Abstract:

This project presents a simple web-based calculator designed using HTML, CSS, and JavaScript. The calculator provides basic arithmetic functionality, including addition, subtraction, multiplication, and division. The user interface is created with HTML for structure and styled with CSS for a clean and intuitive layout. JavaScript is used to implement the calculator's logic, enabling interactive features such as real-time calculations and input validation. This project demonstrates the integration of front-end web technologies to create a functional and user-friendly application. It serves as a foundational exercise for beginners in web development and highlights essential programming concepts such as event handling, DOM manipulation, and data processing in a browser environment.

Introduction:

A simple calculator is a fundamental web application that allows users to perform basic arithmetic operations such as addition, subtraction, multiplication, and division. This project demonstrates how to build a calculator using core web technologies—HTML for the structure, CSS for styling, and JavaScript for the logic and interactivity.

The main goal of this calculator is to provide an easy-to-use interface where users can input numbers and operators to perform quick calculations directly in their web browser. By developing this project, beginners can gain practical experience in designing user interfaces, handling user inputs, and implementing logical operations, all of which are essential skills in front-end web development.

This calculator not only helps reinforce essential programming concepts but also lays the groundwork for more advanced web applications in the future.

Objective:

The objective of this project is to design and develop a simple web-based calculator using HTML, CSS, and JavaScript that can perform basic arithmetic operations including addition, subtraction, multiplication, and division. The calculator aims to provide a user-friendly interface that allows users to input numbers and mathematical operators to perform quick and accurate calculations. This project serves as a foundational exercise in web development, helping learners understand the structure of web pages, styling elements, and implementing interactivity through scripting.

Methodology:

The development of the simple calculator was carried out using a structured approach involving three main web technologies: HTML, CSS, and JavaScript.

1. **Planning and Design**:
   The project began with planning the layout of the calculator interface. The design was kept simple and minimal to focus on functionality. A standard calculator layout was chosen with number buttons (0–9), basic operators (+, −, ×, ÷), a clear (C) button, and an equals (=) button.
2. **HTML Structure**:
   HTML was used to create the basic structure of the calculator. This included a display screen for showing input and results, and a grid of buttons for numbers and operations. Semantic elements and proper organization ensured readability and ease of styling.
3. **CSS Styling**:
   CSS was used to enhance the visual appearance of the calculator. Styling included layout alignment, button design, hover effects, and overall responsiveness to make the calculator more user-friendly and visually appealing.

4. **JavaScript Functionality**:
   JavaScript was implemented to handle the calculator's logic. Event listeners were added to each button to capture user input. Functions were written to perform arithmetic operations, update the display, and manage edge cases such as division by zero or multiple operator inputs.
5. **Testing and Debugging**:
   The final step involved testing the calculator for accuracy and reliability. Common input patterns were tested, and any bugs or unexpected behaviours were corrected to ensure smooth functionality.

Source code:

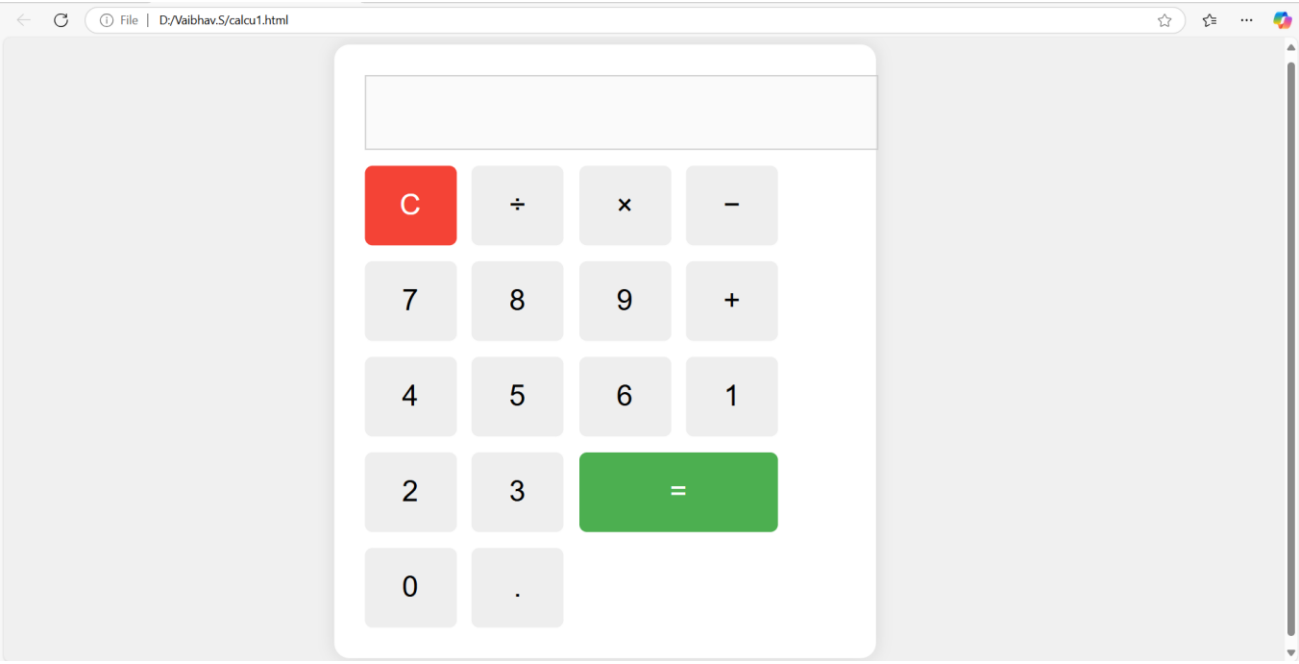
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Simple Calculator</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background: #f0f0f0;
      display: flex;
      height: 100vh;
      align-items: center;
      justify-content: center;
    }

