



UNIVERSITI MALAYSIA TERENGGANU

Faculty of Computer Sciences and Mathematics

Web Based Application Development

CSM3023

Lab Report 2

Prepared for:

Sir. Mohd Arizal Shamsil bin Mat Rifin

Prepared by:

Ahmad Afif Syahmi bin Ahmad Rozali (S65526)

3rd April 2024

Bachelor of Computer Science (Mobile Computing) with Honors

Semester II 2023/2024

Task 1 : Data Sharing in Servlet

- Objective: To use servlet for request forwarding and data sharing by writing a login form and a servlet to authenticate a user.
- Code:

```
<!DOCTYPE html>
<!--
    This is a login page.
-->
<html>
    <head>
        <title>Login Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <style>
            body {
                background-color: black;
                text-align: left;
                color: white;
                font-family: Arial, Helvetica, sans-serif;
            }
        </style>
    </head>
    <body>

        <h1>Welcome to CSM3023</h1>
        <p>Please insert your username and password</p>
        <form name="login" id="login" action="LoginServlet"
method="POST" autocomplete="off">
            Username : <input type="text" name="txtUsername"><br>
            Password : <input type="text" name="txtPassword"><br>
            <br>
            <input type="submit" value="Login" name="btnLogin">
            <input type="reset" value="Reset" name="txtReset"><br>
        </form>
        <p>
            <br>
        </p>

    </body>
</html>
```

```
import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletContext;
import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
```

```

import java.util.HashMap;

/**
 *
 * @author Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class LoginServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and
     * POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */

    HashMap<String, String> users = new HashMap();

    @Override
    public void init() throws ServletException {
        super.init();
        users.put("Ali", "1234");
        users.put("Ahmad", "4567");
        users.put("Muthu", "8910");
    }

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        String username = request.getParameter("txtUsername");
        String password = request.getParameter("txtPassword");

        if (!username.equals("") && !password.equals("")
            && users.get(username).equals(password)) {
            request.setAttribute("userid", username);
            ServletContext sc = getServletContext();
            RequestDispatcher rd =
                sc.getRequestDispatcher("/AccountServlet");
            rd.forward(request, response);
        }
        else {
            // avoid direct access to the servlet
            RequestDispatcher rd =
                request.getRequestDispatcher("/login.html");
            rd.forward(request, response);
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet
    methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP GET method.
     *
     * @param request servlet request

```

```

        * @param response servlet response
        * @throws ServletException if a servlet-specific error occurs
        * @throws IOException if an I/O error occurs
        */
        @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
            throws ServletException, IOException {
            processRequest(request, response);
        }

        /**
        * Handles the HTTP <code>POST</code> method.
        *
        * @param request servlet request
        * @param response servlet response
        * @throws ServletException if a servlet-specific error occurs
        * @throws IOException if an I/O error occurs
        */
        @Override
        protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
            throws ServletException, IOException {
            processRequest(request, response);
        }

        /**
        * Returns a short description of the servlet.
        *
        * @return a String containing servlet description
        */
        @Override
        public String getServletInfo() {
            return "Short description";
        } // </editor-fold>
    }
}

```

```

import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.util.HashMap;

/**
 *
 * @author Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class AccountServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
     <code>POST</code>
     * methods.
     *

```

```

    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
    HashMap<String, String[]> account = new HashMap();

    @Override
    public void init() throws ServletException {
        super.init();
        account.put("Ali", new String[]{"31/01/2019: 2000.00",
"28/02/2019: 3000.00"});
        account.put("Ahmad", new String[]{"31/01/2019: 100.00",
"28/02/2019: 5000.00"});
        account.put("Muthu", new String[]{"31/01/2019: 1000.00",
"28/02/2019: 2000.00"});
    }

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        String userid_login = (String) request.getAttribute("userid");

        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet AccountServlet</title>");
            out.println("</head>");
            out.println("<body>");

            if (account.get(userid_login) == null)
                out.println("<h1>Sorry, no information found!</h1>");

            else {
                out.println("<h1>Account status for: " + userid_login
+ "</h1>");
                for (String tempAcc : account.get(userid_login)) {
                    out.println("<h2>" + tempAcc + "</h2>");
                }
            }

            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet
methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */

```

```

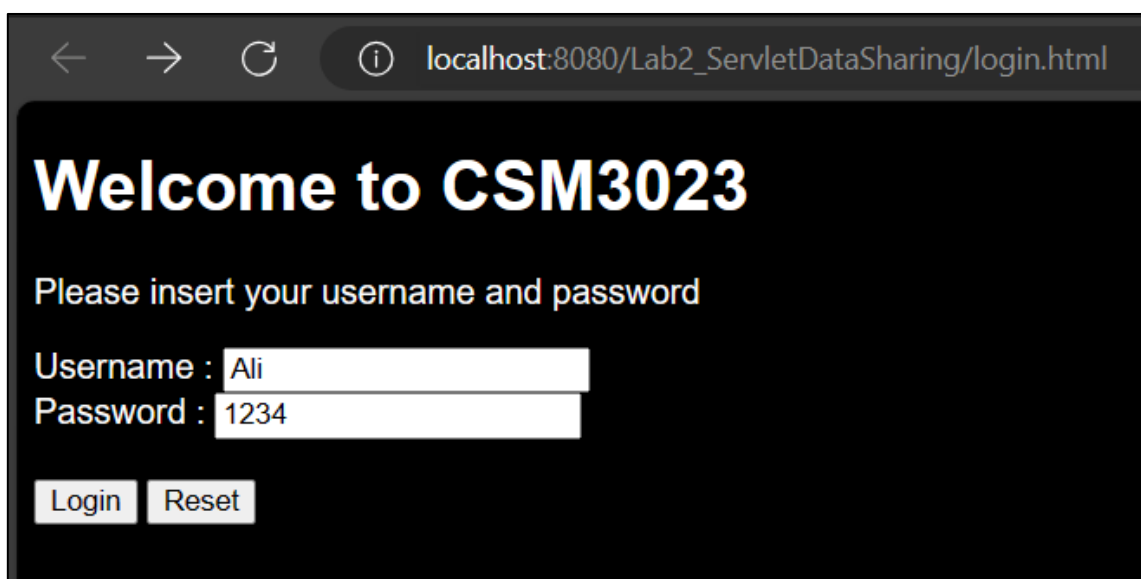
@Override
protected void doGet(HttpServletRequest request,
HttpServletRequest response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request,
HttpServletRequest response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

- Output:



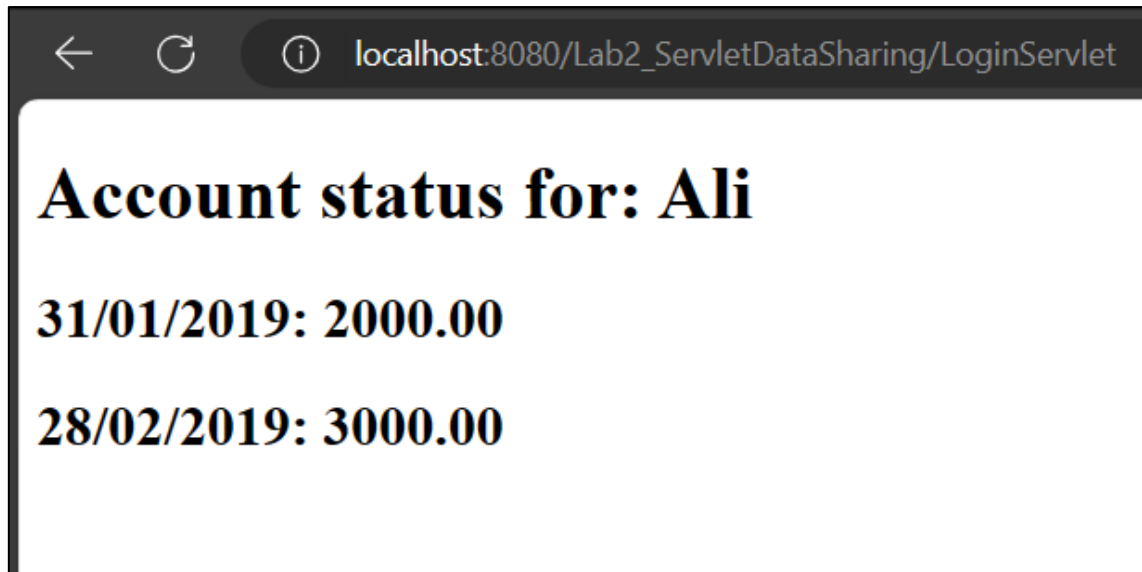
← → ↻ ⓘ localhost:8080/Lab2_ServletDataSharing/login.html

Welcome to CSM3023

Please insert your username and password

Username :

Password :



- Reflection:

1. What have you learnt from this exercise?

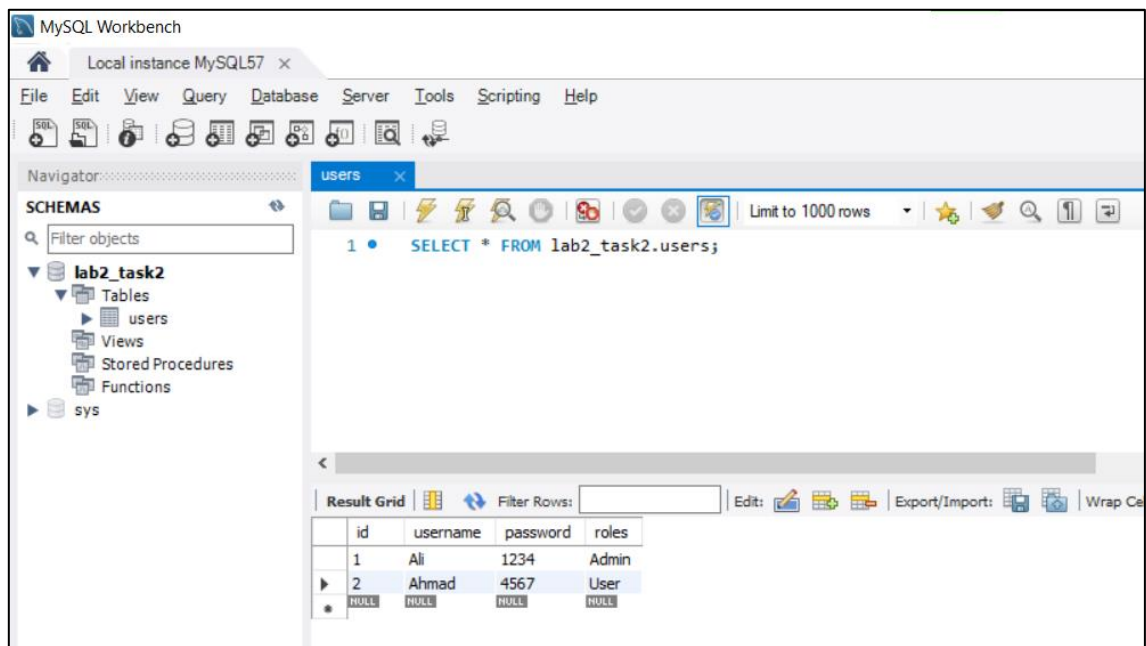
From this exercise I learnt how to use a servlet for request forwarding and data sharing to authenticate a user in a login form. An HTML form was created for user input, which collects the username and password. This form submits data to a LoginServlet for processing. The servlet retrieves the username and password submitted from the HTML form, validates them against a predefined set of credentials stored in a HashMap, and forwards the request accordingly.

2. What are the common methods used in Java Servlet?

- doGet()
- doPost()

Task 2 : Creating A Table in MySQL Database

- Objective: To create a MySQL table to store user credentials by preparing a user table to be used in Web Application.
- Output:

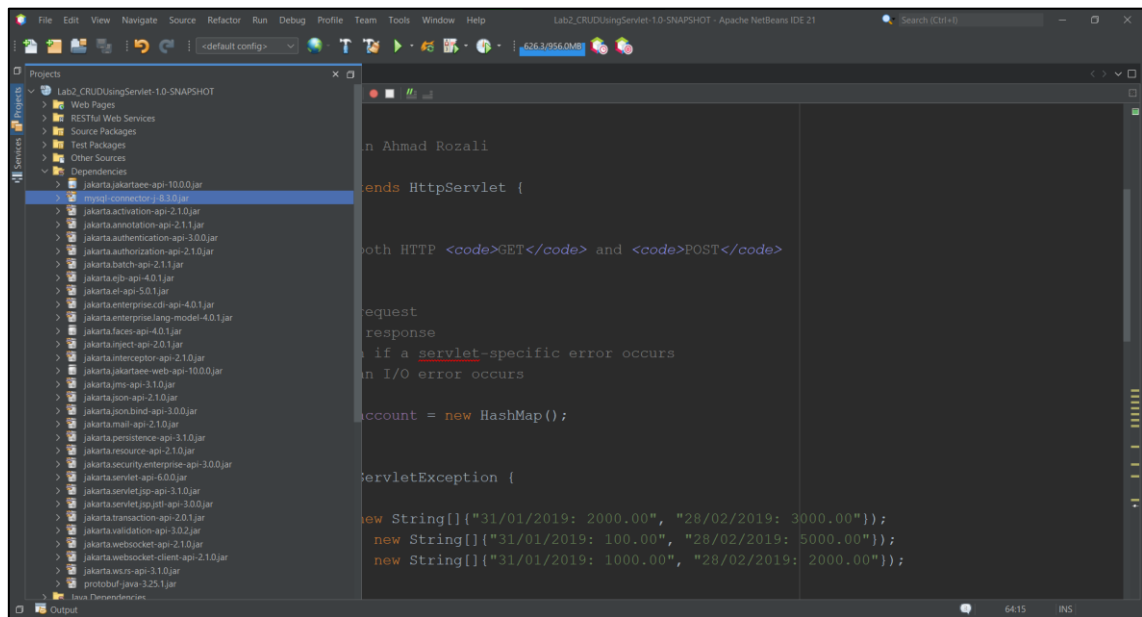


MySQL Workbench interface showing the 'users' table in the 'lab2_task2' database. The query executed is `SELECT * FROM lab2_task2.users;`. The result grid displays the following data:

id	username	password	roles
1	Ali	1234	Admin
2	Ahmad	4567	User
HULL	HULL	HULL	HULL

Connection

- Output:



Task 4 : Using Servlets for Database CRUD Operations

- Objective: To program multiple servlets for manipulating the database by programming five different servlets to handle database operations such as insert, update and delete.
- Code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>User Management</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
  </head>
  <body>

    <h1>Add New User</h1>
    <form action="SaveServlet" method="post">
      <table>
        <tr><td>Username:</td><td><input type="text"
name="name"></td></tr>
        <tr><td>Password:</td><td><input type="password"
name="password"></td></tr>
        <tr><td>Role:</td><td>
          <select name="role">
            <option>Admin</option>
            <option>User</option>
          </select>
        </td></tr>
        <tr><td colspan="2"><input type="submit" value="Save
User"></td></tr>
      </table>
    </form>
    <br>
    <a href="ViewServlet">View users</a>

  </body>
</html>
```

```
/**
 *
 * @author Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class User {

  private int id;
  private String username, password, role;

  public int getId() {
    return id;
  }

  public void setId(int id) {
```

```

        this.id = id;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getRole() {
        return role;
    }

    public void setRole(String role) {
        this.role = role;
    }
}

