે એકત્રિત કરી શકાય છે. બ્લૉકની શરૂઆત અને અંત છગડિયા કૌંસ ({ }) દ્વારા કરવામાં આવે છે.

હવે, ફૉર્મની યથાર્થતા માટેની સરળ જાવાસ્ક્રિપ્ટ લખવાનું શીખીએ. પ્રથમ, આપણે બે ફિલ્ડ સાથેનું એક સરળ ફૉર્મ બનાવીએ : name ફિલ્ડ અને submit બટન.

ફૉર્મની યથાર્થતા માટે, જો ઉપયોગકર્તા name ફિલ્ડને ખાલી રાખી submit બટન પર ક્લિક કરે, તો ઉપયોગકર્તા સમક્ષ નામ ઉમેરવા માટેનો સંદેશ પ્રદર્શિત થવો જોઈએ. આ ક્રિયા માટે નીચે જણાવેલ પગલાંને અનુસરો :

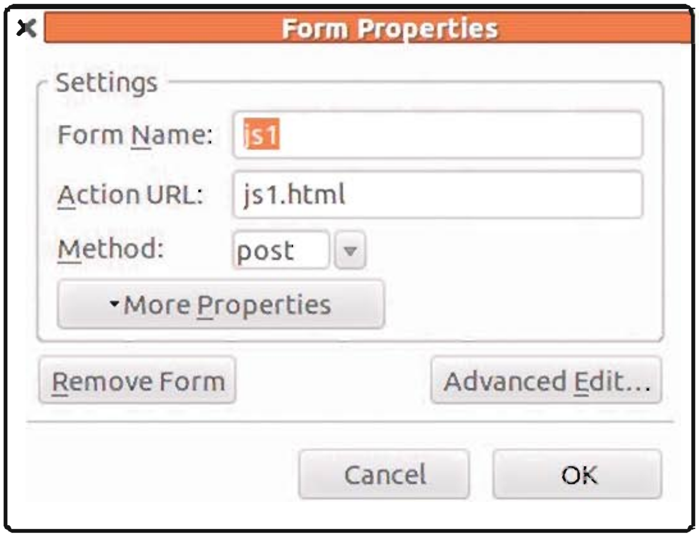
- નવી ફાઇલ બનાવી તેનો સંગ્રહ કરો. આ ઉદાહરણમાં આપણે ફાઇલને js1.html નામ આપી સંગ્રહ કર્યો છે.
- **Form → Define Form** વિકલ્પનો ઉપયોગ કરી નવા ફૉર્મની રચના કરો. ફૉર્મને નામ આપો. આપણા ઉદાહરણમાં ફૉર્મને js1 નામ આપ્યું છે. આકૃતિ 2.15માં દર્શાવ્યા મુજબ નામ માટેનું એક લેબલ અને એક ઇનપુટ ટૅક્સ્ટફિલ્ડ ઉમેરો. એક સબમિટ બટન પણ ઉમેરો.

  નોંધ : ફૉર્મ બનાવતી વખતે આપવામાં આવેલ ફૉર્મ, ઇનપુટ ફિલ્ડ અને કોઈ પણ અન્ય ફિલ્ડનાં નામ સરળ અને યાદ રહે તેવાં હોવાં જોઈએ, કારણકે સ્ક્રિપ્ટમાં વારંવાર તેનો સંદર્ભ આપવાની જરૂર પડશે.



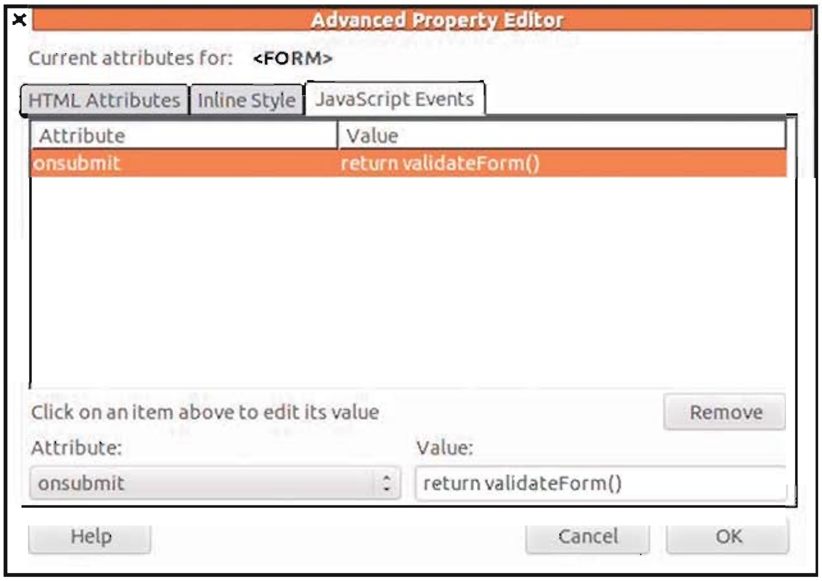
આકૃતિ 2.15 : નમૂનારૂપ ફૉર્મ

- **Form → Define Form** પસંદ કરો. તેથી આકૃતિ 2.16માં દર્શાવવામાં આવેલ Form Properties ડાયલોગબૉક્સ ખૂલશે.



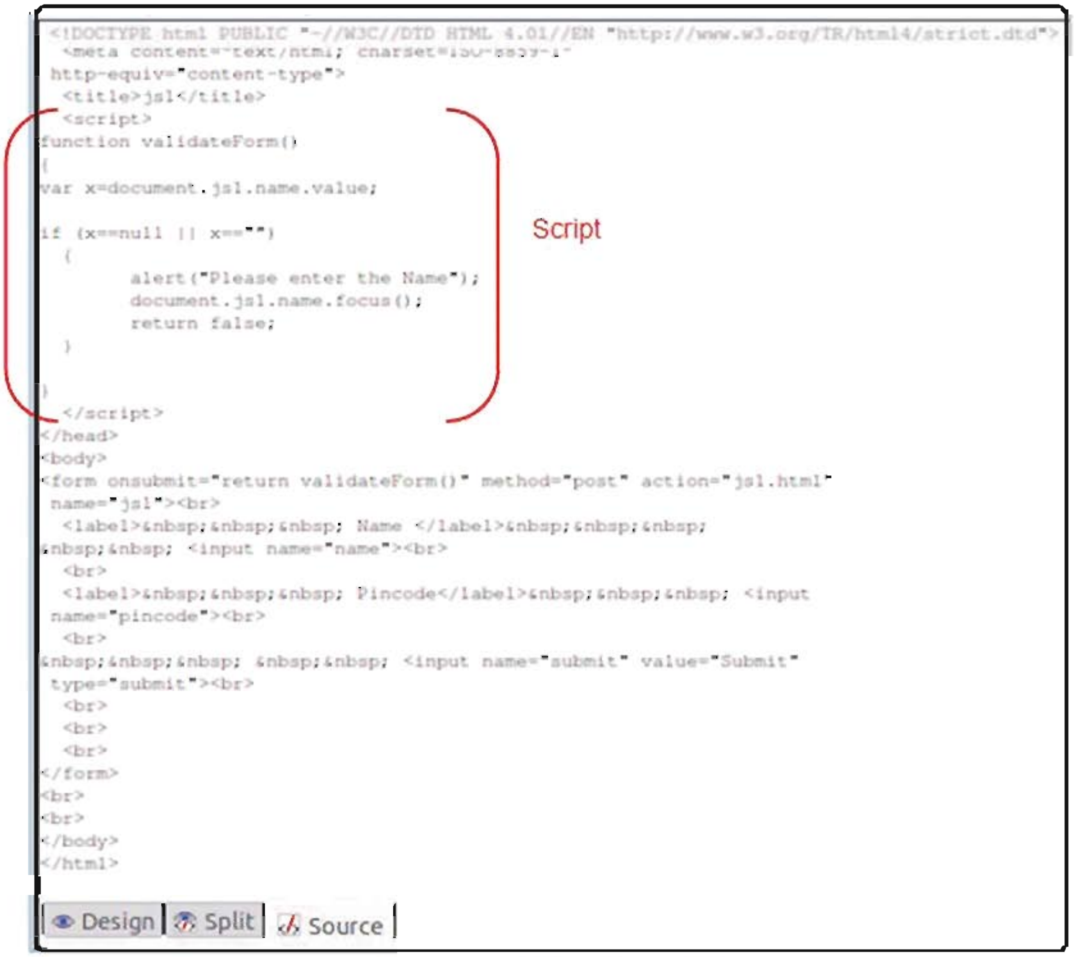
આકૃતિ 2.16 : Form Properties ડાયલોગબૉક્સ

- Advanced Edit બટન પર ક્લિક કરો. તેનાથી આકૃતિ 2.17માં દર્શાવેલ Advanced Property Editor ડાયલોગબૉક્સ ખૂલશે. JavaScript Events વિભાગ પસંદ કરો. આકૃતિ 2.17માં દર્શાવ્યા પ્રમાણે Attribute વિકલ્પમાં "onsubmit" વિકલ્પ પસંદ કરો અને Valueમાં "return validateForm()" લખો. OK બટન પર ક્લિક કરો.



આકૃતિ 2.17 : Advanced Property Editor ડાયલોગબૉક્સ

- હવે, સોર્સવ્યૂ ખોલો. આકૃતિ 2.18માં દર્શાવ્યા મુજબ <head> ટૅગની અંદર સ્ક્રિપ્ટ ઉમેરો.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
  <meta content="text/html; charset=ISO-8859-1"
 http-equiv="content-type">
  <title>js1</title>
  <script>
function validateForm()
{
var x=document.js1.name.value;

if (x==null || x=="")
  {
        alert("Please enter the Name");
        document.js1.name.focus();
        return false;
  }

}
  </script>
</head>
<body>
<form onsubmit="return validateForm()" method="post" action="js1.html"
 name="js1"><br>
  <label>    Name </label>   
   <input name="name"><br>
  <br>
  <label>    Pincode</label>    <input
 name="pincode"><br>
  <br>
       <input name="submit" value="Submit"
 type="submit"><br>
  <br>
  <br>
  <br>
</form>
<br>
<br>
</body>
</html>
```


આકૃતિ 2.18 : સોર્સવ્યૂમાં જાવાસ્ક્રિપ્ટ

આકૃતિ 2.8માં દર્શાવેલ સ્ક્રિપ્ટમાં નીચેના ઘટકો વ્યાખ્યાયિત કરવામાં આવ્યા છે :

- validateForm() વિધેય
- *x* ચલ
- if શરતી વિધાન
- alert() આંતરપ્રસ્થાપિત વિધેય
- document ઑબ્જેક્ટ
- focus પદ્ધતિ

હવે, દરેક ઘટકની સંક્ષેપમાં ચર્ચા કરીએ.

ValidateForm વિધેય (Function ValidateForm())

નિશ્ચિત કાર્યને પાર પાડતા પુન:ઉપયોગમાં લઈ શકાય તેવા કોડના બ્લૉકને વિધેય (અથવા રૂટિન) કહે છે. વિધેયને function કી-વર્ડ વડે વ્યાખ્યાયિત કરવામાં આવે છે અને કોડના બ્લૉકને છગડિયા કૌંસ "{ }"માં લખવામાં આવે છે. અહીં validateForm() એક વિધેય છે.

જાવાસ્ક્રિપ્ટ alert() જેવાં કેટલાંક આંતરપ્રસ્થાપિત વિધેય પૂરાં પાડે છે. alert() વિધેય લખાણ સ્વીકારે છે અને તેને એલર્ટ બૉક્સમાં દર્શાવે છે.

ઉદાહરણ તરીકે,

```
function hello()
{
  alert("Hello Students");
}
```

જ્યારે "hello()" વિધેય બોલાવવામાં આવે (called), ત્યારે Hello Students સંદેશ સાથેનું એલર્ટ બૉક્સ દર્શાવવામાં આવશે.

વિધેયનો અમલ કોઈ ઘટના (event) દ્વારા અથવા સૉર્સકોડમાંથી બોલાવવામાં આવે (called) ત્યારે કરવામાં આવે છે. વિધેયને સૉર્સકોડમાં કોઈ પણ સ્થાનેથી બોલાવી શકાય છે. HTML કોડના head કે body વિભાગમાં વિધેયને વ્યાખ્યાયિત કરી શકાય છે. વિધેયને તેનાં નામથી બોલાવવામાં આવે છે. તે કૌંસમાં આપેલી કિંમતો સ્વીકારી શકે છે. આ કિંમતો વિધેયને પ્રાચલ (parameter) તરીકે મોકલવામાં આવે છે. વિધેય કિંમત પરત કરે, તેવું પણ આપણે ક્યારેક ઇચ્છતા હોઈએ. return વિધાનની મદદથી આ શક્ય બને છે. return વિધાનનો ઉપયોગ વિધેયનો અમલ અટકાવી કોઈ નિશ્ચિત કિંમત પરત કરે છે.

આપણા ઉદાહરણમાં validateForm( ) વિધેયને સબમિટ બટન પર ક્લિક આપવાથી ઉદ્ભવતી ઘટના દ્વારા બોલાવવામાં (call) આવે છે. આમ, જ્યારે ઉપયોગકર્તા સબમિટ બટન પર ક્લિક કરશે, જેને ઘટના (event) તરીકે ઓળખવામાં આવે છે, ત્યારે વિધેયને બોલાવી તેમાં આવેલાં વિધાનોનો અમલ કરવામાં આવશે. વિધેય ખોટી (false) કિંમત પરત કરશે.

ઘટના (Events)

ઉપયોગકર્તા દ્વારા કરવામાં આવતી પ્રક્રિયાનો પ્રતિભાવ આપે તેવા સંવાદિત વેબપેજની રચના કરવા માટે જાવાસ્ક્રિપ્ટ ઉપયોગી છે. ઉપયોગકર્તા અને વેબપેજ વચ્ચેના સંવાદન દ્વારા ઘટના (Event) ઉત્પન્ન થાય છે. બીજા શબ્દોમાં કહીએ તો, જ્યારે ઉપયોગકર્તા કંઈ પણ કાર્ય કરે, ત્યારે ઘટના ઉદ્ભવે છે. કોષ્ટક 2.1માં જાવાસ્ક્રિપ્ટ ઘટનાઓની યાદી આપવામાં આવી છે.

ઘટના	સમજૂતી
abort	ચિત્ર દર્શાવવાનું રદ કરવામાં આવે, ત્યારે
blur	રેડિયો-બટન જેવું કોઈ ઘટક નિષ્ક્રિય બને, ત્યારે
click	ઉપયોગકર્તા ફૉર્મના કોઈ ઘટક પર ક્લિક કરે, ત્યારે
change	ઉપયોગકર્તા દ્વારા ફૉર્મના ફિલ્ડની કિંમત બદલવામાં આવે, ત્યારે
error	દસ્તાવેજ કે ચિત્ર દર્શાવતી વખતે ક્ષતિ ઉદ્ભવે, ત્યારે
focus	બટન જેવો કોઈ ઘટક સક્રિય બને, ત્યારે
load	દસ્તાવેજ કે ચિત્ર દર્શાવવામાં આવે, ત્યારે
mouseout	માઉસનું પૉઈન્ટર ઘટક પરથી દૂર કરવામાં આવે, ત્યારે
mouseover	માઉસનું પૉઈન્ટર ઘટક પર લાવવામાં આવે, ત્યારે
reset	ફૉર્મના ફિલ્ડને પૂર્વનિર્ધારિત કિંમતો સાથે રીસેટ કરવામાં આવે, ત્યારે
select	ઉપયોગકર્તા ફૉર્મનું ફિલ્ડ પસંદ કરે, ત્યારે
submit	ઉપયોગકર્તા ફૉર્મ સબમિટ કરે, ત્યારે
unload	ઉપયોગકર્તા પાનું છોડી દે, ત્યારે

કોષ્ટક 2.1 : જાવાસ્ક્રિપ્ટ ઘટનાઓ

જ્યારે કોઈ ઘટના ઉદ્ભવે છે ત્યારે આપેલ સ્થિતિ અનુસાર એક નિશ્ચિત જાવાસ્ક્રિપ્ટ કોડનો અમલ કરવામાં આવે છે. આ જાવાસ્ક્રિપ્ટ કોડને ઘટના-સંચાલક (event handler) કહે છે. ઈવેન્ટ હૅન્ડલરનું નામ ઘટનાનાં નામ જેવું જ રાખવામાં આવે છે. ઉદાહરણ તરીકે, 'ક્લિક' ઘટના માટે ઈવેન્ટ હૅન્ડલરનું નામ onclick() છે. એ જ રીતે 'સબમિટ' ઘટના માટે onsubmit() ઈવેન્ટ હૅન્ડલરનો ઉપયોગ કરવામાં આવશે.

આપણા ઉદાહરણમાં ઉપયોગકર્તા સબમિટ બટન પર ક્લિક કરે ત્યારે, એટલે કે સબમિટ ઘટના વખતે ફૉર્મની યથાર્થતા ચકાસવાની જરૂર છે. આ માટે આપણે onsubmit ઈવેન્ટ હૅન્ડલરનો ઉપયોગ કરવાની જરૂર છે. આકૃતિ 2.18માં onsumbit() ઈવેન્ટ હૅન્ડલર ઉમેરવામાં આવ્યું છે. આ ઈવેન્ટ હૅન્ડલર validateForm વિધેયને બોલાવશે. 'name' ફિલ્ડમાં કોઈ કિંમત છે કે તે ખાલી (null) છે તેના આધારે True કે False પરત કરશે.

ચલ (Variable)

આપણા ઉદાહરણમાં, validateForm() વિધેયમાં નીચેનું વિધાન લખ્યું છે :

```
var x=document.js1.name.value
```

અહીં આપણે x નામનો ચલ ઘોષિત કર્યો છે. ચલ એ વિગતોનો સંગ્રહ કરતો સંગ્રાહક છે. યાદ રાખો કે, જાવાસ્ક્રિપ્ટમાં ચલ પણ કેસ-સૅન્સિટિવ છે. માટે ચલ "x" અને ચલ "X" સમાન નથી. ચલમાં અંકો, સ્ટ્રિંગ કે લખાણનો સંગ્રહ કરી શકાય છે. જાવાસ્ક્રિપ્ટમાં var કી-વર્ડનો ઉપયોગ કરી ચલની ઘોષણા કરી શકાય છે. ઉદાહરણ તરીકે,

```
var x=3;
var y="Hello";
var z="Hello Students";
```

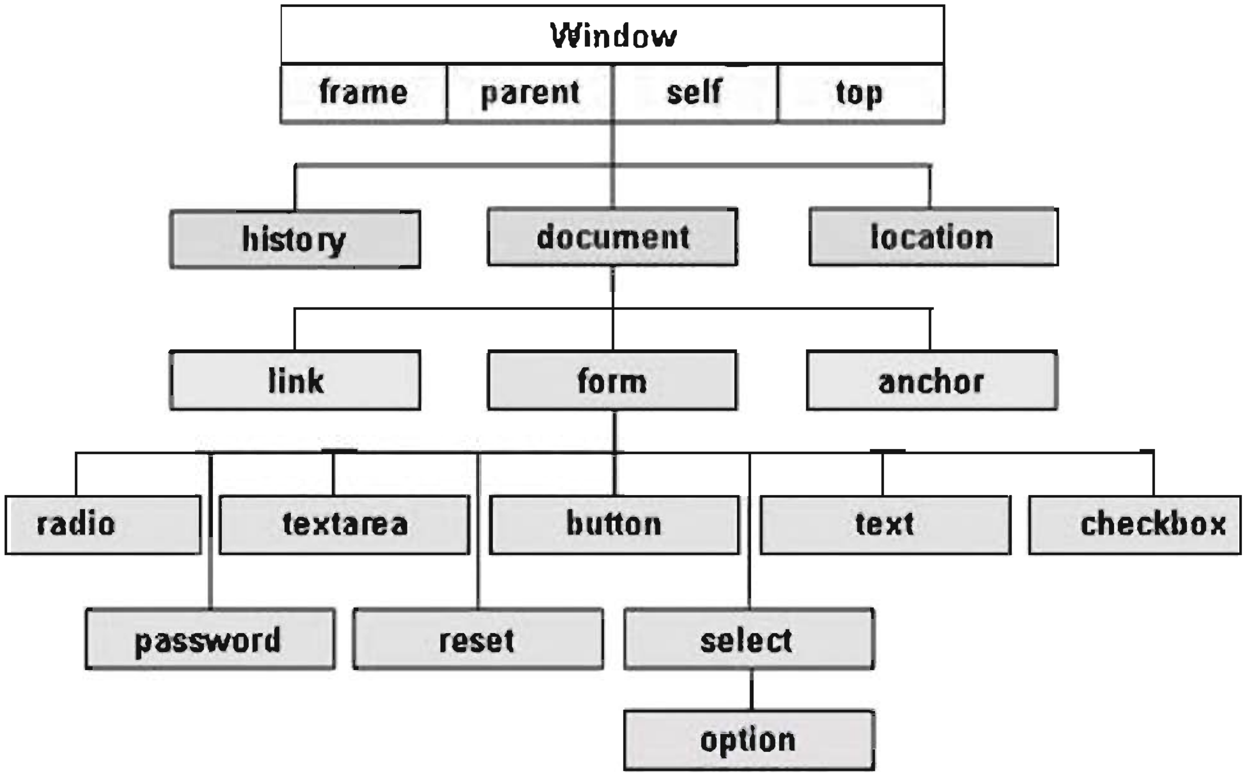
If વિધાન (If Statement)

તમે શરતી વિધાન ifથી સુપરિચિત છો જ. પ્રોગ્રામનો પ્રવાહ બદલવા માટે શરતી વિધાન ifનો ઉપયોગ કરવામાં આવે છે. કોઈ નિશ્ચિત શરતની યથાર્થતા તપાસવા માટે if વિધાન પદાવલીનું મૂલ્યાંકન કરે છે. જો શરત સાચી હોય, તો પ્રોગ્રામ if વિધાનના બ્લૉકમાં પ્રવેશી તેમાં આવેલાં વિધાનોનો અમલ કરે છે. આપણા ઉદાહરણમાં x ચલની કિંમત ખાલી (null) છે કે નહીં તે ચકાસવા માટે if વિધાનનો ઉપયોગ કર્યો છે. જો ચલ xની કિંમત ખાલી હશે, તો if બ્લૉકનો અમલ કરવામાં આવશે.

ડૉક્યુમેન્ટ ઑબ્જેક્ટ મૉડેલ (Document Object Model) - document.js1.name.value

કેટલીક વાર, વેબબ્રાઉઝરના નિયંત્રણ માટે જાવાસ્ક્રિપ્ટનો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ તરીકે, ધારો કે તમારે વેબપેજને બદલવાની કે તેમાં આવેલા ઘટકોનું નિયંત્રણ કરવાની જરૂર છે. વેબ બ્રાઉઝરની વિન્ડો કે વેબપેજના નિયંત્રણ માટે બ્રાઉઝર ઑબ્જેક્ટ મૉડેલ (Browser Object Model - BOM)નો ઉપયોગ કરવામાં આવે છે.

તમામ બ્રાઉઝરને જુદા જુદા વિભાગોમાં અથવા ઘટકોમાં વિભાજિત કરવામાં આવે છે, જેને જાવાસ્ક્રિપ્ટ દ્વારા ઉપયોગમાં લઈ શકાય છે. આ વિભાગોને બ્રાઉઝર ઑબ્જેક્ટ મૉડેલ અથવા BOM તરીકે ઓળખવામાં આવે છે. આકૃતિ 2.19માં દર્શાવ્યા મુજબ આ બ્રાઉઝરના પદાનુક્રમમાં સૌથી ઉપર વિન્ડો ઑબ્જેક્ટ હોય છે.



આકૃતિ 2.19 : બ્રાઉઝર ઑબ્જેક્ટનો પદાનુક્રમ

અહીં ટૂલબાર, મેનુ, સ્ટેટસબાર, વેબપેજ અને અન્ય ઘણા ઘટકો સાથેનો સમગ્ર બ્રાઉઝરનો પદાનુક્રમ રજૂ કરેલો છે. બ્રાઉઝરમાં દેખાતી વિન્ડો કે તેમાં આવેલા ઘટકોને બદલવા માટે બ્રાઉઝર ઑબ્જેક્ટ મૉડેલમાં આવેલા ગુણધર્મો (properties) અને પદ્ધતિઓ (methods)નો ઉપયોગ કરવામાં આવે છે. બ્રાઉઝર ઑબ્જેક્ટ મૉડેલમાં ઑબ્જેક્ટ બનાવવાની જરૂર પડતી નથી; પરંતુ જ્યારે બ્રાઉઝર વેબપેજ દર્શાવે, ત્યારે તેની રચના આપોઆપ થતી હોય છે.

બ્રાઉઝર ઑબ્જેક્ટ મૉડેલમાં સૌથી ઉપરના સ્તરનો ઑબ્જેક્ટ Window છે. Window ઑબ્જેક્ટ બ્રાઉઝરની વિન્ડો અથવા તેમાં આવેલી સ્વતંત્ર ફ્રેમને રજૂ કરે છે. તેની રચના બ્રાઉઝર દ્વારા આપોઆપ કરવામાં આવે છે. Window ઑબ્જેક્ટ વૈશ્વિક (global) ઑબ્જેક્ટ ગણાય છે. કારણકે, મૉડેલના અન્ય તમામ બ્રાઉઝર ઑબ્જેક્ટનો તેની અંદર સમાવેશ કરવામાં આવે છે. ઉદાહરણ તરીકે, Window ઑબ્જેક્ટમાં document ઑબ્જેક્ટનો સમાવેશ થાય છે. Window ઑબ્જેક્ટની પદ્ધતિઓ (methods) અને ગુણધર્મો (properties) દ્વારા વેબબ્રાઉઝરની વિન્ડોને નિયંત્રિત કરવામાં આવે છે, જ્યારે document ઑબ્જેક્ટની પદ્ધતિઓ (methods) અને ગુણધર્મો (properties) દ્વારા વેબપેજને નિયંત્રિત કરવામાં આવે છે.

બ્રાઉઝર ઑબ્જેક્ટ મૉડેલમાં સૌથી અગત્યનો ઑબ્જેક્ટ document ઑબ્જેક્ટ છે. બ્રાઉઝરમાં વેબપેજ રજૂ કરવા માટે તેનો ઉપયોગ કરવામાં આવે છે. વેબપેજના અન્ય તમામ ઘટકો જેવા કે, ફૉર્મ, ચિત્રો, લિંક અને અન્ય ઘટકો document ઑબ્જેક્ટની અંદર સમાવવામાં આવ્યા છે. ઉદાહરણ તરીકે, <form> ઘટક દ્વારા રચવામાં આવેલો form ઑબ્જેક્ટ જાવાસ્ક્રિપ્ટ દ્વારા document ઑબ્જેક્ટની અંદર રજૂ કરવામાં આવે છે. document ઑબ્જેક્ટમાં form ઑબ્જેક્ટનો સમાવેશ કરવામાં આવ્યો છે. તેવી જ રીતે form ઑબ્જેક્ટમાં element ઑબ્જેક્ટને સમાવવામાં આવ્યો છે. ફૉર્મના દરેક ઘટકનો સંદર્ભ મેળવવા element ઑબ્જેક્ટનો ઉપયોગ કરવામાં આવે છે. element ઑબ્જેક્ટ તરીકે રેડિયો-બટન, ટૅક્સ્ટબૉક્સ, ચેકબૉક્સ કે અન્ય કોઈ પણ ઘટક હોઈ શકે છે.

focus પદ્ધતિ (focus Method)

સ્ક્રિપ્ટમાં focus નામની પદ્ધતિનો પણ ઉપયોગ કરવામાં આવે છે. ફૉર્મના નિશ્ચિત ઘટક પર નિયંત્રણ લઈ જવા તેને લાગુ પાડવામાં આવે છે. આમ, ઉપયોગકર્તા એલર્ટ સંદેશ મેળવે પછી, focus પદ્ધતિની મદદથી ઘટકની અંદર કર્સર ગોઠવવામાં આવે છે અને તે નિશ્ચિત ઘટક પ્રકાશિત (highlighted) બને છે. અહીં નિયંત્રણને name ઘટક પર લઈ જવામાં આવ્યું છે.

હવે, ફાઇલને બ્રાઉઝરમાં ખોલીએ. આકૃતિ 2.20માં પરિણામ દર્શાવ્યું છે.

આકૃતિ 2.20 : બ્રાઉઝરમાં પરિણામ

name ફિલ્ડને ખાલી રાખી submit બટન પર ક્લિક કરો. આકૃતિ 2.21માં દર્શાવ્યા પ્રમાણે એક એલર્ટ સંદેશ રજૂ કરવામાં આવશે.

આકૃતિ 2.21 : ખાલી name ફિલ્ડ માટેનો એલર્ટ સંદેશ

alertનો સંદેશ એ સૉર્સવ્યૂમાં લખવામાં આવેલ સ્ક્રિપ્ટનું પરિણામ છે. આ એવી યથાર્થતા છે, જે ફૉર્મને જાવાસ્ક્રિપ્ટ દ્વારા

પૂરી પાડવામાં આવી છે, આમ, અનેક ફિલ્ડ ધરાવતા ફૉર્મમાં કોઈ પણ ફિલ્ડ ખાલી રાખવામાં આવે, તો તેને ચકાસવા માટે યથાર્થતા ઉમેરી શકાય છે.

ઇન્ટરનેટ પર ફૉર્મ ભરતી વખતે તમે કેટલાંક ફિલ્ડને લાલ રંગની ફૂદડી "*" સાથે દર્શાવેલાં જરૂર જોયાં હશે. તે ઉપયોગકર્તાને નિર્દેશ કરે છે કે આ ફિલ્ડ ફરજિયાત છે અને તેને ખાલી છોડવાં જોઈએ નહીં. જાવાસ્ક્રિપ્ટનો ઉપયોગ કરી આ પ્રકારનાં ફિલ્ડની વિગતો તપાસી શકાય છે તથા ઉપયોગકર્તા સમક્ષ એલર્ટ સંદેશ દર્શાવી શકાય છે.

હવે, ફૉર્મમાં Pincode નામનું અન્ય ફિલ્ડ ઉમેરીએ. આ ફિલ્ડ પર નીચે જણાવેલ યથાર્થતા લાગુ પાડવામાં આવશે :

- ઉપયોગકર્તા આ ફિલ્ડને ખાલી છોડી શકશે નહીં.
- માત્ર અંક માન્ય હશે. (અક્ષર માન્ય રાખવામાં આવશે નહીં).
- પિનકોડ છ અંકોનો હોવો જોઈએ.

જો કોઈ પણ યથાર્થતાનો ભંગ કરવામાં આવશે તો ઉપયોગકર્તા સમક્ષ એલર્ટ સંદેશ દર્શાવવામાં આવશે. આ યથાર્થતા ઉમેરવા નીચે જણાવેલ પગલાંને અનુસરો :

- આ જ ફૉર્મમાં પિનકોડ માટેનું એક લેબલ અને ઇનપુટ ફિલ્ડ ઉમેરો.
- આકૃતિ 2.22માં દર્શાવેલ સ્ક્રિપ્ટ સૉર્સવ્યૂમાં ઉમેરો.

```
<script>
function validateForm()
{
 var x=document.js1.name.value;
 var y=document.js1.pincode.value;
 if (x==null || x=="")
 {
  alert("Please enter the Name");
  document.js1.name.focus();
  return false;
 }

 if (y=="" || isNaN(y) || y.length>6 || y.length<6)
 {
  alert("Please enter the Pincode properly");
  document.js1.pincode.focus();
  return false;
 }
}
</script>
```

આકૃતિ 2.22 : નામ અને પિનકોડની યથાર્થતા ચકાસવા માટેની જાવાસ્ક્રિપ્ટ

સ્ક્રિપ્ટમાં જોઈ શકાય છે તે મુજબ, y નામનો અન્ય એક ચલ ઘોષિત કરવામાં આવે છે. y ચલમાં પિનકોડ ફિલ્ડની કિંમત સમાવવામાં આવી છે. if શરતમાં, isNaN() નામના વિધેયનો ઉપયોગ કર્યો છે. આ વિધેયને સમજીએ.

isNaN()

આંકડાકીય કિંમતો સાથે કાર્ય કરવા માટે જાવાસ્ક્રિપ્ટ આંતરપ્રસ્થાપિત વિધેયોનો ઉપયોગ કરે છે. isNaN() એ સામાન્ય રીતે સૌથી વધુ ઉપયોગમાં લેવાતું આંકડાકીય વિધેય છે. NaN એટલે "Not a Number" આપેલ કિંમત આંકડાકીય પ્રકારની ન હોય, તો તે true પરત કરે છે તથા કિંમત આંકડાકીય હોય, તો તે false પરત કરે છે. ઉદાહરણ તરીકે,

isNaN(123) વિધેય false પરત કરશે, કારણકે "123" એ સંખ્યા છે.

isNaN("hello") વિધેય true પરત કરશે, કારણકે "hello" એ સંખ્યા નથી.

આમ, if વિધાનનો ઉપયોગ કરી તપાસી શકાયું છે કે y ખાલી, અંક ન હોય તેવી અથવા છ અંકોની ન હોય તેવી કિંમત હોવી જોઈએ. જો આમાંથી એક પણ શરતનો ભંગ થશે, તો ઉપયોગકર્તા સમક્ષ એલર્ટ સંદેશ દર્શાવવામાં આવશે.

બ્રાઉઝરમાં ફાઇલ ખોલો. આકૃતિ 2.23માં દર્શાવ્યા મુજબનું પરિણામ જોવા મળશે. name ફિલ્ડમાં નામ ઉમેરો. (name ફિલ્ડને ખાલી છોડશો નહીં, નહીં તો તેના માટેનો એલર્ટ સંદેશ દર્શાવવામાં આવશે.)

આકૃતિ 2.23 : બ્રાઉઝરમાં પરિણામ

Pincode ફિલ્ડને ખાલી રાખી Submit બટન પર ક્લિક કરો. આકૃતિ 2.24માં દર્શાવ્યા મુજબનો એલર્ટ સંદેશ દર્શાવવામાં આવશે.

આકૃતિ 2.24 : ખાલી ફિલ્ડ માટેનો એલર્ટ સંદેશ

હવે, Pincode ફિલ્ડમાં થોડા અક્ષર કે છ અંકોથી વધુ કોઈ આંકડાકીય સંખ્યા ઉમેરો. આમ કરવાથી આકૃતિ 2.24માં દર્શાવેલ આ જ પ્રકારનો એલર્ટ સંદેશ દર્શાવવામાં આવશે.

અહીં, Pincode ફિલ્ડમાં ત્રણે યથાર્થતા માટે આપણે એકસમાન એલર્ટ સંદેશ દર્શાવીએ છીએ. જો યથાર્થતા પ્રમાણે યોગ્ય એલર્ટ સંદેશ દર્શાવવા ઇચ્છતા હોઈએ, તો સ્ક્રિપ્ટમાં સુધારા કરવાની જરૂર પડશે. ઉદાહરણમાં ત્રણે યથાર્થતા માટે એક જ if શરતનો ઉપયોગ કરવામાં આવ્યો છે, જે એલર્ટ સંદેશ દર્શાવે છે. યથાર્થતા મુજબ એલર્ટ સંદેશ દર્શાવવા માટે, ત્રણ if શરતો ઉમેરવી પડે. ત્રણ if શરતો સાથેની સ્ક્રિપ્ટ આકૃતિ 2.25માં દર્શાવી છે.

```
<script>
function validateForm()
{
 var x=document.js1.name.value;
 var y=document.js1.pincode.value;
 if (x==null || x=="")
 {
  alert("Please enter the Name");
  document.js1.name.focus();
  return false;
 }
if (y=="")
 {
  alert("please enter the pincode properly");
  document.js1.pincode.focus();
  return false;
 }
if(isNaN(y))
{
 alert("please enter a number");
 document.js1.pincode.focus();
 return false;
}
if(y.length>6 || y.length<6)
{
 alert("please enter a six digits");
 document.js1.pincode.focus();
 return false;
}
}
</script>
```

આકૃતિ 2.25 : યોગ્ય સંદેશ દર્શાવવા માટેનો જાવાસ્ક્રિપ્ટ કોડ

ફાઇલને બ્રાઉઝરમાં ખોલો. Pincode ફિલ્ડમાં કેટલાક અક્ષરો ઉમેરીને સબમિટ બટન પર ક્લિક કરો. આકૃતિ 2.26માં દર્શાવ્યા મુજબ એલર્ટ સંદેશ રજૂ કરવામાં આવશે.

આકૃતિ 2.26 : આંકડાકીય લખાણ તપાસવા માટેનો એલર્ટ સંદેશ

Pincode ફિલ્ડમાં 6 અંકોથી નાની સંખ્યા ઉમેરો અને સબમિટ બટન પર ક્લિક કરો. આકૃતિ 2.27માં દર્શાવ્યા મુજબનો એલર્ટ સંદેશ રજૂ કરવામાં આવશે.

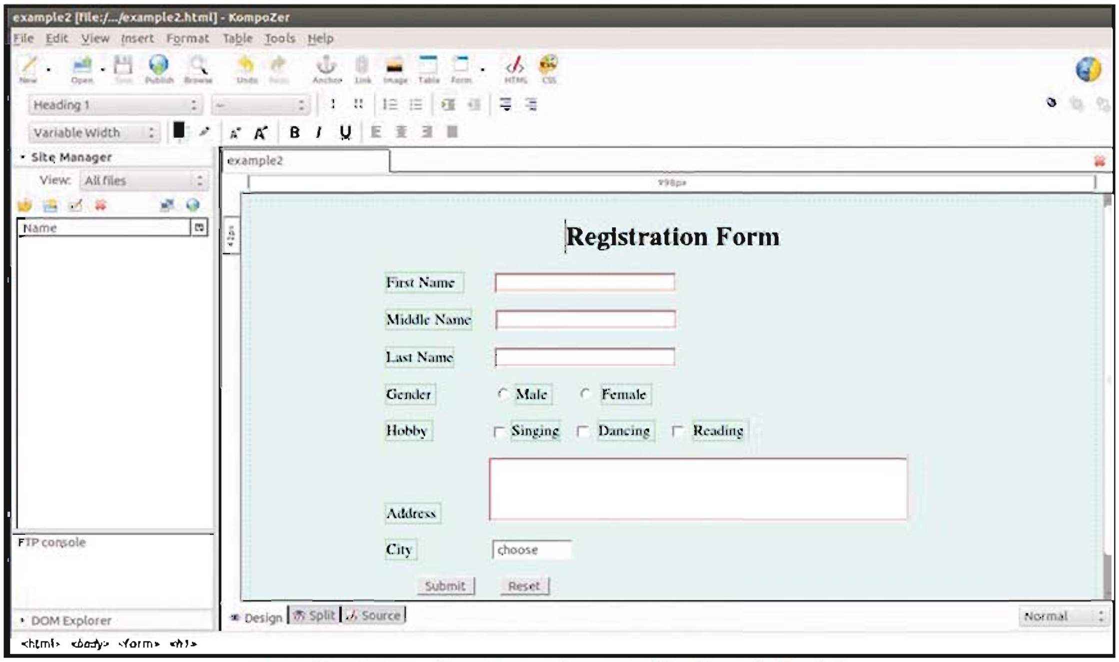
આકૃતિ 2.27 : લંબાઈ તપાસવા માટેનો એલર્ટ સંદેશ

વેબપેજમાં સંવાદન ઉમેરવા માટે જાવાસ્ક્રિપ્ટનો ઉપયોગ આપણે શીખ્યા. હવે, પ્રકરણ-1માં બનાવવામાં આવેલા નોંધણી માટેના ફોર્મ (Registration Form)માં જાવાસ્ક્રિપ્ટ ઉમેરીએ.

યથાર્થતા લાગુ પાડવા માટે ફોર્મમાં નીચેની યાદીમાં જણાવેલ સરળ ફેરફાર કરીશું :

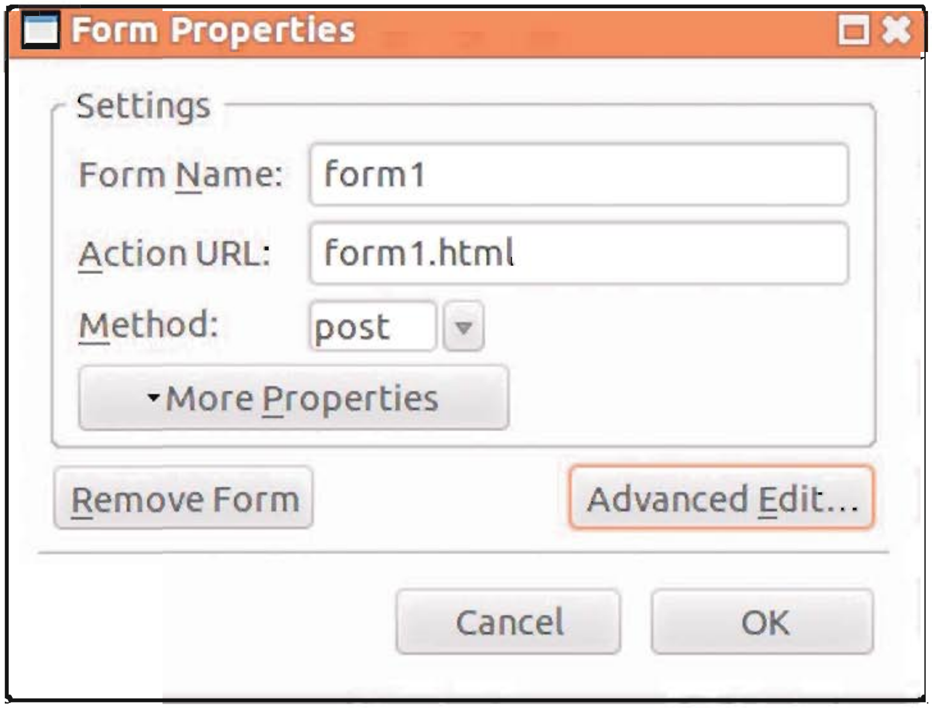
- Gender અને Hobby ફિલ્ડ શરૂઆતમાં પસંદ થયેલાં નહીં હોય.
- Address ફિલ્ડમાં શરૂઆતની કોઈ કિંમત નહીં હોય.
- City ફિલ્ડમાં "choose the city" વિકલ્પ હશે, જેની કિંમત -1 છે. અન્ય તમામ શહેરો, જેવાં કે Ahmedabad, Rajkot અને Suratને અનુક્રમે 1, 2 અને 3 કિંમતો હશે.

ઉપર્યુક્ત ફેરફારો કરવા માટે, ફોર્મ ખોલી તેમાં સંબંધિત ફિલ્ડ પસંદ કરો. **Form → Form Field** વિકલ્પ પસંદ કરો. આમ કરવાથી જે-તે ફિલ્ડમાં ફેરફાર કરવા માટે તેના ગુણધર્મ દર્શાવવામાં આવશે. ફેરફાર કર્યા બાદ ફોર્મ આકૃતિ 2.28માં આપેલ ફોર્મ જેવું દેખાશે.



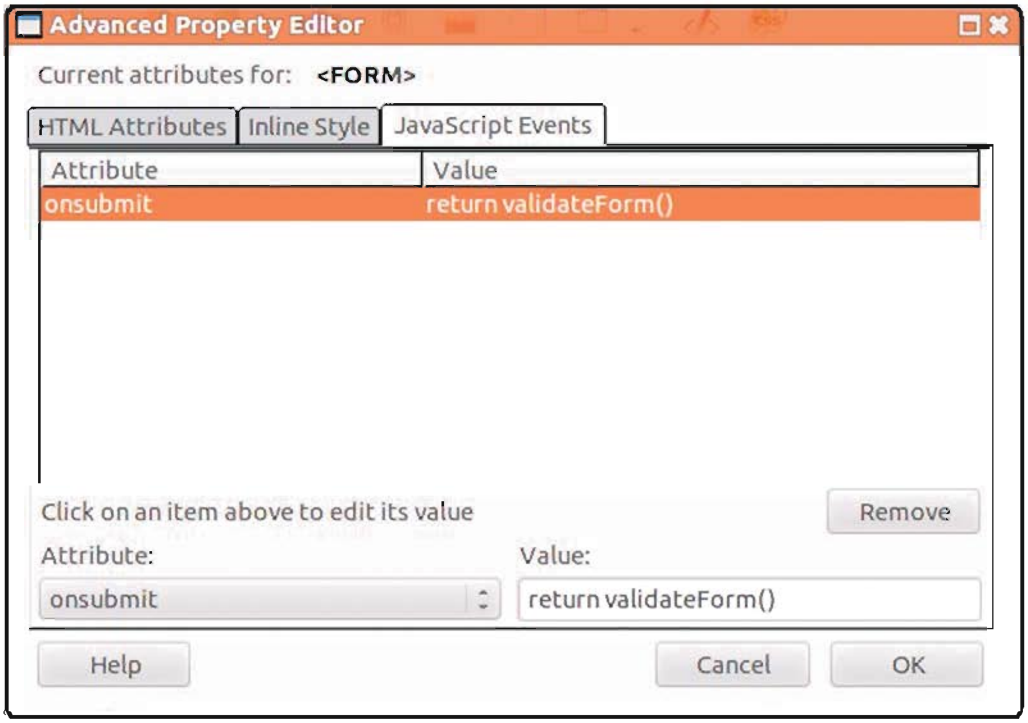
આકૃતિ 2.28 : ફેરફાર કર્યા બાદ નોંધણી માટેનું ફોર્મ

- **Form → Define Form** વિકલ્પ પસંદ કરો. વૈકલ્પિક રીતે, ફૉર્મમાં રાઇટ ક્લિક કરીને Properties વિકલ્પ પસંદ કરી શકાય. તે આકૃતિ 2.29માં દર્શાવેલ Form Properties ડાયલૉગબૉક્સ રજૂ કરશે.



આકૃતિ 2.29 : Form Properties ડાયલૉગબૉક્સ

- Advanced Edit બટન પર ક્લિક કરો. આમ કરવાથી આકૃતિ 2.30માં આપેલ Advanced Property Editor ડાયલૉગબૉક્સ ખૂલશે. JavaScript Events વિભાગ પસંદ કરો. Attribute વિકલ્પમાં "onsubmit" અને Value વિકલ્પમાં "return validateForm() ઉમેરો. OK બટન પર ક્લિક કરો.



આકૃતિ 2.30 : Advanced Property Editor ડાયલૉગબૉક્સ

- આકૃતિ 2.31માં દર્શાવેલ સ્ક્રિપ્ટ HTMLના <head> વિભાગમાં ઉમેરો.

```
<script>
function validateForm()
{
var x=document.form1.firstname.value;
var y=document.form1.middlename.value;
var z=document.form1.lname.value;
var r=document.form1.address.value;
if (x==null || x=="")
{
alert("Please enter the first name properly");
document.form1.firstname.focus();
return false;
}

