Experiment -8

Student Name: Rishu Raj **UID:22BCS15617**

Branch: BE-CSE **Section: IOT-631(B)**

Semester:6th **DOP: 02/04/2025**

Subject Name: JAVA **Subject Code: 22CSH-359**

3.1.1

Aim: To develop a servlet that accepts user credentials from an HTML form and displays a personalized welcome message on successful login.

Objective: Learn form handling with Servlets Understand HTTP request/response handling Practice doPost() method

Code:

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

import java.io.*; import
javax.servlet.*; import
javax.servlet.http.*;

public class LoginServlet extends HttpServlet
{
 protected void doPost(HttpServletRequest request, HttpServletResponse response)
   throws ServletException, IOException
{
```

```
        String user = request.getParameter("username");
        String pass = request.getParameter("password");

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        if ("admin".equals(user) && "1234".equals(pass)) {
        out.println("<h2>Welcome, " + user + "!</h2>");
        }
        else{
        System.out.println("<h2>Login Failed Invalid
           credentials.</h2>");
        }
    }
}
```

```
<web-app>
  <servlet>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>LoginServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/LoginServlet</url-pattern>
  </servlet-mapping>
</web-app>
```

Output:

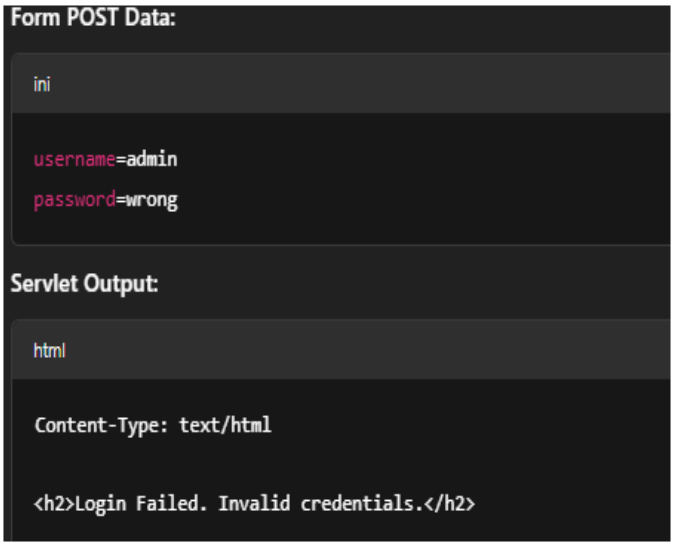
User opens:

```
bash

http://localhost:8080/YourApp/index.html
```

Case 1: Correct Credentials

Case 2: Invalid Credentials



3.1.2

Aim: To build a servlet integrated with JDBC that displays a list of all employees from a database. Include a search form to fetch employee details by ID.

Objective:

1) Use JDBC with Servlet
2) Fetch and display records
3) Implement search functionality

Code:

```
<!DOCTYPE html>
<html>
<head><title>Search Employee</title></head>
<body>
  <form action="EmployeeServlet" method="post">
    Enter Employee ID: <input type="text" name="empId">
    <input type="submit" value="Search">
  </form>
</body>
</html>
```

```
import java.io.*;
import       javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class EmployeeServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)
   throws ServletException, IOException {

      String empId = request.getParameter("empId");
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();

      try {
         Class.forName("com.mysql.jdbc.Driver");
         Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/company","root", "password");
String query = "SELECT * FROM employees WHERE emp_id=?";
         PreparedStatement ps = con.prepareStatement(query);
         ps.setString(1, empId); ResultSet
         rs = ps.executeQuery();

         if (rs.next()) {
            out.println("<h2>Employee       Details</h2>");
            out.println("ID:  "  +  rs.getInt(1)  +  "<br>");
            out.println("Name: " + rs.getString(2) + "<br>");
            out.println("Department: " + rs.getString(3));
         } else {
         System.out.println("No employee found with ID " +
            empId);
         }

         con.close();
      } catch (Exception e) { out.println("Error:
      " + e.getMessage()); }
   }
}
```

Output:

User opens:

```bash
http://localhost:8080/YourApp/index.html
```

MySQL table employees has:

emp_id	name	department
101	John Doe	HR
102	Alice Roy	IT

Case 1: Employee Found

POST Request:

```ini
empId=101
```

Simulated Server Execution:

```sql
> Received POST request at /EmployeeServlet
> Parameter empId = 101
> Connecting to database...
> Running query: SELECT * FROM employees WHERE emp_id=101
> Match found!
```

Output:

```html
Content-Type: text/html

<h2>Employee Details</h2>
ID: 101<br>
Name: John Doe<br>
Department: HR
```

Case 2: Employee Not Found

POST Request:

```ini
empId=999
```

Simulated Server Execution:

```shell
> Received POST request at /EmployeeServlet
> Parameter empId = 999
> Connecting to database...
> Running query: SELECT * FROM employees WHERE emp_id=999
> No results found.
```

Output:

```html
Content-Type: text/html

No employee found with ID 999
```

Learning Outcomes:

1. Understand how to create and deploy Java Servlets to handle user input from HTML forms.

2. Implement conditional logic in Servlets to validate user credentials and provide dynamic responses.

3. Learn to integrate Servlets with JDBC to perform database operations like searching and fetching data.

4. Develop interactive web applications using Servlets and JSP that connect front-end forms with backend databases.