Experiment 7

Student Name: Vikash Singh **UID: 22BCS10491**
Branch: BE-CSE **Section/Group: 903-B**
Semester: 6th **Date of Performance:**
Subject Name: Project based Learning Java **Subject Code: 22CSH-359**

Problem :- 1(Easy-Level)

1.Aim: Write a servlet to accept user credentials through an HTML form and display a personalized welcome message if the login is successful.

2.Objective:

- Implement a servlet to handle HTTP POST requests for user authentication.
- Accept username and password through an HTML form.
- Validate credentials and display a welcome message upon success.

3. Algorithm:

- Create an HTML form with fields for username and password.
- Implement a Java Servlet to handle form submissions.
- Retrieve user credentials from the request.
- Validate credentials (using hardcoded values for simplicity).
- If valid, display a personalized welcome message; otherwise, show an error.

4. Implementation :

HTML Form (index.html)

```
<!DOCTYPE html>

<html>

<head>

  <title>Login Page</title>

</head>
```

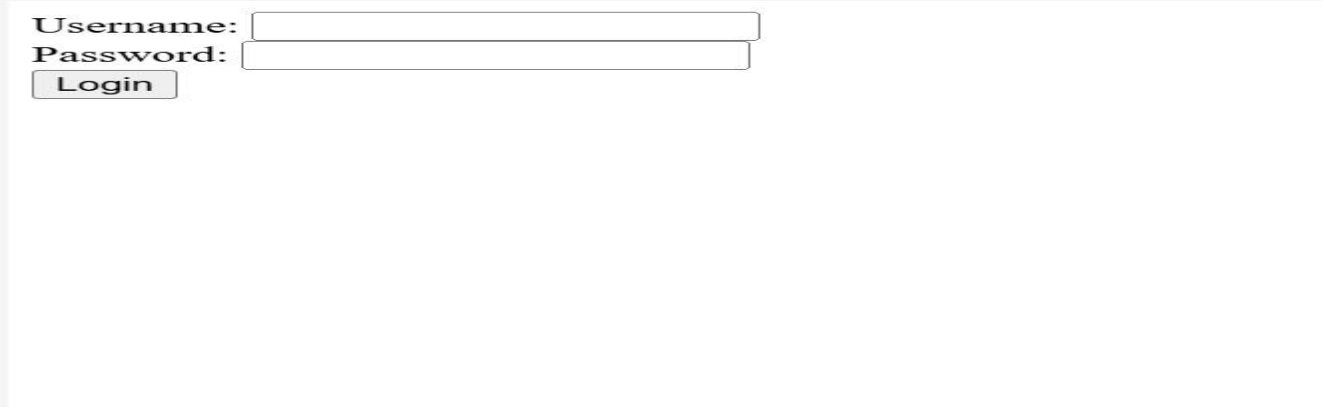
```
<body>
   <form action="LoginServlet" method="post">
      <label>Username:</label>
      <input type="text" name="username" required><br>
      <label>Password:</label>
      <input type="password" name="password" required><br>
      <input type="submit" value="Login">
   </form>
</body>
</html>
```

Servlet (LoginServlet.java)

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html");
```

```
        PrintWriter out = response.getWriter();

        // Retrieve form parameters
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        // Hardcoded credentials for validation
        if ("admin".equals(username) && "password123".equals(password)) {
            out.println("<h2>Welcome, " + username + "!</h2>");
        } else {
            out.println("<h2>Invalid username or password. Try again.</h2>");
        }
    }
}
```

5.Output:



Problem:- 2(Medium-level)

1.Aim: Create a servlet integrated with JDBC to display a list of employees from a database.

Include a search form to fetch employee details by ID.

2.Objective:

- Establish a connection to a MySQL database using JDBC.
- Fetch and display all employees from the database.
- Provide a search form to retrieve employee details by ID.

3.Algorithm:

- Define database connection details (URL, username, password).
- Establish a connection using DriverManager.
- If an employee ID is provided, fetch and display the corresponding record.
- Otherwise, retrieve and display all employee records.
- Close the database connection.

5. Implementation/Code:

Servlet (EmployeeServlet.java)

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/EmployeeServlet")
public class EmployeeServlet extends HttpServlet {
```

```
    private static final String URL = "jdbc:mysql://localhost:3306/your_database"; // Replace with DB
name
    private static final String USER = "your_username"; // Replace with MySQL username
    private static final String PASSWORD = "your_password"; // Replace with MySQL password

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String empId = request.getParameter("empId");

        try {
            Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
            String query;
            PreparedStatement stmt;

            if (empId != null && !empId.isEmpty()) {
                query = "SELECT EmpID, Name, Salary FROM Employee WHERE EmpID = ?";
                stmt = conn.prepareStatement(query);
                stmt.setInt(1, Integer.parseInt(empId));
            } else {
                query = "SELECT EmpID, Name, Salary FROM Employee";
                stmt = conn.prepareStatement(query);
            }

            ResultSet rs = stmt.executeQuery();
            out.println("<h2>Employee Details</h2>");
            while (rs.next()) {
```

```
            out.println("EmpID: " + rs.getInt("EmpID") + ", Name: " + rs.getString("Name") +
                    ", Salary: " + rs.getDouble("Salary") + "<br>");
            }

            conn.close();
        } catch (Exception e) {
            out.println("<h2>Error: " + e.getMessage() + "</h2>");
        }
    }
}
```

5. Output:

1. Viewing All Employees:

```
EmpID: 1, Name: John Doe, Salary: 50000
EmpID: 2, Name: Alice, Salary: 60000
EmpID: 3, Name: Bob, Salary: 55000
```

2. Searching for Employee by ID (Example: ID = 1)

```
EmpID: 1, Name: John Doe, Salary: 50000
```

Problem:- 3(Hard-level)

1.Aim: Develop a JSP-based student portal that includes a form for entering attendance details and saves them to a database using a servlet.

2.Objective:

- Create a JSP form to enter student attendance details.
- Implement a servlet to handle form submissions.

- Store attendance records in a MySQL database using JDBC.

3.Algorithm:

- Create a JSP form to enter student ID, name, and attendance status.
- Implement a servlet to handle form submission and insert data into the database.
- Retrieve form data using HttpServletRequest.
- Establish a database connection using DriverManager.
- Insert the data into the attendance table.
- Close the database connection.

4. Implementation/Code:

JSP Form (attendance.jsp)

```
<!DOCTYPE html>
<html>
<head>
  <title>Student Attendance</title>
</head>
<body>
  <h2>Enter Student Attendance</h2>
  <form action="AttendanceServlet" method="post">
    <label>Student ID:</label>
    <input type="text" name="studentId" required><br>
    <label>Name:</label>
    <input type="text" name="name" required><br>
    <label>Attendance:</label>
    <select name="attendance">
      <option value="Present">Present</option>
      <option value="Absent">Absent</option>
```

```
    </select><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

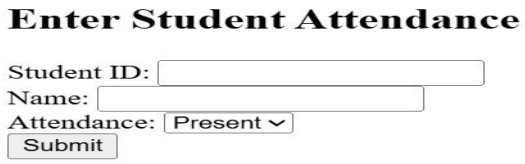
Servlet (AttendanceServlet.java)

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/AttendanceServlet")
public class AttendanceServlet extends HttpServlet {
    private static final String URL = "jdbc:mysql://localhost:3306/your_database"; // Replace with
DB name
    private static final String USER = "your_username"; // Replace with MySQL username
    private static final String PASSWORD = "your_password"; // Replace with MySQL password

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html");
```

```
        PrintWriter out = response.getWriter();

        String studentId = request.getParameter("studentId");
        String name = request.getParameter("name");
        String attendance = request.getParameter("attendance");

        try {
            Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
            String query = "INSERT INTO Attendance (StudentID, Name, Status) VALUES
(?, ?, ?)";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setInt(1, Integer.parseInt(studentId));
            stmt.setString(2, name);
            stmt.setString(3, attendance);
            int rows = stmt.executeUpdate();

            if (rows > 0) {
                out.println("<h2>Attendance recorded successfully.</h2>");
            } else {
                out.println("<h2>Failed to record attendance.</h2>");
            }

            conn.close();
        } catch (Exception e) {
            out.println("<h2>Error: " + e.getMessage() + "</h2>");
        }
    }
```

}

6. Output:



7. Learning Outcomes:

- ✧ **JSP Form Handling:** How to create and submit forms using JSP.
- ✧ **Servlet Communication:** How a servlet processes form data sent from a JSP page.
- ✧ **JDBC Connectivity:** How to establish a connection to a MySQL database using JDBC.
- ✧ **Data Insertion:** How to insert form data into a database using PreparedStatement.
- ✧ **HTTP Request Handling:** How to use doPost() in servlets for handling form submissions.
- ✧ **Web Application Flow:** Understanding how JSP, servlets, and databases interact in a web application.