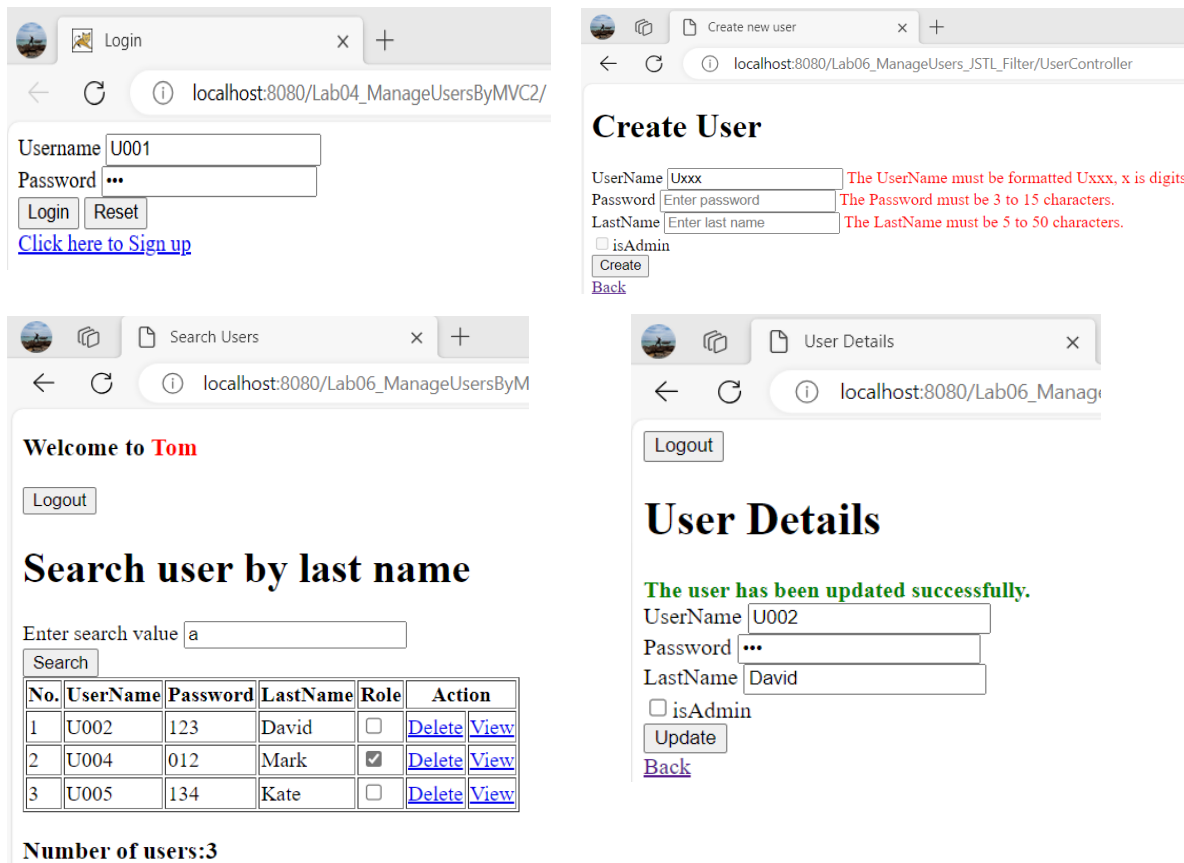


Building an Java Web Application

Using MVC2, JSP & JSTL

Requirements

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



Login

Username: U001
 Password: ...
 Login Reset
[Click here to Sign up](#)

Create User

UserName: Uxxx The UserName must be formatted Uxxx, x is digits.
 Password: Enter password The Password must be 3 to 15 characters.
 LastName: Enter last name The LastName must be 5 to 50 characters.
☐ isAdmin
 Create Back

Search Users

Welcome to **Tom**
 Logout
Search user by last name
 Enter search value: a
 Search

No.	UserName	Password	LastName	Role	Action
1	U002	123	David	<input type="checkbox"/>	Delete View
2	U004	012	Mark	<input checked="" type="checkbox"/>	Delete View
3	U005	134	Kate	<input type="checkbox"/>	Delete View

Number of users:3

User Details

Logout
The user has been updated successfully.
 UserName: U002
 Password: ...
 LastName: David
☐ isAdmin
 Update Back