if (y==""||y==null)
{
alert("Please enter the middle name properly");
document.form1.middlename.focus();
return false;
}
if(z==""||z==null)
{
alert("Please enter a last name properly");
document.form1.lname.focus();
return false;
}
if(r=="")
{
alert("Please enter address");
document.form1.address.focus();
return false;
}
if (( document.form1.gender[0].checked == false ) && ( document.form1.gender[1].checked ==
false ) )
{
alert ( "Please choose your Gender: Male or Female" );
document.form1.gender[0].focus();
return false;
}
if ((document.form1.hobby[0].checked == false) && (document.form1.hobby[1].checked ==
false ) && (document.form1.hobby[2].checked == false))
{
alert ( "Please choose a hobby" );
document.form1.hobby[0].focus();
return false;
}
if( document.form1.city.value == "-1" )
{
alert( "Please provide your city!" );
document.form1.city.focus();
return false;
}
}
</script>
```

આકૃતિ 2.31 : જાવાસ્ક્રિપ્ટની યથાર્થતા

સ્ક્રિપ્ટમાં દર્શાવ્યા મુજબ x, y, z અને r એમ ચાર ચલ ઘોષિત કરવામાં આવ્યા છે.

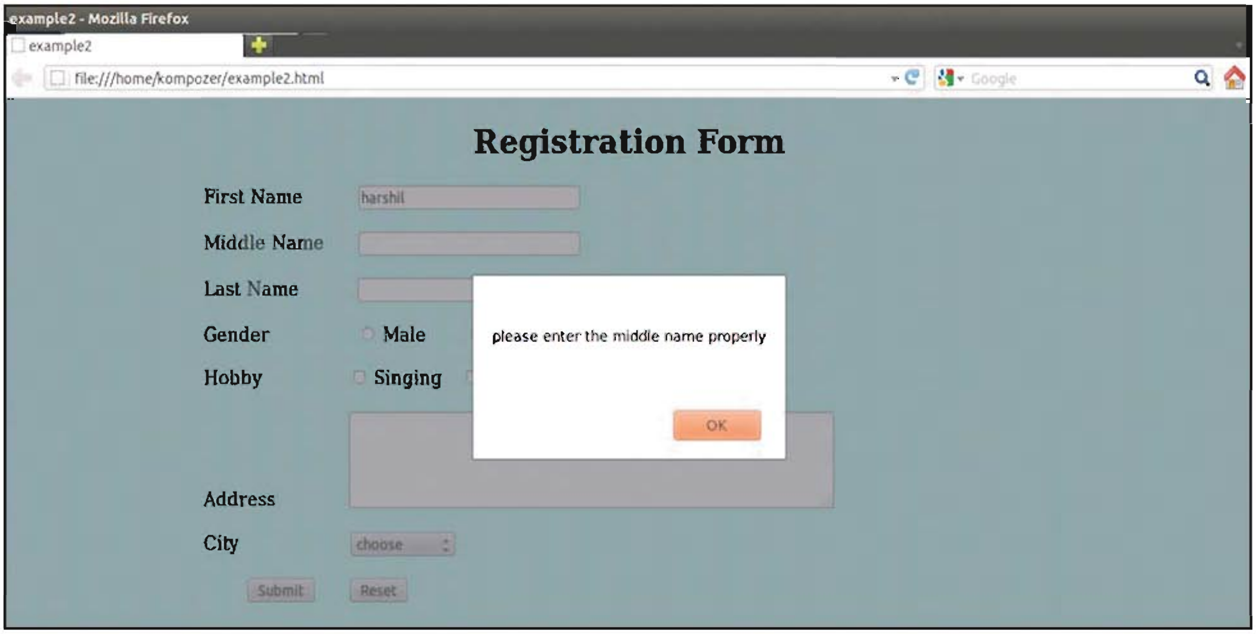
var x=document.form1.firstname.value વિધાનમાં form1 એ ફૉર્મના નામનો સંદર્ભ આપે છે, firstname પદ ઘટકને આપેલ નામનો સંદર્ભ આપે છે. ("first name" ઇનપુટ ફિલ્ડને આપવામાં આવેલ નામ). આમ,

- x ચલમાં first nameની કિંમતનો સંગ્રહ કરવામાં આવશે.
- y ચલમાં middle nameની કિંમતનો સંગ્રહ કરવામાં આવશે.
- z ચલમાં last nameની કિંમતનો સંગ્રહ કરવામાં આવશે.
- r ચલમાં addressની કિંમતનો સંગ્રહ કરવામાં આવશે.

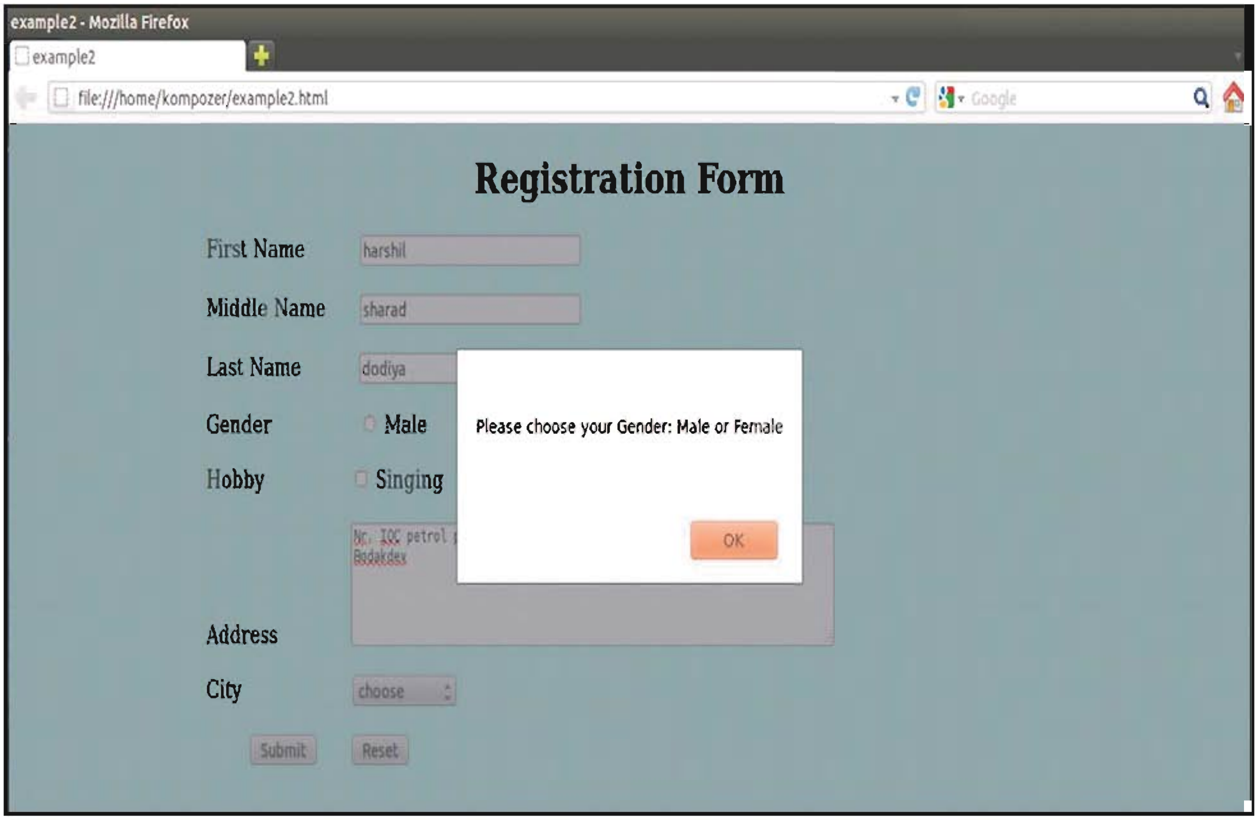
સ્ક્રિપ્ટમાં, if શરતનો ઉપયોગ કરી,

- first name ખાલી છે કે નહીં તે ચલ x દ્વારા ચકાસવામાં આવશે.
- middle name ખાલી છે કે નહીં તે ચલ y દ્વારા ચકાસવામાં આવશે.
- last name ખાલી છે કે નહીં તે ચલ z દ્વારા ચકાસવામાં આવશે.
- address ખાલી છે કે નહીં તે ચલ r દ્વારા ચકાસવામાં આવશે.

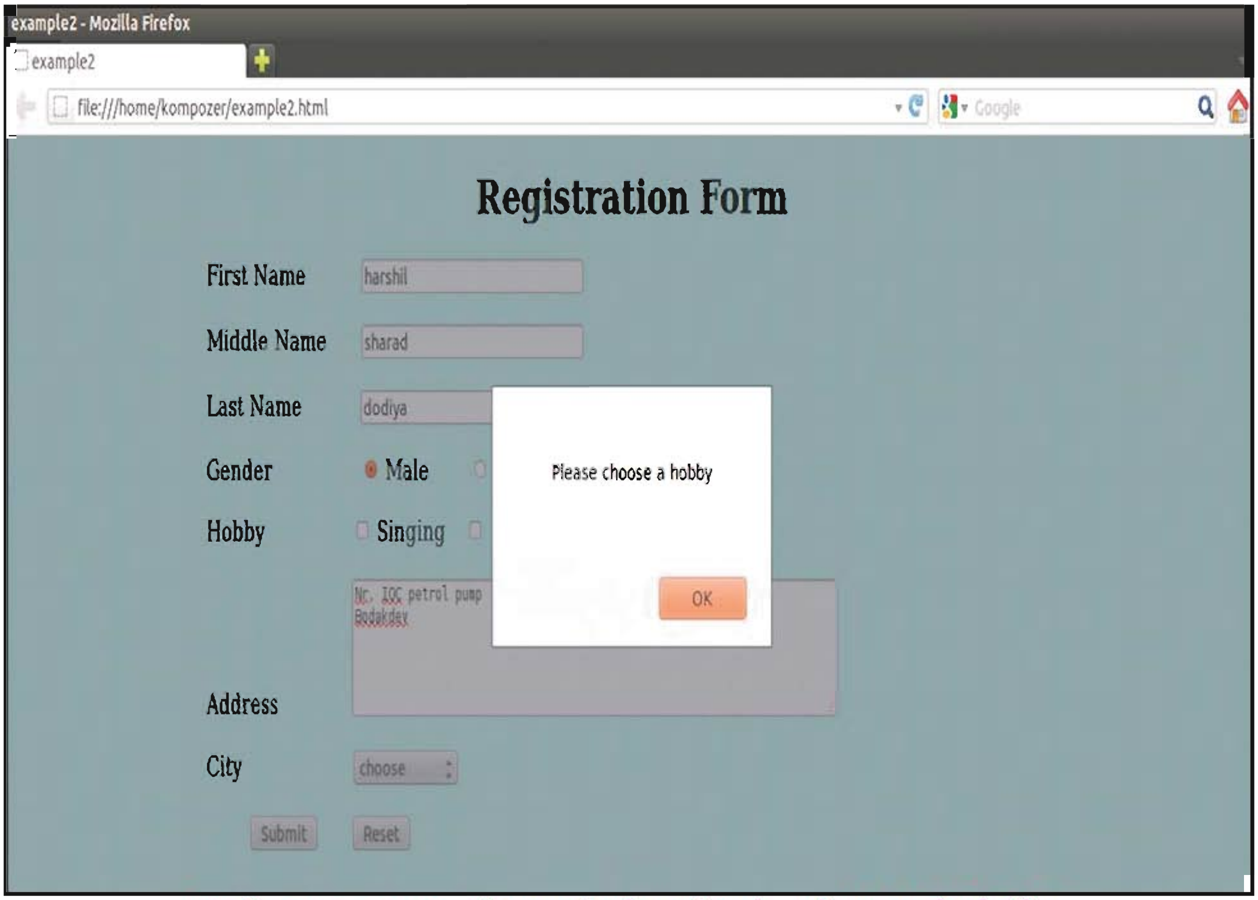
જાતિ (gender) ફિલ્ડ માટેનાં રેડિયો-બટનને "gender" નામ દ્વારા જૂથમાં મૂકવામાં આવ્યાં છે. gender બે ઘટક ધરાવતો ઍરે છે. document.form1.gender[0].checkedનો ઉપયોગ કરી gender ઍરેના પ્રથમ ઘટકની ચકાસણી if વિધાન દ્વારા કરવામાં આવે છે તથા document.form1.gender[1].checkedનો ઉપયોગ કરી if વિધાન દ્વારા ઍરેના બીજા ઘટકની ચકાસણી કરવામાં આવે છે. જો આ બન્નેનું પરિણામ false મળે તો ઉપયોગકર્તાએ કોઈ વિકલ્પ પસંદ કરેલ નથી. આથી એલર્ટ સંદેશ દર્શાવવામાં આવશે અને રેડિયો-બટન પર નિયંત્રણ (focus) લઈ જવામાં આવશે.
આવી જ રીતે, શોખ ફિલ્ડ માટેનાં ચેકબૉક્સને hobby નામ દ્વારા જૂથમાં મૂકવામાં આવ્યાં છે. hobby ત્રણ ઘટક ધરાવતો ઍરે છે. જેવી રીતે if શરત દ્વારા gender ફિલ્ડની કિંમત તપાસવામાં આવી, તેવી જ રીતે hobby ફિલ્ડની કિંમત ચકાસવામાં આવશે. અને જો તમામ ત્રણ ઘટક false દર્શાવે, તો ઉપયોગકર્તાએ કોઈ વિકલ્પ પસંદ કર્યો નથી. તેથી, ઉપયોગકર્તા સમક્ષ શોખ પસંદ કરવા અંગેનો એલર્ટ સંદેશ દર્શાવવામાં આવશે અને ચેકબૉક્સ પર નિયંત્રણ (focus) લઈ જવામાં આવશે.
શહેર માટેના ફિલ્ડ માટે -1 કિંમત તપાસવામાં આવશે. (**નોંધ :** -1 કિંમત "choose the city" માટે રાખવામાં આવી છે.), તે દર્શાવે છે કે ઉપયોગકર્તા દ્વારા એક પણ કિંમત પસંદ કરવામાં આવી નથી. ત્યારે ઉપયોગકર્તા સમક્ષ એલર્ટ સંદેશ દર્શાવવામાં આવશે. ફિલ્ડ ખાલી છોડવાથી મળતા કેટલાક એલર્ટ સંદેશાઓ આકૃતિ 2.32થી 2.35માં દર્શાવ્યા છે.



આકૃતિ 2.32 : middle name ફિલ્ડ ખાલી છોડવામાં આવે ત્યારે મળતા એલર્ટ સંદેશ

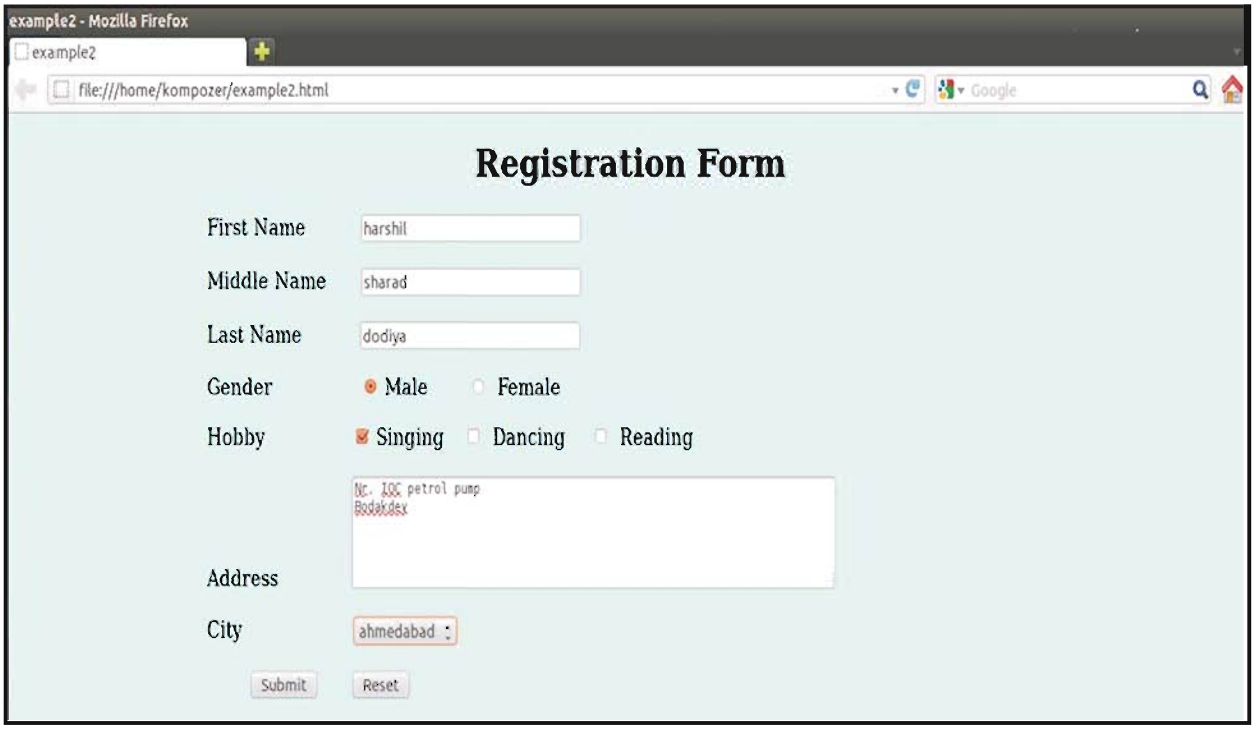


આકૃતિ 2.33 : gender ફિલ્ડ ખાલી છોડવામાં આવે ત્યારે મળતા એલર્ટ સંદેશ



આકૃતિ 2.34 : hobby ફિલ્ડ ખાલી છોડવામાં આવે ત્યારે મળતા એલર્ટ સંદેશ

તમામ ફિલ્ડની કિંમતો ઉમેર્યા પછી ફૉર્મનો અંતિમ દેખાવ આકૃતિ 2.35માં દર્શાવ્યો છે.



આકૃતિ 2.35 : અંતિમ ફૉર્મ

ફૉર્મની યથાર્થતા માટે તથા વેબપેજને વધુ સંવાદિત બનાવવા માટે આપણે જાવાસ્ક્રિપ્ટનો અભ્યાસ કર્યો.

સારાંશ

આ પ્રકરણમાં આપણે કેસ્કેડિંગ સ્ટાઇલશીટ અને જાવાસ્ક્રિપ્ટ વિશે અભ્યાસ કર્યો. વેબસાઇટના દર્શનીય ઘટકો માટેની શૈલી સ્પષ્ટ કરવા CSSનો ઉપયોગ કરવામાં આવે છે. તેની મદદથી દસ્તાવેજની માહિતીને તેના દેખાવ સંબંધી વિગતોથી અલગ રાખવામાં આવે છે. આપણે CSSના ફાયદા અને ગેરફાયદા વિશે ચર્ચા કરી. વેબપેજમાં તાર્કિક પાસું ઉમેરવા માટે જાવાસ્ક્રિપ્ટનો ઉપયોગ કરવામાં આવે છે. સ્ક્રિપ્ટિંગ ભાષા એક સરળ, હળવી પ્રોગ્રામિંગ ભાષા છે, જેમાં અદ્યતન પ્રોગ્રામિંગ કાર્યનો સમાવેશ કરવામાં આવતો નથી. વેબપેજની રૂપરેખા સુધારવા તથા ફૉર્મની યથાર્થતા ચકાસવા માટે તેનો ઉપયોગ કરવામાં આવે છે. તે HTML પાનાંમાં સંવાદન ઉમેરે છે તથા તેને પ્રત્યક્ષ રીતે HTML કોડમાં ઉમેરવામાં આવે છે. HTML પાનામાં જાવાસ્ક્રિપ્ટ ઉમેરવા માટે <script>... </script> ટૅગનો ઉપયોગ કરવામાં આવે છે.

સ્વાધ્યાય

1. કેસ્કેડિંગ સ્ટાઇલશીટનો ઉદ્દેશ જણાવો.
2. વેબપેજમાં શૈલી અને તેમાં આવેલ વિગતોને શા માટે અલગ રાખવી જોઈએ ?
3. CSSની વાક્યરચના સમજાવો.
4. CSSના ફાયદાઓ જણાવો.
5. CSSના ગેરફાયદાઓ જણાવો.
6. HTML પાનામાં જાવાસ્ક્રિપ્ટનો ઉપયોગ શા માટે કરવામાં આવે છે ?

7. HTML પાનામાં જાવાસ્ક્રિપ્ટને કેવી રીતે ઓળખી શકાય ? ઉદાહરણ આપો.

8. આપેલ વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરો :

(1) નીચેનામાંથી કોનો ઉપયોગ વેબસાઇટમાં દર્શનીય ઘટકો માટેની શૈલી સ્પષ્ટ કરવા માટે થાય છે ?

(a) Cascading Style Sheets (b) Webpage

(c) Form (d) Animation

(2) નીચેનામાંથી કોને CSSની વાક્યરચનામાં એક વિશિષ્ટ નિશાની તરીકે ઓળખવામાં આવે છે ?

(a) નિયમ (Rules) (b) પસંદગીકાર (Selector)

(c) ઘોષણા (Declaration) (d) નિવેશ (Input)

(3) નીચેનામાંથી CSS નિયમના મુખ્ય બે વિભાગ ક્યા છે ?

(a) Selector, declaration (b) Select, declaration

(c) Selector, declare (d) Selection, declaration

(4) જેના પર શૈલી લાગુ પાડવાની છે, તેવા HTMLના ઘટકને શું કહે છે ?

(a) declaration (b) selector

(c) select (d) declare

(5) નીચેનામાંથી CSSની વાક્યરચના કઈ છે ?

(a) select {property : value} (b) selector {value : property}

(c) selector {property : value} (d) selection {property : value}

(6) નીચેનામાંથી કઈ સંસ્થાએ જાવાસ્ક્રિપ્ટનો વિકાસ કર્યો ?

(a) Yahoo (b) Google

(c) Wikipedia (d) Netscape

(7) નીચેનામાંથી કઈ સ્ક્રિપ્ટિંગ ભાષાનો ઉપયોગ વેબપેજમાં પ્રોગ્રામિંગ ઉમેરવા માટે કરવામાં આવે છે ?

(a) Action script (b) JavaScript

(c) HTML (d) CSS

(8) નીચેનામાંથી કઈ સરળ, હળવી પ્રોગ્રામિંગ ભાષાને સ્ક્રિપ્ટિંગ ભાષા તરીકે ઓળખવામાં આવે છે કે જેમાં અદ્યતન પ્રોગ્રામિંગ-કાર્યોનો સમાવેશ કરવામાં આવ્યો ન હોય ?

(a) JavaScript (b) HTML

(c) C (d) Java

(9) નીચેનામાંથી ક્યા ટેગનો ઉપયોગ HTML પાનાંમાં જાવાસ્ક્રિપ્ટનો કોડ ઉમેરવા માટે કરવામાં આવે છે ?

(a) <script>... <script> (b) <script>... </script>

(c) <script>... <//script> (d) </script>... </script>

(10) નીચેનામાંથી કઈ નિશાની દ્વારા જાવાસ્ક્રિપ્ટ બ્લૉકની શરૂઆત અને અંત દર્શાવવામાં આવે છે ?

(a) અર્ધવિરામ (semicolon) (b) ચોરસ કૌંસ (square bracket)

(c) છગડિયા કૌંસ (curly bracket) (d) સાદા કૌંસ (round bracket)

(11) નિશ્ચિત કાર્ય કરવા માટે પુનઃઉપયોગમાં લઈ શકાય તે પ્રકારના ઘટકને શું કહે છે ?

(a) Array (b) Code

(c) Program (d) Function

(12) નીચેનામાંથી ક્યા વિધાનનો ઉપયોગ વિધેયમાં કિંમત પરત કરવા માટે થાય છે ?

(a) return (b) function

(c) select (d) send

(13) ઉપયોગકર્તા અને વેબપેજ વચ્ચેના સંવાદનથી શું બને છે ?

(a) વિધેય (Function) (b) પ્રતિભાવ (Response)

(c) ઘટના (Event) (d) કિંમત (Value)

(14) નીચેનામાંથી કોને ઘટના કહી શકાય નહીં ?

(a) Abort (b) Mouseover

(c) Set (d) Load

(15) વિગતોનો સંગ્રહ કરવા માટેના સંગ્રાહકને શું કહે છે ?

(a) ચલ (Variable) (b) પૂર્ણાંક (Integer)

(c) ઘટના (Event) (d) ઘટના-સંચાલક (Event handler)

(16) BOM એટલે શું ?

(a) Browser of Model (b) Browser Object Model

(c) Browser Object Modelling (d) Browse Object Model

(17) બ્રાઉઝર ઑબ્જેક્ટ મૉડેલમાં સૌથી ઉપરના સ્તરે રહેલ ઑબ્જેક્ટ કયો છે ?

(a) Window (b) Document

(c) Page (d) Location

(18) NaN એટલે શું ?

(a) Not a Numeric (b) Not a Number

(c) Not a Noun (d) Not an Numeric

પ્રાયોગિક સ્વાધ્યાય

1. નીચે આપેલ નિયમોના ઉપયોગ દ્વારા H1 ટેગ માટે CSSની રચના કરી તેને લખાણ પર લાગુ કરો :

- Font : Times New Roman
- Color : Red
- Case : Lowercase
- Fontstyle : Italic
- Alignment : Centre
- Text decoration : Underline
- Background color : Light Grey
- Border : Dotted

2. આપેલ ઘટકો સાથેનું ફૉર્મ બનાવો : name, email address, phone number અને submit button. તેમાં નીચે દર્શાવ્યા મુજબ યથાર્થતા ઉમેરો :

- ફિલ્ડ ખાલી ન હોવાં જોઈએ.
- ફોન માટેના ફિલ્ડમાં માત્ર અંક માન્ય છે.
- ફોનનંબર દસ અંકોનો હોવો જોઈએ.

3. વિદ્યાર્થીની અંગત વિગતો ઉમેરવા માટેના પ્રકરણ-1માં બનાવવામાં આવેલ ફૉર્મમાં યથાર્થતા ઉમેરો.

4. પ્રતિસાદ (feedback) માટેના પ્રકરણ-1માં બનાવવામાં આવેલ ફૉર્મમાં યથાર્થતા ઉમેરો.

કમ્પોઝરનો ઉપયોગ કરી સરળ વેબસાઇટની રચના 3

આધુનિક યુગમાં કોઈ પણ વ્યવસાયમાં વેબસાઇટ ખૂબ અગત્યનો ભાગ ભજવે છે. તે વિશ્વકક્ષાએ વ્યવસાયની રજૂઆતમાં મદદરૂપ બને છે. વ્યવસાયના વિકાસમાં, ઉત્પાદનના વેચાણમાં અને ગ્રાહકોને બહોળી સંખ્યામાં આકર્ષવામાં તે મદદરૂપ બને છે. તેથી જ, વેબસાઇટની રચના શ્રેષ્ઠ તકનિક દ્વારા કરવામાં આવી હોય તે જરૂરી છે, જેથી સંસ્થા મહત્તમ સંખ્યામાં મુલાકાતીઓ મેળવી શકે અને ઉચ્ચતમ નફો રળી શકે.

વેબસાઇટ એ વિશિષ્ટ હેતુ માટે પરસ્પર જોડાયેલાં વેબપેજનો સમૂહ છે. સંવાદિત (interactive), ઉપયોગકર્તા માટે સરળ (user friendly) તથા ચોક્કસ (accurate) અને ઉપયોગી માહિતી ધરાવતી વેબસાઇટ બનાવવી એ પડકારજનક કાર્ય છે. વેબસાઇટની રચના એવી હોવી જોઈએ, જેથી ઉપયોગકર્તાને તે માહિતીપ્રદ લાગે તથા ઉપયોગકર્તા વારંવાર તેની મુલાકાત લે. જો સાઇટ વ્યાવસાયિક હોય, તો તે ગ્રાહકોની સંખ્યામાં વધારો કરે. વેબસાઇટની રચના કરતી વખતે પૂરતું ધ્યાન આપવામાં આવે, તો આ હેતુ સિદ્ધ થઈ શકે. આ પ્રકરણમાં વેબસાઇટનું આયોજન કરવા માટે ધ્યાનમાં રાખવાના સામાન્ય મુદ્દાઓની ચર્ચા કરવામાં આવી છે. આપણે કમ્પોઝર (KompoZer) પ્રોગ્રામની મદદથી વેબસાઇટ બનાવી તેને ઇન્ટરનેટ પર પ્રકાશિત (publish) કરીશું.

વેબસાઇટનું આયોજન (Planning for the Website)

સારી વેબસાઇટનો વિકાસ એ કોઈ પણ અન્ય યોજના જેવી જ એક યોજના છે અને તેના માટે વિસ્તૃત આયોજન જરૂરી છે. આયોજન જેટલું બહેતર હશે, વેબસાઇટની ઉપયોગિતાના સંદર્ભે સફળતાની તક પણ તેટલી જ વધુ રહેશે. બીજા શબ્દોમાં કહીએ તો વેબસાઇટ બનાવવા માટેનું ધ્યેય સિદ્ધ કરી શકાશે. સારી વેબસાઇટ બનાવવા માટે આયોજનની પ્રક્રિયામાં ધ્યાનમાં રાખવા માટેના કેટલાક અગત્યના મુદ્દા નીચે દર્શાવ્યા છે :

હેતુ (Purpose)

વેબસાઇટનો હેતુ સ્પષ્ટપણે વ્યાખ્યાયિત થયેલો હોય તે જરૂરી છે. વેબસાઇટની રચના કરતાં પહેલાં તેની વ્યાખ્યા અને ધ્યેય સ્પષ્ટ હોવાં જરૂરી છે. યોગ્ય રીતે રચના પામેલી વેબસાઇટ સંસ્થાને મદદરૂપ બને છે. વેબસાઇટની રચનાનો હેતુ વ્યક્તિઓના સમૂહને માહિતી પૂરી પાડવાનો, નવા ગ્રાહકોને આકર્ષવાનો કે ઉત્પાદનને ઑનલાઇન વેચવાનો હોઈ શકે. હેતુ નક્કી થયા પછી વેબસાઇટના વિષયવસ્તુ (content) અને રૂપરેખા (layout)ને યોગ્ય રીતે વિકસાવી શકાય.

પ્રેક્ષકગણ (Audience)

વેબસાઇટના રચનાત્મક વિભાગનું કાર્ય શરૂ કરતાં પહેલાં અપેક્ષિત ઉપયોગકર્તા અંગેની જાણકારી હોવી જરૂરી છે. વેબસાઇટમાં સામાન્ય અને વિસ્તૃત (detailed) એમ બંને પ્રકારની માહિતી હોવી જોઈએ. તેમાં વિશિષ્ટ અને સરળ માહિતીનો સમાવેશ પણ થયો હોવો જોઈએ. માહિતીના વિવિધ પ્રકાર, વિસ્તાર અને વિશેષતાના વ્યાપનો આધાર અપેક્ષિત પ્રેક્ષકોની રુચિ પર રહેલો છે. વેબસાઇટ પ્રત્યેની ઉપયોગકર્તાની અપેક્ષાઓ જાણવી જરૂરી છે, કારણકે તે જાણકારી વેબસાઇટના યોગ્ય વિષયવસ્તુ અને રૂપરેખાના વિકાસમાં મદદરૂપ બને છે અને વેબસાઇટ ઉપયોગી તથા સફળ હોવાની ખાતરી આપે છે.

એક વધુ ધ્યાન આપવા લાયક બાબત છે ઉપયોગકર્તાના ઇન્ટરનેટ જોડાણની ઝડપ. જો વેબસાઇટ પર વિશાળ ચિત્રાત્મક ફાઇલો રાખવામાં આવી હોય, તો તે ડાઉનલોડ થવામાં લાંબો સમય લે છે. કેટલીક વાર ઉપયોગકર્તા આવી વિશાળ ફાઇલોની રાહ જોવામાં અધીરો બની સાઇટ છોડી દે એવું પણ બને. માટે, વેબસાઇટમાં જરૂરી હોય તેટલા પ્રમાણમાં જ ચિત્રો અને મલ્ટિમીડિયા ફાઇલનો સમાવેશ કરવામાં આવે તે ઇચ્છનીય છે.

વિષયવસ્તુ (Content)

વેબસાઇટમાં સંપૂર્ણ અને સુસંગત માહિતી હોવી અત્યંત જરૂરી છે. અસંગત માહિતીનો વધુ ઉપયોગ ઉપયોગકર્તાને નિરાશ બનાવી શકે છે, કારણકે તે સ્પષ્ટ વિગતો પ્રાપ્ત કરી શકતો નથી. વળી, જો પૂરી પાડવામાં આપેલી માહિતી અપૂર્ણ હોય, તો ઉપયોગકર્તા વેબસાઇટ છોડી દઈ શકે છે. યોગ્ય અને સુસંગત માહિતી મૂકવાથી ઉપયોગકર્તાને સંતોષ પ્રાપ્ત થાય છે તથા વેબસાઇટમાં તેનો રસ જળવાઈ રહે છે. વેબસાઇટના વિષયવસ્તુને સામાન્ય અને વિસ્તૃત વર્ગોમાં વહેંચવું જરૂરી છે.

સામાન્ય વિષયવસ્તુ ઉપયોગકર્તા સમક્ષ સાઇટ, સંસ્થા, ઉત્પાદન અને સેવાઓ તથા અન્ય વસ્તુઓનું વિહંગાવલોકન (overview) પૂરું પાડે છે. માહિતીની સંક્ષિપ્ત સમજૂતી ઉપયોગકર્તાની શોધમાં મદદરૂપ નીવડે છે. તે યોગ્ય અને સ્પષ્ટ વિગતોનો નિર્દેશ કરે છે તથા સાઇટમાં આવેલ અન્ય ઉપલબ્ધ વસ્તુઓ માટેની જાણકારી પૂરી પાડે છે.

વિસ્તૃત વિષયવસ્તુ ઉપયોગકર્તાને ઉત્પાદન અને સેવાઓ અંગેની વિસ્તૃત માહિતી પૂરી પાડે છે. વેબસાઇટની રૂપરેખા એવી હોવી જોઈએ કે ઉપયોગકર્તા તેને ઉપયોગી એવી વિશિષ્ટ માહિતી સરળતાથી મેળવી શકે.

લખાણના ફકરા હંમેશાં યોગ્ય લંબાઈના રાખવા જોઈએ. વેબપેજ પર વિપુલ પ્રમાણમાં વિષયવસ્તુ દર્શાવવાનું હોય તો તેને લખાણના નાના નાના વિભાગોમાં વહેંચી નાખવું જોઈએ. આમ કરવાથી ઉપયોગકર્તા પોતાને જરૂરી માહિતી સરળતાથી મેળવી શકે છે. વેબસાઇટમાં નેવિગેશન (navigation) શક્ય એટલું સરળ હોવું જોઈએ. એક પાના પરથી અન્ય પાના પરનું સ્થાનાંતરણ સરળ હોવું જોઈએ, જેથી ઉપયોગકર્તાને ખબર પડે કે તે ક્યાં છે અને હોમપેજ પર કેવી રીતે પાછા ફરી શકાશે.

સામાન્ય અને વિશિષ્ટ એમ બંને પ્રકારની જરૂરી માહિતી સરળતાથી ઉપલબ્ધ બને, તો જ વેબસાઇટની રચનાનો ઉદ્દેશ્ય પાર પડે છે. સારા વિષયવસ્તુથી સભર વેબસાઇટ ઉદ્યોગ કે અન્ય સંસ્થા માટે એક કિંમતી સાધન બની રહે છે. અન્ય સાધનોની જેમ વેબસાઇટ પણ સંસ્થા માટે શ્રેષ્ઠ કાર્ય કરી શકે છે.

માધ્યમ (Medium)

વધુ ને વધુ વ્યક્તિઓ હવે સ્માર્ટફોન અને ટેબ્લેટ દ્વારા ઇન્ટરનેટનો ઉપયોગ કરી રહી છે. વેબસાઇટની રૂપરેખાનું કદ કમ્પ્યૂટર, સ્માર્ટફોન અને ટેબ્લેટ જેવાં તમામ સાધનોને અનુરૂપ હોવું જોઈએ. આ ઉપરાંત વેબસાઇટની રૂપરેખા તૈયાર કરતી વખતે એ પણ ધ્યાનમાં રાખવું જોઈએ કે તેને મોઝિલા ફાયરફોક્સ, ક્રોમ, ઓપેરા અને ઇન્ટરનેટ એક્સ્પ્લોરર જેવાં તમામ પ્રચલિત વેબબ્રાઉઝરમાં યોગ્ય રીતે દર્શાવવામાં આવે.

કમ્પોઝરનો ઉપયોગ કરી સરળ વેબસાઇટ બનાવવી (Creating a Simple Website Using KompoZer)

અગાઉના પ્રકરણમાં આપણે કમ્પોઝરની મદદથી વેબપેજની રચના તથા સંપાદન માટે જાવા સ્ક્રિપ્ટના ઉપયોગ વિશે ચર્ચા કરી હતી. વેબસાઇટ બનાવવા માટે સામાન્ય રીતે ધ્યાનમાં લેવાતા મુદ્દાઓ પણ આપણે જોયા હતા. હવે આપણે કમ્પોઝરની મદદથી એક નાની ઇ-કૉમર્સની વેબસાઇટ કેવી રીતે બનાવી શકાય, તેનો અભ્યાસ કરીશું.

આપણે "School Plaza" નામની એક સરળ ઇ-કૉમર્સ વેબસાઇટની રચના કરીએ, જે શાળામાં જતા વિદ્યાર્થીઓની જરૂરિયાતો પૂરી કરતી હોય. આ વેબસાઇટ પર પુસ્તકો, સ્ટેશનરી, શાળાની બૅગ, પાણીની બૉટલ, નાસ્તાનો ડબ્બો, ફાઇલ, ફોલ્ડર અને તેના જેવી શાળાને સંબંધિત અન્ય વસ્તુઓ ઉપલબ્ધ હશે અને તેનો ઑર્ડર પણ આપી શકાશે. આપણે એક શૉપિંગ-કાર્ટની પણ રચના કરીશું, જે ખરીદવામાં આવેલી તમામ વસ્તુઓની યાદી અને કુલ રકમ દર્શાવશે. આપણે આયોજન કરેલી વેબસાઇટનું હોમપેજ અને શૉપિંગકાર્ટનો દેખાવ આકૃતિ 3.1 અને 3.2માં દર્શાવ્યો છે.

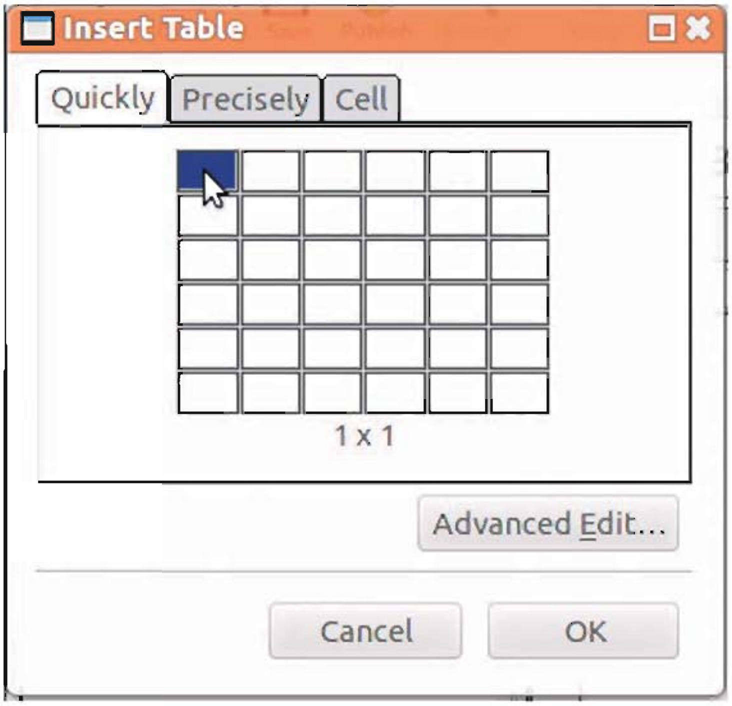
આકૃતિ 3.1 : વેબસાઈટનું હોમપેજ

આકૃતિ 3.2 : શૉપિંગકાર્ટ

આપણે ક્રમિક પગલાં દ્વારા આ વેબસાઇટની રચના કરીએ. સૌપ્રથમ આપણે વેબસાઇટનું હોમપેજ બનાવીશું. જ્યારે ઉપયોગકર્તા વેબબ્રાઉઝરના ઍડ્રેસબારમાં URL સરનામું ઉમેરે, ત્યારે વેબસાઇટના ખોલવામાં આવતાં સૌપ્રથમ પાનાને હોમપેજ કહે છે. તમામ ઉપલબ્ધ વસ્તુઓના વર્ગ (Category)ની યાદી આપણી વેબસાઇટના હોમપેજમાં દર્શાવવી જોઈએ.

કોઈ પણ વર્ગ પર ક્લિક કરવાથી તે વર્ગમાં આવેલી વસ્તુઓ દર્શાવતું અન્ય વેબપેજ રજૂ કરવામાં આવશે. વેબસાઇટ બનાવવા માટે ટેમ્પ્લેટ (Template) નામે ઓળખાતી વિવિધ રૂપરેખાઓના નમૂના ઇન્ટરનેટ પર નિઃશુલ્ક ઉપલબ્ધ છે. આ ટેમ્પ્લેટ સરળતાથી ડાઉનલોડ કરી જરૂર મુજબ ઉપયોગમાં લઈ શકાય છે.

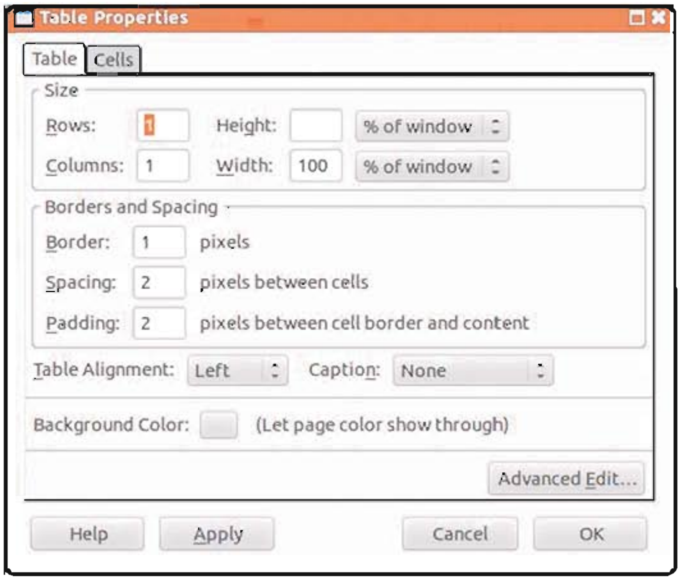
હોમપેજની રચના દ્વારા વેબસાઇટની શરૂઆત કરવા માટે કમ્પોઝર ખોલી નવું વેબપેજ બનાવો. વેબપેજમાં તમામ વિષયવસ્તુ અને ચિત્રોની ગોઠવણ કરવા માટે આકૃતિ 3.3માં દર્શાવ્યા મુજબ એક ખાનું (cell) ધરાવતું કોષ્ટક ઉમેરીશું. કોષ્ટક ઉમેરવા માટે **Insert → Table** પસંદ કરો. એક સેલ ધરાવતું કોષ્ટક પસંદ કરી OK બટન પર ક્લિક કરો.



આકૃતિ 3.3 : 1 × 1 કોષ્ટક ઉમેરવું

યોગ્ય શીર્ષક આપી વેબપેજનો સંગ્રહ કરો. શીર્ષક વેબબ્રાઉઝરના ટાઇટલબારમાં દર્શાવવામાં આવશે. અહીં, આપણે "School Plaza" શીર્ષક આપ્યું છે. "index.html" નામ આપી ફાઇલનો સંગ્રહ કરો. સામાન્ય રીતે હોમપેજને index.html નામ આપવામાં આવે છે. વેબસર્વર પર દરેક વેબસાઇટ ડિરેક્ટરીમાં બનાવવામાં આવે છે અને દરેક વેબપેજ એ વેબસર્વર પરની સ્વતંત્ર ફાઇલ છે. જ્યારે ઉપયોગકર્તા www.schoolplaza.com જેવું વેબસાઇટનું કોઈ URL બ્રાઉઝરમાં લખે છે, ત્યારે તે ખોલવામાં આવનાર ફાઇલની સ્પષ્ટતા કરતો નથી, પરંતુ વેબસર્વરને થોડું વિષયવસ્તુ દર્શાવવા માટે એક પૂર્વનિર્ધારિત ફાઇલની જરૂર પડે છે. આ પૂર્વનિર્ધારિત ફાઇલનું નામ index.html છે. આમ, જ્યારે ફાઇલના નામ વગર URL ઉમેરવામાં આવે છે, ત્યારે સર્વર પૂર્વનિર્ધારિત ફાઇલને ધ્યાનમાં લઈ તેને આપોઆપ દર્શાવે છે.

હવે, આપણે ઉમેરેલા કોષ્ટકને બેકગ્રાઉન્ડ કલર આપીશું. કોષ્ટક પસંદ કરી ડબલ ક્લિક કરો. વૈકલ્પિક રૂપે કોષ્ટક પર રાઇટ ક્લિક કરી 'Table Cell Properties' વિકલ્પ પસંદ કરી શકાય. આમ કરવાથી આકૃતિ 3.4માં દર્શાવેલું Table Properties ડાયલોગબૉક્સ ખૂલશે.

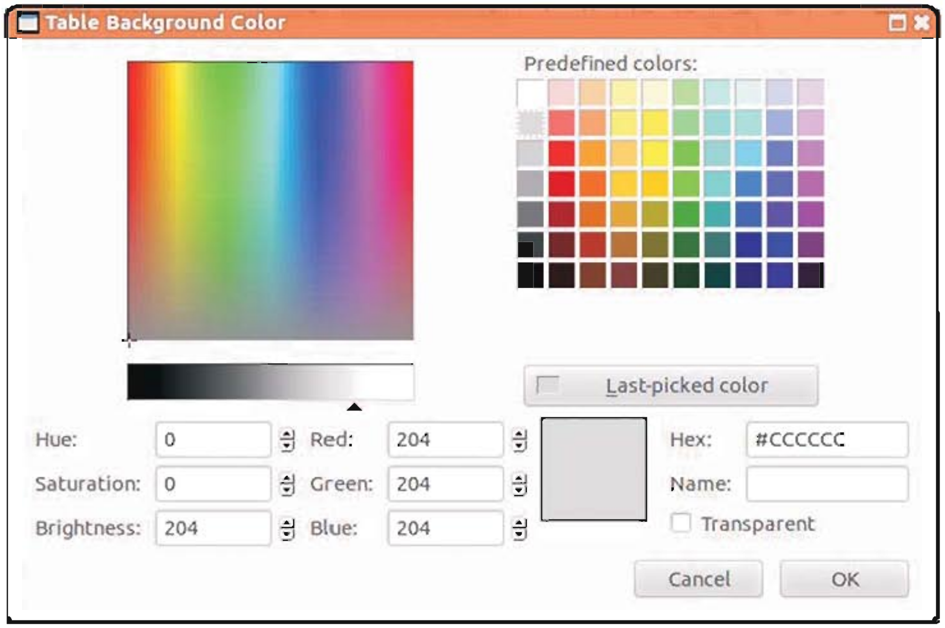


આકૃતિ 3.4 : Table Properties ડાયલોગબૉક્સ

અહીં 'Table' અને 'Cells' બે વિભાગ (Tab) જોવા મળશે, જેના દ્વારા ઉપયોગકર્તા કોષ્ટક કે સ્વતંત્ર સેલનાં વિવિધ પાસાંને નિયંત્રિત કરી શકશે. કોષ્ટકને સંબંધિત સેલ અહીં જોઈ શકાશે. Table વિભાગમાં આવેલા વિકલ્પ નીચે દર્શાવ્યા છે :

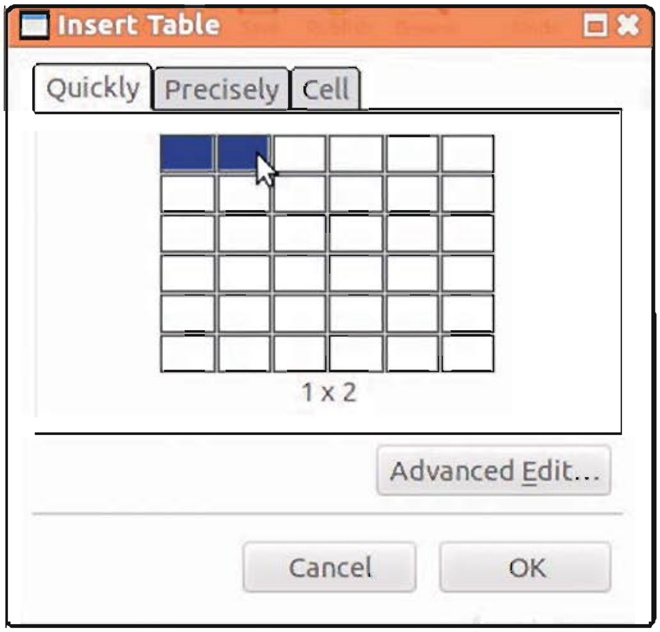
1. **Size :** તે હરોળ (row) અને સ્તંભ (column)ની સંખ્યા તથા કોષ્ટકની ઊંચાઈ અને પહોળાઈ દર્શાવે છે. ઊંચાઈ અને પહોળાઈ પિક્સેલ અથવા વિન્ડોના ટકાના સ્વરૂપે હોઈ શકે.
2. **Borders and Spacing :** કોષ્ટકને સીમારેખા (border) આપવાની હોય, તો તે માટેના વિકલ્પ અહીં દર્શાવવામાં આવે છે. સેલ વચ્ચેની જગ્યા નિશ્ચિત કરવા માટે spacingનો ઉપયોગ કરવામાં આવે છે. સેલની કિનારી અને તેમાં આવેલ લખાણ વચ્ચેની જગ્યા નિશ્ચિત કરવા paddingનો ઉપયોગ કરવામાં આવે છે.
3. **Table Alignment :** કોષ્ટકને ડાબી બાજુ, જમણી બાજુ કે વચ્ચે ગોઠવે છે.
4. **Caption :** જરૂર જણાય, તો કોષ્ટકને શીર્ષક આપી શકાય છે.
5. **Background Color :** કોષ્ટકના બૅકગ્રાઉન્ડનો કલર દર્શાવે છે.

આકૃતિ 3.5માં દર્શાવ્યા મુજબ તમારી ઇચ્છા મુજબ બૅકગ્રાઉન્ડ કલર પસંદ કરો.



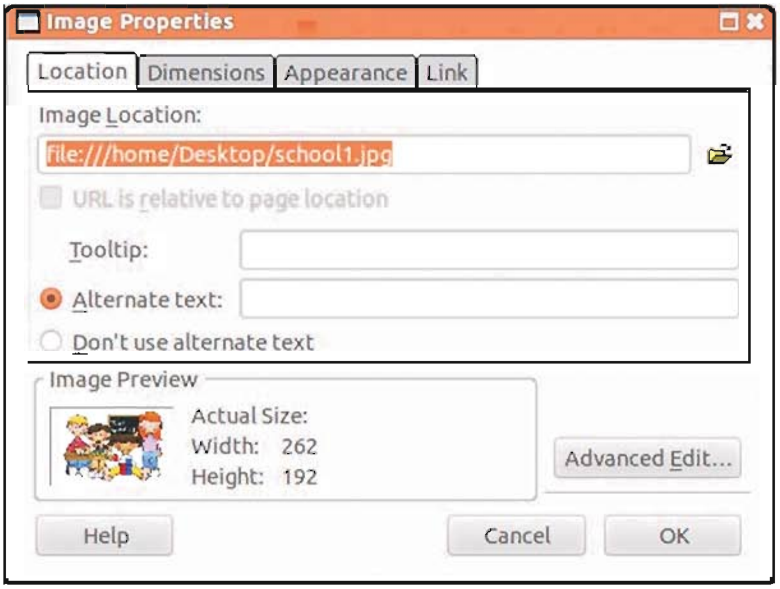
આકૃતિ 3.5 : Table Background Color ડાયલોગબૉક્સ

હવે, "School Plaza" લખાણ અને તેને સંબંધિત ચિત્ર ઉમેરવા માટે એક હરોળ અને બે સ્તંભ ધરાવતું અન્ય કોષ્ટક ઉમેરીએ. એક સ્તંભ લખાણ દર્શાવશે અને બીજામાં ચિત્ર દર્શાવવામાં આવશે. **Insert → Table** વિકલ્પ પસંદ કરો. આકૃતિ 3.6માં દર્શાવ્યા મુજબ 1 × 2 કોષ્ટક પસંદ કરો અને OK બટન પર ક્લિક કરો.



આકૃતિ 3.6 : 1×2 કોષ્ટક ઉમેરવું

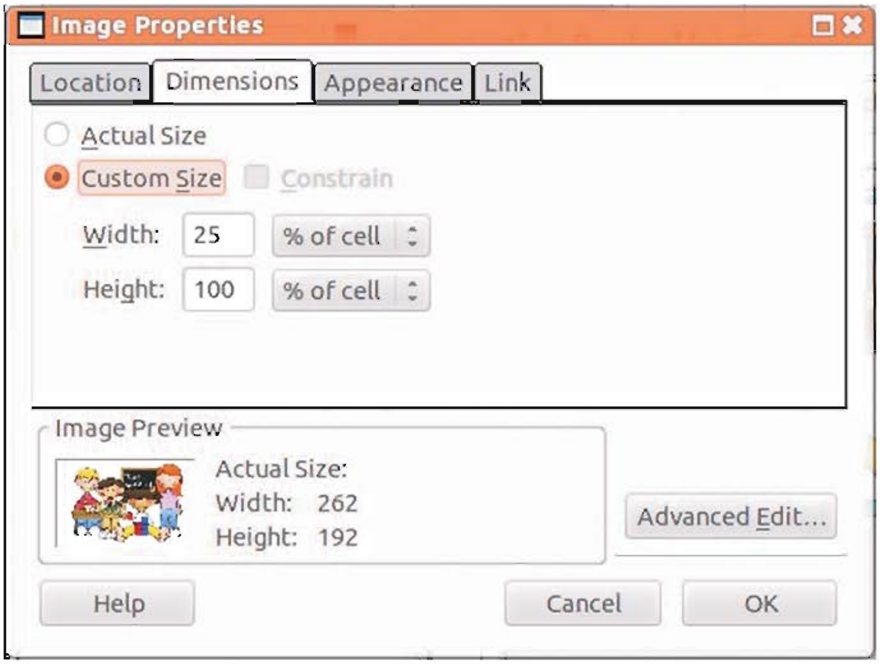
કોષ્ટકને બૅકગ્રાઉન્ડ કલર આપો. હવે કોષ્ટકના ડાબી બાજુના સેલમાં લખાણ અને જમણી બાજુના સેલમાં ચિત્ર ઉમેરીએ. ચિત્ર ઉમેરવા માટે **Insert → Image** વિકલ્પ પસંદ કરો. વૈકલ્પિક રીતે, ટૂલબાર પર આવેલા ચિત્રના આઇકન પર ક્લિક પણ કરી શકાય. આમ કરવાથી આકૃતિ 3.7માં દર્શાવેલ ડાયલોગબૉક્સ ખોલવામાં આવશે. ચિત્રનું સ્થાન શોધો. ઇનપુટ બૉક્સમાં વૈકલ્પિક લખાણ ઉમેરો અને જો વૈકલ્પિક લખાણ ઉમેરવા માંગતા ન હો, તો "Don't use alternate text" રેડિયો બટન પસંદ કરો. ચિત્રને બ્રાઉઝરમાં દર્શાવી શકાય તેમ ન હોય, ત્યારે તેના સ્થાને આ વૈકલ્પિક લખાણ દર્શાવવામાં આવશે.



આકૃતિ 3.7 : Image Properties ડાયલોગબૉક્સ

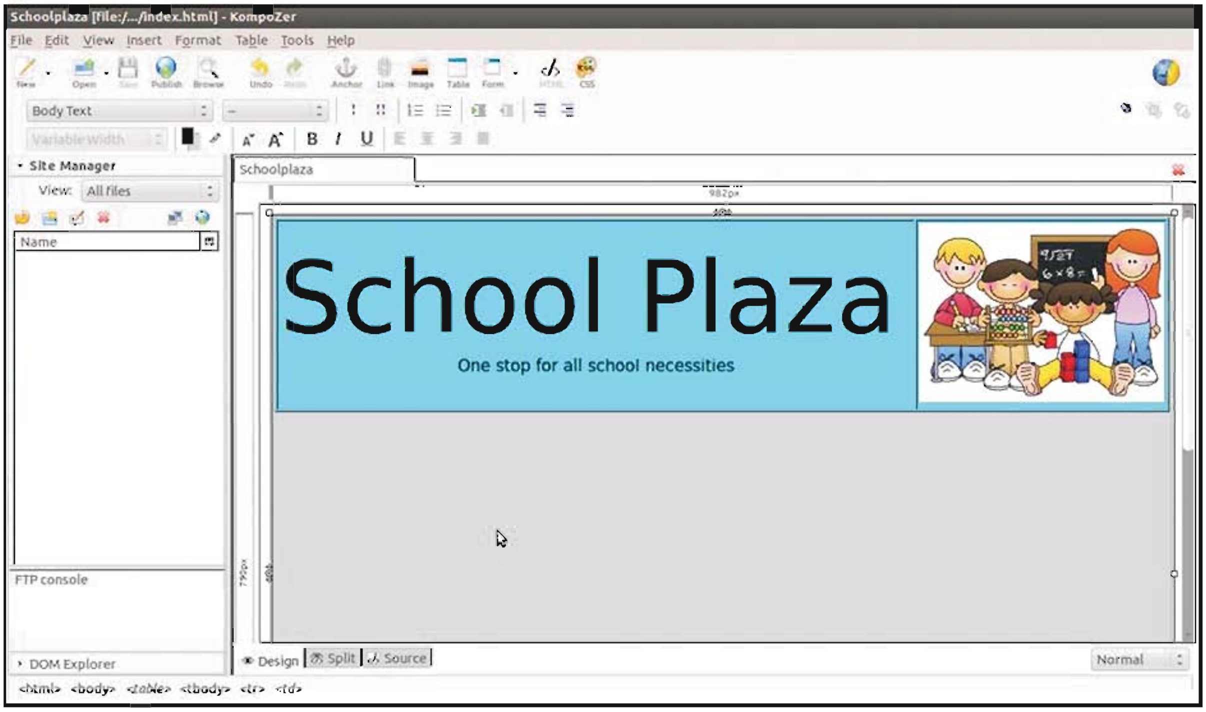
નોંધ : આપણે અહીં school1.jpg ચિત્રનો ઉપયોગ કર્યો છે. વિદ્યાર્થીઓ પોતાની પસંદગીનું ચિત્ર બનાવી શકે છે અથવા અન્ય મનપસંદ ચિત્રનો ઉપયોગ કરી શકે છે.

હવે, Dimension વિભાગ પસંદ કરી તેમાં આવેલ Custom Size વિકલ્પ પસંદ કરો. ચિત્રને આખા સેલમાં દર્શાવવાનું હોવાથી આકૃતિ 3.8માં દર્શાવ્યા મુજબ ઊંચાઈ અને પહોળાઈ 100% રાખીશું. અન્ય સેલમાં આકૃતિ 3.9માં દર્શાવ્યા મુજબ લખાણ ઉમેરો.



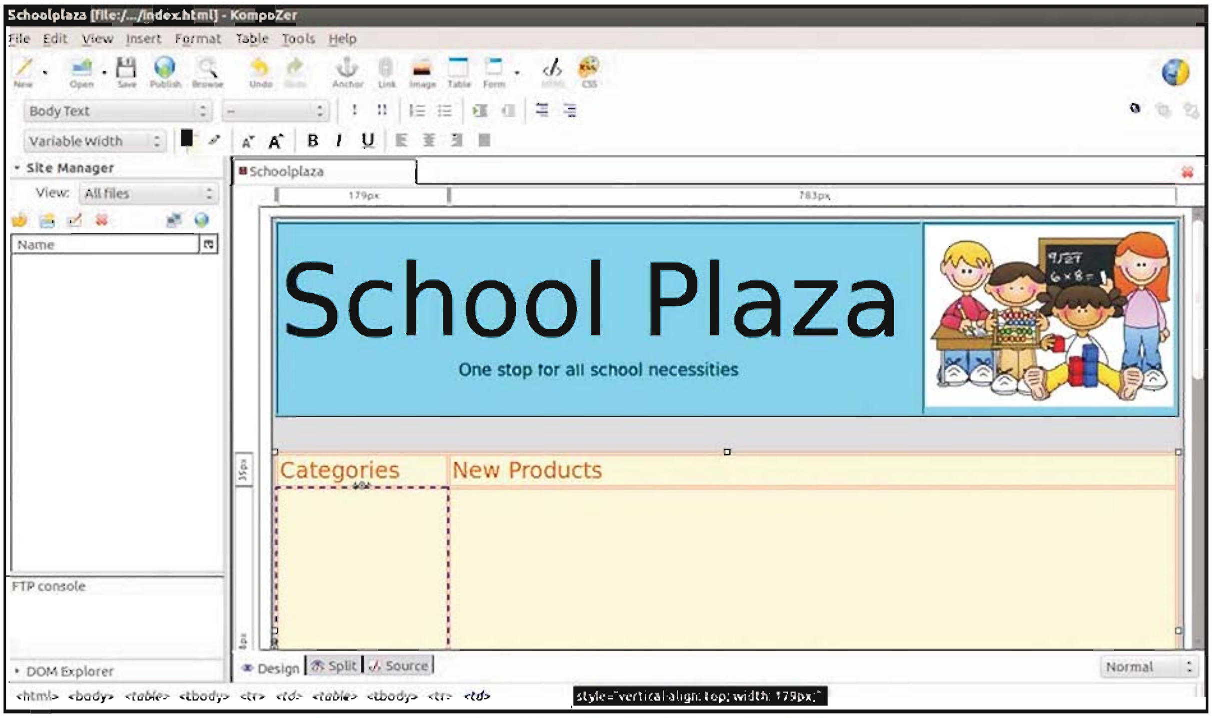
આકૃતિ 3.8 : Image Properties ડાયલોગબૉક્સમાં Dimensions વિભાગ

આકૃતિ 3.9માં ઉમેરવામાં આવેલ લખાણ અને ચિત્ર દર્શાવ્યાં છે.



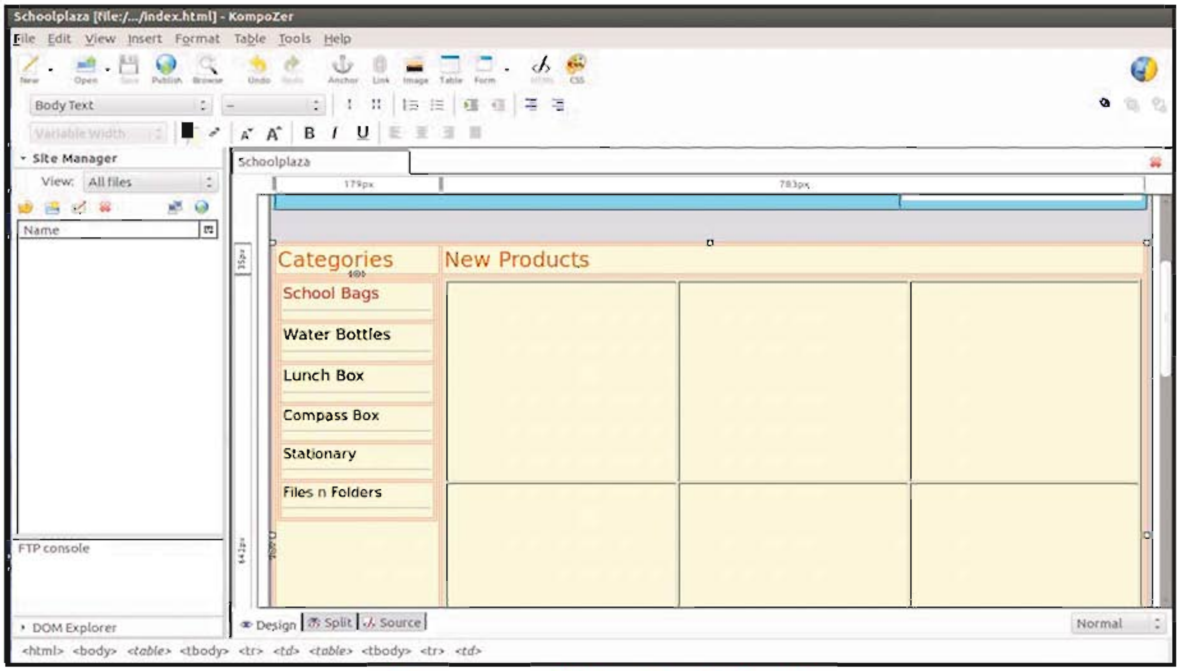
આકૃતિ 3.9 : ચિત્ર અને લખાણ સાથેનું કોષ્ટક

હવે, વર્ગો (Categories) અને નવાં ઉત્પાદનો (New Products) દર્શાવવા માટે 2 હરોળ અને 2 સ્તંભ ધરાવતું અન્ય કોષ્ટક ઉમેરીશું. પ્રથમ હરોળના સેલમાં "Categories" અને "New Products" લખાણ ટાઈપ કરો. **Format → Text color**નો ઉપયોગ કરી લખાણને યોગ્ય રંગ આપો. આકૃતિ 3.10માં લખાણ સાથેની હરોળ દર્શાવી છે.



આકૃતિ 3.10 : કોષ્ટકની હરોળમાં ઉમેરવામાં આવેલ લખાણ

ત્યાર પછી, બીજી હરોળમાં categoriesની નીચે એક સ્તંભ અને વધુ હરોળ સાથેનું એક અન્ય કોષ્ટક ઉમેરીશું. આ કોષ્ટકમાં સ્કૂલબેગ, પાણીની બૉટલ અને અન્ય વસ્તુ માટેના દરેક વર્ગ એક-એક હરોળમાં ઉમેરીશું. લખાણને યોગ્ય રંગ આપો. અહીં વર્ગ છૂટા પાડવા માટે આપણે આડી લીટી પણ ઉમેરીશું. આડી લીટી ઉમેરવા માટે **Insert → Horizontal line** વિકલ્પનો ઉપયોગ કરવામાં આવે છે. વળી, વર્ગોને એવી રીતે લિંક આપવામાં આવશે કે જે આપણને પસંદ કરેલ વર્ગના સંબંધિત પાના પર લઈ જશે. વર્ગ પરની લિંક આપણે પછીથી બનાવીશું. હવે આપણે નવા ઉત્પાદનના સેલની નીચે વસ્તુની છબી અને તેને સંબંધિત વિગતો ઉમેરવા કોષ્ટકની રચના કરીએ. પાનાંમાં દર્શાવવા ઇચ્છતા હોઈએ તે વસ્તુ આધારિત કોષ્ટક ઉમેરો. અહીં આપણે 3 હરોળ અને 3 સ્તંભ સાથેનું કોષ્ટક ઉમેર્યું છે. વેબપેજમાં ઉમેરેલું કોષ્ટક આકૃતિ 3.11માં દર્શાવ્યું છે.



આકૃતિ 3.11 : વર્ગોની યાદી દર્શાવતી સ્ક્રીન

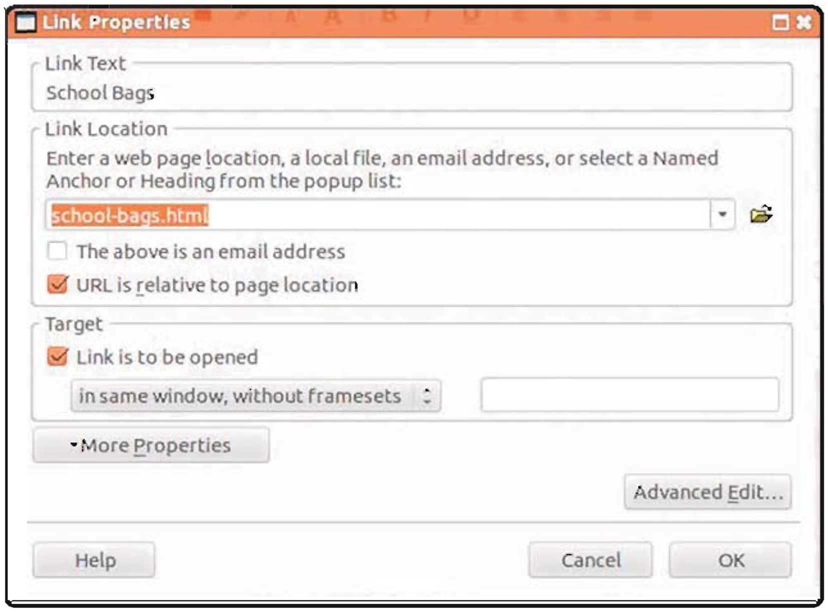
ત્યાર બાદ, પાનાના નીચેના ભાગમાં "About school plaza", "Feedback", "Contact us" અને "Site map"ની વિગતો માટે અન્ય એક કોષ્ટક ઉમેરીશું. કોષ્ટકની રચના કરી લખાણ ઉમેરો. આ લખાણ પણ લિંક બનશે.

આપણે લગભગ આખું હોમપેજ બનાવી લીધું છે. જ્યારે કોઈ પણ વર્ગ પર ક્લિક કરવામાં આવશે ત્યારે, પાનાંની ગોઠવણ સમાન રહેશે, પરંતુ તે વર્ગને અનુરૂપ ચિત્ર બદલાઈ જશે. આમ, હવે નવા વેબપેજની રચના કરવાની જરૂર નથી, પરંતુ હોમપેજની જ એકથી વધુ નકલો કરી જરૂર મુજબ તેમાં ફેરફાર કરી શકાશે.

Save As વિકલ્પ પસંદ કરી ફાઈલનો જુદાં-જુદાં નામથી સંગ્રહ કરો. અહીં ફાઈલનો સંગ્રહ waterbottles.html, school-bags.html અને lucnchbox.html નામ સાથે કરવામાં આવ્યો છે. હવે જ્યારે નિશ્ચિત કોઈ વર્ગના લિંક પર ક્લિક કરવામાં આવે, ત્યારે તેને સંબંધિત વેબપેજ ખૂલશે. તમે એ ધ્યાનમાં લીધું હશે કે બનાવેલાં તમામ વેબપેજને હોમપેજ જેવું જ શીર્ષક મળેલું છે. એક પછી એક વેબપેજ ખોલો અને **Format → Page Title** અને **Properties** વિકલ્પનો ઉપયોગ કરી દરેક વેબપેજનું શીર્ષક બદલો. હવે, દરેક વેબપેજ ખોલી તેના વર્ગને અનુરૂપ ચિત્રો ઉમેરો. હોમપેજમાં ચિત્ર ઉમેરવાની પદ્ધતિને અનુસરવાથી અન્ય તમામ વેબપેજમાં પણ ચિત્રો ઉમેરી શકાશે.

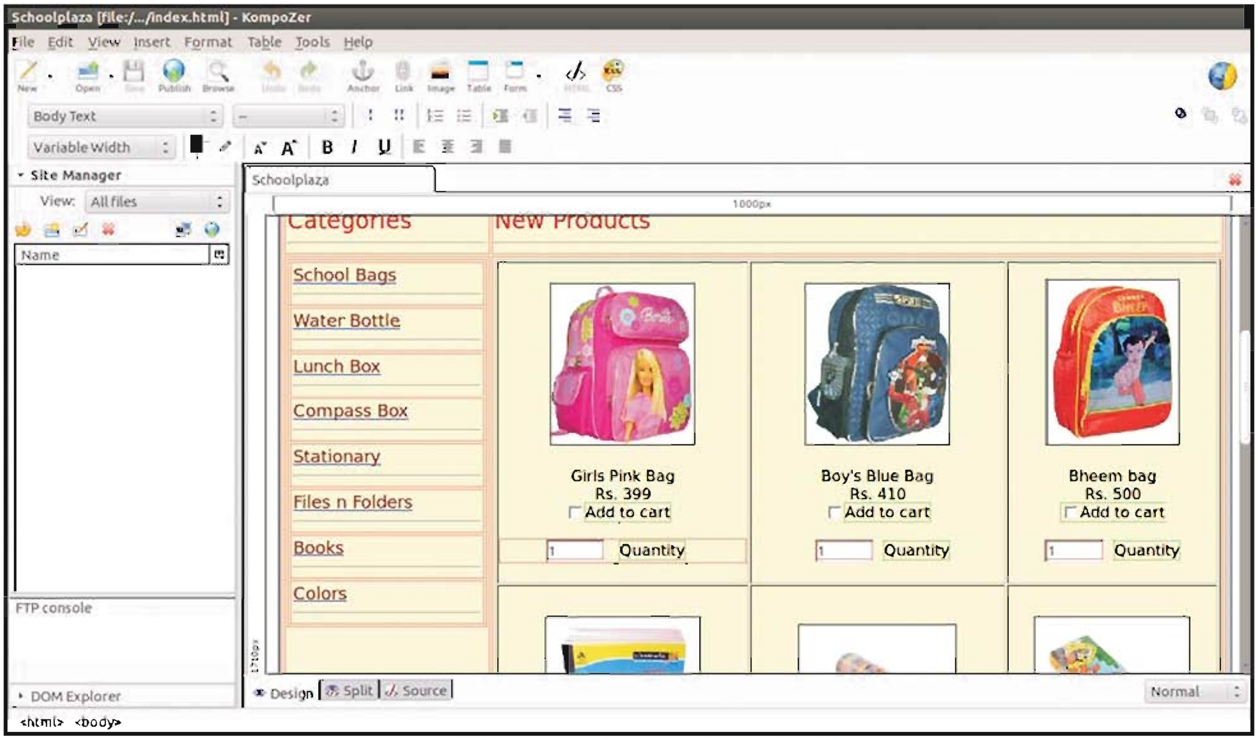
હોમપેજમાં વસ્તુઓનાં ચિત્ર અને તેની વિગતો ઉમેરતાં પહેલાં જુદા-જુદા વર્ગો માટે લિંકની રચના કરીએ. વર્ગનાં નામ પર ક્લિક કરવાથી તેને સંબંધિત વેબપેજ ખૂલશે.

index.html ફાઈલ ખોલો. લિંકની રચના કરવા માટે લખાણ (School-Bags) પસંદ કરો અને **Insert → Link** વિકલ્પ પસંદ કરો. તે આકૃતિ 3.12માં દર્શાવ્યા મુજબનું ડાયલોગબૉક્સ રજૂ કરશે. Link Location વિકલ્પ હેઠળ, જે ફાઈલ ખોલવાની છે, તે દર્શાવો (school-bags.html). OK બટન પર ક્લિક કરો.



આકૃતિ 3.12 : Link Properties ડાયલોગબૉક્સ

હવે, હોમપેજમાં નવા ઉત્પાદનોની વિગતો ઉમેરીએ. કોષ્ટકના દરેક સેલમાં વસ્તુનું ચિત્ર, વસ્તુનું નામ, કિંમત તથા Add to Cart ચેકબૉક્સ અને વસ્તુના જથ્થા માટેનું ટેક્સ્ટબૉક્સ ઉમેરીશું. અગાઉ ઉપયોગમાં લીધેલી પદ્ધતિ મુજબ ચિત્ર ઉમેરો. ઉત્પાદનનું નામ અને કિંમત ઉમેરો. હવે, ફૉર્મફિલ્ડ દ્વારા એક ચેકબૉક્સ ઉમેરો. ચેકબૉક્સ માટે b1 નામની "id" HTML લાક્ષણિકતા ઉમેરીશું, જેનો પછીથી જાવાસ્ક્રિપ્ટમાં ઉપયોગ કરવામાં આવશે. "Add to cart" નામનું લેબલ ઉમેરો. પછીથી, શરૂઆતની કિંમત 1 અને "q1" id સાથે ટેક્સ્ટબૉક્સ ઉમેરો. "Quantity" લેબલ ઉમેરો. જ્યારે ઉપયોગકર્તા Add to Cart ચેકબૉક્સ પસંદ કરશે, ત્યારે વસ્તુ અને તેના જથ્થાનો શૉપિંગ કાર્ટમાં સમાવેશ કરવામાં આવશે. ઉમેરવામાં આવેલ ઉત્પાદનોની વિગતો આકૃતિ 3.13માં દર્શાવી છે.



આકૃતિ 3.13 : ઉત્પાદન અંગેની વિગતો

નોંધ : આપણે બનાવેલ વર્ગ માટેનાં ચિત્રો વેબપેજમાં ઉમેરો. બાકીની વિશેષતાઓ સમાન રહેશે. અહીં અન્ય વેબપેજ દર્શાવવામાં આવ્યાં નથી.

પાનાંના નીચેના ભાગમાં "Purchase" બટન ઉમેરીશું, જે આપણને બિલની વિગતોના વેબપેજ તરફ લઈ જશે. આ પાનું ખરીદવામાં આવેલ કુલ વસ્તુઓની સંખ્યા અને કુલ રકમ દર્શાવે છે. બટનની onclick ઘટના સમયે કોડ લિસ્ટિંગ 3.1માં દર્શાવેલ જાવાસ્ક્રિપ્ટ ઉમેરીએ. તે oncart() વિધેયનો અમલ કરશે અને ત્યાર પછી oncart() વિધેય બિલ માટેનું વેબપેજ દર્શાવશે.

```
<script>
function oncart()
{
var check=false;
var cookievalue="";

if(document.getElementById("b1").checked)
{
check=true;
var quantity=document.getElementById("q1").value;
var price = 399;
var total=(price*quantity);
cookievalue+=":Girls pink bag,"+ price + "," + quantity + "," + total;
}

if(document.getElementById("b2").checked)
{
```

```
check=true;
var quantity=document.getElementById("q2").value;
var price = 410;
var total=(price*quantity);
cookievalue+=":Boy's Blue bag,"+ price + "," + quantity + "," + total;
}

if(document.getElementById("b3").checked)
{
check=true;
var quantity=document.getElementById("q3").value;
var price=500;
var total=(500*quantity);
cookievalue+=":Bheem bag,"+ price + "," + quantity + "," + total;
}

if(document.getElementById("nb1").checked)
{
check=true;
var quantity=document.getElementById("q4").value;
var total=(50*quantity);
cookievalue+=":NoteBook (set of 3),"+total;
}

if(document.getElementById("cp1").checked)
{
check=true;
var quantity=document.getElementById("q5").value;
var total=(45*quantity);
cookievalue+=":Color pencils,"+total;
}

if(document.getElementById("p1").checked)
{
check=true;
var quantity=document.getElementById("q6").value;
var total=(80*quantity);
cookievalue+=":Pencil Box,"+total;
}

if(document.getElementById("l1").checked)
{
```

```
check=true;
var quantity=document.getElementById("q7").value;
var total=(110*quantity);
cookievalue+=":Angry Bird Lunch box,"+total;
}

if(document.getElementById("l2").checked)
{
check=true;
var quantity=document.getElementById("q8").value;
var total=(120*quantity);
cookievalue+=":cartoon lunch box,"+total;
}

if(document.getElementById("e1").checked)
{
check=true;
var quantity=document.getElementById("q9").value;
var total=(50*quantity);
cookievalue+=":Eraser (set of 4),"+total;
}

if(!check)
{
alert("No item selected");
}
else
{
document.cookie=cookievalue;
window.location="bill.html";
}
}
  </script>
```

કોડલિસ્ટિંગ 3.1 : યથાર્થતા માટેની જાવાસ્ક્રિપ્ટ

કોડલિસ્ટિંગ 3.1માં oncart() વિધેય દર્શાવ્યું છે. પ્રથમ if વિધાન ચકાસે છે કે item1 માટેનું ચેકબૉક્સ પસંદ કરવામાં આવ્યું છે કે નહીં. આ પહેલાં આપણે item1ના ચેકબૉક્સને "b1" નામથી id આપ્યું છે. document.getElementById("b1").checked વિધાનનો ઉપયોગ કરી તપાસવામાં આવ્યું છે કે ઉપયોગકર્તાએ વસ્તુ પસંદ કરી છે કે નહીં. જો વસ્તુ પસંદ કરવામાં આવી હોય, તો તેનો જથ્થો ચલમાં સાચવી કુલ રકમની ગણતરી કરવામાં આવશે. અંતમાં, સ્ટ્રિંગ સ્વરૂપે વસ્તુનું નામ, કિંમત, જથ્થો અને કુલ રકમનો ચલમાં સંગ્રહ કરીશું. અહીં દરેક વિગતને અલ્પવિરામ (,) દ્વારા છૂટી પાડવામાં આવી છે. દરેક ઉત્પાદનની વિગતોને વિસર્ગ (:) દ્વારા છૂટી પાડવામાં આવી છે. યાદીમાં આવેલ દરેક વસ્તુ માટે સ્ક્રિપ્ટને આગળ વધારવામાં આવે છે.

જો ઉપયોગકર્તાએ એક પણ વસ્તુ પસંદ કરી ન હોય અને purchase બટન પર ક્લિક કરવામાં આવે, તો "No item is selected" લખાણ સાથેનો સંદેશ દર્શાવવામાં આવશે. નહીં તો, વસ્તુને ‘કૂકી’ (cookie) તરીકે સંગ્રહવામાં આવશે. ઉપયોગકર્તાના કમ્પ્યૂટરમાં સંગ્રહ કરવામાં આવતા ચલને ‘કૂકી’ કહે છે. તે માર્ગ યાદ રાખી મુલાકાતીના વધુ સારા અનુભવ માટે અથવા સાઇટની આંકડાવિષયક માહિતી માટે જરૂરી પસંદગીઓ, ખરીદી, વળતર અને અન્ય માહિતી પૂરી પાડે છે. જાવાસ્ક્રિપ્ટમાં document ઑબ્જેક્ટના Cookies ગુણધર્મ દ્વારા કૂકીનો ઉપયોગ કરી શકાય છે. કૂકીને વાંચી શકાય છે, રચી શકાય છે, સુધારી શકાય છે તથા દૂર કરી શકાય છે.

જ્યારે ઉપયોગકર્તા Purchase બટન પર ક્લિક કરે છે, ત્યારે જાવાસ્ક્રિપ્ટ દ્વારા "bill.html" પાનું દર્શાવવામાં આવે છે. bill.html પાનામાં જાવાસ્ક્રિપ્ટ દ્વારા પસંદ કરેલ વસ્તુઓ માટે કુલ વસ્તુઓની સંખ્યા સાથે ચૂકવવાની કુલ રકમની ગણતરી કરવામાં આવે છે. કુલ રકમની ગણતરી કરવા માટેની જાવાસ્ક્રિપ્ટ કોડલિસ્ટિંગ 3.2માં દર્શાવી છે.

અહીં myfun() નામનું વિધેય વ્યાખ્યાયિત કરવામાં આવ્યું છે. આ વિધેય કૂકી પાસેથી વિગતો લઈ તેને ઍરેમાં સંગૃહીત કરે છે. ત્યાર પછી વિગતોને parseFloat() વિધેય દ્વારા અંકોમાં ફેરવી સરવાળો કરવામાં આવે છે. અંતમાં સરવાળાને દર્શાવવામાં આવે છે.

