TASK-4

CALCULATOR BY USING HTML(html)

```
">5</button>
            <button type="button" value="6">6</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" value="0">0</button><!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Calculator</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="calculator">
        <input type="text" class="calculator-screen" value="" disabled />
        <div class="calculator-keys">
            <button type="button" class="operator" value="+">+</button>
            <button type="button" class="operator" value="-">-</button>
            <button type="button" class="operator" value="*">&times;</button>
            <button type="button" class="operator" value="/">&divide;</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" value="4">4</button>
            <button type="button" value="5
            <button type="button" value=".">.</button>
            <button type="button" class="equal-sign operator"
value="=">=</button>
            <button type="button" class="all-clear" value="all-
clear">AC</button>
        </div>
    </div>
    <script src="scripts.js"></script>
</body>
</html>
```

```css
body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f4f4f4;
}

.calculator {
    border-radius: 10px;
    box-shadow: 0px 0px 20px 0px rgba(0, 0, 0, 0.2);
    padding: 20px;
    background-color: #fff;
}

.calculator-screen {
    width: 100%;
    height: 50px;
    border: none;
    background-color: #252525;
    color: #fff;
    text-align: right;
    padding: 10px;
    border-radius: 5px;
    margin-bottom: 20px;
    font-size: 24px;
}

.calculator-keys {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    gap: 10px;
}

button {
    height: 50px;
    border-radius: 5px;
    border: none;
    background-color: #e0e0e0;
    font-size: 18px;
}

button.operator {
    background-color: #ff9f00;
    color: #fff;
}

button.equal-sign {
```

Style.css

```
    background-color: #ff9f00;
    color: #fff;
    height: calc(100% * 2 + 10px);
}

button.all-clear {
    background-color: #ff6b6b;
    color: #fff;
}
```

Javascript.css

```
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 && !isNaN(inputValue)) {
        calculator.firstOperand = inputValue;
    } else if (operator) {
        const result = calculate(firstOperand, inputValue, operator);

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

    calculator.waitingForSecondOperand = true;
    calculator.operator = nextOperator;
}

function calculate(firstOperand, secondOperand, operator) {
    if (operator === '+') {
        return firstOperand + secondOperand;
    } else if (operator === '-') {
        return firstOperand - secondOperand;
    } else if (operator === '*') {
        return firstOperand * secondOperand;
    } else if (operator === '/') {
        return firstOperand / secondOperand;
    }

    return 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();
});
```