



**UNIVERSITI MALAYSIA TERENGGANU**  
**FACULTY OF COMPUTER SCIENCE AND MATHEMATICS**

**CSM3023 – WEB BASED APPLICATION DEVELOPMENT**  
**LAB REPORT 2 - SERVLET: DATA SHARING AND DATABASE**  
**MANAGEMENT**

**PREPARED BY:**  
**HARINATUL MUFLIHUN BINTI HASNUL MUNAWAR**  
**( S67604 )**

**PREPARED FOR:**  
**DR. MOHAMAD NOR BIN HASSAN**

**BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING)**  
**WITH HONOURS**  
**SEMESTER II 2023/2024**

## TASK 1 DATA SHARING IN SERVLET

### Objective :

- To use servlet for request forwarding and data sharing.

### Login.html Code

```
Source History
1 <!DOCTYPE html>
2 <!--
3 Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
4 Click nbfs://nbhost/SystemFileSystem/Templates/JSP\_Servlet/Html.html to edit this template
5 -->
6 <html>
7 <head>
8 <title>Login Page</title>
9 <meta charset="UTF-8">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11 <style>
12     body {
13         background-color: black;
14         text-align: left;
15         color: white;
16         font-family: Arial, Helvetica, sans-serif;
17     }
18 </style>
19 </head>
20 <body>
21 <h1>Welcome to CSM3023</h1>
22 <p>Please insert your username and password</p>
23 <form name="login" id="login" action="LoginServlet" method="POST" autocomplete="off">
24     Username:<input name="txtUsername" type="text"><br>
25     Password:<input name="txtPassword" type="text"><br>
26     <br>
27     <input name="btnLogin" value="Login" type="submit">
28     <input name="btnReset" value="Reset" type="reset"><br>
29 </form>
30 <p><br>
31 </p>
32 </body>
33 </html>
```

## LoginServlet.java Code

```
Source History
1 import jakarta.servlet.RequestDispatcher;
2 import jakarta.servlet.ServletContext;
3 import java.io.IOException;
4 import java.util.HashMap;
5 import java.io.PrintWriter;
6 import jakarta.servlet.ServletException;
7 import jakarta.servlet.http.HttpServlet;
8 import jakarta.servlet.http.HttpServletRequest;
9 import jakarta.servlet.http.HttpServletResponse;
10
11 /**
12  * Harinatul Muflihun S67604
13  * Lab 2 Task 1
14  */
15 public class LoginServlet extends HttpServlet {
16
17     HashMap <String, String> users = new HashMap();
18
19     @Override
20     public void init() throws ServletException {
21         super.init();
22         users.put("Ali", "1234");
23         users.put("Ahmad", "4567");
24         users.put("Muthu", "8910");
25     }
26
27     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
28         throws ServletException, IOException {
29         response.setContentType("text/html;charset=UTF-8");
30
31         String username = request.getParameter("txtUsername");
32         String password = request.getParameter("txtPassword");
33
34         if (!username.equals("") && !password.equals(""))
35             && users.get(username).equals(password) {
36             request.setAttribute("userid", username);
37             ServletContext sc = getServletContext();
38             RequestDispatcher rd = sc.getRequestDispatcher("/AccountServlet");
39             rd.forward(request, response);
40         } else {
41             //avoid direct access to the servlet
42             RequestDispatcher rd = request.getRequestDispatcher("/login.html");
43             rd.forward(request, response);
44         }
45     }
46
47     HttpServlet methods. Click on the + sign on the left to edit the code.
48
49
50
51
52
53
54
55
56
57 }
```