```
function myfun()
{
var cookiearr=new Array();
cookiearr=document.cookie.split(":");
document.writeln("<center><h1>Welcome To Billing</h1></center>");
document.write("<center><table
border=3><thead><tr><th>Description</th><th>Price</th><th>Quantity</th><th>Total</th></tr></
thead>");
var total2=0;
var total3=0;

for(var i=1;i<cookiearr.length;i++)
{
var cookiedata=cookiearr[i].split(",");
document.writeln("<tr><td>"+cookiedata[0]+"</td><td>"+cookiedata[1]+"</td><td>"+cookiedata[2]
+"</td><td>"+cookiedata[3]+"</td></tr>");

total2+=parseFloat(cookiedata[2]);
total3+=parseFloat(cookiedata[3]);
}
document.write("<tr><td>Total</td><td>"+" "
+"</td><td>"+total2+"</td><td>"+total3+"</td></tr>");
document.write("</tbody></table></center>");
}
 </script>
```

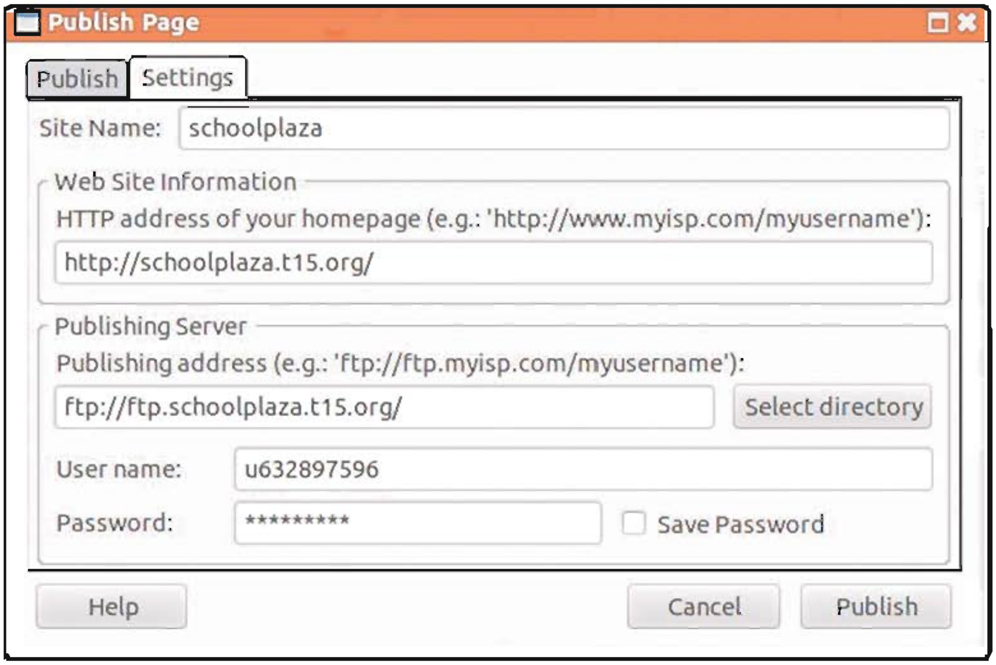
કોડલિસ્ટિંગ 3.2 : કુલ રકમ શોધવા માટેની જાવાસ્ક્રિપ્ટ

અંતમાં, ઉપયોગકર્તા બિલના વેબપેજ પર આવેલા Proceed to Checkout બટન પર ક્લિક કરશે. આ સમયે જો ઉપયોગકર્તાએ લોગ-ઇન કરેલું નહીં હોય, તો તેને તે વિશે પૂછવામાં આવશે અને જો ઉપયોગકર્તાની નોંધણી થયેલ ન હોય, તો તેને નવું ખાતું (account) બનાવવા માટે કહેવામાં આવશે. આ માટે આગળના પ્રકરણમાં ચર્ચા કરી તે મુજબ નોંધણી માટેનું ફૉર્મ (Registration form) બનાવી શકાય છે. ત્યાર પછી સંસ્થાના માલિક દ્વારા નક્કી કરવામાં આવેલ ચૂકવણીની વિગતો ઉમેરી શકાય. સામાન્ય રીતે, આ પ્રકારની વેબસાઇટના વ્યવસ્થાપન માટે ડેટાબેઝની રચના કરવાની જરૂર પડે છે, જેમાં ઉત્પાદન, નોંધણી થયેલાં ઉપયોગકર્તાઓ, લોગ-ઇન નામ, પાસવર્ડ અને અન્ય સંબંધિત વિગતોનો સંગ્રહ કરવામાં આવ્યો હોય છે.

વેબસાઇટ પ્રકાશિત કરવી (Publishing a Website)

અત્યાર સુધી આપણે કમ્પોઝરની મદદથી સરળ વેબસાઇટની રચના વિશે શીખ્યા. હવે વેબસાઇટને કેવી રીતે પ્રદર્શિત (publish) કરવી તે જોઈએ. વેબસાઇટને પ્રદર્શિત કે પ્રકાશિત કરવી એટલે કે ઉપયોગકર્તા સાઇટનાં વેબપેજ, ચિત્રો અને સ્ટાઇલશીટનો ઉપયોગ કરી શકે તે માટે તેને વેબસર્વર પર ખસેડવી. આ પ્રક્રિયાને સામાન્ય રીતે 'અપલોડિંગ' (Uploading) તરીકે ઓળખવામાં આવે છે.

જો વેબસર્વર પર આપણું ખાતું હોય, તો વેબસાઇટને અપલોડ કરી શકાય છે. ઇન્ટરનેટની સેવા પૂરી પાડનાર સંસ્થા (Internet Service Provider - ISP) વેબસર્વર પર મર્યાદિત જગ્યા પૂરી પાડે છે. વ્યાવસાયિક રીતે હોસ્ટિંગ પૂરું પાડતી સંસ્થાઓ પાસેથી જગ્યા ખરીદી પણ શકાય છે. વૈકલ્પિક રીતે, વેબસાઇટને અપલોડ કરવા માટે ઘણા વેબહોસ્ટ (web host) છે, જે ઉપયોગકર્તાને મર્યાદિત જગ્યા નિઃશુલ્ક પૂરી પાડે છે. આ નિઃશુલ્ક જગ્યા મર્યાદિત સમયમર્યાદા માટે પણ હોઈ શકે છે. આપણે બનાવેલ વેબસાઇટના ઉદાહરણમાં આવી જ નિઃશુલ્ક જગ્યા પૂરી પાડતી સાઇટનો ઉપયોગ કરી વેબસાઇટ અપલોડ કરી છે. સાઇટને પ્રદર્શિત કરવા સિસ્ટમની ગોઠવણી માટે તે જગ્યાનાં સેટિંગ જાણવાં જરૂરી છે. index.html વેબપેજ ખોલો અને **File → Publish** વિકલ્પ પસંદ કરો. આકૃતિ 3.14 દર્શાવ્યા મુજબ Publish Page ડાયલોગબૉક્સ રજૂ કરવામાં આવશે.



આકૃતિ 3.14 : Publish Pag e ડાયલોગબૉક્સ

Publish Page ડાયલોગબૉક્સમાં આપેલ ગોઠવણના વિભાગોમાં નીચેની વિગતો પૂરી પાડવામાં આવે છે :

1. Site name ફિલ્ડમાં વેબસાઇટનું નામ ઉમેરો. વેબસાઇટનો સંદર્ભ મેળવવા માટે આ નામનો ઉપયોગ કમ્પોઝર (KompoZer) દ્વારા માત્ર આંતરિક હેતુ માટે કરવામાં આવે છે.

2. Website informationમાં "HTTP address of your homepage" ફિલ્ડ વેબસાઇટનું પ્રત્યક્ષ સરનામું અથવા URL રજૂ કરે છે.

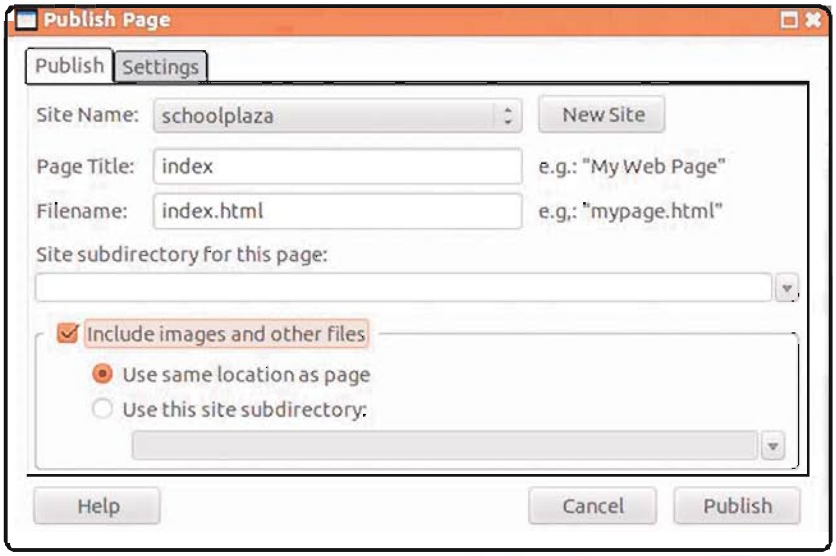
   નોંધ : આપણે ઇન્ટરનેટ પર ડોમેઇનની નોંધણી કરાવી શકીએ છીએ. ઇન્ટરનેટ પર એવી ઘણી વેબસાઇટ ઉપલબ્ધ છે, જેની મદદથી આપણે ડોમેઇન નામની નોંધણી કરી શકીએ છીએ. ઉદાહરણ તરીકે, સામાન્ય ખર્ચ ચૂકવી www.schoolplaza.com જેવું નામ મેળવી શકાય છે.

3. જગ્યા ખરીદવામાં આવી હોય તે ISP કે વેબહોસ્ટ દ્વારા વેબસાઇટ પ્રદર્શિત કરવા માટે વિગતો પૂરી પાડવામાં આવે છે. પ્રકાશન માટેનું સરનામું ટાઇપ કરો. જો FTP દ્વારા જોડાણ કરવામાં આવ્યું હોય, તો FTP સરનામું ટાઇપ કરો. આપણા કમ્પ્યૂટર પરથી વેબહોસ્ટ પર ફાઇલોનાં સ્થાનાંતરણ માટે File Transfer Protocol (FTP)નો ઉપયોગ

કરવામાં આવે છે. અગાઉ ચર્ચા કરી તે મુજબ, આ પ્રક્રિયાને અપલોડિંગ કે પ્રકાશન (publishing) કહે છે. FTP યૂઝરનેમ અને પાસવર્ડ આપો.

નોંધ : વેબહોસ્ટ દ્વારા પૂરી પાડવામાં આવેલી નિ:શુલ્ક જગ્યાની વિગતો આકૃતિ 3.14માં દર્શાવી છે.

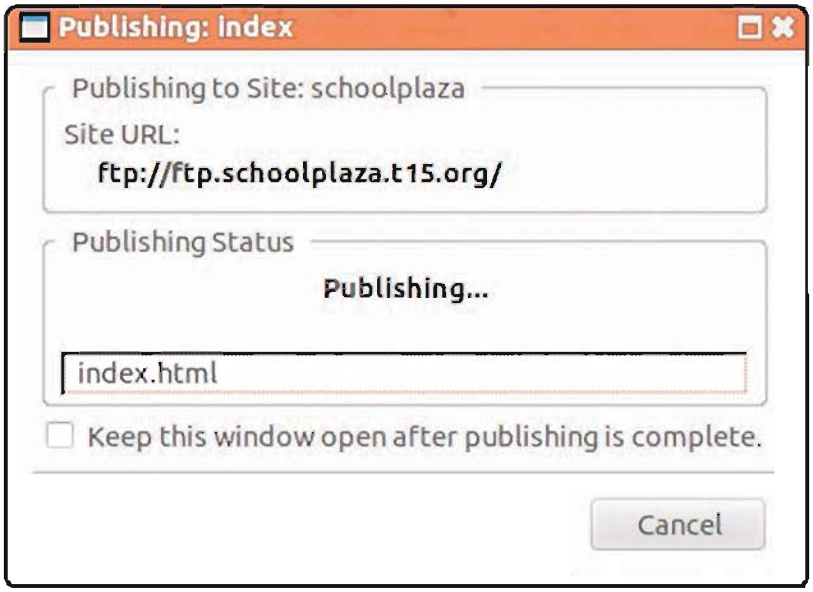
આકૃતિ 3.14માં આવેલા Publish વિભાગ પર ક્લિક કરો. આમ કરવાથી આકૃતિ 3.15માં દર્શાવેલ ડાયલોગબૉક્સ રજૂ થશે.



આકૃતિ 3.15 : Publish Page ડાયલોગબૉક્સનો Publish વિભાગ

હવે, નીચે દર્શાવેલ પગલાંને અનુસરો :

1. જો પસંદ થયેલ ન હોય, તો ડ્રૉપડાઉન મેનુમાંથી સાઇટનું નામ પસંદ કરો.
2. Page Title ઉમેરો, તે index.htmlના શીર્ષકનો સંદર્ભ આપે છે.
3. ફાઇલનું નામ આપો. અહીં index.html ફાઇલ ખૂલેલી હોવાને કારણે તે આપોઆપ ફાઇલનું નામ લઈ લેશે.
4. જો વેબપેજ સાઇટની પેટા ડિરેક્ટરીમાં હોવું જરૂરી હોય, તો તેને "Site subdirectory" ફિલ્ડમાં ઉમેરો. કેટલીક હોસ્ટ સંસ્થાઓની એ જરૂરિયાત હોય છે કે વેબપેજને public_html જેવી સ્વતંત્ર ડિરેક્ટરીમાં રાખવામાં આવે.
5. ચિત્રો માટેની ડિરેક્ટરી નિશ્ચિત કરો. સામાન્ય રીતે ચિત્રોને તે જ ડિરેક્ટરીમાં રાખવામાં આવે છે.
6. Publish બટન પર ક્લિક કરો. આકૃતિ 3.16માં દર્શાવેલ Publishing ડાયલોગબૉક્સ રજૂ કરવામાં આવશે.



આકૃતિ 3.16 : Publishing ડાયલોગબૉક્સ

હવે કમ્પોઝર (KompoZer) ફાઈલોને અપલોડ કરશે અને અંતમાં Pop-up સંદેશ પ્રદર્શિત કરશે. અપલોડિંગ થઈ ગયા બાદ, બ્રાઉઝરમાં URL ટાઈપ કરી વેબસાઇટને તપાસી લો.

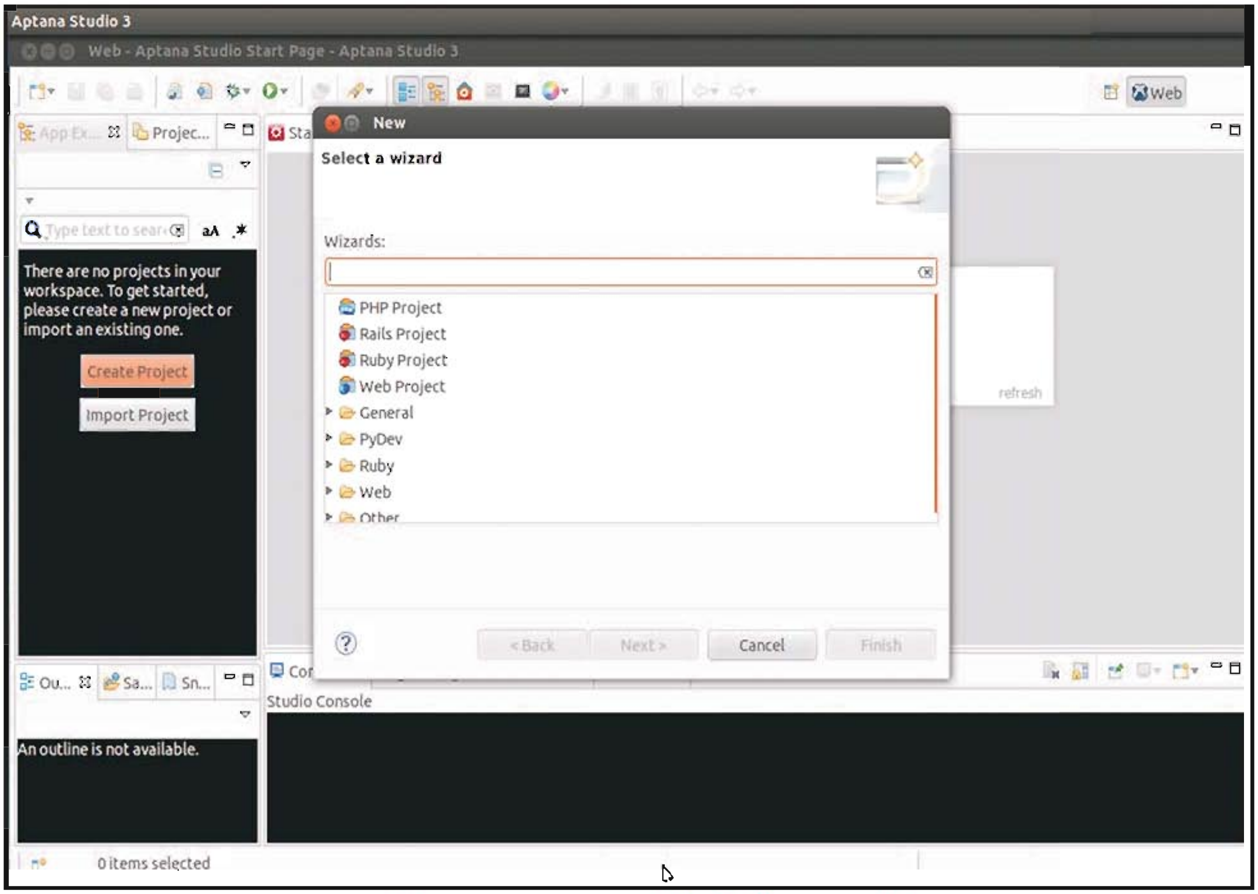
વેબવિકાસ માટેનાં અન્ય ઑપનસૉર્સ ટૂલ્સ (Other Open Source Web Development Tools)

કમ્પોઝરની મદદથી વેબસાઇટની રચના કેટલી સરળતાપૂર્વક કરી શકાય, તેનો આપણે અભ્યાસ કર્યો. આપણે અગાઉ જોયું તે મુજબ કમ્પોઝર એ વેબવિકાસ માટેનું એક નિઃશુલ્ક અને ઑપનસૉર્સ IDE છે. કમ્પોઝરની જેમ જ વેબવિકાસ માટેનાં ઘણાં નિઃશુલ્ક ટૂલ્સ ઇન્ટરનેટ પર ઉપલબ્ધ છે.

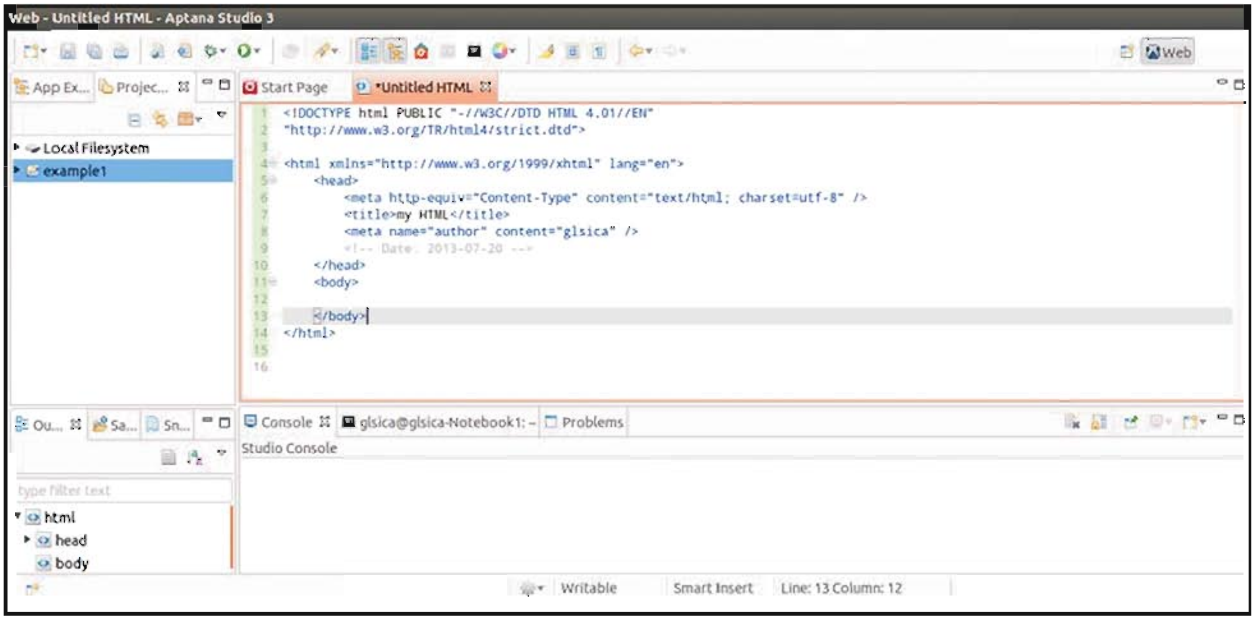
નેટ પર સરળતાથી ઉપલબ્ધ હોય અને વેબ ડેવલપર દ્વારા વેબસાઇટના વિકાસ માટે સરળતાથી ઉપયોગમાં લઈ શકાય તેવાં કેટલાંક ઑપનસૉર્સ ટૂલ્સ વિશે ચર્ચા કરીએ.

અપ્ટાના સ્ટુડિયો (Aptana Studio)

વેબવિનિયોગ બનાવવા માટે અપ્ટાના સ્ટુડિયો એક સક્ષમ ઑપનસૉર્સ IDE (Integrated Development Environment) છે. તે HTML, CSS, JavaScript, Ruby, Rails, PHP, Python અને અન્ય ઘણી ભાષાઓને સમર્થન આપતું એક સંપૂર્ણ વેબવિકાસ માટેનું વાતાવરણ પૂરું પાડે છે. તે વિપુલ સંખ્યામાં વધારાના પ્લગ-ઇન સાથે આપવામાં આવે છે. અપ્ટાના સ્ટુડિયો 2.0.5 (જે અપ્ટાના સ્ટુડિયો 2 પણ કહેવાય છે)નો ઉપયોગ HTML, CSS અને JavaScriptની મદદથી વેબવિનિયોગોના વિકાસ માટે થાય છે. www.aptana.com વેબસાઇટ પરથી તે સરળતાથી ડાઉનલોડ કરી શકાય છે. અપ્ટાના સ્ટુડિયોનો દેખાવ આકૃતિ 3.17માં દર્શાવ્યો છે. આકૃતિમાં દર્શાવ્યા મુજબ તેની મદદથી PHP, Rails, Ruby કે Webમાં પ્રોજેક્ટ બનાવી શકાય છે. આ ઇન્ટરફેસમાં લખેલ HTML કોડ આકૃતિ 3.18માં દર્શાવેલ છે.



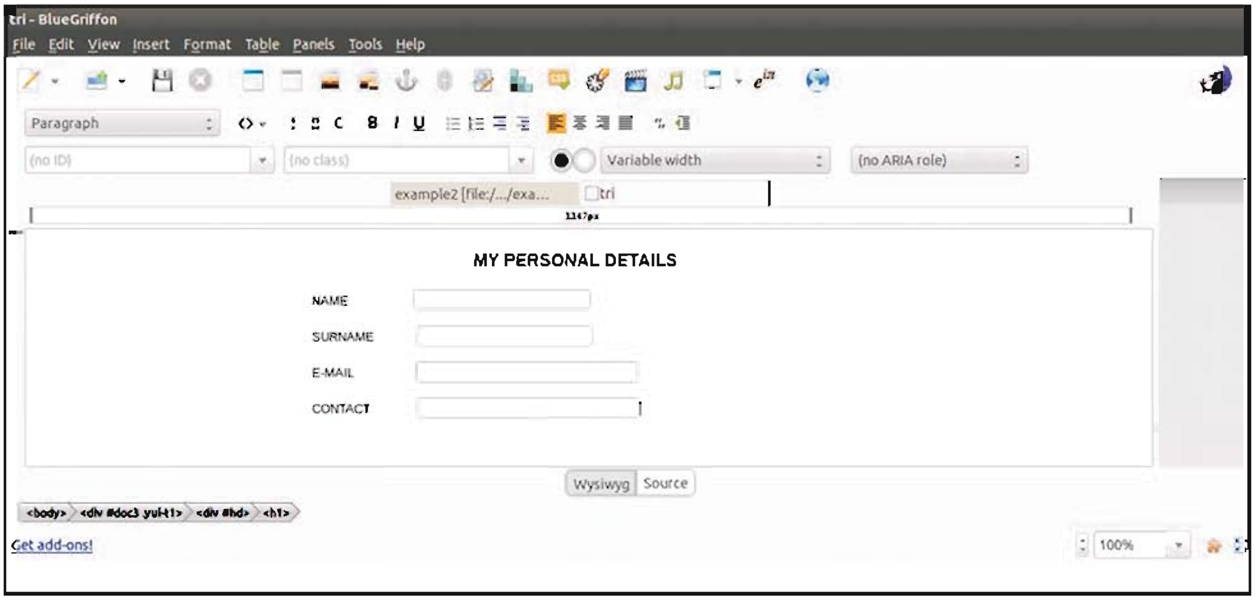
આકૃતિ 3.17 : અપ્ટાના સ્ટુડિયોમાં પ્રોજેક્ટ બનાવવો



આકૃતિ 3.18 : અપ્ટાના સ્ટુડિયોનો ઇન્ટરફેસ

બ્લૂગ્રિફોન (BlueGriffon)

બ્લૂગ્રિફોન એક અન્ય ઓપનસૉર્સ WYSIWYG પ્રકારનું HTML એડિટર છે. તેને www.bluegriffon.org પરથી સરળતાથી ડાઉનલોડ કરી શકાય છે. તે અંગ્રેજી, ડચ, જર્મન, ચાઇનીઝ અને અન્ય ઘણી ભાષાઓનું સમર્થન કરે છે. વેબ-ધારાધોરણના ગહન તકનિકી જ્ઞાનના અભાવમાં પણ આકર્ષક વેબસાઇટની રચના કરી શકાય, તેવું સરળ ઇન્ટરફેસ પૂરી પાડતો એક સાહજિક વિનિયોગ છે. બ્લૂગ્રિફોનની આવૃત્તિ 1.6.2નો ઇન્ટરફેસ આકૃતિ 3.19માં દર્શાવ્યો છે. તેમાં ટૂલ્સનો ઉપયોગ કરી સરળ ફોર્મની રચના કરવામાં આવી છે.

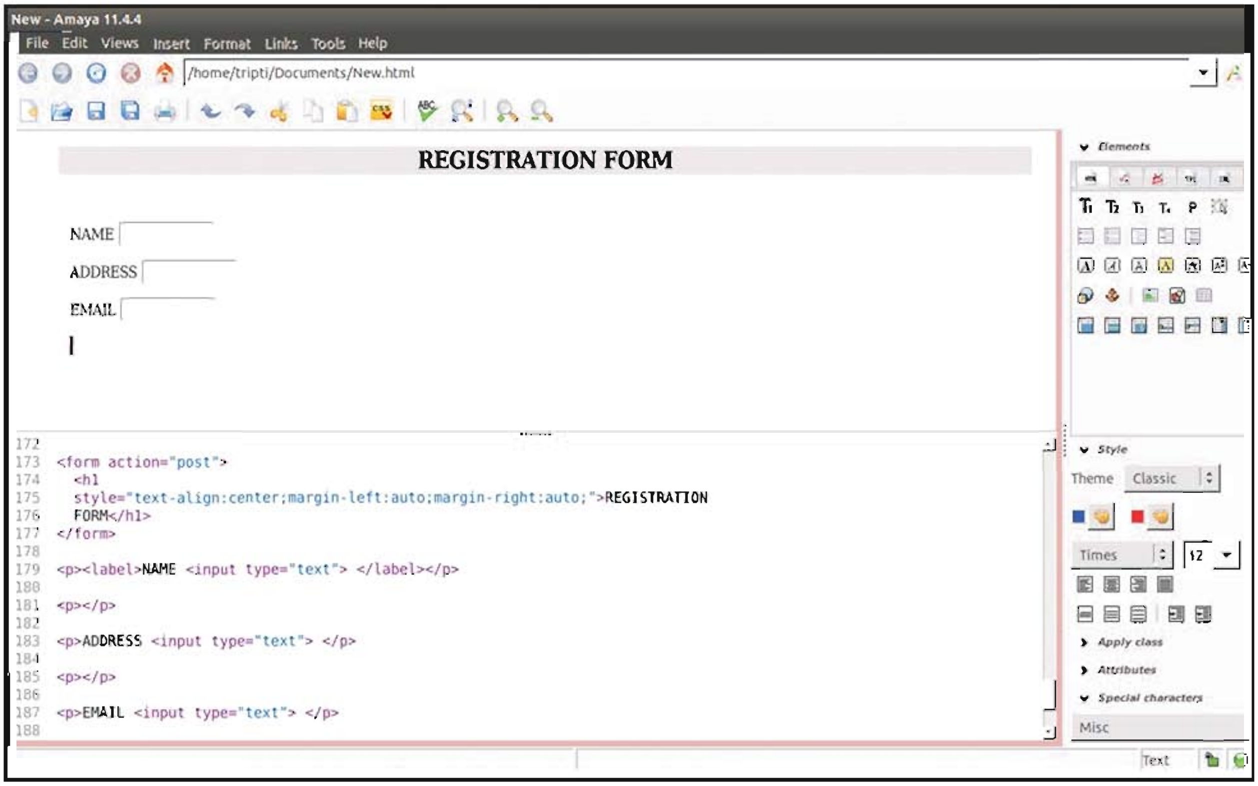


આકૃતિ 3.19 : બ્લૂગ્રિફોનનો ઉપયોગ કરી વેબપેજની રચના

આમાયા (Amaya)

World Wide Web Consortium (W3C) દ્વારા વિકસિત આમાયા એક અન્ય નિઃશુલ્ક, ઓપનસૉર્સ WYSIWYG પ્રકારનું વેબ-એડિટર છે. તેની શરૂઆત HTML/CSS એડિટર તરીકે કરવામાં આવી હતી અને હવે તે ઘણી

XML-આધારિત સિસ્ટમ માટેના એડિટર તરીકે વિકાસ પામ્યું છે. www.w3.org/Amaya પરથી તેને સરળતાથી ડાઉનલોડ કરી શકાય છે. આકૃતિ 3.20માં આમાયા દ્વારા રચિત વેબપેજ દર્શાવ્યું છે. વેબપેજની નીચે જ તેના સૉર્સને પણ જોઈ શકાય છે.



આકૃતિ 3.20 : આમાયાનો ઉપયોગ કરી રચવામાં આવેલું વેબપેજ

સારાંશ

આ પ્રકરણમાં આપણે કમ્પોઝરની મદદથી સરળ વેબસાઇટની રચના વિશે ચર્ચા કરી. વેબસાઇટ બનાવ્યા પછી તેને ઇન્ટરનેટ પર અપલોડ કરવી જરૂરી છે. વેબસાઇટને પ્રદર્શિત કરવી એટલે તેનાં પાનાં, ચિત્રો અને સ્ટાઇલશીટને વેબસર્વર પર ખસેડવાં, જેથી ઉપયોગકર્તા તેનો ઉપયોગ કરી શકે. આ પ્રક્રિયા 'અપલોડિંગ' તરીકે ઓળખાય છે. વેબસર્વર પર ખાતું (account) હોય, તો વેબસાઇટને અપલોડ કરી શકાય છે. ઇન્ટરનેટની સેવા પૂરી પાડનાર સંસ્થા (Internet Service Providers - ISP) વેબસર્વર પર મર્યાદિત જગ્યા પૂરી પાડે છે. વ્યાવસાયિક હોસ્ટિંગ સંસ્થાઓ પાસેથી જગ્યા ખરીદી પણ શકાય છે. આપણે Aptana studio, BlueGriffon અને Amaya જેવાં જુદાં-જુદાં નિઃશુલ્ક, ઑપનસૉર્સ વેબવિકાસ માટે ઉપલબ્ધ IDE વિશે પણ ચર્ચા કરી.

સ્વાધ્યાય

1. વ્યવસાયમાં વેબસાઇટ શા માટે અગત્યનો ભાગ ભજવે છે ?
2. સારી વેબસાઇટના આયોજનની પ્રક્રિયામાં ધ્યાનમાં રાખવા જેવા મુદ્દાઓની યાદી બનાવો.

3. વેબસાઇટ બનાવવા માટેનો હેતુ શા માટે સ્પષ્ટપણે વ્યાખ્યાયિત થયેલ હોવો જોઈએ ?

4. વેબસાઇટની રચના કરતી વખતે અપેક્ષિત પ્રેક્ષકગણ વિશેનું જ્ઞાન કઈ રીતે મદદરૂપ બને છે ?

5. વેબસાઇટની રચના વખતે ક્યા વિષયવસ્તુનો સમાવેશ કરવો જોઈએ ?

6. વેબસાઇટના હોમપેજને શા માટે index.html નામ આપવું જોઈએ ?

7. કોષ્ટકની લાક્ષણિકતાઓ ગોઠવવા માટે ક્યા વિકલ્પ ઉપલબ્ધ છે ?

8. કૂકી એટલે શું ?

9. વેબસાઇટનું પ્રકાશન (publishing) એટલે શું ?

10. વેબવિકાસ માટે ઉપલબ્ધ કેટલાંક ઓપનસોર્સ ટૂલ્સનાં નામ આપો.

11. આપેલ વિકલ્પમાંથી યોગ્ય વિકલ્પ પસંદ કરો :

(1) વ્યવસાયને આગળ વધારવા, ઉત્પાદન વેચવા અને વિશાળ પ્રમાણમાં ગ્રાહકોને આકર્ષવા માટે નીચેનામાંથી શું મદદરૂપ બને છે ?

(a) વેબસાઇટ (b) વેબપેજ (c) ફૉર્મ (d) CSS

(2) વેબસાઇટની વિકાસપ્રક્રિયામાં નીચેનામાંથી ક્યો મુદ્દો અગત્યનો નથી ?

(a) હેતુ (b) પ્રેક્ષકગણ (c) વિષયવસ્તુ (d) નિવેશ

(3) વેબસાઇટમાં કેવી માહિતી હોવી જોઈએ ?

(a) સંપૂર્ણ, સંબંધિત (b) સંપૂર્ણ, અસંબંધિત

(c) અપૂર્ણ, અસંબંધિત (d) અપૂર્ણ, સંબંધિત

(4) નીચેનામાંથી ક્યા પ્રકારની વિષયવસ્તુ વેબસાઇટ, સંસ્થા, ઉત્પાદન, સેવા અને અન્ય વસ્તુઓનું વિહંગાવલોકન પૂરું પાડે છે ?

(a) વિસ્તૃત (b) લાંબી (c) સામાન્ય (d) ટૂંકી

(5) વેબપેજના આંતરજોડાણને શું કહે છે ?

(a) વેબપેજ (b) ફૉર્મ (c) કમ્પોઝર (d) વેબસાઇટ

(6) બ્રાઉઝરના ઍડ્રેસબારમાં URL ઉમેરવામાં આવે, ત્યારે ખોલવામાં આવતા પ્રથમ પાનાંને શું કહે છે ?

(a) હોમપેજ (b) અંતિમ પેજ (c) વેબપેજ (d) પ્રથમ પેજ

(7) વેબસાઇટના હોમપેજને ક્યું નામ આપી સંગ્રહ કરવો જોઈએ ?

(a) first.html (b) index.html (c) home.html (d) one.html

(8) ઉપયોગકર્તાના કમ્પ્યૂટરમાં સંગ્રહ કરવામાં આવતા ચલને શું કહે છે ?

(a) Integer (b) HTML (c) Cookie (d) Java

(9) FTPનું પૂરું નામ શું છે ?

(a) File Truncate Protocol (b) File Transfer Process

(c) Fine Tune Protocol (d) File Transfer Protocol

પ્રાયોગિક સ્વાધ્યાય

1. કમ્પોઝરનો ઉપયોગ કરી તમારી શાળાની વેબસાઇટ બનાવો.
2. કમ્પોઝરનો ઉપયોગ કરી 'ગરવી ગુજરાત' વિષય પર વેબસાઇટ બનાવો.
3. કમ્પોઝરનો ઉપયોગ કરી 'ગુજરાત ટૂરિઝમ' વિષય પર વેબસાઇટ બનાવો.
4. કમ્પોઝરનો ઉપયોગ કરી રમકડાંની દુકાન માટે ઇ-કૉમર્સની વેબસાઇટ બનાવો.

ઇ-કૉમર્સનો પરિચય 4

આપણે માહિતીના યુગમાં જીવી રહ્યા છીએ. રેડિયો, ટેલિવિઝન, વર્તમાનપત્ર અને ઇન્ટરનેટ જેવા સ્રોત પરથી માહિતી ઉપલબ્ધ થતી હોય છે. ઇન્ટરનેટ અને મોબાઇલ સાધનોએ માહિતીનો ઉપયોગ કરવાની આપણી રીતો બદલી નાખી છે. લોકોએ દિવસના લગભગ દરેક કલાકે ઇન્ટરનેટનો ઉપયોગ કરવાનું શરૂ કર્યું છે. પહેલાંની વેબસાઇટનો મુખ્ય ઉપયોગ ઉત્પાદન અને સંસ્થા અંગેની માહિતીના પ્રસારણ માટે થતો. જેમકે, શૈક્ષણિક સંસ્થાની વેબસાઇટ વિવિધ પાઠ્યક્રમ, પાઠ્યક્રમનું વિષયવસ્તુ તથા વિદ્યાર્થીઓ અને શિક્ષકો અંગેની માહિતી પૂરી પાડે છે. એ જ રીતે વ્યાવસાયિક સંસ્થાઓ તેમનાં ઉત્પાદનો, ઉત્પાદનની વિશેષતાઓ, તેમના પુરવઠાકાર (supplier), ઑર્ડર આપવાની રીત વગેરે માહિતી પૂરી પાડે છે.

છેલ્લા કેટલાક દાયકાઓમાં ટેલિકૉમ્યુનિકેશન માળખાં તથા સૉફ્ટવેર તક્નિકમાં થયેલા ધરખમ વિકાસને કારણે ઇન્ટરનેટ વ્યાવસાયિક સંસ્થાઓ જેટલું જ વ્યક્તિગત રીતે પણ ઘણું લોકપ્રિય બન્યું છે. આજકાલ ઇન્ટરનેટે વ્યવસાયના સંચાલનની પદ્ધતિમાં ક્રાંતિકારી ફેરફારો કર્યા છે. બિલની ચુકવણી, બૅન્કને લગતાં કાર્યો અને ખરીદી જેવા વિવિધ હેતુઓ માટે લોકો ઇન્ટરનેટનો ઉપયોગ કરતા થયા છે. ઉત્પાદનનું માર્કેટિંગ અને વેચાણ, સૂચિપત્રો દર્શાવવા, શૅરની લે-વેચ અને ગ્રાહક-સેવા જેવી પ્રવૃત્તિઓ માટે વ્યાવસાયિક સંસ્થાઓ ઇન્ટરનેટનો ઉપયોગ કરે છે, જેને ઇ-કૉમર્સ તરીકે ઓળખવામાં આવે છે.

ઇન્ટરનેટ જેવા ઇલેક્ટ્રૉનિક માધ્યમનો ઉપયોગ કરી ઉત્પાદનનું ખરીદ-વેચાણ, સેવા અને માહિતી પૂરી પાડવાના કાર્યને પણ ઇ-કૉમર્સ તરીકે વ્યાખ્યાયિત કરી શકાય. ઇ-કૉમર્સ એ સંસ્થાઓ, વેપારીઓ અને ગ્રાહકોને તેમના ખર્ચમાં રાહત આપી માલ અને સેવાની ગુણવત્તા વધારવામાં સહાયભૂત થનારી આધુનિક પદ્ધતિ છે. વિતરણની ઝડપમાં પણ તે વધારો કરે છે. તે લોકોને સમય અને અંતરના અવરોધથી ઉગારી વૈશ્વિક બજાર અને વ્યાવસાયિક તકોનો લાભ લેવાની સુવિધા પૂરી પાડે છે.

વેબસાઇટ દ્વારા ઉત્પાદનનું વેચાણ એ વિશ્વભરમાં સૌથી ઝડપી પ્રગતિ કરી રહેલ વેપારની પદ્ધતિ છે. પુસ્તકો, ઇલેક્ટ્રૉનિક સાધનો, મોટરકાર, પ્રવાસન યોજનાઓ (holiday packages) અને બીજાં અનેક પ્રકારનાં ઉત્પાદનો અને સેવાઓનો વ્યાપાર ઑનલાઇન થાય છે. ઘણી વ્યાવસાયિક સંસ્થાઓએ તેમની ઇ-કૉમર્સ આધારિત પ્રવૃત્તિઓ માટેની વેબસાઇટ સ્થાપિત કરી છે. ઇ-કૉમર્સે આખા વિશ્વને એક વૈશ્વિક સ્થાન બનાવી દીધું છે, જ્યાંથી કોઈ પણ વ્યક્તિ કોઈ પણ સ્થળેથી કંઈ પણ ખરીદી શકે છે.

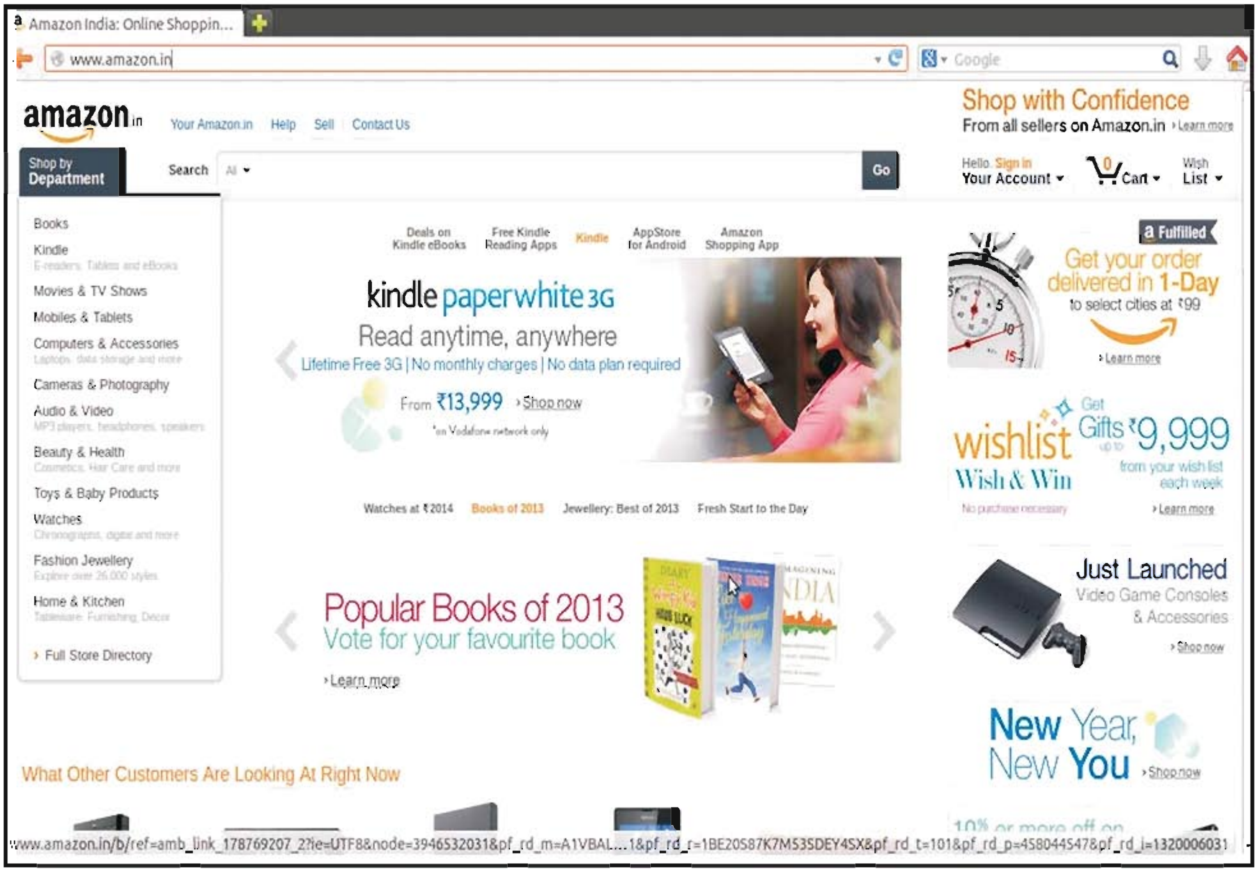
ઈ-કૉમર્સના વિનિયોગ (Applications of E-commerce)

આજકાલ જે વ્યાવસાયિક પ્રવૃત્તિઓમાં ઇ-કૉમર્સનો બહોળો ઉપયોગ કરવામાં આવે છે, તેમાં માલનો વેપાર એટલે કે, માર્કેટિંગ અને વેચાણ, માલની હરાજી તથા બૅંકિંગ અને વીમા જેવી નાણાકીય પ્રવૃત્તિઓનો સમાવેશ થાય છે. આ પ્રવૃત્તિઓની ટૂંકમાં ચર્ચા કરીએ. ઇ-કૉમર્સ અહીં ચર્ચા કરવામાં આવેલ પ્રવૃત્તિઓ પૂરતું જ મર્યાદિત નથી, પરંતુ વ્યવસાય અને અન્ય પ્રવૃત્તિઓ સાથે સંકળાયેલા અન્ય વિકાસશીલ વિસ્તારોમાં પણ વિસ્તર્યું છે.

ઇન્ટરનેટ પર પુસ્તકની દુકાન (Internet Bookshops)

ઇન્ટરનેટ પર ઇ-કૉમર્સનો આ સૌપ્રથમ વિનિયોગ હતો. ગ્રાહકો ઇન્ટરનેટ પર પુસ્તકો ખરીદવાનું પસંદ કરે છે, કારણ કે પુસ્તકોને ભૌતિક રીતે તપાસવાની જરૂર પડતી નથી અને તેનું સરળતાથી વર્ણન કરી શકાય છે. પુસ્તકોને ગ્રાહકો સુધી ક્ષતિરહિત કે અલ્પ ક્ષતિસહિત સરળતાપૂર્વક પહોંચાડી શકાય છે. ઑનલાઇન પુસ્તક-વિક્રેતાને પુસ્તકોની વર્ગીકૃત

યાદી, મુખપૃષ્ઠનાં ચિત્ર, પુસ્તકનું વર્ણન, પાનાંની કુલ સંખ્યા, પુસ્તકની કિંમત, વળતર અને ગ્રાહકોના પ્રતિભાવ દર્શાવવા સારી વેબસાઇટની જરૂર પડે છે. પુસ્તકનાં શીર્ષક, લેખકનાં નામ કે પ્રકાશકનાં નામ દ્વારા પુસ્તકને શોધી શકાય છે. પુસ્તકોનો ઑનલાઇન વિક્રેતા ગ્રાહકોની અભિરુચિની નોંધ રાખે છે અને તેમનો રસ જળવાઈ રહે તે માટે તેમને નવાં પુસ્તકોની જાણકારી આપતા રહે છે. ઑનલાઇન પુસ્તકોની દુકાનોમાંથી પ્રથમ એવી www.amazon.comનું હોમપેજ આકૃતિ 4.1માં દર્શાવ્યું છે.



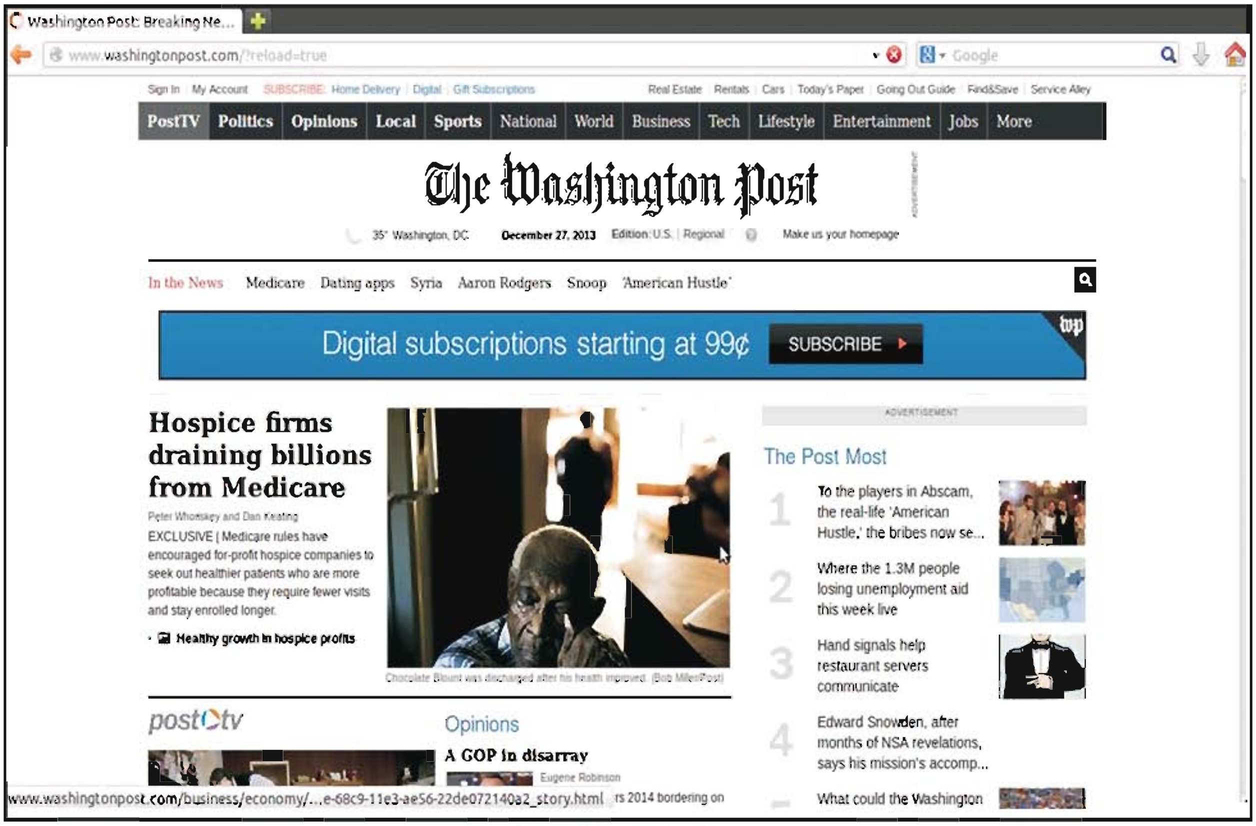
આકૃતિ 4.1 : પુસ્તકની ઑનલાઇન દુકાન

પુસ્તકોની વિશાળ ઑનલાઇન દુકાન ધરાવતી કેટલીક વેબસાઇટ નીચે દર્શાવી છે :

- www.amazon.com
- shopping.indiatimes.com
- www.buybooksindia.com
- www.bookshopofindia.com

ઇલેક્ટ્રૉનિક વર્તમાનપત્ર (Electronic newspaper)

ઇન્ટરનેટ પર ડિજિટલ સ્વરૂપે ઉપલબ્ધ વર્તમાનપત્રને ઇલેક્ટ્રૉનિક વર્તમાનપત્ર કે ઇ-ન્યૂઝપેપર (E-newspaper) તરીકે ઓળખવામાં આવે છે. મુદ્રિત વર્તમાનપત્ર કે ટેલિવિઝન અને રેડિયો દ્વારા પ્રસારિત કરવામાં આવતા સમાચારો કરતાં તે વધુ લાભદાયી છે. વિશ્વસ્તરે થતી ઘટનાઓના તત્કાલ સમાચાર તે આપી શકે છે. ડિજિટલ તકનિકની સહાયથી દર્શકની અભિરુચિને અનુરૂપ બ્રાઉઝરની સંરચના દ્વારા સમાચારોને પ્રદર્શિત કરી શકાય છે. અને મુદ્રણ-પ્રક્રિયામાંથી મુક્તિ મેળવી શકાય છે, જે કિંમત ઘટાડવામાં મદદરૂપ બને છે. મોટા ભાગનાં પ્રતિષ્ઠિત વર્તમાનપત્રો હવે વાચકોને ઇ-વર્તમાનપત્ર પૂરાં પાડે છે. આકૃતિ 4.2માં વૉશિંગ્ટન-પોસ્ટના ઇ-વર્તમાનપત્રનું હોમપેજ દર્શાવ્યું છે.



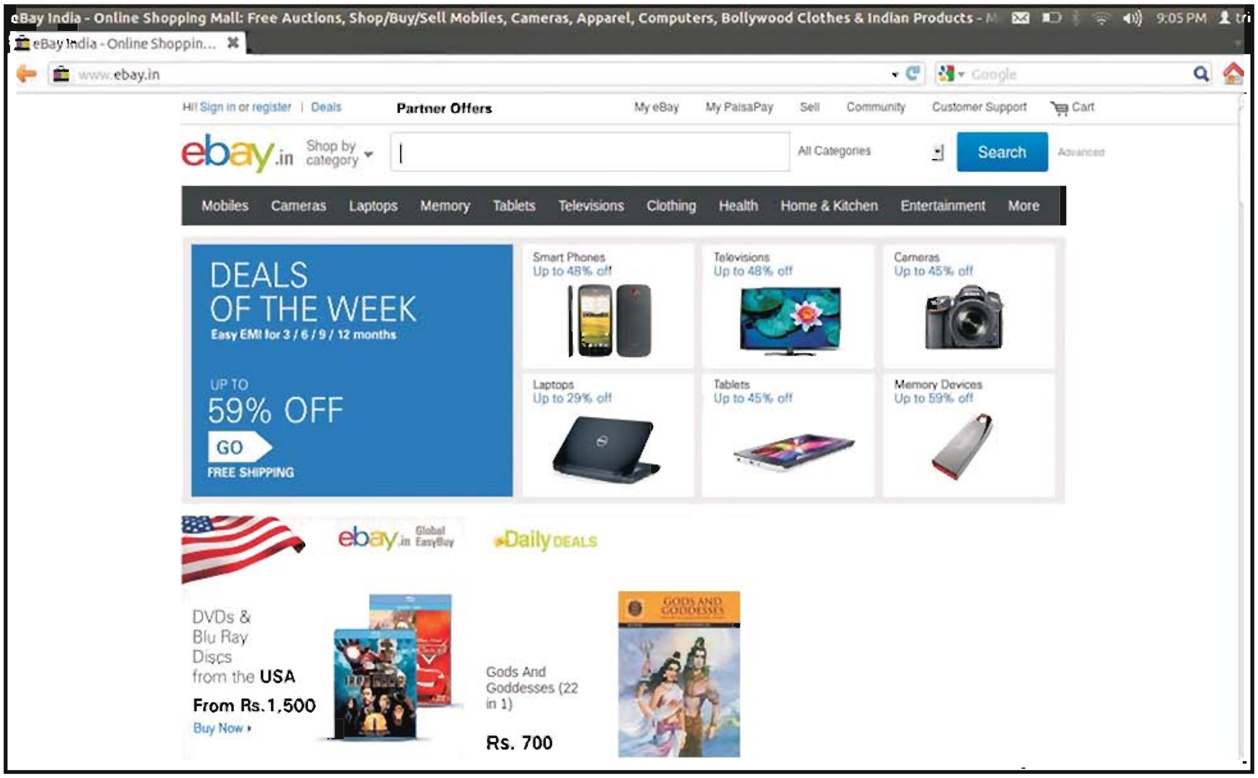
આકૃતિ 4.2 : ઈ-વર્તમાનપત્ર

ઑનલાઈન હરાજી (Online Auctions)

હરાજી ખરીદ અને વેચાણની એવી પ્રક્રિયા છે જે ગ્રાહકને ઉત્પાદનની બોલી બોલવાની તક આપી, ઉચ્ચ બોલી બોલનારને વસ્તુ ખરીદવાની સુવિધા પ્રદાન કરે છે. પરંપરાગત હરાજીમાં મર્યાદિત લોકો ભાગ લે છે. હવે હરાજીની આ પ્રક્રિયાને ઇ-કૉમર્સ તક્નિકની મદદથી અમલમાં મૂકી શકાય છે, જે લોકોને ઇન્ટરનેટ પર બોલી બોલવાની સુવિધા આપે છે. તેને ઑનલાઇન હરાજી (Online Auction) તરીકે ઓળખવામાં આવે છે. માલની જીવંત હરાજી પૂરી પાડતી ઘણી વેબસાઇટ ઉપલબ્ધ છે. આ વેબસાઇટ વેચનાર અને બોલી લગાવનાર બંનેને આધાર પૂરો પાડે છે. જ્યારે તમે આ પ્રકારની સાઇટ પર વસ્તુ મૂકો છો, ત્યારે તમે 'વેચનાર' છો. આ જ સમયે તમે અન્ય વેચનાર દ્વારા સાઇટ પર મૂકેલા ઉત્પાદન માટે બોલી લગાવી શકો છો, તો હવે તમે 'બોલી લગાવનાર' છો.

હરાજીની ઑનલાઇન સાઇટ પર વસ્તુ વેચવા માટે પ્રથમ તમારે તે સાઇટ પર નોંધણી કરાવવી જરૂરી છે. તમે વેચેલી કે બોલી લગાવેલી વસ્તુનું સંધાન (track) મેળવવા, સ્વીકૃત બોલીને શોધવા અને વેચનાર તથા બોલી લગાવનારના પ્રતિસાદનો ડેટાબેઝ બનાવવા નોંધણી જરૂરી બને છે. વેચાણ પહેલાં સભ્યોએ પોતાની મૂળભૂત સંપર્ક-માહિતી પણ પૂરી પાડવી જરૂરી છે. ત્યાર પછી, તેઓ હરાજી માટે મૂકવા માગતા હોય તે વસ્તુ માટે સાઇટ પર આપેલી સૂચનાઓને ક્રમાનુસાર અનુસરે તે જરૂરી છે. સામાન્ય રીતે વિક્રેતા વસ્તુના ડિજિટલ ચિત્ર સહિતની ટૂંકી માહિતી પ્રસ્તુત કરે છે.

હરાજી માટે વેબસાઇટ પર માલ પ્રદર્શિત કરવાથી વ્યક્તિ તેની સારી કિંમત ઉપજાવી શકે છે અને અપેક્ષિત કિંમત સુધી હરાજી પહોંચી જાય કે સમયમર્યાદા પૂરી થાય, ત્યારે વસ્તુ મહત્તમ બોલી લગાવનારને મોકલવામાં આવે છે. બોલી બોલનારને ચુકવણીના વિવિધ વિકલ્પ પણ પૂરા પાડવામાં આવે છે. વિક્રેતા તેના ઉત્પાદન માટે શ્રેષ્ઠ કિંમત મેળવી શકે છે તથા બોલી બોલનાર પોતાની પસંદગીનું ઉત્પાદન ઓછા સમયમાં મેળવી શકે છે. આકૃતિ 4.3માં ઑનલાઇન હરાજીની સાઇટ www.ebay.comનું હોમપેજ દર્શાવ્યું છે.



આકૃતિ 4.3 : ઑનલાઇન હરાજી માટેની સાઈટ

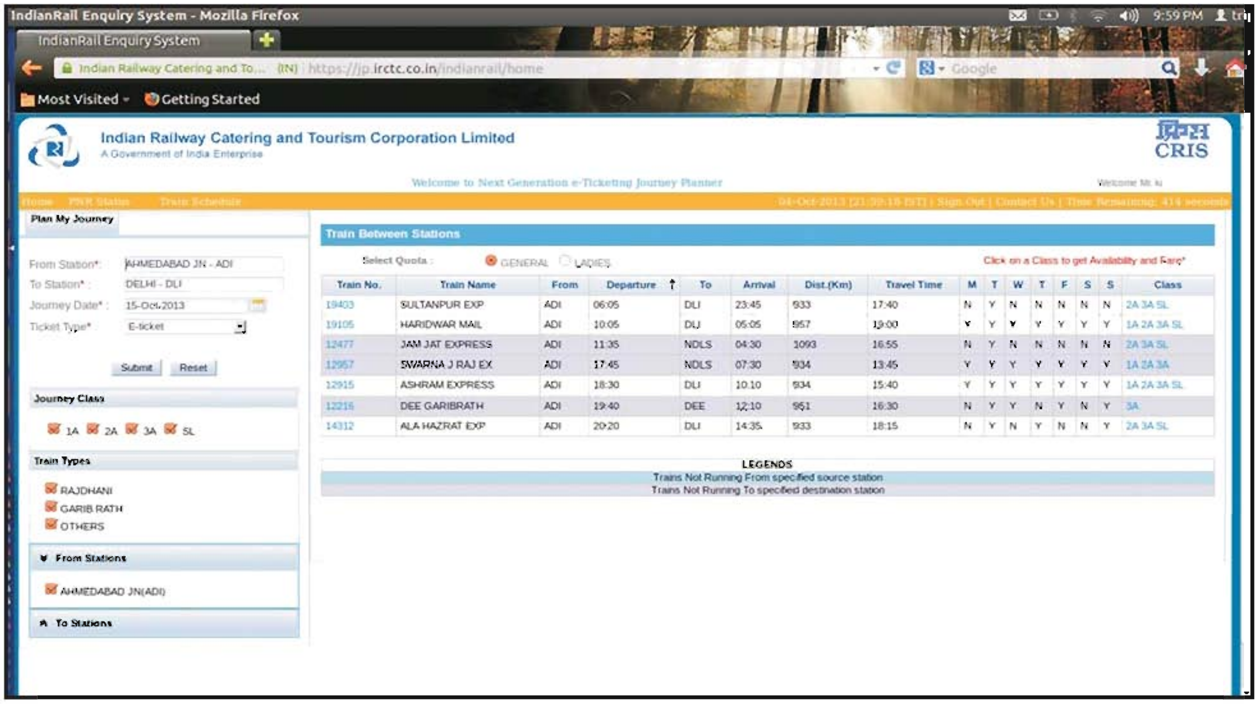
ઑનલાઇન હરાજી માટેની કેટલીક પ્રચલિત વેબસાઇટની યાદી નીચે આપવામાં આવી છે :

- www.ebay.com
- www.onlineauction.com
- www.mybids.in
- www.ubid.com

માર્કેટિંગ અને વેચાણ (Marketing and Selling)

હાલમાં ઘણી સંસ્થાઓ તેમના ઉત્પાદનના પ્રચાર અને સેવાના વ્યવસાયનું સંચાલન વેબસાઇટ દ્વારા કરે છે. વધુ સારા માર્કેટિંગ માટે તેઓ તેમના ઉત્પાદનની સૂચિ (catalogue) ઇન્ટરનેટ પર ઑનલાઇન પૂરી પાડે છે. સૂચિ જુદી જુદી શ્રેણીઓમાં સચિત્ર તો ક્યારેક વીડિયો દ્વારા ઉત્પાદનનું સંક્ષિપ્ત વિવરણ અને વિશેષતાઓ દર્શાવે છે. ગ્રાહક સૂચિને જોઈ શકે છે તથા પોતાની પસંદગીના ઉત્પાદનને શૉપિંગકાર્ટ (shopping cart)માં ઉમેરે છે. ઑનલાઇન શૉપિંગકાર્ટ એ દુકાનમાં આપવામાં આવેલ મૂળભૂત શૉપિંગકાર્ટ જેવું જ હોય છે. સ્થાનિક દુકાનમાં ગ્રાહક ઉત્પાદનને પસંદ કરી શૉપિંગકાર્ટમાં મૂકતો હોય છે. આ પ્રક્રિયા પૂર્ણ થયા બાદ ગ્રાહક દ્વારા ખરીદવામાં આવેલ તમામ ઉત્પાદનોનું શૉપિંગકાર્ટમાંથી બિલ બનાવવામાં આવે છે. આવી જ રીતે, ગ્રાહક ઑનલાઇન શૉપિંગકાર્ટમાં ઉત્પાદન ઉમેરી શકે છે, પસંદ કરેલ ઉત્પાદનને જોઈ શકે છે, તેમાં જરૂરી ફેરફારો કરી શકે છે કે ઉમેરી શકે છે અને અંતમાં ચેકઆઉટ (checkout) સમયે ઉત્પાદનનો ઑર્ડર આપી શકે છે. ઉપયોગકર્તાએ વસ્તુ મેળવવા માટે પોતાના સ્થળની વિગતો (shipping detail) આપવી જરૂરી છે. ખરીદીના બિલની ચુકવણી પણ ઇન્ટરનેટ દ્વારા કરી શકાય છે.

આજે ઇન્ટરનેટ પરથી કરિયાણું, રમકડાં, કમ્પ્યૂટરનાં સાધનો, મોબાઇલ અને અન્ય ઘણાં ઉત્પાદનો ખરીદી શકાય છે. કેટલીક વેબસાઇટ હવાઈ ટિકિટ સાથે પર્યટન-યોજનાઓ પૂરી પાડે છે. હવાઈ કે રેલવે-ટિકિટ હવે સરળતાથી ઑનલાઇન ખરીદી શકાય છે. ઉદાહરણ તરીકે, ભારતીય રેલવેની વેબસાઇટ www.irctc.co.in વિવિધ ટ્રેન અંગેની માહિતી પૂરી પાડે છે તથા ટિકિટની ઑનલાઇન નોંધણી અને ચુકવણી પણ થઈ શકે છે. ટિકિટની નોંધણી થઈ ગયા બાદ તેની ઇ-કૉપી (E-copy) ગ્રાહકના ઇ-મેઇલ પર તથા તેના મોબાઇલમાં SMS દ્વારા મોકલવામાં આવે છે. રેલવે-ટિકિટની ઑનલાઇન નોંધણી આકૃતિ 4.4માં દર્શાવી છે.



આકૃતિ 4.4 : રેલવે-ટિકિટની ઑનલાઇન નોંધણીની સુવિધા

માર્કેટિંગ અને વેચાણ માટેની કેટલીક અન્ય પ્રચલિત સાઇટની યાદી આ મુજબ છે :

- www.homeshop18.com
- www.flipkart.com
- www.myntra.com
- www.makemytrip.com

ઑનલાઇન બિલિંગ (Online Billing)

ઑનલાઇન બિલિંગમાં સંસ્થાઓ પોતાના બિલ ઇ-મેઇલ દ્વારા મોકલે છે. બિલ મેળવ્યા બાદ ગ્રાહક સંસ્થાની વેબસાઇટનો ઉપયોગ કરી ક્રેડિટકાર્ડ કે નેટબૅંકિંગની સુવિધા દ્વારા ઑનલાઇન ચુકવણી કરી શકે છે. જે સંસ્થાઓ બહોળી સંખ્યાના તેમના ગ્રાહકોને સમયાંતરે બિલ મોકલતી હોય, તે આ સુવિધાનો ઉપયોગ કરી શકે છે. ઉદાહરણ તરીકે, BSNL તેના ગ્રાહકોને ઑનલાઇન બિલ મોકલે છે અને તેના ગ્રાહકો ઇન્ટરનેટનો ઉપયોગ કરી ઑનલાઇન ચુકવણી પણ કરી શકે છે.

માહિતીસેવા (Information Services)

ઘણી સંસ્થાઓ પોતાના કર્મચારીઓ કે સભ્યોને અદ્યતન માહિતી પૂરી પાડવા માટે ઇન્ટરનેટનો ઉપયોગ કરે છે. તેમાં શૈક્ષણિક સંસ્થાઓ અને યુનિવર્સિટીઓનો સમાવેશ થાય છે, જે પરીક્ષાનાં પરિણામ, પ્રવેશફૉર્મ, પરીક્ષાનો કાર્યક્રમ, બેઠક-વ્યવસ્થા અને અગત્યની સૂચનાઓ ઇન્ટરનેટ દ્વારા પૂરી પાડે છે. વિદ્યાર્થીઓ કોઈ પણ સ્થળેથી પોતાનું પરિણામ જાણી શકે છે. સંસ્થાઓ કે બૅન્કો દ્વારા ગ્રાહકોને મોકલવામાં આવતી સૂચનાઓ અને સ્મરણપત્રો (reminders) માહિતીસેવાનાં અન્ય ઉદાહરણ છે.

અનેક સંસ્થાઓ વેબસાઇટ પર વિવિધ ફૉર્મ પૂરાં પાડે છે, જેને ગ્રાહક સરળતાથી ડાઉનલોડ કરી ઉપયોગમાં લઈ શકે છે. ઑનલાઇન ફૉર્મ ભરાવીને સંસ્થાઓ પોતાના ગ્રાહકોની માહિતી કે વિગતો પણ મેળવી શકે છે.

સહાયસેવાઓ (Support Services)

છેલ્લા દાયકામાં થયેલા વિશાળ તક્નિકી ફેરફારોને કારણે સહાયસેવાઓનું મહત્ત્વ વધવા પામ્યું છે. આજે સરળતમ ઇલેક્ટ્રૉનિક ઉત્પાદનમાં પણ જ્યારે ક્ષતિ ઉદ્ભવે, ત્યારે તેના નિવારણ અને જાળવણી માટે વિશિષ્ટ જ્ઞાન અને તક્નિકી ક્ષમતાની

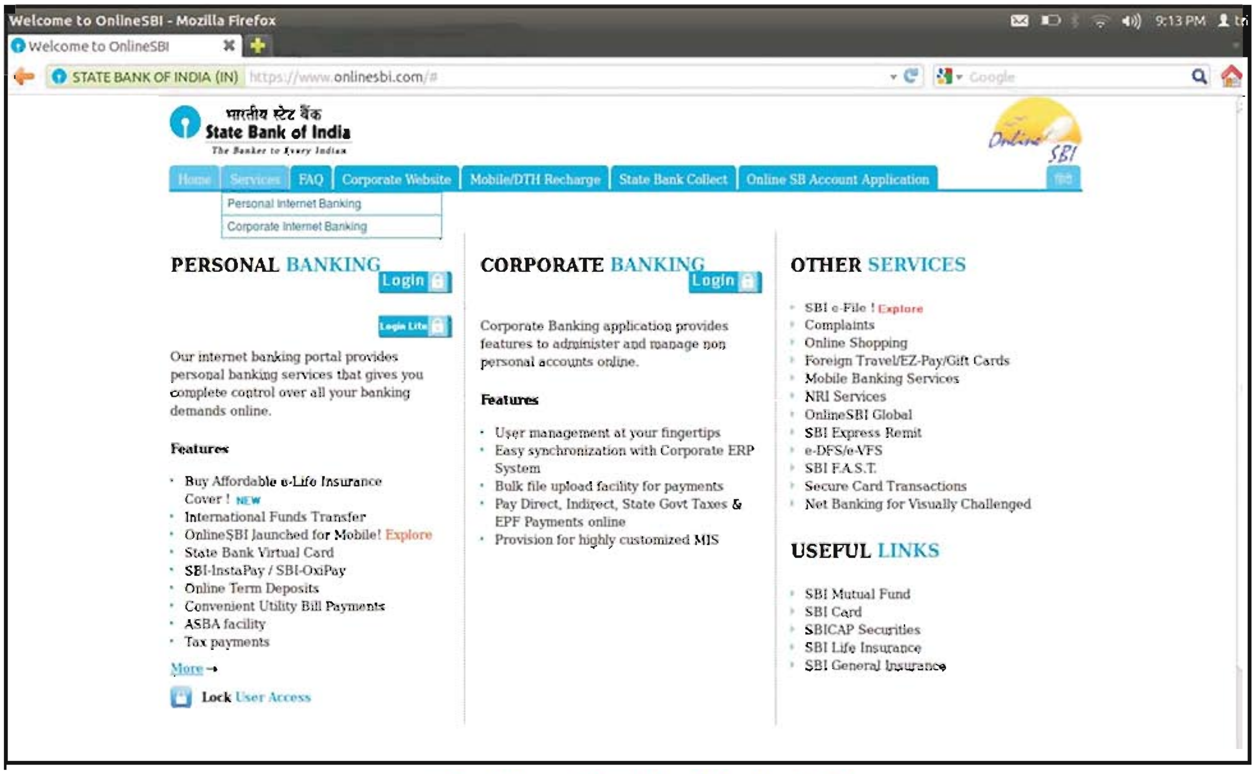
જરૂર પડે છે. ઉત્પાદનના વેચાણ પછી સંસ્થા ગ્રાહકોને ઑનલાઇન સહાય પૂરી પાડે છે. ઉદાહરણ તરીકે, ઇલેક્ટ્રૉનિક ઉત્પાદનનું વેચાણ કરતી સંસ્થા તેના ગ્રાહકોને ફરિયાદની ઑનલાઇન નોંધણી કરવાની સુવિધા પૂરી પાડે છે, જે ત્યાર પછી સહાયક ઇજનેરને મોકલવામાં આવે છે. ગ્રાહકો ઑનલાઇન મૂકેલી ફરિયાદની સ્થિતિ પણ જાણી શકે છે. સૉફ્ટવેર સંસ્થાઓ સૉફ્ટવેરનાં સ્થાપન, સંરચના કે ઉપયોગ સંબંધી કોઈ પણ મુશ્કેલી સંદર્ભે ઑનલાઇન સહાય પૂરી પાડે છે. સૉફ્ટવેર-વિક્રેતા પોતાના લાઇસન્સ-ધારક ગ્રાહકને સૉફ્ટવેરના અદ્યતન સુધારા (updates) ડાઉનલોડ કરવાની સુવિધા પૂરી પાડે છે. હાર્ડવેર-વિક્રેતા તેમનાં સાધનો માટે સૉફ્ટવેર-ડ્રાઇવર પૂરાં પાડે છે. આ ડ્રાઇવરને ગ્રાહક તેના ઉત્પાદનના પ્રકાર અને મૉડેલ પસંદ કરી ડાઉનલોડ કરી શકે છે.

નેટબૅંકિંગ (Net Banking)

આધુનિક યુગમાં નેટબૅંકિંગ કે ઇલેક્ટ્રૉનિક બૅંકિંગ ઘણું પ્રચલિત થઈ રહ્યું છે. ક્યારેક એવું પણ બને કે કોઈક કારણસર ગ્રાહક બૅન્કમાં ગયા વિના તાત્કાલિક ચુકવણી કરવા ઇચ્છતો હોય કે તેના ખાતાની સિલક જાણવા ઇચ્છતો હોય. આના ઉકેલ માટે ઑનલાઇન બૅંકિંગ મદદરૂપ બની શકે છે. ઇન્ટરનેટ દ્વારા બૅન્કના વ્યવહારોનો અમલ કરવાની પ્રક્રિયાને ઑનલાઇન બૅંકિંગ કહે છે. આજે મોટા ભાગની પ્રતિષ્ઠિત બૅન્કોએ પોતાના ગ્રાહકોને ઑનલાઇન બૅંકિંગની સુવિધા પૂરી પાડવાનું શરૂ કર્યું છે. ઑનલાઇન બૅંકિંગની મદદથી ગ્રાહકોને નીચે જણાવેલ સેવાઓ પૂરી પાડવામાં આવે છે :

- કોઈ પણ સમયે ખાતાની સિલકની જાણકારી
- એક ખાતામાંથી અન્ય ખાતામાં રકમની લેવડદેવડ
- કોઈ પણ આવક કે જાવક માટેનાં પત્રક (statement)
- નાણાકીય વ્યવહારની સ્થિતિ વિશે જાણકારી
- બૅન્કમાં ગયા વગર ટેલિફોન, ઇલેક્ટ્રિસિટી અને અન્ય બિલની ચુકવણી

ગ્રાહકને ઑનલાઇન સેવાઓ માટે પાસવર્ડ આપવામાં આવે છે, જેના દ્વારા કમ્પ્યૂટર કે મોબાઇલની મદદથી તે બૅન્કની સાઇટ પર લૉગ-ઇન કરી બૅન્ક સાથેની તમામ પ્રવૃત્તિઓ કરી શકે છે. આકૃતિ 4.5માં સ્ટેટ બૅન્ક ઑફ ઇન્ડિયાની વેબસાઇટનું હોમપેજ દર્શાવ્યું છે. આ માટે www.onlinesbi.com એ URLનો ઉપયોગ કરવામાં આવે છે.



આકૃતિ 4.5 : ઑનલાઇન બૅંકિંગની સુવિધા

આજે મોટા ભાગની બૅન્કો ઑનલાઇન બૅંકિંગ પૂરું પાડે છે; તેમાંની કેટલીક બૅન્કોની વેબસાઇટ નીચે આપેલ છે :

- www.centralbankofindia.co.in
- www.bankofbaroda.co.in
- www.iob.in
- www.pnbindia.in
- www.denabank.co.in

પરંપરાગત વ્યવસાય વિરુદ્ધ ઈ-કૉમર્સ (Traditional Commerce Vs. E-commerce)

ઇન્ટરનેટના વિકાસને કારણે ઇ-કૉમર્સનો ઉપયોગ વધવાથી પરંપરાગત વ્યવસાયોની સ્પર્ધામાં નોંધપાત્ર પરિવર્તન આવ્યું છે. પરંપરાગત વ્યવસાયમાં સંસ્થાએ કોઈ એક જ ઉદ્યોગ સાથે અને તે પણ સીમિત ભૌગોલિક વિસ્તારમાં જ સ્પર્ધા કરવાની રહે છે. ઘણા કિસ્સાઓમાં વ્યાવસાયિક પ્રક્રિયાઓમાં પરંપરાગત પ્રવૃત્તિઓનો ઉપયોગ ખૂબ જ અસરકારક હોય છે અને તેને ટેક્નોલૉજી દ્વારા વધુ સારો બનાવી શકાતો નથી. જે ઉત્પાદનોને ગ્રાહક સ્પર્શ કરીને, સૂંઘીને કે તપાસીને લેવા ઇચ્છતો હોય, તેનું ઇ-કૉમર્સની મદદથી વેચાણ કરવું મુશ્કેલ છે. ઉદાહરણ તરીકે, ખરીદદારો નાશવંત ખાદ્યપદાર્થો, કિંમતી ઘરેણાં અને ઉચ્ચ બનાવટનાં કપડાં ઑનલાઇન ખરીદતા અચકાશે, કારણકે આ વસ્તુઓને ખરીદતાં પહેલાં બરાબર તપાસી શકાશે નહીં. પરંપરાગત વ્યવસાયમાં છૂટક વ્યાપારીઓ તેમના વર્ષોના અનુભવ પરથી દુકાનની એવી રૂપરેખા અમલમાં મૂકે છે, જે ગ્રાહકને પોતાની વસ્તુ ખરીદવા માટે અનુકૂળતા પૂરી પાડે. ઉત્પાદનોની ગોઠવણ, દુકાનની રૂપરેખા અને ડિઝાઇનને જ વ્યાપાર કહે છે. વ્યાપારીઓમાં ગ્રાહકની જરૂરિયાતોને પારખી શકવાનું કૌશલ્ય હોય છે અને તેઓ જરૂરિયાત અનુસાર ઉત્પાદનો અને સેવાઓ શોધી કાઢી શકે છે. પરંતુ વ્યાપારની આ કળા ઇન્ટરનેટ પર અમલમાં મૂકવી મુશ્કેલ છે. પરંપરાગત વ્યાપારની થોડી સામાન્ય લાક્ષણિકતાઓ નીચે મુજબ છે :

- નિશ્ચિત સમયગાળા દરમિયાન અથવા વ્યવસાયના કલાકો દરમિયાન વ્યાપાર કરવામાં આવે છે.
- હરીફો સમક્ષ માહિતી વહેંચવી પડતી નથી.
- વેચાણ અધિકારી, વેચાણ વહીવટકાર અને તેવી અન્ય વ્યક્તિઓની નિમણૂંક.
- જગ્યાનું ભાડું, માલની ખરીદી, જાહેરાત, માલ-યાદી (inventory), માલ મોકલવાની વ્યવસ્થા વગેરે.

જોકે, આધુનિક યુગની ગતિશીલતાને જોતાં આ પરંપરાગત નિયમો તોડવા ઘણા જરૂરી બને છે. ઇન્ટરનેટનો ઉપયોગ કરી દુનિયાભરમાં વ્યવસાય કરી શકાય તે માટેના નવા માર્ગ અપનાવવા જરૂરી છે. ઇ-કૉમર્સની કેટલીક લાક્ષણિકતાઓ નીચે દર્શાવી છે :

- ઉત્પાદનની જાહેરાત ઇલેક્ટ્રોનિક સંસાધનો દ્વારા થાય છે.
- ઉત્પાદનની સૂચિ અને ઉપલબ્ધ યોજનાઓ વિષે ગ્રાહકને જાણકારી આપી શકાય છે.
- ચુકવણીની ઇ-પેમેન્ટ પદ્ધતિઓનો ઉપયોગ કરી શકાય છે. તદુપરાંત માલ મેળવતી વખતે કરાતી ચુકવણીની પદ્ધતિનો વિકલ્પ પણ ઉપલબ્ધ હોય છે.
- ગ્રાહકને થોડા દિવસમાં માલ પહોંચાડવામાં આવે છે.
- વ્યવહારનો સરેરાશ ખર્ચ ઘટે છે.
- સંપૂર્ણ વ્યવહાર પૂરો કરવા માટેનો સમય ઘટે છે.

ભારતમાં ઈ-કૉમર્સ (E-commerce in India)

ભારતમાં ઇન્ટરનેટના વપરાશકાર ઉત્તરોત્તર વધતા રહ્યા છે. ઇ-કૉમર્સની વૃદ્ધિ પણ નોંધપાત્ર રીતે અનુભવાઈ રહી છે અને લોકોની વ્યવહાર કરવાની પદ્ધતિ સફળતાપૂર્વક બદલાઈ રહી છે. ભારતીય ઑનલાઇન ખરીદદારો માટે ઉચ્ચકક્ષાનાં સાધનોથી લઈને જૂનાં (second hand) પુસ્તકો સુધીની તમામ વસ્તુઓ હવે એક બટન ક્લિક કરવાથી મેળવી શકાય તે રીતે ઉપલબ્ધ છે. લાખો લોકો માટે તે નવી ક્ષિતિજો અને તક ખોલી આપે છે.

ઉત્પાદન અને સેવાની વિવિધતા ઇન્ટરનેટ પરની ઑનલાઇન ખરીદીને આકર્ષક અને અનુકૂળ બનાવે છે. ઇન્ટરનેટ ભારતના ગ્રામ્ય વિસ્તારમાં પણ પહોંચી ચૂક્યું હોવાથી ત્યાંના લોકોની જીવનપદ્ધતિ, વપરાશની ગુણવત્તા અને જથ્થો તથા વિચારધારામાં આમૂલ પરિવર્તન આવ્યું છે.

ભારતમાં ઇ-કૉમર્સની વૃદ્ધિમાં ઘણાં પરિબળો ભાગ ભજવે છે. તેમાંનાં કેટલાંક મુખ્ય પરિબળો નીચે આપ્યાં છે :

- ઇન્ટરનેટનું જોડાણ, બ્રૉડબૅન્ડ અને ત્રીજી પેઢી (third generation - 3G)ની સેવાઓ, લેપટૉપ, સ્માર્ટફોન, ટેબ્લેટ અને ડોંગલ જેવી તક્નિકી સરળતામાં થયેલી વૃદ્ધિ.
- મોબાઇલ સાધનોના ઉપયોગમાં થયેલો વધારો.
- વધુ વ્યાપક ઉત્પાદનોના વિસ્તારની ઉપલબ્ધતા.
- પરંપરાગત ખરીદી માટે વ્યસ્ત જીવન, વાહન-વ્યવહારની ગીચતા અને સમયનો અભાવ.
- મધ્યસ્થીઓ દ્વારા ચલાવવામાં આવતા છૂટક વેચાણમાં થતા માલસંગ્રહ અને સ્થાયી મિલકતમાં ઘટાડો થવાથી પરંપરાગત પદ્ધતિ કરતા ઓછી કિંમતે વેચાણ.
- ઑનલાઇન વર્ગીકૃત સાઇટના ઉપયોગમાં થયેલા વધારાને કારણે વધુ ને વધુ ગ્રાહકો દ્વારા જૂના માલનું ખરીદ-વેચાણ.
- eBay, Flipkart, Snapdealને તેના જેવી અનેક સાઇટ સાથે ઑનલાઇન માર્કેટ-જગ્યાનો થયેલો વિકાસ.

અન્ય દેશોની તુલનામાં ભારતમાં ઇ-કૉમર્સનો પ્રવેશ અપેક્ષાકૃત મોડો થયો હોવા છતાં, તેમાં નવા પ્રવેશોની વિપુલ સંખ્યાને કારણે હવે તેનો અભૂતપૂર્વ ગતિથી વિકાસ થઈ રહ્યો છે. Flipkart, eBay India, Snapdeal, Amazon India, Myntra, Domino, PayTM, Jabong અને તેના જેવી અનેક ઑનલાઇન દુકાનો ભારતમાં પ્રચલિત બની રહી છે.

છૂટક વેપારીઓએ માલ ખરીદતી વખતે તત્કાલ ચુકવણીની સુવિધા (cash-on-delivery) પણ શરૂ કરી છે, તે ભારતમાં ચુકવણી માટેની સૌથી વધુ લોકપ્રિય પદ્ધતિ છે. લગભગ 80 % ભારતીય ઇ-કૉમર્સ વ્યવસાય માલ મેળવતી વખતે તત્કાલ ચુકવણીની સુવિધાનો ઉપયોગ કરે છે. જોકે, ઑનલાઇન છૂટક વેપારીની દષ્ટિએ જોતાં આ એક ખર્ચાળ દરખાસ્ત છે, કારણકે જ્યાં સુધી માલ મોકલવામાં ન આવે, ત્યાં સુધી તેમણે વેચાણ માટેનાં નાણાંની જોગવાઈ કરી રાખવી પડે છે. ગ્રાહક માટે તત્કાલ ચુકવણીની સુવિધાથી માલનો અસ્વીકાર કરવાનું સરળ બને છે.

ઉદ્યોગ-નિષ્ણાતો ભારતમાં ઇ-કૉમર્સ ઉદ્યોગના સંભવિત ભવિષ્ય માટે ઘણા આશાવાદી છે. તેમાં ઘણા પડકારોનો સામનો કરવાનો છે, પરંતુ સુનિયોજિત અને પૂર્વનિશ્ચિત યોજનાઓ દ્વારા ભારતમાં ઇ-કૉમર્સ સરળતાપૂર્વક વિકાસ સાધશે, તેમ માનવામાં આવે છે.

ઇ-કૉમર્સના ફાયદા (Advantages of E-commerce)

ઇન્ટરનેટના વિકાસને લીધે સંસ્થાઓ પોતાના વ્યવસાયના પ્રસાર માટે અનેક નવીન અને આકર્ષક યોજનાઓ શોધી રહી છે. આજે એવી સફળ સંસ્થા શોધવી મુશ્કેલ છે, જે પોતાની પ્રવૃત્તિઓ માટે કમ્પ્યૂટરનો ઉપયોગ કરતી ન હોય. ઇ-કૉમર્સ ગ્રાહકોને સસ્તા ભાવે માલની ઉપલબ્ધતા, વિશાળ પસંદગી અને સમયના બચાવ જેવા અનેક લાભ પૂરા પાડે છે. વધુ ઝડપી બ્રૉડબૅન્ડ સેવાઓ અને નવા વિનિયોગોની ઉપલબ્ધતાથી આવનારા વર્ષોમાં ઇ-કૉમર્સના વેચાણમાં વધારો થશે. ઇ-કૉમર્સના કેટલાક ફાયદા નીચે જણાવ્યા છે :

- **અવિરત સમય (24×7) માટેની વ્યાપારવ્યવસ્થા (Conduct Business 24×7)**

  ઇ-કૉમર્સની મદદથી કોઈ પણ સમયે અને સ્થળે વ્યવસાય કરી શકાય છે. વ્યાવસાયિક પ્રવૃત્તિઓને સમયનું કોઈ બંધન રહેતું નથી. દિવસમાં કોઈ પણ સમયે ગ્રાહક ઉત્પાદન ખરીદી શકે છે અને કોઈ પણ સમયે ઑર્ડર પણ સ્વીકારવામાં આવે છે.

- **ઓછો ખર્ચ (Lower Cost)**