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

    #result {
      width: 100%;
      padding: 10px;
      font-size: 1.5em;
      text-align: right;
      margin-bottom: 10px;
      border: 1px solid #ccc;
    }

    .buttons {
      display: grid;
```

```
    grid-template-columns: repeat(4, 60px);
    gap: 10px;
   }

   button {
    padding: 15px;
    font-size: 1.2em;
    cursor: pointer;
    border: none;
    background: #eee;
    border-radius: 5px;
    transition: background 0.2s;
   }

   button:hover {
    background: #ddd;
   }

   .equal {
    grid-column: span 2;
    background-color: #4CAF50;
    color: white;
   }

   .clear {
    background-color: #f44336;
    color: white;
   }
  </style>
</head>
<body>

<div class="calculator">
  <input type="text" id="result" disabled>
  <div class="buttons">
    <button onclick="clearResult()" class="clear">C</button>
    <button onclick="appendValue('/')">÷</button>
```

```
    <button onclick="appendValue('*')">×</button>
    <button onclick="appendValue('-')">−</button>

    <button onclick="appendValue('7')">7</button>
    <button onclick="appendValue('8')">8</button>
    <button onclick="appendValue('9')">9</button>
    <button onclick="appendValue('+')">+</button>

    <button onclick="appendValue('4')">4</button>
    <button onclick="appendValue('5')">5</button>
    <button onclick="appendValue('6')">6</button>

    <button onclick="appendValue('1')">1</button>
    <button onclick="appendValue('2')">2</button>
    <button onclick="appendValue('3')">3</button>
    <button onclick="calculateResult()" class="equal">=</button>

    <button onclick="appendValue('0')">0</button>
    <button onclick="appendValue('.')">.</button>
  </div>
</div>

<script>
  function appendValue(value) {
    document.getElementById('result').value += value;
  }

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

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

```
    }
  }
</script>

</body>
</html>
```

Output:



Conclusion:

This simple calculator demonstrates the basic functionality of a web-based calculator using HTML, CSS, and JavaScript.

It allows users to perform basic arithmetic operations such as addition, subtraction, multiplication, and division.

This project is a great starting point for beginners to understand how web technologies work together to create interactive applications.

Reference:

```
<section style="text-align: center; margin-top: 40px;">
  <h2>References</h2>
  <ul style="list-style-type: none; padding: 0;">
    <li>
      <a href="https://developer.mozilla.org/en-
US/docs/Web/HTML" target="_blank">
        MDN Web Docs – HTML
      </a>
    </li>
    <li>
      <a href="https://developer.mozilla.org/en-
US/docs/Web/CSS" target="_blank">
        MDN Web Docs – CSS
      </a>
    </li>
    <li>
      <a href="https://developer.mozilla.org/en-
US/docs/Web/JavaScript" target="_blank">
        MDN Web Docs – JavaScript
      </a>
    </li>
    <li>
```

```
      <a
href="https://www.w3schools.com/js/js_calculator.asp"
target="_blank">
        W3Schools – JavaScript Calculator Tutorial
      </a>
    </li>
  </ul>
</section>
```

JSPM University Pune
Faculty of Science and Technology
School of Basic and Applied Sciences
Bachelor of Computer Applications

FULL STACK DEVELOPMENT PROJECT

SIMPLE CALCULATOR

NAME: SHUBHAM KAWADE
ROLL.NO: 61, DIV: B
PRN: 22356030138