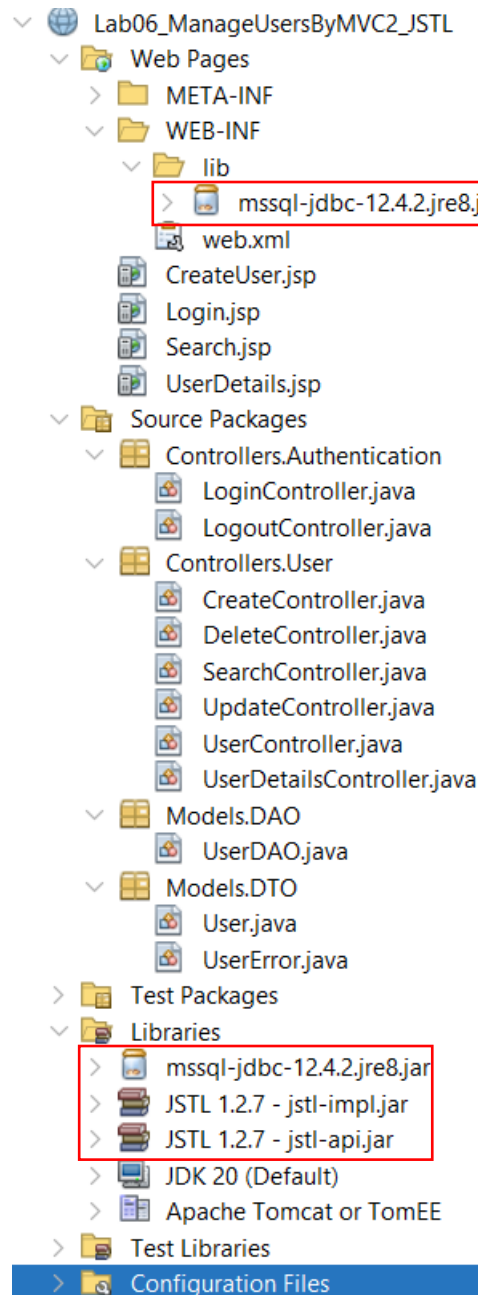
Lab Objectives

In this lab, you will be:

- Create Java web project by MVC2 Model
- Create the web pages by HTML, and JSP (JSTL & EL)
- Apply passing data from HTML to Servlet by GET and POST method
- Apply Error handling in the web app
- Apply RequestDispatcher to pass data between Servlets

- Performs CRUD functions by SqlJDBC driver and SQL Server
- Run the project and test the functions of the project

Step 01. Create a Java Web App named **Lab06_ManageUsersByMVC2_JSTL** is structured as follows:



- On the project, right-click on Libraries | Add JAR/Folder then select **sqljdbc(sqljdbc42.jar)** driver that **matches the Sql Server version** on the machine.

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

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"/>

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

```

User.java x
Source History
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 }

```

• UserError.java

```

UserError.java x
Source History
1 package Models.DTO;
2 /** * @author SwordLake ...3 lines */
5 public class UserError {
6     private String userNameError;
7     private String lastNameError;
8     private String passwordError;
9     private String duplicateUserName;
10    public UserError() { }
11    public UserError(String userNameError, String lastNameError,
12                     String passwordError, String duplicateUserName) {
13        this.userNameError = userNameError;
14        this.lastNameError = lastNameError;
15        this.passwordError = passwordError;
16        this.duplicateUserName = duplicateUserName;
17    }
18    public String getUserNameError() {...3 lines }
21    public void setUserNameError(String userNameError) {...3 lines }
24    public String getLastNameError() {...3 lines }
27    public void setLastNameError(String lastNameError) {...3 lines }
30    public String getPasswordError() {...3 lines }
33    public void setPasswordError(String passwordError) {...3 lines }
36    public String getDuplicateUserName() {...3 lines }
39    public void setDuplicateUserName(String duplicateUserName) {...3 lines }
42 }
  
```

2.2. Models.DAO package

• UserDao.java

```

UserDAO.java x
Source History
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
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 loginPage = "Login.jsp";
23      protected void processRequest(HttpServletRequest request,
24          HttpServletResponse response) throws ServletException, IOException {
25          String url = loginPage;
26          response.setContentType("text/html;charset=UTF-8");
27          String userName = request.getParameter("txtUsername");
28          String password = request.getParameter("txtPassword");
29          try {
30              UserDao userDao = new UserDao();
31              User user = userDao.login(userName, password);
32              if (user != null) {
33                  HttpSession session=request.getSession();
34                  session.setAttribute("userLoggedIn",user);
35                  if(user.isIsAdmin()){
36                      url = searchPage;
37                  }
38                  else{
39                      url = userController+"?action=Details&&UserName="+userName;
40                  }
41              }
42              else{
43                  request.setAttribute("message","The user name or password is invalid");
44              }
45          } catch (Exception ex) {
46              log(ex.getMessage());
47          }
48          finally {
49              RequestDispatcher rd = request.getRequestDispatcher(url);
50              rd.forward(request, response);
51          }
52      }
53      HttpServlet methods. Click on the + sign on the left to edit the code.
91  }
    
```

- **LogoutController.java**

```

LogoutController.java x
Source History
13  * @author SwordLake
14  */
15  @WebServlet(name = "LogoutController", urlPatterns = {"/LogoutController"})
16  public class LogoutController extends HttpServlet {
17      private final String loginPage = "Login.jsp";
18      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
19          throws ServletException, IOException {
20          response.setContentType("text/html;charset=UTF-8");
21          String url = loginPage;
22          response.setContentType("text/html;charset=UTF-8");
23          try {
24              HttpSession session = request.getSession(false);
25              if (session != null) {
26                  session.invalidate();
27              }
28          } catch (Exception ex) {
29              log(ex.getMessage());
30          }
31          finally{
32              RequestDispatcher rd = request.getRequestDispatcher(url);
33              rd.forward(request, response);
34          }
35      }
36      HttpServlet methods. Click on the + sign on the left to edit the code.
74  }