```

```

/**
 *
 * @author Ahmad Afif Syahmi bin Ahmad Rozali
 */
import java.util.*;
import java.sql.*;

public class UserDao {

    public static Connection getConnection() {
        Connection con = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/lab2_task2",
"root", "admin");
        }
        catch (Exception e) {
            System.out.println(e);
        }
        return con;
    }

    public static int save(User e) {
        int status = 0;
        try {
            Connection con = UserDao.getConnection();
            PreparedStatement ps = con.prepareStatement(

```

```

        "INSERT INTO users(username, password, roles)
VALUES (?, ?, ?)"
    );
    ps.setString(1, e.getUsername());
    ps.setString(2, e.getPassword());
    ps.setString(3, e.getRole());

    status = ps.executeUpdate();

    con.close();
} catch (Exception ex) {
    ex.printStackTrace();
}

return status;
}

public static int update(User e) {
    int status = 0;
    try {
        Connection con = UserDao.getConnection();
        PreparedStatement ps = con.prepareStatement(
            "UPDATE users SET username = ?, password = ?,
roles = ? WHERE id = ?"
        );
        ps.setString(1, e.getUsername());
        ps.setString(2, e.getPassword());
        ps.setString(3, e.getRole());
        ps.setInt(4, e.getId());

        status = ps.executeUpdate();

        con.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    return status;
}

public static int delete(int id) {
    int status = 0;

    try {
        Connection con = UserDao.getConnection();
        PreparedStatement ps = con.prepareStatement(
            "DELETE FROM users WHERE id = ?"
        );
        ps.setInt(1, id);

        status = ps.executeUpdate();

        con.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    return status;
}
}

