SUBJECT CODE : AI23431 WEB TECHNOLOGY AND MOBILE APPLICATION
JAYAKRISHNAN AIML FA

ROLL NO : 231501067 Artificial Intelligence Data Science and Artificial Intelligence Machine Learning

EXPERIMENT : 05

Write a Servlet to demonstrate the difference between HTTP GET and POST methods by creating a form and handling requests accordingly

AIM : The aim of this servlet is to demonstrate the differences between HTTP GET and POST methods. A form will be created where a user can input their data, and based on the method (GET or POST) selected, the servlet will handle the request differently.

ALGORITHM :

- **Create the HTML Form**:
  - Create an HTML form that includes both GET and POST methods.
  - The form will include a text input field and a submit button.

- There will be two buttons: one for GET method and one for POST method.

- **Create the Servlet**:
  - The servlet will handle both GET and POST requests.
  - It will differentiate the request based on the HTTP method (GET or POST) used.
  - For GET requests, the servlet will retrieve data and display it back.
  - For POST requests, the servlet will process the submitted data and display a confirmation.

- **Configure the Servlet**:
  - Configure the servlet in the web.xml (deployment descriptor) or use annotations to map the servlet URL.

- **Implement the Servlet Logic**:
  - In the servlet's doGet and doPost methods, handle the respective logic for GET and POST requests.
- **Compile and Deploy the Servlet**:
- Compile the Java Servlet and deploy it in a servlet container like Apache Tomcat.

STEP BY STEP GUIDE:

1. Create the HTML Form (index.html):

This HTML file provides the form where the user can input data and select the HTTP method (GET or POST) for submission.

```
<!DOCTYPE html>
<html>
<head>
  <title>GET vs POST Method Demonstration</title>
</head>
<body>
```

```
 <h2>GET and POST Method Example</h2>

 <!-- GET Form -->
 <h3>GET Method</h3>
 <form action="DemoServlet" method="GET">
    Enter your name: <input type="text" name="name"><br><br>
    <input type="submit" value="Submit using GET">
 </form>

 <hr>

 <!-- POST Form -->
 <h3>POST Method</h3>
 <form action="DemoServlet" method="POST">
    Enter your name: <input type="text" name="name"><br><br>
    <input type="submit" value="Submit using POST">
 </form>


</body>
</html>
```

2. **Create the Java Servlet (DemoServlet.java):**

This Java servlet will handle GET and POST requests based on the user's input

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

```
public class DemoServlet extends HttpServlet {
  // Handle GET request
  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
     // Get data from the query string (URL parameters)
     String name = request.getParameter("name");

     // Set the content type and get the PrintWriter to write the response
     response.setContentType("text/html");
     PrintWriter out = response.getWriter();

     // Respond to the client with the GET request result
     out.println("<html><body>");
     out.println("<h2>GET Request Result</h2>");
     if (name != null && !name.isEmpty()) {
        out.println("<p>Hello, " + name + "! You submitted using the GET method.</p>");
     } else {
        out.println("<p>Please provide your name in the GET form.</p>");
     }
     out.println("</body></html>");
  }

  // Handle POST request
  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
     // Get data from the form (POST request body)
     String name = request.getParameter("name");

     // Set the content type and get the PrintWriter to write the response
     response.setContentType("text/html");
     PrintWriter out = response.getWriter();
```

```
        // Respond to the client with the POST request result
        out.println("<html><body>");
        out.println("<h2>POST Request Result</h2>");
        if (name != null && !name.isEmpty()) {
            out.println("<p>Hello, " + name + "! You submitted using the POST method.</p>");
        } else {
            out.println("<p>Please provide your name in the POST form.</p>");
        }
        out.println("</body></html>");
    }
}
```

3. **Configure the Servlet (web.xml):**

If you are using a web.xml configuration file, you must add the servlet mapping to route requests to your servlet.

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
     http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" version="3.0">

  <servlet>
     <servlet-name>DemoServlet</servlet-name>
     <servlet-class>DemoServlet</servlet-class>
  </servlet>

  <servlet-mapping>
     <servlet-name>DemoServlet</servlet-name>
     <url-pattern>/DemoServlet</url-pattern>
  </servlet-mapping>
```

</web-app>

Alternatively, if you're using annotations, you can replace the web.xml configuration with the following annotation in your DemoServlet.java:

java

@WebServlet("/DemoServlet")

4. **Deploy the Application:**

- Compile the Java Servlet (DemoServlet.java).
- Place the servlet in the WEB-INF/classes directory.
- If you're using a servlet container like Apache Tomcat, deploy the web.xml and index.html in the appropriate web application structure.
- Start the server and access index.html through the browser.

RESULT:

After deploying the servlet, you can open the browser and navigate to the form (index.html).

- If the user submits the form using the GET method, the URL will change to include the submitted parameters in the query string (e.g., ?name=John). The doGet method will handle the request and display the result.

- If the user submits the form using the POST method, the data will be sent in the body of the request, and the doPost method will process and display the result.\

OUTPUT :