  ઈન્ટરનેટના વ્યાવસાયિક ઉપયોગથી માર્કેટિંગ, વહેંચણી, ફોન, ટપાલ, છાપકામ અને તેના જેવી અનેક પ્રવૃત્તિઓમાં થતો ખર્ચ ઘટે છે. સ્થાનિક વ્યવસાયમાં કરાતા મૂડીરોકાણની તુલનામાં ઇ-કૉમર્સનું મૂડીરોકાણ અલ્પતમ હોય છે. ઈન્ટરનેટ દ્વારા કરવામાં આવતા વ્યવસાય કરતાં સ્થાનિક વ્યવસાય/દુકાન શરૂ કરવાનો ખર્ચ ઘણો વધુ આવે છે. ઉત્પાદનનાં વેચાણ માટે વિતરકો કે અન્ય મધ્યસ્થીઓની ગેરહાજરીથી ગ્રાહકોને ઓછા ખર્ચે ઉત્પાદનની વિશાળ શ્રેણી મેળવવાનો લાભ મળે છે.

- **સરહદ કે ભૌગોલિક મર્યાદાનો અભાવ (No Boundaries or Geographical Limitations)**

  સ્થાનિક દુકાનની સીમા નિશ્ચિત ક્ષેત્ર સુધી વિસ્તરેલી હોય છે. ઇ-કૉમર્સની મદદથી વ્યવસાયને તત્કાલ અસંખ્ય ગ્રાહકો સુધી પહોંચાડી શકાય છે. પરંપરાગત માર્કેટિંગની કોઈ પણ યોજના દ્વારા આ શક્ય નથી. નવા બજારની રચના સહિત સંભવિત ખરીદદારો સુધી સરળતાથી પહોંચી શકવાની ક્ષમતા તે ધરાવે છે. વ્યવસાયની વેબસાઇટ બનાવી તેને ઈન્ટરનેટ પર અપલોડ કરવાથી વૈશ્વિક વિસ્તાર સુધી પહોંચવું સરળ બને છે.

- **ઉન્નત અને વધુ સારી ગ્રાહકસેવા (Improved and Better Customer Service)**

  ગ્રાહક સાથે પ્રત્યક્ષ સંવાદ શક્ય હોવાને કારણે ઉત્પાદનની કિંમત, જથ્થા, વિશેષતાઓ અને અન્ય પ્રશ્નોનો ઉકેલ સરળતાથી મેળવી શકાય છે, જે વધુ સારી ગ્રાહક સેવા પૂરી પાડે છે. જે સંસ્થાઓ અન્ય સંસ્થાઓ સાથે મળીને વ્યવસાય કરે છે, તેમના માટે ઑનલાઇન ગ્રાહકસેવા ઉમેરવી એ એક સ્પર્ધાત્મક લાભ છે. ઑનલાઇન ગ્રાહકસેવાને ગ્રાહક વધુ પસંદ કરે છે. ઇ-કૉમર્સ વધુ સારી અને ત્વરિત ગ્રાહકસેવા પૂરી પાડે છે. ઈન્ટરનેટ પર માહિતીનો પ્રસાર ઘણી સરળતાથી થઈ શકે છે. વ્યાપારી તેના નવા ઉત્પાદનની જાણ ગ્રાહકને ઇ-મેઇલ દ્વારા કરી શકે છે અને ઉત્પાદન સંબંધી મુશ્કેલીઓનું નિવારણ પણ લાવી શકે છે.

- **જૂથકાર્ય (Teamwork)**

  ઇ-કૉમર્સની મદદથી એકાધિક સંસ્થાઓ સાથે મળી કાર્ય કરી શકે છે. આનું એક ઉદાહરણ ઇ-મેઇલ છે, જેની મદદથી લોકો માહિતીનો વિનિમય કરે છે. તેના દ્વારા વિતરકો, વ્યાપારીઓ, ધંધાકીય ભાગીદારો અને ગ્રાહકો સાથેના સંસ્થાના સંવાદની પદ્ધતિમાં સુધારો થયો છે. વધુ સંવાદ વધુ સારું પરિણામ આપે છે.

- **પરિવહનના સમય અને ખર્ચમાં ઘટાડો (Eliminate Travel Time and Cost)**

  ઇચ્છિત દુકાન સુધી પહોંચવા માટે ગ્રાહકે અહીં લાંબું અંતર કાપવાનું રહેતું નથી. ઇ-કૉમર્સની મદદથી આ દુકાનની મુલાકાત માઉસની કેટલીક ક્લિક કરવાથી લઈ શકાય છે.

- **ઝડપ (Speed)**

  પરંપરાગત પદ્ધતિઓની તુલનામાં ઇ-કૉમર્સ વ્યવસાય ઘણો ઝડપી છે. ઑનલાઇન વ્યવહારોને કારણે સંચારમાં થતો સમયનો વ્યય નાબૂદ થાય છે, જે વ્યવસાયની ઝડપ નોંધપાત્ર રીતે વધારે છે. વ્યાવસાયિક સંસ્થાઓ વિતરકને ખરીદીના ઑર્ડર ઑનલાઇન બનાવીને મોકલી શકે છે, જે સમયનો વ્યય અટકાવે છે. વિતરકો તત્કાલ ઑર્ડર મેળવી તેના પર કાર્ય શરૂ કરી શકે છે, આથી ઑર્ડર મોકલવાનો સમય પણ બચે છે. ઈન્ટરનેટ પર આપવામાં આપેલ માહિતીને અવારનવાર બદલી શકાય છે. આ સુવિધાની મદદથી સંસ્થાનાં ઉત્પાદનો કે સેવામાં થતા ફેરફાર વિશે ગ્રાહકને જાણકારી આપી શકાય છે.

ઇ-કૉમર્સ સમાજને નીચે જણાવેલ લાભ પણ પૂરા પાડે છે :

- ઘર, કાર્યાલય કે કોઈ પણ સ્થળેથી ખરીદી.
- ઉત્પાદનની ખરીદી માટે પરિવહન ઓછું થતું હોવાથી યાતાયાત અને પ્રદૂષણમાં ઘટાડો.
- આરોગ્યલક્ષી સેવાઓ.
- ડિસ્ટન્સલર્નિંગ અને અભ્યાસ.

ઈ-કૉમર્સની મર્યાદાઓ (Limitations of E-commerce)

ઈ-કૉમર્સના ફાયદાઓની યાદી બનાવી શકાય તેમ હોવા છતાં તેમાં કેટલીક મર્યાદાઓ અને ગેરફાયદા પણ રહેલાં છે. જોકે, સમયાંતરે ઘણી મર્યાદાઓ લુપ્ત થતી જાય છે. ઈ-કૉમર્સની કેટલીક મર્યાદાઓ નીચે દર્શાવી છે :

- **પરિવર્તનનો પ્રતિકાર (Resistance to Change)**

  વ્યાવસાયિક સંસ્થાઓને પરંપરાગત વ્યવસાયમાંથી ઈ-કૉમર્સ વ્યવસાય તરફ અભિમુખ થવું પડે છે, જેમાં તેમને ઘણો પ્રતિકાર સહન કરવો પડે છે. લોકોને પણ વ્યવસાયના કાગળરહિત અને પ્રત્યક્ષ ન હોય તેવા વ્યવહારોની આદત પાડવી પડે છે.

- **પ્રારંભિક ખર્ચ (Initial Cost)**

  ઈ-કૉમર્સની નવી તક્નિકો માટે નોંધપાત્ર પ્રારંભિક મૂડીરોકાણ જરૂરી છે, જે સંસ્થાની મૂળભૂત વ્યાવસાયિક પદ્ધતિઓ બદલી નાંખે છે. હાર્ડવેર અને સૉફ્ટવેરની તક્નિકોમાં મૂડીરોકાણ જરૂરી બને છે. ઈ-કૉમર્સની મદદથી કરવામાં આવતા વ્યવસાયને સમજવા માટે કર્મચારીઓને તાલીમની પણ જરૂર પડે છે.

- **સુરક્ષા (Security)**

  ઈ-કૉમર્સમાં મૂળભૂત રીતે ધ્યાન પર લેવા જેવી બાબત સુરક્ષાની છે. સંસ્થા કે વ્યક્તિની સંબંધિત માહિતીનું જ્યારે ઈન્ટરનેટ પર પ્રસારણ થતું હોય, ત્યારે અનધિકૃત ઉપયોગથી તેમનું રક્ષણ કરવું જરૂરી બને છે. ઈન્ટરનેટ વૈશ્વિક ઉપયોગ પૂરો પાડે છે. પરંતુ સંસ્થાઓએ દૂષિત (malicious) ઉપયોગ કે અકસ્માતે થતા દુરુપયોગ સામે પોતાની માહિતીને સુરક્ષા પૂરી પાડવી જોઈએ.

- **ગોપનીયતા (Privacy)**

  ગ્રાહકની ગોપનીયતા એ ઘણો ગંભીર મુદ્દો છે. દુરુપયોગ થવાના ભયને કારણે ગ્રાહક પોતાની અંગત માહિતી ઈન્ટરનેટ પર દર્શાવતા અચકાય છે. કેટલીક સંસ્થાઓ અન્ય માર્કેટિંગ સંસ્થાઓને તેમના ડેટાબેઝની માહિતી વેચે છે, જેનો દુરુપયોગ કરી ગ્રાહકોને બિનજરૂરી (spam) મેઇલ મોકલવામાં આવે છે. હેકર પણ ઈન્ટરનેટ પરથી માહિતીને આંતરીને તેનો દુરુપયોગ કરી શકે છે. ક્રેડિટકાર્ડની છેતરપિંડી ગ્રાહકને આર્થિક નુકસાન પહોંચાડે છે. આમ, હેકર, વાઇરસ, વિગતોનું સ્થાનાંતરણ અને વ્યવહારનાં જોખમો સામે ઈ-કૉમર્સને સુરક્ષા પૂરી પાડવી જરૂરી બને છે.

- **વિશ્વાસનો અભાવ (Lack of Trust)**

  કેટલીક વાર, ઉત્પાદન ન પહોંચાડવાની છેતરપિંડી, ઉત્પાદન અંગેની ભ્રામક રજૂઆત, ચુકવણીની સુરક્ષાનો અભાવ વગેરે કારણોસર ગ્રાહકોમાં ઈ-કૉમર્સ પ્રત્યે વિશ્વાસનો અભાવ જોવા મળે છે. તદુપરાંત ઑનલાઇન ખરીદીમાં ક્ષતિગ્રસ્ત માલને પરત કરવાનું પણ મુશ્કેલ છે. પરત કરેલ માલનો વહનખર્ચ કોણ ભોગવશે, વેપારી કિંમત પરત કરશે કે નહીં અને ઉત્પાદન તેના મૂળ સ્રોત સુધી પરત થશે કે નહીં જેવા પ્રશ્નો સામાન્ય રીતે અહીં ઉદ્ભવતા હોય છે.

- **ઉત્પાદનને પહોંચાડવાનો સમય (Time for Delivery of Products)**

  સ્થાનિક દુકાનમાં આપણે ઉત્પાદનની ખરીદી કરી તેને સાથે લઈ જઈ શકીએ છીએ. ઈ-કૉમર્સનો ઉપયોગ કરી આપણે એવાં ઉત્પાદનો ખરીદતાં હોઈએ છીએ કે સ્થાનિક રીતે ઉપલબ્ધ નથી હોતાં. એનો અર્થ એ થાય કે ઉત્પાદનને મોકલવામાં સમય લાગશે અને ખર્ચ થશે. દૂરના સ્થળેથી ઉત્પાદનનો ઑર્ડર આપવામાં આવ્યો હોય, તો તેને મોકલવાનો ખર્ચ વધારે ચૂકવવો પડે છે.

- ફળ, શાકભાજી અને અન્ય નાશવંત ઉત્પાદનો ઑનલાઇન ખરીદવા માટે પસંદ કરવામાં આવતાં નથી. ઑનલાઇન દુકાનો ઉત્પાદનને સ્પર્શવાની, ધારણ કરવાની કે પ્રત્યક્ષ અનુભવવાની સુવિધા આપી શકે નહીં. માટે વસ્ત્રો અને ઘરેલું રાચરચીલાં જેવાં ઉત્પાદનોનું ઑનલાઇન વેચાણ યુક્તિપૂર્વક કરવું પડે છે.

- ઈ-કૉમર્સમાં ચુકવણી માટે મોટે ભાગે ક્રેડિટકાર્ડનો ઉપયોગ કરવામાં આવે છે. ક્રેડિટકાર્ડ દ્વારા ઑનલાઇન તદ્દન નાના કે બહુ મોટા નાણાકીય વ્યવહારો પસંદ કરવામાં આવતા નથી. નાની સંસ્થાઓને પોતાનો ઑનલાઇન વ્યવસાય

સ્થાપિત કરવામાં સમયાંતરે કરવા પડતા સુધારાઓ (updates) મુશ્કેલી ઊભી કરે છે. ઇ-કૉમર્સ ઇન્ટરનેટ પર આધારિત છે અને ઇન્ટરનેટના જોડાણમાં થતો અવરોધ વ્યવસાય માટે જોખમી બની રહે છે.

ઇ-કૉમર્સની વ્યાવસાયિક પ્રતિકૃતિઓ (E-commerce Business Models)

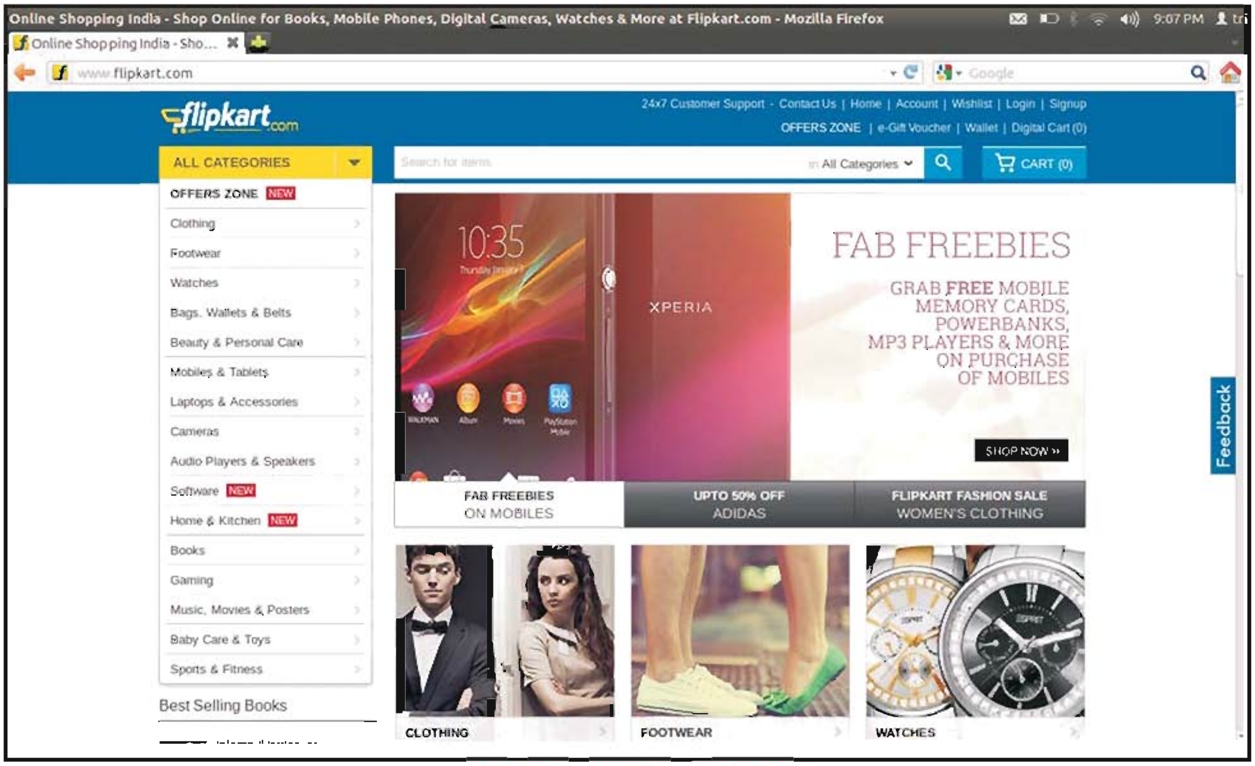
ઇ-કૉમર્સમાં સમાવિષ્ટ પક્ષ અને તેના દ્વારા પૂરી પાડવામાં આવતી પ્રવૃત્તિઓ તથા સેવાઓને આધારે ઇ-કૉમર્સની વ્યાવસાયિક પ્રતિકૃતિઓની વ્યાખ્યા કરવામાં આવે છે. ઇ-કોમર્સમાં અનેક વ્યાવસાયિક પ્રતિકૃતિઓ સૂચિત કરવામાં આવી છે, પરંતુ નીચે આપવામાં આવેલી પ્રતિકૃતિઓ સૌથી વધુ પ્રચલિત છે. સામાન્ય રીતે આ માટેનું વર્ગીકરણ 'કોણ કોને માલ વેચે છે', તેના પર આધારિત હોય છે.

- વ્યવસાયીથી ગ્રાહક (Business to Consumer - B2C)
- વ્યાવસાયીથી વ્યવસાયી (Business to Business - B2B)
- ગ્રાહકથી ગ્રાહક (Consumer to Consumer - C2C)
- ગ્રાહકથી વ્યવસાયી (Consumer to Business - C2B)

હવે, દરેકની વિસ્તૃત સમજૂતી મેળવીએ.

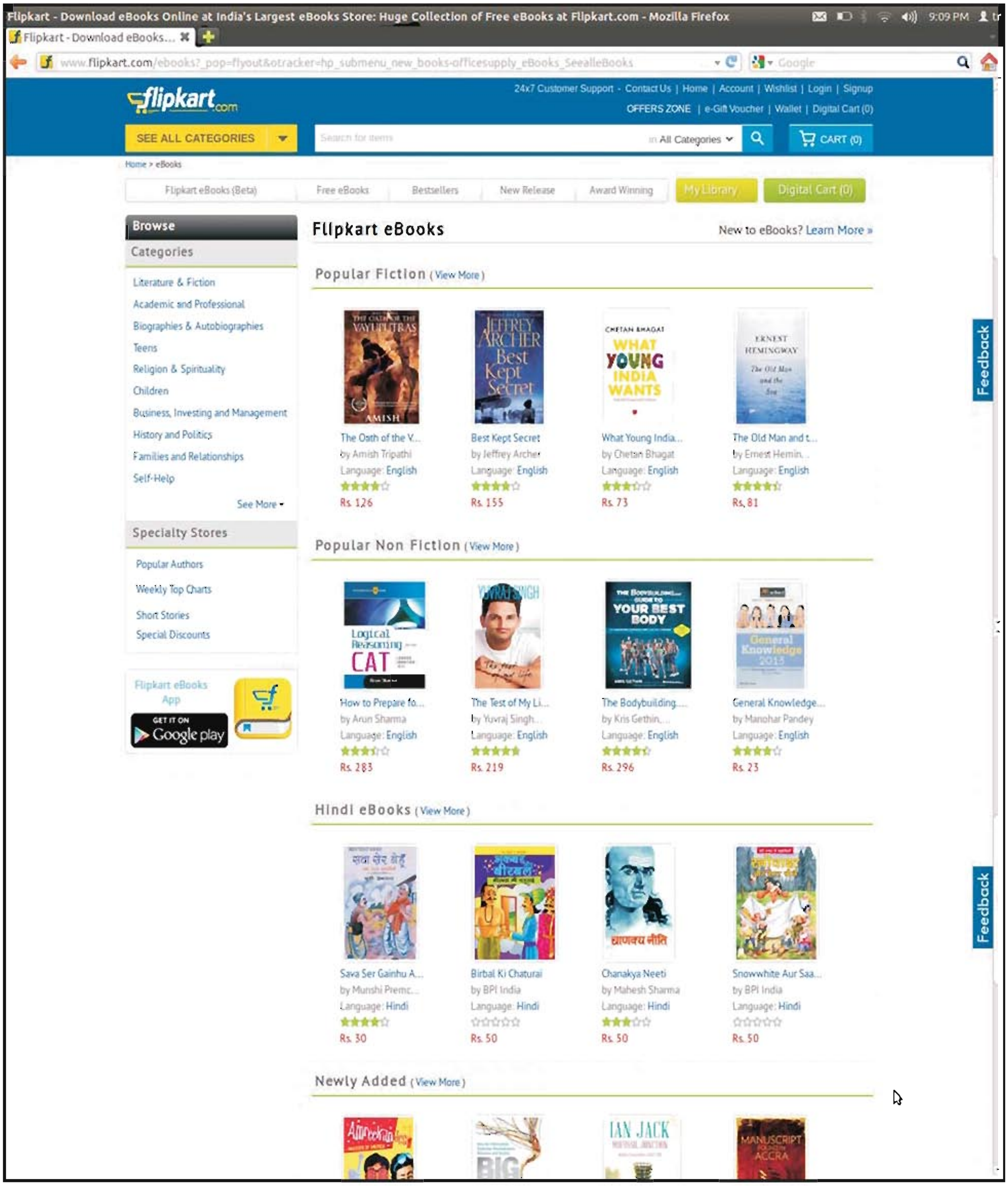
વ્યવસાયીથી ગ્રાહક (Business to Consumer)

ઇન્ટરનેટ દ્વારા વેબસાઇટના ઉપયોગથી જે વ્યાવસાયિકો કે સંસ્થાઓ ગ્રાહકોને તેમનાં ઉત્પાદનો કે સેવાઓનું વેચાણ કરે છે, તેમના માટે 'વ્યવસાયીથી ગ્રાહક' પ્રતિકૃતિનો સંદર્ભ આપી શકાય. ગ્રાહકો કોઈ પણ સ્થળેથી કોઈ પણ સમયે ઉત્પાદન કે સેવાઓ પસંદ કરી તેનો ઑર્ડર આપી શકે છે. વેચનાર પોતાના ઉત્પાદનનું વેચાણ મધ્યસ્થી વિના સીધું જ ખરીદદારને કરી શકે છે, જ્યાં ખરીદદાર એક સ્વતંત્ર ગ્રાહક છે. ઇન્ટરનેટ પર આ પ્રતિકૃતિ સૌથી વધુ જોવા મળે છે. ઉત્પાદન સરળતાથી અને ઝડપથી ખરીદી શકાતાં હોવાને કારણે આજકાલ તે ગ્રાહકોમાં વધુ પ્રચલિત છે. વસ્તુના છૂટક વેચાણ ઉપરાંત B2Cમાં ઑનલાઇન બૅંકિંગ, મકાનની લે-વેચ, પરિવહન-સુવિધાઓ અને તેના જેવી અન્ય અનેક સેવાઓનો સમાવેશ કરવામાં આવ્યો છે. amazon.com, rediff.com, fabmart.com, flipkart.com વગેરે B2C વેબસાઇટનાં કેટલાંક ઉદાહરણ છે. આકૃતિ 4.6માં Flipkart વેબસાઇટનું હોમપેજ દર્શાવ્યું છે. વિવિધ પ્રકારની વસ્તુ ખરીદવા માટે આ ઘણી પ્રચલિત વેબસાઇટ છે.



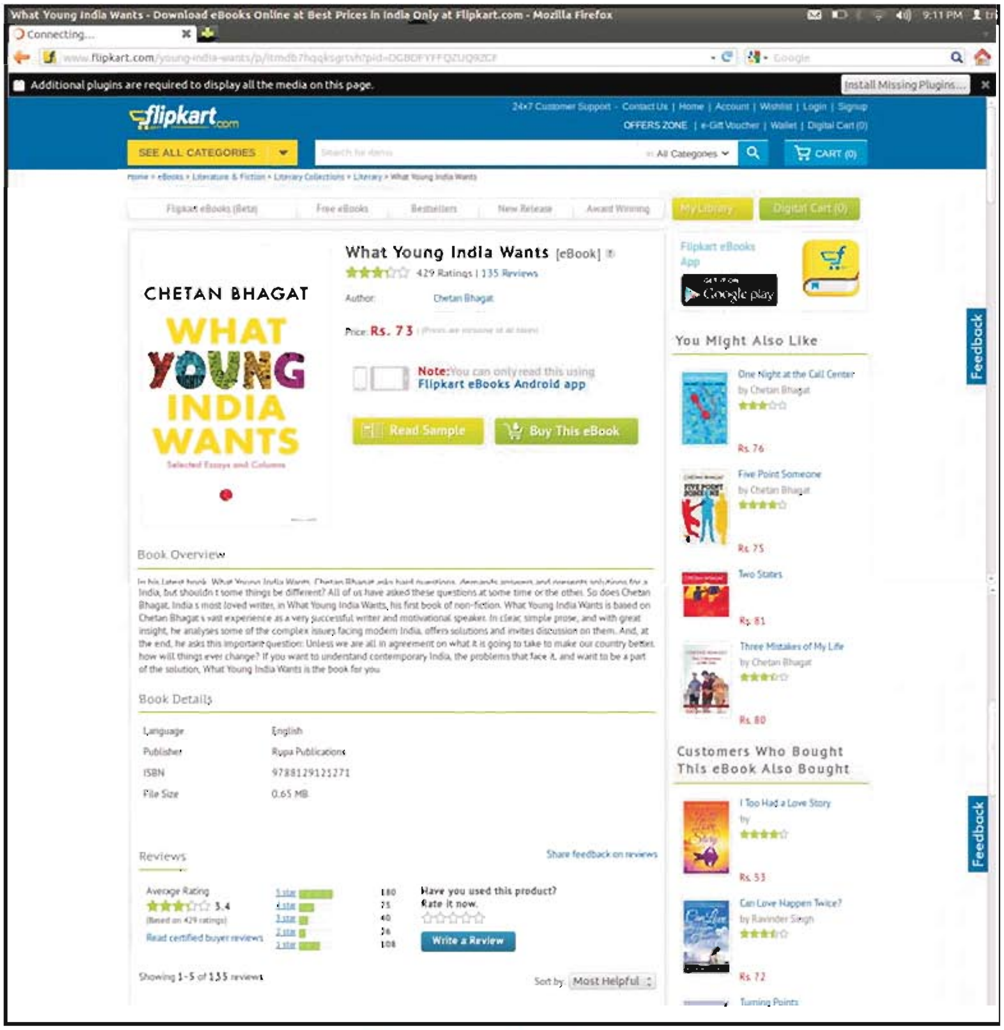
આકૃતિ 4.6 : Flipkartનું હોમપેજ

આકૃતિ 4.6માં દર્શાવ્યા મુજબ વિન્ડોની ડાબી બાજુ ઉત્પાદનોને વર્ગીકૃત સ્વરૂપે દર્શાવવામાં આવે છે. તમારી ઇચ્છા મુજબનો વર્ગ પસંદ કરો. આકૃતિ 4.7માં દર્શાવ્યા મુજબ અહીં eBooks વર્ગ પસંદ કરવામાં આવ્યો છે. તે તમામ ઇ-બુક્સને સૂચિ (category) દ્વારા પ્રદર્શિત કરે છે.



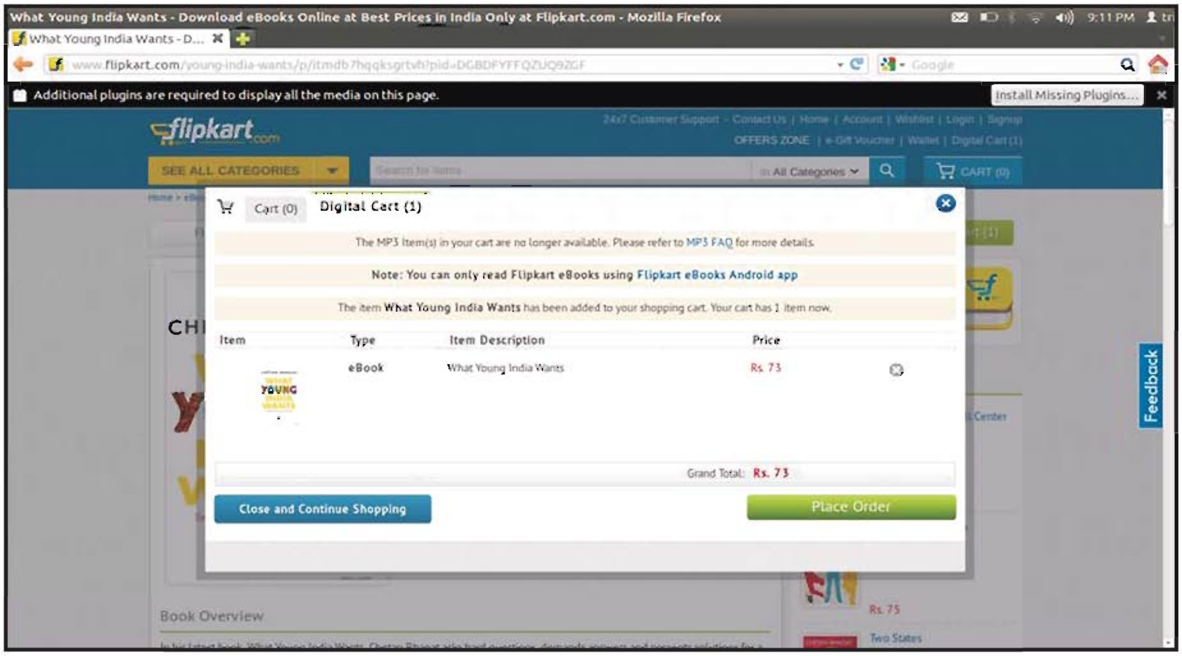
આકૃતિ 4.7 : Flipkart.comમાં eBooks માટેનો વર્ગ પસંદ કરવો

જ્યારે કોઈ પણ ઇ-બુક પર ક્લિક કરવામાં આવે છે, ત્યારે આકૃતિ 4.8માં દર્શાવ્યા મુજબ લેખકનું નામ, કિંમત, વાચકોના રેટિંગ, પુસ્તકના પ્રતિભાવ, પ્રકાશકનું નામ અને ISBN અંક અંગેની વિગતો રજૂ કરવામાં આવે છે. વિન્ડોના સૌથી ઉપરના ભાગમાં આપેલ સર્ચ-ટૂલનો ઉપયોગ કરી વિવિધ વર્ગોમાંથી નિશ્ચિત પુસ્તકને શોધી શકાય છે.



આકૃતિ 4.8 : ઈ-બુકની પસંદગી

"Buy this eBook" બટન પર ક્લિક કરવાથી તેને આકૃતિ 4.9માં દર્શાવ્યા મુજબ શૉપિંગકાર્ટમાં ઉમેરવામાં આવશે.



આકૃતિ 4.9 : પસંદ કરેલી ઈ-બુક દર્શાવતું શૉપિંગકાર્ટ

આવી જ રીતે, તમે અનેક વર્ગોમાંથી તમારી પસંદગીની ગમે તેટલી વસ્તુઓ પસંદ કરી શૉપિંગકાર્ટમાં ઉમેરી શકો છો, શૉપિંગકાર્ટમાં આપેલ ઉત્પાદન કાઢી નાંખી શકાય છે તથા તેનો જથ્થો પણ બદલી શકાય છે.

ઑર્ડરની પુષ્ટિ કરવા માટે આકૃતિ 4.9માં દર્શાવ્યા મુજબ 'Place Order' બટન પર ક્લિક કરો. આમ કરવાથી ગ્રાહકને અંતિમ સ્ક્રીન પર લઈ જવામાં આવશે અને જો લૉગ-ઇન કરેલું ન હોય, તો તે માટે પૂછવામાં આવશે. અહીં બિલિંગ માટેનું સરનામું, ઑર્ડરનો સારાંશ અને ચુકવણીનો વિકલ્પ પૂરો પાડવામાં આવશે. એકવાર આ તમામ પ્રક્રિયા પૂર્ણ થઈ જાય પછી ખરીદ-ઑર્ડર તૈયાર કરવામાં આવે છે અને ગ્રાહકે આપેલ સરનામાં પર ઇ-બુક મોકલી આપવામાં આવે છે. હાલમાં 'માલ મેળવ્યા બાદ ચુકવણી'નો વિકલ્પ પણ પૂરો પાડવામાં આવે છે, જેના દ્વારા ગ્રાહક ખરીદેલું ઉત્પાદન પોતાના સરનામા પર મેળવ્યા બાદ ચુકવણી કરે છે.

વ્યવસાયીથી વ્યવસાયી (Business to Business)

જુદા જુદા વ્યવસાયીઓ વચ્ચે થતી ઇ-કૉમર્સની પ્રવૃત્તિઓને 'વ્યવસાયીથી વ્યવસાયી' પ્રતિકૃતિનો સંદર્ભ આપવામાં આવે છે. B2Bમાં વેચનાર અને ખરીદનાર બન્ને વ્યવસાયી છે. તે પુરવઠાકાર (suppliers), વિતરક (distributors) કે અન્ય મધ્યસ્થી (agent) સાથે ઇ-સંબંધ (e-relationship)ની સ્થાપના કરે છે. સામેલ થયેલા વ્યવસાયીઓ વચ્ચે વધુ પારદર્શિતાને કારણે વધુ કાર્યક્ષમતા પ્રાપ્ત કરી શકાય છે. ઉદાહરણ તરીકે, સંગ્રાહક, વિતરક અને જથ્થાબંધ વેપારી સાથે ઉત્પાદક સોદો કરે છે. B2Bનો ઉપયોગ કરી ઉત્પાદક વિતરક સાથે માલના જથ્થામાં થતા ઘટાડા વિશે તત્કાલ સંવાદ કરી શકે છે તથા વિતરક તેનો ત્વરિત પ્રત્યુત્તર આપી શકે છે.

સંસ્થાઓ તેમની સામાન્ય વ્યાવસાયિક પ્રવૃત્તિઓ જેવી કે પુરવઠાકારનું વ્યવસ્થાપન, માલના જથ્થાની યાદીનું વ્યવસ્થાપન, ચુકવણીનું વ્યવસ્થાપન વગેરેની કાર્યક્ષમતામાં B2Bની મદદથી વધારો કરી શકે છે. તે ટેલિમાર્કેટિંગ (telemarketing)ના વ્યવસ્થાપન, પુરવઠા-સાંકળ (supply chain), માલની પ્રાપ્તિ, નિયમિત સમયે માલ પહોંચાડવો, ઑનલાઇન સેવા વગેરે માટે અસરકારક માધ્યમ છે. commodity.com અને tradeindia.com વેબસાઇટ B2Bનાં ઉદાહરણ છે.

ગ્રાહકથી ગ્રાહક (Consumer to Consumer)

ગ્રાહકો વચ્ચે થતી ઇ-કૉમર્સની પ્રવૃત્તિના વ્યવહારોને 'ગ્રાહકથી ગ્રાહક' પ્રતિકૃતિનો સંદર્ભ આપવામાં આવે છે. કોઈ પણ ત્રાહિત પક્ષની સંડોવણી વગર ઑનલાઇન હરાજી અને જાહેરાત દ્વારા તે ગ્રાહકોને પરસ્પર સોદા કરવાની સુવિધા પૂરી પાડે છે. C2C વેબસાઇટ પર ઇન્ટરનેટનો ઉપયોગકર્તા વેચનાર કે ખરીદનાર બની શકે છે. C2C પ્રતિકૃતિનું ઉત્તમ ઉદાહરણ હરાજીની સાઇટ છે. જો આપણે કોઈ ઉત્પાદન વેચવું હોય, તો તેને હરાજીની સાઇટ પર યાદી સ્વરૂપે મૂકી શકીએ છીએ અને અન્ય વ્યક્તિઓ તેની બોલી લગાવે છે. eBay.com, OLX.com અને Quikr.com એ C2C વેબસાઇટનાં ઉદાહરણ છે.

ગ્રાહકથી વ્યવસાયી (Consumer to Business)

જ્યારે ગ્રાહક દ્વારા ઉત્પાદન કે સેવાની કિંમત નક્કી કરવામાં આવે, ત્યારે ગ્રાહકથી વ્યવસાયી પ્રતિકૃતિ અવળી હરાજીને સમાવતી પદ્ધતિ છે. આ પ્રકારના ઇ-કૉમર્સમાં ગ્રાહકને પરવડી શકે તે રીતે અથવા તો નિશ્ચિત ઉત્પાદન કે સેવા માટે તેઓ ચૂકવવા ઇચ્છતા હોય તે કિંમતથી વ્યાપક શ્રેણીમાં ઉત્પાદન અને સેવાની પસંદગી મળે છે. સંસ્થાઓ ગ્રાહકને ઉત્પાદન કે સેવા આપવા માટે બોલી લગાવે છે. આ પદ્ધતિ ભાવતાલના સમયમાં ઘટાડો કરવામાં મદદરૂપ બને છે તથા ગ્રાહક અને સંસ્થા બંનેની લવચિકતા (flexibility) માં વધારો કરે છે. ખરીદીની સામાન્ય પ્રક્રિયાને ઉલટાવવા માટે C2B ઇન્ટરનેટનો ઉપયોગ કરે છે, જ્યાં ગ્રાહક દ્વારા ચુકવણી નક્કી કરવામાં આવે છે અને સંસ્થા નક્કી કરે છે કે તે સ્વીકારવું કે નહીં. bidstall.com અને JeetLe.in એ C2B પ્રકારની વેબસાઇટનાં ઉદાહરણ છે.

ઇ-કૉમર્સની વ્યવસાય પ્રતિકૃતિઓમાં કેટલીક અન્ય વિવિધતા પણ છે. જો આપણે સરકારને એક સ્વાયત્ત અસ્તિત્વ માની લઈએ, તો તે સંદર્ભે નીચે જણાવેલ પ્રતિકૃતિઓ પણ અસ્તિત્વ ધરાવે છે :

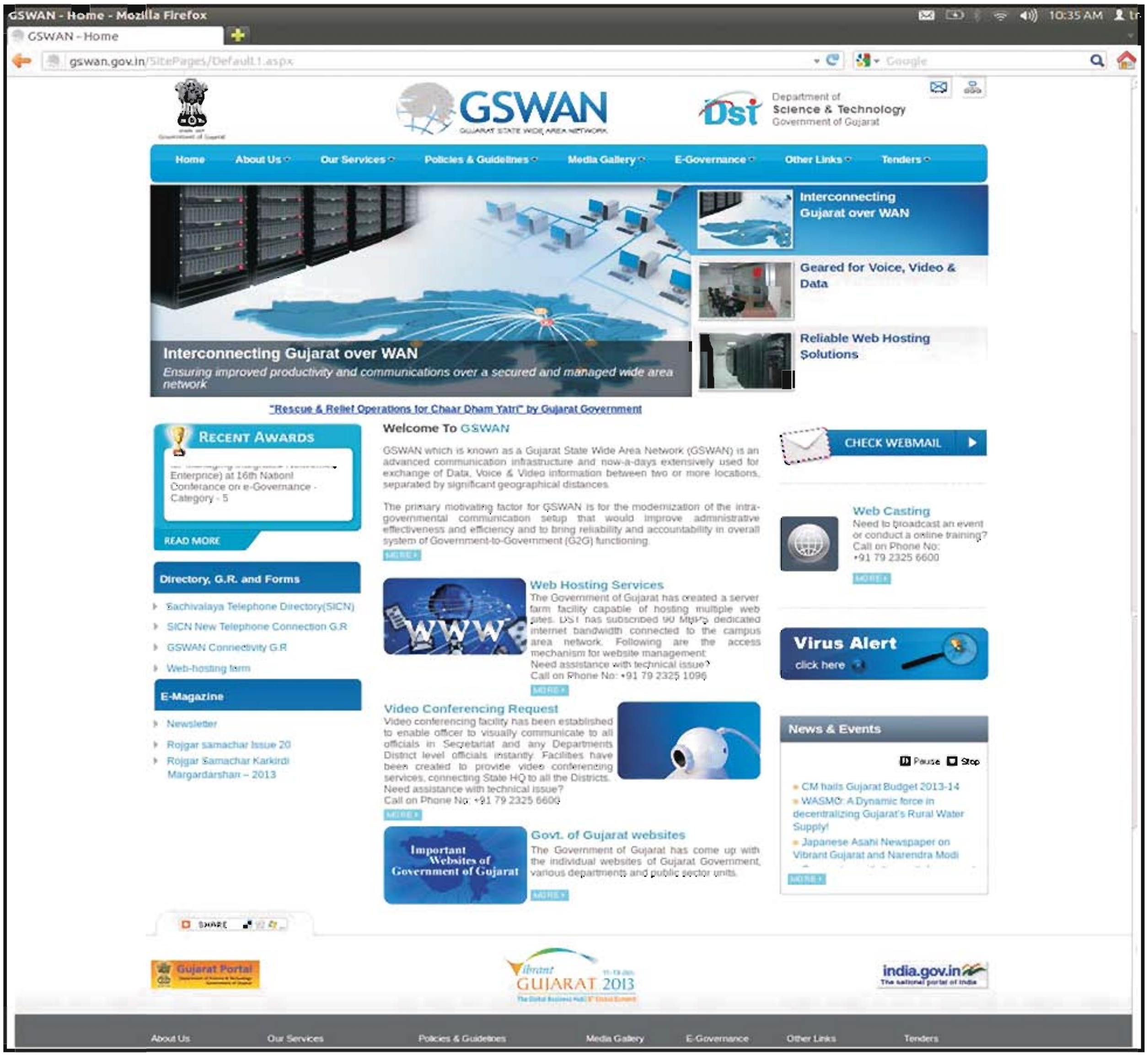
- સરકારથી વ્યવસાયી (Government to Business - G2B)
- સરકારથી નાગરિક (Government to Citizen - G2C)
- સરકારથી સરકાર (Government to Government - G2G)

સરકારથી વ્યવસાયી (Government to Business)

સરકારી વેબસાઇટના વિશાળ નેટવર્કનો ઉપયોગ કરી સરકાર દ્વારા વ્યવસાયી સંસ્થાઓને પૂરી પાડવામાં આવતી સેવાઓ અને માહિતીને સરકારથી વ્યવસાયી (G2B)ના સંદર્ભે જોવામાં આવે છે. વ્યવસાયી આ વેબસાઇટ પરથી સંસ્થાઓ સંબંધિત માહિતી જેવી કે, વ્યવસાયનીતિ, વ્યવસાય શરૂ કરવાની મંજૂરી, વ્યવસાયના સુયોજન (setup) માટેની જરૂરિયાતો અને અન્ય વિવરણો વગેરે મેળવી શકે છે. અહીં સરકારી કાર્યાલયોને લગતાં વિવિધ ફૉર્મ ભરીને રજૂ કરી શકાય છે. ઉદાહરણ તરીકે, ભારતીય સરકારના કરવેરા વિભાગની વેબસાઇટ www.incometaxindia.gov.in છે, જેમાં કરવેરાને લગતા તમામ નિયમો, વિવિધ ફૉર્મ અને કરવેરાનાં ઑનલાઇન રિટર્ન ભરવા માટેની સુવિધા આપવામાં આવી છે. આ વ્યવસાયી પ્રતિકૃતિ ઇ-ગવર્નન્સની એક પહેલ છે.

સરકારથી નાગરિક (Government to Citizen)

સરકારથી નાગરિક (G2C) પણ ઇ-ગવર્નન્સનો એક ભાગ છે. આ વ્યવસાયી પ્રતિકૃતિનો હેતુ સ્વતંત્ર નાગરિકને સારી અને અસરકારક સેવાઓ પૂરી પાડવાનો છે. આ વેબસાઇટ જુદાં-જુદાં સરકારી ખાતાંઓ, વિવિધ કલ્યાણકારી યોજનાઓ, તથા નાગરિકોને ઉપયોગી અનેક પ્રકારનાં અરજીપત્રકો વગેરેની માહિતી પૂરી પાડે છે. આ ઉદ્દેશ્યની પૂર્તિ માટે ગુજરાત સરકારે Gujarat State Wide Area Network (GSWAN) નામનું પોતાનું નેટવર્ક વિકસિત કર્યું છે, જે www.gswan.gov.in પર ઑનલાઇન ઉપલબ્ધ છે. આકૃતિ 4.10 GSWANનું હોમપેજ દર્શાવે છે.



આકૃતિ 4.10 : GSWAN વેબસાઇટનું હોમપેજ

સરકારથી સરકાર (Government to Government)

એક સરકારી સંસ્થા, એજન્સી કે વિભાગ અન્ય સરકારી સંસ્થા, એજન્સી કે વિભાગ સાથે વાણિજ્યરહિત (non-commercial) સંદેશાવ્યવહાર કરે, તો તેને સરકારથી સરકાર (G2G) તરીકે ઓળખવામાં આવે છે. IT ખર્ચાને ઘટાડવા, પ્રક્રિયાઓને સુનિયોજિત બનાવવા અને કાર્યાલયોને વધુ અસરકારક બનાવવા આ માહિતીનું વિવરણ મદદરૂપ બને છે.

ઉપર વિવેચિત તમામ ઇ-કૉમર્સ પ્રતિકૃતિઓમાં B2C અને B2B વિપુલ પ્રમાણમાં ઉપયોગમાં લેવામાં આવતી પ્રતિકૃતિઓ છે. આ બંને પ્રતિકૃતિમાં મુખ્ય તફાવત 'ગ્રાહક' અંગે છે. B2B પ્રતિકૃતિમાં ગ્રાહક એક સંસ્થા છે, જ્યારે B2C પ્રતિકૃતિમાં ગ્રાહક એક સ્વતંત્ર વ્યક્તિ છે.

સારાંશ

આ પ્રકરણમાં આપણે ઇ-કૉમર્સ, તેના ફાયદા અને ગેરફાયદા, ઇ-કૉમર્સના વિનિયોગો અને તેની વ્યવસાય પ્રતિકૃતિઓ વિશે ચર્ચા કરી. વ્યાવસાયિક પ્રવૃત્તિઓ માટે કરવામાં આવતા ઇન્ટરનેટના ઉપયોગને ઇ-કૉમર્સ તરીકે વ્યાખ્યાયિત કરવામાં આવે છે. ઇ-કૉમર્સનો ઉપયોગ માર્કેટિંગ અને વેચાણ, ઉત્પાદનની ખરીદી, ઑનલાઇન વર્તમાનપત્ર, માહિતી સેવાઓ, સહાયસેવાઓ, નેટબૅંકિંગ અને તેને સમકક્ષ અન્ય અનેક પ્રવૃત્તિઓ માટે કરવામાં આવે છે. ઇ-કૉમર્સની વ્યવસાય-પ્રતિકૃતિનું વર્ગીકરણ તેમાં સમાવિષ્ટ પક્ષો અને તેના દ્વારા પૂરી પાડવામાં આવતી વ્યાવસાયિક પ્રવૃત્તિઓ કે સેવાના પ્રકારના આધારે કરવામાં આવે છે. આપણે વ્યવસાયીથી ગ્રાહક (B2C), વ્યવસાયીથી વ્યવસાયી (B2B), ગ્રાહકથી ગ્રાહક (C2C), ગ્રાહકથી વ્યવસાયી (C2B) તેમજ સરકારથી વ્યવસાયી (G2B) જેવી પ્રતિકૃતિઓનો અભ્યાસ કર્યો.

સ્વાધ્યાય

1. ઇ-કૉમર્સને વ્યાખ્યાયિત કરો. ઇ-કૉમર્સના વિનિયોગની યાદી બનાવો.
2. ઑનલાઇન હરાજી એટલે શું ?
3. ઇલેક્ટ્રૉનિક વર્તમાનપત્ર એટલે શું ? આ પ્રકારની વેબસાઇટની વિશેષતાઓની યાદી બનાવો.
4. પુસ્તકોની ઑનલાઇન દુકાનોની યાદી બનાવો અને કોઈ પણ એકની વિશેષતાઓ વિશે ચર્ચા કરો.
5. નેટબૅંકિંગ એટલે શું ? ઑનલાઇન બૅંકિંગ માટેની કેટલીક વેબસાઇટનાં નામ આપો.
6. પરંપરાગત વ્યવસાય અને ઇ-કૉમર્સ વચ્ચેનો તફાવત સ્પષ્ટ કરો.
7. ઇ-કૉમર્સના ફાયદા જણાવો.
8. નીચે આપેલ ઇ-કૉમર્સની વ્યવસાય પ્રતિકૃતિઓ વિશે માહિતી આપો. દરેકનાં ઉદાહરણ પણ આપો :
   (1) વ્યવસાયીથી ગ્રાહક (Business to Consumer - B2C)
   (2) વ્યવસાયીથી વ્યવસાયી (Business to Business - B2B)
   (3) ગ્રાહકથી ગ્રાહક (Consumer to Consumer - C2C)
   (4) ગ્રાહકથી વ્યવસાયી (Consumer to Business - C2B)
   (5) સરકારથી વ્યવસાયી (Government to Business - G2B)
9. **આપેલ વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરો :**
   (1) નીચેનામાંથી પુસ્તકોની ઑનલાઇન દુકાનનું ઉદાહરણ કયું છે ?
   (a) Amazon (b) irctc (c) Gmail (d) yahoo

(2) ઇન્ટરનેટ પર ડિજિટલ સ્વરૂપે ઉપલબ્ધ વર્તમાનપત્રને શું કહે છે ?

(a) I-newspaper (b) Internet-newspaper

(c) www-newspaper (d) E-newspaper

(3) કઈ પ્રક્રિયા દ્વારા ગ્રાહકને ઉત્પાદનની કિંમત માટે બોલી લગાવીને ખરીદ કે વેચાણની સુવિધા પૂરી પાડવામાં આવે છે ?

(a) માર્કેટિંગ (b) હરાજી (c) પુસ્તકોની દુકાન (d) નોંધણી

(4) ઇન્ટરનેટ પર કરવામાં આવતા બૅન્કના વ્યવહારોને નીચેનામાંથી કઈ પ્રક્રિયા તરીકે ઓળખવામાં આવે છે ?

(a) હરાજી (b) બોલી (c) નેટબૅંકિંગ (d) www-બૅંકિંગ

(5) નીચેનામાંથી કઈ વિશેષતા પરંપરાગત વ્યવસાયની છે ?

(a) નિશ્ચિત સમયગાળા માટે અથવા તો વ્યવસાયના કલાકોમાં પ્રક્રિયા કરવામાં આવે.

(b) ઇન્ટરનેટ પર જાહેરાત કરવામાં આવે.

(c) ચુકવણી મેળવવા માટે ઇ-ચુકવણીની પદ્ધતિનો ઉપયોગ કરવામાં આવે.

(d) ગ્રાહકો ઉત્પાદનો અને યોજનાઓ શોધી શકે.

(6) ઇ-કૉમર્સમાં નીચેનામાંથી કઈ વિશેષતા જોવા મળે છે ?

(a) નિશ્ચિત સમયગાળા માટે અથવા તો વ્યવસાયના કલાકોમાં પ્રક્રિયા કરવામાં આવે.

(b) સ્પર્ધકો સાથે કોઈ માહિતી વહેંચવામાં ન આવે.

(c) સ્થળને ભાડે લેવું પડે અથવા ખરીદવું પડે.

(d) ઉત્પાદનની જાહેરાત ઇન્ટરનેટ દ્વારા કરવામાં આવે.

(7) ઇ-કૉમર્સ દ્વારા નીચેનામાંથી કયો ફાયદો થતો નથી ?

(a) ઓછી કિંમત (b) વ્યવસાયનો 24×7 સમય

(c) સુરક્ષા (d) ભૌગોલિક મર્યાદાનો અભાવ

(8) નીચેનામાંથી ઇ-કૉમર્સનો ગેરફાયદો કયો છે ?

(a) ગોપનીયતા (b) ગ્રાહકસેવામાં સુધારો

(c) ઝડપ (d) વ્યવસાયનો 24×7 સમય

(9) ઉત્પાદન કે સેવા પૂરી પાડતા વ્યાવસાયિકો કે સંસ્થાઓ ઇન્ટરનેટની વેબસાઇટ પર કઈ વ્યવસાયી પ્રતિકૃતિનો ઉપયોગ કરે છે ?

(a) વ્યવસાયીથી ગ્રાહક (Business to Consumer - B2C)

(b) વ્યાવસાયીથી વ્યવસાયી (Business to Business - B2B)

(c) ગ્રાહકથી વ્યવસાયી (Consumer to Business - C2B)

(d) સરકારથી વ્યવસાયી (Government to Business - G2B)

(10) જુદા-જુદા વ્યવસાયી ભાગીદારો વચ્ચે કરવામાં આવતી પ્રવૃત્તિઓને ઇ-કૉમર્સની કઈ પ્રતિકૃતિ કહેવામાં આવે છે ?

(a) સરકારથી વ્યવસાયી (Government to Business - G2B)

(b) ગ્રાહકથી વ્યવસાયી (Consumer to Business - C2B)

(c) વ્યવસાયીથી વ્યવસાયી (Business to Business - B2B)

(d) વ્યવસાયીથી ગ્રાહક (Business to Consumer - B2C)

(11) નીચેનાંમાંથી કયું ઉદાહરણ C2C પ્રતિકૃતિનું છે ?

(a) હરાજી માટેની સાઇટ

(b) ઇ-વર્તમાનપત્ર

(c) ઑનલાઇન ખરીદી

(d) માહિતીસેવા

(12) ગ્રાહકો વચ્ચે કરવામાં આવ્યા હોય, તેવા વ્યવહારોને સમાવતી પ્રવૃત્તિને ઇ-કૉમર્સની કઈ વ્યવસાય પ્રતિકૃતિ કહે છે ?

(a) સરકારથી વ્યવસાયી (Government to Business - G2B)

(b) ગ્રાહકથી ગ્રાહક (Consumer to Consumer - C2C)

(c) વ્યવસાયીથી વ્યવસાયી (Business to Business - B2B)

(d) વ્યવસાયીથી ગ્રાહક (Business to Consumer - B2C)

(13) નીચેનામાંથી કઈ ઇ-કૉમર્સ પ્રતિકૃતિમાં અવળી હરાજીનો સમાવેશ કરવામાં આવે છે, કે જેમાં ગ્રાહક દ્વારા ઉત્પાદન કે સેવાની કિંમત નક્કી કરવામાં આવે છે ?

(a) ગ્રાહકથી વ્યવસાયી (Consumer to Business - C2B)

(b) વ્યવસાયીથી વ્યવસાયી (Business to Business - B2B)

(c) ગ્રાહકથી ગ્રાહક (Consumer to Consumer - C2C)

(d) સરકારથી વ્યવસાયી (Government to Business - G2B)

(14) નીચેનામાંથી કઈ ઇ-કૉમર્સની પ્રતિકૃતિ ઇ-ગવર્નન્સનો એક ભાગ છે ?

(a) વ્યવસાયીથી વ્યવસાયી (Business to Business - B2B)

(b) ગ્રાહકથી વ્યવસાયી (Consumer to Business - C2B)

(c) ગ્રાહકથી ગ્રાહક (Consumer to Consumer - C2C)

(d) સરકારથી વ્યવસાયી (Government to Business - G2B)

(15) નીચેનામાંથી કઈ ઇ-કૉમર્સ પ્રતિકૃતિ દ્વારા એક સરકારી એજન્સી, સંસ્થા કે વિભાગ અન્ય સરકારી એજન્સી, સંસ્થા કે વિભાગ સાથે વાણિજ્યરહિત સંદેશાવ્યવહાર કરે છે ?

(a) વ્યવસાયીથી વ્યવસાયી (Business to Business - B2B)

(b) ગ્રાહકથી વ્યવસાયી (Consumer to Business - C2B)

(c) સરકારથી સરકાર (Government to Government - G2G)

(d) ગ્રાહકથી ગ્રાહક (Consumer to Consumer - C2C)

એમ-કૉમર્સનો પરિચય 5

મોબાઇલ કોમર્સ (Mobile Commerce)ને એમ-કૉમર્સ (M-commerce) તરીકે ઓળખવામાં આવે છે. તે મોબાઇલ ફોન, પર્સનલ ડિજિટલ આસિસ્ટન્ટ (Personal Digital Assistant - PDAs), સ્માર્ટફોન, ટેબ્લેટ, પામટૉપ કે તેના જેવા કોઈ પણ મોબાઇલ સાધન દ્વારા ઇન્ટરનેટની મદદથી માલ કે સેવાના ખરીદ કે વેચાણને રજૂ કરે છે.

એમ-કૉમર્સ ઉપયોગકર્તાને લવચિકતા (flexibility) અને સાર્વત્રિકતા (ubiquity)ના લાભ પૂરા પાડે છે. મોબાઇલ ફોનના ઉપયોગથી ગ્રાહક કમ્પ્યૂટરના ટર્મિનલ પાસે કે દુકાનમાં પ્રત્યક્ષ હાજર ન હોવા છતાં વ્યાવસાયિક વ્યવહાર કરી શકે છે. આને લગતાં ઉપકરણો ઉપયોગકર્તા જ્યાં જાય ત્યાં તેની પાસે જ હોય છે, જે કોઈ પણ સ્થળેથી ઇન્ટરનેટના ઉપયોગને શક્ય બનાવે છે. પરિવહન વખતે તે વાસ્તવિક સમયના વ્યવહારની સુવિધા આપે છે. સ્માર્ટફોન અને ટેબ્લેટની લોકપ્રિયતા દિન-પ્રતિદિન થતી વૃદ્ધિને કારણે વધુ ને વધુ ઉપયોગકર્તાઓ એમ-કૉમર્સ તરફ વળે છે. એમ-કૉમર્સનાં કેટલાંક ઉદાહરણ આ મુજબ છે :

- હવાઈ ટિકિટની ખરીદી
- ચલચિત્રની ટિકિટની ખરીદી
- રેસ્ટોરન્ટનું બુકિંગ અને આરક્ષણ
- હોટલનું બુકિંગ અને આરક્ષણ
- શૅરબજારનું પૃથક્કરણ

બૅન્ક અને તેના જેવી અન્ય વાણિજ્યિક સંસ્થાઓ પોતાના વ્યવસાયને સુરક્ષિત રાખવા એમ-કૉમર્સનો વધુ ને વધુ ઉપયોગ કરી રહી છે. તેઓ પોતાના ગ્રાહકને મોબાઇલ ફોન દ્વારા ખાતાની સિલક જાણવા, શૅરબજારની વધઘટ જોવા અને તેમાં ખરીદ-વેચાણ કરવા માટેની સુવિધા આપે છે. આ સેવાને મોબાઇલ-બૅંકિંગ (Mobile Banking) અથવા એમ-બૅંકિંગ (M-Banking) કહે છે. મોબાઇલ ઉપકરણો દ્વારા પૂરી પાડવામાં આવતી શૅરબજારને લગતી સેવાઓ પણ હવે પ્રચલિત બની રહી છે અને તેને મોબાઇલ બ્રોકરેજ (Mobile Brokerage) તરીકે ઓળખવામાં આવે છે. મોબાઇલ સંબંધિત સેવાઓની માંગણી સાથે સમાચાર-માહિતી, રમતગમત, મનોરંજન, ખરીદી અને આરક્ષણ અંગેની માહિતી જેવાં ક્ષેત્રો પણ વિસ્તાર પામ્યાં છે.

એમ-કૉમર્સના લાભ (Benefits of M-Commerce)

મોબાઇલના ઉપયોગમાં ઉત્તરોત્તર વૃદ્ધિ થઈ રહી છે અને તે હવે માત્ર પ્રત્યાયન-વ્યવહાર એટલે કે પરસ્પર વાત કરવા પૂરતો મર્યાદિત રહ્યો નથી. મોબાઇલ સંસ્થાઓ નવી સુવિધાઓ ધરાવતા સ્માર્ટફોન સાથે આવી રહી છે, જેના દ્વારા ગ્રાહકને સરળતા, લવચિકતા અને સુરક્ષા એકસાથે પૂરી પાડવામાં આવે છે. સહજ ઉપલબ્ધતા અને ગતિશીલતાને કારણે એમ-કૉમર્સ આજકાલ વધુ પ્રચલિત થઈ રહ્યું છે. વેબવિકાસ કરતી સંસ્થાઓ પણ મોબાઇલ ઉપકરણો પર વેબસાઇટ યોગ્ય રીતે દર્શાવી શકાય તે માટે સક્રિયતા દાખવી રહ્યા છે.

એમ-કૉમર્સ વાયરલેસ નેટવર્ક અને ઇન્ટરનેટનું એવું સંચાલન છે, જેને હસ્તગત ઉપકરણ દ્વારા સતત પાસે રાખી શકાય છે. ઇન્ટરનેટ અને ઇ-કૉમર્સના ફાયદા એમ-કૉમર્સ દ્વારા પણ ઉપલબ્ધ થાય છે. એમ-કૉમર્સના કેટલાક લાભ નીચેની યાદીમાં દર્શાવ્યા છે :

- એમ-કૉમર્સ ઉપયોગકર્તાને અનુકૂલન પૂરું પાડે છે. મોબાઇલ ઉપકરણ પરની કેટલીક ક્લિક દ્વારા ઉપયોગકર્તા પરિવહનના સમય દરમિયાન પણ ખરીદી, બૅંકિંગ અને મીડિયા ફાઇલ ડાઉનલોડ કરવા જેવી પ્રવૃત્તિઓ કરી શકે છે.

- મોબાઈલ સાધન દ્વારા ઉપયોગકર્તાનો પરોક્ષ રીતે કોઈ પણ સમયે ને સ્થાને સંપર્ક કરી શકાય છે.
- વ્યવહારખર્ચમાં ઘટાડો થાય છે.
- વિક્રેતાને ઑર્ડર આપવામાં ઓછો સમય લાગે છે. તદુપરાંત ઉપયોગકર્તા પાસે આ માટે PC કે લેપટૉપ હોવું જરૂરી નથી.
- વ્યાવસાયિક પ્રક્રિયાઓ સુવ્યવસ્થિત બને છે.
- વૈશ્વિક સંપર્ક શક્ય બને છે.
- વ્યવસાય નિરંતર એટલે કે 24*7 સમયે ચલાવી શકાય છે.
- કોઈ પણ મોબાઇલ ઉપકરણ દ્વારા માહિતીનો ઉપયોગ કરી શક્વાની લવચિકતા પૂરી પાડે છે.
- અંગત કમ્પ્યૂટરના ઉપયોગની જેમ જ મોબાઈલ સાધનના ઉપયોગ દ્વારા પણ ચુકવણી કરી શકાય છે.
- સમયાધારિત જટિલ (time critical) અને સંકટકાલીન માહિતી મોકલવા ઉપયોગી છે.
- હસ્તગત ઉપકરણનું ભૌતિક સ્થાન સરળતાથી શોધી શકાય છે. સ્થાન આધારિત વિનિયોગોના ઉદ્ભવથી ઉપયોગકર્તા સુસંગત માહિતી મેળવી શકે છે. આ પ્રકરણમાં આગળ આપણે સ્થાન આધારિત વિનિયોગો વિશે ચર્ચા કરીશું.
- મોબાઈલ ઉપકરણ પર વિવિધ ચેતવણીઓ (Customized alerts) સરળતાથી મેળવી શકાય છે.
- વર્તમાન યુગમાં તત્કાલ જોડાણ અને વધુ ઝડપી 3G સેવાઓની ઉપલબ્ધિએ એમ-કૉમર્સને વધુ પ્રચલિત બનાવ્યું છે.
- ઉપયોગકર્તા સુધી માહિતી સમયસર પહોંચાડી શકાય છે. હવાઈ કે ટ્રેનની અનુસૂચિ, વિલંબ કે રદ કર્યાની માહિતી ઉપયોગકર્તાને તેના મોબાઈલ ઉપકરણ પર વાસ્તવિક સમયે પૂરી પાડી શકાય છે.

એમ-કૉમર્સની મર્યાદાઓ (Limitations of M-commerce)

એમ-કૉમર્સના લાભની યાદી લાંબી હોવા છતાં તેની કેટલીક મર્યાદાઓ પણ છે, જે નીચે દર્શાવેલી છે :

- હાલમાં ઉપલબ્ધ હસ્તગત (handheld) સાધનો સ્ક્રીનનું મર્યાદિત કદ ધરાવે છે. વળી, તે ફાઇલના પ્રકાર અને ડેટાના સ્થાનાંતરમાં પણ મર્યાદિત છે. ઘણીવાર વીડિયો દર્શાવવા પણ મુશ્કેલ બને છે.
- અંગત કમ્પ્યૂટરની સરખામણીમાં ઉપયોગકર્તા માટેનો ઈન્ટરફેસ ઓછો અનુકૂળ હોય છે.
- મોબાઈલ ઉપકરણો પાસે કમ્પ્યૂટિંગ ક્ષમતા, મેમરી અને સંગ્રહક્ષમતા મર્યાદિત હોય છે.
- એમ-કૉમર્સ વાયરલેસ નેટવર્ક પર ચાલે છે, જે વાયરયુક્ત નેટવર્કની તુલનામાં ઓછું સુરક્ષિત છે.
- તે મર્યાદિત બૅન્ડવિડ્થ (ગતિ) ધરાવે છે.
- મોબાઈલ અને વાયરલેસ બ્રૉડબૅન્ડ માળખાંને સ્થાપિત કરવાનો ખર્ચ વધુ હોય છે.

એમ-કૉમર્સના વિનિયોગ (Applications of M-Commerce)

એમ-કૉમર્સના વિનિયોગ ઝડપથી વૃદ્ધિ પામી રહ્યા છે. તેના ઉપયોગ અહીં ચર્ચવામાં વિસ્તારો પૂરતાં મર્યાદિત નથી. નિકટના ભવિષ્યમાં વધુ ને વધુ વિનિયોગોનો વિકાસ થવાનો છે. એમ-કૉમર્સના કેટલાક વિનિયોગ વિશે ચર્ચા કરીએ.

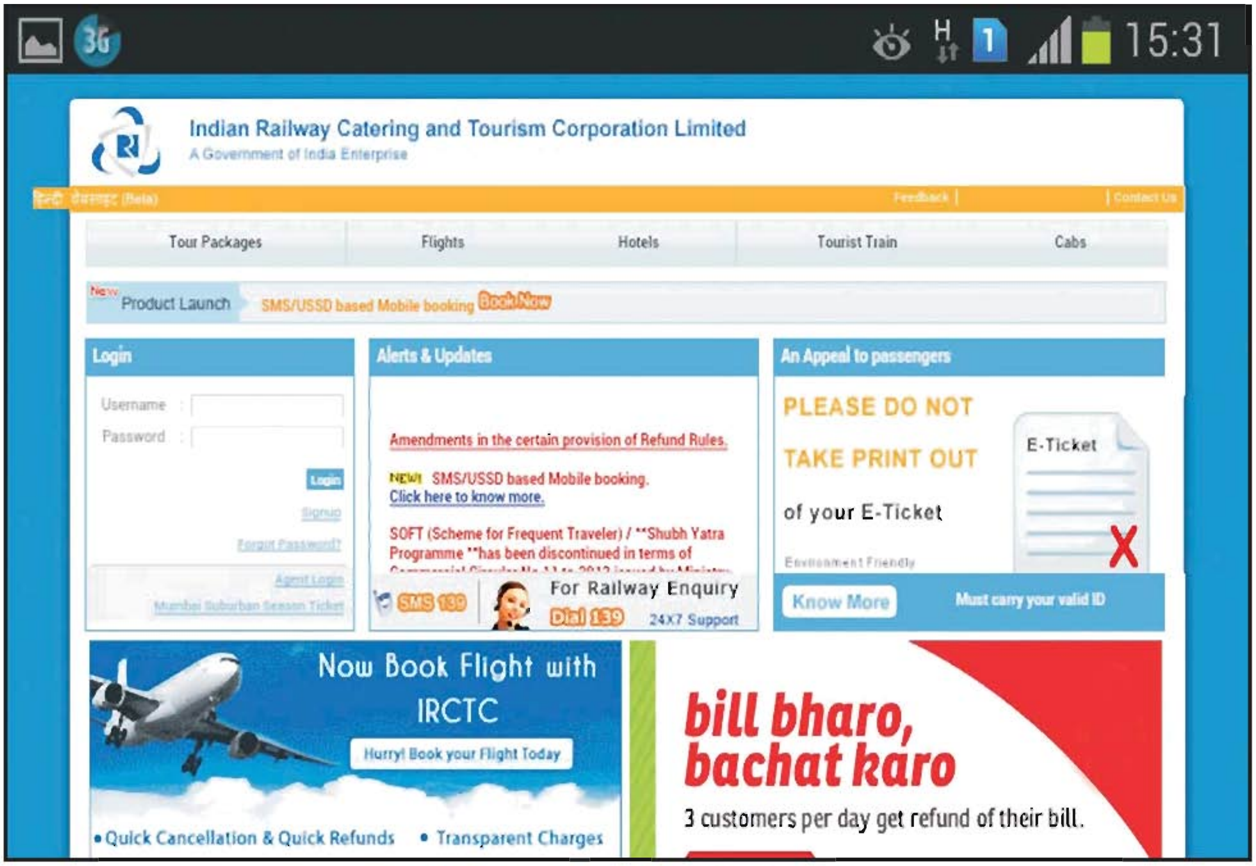
મોબાઈલ માર્કેટિંગ અને વિજ્ઞાપન (Mobile Marketing and Advertising)

હવે સંસ્થાઓ માર્કેટિંગથી વિજ્ઞાપન સુધીની સેવાઓના વિસ્તાર માટે એમ-કૉમર્સનો ઉપયોગ કરે છે. સંસ્થાઓ દ્વારા વિશાળ સંખ્યામાં પ્રેક્ષકો સુધી પહોંચવા માટે મોબાઈલ વિજ્ઞાપનનો માર્ગ સૌથી વધુ પ્રચલિત બન્યો છે. મોટા ભાગના પોર્ટલ માટે ઈન્ટરનેટ પર વિજ્ઞાપન એ આવકનો એક મોટો સ્રોત બની રહ્યો છે. ઘણા છૂટક વેપારીઓ સ્થાન આધારિત મોબાઈલ વિજ્ઞાપનનો વિકલ્પ પૂરો પાડે છે, જેના દ્વારા ગ્રાહકો સુધી પહોંચી વેચાણ વધારી શકાય છે. આમ, મોબાઈલ ઉપકરણ પર મૂકવામાં આવેલ વિજ્ઞાપન અંગત જરૂરિયાત અને ચોકસાઈ સ્થાન પર આધારિત હોઈ શકે છે. ઉપયોગકર્તાના વર્તમાન સ્થાન પાસે આપવામાં આવતા વિવિધ વળતર અને યોજનાઓ અંગે તેને વાકેફ કરી શકાય છે.

મોબાઈલ પર ટિકિટ (Mobile Ticketing)

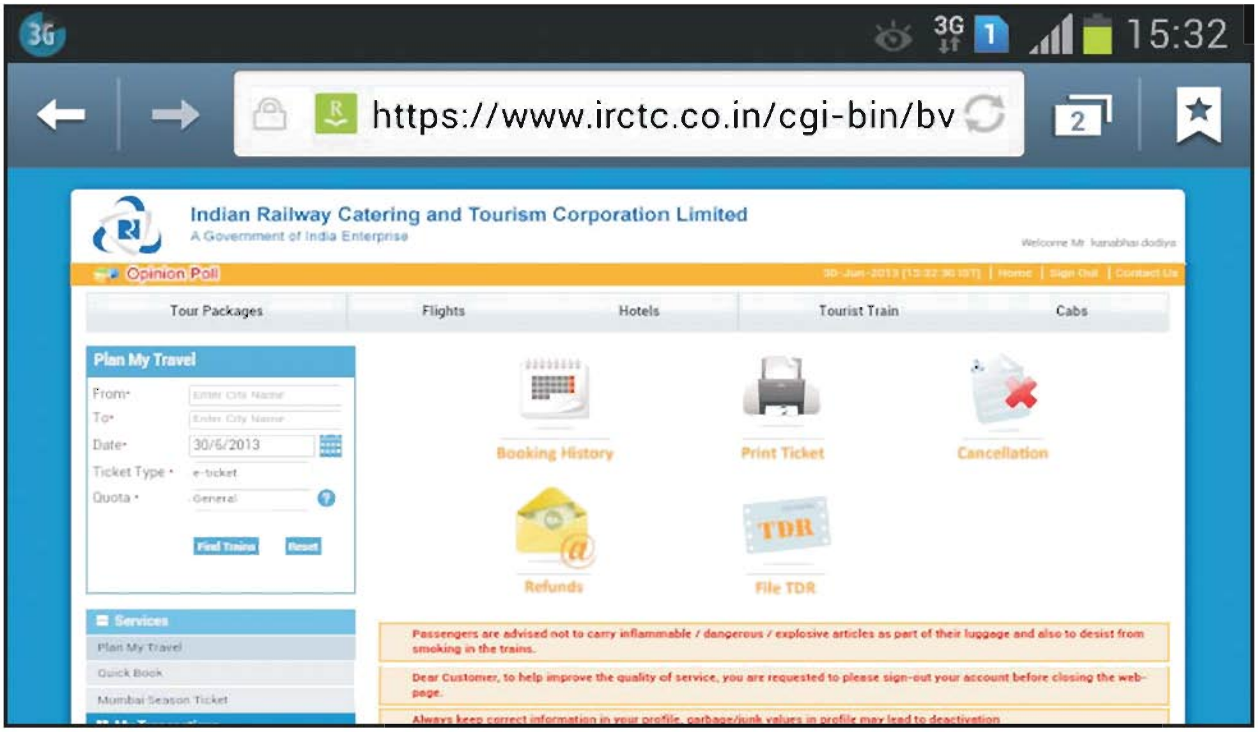
ઉપયોગકર્તા મોબાઈલનો ઉપયોગ કરી સરળતાથી હવાઈ, રેલવે કે ચલચિત્રની ટિકિટ ખરીદી શકે છે. ઉપયોગકર્તાના મોબાઈલ ઉપકરણ પર આ ટિકિટ મોકલવામાં આવે છે. ઉપયોગકર્તા જરૂરી સ્થાન પર પોતાના મોબાઈલ ઉપકરણમાં આ ટિકિટને રજૂ કરી શકે છે. વિનિયોગનો ઉપયોગ કરીને અથવા હવાઈ યાત્રાના એજન્ટનો પોર્ટલનો ઉપયોગ કરીને મોબાઈલ ફોન દ્વારા ટિકિટ સરળતાથી રદ પણ કરાવી શકાય છે. તે અવરજવર કે પાર્કિંગના કષ્ટમાંથી મુક્તિ અપાવે છે, કારણકે ઉપયોગકર્તાએ ટિકિટ ખરીદવા જે-તે સ્થળે જવાની જરૂર રહેતી નથી.

આકૃતિ 5.1 Indian Railway Catering And Tourism Corporation limited (IRCTC)નું લૉગ-ઇન પાનું દર્શાવે છે.



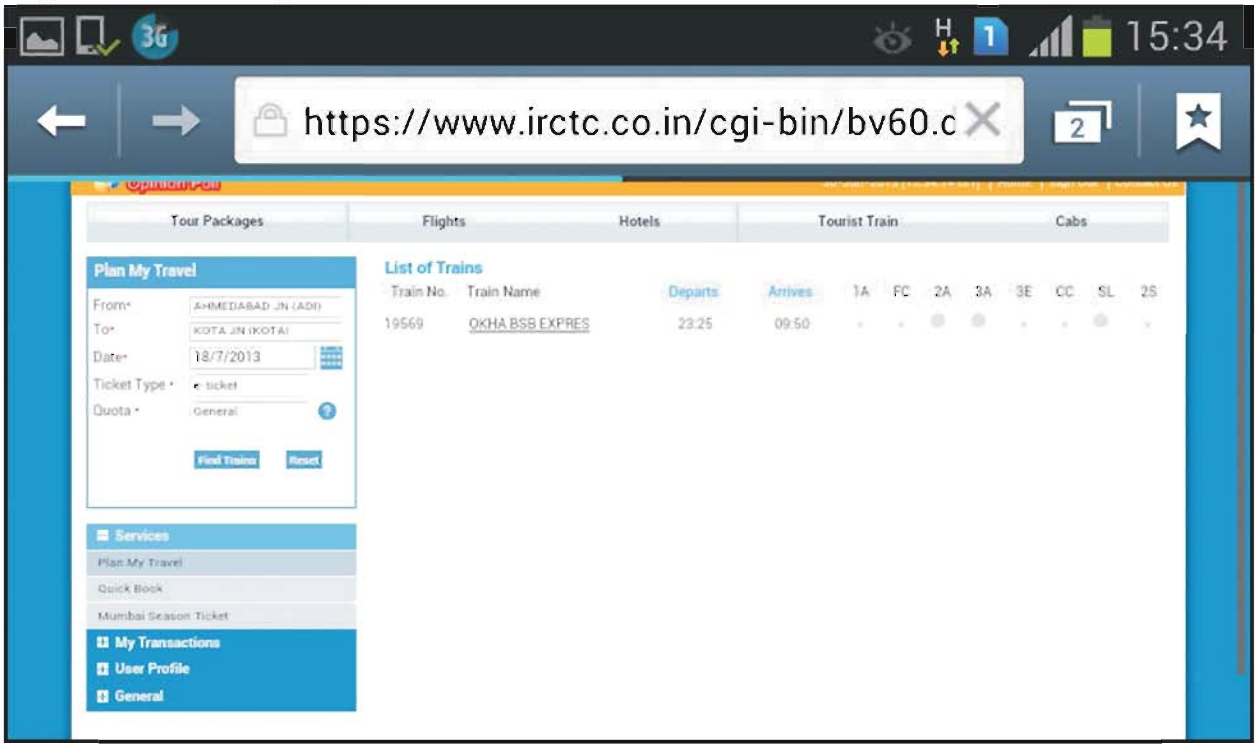
આકૃતિ 5.1 : IRCTCનું લૉગ-ઇન પાનું

આપણે મોબાઈલ સાધનનો ઉપયોગ કરી ઑનલાઈન ટિકિટની નોંધણી કરવાનો પ્રયત્ન કરીએ. યૂઝરનેમ અને પાસવર્ડ પૂરાં પાડ્યા બાદ (નોંધ : ઉપયોગકર્તા પાસે પહેલેથી ખાતું છે) ઉપયોગકર્તા આકૃતિ 5.2માં દર્શાવ્યા મુજબ મૂળ સ્થાન (source) અને ગંતવ્ય (destination)નાં નામ, મુસાફરીની તારીખ, ટિકિટનો પ્રકાર અને નિયત હિસ્સા (quota) અંગેની તેની યોજના દાખલ કરી શકે છે.



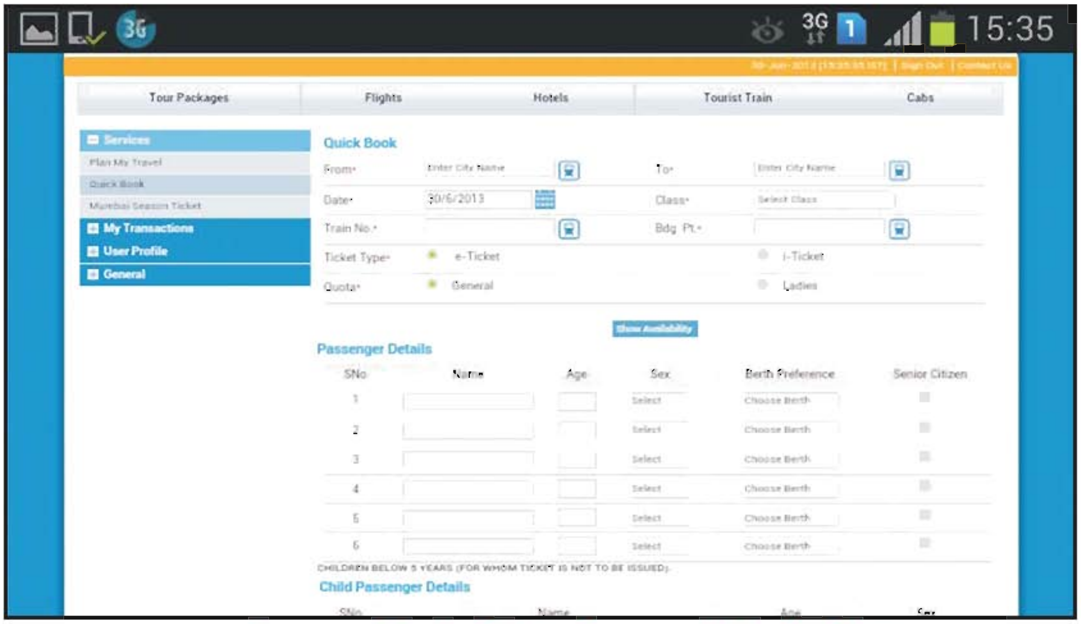
આકૃતિ 5.2 : મુસાફરીની વિગતો

જ્યારે ઉપયોગકર્તા Find Trains બટન પર ક્લિક કરે, ત્યારે તેની જરૂરિયાત મુજબની તારીખે ઉપલબ્ધ ટ્રેનની યાદી આકૃતિ 5.3માં દર્શાવ્યા મુજબ રજૂ કરવામાં આવે છે.



આકૃતિ 5.3 : ઉપલબ્ધ ટ્રેનની યાદી સાથે મુસાફરીની વિગતો

ઉપયોગકર્તા ટ્રેનમાં નામ પર ક્લિક કરી વધુ વિગતો મેળવી શકે છે. આકૃતિ 5.3માં આપણને માત્ર એક ટ્રેનનું નામ જોવા મળી શકે છે, પરંતુ એથી વધુ ટ્રેનની યાદી મેળવવી પણ શક્ય છે. ત્યાર પછી, આકૃતિ 5.4માં દર્શાવ્યા મુજબ ટિકિટની નોંધણી દરમિયાન મુસાફર પોતાની વિગતો પૂરી પાડી શકે છે.

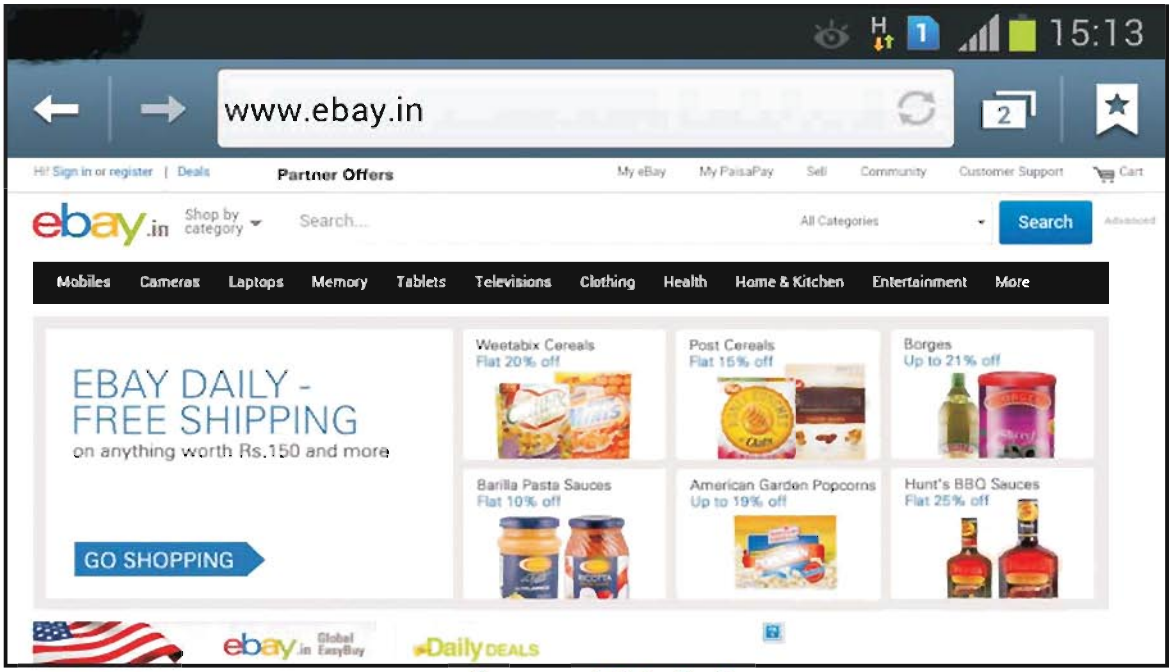


આકૃતિ 5.4 : ટિકિટની નોંધણી માટે મુસાફરની વિગતો ઉમેરવી

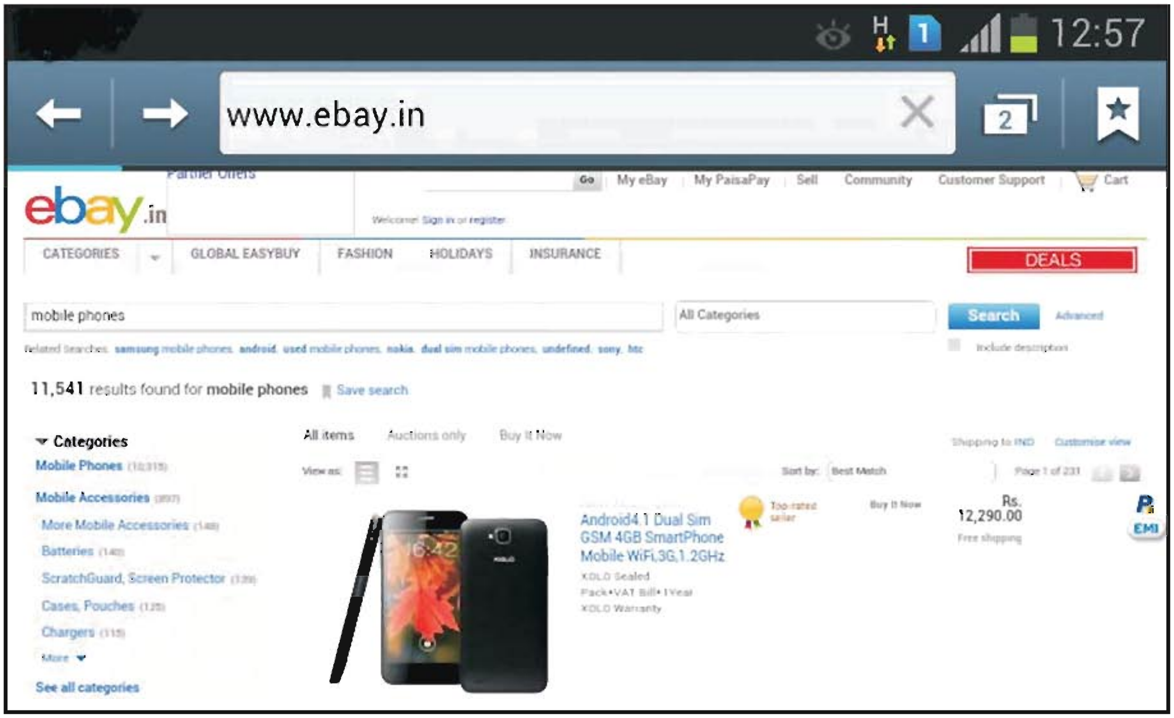
નોંધણી થયા બાદ ઉપયોગકર્તાને IRCTC તરફથી ઇ-ટિકિટનો સંદેશ મોકલવામાં આવે છે, જે મુસાફરી દરમિયાન રજૂ કરી શકાય છે. તમે જોઈ શકો છો કે, કમ્પ્યૂટર સાથે જોડાયા વિના કે રેલવે-સ્ટેશને પ્રત્યક્ષ હાજરી આપ્યા વિના ઉપયોગકર્તા કોઈ પણ સ્થળે, કોઈ પણ સમયે મોબાઇલ ઉપકરણની મદદથી આ સેવાનો ઉપયોગ કરી શકે છે.

મોબાઇલ હરાજી (Mobile Auctions)

હાલના સમયમાં હરાજીની સાઇટ ઘણી પ્રચલિત થઈ રહી છે. મોબાઇલ ઉપકરણ હરાજીની આવી સાઇટનો સંપર્ક સાધવામાં મદદરૂપ બને છે. ઉપયોગકર્તા હરતાં-ફરતાં પણ આ સાઇટનો ઉપયોગ કરી શકે છે, બોલી લગાવી શકે છે, બોલી પર સતત દેખરેખ રાખી શકે છે અને આ પ્રક્રિયામાં યોગ્ય રીતે સમયસર સક્રિય રહી શકે છે. હરાજી માટેની અનેક વેબસાઇટોએ વાયરલેસ નેટવર્ક દ્વારા મોબાઇલ ઉપકરણનો ઉપયોગ કરી શકાય તે પ્રકારના ગેટ-વે અને ઇન્ટરફેસની રચના કરી છે. મોબાઇલ ઉપકરણ પર હરાજી માટેની ઇ-બે સાઇટનું દૃશ્ય આકૃતિ 5.5 અને 5.6માં દર્શાવ્યું છે. ઉપયોગકર્તા મોબાઇલ ઉપકરણ વડે અહીં ઉત્પાદન માટે બોલી લગાવી શકે છે.



આકૃતિ 5.5 : ઇ-બેનું હોમપેજ



આકૃતિ 5.6 : ઉત્પાદન માટે બોલી લગાવવી

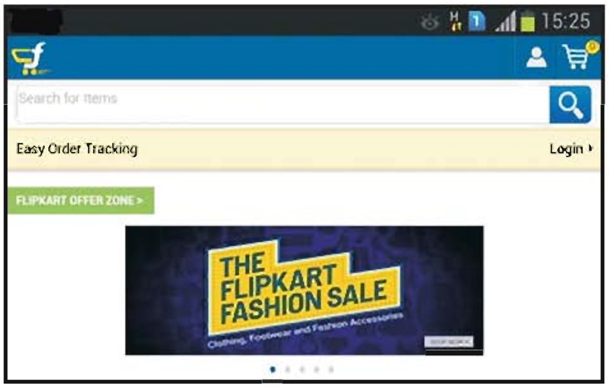
મોબાઈલ દ્વારા મનોરંજન (Mobile Entertainment)