```

```

public static User getUserById(int id) {
    User e = new User();

    try {
        Connection con = UserDao.getConnection();
        PreparedStatement ps = con.prepareStatement(
            "SELECT * FROM users WHERE id = ?"
        );
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            e.setId(rs.getInt(1));
            e.setUsername(rs.getString(2));
            e.setPassword(rs.getString(3));
            e.setRole(rs.getString(4));
        }

        con.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    return e;
}

public static List<User> getAllUsers() {
    List<User> list = new ArrayList<User>();

    try {
        Connection con = UserDao.getConnection();
        PreparedStatement ps = con.prepareStatement(
            "SELECT * FROM users"
        );
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            User e = new User();
            e.setId(rs.getInt(1));
            e.setUsername(rs.getString(2));
            e.setPassword(rs.getString(3));
            e.setRole(rs.getString(4));
            list.add(e);
        }
        con.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    return list;
}
}

```

```

import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;

```

```

import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

/**
 *
 * @author Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class SaveServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and
     * POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        String name = request.getParameter("name");
        String password = request.getParameter("password");
        String role = request.getParameter("role");

        User e = new User();
        e.setUsername(name);
        e.setPassword(password);
        e.setRole(role);

        int status = UserDao.save(e);
        if (status > 0) {
            out.print("<p>Record saved successfully!</p>");
        }

        request.getRequestDispatcher("index.html").include(request, response);
        else {
            out.println("Sorry! Unable to save record.");
        }

        out.close();
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet
    methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP GET method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override

```

```

        protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
            throws ServletException, IOException {
            processRequest(request, response);
        }

        /**
         * Handles the HTTP <code>POST</code> method.
         *
         * @param request servlet request
         * @param response servlet response
         * @throws ServletException if a servlet-specific error occurs
         * @throws IOException if an I/O error occurs
         */
        @Override
        protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
            throws ServletException, IOException {
            processRequest(request, response);
        }

        /**
         * Returns a short description of the servlet.
         *
         * @return a String containing servlet description
         */
        @Override
        public String getServletInfo() {
            return "Short description";
        } // </editor-fold>
    }
}

```

```

import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.util.List;

/**
 *
 * @author Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class ViewServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
     <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
}

```

```

        protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
            throws ServletException, IOException {
            response.setContentType("text/html;charset=UTF-8");
            PrintWriter out = response.getWriter();
            out.println("<a href='index.html'>Add New User</a>");
            out.println("<h1>User List</h1>");

            List<User> list = UserDao.getAllUsers();

            out.print("<table border='1' width='100%'>");

            out.print("<tr><th>Id</th><th>Name</th><th>Password</th><th>Role</th><th>Edit</th><th>Delete</th>");
            for (User e : list) {
                out.print("<tr><td>" + e.getId() + "</td><td>" +
                e.getUsername() + "</td><td>"
                    + e.getPassword() + "</td><td>" + e.getRole() +
                "</td><td><a href='EditServlet?id="
                    + e.getId() + "'>Edit</a></td><td><a
                href='DeleteServlet?id="
                    + e.getId() + "'>Delete</a></td></tr>"
                );
            }
            out.print("</table>");

            out.close();
        }

        // <editor-fold defaultstate="collapsed" desc="HttpServlet
        methods. Click on the + sign on the left to edit the code.">
        /**
         * Handles the HTTP <code>GET</code> method.
         *
         * @param request servlet request
         * @param response servlet response
         * @throws ServletException if a servlet-specific error occurs
         * @throws IOException if an I/O error occurs
         */
        @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
            throws ServletException, IOException {
            processRequest(request, response);
        }

        /**
         * Handles the HTTP <code>POST</code> method.
         *
         * @param request servlet request
         * @param response servlet response
         * @throws ServletException if a servlet-specific error occurs
         * @throws IOException if an I/O error occurs
         */
        @Override
        protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
            throws ServletException, IOException {
            processRequest(request, response);
        }
    }

