Aim

To develop a Java Servlet that accepts user credentials through an HTML form and displays a personalized welcome message upon successful login.

Objectives

1. To understand the interaction between HTML forms and Java servlets.
2. To learn how to handle POST requests using HttpServlet.
3. To implement basic user authentication using servlets.
4. To dynamically generate HTML content based on user input.

Code

HTML Code (login.html)

```html
html
CopyEdit
<!DOCTYPE html>
<html>
<head>
    <title>Login Page</title>
</head>
<body>
    <h2>Login</h2>
    <form action="LoginServlet" method="post">
        Username: <input type="text" name="username" required><br><br>
        Password: <input type="password" name="password" required><br><br>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

Servlet Code (LoginServlet.java)

```java
java
CopyEdit
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;

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

    private static final String VALID_USERNAME = "admin";
    private static final String VALID_PASSWORD = "password123";
```

```java
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    String username = request.getParameter("username");
    String password = request.getParameter("password");

    if (VALID_USERNAME.equals(username) && VALID_PASSWORD.equals(password)) {
        out.println("<h2>Welcome, " + username + "!</h2>");
    } else {
        out.println("<h2>Invalid username or password.</h2>");
    }

    out.close();
}
}
```

Output

◇ Case 1: Successful Login

If the user enters:

- **Username**: admin
- **Password**: password123

The output will be:

```
pgsql
CopyEdit
Welcome, admin!
```

◇ Case 2: Failed Login

If the user enters incorrect credentials:

```
nginx
CopyEdit
Invalid username or password.
```

Learning Outcome

After completing this exercise, students will be able to:

- Understand the process of sending form data from a client (HTML) to a server (Servlet).
- Create and configure a Java servlet using annotations.
- Handle HTTP POST requests using doPost() method.
- Dynamically generate responses based on user input.
- Apply basic authentication logic in web applications.

Aim

To write a Java Servlet that accepts user credentials from an HTML form and displays a personalized message upon successful authentication.

Objectives

- To collect user input via an HTML form.
- To process form data using a Java servlet.
- To validate login credentials.
- To dynamically generate a response based on the login result.

Code

HTML Form (`login2.html`)

```
html
CopyEdit
<!DOCTYPE html>
<html>
<head>
    <title>Login Page - Servlet 2</title>
</head>
<body>
    <h2>User Login</h2>
    <form action="LoginServlet2" method="post">
        Username: <input type="text" name="username" required><br><br>
        Password: <input type="password" name="password" required><br><br>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

Servlet Code (`LoginServlet2.java`)

```
java
CopyEdit
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;

@WebServlet("/LoginServlet2")
public class LoginServlet2 extends HttpServlet {
    private static final long serialVersionUID = 1L;
```

```
    // Alternate set of hardcoded credentials
    private static final String VALID_USERNAME = "user1";
    private static final String VALID_PASSWORD = "welcome123";

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

        String username = request.getParameter("username");
        String password = request.getParameter("password");

        if (VALID_USERNAME.equals(username) &&
VALID_PASSWORD.equals(password)) {
            out.println("<h2>Hello, " + username + "! Welcome back!</h2>");
        } else {
            out.println("<h2>Login failed. Invalid credentials.</h2>");
        }

        out.close();
    }
}
```

Output

◇ Successful Login:

For:

- Username: `user1`
- Password: `welcome123`

Output:

```
CopyEdit
Hello, user1! Welcome back!
```

◇ Invalid Login:

Any wrong username/password:

```
nginx
CopyEdit
Login failed. Invalid credentials.
```

Learning Outcome

From this task, you will learn:

- How to build a basic login system using servlets.
- How servlets handle form data with `doPost()`.
- How to conditionally generate HTML responses.
- Basics of secure credential handling in a web app (though this example uses hardcoded values).