

Experiment#		Student ID	
Date		Student Name	THANOS

13. Handling Form Data in JSP, Processing with Servlet, and Persisting into Database

Aim/Objective: To understand the how to fetch form data in JSP and handle the data by using Servlet and persist the data into database.

Description: The student will understand creating a web application that collects user input through a form in a JSP page, processes the submitted data using a servlet, and persists the data into a database using JDBC.

Pre-Requisites: Classes, Objects and Databases

Tools: Eclipse IDE for Enterprise Java and Web Developers

Pre-Lab:

- 1) How does JSP differ from servlet?

JSP vs Servlet:

- **JSP:** HTML + Java code, compiles to a servlet.
- **Servlet:** Java class that handles requests, returns responses.

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 1

Experiment#		Student ID	
Date		Student Name	THANOS

2) Explain how a Servlet handle form data?

Servlet handling form data:

- GET/POST request from the form.
- Use `request.getParameter("paramName")` to access form data.
- Process and send response back.

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 2

Experiment#		Student ID	
Date		Student Name	THANOS

In-Lab:

- 1) Develop a web application that allows users to submit their contact details through a form on a JSP page. The form data will be processed by a servlet and stored in a PostgreSQL database.

A. Set Up the Database

- a. Create a PostgreSQL database and a table to store the contact details.

B. Create a JSP Form:

- a. Develop a JSP page with a form to collect user contact details.

C. Create a Servlet to Handle Form Submission:

- a. Develop a servlet that processes the form data submitted from the JSP page.
- b. Use the servlet to validate and prepare the data for persistence.

D. Persist Data into the Database Using JDBC:

- a. Establish a connection to the PostgreSQL database.
- b. Insert the submitted form data into the database.

Procedure/Program:

A. Set Up the Database

```
CREATE DATABASE contact_db;
```

```
CREATE TABLE contact_details (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL,
    phone VARCHAR(15) NOT NULL
);
```

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 3

Experiment#		Student ID	
Date		Student Name	THANOS

B. Create a JSP Form

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>Contact Form</title>
</head>
<body>
    <h2>Contact Us</h2>
    <form action="ContactServlet" method="post">
        Name: <input type="text" name="name" required><br>
        Email: <input type="email" name="email" required><br>
        Phone: <input type="text" name="phone" required><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

C. Create a Servlet to Handle Form Submission

```
import java.io.IOException;
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("/ContactServlet")
public class ContactServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String name = request.getParameter("name");
        String email = request.getParameter("email");
        String phone = request.getParameter("phone");
```

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 4

Experiment#		Student ID	
Date		Student Name	THANOS

```

        if (name != null && email != null && phone != null) {
            DatabaseHelper.insertContact(name, email, phone);
            response.getWriter().println("Contact details submitted successfully.");
        } else {
            response.getWriter().println("Invalid input.");
        }
    }
}

```

D. Persist Data into the Database Using JDBC

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class DatabaseHelper {
    private static final String URL =
"jdbc:postgresql://localhost:5432/contact_db";
    private static final String USER = "root";
    private static final String PASSWORD = "Root@123";

    public static void insertContact(String name, String email, String phone) {
        String sql = "INSERT INTO contact_details (name, email, phone) VALUES (?, ?, ?)";
        try (Connection conn = DriverManager.getConnection(URL, USER,
PASSWORD);
            PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setString(1, name);
            pstmt.setString(2, email);
            pstmt.setString(3, phone);
            pstmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 5

Experiment#		Student ID	
Date		Student Name	THANOS

- 2) A Programmer, is developing a web-based calculator application for his client using servlets, JSP (Java Server Pages), and integrating with a database. The application allows users to perform basic arithmetic operations such as addition, subtraction, multiplication, division, and modulus.

Requirements and Implementation Steps:

Step 1: Design UI (VIEW) Using HTML

Step 2: Create a Calculator (MODEL) Class

Step 3: Create a Servlet (CONTROLLER) to Handle Requests

Step 4: Integrate with JSP for Dynamic Views

Step 5: Deployment and Testing

Procedure/Program:

Step 1: Design UI (VIEW) Using HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Calculator</title>
</head>
<body>
  <h1>Web-Based Calculator</h1>
  <form action="CalculatorServlet" method="post">
    <input type="text" name="num1" placeholder="Enter first number"
required>
    <select name="operation">
      <option value="add">+</option>
      <option value="subtract">-</option>
      <option value="multiply">*</option>
```

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 6

Experiment#		Student ID	
Date		Student Name	THANOS

```

        <option value="divide">/</option>
        <option value="modulus">%</option>
    </select>
    <input type="text" name="num2" placeholder="Enter second number"
required>
    <input type="submit" value="Calculate">
</form>
<div id="result">
    <h2>Result: <span id="output"></span></h2>
</div>
</body>
</html>

```

Step 2: Create a Calculator (MODEL) Class

```

public class Calculator {
    public double add(double a, double b) {
        return a + b;
    }

    public double subtract(double a, double b) {
        return a - b;
    }
}

```

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 7

Experiment#		Student ID	
Date		Student Name	THANOS

```
public double multiply(double a, double b) {
    return a * b;
}
```

```
public double divide(double a, double b) {
    if (b != 0) {
        return a / b;
    } else {
        throw new ArithmeticException("Division by zero");
    }
}
```