```



```

    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    } // </editor-fold>
}

```

```

import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

/**
 *
 * @author Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class EditServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
     <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<h1>Update User</h1>");
        String sid = request.getParameter("id");
        int id = Integer.parseInt(sid);

        User e = UserDao.getUserById(id);

        out.print("<form action='EditServlet2' method='post'>");
        out.print("<table>");
        out.print("<tr><td></td><td><input type='hidden' name='id'
value='"
            + e.getId() + "'></td></tr>"
        );
        out.print("<tr><td>Name:</td><td><input type='text'
name='username' value='"
            + e.getUsername() + "'></td></tr>"

```

```

        );
        out.print("<tr><td>Password:</td><td><input type='password'
name='password' value='"
            + e.getPassword() + "'></td></tr>"
        );
        out.print("<tr><td>Role:</td><td>");
        out.print("<select name='role' style='width:150px'>");
        out.print("<option>Admin</option>");
        out.print("<option>User</option>");
        out.print("</select>");
        out.print("</td></tr>");
        out.print("<tr><td colspan='2'><input type='submit'
value='Edit & Save'></td></tr>");
        out.print("</table></form>");

        out.close();
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet
methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }
}
</editor-fold>

```

```
}
```

```
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

/**
 *
 * @author Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class EditServlet2 extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and
     * POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        int id = Integer.parseInt(request.getParameter("id"));
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String role = request.getParameter("role");

        User e = new User();
        e.setId(id);
        e.setUsername(username);
        e.setPassword(password);
        e.setRole(role);

        int status = UserDao.update(e);
        if (status > 0) {
            out.print("<p>Record saved successfully!</p>");
        }

        request.getRequestDispatcher("index.html").include(request, response);
    }
    else {
        out.println("Sorry! Unable to save record");
    }

    out.close();
}
```

```

// <editor-fold defaultstate="collapsed" desc="HttpServlet
methods. Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

```

import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

/**
 *
 * @author Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class DeleteServlet extends HttpServlet {

```

```

/**
 * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
 * methods.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    String sid = request.getParameter("id");
    int id = Integer.parseInt(sid);
    UserDao.delete(id);
    response.sendRedirect("ViewServlet");
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet
methods. Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

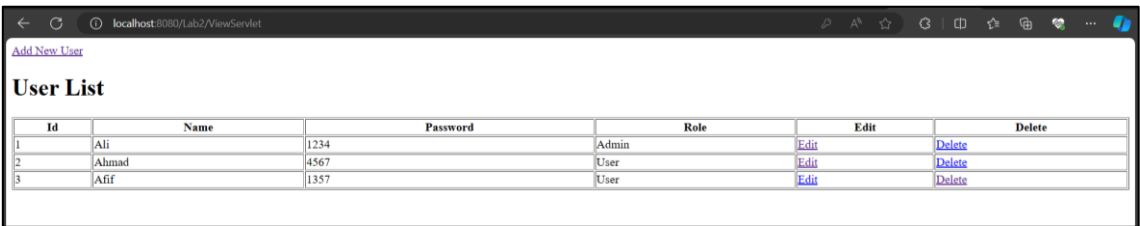
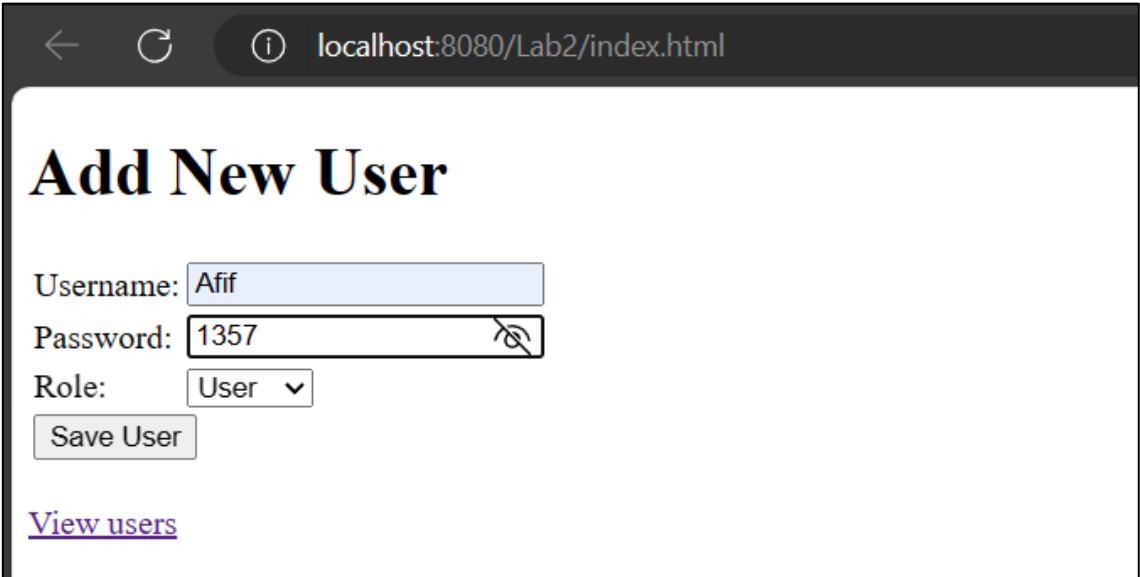
/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {

```

