Experiment 4: Servlet to Accept and Display User's Name and Age

Aim:

To create a **web form** that accepts a user's name and age, and write a **Java Servlet** to process and display the submitted data in the browser.

Algorithm:

1. **Start**

2. Create an HTML form (`index.html`) with input fields for `name` and `age`.

3. Set the form's `action` to point to the servlet URL and method to `POST`.

4. Create a Java class `UserDataServlet` that extends `HttpServlet`.

5. Override the `doPost()` method to process the form data.

6. Retrieve form data using `request.getParameter()`.

7. Set the response type and use `PrintWriter` to display the data.

8. Compile the servlet and deploy it along with the form in a web application directory.

9. Configure `web.xml` with servlet mappings.

10. Deploy on Apache Tomcat and test using the browser.

11. **End**

Code:

`index.html`

```html
<!DOCTYPE html>
<html>
<head>
    <title>User Form</title>
</head>
<body>
    <h2>Enter Your Details</h2>
    <form action="userdata" method="post">
        Name: <input type="text" name="username"><br><br>
        Age: <input type="text" name="age"><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

UserDataServlet.java

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

public class UserDataServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String name = request.getParameter("username");
        String age = request.getParameter("age");

        out.println("<h1>User Data Received</h1>");
        out.println("<p>Name: " + name + "</p>");
        out.println("<p>Age: " + age + "</p>");
    }
```

```
}
```

web.xml (inside WEB-INF)

xml
CopyEdit

```xml
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee" version="3.1">
    <servlet>
        <servlet-name>UserData</servlet-name>
        <servlet-class>UserDataServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>UserData</servlet-name>
        <url-pattern>/userdata</url-pattern>
    </servlet-mapping>
</web-app>
```

OUTPUT

Enter Your Details

Name: Ishana Sabrish

Age: 18

Submit

Result:

After submitting the form, the browser displays:

```
User Data Received
Name: <entered name>
Age: <entered age>
```

Thus, the web form and servlet were successfully implemented and deployed to accept and display user input.