Experiment#			
Date		Student ID	
		Student Name	

20. 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) Explain the concept of JSP and its role in web development. How does JSP differ from servlets? Describe the lifecycle of a JSP page.

JSP (Java Server page) is a technology that allows embedding Java code directly into HTML pages to create dynamically web content. It simplies the development of dynamic web applications by separating presentation and logic. Unlike servlets, which are Java classes handling HTTP requests, JSP files are easier. The JSP life cycles include translation completion, and destruction and JSP Page.

Experiment#			
Date		Student ID	
		Student Name	

2) Explain how a Servlet handle form data?

A servlet handles form data by ~~an~~ retrieving the submitted values from HTTP requests using methods like request.getParameter("parameter-name"). when a user submits a form the servlet process the data by accessing the input fields (eg textfield, radio buttons, checkboxes) from request object. This process typically happens with in the doPost() & doGet() methods of servlet.

Experiment#			
Date		Student ID	
		Student Name	

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:

① Postgresql :-

```
CREATE TABLE contact_details (
    id  social PRIMARY KEY ;
    name VARCAR (100);
    email VARCAR (100),
    Phone ( 15);
    message TEXT;
);
```

Experiment#			
Date		Student ID	
		Student Name	

(2) contactform.jsp:-

```
<!DOCTYPE html>
<html>
<head><title>contactform</title></head>
    <body>
      <h2>contact us</h2>
<form action="contactserved" method="post">
Name: <input type="text" name "required">
Email: <input type='email" name='email" reqd>
phone: <input type=" text' name :" phone"/>
message: <textarea name :"message "required>
    </textarea><br><input type=
    "submit" value :"submit">
    </form>
    </body>
    </html>
```

Experiment#			
Date		Student ID	
		Student Name	

(3) contactservlet.java

```
Public Class Contactservlet extends HttpServlet {
Protected void doPost (HttpServlet Request request,
        HttpServletResponse response) throws
Servlet Exception, IO Exception {
 String name = request.getparameter("name");
 String Email = request.getparameter("email");
 String message = request.getparameter("message");
try{Connection conn = DriverManager.getConnection(Url,
        user, pass);
Prepared statement pstmt=conn.
Prepared statement (Query) {
   pstmt.setString(1, name);
   pstmt.setString(2, email);
   pstmt.setString(3, phone);
 pstmt.setString(4, message);
 int rows = pstmt.executeUpdate();
```

(4) web.xml :-

```
<web-app xmlns="http://java.sun.com/xml/
    ns/javaee" version="3.0">
<servlet-name>contactservlet</
    servlet-name>
<servlet-class>contactservlet</servlet-
    class>
</servlet>
<servlet-mapping>
<servlet-name>contactservlet</servlet-
    name>
<url-pattern>/contactservlet</url-pattern>
</servlet-mapping>
</web-app>
```

Experiment#			
Date		Student ID	
		Student Name	

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:

① Calculator.jsp:-

```
<!DOCTYPE html>
<html>
<body>
<h2> Calculator </h2>
<form action="calculatorservlet" method="POST">
  <input type="number" name="num1" required>
<input type="number" name="num2" required>
  <select name="operation">
    <option value="add">Add</option>
    <option value="subtract">Subtract</option>
    <option value="multiply">multiply</option>
    <option value="divide">divide</option>
<option value="modulus">modulus</option>
</select>
```

② calculator.java:

```
Public class calculator {
    Public double add (double num1, double num2)
    { return num1+num2; }
Public double subtract (double num1, double num2){
    return num1-num2; }
Public double multiply (double num1, double
                        num2) {
    return num1*num2 ;}
Public double divide (double num1, double
    num2 ){ return num2 ==0?0 : num1/num2 ;}
Public double modulus (double num1,
    double num 2) { return num2==0?0:
                    num1 % num2 ;}
```

③ calculatorservlet.java: (servlet to handle requests)

```
Public class calculatorservlet extends
    HttpServlet {
Protected void doPost (HttpServletRequest
    HttpServletResponse response) throws
    ServletException, expection. {
```

Experiment#		Student ID	
Date		Student Name	

```
double num1 = Double.parseDouble(
        request.getparameter("num1"));
double num2 = Double.parseDouble(request.
        getparameter("num2"));
String operation = request.getparameter
                ("operation");
Calculator calc = new calculator();
double result = 0;
switch(operation) {
case "add": result = calc.add(num1,num2);
                break;
case "substract": result = calc.substract(num1,num2);
                break;
case "multiply": result = calc.multiply
        (num1,num2); break;
case "divide": result = calc.divide(num1,num2);
                break;
case "modules": result = calc.modules(num1,
        num2); break;
```

Experiment#			
Date		Student ID	
		Student Name	

(5) Database setup:-

CREATE DATABASE calculator db;

\c calculator;

CREATE TABLE calculation_history (

id serial PRIMARY KEY;

num1 DOUBLE, num2 DOUBLE, operation

VARCHAR(20), result DOUBLE

);

Experiment#			
Date		Student ID	
		Student Name	

✓ **Data and Results:**

The calculator application successfully performs arithmetic operations (addition, substraction, multiplication, division, modulus) and stores the calculations history in database.

✓ **Analysis and Inferences:**

The web application works as excepted, allowing users to perform calculations and view results while the database stores each calculation for record-keeping.

Experiment#		Student ID	
Date		Student Name	

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?**

A servlet processes form data submitted by a JSP page via HTTP requests, handling bussiness logic & forwarding responses.

2) **What is the difference between the request and response objects in servlets?**

The request object retrieves client data, while the response object sends the server's reply to client.

3) **Explain the process of sending form data from a JSP page to a servlet for processing.**

A JSP form send data to a servlet and using HTTP POST or GET method, which the servlet processes.

Experiment#		Student ID	
Date		Student Name	

4) What is the purpose of the web.xml deployment descriptor file in a Java web application?

It configures servlets, URL mappings, and other application settings for deployment.

5) How can you access the request parameters sent from a JSP form in a servlet?

Use request.getparameter("parameter Name") to retrieve from data in a servlet.

Experiment#			
Date		Student ID	
		Student Name	

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:

① login.JSP

```
<!DOCTYPE html>
<html>
<head>
<title>Login Form</title>
</head>
<body>
<h2>Login</h2>
<form action="LoginServlet" method="POST">
<label for="username">Username:</label>
<input type="text" name="username"
required /><br><br>
<label for="password">Password:</label>
```

② login servlet.java

```
Public class loginservlet extends HttpServlet {
    Protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws
    ServletException, IOException {
        String username = request.getParameter
                          ("username");
        String password = request.getParameter
                          ("password");
        If("admin".equals(username)&&"admin123".
        equals(password)){
        response.sendRequest("success.jsp");
        }else{
        response.sendDirect("error.jsp");
        }}
```

③ <!DOCTYPE html>

<html>

<head>

<title> Login Failed </title>

</head>

<body>

<h2> Login Failed </h2>

 Invalid username or password. please

try again

</body>

</html>

Experiment#		Student ID	
Date		Student Name	

✓ **Data and Results:**

The login form successfully captures the Username and password, the servlets validates the credentials and based on validation.

✓ **Analysis and Inferences:**

The application demonstrates simple form handling, validation, and redirection, in a JAVA web environment.

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.