CALCULATOR

Name : Anshika Sangal

Roll No: 2300290100062

Branch : CSE

Year : 2

INTRODUCTION

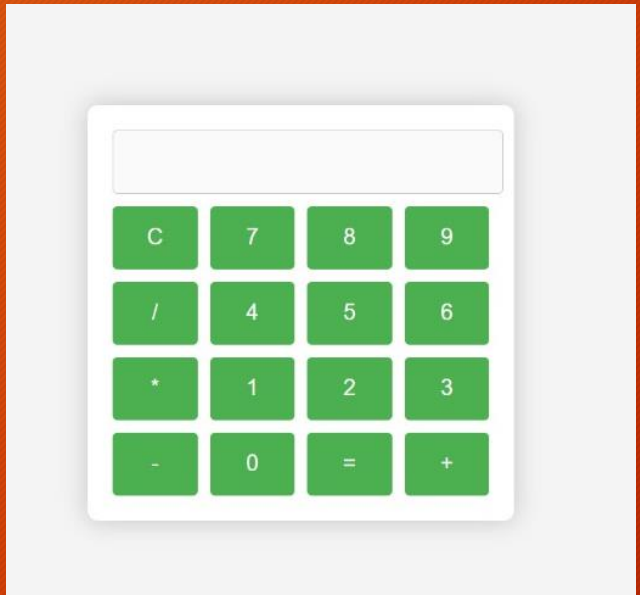
- **Overview**: This project focuses on developing a simple calculator application that performs basic arithmetic operations such as addition, subtraction, multiplication, and division.
- **Purpose**:
  - To demonstrate fundamental programming concepts.
  - To provide a practical tool that enhances user convenience for performing calculations.
- **Technology Stack**:
  - HTML for structure.
  - CSS for styling.
  - JavaScript for functionality.

PROJECT DESCRIPTION

- **Features**:
- **Basic Operations**: Perform addition (+), subtraction (-), multiplication (×), and division (÷).
- **User Input**: Accepts user input through buttons.
- **Clear Functionality**: Reset the current calculation.
- **Display Area**: Shows the current input and the result.
- **Target Audience**:
- Students, professionals, or anyone needing quick arithmetic calculations.

ARCHITECTURE AND DESGIN

- Client-Side Architecture:Entire application runs in the browser, no server-side processing needed.
- Responsive Design:The calculator adapts to different screen sizes, ensuring usability on desktops, tablets, and smartphones.
- Wireframe



DEVELOPMENT PROCESS

Tools Used:

- **Text Editor**: Visual Studio Code for coding.
- **Version Control**: Git for tracking changes and managing versions.
- **Browser Developer Tools**: For debugging and testing during development.

IMPLEMENTATION

- HTML Structure:Utilized semantic elements to create a clean and organized layout.
- Example Elements:
- <input> for the display.
- <button> for calculator buttons.
- CSS Styling:Applied styles to create a visually appealing interface.Used color contrasts for buttons and display area to enhance readability.
- JavaScript Functionality:Logic for performing calculations based on user input.

HTML

```html
<!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="display" disabled>
        <div class="buttons">
            <button onclick="clearDisplay()">C</button>
            <button onclick="appendToDisplay('7')">7</button>
            <button onclick="appendToDisplay('8')">8</button>
            <button onclick="appendToDisplay('9')">9</button>
            <button onclick="appendToDisplay('/')">/</button>
            <button onclick="appendToDisplay('4')">4</button>
            <button onclick="appendToDisplay('5')">5</button>
            <button onclick="appendToDisplay('6')">6</button>
            <button onclick="appendToDisplay('*')">*</button>
            <button onclick="appendToDisplay('1')">1</button>
            <button onclick="appendToDisplay('2')">2</button>
            <button onclick="appendToDisplay('3')">3</button>
            <button onclick="appendToDisplay('-')">-</button>
            <button onclick="appendToDisplay('0')">0</button>
            <button onclick="calculateResult()">=</button>
            <button onclick="appendToDisplay('+')">+</button>
        </div>
    </div>
```

CSS

```css
body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f4f4f4;
    font-family: Arial, sans-serif;
}

.calculator {
    background-color: white;
    border-radius: 8px;
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);
    padding: 20px;
}

#display {
    width: 100%;
    height: 40px;
    font-size: 24px;
    text-align: right;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 4px;
    padding: 5px;
}
```

```
27
28  .buttons {
29      display: grid;
30      grid-template-columns: repeat(4, 1fr);
31      gap: 10px;
32  }
33
34  button {
35      padding: 15px;
36      font-size: 18px;
37      border: none;
38      border-radius: 4px;
39      background-color: #4CAF50;
40      color: white;
41      cursor: pointer;
42      transition: background-color 0.3s;
43  }
44
45  button:hover {
46      background-color: #45a049;
47  }
48
```

Java Script

```javascript
function appendToDisplay(value) {
    document.getElementById("display").value += value;
}

function clearDisplay() {
    document.getElementById("display").value = "";
}

function calculateResult() {
    const display = document.getElementById("display");
    try {
        display.value = eval(display.value);
    } catch (error) {
        display.value = "Error";
    }
}
```

TESTING

•**Cross-Browser Testing**:
•Ensured functionality across major browsers: Chrome, Firefox, and Safari.
•**Device Testing**:
•Conducted tests on multiple devices (desktop, tablet, and mobile) for responsive design.
•**Testing Methods**:
•**Functional Testing**: Confirmed all calculator functions (add, subtract, multiply, divide) performed correctly.
•**Usability Testing**: Gathered feedback from users on the layout and ease of use; made adjustments based on feedback.
•**Visual Testing**: Ensured consistent appearance of the calculator across different screen sizes and resolutions.
•**Outcomes**:
•Resolved issues such as button responsiveness and display accuracy based on user testing.

CONCLUSION

•**Project Success**:
Developed a fully functional calculator application that meets the initial requirements.
•**Learning Outcomes**:
Enhanced understanding of web development technologies: HTML, CSS, and JavaScript.
Gained experience in designing user interfaces and ensuring usability.
•**Future Enhancements**:
Plan to incorporate more advanced mathematical functions (e.g., square root, percentages).
Consider improving the design with animations and transitions for better user experience.
Explore adding features like history tracking of calculations.