મોબાઇલ સાધનોનો વ્યાપક ઉપયોગ ઑડિયો સાંભળવા, વીડિયો જોવા અને રમતો રમવા માટે કરવામાં આવે છે. મોબાઇલના ઉપયોગકર્તાઓ મનોરંજનની ઑનલાઇન લાઇબ્રેરીમાં સભ્ય થઈ શકે છે, જ્યાં તેઓ ગીતો, વીડિયો કે રમતોને શોધી પોતાના સંગ્રહમાં સરળતાથી ડાઉનલોડ કરી અનુકૂળતા મુજબ તેનો ઉપયોગ કરી શકે છે. વિપુલ સંખ્યાના મોબાઇલ ધારકોને ડાઉનલોડ આધારિત ચુકવણી, કાર્યક્રમ આધારિત ચુકવણી કે લવાજમ આધારિત સેવાઓ પૂરી પાડી શકાય છે. અને તેમણે આ સેવાઓ માટે ચુકવણી કરવા તૈયાર રહેવું પડે છે.

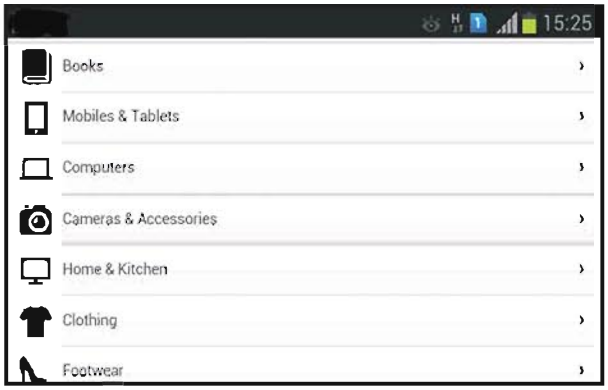
મોબાઈલ દ્વારા ખરીદી (Mobile Purchase)

મોબાઇલ દ્વારા ખરીદીની પ્રક્રિયા ગ્રાહકને કોઈ પણ સમયે ને સ્થળે ઑનલાઇન ખરીદીની સુવિધા પૂરી પાડે છે. ગ્રાહકો ઉત્પાદનની જાણકારી મેળવી ચુકવણીની સુરક્ષિત પદ્ધતિનો ઉપયોગ કરી ખરીદીનો ઑર્ડર આપી શકે છે. છૂટક વેપારી ગ્રાહકને લેખિત સૂચિને બદલે તેને જોઈતાં ઉત્પાદનોની યાદી સીધી જ તેના મોબાઇલ પર મોકલી શકે છે. આ જ રીતે ગ્રાહક છૂટક વેપારીની ઇ-કોમર્સ સાઇટની મોબાઇલ આવૃત્તિની મુલાકાત લઈ શકે છે. છૂટક વેપારીઓ ગ્રાહકનો સંપર્ક સાધી તેમને સ્થાનિક દુકાનો પર મળતા વળતર વિશેની જાણકારી પૂરી પાડી શકે છે.

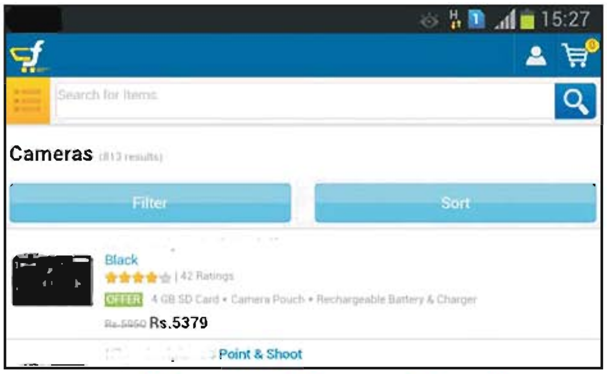
આકૃતિ 5.7(a)માં મોબાઇલ ઉપકરણ પર જોવા મળતું ફ્લિપકાર્ટનું હોમપેજ રજૂ કરવામાં આવ્યું છે. મોબાઇલ દ્વારા થતી ખરીદી સંદર્ભે ઉત્પાદનના વિભાગની પસંદગી, ઉત્પાદનની પસંદગી, ઉત્પાદન અંગેની માહિતી, ઑર્ડર-પ્રક્રિયા અને વિગતો આકૃતિ 5.7 (bથી f)માં દર્શાવવામાં આવેલ છે.



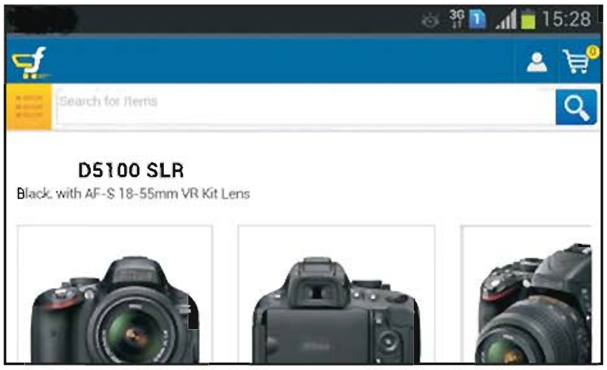
આકૃતિ 5.7(a) : ફ્લિપકાર્ટનું હોમપેજ



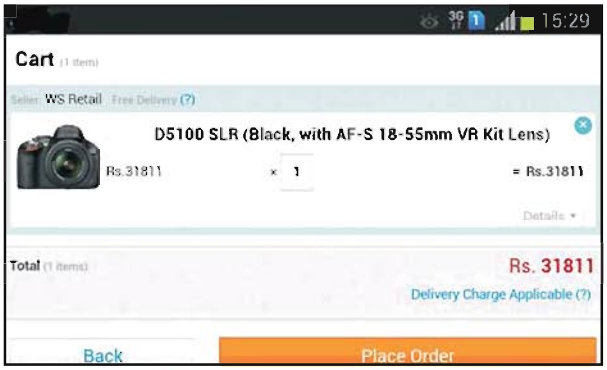
આકૃતિ 5.7(b) : વિભાગની પસંદગી



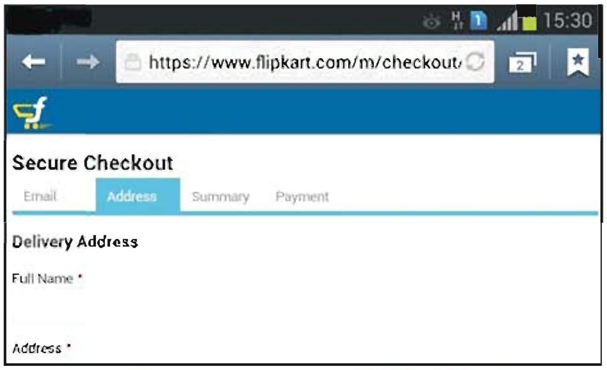
આકૃતિ 5.7(c) : વિભાગના ઉત્પાદનો



આકૃતિ 5.7(d) : ઉત્પાદનની પસંદગી



આકૃતિ 5.7(e) : ઑર્ડર-પ્રક્રિયા



આકૃતિ 5.7(f) : ગ્રાહક સંબંધિત વિગતો

મોબાઈલ દ્વારા નાણાકીય સેવાઓ (Mobile Financial Services)

આજકાલ, ઘણી પ્રતિષ્ઠિત બૅન્કો અને નાણાકીય સંસ્થાઓ એમ-કૉમર્સનો ઉપયોગ કરે છે. તેઓ તેમના ગ્રાહકોને મોબાઇલ ફોન કે અન્ય મોબાઇલ સાધનો દ્વારા તેમનાં ખાતાંની જાણકારી, શેરનું ખરીદ-વેચાણ, દેવાં-માફી વગેરે જેવી સુવિધાઓ પૂરી પાડે છે. યુનિયન બૅન્ક દ્વારા પૂરી પાડવામાં આવતી મોબાઇલ-સેવા આકૃતિ 5.8માં દર્શાવી છે. ઍન્ડરોઇડ માર્કેટમાંથી ગ્રાહક "umobile" નામની ઍપ્લિકેશન ડાઉનલોડ કરી શકે છે. નોંધણી થઈ ગયા બાદ, આકૃતિ 5.9માં દર્શાવ્યા મુજબ ગ્રાહક તેના મોબાઇલ સાધન પર બૅન્ક દ્વારા પૂરી પાડવામાં આવેલી સેવાઓ મેળવી શકે છે. મુખ્ય મેનુમાં વિવિધ વિકલ્પો આકૃતિ 5.10માં જોઈ શકાય છે. ગ્રાહક તેની ખાતાકીય સિલક તપાસી શકે છે, અન્ય બૅન્કનાં ખાતાંમાં રકમનું હસ્તાંતરણ કરી શકે છે, ચૅકબુક મેળવવા માટે વિનંતી કરી શકે છે અને આ સિવાય પણ અન્ય પ્રકારની અનેક સેવાઓ પોતાના મોબાઇલ ઉપકરણ દ્વારા મેળવી શકે છે.

આકૃતિ 5.8 : મોબાઈલ દ્વારા નાણાકીય સેવાઓ

આકૃતિ 5.9 : પાસવર્ડની સહાયથી લૉગ-ઇન

આકૃતિ 5.10 : વિવિધ સેવાઓની યાદી

મોબાઈલ દ્વારા માહિતીસેવા (Mobile Information Services)

અંગત કમ્પ્યૂટરની જેમ જ મોબાઈલધારકોને વિપુલ પ્રમાણમાં માહિતીને લગતી સેવાઓ પૂરી પાડી શકાય છે. આમાં સમાવિષ્ટ છે :

- સમાચારસેવા
- શેરબજારની વિગતો
- ખેલકૂદ-સમાચાર
- નાણાકીય નોંધ
- યાતાયાતની માહિતી

સ્થળ અને શોધસેવા (Location and Search Service)

મોબાઈલ કૉમર્સના ઉપયોગ સમયે ઉપયોગકર્તાના મોબાઈલ ફોનનું સ્થાન એક અગત્યની માહિતી બની રહે છે. ઉદાહરણ તરીકે, મર્યાદિત કિંમતમાં અને ચોક્કસ વિશિષ્ટતાઓ સાથે ઉપલબ્ધ સ્માર્ટફોનને ખરીદવા માંગતા ઉપયોગકર્તાને તેની નજીકની દુકાનનું સ્થાન જાણવામાં રસ હશે કે જ્યાંથી તે પોતાની અપેક્ષા મુજબનું ઉત્પાદન મેળવી શકે. સ્થાન અને શોધની સેવા ઉપયોગકર્તાને તેના વર્તમાન સ્થાનથી શહેરની નિકટવર્તી દુકાનોની માહિતી મળે એ જરૂરી છે. ઉપયોગકર્તાના મોબાઈલના સ્થાન અંગેની જાણકારી વ્યાપારીને સ્થાનિક નકશા, સ્થાનિક રજૂઆતો, સ્થાનિક હવામાન, વ્યક્તિ અંગેની ખબર અને દેખરેખ જેવી સ્થળ આધારિત સેવાઓ પૂરી પાડવાની સુવિધા આપે છે. મોબાઈલ સાધનોની મદદથી કોઈ નિશ્ચિત જગ્યા, સિનેમાગૃહ, ભોજનાલય, દવાખાનું કે અન્ય સુવિધાઓ માટેની માહિતી મેળવી શકાય છે. L-commerce નામથી ઓળખાતી સ્થળ આધારિત શોધ વિશે અભ્યાસ કરીએ.

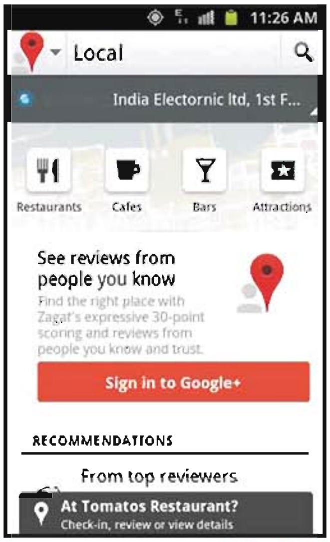
એલ-કૉમર્સ (L-Commerce)

PDA, સેલ્યુલર ફોન અને પૉકેટ પીસી જેવા વધુ ને વધુ વાયરલેસ સાધનોએ એમ-કૉમર્સની વૃદ્ધિ માટે નોંધપાત્ર તકો ઊભી કરી છે. ગ્રાહક કે વિક્રેતાની ઉત્પાદન તથા સેવા સંબંધી બધી જ આવશ્યકતાઓની પૂર્તિ માટે મોબાઈલ કૉમર્સ સક્ષમ સાબિત થયું છે. આમ છતાં, કેટલીક વિશિષ્ટ સ્થિતિઓમાં ગ્રાહક અને વિક્રેતાઓનું નિશ્ચિત સ્થાન વ્યવહાર માટે મહત્ત્વનું હોય છે. આજકાલ સ્થાન આધારિત ઘણા વિનિયોગો અને સેવાઓ આપવામાં આવે છે. આ વિનિયોગો સેવા કે ઉત્પાદનને પહોંચાડવા માટે ઉપયોગકર્તાના સ્થાનની દેખરેખ રાખે છે. વ્યાવસાયિક હેતુ માટે સ્થાનની માહિતી પૂરી પાડતી તક્નિકના ઉપયોગને એલ-કૉમર્સ તરીકે ઓળખવામાં આવે છે.

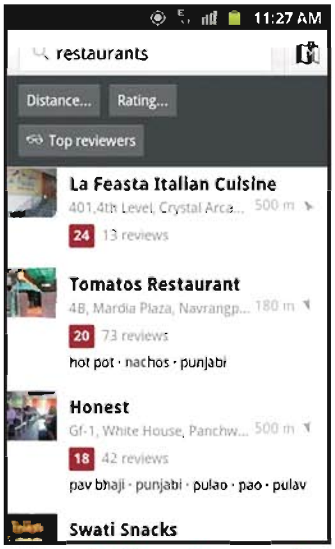
મોબાઈલ ઉપકરણના ભૌતિક સ્થાનને આધારે તક્નિક વિનિયોગના ઉપર્યુક્ત ક્ષેત્રનું નિર્ધારણ કરે છે. અને તેના દ્વારા ઉપયોગકર્તા સ્થળવિશેષને લૉગ-ઇન કરી શકે છે, અન્ય વ્યક્તિનું સ્થાન જાણી શકે છે અને બૅન્ક કે ભોજનાલય જેવાં સ્થળ શોધી શકે છે. આ તક્નિક GPS, સેલ્યુલર અને Wi-Fi સ્રોત તરફથી મળતા સિગ્નલ પર કાર્ય કરે છે. મોબાઈલ ઉપકરણોની સ્થિતિ નક્કી કરવા માટેની સૌથી વધુ ચોક્કસ પદ્ધતિ ગ્લોબલ પોઝિશનિંગ સિસ્ટમ (Global Positioning

System - GPS) છે. તે સમગ્ર વિશ્વમાં વ્યાપ્ત સેટેલાઇટ ટ્રેકિંગ પદ્ધતિ પર આધારિત છે, જ્યાં પૃથ્વીની આસપાસ પરિભ્રમણ કરતા ત્રણ ઉપગ્રહોના સમૂહ દ્વારા GPS સિગ્નલ દ્વારા ઉત્પન્ન કરેલાં સિગ્નલના છેદબિંદુનો ઉપયોગ કરવામાં આવે છે. તે 500 મીટરના ઘેરાવામાં ઉપકરણનું સ્થાન નક્કી કરે છે. આને ત્રિકોણીય (Triangulation) તરીકે ઓળખવામાં આવે છે. GPS સિગ્નલ અલ્પ કે અવરુદ્ધ હોય, તો મોબાઇલ ઉપકરણ સેલ ટાવર અને Wi-Fi હોટ સ્પોટ તરફથી મળતા સિગ્નલનો ઉપયોગ કરી શકે છે. આ સિગ્નલ સ્વયં સ્થાનને પ્રસારિત કરી શકતા નથી, પરંતુ સ્માર્ટફોન સંસ્થાઓ પોતાના ડેટાબેઝમાં સંગૃહીત સ્રોતનો ઉપયોગ કરી આ સ્થાનની જાણકારી આપે છે.

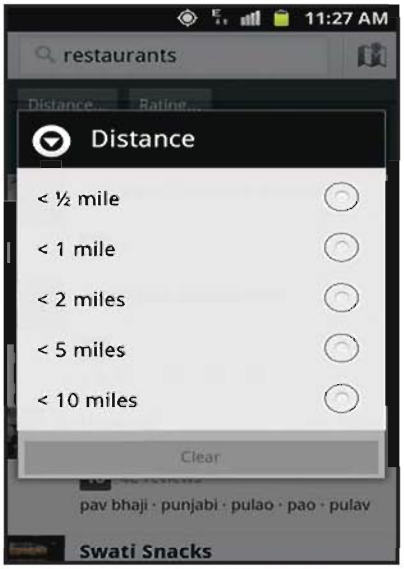
GPS દ્વારા મોબાઇલ સાધનનો ઉપયોગ કરી ઉપયોગકર્તાના સ્થાનની જાણકારી મેળવવા માટેનો વિનિયોગ આકૃતિ 5.11માં દર્શાવ્યો છે. (**નોંધ :** આ માટે સાધન પર GPSનું સમર્થન હોવું જરૂરી છે.) આકૃતિ 5.11માં મોબાઇલ સાધન પર દર્શાવવામાં આવેલું વિજ્ઞાપન પણ સ્થાનઆધારિત છે. આકૃતિમાં ઉપયોગકર્તાનું વર્તમાન મોબાઇલ સ્થાન સી.જી.રોડ, અમદાવાદ છે. જો ઉપયોગકર્તા તેના સ્થાનથી નજીક આવેલા ભોજનાલય શોધવા ઇચ્છતો હોય, તો તે આકૃતિ 5.12માં દર્શાવ્યા મુજબનો વિકલ્પ પસંદ કરશે. આમ કરવાથી તેના સ્થાનની નજીક આવેલ ભોજનાલયની યાદી દર્શાવવામાં આવશે. ઉપયોગકર્તા આ શોધ માટે નિશ્ચિત અંતરની સ્પષ્ટતા પણ કરી શકે છે, જે આકૃતિ 5.13માં દર્શાવ્યું છે. આમ કરવાથી આકૃતિ 5.14માં દર્શાવ્યા મુજબ શોધના પરિણામને વધુ ફિલ્ટર કરીને રજૂ કરી શકાય છે.



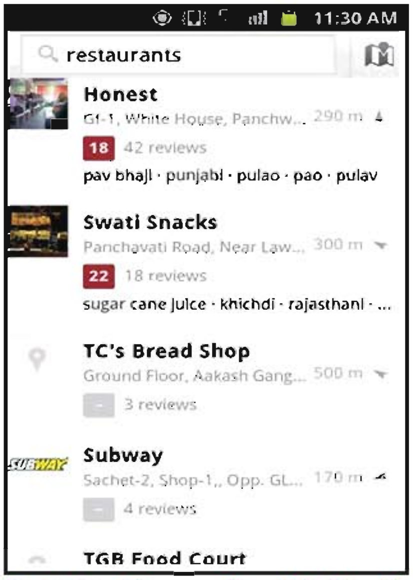
આકૃતિ 5.11 : સ્થાન આધારિત વિનિયોગ



આકૃતિ 5.12 : ઉપયોગકર્તાના સ્થાનને અનુરૂપ નજીકના ભોજનાલયની શોધ

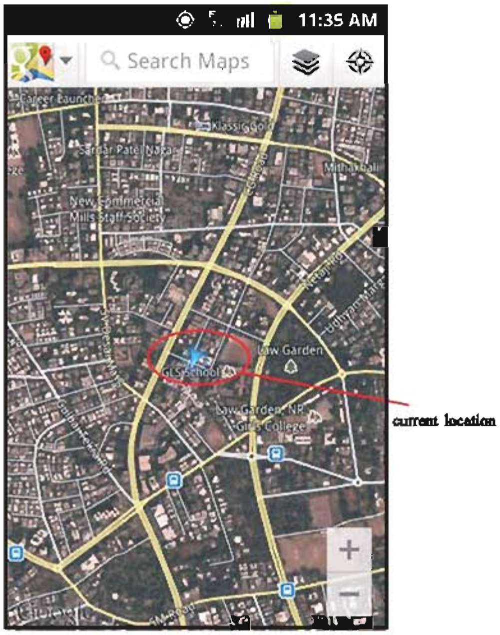


આકૃતિ 5.13 : શોધપરિણામને ચાળવું

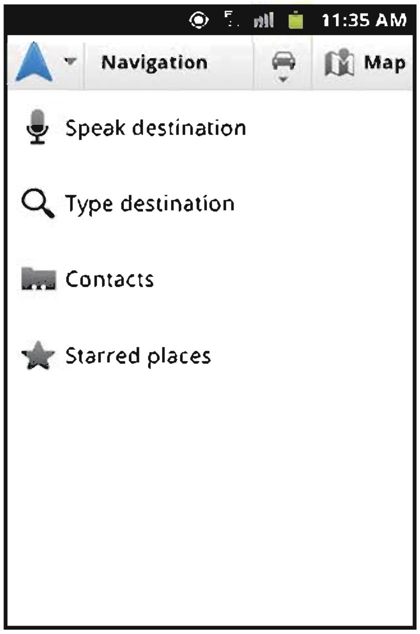


આકૃતિ 5.14 : ફિલ્ટર કર્યા પછીનું શોધપરિણામ

હવે, સ્થાન આધારિત સેવાઓના એક અન્ય ઉદાહરણની ચર્ચા કરીએ. જેમાં આપણે ગંતવ્યસ્થાન સુધીનો માર્ગ શોધવા માટે નકશાનો ઉપયોગ કરી શકીએ છીએ. ઉદાહરણ તરીકે, ઉપયોગકર્તાને ગંતવ્યસ્થાન સુધી પહોંચવું છે પરંતુ તેનો માર્ગ તે જાણતો નથી. આકૃતિ 5.15માં દર્શાવ્યા મુજબ નકશાના ઉપયોગ દ્વારા સ્થાન આધારિત સેવાઓ ઉપયોગકર્તાનું વર્તમાન સ્થાન શોધી શકે છે, જેને ભૂરા રંગના નિર્દેશક દ્વારા દર્શાવવામાં આવે છે. હવે આકૃતિ 5.16માં દર્શાવ્યા મુજબ આપણે ગંતવ્યસ્થાનનું નામ ટાઇપ કરીને કે બોલીને તેનો નિર્દેશ કરી શકીએ છીએ.

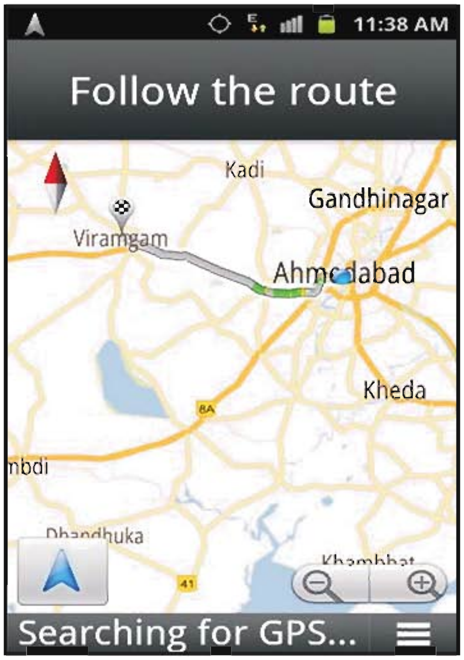


આકૃતિ 5.15 : ઉપયોગકર્તાનું સ્થાન શોધવા માટે નકશાનો ઉપયોગ



આકૃતિ 5.16 : ગંતવ્યસ્થાન ઉમેરવા માટેના વિકલ્પો

આકૃતિ 5.17 : ગંતવ્યસ્થાન નક્કી કરવું



આકૃતિ 5.18 : ગંતવ્યસ્થાન સુધીનો માર્ગ

આકૃતિ 5.17માં દર્શાવ્યા મુજબ ગંતવ્યસ્થાન તરીકે 'વીરમગામ' પસંદ કરવામાં આવ્યું છે. આકૃતિ 5.18માં આપેલ નકશો સ્રોતસ્થાનથી ગંતવ્યસ્થાન સુધીનો માર્ગ દર્શાવે છે.

આકૃતિ 5.19 : ગંતવ્યસ્થાન સુધી પહોંચવા માટેનું અંતર અને સંભવિત સમય

આકૃતિ 5.20 : વાહનચાલકને માર્ગદર્શન

આકૃતિ 5.19માં ગંતવ્યસ્થાને પહોંચવા માટેનું અંતર અને અંદાજિત સમય પણ દર્શાવવામાં આવ્યા છે. આકૃતિ 5.20માં દર્શાવ્યા મુજબ ગંતવ્યસ્થાન સુધી પહોંચવા માટેની મદદ પણ મેળવી શકાય છે સ્થાન આધારિત સેવાઓ ગ્રાહકને જરૂરિયાત અનુસાર વધુ ઝડપી અને નિશ્ચિત સેવાઓ માટે અનુકૂલન અને અવસર પૂરાં પાડે છે. સ્થાન આધારિત કેટલીક સેવાઓનાં ઉદાહરણ નીચે દર્શાવ્યાં છે :

- માહિતી અથવા નિર્દેશન-સેવાઓ (Information or directory services) : ક્રિયાશીલ 'યલોપેજ' ઉપયોગકર્તાને આપોઆપ નજીકના ભોજનાલય, પાર્કિંગ-સુવિધા, યાતાયાતની માહિતી વગેરેથી માહિતગાર કરે છે.
- અનુસંધાન-સેવા (Tracking services) : અકસ્માતની શોધખોળ, મિત્રના ભૌગોલિક સ્થાનની શોધ, ખોવાયેલી મોટરકારની શોધ, વાલી દ્વારા પોતાનાં બાળકોની શોધ વગેરે.
- સંકટકાળ-સેવાઓ (Emergency services) : સંકટ સમયે આરોગ્યલક્ષી અને ઍમ્બ્યુલન્સ સેવા, શોધ અને બચાવ-કામગીરી, માર્ગ અકસ્માતોમાં સહાય, પોલીસ, સંરક્ષણ અને અગ્નિશામક સહાય.
- વિજ્ઞાપન દ્વારા પ્રોત્સાહન (Advertising promotion) : લક્ષ્યવેધી વિજ્ઞાપનો, પ્રોત્સાહક સંદેશાઓ, દુકાનમાં ગ્રાહકની ઓળખાણ.
- નકશા (Mapping) : વિશિષ્ટ ભૌગોલિક સ્થાનના નકશાઓની રચના.
- નેવિગેશન (Navigation) : એક સ્થળથી અન્ય સ્થળ સુધીનું દિશાસૂચન

ઈ-કૉમર્સ અને એમ-કૉમર્સમાં સુરક્ષાની સમસ્યાઓ (Security Issues in E-commerce & M-commerce)

ઈન્ટરનેટ એવું સાર્વજનિક નેટવર્ક છે, જેમાં હજારોની સંખ્યામાં અંગત કમ્પ્યૂટર નેટવર્ક પરસ્પર સંલગ્ન હોય છે. આ અંગત નેટવર્ક ક્યારેક સાર્વજનિક નેટવર્ક પર સુરક્ષારહિત બની જતું હોય છે. વ્યવસાય માટે ઈન્ટરનેટ સારી તકો પૂરી પાડે છે, પણ અનુકૂળતાઓ સાથે હંમેશાં નવાં જોખમો આવતાં હોય છે. ઈન્ટરનેટ પર સ્થાનાંતરિત થતી મહત્ત્વની વિગતો કે માહિતીનો દુરુપયોગ થઈ શકે તેની ઉચાપત થઈ શકે કે તેને વિકૃત કે નષ્ટ કરી દેવામાં પણ આવે. ઉદાહરણ તરીકે, ઈ-કૉમર્સ વેબસાઈટ દ્વારા ઑનલાઈન ખરીદી કરતી વખતે ગ્રાહકે ક્રેડિટકાર્ડનો નંબર અને અંગત માહિતી પૂરી પાડવી પડે છે. આ માહિતી વ્યાપારીના સર્વર પર મોકલવામાં આવે છે. વ્યાપારીનું સર્વર આ માહિતી પેમેન્ટ ગેટ-વે દ્વારા કાર્ડ આપનાર બૅન્કને ખરાઈ માટે મોકલી આપે છે. આ તમામ વ્યવહારો ઈન્ટરનેટ જેવાં સાર્વજનિક નેટવર્ક પર કરવામાં આવે છે. આ પ્રક્રિયા દરમિયાન અનધિકૃત ઉપયોગકર્તા ક્રેડિટકાર્ડનો નંબર જાણી લઈને ભવિષ્યમાં તેનો

દુરુપયોગ કરી શકે છે. તદુપરાંત, ઑર્ડરની માહિતીને અધવચ્ચે બદલી નાંખવામાં આવે તેવી પણ સંભાવના છે. ગ્રાહકે 10 વસ્તુઓનો ઑર્ડર આપ્યો હોય, પરંતુ કોઈક કારણોસર વિક્રેતાને 100 વસ્તુઓનો ઑર્ડર મળે, તો તે 100 વસ્તુઓની કિંમત ચૂકવવા માંગણી કરશે. કોઈક ઘૂસણખોર દુનિયાના કોઈ પણ ખૂણેથી પોતાના કમ્પ્યૂટર દ્વારા માહિતી ચોરી લઈ શકે છે અને તેમાં ફેરફાર પણ કરી શકે છે. તે પોતાની અંગત ઓળખ છુપાવી નવા પ્રોગ્રામની રચના દ્વારા દૂરસ્થ કમ્પ્યૂટર પર ભંગાણ સર્જી શકે છે અને અતિખરાબ સ્થિતિમાં કમ્પ્યૂટરને સ્થગિત પણ કરી શકે છે.

ઇ-કૉમર્સ/એમ-કૉમર્સની સાઇટ પર ગ્રાહકની અંગત માહિતી, તેમની બૅન્ક અંગેની માહિતી અને તેના જેવી અનેક મહત્ત્વપૂર્ણ માહિતીઓ મૂકવામાં આવે છે, માટે સંસ્થાઓએ આવી તમામ પ્રકારની છેતરપિંડીથી સચેત રહેવું જરૂરી છે. ઇ-કૉમર્સમાં ચૂકવણી માટે ઑનલાઇન બૅંકિંગ, ઇલેક્ટ્રૉનિક વ્યવહાર, ક્રેડિટકાર્ડ અને ડેબિટકાર્ડનો ઉપયોગ થતો હોવાથી, ઇ-કૉમર્સ/એમ-કૉમર્સ વેબસાઇટમાં સુરક્ષા અંગેની વધુ સમસ્યા જોવા મળે છે. અન્ય વેબસાઇટ કરતા આ પ્રકારની વેબસાઇટ છેતરપિંડીનો વધુ ભોગ બને છે. આમ, ઇન્ટરનેટ પર વિગતોની સુરક્ષાનું વિશેષ મહત્ત્વ છે. ઇ-કૉમર્સ/એમ-કૉમર્સની સુરક્ષા નીચે દર્શાવેલ ચાર અગત્યના મુદ્દા અનુસાર હોવી જરૂરી છે :

- **ગુપ્તતા (Confidentiality)**

  અનધિકૃત ઉપયોગકર્તા વાંચી ન શકે તે માટે અહીં માહિતીની ગુપ્તતાનો નિર્દેશ કરવામાં આવ્યો છે, જે સાંકેતીકરણ (cryptography) દ્વારા પ્રાપ્ત કરી શકાય છે. તેમાં તમામ સંદેશાઓ સાંકેતિક લખાણમાં પ્રસારિત કરવામાં આવે છે. અને યોગ્ય કી (key)નો ઉપયોગ કરી સંદેશને મૂળ રૂપમાં ફેરવ્યા પછી માત્ર પ્રાપ્તકર્તા જ તે વાંચી શકે છે. આ પ્રક્રિયા અંગત વિગતોને બાહ્ય આક્રમણથી સુરક્ષિત રાખે છે અને ખાતરી આપે છે કે સંદેશ ગંતવ્યસ્થાને પહોંચતા સુધીમાં જાહેર થયેલ નથી કે અન્ય કોઈને તેની જાણ થયેલ નથી. ક્રેડિટકાર્ડના નંબર જેવી ખાનગી વિગતોની સુરક્ષામાં આ પ્રક્રિયા મદદરૂપ બને છે.

- **અખંડિતતા (Integrity)**

  તે ખાતરી આપે છે કે પ્રસારણ દરમિયાન માહિતી આકસ્મિક રીતે કે દ્વેષપૂર્વક બદલવામાં આવી નથી કે તેની સાથે કોઈ ફેરફાર કરવામાં આવ્યો નથી. પ્રાપ્તકર્તા એ જ સંદેશ મેળવે છે, જે પ્રેષક દ્વારા મોકલવામાં આવ્યો હતો. જો પ્રસારણ દરમિયાન સંદેશ બદલાયો હોય, તો તે શોધી શકાવો જોઈએ. આમ કરવાથી ઑર્ડરના જથ્થામાં થતા ફેરફારની ક્ષતિ દૂર કરી શકાય છે, જે પછીથી ઉદ્ભવતી ચુકવણીની ક્ષતિને પણ રોકે છે.

- **અધિકૃતતા (Authorization)**

  તે ખાતરી આપે છે કે માત્ર અધિકૃત ઉપયોગકર્તાને સિસ્ટમના ઉપયોગની અનુમતિ આપવામાં આવી છે. અધિકૃતતા મેળવવા માટે લૉગ-ઇન અને પાસવર્ડ એક માર્ગ છે.

- **અસ્વીકાર (Non-repudiation)**

  સંદેશ મોકલનાર વ્યક્તિ સંદેશ મોકલ્યાનો ઇનકાર કરી શકે નહીં, તેની અહીં ખાતરી આપવામાં આવે છે. મોકલનારે ખરેખર સંદેશો મોકલ્યો હોય, તો તેના અસ્વીકારને અટકાવે છે. ઉદાહરણ તરીકે, જો ગ્રાહક ખરીદ-ઑર્ડર મોકલ્યા છતાં કોઈ કારણસર તેનો ઇનકાર કરે તો સાબિત થઈ શકે છે કે ગ્રાહકે તે સંદેશ મોકલ્યો છે. સામાન્ય રીતે ડિજિટલ સહી (digital signatures) અથવા વિશ્વસનીય ત્રાહિત પક્ષ (Trusted Third Party - TTP) દ્વારા તેને પ્રમાણિત કરવામાં આવે છે.

ઇન્ટરનેટ સુરક્ષાનાં ભયસ્થાનો (Internet Security Threats)

ઇન્ટરનેટ પર સામાન્ય રીતે જે ધમકીઓનો સામનો કરવો પડે છે, તે નીચે મુજબ છે :

- **દૂષિત કોડ (Malicious code)**

  આપણા કમ્પ્યૂટર કે સિસ્ટમને નુકસાન પહોંચાડતા કોડને દૂષિત કોડ તરીકે ઓળખવામાં આવે છે. દૂષિત કોડ જાતે જ સક્રિય બને છે અથવા ઉપયોગકર્તા કોઈ સ્થાને ક્લિક કરે અથવા તો ઇ-મેઇલનું જોડાણ (attachment) ખોલે, ત્યારે કોઈ વાઇરસની જેમ તે કાર્ય કરે છે અને માહિતી ચોરી લે છે અથવા તો ફાઇલને નષ્ટ કરી વધુ નુકસાન પણ પહોંચાડી શકે છે.

- **સ્નિફિંગ (Sniffing)**

સ્નિફર એવો પ્રોગ્રામ છે જે ઇન્ટરનેટનો ઉપયોગ કરી કમ્પ્યૂટર કે રાઉટર દ્વારા પ્રેષકથી પ્રાપ્તકર્તા સુધી પ્રસારિત થતી માહિતીની નોંધ કરે છે. સ્નિફર-પ્રોગ્રામનો ઉપયોગ ટેલીફોન વાયરને જોડીને વાર્તાલાપની નોંધ કરવા સમાન છે. સ્નિફર-પ્રોગ્રામ ઇ-મેઇલ સંદેશ, ઉપયોગકર્તાનું લૉગ-ઇન, પાસવર્ડ અને ક્રેડિટકાર્ડ નંબર વાંચી શકે છે.

- **સેવાના અસ્વીકાર સ્વરૂપે આક્રમણ (Denial of Service Attack)**

આ એક એવું આક્રમણ છે જેના દ્વારા મશીન કે નેટવર્ક બંધ થઈ જાય છે અને તેના ઉપયોગકર્તાઓ માટે તે નિરુપયોગી બની જાય છે. હુમલાખોર ઉપયોગકર્તાના કમ્પ્યૂટર અને તેના નેટવર્કજોડાણને કે સાઇટને લક્ષ્ય બનાવી ઉપયોગકર્તાને ઇ-મેઇલ, વેબસાઇટ અને અસરગ્રસ્ત કમ્પ્યૂટર પર રહેલી બૅંકિંગ કે અન્ય સેવાઓ પૂરી પાડતાં ઑનલાઇન ખાતાંઓનો ઉપયોગ કરતાં અટકાવે છે. ઉપયોગકર્તાને હજારો સંદેશ મોકલવામાં આવે છે, જે નેટવર્ક પર યાતાયાતની મુશ્કેલીઓ ઊભી કરે છે.

- **સાયબર જંગાલિયાત (Cyber Vandalism)**

સાયબર જંગાલિયાત એ હયાત વેબસાઇટ પર કરવામાં આવતી ઇલેક્ટ્રૉનિક ભાંગફોડિયા પ્રવૃત્તિ છે. આક્રમણકાર વેબસાઇટની મૂળભૂત વિગતોને પોતાની વિગતો સાથે બદલી નાંખે છે. તે અખંડિતતા ઉલ્લંઘનનું ઉદાહરણ છે. તે મિલકતનો નાશ અથવા કોઈની તસવીર પર કરવામાં આવતા દૂષિત ફેરફારોનું ઇલેક્ટ્રૉનિક સ્વરૂપ છે. આજકાલ સાયબર જંગાલિયાતના અનેક કિસ્સા બને છે, જેમાં વ્યાવસાયિક વિગતોને અરુચિકર સામગ્રી સાથે બદલવામાં આવે છે.

- **છેતરપિંડી (Spoofing)**

છેતરપિંડી અથવા છદ્મવેશીપણું (masquerading) એટલે અન્ય વ્યક્તિ હોવાનો સ્વાંગ ભરવો અથવા અન્યની વેબસાઇટનો પોતાની અધિકૃત વેબસાઇટ ગણાવવી. આ તકનિકમાં હુમલાખોર ત્રાહિત વ્યક્તિ કે વેબસાઇટની ઓળખ ધારણ કરી વ્યવહાર કરે છે. ઉદાહરણ તરીકે, હુમલાખોર www.gswan.co.in નામની નકલી વેબસાઇટ બનાવી શકે છે અને તેના IP સરનામાને મૂળ વેબસાઇટના IP સરનામા સાથે બદલી નાખે છે. આમ, મૂળ સાઇટની મુલાકાત લેવા ઇચ્છતા તમામ ઉપયોગકર્તાઓ નકલી વેબસાઇટ તરફ જાય છે.

સુરક્ષાના ઉપાયો (Security Measures)

સુરક્ષા સામેની જુદી-જુદી ધમકીઓને પહોંચી વળવા ઘણા ઉપાયો યોજવામાં આવે છે. તેમાંના કેટલાકની ચર્ચા કરીએ.

- **ઍન્ટિવાઇરસ સૉફ્ટવેર (Antivirus Software)**

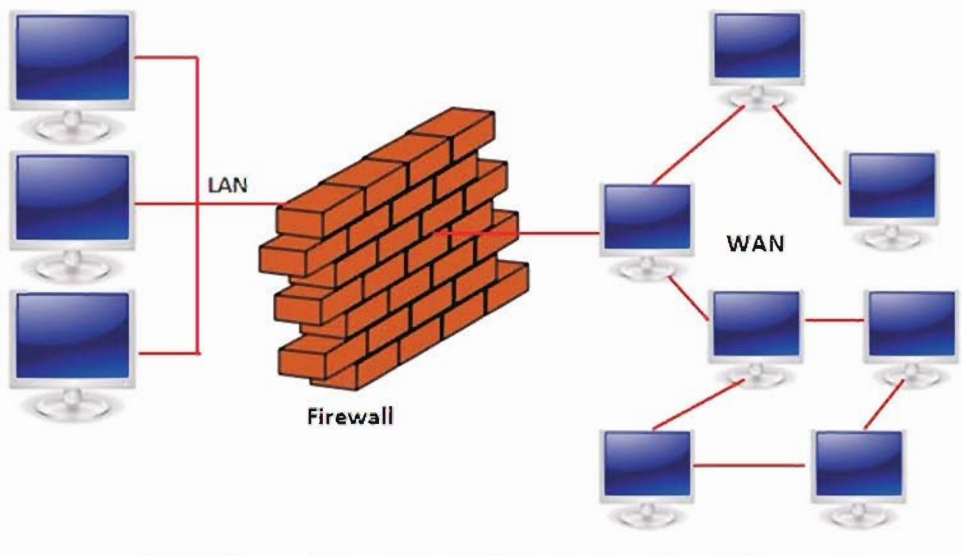
ઍન્ટિવાઇરસ સૉફ્ટવેર એ એવો કમ્પ્યૂટર-પ્રોગ્રામ છે, જે અસરગ્રસ્ત સિસ્ટમમાં આવેલ વાઇરસ, વૉર્મ અને ટ્રોજન હોર્સ જેવા દુષિત કોડને શોધે છે, અટકાવે છે અને દૂર કરવા જરૂરી પગલાં લે છે. કમ્પ્યૂટરની સુરક્ષા માટે સારું ઍન્ટિવાઇરસ સૉફ્ટવેર જરૂરી છે. ઍન્ટિવાઇરસ સૉફ્ટવેર વગરની સિસ્ટમ ઇન્ટરનેટ પર બહુ અલ્પ સમયમાં દૂષિત કોડનું સરળ લક્ષ્ય બની જાય છે. દૂષિત કોડને કારણે તેને અનેક પ્રકારની મુશ્કેલીઓ કે નુકસાનમાંથી પસાર થવું પડે છે. દૂષિત કોડનો આ ચેપ વિચિત્ર ધ્વનિ કે પોપ-અપ વિન્ડો ઉત્પન્ન કરવી અને અન્ય ત્રાસજન્ય કાર્યો કરવા જેવા સામાન્ય પણ હોઈ શકે છે. તે ફાઇલોને નષ્ટ કરી શકે છે, સિસ્ટમને ધીમી પાડી દે છે અથવા હાર્ડવેરને નુકસાન કરી આખી કમ્પ્યૂટર સિસ્ટમનો નાશ પણ કરી શકે છે. એક વાર વાઇરસથી સિસ્ટમ અસરગ્રસ્ત બને પછી સિસ્ટમ સાથે જોડાયેલા અન્ય પ્રોગ્રામ અને ફાઇલોમાં પણ તે પ્રસરે છે. વાઇરસ માત્ર પોતાની પ્રતિકૃતિ ઉત્પન્ન કરવાનું કાર્ય કરતા નથી, પરંતુ ઉપયોગકર્તા દ્વારા સંપર્કસૂચિમાં રહેલી અન્ય વ્યક્તિઓની વિગતોને પણ તે પ્રભાવિત

કરે છે. સિસ્ટમ પર હુમલો થવાનો સૌથી સરળ માર્ગ ઇ-મેઇલ સાથે મોકલવામાં આવતા ઇ-મેઇલનાં દૂષિત જોડાણો (attachments)નો છે. આ જોડાણો ઇ-મેઇલ સાથે જોડી શકાય તેવા ચિત્રો, વીડિયો, ધ્વનિ કે અન્ય પ્રકારનાં હોઈ શકે છે. ઇન્ટરનેટ પરથી કરવામાં આવતા ડાઉનલોડ દ્વારા પણ વાઇરસનો ચેપ ફેલાઈ શકે છે.

સિસ્ટમને થતું નુકસાન રોકવા માટે ઍન્ટિવાઇરસ સૉફ્ટવેર ડાઉનલોડ થયેલ વિગતો સહિત આખી કમ્પ્યૂટર સિસ્ટમને વાઇરસની હાજરી શોધવા માટે તપાસે છે. આજકાલ, ઇન્ટરનેટના ઉપયોગ માટે મોબાઇલ ઉપકરણો પણ પ્રચલિત થવાને કારણે તેના માટે પણ ઍન્ટિવાઇરસ સૉફ્ટવેર ઉપલબ્ધ છે. બજારમાં અનેક પ્રકારનાં ઍન્ટિવાઇરસ સૉફ્ટવેર ઉપલબ્ધ છે. ઍન્ટિવાઇરસ સૉફ્ટવેર કમ્પ્યૂટરમાં સ્થાપિત કરવું અને તેને નિયમિત રીતે અદ્યતન (update) બનાવતા રહેવું અત્યંત જરૂરી છે.

- **ફાયરવૉલ (Firewall)**

પોતાની વેબસાઇટ ધરાવતી સંસ્થાઓએ સંસ્થાની અંદર અને બહાર બંનેની નેટવર્ક-સેવાઓ પર નિયંત્રણ રાખવું પડે છે. સંસ્થાના નેટવર્ક અને બહારની દુનિયા વચ્ચે સામાન્ય રીતે ઉપયોગમાં લેવાતી નેટવર્કની સુરક્ષા-સીમાને ફાયરવૉલ કહે છે. આકૃતિ 5.21માં દર્શાવ્યા મુજબ ફાયરવૉલ એ સંસ્થાના સ્થાનિક નેટવર્ક અને બહારની દુનિયા વચ્ચેના યાતાયાત પર નિરીક્ષણ અને નિયંત્રણ રાખવા માટે મૂકવામાં આવેલું એક સાધન (કમ્પ્યૂટર અથવા રાઉટર) છે. અનધિકૃત વ્યક્તિઓને સંસ્થાની ઇ-કૉમર્સ વેબસાઇટના માળખાથી દૂર રાખવા એ ફાયરવૉલનો પ્રાથમિક હેતુ છે. તે ખાતરી આપે છે કે સંસ્થાની અતિ મહત્ત્વની વિગતો સલામત રીતે સચવાયેલી છે અને અનધિકૃત વ્યક્તિ તેનો દુરુપયોગ કરી શકશે નહિ.



આકૃતિ 5.21 : સ્થાનિક અને સાર્વજનિક નેટવર્ક વચ્ચે આપેલી ફાયરવૉલ

ફાયરવૉલ સ્થાનિક નેટવર્કને નીચેનાં જોખમો સામે સુરક્ષા આપે છે :

- ઇ-મેઇલ સેવા જે કેટલીક વાર મુશ્કેલી ઊભી કરે છે.
- સ્થાનિક નેટવર્કમાં અનપેક્ષિત ચિત્રો, વીડિયો જેવી વિગતોનો પ્રવેશ રોકવો.
- સ્થાનિક નેટવર્કનો અનધિકૃત વ્યક્તિ દ્વારા ઉપયોગ અટકાવવો.
- સંસ્થાના નેટવર્કમાંથી બહાર જતી અનધિકૃત વિગતો કે માહિતી રોકવી.
- બહારની દુનિયામાંથી સ્થાનિક નેટવર્કમાં થતા યાતાયાતને અટકાવવો.
- કોઈ પણ પ્રકારના નેટવર્ક આક્રમણ સામે સુરક્ષા આપવી.

- **ડિજિટલ સર્ટિફિકેટ (Digital Certificate)**

ઇલેક્ટ્રૉનિક વ્યવહારમાં આપણી ઓળખાણ સાબિત કરવા માટે ડિજિટલ સર્ટિફિકેટ કે ડિજિટલ IDનો ઉપયોગ કરવામાં આવે છે. જેમ વાસ્તવિક દુનિયામાં આપણી ઓળખ પૂરી પાડવા ડ્રાઇવિંગ લાઇસન્સ કે પાસપોર્ટની જરૂર હોય છે. તેવી જ રીતે ડિજિટલ સર્ટિફિકેટની મદદથી વ્યાવસાયિક સંસ્થાઓ ઑનલાઇન સેવાઓ અને મિત્રોને ખાતરી કરાવી શકાય છે કે આપણી પાસેથી તેમણે મેળવેલી માહિતી અધિકૃત છે. ધારકની ઓળખ પ્રસ્થાપિત કરવા માટે વિશ્વસનીય ત્રાહિત પક્ષ (trusted third party) દ્વારા ડિજિટલ સર્ટિફિકેટ આપવામાં આવે છે. સર્ટિફિકેટ આપનાર ત્રાહિત પક્ષને સર્ટિફિકેશન ઑથોરિટી (Certification Authority - CA) કહે છે. ડિજિટલ સર્ટિફિકેટમાં ધારકનું નામ, ક્રમ, સમાપ્તિ તારીખ, સંદેશને સાંકેતિક બનાવવા માટે ડિજિટલ સર્ટિફિકેટ ધારકની પબ્લિક કી, ધારકની સહી તથા સર્ટિફિકેટ આપનાર અધિકૃત પક્ષની સહી હોય છે, જેના દ્વારા સર્ટિફિકેટની સત્યતા સાબિત થઈ શકે છે.

- **સાંકેતીકરણ (Cryptography)**

માહિતીને અવાચ્ય રૂપમાં ફેરવી સુરક્ષિત રાખવાની કળાને સાંકેતીકરણ (Cryptography) કહે છે. સાંકેતીકરણની પ્રક્રિયામાં સામાન્ય લખાણ કે સાદા લખાણ (plain text)ને સાંકેતિક અલ્ગોરિધમના ઉપયોગ દ્વારા ખાનગી કે ગુપ્ત લખાણ (cipher text)માં રૂપાંતરિત કરી અવાચ્ય બનાવવામાં આવે છે. સંદેશના સાંકેતીકરણ કે બિનસાંકેતીકરણ માટે ગુપ્ત કી (secret key)નો ઉપયોગ કરવામાં આવે છે. તે લખાણને અદૃશ્ય બનાવતું નથી, પરંતુ લખાણને અન્ય એવા લખાણમાં પરિવર્તિત કરે છે, જે અર્થપૂર્ણ હોતું નથી. તેનો ઉદ્દેશ ઇન્ટરનેટ પર સંદેશના અધિકૃત પ્રાપ્તકર્તા સિવાયની અન્ય વ્યક્તિઓની માહિતીને અવાચ્ય રાખી ગુપ્તતાની જાળવણીનો છે. ઇન્ટરનેટ કે નેટવર્ક પર પ્રસારિત કરતાં પહેલાં સંદેશનું સાંકેતીકરણ (encryption) કરવામાં આવે છે. સાંકેતીકરણ દ્વારા પરિવર્તિત થયેલો ગુપ્ત સંદેશ પ્રાપ્તકર્તાને મળે, ત્યારે તેને મૂળ સ્વરૂપમાં લાવવો જરૂરી છે. આ માટે સાંકેતીકરણ (encryption)થી વિરુદ્ધ એવી બિનસાંકેતીકરણ (decryption)ની પ્રક્રિયાનો અમલ કરવામાં આવે છે. આ પ્રક્રિયા અન્વયે ગુપ્ત લખાણને પુનઃ સાદા લખાણમાં રૂપાંતરિત કરવામાં આવે છે. સાંકેતીકરણ માટેના અનેક અલ્ગોરિધમ બજારમાં ઉપલબ્ધ છે.

છેલ્લાં કેટલાંક વર્ષોમાં એવા અનેક કિસ્સા બન્યા છે, જેમાં પ્રસારિત થતી વિગતોને આંતરવામાં આવી હોય. ઇન્ટરનેટ, ઇ-કૉમર્સ, મોબાઇલ ટેલિફોન, બ્લ્યુટૂથ સાધનો, બૅન્કનાં ઑટોમેટિક ટેલર મશીન (Automatic Teller Machines - ATM) જેવા નેટવર્કમાં પ્રસારણ દરમિયાન વિગતોની સુરક્ષા માટે સાંકેતીકરણનો ઉપયોગ કરવામાં આવે છે. ધારો કે, મિત્રને "HOW ARE YOU?" સંદેશ મોકલવાનો છે. સંદેશને સુરક્ષિત બનાવવા માટે સાંકેતિક ભાષામાં રૂપાંતરિત કરવો પડશે. સાંકેતીકરણનો ઉપયોગ કરી ગુપ્ત સંદેશ મિત્રને મોકલવામાં આવશે. આ ઉદાહરણમાં સાંકેતીકરણની પ્રક્રિયા દ્વારા દરેક અક્ષરને તેના પછીના અક્ષર સાથે બદલીને રજૂ કરવામાં આવ્યો છે. આનો અર્થ એ થયો કે, "A" એ "B" બનશે, "B" એ "C" બનશે અને તેવી રીતે તમામ અક્ષરોને બદલવામાં આવશે. તમે મિત્રને એ પણ જાણ કરી હશે કે આ કોડને સાદા લખાણમાં ફેરવવા માટે "1થી ખસેડો" (shift by 1). આકૃતિ 5.22માં દર્શાવ્યા મુજબ સંદેશને ગુપ્ત ભાષામાં ફેરવવામાં આવશે.

H	O	W	A	R	E	Y	O	U	← Plain text
I	P	X	B	S	F	Z	P	V	← Cipher text

આકૃતિ 5.22 : કીની મદદથી સાંકેતીકરણ

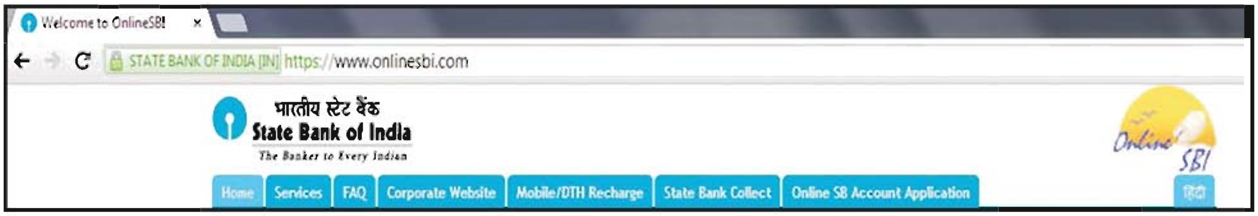
જ્યારે તમારો મિત્ર સંદેશ મેળવશે ત્યારે તેનું બિનસાંકેતીકરણ કરશે. અન્ય કોઈ પણ વ્યક્તિ પ્રસારણ દરમિયાન સંદેશના નિરર્થક અક્ષરો જ જોઈ શકશે. અહીં રાખવામાં આવેલી કી સરળ છે, પરંતુ સામાન્ય રીતે તે ઘણી લાંબી હોય છે.

ઇ-કૉમર્સની સુરક્ષા માટે વિવિધ પ્રોટોકોલ પણ ઉપયોગમાં લેવામાં આવે છે. તેમાંના એક SSL પ્રોટોકોલને સમજીએ.

- **સિક્યોર સૉકેટ લેયર (Secure Socket Layer - SSL)**

આજકાલ દરેક ઉપયોગકર્તા જુદી-જુદી જાતની અનેક વિગતો ઇ-મેઇલ દ્વારા ક્રેડિટકાર્ડ માટે મોકલતો હોય છે. ઉપયોગકર્તા

ઇચ્છે છે કે સાર્વજનિક નેટવર્ક પર પ્રસારિત થતી વખતે આ વિગતો સુરક્ષિત રહે. ઇન્ટરનેટ પરના વેબ-વ્યવહારોની સુરક્ષા માટે SSL પ્રોટોકોલનો ઉપયોગ કરવામાં આવે છે. તેની રચના નેટસ્કેપ દ્વારા કરવામાં આવી હતી. ઇ-કૉમર્સના વ્યવહાર કરી માહિતીનું સુરક્ષિત પ્રસારણ કરવામાં આવે છે. તે ડિજિટલ સર્ટિફિકેટ દ્વારા વ્યાપારી કે દુકાનદારની અધિકૃતતા પ્રમાણિત કરે છે, જેથી ગ્રાહકને માન્ય માલિક સાથેના વ્યવહારની ખાતરી થાય છે. સાઇટ સુરક્ષિત છે કે નહિ તે તેમાં આવેલા લોગો દ્વારા જાણી શકાય છે. જો સાઇટને વેરિસાઇન (VeriSign) દ્વારા સુરક્ષિત કરવામાં આવી હોય તો સાઇટની લોગ ઇન સ્ક્રિન પર વેરિસાઇનનો લોગો જોવા મળે છે. તેના પર ક્લિક કરવાથી માલિક અંગેની માહિતી તથા સર્ટિફિકેટની માન્યતા જાણી શકાય છે. આ દર્શાવે છે કે સાઇટ સાથેનો વ્યવહાર સુરક્ષિત છે, વેબસાઇટનો માલિક માન્ય છે તથા તે સાઇટ વેરિસાઇન જેવી સત્તા દ્વારા પ્રમાણિત છે. સુરક્ષાનો અન્ય નિર્દેશ એ છે કે સુરક્ષિત જોડાણ ધરાવતી સાઇટનું સરનામું આકૃતિ 5.23માં દર્શાવ્યા મુજબ http:// ને બદલે https:// થી શરૂ થાય છે.



આકૃતિ 5.23 : https://નું ઉદાહરણ

ઇ-કૉમર્સ અને એમ-કૉમર્સમાં કાયદાકીય પ્રશ્નો (Legal Issues in E-commerce/M-commerce) :

વ્યવસાય માટે, વૈશ્વિક બજારના સંપર્ક માટે અને ઑનલાઇન ખરીદીની સુવિધા માટેનો અવસર ઇ-કૉમર્સ અને એમ-કૉમર્સ પૂરો પાડે છે. તે વ્યાવસાયિક પ્રક્રિયામાં વૃદ્ધિની તકો પૂરી પાડે છે. જોકે, નવા કોઈ વ્યવસાયની જેમ ઇ-કૉમર્સ અને એમ-કૉમર્સમાં પણ કેટલાંક પ્રશ્નો અને જોખમો રહેલાં છે. બંનેમાં ઘણા કાયદાકીય પડકાર પણ રહેલા છે, કારણ કે રાષ્ટ્રીય સીમાને ધ્યાનમાં રાખ્યા વગર તે વૈશ્વિક ઇન્ટરનેટ પર પ્રવૃત્ત બને છે. દરેક દેશના પોતાના કાયદા અને નિયમો હોય છે. અહીં બૌદ્ધિક મિલકતોના અધિકાર (intellectual property rights), સ્વાધિકાર (copyrights), અને ગુપ્તતા (privacy) તથા કેટલીક વાર પક્ષો વચ્ચેની તકરારને લગતા પડકાર જોવા મળે છે. આ પ્રશ્નોના નિરાકરણ માટે કાયદાકીય માળખું જરૂરી છે. ઘણા દેશોએ ઇલેક્ટ્રૉનિક કૉમર્સ માટે તેમના કાયદાકીય માળખાની સ્થાપના કરી છે. ભારત સરકારે પણ IT કાયદા હેઠળ IT laws (Information technology laws)ની સ્થાપના કરી છે.

કાયદાકીય નીતિ અને નિયમોના પાલનથી ગ્રાહકને વિશ્વાસ બેસે છે કે તેની અંગત વિગતો ખાનગી રહેશે અને તેનો દુરુપયોગ થશે નહીં. જો તેનો દુરુપયોગ કરવામાં આવ્યો હોય, તો તે માટે જવાબદાર પક્ષોને સજા પણ કરી શકાશે. આ પ્રકારની સુરક્ષા ઇ-કૉમર્સ અને એમ-કૉમર્સ માટે અનિવાર્ય છે, કારણકે તેના દ્વારા ગ્રાહક ક્રેડિટકાર્ડની વિગતો જેવી સંવેદનશીલ માહિતીઓ પૂરી પાડે છે. ઑનલાઇન વ્યવસાય કરતી સંસ્થાઓને તેમના લોગો કે કૉપીરાઇટ જેવી ડિજિટલ માહિતીના દુરુપયોગ સામે કાયદાકીય સમર્થન પૂરું પાડવામાં આવે છે. આ કાયદા અન્વયે બે પક્ષો વચ્ચે થયેલ તકરારનું પણ નિરાકરણ લાવી શકાય છે.

બૌદ્ધિક સંપત્તિની સુરક્ષા (Securing Intellectual Property)

બૌદ્ધિક સંપત્તિમાં પુસ્તકો, સૉફ્ટવેર, સંગીત, વીડિયો, કૉપીરાઇટ, ટ્રેડમાર્ક અને વેબપેજનો સમાવેશ કરવામાં આવે છે. તેને સંબંધિત કેટલાક મુદ્દાઓની ચર્ચા કરીએ.

- **સ્વાધિકાર (Copyright)**

  સર્જકના મૂળભૂત કાર્ય પર તેની અનુમતિ વિના અન્ય વ્યક્તિ અધિકાર ધરાવે કે પોતાના ઉપયોગમાં લે તેની સામે રક્ષણ આપતા અધિકારને સ્વાધિકાર (Copyright) કહે છે. પુસ્તકો, સૉફ્ટવેર-પ્રોગ્રામ અને લખાણને તે લાગુ પાડવામાં આવે છે. સ્વાધિકારનો કાયદો બૌદ્ધિક સંપત્તિને અનેક રીતે સુરક્ષા પૂરી પાડે છે. સ્વાધિકાર સંપત્તિ મુક્તપણે ઉપયોગમાં લઈ શકાતી નથી. ઇ-કૉમર્સ કે એમ-કૉમર્સમાં બૌદ્ધિક સંપત્તિની સુરક્ષાનું કાર્ય ઘણું કઠિન છે. ઉદાહરણ તરીકે, તમે

કોઈ સૉફ્ટવેર ખરીદ્યું હોય, તો તમને તેનો ઉપયોગ કરવાનો અધિકાર છે, પરંતુ તેના વિતરણનો નહીં. વિતરણનો અધિકાર માત્ર સ્વાધિકારધારક (copyright holder) પાસે હોય છે. મોટા ભાગનાં વેબપેજ સ્વાધિકારની સુરક્ષા ધરાવતાં હોય છે. વેબપેજ પરથી વિગતોની નકલ કરવી એ પણ સ્વાધિકારના કાયદાનો ભંગ છે.

- **ટ્રેડમાર્ક (Trademark)**

  આ એક નિર્ધારિત લોગો, ચિહ્ન, શબ્દ, નિશાની, શબ્દસમૂહ કે ચિત્ર છે, જેનો ઉપયોગ કોઈ વ્યક્તિગત કે સંસ્થા દ્વારા કોઈ ઉત્પાદન કે સેવાને બજારના અન્ય ઉત્પાદન કે સેવાથી અલગ પાડવા માટે કરવામાં આવે છે. ટ્રેડમાર્કને TM, SM અને ® નિશાની દ્વારા દર્શાવવામાં આવે છે.

- **ડોમેઈન નામની તકરાર (Domain Name Disputes)**

  ડોમેઇન નામ માટેની સ્પર્ધા એ એક અન્ય કાયદાકીય મુદ્દો છે. પહેલાંના સમયમાં ડોમેઇન નામ વહેલા તે પહેલાનાં ધોરણે આપવામાં આવતાં હતાં. પણ લોકો દ્વારા ઉપયોગમાં લેવાતાં ન હોય તેવાં નામની નોંધણી કરાવી લેવામાં આવતી. પછીથી આ ડોમેઇન નામ કોઈક સંસ્થાને ઘણી ઊંચી કિંમતે વેચી દેવામાં આવતું. આને સાયબર સ્ક્વૉટિંગ (cyber squatting) તરીકે ઓળખવામાં આવે છે. બીજી સમસ્યા નામ બદલવા માટેની છે. જેમાં કોઈ પ્રચલિત ડોમેઇન નામની જોડણીમાં ઇરાદાપૂર્વક સામાન્ય ફેરફાર કરી નોંધણી કરાવવામાં આવે છે. જેણે સામાન્ય રીતે URL ટાઇપ કરવામાં ટાઇપિંગની ભૂલ કરી હોય તેવા ગ્રાહક ગેરમાર્ગે દોરાઈ શકે છે.

બૌદ્ધિક સંપત્તિની સુરક્ષા (Protecting Intellectual Property)

બૌદ્ધિક સંપત્તિની સુરક્ષા માટે ઘણી નવી સુધારેલી પદ્ધતિઓનો સતત વિકાસ થઈ રહ્યો છે. તેમાંની કેટલીક પદ્ધતિઓની ચર્ચા અહીં કરવામાં આવી છે.

સ્ટેગ્નોગ્રાફી (Steganography)

એક માહિતીમાં અન્ય માહિતી સંતાડવાની ક્રિયાને સ્ટેગ્નોગ્રાફી (Steganography) કહે છે. જો ફાઇલમાં આવેલી માહિતીને સુરક્ષિત બનાવવામાં ન આવે, તો તેને કોઈ પણ દૂષિત હેતુ માટે ઉપયોગમાં લઈ શકાય છે. તેનું કાર્ય કમ્પ્યૂટરમાં રહેલાં પણ ઉપયોગમાં લેવાયાં ન હોય તેવા ચિત્રો, ધ્વનિ કે લખાણ ધરાવતી ફાઇલને અદૃશ્ય માહિતી સાથે બદલવાનું છે. આ અદૃશ્ય માહિતી સાદું લખાણ, ગુપ્ત લખાણ કે ચિત્ર પણ હોઈ શકે છે. સ્ટેગ્નોગ્રાફી માટે વિશિષ્ટ સૉફ્ટવેર જરૂરી છે અને ઇન્ટરનેટ પરથી સરળતાથી ડાઉનલોડ થઈ શકે તેવી ફ્રી-વેર આવૃત્તિઓ પણ ઉપલબ્ધ છે.

ડિજિટલ ઓળખચિહ્ન (Digital Watermarking)

વૉટરમાર્ક એ ડિજિટલ ચિત્ર, ધ્વનિ કે વીડિયોમાં ઉમેરવામાં આવતો ડિજિટલ કોડ છે જે ફાઇલના સ્વાધિકાર (copyright)ની ઓળખ આપી શકે છે. તે માહિતીનો સંપૂર્ણપણે અદૃશ્ય સ્વરૂપે સંગ્રહ કરવાની સુવિધા આપે છે. શરૂઆતના સમયમાં કલાકારો ચિત્રની ઉપર પીંછી દ્વારા સર્જનાત્મક નિશાની કરી સ્વાધિકારનો હક દાવો કરતા. પરંતુ ડિજિટલની દુનિયામાં કલાકારો ચિત્રની અંદર તેમનું નામ છુપાવી ઓળખચિહ્ન (watermark) બનાવી શકે છે. આમ, સર્જકની ઓળખાણ માટે વૉટરમાર્ક અદૃશ્ય રીતે મદદરૂપ બને છે.

આ અભિગમને ડિજિટલ ઑડિયો અને વીડિયો જેવાં અન્ય માધ્યમોમાં પણ ક્રિયાન્વિત કરી શકાય છે. હાલમાં ઇન્ટરનેટ પર ડિજિટલ ઑડિયોનું MP3 સ્વરૂપમાં અનધિકૃત વિતરણ એક મોટી સમસ્યા છે. ડિજિટલ ઓળખચિહ્ન દ્વારા ઑડિયોના વિતરણને નિયંત્રિત બનાવી શકાય છે. અને અસરકારક રીતે સ્વાધિકાર સુરક્ષા પૂરી પાડી શકાય છે. ડેટાસુરક્ષાના ક્ષેત્રમાં ઓળખચિહ્નનો ઉપયોગ પ્રમાણપત્ર અને અધિકૃતતા માટે પણ કરી શકાય છે. ઉદાહરણ તરીકે, વ્યક્તિના ફોટો-ઓળખપત્ર પર 123456 સંખ્યા લખીને અને તેને ડિજિટલ વૉટરમાર્ક ઓળખચિહ્ન રૂપે અદૃશ્ય રાખીને ફોટો ઓળખને સુરક્ષિત બનાવી શકાય છે. આમ, તસવીરમાં થયેલ ફેરફારને સરળતાથી શોધી શકાય છે.

ડિજિટલ ઓળખચિહ્ન દ્વારા દસ્તાવેજમાં માહિતીને લિંક પણ આપી શકાય છે. ઉદાહરણ તરીકે, સામાન્ય રીતે પાસપોર્ટધારક વ્યક્તિનું નામ સ્પષ્ટ શબ્દોમાં લખાયેલું હોય છે. પરંતુ પાસપોર્ટની તસવીર સાથેના નામને ડિજિટલ ઓળખચિહ્નની મદદથી અદૃશ્ય બનાવી શકાય છે. જો કોઈએ પાસપોર્ટમાં ચેડાં કરી ફોટો બદલી નાખ્યો હોય, તો તે જાણવા માટે પાસપોર્ટને