```

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

- **CreateController.java**

```

CreateController.java x
Source History
15  * @author SwordLake
16  */
17  @WebServlet(name = "CreateController", urlPatterns = {"/CreateController"})
18  public class CreateController extends HttpServlet {
19      private final String ceateUserPage = "CreateUser.jsp";
20      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
21          throws ServletException, IOException {
22          String userName, password, lastName;
23          boolean isAdmin = true, isError = false;
24          String message = null;
25          response.setContentType("text/html;charset=UTF-8");
26          String url = ceateUserPage;
27          try {
28              UserDao userDao = new UserDao();
29              userName = request.getParameter("txtUserName");
30              password = request.getParameter("txtPassword");
31              lastName = request.getParameter("txtLastName");
32              String admin = request.getParameter("chkIsAdmin");
33              UserError userError = new UserError();

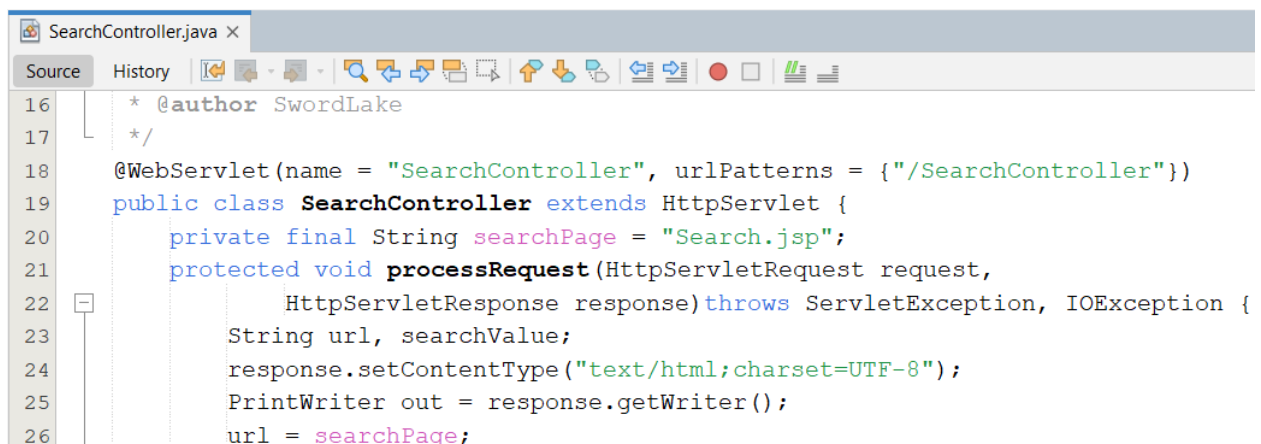
```

```

34         if (userName.matches("U\\d{3}") == false) {
35             userError.setUserNameError("The UserName must be formatted Uxxx, x is digits.");
36             isError = true;
37         } //end check userName
38         if (password.matches("(.){3,15}") == false) {
39             userError.setPasswordError("The Password must be 3 to 15 characters.");
40             isError = true;
41         } //end check Password
42         if (lastName.matches("(.){5,50}") == false) {
43             userError.setLastNameError("The LastName must be 5 to 50 characters.");
44             isError = true;
45         } //end check Password
46         if (userDao.getUserByUserName(userName) != null) {
47             userError.setDuplicateUserName("The UserName already exists.");
48             isError = true;
49         }
50         if (isError == false) {
51             if (admin == null) {
52                 isAdmin = false;
53             }
54             User user = new User(userName, password, lastName, isAdmin);
55             if (userDao.addUser(user) == true) {
56                 message = " <b style='color: green'>The user has been created successfully</b>";
57             }
58             else{
59                 message = "<b style='color: red'>Something went wrong.</b>";
60             }
61         } //end isError
62         else {
63             request.setAttribute("ErrorDetails", userError);
64         }
65     } catch (Exception ex) {
66         log(ex.getMessage());
67     } finally {
68         request.setAttribute("message", message);
69         RequestDispatcher rd = request.getRequestDispatcher(url);
70         rd.forward(request, response);
71     }
72 } //end processRequest
73 + HttpServlet methods. Click on the + sign on the left to edit the code.
112 } //end CreateController

```

• SearchController.java



```

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 //end SearchController

```

HttpServlet methods. Click on the + sign on the left to edit the code.

