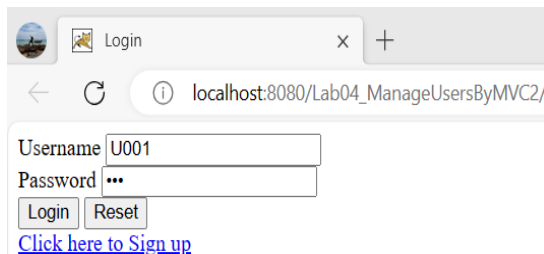


# Building an Java Web Application

## Using MVC2 & JSP

### Requirements

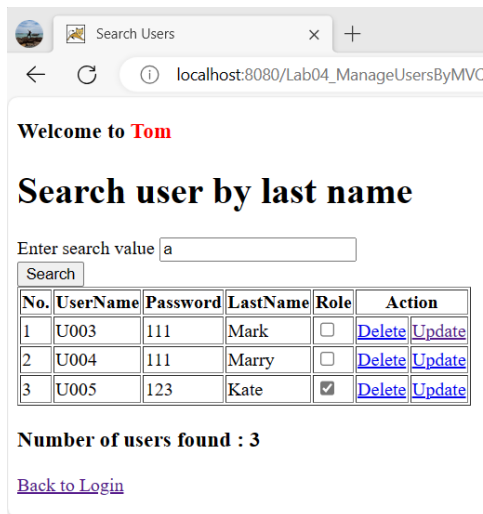
Create an Java web application (using HTML, Servlets , JSP and JDBC) by MVC2 Model to perform functions as manage users.



Username

Password

[Click here to Sign up](#)



Welcome to **Tom**

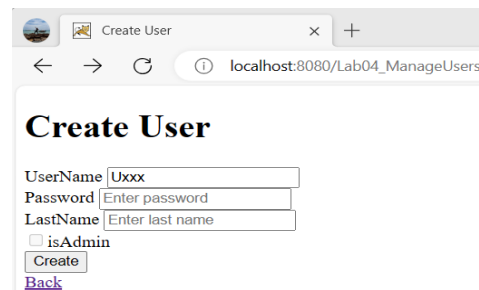
### Search user by last name

Enter search value

No.	UserName	Password	LastName	Role	Action
1	U003	111	Mark	<input type="checkbox"/>	<a href="#">Delete</a> <a href="#">Update</a>
2	U004	111	Marry	<input type="checkbox"/>	<a href="#">Delete</a> <a href="#">Update</a>
3	U005	123	Kate	<input checked="" type="checkbox"/>	<a href="#">Delete</a> <a href="#">Update</a>

Number of users found : 3

[Back to Login](#)



### Create User

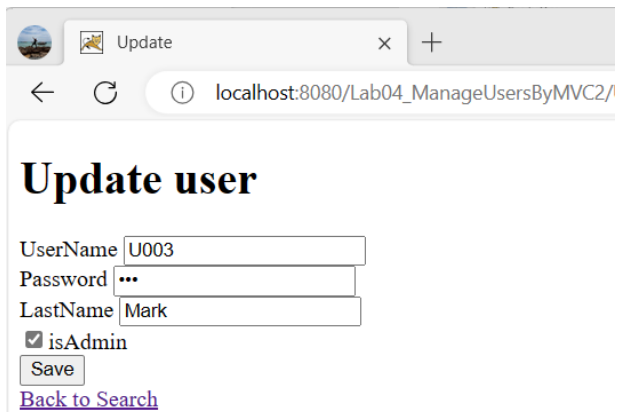
UserName

Password

LastName

☐ isAdmin

[Back](#)



### Update user

UserName

Password

LastName

☒ isAdmin

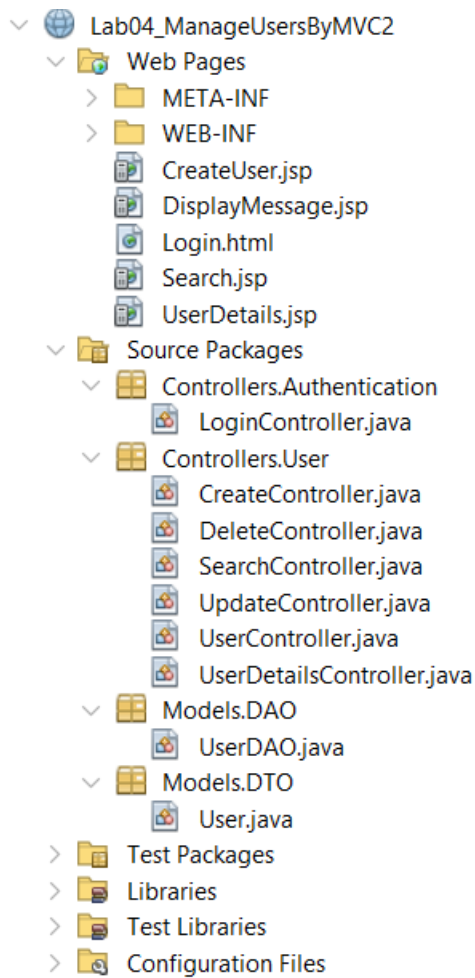
[Back to Search](#)

### Lab Objectives

In this lab, you will be:

- Create Java web project by MVC2 Model
- Create the web pages by HTML, and JSP (Views)
- Apply passing data from HTML to Servlet by GET and POST method
- Apply RequestDispatcher to pass data between Servlets (Controllers)
- Performs CRUD functions by SqlJDBC driver and SQL Server (Models)
- Run the project and test the functions of the project

**Step 01.** Create a Java Web App named **Lab04\_ManageUsersByMVC2** is structured as follows:



-Open SQL Server Management Studio and create a database named **SampleDB** has a table named **Registration** as follows:

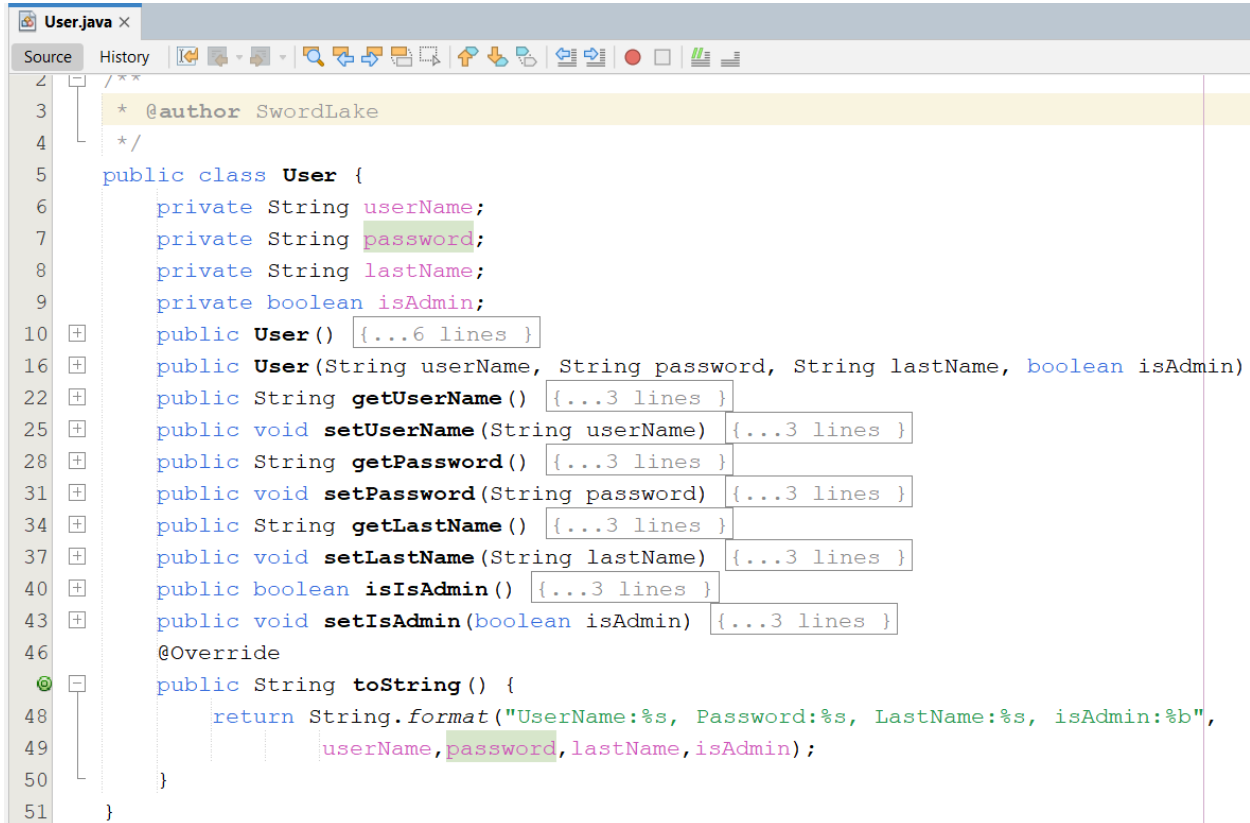
T15PGen2.SampleDB...dbo.Registration			
Column Name	Data Type	Allow Nulls	
UserName	varchar(30)	<input type="checkbox"/>	
Password	varchar(30)	<input type="checkbox"/>	
LastName	nvarchar(50)	<input type="checkbox"/>	
isAdmin	bit	<input type="checkbox"/>	

T15PGen2.SampleDB - dbo.Registration				
UserName	Password	LastName	isAdmin	
U001	123	Tom	True	
U002	456	David	False	
U003	789	John	False	
U004	012	Mark	True	
U005	134	Kate	False	

**Step 02.** Write codes for the classes in the **Models.DTO** and **Models.DAO** packages as the follows:

## 2.1. Models.DTO package

### • User.java



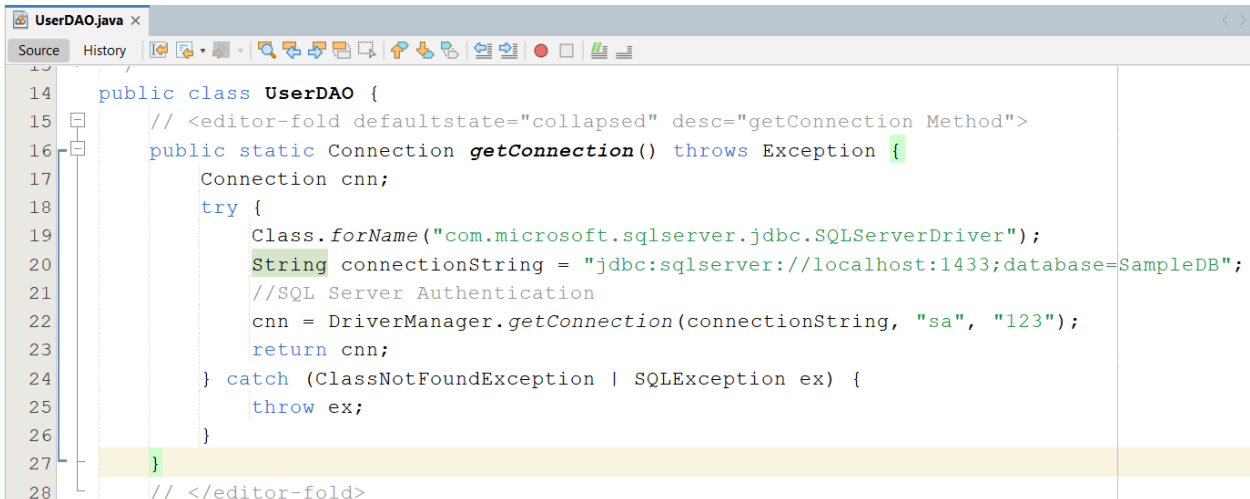
```

1  /**
2   *
3   * @author SwordLake
4   */
5  public class User {
6      private String userName;
7      private String password;
8      private String lastName;
9      private boolean isAdmin;
10     public User() {...6 lines }
16     public User(String userName, String password, String lastName, boolean isAdmin)
22     public String getUserName() {...3 lines }
25     public void setUserName(String userName) {...3 lines }
28     public String getPassword() {...3 lines }
31     public void setPassword(String password) {...3 lines }
34     public String getLastName() {...3 lines }
37     public void setLastName(String lastName) {...3 lines }
40     public boolean isIsAdmin() {...3 lines }
43     public void setIsAdmin(boolean isAdmin) {...3 lines }
46     @Override
47     public String toString() {
48         return String.format("UserName:%s, Password:%s, LastName:%s, isAdmin:%b",
49             userName, password, lastName, isAdmin);
50     }
51 }

