**Week :1**

**Task**: Advanced Calculator Web Application

**Tech Stack:** HTML, CSS, JavaScript

**Features Implemented:**

* Clean and responsive calculator UI
* Basic arithmetic operations (+, −, ×, ÷)
* Advanced operations like **square, square root, and percentage**
* Keyboard input support
* Error handling for invalid inputs

**Index Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Advanced Calculator</title>

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

</head>

<body>

<nav class="navbar">

    <div class="logo">CalcPro</div>

</nav>

<div class="calculator">

    <input type="text" id="display" value="0" readonly>

    <div class="buttons">

        <button class="control" onclick="clearDisplay()">C</button>

        <button class="control" onclick="deleteLast()">DEL</button>

        <button class="operator" onclick="calculatePercent()">%</button>

        <button class="operator" onclick="appendOperator('/')">÷</button>

        <button class="operator" onclick="calculateSqrt()">√</button>

        <button class="operator" onclick="appendOperator('\*')">×</button>

        <button class="operator" onclick="calculateSquare()">x²</button>

        <button class="operator" onclick="appendOperator('-')">−</button>

        <button onclick="appendNumber('7')">7</button>

        <button onclick="appendNumber('8')">8</button>

        <button onclick="appendNumber('9')">9</button>

        <button class="operator" onclick="appendOperator('+')">+</button>

        <button onclick="appendNumber('4')">4</button>

        <button onclick="appendNumber('5')">5</button>

        <button onclick="appendNumber('6')">6</button>

        <button class="equals" onclick="calculateResult()">=</button>

        <button onclick="appendNumber('1')">1</button>

        <button onclick="appendNumber('2')">2</button>

        <button onclick="appendNumber('3')">3</button>

        <button class="zero" onclick="appendNumber('0')">0</button>

        <button onclick="appendNumber('.')">.</button>

    </div>

</div>

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

</body>

</html>

CSS code:

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

    min-height: 100vh;

    background: linear-gradient(135deg, #667eea, #764ba2);

    font-family: 'Segoe UI', sans-serif;

    display: flex;

    justify-content: center;

    align-items: center;

}

.navbar {

    position: fixed;

    top: 0;

    left: 0;

    width: 100%;

    padding: 20px 28px;

    z-index: 10;

}

.logo {

    display: inline-block;

    padding: 8px 18px;

    font-size: 22px;

    font-weight: 800;

    letter-spacing: 1px;

    color: #4f46e5;

    background: #ffffff;

    border-radius: 14px;

    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.25);

}

.calculator {

    width: 340px;

    padding: 22px;

    background: #ffffff;

    border-radius: 18px;

    box-shadow: 0 25px 50px rgba(0,0,0,0.3);

    margin-top: 80px; /\* prevents navbar overlap \*/

}

#display {

    width: 100%;

    height: 64px;

    font-size: 26px;

    text-align: right;

    padding: 12px;

    margin-bottom: 18px;

    border-radius: 12px;

    border: none;

    background: #f3f4f6;

}

.buttons {

    display: grid;

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

    grid-auto-rows: 56px;

    gap: 14px;

}

button {

    height: 56px;

    font-size: 18px;

    font-weight: 600;

    border: none;

    border-radius: 14px;

    cursor: pointer;

    background: #e5e7eb;

    transition: all 0.15s ease;

}

button:hover {

    transform: translateY(-2px);

    box-shadow: 0 8px 16px rgba(0,0,0,0.2);

}

button:active {

    transform: scale(0.95);

}

.operator {

    background: linear-gradient(135deg, #ff9a9e, #fad0c4);

}

.control {

    background: linear-gradient(135deg, #fbc2eb, #a6c1ee);

}

.equals {

    background: linear-gradient(135deg, #43e97b, #38f9d7);

    color: #000;

    grid-row: span 3;

    height: 100%;

}

.zero {

    grid-column: span 2;

}

JS Code:

const display = document.getElementById("display");

let expression = "";

function appendNumber(num) {

    if (display.value === "Error") clearDisplay();

    if (num === "." && expression.includes(".")) return;

    expression += num;

    display.value = expression;

}

function appendOperator(op) {

    if (!expression) return;

    const lastChar = expression.slice(-1);

    if ("+-\*/".includes(lastChar)) return;

    expression += op;

    display.value = expression;

}

function clearDisplay() {

    expression = "";

    display.value = "0";

}

function deleteLast() {

    expression = expression.slice(0, -1);

    display.value = expression || "0";

}

function calculateResult() {

    try {

        const result = Function(`"use strict"; return (${expression})`)();

        expression = result.toString();

        display.value = expression;

    } catch {

        display.value = "Error";

        expression = "";