• DeleteController.java

```

DeleteController.java x
Source History
13 import javax.servlet.http.HttpSession;
14 /* @author SwordLake */
15 @WebServlet(name = "DeleteController", urlPatterns = {"/DeleteController"})
16 public class DeleteController extends HttpServlet {
17     private final String userController = "UserController";
18     protected void processRequest(HttpServletRequest request,
19         HttpServletResponse response) throws ServletException, IOException {
20         response.setContentType("text/html;charset=UTF-8");
21         PrintWriter out = response.getWriter();
22         String url = userController;
23         String message = null;
24         try {
25             UserDao userDao = new UserDao();
26             HttpSession session = request.getSession();
27             User userLoggedIn = (User) session.getAttribute("userLoggedIn");
28             String userName = request.getParameter("UserName");
29             String searchValue = request.getParameter("txtSearchValue");
30             if (userName.equals(userLoggedIn.getUserName())) {
31                 message = "<b style='color: red'>This user logged in, can not delete.</b>";
32             } else {
33                 if (!userName.isEmpty()) {
34                     if (userDao.deleteUser(userName) == true) {
35                         message = "<b style='color: green'>The user has been deleted successfully.</b>";
36                     } else {
37                         message = "<b style='color: red'>Something went wrong.</b>";
38                     }
39                 }
40             }
41             url = userController+"?action=Search&txtSearchValue="+searchValue;
42         } catch (Exception ex) {
43             log(ex.getMessage());
44         } finally {

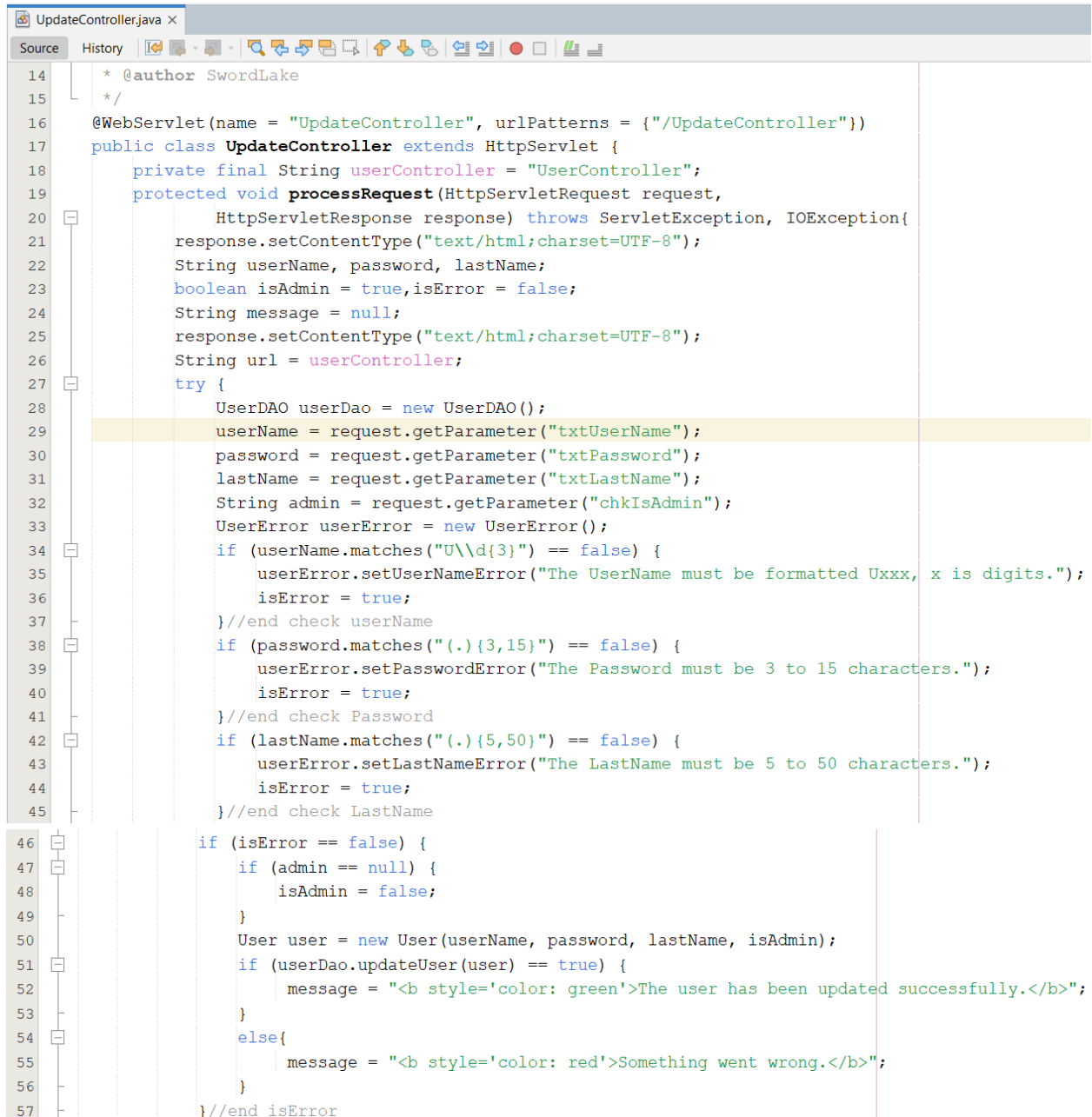
```

```

45         request.setAttribute("message", message);
46         RequestDispatcher rd = request.getRequestDispatcher(url);
47         rd.forward(request, response);
48         out.close();
49     }
50 } //end processRequest
51 + HttpServlet methods. Click on the + sign on the left to edit the code.
90 } //end DeleteController

```

• UpdateController.java



```

UpdateController.java x
Source History
14  * @author SwordLake
15  */
16  @WebServlet(name = "UpdateController", urlPatterns = {"/UpdateController"})
17  public class UpdateController extends HttpServlet {
18      private final String userController = "UserController";
19      protected void processRequest(HttpServletRequest request,
20      HttpServletResponse response) throws ServletException, IOException{
21          response.setContentType("text/html;charset=UTF-8");
22          String userName, password, lastName;
23          boolean isAdmin = true, isError = false;
24          String message = null;
25          response.setContentType("text/html;charset=UTF-8");
26          String url = userController;
27          try {
28              UserDao userDao = new UserDao();
29              userName = request.getParameter("txtUserName");
30              password = request.getParameter("txtPassword");
31              lastName = request.getParameter("txtLastName");
32              String admin = request.getParameter("chkIsAdmin");
33              UserError userError = new UserError();
34              if (userName.matches("U\\d{3}") == false) {
35                  userError.setUserNamedError("The UserName must be formatted Uxxx, x is digits.");
36                  isError = true;
37              } //end check userName
38              if (password.matches("(.){3,15}") == false) {
39                  userError.setPasswordError("The Password must be 3 to 15 characters.");
40                  isError = true;
41              } //end check Password
42              if (lastName.matches("(.){5,50}") == false) {
43                  userError.setLastNameError("The LastName must be 5 to 50 characters.");
44                  isError = true;
45              } //end check LastName
46              if (isError == false) {
47                  if (admin == null) {
48                      isAdmin = false;
49                  }
50                  User user = new User(userName, password, lastName, isAdmin);
51                  if (userDao.updateUser(user) == true) {
52                      message = "<b style='color: green'>The user has been updated successfully.</b>";
53                  }
54                  else{
55                      message = "<b style='color: red'>Something went wrong.</b>";
56                  }
57              } //end isError

