Laboratory 8

Title of the Laboratory Exercise: Client Side validation using JavaScript

1. Introduction and Purpose of Experiment

Students will learn to implement Client Side validation using JavaScript for all the functional requirements identified.

2. Aim and Objectives

Aim

Design and implement the Business Logic for the given scenario using JAX-RS services.

3. Objectives

4. Experimental Procedure

5. Calculations/Computations/Algorithms

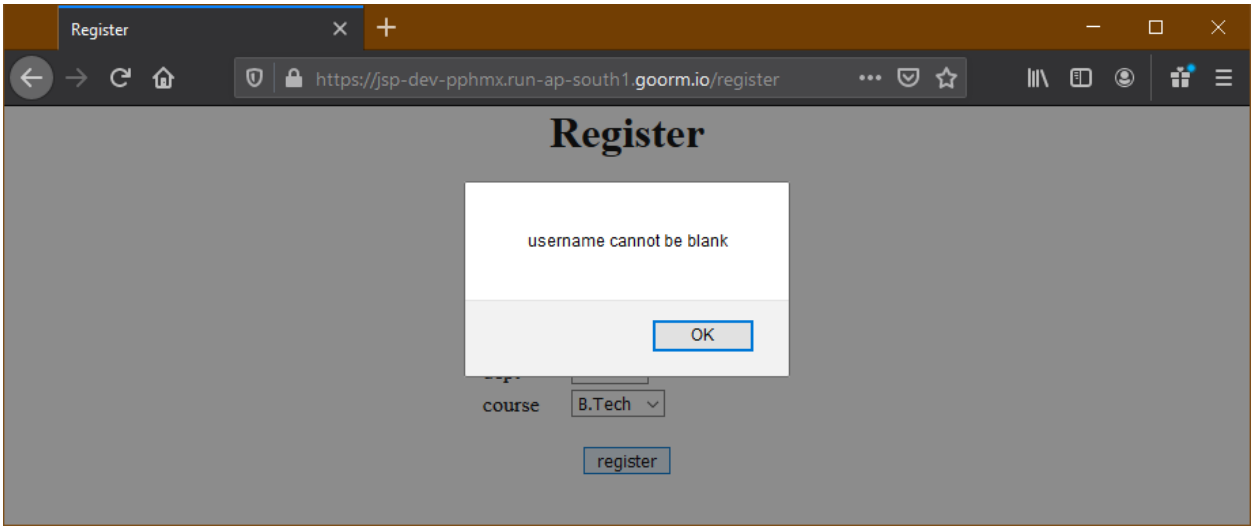
register.jsp

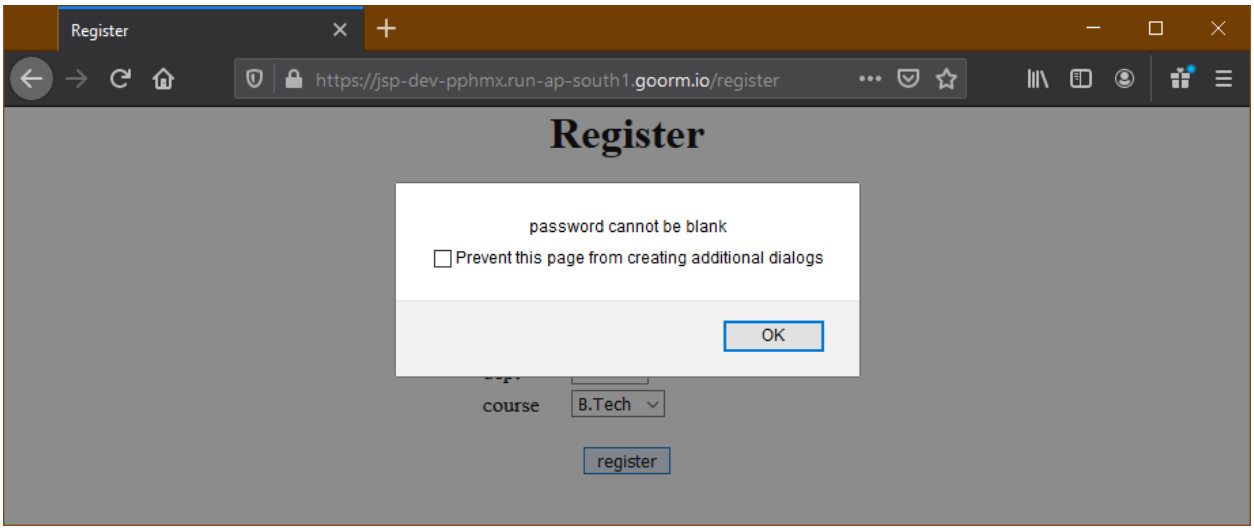
```
<%@ page contentType = "text/html;charset=utf-8" %>
<html>
    <head>
    <title>Register</title>

    <script>

        function validateForm() {
            var username = document.regform.username.value;
            var password = document.regform.password.value;
            var fullname = document.regform.fullname.value;
            var usnno    = document.regform.usnno.value;

            if (username == "") {
                alert("username cannot be blank");
                return false;
            } else if (password == "") {
                alert("password cannot be blank");
                return false;
            } else if (fullname == "") {
                alert("fullname cannot be blank");
                return false;
            } else if (usnno == "") {
```