## AccountServlet.java Code

```
Source History 
1  import java.io.IOException;
2  import java.util.HashMap;
3  import java.io.PrintWriter;
4  import jakarta.servlet.ServletException;
5  import jakarta.servlet.http.HttpServlet;
6  import jakarta.servlet.http.HttpServletRequest;
7  import jakarta.servlet.http.HttpServletResponse;
8
9  /**
10   * Harinatul Muflihun S67604
11   * Lab 2 Task 1
12   */
13  public class AccountServlet extends HttpServlet {
14
15      HashMap <String, String[]> account = new HashMap();
16
17      @Override
18      public void init() throws ServletException {
19          super.init();
20          account.put("Ali", new String[]{"31/01/2019: 2000.00", "28/02/2019: 3000.00"});
21          account.put("Ahmad", new String[]{"31/01/2019: 100.00", "28/02/2019: 5000.00"});
22          account.put("Muthu", new String[]{"31/01/2019: 1000", "28/02/2019: 2000"});
23      }
24
25      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
26          throws ServletException, IOException {
27          response.setContentType("text/html;charset=UTF-8");
28
29          String userid_login = (String)request.getAttribute("userid");
30
31          try (PrintWriter out = response.getWriter()) {
32              out.println("<!DOCTYPE html>");
33              out.println("<html>");
34              out.println("<head>");
35              out.println("<title>Servlet AccountServlet</title>");
36              out.println("</head>");
37              out.println("<body>");
38
39              if(account.get(userid_login) == null) {
40                  out.println("<h1>Sorry, no information found!</h1>");
41              }
42              else {
43                  out.println("<h1>Account status for: " + userid_login + "</h1>");
44                  for(String tempAcc: account.get(userid_login)) {
45                      out.println("<h2>" + tempAcc + "</h2>");
46                  }
47              }
48              out.println("</body>");
49              out.println("</html>");
50          }
51      }
52
53      

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


54
55      91
56      92  }
```

This is the output.

a) Output for Ahmad

localhost:8080/ServletDataSharing/

Bahan PdP Kimia SDS | kms maths SM025 | Channel

## Welcome to CSM3023

Please insert your username and password

Username:

Password:

Login Reset

localhost:8080/ServletDataSharing/Logi

Bahan PdP Kimia SDS | kms maths SM025 | Channel Yo

## Account status for: Ahmad

31/01/2019: 100.00

28/02/2019: 5000.00

b) Output for Ali

localhost:8080/ServletDataSharing/

Bahan PdP Kimia SDS | kms maths SM025 | Channel

## Welcome to CSM3023

Please insert your username and password

Username:

Password:

Login Reset

localhost:8080/ServletDataSharing/Logi

Bahan PdP Kimia SDS | kms maths SM025 | Cha

## Account status for: Ali

31/01/2019: 2000.00

28/02/2019: 3000.00

c) Output for Muthu

localhost:8080/ServletDataSharing/

Bahan PdP Kimia SDS | kms maths SM025 | Channel

## Welcome to CSM3023

Please insert your username and password

Username:

Password:

Login Reset

localhost:8080/ServletDataSharing/Logi

Bahan PdP Kimia SDS | kms maths SM025 | Channel \

## Account status for: Muthu

31/01/2019: 1000

28/02/2019: 2000

**Reflections :**

1. What have you learnt from this exercise?
  - I learnt to create the servlet, used HashMap and how to use a post method. HashMap is one way to store data. With HashMap, the data will efficiently retrieve and manipulate.
2. What are the common methods used in Java Servlet?
  - The command methods used in Java Servlet are Get method and Post method. But in task 1, the method that we used is Post method. This is because Post method for sensitive information and used to create or update a resource. For this case, amount account is a sensitive information and the data about amount account always needs to update.

## TASK 4 : USING SERVLETS FOR DATABASE CRUD OPERATIONS

### Objective :

- To program multiple servlets for manipulating the database.

### Index.html Code

```
Source History
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>User Management</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   </head>
8   <body>
9     <h1>Add New User</h1>
10    <form action="SaveServlet" method="post">
11      <table>
12        <tr><td>Username:</td><td><input type="text" name="name"/></td></tr>
13        <tr><td>Password:</td><td><input type="password" name="password"/></td></tr>
14        <tr><td>Role:</td><td>
15          <select name="role" style="width:150px">
16            <option>admin</option>
17            <option>user</option>
18          </select>
19        </td></tr>
20        <tr><td colspan="2"><input type="submit" value="Save User"/></td></tr>
21      </table>
22    </form>
23
24    <br/>
25    <a href="ViewServlet">view users</a>
26  </body>
27 </html>
```