```

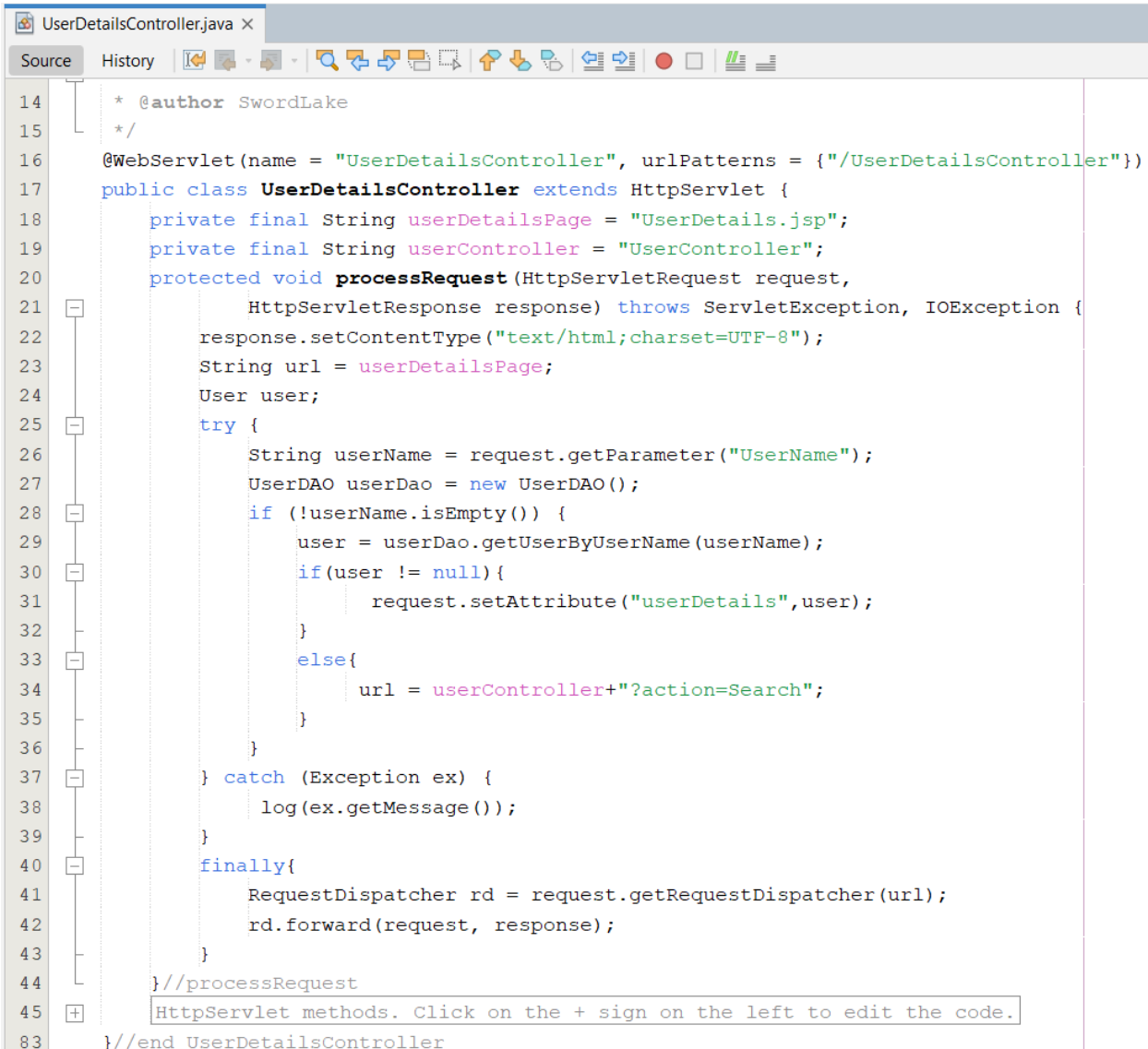
```

58         else {
59             request.setAttribute("ErrorDetails", userError);
60         }
61         url = userController+"?action=Details&UserName="+userName;
62     } catch (Exception ex) {
63         log(ex.getMessage());
64     }
65     finally{
66         request.setAttribute("message",message);
67         RequestDispatcher rd = request.getRequestDispatcher(url);
68         rd.forward(request, response);
69     }
70 } //processRequest
71 //end UpdateController
109

```

HttpServlet methods. Click on the + sign on the left to edit the code.

• UserDetailsController.java



```

14  * @author SwordLake
15  */
16  @WebServlet(name = "UserDetailsController", urlPatterns = {"/UserDetailsController"})
17  public class UserDetailsController extends HttpServlet {
18      private final String userDetailsPage = "UserDetails.jsp";
19      private final String userController = "UserController";
20      protected void processRequest(HttpServletRequest request,
21                                  HttpServletResponse response) throws ServletException, IOException {
22          response.setContentType("text/html;charset=UTF-8");
23          String url = userDetailsPage;
24          User user;
25          try {
26              String userName = request.getParameter("UserName");
27              UserDao userDao = new UserDao();
28              if (!userName.isEmpty()) {
29                  user = userDao.getUserByUserName(userName);
30                  if (user != null) {
31                      request.setAttribute("userDetails", user);
32                  }
33                  else {
34                      url = userController+"?action=Search";
35                  }
36              }
37          } catch (Exception ex) {
38              log(ex.getMessage());
39          }
40          finally{
41              RequestDispatcher rd = request.getRequestDispatcher(url);
42              rd.forward(request, response);
43          }
44      } //processRequest
45      //end UserDetailsController
83

```

HttpServlet methods. Click on the + sign on the left to edit the code.

