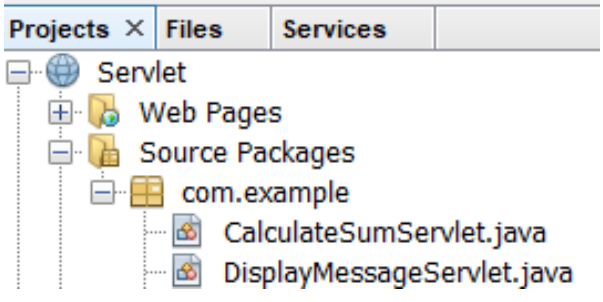
Java Servlet

Task 1: Simple Servlet - Display Static Message

Goal: Create a simple servlet that displays a static message to the user.

1. **Create a Java Servlet (DisplayMessageServlet)** that outputs a static message.
2. **Configure the servlet** using the @WebServlet annotation or the web.xml deployment descriptor.

```
package com.example;
import java.io.IOException;
import java.io.PrintWriter;
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("/getUserInput")
public class GetUserInputServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String username = request.getParameter("username");
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("<h1>Hello, " + username + "!</h1>");
    out.println("</body></html>");
    }
}
```



Deployment Descriptor (web.xml) (if not using annotations):

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<web-app xmlns=http://java.sun.com/xml/ns/javaee version="3.0">
<servlet>
<servlet-name>DisplayMessageServlet</servlet-name>
<servlet-class>com.example.DisplayMessageServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>DisplayMessageServlet</servlet-name>
<url-pattern>/displayMessage</url-pattern>
</servlet-mapping>
</web-app>
</root>
```

Task 2: Get User Input from Form and Display

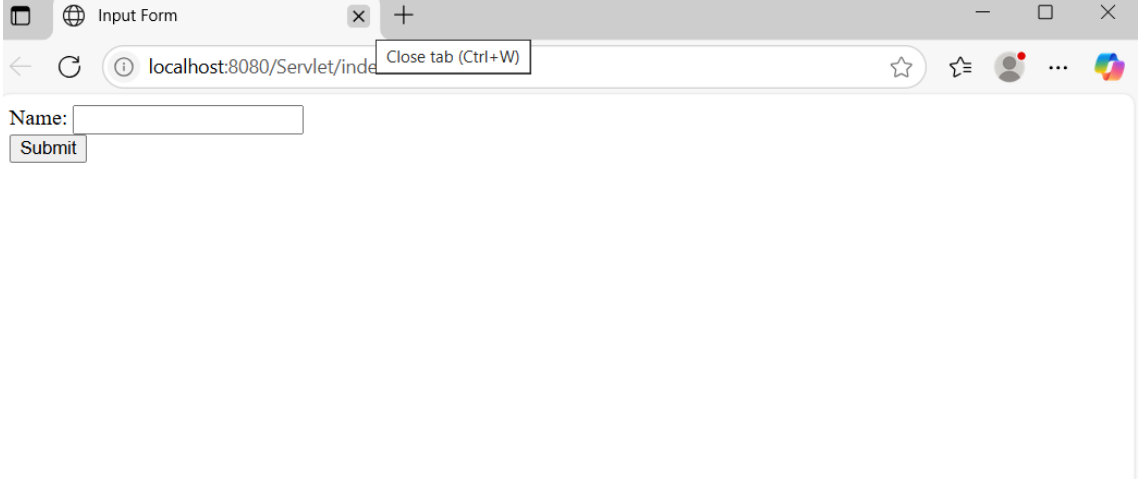
Goal: Create a servlet that receives user input from an HTML form and displays it back to the user.

Steps:

1. **Create an HTML form** to collect the user's name.
2. **Create a Servlet** (GetUserInputServlet) to handle the form submission and display the user's name.

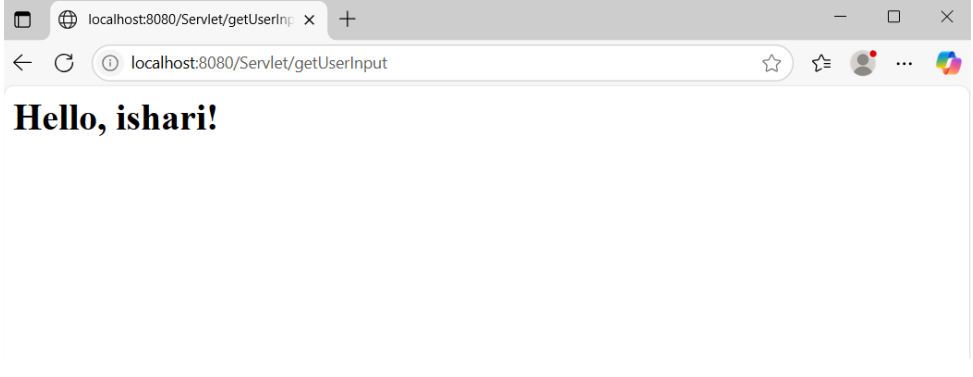
HTML Form (index.html):