```

## 2.2. Models.DAO package

### • UserDao.java



```

14  public class UserDao {
15      // <editor-fold defaultstate="collapsed" desc="getConnection Method">
16      public static Connection getConnection() throws Exception {
17          Connection cnn;
18          try {
19              Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
20              String connectionString = "jdbc:sqlserver://localhost:1433;database=SampleDB";
21              //SQL Server Authentication
22              cnn = DriverManager.getConnection(connectionString, "sa", "123");
23              return cnn;
24          } catch (ClassNotFoundException | SQLException ex) {
25              throw ex;
26          }
27      }
28      // </editor-fold>

```

```

30 // <editor-fold defaultstate="collapsed" desc="login Method">
31 public User login(String userName, String password) throws Exception {
32     User user = null;
33     Connection cnn = null;
34     PreparedStatement preStm = null;
35     ResultSet rs = null;
36     String lastName;
37     boolean isAdmin;
38     try {
39         cnn = getConnection();
40         String sql =
41             "select LastName,isAdmin from Registration where [UserName]=? and [Password]=?";
42         preStm = cnn.prepareStatement(sql);
43         preStm.setString(1, userName);
44         preStm.setString(2, password);
45         rs = preStm.executeQuery();
46         while (rs.next()) {
47             lastName = rs.getString(1);
48             isAdmin = rs.getBoolean(2);
49             user = new User(userName, password, lastName, isAdmin);
50         } //end while
51     } catch (Exception ex) {
52         throw ex;
53     } finally {
54         if (rs != null) {
55             rs.close();
56         }
57         if (preStm != null) {
58             preStm.close();
59         }
60         if (cnn != null) {
61             cnn.close();
62         }
63     }
64     return user;
65 } //end login
66 // </editor-fold>

68 // <editor-fold defaultstate="collapsed" desc="getUserByUserName Method">
69 public User getUserByUserName(String userName) throws Exception {
70     User user = null;
71     Connection cnn = null;
72     PreparedStatement preStm = null;
73     ResultSet rs = null;
74     String lastName,password;
75     boolean isAdmin;
76     try {
77         cnn = getConnection();
78         String sql =
79             "select Password,LastName,isAdmin from Registration where [UserName]=?";
80         preStm = cnn.prepareStatement(sql);
81         preStm.setString(1, userName);
82         rs = preStm.executeQuery();

```

```

83         while (rs.next()) {
84             password = rs.getString(1);
85             lastName = rs.getString(2);
86             isAdmin = rs.getBoolean(3);
87             user = new User(userName, password, lastName, isAdmin);
88         } //end while
89     } catch (Exception ex) {
90         throw ex;
91     } finally {
92         if (rs != null) {
93             rs.close();
94         }
95         if (preStm != null) {
96             preStm.close();
97         }
98         if (cnn != null) {
99             cnn.close();
100         }
101     }
102     return user;
103 } //end getUserByUserName
104 // </editor-fold>

106 // <editor-fold defaultstate="collapsed" desc="searchUserByLastName Method">
107 public List<User> searchUserByLastName(String searchValue) throws Exception {
108     String userName, password, lastName;
109     boolean isAdmin;
110     Connection cnn = null;
111     PreparedStatement preStm = null;
112     ResultSet rs = null;
113     List<User> userList = new ArrayList();
114     try {
115         cnn = getConnection();
116         String sql = "select UserName,Password,LastName,isAdmin from Registration "
117             + " where LastName like ?";
118         preStm = cnn.prepareStatement(sql);
119         preStm.setString(1, "%" + searchValue + "%");
120         rs = preStm.executeQuery();
121         while (rs.next()) {
122             userName = rs.getString(1);
123             password = rs.getString(2);
124             lastName = rs.getString(3);
125             isAdmin = rs.getBoolean(4);
126             User user = new User(userName, password, lastName, isAdmin);
127             userList.add(user);
128         } //end while

```

```

129         } catch (Exception ex) {
130             throw ex;
131         } finally {
132             if (rs != null) {
133                 rs.close();
134             }
135             if (preStm != null) {
136                 preStm.close();
137             }
138             if (cnn != null) {
139                 cnn.close();
140             }
141         }
142         if (userList.isEmpty()) {
143             return null;
144         }
145         return userList;
146     } //end searchUserByLastName
147 } // </editor-fold>

```

```

149 // <editor-fold defaultstate="collapsed" desc="addUser Method">
150 public boolean addUser(User user) throws Exception {
151     PreparedStatement preStm = null;
152     Connection cnn = null;
153     try {
154         cnn = getConnection();
155         String sql = "Insert Registration values(?,?,?,?)";
156         preStm = cnn.prepareStatement(sql);
157         preStm.setString(1, user.getUserName());
158         preStm.setString(2, user.getPassword());
159         preStm.setString(3, user.getLastName());
160         preStm.setBoolean(4, user.isIsAdmin());
161         return preStm.executeUpdate() > 0;
162     } catch (Exception ex) {
163         throw ex;
164     } finally {
165         if (preStm != null) {
166             preStm.close();
167         }
168         if (cnn != null) {
169             cnn.close();
170         }
171     }
172 } //end addUser