• UserController.java

```

14  * @author SwordLake
15  */
16  @WebServlet(name = "UserController", urlPatterns = {"/UserController"})
17  public class UserController extends HttpServlet {
18      private final String loginPage = "Login.jsp";
19      private static final String searchController = "SearchController";
20      private static final String userDetailsController = "UserDetailsController";
21      private static final String createController = "CreateController";
22      private static final String deleteController = "DeleteController";
23      private static final String updateController = "UpdateController";
24      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
25          throws ServletException, IOException {
26          response.setContentType("text/html;charset=UTF-8");
27          String action = request.getParameter("action");
28          String url = null;
29          try {
30              HttpSession session = request.getSession();
31              boolean isLoggedIn = (session.getAttribute("userLoggedIn") != null);
32              if (action.equalsIgnoreCase("Create")) {
33                  url = createController;
34              } else if (isLoggedIn) {
35                  if (action.equalsIgnoreCase("Delete")) {
36                      url = deleteController;
37                  } else if (action.equalsIgnoreCase("Update")) {
38                      url = updateController;
39                  } else if (action.equalsIgnoreCase("Search")) {
40                      url = searchController;
41                  } else if (action.equalsIgnoreCase("Details")) {
42                      url = userDetailsController;
43                  }
44              } else {
45                  url = loginPage;
46              }
47          } finally {
48              RequestDispatcher rd = request.getRequestDispatcher(url);
49              rd.forward(request, response);
50          }
51      } //end processRequest
52      //HttpServlet methods. Click on the + sign on the left to edit the code.
91  } //end UserController
    
```

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

• Login.html page


```

Login.jsp x
Source History
4      Author      : SwordLake
5      --%>
6      <%@page contentType="text/html" pageEncoding="UTF-8"%>
7      <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
8      <!DOCTYPE html>
9      <html>
10     <...4 lines />
14     <body>
15         <h1>Login to website</h1>
16         <form action="LoginController" method="post">
17             Username <input type="text" name="txtUsername" /><br/>
18             Password <input type="password" name="txtPassword" /><br/>
19             <input type="submit" value="Login" name="action" />
20             <input type="reset" value="Reset" /> </br>
21             <a href='CreateUser.jsp'>Click here to Sign up</a><br>
22         </form>
23         <c:set var="message" value="${requestScope.message}" />
24         <c:if test="${message != null}">
25             <text style="color: red">${message}</text>
26         </c:if>
27     </body>
28 </html>

```

• CreateUser.jsp page

```

CreateUser.jsp x
Source History
4      Author      : SwordLake
5      --%>
6      <%@page contentType="text/html" pageEncoding="UTF-8"%>
7      <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
8      <!DOCTYPE html>
9      <html>
10     <...4 lines />
14     <body>
15         <h1>Create User</h1>
16         <c:set var="message" value="${requestScope.message}" />
17         <c:if test="${message != null}">
18             ${message}
19         </c:if>
20         <form action="UserController" method="post">
21             <c:set var="error" value="${requestScope.errorDetails}" />
22             UserName <input type="text" name="txtUserName" value="Uxxx" />
23             <c:if test="${not empty error.userNameError}">
24                 <text style="color: red">${error.userNameError}</text>
25             </c:if>
26             <c:if test="${not empty error.duplicateUserName}">
27                 <text style="color: red">${error.duplicateUserName}</text>
28             </c:if><br/>
29             Password <input type="password" placeholder="Enter password" name="txtPassword" />
30             <c:if test="${not empty error.passwordError}">
31                 <text style="color: red">${error.passwordError}</text>
32             </c:if><br/>