    }

}

function calculatePercent() {

    if (!expression) return;

    expression = (parseFloat(expression) / 100).toString();

    display.value = expression;

}

function calculateSqrt() {

    const value = Number(expression);

    if (value < 0 || isNaN(value)) {

        display.value = "Error";

        expression = "";

        return;

    }

    expression = Math.sqrt(value).toString();

    display.value = expression;

}

function calculateSquare() {

    const value = Number(expression);

    if (isNaN(value)) {

        display.value = "Error";

        expression = "";

        return;

    }

    expression = (value \* value).toString();

    display.value = expression;

}

document.addEventListener("keydown", e => {

    if (!isNaN(e.key) || e.key === ".") {

        appendNumber(e.key);

    } else if ("+-\*/".includes(e.key)) {

        appendOperator(e.key);

    } else if (e.key === "Enter") {

        calculateResult();

    } else if (e.key === "Backspace") {

        deleteLast();

    } else if (e.key === "Escape") {

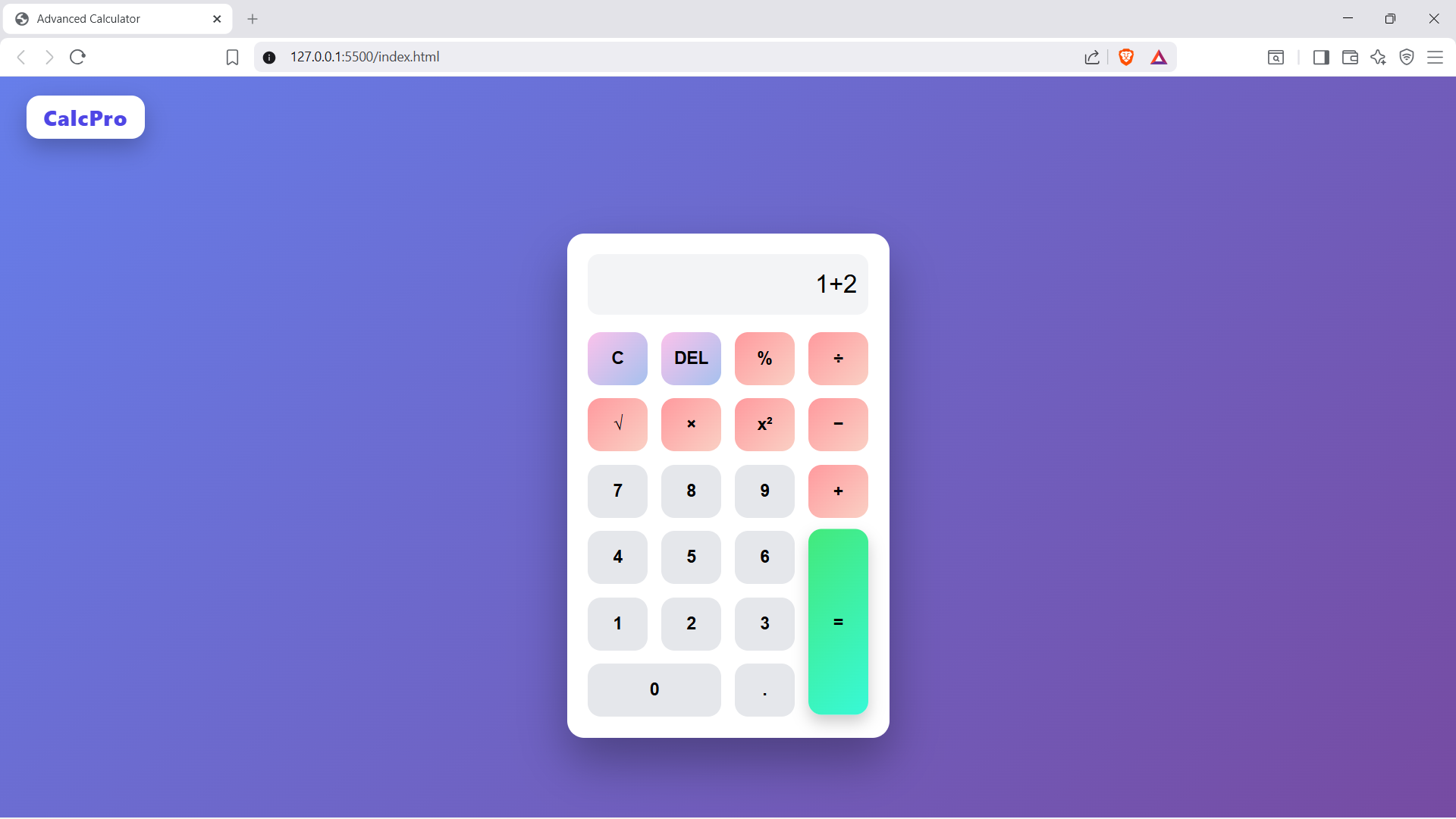
        clearDisplay();

