Experiment 9

Student Name: Jatin Garg
Branch: B.E. CSE
Semester: 6th
Subject Name:Java
UID: 22BCS15676
Section/Group: IOT-640/B
Date of Performance: 16/04/25
Subject Code: 22CSH-359

(a) Login Servlet

□ Aim:

To develop a servlet that handles user login using an HTML form and displays a personalized welcome message upon successful login.

□ Objective:

- Collect user credentials through an HTML form.
- Validate credentials in a Servlet.
- Display a personalized message if login is successful.

□ Code:

login.html

```
html
CopyEdit
<!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>
```

LoginServlet.java

```
java
CopyEdit
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) && "password123".equals(pass)) {
            out.println("<h2>Welcome, " + user + "!</h2>");
        } else {
            out.println("<h3>Invalid credentials.</h3>");
        }
    }
}
```

□ Output:

- If credentials match: `Welcome, admin!`
- Else: `Invalid credentials.`

□ Learning Outcomes:

- Understanding how to use Servlets for handling form data.
- Handling HTTP POST requests.
- Generating dynamic HTML content using Java Servlets.

□**(b) Employee List + Search using JDBC**

□ Aim:

To develop a servlet that connects to a database using JDBC to fetch and display employee records and perform searches by ID.

□ Objective:

- Connect a servlet to a MySQL database.
- Display employee details in an HTML table.
- Add a search feature to filter by employee ID.

□ Code:

`EmployeeServlet.java`

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

```
public class EmployeeServlet extends HttpServlet {
    String dbURL = "jdbc:mysql://localhost:3306/yourdb";
    String dbUser = "root";
    String dbPass = "password";

    protected void doGet(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.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(dbURL, dbUser, dbPass);
            Statement stmt = con.createStatement();

            out.println("<form action='EmployeeServlet' method='get'>Search ID:
<input type='text' name='empid'><input type='submit' value='Search'></form>");

            String query = (empId != null && !empId.isEmpty()) ?
                "SELECT * FROM employees WHERE id=" + empId :
                "SELECT * FROM employees";

            ResultSet rs = stmt.executeQuery(query);
            out.println("<table
border='1'><tr><th>ID</th><th>Name</th><th>Dept</th></tr>");
            while (rs.next()) {
                out.println("<tr><td>" + rs.getInt("id") + "</td><td>" +
                        rs.getString("name") + "</td><td>" +
rs.getString("department") + "</td></tr>");
            }
            out.println("</table>");
            con.close();
        } catch (Exception e) {
            out.println("Error: " + e.getMessage());
        }
    }
}
```

□ Output:

- All employees are displayed in a table.
- Search box filters data by employee ID.

□ Learning Outcomes:

- Establishing JDBC connection in servlets.
- Executing SQL queries and displaying results.
- Adding dynamic search functionality.

□**(c) JSP-based Student Attendance Portal**

□ Aim:

To build a JSP-based student attendance portal where data is submitted through a form and stored in a database using a Servlet.

□ Objective:

- Create a JSP form to take attendance input.
- Send the data to a servlet.
- Insert the data into a database.

□ Code:

attendance.jsp

```
jsp
CopyEdit
<!DOCTYPE html>
<html>
<head><title>Student Attendance</title></head>
<body>
    <h2>Student Attendance Form</h2>
    <form action="AttendanceServlet" method="post">
        Student ID: <input type="text" name="sid"><br/>
        Date: <input type="date" name="date"><br/>
        Status:
        <select name="status">
            <option value="Present">Present</option>
            <option value="Absent">Absent</option>
        </select><br/>
        <input type="submit" value="Submit Attendance">
    </form>
</body>
</html>
```

AttendanceServlet.java

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

public class AttendanceServlet extends HttpServlet {
    String dbURL = "jdbc:mysql://localhost:3306/yourdb";
    String dbUser = "root";
    String dbPass = "password";

    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String sid = request.getParameter("sid");
        String date = request.getParameter("date");
        String status = request.getParameter("status");

        response.setContentType("text/html");
```

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

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(dbURL, dbUser, dbPass);
            PreparedStatement pst = con.prepareStatement("INSERT INTO attendance
(student_id, date, status) VALUES (?, ?, ?)");
            pst.setString(1, sid);
            pst.setString(2, date);
            pst.setString(3, status);
            int i = pst.executeUpdate();

            if (i > 0)
                out.println("<h3>Attendance recorded successfully!</h3>");
            else
                out.println("<h3>Failed to record attendance.</h3>");

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

□ Output:

- Student fills attendance form on JSP.
- Servlet processes and saves to the DB.
- Displays success/failure message.

□ Learning Outcomes:

- Integrating JSP with Servlets.
- Using Servlets to handle database operations.
- Using forms to submit data dynamically.

□ Database Setup Scripts

`employees` table:

```
sql
CopyEdit
CREATE TABLE employees (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    department VARCHAR(100)
);
```

`attendance` table:

```
sql
CopyEdit
```

```
CREATE TABLE attendance (
    id INT AUTO_INCREMENT PRIMARY KEY,
    student_id VARCHAR(20),
    date DATE,
    status VARCHAR(10)
);
```