```

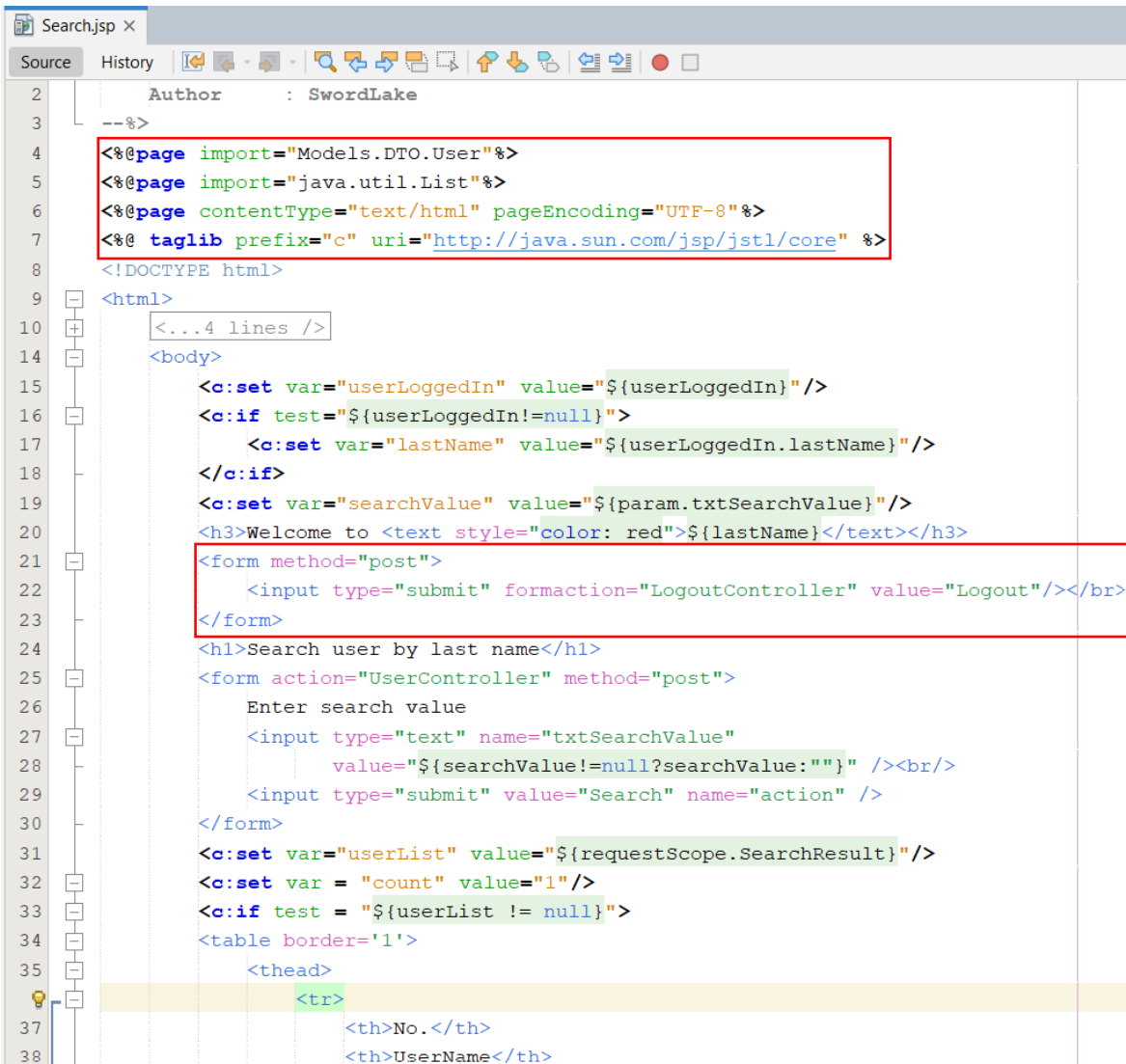


```

33      LastName <input type="text" placeholder="Enter last name" name="txtLastName"/>
34      <c:if test="${not empty error.lastNameError}">
35          <text style="color: red">${error.lastNameError}</text>
36      </c:if><br/>
37      <input type="checkbox" name="chkIsAdmin" disabled="true" />isAdmin<br/>
38      <input type="submit" value="Create" name="action" /></br>
39      <a href='Login.jsp'>Back</a></br>
40  </form>
41  </body>
42  </html>

```

• Search.jsp page



```

Source  History
Search.jsp x
2      Author      : SwordLake
3      --%>
4      <%@page import="Models.DTO.User"%>
5      <%@page import="java.util.List"%>
6      <%@page contentType="text/html" pageEncoding="UTF-8"%>
7      <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
8      <!DOCTYPE html>
9      <html>
10     <...4 lines />
11     <body>
12         <c:set var="userLoggedIn" value="${userLoggedIn}" />
13         <c:if test="${userLoggedIn!=null}">
14             <c:set var="lastName" value="${userLoggedIn.lastName}" />
15         </c:if>
16         <c:set var="searchValue" value="${param.txtSearchValue}" />
17         <h3>Welcome to <text style="color: red">${lastName}</text></h3>
18         <form method="post">
19             <input type="submit" formaction="LogoutController" value="Logout"/></br>
20         </form>
21         <h1>Search user by last name</h1>
22         <form action="UserController" method="post">
23             Enter search value
24             <input type="text" name="txtSearchValue"
25                 value="${searchValue!=null?searchValue:''}" /><br/>
26             <input type="submit" value="Search" name="action" />
27         </form>
28         <c:set var="userList" value="${requestScope.SearchResult}" />
29         <c:set var="count" value="1" />
30         <c:if test="${userList != null}">
31             <table border="1">
32                 <thead>
33                     <tr>
34                         <th>No.</th>
35                         <th>UserName</th>

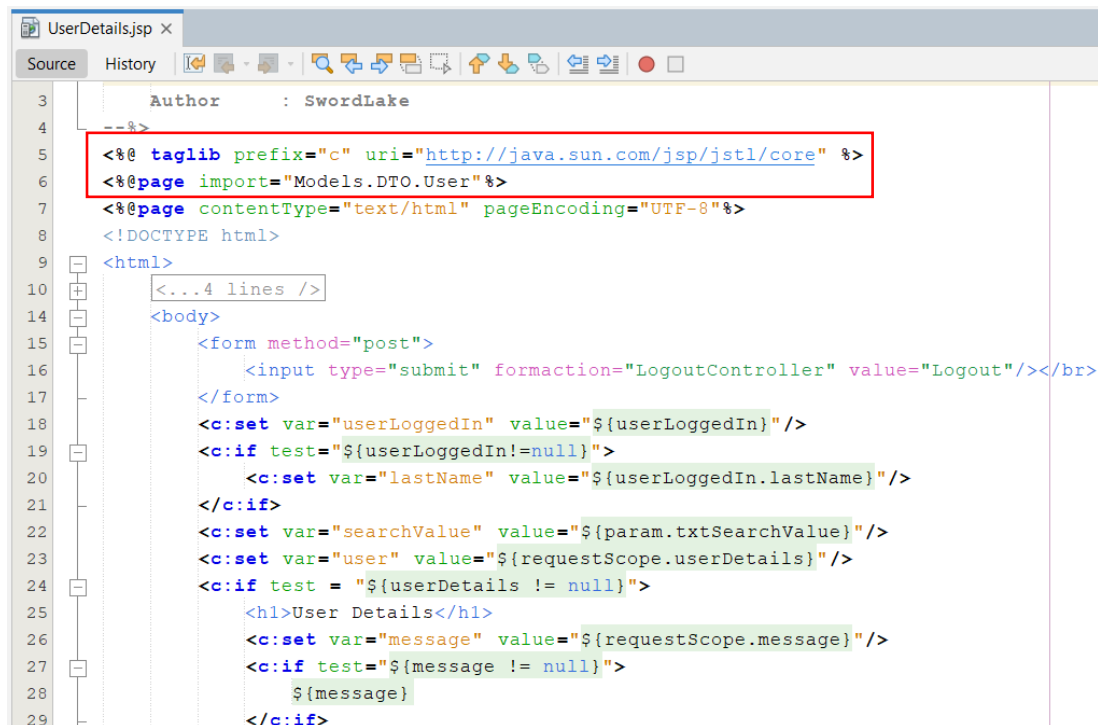
```

```

39         <th>Password</th>
40         <th>LastName</th>
41         <th>Role</th>
42         <th colspan="2">Action</th>
43     </tr>
44 </thead>
45 <tbody>
46     <c:forEach var="user" items="${userList}">
47         <tr>
48             <td>${count}</td>
49             <td>${user.userName}</td>
50             <td>${user.password}</td>
51             <td>${user.lastName}</td>
52             <td><input type="checkbox" <c:if test="${user.isAdmin==true}">checked</c:if>/></td>
53             <td><a href='UserController?action=Delete&UserName=${user.userName}&txtSearchValue=${searchValue}'>
54                 Delete
55             </td>
56             <td><a href='UserController?action=Details&UserName=${user.userName}&txtSearchValue=${searchValue}'>
57                 View
58             </td>
59         </tr>
60         <c:set var = "count" value="${count+1}" />
61     </c:forEach>
62 </tbody>
63 </table>
64
65     <c:set var="message" value="${requestScope.message}" />
66     <c:if test="${message != null}">
67         ${message}
68     </c:if>
69     <h3>Number of users: ${userList.size()}</h3>
70 </c:if>
71     <c:if test = "${searchValue != null}">
72         <c:if test = "${userList.size()==0 || userList== null}">
73             <h3>No users were found.</h3>
74         </c:if>
75     </c:if>
76 </body>
</html>