```
<!DOCTYPE html>
<html>
<head><title>Input Form</title></head>
<body>
<form action="getUserInput" method="POST">
Name: <input type="text" name="username" required><br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```



Servlet Code (GetUserInputServlet.java):

```
package com.example;
import java.io.IOException;
import java.io.PrintWriter;
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("/getUserInput")

public class GetUserInputServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String username = request.getParameter("username");
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("<h1>Hello, " + username + "!</h1>");
    out.println("</body></html>");
    }
}
```



Task 3: Get Multiple Inputs, Perform Calculation, and Display Result

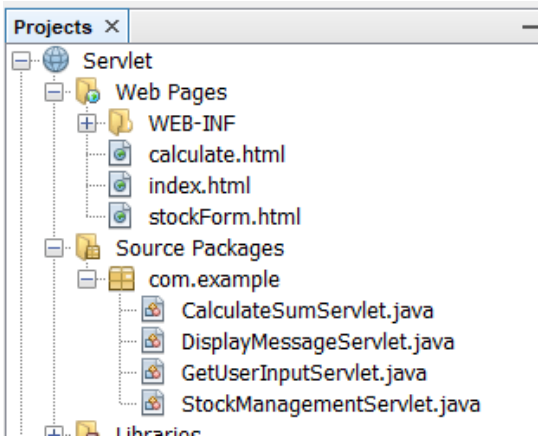
Create a servlet that receives multiple user inputs from a form, performs a calculation, and displays the result.

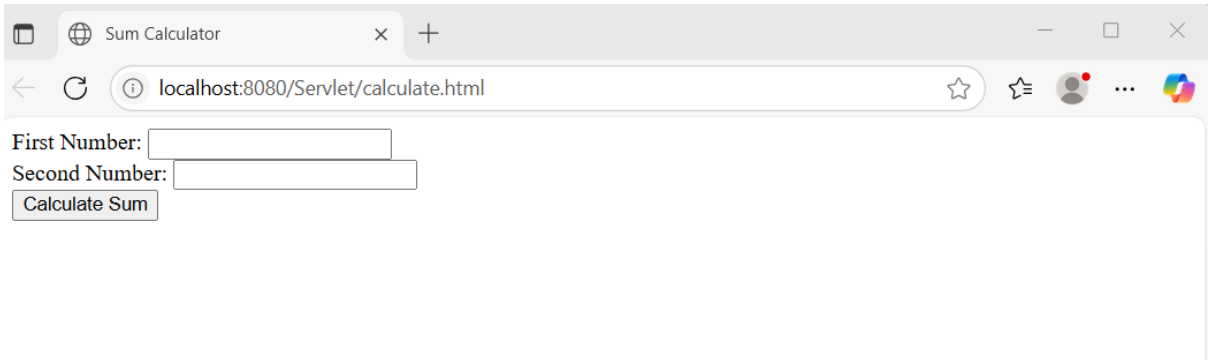
Steps:

1. **Create an HTML form** to take inputs for two numbers.