```

```

175 // <editor-fold defaultstate="collapsed" desc="deleteUser Method">
176 public boolean deleteUser(String userName) throws Exception {
177     PreparedStatement preStm = null;
178     Connection cnn = null;
179     try {
180         cnn = getConnection();
181         String sql = "delete Registration Where UserName=?";
182         preStm = cnn.prepareStatement(sql);
183         preStm.setString(1, userName);
184         return preStm.executeUpdate() > 0;
185     } catch (Exception ex) {
186         throw ex;
187     } finally {
188         if (preStm != null) {
189             preStm.close();
190         }
191         if (cnn != null) {
192             cnn.close();
193         }
194     }
195 } //end deleteUser
196 // </editor-fold>

226 public boolean updateUser(User user) throws Exception {
227     PreparedStatement preStm = null;
228     Connection cnn = null;
229     try {
230         cnn = getConnection();
231         String sql =
232             "Update Registration Set Password=?,LastName=?,isAdmin=? Where UserName=?";
233         preStm = cnn.prepareStatement(sql);
234         preStm.setString(1, user.getPassword());
235         preStm.setString(2, user.getLastName());
236         preStm.setBoolean(3, user.isIsAdmin());
237         preStm.setString(4, user.getUserName());
238         return preStm.executeUpdate() > 0;
239     } catch (Exception ex) {
240         throw ex;
241     } finally {
242         if (preStm != null) {
243             preStm.close();
244         }
245         if (cnn != null) {
246             cnn.close();
247         }
248     }
249 } //end updateUser
250 // </editor-fold>
251 }

```

**Step 03.** Write codes for the classes in the **Controllers.Authentication** package as the follows:

- **LoginController.java**

```

LoginController.java x
Source History
14  /**
15   *
16   * @author SwordLake
17   */
18  @WebServlet(name = "LoginController", urlPatterns = {"/LoginController"})
19  public class LoginController extends HttpServlet {
20      private final String searchPage = "Search.jsp";
21      private final String userController = "UserController";
22      private final String messagePage = "DisplayMessage.jsp";
23      protected void processRequest(HttpServletRequest request,
24                                  HttpServletResponse response)
25          throws ServletException, IOException {
26          String url = messagePage;
27          response.setContentType("text/html;charset=UTF-8");
28          String userName = request.getParameter("txtUsername");
29          String password = request.getParameter("txtPassword");
30          try {
31              UserDao userDao = new UserDao();
32              User user = userDao.login(userName, password);
33              if (user != null) {
34                  HttpSession session=request.getSession();
35                  session.setAttribute("userLoggedIn",user);
36                  if(user.isIsAdmin()){
37                      url = searchPage;
38                  }
39                  else{
40                      url = userController+"?action=Details&&UserName="+userName;
41                  }
42              }
43              else{
44                  request.setAttribute("action","Login to website");
45                  request.setAttribute("page","Login.html");
46                  request.setAttribute("message","The user name or password is invalid");
47              }
48          } catch (Exception ex) {
49              log(ex.getMessage());
50          }
51          finally
52          {
53              RequestDispatcher rd = request.getRequestDispatcher(url);
54              rd.forward(request, response);
55          }
56      }
57      + HttpServlet methods. Click on the + sign on the left to edit the code.
95  }

```



**Step 04.** Write codes for the classes in the **Controllers.User** package as the follows:

- **CreateController.java**

```

12  /**
13   * @author SwordLake
14   */
15  @WebServlet(name = "CreateController", urlPatterns = {"/CreateController"})
16  public class CreateController extends HttpServlet {
17      private final String displayMessagePage = "DisplayMessage.jsp";
18      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
19          throws ServletException, IOException {
20          String userName, password, lastName;
21          boolean isAdmin = true;
22          String message = "Something went wrong.";
23          response.setContentType("text/html;charset=UTF-8");
24          String url = displayMessagePage;
25          try {
26              userName = request.getParameter("txtUserName");
27              password = request.getParameter("txtPassword");
28              lastName = request.getParameter("txtLastName");
29              String admin = request.getParameter("chkIsAdmin");
30              if (admin == null) {
31                  isAdmin = false;
32              }
33              if (!userName.isEmpty()) {
34                  User user = new User(userName, password, lastName, isAdmin);
35                  UserDao userDao = new UserDao();
36                  if (userDao.addUser(user) == true) {
37                      message = "The user '"+userName+"' has been added successfully";
38                  }
39              }
40          } catch (Exception ex) {
41              log(ex.getMessage());
42          }
43          finally{
44              request.setAttribute("action", "Create User");
45              request.setAttribute("page", "Login.html");
46              request.setAttribute("message", message);
47              RequestDispatcher rd = request.getRequestDispatcher(url);
48              rd.forward(request, response);
49          }
50      } //end processRequest
51      // Servlet methods. Click on the + sign on the left to edit the code.
89  } //end CreateController
    
```

- **SearchController.java**

```

SearchController.java x
Source History
15  /**
16   * @author SwordLake
17   */
18  @WebServlet(name = "SearchController", urlPatterns = {"/SearchController"})
19  public class SearchController extends HttpServlet {
20      private final String searchPage = "Search.jsp";
21      protected void processRequest(HttpServletRequest request,
22          HttpServletResponse response) throws ServletException, IOException {
23          String url, searchValue;
24          response.setContentType("text/html;charset=UTF-8");
25          PrintWriter out = response.getWriter();
26          url = searchPage;
27          try {
28              searchValue = request.getParameter("txtSearchValue");
29              if (!searchValue.isEmpty()) {
30                  UserDao userDao = new UserDao();
31                  List<User> userList = userDao.searchUserByLastName(searchValue);
32                  request.setAttribute("SearchResult", userList);
33              }
34          } catch (Exception ex) {
35              log(ex.getMessage());
36          }
37          finally{
38              RequestDispatcher rd = request.getRequestDispatcher(url);
39              rd.forward(request, response);
40              out.close();
41          }
42      } //processRequest
43      //HttpServlet methods. Click on the + sign on the left to edit the code.
81  } //end SearchController
  
