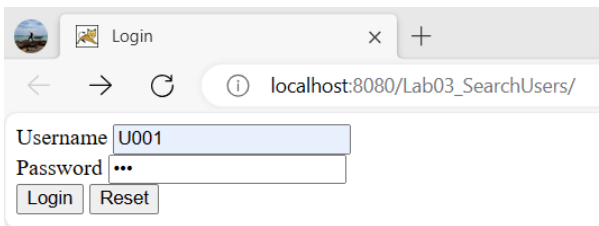
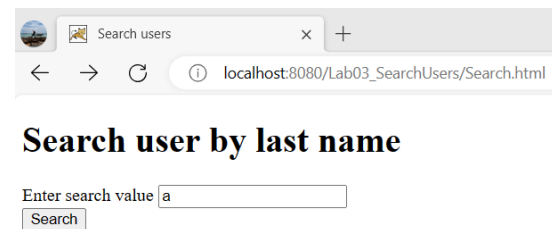
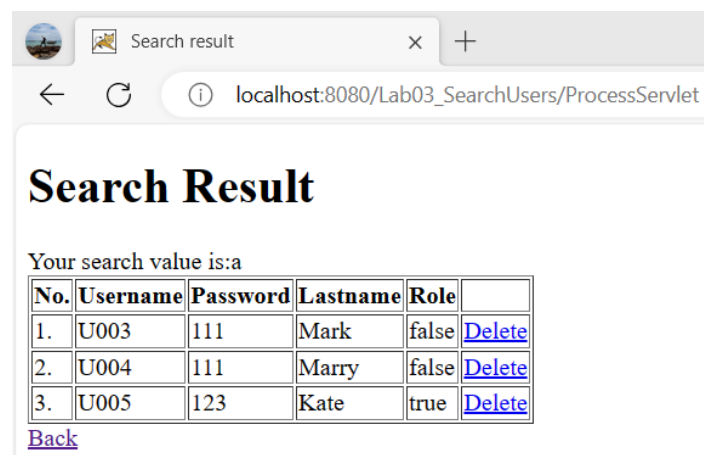


Building an Java Web Application

Using HTML, Servlet, JDBC & RequestDispatcher

Requirements

Create a java web application (using HTML, Servlets and JDBC) to perform functions as login and search users.

Search Result

Your search value is:a

No.	Username	Password	Lastname	Role	
1.	U003	111	Mark	false	Delete
2.	U004	111	Marry	false	Delete
3.	U005	123	Kate	true	Delete

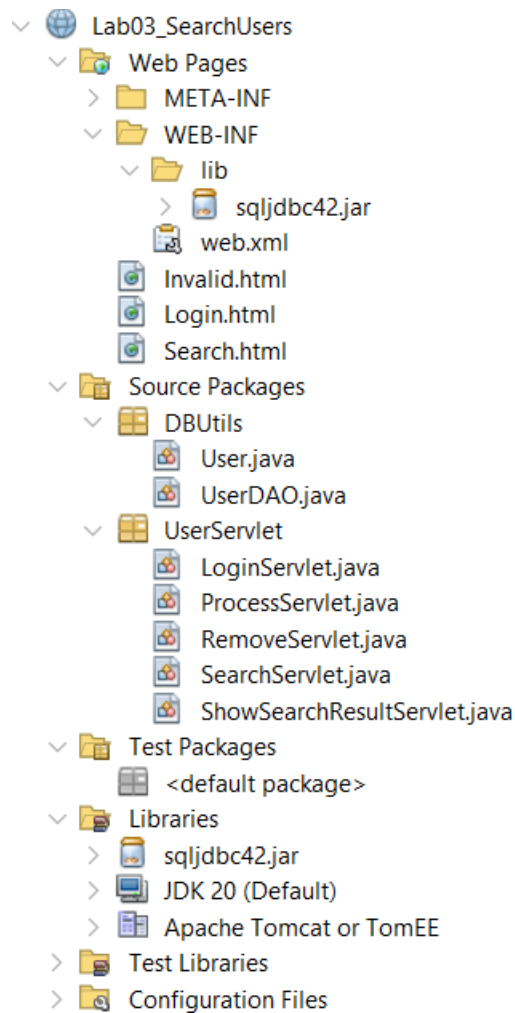
[Back](#)

