Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

Calculator

Supervised By:

Raveesh Samkaria

Submitted By:

Name:- Saniya Ahuja

Roll no. 2210990787

Group :- G12

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

TABLE OF CONTENTS

ABSTRACT

This project introduces a dynamic calculator developed using JavaScript for computation and Bootstrap for crafting an intuitive user interface (UI). The calculator empowers users with a seamless and visually engaging platform for performing a wide range of mathematical operations.

Key features of this calculator project include:

1. **Efficient Calculation Engine**: JavaScript serves as the backbone for the calculator's computation, enabling swift and accurate processing of arithmetic expressions. From simple addition to complex equations, the calculator leverages JavaScript's computational prowess to deliver precise results.
2. **User-Friendly Design**: Bootstrap's UI components are seamlessly integrated to create an intuitive and responsive user interface. The calculator adapts flawlessly to various screen sizes and devices, ensuring a consistent user experience across desktops, tablets, and mobile devices.
3. **Interactive User Experience**: The calculator's interface is designed with user interaction in mind, featuring clear input fields, intuitive operators, and responsive buttons. Bootstrap's UI elements enhance the aesthetic appeal and usability of the calculator, making mathematical tasks effortless and engaging.
4. **Error Handling and Validation**: Robust error handling mechanisms are implemented to provide informative feedback to users in case of invalid inputs or mathematical errors. Error messages are displayed in a user-friendly manner, guiding users towards correct usage and enhancing overall usability.
5. **Customization Options**: Users have the flexibility to customize the calculator's appearance and layout using Bootstrap's extensive styling options. Themes, colors, and typography can be easily adjusted to suit individual preferences, offering a personalized experience.

By combining the computational capabilities of JavaScript with the design flexibility of Bootstrap, this calculator project delivers a powerful yet user-friendly tool for performing mathematical calculations. Whether for educational purposes, professional use, or everyday tasks, the calculator offers a versatile solution with a modern and visually appealing interface.

PROBLEM DEFINITION AND REQUIREMENTS

Problem Definition:

The aim of this project is to develop a web-based calculator application using JavaScript for computation and Bootstrap for the user interface (UI). The goal is to create a calculator that is intuitive, visually appealing, and responsive across various devices. The calculator should support basic arithmetic operations as well as more advanced features such as handling parentheses and exponentiation. Error handling should be robust, providing clear feedback to users in case of invalid inputs or mathematical errors.

Requirements:

1. **Functional Requirements:**
    - **Arithmetic Operations:** The calculator should support addition, subtraction, multiplication, and division.
    - **Advanced Operations:** It should handle more complex operations such as exponentiation (^) and parentheses for order of operations.
    - **Real-time Calculation:** The calculator should provide real-time calculation as users input numbers and operators.
    - **Error Handling:** Robust error handling should be implemented to provide informative feedback to users for invalid inputs or mathematical errors.
2. **User Interface Requirements:**
    - **Clear Display:** The display should be clear and easily readable, showing both input expression and result.
    - **Intuitive Controls:** User interface elements such as buttons for numbers and operators should be intuitive and easy to use.

    .
3. **Development Requirements:**

- **JavaScript Implementation:** All computation logic should be implemented using JavaScript.
- **Bootstrap Integration:** Bootstrap should be integrated to design and style the calculator UI efficiently.
- **Modular Code:** The codebase should be well-structured and modular, facilitating easy maintenance and future enhancements.
- **Testing:** Comprehensive testing should be conducted to ensure the calculator functions correctly under various scenarios, including edge cases and invalid inputs.

By meeting these requirements, the JavaScript calculator with Bootstrap UI will provide users with a reliable, user-friendly, and visually appealing tool for performing mathematical calculations on the web.

PROPOSED DESIGN

1. Layout Structure:

- Utilize a clean and minimalist layout with distinct sections for input, output, and buttons.
- Arrange the calculator components in a grid layout for better organization and aesthetics.

2. Input Field:

- Implement a responsive input field at the top of the calculator for users to enter their calculations.
- Ensure the input field adjusts dynamically to accommodate various screen sizes.