## User.java

```
7  /**
8   *Harinatul Muflihun S67604
9   * Task 2 lab 2
10  */
11  public class User {
12
13      private int id;
14      private String username, password, role;
15
16      public int getId() {
17          return id;
18      }
19
20      public void setId(int id) {
21          this.id = id;
22      }
23
24      public String getUsername() {
25          return username;
26      }
27
28      public void setUsername(String username) {
29          this.username = username;
30      }
31
32      public String getPassword() {
33          return password;
34      }
35
36      public void setPassword(String password) {
37          this.password = password;
38      }
39
40      public String getRole() {
41          return role;
42      }
43
44      public void setRole(String role) {
45          this.role = role;
46      }
47  }
```

## UserDao.java

```
8  import java.util.*;
9  import java.sql.*;
10  /**
11   *Harinatul Muflihun S67604
12   * Task 2 lab 2
13  */
14  public class UserDao {
15
16      public static Connection getConnection() {
17          Connection con = null;
18          try {
19              Class.forName("com.mysql.jdbc.Driver");
20              con = DriverManager.getConnection("jdbc:mysql://localhost:3306/csm3023-lab2","root", "admin");
21          } catch (Exception e) {
22              System.out.println(e);
23          }
24          return con;
25      }
```



```

27 public static int save(User e) {
28     int status = 0;
29     try {
30         Connection con = UserDao.getConnection();
31         PreparedStatement ps = con.prepareStatement(
32             "insert into users(username,password,roles) values (?, ?, ?)");
33         ps.setString(1, e.getUsername());
34         ps.setString(2, e.getPassword());
35         ps.setString(3, e.getRole());
36
37         status = ps.executeUpdate();
38
39         con.close();
40     } catch (Exception ex) {
41         ex.printStackTrace();
42     }
43
44     return status;
45 }

```

```

47 public static int update(User e) {
48     int status = 0;
49     try {
50         Connection con = UserDao.getConnection();
51         PreparedStatement ps = con.prepareStatement(
52             "update users set username=?,password=?,roles=? where id=?");
53         ps.setString(1, e.getUsername());
54         ps.setString(2, e.getPassword());
55         ps.setString(3, e.getRole());
56         ps.setInt(4, e.getId());
57
58         status = ps.executeUpdate();
59
60         con.close();
61     } catch (Exception ex) {
62         ex.printStackTrace();
63     }
64
65     return status;
66 }

```

```

68 public static int delete(int id) {
69     int status = 0;
70     try {
71         Connection con = UserDao.getConnection();
72         PreparedStatement ps = con.prepareStatement("delete from users where id=?");
73         ps.setInt(1, id);
74         status = ps.executeUpdate();
75
76         con.close();
77     } catch (Exception e) {
78         e.printStackTrace();
79     }
80
81     return status;
82 }

```

```

84 public static User getUserById(int id) {
85     User e = new User();
86
87     try {
88         Connection con = UserDao.getConnection();
89         PreparedStatement ps = con.prepareStatement("select * from users where id=?");
90         ps.setInt(1, id);
91         ResultSet rs = ps.executeQuery();
92         if(rs.next()) {
93             e.setId(rs.getInt(1));
94             e.setUsername(rs.getString(2));
95             e.setPassword(rs.getString(3));
96             e.setRole(rs.getString(4));
97         }
98         con.close();
99     } catch (Exception ex) {
100         ex.printStackTrace();
101     }
102
103     return e;
104 }


```

```

106 public static List<User> getAllUsers() {
107     List<User> list = new ArrayList<User>();
108
109     try {
110         Connection con = UserDao.getConnection();
111         PreparedStatement ps = con.prepareStatement("select * from users");
112         ResultSet rs = ps.executeQuery();
113         while (rs.next()) {
114             User e = new User();
115             e.setId(rs.getInt(1));
116             e.setUsername(rs.getString(2));
117             e.setPassword(rs.getString(3));
118             e.setRole(rs.getString(4));
119             list.add(e);
120         }
121         con.close();
122     } catch (Exception e) {
123         e.printStackTrace();
124     }
125
126     return list;
127 }
128 }

