Today I have completed calculator task project in that calculator task to give add functionality

and subtraction the values and multiples and divisions.

Code of HTML:-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

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

    <title>calculator</title>

</head>

<body>

    <div class="calculator">

        <section class="cal-screen">

            0

        </section>

        <section class="cal-btn">

            <div class="cal-row">

                <button class="double cal-button">C</button>

                <button class="cal-button">←</button>

                <button class="cal-button">/</button>

            </div>

            <div class="cal-row">

                <button class="cal-button">7</button>

                <button class="cal-button">8</button>

                <button class="cal-button">9</button>

                <button class="cal-button">\*</button>

            </div>

            <div class="cal-row">

                <button class="cal-button">4</button>

                <button class="cal-button">5</button>

                <button class="cal-button">6</button>

                <button class="cal-button">-</button>

            </div>

            <div class="cal-row">

                <button class="cal-button">3</button>

                <button class="cal-button">2</button>

                <button class="cal-button">1</button>

                <button class="cal-button">+</button>

            </div>

            <div class="cal-row">

                <button class="triple cal-button">0</button>

                <button class="cal-button">=</button>

            </div>

        </section>

    </div>

   <script src="cal.js"></script>

</body>

</html>

Code of CSS:-

\* {

    box-sizing: border-box;

}

body {

    margin: 0;

    padding: 0;

    background-color: whitesmoke;

}

.calculator {

    width: 370px;

    background-color: gray;

    color: white;

}

.cal-screen {

    font-size: 40px;

    font-family: 'Courier New', Courier, monospace;

    text-align: right;

    padding: 20px 5px;

}

.cal-button {

    background-color: #222;

    color: white;

    height: 70px;

    width: 24.5%;

    border: none;

    border-radius: 0;

    cursor: pointer;

    font-size: 25px;

    border: solid 2px black;

    text-shadow: 2rem;

}

.cal-button:hover {

    background-color: lightblue;

    color: black;

}

.cal-button:active {

    background-color: lightsalmon;

}

.cal-button:last-child {

    background-color: goldenrod;

}

.cal-button:last-child:hover {

    background-color: skyblue;

}

.double {

    width: 49.9%;

}

.triple {

    width: 75.8%;

}

.cal-row {

    display: flex;

    align-content: stretch;

    justify-content: space-between;

    margin-bottom: 0.5%;

}

Code of JS:-

let runningtotal = 0;

let buffer = '0';

let previousoperator = null;

const screen = document.querySelector(".cal-screen");

document.querySelector('.cal-btn').addEventListener('click', function (event) {

    buttonClick(event.target.innerText);

});

function buttonClick(value) {

    if (isNaN(parseInt(value))) {

        handleSymbol(value)

    } else {

        handleNumber(value)

    }

    rerender();

}

function handleNumber(value) {

    if (buffer === '0') {

        buffer = value;

    } else {

        buffer += value;

    }

}

function handleSymbol(value) {

    switch (value) {

        case 'C':

            buffer = '0';

            runningtotal = 0;

            previousoperator = null;

            break;

        case "=":

            if (previousoperator === null) {

                return;

            }

            flushOperation(parseInt(buffer));

            previousoperator = null

            buffer = "" + runningtotal;

            runningtotal = 0;

            break;

        case "←":

            if (buffer.length === 1) {

                buffer = "0";

            } else {

                buffer = buffer.substring(0, buffer.length - 1);

            }

            break;

        default:

            handleMath(value);

            break;

    }

}

function rerender() {

    screen.innerText = buffer;

}

function handleMath(value) {

    const intbuffer = parseInt(buffer);

    if (runningtotal === 0) {

        runningtotal = intbuffer;

    } else {

        flushOperation(intbuffer);

    }

    previousoperator = value;

    buffer = "0";

}

function flushOperation(intbuffer) {

    if (previousoperator === "+") {

        runningtotal += intbuffer;

    } else if (previousoperator === "-") {

        runningtotal -= intbuffer;

    } else if (previousoperator === "\*") {

        runningtotal \*= intbuffer;

    } else {

        runningtotal /= intbuffer

    }

}

Output:-