3. Output Display:

- Display the calculation results prominently below the input field.
- Use a clear and readable font for better legibility.

4. Button Design:

- Design calculator buttons using Bootstrap's button components.
- Apply consistent styling for buttons to maintain visual coherence.
- Differentiate numeric, operator, and special function buttons using color coding or icons.

5. Operator Buttons:

- Use recognizable symbols for arithmetic operators (+, -, ×, ÷) for easy identification.

- Ensure operator buttons are spaced adequately to prevent accidental clicks.

6. **Accessibility Features:**

- Ensure the calculator is accessible to users with disabilities by incorporating features like keyboard navigation and screen reader compatibility.
- Follow best practices for web accessibility to enhance inclusivity.

7. Testing and Optimization:

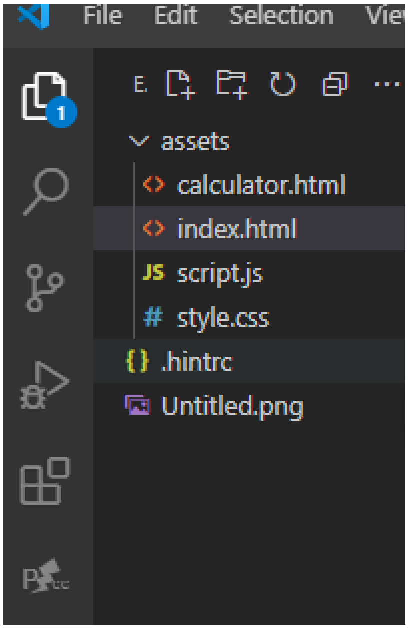
- Conduct thorough testing across different browsers and devices to ensure cross-compatibility and functionality.
- Optimize the code and assets for performance to minimize loading times and enhance user experience.

8. Documentation:

- Provide clear and comprehensive documentation outlining the calculator's features, usage instructions, and customization options.
- Include examples and code snippets to assist developers in integrating the calculator into their projects.

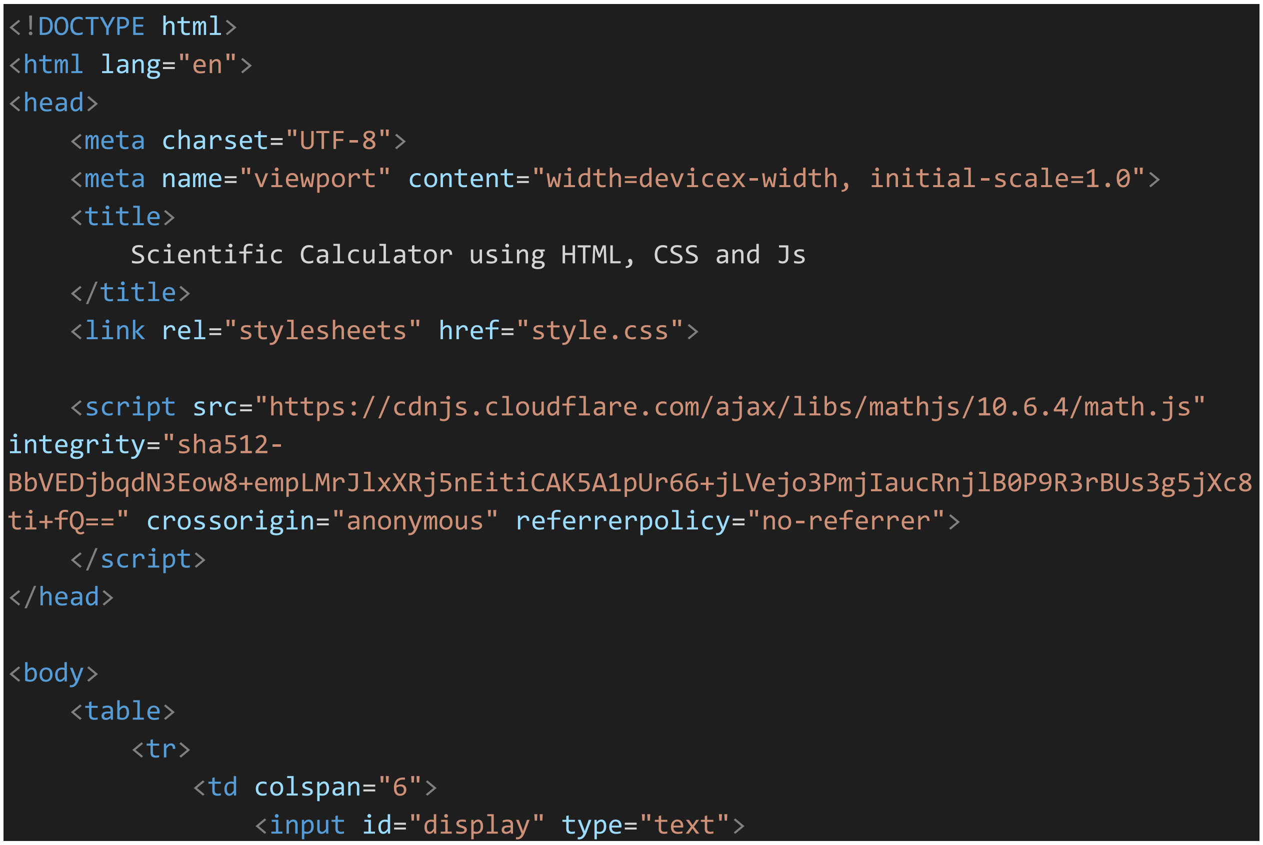
By following these design guidelines, the JavaScript calculator with Bootstrap UI can offer users a visually appealing, intuitive, and functional tool for performing mathematical calculations efficiently across various platforms.