Lab Objectives

In this lab, you will be:

- Create Java web project using the NetBean IDE
- Create the web pages by HTML
- Apply passing data from HTML to Servlet by GET and POST method
- 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 application

Step 01. Create a Java Web application named Lab03_SearchUsers is structured as follows:



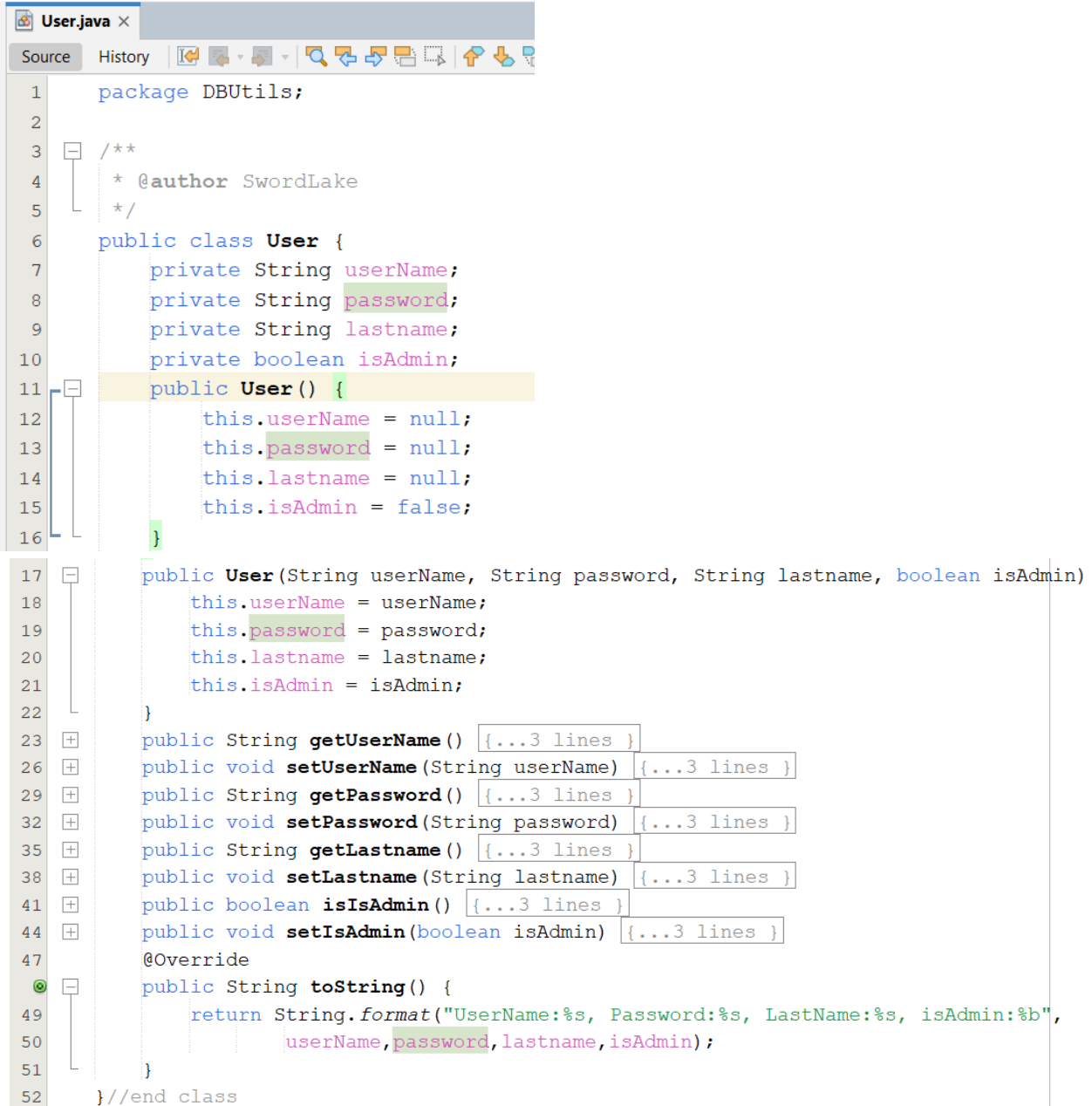
-Open SQL Server Management Studio and create a database named **SampleDB** has a table named **Registration** as follows:

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 **DBUtils** package as the follows:

- **User.java**



```

1  package DBUtils;
2
3  /**
4   * @author SwordLake
5   */
6  public class User {
7      private String userName;
8      private String password;
9      private String lastname;
10     private boolean isAdmin;
11     public User () {
12         this.userName = null;
13         this.password = null;
14         this.lastname = null;
15         this.isAdmin = false;
16     }
17     public User(String userName, String password, String lastname, boolean isAdmin) {
18         this.userName = userName;
19         this.password = password;
20         this.lastname = lastname;
21         this.isAdmin = isAdmin;
22     }
23     public String getUsername () { ...3 lines }
24     public void setUsername (String username) { ...3 lines }
25     public String getPassword () { ...3 lines }
26     public void setPassword (String password) { ...3 lines }
27     public String getLastName () { ...3 lines }
28     public void setLastName (String lastname) { ...3 lines }
29     public boolean isIsAdmin () { ...3 lines }
30     public void setIsAdmin (boolean isAdmin) { ...3 lines }
31     @Override
32     public String toString () {
33         return String.format("UserName:%s, Password:%s, LastName:%s, isAdmin:%b",
34             userName, password, lastname, isAdmin);
35     }
36 }
37 //end class

```

- **UserDAO.java**

```

UserDAO.java x
Source History
3  import java.sql.Connection;
4  import java.sql.DriverManager;
5  import java.sql.PreparedStatement;
6  import java.sql.ResultSet;
7  import java.sql.SQLException;
8  import java.util.ArrayList;
9  import java.util.List;
10 /**
11  * @author SwordLake
12  */
13 public class UserDAO {
14     // <editor-fold defaultstate="collapsed" desc="getConnection Method">
15     public static Connection getConnection() throws Exception {
16         try {
17             Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
18             String connectionString = "jdbc:sqlserver://localhost:1433;database=SampleDB";
19             //SQL Server Authentication
20             Connection cnn = DriverManager.getConnection(connectionString, "sa", "123");
21             return cnn;
22         } catch (ClassNotFoundException | SQLException ex) {
23             throw ex;
24         }
25     }
26     // </editor-fold>
27
28     // <editor-fold defaultstate="collapsed" desc="login Method">
29     public User login(String userName, String password) throws Exception {
30         User user = null;
31         Connection cnn = null;
32         PreparedStatement preStm = null;
33         ResultSet rs = null;
34         String lastName;
35         boolean isAdmin;
36         try {
37             cnn = getConnection();
38             String sql =
39                 "select LastName,isAdmin from Registration where [UserName]=? and [Password]=?";
40             preStm = cnn.prepareStatement(sql);
41             preStm.setString(1, userName);
42             preStm.setString(2, password);
43             rs = preStm.executeQuery();
44             while (rs.next()) {
45                 lastName = rs.getString(1);
46                 isAdmin = rs.getBoolean(2);
47                 user = new User(userName, password, lastName, isAdmin);
48             } //end while
49         } catch (Exception ex) {
50             throw ex;
51         }
52     }
53     // </editor-fold>

