

EXP NO: 5

Servlet Example Demonstrating HTTP GET vs POST Methods with Form Handling

AIM:

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

CODE:

SignUp.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Sign_up</title>
</head>
<body>
<form action="/SignUpdetails" method="post">
  <label for="fname">FirstName:</label>
  <input type="text" name="fname" placeholder="Enter Your First Name"><br><br>

  <label for="lname">LastName:</label>
  <input type="text" name="lname" placeholder="Enter Your First Name"><br><br>

  <label for="age">Age:</label>
  <input type="number" name="age" placeholder="Enter Your Age"><br><br>

  <label for="date">Date Of Birth:</label>
  <input type="date" name="datee"><br><br>

  <label for="gender">Male</label>
  <input type="radio" name="gender"><br><br>
  <label for="gender">Female</label>
  <input type="radio" name="gender"><br><br>
```

```
<label for="address">Address:</label>
<textarea name="address" ></textarea>

<input type="submit" value="Sign-Up">

</form>
</body>
</html>
```

SignupDetails.java

```
package com.karthick.servlet;

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;

/**
 * Servlet implementation class SignUpdetails
 */
@WebServlet("/SignUpdetails")
public class SignUpdetails extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public SignUpdetails() {
        super();
        // TODO Auto-generated constructor stub
    }
}
```

```
}
```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

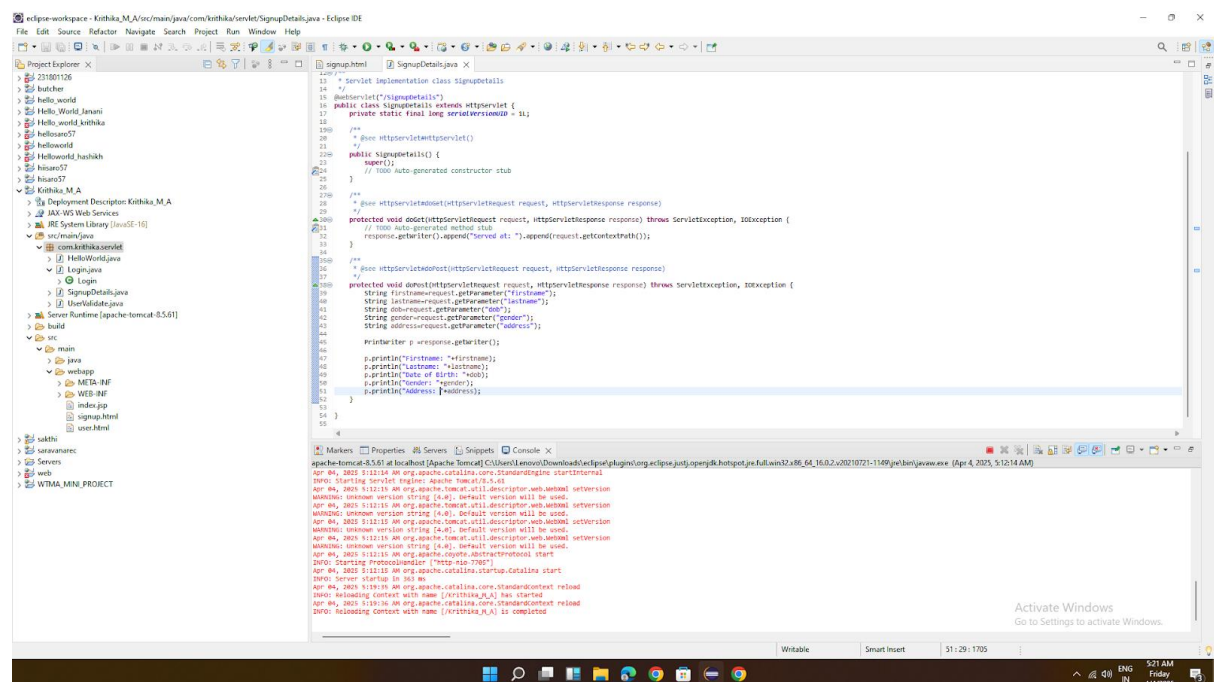
```
String firstname=request.getParameter("fname");  
String lastname=request.getParameter("lname");  
int age=Integer.parseInt(request.getParameter("age"));  
String date=request.getParameter("date");  
String gender=request.getParameter("gender");  
String address=request.getParameter("address");
```

```
PrintWriter p = response.getWriter();  
p.println("FirstName :"+firstname);  
p.println("LastName :"+lastname);  
p.println("Age :"+age);  
p.println("Date Of Birth :"+date);  
p.println("Gender :"+gender);  
p.println("Address :"+address);
```