    }

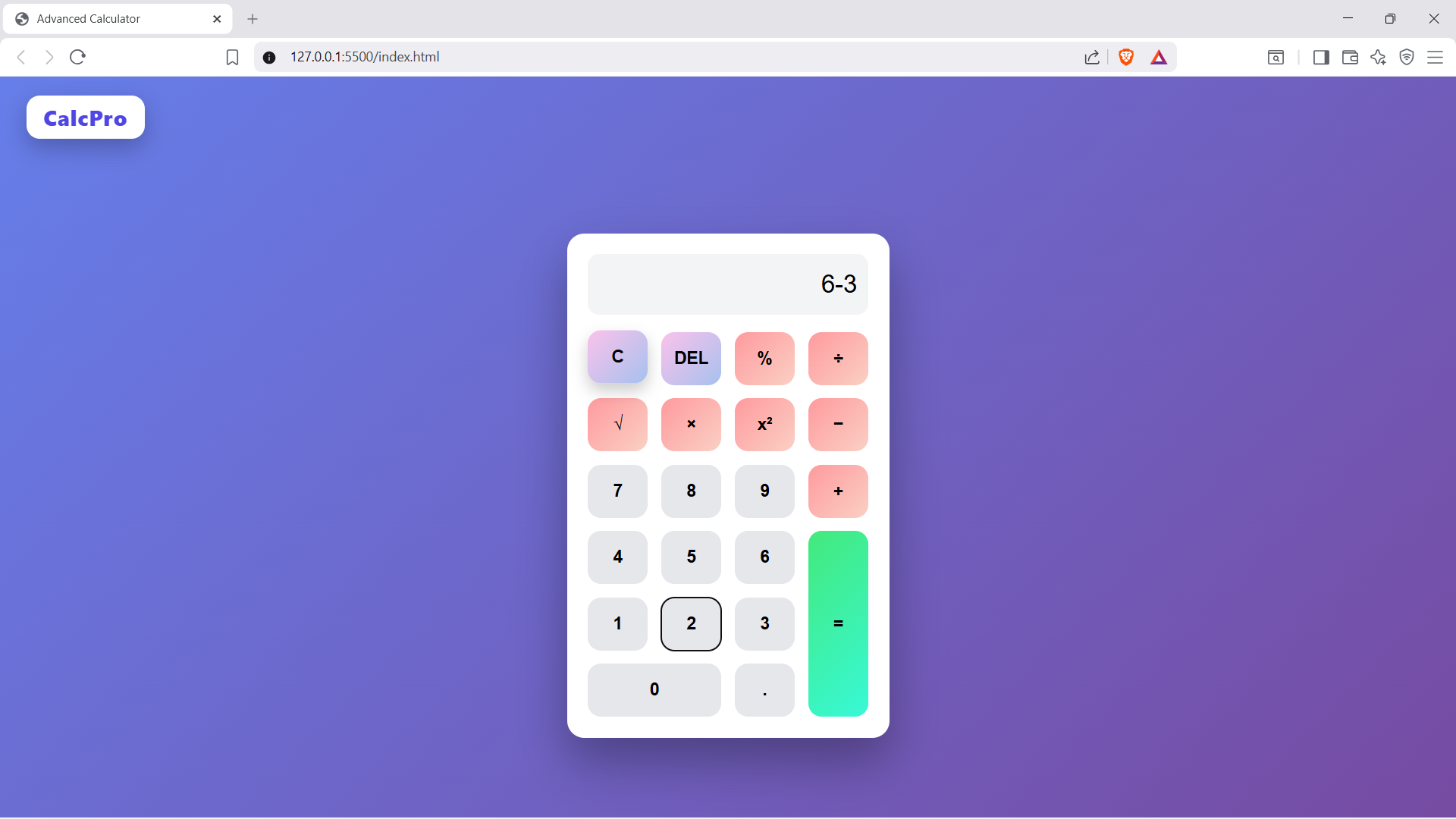
});

OUTPUT:

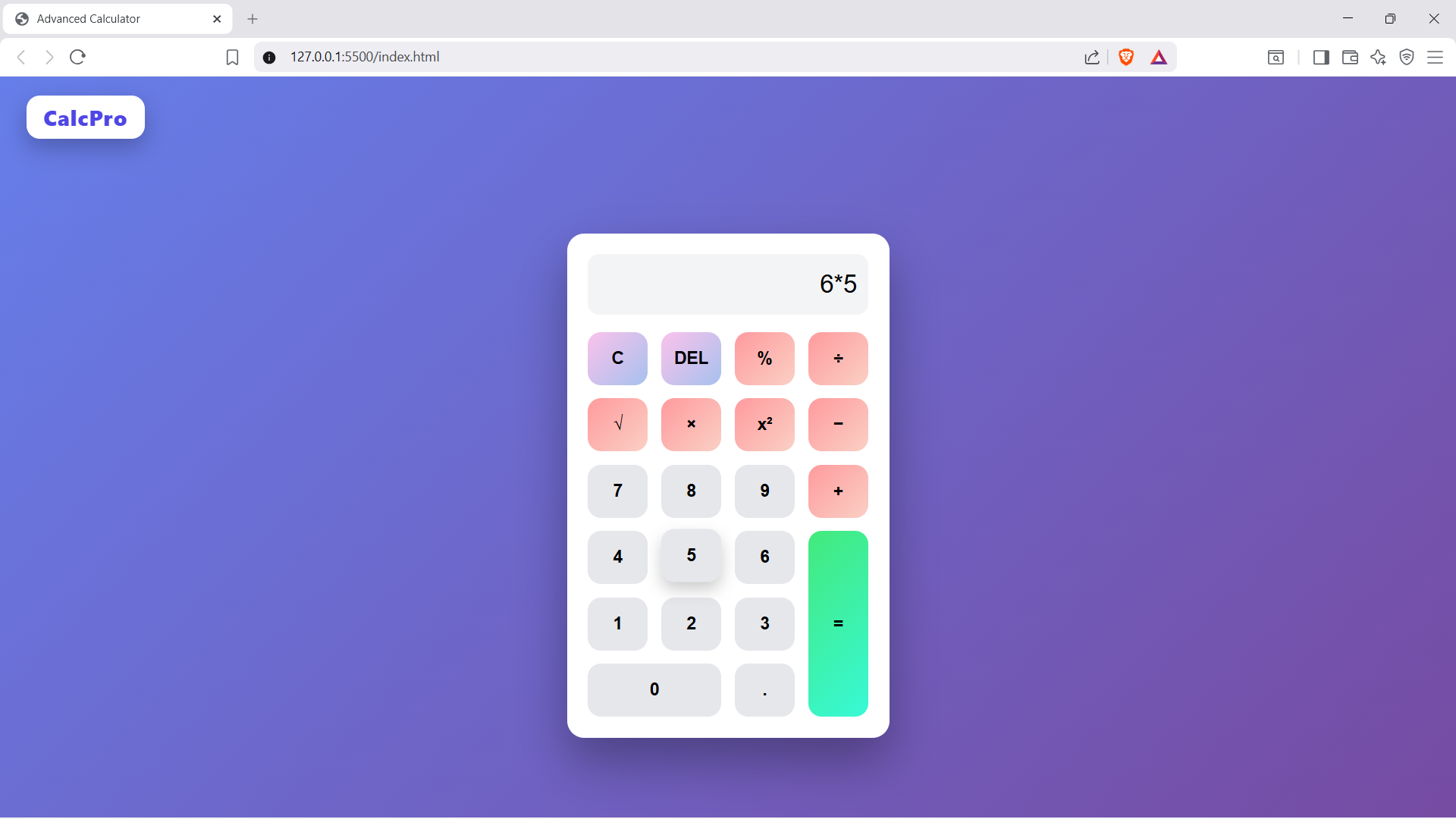
Addition



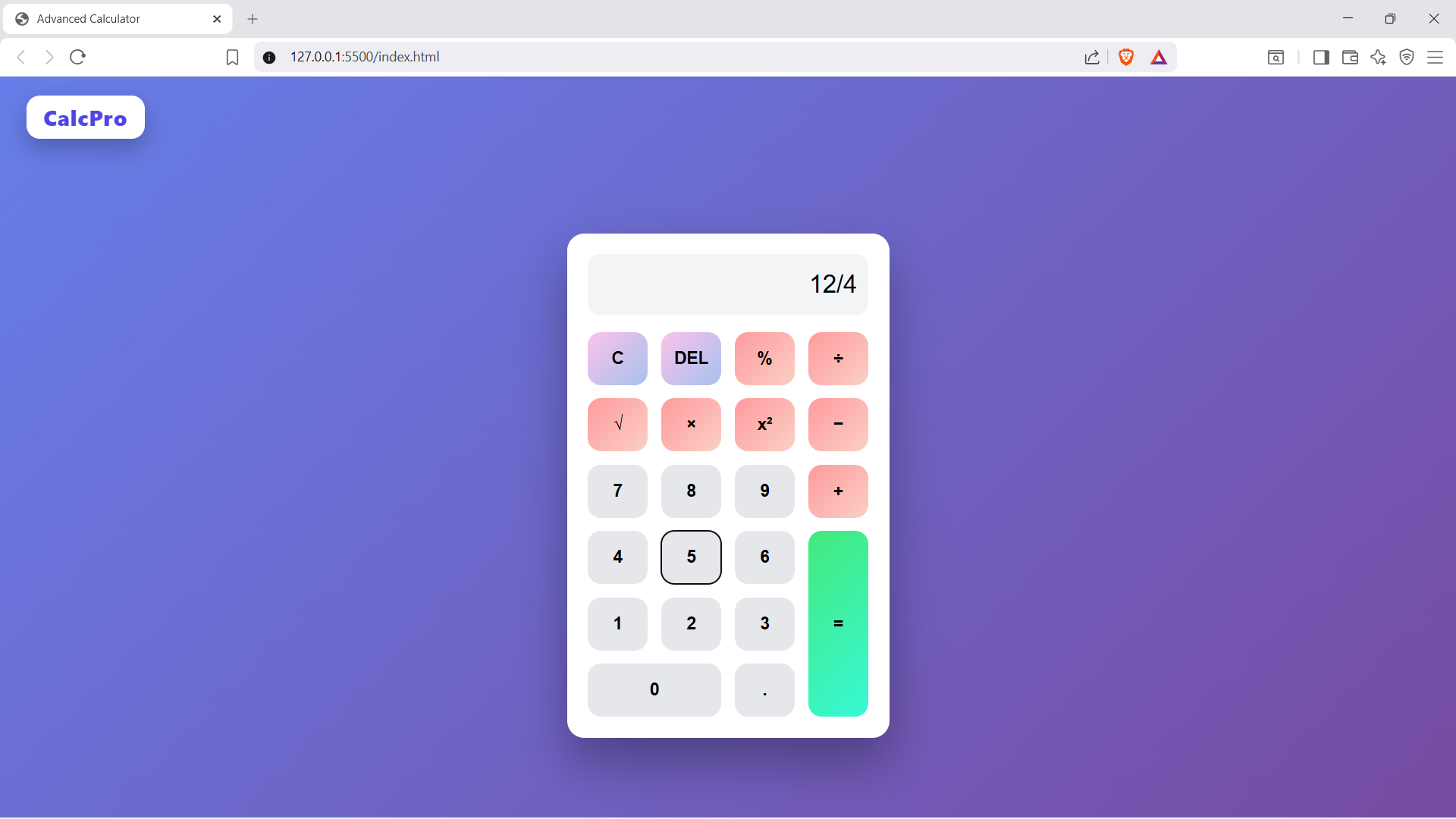
