<!DOCTYPE html>

<html>

<head>

<title>Calculator</title>

<link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

<div class="calculator">

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

<div class="calculator-keys">

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

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

<button class="operator" value="\*">&times;</button>

<button class="operator" value="/">&divide;</button>

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

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

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

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

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

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

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

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

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

<button value="0">0</button>

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

<button class="all-clear" value="all-clear">AC</button>

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

</div>

</div>

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

</body>

</html>

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f0f0f0;

}

.calculator {

border: 1px solid #ccc;

padding: 20px;

border-radius: 5px;

background-color: #fff;

}

.calculator-screen {

width: 100%;

height: 40px;

background-color: #fff;

border: 1px solid #ccc;

margin-bottom: 10px;

text-align: right;

padding: 10px;

font-size: 20px;

}

.calculator-keys button {

width: 23%;

height: 40px;

margin: 1%;

font-size: 18px;

}

.operator {

background-color: #ff9500;

color: #fff;

}

.equal-sign {

background-color: #007AFF;

color: #fff;

height: 84px;

}

.all-clear {

background-color: #ff3b30;

color: #fff;

}

const calculator = {

displayValue: '0',

firstOperand: null,

waitingForSecondOperand: false,

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;

const { value } = target;

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

return;

}

switch (value) {

case '+':

case '-':

case '\*':

case '/':

case '=':

handleOperator(value);

break;

case '.':

inputDecimal(value);

break;

case 'all-clear':

resetCalculator();

break;

default:

if (Number.isInteger(parseFloat(value))) {

inputDigit(value);

}

}

updateDisplay();

});

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 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;

}