```

## • DeleteController.java

```

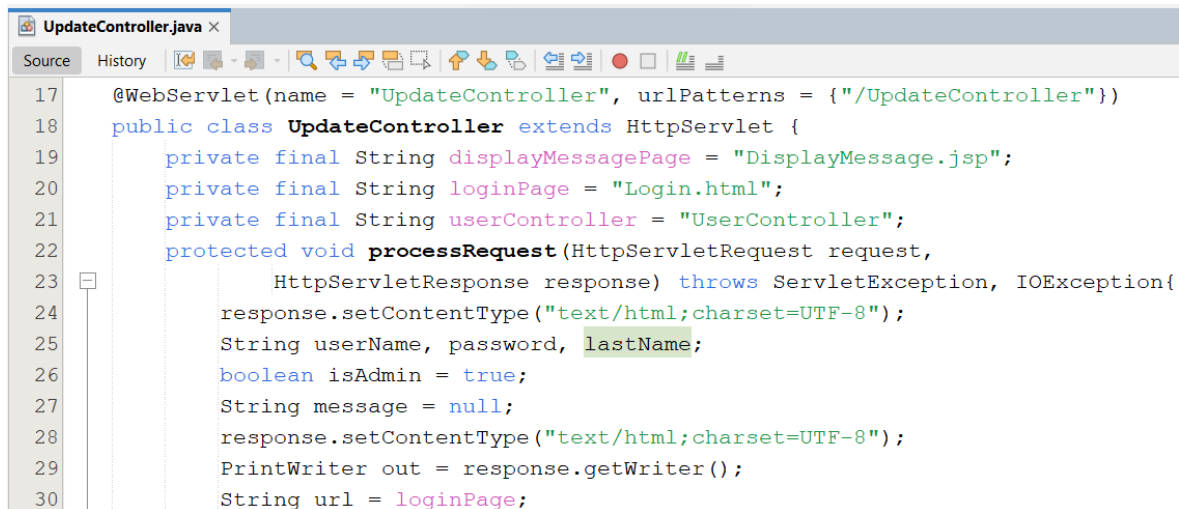
DeleteController.java x
Source History
14  /* @author SwordLake */
15  @WebServlet(name = "DeleteController", urlPatterns = {"/DeleteController"})
16  public class DeleteController extends HttpServlet {
17      private final String loginPage = "Login.html";
18      private final String displayMessagePage = "DisplayMessage.jsp";
19      private final String userController = "UserController";
20
21      protected void processRequest(HttpServletRequest request,
22          HttpServletResponse response) throws ServletException, IOException {
23          response.setContentType("text/html;charset=UTF-8");
24          PrintWriter out = response.getWriter();
25          String url = displayMessagePage;
26          String message = null, page = null;
  
```

```

27         try {
28             HttpSession session = request.getSession();
29             User userLoggedIn = (User) session.getAttribute("userLoggedIn");
30             String userName = request.getParameter("UserName");
31             String searchValue = request.getParameter("txtSearchValue");
32             if (userName.equals(userLoggedIn.getUserName())) {
33                 message = "This user logged in, can not delete.";
34                 page = userController+"?action=Search&txtSearchValue="+searchValue;
35             } else {
36                 UserDao userDao = new UserDao();
37                 if (!userName.isEmpty()) {
38                     if (userDao.deleteUser(userName) == true) {
39                         url = userController+"?action=Search";
40                     } else {
41                         message = "Something went wrong.";
42                         page = loginPage;
43                     }
44                 }
45             }
46         } catch (Exception ex) {
47             log(ex.getMessage());
48         } finally {
49             request.setAttribute("action", "Delete User");
50             request.setAttribute("message", message);
51             request.setAttribute("page", page);
52             RequestDispatcher rd = request.getRequestDispatcher(url);
53             rd.forward(request, response);
54             out.close();
55         }
56     } //end processRequest
57     + HttpServlet methods. Click on the + sign on the left to edit the code.
96 } //end DeleteController