```

```

51         } finally {
52             if (rs != null) {
53                 rs.close();
54             }
55             if (preStm != null) {
56                 preStm.close();
57             }
58             if (cnn != null) {
59                 cnn.close();
60             }
61         }
62         return user;
63     } //end login
64 // </editor-fold>

66 // <editor-fold defaultstate="collapsed" desc="searchUserByLastName Method">
67 public List<User> searchUserByLastName(String searchValue) throws Exception {
68     String userName, password, lastName;
69     boolean isAdmin;
70     Connection cnn = null;
71     PreparedStatement preStm = null;
72     ResultSet rs = null;
73     List<User> userList = new ArrayList();
74     try {
75         cnn = getConnection();
76         String sql = "select UserName,Password,LastName,isAdmin from Registration "+
77             " where LastName like ?";
78         preStm = cnn.prepareStatement(sql);
79         preStm.setString(1, "%" + searchValue + "%");
80         rs = preStm.executeQuery();
81         while (rs.next()) {
82             userName = rs.getString(1);
83             password = rs.getString(2);
84             lastName = rs.getString(3);
85             isAdmin = rs.getBoolean(4);
86             User user = new User(userName, password, lastName, isAdmin);
87             userList.add(user);
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     if (userList.isEmpty()) {
103         return null;
104     }
105     return userList;
106 } //end searchUserByLastName
107 // </editor-fold>

```

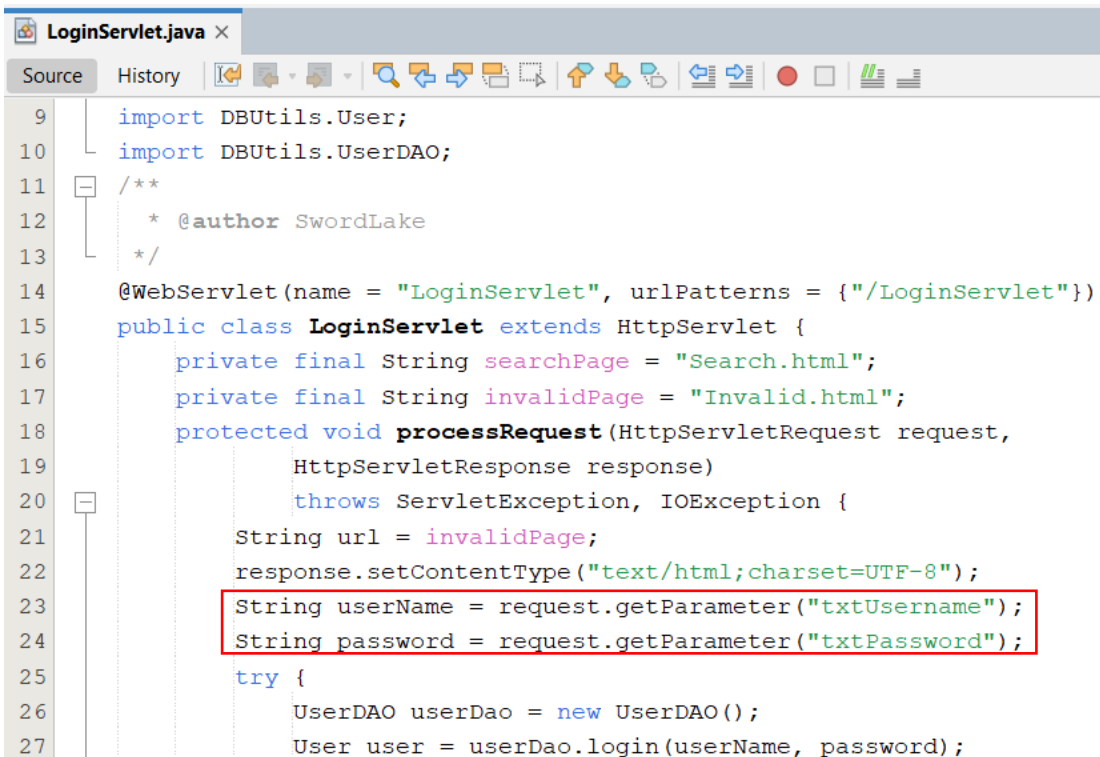
```

109 // <editor-fold defaultstate="collapsed" desc="deleteUser Method">
110 public boolean deleteUser(String userName) throws Exception {
111     PreparedStatement preStm = null;
112     Connection cnn = null;
113     try {
114         cnn = getConnection();
115         String sql = "delete Registration Where UserName=?";
116         preStm = cnn.prepareStatement(sql);
117         preStm.setString(1, userName);
118         return preStm.executeUpdate() > 0;
119     } catch (Exception ex) {
120         throw ex;
121     } finally {
122         if (preStm != null) {
123             preStm.close();
124         }
125         if (cnn != null) {
126             cnn.close();
127         }
128     }
129 } //end deleteUser
130 // </editor-fold>
131 } //end class