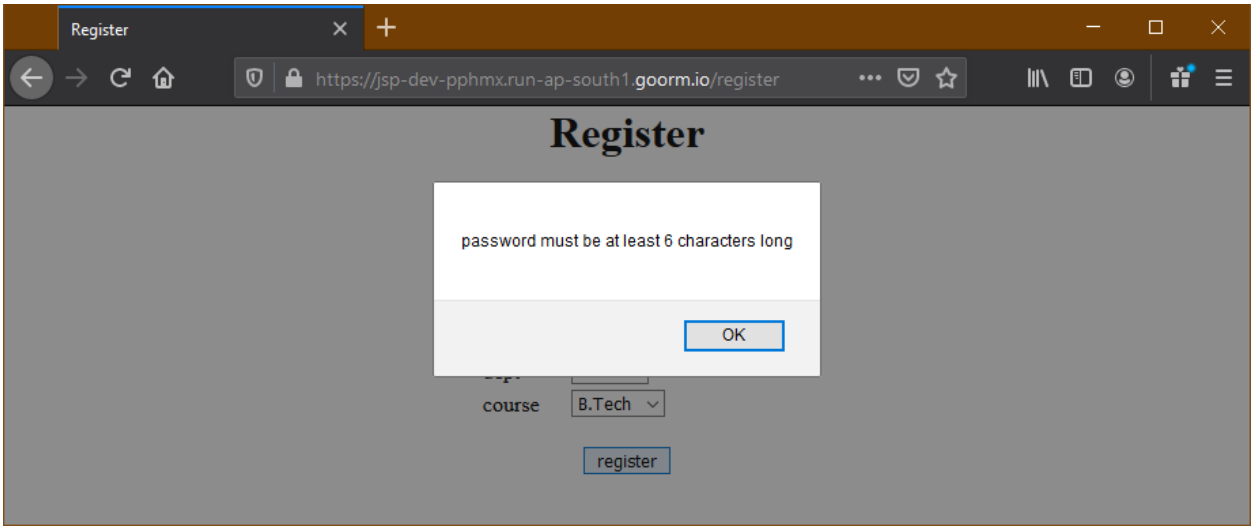
```
                alert("usnno cannot be blank");
                return false;
            } else if (password.length < 6) {
                alert("password must be at least 6 characters long");
                return false;
            }

            return true;
        }

    </script>
    </head>

    <body style="text-align:center; margin:auto">
        <h1>Register</h1>
        <br/>
        <center>
            <form align="center" name="regform" action="register" method="post" onsubmit="return validat
eForm();">
                <table>
                    <tr>
                        <td>username</td>
                        <td><input type="text" name="username" /></td>
                    </tr>
                    <tr>
                        <td>password</td>
                        <td><input type="password" name="password" /></td>
                    </tr>
                    <tr>
                        <td>full name</td>
                        <td><input type="text" name="fullname" /></td>
                    </tr>
                    <tr>
                        <td>usn no</td>
                        <td><input type="text" name="usnno" /></td>
                    </tr>
                    <tr>
                        <td>dept</td>
                        <td>
                            <select name="dept">
                              <option value="CSE">CSE</option>
                              <option value="EEE">EEE</option>
                              <option value="ECE">ECE</option>
                              <option value="CIVIL">CIVIL</option>
                            </select>
                        </td>
                    </tr>
                    <tr>
                        <td>course</td>
                        <td>
                            <select name="course">
```