```

## • UpdateController.java



```

17 @WebServlet(name = "UpdateController", urlPatterns = {"/UpdateController"})
18 public class UpdateController extends HttpServlet {
19     private final String displayMessagePage = "DisplayMessage.jsp";
20     private final String loginPage = "Login.html";
21     private final String userController = "UserController";
22     protected void processRequest(HttpServletRequest request,
23         HttpServletResponse response) throws ServletException, IOException{
24         response.setContentType("text/html; charset=UTF-8");
25         String userName, password, lastName;
26         boolean isAdmin = true;
27         String message = null;
28         response.setContentType("text/html; charset=UTF-8");
29         PrintWriter out = response.getWriter();
30         String url = loginPage;

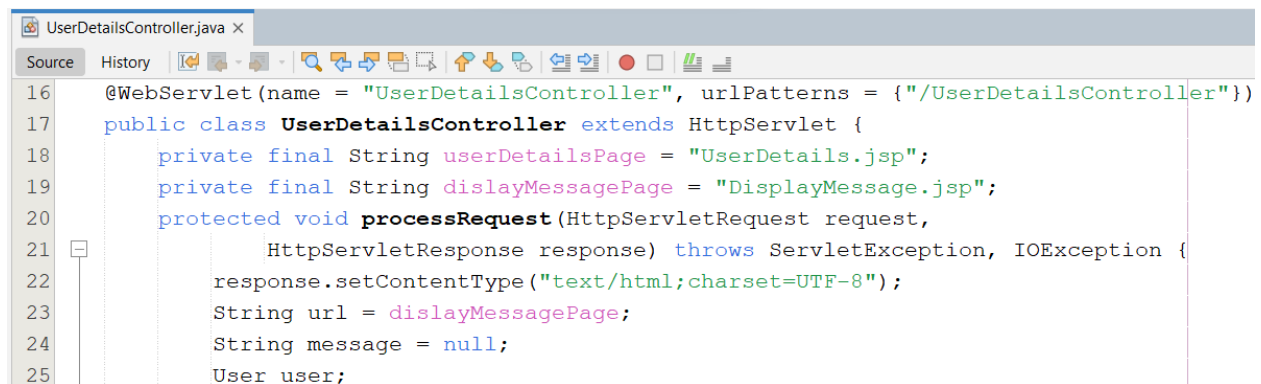
```

```

31     try {
32         HttpSession session=request.getSession();
33         User userLoggedIn =(User)session.getAttribute("userLoggedIn");
34         userName = request.getParameter("txtUserName");
35         password = request.getParameter("txtPassword");
36         lastName = request.getParameter("txtLastName");
37         String admin = request.getParameter("chkIsAdmin");
38         if (admin == null) {
39             isAdmin = false;
40         }
41         if (!userName.isEmpty()) {
42             User user = new User(userName, password, lastName, isAdmin);
43             UserDao userDao = new UserDao();
44             if (userDao.updateUser(user) == true) {
45                 if(userLoggedIn.isIsAdmin()){
46                     url = userController+"?action=Search";
47                 }
48             }
49             else{
50                 url = displayMessagePage;
51                 message = "The user '"+userName+"' has been updated successfully";
52             }
53         }
54     } catch (Exception ex) {
55         log(ex.getMessage());
56     }
57     finally{
58         request.setAttribute("action","Update User");
59         request.setAttribute("page","Search.jsp");
60         request.setAttribute("message",message);
61         RequestDispatcher rd = request.getRequestDispatcher(url);
62         rd.forward(request, response);
63         out.close();
64     }
65 } //processRequest
66 //HttpServlet methods. Click on the + sign on the left to edit the code.
104 } //end UpdateController

```

## • UserDetailsController.java



```

UserDetailsController.java x
Source History
16 @WebServlet(name = "UserDetailsController", urlPatterns = {"/UserDetailsController"})
17 public class UserDetailsController extends HttpServlet {
18     private final String userDetailsPage = "UserDetails.jsp";
19     private final String displayMessagePage = "DisplayMessage.jsp";
20     protected void processRequest (HttpServletRequest request,
21         HttpServletResponse response) throws ServletException, IOException {
22         response.setContentType("text/html;charset=UTF-8");
23         String url = displayMessagePage;
24         String message = null;
25         User user;

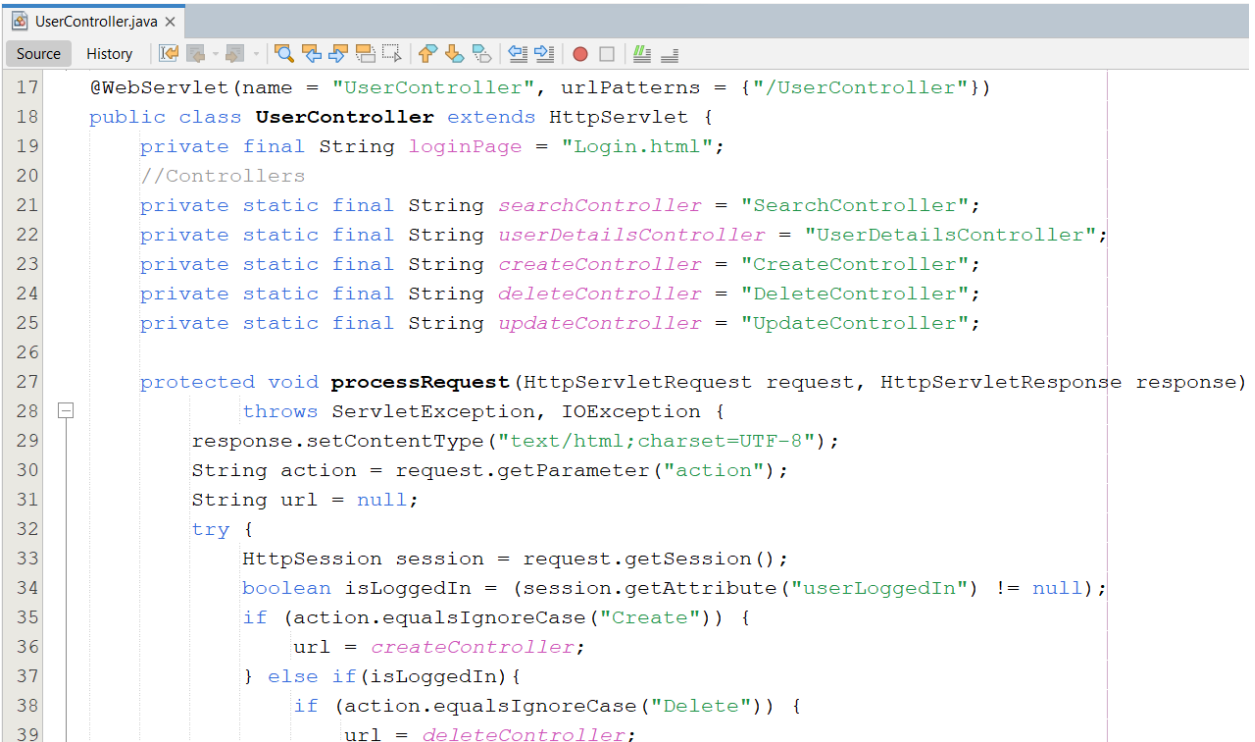
```

```

26         try {
27             String userName = request.getParameter("UserName");
28             UserDao userDao = new UserDao();
29             if (!userName.isEmpty()) {
30                 user = userDao.getUserByUserName(userName);
31                 if (user != null) {
32                     request.setAttribute("userDetails", user);
33                     url = userDetailsPage;
34                 }
35                 else {
36                     message = "The user '" + userName + "' not found.";
37                     request.setAttribute("action", "User Details");
38                     request.setAttribute("page", "Search.jsp");
39                     request.setAttribute("message", message);
40                 }
41             }
42         } catch (Exception ex) {
43             log(ex.getMessage());
44         }
45         finally {
46             RequestDispatcher rd = request.getRequestDispatcher(url);
47             rd.forward(request, response);
48         }
49     } //processRequest
50     //HttpServlet methods. Click on the + sign on the left to edit the code.
88 } //end UserDetailsController