```

## SaveServlet.java

```
Source History 
1  package com.mycompany.crudusingservlet;
2
3  /*
4   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
5   * Click nbfs://nbhost/SystemFileSystem/Templates/JSP\_Servlet/Servlet.java to edit this template
6   */
7
8  import java.io.IOException;
9  import java.io.PrintWriter;
10 import jakarta.servlet.ServletException;
11 import jakarta.servlet.http.HttpServlet;
12 import jakarta.servlet.http.HttpServletRequest;
13 import jakarta.servlet.http.HttpServletResponse;
14
15 /**
16  *Harinatul Muflihun S67604
17  * Task 2 lab 2
18  */
19 public class SaveServlet extends HttpServlet {
20
21     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
22         throws ServletException, IOException {
23         response.setContentType("text/html;charset=UTF-8");
24         PrintWriter out = response.getWriter();
25
26         String name = request.getParameter("name");
27         String password = request.getParameter("password");
28         String role = request.getParameter("role");
29
30         User e = new User();
31         e.setUsername(name);
32         e.setPassword(password);
33         e.setRole(role);
34
35         int status = UserDao.save(e);
36         if(status > 0) {
37             out.print("<p> Record saved successfully!</p>");
38             request.getRequestDispatcher("index.html").include(request, response);
39         }else {
40             out.println("Sorry! unable to save record");
41         }
42
43         out.close();
44     }
45
46     

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


47
48     84
49     85
50 }
```

## EditServlet.java

```
8 import java.io.IOException;
9 import java.io.PrintWriter;
10 import jakarta.servlet.ServletException;
11 import jakarta.servlet.http.HttpServlet;
12 import jakarta.servlet.http.HttpServletRequest;
13 import jakarta.servlet.http.HttpServletResponse;
14 /**
15  *Harinatul Muflihun S67604
16  * Task 2 lab 2
17  */
18 public class EditServlet extends HttpServlet {
19
20     /**
21      * Processes requests for both HTTP GET and POST
22      * methods.
23      *
24      * @param request servlet request
25      * @param response servlet response
26      * @throws ServletException if a servlet-specific error occurs
27      * @throws IOException if an I/O error occurs
28      */
29     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
30         throws ServletException, IOException {
31         response.setContentType("text/html");
32         PrintWriter out=response.getWriter();
33         out.println("<h1>Update User</h1>");
34         String sid=request.getParameter("id");
35         int id=Integer.parseInt(sid);
36
37         User e = UserDao.getUserById(id);
38
39         out.println("<form action='EditServlet2' method='post'>");
40         out.println("<table>");
41         out.println("<tr><td></td><td><input type='hidden' name='id' value='"
42             +e.getId()+"'/></td></tr>");
43         out.println("<tr><td>Name:</td><td><input type='text' name='username' value='"
44             +e.getUsername()+"'/></td></tr>");
45         out.println("<tr><td>Password:</td><td><input type='password' name='password' value='"
46             +e.getPassword()+"'/></td></tr>");
47         out.println("<tr><td>Role:</td><td>");
48         out.println("<select name='role' style='width:150px'>");
49         out.println("<option> admin </option>");
50         out.println("<option> user </option>");
51         out.println("</select>");
52         out.println("</td></tr>");
53         out.println("<tr><td colspan='2'><input type='submit' value='Edit & Save '/></td></tr>");
54         out.println("</table>");
55         out.println("</form>");
56
57         out.close();
58     }
59
60     /**
61      * HttpServlet methods. Click on the + sign on the left to edit the code.
62      */
63 }
```

## EditServlet2.java

```
8 import java.io.IOException;
9 import java.io.PrintWriter;
10 import jakarta.servlet.ServletException;
11 import jakarta.servlet.http.HttpServlet;
12 import jakarta.servlet.http.HttpServletRequest;
13 import jakarta.servlet.http.HttpServletResponse;
14
15 /**
16  *Harinatul Muflihun S67604
17  * Task 2 lab 2
18  */
19 public class EditServlet2 extends HttpServlet {
20
21     /**
22      * Processes requests for both HTTP GET and POST
23      * methods.
24      *
25      * @param request servlet request
26      * @param response servlet response
27      * @throws ServletException if a servlet-specific error occurs
28      * @throws IOException if an I/O error occurs
29      */
30     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
31         throws ServletException, IOException {
32         response.setContentType("text/html");
33         PrintWriter out=response.getWriter();
34
35         int id = Integer.parseInt(request.getParameter("id"));
36         String name = request.getParameter("username");
37         String password = request.getParameter("password");
38         String role = request.getParameter("role");
39
40         User e = new User();
41         e.setId(id);
42         e.setUsername(name);
43         e.setPassword(password);
44         e.setRole(role);
45
46         int status = UserDao.update(e);
47         if (status > 0) {
48             out.print("<p>Record saved successfully!</p>");
49             request.getRequestDispatcher("index.html").include(request, response);
50         } else {
51             out.println("Sorry! unable to save record");
52         }
53
54         out.close();
55     }
56
57     

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


58
59     }
60 }
```

