



CET-223

Web Technologies

Experiment # 09

Experiment Title

Building a Functional Calculator with HTML, CSS, and JavaScript

Assessment of CLO(s): 04

Performed on _____

Student Name:			
Roll No.		Group	
Semester		Session	

Total (Max)	Performance (03)	Viva (03)	File (04)	Total (10)
Marks Obtained				
Remarks (if any)				

Experiment evaluated by

Instructor's Name	Engr. Bilal Iqbal		
Date		Signature	

OBJECTIVE:

The objective of this lab is to provide students with hands-on experience in building a functional calculator using web technologies. You will learn to structure a web page using HTML, style it effectively with CSS, and add interactivity through JavaScript.

HTML Structure:

HTML (HyperText Markup Language) is used to define the structure and content of web pages. In this step, we create the layout for our calculator using basic HTML tags like div, input, and button.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple Calculator</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="calculator">
    <input type="text" id="result" disabled>
    <div class="calculator-buttons">
      <button class="number" onclick="appendValue('7')">7</button>
      <button class="number" onclick="appendValue('8')">8</button>
      <button class="number" onclick="appendValue('9')">9</button>
      <button class="operation" onclick="appendValue('/')">/</button>
      <button class="number" onclick="appendValue('4')">4</button>
      <button class="number" onclick="appendValue('5')">5</button>
      <button class="number" onclick="appendValue('6')">6</button>
      <button class="operation" onclick="appendValue('*')">*</button>
      <button class="number" onclick="appendValue('1')">1</button>
      <button class="number" onclick="appendValue('2')">2</button>
      <button class="number" onclick="appendValue('3')">3</button>
      <button class="operation" onclick="appendValue('-')">-</button>
      <button class="number" onclick="appendValue('0')">0</button>
      <button class="number" onclick="appendValue('.')">.</button>
      <button class="clear" onclick="clearResult()">C</button>
      <button class="operation" onclick="appendValue('+')">+</button>
      <button class="equal" onclick="calculateResult()">=</button>
    </div>
  </div>
<script src="script.js"></script>
</body>
</html>
```

- The `<input type="text">` is used to display the result of the calculations. It is disabled so that users cannot type in it manually.
- The buttons are divided into numbers, operations (+, -, *, /), and special functions like C for clearing the input and = for calculating the result.
- The `onclick` attribute calls JavaScript functions to perform actions when the user clicks any button.

CSS Styling:

CSS (Cascading Style Sheets) is used to style the HTML elements, such as controlling the layout, colors, fonts, and spacing.

```
body {
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  background-color: #f3f3f3;
}

.calculator {
  background: #fff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  width: 320px;
}

.calculator input[type="text"] {
  width: 100%;
  height: 50px;
  margin-bottom: 15px;
  text-align: right;
  padding: 10px;
  font-size: 1.5em;
  border: 1px solid #ddd;
  border-radius: 5px;
  box-sizing: border-box;
}

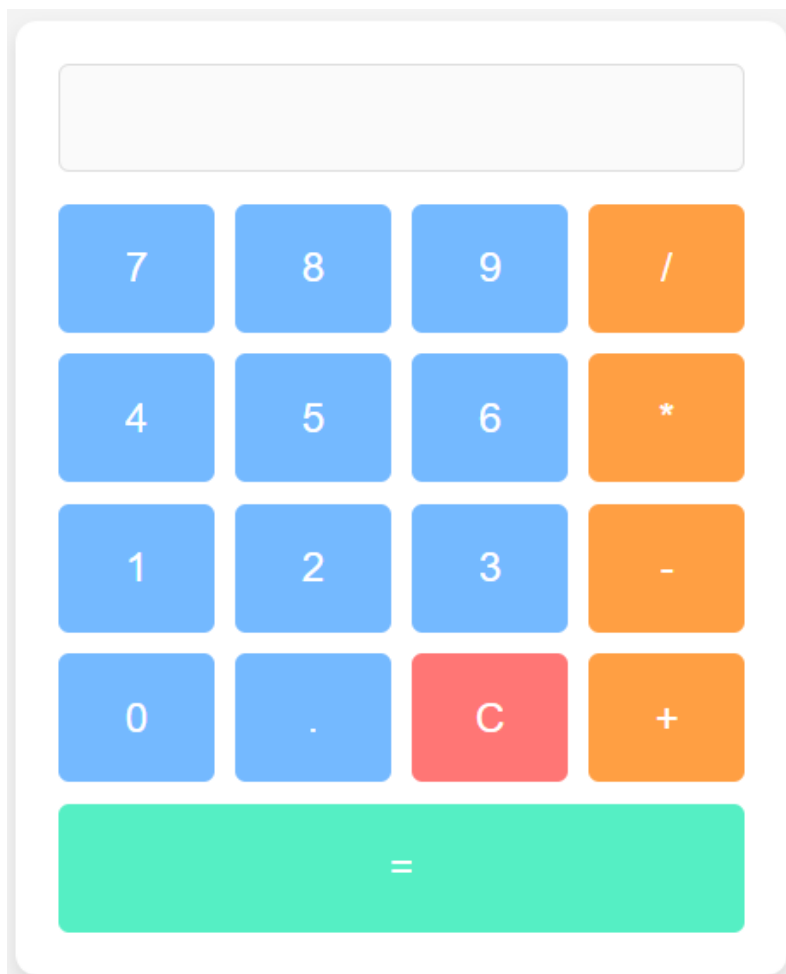
.calculator-buttons {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 10px;
}

.calculator button {
  height: 60px;
  font-size: 1.2em;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

.calculator button.operation {
  background-color: #ff9f43;
  color: white;
}
```

```
.calculator button.number {  
    background-color: #74b9ff;  
    color: white;  
}  
  
.calculator button.clear {  
    background-color: #ff7675;  
    color: white;  
}  
  
.calculator button.equal {  
    background-color: #55efc4;  
    color: white;  
    grid-column: span 4;  
}
```

- The body is styled to center the calculator in the middle of the screen with a light gray background.
- The .calculator class applies a white background, padding, rounded corners, and a shadow effect to make the calculator visually appealing.
- Buttons are styled to have consistent height, font size, and color coding (operations, numbers, clear, and equal).



JavaScript Functionality:

JavaScript is used to make the calculator functional by adding event listeners to buttons, performing arithmetic operations, and handling errors.

```
function appendValue(value) {
    document.getElementById('result').value += value;
}

function clearResult() {
    document.getElementById('result').value = '';
}

function calculateResult() {
    try {
        const result = eval(document.getElementById('result').value);
        document.getElementById('result').value = result;
    } catch {
        document.getElementById('result').value = 'Error';
    }
}
```

- **appendValue(value):** Adds the clicked button's value to the current input field (#result).
- **clearResult():** Clears the input field when the 'C' button is pressed.
- **calculateResult():** Uses JavaScript's eval() function to evaluate the expression entered in the input field and display the result. If the expression is invalid, it shows "Error" in the input field.

After completing all the steps, calculator should be fully functional. Users can click buttons for numbers and operations, and the calculator will display the result or show an error if the input is invalid.

Lab Tasks:

1. Add animations to buttons when they are clicked.
2. Implement keyboard support to allow users to type numbers and operations directly.
3. Modify the design to make it more responsive for different screen sizes (mobile and desktop).