```

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

- **LoginServlet.java**



```

9 import DBUtils.User;
10 import DBUtils.UserDAO;
11 /**
12  * @author SwordLake
13  */
14 @WebServlet(name = "LoginServlet", urlPatterns = {"/LoginServlet"})
15 public class LoginServlet extends HttpServlet {
16     private final String searchPage = "Search.html";
17     private final String invalidPage = "Invalid.html";
18     protected void processRequest(HttpServletRequest request,
19                                 HttpServletResponse response)
20         throws ServletException, IOException {
21         String url = invalidPage;
22         response.setContentType("text/html;charset=UTF-8");
23         String userName = request.getParameter("txtUsername");
24         String password = request.getParameter("txtPassword");
25         try {
26             UserDAO userDao = new UserDAO();
27             User user = userDao.login(userName, password);

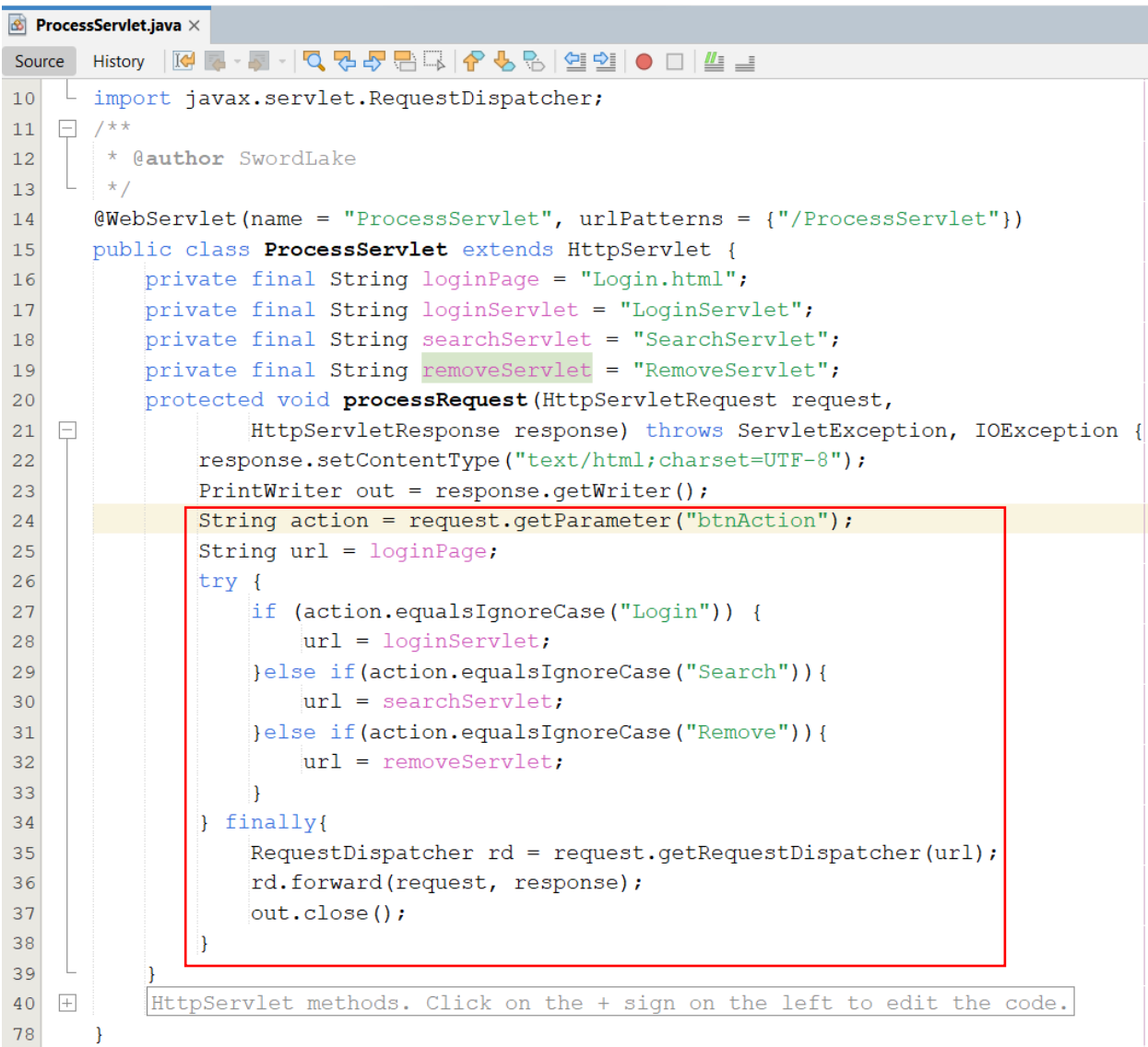
```

```

28         if (user != null) {
29             url = searchPage;
30         }
31         response.sendRedirect(url);
32     } catch (Exception ex) {
33         System.out.println(ex.getMessage());
34     }
35 } //end processRequest
36 + HttpServlet methods. Click on the + sign on the left to edit the code.
74 }