## ViewServlet.java

```
8 import java.io.IOException;
9 import java.io.PrintWriter;
10 import jakarta.servlet.ServletException;
11 import jakarta.servlet.http.HttpServlet;
12 import jakarta.servlet.http.HttpServletRequest;
13 import jakarta.servlet.http.HttpServletResponse;
14 import java.util.List;
15 /**
16  *Harinatul Muflihun S67604
17  * Task 2 lab 2
18  */
19 public class ViewServlet extends HttpServlet {
20
21     /**
22      * Processes requests for both HTTP GET and POST
23      * methods.
24      *
25      * @param request servlet request
26      * @param response servlet response
27      * @throws ServletException if a servlet-specific error occurs
28      * @throws IOException if an I/O error occurs
29      */
30     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
31         throws ServletException, IOException {
32         response.setContentType("text/html;charset=UTF-8");
33         PrintWriter out = response.getWriter();
34         out.println("<a href='index.html'>Add New User</a>");
35         out.println("<h1>User List</h1>");
36
37         List<User> list = UserDao.getAllUsers();
38
39         out.print("<table border='1' width='100%'>");
40         out.print("<tr><th>Id</th><th>Name</th><th>Password</th><th>Role</th>"
41             + "<th>Edit</th><th>Delete</th></tr>");
42         for (User e: list) {
43             out.print("<tr><td>" + e.getId() + "</td><td>" + e.getUsername() + "</td><td>"
44                 + e.getPassword() + "</td><td>" + e.getRole() + "</td><td><a href='EditServlet?id="
45                 + e.getId() + "'>edit</a></td> <td><a href='DeleteServlet?id="
46                 + e.getId() + "'>delete</a></td></tr>");
47         }
48         out.print("</table>");
49
50         out.close();
51     }
52
53     HttpServlet methods. Click on the + sign on the left to edit the code.
54
55 }
```

## DeleteServlet.java

```
8  import java.io.IOException;
9  import java.io.PrintWriter;
10 import jakarta.servlet.ServletException;
11 import jakarta.servlet.http.HttpServlet;
12 import jakarta.servlet.http.HttpServletRequest;
13 import jakarta.servlet.http.HttpServletResponse;
14
15 /**
16  *Harinatul Muflihun S67604
17  * Task 2 lab 2
18  */
19 public class DeleteServlet extends HttpServlet {
20
21     /**
22      * Processes requests for both HTTP GET and POST
23      * methods.
24      *
25      * @param request servlet request
26      * @param response servlet response
27      * @throws ServletException if a servlet-specific error occurs
28      * @throws IOException if an I/O error occurs
29      */
30     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
31         throws ServletException, IOException {
32         String sid = request.getParameter("id");
33         int id = Integer.parseInt(sid);
34         UserDao.delete(id);
35         response.sendRedirect("ViewServlet");
36     }
37
38     // HttpServlet methods. Click on the + sign on the left to edit the code.
39
40 }
76
77 }
```

The output

a) Add new user

### Add New User

Username:

Password:  

Role:

[view users](#)

- This is the output shown when the user success register.

Record saved successfully!

## Add New User

Username:

Password:

Role:

[view users](#)

- b) Output for view users

[Add New User](#)

## User List

Id	Name	Password	Role	Edit	Delete
1	ahmad	1234	user	<a href="#">edit</a>	<a href="#">delete</a>
6	lily	2345	user	<a href="#">edit</a>	<a href="#">delete</a>
7	ahmad	12345	admin	<a href="#">edit</a>	<a href="#">delete</a>

- c) Output when click the edit button.

## Update User

Name:

Password:

Role:

- The result after I edit the Id=1, name=Sufian.

[Add New User](#)

## User List

Id	Name	Password	Role	Edit	Delete
1	Sufian	1234	admin	<a href="#">edit</a>	<a href="#">delete</a>
6	lily	2345	user	<a href="#">edit</a>	<a href="#">delete</a>
7	ahmad	12345	admin	<a href="#">edit</a>	<a href="#">delete</a>

- d) Output for deleted users.