```

• UserDetails.jsp page



```

3      Author      : SwordLake
4      -->
5      <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
6      <%@page import="Models.DTO.User"%>
7      <%@page contentType="text/html" pageEncoding="UTF-8"%>
8      <!DOCTYPE html>
9      <html>
10         <...4 lines />
11     <body>
12         <form method="post">
13             <input type="submit" formaction="LogoutController" value="Logout"/></br>
14         </form>
15         <c:set var="userLoggedIn" value="${userLoggedIn}" />
16         <c:if test="${userLoggedIn!=null}">
17             <c:set var="lastName" value="${userLoggedIn.lastName}" />
18         </c:if>
19         <c:set var="searchValue" value="${param.txtSearchValue}" />
20         <c:set var="user" value="${requestScope.userDetails}" />
21         <c:if test = "${userDetails != null}">
22             <h1>User Details</h1>
23             <c:set var="message" value="${requestScope.message}" />
24             <c:if test="${message != null}">
25                 ${message}
26             </c:if>

```

```

30 <form action="UserController" method="post">
31 <c:set var="error" value="${requestScope.ErrorDetails}"/>
32 UserName <input type="text" readonly="true" name="txtUserName" value='${user.userName}'/>
33 <c:if test="${not empty error.userNameError}">
34 <text style="color: red">${error.userNameError}</text>
35 </c:if><br/>
36 Password <input type="password" name="txtPassword" value='${user.password}'/>
37 <c:if test="${not empty error.passwordError}">
38 <text style="color: red">${error.passwordError}</text>
39 </c:if><br/>
40 LastName <input type="text" name="txtLastName" value='${user.lastName}'/>
41 <c:if test="${not empty error.lastNameError}">
42 <text style="color: red">${error.lastNameError}</text>
43 </c:if><br/>
44 <input type="hidden" value='${searchValue}' name="txtSearchValue"/>
45 <input type="checkbox" name="chkIsAdmin"
46 <c:if test="${user.isAdmin==true}">checked</c:if>
47 <c:if test="${userLoggedIn.isAdmin==false}">disabled</c:if>>isAdmin
48 </input><br/>
49 <input type="submit" value="Update" name="action" /></br>
50 </form>
51 <c:if test="${userLoggedIn.isAdmin}" >
52 <a href='UserController?action=Search&txtSearchValue=${searchValue}'>Back</a></br>
53 </c:if>
54 <c:if test="${!userLoggedIn.isAdmin}" >
55 <a href='Login.jsp'>Back</a></br>
56 </c:if>
57 </c:if>
58 </body>
59 </html>

```

Step 06. Open the **web.xml** in the WEB-INF folder then update as the follows:



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="3.1" xmlns="http://xmlns.jcp.
3 <session-config>
4 <session-timeout>
5 30
6 </session-timeout>
7 </session-config>
8 <welcome-file-list>
9 <welcome-file>Login.jsp</welcome-file>
10 </welcome-file-list>
11 </web-app>

```

Step 07. Right-click on **Lab06_ManageUsersByMVC2_JSTL** project, select Run to run the web app then test all functions.