```
        return "Short description";
    } // </editor-fold>
}
```

- Output:



← ↻ ⓘ localhost:8080/Lab2/EditServlet?id=3

Update User

Name:

Password:

Role: ▼ 

←

↺

🔍

localhost:8080/Lab2/ViewServlet

⌕

☆

🔄

📄

🔖

📁

💡

⋮

🧑

[Add New User](#)

User List

	Id	Name	Password	Role	Edit	Delete
1		Ali	1234	Admin	Edit	Delete
2		Ahmad	4567	User	Edit	Delete

←

↺

🔍

localhost:8080/Lab2/ViewServlet

🔍

🏠

🔖

🔄

📄

🔗

📧

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

📌

- Reflection:

1. What is the name of the Java Library that you need to import before coding the web application with database operations?

The name of the Java library commonly used for database operations in web applications is JDBC (Java Database Connectivity). JDBC provides a set of classes and interfaces for accessing and manipulating relational databases from Java

programs. It allows Java applications to connect to a database, send SQL queries, retrieve results, and perform database transactions.

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 web.xml file is located in the WEB-INF folder of a Java web application.

```
<?xml version="1.0" encoding="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">
    <servlet>
        <servlet-name>SaveServlet</servlet-name>
        <servlet-class>SaveServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>ViewServlet</servlet-name>
        <servlet-class>ViewServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>EditServlet</servlet-name>
        <servlet-class>EditServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>EditServlet2</servlet-name>
        <servlet-class>EditServlet2</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>DeleteServlet</servlet-name>
        <servlet-class>DeleteServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>SaveServlet</servlet-name>
        <url-pattern>/SaveServlet</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>ViewServlet</servlet-name>
        <url-pattern>/ViewServlet</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>EditServlet</servlet-name>
        <url-pattern>/EditServlet</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>EditServlet2</servlet-name>
        <url-pattern>/EditServlet2</url-pattern>
    </servlet-mapping>
```



```
<servlet-mapping>
  <servlet-name>DeleteServlet</servlet-name>
  <url-pattern>/DeleteServlet</url-pattern>
</servlet-mapping>
<session-config>
  <session-timeout>
    30
  </session-timeout>
</session-config>
</web-app>
```

<servlet-name> specifies a unique name for the servlet configuration. <servlet-class> specifies the fully qualified class name of the servlet implementation. <servlet-mapping> maps a servlet to a URL pattern. It defines the URLs that invoke the servlet. <url-pattern> specifies the URL pattern to which the servlet is mapped.

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

A Data Access Object (DAO) servlet is a design pattern used to separate the data access logic from the business logic of a servlet-based web application. The DAO pattern involves creating a separate class or set of classes responsible for interacting with the database, querying data, and performing database operations. By using a DAO servlet, the business logic in servlets can focus on handling user requests, processing data, and generating responses, while the data access logic is encapsulated within the DAO classes, improving the overall organization and clarity of the application architecture.