FOLDER STRUCTURE

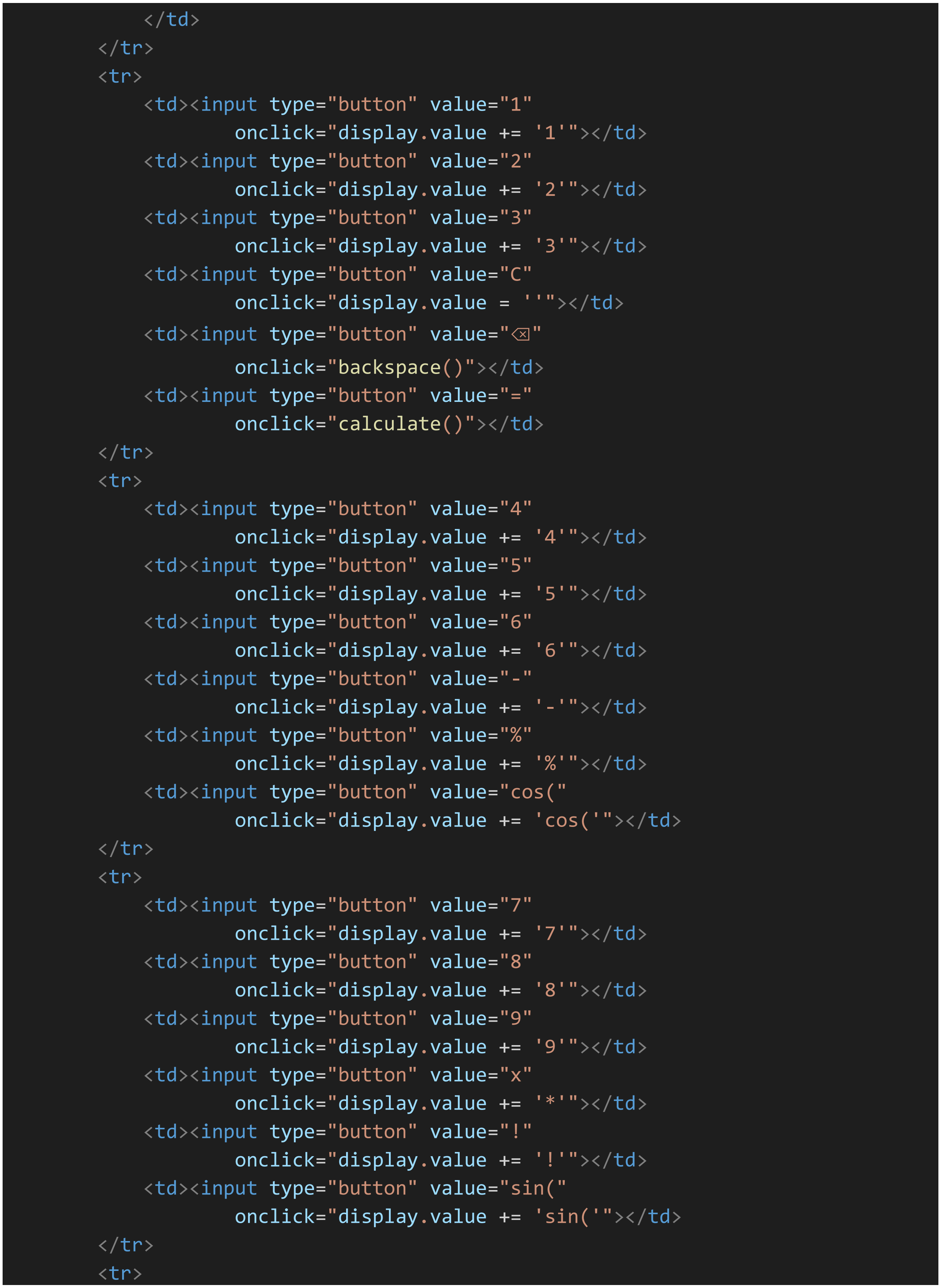


CODE

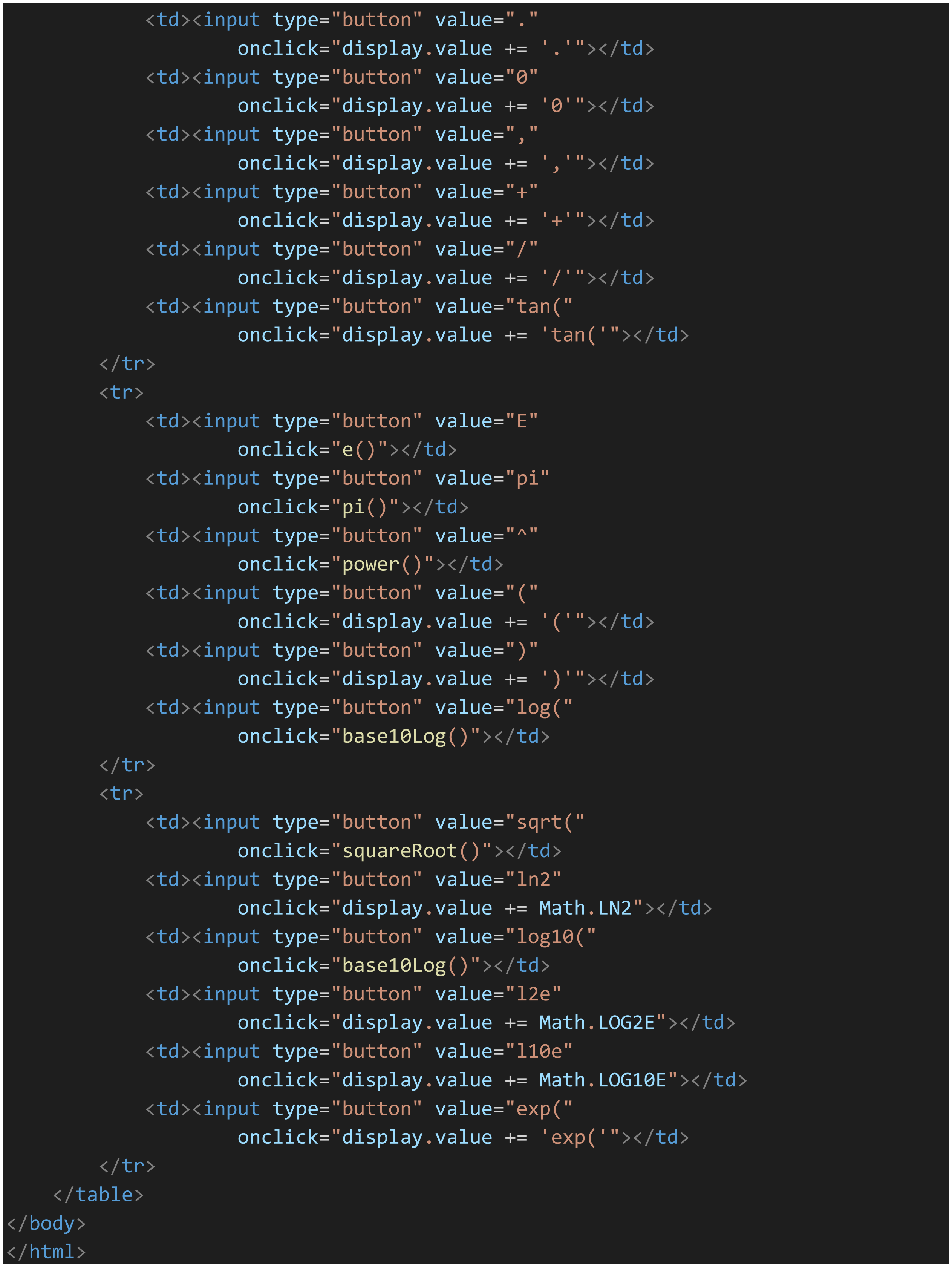
HTML CODE:-



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=devicex-width, initial-scale=1.0">
    <title>
        Scientific Calculator using HTML, CSS and Js
    </title>
    <link rel="stylesheets" href="style.css">

    <script src="https://cdnjs.cloudflare.com/ajax/libs/mathjs/10.6.4/math.js"
integrity="sha512-BbVEDjbqdN3Eow8+empLMrJlxXRj5nEitiCAK5A1pUr66+jLVejo3PmjIaucRnjlB0P9R3rBUs3g5jXc8ti+fQ==" crossorigin="anonymous" referrerpolicy="no-referrer">
    </script>
</head>

<body>
    <table>
        <tr>
            <td colspan="6">
                <input id="display" type="text">
```



```
            </td>
        </tr>
        <tr>
            <td><input type="button" value="1"
                    onclick="display.value += '1'"></td>
            <td><input type="button" value="2"
                    onclick="display.value += '2'"></td>
            <td><input type="button" value="3"
                    onclick="display.value += '3'"></td>
