

Experiment No. 05

Title: Implement Javascript validation for Website Forms.

Batch: A1**Roll No.: 16010422013****Experiment No.:5**

Aim: To Implement Javascript validation for Website Forms.

Resources needed: Notepad++, Web Browser

Theory:

JavaScript is a scripting language produced by Netscape for use within HTML Web pages. JavaScript is loosely based on Java and it is built into all the major modern browsers. JavaScript is a lightweight, interpreted programming language, Complementary to and integrated with Java , Complementary to and integrated with HTML , Open and cross-platform and is case sensitive.

Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- Has the user filled in all required fields?
- Has the user entered a valid date?
- Has the user entered text in a numeric field?
- Most often, the purpose of data validation is to ensure correct user input.

Validation can be defined by many different methods, and deployed in many different ways.

- Server side validation is performed by a web server, after input has been sent to the server.
- Client side validation is performed by a web browser, before input is sent to a web server.

For example HTML form validation can be done by JavaScript. If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted:

```
function validateForm() {  
  
    var x = document.forms["myForm"]["fname"].value;  
  
    if (x == "") {  
  
        alert("Name must be filled out");  
  
        return false;  
    }  
}
```

```

    }
}

```

Activity:

Add validations for the Website Forms Such as

- 1) Name should string
- 2) Roll number should a number
- 3) Email id should have @ and . in it
- 4) Telephone number should be a ten digit number.

Students need to add various validations to their form input as per the requirement of the user interface.

Results: (Program printout with output)

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Form Validation</title>
  <style>
    .error {
      color: red;
    }
  </style>
</head>
<body>

  <form id="myForm">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" onblur="validateName()"
required>
    <span id="nameError" class="error"></span><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" onblur="validateEmail()">
    <span id="emailError" class="error"></span><br>

    <label for="age">Age:</label>
    <input type="number" id="age" name="age" onblur="validateAge()">
    <span id="ageError" class="error"></span><br>

    <label for="phone">Phone:</label>
    <input type="tel" id="phone" name="phone" onblur="validatePhone()">
    <span id="phoneError" class="error"></span><br>

    <input type="button" value="Submit" onclick="validateForm()">
  </form>

  <script>

```

```

function validateName() {
    var name = document.getElementById("name").value;
    var nameRegex = /^[A-Za-z]+$/;

    if (!name || !nameRegex.test(name)) {
        document.getElementById("nameError").innerText = "Please enter a
valid name with only alphabets.";
    } else {
        document.getElementById("nameError").innerText = "";
    }
}

function validateEmail() {
    var email = document.getElementById("email").value;
    var emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

    if (email && !emailRegex.test(email)) {
        document.getElementById("emailError").innerText = "Please enter
a valid email address.";
    } else {
        document.getElementById("emailError").innerText = "";
    }
}

function validateAge() {
    var age = document.getElementById("age").value;

    if (isNaN(age) || age < 2 || age > 200) {
        document.getElementById("ageError").innerText = "Please enter a
valid age between 2 and 200.";
    } else {
        document.getElementById("ageError").innerText = "";
    }
}

function validatePhone() {
    var phone = document.getElementById("phone").value;

    if (isNaN(phone) || phone.length !== 10) {
        document.getElementById("phoneError").innerText = "Please enter
a valid 10-digit phone number.";
    } else {
        document.getElementById("phoneError").innerText = "";
    }
}

function validateForm() {
    validateName();
    validateEmail();
    validateAge();
    validatePhone();
}
</script>

</body>
</html>

```

TEST CASES OF OUTPUT –

← → ↻ ⓘ File | C:/Users/HP/Desktop/content.2016/java.html

Name:

Email: Please enter a valid email address.

Age:

Phone:

Name: Please enter a valid name with only alphabets.

Email:

Age:

Phone:

Name:

Email:

Age: Please enter a valid age between 2 and 200.

Phone:

Name:

Email:

Age:

Phone: Please enter a valid 10-digit phone number.

Questions:

Q1) Why to carry out Validation at client side using scripting language?

Client-side validation using scripting languages, such as JavaScript, offers several advantages. First, it enhances user experience by providing immediate feedback, reducing the need for round trips to the server. Second, it helps in validating user input without requiring a server request, improving responsiveness. Third, it reduces server load by handling validation on the client side, leading to faster processing. Fourth, it can enhance security by preventing unnecessary data submission, detecting errors before reaching the server. Lastly, client-side validation contributes to a more seamless and interactive web application, creating a smoother overall user interaction.

Q2) What is the difference between client side validation and server side validation?

Client-side validation occurs on the user's device, typically using scripting languages like JavaScript, providing immediate feedback to users without server involvement. It enhances user experience by reducing the need for server requests, improving responsiveness. However, it is essential to note that client-side

validation can be bypassed, so server-side validation is crucial for security and data integrity. Server-side validation takes place on the server after the user submits data, ensuring a thorough and secure examination of input. While it may result in a slightly slower user experience due to server requests, server-side validation is crucial for maintaining the integrity of data and preventing malicious attacks. A combination of both client-side and server-side validation is often employed in web applications to provide a seamless and secure user experience.

Outcomes:

APPLY JAVASCRIPT AND JSON FOR WEB DEVELOPMENT.

Conclusion: (Conclusion to be based on the outcomes achieved)

Validated HTML forms using JAVASCRIPT.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

- “Web technologies: Black Book”, Dreamtech Publications
 - <http://www.w3schools.com>
-