Calculator

#html code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Styled Calculator</title>

<link rel="stylesheet" href="cal.css">

</head>

<body>

<div class="calculator">

<input type="text" class="calculator-screen" value="0" disabled />

<div class="calculator-keys">

<button type="button" class="clear">AC</button>

<button type="button" class="operator">±</button>

<button type="button" class="operator">%</button>

<button type="button" class="operator" value="/">÷</button>

<button type="button" value="7">7</button>

<button type="button" value="8">8</button>

<button type="button" value="9">9</button>

<button type="button" class="operator" value="\*">×</button>

<button type="button" value="4">4</button>

<button type="button" value="5">5</button>

<button type="button" value="6">6</button>

<button type="button" class="operator" value="-">−</button>

<button type="button" value="1">1</button>

<button type="button" value="2">2</button>

<button type="button" value="3">3</button>

<button type="button" class="operator" value="+">+</button>

<button type="button" value="0" class="zero">0</button>

<button type="button" value=".">.</button>

<button type="button" class="equal-sign" value="=">=</button>

</div>

</div>

<script src="cal.js"></script>

</body>

</html>

#css code

/\* General page styles \*/

body {

font-family: 'Arial', sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-color: #f0f0f0;

}

/\* Calculator container \*/

.calculator {

width: 320px;

height: auto;

border-radius: 10px;

background-color: #e0e0e0;

box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);

padding: 20px;

}

/\* Display area \*/

.calculator-screen {

width: 100%;

height: 80px;

border: none;

background-color: #3a3a3a; /\* Dark bluish-grey \*/

color: #fff;

text-align: right;

font-size: 2.5rem;

padding: 10px 15px;

margin-bottom: 15px;

border-radius: 10px;

box-shadow: inset 0 0 5px #00000055;

}

/\* Button grid layout \*/

.calculator-keys {

display: grid;

grid-template-columns: repeat(4, 1fr);

grid-gap: 10px;

}

/\* Common button styles \*/

button {

height: 60px;

font-size: 1.5rem;

border: none;

border-radius: 10px;

box-shadow: 0px 3px 5px rgba(0, 0, 0, 0.1);

cursor: pointer;

transition: all 0.3s ease;

}

/\* Operator buttons \*/

button.operator {

background-color: #e6a81c; /\* Orange-yellow \*/

color: white;

}

button.operator:hover {

background-color: #d59718;

}

/\* Number buttons \*/

button:not(.operator):not(.clear):not(.equal-sign) {

background-color: #b5b5b5; /\* Light grey \*/

color: black;

}

button:hover {

background-color: #a3a3a3;

}

/\* Clear button \*/

button.clear {

background-color: #d2d2d2; /\* Light grey \*/

color: black;

}

button.clear:hover {

background-color: #c0c0c0;

}

/\* Equal button \*/

button.equal-sign {

background-color: #e6a81c; /\* Orange-yellow \*/

color: white;

}

button.equal-sign:hover {

background-color: #d59718;

}

/\* Zero button span across two columns \*/

button.zero {

grid-column: span 2;

}

#js code

const calculator = {

displayValue: '0',

firstOperand: null,

waitingForSecondOperand: false,

operator: null,

};

function inputDigit(digit) {

const { displayValue, waitingForSecondOperand } = calculator;

if (waitingForSecondOperand === true) {

calculator.displayValue = digit;

calculator.waitingForSecondOperand = false;

} else {

calculator.displayValue = displayValue === '0' ? digit : displayValue + digit;

}

}

function inputDecimal(dot) {

if (calculator.waitingForSecondOperand === true) {

calculator.displayValue = '0.';

calculator.waitingForSecondOperand = false;

return;

}

if (!calculator.displayValue.includes(dot)) {

calculator.displayValue += dot;

}

}

function handleOperator(nextOperator) {

const { firstOperand, displayValue, operator } = calculator;

const inputValue = parseFloat(displayValue);

if (operator && calculator.waitingForSecondOperand) {

calculator.operator = nextOperator;

return;

}

if (firstOperand == null) {

calculator.firstOperand = inputValue;

} else if (operator) {

const result = performCalculation[operator](firstOperand, inputValue);

calculator.displayValue = `${parseFloat(result.toFixed(7))}`;

calculator.firstOperand = result;

}

calculator.waitingForSecondOperand = true;

calculator.operator = nextOperator;

}

const performCalculation = {

'/': (firstOperand, secondOperand) => firstOperand / secondOperand,

'\*': (firstOperand, secondOperand) => firstOperand \* secondOperand,

'+': (firstOperand, secondOperand) => firstOperand + secondOperand,

'-': (firstOperand, secondOperand) => firstOperand - secondOperand,

'=': (firstOperand, secondOperand) => secondOperand

};

function resetCalculator() {

calculator.displayValue = '0';

calculator.firstOperand = null;

calculator.waitingForSecondOperand = false;

calculator.operator = null;

}

function updateDisplay() {

const display = document.querySelector('.calculator-screen');

display.value = calculator.displayValue;

}

updateDisplay();

const keys = document.querySelector('.calculator-keys');

keys.addEventListener('click', (event) => {

const { target } = event;

if (!target.matches('button')) {

return;

}

if (target.classList.contains('operator')) {

handleOperator(target.value);

updateDisplay();

return;

}

if (target.classList.contains('equal-sign')) {

handleOperator(target.value);

updateDisplay();

return;

}

if (target.classList.contains('clear')) {

resetCalculator();

updateDisplay();

return;

}

inputDigit(target.value);

updateDisplay();

});