```
}
```

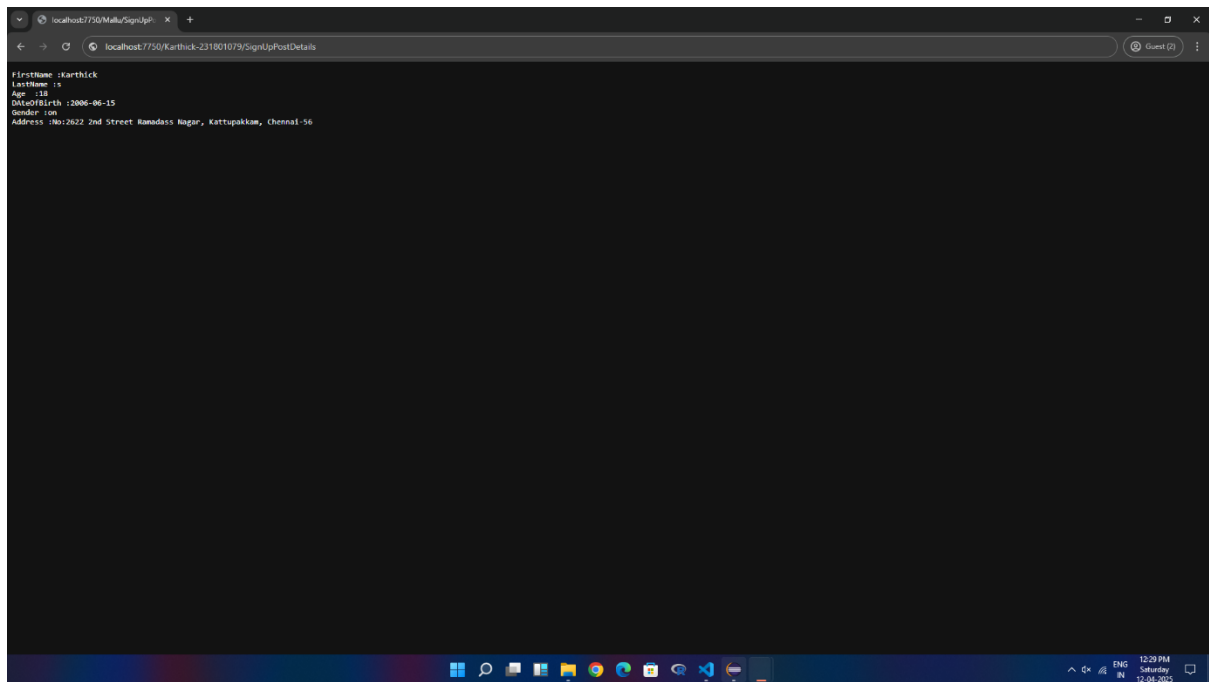
```
}
```

OUTPUT:



The screenshot shows the Eclipse IDE with the file `SignUpDetails.java` open. The code is a servlet implementation that handles POST requests. The console output shows the following sequence of events:

```
2023.04.04 11:12:14 AM org.apache.catalina.core.StandardEngine startInternal  
INFO: Starting Servlet engine: Apache Tomcat/9.0.54  
2023.04.04 11:12:15 AM org.apache.tomcat.util.descriptor.web.MimeMap setVersion  
WARNING: Unknown version string (4.0), default version will be used.  
2023.04.04 11:12:15 AM org.apache.tomcat.util.descriptor.web.MimeMap setVersion  
WARNING: Unknown version string (4.0), default version will be used.  
2023.04.04 11:12:15 AM org.apache.tomcat.util.descriptor.web.MimeMap setVersion  
WARNING: Unknown version string (4.0), default version will be used.  
2023.04.04 11:12:15 AM org.apache.tomcat.util.descriptor.web.MimeMap setVersion  
WARNING: Unknown version string (4.0), default version will be used.  
2023.04.04 11:12:15 AM org.apache.coyote.AbstractProtocol start  
INFO: Starting ProtocolHandler ["http-apr-7005"]  
2023.04.04 11:12:15 AM org.apache.catalina.startup.Catalina start  
INFO: Server startup in 0.046 ms  
2023.04.04 11:12:19 AM org.apache.catalina.core.StandardContext reload  
INFO: Reloading context with name [/Kithika_M_A] has started  
2023.04.04 11:12:19 AM org.apache.catalina.core.StandardContext reload  
INFO: Reloading context with name [/Kithika_M_A] is completed
```

RESULT:

Thus, the difference between HTTP GET and POST Method is Successful demonstrated by creating a form and handling requests.