```

## • UserController.java



```

17 @WebServlet(name = "UserController", urlPatterns = {"/UserController"})
18 public class UserController extends HttpServlet {
19     private final String loginPage = "Login.html";
20     //Controllers
21     private static final String searchController = "SearchController";
22     private static final String userDetailsController = "UserDetailsController";
23     private static final String createController = "CreateController";
24     private static final String deleteController = "DeleteController";
25     private static final String updateController = "UpdateController";
26
27     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
28         throws ServletException, IOException {
29         response.setContentType("text/html; charset=UTF-8");
30         String action = request.getParameter("action");
31         String url = null;
32         try {
33             HttpSession session = request.getSession();
34             boolean isLoggedIn = (session.getAttribute("userLoggedIn") != null);
35             if (action.equalsIgnoreCase("Create")) {
36                 url = createController;
37             } else if (isLoggedIn) {
38                 if (action.equalsIgnoreCase("Delete")) {
39                     url = deleteController;

```

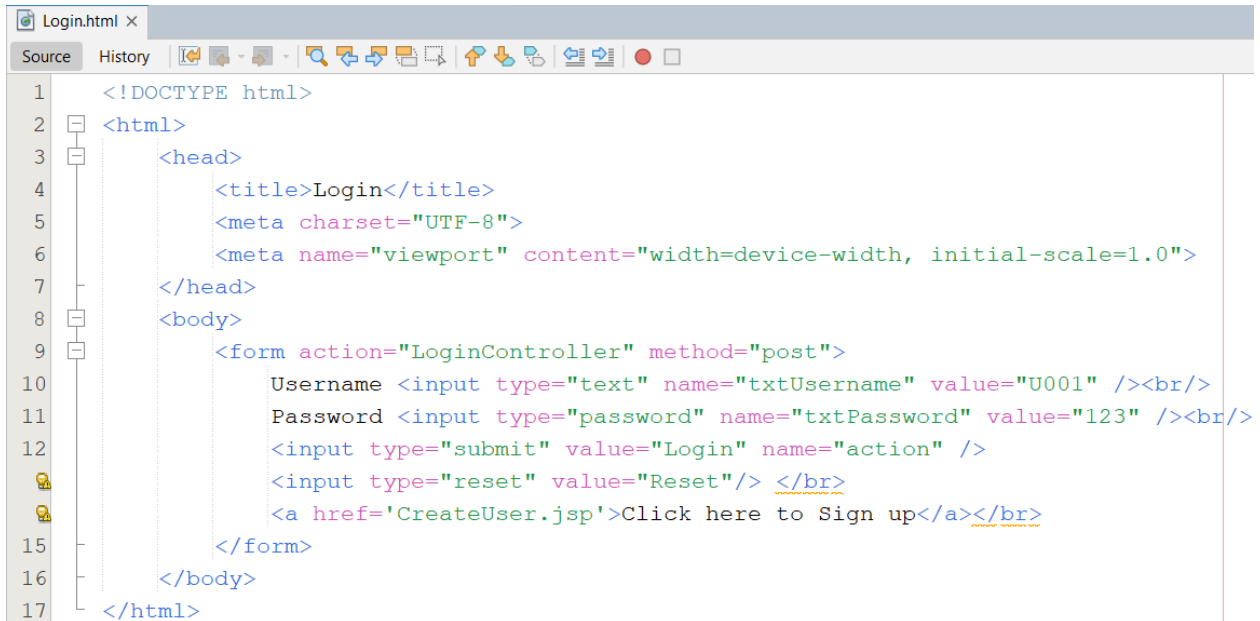
```

40         } else if (action.equalsIgnoreCase("Update")) {
41             url = updateController;
42         } else if (action.equalsIgnoreCase("Search")) {
43             url = searchController;
44         } else if (action.equalsIgnoreCase("Details")) {
45             url = userDetailsController;
46         }
47     } else {
48         url = loginPage;
49     }
50 } finally {
51     RequestDispatcher rd = request.getRequestDispatcher(url);
52     rd.forward(request, response);
53 }
54 } //end processRequest
55 + HttpServlet methods. Click on the + sign on the left to edit the code.
94 } //end UserController

```

**Step 05.** Write codes for the web pages as the follows:

- **Login.html page**



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Login</title>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 </head>
8 <body>
9 <form action="LoginController" method="post">
10     Username <input type="text" name="txtUsername" value="U001" /><br/>
11     Password <input type="password" name="txtPassword" value="123" /><br/>
12     <input type="submit" value="Login" name="action" />
13     <input type="reset" value="Reset" /> </br>
14     <a href='CreateUser.jsp'>Click here to Sign up</a><br>
15 </form>
16 </body>
17 </html>

```

- **CreateUser.jsp page**

```

CreateUser.jsp x
Source History
<!--
2      Document      : CreateUser
3      Author       : SwordLake
4  -->
5  <%@page contentType="text/html" pageEncoding="UTF-8"%>
6  <!DOCTYPE html>
7  <html>
8  <head>
9      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
10     <title>Create User</title>
11 </head>
12 <body>
13     <h1>Create User</h1>
14     <form action="UserController" method="post">
15         UserName <input type="text" placeholder="Enter user name" name="txtUserName"/><br/>
16         Password <input type="password" placeholder="Enter password" name="txtPassword"/><br/>
17         LastName <input type="text" placeholder="Enter last name" name="txtLastName"/><br/>
18         <input type="checkbox" name="chkIsAdmin" disabled="true" />isAdmin<br/>
19         <input type="submit" value="Create" name="action" /><br/>
20         <a href='Login.html'>Back</a><br/>
21     </form>
22 </body>
23 </html>

```

## • Search.jsp page

