UNIVERSITI MALAYSIA TERENGGANU

CSM3023 WEB BASED APPLICATION DEVELOPMENT

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

LAB 2

SEMESTER II 2023/2024

Prepared for:

SIR MOHD ARIZAL SHAMSIL BIN MAT RIFIN

Prepared by:

MUHAMMAD ZAHIER BIN RAZMI

(S67943)

Task 1 : Data Sharing in Servlet

Login.html:

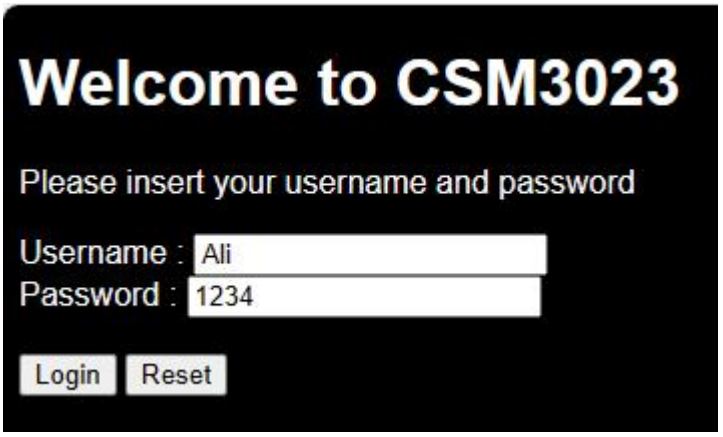
```
<html>
    <head>
        <title>Login Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style>
            body {
                background-color: black;
                text-align: left;
                color: white;
                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>
```

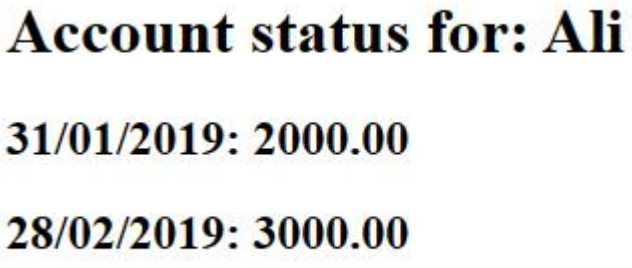
LoginServlet.java

```
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 Zahier
 */
public class LoginServlet extends HttpServlet {

    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");
    }

    /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> ...9 lines */
```

```
    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{
            RequestDispatcher rd = request.getRequestDispatcher("/login.html");
            rd.forward(request, response);
        }
    }

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

}
```

Output:





Reflections:

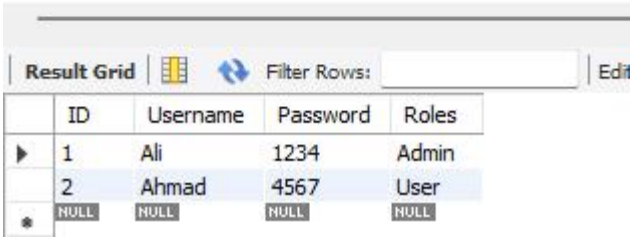
1. What have you learnt from this exercise?

   I have learnt how the data sharing process works by using servlet.

2. What are the common methods used in Java Servlet?
   doGet() and doPost()

Task 2: Creating a table in mySQL Database

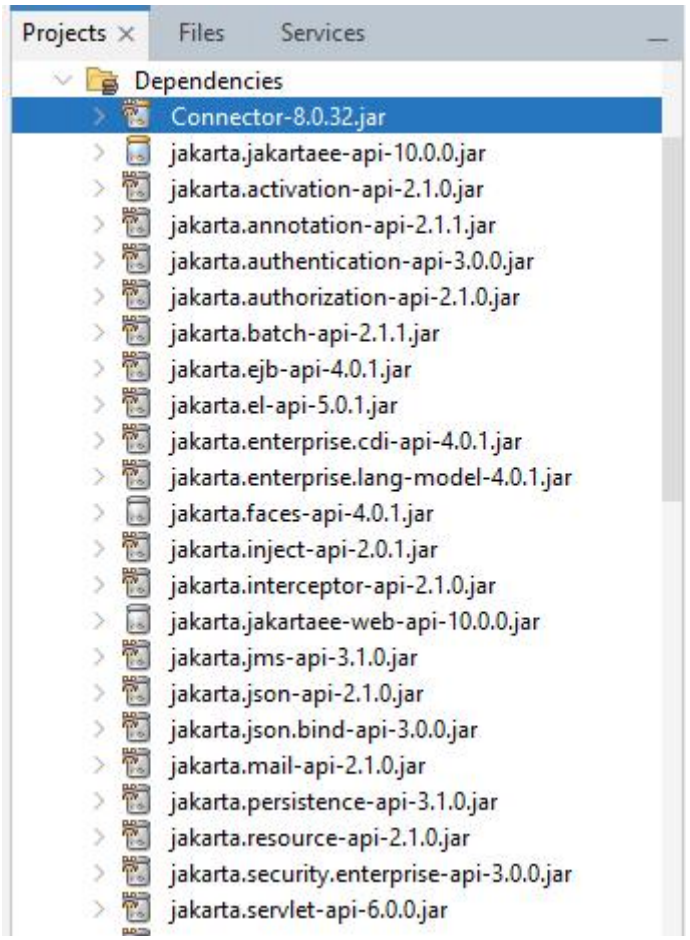
Output:



ID	Username	Password	Roles
1	Ali	1234	Admin
2	Ahmad	4567	User
NULL	NULL	NULL	NULL