સ્કેન કરી ફોટા સાથેના અદૃશ્ય નામની ચકાસણી પણ કરી શકાય છે. GIMP જેવા ફોટો-એડિટર પ્રોગ્રામની મદદથી ચિત્રમાં દૃશ્યમાન વૉટરમાર્ક ઉમેરી શકાય છે.

ઇ-કૉમર્સ/એમ-કૉમર્સમાં ચુકવણી (Payment in E-commerce/M-commerce)

ઇ-કૉમર્સ અને એમ-કૉમર્સમાં ચુકવણી એ સૌથી મહત્ત્વનું પાસું છે. ચુકવણીની પરંપરાગત પદ્ધતિઓમાં રોકડ, ચેક કે ક્રેડિટકાર્ડ વડે થતી ચુકવણીઓનો સમાવેશ થાય છે. આજકાલ ઑનલાઇન કરવામાં આવતા વ્યવસાયોમાં ચુકવણીની ઇલેક્ટ્રૉનિક પદ્ધતિ વધુ મહત્ત્વની સાબિત થઈ રહી છે. કારણકે સંસ્થાઓ ગ્રાહકોને ઝડપી અને ઓછા ખર્ચે સેવા આપવાના જુદા-જુદા માર્ગ શોધતી હોય છે. ગ્રાહક અને વિક્રેતા વચ્ચે થયેલા ઑનલાઇન નાણાકીય વિનિમયને ઇલેક્ટ્રૉનિક ચુકવણી કહે છે. આજકાલ બજારમાં ઇલેક્ટ્રૉનિક ચુકવણીના અનેક વિકલ્પ ઉપલબ્ધ છે. ઇલેક્ટ્રૉનિક ચુકવણી માટે નીચે જણાવેલ પ્રકાર ઉપયોગમાં લેવાય છે :

- **ચુકવણી માટેનાં કાર્ડ (Payment Cards)**

ચુકવણી માટેનાં કાર્ડમાં ક્રેડિટકાર્ડ, ડેબિટ કાર્ડ અને સ્માર્ટકાર્ડનો સમાવેશ કરી શકાય. સામાન્ય રીતે ગ્રાહક દ્વારા ખરીદી માટે ઉપયોગમાં લેવાતા તમામ પ્રકારનાં પ્લાસ્ટિકકાર્ડ માટે 'પેમેન્ટકાર્ડ' (payment card) પદનો ઉપયોગ કરવામાં આવે છે.

ક્રેડિટકાર્ડ (Credit Card)

ઇન્ટરનેટ પર ચુકવણી માટેની આ સૌથી વધુ પ્રચલિત અને વ્યાપક પ્રમાણમાં સ્વીકૃત પદ્ધતિ છે. ઇશ્યૂઇંગ બૅન્ક (issuing bank) નામે ઓળખાતી બૅન્ક દ્વારા ગ્રાહકને ક્રેડિટકાર્ડ આપવામાં આવે છે. ઇશ્યૂઇંગ બૅન્ક ક્રેડિટકાર્ડના વ્યવસાયમાં સ્થાપિત અને નામાંકિત નાણાકીય સંસ્થાઓનાં ક્રેડિટકાર્ડ પૂરાં પાડે છે. MasterCard® અથવા Visa® તેનાં ઉદાહરણ છે. ગ્રાહકને ધિરાણક્ષમતા અને આવકના સ્તર પર આધારિત ધિરાણ પૂરું પાડવામાં આવે છે. બિલિંગ સમયગાળા દરમિયાન ગ્રાહક તે ધિરાણની મર્યાદામાં રહી ઇશ્યૂઇંગ બૅન્ક સાથે ખર્ચ કે ચુકવણી કરી શકે છે.

ક્રેડિટકાર્ડ બૅન્કનાં ખાતાં સાથે સંલગ્ન હોવાથી ગ્રાહક જ્યારે તેનો ઉપયોગ ઑનલાઇન ચુકવણી માટે કરે ત્યારે વિક્રેતા પોતાના માલની કિંમત ઇશ્યૂઇંગ બૅન્ક પાસેથી વસૂલ કરે છે અને બૅન્ક ગ્રાહકના હવે પછીના સ્ટેટમેન્ટમાં તે કિંમત બાદ કરીને બતાવે છે. ગ્રાહક બૅન્કને ચૂકવણી કરે છે. વેબસાઇટ દ્વારા ક્રેડિટકાર્ડથી ચુકવણી સ્વીકારવા માટે વ્યાપારી એક્વાયરિંગ બૅન્ક (acquiring bank)ને નામે ઓળખાતી બૅન્કમાં વ્યાવસાયિક ખાતું ખોલાવે છે, જેથી તેને ઑનલાઇન અધિકૃતતા અને ચુકવણીની સેવા મળે છે. કાર્ડની સક્રિયતા, ખરીદી માટે અપાતા ધિરાણની મર્યાદા અને બિલિંગની માહિતી જેવી વિગતોની ચકાસણીને ગ્રાહકની અધિકૃતતા (Authorization) કહેવામાં આવે છે. વેપારીઓ દ્વારા ક્રેડિટકાર્ડનો વ્યાપક પ્રમાણમાં ઉપયોગ કરવામાં આવી રહ્યો છે. અને તે વિક્રેતા તથા ગ્રાહકને વિશ્વસનીયતા પૂરી પાડે છે. આકૃતિ 5.24માં કેટલાંક ક્રેડિટકાર્ડ દર્શાવ્યાં છે.

આકૃતિ 5.24 : ક્રેડિટકાર્ડનાં ઉદાહરણ

ઇન્ટરનેટ પર કરવામાં આવતા ક્રેડિટકાર્ડના વ્યવહારમાં ગ્રાહક વિક્રેતાની વેબસાઇટની મુલાકાત લે છે અને ખરીદી માટે માલ પસંદ કરે છે. ત્યાર બાદ ક્રેડિટકાર્ડને સંબંધિત તમામ માહિતી ઉમેરી આ માહિતી ઇલેક્ટ્રૉનિક પદ્ધતિથી વ્યાપારીને મોકલી આપવામાં આવે છે. આ વ્યવહારમાં ચાર પક્ષ સમાવિષ્ટ થાય છે :

- ક્રેડિટકાર્ડધારક ગ્રાહક
- ક્રેડિટકાર્ડ સ્વીકારતો વિક્રેતા
- ઇશ્યૂઇંગ બૅન્ક (Issuing Bank) : જે ગ્રાહકને ક્રેડિટકાર્ડ આપે છે અને વિક્રેતાને ચુકવણીની ખાતરી આપે છે. બૅન્ક ગ્રાહક પાસેથી ચુકવણીની રકમ પ્રાપ્ત કરે છે.
- એક્વાયરિંગ બૅન્ક (Acquiring banks) : વિક્રેતા સાથે ખાતું સ્થાપિત કરતી નાણાકીય સંસ્થા, જે ક્રેડિટકાર્ડની યથાર્થતા ચકાસે છે અને ગ્રાહકની ધિરાણમર્યાદાને આધારે વેચાણને અધિકૃત કરે છે.

ઑનલાઇન ચુકવણીમાં અન્ય બે પક્ષ ભાગ ભજવે છે : પેમેન્ટ ગેટ-વે (payment gateways) અને પેમેન્ટ પ્રોસેસર (payment processor). પે-પાલ (PayPal) જેવા ત્રાહિત પક્ષો દ્વારા પૂરી પાડવામાં આવતી સેવાને પેમેન્ટ ગેટ-વે તરીકે ઓળખવામાં આવે છે. તે અધિકૃતતા સ્થાપિત કરી ચુકવણી માટેના તમામ પક્ષોના નેટવર્કનું સુરક્ષિત સંકલન કરે છે. પ્રોસેસર્સ એવાં ડેટાસેન્ટર છે, જે ક્રેડિટકાર્ડના વ્યવહાર પૂરા કરે છે તથા વિક્રેતાને ભંડોળની પતાવટ કરે છે. પેમેન્ટ ગેટ-વે દ્વારા વ્યાપારીની ઇ-કૉમર્સ વેબસાઇટ સાથે જોડાયેલી હોય છે.

ઇન્ટરનેટ પર ક્રેડિટકાર્ડ દ્વારા થતી ઑનલાઇન ચુકવણી બે ભાગમાં વહેંચાયેલી છે : અધિકૃતતા (Authorization) અને વ્યવહારકરાર (Settlement).

અધિકૃતતાની પ્રક્રિયા દરમિયાન નીચેનાં પગલાં લેવામાં આવે છે :

- ચકાસણી (checkout) સમયે ગ્રાહક ક્રેડિટકાર્ડની માહિતી સાથે વ્યવહારની તમામ વિગતો (જેવી કે, વસ્તુની વિગત, ખરીદ તારીખ વગેરે) ઇ-કૉમર્સ વેબસાઇટને પૂરી પાડે છે, જે પેમેન્ટ ગેટ-વેને મોકલી આપવામાં આવે છે.
- પેમેન્ટ ગેટ-વે આ માહિતી પ્રોસેસરને મોકલે છે, જે માહિતીની ચકાસણી માટે ઇશ્યૂઇંગ બૅન્કનો સંપર્ક કરે છે.
- ઇશ્યૂઇંગ બૅન્ક ચકાસણી બાદ તેનું પરિણામ (વ્યવહારનો સ્વીકાર કે અસ્વીકાર) પેમેન્ટ ગેટ-વેને મોકલી આપે છે.
- જો વિક્રેતાને વ્યવહારને સ્વીકારે તો વ્યવહાર-કરાર (settlement) બને છે અને તે સમયે વ્યવહારની રકમ ગ્રાહકના ખાતામાંથી વિક્રેતાના ખાતામાં હસ્તાંતરિત કરવામાં આવે છે.

કરાર કે ચુકવણીની પ્રક્રિયા દરમિયાન નીચેનાં પગલાં લેવામાં આવે છે :

- વિક્રેતા તમામ વિગતો સાથેની વ્યવહારની વિનંતી પેમેન્ટ ગેટ-વેને મોકલે છે. જે ત્યાર પછી પ્રોસેસરને મોકલવામાં આવે છે.
- પ્રોસેસર ચુકવણીની વિગતો ગ્રાહકની ઇશ્યૂઇંગ બૅન્કને મોકલી આપે છે. તે ચુકવણીની વિગતો એક્વાયરિંગ બૅન્કને પણ મોકલે છે, જ્યાં વિક્રેતાનું ખાતું છે.
- એક્વાયરિંગ બૅન્ક વિક્રેતાના ખાતામાં રકમ જમા કરે છે. ઇશ્યૂઇંગ બૅન્ક તમામ ખર્ચ સાથેનું બિલ ગ્રાહકને મોકલે છે, જે તેણે સૂચિત સમયમર્યાદામાં ચૂકવવાનું રહે છે.

ઑનલાઇન વ્યવહારોને સુરક્ષિત બનાવવા માટે મોટા ભાગની ક્રેડિટકાર્ડ સંસ્થાઓ સિક્યોર ઇલેક્ટ્રૉનિક ટ્રાન્સફર (Secure Electronic Transfer - SET)નો ઉપયોગ કરે છે.

ક્રેડિટકાર્ડના ફાયદા નીચે મુજબ છે :

- ગ્રાહકને મોટી રોકડ રકમ લઈને ફરવું પડતું નથી. ગ્રાહક માલ અને સેવા માટે ઑનલાઇન અને ઑફલાઇન બંને રીતે ચુકવણી કરી શકે છે.
- બૅન્કના સ્ટેટમેન્ટની મદદથી ગ્રાહકની ખરીદીની નોંધ રાખી શકાય છે.
- બૅન્કના ખાતામાં રોકડ ઉપલબ્ધ ન હોય તે છતાં ગ્રાહકને માલ ખરીદવાની સુવિધા મળે છે. મહિનાના અંતે તેઓ રોકડ ચૂકવીને ખાતું સરભર કરે છે.

ક્રેડિટકાર્ડની મર્યાદાઓ નીચે મુજબ છે :

- બહુ નાની કે બહુ મોટી ચુકવણીઓ માટે તે અયોગ્ય છે. સુરક્ષાની દૃષ્ટિએ આ કાર્ડની એક મર્યાદા છે. ખૂબ જ મોટી રકમના વ્યવહારો માટે તેનો ઉપયોગ શક્ય નથી.
- ગ્રાહક ક્રેડિટકાર્ડનો ઉપયોગ કરી પોતાની ક્ષમતાથી વધુ ખર્ચ કરવાની માનસિકતા ધરાવી શકે છે.
- ક્રેડિટકાર્ડ ખોવાઈ કે ચોરાઈ જવાની શક્યતા રહે છે.

ડેબિટકાર્ડ (Debit Cards)

ડેબિટકાર્ડ દેખાવમાં ક્રેડિટકાર્ડ જેવું લાગે છે, પરંતુ તેનું કાર્ય અલગ છે. ગ્રાહકના બૅન્કખાતામાંથી સીધા જ વ્યાપારીને રકમનું હસ્તાંતરણ કરી શકાય તે પ્રકારનું આ કાર્ડ છે. ગ્રાહકના ખાતામાંથી રકમની તત્કાલ કપાત કરવામાં આવે છે. ડેબિટકાર્ડ ગ્રાહકને ખરીદીની મર્યાદામાં રાખે છે અને તેની ક્ષમતાથી વધુ ખરીદી કરવાની પરવાનગી આપતું નથી. ખરીદી માટે ડેબિટકાર્ડનો ઉપયોગ કરતી વખતે ગ્રાહકે હંમેશાં પોતાનાં ખાતાંની સિલકનું ધ્યાન રાખવું જરૂરી છે. આકૃતિ 5.25માં ડેબિટકાર્ડનું ઉદાહરણ આપવામાં આવ્યું છે.

આકૃતિ 5.25 : ડેબિટકાર્ડનું ઉદાહરણ

સ્માર્ટકાર્ડ (Smart Cards)

સ્માર્ટકાર્ડ ક્રેડિટકાર્ડ જેવું જ દેખાય છે, પરંતુ સપાટી પર માઇક્રોચિપ જડેલી હોવાથી તેનું કાર્ય અલગ પડે છે. ખાતાંઓની માહિતી, આરોગ્ય સંબંધિત વિમાની માહિતી, અંગત કી અને ઉપયોગકર્તાની અન્ય અંગત માહિતીનો સ્માર્ટકાર્ડમાં સંગ્રહ કરવામાં આવે છે. સામાન્ય કાર્ડ કરતા તે સો ગણી વધુ માહિતીનો સંગ્રહ કરી શકે છે. સ્માર્ટકાર્ડમાં સંગૃહીત માહિતી સાંકેતિક સ્વરૂપમાં હોવાથી તે ક્રેડિટ કે ડેબિટકાર્ડથી વધુ સુરક્ષિત છે. ગ્રાહકો કાર્ડમાં રોકડ પણ ઉમેરી શકે છે અને ત્યાર પછી છૂટક વ્યાપારીની વેબસાઇટ કે દુકાન પર માલની ખરીદી તેની ચુકવણી માટે તેનો ઉપયોગ પણ કરી શકે છે. છૂટક વ્યાપારીની દુકાન પર કાર્ડરીડર ઉપલબ્ધ હોય છે તથા તેને PC સાથે પણ જોડી શકાય છે. આ અનુકૂળતા સ્માર્ટકાર્ડને ઘણું ફાયદાકારક બનાવે છે. ડિજિટલ રોકડનો સંગ્રહ કરવા, દર્દીનો તબીબી અહેવાલ મેળવવા જેવા ઘણા હેતુઓ માટે તેનો ઉપયોગ કરી શકાય છે. યુ.એસ., યુરોપ, જાપાન અને એશિયાના કેટલાક ભાગમાં સ્માર્ટકાર્ડ ઘણાં પ્રચલિત છે.

સ્માર્ટકાર્ડમાં રહેલી માહિતી જોવા માટે તથા નવી માહિતી ઉમેરવા માટે કાર્ડરીડર જરૂરી છે. આ એક એવું નાનું ઉપકરણ છે જેમાં કાર્ડ દાખલ કરવામાં આવે છે. ઉદાહરણ તરીકે, તમે તમારા ફેમિલી ડૉક્ટર પાસે જાઓ છો અને તમારી બીમારીનું પુનઃનિરીક્ષણ કરી દવાઓ સૂચવવા તેમને સ્માર્ટકાર્ડ આપો છો. તમારા વિગતો તમારા સ્માર્ટકાર્ડમાં દાખલ કરી દેવાય છે. એ પછી દવાની દુકાનમાં જઈ ફાર્માસિસ્ટને તમે તમારું સ્માર્ટકાર્ડ આપો છો. તેમાંથી તે ડૉક્ટરની સૂચનાઓ વાંચી તમને નિયત દવાઓ આપી શકે છે. સ્માર્ટકાર્ડનો ઉપયોગ કરી ચુકવણી પણ કરી શકાય છે.

ચાર્જકાર્ડ (Charge Cards)

ચાર્જકાર્ડ પણ એક પ્રકારની ચુકવણીની પ્રક્રિયા છે, જેમાં ગ્રાહક કાર્ડ દ્વારા વિક્રેતાને ચુકવણી કરી શકે છે. ક્રેડિટકાર્ડની તુલનામાં ચાર્જકાર્ડમાં ધિરાણની કોઈ મર્યાદા હોતી નથી. કાર્ડ આપનાર સંસ્થાને ગ્રાહકે બિલિંગ સમયગાળાને અંતે પૂરેપૂરી રકમની ચુકવણી કરવાની હોય છે. જો આમ ન બને તો તેણે લેઇટ ફી ભરવી પડે છે.

- ### નેટબૅંકિંગ (Net-Banking)

નેટબૅંકિંગ અથવા ઑનલાઇન બૅંકિંગનો વિકલ્પ આજકાલ ઘણો પ્રચલિત થઈ રહ્યો છે. તેમાં કોઈ પ્રકારના કાર્ડનો સમાવેશ કરવામાં આવતો નથી. ઇન્ટરનેટ બૅંકિંગ સહિતનું બૅન્કખાતું ધરાવતો ગ્રાહક તેનો ઉપયોગ કરી શકે છે. ગ્રાહક આ ખાતાના ઉપયોગ દ્વારા ઑનલાઇન ખરીદી તેમજ ચૂકવણી કરી શકે તે માટે બૅન્ક દ્વારા તેને નેટબૅંકિંગ પાસવર્ડ પૂરો પાડવામાં આવે છે. ઇ-કૉમર્સ/એમ-કૉમર્સની અનેક વેબસાઇટ નેટબૅંકિંગ દ્વારા ચુકવણીની સુવિધા પૂરી પાડે છે. વેબસાઇટ પર કાર્ડની વિગતો રજૂ કરવાને બદલે જે બૅન્ક દ્વારા ચુકવણી કરવાની હોય તેનું નામ પૂછવામાં આવે છે. આ વેબસાઇટ પર તમે ચુકવણીની પ્રક્રિયામાં આગળ વધો છો ત્યારે બૅન્કનાં નામની પસંદગી પૂછવામાં આવે છે. બૅન્કની પસંદગી કર્યા બાદ સ્ક્રીન પર તે બૅન્કની વેબસાઇટ જોવા મળે છે. જ્યાં ખાતા-નંબર અને નેટબૅંકિંગ પાસવર્ડ દ્વારા લૉગ ઇન કરવાની જરૂર પડે છે. હવે તમે વ્યવહારની પ્રક્રિયા દ્વારા તમારા ખાતામાંથી વિક્રેતાનાં ખાતાંમાં રકમનું હસ્તાંતરણ કરી શકો છો.

ભારતીય રેલવે તેની વેબસાઇટ www.irctc.co.in પર ઑનલાઇન ટિકિટની નોંધણીની સુવિધા પૂરી પાડે છે. આ માટે ઉપયોગકર્તા સૌથી પહેલાં સાઇટ પર પોતાની નોંધણી (registration) કરાવી મુસાફરોની માહિતી સહિત મુસાફરીની વિગતો આપશે. એ પછી ચુકવણી માટે નેટબૅંકિંગના વિકલ્પ દ્વારા બૅન્કની પસંદગી કરી શકાય છે. અહીં તમારી બૅન્કની લૉગ-ઇન સ્ક્રીન દર્શાવવામાં આવશે. અને ભારતીય રેલવેને ચુકવણી કરી ટિકિટ નોંધણીની પ્રક્રિયા પૂર્ણ કરી શકાશે. સફળતાપૂર્વક ચૂકવણી કર્યા બાદ ઉપયોગકર્તા નોંધાયેલ મોબાઇલ નંબર પર irctc પરથી સંદેશ મેળવશે, જે મુસાફરી દરમિયાન ટિકિટની અવેજીમાં દર્શાવી શકાશે અથવા ટિકિટની ઇ-કૉપી (E-copy) પણ મુદ્રિત કરી શકાશે. નેટબૅંકિંગની ચુકવણી પદ્ધતિ ક્રેડિટકાર્ડની તુલનામાં વધુ સુરક્ષિત માનવામાં આવે છે અને ભારતમાં લગભગ દરેક વ્યાપારી સંસ્થાઓ આ વિકલ્પ પૂરો પાડે છે.

- ### ઇલેક્ટ્રૉનિક ફંડ-ટ્રાન્સફર (Electronic Fund Transfer)

એક બૅન્કના ખાતામાંથી અન્ય બૅન્કના ખાતામાં રકમ સ્થાનાંતરિત કરવાના કાર્યને ઇલેક્ટ્રૉનિક ફંડ-ટ્રાન્સફર તરીકે ઓળખવામાં આવે છે. કાગળના રૂપમાં ચેક કે પ્રત્યક્ષ ઉઘરાણી કરતાં તે સુરક્ષિત, નિશ્ચિત, અસરકારક અને ઓછું ખર્ચાળ છે. દુનિયાની બૅન્કોના પારસ્પરિક વ્યવહાર, ATMના ઉપયોગથી ટ્યૂશન ફીની ચુકવણી, કર્મચારીઓના પગાર તેમનાં ખાતાંમાં જમા કરવા, બૅન્કખાતાની માસિક કપાત વગેરે EFTનાં ઉદાહરણ છે. ઑનલાઇન ચુકવણી માટે EFTની અસરકારકતામાં વધારો, સરળ નામાપદ્ધતિ અને વધુ સુરક્ષા EFTના લાભ છે. જોકે, ઇન્ટરનેટ દ્વારા બિલ મોકલતી અને મેળવતી સંસ્થાઓની સંખ્યા હજી ઓછી છે.

- **ઈ-વૉલેટ (E-wallet)**

જ્યારે તમે વેબ પરથી ખરીદી કરો છો, ત્યારે તમારું નામ, માલ મોકલવાનું સરનામું, બિલ મોકલવા માટેનું સરનામું, ક્રેડિટકાર્ડની માહિતી સહિતનું એક ફૉર્મ ભરવાનું રહે છે. એકાદવાર આ કાર્ય યોગ્ય છે, પરંતુ દરેક સમયે ખરીદી કરતી વખતે આ કાર્ય કષ્ટસાધ્ય બની રહે છે. આ માટે, કેટલાક વ્યાપારીઓ ગ્રાહકે એક વાર ભરેલા ફૉર્મની વિગતોના પુનઃઉપયોગ માટે તેને પોતાના સર્વર પર સંગ્રહ કરે છે. આ વ્યાપારીઓ ગ્રાહકને ઇ-વૉલેટ સેવાઓ પૂરી પાડે છે, જેના દ્વારા ખરીદી કે ટિકિટ-નોંધણીનું કાર્ય વધુ સરળ બને છે. આજે ઘણી બૅન્કો, કરિયાણાની ઑનલાઇન દુકાનો, ટેલિફોનસેવાઓ વગેરે ઇ-વૉલેટ સેવાઓ પૂરી પાડે છે. વ્યાપારીઓને સુરક્ષિત રીતે ઑનલાઇન ચુકવણી માટેનાં ઇલેક્ટ્રૉનિક કાર્ડને ઇ-વૉલેટ કહે છે. તે ક્રેડિટકાર્ડ કે ડેબિટકાર્ડની જેમજ કાર્ય કરે છે. ઇ-વૉલેટ દ્વારા ચુકવણી કરતી વખતે ગ્રાહકે ક્રેડિટ કે ડેબિટકાર્ડ નંબર પૂરો પાડવો જરૂરી નથી. આથી, ક્રેડિટ/ડેબિટ કાર્ડની ગોપનીયતા નષ્ટ થવાનું જોખમ રહેતું નથી.

ઉદાહરણ તરીકે, ઑનલાઇન નોંધણી સરળ બનાવવા માટે IRCTCએ ગ્રાહકો માટે ઇ-વૉલેટ યોજના શરૂ કરી છે. ખાતું ધરાવનાર ગ્રાહક IRCTCમાં અગાઉથી પૈસા જમા કરાવી શકે છે, જેનો ઉપયોગ ભવિષ્યમાં ટિકિટની ઑનલાઇન નોંધણી માટે ચુકવણીના વિકલ્પ તરીકે કરી શકાય છે. હાલમાં ઉપયોગકર્તાને ટિકિટની નોંધણી માટે ક્રેડિટ/ડેબિટ-કાર્ડની જરૂર પડે છે. આ પ્રક્રિયામાં સમય લાગે છે તથા ગ્રાહકને ચુકવણી માટે બૅન્કના સર્વર પર સ્થાનાંતરિત કરવામાં આવે છે.

- **રુ-પે (RuPay)**

Rupee અને Payment એવાં બે પદ પરથી RuPay પદ તારવવામાં આવ્યું છે. નેશનલ પેમેન્ટ કૉર્પોરેશન ઑફ ઇન્ડિયા (NPCI)એ શરૂ કરેલી કાર્ડ દ્વારા ચુકવણીની આ એક નવી રીત છે. આકૃતિ 5.26માં RuPay કાર્ડ દર્શાવ્યું છે, જેનો ઉપયોગ ક્રેડિટકાર્ડ અને ડેબિટકાર્ડની જેમ જ કરી શકાય છે.

આકૃતિ 5.26 : RuPay કાર્ડ

હાલમાં, સમગ્ર વિશ્વમાં ફેલાયેલા કાર્ડધારકો, વ્યાપારીઓ અને ઇશ્યૂઇંગ બૅન્કો માસ્ટરકાર્ડ અને વીસા સાથે જોડાણ કરી આપે તેવી કોઈ સ્થાનિક ભારતીય બૅન્ક અસ્તિત્વમાં નથી. માસ્ટરકાર્ડ અને વીસાકાર્ડ ચુકવણીની દૃષ્ટિએ વૈશ્વિક અગ્રણીઓ છે. આ પ્રકારના તમામ વ્યવહાર તેમના દ્વારા કરવામાં આવે છે. સ્થાનિક બૅન્કો દ્વારા થતા ક્રેડિટકાર્ડ કે ડેબિટકાર્ડના વ્યવહારોને દેશબહાર આવેલા નેટવર્ક દ્વારા આગળ વધારવામાં આવે છે. આ સેવાઓ પૂરી પાડવા માટે તેઓ અતિરિક્ત કિંમત વસૂલ કરે છે, જેથી બૅન્કે તમામ ડેબિટ અને ક્રેડિટકાર્ડ ચુકવણીઓ માટેનો ખર્ચ ભોગવવો પડે છે. RuPay ચુકવણી માટે અન્ય કાર્ડનો સ્થાનિક વિકલ્પ છે. RuPay કાર્ડના ઉપયોગ દ્વારા થતા તમામ વ્યવહારો પર ભારતમાં જ પ્રક્રિયા કરવામાં આવે છે. વ્યવહારની પ્રક્રિયા સ્થાનિક સ્તરે થતી હોવાથી દરેક વ્યવહારની લેવડ-દેવડ અને કરારનો ખર્ચ ઘટાડી શકાય છે. આમ, ઓછું ખર્ચાળ હોવાને કારણે RuPay ગ્રાહક અને બૅન્ક માટે લાભદાયી છે.

સારાંશ

આ પ્રકરણમાં આપણે મોબાઇલ કૉમર્સ વિશે ચર્ચા કરી. ઇન્ટરનેટના ઉપયોગ દ્વારા મોબાઇલ ફોન, પર્સનલ ડિજિટલ આસિસ્ટન્ટ (PDA), સ્માર્ટફોન, ટેબ્લેટ PC અને પામટૉપ જેવાં વાયરલેસ સાધનોની મદદથી માલ અને સેવાના ખરીદ-વેચાણની પ્રક્રિયાને એમ-કૉમર્સ કહે છે. એમ-કૉમર્સના ફાયદાઓ અને મર્યાદાઓ વિશે પણ આપણે ચર્ચા કરી. મોબાઇલ ઉપયોગકર્તાઓની સંખ્યામાં દિન-પ્રતિદિન થતી વૃદ્ધિને કારણે એમ-કૉમર્સ વિનિયોગ ઉત્તરોત્તર પ્રચલિત બની રહ્યા છે. ઇ-કૉમર્સના ભવિષ્ય માટે સ્થાન આધારિત વ્યાવસાયિક વિનિયોગો અને સેવાઓ તેના સૂચિતાર્થો સહિત અગ્રસર થઈ રહી છે. આ વિનિયોગ દ્વારા ઉત્પાદન કે સેવા પૂરી પાડવા માટે ઉપયોગકર્તાનું નિશ્ચિત સ્થાન શોધી કાઢવામાં આવે છે. આપણે વિવિધ પ્રકારના સુરક્ષાના મુદ્દા અને ભયસ્થાનો વિશે ચર્ચા કરી. ઇ-કૉમર્સની સુરક્ષામાં અવરોધરૂપ ધમકીઓના નિવારણ માટે લઈ શકાતાં પગલાં અંગેની ચર્ચા પણ કરવામાં આવી. ઇ-કૉમર્સને લગતા કાયદાકીય મુદ્દાઓ પણ આપણે જોયા. ઇ-કૉમર્સ દ્વારા પૂરાં પાડવામાં આવતાં ક્રેડિટકાર્ડ, ડેબિટકાર્ડ, સ્માર્ટકાર્ડ, રુ-પે, નેટબૅંકિંગ અને ઇ-વૉલેટ જેવા ચુકવણીના વિવિધ વિકલ્પો વિશે પણ ચર્ચા કરવામાં આવી.

સ્વાધ્યાય

1. એમ-કૉમર્સ એટલે શું ? એમ-કૉમર્સનાં ઉદાહરણોની યાદી બનાવો.
2. એમ-કૉમર્સ ઉપયોગકર્તાને શા માટે ઉપયોગી છે ?
3. એમ-કૉમર્સના ફાયદાની યાદી બનાવો.
4. એમ-કૉમર્સની મર્યાદાઓ કઈ છે ?
5. એમ-કૉમર્સના વિનિયોગની યાદી બનાવો.
6. નીચેના એમ-કૉમર્સ વિનિયોગો માટે વેબસાઇટનાં ઉદાહરણ આપો :
   (1) મોબાઇલ દ્વારા ટિકિટની નોંધણી
   (2) મોબાઇલ દ્વારા હરાજી
   (3) મોબાઇલ દ્વારા ખરીદી
   (4) મોબાઇલ દ્વારા માહિતી સેવા
   (5) મોબાઇલ દ્વારા વાણિજ્યિક સેવા
7. એલ-કૉમર્સ એટલે શું ?
8. GPS એટલે શું ? તે સાધનને કેવી રીતે શોધે છે ?
9. સ્થાનઆધારિત સેવાના વિનિયોગોની યાદી બનાવો.
10. ઇ-કૉમર્સની સુરક્ષાનાં ચાર મહત્ત્વના પાસાં ક્યાં છે ?
11. ઇન્ટરનેટની સુરક્ષાનાં નીચે જણાવેલ જોખમો વિશે ખ્યાલ આપો :
    (1) દૂષિત કોડ (2) સ્નિફર (3) સેવામાં અવરોધરૂપ આક્રમણ
12. સાયબર જંગલીપણું (vandalism) એટલે શું ?
13. 'સ્પૂફિંગ' (spoofing) પદની સમજૂતી આપો.
14. સુરક્ષા સામેની ધમકીઓ માટે લઈ શકાતાં પગલાંની યાદી બનાવો.
15. ડિજિટલ સર્ટિફિકેટમાં શેનો સમાવેશ કરવામાં આવે છે ?
16. સાંકેતીકરણ એટલે શું ? સાંકેતીકરણ પદ્ધતિનો ઉપયોગ કરી "Gandhi Ashram" લખાણના દરેક અક્ષરને તેની પહેલાંના અક્ષર સાથે બદલી ગુપ્ત ભાષામાં ફેરવો.

17. SSLનો ઉદ્દેશ શું છે ?

18. બૌદ્ધિક સંપત્તિને સંબંધિત મુદ્દાઓની યાદી બનાવો.

19. બૌદ્ધિક સંપત્તિની સુરક્ષા માટે જુદા-જુદા ક્યા માર્ગો ઉપલબ્ધ છે ?

20. ઇલેક્ટ્રોનિક ચુકવણીની જુદી-જુદી પદ્ધતિઓની યાદી બનાવો.

21. ક્રેડિટકાર્ડના ફાયદા અને મર્યાદાઓની યાદી બનાવો.

22. ક્રેડિટકાર્ડથી સ્માર્ટકાર્ડ કેવી રીતે અલગ પડે છે ?

23. ઇલેક્ટ્રોનિક ફંડનું હસ્તાંતરણ (transfer) એટલે શું ?

24. આપેલા વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરો :

(1) ઇન્ટરનેટ ધરાવતા વાયરલેસ સાધનનો ઉપયોગ કરી માલ કે સેવાના ખરીદ-વેચાણને શું કહે છે ?

(a) ઇન્ટરનેટ (b) એમ-કૉમર્સ (c) એમ-બૅન્કિંગ (d) WWW

(2) વ્યાવસાયિક હેતુ માટે સ્થાનની માહિતી પૂરી પાડતી તક્નિકના ઉપયોગને શું કહે છે ?

(a) ઇ-કૉમર્સ (b) એમ-કૉમર્સ

(c) એલ-કૉમર્સ (d) પરંપરાગત કૉમર્સ

(3) GPS એટલે શું ?

(a) Global Positioning System (b) Global Postal System

(c) Grand Positioning System (d) Google Positioning System

(4) અનધિકૃત ઉપયોગકર્તા વાંચી ન શકે તે માટે માહિતીને ગુપ્ત રાખવાની પદ્ધતિને સુરક્ષાના ક્યા પ્રકાર તરીકે ઓળખવામાં આવે છે ?

(a) ગુપ્તતા (Confidentiality) (b) અખંડિતતા (Integrity)

(c) અસ્વીકાર (Non-repudiation) (d) અધિકૃતતા (Authorization)

(5) વિતરણ દરમિયાન વિગતને અકસ્માતથી કે દૂષિત ઇરાદાઓથી બદલવામાં આવ્યા નથી કે તેની સાથે કોઈ ચેડાં કરવામાં આવ્યાં નથી – આને સુરક્ષાનો ક્યો પ્રકાર કહેવાય ?

(a) ગુપ્તતા (b) અખંડિતતા

(c) અસ્વીકાર (d) અધિકૃતતા

(6) માત્ર અધિકૃત ઉપયોગકર્તાને જ સિસ્ટમના ઉપયોગની પરવાનગી આપવામાં આવે – તે સુરક્ષાનો ક્યો પ્રકાર દર્શાવે છે ?

(a) અધિકૃતતા (b) ગુપ્તતા (c) અસ્વીકાર (d) અખંડિતતા

(7) નીચેનામાંથી સુરક્ષાનો ક્યો પ્રકાર ખાતરી આપે છે કે સંદેશ મોકલનાર સંદેશ મોકલ્યાનો ઇનકાર કરી શકશે નહીં ?

(a) અધિકૃતતા (b) ગુપ્તતા (c) અસ્વીકાર (d) અખંડિતતા

(8) ઇન્ટરનેટ પર કમ્પ્યૂટર કે રાઉટર દ્વારા પ્રેષકથી પ્રાપ્તકર્તા સુધી મોકલાતી માહિતીની નોંધ રાખતા પ્રોગ્રામને શું કહે છે ?

(a) Sniffer (b) Denial of service attack

(c) Malicious code (d) Spoofing

(9) ઉપયોગકર્તાના મશીન કે નેટવર્કને સ્થગિત કરી નાખી તેને નિરુપયોગી બનાવી દેવાનું આક્રમણ નીચેનાંમાંથી ક્યા નામે ઓળખાય છે ?

(a) Malicious code (b) Denial-of-Service

(c) Spoofing (d) Cyber vandalism

(10) હયાત વેબસાઇટના પાનાને ઇલેક્ટ્રોનિક પદ્ધતિથી સુધારવાની પદ્ધતિને શું કહે છે ?

(a) Cyber vandalism (b) Denial-of-Service

(c) Spoofing (d) Malicious code

(11) કોઈની સમક્ષ તમે જે નથી તે સ્વરૂપે રજૂ થવું અથવા કોઈ નકલી વેબસાઇટને પ્રમાણભૂત વેબસાઇટ તરીકે રજૂ કરવી તેને શું કહેવાય છે ?

(a) Cyber vandalism (b) Malicious code

(c) Denial-of-Service (d) Spoofing

(12) વાયરસ, વૉર્મ અને ટ્રોજનહૉર્સ જેવા દૂષિત કોડને શોધી, અટકાવી અને દૂર કરવાની પ્રક્રિયા કરતા કમ્પ્યૂટર-પ્રોગ્રામને શું કહે છે ?

(a) ઍન્ટિવાઇરસ સૉફ્ટવેર (b) ડિજિટલ સર્ટિફિકેટ

(c) ફાયરવૉલ (d) ક્રિપ્ટોગ્રાફી

(13) એન્ક્રિપ્શન અલ્ગોરિધમનો ઉપયોગ કરી 'સાદા લખાણ' નામે ઓળખાતા મૂળ લખાણને 'ગુપ્ત લખાણ' તરીકે ઓળખાતા અવાચ્ય લખાણમાં પરિવર્તિત કરવાની ક્રિયાને શું કહે છે ?

(a) ફાયરવૉલ (b) એન્ક્રિપ્શન

(c) ઍન્ટિવાઇરસ સૉફ્ટવેર (d) ડિજિટલ સર્ટિફિકેટ

(14) ગુપ્ત લખાણને ફરી મૂળ લખાણમાં પરિવર્તિત કરવાની ક્રિયાને શું કહે છે ?

(a) ફાયરવૉલ (b) ડિજિટલ સર્ટિફિકેટ

(c) ડિક્રિપ્શન (d) વાઇરસ

(15) ઇન્ટરનેટ પર વેબવ્યવહારોને સુરક્ષિત રાખવા માટે ક્યા પ્રોટોકોલનો ઉપયોગ કરવામાં આવે છે ?

(a) TCP/IP (b) HTTP (c) Bluetooth (d) SSL

(16) SSL પ્રોટોકોલની રચના કોણે કરી ?

(a) Google (b) Netscape (c) Yahoo (d) Firefox

(17) નીચેનામાંથી વેબસાઇટની કઈ શરૂઆત દર્શાવે છે કે સાઇટને SSL દ્વારા સુરક્ષિત કરવામાં આવેલી છે ?

(a) http:// (b) ssl:// (c) https:// (d) http-ssl://

(18) સર્જકના મૂળભૂત કાર્યનો ઉપયોગ અન્ય અનધિકૃત ઉપયોગકર્તા દ્વારા થતો અટકાવવા સર્જકને કઈ સુવિધા પૂરી પાડવામાં આવે છે ?

(a) ટ્રેડમાર્ક (b) કૉપીરાઇટ (c) ડિજિટલ વૉટરમાર્કિંગ (d) સ્ટેગેનોગ્રાફી

(19) ઉત્પાદન કે સેવાને બજારના અન્ય ઉત્પાદન કે સેવાથી અલગ રાખવા વ્યક્તિ કે સંસ્થા દ્વારા ઉપયોગમાં લેવામાં આવતા નિશ્ચિત લોગો, શબ્દ, નિશાની, શૈલી કે શબ્દસમૂહ કે ચિત્રને શું કહે છે ?

(a) ટ્રેડમાર્ક (b) કૉપીરાઇટ (c) ડિજિટલ વૉટરમાર્કિંગ (d) સ્ટેગેનોગ્રાફી

(20) નીચેનામાંથી કઈ ટ્રેડમાર્કની નિશાની છે ?

(a) TM, MT and © (b) TM, MS and ®

(c) TM, SM and ® (d) TM, SM and ©

(21) એક માહિતીમાં અન્ય માહિતી સંતાડવાની ક્રિયાને શું કહે છે ?

(a) Squatting (b) Steganography (c) Name changing (d) Copyright

(22) ફાઇલની કૉપીરાઇટ માહિતીને ઓળખવા માટે ચિત્ર, ઑડિયો કે વીડિયોમાં ઉમેરવામાં આવતા ડિજિટલ કોડને શું કહે છે ?

(a) Image mark (b) Digital mark

(c) Code mark (d) Watermark

(23) ઉપયોગકર્તાને ક્રેડિટકાર્ડ તથા વેપારીને ચુકવણીની ખાતરી કોણ આપે છે ?

(a) વેપારી (b) ઇશ્યૂઇંગ બૅન્ક (c) એક્વાયરિંગ બૅન્ક (d) ગ્રાહક

(24) નીચેનામાંથી ચુકવણીના ક્યા કાર્ડની સપાટી પર માઇક્રોચિપ જડેલી હોય છે ?

(a) સ્માર્ટકાર્ડ (b) ડેબિટકાર્ડ (c) ક્રેડિટકાર્ડ (d) ચાર્જકાર્ડ

●

ઑબ્જેક્ટ આધારિત ખ્યાલો 6

આપણે આજે ઇન્ટરનેટ, વેબસાઇટ અને વેબ આધારિત પ્રક્રિયાઓના યુગમાં છીએ કે જ્યાં વિનિયોગનો ઝડપી વિકાસ તેમજ સૉર્સ કોડ (source code)નો પુનઃ ઉપયોગ ઘણો અગત્યનો છે. સૉફ્ટવેર સિસ્ટમનાં વિશ્લેષણ, ડિઝાઇન અને અમલીકરણમાં ઑબ્જેક્ટ (object) આધારિત પદ્ધતિ કે ક્રિયા નોંધપાત્ર ભૂમિકા ભજવે છે. ઑબ્જેક્ટ આધારિત પદ્ધતિનો ઉપયોગ કરીને બનાવવામાં આવેલા સૉફ્ટવેર વધુ વિશ્વસનીય છે અને તેની જાળવણી, પુનઃઉપયોગ અને તેના વિકાસનું કાર્ય ખૂબ સરળ છે. આ પ્રકરણમાં ઑબ્જેક્ટ આધારિત ખ્યાલોની સામાન્ય સમજ આપેલી છે. જાવા પ્રોગ્રામિંગ ભાષાનો ઉપયોગ કરીને આ ખ્યાલોનું અમલીકરણ હવે પછીનાં પ્રકરણોમાં આવરી લેવામાં આવ્યું છે.

પરિચય (Introduction)

ઑબ્જેક્ટ (object) આધારિત પ્રોગ્રામિંગની શરૂઆત 1960ના સમયગાળામાં થઈ અને 1980ના દાયકાની મધ્યથી નવા સૉફ્ટવેર બનાવવામાં પ્રોગ્રામિંગની તે મુખ્ય પદ્ધતિ બની ગઈ. સૉફ્ટવેર સિસ્ટમના અતિ ઝડપથી વધતા કદ અને જટિલતાનું નિયંત્રણ કરવા માટે તેમજ મોટી અને જટિલ સિસ્ટમને સમય સાથે સુધારવાના કાર્યને સરળ બનાવવાના એક માર્ગ તરીકે આ પદ્ધતિને વિકસાવવામાં આવી હતી. કેટલીક પ્રચલિત પ્રોગ્રામિંગ ભાષાઓ જેવી કે C++, Java, C#, VB.net, ASP.net અને PHP ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગને સમર્થન આપે છે.

પ્રોગ્રામિંગની રીતને આપણે બે પ્રકારમાં વહેંચી શકીએ, જેમકે પ્રક્રિયાગત પ્રોગ્રામિંગ (Structure/Procedural Programming) અને ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગ. પ્રક્રિયાગત પ્રોગ્રામિંગમાં આપણું કેન્દ્રબિંદુ ડેટા ઉપરની કાર્યપ્રણાલી (functions) તેમજ પ્રક્રિયાઓ (procedures) લખવામાં કેન્દ્રિત રહે છે. ઉદાહરણ તરીકે, પુસ્તકાલય વિનિયોગ સૉફ્ટવેર માટે આપણે પુસ્તકાલયના વિનિયોગની બધી પ્રક્રિયા બાબત વિચારીશું અને આપણું ધ્યાન વિદ્યાર્થી-નોંધણી, પુસ્તક-વિતરણ, પુસ્તક પરત કરવું અને દંડની ગણતરી જેવા વિભાગો ઉપર કેન્દ્રિત રહેશે.

ઑબ્જેક્ટ આધારિત પદ્ધતિમાં આપણા કેન્દ્રબિંદુએ **ઑબ્જેક્ટ (object)** હોય છે કે જે ડેટા તેમજ ક્રિયાત્મકતા (functionality) એમ બંને એકસાથે ધરાવે છે. પુસ્તકાલય વિનિયોગ બનાવવામાં આપણું ધ્યાન વિનિયોગમાં સમાવેશ વિવિધ ઑબ્જેક્ટ ઉપર કેન્દ્રિત હોય છે. અહીં આપણે વિદ્યાર્થી, પુસ્તક અને ગ્રંથપાલ જેવા ઑબ્જેક્ટ વિચારી શકીએ. આવા ઑબ્જેક્ટ એકબીજા સાથે કઈ રીતે સંકળાયેલા છે (association) તે બાબત પણ વિચારવું જોઈએ. ઉદાહરણ તરીકે, વિદ્યાર્થી પુસ્તકાલયમાં પુસ્તક પરત કરે છે.

ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગ ભાષાઓની તાકાત (ક્ષમતા) પ્રોગ્રામરને વિભાગીય (modular), પુનઃઉપયોગમાં લઈ શકાય તેમજ તે પ્રોગ્રામનો વિકાસ કરી શકાય (extendable) તે પ્રકારનો પ્રોગ્રામનો કોડ લખવાનું સામર્થ્ય પૂરું પાડે છે. આ ક્ષમતાને કારણે પ્રોગ્રામર હયાત મૉડ્યુલમાં ફેરફાર કરીને નવા પ્રોગ્રામની રચના કરી શકે છે. સૉફ્ટવેરના અન્ય ભાગના કોડમાં ખલેલ પહોંચાડ્યા વગર મૉડ્યુલમાં ફેરફાર કરવા કે બદલવાની સમર્થતા વડે તે ભાષાઓને લવચિકતા બક્ષે છે. હયાત કોડનો ફરી ઉપયોગ તેમજ હયાત કોડને સુધારીને ઉપયોગ કરીને સૉફ્ટવેર બનાવવાની ઝડપ વધારી શકાય છે.

ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગ ઑબ્જેક્ટને (object) પાયાના એકમ તરીકે વાપરે છે. એકસરખા ઑબ્જેક્ટનું ક્લાસ (class)ના ખ્યાલ દ્વારા વર્ગીકરણ કરવામાં આવે છે. કમ્પ્યૂટરની જે ભાષાઓ ઑબ્જેક્ટના ચાર ચોક્કસ ગુણધર્મો (1) **ઍબ્સ્ટ્રૅક્શન** (abstraction) (2) **ઇનકૅપ્સ્યુલેશન** (encapsulation) (3) **પોલિમૉર્ફિઝમ** (polymorphism) અને (4) **ઇનહેરીટન્સ** (inheritance) પૂરા પાડે છે, તે ભાષા ઑબ્જેક્ટ આધારિત ભાષા તરીકે ઓળખાય છે.

ઑબ્જેક્ટ (Object)

વાસ્તવિક વિશ્વમાં ઑબ્જેક્ટ એ આ દુનિયા જે વસ્તુઓ વડે બનેલી છે, તે વસ્તુ છે. આમાંની કેટલીક વસ્તુઓ વ્યક્તિ, કાર કે કૉફીના પ્યાલા જેવા કોઈ ભૌતિક સ્વરૂપમાં હયાતી ધરાવે છે. અન્ય વસ્તુઓ અમૂર્ત સ્વરૂપે (abstract) હોઈ શકે કે જેને સ્પર્શ ન કરી શકાય અથવા જોઈ ન શકાય, ઉદાહરણ તરીકે તારીખ અને સમય જેવા ખ્યાલો.

દરેક ઑબ્જેક્ટ (object)ની એક અનન્ય ઓળખ હોય છે અને દરેક ઑબ્જેક્ટને એકબીજાથી અલગ ઓળખી શકાય છે. ઉદાહરણ તરીકે, દરેક વ્યક્તિ નામ, શહેર, જાતિ, જન્મતારીખ અને વ્યવસાય જેવી લાક્ષણિકતાઓ ધરાવે છે. ઑબ્જેક્ટ આધારિત પરિભાષામાં આવાં લક્ષણો **પ્રોપર્ટી** (property) અથવા **ઍટ્રિબ્યૂટ** (attribute) તરીકે ઓળખાય છે. (જેને આપણે

ગુણધર્મ કે સંબંધિત ગુણધર્મ પણ કહી શકીએ.) એક વ્યક્તિને બીજી વ્યક્તિથી અલગ પાડવા માટે આપણે તેની લાક્ષણિકતાની કિંમતનો ઉપયોગ કરીએ છીએ. 'રામ' અને 'શ્યામ' નામ બે અલગ-અલગ વ્યક્તિઓની ઓળખ આપે છે, પણ જ્યારે બે વ્યક્તિઓનાં નામ એક જ હોય, ત્યારે જન્મતારીખ જેવી અન્ય લાક્ષણિકતાની મદદથી આપણે તેમને અલગ પાડી શકીએ. આ રીતે ઑબ્જેક્ટને ઓળખવા માટે આપણે આ લાક્ષણિકતા (attribute)ની કિંમત વાપરીએ છીએ. આ કિંમતો **સ્ટેટ** (state) તરીકે ઓળખાય છે. આ ઉપરાંત ઑબ્જેક્ટ સાથે **બિહેવ્યર** (behaviour) સંકળાયેલ હોય છે. ઉદાહરણ તરીકે, વ્યક્તિ જન્મ લે છે, નામ મેળવે છે, સ્થાન બદલે છે. બિહેવ્યર **મેથડ** (method) તરીકે પણ ઓળખાય છે. ઑબ્જેક્ટની કિંમત તેના બિહેવ્યરના કારણે બદલાઈ શકે છે. આ રીતે, વાસ્તવિક વિશ્વમાં કોઈ પણ વસ્તુ (ઑબ્જેક્ટ) તે કયા નામથી ઓળખાય છે (ઓળખ - identity), તે શું છે (તેની સ્થિતિ - state) અને તે શું કરે છે (behaviour) તેના દ્વારા વર્ણવી શકાય છે.

ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગમાં ઑબ્જેક્ટનું વર્ણન કરતી જુદી-જુદી લાક્ષણિકતાઓ **ડેટાફિલ્ડ** (data-field) તરીકે પણ ઓળખવામાં આવે છે. ઑબ્જેક્ટ સાથે સંકળાયેલા વિવિધ ડેટા ઍટ્રિબ્યૂટ અને બિહેવ્યર મેથડને સામૂહિક રીતે તેના **મેમ્બર** કે ફિચર (member અથવા feature) તરીકે ઓળખવામાં આવે છે.

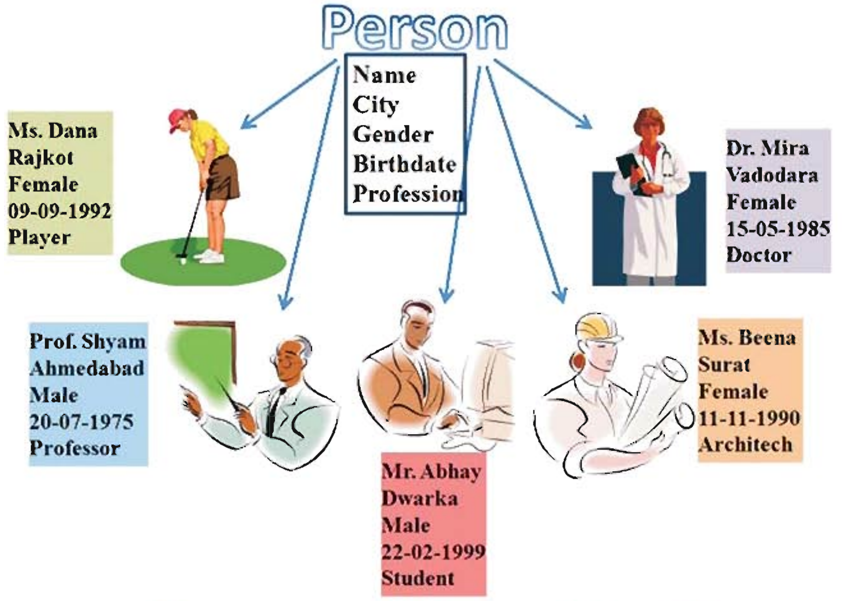
જ્યારે આપણે કોઈ સૉફ્ટવેર વિનિયોગની રચના કરીએ છીએ, ત્યારે આપણે જે ઑબ્જેક્ટ પસંદ કરીએ, તે વિનિયોગ માટે અર્થપૂર્ણ હોવા જોઈએ. ઉદાહરણ તરીકે, રેલ્વે-આરક્ષણ વિનિયોગ માટે રેલ્વે ટ્રેન, પ્રવાસી, ટિકિટ અને સ્ટેશન જેવા ઑબ્જેક્ટ યોગ્ય છે, પણ કાર, કમ્પ્યૂટર અને ઘડિયાળ જેવા ઑબ્જેક્ટ આ વિનિયોગ માટે અસંગત છે.

ક્લાસ (Class)

ઉપર જણાવેલાં person ઑબ્જેક્ટના ઉદાહરણમાં સહેલાઈથી જોઈ શકાય છે કે કેટલાક ઑબ્જેક્ટ એકસરખી લાક્ષણિક્તા અને બિહેવ્યર ધરાવતા હોય છે પણ ફક્ત તેના સ્ટેટથી (state) (ડેટાની કિંમત) એકબીજાથી જુદા પડે છે. ઑબ્જેક્ટ આધારિત પદ્ધતિ ક્લાસ (class)ના ખ્યાલનો ઉપયોગ કરે છે, જે ફક્ત તેના ઍટ્રિબ્યૂટની કિંમતોથી જ અલગ હોય તેવા અમૂર્ત સમકક્ષ (abstractly equivalent) ઑબ્જેક્ટને અભિવ્યક્ત કરવા સમર્થ બનાવે છે. ક્લાસને અનેક જુદા-જુદા ઑબ્જેક્ટની એક બ્લ્યુપ્રિન્ટ (નકશો) તરીકે ગણી શકાય.

ક્લાસ એ સમાન લક્ષણો ધરાવતાં બહુવિધ ઑબ્જેક્ટનું એક ટેમ્પ્લેટ (template) છે. તે એકસમાન ઍટ્રિબ્યૂટ અને બિહેવ્યર ધરાવતાં વિવિધ ઑબ્જેક્ટનું એક જૂથ છે. એક જ ક્લાસના જુદા-જુદા ઑબ્જેક્ટનો સિમેન્ટિક હેતુ (semantic purpose) એકસરખો હોય છે. આ રીતે ક્લાસ એ કોઈ ચોક્કસ ઑબ્જેક્ટના જૂથની બધી જ સમાન લાક્ષણિક્તાનો સમાવેશ કરવા માટેનો એક સર્વસાધારણ ખ્યાલ છે.

ઉદાહરણ તરીકે, આપણી પાસે બધી વ્યક્તિઓના સામાન્ય ઍટ્રિબ્યૂટ અને બિહેવ્યરનો સમાવેશ કરતો 'Person' નામનો ક્લાસ છે. અહીં દરેક વ્યક્તિ એ ઑબ્જેક્ટ છે અને તે તેના સ્ટેટ એટલે કે ઍટ્રિબ્યૂટના કિંમતથી તે ઓળખાય છે. આકૃતિ 6.1માં ક્લાસ અને તેના ઑબ્જેક્ટનો તફાવત દર્શાવેલો છે.



આકૃતિ 6.1 : 'Person' નામનો ક્લાસ અને તેના ઑબ્જેક્ટ

હવે આપણે અન્ય ઑબ્જેક્ટ આધારિત ખ્યાલો વિશે શીખતાં પહેલાં ક્લાસ-ડાયાગ્રામ (class-diagram) વિશે ટૂંકમાં પરિચય મેળવીએ.

ક્લાસ-ડાયાગ્રામનો પરિચય (Introduction to Class-Diagram)

ક્લાસ-ડાયાગ્રામ અનેક ક્લાસનો સમૂહ, અવરોધો (constraints) અને જુદા-જુદા ક્લાસ વચ્ચેના સંબંધની સ્થિતિ રજૂ કરે છે.

યુનિફાઇડ મૉડેલિંગ લૅંગ્વેજ (Unified Modelling Language - UML)નો ઉપયોગ ઑબ્જેક્ટ આધારિત સૉફ્ટવેરની પ્રતિકૃતિ (model) તૈયાર કરવામાં કરી શકાય કે જે વિનિયોગની રચના તૈયાર કરવામાં મદદરૂપ થાય. UML એ એક દશ્ય (visual) મૉડેલિંગ ભાષા છે અને તે ઑબ્જેક્ટ મૅનેજમૅન્ટ ગ્રૂપ (Object Management Group - OMG) દ્વારા વ્યાખ્યાયિત કરેલી છે અને તેની જાળવણી કરે છે. UML સૉફ્ટવેર વિનિયોગનાં વિવિધ પાસાંઓ રજૂ કરવા માટે નામનિર્દેશવાળી અનેક આકૃતિઓ જણાવે છે.

ક્લાસ-ડાયાગ્રામનો હેતુ વિનિયોગના સ્થાયી દેખાવની પ્રતિકૃતિ બનાવવાનો છે. ક્લાસ-ડાયાગ્રામ જ ફક્ત રેખાકૃતિઓ (diagrams) છે કે જે ઑબ્જેક્ટ આધારિત ભાષાઓ સાથે સીધેસીધી જોડી શકાય છે અને આથી સૉફ્ટવેર બનાવનાર વ્યક્તિઓમાં તે વ્યાપકપણે વપરાય છે.

ક્લાસ-ડાયાગ્રામમાં કોઈ પણ ક્લાસને રજૂ કરવા માટેનો આઇકોન નીચે જણાવ્યા પ્રમાણે name, attribute અને behaviourનો સમાવેશ કરવા માટે એક લંબચોરસ ત્રણ વિભાગમાં વિભાજિત કરેલો હોય છે :

1. સૌથી ઉપરના વિભાગમાં ક્લાસનું નામ (class name) હોય છે.
