**Name - Gaurang A Raorane Roll No - 49**

**Class - D15A Batch - C**

**EXPERIMENT NO - 6**

**Aim:- Explain DOM and different properties and methods, Event Handling and different types of events, Regular expression and validation using Java script.**

**Theory:-**

**1. Document Object Model (DOM):** The Document Object Model represents the structure of an HTML or XML document as a tree-like structure. It provides a way for programs and scripts to access and manipulate the content and structure of web documents. Key concepts include:

* Nodes: The basic building blocks of the DOM, representing elements, attributes, and text content.
* Hierarchy: The DOM is organized hierarchically, with nodes nested inside other nodes.
* Properties and Methods: DOM nodes have properties (such as textContent or innerHTML) and methods (such as getElementById or createElement) that enable manipulation.
* Traversal: You can navigate the DOM tree using methods like parentNode, childNodes, and nextSibling.

**2. Event Handling:** Event handling in JavaScript allows for user interaction with web pages. It involves responding to events like clicks, key presses, and mouse movements. Key points include:

* Event Listeners: Event listeners are used to detect and respond to specific events. They are attached to DOM elements and execute code when the event occurs.
* Event Object: Event objects contain information about the event, such as the target element and event type.
* Event Propagation: Events can propagate through the DOM tree, following a capturing and bubbling phase. Understanding event propagation is crucial for event handling.

**3. Types of Events:** JavaScript provides various types of events. Common categories include:

* Mouse Events: Events triggered by mouse actions, such as click, mouseover, and mouseout.
* Keyboard Events: Events related to keyboard input, like keydown, keyup, and keypress.
* Form Events: Events for form elements, including submit, reset, and change.
* Document and Window Events: Events like load, unload, and resize that pertain to the document or browser window.

**4. Regular Expressions and Validation:** Regular expressions (regex) are patterns used to match and validate strings. They are essential for data validation and manipulation. Key points include:

* Pattern Matching: Regular expressions define patterns for searching and manipulating text.
* Modifiers: Modifiers like i (case-insensitive) and g (global) control how regular expressions are applied.
* Methods: JavaScript provides methods like test, match, and replace to work with regular expressions.
* Common Use Cases: Regular expressions can be used for tasks such as email validation, password strength checks, and data formatting.

**Input:-**

<!DOCTYPE html>

<html>

<head>

<title>Grade Calculation</title>

</head>

<body>

<h1>Grade Calculation</h1>

<form id="gradeCalculator">

<label for="subject1">Subject 1:</label>

<input type="number" id="subject1" required>

<br>

<label for="subject2">Subject 2:</label>

<input type="number" id="subject2" required>

<br>

<label for="subject3">Subject 3:</label>

<input type="number" id="subject3" required>

<br>

<label for="subject4">Subject 4:</label>

<input type="number" id="subject4" required>

<br>

<label for="subject5">Subject 5:</label>

<input type="number" id="subject5" required>

<br>

<button type="button" onclick="calculateGrades()">Calculate Grades</button>

</form>

<p id="result">Average Grade: </p>

<script>

function calculateGrades() {

const subject1 = parseFloat(document.getElementById("subject1").value);

const subject2 = parseFloat(document.getElementById("subject2").value);

const subject3 = parseFloat(document.getElementById("subject3").value);

const subject4 = parseFloat(document.getElementById("subject4").value);

const subject5 = parseFloat(document.getElementById("subject5").value);

const totalSubjects = 5; // You can change this if you have a different number of subjects

const totalScore = subject1 + subject2 + subject3 + subject4 + subject5;

const average = totalScore / totalSubjects;

let grade = "";

if (average >= 90) {

grade = "A+";

} else if (average >= 80) {

grade = "A";

} else if (average >= 70) {

grade = "B";

} else if (average >= 60) {

grade = "C";

} else if (average >= 50) {

grade = "D";

} else {

grade = "F";

}

document.getElementById("result").textContent = "Average Grade: " + grade;

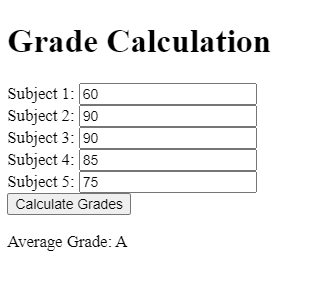
}

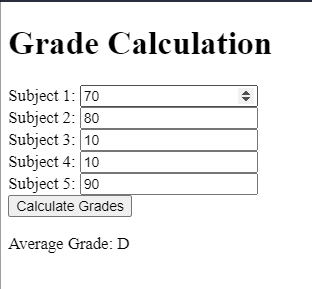
</script>

</body>

</html>

**Output:-**





**Input:-**

<!DOCTYPE html>

<html>

<head>

<title>Registration Form</title>

</head>

<body>

<h1>Registration Form</h1>

<form id="registrationForm" onsubmit="return validateForm()">

<label for="name">Full Name:</label>

<input type="text" id="name" name="name" required>

<span id="nameError" class="error"></span>

<br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

<span id="emailError" class="error"></span>

<br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<span id="passwordError" class="error"></span>

<br>

<button type="submit">Register</button>

</form>

<script>

function validateForm() {

const name = document.getElementById("name").value;

const email = document.getElementById("email").value;

const password = document.getElementById("password").value;

const nameRegex = /^[A-Za-z\s]+$/;

const emailRegex = /^[A-Za-z0-9.\_-]+@[A-Za-z]+\.[A-Za-z]{2,}$/;

const passwordRegex = /^(?=.\*[A-Za-z])(?=.\*\d)[A-Za-z\d]{8,}$/;

const nameError = document.getElementById("nameError");

const emailError = document.getElementById("emailError");

const passwordError = document.getElementById("passwordError");

let valid = true;

if (!name.match(nameRegex)) {

nameError.textContent = "Invalid name";

valid = false;

} else {

nameError.textContent = "";

}

if (!email.match(emailRegex)) {

emailError.textContent = "Invalid email";

valid = false;

} else {

emailError.textContent = "";

}

if (!password.match(passwordRegex)) {

passwordError.textContent = "Invalid password";

valid = false;

} else {

passwordError.textContent = "";

}

return valid;

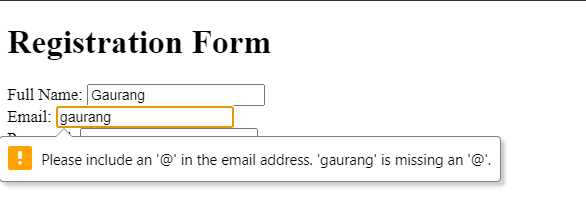
}

</script>

</body>

</html>

**Output:-**

****