**Html Code:**

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

<h1 align="center">calculator</h1>

<div class="calculator card">

<input type="text" class="calculator-screen z-depth-1"/>

<div class="calculator-keys">

<button type="button" class="operator btn btn-info" value="+" style="background-color:lightblue; border-color:tomato;">+</button>

<button type="button" class="operator btn btn-info" value="-"style="background-color:lightblue;border-color:tomato;">-</button>

<button type="button" class="operator btn btn-info" value="\*"style="background-color:lightblue;border-color:tomato;">&times;</button>

<button type="button" class="operator btn btn-info" value="/"style="background-color:lightblue;border-color:tomato;">&divide;</button>

<button type="button" value="7" class="btn btn-light waves-effect">7</button>

<button type="button" value="8" class="btn btn-light waves-effect">8</button>

<button type="button" value="9" class="btn btn-light waves-effect">9</button>

<button type="button" value="4" class="btn btn-light waves-effect">4</button>

<button type="button" value="5" class="btn btn-light waves-effect">5</button>

<button type="button" value="6" class="btn btn-light waves-effect">6</button>

<button type="button" value="1" class="btn btn-light waves-effect">1</button>

<button type="button" value="2" class="btn btn-light waves-effect">2</button>

<button type="button" value="3" class="btn btn-light waves-effect">3</button>

<button type="button" value="0" class="btn btn-light waves-effect">0</button>

<button type="button" class="decimal function btn btn-secondary" value=".">.</button>

<button type="button" class="all-clear function btn btn-danger btn-sm" value="all-clear" style="background-color:orangered;">AC</button>

<button type="button" class="equal-sign operator btn btn-default" value="=" style="background-color:lightseagreen;">=</button>

</div>

</div>

</div>

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

**CSS :**

html {

font-size: 62.5%;

box-sizing: border-box;

}

\*,

\*::before,

\*::after {

margin: 0;

padding: 0;

box-sizing: inherit;

}

.h1{

height: 100px;

width: 120;

}

.body{

background-color: red;

}

.calculator {

border: 1px solid #ccc;

border-radius: 5px;

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

width: 400px;

}

.calculator-screen {

width: 100%;

height: 80px;

border: none;

background-color: #252525;

color: #fff;

text-align: right;

padding-right: 20px;

padding-left: 10px;

font-size: 4rem;

}

button {

height: 60px;

font-size: 2rem!important;

}

.equal-sign {

height: 98%;

grid-area: 2 / 4 / 6 / 5;

}

.calculator-keys {

display: grid;

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

grid-gap: 20px;

padding: 20px;

}

**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 the `displayValue` does not contain a decimal point

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

// Append the decimal point

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 currentValue = firstOperand || 0;

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

calculator.displayValue = String(result);

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('decimal')) {

inputDecimal(target.value);

updateDisplay();

return;

}

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

resetCalculator();

updateDisplay();

return;

}

inputDigit(target.value);

updateDisplay();

});

**OUTPUT:**

