INTRODUCTION

NAME : TASHI VERMA

•University Roll No: 2300290120260

•Branch: Computer Science

•Year: 2

•Section: D

Building a Calculator in Web Development

This is a basic calculator application built using HTML, CSS, and JavaScript. The calculator performs basic arithmetic operations and has a user-friendly interface.

Introduction to HTML, CSS, and JavaScript

HTML (HyperText Markup Language)

The foundation of a web page. HTML defines the structure and content of your calculator.

CSS (Cascading Style Sheets)

The visual style of your calculator. CSS controls the layout, colors, fonts, and other design elements.

JavaScript

The programming language that adds interactivity. JavaScript handles user input, performs calculations, and updates the display.

```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="calculator.css">
</head>
<body>
    <div class="calculator">
        <input type="text" class="display" id="display" disabled>
        <div class="buttons">
            <button onclick="clearDisplay()">C</button>
            <button onclick="deleteLast()">DEL</button>
            <button onclick="appendToDisplay('/')">/</button>
            <button onclick="appendToDisplay('')"></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="calculateResult()">=</button>
            <button onclick="appendToDisplay('0')" class="zero">0</button>
            <button onclick="appendToDisplay('.')">.</button>
        </div>
    </div>
    <script src="calculator.js"></script>
</body>
</html>
```

Basic HTML Structure for the Calculator

1 **Calculator Container**

A main div element to hold all calculator components.

2 **Display Area**

A div for displaying the input and results.

3 **Number Buttons**

Button elements for each number 0-9.

4 **Operator Buttons**

Button elements for addition, subtraction, multiplication, and division.

Styling the Calculator with CSS

Color Palette

Choose colors that are visually appealing and contrast well.

Font Selection

Pick fonts that are easy to read and match the calculator's style.

Layout and Spacing

Arrange elements to create a user-friendly layout.

Button Styles

Customize button appearance with colors, borders, and hover effects.

```css
* {
    box-sizing: border-box;
    margin: 0;
    padding: 0;
    font-family: Arial;
}

body {
    display: flex;
    align-items: center;
    justify-content: center;
    height: 100vh;
    background-color: #a5f7d2;
}

.calculator {
    width: 350px;
    background-color: #222;
    border-radius: 8px;
    padding: 20px;
}
```

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

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

function deleteLast() {
    const display = document.getElementById('display');
    display.value = display.value.slice(0, -1);
}

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

Implementing Functionality with JavaScript

1

Event Listeners

Attach event listeners to buttons to detect clicks.

2

Input Handling

Store user input and manipulate it.

3
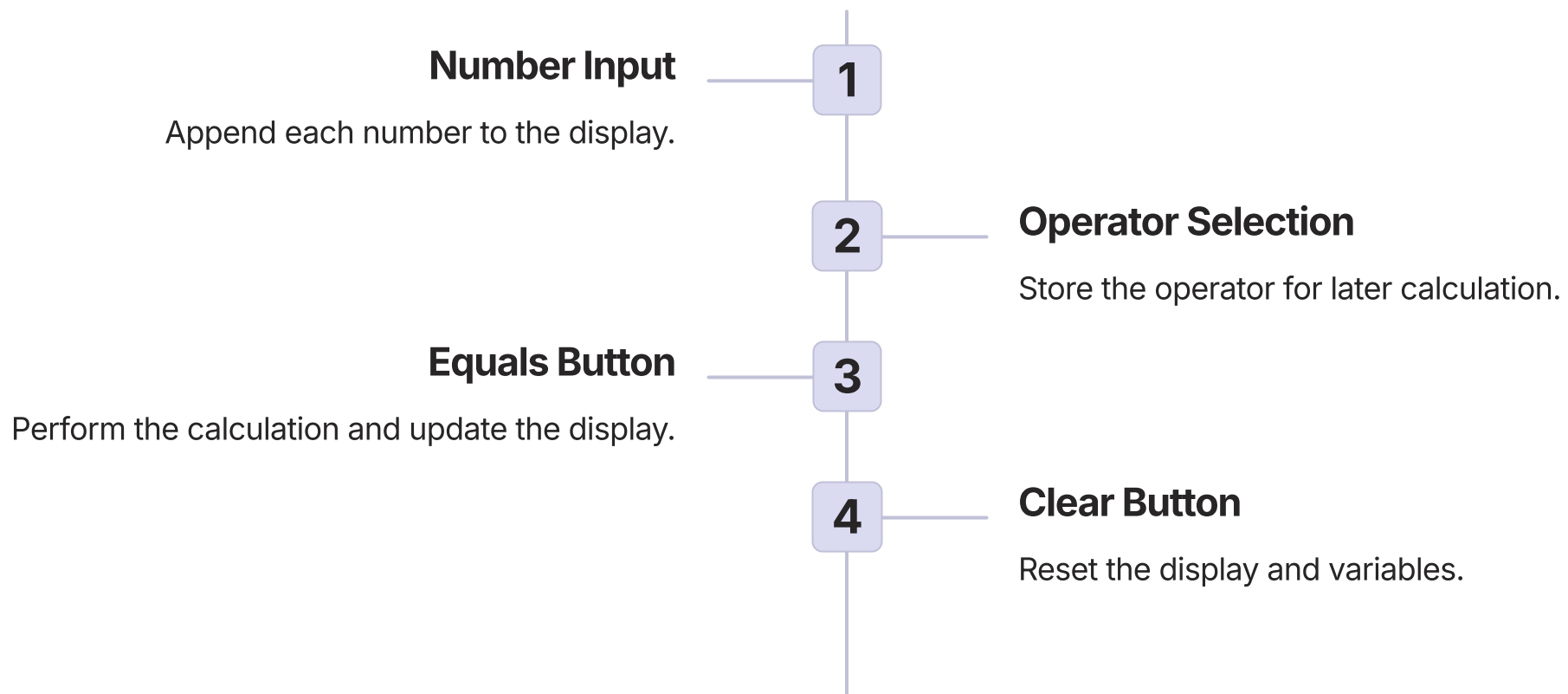
Calculations

Perform mathematical operations based on user input.

4

Display Updates

Update the display with results or error messages.

3346239

Handling User Interactions

Number Input — 1

Append each number to the display.

2 — **Operator Selection**

Store the operator for later calculation.

Equals Button — 3

Perform the calculation and update the display.

4 — **Clear Button**

Reset the display and variables.

Conclusion and Next Steps

Building a calculator is a great way to learn the fundamentals of web development. It helped to learn various concepts of JACASRIPT , HTML and CSS.