POLITEKNIK MALAYSIA

DEPARTMENT OF INFORMATION & COMMUNICATION TECHNOLOGY

LAB ACTIVITY 7 : DATA VALIDATION AND SERVER RESPONSE IN DYNAMIC WEB PAGES

COURSE INFORMATION

Course Name	Java-Based Application Development
Topic	3.0 Introduction to Java Server Pages (JSP)
Lab Activity No.	Lab 7
Duration	2 – 3 Hours

SOFTWARE / HARDWARE REQUIREMENTS

Software	Hardware
Netbeans GlassFish	Laptop / PC

LEARNING OUTCOMES

- Explain the purpose of data validation in dynamic web pages.
- Perform client-side validation using JavaScript.
- Perform server-side validation using Servlet/JSP.
- Display appropriate validation messages to users.
- Set HTTP response headers using JSP.
- Use HTTP redirect response in JSP.

Task 1 : Client-Side Validation Using Javascript

Scenario

SweetDrink Sdn. Bhd. wants to improve user experience by validating all input data before it is submitted to the server using JavaScript in a JSP page.

Requirement

a. Create a JSP page that contains a login form interface for user input.

The form must include:

- Username
- Password
- Phone Number

b. Apply JavaScript Validation

Add JavaScript validation to check input before form submission.

Validation Rules

Field	**Validation**
Username	Cannot be empty
Password	Minimum 6 characters
Phone Number	Numbers only

INSTRUCTION

Step 1

Create login.jsp and type the following code to design a user login form interface.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
   <title>Login Form</title>
     <!—insert JavaScript -->
<head>

<body>

   <h2>Login Form</h2>

   <form name="myForm" action="process.jsp" method="post"
       onsubmit="return validateForm()">

      Username:
      <input type="text" name="username"><br><br>

      Password:
      <input type="password" name="password"><br><br>

      Phone Number:
      <input type="text" name="phone"><br><br>

      <input type="submit" value="Login">

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

Figure 1 : Code snippet of login.jsp

Activity: Paste screenshot evidence

Step 2

Type the **validateForm()** function in the <head> section using JavaScript.

```
  <script>
    function validateForm() {

      var username = document.forms["myForm"]["username"].value;
      var password = document.forms["myForm"]["password"].value;
      var phone = document.forms["myForm"]["phone"].value;

      if(username == "") {
        alert("Username cannot be empty");
        return false;
      }

      if(password.length < 6) {
        alert("Password must be at least 6 characters");
        return false;
      }
      if(!phone.match(/^[0-9]+$/)) {
        alert("Phone number must contain numbers only");
        return false;
      }
      return true;
    }
  </script>
</head>
```

Figure 2 : Code snippet of validateForm()

Activity: Paste screenshot evidence

Task 2 : Server-Side Validation Using Servlet/JSP.

Scenario

SweetDrink Sdn. Bhd. also wants all submitted data to be validated at the server side using JSP to improve system security.

Requirements

a. Create a JSP page that contains a login form interface for user input.

The form must include:

- Username
- Password
- Phone Number

b. Create a JSP page named process.jsp to validate the following input:

Field	**Validation**
Username	Cannot be empty
Password	Minimum 6 characters
Phone Number	Numbers only

Step 1

Create login.jsp and type the following code to design a user login form interface.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
   <title>Login Form</title>
</head>
<body>
<h2>User Login</h2>
<form action="process.jsp" method="post">
   Username:
   <input type="text" name="username"><br><br>
   Password:
   <input type="password" name="password"><br><br>
   Phone Number:
   <input type="text" name="phone"><br><br>
   <input type="submit" value="Login">
</form>
</body>
</html>
```

Figure 1 : Code snippet of login.jsp

Activity: Paste screenshot evidence

Step 2

Create **process.jsp** and type the following code to process and validate the user input data submitted from the login form.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%
   String username = request.getParameter("username");
   String password = request.getParameter("password");
   String phone = request.getParameter("phone");

   if(username == null || username.equals("")) {

%>
      <h3>Username cannot be empty</h3>
<%
   } else if(password == null || password.length() < 6) {

%>
      <h3>Password must be at least 6 characters</h3>
<%
   } else if(phone == null || !phone.matches("[0-9]+")) {

%>
      <h3>Phone number must contain numbers only</h3>
<%
   } else {
%>
      <h2>Login Successful</h2>
<%
   }
%>
```

Figure 2 : Code snippet of process.jsp

d. Retrieve form data using request.getParameter().

e. Validate user input using JSP scriptlet.

f. Display validation messages if input is invalid.

g. Display “Login Successful” if validation passes.