```

• ProcessServlet.java



```

ProcessServlet.java x
Source History
10 import javax.servlet.RequestDispatcher;
11 /**
12  * @author SwordLake
13  */
14 @WebServlet(name = "ProcessServlet", urlPatterns = {"/ProcessServlet"})
15 public class ProcessServlet extends HttpServlet {
16     private final String loginPage = "Login.html";
17     private final String loginServlet = "LoginServlet";
18     private final String searchServlet = "SearchServlet";
19     private final String removeServlet = "RemoveServlet";
20     protected void processRequest(HttpServletRequest request,
21         HttpServletResponse response) throws ServletException, IOException {
22         response.setContentType("text/html;charset=UTF-8");
23         PrintWriter out = response.getWriter();
24         String action = request.getParameter("btnAction");
25         String url = loginPage;
26         try {
27             if (action.equalsIgnoreCase("Login")) {
28                 url = loginServlet;
29             } else if (action.equalsIgnoreCase("Search")) {
30                 url = searchServlet;
31             } else if (action.equalsIgnoreCase("Remove")) {
32                 url = removeServlet;
33             }
34         } finally {
35             RequestDispatcher rd = request.getRequestDispatcher(url);
36             rd.forward(request, response);
37             out.close();
38         }
39     }
40 + HttpServlet methods. Click on the + sign on the left to edit the code.
78 }

```

• SearchServlet.java

```

SearchServlet.java x
Source History
15  /**...3 lines */
18  @WebServlet(name = "SearchServlet", urlPatterns = {"/SearchServlet"})
19  public class SearchServlet extends HttpServlet {
20      private final String searchPage = "Search.html";
21      private final String showSearchResultServlet = "ShowSearchResultServlet";
22      protected void processRequest (HttpServletRequest request, HttpServletResponse response)
23          throws ServletException, IOException {
24          String url, searchValue;
25          response.setContentType("text/html;charset=UTF-8");
26          PrintWriter out = response.getWriter();
27          url = searchPage;
28          try {
29              searchValue = request.getParameter("txtSearchValue");
30              if (!searchValue.isEmpty()) {
31                  UserDao userDao = new UserDao();
32                  List<User> userList = userDao.searchUserByLastName(searchValue);
33                  request.setAttribute("SearchResult", userList);
34                  url = showSearchResultServlet;
35              }
36          } catch (Exception ex) {
37              out.println("Error:" + ex.getMessage());
38          }
39          finally{
40              RequestDispatcher rd = request.getRequestDispatcher(url);
41              rd.forward(request, response);
42              out.close();
43          }
44      }
45      HttpServlet methods. Click on the + sign on the left to edit the code.
83  }

