

## Experiment\_6 Web Technology

Name: Veeransh Shah

Reg ID: 221070063

### Aim:

To invoke servlets from HTML formats.

### Theory:

Java Servlets are server-side Java programs that handle requests and responses in a web application environment. They are used for managing interactions with web clients, processing form data, managing sessions, and creating dynamic content.

To configure and use Java Servlets, follow these steps:

1. Set Up Environment: Install a servlet container like Apache Tomcat and configure your Java IDE.
2. Write Servlet Code: Create a Java class that extends `HttpServlet` and override methods like `doGet()` and `doPost()` to handle requests.
3. Compile Servlet: Use the Java compiler or your IDE to compile the servlet.
4. Configure Web Application: Create a `web.xml` file to define servlet mappings and configurations.
5. Deploy and Run: Place compiled classes and configuration files in the appropriate directories of your servlet container, start the server, and access your servlet via a web browser.

### Code:

JAVA:

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

public class MyServlet extends HttpServlet {
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String name = request.getParameter("name");
    out.println("<h1>Hello, " + name + "</h1>");
}
}
```

## HTML:

```
<!DOCTYPE html>
<html>
<head>
    <title>Sample Form</title>
</head>
<body>
    <form action="MyServlet" method="POST">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name"><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

## Conclusion:

This example illustrates the interaction between a client-side HTML form and a server-side Java servlet. It provides a straightforward demonstration of how data can be submitted from a web form, processed on the server using Java, and dynamically included in the HTML response. The code serves as a foundational example for understanding and building more complex web applications using servlets and HTML forms.