[Add New User](#)

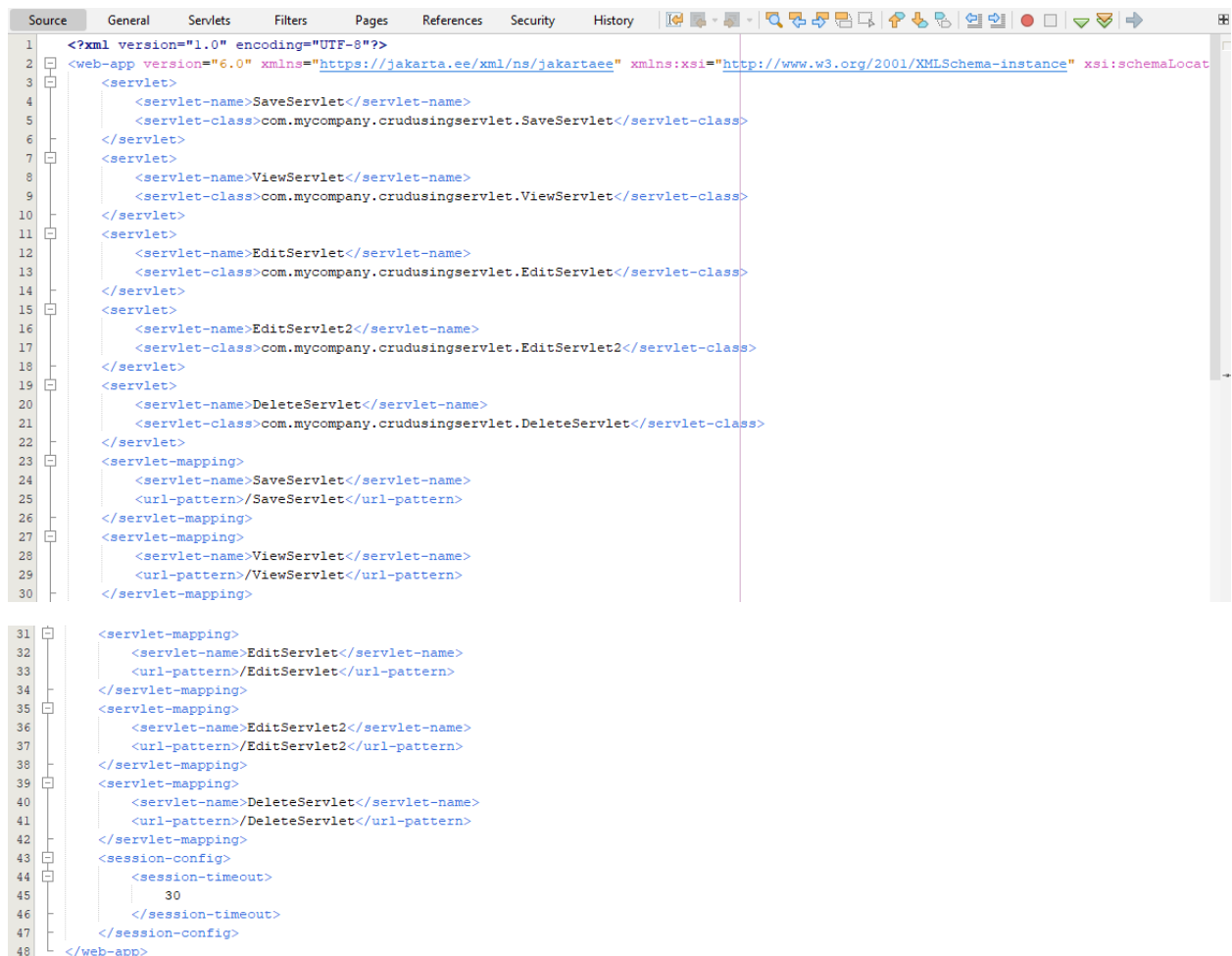
## User List

Id	Name	Password	Role	Edit	Delete
1	Sufian	1234	admin	<a href="#">edit</a>	<a href="#">delete</a>
6	lily	2345	user	<a href="#">edit</a>	<a href="#">delete</a>



## Reflection :

1. What is the name of the Java Library that you need to import before coding the web application with database operations?
  - I need to import the MySQL JDBC Driver Library. With this MySQL JDBC, we can connect to the database for CRUD which is create, read, update and delete all the data and perform all the database operation.
2. Which folder keeps the web.xml file? Copy the contents of the file and explain in brief the tags included such as <servlet-name><servlet-class><servlet-mapping>. etc.
  - The folder that keeps the web.xml file is a WEB-INF folder.