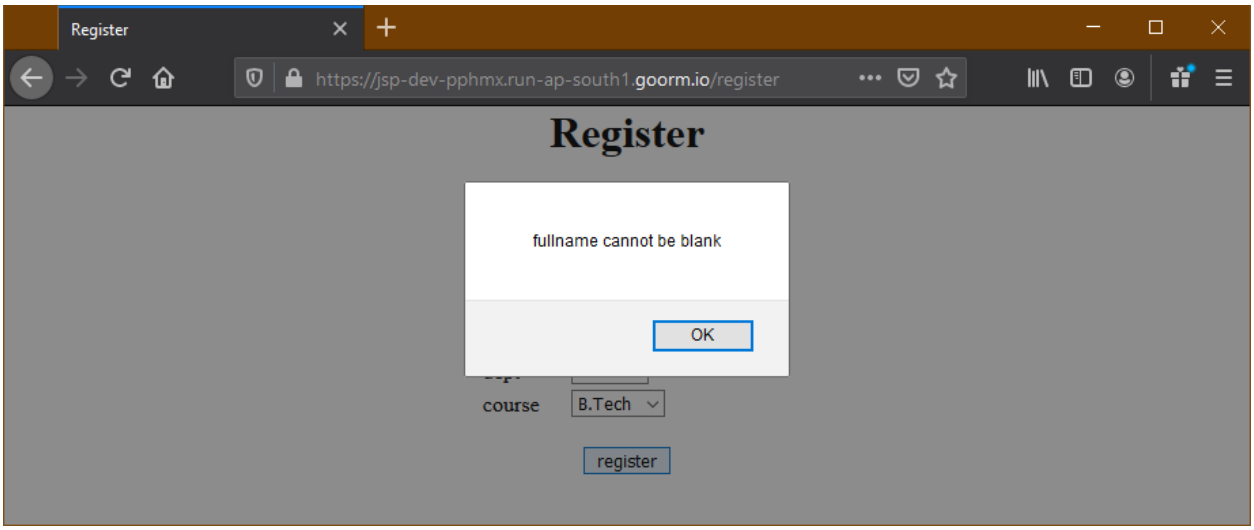
```
                            <option value="B.Tech">B.Tech</option>
                            <option value="M.Tech">M.Tech</option>
                        </select>
                    </td>
                </tr>
            </table>
            <br/>
            <input type="submit" value="register" />
        </form>
    </center>
</body>
</html>
```

6. Presentation of Results









7. Analysis and Discussions

JavaScript is a client-side scripting language, which means that the the script runs on the your viewer or client's browser and not on the server. JavaScript extends the web functionality with a combination of HTML and server side languages. JavaScript validation is one of the common features used with the language.

A basic null validation code block uses an 'if' condition to check whether a specific HTML element returns a value. The following example checks to see if the user entered any value at all:

```
if( document.Form_validate.Name.value == "" )
{
```

```
alert( "Please provide your name!" );
document.Form_validate.Name.focus() ;
return false;
}
```

Data that you can validate using JavaScript includes:

- Required fields
- Username
- Password
- Email address
- Phone number

8. Conclusions

Client-side validation is an initial check and an important feature of good user experience; by catching invalid data on the client-side, the user can fix it straight away. If it gets to the server and is then rejected, a noticeable delay is caused by a round trip to the server and then back to the client-side to tell the user to fix their data.

Types of Client-Side Validation

- Built-in form validation uses HTML5 form validation features, which we've discussed in many places throughout this module. This validation generally doesn't require much JavaScript. Built-in form validation has better performance than JavaScript, but it is not as customizable as JavaScript validation.
- JavaScript validation is coded using JavaScript. This validation is completely customizable, but you need to create it all (or use a library).

9. Comments

a. Limitations of Experiments

The disadvantage, however, is big: client-side support for scripting languages varies wildly, with some browsers supporting scripts very well, others supporting bits and pieces, and others supporting nothing

at all. Furthermore, wily users can disable your client-side checking in order to feed you bad data - if you rely solely on client-side checking, you are bound to get hacked eventually.

b. Limitations of Results

Most client-side validation is accomplished using the special "onSubmit" event of a form, which allows you to run JavaScript code to handle form validation when your visitor attempts to submit the form. If you return false from your code in onSubmit, web browsers will not proceed with submitting the form, which allows you to prompt visitors to correct any errors before submission.

c. Learning happened

We Learnt how to do Client-Side Validation using JavaScript

d. Recommendations

None

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	