2. **Create a Servlet** (CalculateSumServlet) to calculate the sum of the numbers and display the result.

```
<!DOCTYPE html>
<html>
<head><title>Sum Calculator</title></head>
<body>
 <form action="calculateSum" method="POST">
 First Number: <input type="number" name="num1" required><br> Second Number: <input type="number" name="num2" required><br>
<input type="submit" value="Calculate Sum">
 </form>
</body>
</html>
```





Servlet Code (CalculateSumServlet.java):

```
package com.example;
import java.io.IOException;
import java.io.PrintWriter;
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("/calculateSum")
public class CalculateSumServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    int num1 = Integer.parseInt(request.getParameter("num1"));
    int num2 = Integer.parseInt(request.getParameter("num2"));
    int sum = num1 + num2;
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("<h1>The sum of " + num1 + " and " + num2 + " is: " + sum + "</h1>");
    out.println("</body></html>");
    }
}
```

Task 4: Java Servlet with Database CRUD Operations

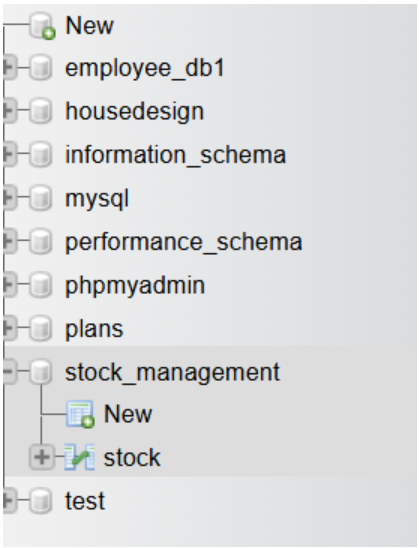
Implement a servlet that interacts with a database to perform **CRUD (Create, Read, Update, Delete)** operations for a stock management system.

Steps:

1. **Set up a database** with a table named stock (fields: id, product_name, quantity).
2. **Create a simple web form** to interact with the database (add, update, delete products).
3. **Create a Servlet** (StockManagementServlet) that handles database operations.

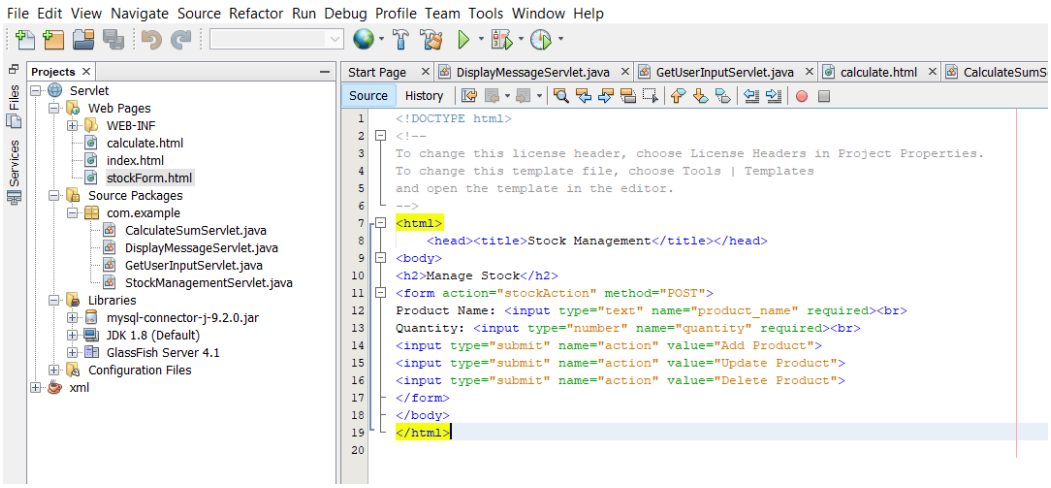
Database Setup (MySQL example):

```
CREATE DATABASE stock_management;

USE stock_management;

CREATE TABLE stock (

 id INT AUTO_INCREMENT PRIMARY KEY,

 product_name VARCHAR(255),

 quantity INT

);
```



HTML Form (stockForm.html):