Task 3: Setting the Environment of Web Application for Database Connection

Output:



Task 4: Using servlets for database CRUD operations

Index.html:

```
<!DOCTYPE html>
<html>
    <head>
        <title>User Management</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </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" style="width:150px">
                            <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="ViewServlets">View Users</a>
    </body>
</html>
```

User.java:

```
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;
    }
}
```

UserDao.java:

```
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;
    }

}
```

SaveServlet.java:

```java
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 Zahier
 */
public class SaveServlet 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();

        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();
    }

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

}
```

ViewServlets.java:

```
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 Zahier
 */
public class ViewServlets 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();
    }

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

}
```

EditServlet.java:

```
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 Zahier
 */
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();
    }

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

}
```

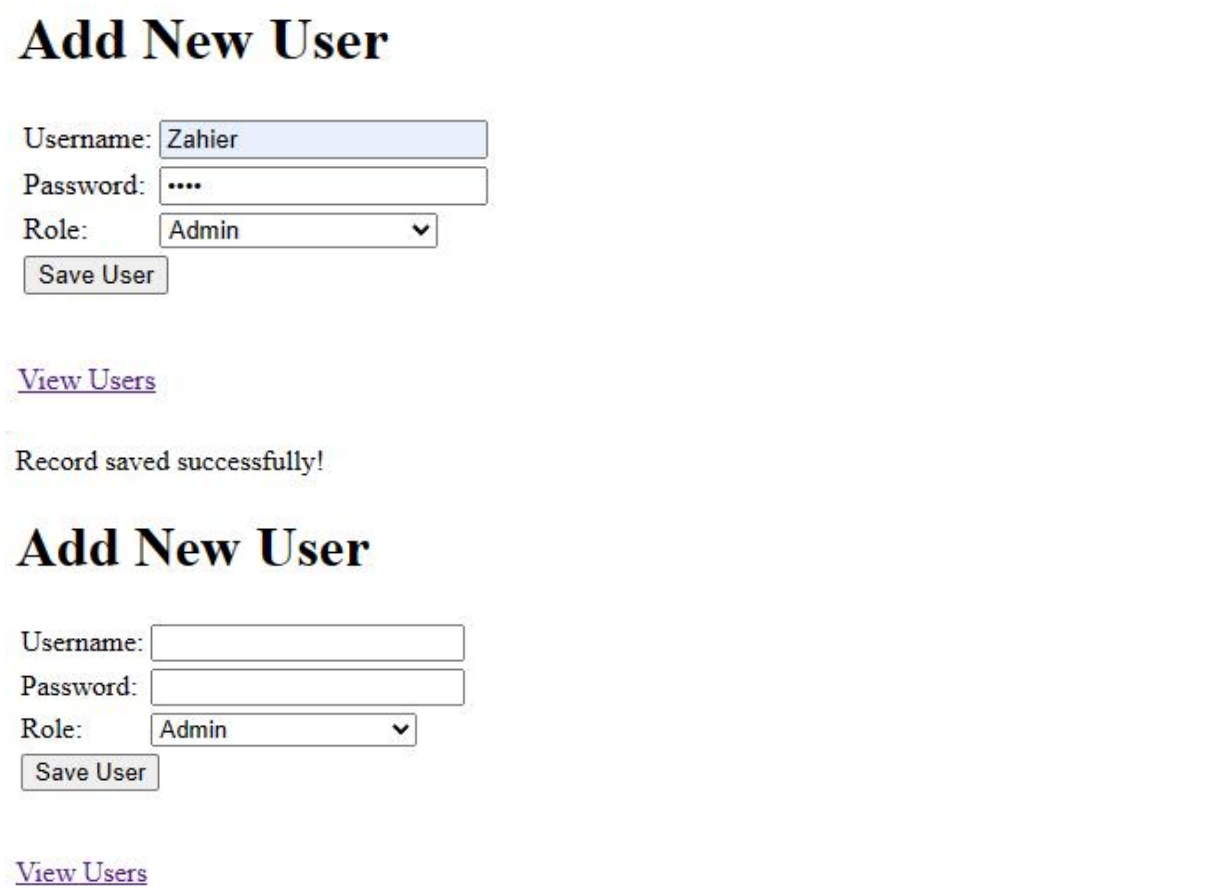
EditServlet2.java:

```
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 Zahier
 */
public class EditServlet2 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();

        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();
    }

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

}
```

DeleteServlet.java:

```java
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 Zahier
 */
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");
    }

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

}
```

Output:



Add New User

User List

Id	Name	Password	Role	Edit	Delete
1	Ali	1234	Admin	Edit	Delete
2	Ahmad	4567	User	Edit	Delete
3	Zahier	2468	User	Edit	Delete

Update:

Add New User

User List

Id	Name	Password	Role	Edit	Delete
1	Ali	1234	Admin	Edit	Delete
2	Ahmad	4567	User	Edit	Delete
3	Zahier	2468	Admin	Edit	Delete

Delete:

Add New User

User List

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

Reflections:

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

   JDBC (Java Database Connectivity).It provides a set of classes and interfaces for accessing and manipulating relational databases from a Java program.It also allows Java applications to connect to a database and perform CRUD process by sending SQL queries to database.

2. Which folder keeps the web.xml file? Copy the contents of the file and explain in brief the tags included such as . 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 permissioned class name of the 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 ease the business process in your servlet-based web application?

DAO servlet is a design pattern used to separate the data access logic from the business logic of a servlet-based web application. DAO involves creating a separate class or set of classes responsible for interacting with the database, querying data, and perform database CRUD 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.