2. વચ્ચેના વિભાગમાં ક્લાસનાં ઍટ્રિબ્યૂટ કે પ્રોપર્ટી હોય છે.
3. સૌથી નીચેના વિભાગમાં ક્લાસનું બિહેવ્યર અથવા ઑપરેશન (operation) અથવા મેથડ (method) હોય છે.

આકૃતિ 6.2માં UML પ્રણાલિકા પ્રમાણે ક્લાસની સચિત્ર રજૂઆત દર્શાવી છે, જ્યારે આકૃતિ 6.3માં 'Person'નો ક્લાસ-ડાયાગ્રામ દર્શાવ્યો છે.

Class Name
Visibility attribute : data type = initial value
Visibility operation (argument list) : return type

આકૃતિ 6.2 : UML પ્રણાલિકાગત ક્લાસની રજૂઆત

Person
name : string city: string gender : char = 'M' -birthdate : date profession : string
setBirthdate (d:int,m:int,y:int): date changeCity (newCity:string) :string display ( )

આકૃતિ 6.3 : 'Person' ક્લાસની રેખાકૃતિ

'Person' ક્લાસનો હેતુ ક્લાસ-ડાયાગ્રામના સંદર્ભમાં ક્લાસની કામગીરી સમજવામાં મદદરૂપ માહિતી પૂરી પાડવાનો છે. તેણે ક્લાસનાં દરેક ઍટ્રિબ્યૂટ અને કાર્યોનો સમાવેશ કરવાની જરૂર નથી.

આકૃતિ 6.2માં દર્શાવ્યા પ્રમાણે, UMLની સંકેતલિપિમાં ઍટ્રિબ્યૂટની વાક્યરચના નીચે દર્શાવ્યા પ્રમાણે કરી શકાય :

[<visibility>] <attribute name> [: <attribute data type> [= <initial value>]]

અહીં, ચોરસ કૌંસ [ ]ની જોડીમાં લખવામાં આવેલી બાબત વૈકલ્પિક છે અને કોણીયકૌંસ < > ની જોડીમાં લખેલી બાબતની કિંમત વપરાશકર્તાએ જણાવવી પડે છે. અહીં દ્રશ્યતા (visibility) અંગત, સુરક્ષિત, જાહેર અથવા પૅકેજ (package) હોઈ શકે અને તે જણાવવા માટે અનુક્રમે –, #, + અને ~ સંકેતો વાપરવામાં આવે છે. આપણે દ્રશ્યતા વિશે હવે પછીના પ્રકરણમાં અભ્યાસ કરીશું. ઍટ્રિબ્યૂટ સામાન્ય રીતે ચલ (variable)નો નિર્દેશ કરે છે. ડેટાટાઇપ અને તેની પ્રારંભિક કિંમત પ્રોગ્રામની શરૂઆતમાં ક્યા પ્રકારનો ડેટા સંગ્રહ કર્યો છે અને તેની કિંમત શું છે તેનો નિર્દેશ કરે છે. અહીં જોઈ શકાય છે કે માત્ર ઍટ્રિબ્યૂટ નેઇમ (attribute-name) જ ફરજિયાત છે અને અન્ય તમામ બાબત વૈકલ્પિક છે.

ઍટ્રિબ્યૂટ ઘોષિત કરવા માટે નીચે કેટલાંક ઉદાહરણ આપેલાં છે :

name : string

- balance : float = 0.0

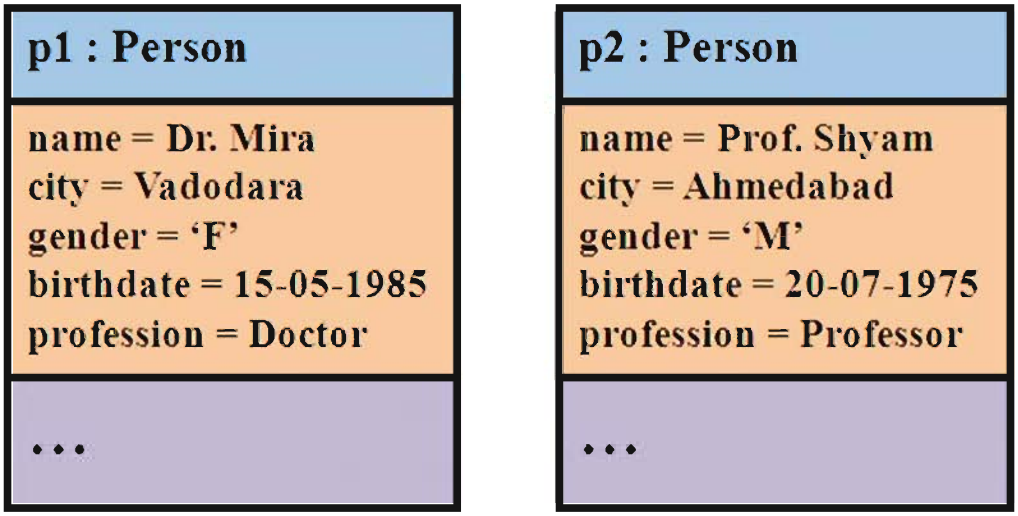
આકૃતિ 6.2માં દર્શાવ્યા પ્રમાણે, UMLની સંકેતલિપિમાં કોઈ કાર્યને નીચે જણાવેલી વાક્યરચના વાપરીને જણાવી શકાય છે :

[<visibility>] <method name> (parameter list separated by comma) : <return data type>

અગાઉના ઉદાહરણમાં મેથડ (method)નું ઉદાહરણ setBirthdate(d:int, m:int, y:int) : date છે. અહીં પ્રાચલ ડેટાટાઇપ અને વૈકલ્પિક પ્રારંભિક કિંમત સાથે જણાવી શકાય છે; ઉદાહરણ તરીકે gender : char = 'M'. પ્રાચલ ફક્ત વાંચી જ શકાય (read only) તેવા છે કે નહીં, તેના આધારે તે નિવેશ અથવા નિર્ગમ તરીકે જણાવી શકાય છે.

UML રેખાકૃતિ વિનિયોગ કઈ પ્રોગ્રામિંગ ભાષામાં કોડ કરવાનો છે, તેના ઉપર આધારિત નથી. કેટલીક સૉફ્ટવેર બનાવનાર વ્યક્તિઓ ઍટ્રિબ્યૂટ અને કાર્યોને પ્રમાણભૂત UML સંકેતલિપિમાં જણાવવાને બદલે પ્રોગ્રામિંગ ભાષાના વધુ પરિચિત માળખામાં જણાવવાનું પસંદ કરે છે. આ પદ્ધતિ જ્યાં સુધી દરેક સંબંધિત વ્યક્તિને માટે અર્થપૂર્ણ રહે છે, ત્યાં સુધી તે બરાબર છે.

વિનિયોગના અમલ દરમિયાન ઑબ્જેક્ટને તેના સ્ટેટ (state) વડે રજૂ કરવામાં આવે છે. આ રીતે ઑબ્જેક્ટ ચલિત (dynamic) હોય છે. આકૃતિ 6.1ને અનુરૂપ 'person' ક્લાસના ઑબ્જેક્ટ (ઇન્સ્ટન્સ - instance તરીકે પણ ઓળખવામાં આવે છે)ને આકૃતિ 6.4માં દર્શાવ્યા પ્રમાણે રજૂ થાય છે.



આકૃતિ 6.4 : 'Person' ક્લાસના p1 અને p2 ઑબ્જેક્ટની રેખાકૃતિ

ઈનકૅપ્સ્યુલેશન (Encapsulation)

કોઈ પણ કમ્પ્યૂટર પ્રોગ્રામનાં બે મૂળ અંગ ડેટા અને કાર્ય છે. સંગઠિત પ્રોગ્રામિંગ (Structured Programming) આ બે ઘટકોને અલગ-અલગ ઘટક તરીકે ગણે છે જ્યારે ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગ તેને એક ઘટક તરીકે જુએ છે.

પ્રક્રિયાગત પ્રોગ્રામિંગમાં (Procedural Programming) પ્રોગ્રામના કોઈ પણ ભાગમાં ડેટાની કિંમત બદલી શકાય છે. તે ફેરફારથી સુરક્ષિત નથી. આ સમસ્યાનો ઉકેલ ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગમાં ઈનકૅપ્સ્યુલેશન દ્વારા લાવી શકાય છે. અહીં ડેટા અને મૅથડ વડે માહિતીની ગણતરી થાય તે સમયે પ્રોગ્રામના અન્ય ઘટકો વડે ડેટામાં ફેરફાર અથવા દુરુપયોગ સામે સુરક્ષિતતા બક્ષવામાં આવે છે. આ પ્રકારની રચના કે જેના વડે ડેટા અને મેથડ સામે રક્ષણ પૂરું પાડવામાં આવે છે તેને **ઈનકૅપ્સ્યુલેશન (Encapsulation)** કહેવામાં આવે છે. ડેટા અને મેથડને લપેટીને (વીંટીને) બનાવેલા એક એકમ જે ક્લાસના નામથી જાણીતા છે તેને અંગત (private) ઘોષિત કરવાથી શક્ય બને છે, અને ક્લાસના અંગત સભ્ય (private member) બહારના વિશ્વને સીધા ઉપલબ્ધ થવા દેતા નથી. જો તેની જરૂર હોય, તો સાર્વજનિક મેથડ (public method) વડે ડેટાને ઉપલબ્ધ બનાવી શકાય છે. આ રીતે, ઈનકૅપ્સ્યુલેશન ડેટાને છુપાવવાની ક્ષમતા પૂરી પાડે છે. આપણે આ પછીનાં પ્રકરણોમાં ક્લાસ અને અંગત/સાર્વજનિક ડેટા અને મેથડ વિશે અભ્યાસ કરીશું.

ઈનકૅપ્સ્યુલેશન બિનઈરાદાપૂર્વકની ક્રિયાઓ અને બહારના અન્ય ઑબ્જેક્ટ વડે જરૂર વગરના ડેટાને મેળવવાથી સલામત રાખે છે. પ્રક્રિયાગત પ્રોગ્રામિંગમાં સાર્વજનિક ડેટાવિસ્તાર માહિતીની વહેંચણી માટે સામાન્ય રીતે વપરાય છે, જ્યારે ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગ તેનાથી વિરુદ્ધ અન્ય પ્રોગ્રામ્સ વડે સાર્વજનિક ડેટાને (વૈશ્વિક ચલ સિવાયના ડેટાના ઉપયોગને) સીધો મેળવવાની ક્રિયાને અટકાવે છે. ઑબ્જેક્ટ પોતાના ડેટાની કિંમતમાં જ ફેરફાર કરી શકે છે. અન્ય ઑબ્જેક્ટ તેને જોઈ શકે છે અથવા તે ઑબ્જેક્ટના માલિક (owner)ને સંદેશો મોકલીને આ ડેટાને બદલી શકે છે.

ડેટા-ઍબ્સ્ટ્રૅક્શન (Data-Abstraction)

ડેટા-ઍબ્સ્ટ્રૅક્શન એ ઑબ્જેક્ટની જરૂરી લાક્ષણિકતાઓને તેના અમલીકરણની વિગત સિવાયની માહિતીને રજૂ કરવાની પ્રક્રિયા છે. ઍબ્સ્ટ્રૅક્શન એ એક ખ્યાલ છે જે આંટીઘૂંટી કે ગૂંચને છુપાવે છે અને તે જે કાર્ય કરે છે, તે જણાવે છે પણ કઈ રીતે કરે છે, તે જણાવતો નથી.

હવે આપણે આપણી જિંદગીનું એક વાસ્તવિક ઉદાહરણ ટેલિવિઝનનું લઈએ. આપણે ટેલિવિઝનને ચાલુ અને બંધ કરી શકીએ છીએ, ચેનલ બદલી શકીએ છીએ તેમજ આપણને અનુકૂળ અવાજનું માપ રાખી શકીએ છીએ. પણ આપણે તેની આંતરિક રચના જાણતા નથી કે તે હવા કે તાર દ્વારા કઈ રીતે સંકેતો મેળવે છે, તે સંકેતોનું કઈ રીતે રૂપાંતર કરે છે અને અંતમાં તેને પડદા ઉપર પ્રદર્શિત કરે છે. આ રીતે ટેલિવિઝનનું આંતરિક કાર્ય તેના બાહ્ય સેતુથી તદ્દન અલગ છે. આપણે તેની આંતરિક રચનાની સમજ કે જ્ઞાન વિના તેનો ઉપયોગ પાવર-બટન, ચેનલ બદલવાનાં બટન અને અવાજના વૉલ્યુમમાં ફેરફાર તેના બાહ્ય સેતુ વડે કરી શકીએ છીએ.

આ રીતે ડેટા-ઍબ્સ્ટ્રૅક્શન એ એક એવી કાર્યરીતિ (ટૅક્નિક) છે જે તે વાપરવાના સેતુ અને અમલીકરણને અલગ કરવાના ખ્યાલ પર આધારિત છે. તે પ્રોગ્રામિંગમાં નવો ખ્યાલ નથી. અનેક વ્યક્તિઓ કદાચ આ બાબતની જાણ વગર અમુક અંશે તેનો ઉપયોગ અત્યાર સુધી કરે જ છે. ઉદાહરણ તરીકે, આપણે જ્યારે C પ્રોગ્રામિંગમાં sqrt(25) અને printf("Hello world") જેવા વિધેયનો ઉપયોગ કરીએ છીએ ત્યારે આપણે કદી પણ એ નથી વિચાર્યું કે તે કઈ રીતે કાર્ય કરતા હશે.

જરૂરી નિવેશ ડેટા પ્રાચલો સાથે ઉપયોગકર્તા નિર્મિત (user-defined) વિધેય પણ ડેટા-ઍબ્સ્ટ્રૅક્શન પૂરું પાડે છે. હવે આપણે સ્ટ્રક્ચર (structure)નો ઉપયોગ કરીને એક C પ્રોગ્રામ વડે 'date' પ્રકારના ડેટા જણાવીએ. અહીં ડેટા ત્રણ ઘટકોનો બનેલો છે : દિવસ, માસ અને વર્ષ. આપણે આ ઘટકો માટે ડેટાનો મૂળભૂત પૂર્ણાંક સંખ્યા પ્રકાર (primitive integer datatype)નો ઉપયોગ કરીએ. હવે 'dateDiff' નામનું વિધેય લો, જેમાં બે તારીખ પ્રાચલ તરીકે લેવામાં આવે છે અને આ બે તારીખ વચ્ચે કેટલા દિવસ છે તે કિંમત પરત આપે છે. આ વિધેયના ઉપયોગકર્તાએ જાણવાની જરૂર નથી કે બે તારીખના તફાવતની ગણતરી કઈ રીતે થાય છે. પણ ઉપયોગકર્તા તે વિધેયની ફક્ત સિગ્નેચર(signature), એટલે કે વિધેયનું નામ, પ્રાચલની સંખ્યા અને તેનો પ્રકાર તેમજ પરત મળતી કિંમતના પ્રાચલનો પ્રકાર જાણે તે જરૂરી છે. જો કોઈ કારણથી તફાવત શોધવાની પ્રક્રિયામાં ફેરફાર થાય, તો આ વિધેય વાપરનાર પ્રોગ્રામના કોઈ પણ ભાગમાં તેની અસર થતી નથી.

આ રીતે ડેટા-ઍબ્સ્ટ્રૅક્શન આપણને વાપરવા માટેની એક રૂપરેખા કે ઢાંચો (templates) પૂરો પાડે છે. સિસ્ટમ ડેટાનો સંગ્રહ, નિર્માણ અને જાળવણી કઈ રીતે થાય છે, તેની કેટલીક માહિતી ગુપ્ત રાખે છે. બહારની દુનિયાને જે કંઈ દશ્યમાન

છે, તે ડેટા પ્રકારની અમૂર્ત (abstract) રીતભાત (behaviour) છે; પણ તેનું અમલીકરણ કઈ રીતે થયેલું છે, તે ગોપનીય રહે છે અને આથી તેનો ઉપયોગ કરનાર વ્યક્તિને અસર કર્યા વગર જરૂરિયાત પ્રમાણે તેમાં ફેરફાર કરી શકાય છે.

C અને C++ ના ઍબ્સ્ટ્રેક્ટ ડેટાટાઇપ્સ (Abstract Data Types - ADT) અથવા સ્ટ્રક્ચર (structures - struct) અને C++/જાવાના ક્લાસ એ ડેટા-ઍબ્સ્ટ્રેક્શનનાં ઉદાહરણ છે. ADTમાં આપણે ડેટાનો પ્રકાર વ્યાખ્યાયિત કરીને તેના ઉપરની પ્રક્રિયાઓ ફક્ત જણાવીએ છીએ. આપણે તે પ્રક્રિયાઓનું અમલીકરણ કઈ રીતે થાય છે, તે બતાવતા નથી.

ડેટા ઇનકૅપ્સ્યુલેશન અને ડેટા-ઍબ્સ્ટ્રેક્શનનો મૂળભૂત તફાવત : ઇનકૅપ્સ્યુલેશન પ્રોગ્રામની બહારથી ડેટા મેળવવાનું અટકાવીને (inaccessible) ડેટાને સુરક્ષિત કરે છે, જ્યારે ઍબ્સ્ટ્રેક્શન પ્રક્રિયાના અમલીકરણની માહિતી ગુપ્ત રાખીને ડેટાની રજૂઆત વડે ડેટાને સુરક્ષિત બનાવે છે.

મૅસેજિંગ (Messaging)

ઑબ્જેક્ટ આધારિત પરિભાષામાં મેથડ (method) બોલાવવાની ક્રિયાને **મૅસેજ (message)** કહેવામાં આવે છે. ઇનકૅપ્સ્યુલેશનને કારણે બધી મૅથડકૉલનું નિયંત્રણ ઑબ્જેક્ટ વડે થાય છે, જે (ઑબ્જેક્ટ) મૅથડને ઓળખે છે.

જુદા-જુદા ક્લાસમાં સમાન નામ ધરાવતી એકસરખી મૅથડ હોઈ શકે. ઉદાહરણ તરીકે, 'date' ક્લાસમાં 'display' નામની મૅથડ છે જે 'date' ઑબ્જેક્ટને અમુક ચોક્કસ સ્વરૂપમાં પ્રદર્શિત કરે છે. ધારો કે 'time' અને 'person' નામના અન્ય ક્લાસ છે, અને સમય ચોક્કસ સ્વરૂપમાં પ્રદર્શિત કરવા અને વ્યક્તિની માહિતી પ્રદર્શિત કરવા માટે બંનેની એક જ નામની મૅથડ 'display' છે. જ્યારે પ્રોગ્રામમાં 'display' મૅથડ વાપરવામાં આવે છે, ત્યારે કઈ મૅથડનો અમલ કરવાનો છે તે કઈ રીતે જાણવું ? જે ઑબ્જેક્ટ વડે મૅથડ વાપરવામાં આવે તેની મદદથી તે નક્કી થાય છે. ઉદાહરણ તરીકે, જો 'person' ક્લાસના ઑબ્જેક્ટ વડે 'display' વપરાય, તો તે 'person' ક્લાસની 'display' મૅથડનો અમલ કરશે.

પોલિમૉર્ફિઝમ (Polymorphism)

પોલિમૉર્ફિઝમ એટલે 'અનેક સ્વરૂપ' કે 'બહુરૂપતા'. એક મૅથડ કે પ્રક્રિયાનાં જુદાં-જુદાં સ્વરૂપ (form) હોઈ શકે.

ધારો કે આપણે બે સંખ્યામાંથી મોટી સંખ્યા શોધવાનું વિધેય 'max' લખેલું છે. આ વિધેય પ્રાચલ તરીકે બે પૂર્ણાંક સંખ્યા લે છે અને મોટી પૂર્ણાંક કિંમત પરત આપે છે. હવે ધારો કે 'max' નામનું એક વધારે વિધેય આપણે ઉમેરવા ઇચ્છીએ છીએ, જે કોઈ ઍરે (array)માં સંગ્રહ કરેલી પૂર્ણાંક સંખ્યાઓમાંથી મહત્તમ સંખ્યા શોધે. આ વિધેય તે ઍરે અને તેના કદને પ્રાચલ તરીકે લેશે અને મહત્તમ પૂર્ણાંક સંખ્યા પરત આપશે. શું એક જ નામનાં એક કરતાં વધુ વિધેયો વ્યાખ્યાયિત કરવા શક્ય છે ? કેટલીક પ્રોગ્રામિંગ ભાષામાં આ પ્રશ્નનો જવાબ 'ના' છે, પણ ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગમાં જ્યાં સુધી મેથડ તેની સિગ્નેચર (પ્રાચલની સંખ્યા અને પ્રકાર)થી અલગ છે, ત્યાં સુધી તે પ્રશ્નનો જવાબ 'હા' છે.

ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગ એક જ ક્લાસમાં એક કરતાં વધારે મૅથડ કે જેનાં નામ સરખાં હોય પણ સિગ્નેચરથી અલગ હોય તેને વ્યાખ્યાયિત કરવાની મંજૂરી આપે છે. આ લાક્ષણિકતાને **ફંક્શન કે મૅથડ ઓવરલોડિંગ (function or method overloading)** કહેવામાં આવે છે.

ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગ ઑબ્જેક્ટ ઉપર પ્રક્રિયકો સાથેની પદાવલિ લખવાની પણ છૂટ આપે છે. ઉદાહરણ તરીકે, આપણે 'date1-date2' જેવી પદાવલિ પણ વાપરી શકીએ, જ્યાં બંને સંકાર્ય (operands) એ 'date' ક્લાસના ઑબ્જેક્ટ છે. અહીં '-' પ્રક્રિયા એ બે સંખ્યાની બાદબાકી કરવાની પ્રક્રિયા જેમકે n1-n2, કરતાં અલગ રીતે કરવામાં આવે છે. અહીં એ જ પ્રક્રિયાનો અર્થ સંકાર્ય (operands)ના ડેટા પ્રકારના આધારે અલગ કરવામાં આવે છે. આ પ્રકારની બહુરૂપતા (પોલિમૉર્ફિઝમ) **ઑપરેટર ઓવરલોડિંગ (operator overloading)**ને કારણે શક્ય બને છે.

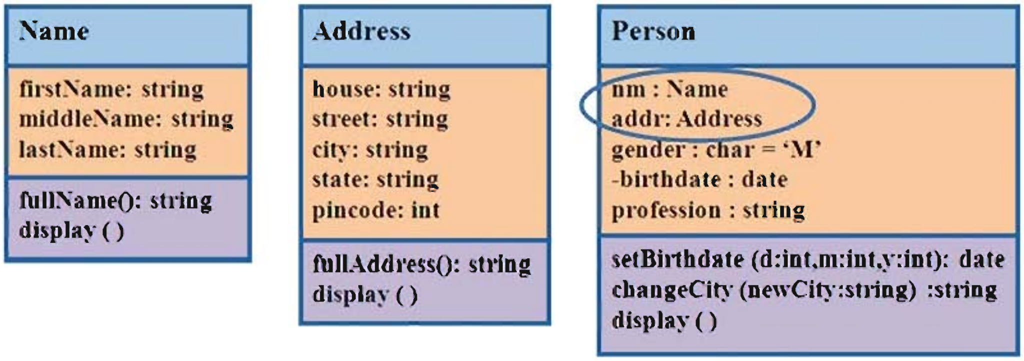
આ રીતે બે પ્રકારનાં ઓવરલોડિંગ વડે પોલિમૉર્ફિઝમ શક્ય બને છે : (1) ફંક્શન ઓવરલોડિંગ (function overloading) અને (2) ઑપરેટર ઓવરલોડિંગ (operator overloading). સામાન્ય રીતે, અલગ-અલગ સંદર્ભમાં એક જ નામના જુદા-જુદા અર્થની ક્ષમતા કે સામર્થ્યને ઓવરલોડિંગ (overloading) કહેવામાં આવે છે.

એગ્રિગેશન અને કૉમ્પોઝિશન (Aggregation and Composition)

જ્યારે એક ક્લાસના ઑબ્જેક્ટ બીજા ક્લાસના ઑબ્જેક્ટથી બનેલા હોય, ત્યારે તેને એગ્રિગેશન (aggregation) અથવા કૉમ્પોઝિશન (composition) કહેવામાં આવે છે. તે અલગ-અલગ ક્લાસ વચ્ચેનો 'પૂર્ણ' અથવા 'તેમાંનો અંશ' સંબંધ રજૂ કરે છે. ઉદાહરણ તરીકે, મધરબોર્ડ કમ્પ્યૂટરનો એક ભાગ છે.

આપણે જાણીએ છીએ કે કમ્પ્યૂટર મધરબોર્ડ, સ્ક્રીન, કી-બોર્ડ અને માઉસ જેવા અનેક ઘટકોનું બનેલું હોય છે. સ્ક્રીન સ્વયં જ લંબાઈ, પહોળાઈ અને મૉડલ જેવા ગુણધર્મો (ઍટ્રિબ્યૂટ) સાથેનો એક ક્લાસ હોઈ શકે. એ જ રીતે મધરબોર્ડને પણ મોડલ અને કંપની જેવા ગુણધર્મો સાથેનો ક્લાસ બનાવી શકાય. જ્યારે આપણે 'computer' નામનો ક્લાસ વ્યાખ્યાયિત કરીએ, ત્યારે તેમાં 'motherboard' અને 'screen' ક્લાસના ઑબ્જેક્ટ જેવા ઍટ્રિબ્યૂટનો સમાવેશ કરેલો હશે. આ રીતે તેમાં સમાવેશનો નિર્દેશ સૂચવે છે.

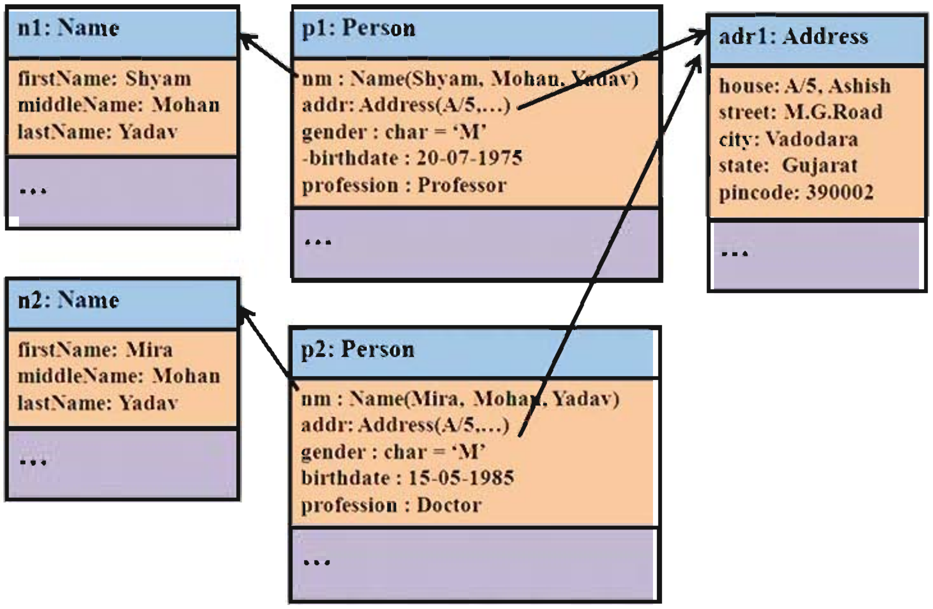
હવે આપણે અગાઉ ચર્ચા કરેલા 'Person' ક્લાસમાં ફેરફાર કરીએ. સૌપ્રથમ આકૃતિ 6.5માં દર્શાવ્યા પ્રમાણે આપણે બે નવા ક્લાસ 'Name' અને 'Address' બનાવીએ. આ 'Name' ક્લાસમાં firstname, middlename અને lastname એટ્રિબ્યૂટ છે અને 'Address' ક્લાસમાં house (જે ઘરની વિગતોનો નિર્દેશ કરે છે), street, city, state અને pin code ઍટ્રિબ્યૂટ છે. હવે 'Person' ક્લાસને બદલીએ અને ઍટ્રિબ્યૂટનાં નામ Name અને Addressને બદલી અનુક્રમે nm અને addr રાખીએ. nm અને addr ઍટ્રિબ્યૂટનો ડેટાપ્રકાર અનુક્રમે 'Name' અને 'Address' ક્લાસ છે. આ રીતે 'Person' ક્લાસમાં 'Name' અને 'Address' ક્લાસના ઑબ્જેક્ટ છે.



આકૃતિ 6.5 : 'Name' અને 'Address' ક્લાસના ઍટ્રિબ્યૂટ સાથેનો 'Person' ક્લાસ

ઍગ્રિગેશન અને કૉમ્પોઝિશનની તુલના (Aggregation Vs Composition)

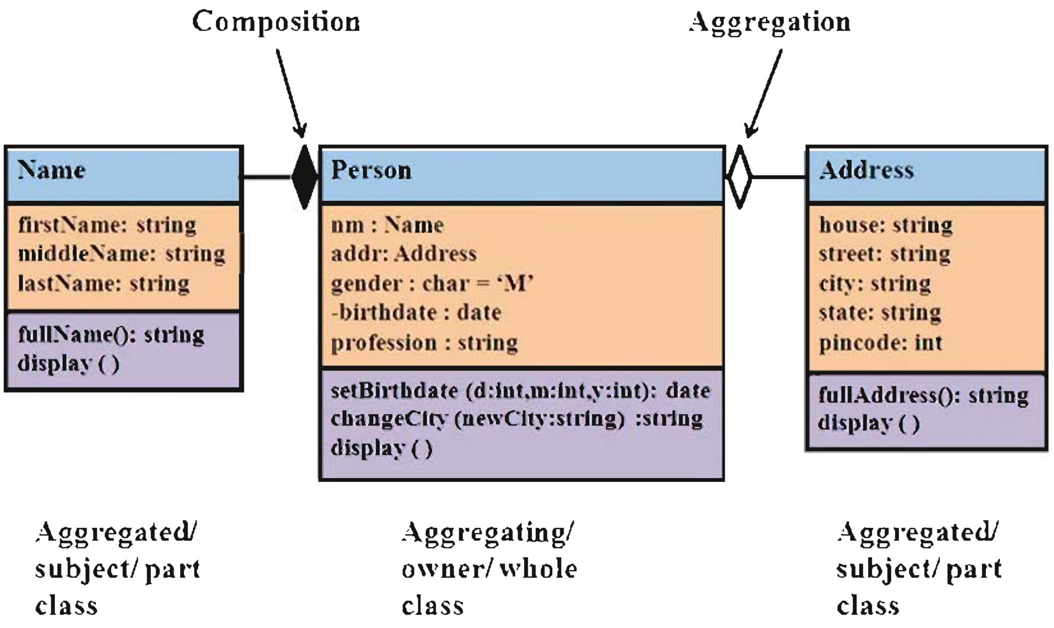
ઍગ્રિગેશન બે ક્લાસ વચ્ચેના ભિન્ન (non-exclusive) સંબંધને રજૂ કરે છે. ઍગ્રિગેશનમાં કોઈ ક્લાસ કે જે ઓનર ક્લાસનો એક ભાગ છે, તે સ્વતંત્ર રીતે અસ્તિત્વ ધરાવે છે. આ આંશિક ક્લાસ ઑબ્જેક્ટનો કાર્યકાળ ઓનર ક્લાસ (ownerclass)થી નક્કી થતો નથી. ઉદાહરણ તરીકે, મધરબોર્ડ જોકે કમ્પ્યૂટરનો એક ભાગ છે, પણ તે કમ્પ્યૂટરથી સ્વતંત્ર એક અલગ ઘટક તરીકે અસ્તિત્વ ધરાવે છે. આ રીતે મધરબોર્ડ કમ્પ્યૂટર સાથે અભિન્ન રીતે (exclusively) સંકળાયેલ નથી. આ જ રીતે આકૃતિ 6.6માં દર્શાવ્યા પ્રમાણે બે અથવા વધારે વ્યક્તિઓ address ઑબ્જેક્ટનો ઉપયોગ કરે છે. આથી, address એ કોઈ એક જ વ્યક્તિ માટે હોય તે જરૂરી નથી.



આકૃતિ 6.6 : અલગ-અલગ name ધરાવતા પણ એક જ address ધરાવતા બે 'Person' ઑબ્જેક્ટ

આકૃતિ 6.7માં દર્શાવ્યા પ્રમાણે મૂળભૂત એગ્રિગેશનને પૂર્ણ ક્લાસને અડીને એક ખાલી હીરાના ચિહ્નનો ઉપયોગ કરીને બનાવવામાં આવે છે.

કોમ્પોઝિશન બે ક્લાસ વચ્ચેનો અજોડ (exclusive) સંબંધ જણાવે છે. કોમ્પોઝિશન એક પ્રબળ કે મજબૂત પ્રકારનું એગ્રિગેશન છે, જેમાં આંશિક ક્લાસનો કાર્યકાળ ઓનર ક્લાસના અસ્તિત્વ આધારિત હોય છે. જો એકત્ર ક્લાસ (aggregating class)નો ઑબ્જેક્ટ દૂર કરવામાં આવે, તો તેના આંશિક ક્લાસ ઑબ્જેક્ટ પણ નષ્ટ થશે. ઉદાહરણ તરીકે, 'Person' ક્લાસનો કોઈ ઑબ્જેક્ટ ડિલીટ કરવામાં આવે, તો 'Name' ક્લાસનો ઑબ્જેક્ટ પણ નષ્ટ થશે. આકૃતિ 6.6માં દર્શાવ્યા પ્રમાણે Name અજોડ રીતે માત્ર એક જ વ્યક્તિ સાથે જોડાયેલું છે. કોમ્પોઝિશનનો સંબંધ આકૃતિ 6.7માં દર્શાવ્યા પ્રમાણે એક ભરાયેલા હીરાના ચિહ્નને પૂર્ણ ક્લાસને અડીને બતાવવામાં આવે છે.



આકૃતિ 6.7 : કોમ્પોઝિશન અને એગ્રિગેશન

ઉપર આપેલા ઉદાહરણમાં, 'Person' ક્લાસ અને 'Name' ક્લાસ વચ્ચેનો સંબંધ કોમ્પોઝિશન રિલેશનશિપ છે, જ્યારે 'Person' ક્લાસ અને 'Address' ક્લાસ વચ્ચેનો સંબંધ એગ્રિગેશન રિલેશનશિપ છે. એક જ ઍડ્રેસ એક કરતાં વધારે વ્યક્તિઓનું હોઈ શકે. આથી, જ્યારે કોઈ વ્યક્તિ નષ્ટ કરવામાં આવે છે, ત્યારે તેને સુસંગત 'Name' ઑબ્જેક્ટને નષ્ટ કરવામાં આવે છે, પણ 'Address' નષ્ટ કરવામાં આવતું નથી.

નોંધ : કોઈ ક્લાસ જ્યારે અન્ય ક્લાસના ઑબ્જેક્ટ ધરાવતા હોય ત્યારે તે ઓનર ક્લાસ (ownerclass) અથવા પૂર્ણ ક્લાસ (whole class) અથવા એકત્ર ક્લાસ (aggregating class) તરીકે ઓળખાય છે. ઉદાહરણ તરીકે, આકૃતિ 6.7માં દર્શાવેલ 'Person' ક્લાસ એગ્રિગેટિંગ ક્લાસનું ઉદાહરણ છે.

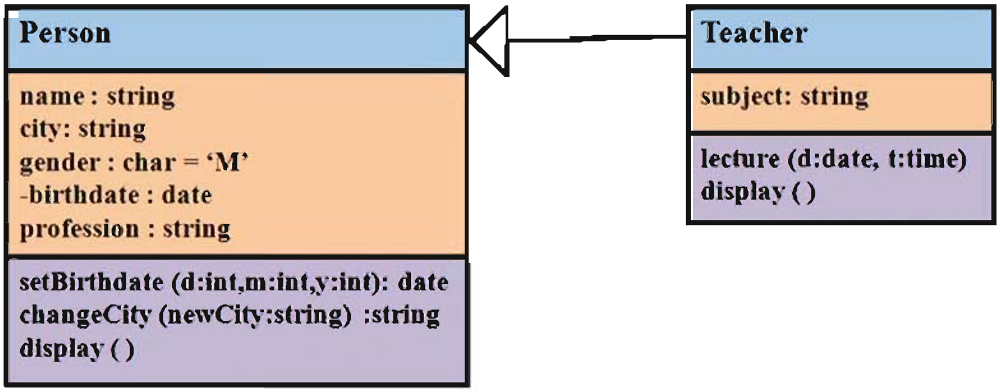
જે ક્લાસ ઓનરક્લાસમાં સમાયેલ છે, તેને સબ્જેક્ટ ક્લાસ (subject class) અથવા આંશિક ક્લાસ (part class) અથવા એકત્રિત ક્લાસ (aggregated class) તરીકે ઓળખાય છે. ઉદાહરણ તરીકે, આકૃતિ 6.7માં દર્શાવેલ 'Name' અને 'Address' ક્લાસ એગ્રિગેટેડ ક્લાસનાં ઉદાહરણ છે.

ઇનહેરિટન્સ (Inheritance)

ઇનહેરિટન્સ સામાન્ય રીતે બે ક્લાસ વચ્ચે 'એક પ્રકારનો' ('is-a-kind-of') સંબંધ જણાવે છે. જ્યારે એક ક્લાસ અન્ય ક્લાસનો પ્રકાર હોય ત્યારે આ યોગ્ય છે. ઉદાહરણ તરીકે, શિક્ષક એક પ્રકારની વ્યક્તિ છે. આથી, 'Person' ક્લાસના બધા ઍટ્રિબ્યૂટ અને મેથડ 'Teacher' ક્લાસને પણ લાગુ પડે છે. બીજા શબ્દોમાં કહીએ તો 'Person' ક્લાસના બધા ઍટ્રિબ્યૂટ અને બિહેવ્યર (ગુણધર્મ) 'Teacher' ક્લાસમાં વારસા (inherit)માં મળે છે. આ ઉપરાંત 'Teacher' ક્લાસના વધારાના ઍટ્રિબ્યૂટ જેવાં કે subject અને મેથડ જેમકે વિષયનાં lecture હોઈ શકે. આવા સંજોગોમાં 'Person' ક્લાસનો ઉપયોગ કરીને 'Teacher' ક્લાસ વ્યાખ્યાયિત કરી શકાય.

ઈનહેરિટન્સ અન્ય હયાત ક્લાસના ગુણધર્મોને વારસામાં મેળવીને ઑબ્જેક્ટના નવા ક્લાસને વ્યાખ્યાયિત કરવાના સામર્થ્યનો નિર્દેશ કરે છે. ઑબ્જેક્ટ આધારિત પરિભાષામાં નવા ક્લાસને સબક્લાસ (subclass) અથવા ચાઇલ્ડક્લાસ (childclass) અથવા ડિરાઇવ્ડ ક્લાસ (derived class) કહેવામાં આવે છે, જ્યારે હયાત ક્લાસને સુપર ક્લાસ (super class) અથવા પૅરેન્ટક્લાસ (parent class) અથવા બેઇઝ ક્લાસ (base class) કહેવામાં આવે છે. સુપર ક્લાસનાં ડેટા ઍટ્રિબ્યૂટ અને મેથડ ઑબ્જેક્ટનો સબક્લાસમાં તેનાં declarations ફરી લખ્યા વિના ઉપલબ્ધ હોય છે. જ્યાં હયાત મેથડ ફરી વ્યાખ્યાયિત કર્યા સિવાય વાપરવાની હોય ત્યાં આ ગુણધર્મ પુનઃઉપયોગી સુવિધા પૂરી પાડે છે. વધારામાં સબક્લાસમાં નવો ડેટા અને મેથડ સભ્યનો ઉમેરો વિસ્તૃત કરવા માટે કરી શકાય છે. સબક્લાસમાં જરૂર પ્રમાણે મેથડને ફરી વ્યાખાયિત કરવાની મંજૂરી આપે છે.

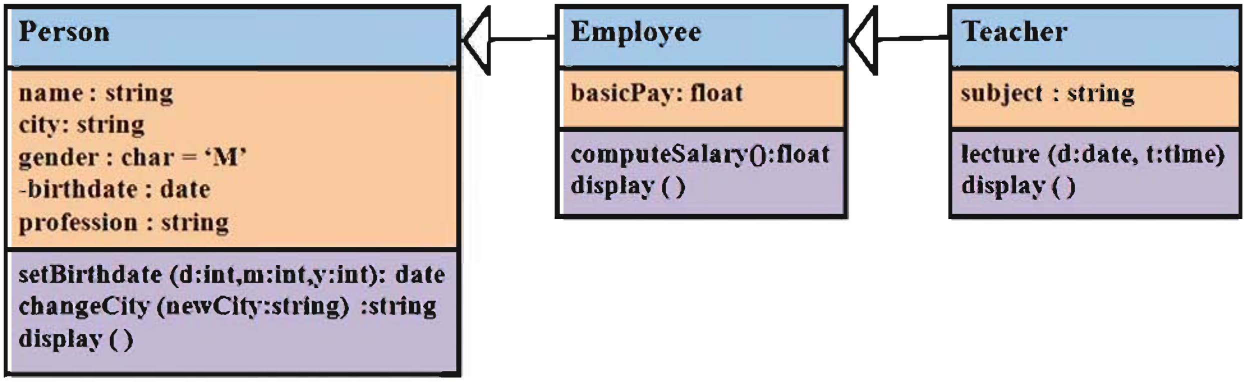
ક્લાસ ડાયાગ્રામમાં ઈનહેરિટન્સ બતાવવા માટે આકૃતિ 6.8માં દર્શાવ્યા પ્રમાણે સુપરક્લાસ તરફ નિર્દેશિત કરતા તીરનો ઉપયોગ કરવામાં આવે છે. ઉપરના ઉદાહરણમાં 'Person' એ સુપરક્લાસ છે અને 'Teacher' સબક્લાસ છે.



આકૃતિ 6.8 : ઈનહેરિટન્સનું ઉદાહરણ

સર્વસામાન્ય નિયમ એ ઈનહેરિટન્સનું બીજું નામ છે અથવા એક સંબંધ છે. બે ક્લાસમાંથી જ્યાં એક ક્લાસ બીજા ક્લાસની વિશિષ્ટ આવૃત્તિ છે, તેની વચ્ચેના સંબંધનો નિર્દેશ કરે છે. સુપરક્લાસમાં સામાન્ય ઍટ્રિબ્યૂટ અને મેથડને વ્યાખ્યાયિત કરવામાં આવે છે. સબક્લાસ એ વધારાના ઍટ્રિબ્યૂટ અને મેથડ સાથેની વિશિષ્ટ આવૃત્તિ છે.

અમુક સમયે વિવિધ ક્લાસ વચ્ચે ઈનહેરિટન્સનો આદર્શ પદાનુક્રમ હોઈ શકે. ઉદાહરણ તરીકે, 'Person' ક્લાસમાંથી 'Employee' ક્લાસ મેળવી શકાય અને પછી 'Employee' ક્લાસમાંથી 'Teacher' ક્લાસ મેળવી શકાય. અહીં કર્મચારી એ એક વ્યક્તિ છે અને શિક્ષક એ એક કર્મચારી છે. આ પ્રકારના ઈનહેરિટન્સ **મલ્ટિલેવલ ઈનહેરિટન્સ (multilevel inheritance)** કહેવાય છે. આકૃતિ 6.9માં મલ્ટિલેવલ ઈનહેરિટન્સનું ઉદાહરણ આપેલું છે.



આકૃતિ 6.9 : પદાનુક્રમિત ઉત્તરાધિકાર (Multilevel inheritance)

ક્લાસને એક કરતા વધુ પેરન્ટ ક્લાસનો ઉપયોગ કરીને પણ તારવી શકાય છે. ઉદાહરણ તરીકે, બાળક વારસામાં તેના માતા અને પિતા બંનેની લાક્ષણિકતાઓ ધરાવે છે; એરપ્લેન એક પ્રકારનું વાહન છે તથા ઊડી શકે તેવો ઑબ્જેક્ટ પણ છે. જ્યારે કોઈ ક્લાસ બે કે વધુ ક્લાસ પરથી તારવવામાં આવે, તો તેને **મલ્ટિપલ ઈનહેરિટન્સ (multipal inheritance)** કહેવામાં આવે છે.

કૉમ્પોઝિશન અને ઇનહેરિટન્સની તુલના (Composition Vs Inheritance)

ઇનહેરિટન્સમાં ક્લાસની ક્રિયાત્મકતા (functionality) ભેગી વાપરવા, પુનઃઉપયોગ કરવા અથવા વધારવા માટે અન્ય ક્લાસમાંથી વારસામાં મેળવે છે. અહીં સુપર ક્લાસ અને સબક્લાસ વચ્ચે 'એક પ્રકારનો' ('a kind of') સંબંધ હોય છે.

કૉમ્પોઝિશનમાં ક્લાસ અન્ય ક્લાસમાંથી વારસામાં બનતો નથી, પણ બીજા ક્લાસ વડે 'બનેલો' ('composed of') હોય છે. ક્લાસ અમુક ઍટ્રિબ્યૂટ ધરાવે છે, જેમાંના કેટલાક ઍટ્રિબ્યુટ બીજા પ્રકારના ક્લાસના હોય છે.

અહીં નોંધ કરો કે ક્લાસ-ડાયાગ્રામમાં બીજી અન્ય પ્રકારની રિલેશનશિપ (relationship) અને કન્સ્ટ્રેઇન્ટ (constraints) પણ દર્શાવવામાં આવે છે. આ બધા ખ્યાલોનો અભ્યાસ આ પુસ્તકના કાર્યક્ષેત્રની મર્યાદા બહાર છે.

સારાંશ

સૉફ્ટવેર સિસ્ટમનાં વિશ્લેષણ, ડિઝાઇન અને અમલીકરણમાં ઑબ્જેક્ટ આધારિત પદ્ધતિ ઘણી નોંધપાત્ર ભૂમિકા ભજવે છે. આ ફેરફારમાં **ઑબ્જેક્ટ (object)** કેન્દ્રબિંદુ છે કે જેમાં ડેટા અને ક્રિયાત્મકતા બંનેનો સમાવેશ થયેલો હોય છે. ક્લાસ ઍટ્રિબ્યૂટ (ડેટા) અને મેથડ (બિહેવ્યર અથવા ક્રિયાત્મકતા)ને ભેગી કરીને (encapsulates) માળખા તરીકે (template) બધા ઑબ્જેક્ટ વચ્ચે સહિયારો ઉપયોગ કરે છે. ઑબ્જેક્ટ એકબીજાથી તેના સ્ટેટ (state)થી જુદા પડે છે, એટલે કે તેના ઍટ્રિબ્યૂટની કિંમતોથી એક કરતાં વધારે ક્લાસ એકબીજા સાથે જોડાયેલો હોય છે. જ્યારે બે ક્લાસ વચ્ચે 'પૂર્ણ' અથવા 'આંશિક' સંબંધની સ્થિતિ હોય છે, ત્યારે તેને એગ્રિગેશન અથવા કૉમ્પોઝિશન કહેવાય છે. જ્યારે એક ક્લાસ અન્ય ક્લાસના ઑબ્જેક્ટ ધરાવે, ત્યારે તે ધરાવનાર ક્લાસ ઑનર ક્લાસ (owner class) અથવા હોલક્લાસ (whole class) અથવા એગ્રિગેટિંગ ક્લાસ (aggregating class) કહેવાય છે. ઓનર ક્લાસમાં સમાયેલ ક્લાસ સબ્જેક્ટ ક્લાસ (subject class) અથવા પાર્ટક્લાસ (part class) અથવા એગ્રિગેટેડ ક્લાસ (aggregated class) કહેવાય છે. એગ્રિગેશન બે ક્લાસ વચ્ચે ભિન્ન (non-exclusive) સંબંધની સ્થિતિ રજૂ કરે છે. કૉમ્પોઝિશન બે ક્લાસ વચ્ચે અભિન્ન સંબંધની સ્થિતિ જણાવે છે. જ્યારે બે ક્લાસ વચ્ચે 'એક પ્રકારનો' સંબંધ હોય, ત્યારે તેને ઇનહેરિટન્સની સ્થિતિ કહેવામાં આવે છે. સામાન્ય લાક્ષણિકતાઓને સુપરક્લાસમાં રાખવામાં આવે છે અને વિશિષ્ટ લાક્ષણિકતાઓને સબક્લાસમાં રાખવામાં આવે છે.

સ્વાધ્યાય

1. ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગ ભાષામાં ઉપલબ્ધ લાક્ષણિકતાઓની યાદી બનાવો.
2. ક્લાસ અને ઑબ્જેક્ટનો તફાવત જણાવો.
3. 'ઇનકૅપ્સ્યુલેશન' અને 'ડેટા-ઍબ્સ્ટ્રૅક્શન' વચ્ચેની ભિન્નતા જણાવો.
4. પોલિમૉર્ફિઝમનો અર્થ શો છે ? પોલિમૉર્ફિઝમ મેળવવા માટે વપરાતા બે પ્રકારનાં ઓવરલોડિંગનાં નામ જણાવો.
5. એગ્રિગેશન અને કૉમ્પોઝિશનનો ઉપયોગ સમજાવો.
6. ઇનહેરિટન્સ ક્યારે વાપરવું જોઈએ ? ઉદાહરણ આપો.
7. વિવિધ પ્રકારના ઇનહેરિટન્સ સમજાવો.
8. **નીચેના પ્રશ્નોનો સાચો જવાબ પસંદ કરો :**

   (1) ઑબ્જેક્ટ આધારિત પદ્ધતિમાં કઈ વસ્તુ કેન્દ્રબિંદુ ઉપર હોય છે ?

   (a) ડેટા (b) વિધેય
   (c) ઑબ્જેક્ટ (d) ઉપરના બધા વિકલ્પ

(2) જાવા માટે નીચેનામાંથી શું વધારે યોગ્ય છે ?

(a) પ્રક્રિયાગત પ્રોગ્રામિંગ ભાષા

(b) ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગ ભાષા

(c) ક્વેરી-લૅંગ્વેજ

(d) ઉપરના બધા વિકલ્પ

(3) નીચેનામાંથી શું ઑબ્જેક્ટને એકબીજાથી જુદા પાડે છે ?

(a) એટ્રિબ્યૂટ (b) સ્ટેટ

(c) બિહેવ્યર (d) ઉપરના બધા વિકલ્પ

(4) એકસમાન ઑબ્જેક્ટની સામાન્ય લાક્ષણિકતાઓને વ્યાખ્યાયિત કરવા માટે નીચેનામાંથી શું વપરાય છે ?

(a) ક્લાસ (b) ઑબ્જેક્ટ

(c) મેથડ (d) ઉપરના બધા વિકલ્પ

(5) નીચેનામાંથી કયું દૃશ્યતાનું ચિહ્ન નથી ?

(a) ~ (b) * (c) # (d) -

(6) ઈનકૅપ્સ્યુલેશન વડે નીચેનામાંથી શું પૂરું પાડવામાં આવે છે ?

(a) ડેટાની સુરક્ષિતતા (b) ડેટાનો સહિયારો ઉપયોગ

(c) ડેટા અને મેથડ છૂટા પાડવા (d) આપેલ બધા વિકલ્પ

(7) ડેટા-ઍબ્સ્ટ્રૅક્શન વડે શું શક્ય બનાવી શકાય છે ?

(a) ડેટા-સુરક્ષિતતા (b) ડેટા છુપાવવો

(c) ડેટાની ગણતરી કરવા બાબતની માહિતીનું અમલીકરણ છુપાવવું

(d) ઉપરના બધા વિકલ્પ

(8) નીચેનામાંથી ક્યો વિકલ્પ પોલિમૉર્ફિઝમ વડે પ્રાપ્ત થતો નથી ?

(a) મેથડ ઓવરલોડિંગ (b) ઑપરેટર ઓવરલોડિંગ

(c) ડેટા-હાઈડિંગ (d) આપેલ બધા વિકલ્પ

(9) એગ્રિગેશન ક્યા પ્રકારના સંબંધની સ્થિતિ જણાવે છે ?

(a) 'પૂર્ણ' સંબંધ ('is-a' relationship) (b) સમાન સંબંધ ('is-like' relationship)

(c) અંશતઃ સંબંધ (a-part-of' relationship) (d) ઉપરના બધા વિકલ્પ

(10) ઈનહેરિટન્સ ક્યા પ્રકારના સંબંધની સ્થિતિ જણાવે છે ?

(a) 'પૂર્ણ' સંબંધ ('is-a' relationship) (b) 'એક છે' સંબંધ ('has-a' relationship)

(c) અંશતઃ સંબંધ ('a-part-of' relationship) (d) ઉપરના બધા વિકલ્પ

(11) ક્લાસ-ડાયાગ્રામમાં કોમ્પોઝિશનને ક્યા ચિહ્ન વડે દર્શાવવામાં આવે છે ?

(a) ખાલી હીરાના ચિહ્ન વડે (b) ભરેલ હીરાના ચિહ્ન વડે

(c) ખાલી ત્રિકોણ ચિહ્ન વડે (d) ઉપરના બધા વિકલ્પ

જાવાની મૂળભૂત બાબતો 7

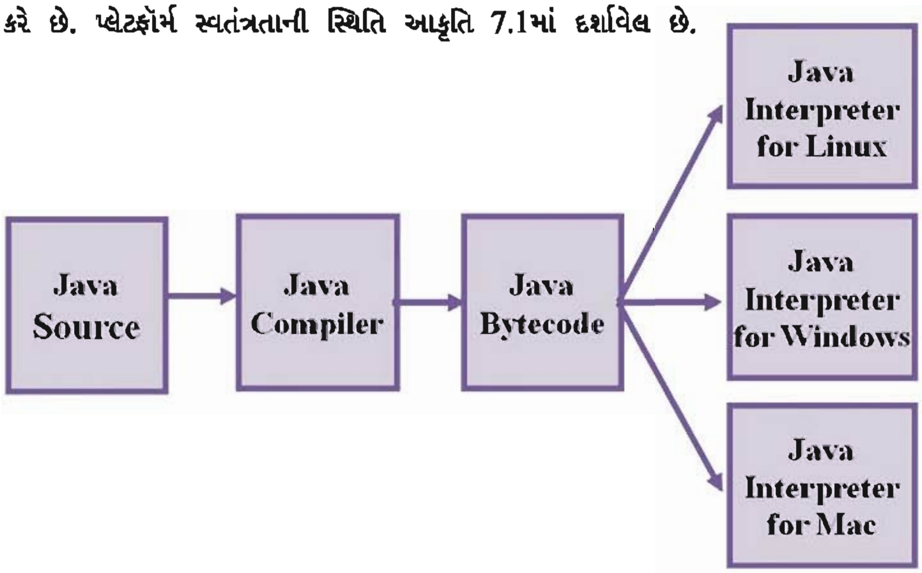
જાવા એક ઑબ્જેક્ટ આધારિત પ્રોગ્રામિંગ ભાષા છે જે ઉચ્ચ કક્ષાના યુનિક્સ વર્કસ્ટેશન માટે પ્રખ્યાત Sun Microsystems દ્વારા વિકસાવવામાં આવી છે. C++ પછી વિકાસ પામેલી જાવા ભાષા સૉર્સ પ્રોગ્રામ અને દ્વિઅંકી બંને સ્તરે વિવિધ પ્લેટફૉર્મ (જુદાં-જુદાં હાર્ડવેર) અને ઑપરેટિંગ સિસ્ટમ ઉપર નાની, સરળ અને સુવાહ્ય બની રહે તેવી બનાવેલી છે. આ પ્રકરણમાં આપણે જાવા સાથે પ્રોગ્રામિંગ શીખવાની શરૂઆત કરીશું. આપણે જાવામાં main() મેથડ વડે કરી શકાતાં પાયાનાં કાર્યો અને સાદાં જાવા વિધાનો બાબત ચર્ચા કરીશું. આપણે અગાઉ C ભાષા ભણી ગયા હોવાથી જાવાના પાયારૂપ બંધારણો સમજવા માટે ખૂબ જ સરળ બનશે. મોટાભાગનાં જાવાવિધાનોની વાક્યરચના (syntax) C ભાષા જેવી જ છે.

જાવાનો પરિચય (Introduction to Java)

જાવા ભાષા 1991માં Sun Microsystems દ્વારા વિકસાવવામાં આવી છે. જાવા નાની, ઝડપી, કાર્યક્ષમ અને હાર્ડવેર ઉપકરણોની વિશાળ શ્રેણી માટે સુવાહ્ય છે. તેને વર્લ્ડવાઇડ વેબ (World Wide Web) દ્વારા અમલમાં મૂકી શકાય તેવા (executable) પ્રોગ્રામ્સના વિતરણ માટેની એક આદર્શ ભાષા અને વિવિધ પ્લેટફૉર્મ ઉપર સરળતાથી વાપરી શકાય અને સુવાહ્ય હોય તેવા પ્રોગ્રામ્સ બનાવવા માટેની સામાન્ય ઉદ્દેશવાળી પ્રોગ્રામિંગ ભાષા (General Purpose Programming Language) પણ ગણવામાં આવે છે.

જાવા એક ઑબ્જેક્ટ આધારિત ભાષા છે અને અહીં તે C ભાષાથી અલગ પડે છે. જાવાનો ઉપયોગ કરીને આપણે ઑબ્જેક્ટ આધારિત પદ્ધતિ અને લવચિક, વિભાગીય (modular) અને ફરી વાપરી શકાય તેવા પ્રોગ્રામના કોડ લખવાના સામર્થ્યનો પૂરેપૂરો લાભ લઈ શકીએ છીએ. તે વિવિધ પ્રકારના ક્લાસની લાઇબ્રેરી ધરાવે છે, જે મૂળભૂત ડેટાપ્રકારો, સિસ્ટમની નિવેશ અને નિર્ગમ ઉપયોગી ક્ષમતા તેમજ અન્ય યુટિલિટી કાર્યો માટેની સગવડ પૂરી પાડે છે. આ મૂળભૂત ક્લાસનો સમૂહ જાવા ડેવલપમેન્ટ કિટ (JDK)નો ભાગ છે. JDK માં નેટવર્કિંગ માટેના, સામાન્ય ઇન્ટરનેટ પ્રોટોકોલના અને વપરાશકર્તાના સેતુ ટૂલકિટનાં કાર્યો માટેના અનેક ક્લાસ હોય છે. લાઇબ્રેરીના આ બધા ક્લાસ જાવામાં લખાયેલા હોવાથી બધા જાવા વિનિયોગની જેમ સમગ્ર પ્લેટફૉર્મ ઉપર તે સુવાહ્ય (portable) હોય છે.

જાવા સૉર્સપ્રોગ્રામ અને દ્વિઅંકી બંને સ્તરે પ્લેટફૉર્મ સ્વતંત્ર છે. પ્લેટફૉર્મ સ્વતંત્રતા એ પ્રોગ્રામની એક કમ્પ્યૂટર-સિસ્ટમથી બીજી ઉપર સરળતાથી લઈ જવાની ક્ષમતા છે. સૉર્સપ્રોગ્રામના સ્તરે જાવાના પ્રાથમિક ડેટા પ્રકારોનું કદ બધા વિકાસ પ્લેટફૉર્મ ઉપર અચળ છે. અને દ્વિઅંકી સ્તરે બાઇટકોડ ઇન્ટરપ્રિન્ટર (bytecode interpreter)ને કારણે પ્લેટફૉર્મ સ્વતંત્રતા શક્ય છે. જાવાના રચયિતા કમ્પાઇલેશન અને ઇન્ટરપ્રિટેશન (compilation and interpretation)ના મિશ્રણનો ઉપયોગ કરવાનું પસંદ કરે છે. પ્લેટફૉર્મ સ્વતંત્રતાની સ્થિતિ આકૃતિ 7.1માં દર્શાવેલ છે.



આકૃતિ 7.1 : જાવા : પ્લેટફૉર્મ સ્વતંત્ર

જાવામાં લખેલા પ્રોગ્રામનું એવા કમ્પ્યૂટરની મશીનની ભાષામાં કંપાઈલેશન (compilation) કરવામાં આવે છે, જે હકીકતમાં અસ્તિત્વમાં નથી. આ પ્રકારનાં "virtual" કમ્પ્યૂટર જાવા વર્ચ્યુઅલ મશીન (Java Virtual Machine - JVM) તરીકે ઓળખાય છે. જાવા વર્ચ્યુઅલ મશીનની યાંત્રિક ભાષાને જાવા બાઈટકોડ (bytecode) કહેવામાં આવે છે. દરેક પ્રકારના કમ્પ્યૂટર માટે અલગ-અલગ જાવા બાઈટકોડ ઈન્ટરપ્રિટરની જરૂર રહે છે. જાવા બાયનરી ફાઈલ્સ વાસ્તવમાં બાઈટકોડના સ્વરૂપમાં હોય છે જે કોઈ પણ એક પ્રોસેસર અથવા કોઈ પણ ઑપરેટિંગ સિસ્ટમ માટે ચોક્કસ નથી. બાઈટકોડ વાપરવાનો ગેરલાભ તેના અમલની ઓછી ઝડપ છે. એ પ્રકારનાં ટૂલ્સ ઉપલબ્ધ છે, જે જાવા બાઈટકોડનું નેટિવ કોડ (native code)માં રૂપાંતર કરે છે. નેટિવ કોડ અમલ કરવામાં ઝડપી છે પણ પછી તે મશીન સ્વતંત્ર રહેતાં નથી.

સાદા જાવા-વિનિયોગનું નિર્માણ (Creating Simple Java Application)

જાવાપ્રોગ્રામિંગ ભાષાની મૂળભૂત બાબતો શીખતાં પહેલાં, ચાલો આપણે એક સરળ પ્રોગ્રામ દ્વારા જાવાનું નિરીક્ષણ કરવાનું શરૂ કરીએ કે જે ફોનકોલના ખર્ચની ગણતરી કરે અને પહેલાંથી ચૂકવેલી રકમ (pre-paid balance)માં સુધારો કરે. અહીં આપણે કઈ રીતે જાવાપ્રોગ્રામ બનાવવા, કંપાઈલ કરવા અને તેનો અમલ કરવો તે બાબત શીખીશું.

જાવાપ્રોગ્રામ અનેક ક્લાસનો બનેલો હોય છે. તેમાં ઓછામાં ઓછો એક ક્લાસ હોવો જોઈએ અને main મેથડ હોવી જ જોઈએ. C ભાષાના પ્રોગ્રામરો ક્લાસને struct અને typedefનો ઉપયોગ કરીને નવો સંયુક્ત ડેટાપ્રકાર (composite data type) બનાવવા જેવો વિચાર કરી શકે. જોકે ક્લાસ એ માત્ર માહિતીનો સંચય કરતા ઘણું વધારે આપી શકે છે. અહીં એ નોંધ કરો કે જાવામાં typedef ઉપલબ્ધ નથી.

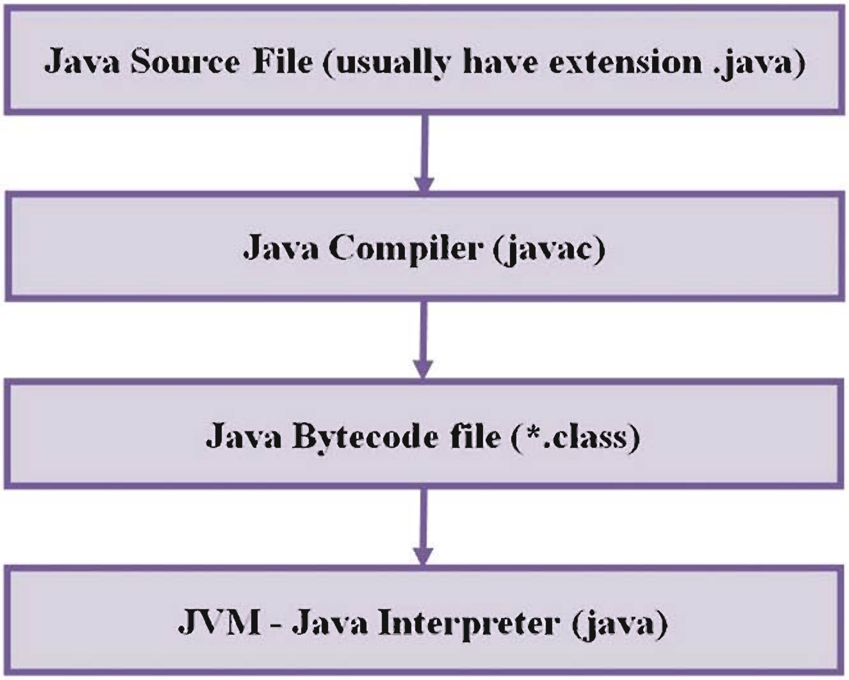
આપણે જાવા-વિનિયોગ કે જે ફોનકૉલના ખર્ચની ગણતરી કરી સિલક રકમમાં સુધારો કરે છે તેને 'CallCost' નામ આપીએ. આ 'CallCost' વિનિયોગ બનાવીને તેનો અમલ કરવા માટે આપણે નીચે જણાવેલાં પગલાં પ્રમાણે કાર્ય કરવું જરૂરી છે :

1. કોઈ પણ સાદા ASCII ટૅક્સ્ટ-એડિટર વડે જાવા સૉર્સફાઈલ બનાવો.
   - કોઈ પણ ટૅક્સ્ટ-એડિટર પસંદ કરી કોડલિસ્ટિંગ 7.1માં આપેલો પ્રોગ્રામ ટાઈપ કરો.
   - આ સૉર્સફાઈલનો 'CallCost.java' નામથી સંગ્રહ કરો. રૂઢિપરંપરાથી, જે નામથી ક્લાસ વ્યાખ્યાયિત કરેલો હોય, તે જ નામ અને '.java' અનુલંબન સાથેનું નામ જાવા સૉર્સફાઈલને આપવામાં આવે છે. અહીં એ નોંધો કે, ક્લાસનું નામ અને ફાઈલનું નામ કેસ સેન્સિટિવ હોય છે. માટે, જો ક્લાસનું નામ CallCost હોય તો, ફાઈલનું નામ પણ CallCost.java હોવું જરૂરી છે. (SciTE એડિટર માટે)
2. જાવા-કંપાઈલરનો ઉપયોગ કરીને સૉર્સફાઈલને કંપાઈલ (compile) કરો.
   - આ કાર્ય માટે javac પછી સૉર્સફાઈલનું નામ એટલે કે javac CallCost.java ટાઈપ કરીને જાવાપ્રોગ્રામ કંપાઈલ કરો.
   - જો કંપાઈલર કંઈ પણ ભૂલ દર્શાવે, તો પાછા જાઓ અને ખાતરી કરો કે કોડલિસ્ટિંગ 7.1માં જણાવ્યા પ્રમાણે તમે પ્રોગ્રામ બરાબર ટાઈપ કર્યો છે કે કેમ.
   - પ્રોગ્રામ કંઈ પણ ભૂલ વિના કંપાઈલ થાય, ત્યારે તમારી સૉર્સફાઈલની ડિરેક્ટરીમાં કંપાઈલર .class અનુલંબન સાથેની ફાઈલ બનાવશે. ખાતરી કરો કે કંપાઈલર વડે 'CallCost.class' ફાઈલ બનેલી છે. આ આપણી જાવા બાઈટકોડ ફાઈલ છે, જેનો અમલ કરવામાં આવશે.
3. જાવા-ઈન્ટરપ્રિટર વડે વિનિયોગનો અમલ કરો.
   - JDKમાં ફક્ત java આપીને જાવા ઈન્ટરપ્રિટર વાપરી શકાય છે. java CallCost ટાઈપ કરો.
   - પ્રોગ્રામનો અમલ કરવા માટે આપણે સૉર્સકોડની જરૂર નથી, પણ ફક્ત કંપાઈલ કરેલી ક્લાસફાઈલની જરૂર છે.
   - જાવા ઈન્ટરપ્રિટર બાઈટકોડ CallCost.classનો ઉપયોગ કરીને તેનો અમલ કરે છે.

