Ex No: 5
Date:

HTTP POST AND GET METHODS

Aim:

To demonstrate the difference between the HTTP GET and POST methods using a Java Servlet, where a form will be created and requests will be handled accordingly by the GET and POST methods.

Algorithm

1. **Create an HTML Form**:
   - The form will have two buttons: one for sending a GET request and one for sending a POST request. The form will also contain a field to input some text.
2. **Handle GET Request**:
   - In the servlet, when a GET request is received, display the value of the parameter sent via the URL (query string) on the webpage.
3. **Handle POST Request**:
   - When a POST request is received, retrieve and display the data sent in the body of the request.
4. **Result**:
   - The user will see different results depending on whether they use the GET or POST method to submit the form.

CODE:

HTML :

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>GET vs POST Example</title>
<style>
body {
font-family: Arial, sans-serif; background-color: #f4f4f9; color: #333;

margin: 0;
padding: 0;
}


h2 {
color: #4CAF50; text-align: center; margin-top: 50px;
}

.container { width: 50%;
margin: 0 auto; background-color: #fff; padding: 30px;
border-radius: 8px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
.form-group {
margin-bottom: 20px;
}
label {
font-size: 16px; color: #555;
}


input[type="text"] { width: 100%;

padding: 10px; margin-top: 5px; border-radius: 4px;
border: 1px solid #ccc; box-sizing: border-box;
}


button {
background-color: #4CAF50; color: white;
padding: 10px 20px; border: none;
border-radius: 4px; cursor: pointer; font-size: 16px;
}


button:hover {
background-color: #45a049;
}


.form-container { margin-bottom: 30px;
}


hr {
border: 1px solid #ddd;
}


.result {

background-color: #f9f9f9; border-left: 4px solid #4CAF50; padding: 20px;
margin-top: 20px; font-size: 16px; border-radius: 4px;
}
</style>
</head>
<body>
<div class="container">
<h2>GET vs POST Request Demonstration</h2>


<!-- Form to demonstrate GET method -->
```

```
<div class="form-container">
<h3>GET Method</h3>
<form action="DemoServlet" method="GET">
<div class="form-group">
<label for="data">Enter Data (GET):</label>
<input type="text" name="data" required>
</div>
<button type="submit">Submit (GET)</button>
</form>
</div>


<hr>


<!-- Form to demonstrate POST method -->
<div class="form-container">
<h3>POST Method</h3>

<form action="DemoServlet" method="POST">
<div class="form-group">
<label for="data">Enter Data (POST):</label>
<input type="text" name="data" required>
</div>
<button type="submit">Submit (POST)</button>
</form>
</div>


<!-- Result section will be dynamically updated -->
<div class="result" id="result">
<!-- Display GET or POST request result here -->
</div>
</div>
</body>
</html>
```

Servlet.java:

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

public class DemoServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String data = request.getParameter("data");

        out.println("<html><body>");
        out.println("<h2>GET Request Received</h2>");

        if (data != null && !data.isEmpty()) {
            out.println("<p>Data received via GET method: " + data + "</p>");
        } else {
            out.println("<p>No data received in GET request.</p>");
        }

        out.println("</body></html>");
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String data = request.getParameter("data");

        out.println("<html><body>");
        out.println("<h2>POST Request Received</h2>");

        if (data != null && !data.isEmpty()) {
            out.println("<p>Data received via POST method: " + data + "</p>");
        } else {
            out.println("<p>No data received in POST request.</p>");
        }

        out.println("</body></html>");
    }
}
```

Web.xml config

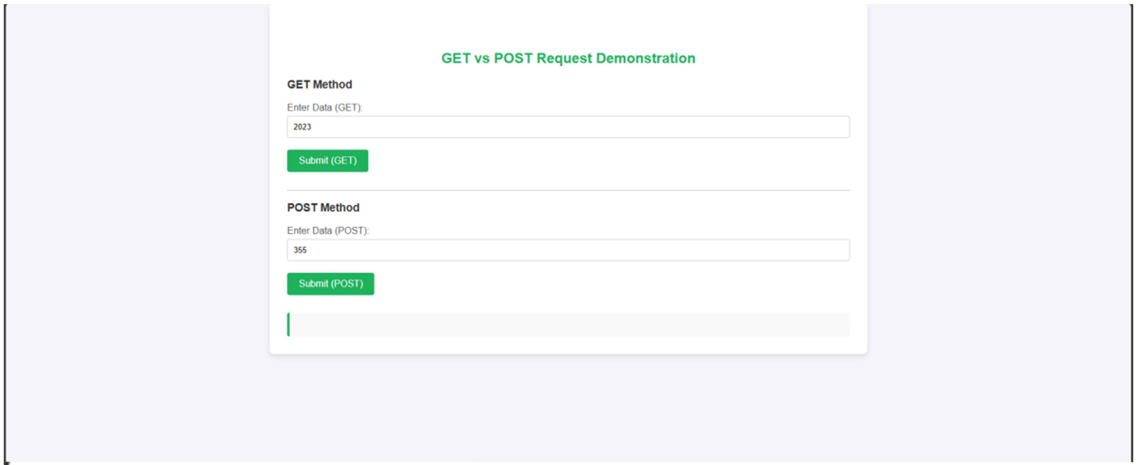
```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
         http://java.sun.com/xml/ns/javaee/web-app_3_1.xsd"
         version="3.1">

    <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>
```

OUTPUT :



GET vs POST Request Demonstration

GET Method

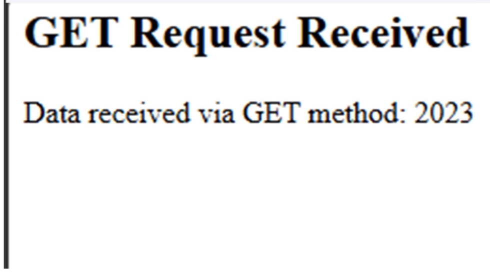
Enter Data (GET):

2023

POST Method

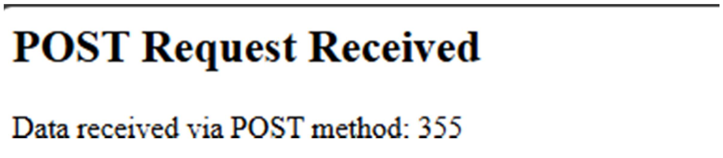
Enter Data (POST):

355



GET Request Received

Data received via GET method: 2023



POST Request Received

Data received via POST method: 355

RESULT :

The Java Servlet application demonstrating the distinction between HTTP GET and POST methods, with a form designed to submit data via both methods and corresponding handling logic implemented within the servlet, has been successfully created and tested. The application effectively showcases the different behaviors and use cases of GET and POST requests.