```

• ShowSearchResultServlet.java

```

ShowSearchResultServlet.java x
Source History
16  @WebServlet(name = "ShowSearchResultServlet", urlPatterns = {"/ShowSearchResultServlet"})
17  public class ShowSearchResultServlet extends HttpServlet {
18      protected void processRequest (HttpServletRequest request,
19          HttpServletResponse response)
20          throws ServletException, IOException {
21          String searchValue;
22          int count;
23          response.setContentType("text/html;charset=UTF-8");
24          try (PrintWriter out = response.getWriter()) {
25              out.println("<!DOCTYPE html>");
26              out.println("<html>");
27              out.println("<head>");
28              out.println("<title>Search result</title>");
29              out.println("</head>");
30              out.println("<body>");

```



```

31 out.println("<h1>Search Result</h1>");
32 searchValue = request.getParameter("txtSearchValue");
33 out.println("Your search value is:" + searchValue+"<br>");
34 List<User> result = (List<User>) request.getAttribute("SearchResult");
35 StringBuilder strResult = new StringBuilder();
36 if (result != null) {
37     strResult.append("<table border='1'>");
38     strResult.append("<thead>");
39     strResult.append("<tr>");
40     strResult.append("<th>No.</th>");
41     strResult.append("<th>Username</th>");
42     strResult.append("<th>Password</th>");
43     strResult.append("<th>Lastname</th>");
44     strResult.append("<th>Role</th>");
45     strResult.append("<th></th>");
46     strResult.append("</tr>");
47     strResult.append("</thead>");
48     strResult.append("<tbody>");
49     count = 0;
50     for (User user : result) {
51         strResult.append("<tr>");
52         strResult.append("<td>"+(++count)+"</td>");
53         strResult.append("<td>"+user.getUserName()+"</td>");
54         strResult.append("<td>"+user.getPassword()+"</td>");
55         strResult.append("<td>"+user.getLastname()+"</td>");
56         strResult.append("<td>"+user.isIsAdmin()+"</td>");
57         strResult.append("<td><a href='ProcessServlet?btnAction=Remove&&UserName='>");
58         strResult.append(user.getUserName()+"</td>");
59         strResult.append("Delete</td>");
60         strResult.append("</tr>");
61     }
62     strResult.append("</tbody>");
63     strResult.append("</table>");
64     out.println(strResult.toString());
65 }else{
66     out.println("<h3>No the users found.</h3>");
67 }
68 out.println("<a href='Search.html'>Back</a><br>");
69 out.println("</body>");
70 out.println("</html>");
71 }
72 }
73
111 }
112

```

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

• RemoveServlet.java

```

13 /**
14  * @author SwordLake
15  */
16 @WebServlet(name = "RemoveServlet", urlPatterns = {"/RemoveServlet"})
17 public class RemoveServlet extends HttpServlet {
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();