Subtraction



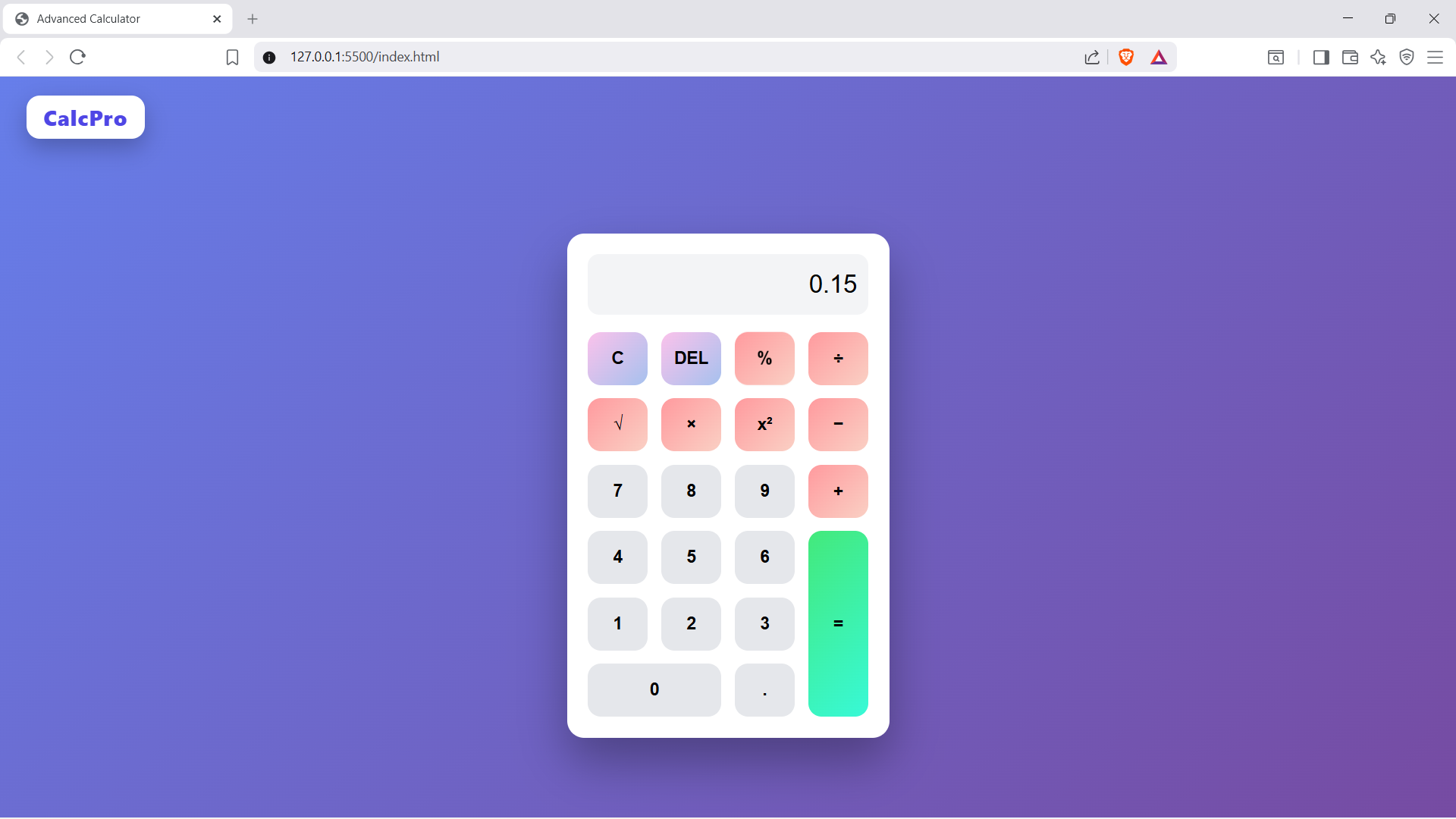