```
public double modulus(double a, double b) {
    return a % b;
}
}
```

Step 3: Create a Servlet (CONTROLLER) to Handle Requests

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

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 8

Experiment#		Student ID	
Date		Student Name	THANOS

@WebServlet("/CalculatorServlet")

```
public class CalculatorServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        double num1 = Double.parseDouble(request.getParameter("num1"));
        double num2 = Double.parseDouble(request.getParameter("num2"));
        String operation = request.getParameter("operation");

        Calculator calculator = new Calculator();
        double result = 0;

        switch (operation) {
            case "add":
                result = calculator.add(num1, num2);
                break;
            case "subtract":
                result = calculator.subtract(num1, num2);
                break;
            case "multiply":
                result = calculator.multiply(num1, num2);
                break;
            case "divide":
                result = calculator.divide(num1, num2);
                break;
        }
    }
}
```

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 9

Experiment#		Student ID	
Date		Student Name	THANOS

```
case "modulus":
```

```
    result = calculator.modulus(num1, num2);
```

```
    break;
```

```
}
```

```
request.setAttribute("result", result);
```

```
request.getRequestDispatcher("result.jsp").forward(request, response);
```

```
}
```

```
}
```

Step 4: Integrate with JSP for Dynamic Views

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
```

```
<html>
```

```
<head>
```

```
    <title>Calculation Result</title>
```

```
</head>
```

```
<body>
```

```
    <h1>Calculation Result</h1>
```

```
    <h2>Result: <%= request.getAttribute("result") %></h2>
```

```
    <a href="index.html">Back to Calculator</a>
```

```
</body>
```

```
</html>
```

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 10

Experiment#		Student ID	
Date		Student Name	THANOS

Step 5: Deployment and Testing

Deploy the application on a servlet container like Apache Tomcat
Ensure the web.xml is configured properly for servlet mapping

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 11

Experiment#		Student ID	
Date		Student Name	THANOS

✓ **Data and Results:**

Data:

User inputs two numbers and selects an arithmetic operation.

Result:

Server processes input, performs calculation, and displays the output.

✓ **Analysis and Inferences:**

Analysis:

Application follows MVC architecture for efficient request and response handling.

Inferences:

Project demonstrates successful integration of servlets, JSP, and database.

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 12

Experiment#		Student ID	
Date		Student Name	THANOS

VIVA-VOCE Questions (In-Lab):

- 1) What is the role of a servlet in handling form data submitted by a user in a JSP page?
- 2) What is the difference between the request and response objects in servlets?
- 3) Explain the process of sending form data from a JSP page to a servlet for processing.
- 4) What is the purpose of the web.xml deployment descriptor file in a Java web application?
- 5) How can you access the request parameters sent from a JSP form in a servlet?

1) Servlet processes form data and sends response.

2) `request` gets data; `response` sends data.

3) JSP form submits; servlet reads using `getParameter()`.

4) `web.xml` maps URLs, configures app settings.

5) Use `request.getParameter("name")` in servlet.

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 13

Experiment#		Student ID	
Date		Student Name	THANOS

Post-Lab:

- 1) Write a code snippet demonstrating the use of JSP and servlet to handle a login form.
Use a scenario where a user submits a login form, and the servlet validates the credentials. Your code should include the following:
 - a. Create a JSP page with a login form that collects username and password.
 - b. Implement a servlet that receives the form data, validates the credentials, and redirects the user to a success or error page based on the validation result.
 - c. Test your program by submitting different login credentials to ensure proper validation and redirection.

Procedure/Program:

login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>Login Form</title>
</head>
<body>
<h2>Login</h2>
<form action="LoginServlet" method="post">
  Username: <input type="text" name="username" required><br>
  Password: <input type="password" name="password" required><br>
  <input type="submit" value="Login">
</form>
</body>
</html>
```

LoginServlet.java

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

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 14

Experiment#		Student ID	
Date		Student Name	THANOS

```
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

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

        if ("admin".equals(username) && "password123".equals(password)) {
            response.sendRedirect("success.jsp");
        } else {
            response.sendRedirect("error.jsp");
        }
    }
}
```

success.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>Success</title>
</head>
<body>
    <h2>Login Successful!</h2>
    <p>Welcome to the application.</p>
</body>
</html>
```

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 15

Experiment#		Student ID	
Date		Student Name	THANOS

error.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>Error</title>
</head>
<body>
<h2>Login Failed!</h2>
<p>Invalid username or password. Please try again.</p>
</body>
</html>
```

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 16

Experiment#		Student ID	
Date		Student Name	THANOS

✓ **Data and Results:**

Data

A simple login system using JSP and Servlet for validation.

Result

Successful login redirects to success page, else redirects to error page.

✓ **Analysis and Inferences:**

Analysis

Form data is sent to servlet, credentials validated, redirection happens.

Inferences

JSP and Servlets efficiently handle login validations and user redirection.

Evaluator Remark (if Any):	Marks Secured ____ out of 50
	Signature of the Evaluator with Date

Evaluator MUST ask Viva-voce prior to signing and posting marks for each experiment.

Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103R	Page 17