```
/**
* This class implements a simple program that
* will compute the cost of phone call and update balance
*/
public class CallCost
{
public static void main(String[] args)
{
/* declare variables */
double balance;          // balance amount in rupees
double rate;             // call rate; rupees per second
double duration;         // call duration in seconds
double cost;             // cost of last call

/* computations. */
balance = 170;
rate = 1.02;
duration = 37;
cost = duration * rate;                  // compute the cost
balance = balance - cost;     // update balance amount

/* display results */
System.out.print("Call Duration: ");
System.out.print(duration);
System.out.println(" Seconds");
System.out.println("Balance: " + balance + "Rupees ");

} // end of main()
} // end of class CallCost
```

કોડલિસ્ટિંગ 7.1 : નમૂનારૂપ જાવાપ્રોગ્રામ

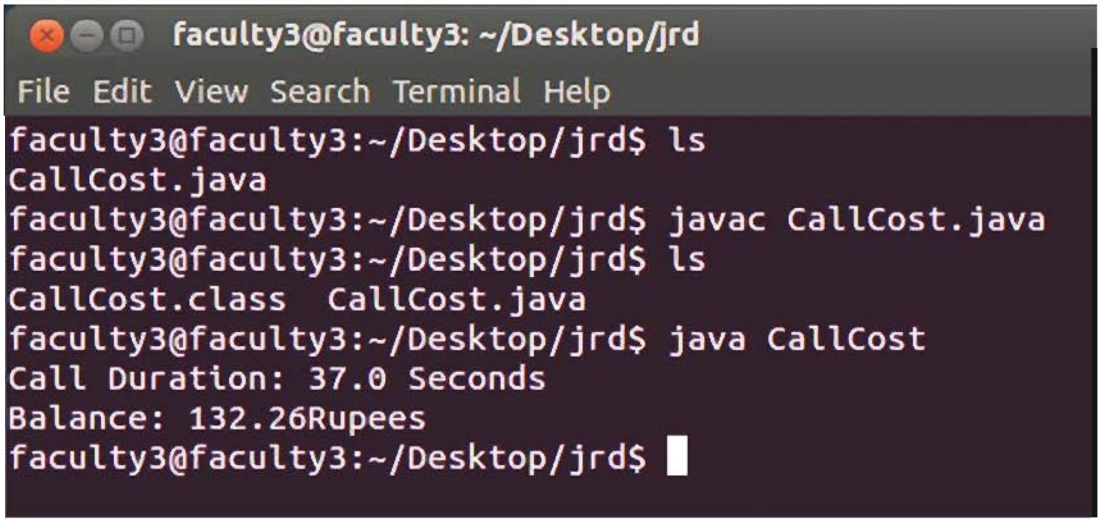
જાવાપ્રોગ્રામને કંપાઇલ કરવાની અને અમલ કરવાની પ્રક્રિયા આકૃતિ 7.2માં દર્શાવી છે.



આકૃતિ 7.2 : કંપાઇલેશન પ્રક્રિયા

જાવા સૉર્સપ્રોગ્રામ ફાઇલનું અનુલંબન '.java' હોવું જરૂરી છે. અને તે ફાઇલનું નામ જ્યારે ક્લાસ public હોય ત્યારે ક્લાસનું નામ હોવું જોઈએ. અહીં નોંધ કરો કે નામ કેસ-સૅન્સિટિવ (case sensitive) હોય છે. 'javac' નામનું કંપાઇલર સૉર્સફાઇલને કંપાઇલ કરી બાઇટકોડ ફાઇલ બનાવે છે. main મેથડ ધરાવતા ક્લાસનું નામ જ બાઇટકોડ ફાઇલનું હોય છે અને તેનું અનુલંબન '.class' હોય છે. 'java' ઇન્ટરપ્રિટર બાઇટકોડનું અર્થઘટન કરે છે અને તેનો અમલ કરે છે.

ટર્મિનલના ઉપયોગ દ્વારા linuxમાં પ્રોગ્રામને કંપાઇલ કરી તેનો અમલ કરી શકાય છે તે આકૃતિ 7.3માં દર્શાવ્યું છે.



આકૃતિ 7.3 : linux ટર્મિનલમાં જાવાપ્રોગ્રામનું કંપાઇલેશન અને અમલ

કોડલિસ્ટિંગ 7.1માં બે મુખ્ય ભાગ પર ધ્યાન આપો :

1. પ્રોગ્રામ ક્લાસની વ્યાખ્યામાં અહીં, 'CallCost' નામના ક્લાસમાં સમાયેલો છે.

2. પ્રોગ્રામની બૉડી (body) main() નામના રૂટિન (routine)માં સમાયેલી હોય છે. જાવાવિનિયોગમાં જ્યારે પ્રોગ્રામનો અમલ કરવામાં આવે છે, ત્યારે main() નામનું રૂટિન સૌપ્રથમ અમલમાં આવે છે.

કોડલિસ્ટિંગ 7.1ની સમજૂતી (Explanation of Code Listing 7.1)

પ્રોગ્રામમાં // પછી લખાયેલી શાબ્દિક માહિતી (text) અને /* અને */ વચ્ચે બંધ હોય તેને કૉમેન્ટ (comment) કહેવામાં આવે છે. આપણે જાણીએ છીએ કે કૉમેન્ટ કંપાઇલ કે ઇન્ટરપ્રિટ થતી નથી.

- ડેટાપ્રકાર (datatype) પછી ચલનું નામ (variable name) જણાવી ચલને ઘોષિત કરવામાં આવે છે.
- ગણતરીના ભાગમાં એસાઇન્મેન્ટ વિધાનો સાથેની પદાવલીઓ હોય છે.
- અહીં, પ્રોગ્રામની માહિતી ઉપયોગકર્તાને પ્રદર્શિત કરવા માટે અનેક સબરૂટિન (જાવામાં ફંક્શન અથવા મેથડ પણ કહેવામાં આવે છે.) કૉલ (subroutine call)વિધાનો વાપર્યાં છે.
  - પરિણામ પ્રદર્શિત કરવા માટે વાપરેલી મેથડનાં નામ : System.out.print અને System.out.println. આ બંને મેથડ જે કિંમત પ્રદર્શિત કરવાની છે, તે ચલ (argument) તરીકે લે છે.
  - System.out.println મેથડ જે માહિતી પ્રદર્શીત કરે છે, તેના પછી એક નવી લીટી (linefeed) ઉમેરે છે, જ્યારે System.out.print ફક્ત માહિતી જ પ્રદર્શિત કરે છે.
  - ફોનકૉલની અવધિ પ્રોગ્રામમાં ત્રણ Call વિધાન વડે પ્રદર્શિત થાય છે. પહેલાં Callથી શાબ્દિક માહિતી 'Call Duration:' પ્રદર્શિત થાય છે, બીજા Callથી 'duration' ચલની કિંમત પ્રદર્શિત થાય છે અને ત્રીજા Callથી 'Seconds' લખાણ પ્રદર્શિત થાય છે અને તે પછી કર્સર આગળની લીટીમાં લાવે છે. અહીં નોંધ કરો કે શાબ્દિક લખાણ (string literal) બેવડા અવતરણ (" ") વચ્ચે ઉમેરવામાં આવે છે.
  - એક Callથી balance પ્રદર્શિત થાય છે. પ્રાચલ તરીકે જાવામાં + પ્રક્રિયકનો ઉપયોગ ધ્યાન ઉપર લો. પ્રાચલ તરીકે આપેલી પદાવલીનું પ્રથમ મૂલ્યાંકન કરવામાં આવે છે અને પછી તે પ્રદર્શિત થાય છે.

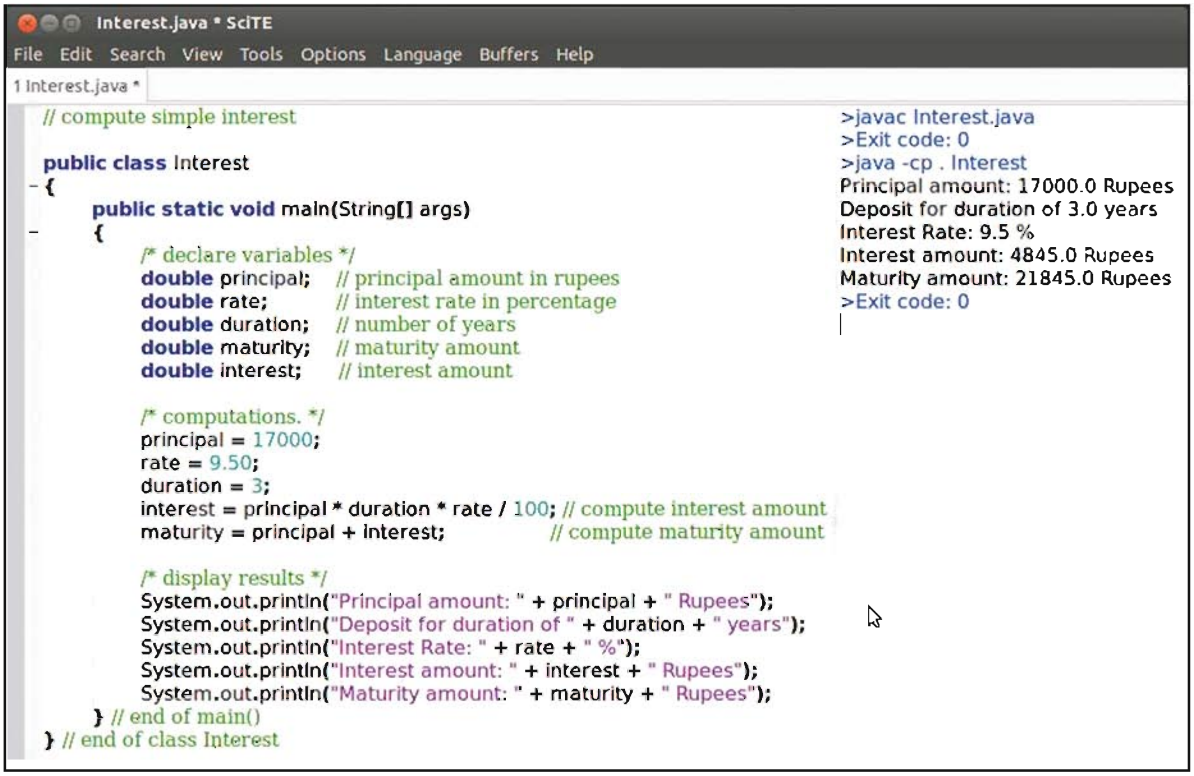
જ્યારે આપણે પ્રોગ્રામનો અમલ કરીએ છીએ, ત્યારે જાવા ઇન્ટરપ્રિટર પ્રથમ main() મેથડ ઉપર જાય છે અને તેમાં રહેલાં વિધાનોનો અમલ કરે છે. આ વિધાનો જ્યારે પ્રોગ્રામનો અમલ થાય ત્યારે કમ્પ્યૂટરે ચોક્કસ શું કરવાનું છે તે જણાવે છે. main() રૂટિન એ જ ક્લાસમાં અથવા અન્ય ક્લાસમાં પણ વ્યાખ્યાયિત સબરૂટિનને બોલાવી શકે છે, પણ ફક્ત main() રૂટિન જ કેવી રીતે અને ક્યા ક્રમમાં અન્ય સબરૂટિન વાપરવા તે નક્કી કરે છે.

main()ની પ્રથમ લીટીના શબ્દ "public"નો અર્થ એ છે કે આ પ્રોગ્રામની બહારથી આ રૂટિન બોલાવી શકાય છે. આ અનિવાર્ય છે, કારણકે જાવા ઇન્ટરપ્રિટર વડે main() રૂટિન બોલાવવામાં આવે છે કે જે પ્રોગ્રામની બહાર છે. રૂટિનની પ્રથમ લીટીની શેષ બાબત આ ક્ષણે સમજાવવી મુશ્કેલ છે; આથી હાલ પૂરતું એવું વિચારો કે તે વાક્યરચના (syntax)નો જરૂરી ભાગ છે.

SciTEનો ઉપયોગ (Using SciTE)

હવે આપણે SciTE એડિટર (editor)નો ઉપયોગ કરીને એક વધુ જાવા-વિનિયોગ બનાવીએ. અહીં આપણા વિનિયોગમાં સાદા વ્યાજની ગણતરી કરીને પરિણામ પ્રદર્શિત કરશે. આકૃતિ 7.4માં પ્રોગ્રામનું કોડલિસ્ટિંગ અને તેનો આઉટપુટ દર્શાવ્યો છે. નીચેનાં પગલાંને અનુસરો :

- SciTE વિનિયોગ ચાલુ કરો. હવે પસંદ કરો : **File → New.**
- આકૃતિ 7.4માં આપેલા કોડલિસ્ટિંગ પ્રમાણે જાવાપ્રોગ્રામ ટાઇપ કરો અને Interest.java નામ સાથે આ સૉર્સફાઇલનો સંગ્રહ કરો. ફાઇલનો સંગ્રહ કરવા માટે **File → Save** આદેશ આપો.
- **Tools → Compile** આદેશ આપીને સૉર્સપ્રોગ્રામને કંપાઇલ કરો.
- જો પ્રોગ્રામ કોઈ પણ ભૂલ (error) વિના કંપાઇલ થઈ જાય, તો **Tools → Go** આદેશ આપીને તેનો અમલ કરો.



આકૃતિ 7.4 : SciTEનો ઉપયોગ કરીને જાવાપ્રોગ્રામનો અમલ

જાવાપ્રોગ્રામનું બંધારણ (Structure of a Java Program)

પ્રોગ્રામિંગ ભાષાઓ સામાન્ય ભાષાઓથી અલગ એ રીતે છે કે તે સંપૂર્ણપણે સ્પષ્ટ (unambiguous) અને પ્રોગ્રામમાં કઈ બાબતને મંજૂરી છે અને કઈ બાબતને નથી, તે વિશે ખૂબ જ દૃઢ છે. એ નિયમો કે જે નક્કી કરે કે શું માન્ય છે, તેને ભાષાની વાક્યરચના (syntax) કહેવામાં આવે છે.

વાક્યરચનાના નિયમો (syntax rules) ભાષાના મૂળભૂત શબ્દભંડોળનો ઉલ્લેખ કરે છે અને ચલ, પદાવલીઓ, વિધાનો, બ્રાન્ચ (branches), લૂપ (loops) અને મેથડ (methods)નો ઉપયોગ કરીને કઈ રીતે પ્રોગ્રામની રચના કરી શકાય તે જણાવે છે. વાક્યરચનાની રીતે (syntactically) સાચો પ્રોગ્રામ એ છે કે જે સફળતાપૂર્વક કંપાઇલ કે ઇન્ટરપ્રિટ થઈ શકે.

જાવાપ્રોગ્રામનું માળખું આકૃતિ 7.5માં દર્શાવ્યું છે. અહીં, < અને > કોણીય કૌંસ એક પ્લેસહોલ્ડર (place holder) તરીકે વપરાય છે, જેમાં ખરેખર પ્રોગ્રામ લખતા સમયે કંઈક વાસ્તવિક ટાઇપ કરવાનું હોય છે. જાવામાં મેથડ (ફંક્શન)ની વ્યાખ્યા ફંક્શન હેડર તથા { અને } કૌંસ વચ્ચે વિધાનોની શ્રેણી હોય છે.

જાવા ઑબ્જેક્ટ આધારિત ભાષા હોવાથી અહીં તમામ ક્લાસના ભાગ તરીકે વ્યાખ્યાયિત કરવામાં આવે છે. આ રીતે, મેથડ સ્વતંત્ર રીતે હયાતી ધરાવતી નથી. તે ક્લાસનો એક ભાગ હોઈ શકે છે.

```
public class <class-name>
{
        <optional-variable-declarations-and-methods>
        public static void main(String[] args)
        {
                <statements >
        }
        <optional-variable-declarations-and-methods>
}
```

આકૃતિ 7.5 : જાવાપ્રોગ્રામનું માળખું

- પ્રથમ લીટીમાં <class-name> એ ક્લાસનું નામ છે, જેમાં main મેથડ હોય છે.

  જો ક્લાસનું નામ CallCost હોય તો રૂઢિપરંપરાથી જાવા સૉર્સફાઇલનો સંગ્રહ CallCost.java નામથી કરવો જોઈએ. જ્યારે આ ફાઇલને કંપાઇલ કરવામાં આવે છે, ત્યારે CallCost.class નામની બીજી ફાઇલ બને છે. ક્લાસના નામનો ઉપયોગ કરીને ક્લાસફાઇલ (class file)ને નામ આપવામાં આવે છે. આ CallCost.class નામની ક્લાસ ફાઇલ પ્રોગ્રામનું જાવા બાઇટકોડમાં થયેલું રૂપાંતર (અનુવાદ) ધરાવે છે જેનો જાવા-ઇન્ટરપ્રિટર દ્વારા અમલ કરી શકાય છે.

- main() મેથડ પહેલાં અને પછી ચલ (variable) અને મેથડ (method) ઘોષિત કરવા વૈકલ્પિક છે.

- દરેક પ્રોગ્રામમાં એક ક્લાસ હોવો જ જોઈએ જે પબ્લિક મેથડ main() ધરાવે.

- જાવા કોઈ ચોક્કસ બંધારણની ભાષા નથી, એટલે કે તે ફ્રી-ફૉર્મેટ (free-format) ભાષા છે. કોડલિસ્ટિંગ 7.1માં આપેલું પ્રોગ્રામનું માળખું (જેમ કે ખાલી લાઇન અને ઇન્ડેન્ટેશનનો ઉપયોગ) ભાષાની વાક્યરચના અથવા સિમેન્ટિક્સનો ભાગ નથી. કમ્પ્યૂટર પ્રોગ્રામના માળખાને ધ્યાનમાં લેતું નથી. આપણે આખો પ્રોગ્રામ એક જ લીટીમાં પણ લખી શકીએ. જોકે પ્રોગ્રામનું માળખું માનવીય વાચકો માટે મહત્ત્વનું છે.

- પ્રોગ્રામ main() સાથે બીજી મેથડ તેમજ અન્ય ચલ ધરાવી શકે છે. આપણે આ બાબત હવે પછી શીખીશું.

હવે, આપણે સાદાં જાવાવિધાનો વિશે શીખીએ જેના દ્વારા આપણે main() મેથડમાં સામાન્ય કાર્યો કરી શકીએ. આપણે વાક્યરચના (syntax) સાથે નીચે જણાવેલી બાબતો વિશે શીખીએ :

- ડેટાપ્રકાર (Data types)
- ચલ (Variables)
- લિટરલ (Literals)
- કૉમેન્ટ (Comments)
- જાવાવિધાનો અને પદાવલીઓ (Java statements and expressions)
- અંકગણિતીય પ્રક્રિયકો (Arithmetic operators)
- સરખામણી (Comparisons)
- તાર્કિક પ્રક્રિયકો (Logical operators)

ડેટાપ્રકારો (Data-types)

ડેટાપ્રકાર મેમરીનું જરૂરી કદ, મેમરીમાં રહેલી કિંમતનો પ્રકાર, કિંમતનો ક્ષેત્રવિસ્તાર (રેન્જ) અને તેની ઉપર શક્ય ક્રિયાઓનો પ્રકાર નક્કી કરે છે. જાવા આઠ પ્રકારના પ્રાથમિક (પ્રિમિટિવ-primitive) ડેટાપ્રકાર પૂરા પાડે છે, જે સામાન્ય પ્રકારના પૂર્ણાંકો, અપૂર્ણાંકો, કૅરેક્ટર (character-અક્ષર) અને બુલિયન કિંમતો (true અથવા false)ને સંભાળે છે.

પ્રાથમિક ડેટાપ્રકારોમાં **byte, short, int, long, float, double, char અને boolean**નો સમાવેશ થાય છે. પ્રથમ ચાર પ્રકારોમાં પૂર્ણાંક સંખ્યા (જેમકે 17, -38477 અને 0) હોય છે, તે પછીના બે પ્રકારોમાં અપૂર્ણાંક સંખ્યા (જેમકે 5.8, -129.35)નો સમાવેશ કરી શકાય છે, char પ્રકારમાં યુનિકોડ અક્ષરસમૂહ (Unicode character set)માંથી એક અક્ષર હોય છે અને બુલિયન (boolean) પ્રકાર બે તાર્કિક કિંમતો true અથવા falseમાંથી એક કિંમત હોય છે.

આ ડેટાપ્રકારોને પ્રાથમિક ડેટાપ્રકાર કહેવામાં આવે છે, કારણકે તે સિસ્ટમમાં જ સમાવિષ્ટ હોય છે. અહીં નોંધ કરશો કે આ ડેટાપ્રકાર મશીન સ્વતંત્ર છે, એટલે કે બધાં મશીન ઉપર તમામ જાવાપ્રોગ્રામ્સમાં તેનાં કદ અને લાક્ષણિકતાઓ સુસંગત (consistent) હોવાથી આપણે તેના ઉપર ભરોસો રાખી શકીએ છીએ. કોષ્ટક 7.1માં ડેટાપ્રકારોની વિગત આપેલી છે.

ડેટાપ્રકાર	સંગ્રહની જગ્યા	કિંમતનો પ્રકાર	કિંમતનો ક્ષેત્ર વિસ્તાર (range)	પૂર્વનિર્ધારિત કિંમત
byte	1 byte	Integer	-128 and 127	0
short	2 bytes	Integer	-32768 to 32767	0
int	4 bytes	Integer	-2147483648 to 2147483647	0
long	8 bytes	Integer	-9223372036854775808 to 9223372036854775807	0
float	4 bytes	Real	10 ±38 with about 7 significant digits	0
double	8 bytes	Real	10 ±308 with about 15 significant digits	0
char	2 bytes	Character	16-bit Unicode character	0
boolean	1 byte	Boolean	true, false	false

કોષ્ટક 7.1 : જાવામાં ડેટાના પ્રકારો

b બીટની યથાર્થતા (precision) સાથેની પૂર્ણાંક સંખ્યામાં ($-2^{b-1}-1$ થી 2^{b-1})ની રેન્જમાં ચિહ્નિત કિંમતોનો સંગ્રહ કરી શકાય છે. જ્યારે તેની આગળ unsigned ચાવીરૂપ શબ્દ (keyword) હોય, ત્યારે તેની કિંમત (0 થી $2^{b}-1$)ની રેન્જમાં હોઈ શકે. જાવામાં અપૂર્ણાંક સંખ્યા IEEE 754 (અપૂર્ણાંક સંખ્યાઓ અને અંકગણિતીય પ્રક્રિયાને વ્યાખ્યાયિત કરવા માટેના આંતરરાષ્ટ્રીય ધોરણો) સુસંગત છે. જાવા યુનિકોડ અક્ષરસમૂહનો ઉપયોગ કરે છે. char ડેટાપ્રકારની 16 બીટની પ્રીસિઝન (precision) અને અચિહ્નિત (unsigned) હોય છે. આ કારણે જાવા અનેક વિવિધ ભાષાઓના અલગ-અલગ મૂળાક્ષરોના હજારો અક્ષરોનો ઉપયોગ કરી શકે છે. boolean ડેટાપ્રકાર એ સંખ્યા નથી અને તેને સંખ્યા તરીકે ગણી પણ ન શકાય.

ચલ (Variable)

જો પ્રોગ્રામના અમલ દરમિયાન તમે કમ્પ્યૂટર દ્વારા કંઈ પણ યાદ રાખવા માંગો છો તો તે કમ્પ્યૂટરની મેમરીમાં સંગૃહીત કરવાની જરૂર રહે છે. મેમરીમાં સંગ્રહ કરેલા ડેટા ઉપર પ્રોગ્રામ ગણતરી કરે છે. યાંત્રિક ભાષામાં (મશીન-લૅંગ્વેજ) ડેટાનો નિર્દેશ જે મેમરી જગ્યાએ સંગ્રહ કરેલો છે, તેના આંકડાકીય સ્થાનાંક વડે કરવામાં આવે છે. જાવા જેવી ઉચ્ચ-સ્તરીય ભાષામાં ડેટાનો નિર્દેશ કરવા માટે મેમરીની જગ્યાનાં આંકડાકીય સ્થાનાંકના બદલે નામનો ઉપયોગ કરવામાં આવે છે. પ્રોગ્રામરે ફક્ત નામ જ યાદ રાખવું પડે છે. મેમરીમાં સંગ્રહ કરેલા ડેટાનો નિર્દેશ કરવા માટે વાપરેલા નામને **ચલ (variable)** કહેવામાં આવે છે.

પ્રોગ્રામના અમલ દરમિયાન જુદા-જુદા સમયે ચલમાં અલગ-અલગ ડેટાકિંમત હોઈ શકે પણ તે હંમેશાં એક જ મેમરી-સ્થાનાંકનો નિર્દેશ કરે છે. જાવાપ્રોગ્રામમાં જો પહેલાં ચલ ઘોષિત કરેલો હોય (declared) તો જ વાપરી શકાય છે.