Multiplication:



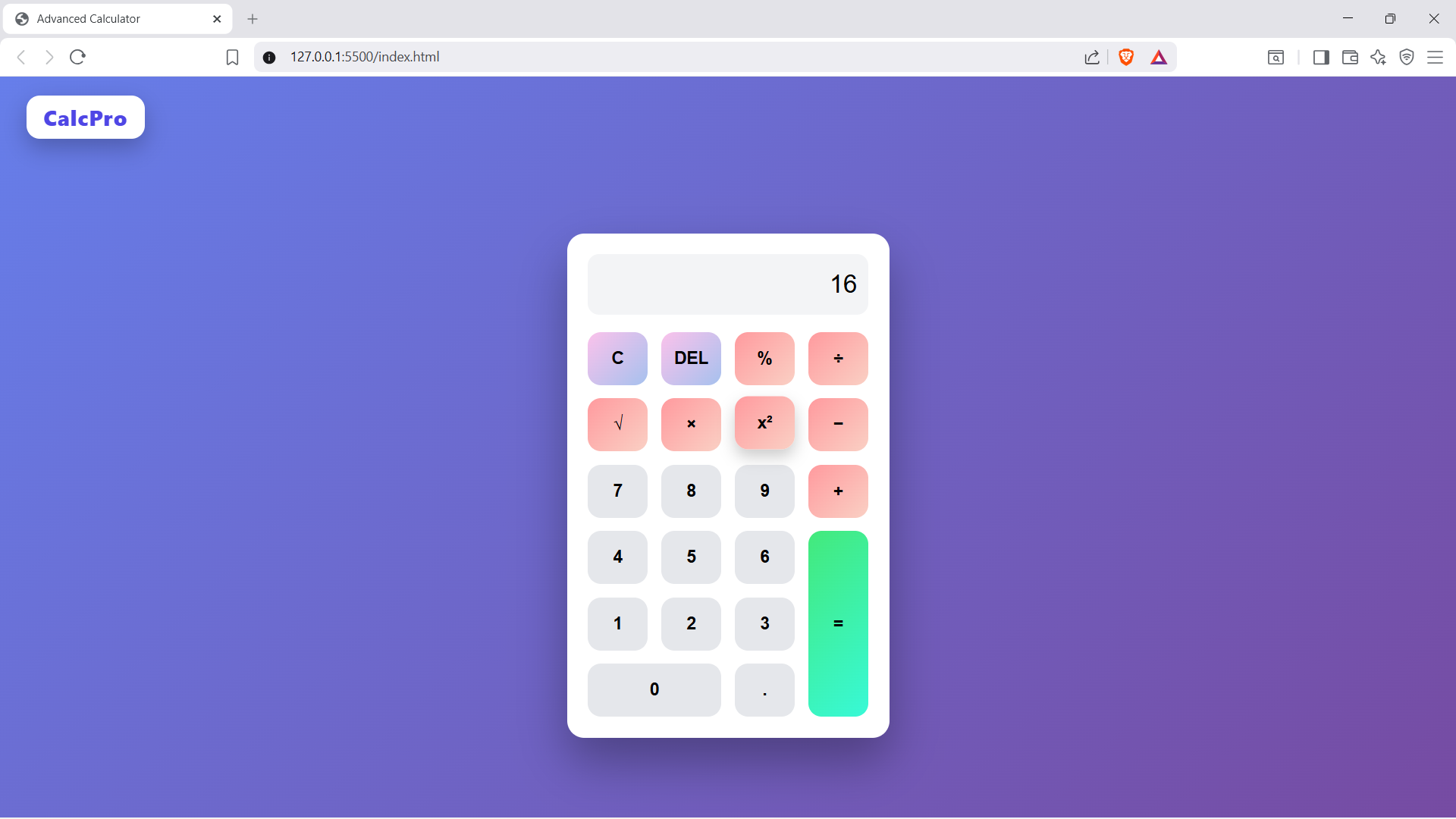
Division:



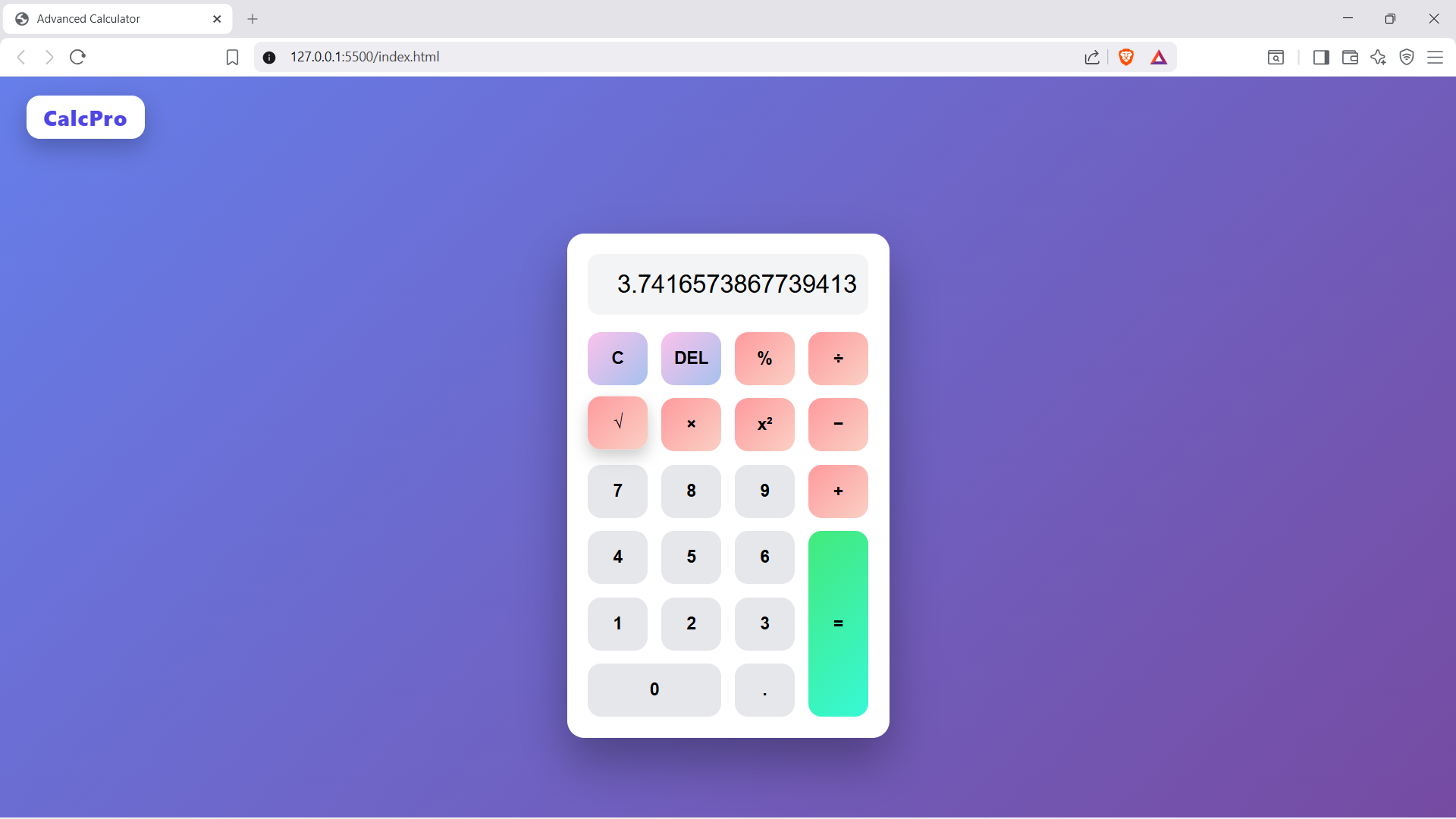
Percentage:



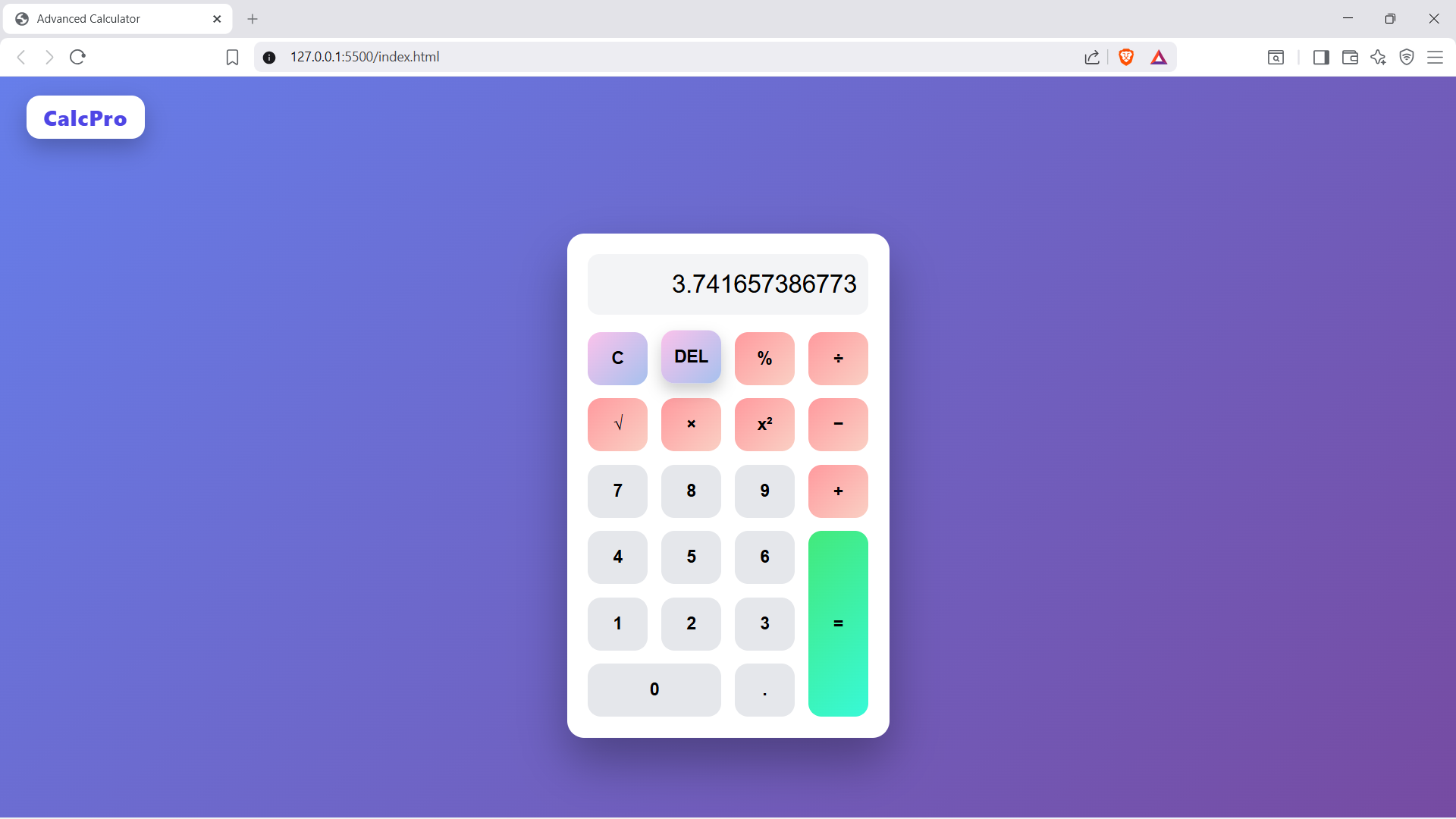
Square:



Root:



Del:



Clear:

