Task :9

JSP

Date: 28-10-2025 & 28-11-2025

Write a java program/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries. Insert the details of the employees who register with the web site, whenever a new employee clicks the submit button in the registration page.

Aim:

To write a JSP program that connects to a MySQL database, extracts data from the employee table, and allows inserting new employee details through a registration form.

Algorithm:

1. Create a database and table in MySQL.
2. Create a JSP registration page to collect employee details.
3. When the user clicks Submit, the data is inserted into the employee table.
4. Another JSP page retrieves and displays all employee records.
5. Use JDBC connectivity within JSP to perform SQL queries like INSERT and SELECT.

Step 1 — Create Database and Table in MySQL

```
CREATE DATABASE company_db;
USE company_db;
CREATE TABLE employee (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(50),
   email VARCHAR(50),
   department VARCHAR(30)
);
```

Step 2 — registration.jsp (Employee Registration Page)

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
  <title>Employee Registration</title>
</head>
<body style="font-family: Arial; text-align: center;">
  <h2>Employee Registration Form</h2>
  <form action="insertEmployee.jsp" method="post">
    <label>Employee Name:</label><br>
    <input type="text" name="name" required><br><br>

    <label>Email:</label><br>
    <input type="email" name="email" required><br><br>

    <label>Department:</label><br>
    <input type="text" name="department" required><br><br>

    <input type="submit" value="Register Employee">
  </form>

  <br><a href="viewEmployees.jsp">View All Employees</a>
</body>
</html>
```

Step 3 — insertEmployee.jsp (Insert Data into Database)

```
<%@ page import="java.sql.*" %>
```

```
<%
    String name = request.getParameter("name");
    String email = request.getParameter("email");
    String department = request.getParameter("department");

    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/company_db",
"root", "your_password");

        String sql = "INSERT INTO employee (name, email, department) VALUES (?, ?, ?)";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, name);
        pstmt.setString(2, email);
        pstmt.setString(3, department);

        int rows = pstmt.executeUpdate();

        if (rows > 0) {
%>
            <h3>Employee Registered Successfully!</h3>
            <a href="viewEmployees.jsp">View Employee List</a>
<%
        } else {
%>
```

```
            <h3>Error: Unable to register employee.</h3>
<%
        }
    } catch (Exception e) {
        out.println("Error: " + e.getMessage());
    } finally {
        try { if (pstmt != null) pstmt.close(); } catch (Exception e) {}
        try { if (conn != null) conn.close(); } catch (Exception e) {}
    }
%>
```

Step 4 — viewEmployees.jsp (Display All Records)

```
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
    <title>Employee Records</title>
</head>
<body style="font-family: Arial; text-align: center;">
    <h2>Registered Employee Details</h2>

    <table border="1" align="center" cellpadding="10">
        <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Email</th>
            <th>Department</th>
        </tr>
```

```
<%
   try {
      Class.forName("com.mysql.cj.jdbc.Driver");
      Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/company_db", "root",
"your_password");
      Statement stmt = conn.createStatement();

      String sql = "SELECT * FROM employee";
      ResultSet rs = stmt.executeQuery(sql);

      while (rs.next()) {
%>
      <tr>
         <td><%= rs.getInt("id") %></td>
         <td><%= rs.getString("name") %></td>
         <td><%= rs.getString("email") %></td>
         <td><%= rs.getString("department") %></td>
      </tr>
<%
      }
      conn.close();
   } catch (Exception e) {
      out.println("<tr><td colspan='4'>Error: " + e.getMessage() + "</td></tr>");
   }
%>
   </table>
```

```
   <br><a href="registration.jsp">Register New Employee</a>
</body>
</html>
```

Step 5 — Example SQL Queries to Experiment

You can try these inside your JSP or directly in MySQL:

```
-- Select all employees
SELECT * FROM employee;

-- Select employees from a specific department
SELECT * FROM employee WHERE department = 'IT';

-- Update an employee record
UPDATE employee SET department = 'HR' WHERE id = 2;

-- Delete an employee record
DELETE FROM employee WHERE id = 3;
```

Sample Output

registration.jsp

Employee Registration Form

[Name: John Doe]

[Email: john@gmail.com]

[Department: IT]

[Register Employee]

Result:

Thus, a JSP web application was successfully created to connect to MySQL, insert new employee details, and display all registered employees using JDBC connectivity and SQL queries**.**