A screenshot of an IDE window showing the contents of a web.xml file. The file is an XML document with a root element <web-app>. It defines several servlets: SaveServlet, ViewServlet, EditServlet, EditServlet2, and DeleteServlet, each with a specific class name. It also includes mappings for each servlet to a URL pattern. Finally, it has a session-config section with a session-timeout of 30. The code is as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="6.0" xmlns="https://jakarta.ee/xml/ns/jakartaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd">
3   <servlet>
4     <servlet-name>SaveServlet</servlet-name>
5     <servlet-class>com.mycompany.crudusingservlet.SaveServlet</servlet-class>
6   </servlet>
7   <servlet>
8     <servlet-name>ViewServlet</servlet-name>
9     <servlet-class>com.mycompany.crudusingservlet.ViewServlet</servlet-class>
10  </servlet>
11  <servlet>
12    <servlet-name>EditServlet</servlet-name>
13    <servlet-class>com.mycompany.crudusingservlet.EditServlet</servlet-class>
14  </servlet>
15  <servlet>
16    <servlet-name>EditServlet2</servlet-name>
17    <servlet-class>com.mycompany.crudusingservlet.EditServlet2</servlet-class>
18  </servlet>
19  <servlet>
20    <servlet-name>DeleteServlet</servlet-name>
21    <servlet-class>com.mycompany.crudusingservlet.DeleteServlet</servlet-class>
22  </servlet>
23  <servlet-mapping>
24    <servlet-name>SaveServlet</servlet-name>
25    <url-pattern>/SaveServlet</url-pattern>
26  </servlet-mapping>
27  <servlet-mapping>
28    <servlet-name>ViewServlet</servlet-name>
29    <url-pattern>/ViewServlet</url-pattern>
30  </servlet-mapping>
31  <servlet-mapping>
32    <servlet-name>EditServlet</servlet-name>
33    <url-pattern>/EditServlet</url-pattern>
34  </servlet-mapping>
35  <servlet-mapping>
36    <servlet-name>EditServlet2</servlet-name>
37    <url-pattern>/EditServlet2</url-pattern>
38  </servlet-mapping>
39  <servlet-mapping>
40    <servlet-name>DeleteServlet</servlet-name>
41    <url-pattern>/DeleteServlet</url-pattern>
42  </servlet-mapping>
43  <session-config>
44    <session-timeout>
45      30
46    </session-timeout>
47  </session-config>
48 </web-app>
```

- <?xml version="1.0" encoding="UTF-8"?> (This explain that the file is XML with version 1.0 encoded using UTF-8.)
- <web-app version="6.0" xmlns="https://jakarta.ee/xml/ns/jakartaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee

[https://jakarta.ee/xml/ns/jakartaee/web-app\\_6\\_0.xsd](https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd)"> ( This web-app refer to the XML file will represent in Java Web Application and servlet API version 6.0 )

- <servlet> define a servlet.
- <servlet-name> specific name of servlet
- <servlet-class> specific package for servlet
- <servlet-mapping> this refers to maps where notices the web container which servlet need to handle.
- <servlet-name> this refers to servlet name in servlet maps.
- <url-pattern> this refers to servlet name for servlet response.
- <session-config> configures session-related settings for the application.
- <session-timeout> this refers to timeout maximum in minutes between client request and web server.

3. Define the usage of Data Access Object (DAO) servlet. How it eases the business process in your servlet-based web application?

- Data Access Object (DAO) function is to create and end connection to the data source. It also performs a CRUD operation which is create, read, update, delete all the data and perform all the database operation.
- DAO eases the business process in servlet-based web application because it can separate the data access login from the servlets responsible for handle HTTP requests and generate respond. The separate make the code easy to understand.