```

Search.jsp x
Source History
<!--
2      Author       : SwordLake
3  -->
4  <%@page import="Models.DTO.User"%>
5  <%@page import="Models.DAO.UserDAO"%>
6  <%@page import="java.util.List"%>
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10 <head>
11     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12     <title>Search Users</title>
13 </head>
14 <body>
15 <%
16     String lastName = null, searchValue=null;
17     User userLoggedIn =(User)session.getAttribute("userLoggedIn");
18     if (userLoggedIn != null) {
19         lastName = userLoggedIn.getLastName();
20     }
21     searchValue = request.getParameter("txtSearchValue");
22 %>

```

```

23 <h3>Welcome to <text style="color: red"><%=lastName%></text></h3>
24 <h1>Search user by last name</h1>
25 <form action="UserController" method="post">
26     Enter search value
27     <input type="text" name="txtSearchValue"
28         value="<%=searchValue!=null?searchValue:"">" /><br/>
29     <input type="submit" value="Search" name="action" />
30 </form>
31 </body>
32 <%
33 // if (searchValue != null) {
34     UserDao userDao = new UserDao();
35     List<User> userList = (List<User>) request.getAttribute("SearchResult");
36     if (userList != null) {
37         %>
38         <table border="1">
39             <thead>
40                 <tr>
41                     <th>No.</th>
42                     <th>UserName</th>
43                     <th>Password</th>
44                     <th>LastName</th>
45                     <th>Role</th>
46                     <th colspan="2">Action</th>
47                 </tr>
48             </thead>
49             <tbody>
50                 <%
51                     int count = 0;
52                     for (User user : userList) {
53
54                         <tr>
55                             <td><%=++count)%></td>
56                             <td><%=user.getUserName()%></td>
57                             <td><%=user.getPassword()%></td>
58                             <td><%=user.getLastName()%></td>
59                             <td><input type="checkbox" <% if (user.isIsAdmin()) { %> checked <% }%> /></td>
60                             <td><a href=UserController?action=Delete&UserName=<%=user.getUserName()%>&txtSearchValue=<%=searchValue%>>
61                                 Delete</td>
62                             <td><a href=UserController?action=Details&UserName=<%=user.getUserName()%>&txtSearchValue=<%=searchValue%>>
63                                 View</td>
64                         </tr>
65                     <%
66                         }//end for
67                     %>
68                 </tbody>
69             </table>
70             <h3>Number of users found : <%=userList.size()%></h3>
71             <%
72             }//end if
73             else if(searchValue!=null){
74                 %>
75                 <h3>No users were found.</h3>
76                 <%
77                 }
78                 %>
79             <a href='Login.html'>Back</a></br>
80 </html>

```

- **DisplayMessage.jsp page**



```

DisplayMessage.jsp x
Source History
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>Information</title>
13 </head>
14 <body>
15 <h1><%=request.getAttribute("action")%></h1>
16 <%
17 String message = (String)request.getAttribute("message");
18 if (message != null) {
19 %>
20 <h3 style="color: red"><%=message%></h3>
21 <%
22 }
23 %>
24 <a href=<%=request.getAttribute("page")%>>Back</a></br>
25 </body>
26 </html>

```

## • UserDetails.jsp page

```

UserDetails.jsp x
Source History
2 <!--
3 Document : AddUser
4 Author : SwordLake
5 -->
6 <%@page import="Models.DTO.User"%>
7 <%@page import="Models.DAO.UserDAO"%>
8 <%@page contentType="text/html" pageEncoding="UTF-8"%>
9 <!DOCTYPE html>
10 <html>
11 <...4 lines />
12 <body>
13 <%
14 User userLoggedIn = (User)session.getAttribute("userLoggedIn");
15 if (userLoggedIn == null) {
16 response.sendRedirect("Login.html");
17 }
18 String searchValue = request.getParameter("txtSearchValue");
19 User user = (User) request.getAttribute("userDetails");
20 if (user != null) {
21 %>
22 <h1>User Details</h1>
23 <form action="UserController" method="post">
24 Username <input type="text" readonly="true" name="txtUserName" value='<%=user.getUserName() %>' /><br/>
25 Password <input type="password" name="txtPassword" value='<%=user.getPassword() %>' /><br/>
26 LastName <input type="text" name="txtLastName" value='<%=user.getLastName() %>' /><br/>
27 <input type="hidden" value='<%=searchValue %>' name="txtSearchValue"/>
28 <%if (userLoggedIn.isIsAdmin() == true) { %>
29 <input type="checkbox" name="chkIsAdmin" <%if (user.isIsAdmin()) { %> checked <%} %> /> isAdmin<br/>
30 <% } %>
31
32
33

```

```

34      <input type="submit" value="Update" name="action" /></br>
35      <%if (userLoggedIn.isIsAdmin()==true) { %>
36      <a href='UserController?action=Search&&txtSearchValue=<%=searchValue%>'>Back</a></br>
37      <% }
38      else{
39      %>
40      <a href='Login.html'>Back</a></br>
41      <%
42      }
43      %>
44      </form>
45      <%
46      }
47      %>
48      </body>
49      </html>

```

**Step 05.** Open the **web.xml** in the WEB-INF folder then update as the follows:



```

web.xml x
Source  General  Servlets  Filters  Pages  References  Security  History
1      <?xml version="1.0" encoding="UTF-8"?>
2      <web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3      <session-config>
4          <session-timeout>
5              30
6          </session-timeout>
7      </session-config>
8      <welcome-file-list>
9          <welcome-file>Login.html</welcome-file>
10     </welcome-file-list>
11     </web-app>

```

**Step 06.** Right-click on **Lab04\_ManageUsersByMVC2** project, select Run to run the web app then test all functions.