```

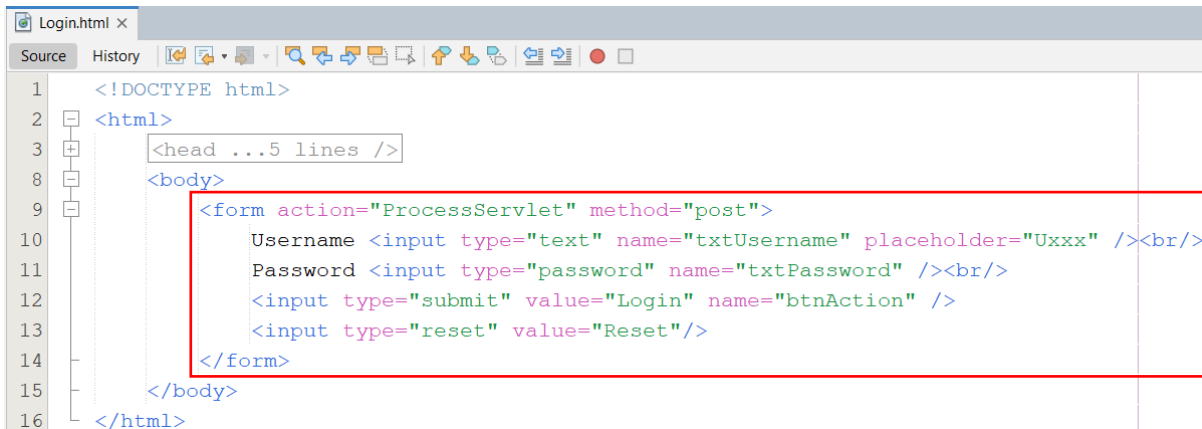
```

22     try {
23         String userName = request.getParameter("UserName");
24         if (!userName.isEmpty()) {
25             UserDao userDao = new UserDao();
26             if (userDao.deleteUser(userName) == true) {
27                 out.println(userName + " has been deleted successfully.<br>");
28             }
29             else{
30                 out.println("Something went wrong.<br>");
31             }
32             out.println("<a href='Search.html'>Back to search</a><br>");
33         }
34     } catch (Exception ex) {
35         out.println("Error:" + ex.getMessage());
36     }
37 }
38 + HttpServlet methods. Click on the + sign on the left to edit the code.
76 }

```

Step 04. Write HTML codes for the web pages as the follows:

- **Login.html page**

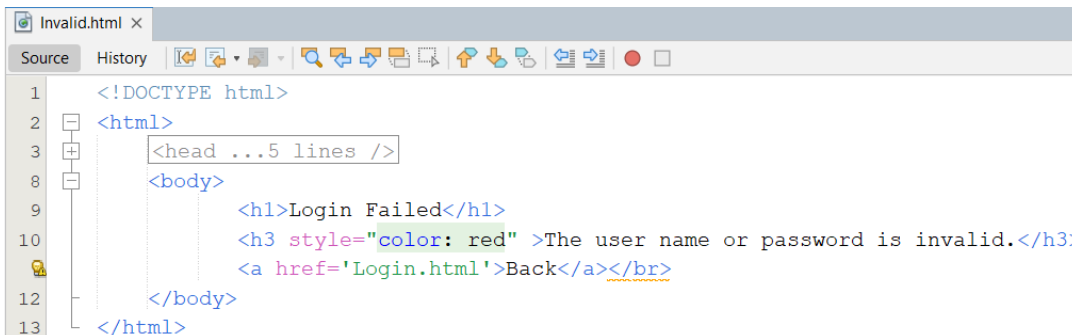


```

1 <!DOCTYPE html>
2 <html>
3   <head ...5 lines />
8   <body>
9     <form action="ProcessServlet" method="post">
10      Username <input type="text" name="txtUsername" placeholder="Uxxx" /><br/>
11      Password <input type="password" name="txtPassword" /><br/>
12      <input type="submit" value="Login" name="btnAction" />
13      <input type="reset" value="Reset" />
14    </form>
15  </body>
16 </html>

```

- **Invalid.html page**



```

1 <!DOCTYPE html>
2 <html>
3   <head ...5 lines />
8   <body>
9     <h1>Login Failed</h1>
10    <h3 style="color: red" >The user name or password is invalid.</h3>
11    <a href='Login.html'>Back</a><br>
12  </body>
13 </html>

```

- **Search.html page**

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Search users</title>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  </head>
8  <body>
9      <h1>Search user by last name</h1>
10     <form action="ProcessServlet" method="post">
11         Enter search value <input type="text" name="txtSearchValue"/><br/>
12         <input type="submit" value="Search" name="btnAction" />
13     </form>
14 </body>
15 </html>
  
```

Step 05. 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.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 **Lab03_SearchUsers** project, select Run to run the web app then test all functions.