Activity: Paste screenshot evidence

Task 3 : Builds JSP file to generate Server Response: HTTP Response Headers

Scenario

SweetDrink Sdn. Bhd. has developed an online login system. After a successful login, the system must:

- Send proper HTTP response headers (e.g., content type)
- Display response information to the user
- Redirect users to a dashboard page automatically after login

The system must ensure that all responses follow proper HTTP response structure.

Requirement

a. Create login.jsp

Create a login form with:

- Username
- Password

The form should submit data to process.jsp.

b. Create process.jsp

Create a JSP page to:

- Process login data
- Set HTTP response headers
- Display response output
- Redirect user to **dashboard.jsp** if login is successful

c. Setting Response Headers

In process.jsp, set response header:

- Content-Type: text/html

```
response.setContentType("text/html");
```

d. Use Redirect Response

If login is successful, redirect user to dashboard:

```
response.sendRedirect("dashboard.jsp");
```

Instructions

Step 1

Create **login.jsp** and type the following code to design a user login form interface.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
   <title>Login Page</title>
</head>
<body>

<h2>User Login</h2>

<form action="process.jsp" method="post">
   Username:
   <input type="text" name="username"><br><br>

   Password:
   <input type="password" name="password"><br><br>
   <input type="submit" value="Login">

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

Figure 1 : Code Snippet of login.jsp

Activity: Paste screenshot evidence

Step 2

Create **process.jsp** and type the following code to process the submitted login data, validate user input, and redirect the user to dashboard.jsp using **response.sendRedirect()** when the login is successful.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%
   response.setContentType("text/html");

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

   if(username != null && password != null &&
     username.equals("admin") && password.equals("123456")) {

      // Redirect to dashboard
      response.sendRedirect("dashboard.jsp");

   } else {

      out.println("<h3>Invalid Username or Password</h3>");
      out.println("<a href='login.jsp'>Try Again</a>");
   }
%>
```

Figure 2 : Code Snippet of process.jsp

Activity: Paste screenshot evidence

Step 3

Create **dashboard.jsp** that displays a successful login message. This page functions as the destination page after the user is redirected **from process.jsp using response.sendRedirect().**

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
  <title>Dashboard</title>
</head>
<body>

<h2>Welcome to Dashboard</h2>
<p>Login Successful!</p>

</body>
</html>
```

Figure 3 :Code Snippet of dashboard.jsp

Activity: Paste screenshot evidence

EXERCISE 1

Scenario

Rina is a web developer at **EduLearn Online System**, a platform that allows students to register for online courses. Recently, the system received many invalid registrations such as empty usernames, very short passwords, and incomplete email addresses.
To improve the system, Rafiq implements **client-side validation using JavaScript** on the registration form to ensure users enter correct data before submitting it to the server.
The validation rules are:

- Username must not be empty
- Email must contain "@" symbol
- Password must be at least 6 characters

Requirement

Based on the scenario above, create a JSP-based login system that includes HTTP response handling. Create a JSP page named login.jsp to design a user registration form interface. Allow users to enter username, email, and password, and validate the input data using JavaScript before submitting it to the server.

EXERCISE 2

Scenario

Daniel is a junior web developer at SmartShop Online System. He develops a login module using JSP for staff access. When the staff successfully logs in, the system must:

- Set the correct HTTP response content type
- Process login data in process.jsp
- Redirect the user to orderBook.jsp using response.sendRedirect()

If the login fails, the system will display an error message on the same page.

Requirement

Based on the scenario above, create a JSP-based login system that includes HTTP response handling.
a. Create login.jsp to design a login form interface for users.
b. Create process.jsp to process the login data submitted from the form and perform authentication.

c. In process.jsp, set the HTTP response content type using response.setContentType("text/html").

d. Use response.sendRedirect() to redirect users to ordderBook.jsp when login is successful.

e. Create dashboard.jsp to display a successful login message.

f. Run and test the system using both valid and invalid login inputs.

REFLECTION

_____

_____

_____

_____

CONCLUSION

Students were able to understand data validation and HTTP response handling in JSP-based web applications. Students were also able to perform client-side and server-side validation, process user input, set HTTP response headers, and apply response.sendRedirect() for page navigation. Overall, the lab activity improved students' skills in developing dynamic web applications using JSP and JavaScript.