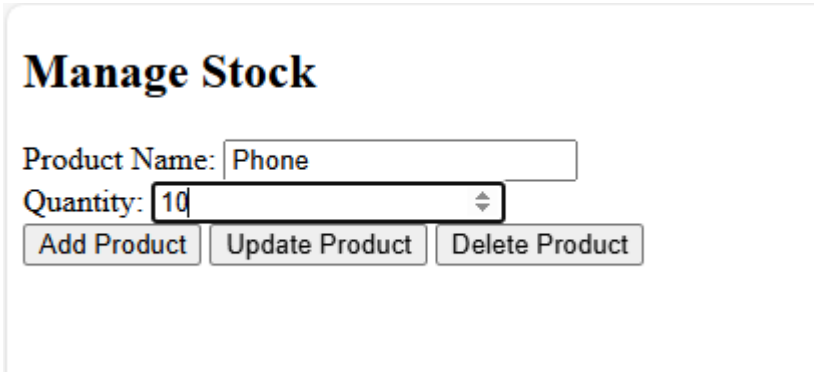
```
<!DOCTYPE html>
<html>
<head><title>Stock Management</title></head>
<body>
<h2>Manage Stock</h2>
<form action="stockAction" method="POST">
Product Name: <input type="text" name="product_name" required><br>
Quantity: <input type="number" name="quantity" required><br>
<input type="submit" name="action" value="Add Product">
<input type="submit" name="action" value="Update Product">
<input type="submit" name="action" value="Delete Product">
</form>
</body>
</html>
```



Servlet Code (StockManagementServlet.java):

```
package com.example;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
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("/stockAction")
public class StockManagementServlet extends HttpServlet {
private Connection getConnection() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/stock_management";
        String username = "root";
        String password = "root"; // replace with your database password
        return DriverManager.getConnection(url, username, password);
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        String action = request.getParameter("action");
        String productName = request.getParameter("product_name");
        int quantity = Integer.parseInt(request.getParameter("quantity"));

try (Connection conn = getConnection()) {
switch(action) {
        case "Add Product":
try (PreparedStatement stmt = conn.prepareStatement(
"INSERT INTO stock (product_name, quantity) VALUES (?, ?)")) {
        stmt.setString(1, productName);
        stmt.setInt(2, quantity);
```

```
            stmt.executeUpdate();
            response.getWriter().write("<h1>Product Added Successfully</h1>");
}
break;
case "Update Product":
try (PreparedStatement stmt = conn.prepareStatement(
"UPDATE stock SET quantity = ? WHERE product_name = ?")) {
            stmt.setInt(1, quantity);
            stmt.setString(2, productName);
            stmt.executeUpdate();
            response.getWriter().write("<h1>Product Updated Successfully</h1>");
}
break;
case "Delete Product":
try (PreparedStatement stmt = conn.prepareStatement(
            "DELETE FROM stock WHERE product_name = ?")) {
            stmt.setString(1, productName);
            stmt.executeUpdate();
            response.getWriter().write("<h1>Product Deleted Successfully</h1>");
}
break;
default:
            response.getWriter().write("<h1>Invalid Action</h1>");
            }
} catch (SQLException e) {

e.printStackTrace();
response.getWriter().write("<h1>Database Error: " + e.getMessage() + "</h1>");
                        }
            }
}
```



Product Added Successfully

Back to Form

	id	product_name	quantity
Edit Copy Delete	1	Phone	10

Task 5: Display Data from Database on Another Web Page

Goal:

Extend the Stock Management System to display the list of all products on another webpage.

Steps:

1. **Create a Servlet** to fetch and display all products from the database.
2. **Create a new HTML page** to show the product list.

Servlet Code (DisplayProductsServlet.java):

```
	@WebServlet("/displayProducts")
public class DisplayProductsServlet extends HttpServlet {
	protected void doGet(HttpServletRequest request, HttpServletResponse response)
	throws ServletException, IOException {
	response.setContentType("text/html");
	PrintWriter out = response.getWriter();
	try (Connection conn = getConnection()) {
	Statement stmt = conn.createStatement();
	ResultSet rs = stmt.executeQuery("SELECT * FROM stock");
	out.println("<html><body><h1>Stock List</h1>");
	while (rs.next()) {
	out.println("<p>" + rs.getString("product_name") + ": " + rs.getInt("quantity") +
	"</p>");
}
	out.println("</body></html>");
	} catch (SQLException e) {
	e.printStackTrace();
	out.println("<h1>Database Error</h1>");
		}
	}
}
```