            <td><input type="button" value="C"
                    onclick="display.value = ''"></td>
            <td><input type="button" value="⌫"
                    onclick="backspace()"></td>
            <td><input type="button" value="="
                    onclick="calculate()"></td>
        </tr>
        <tr>
            <td><input type="button" value="4"
                    onclick="display.value += '4'"></td>
            <td><input type="button" value="5"
                    onclick="display.value += '5'"></td>
            <td><input type="button" value="6"
                    onclick="display.value += '6'"></td>
            <td><input type="button" value="-"
                    onclick="display.value += '-'"></td>
            <td><input type="button" value="%"
                    onclick="display.value += '%'"></td>
            <td><input type="button" value="cos("
                    onclick="display.value += 'cos('"></td>
        </tr>
        <tr>
            <td><input type="button" value="7"
                    onclick="display.value += '7'"></td>
            <td><input type="button" value="8"
                    onclick="display.value += '8'"></td>
            <td><input type="button" value="9"
                    onclick="display.value += '9'"></td>
            <td><input type="button" value="x"
                    onclick="display.value += '*'"></td>
            <td><input type="button" value="!"
                    onclick="display.value += '!'"></td>
            <td><input type="button" value="sin("
                    onclick="display.value += 'sin('"></td>
        </tr>
        <tr>
```



```
            <td><input type="button" value="."
                    onclick="display.value += '.'"></td>
            <td><input type="button" value="0"
                    onclick="display.value += '0'"></td>
            <td><input type="button" value=","
                    onclick="display.value += ','"></td>
            <td><input type="button" value="+"
                    onclick="display.value += '+'"></td>
            <td><input type="button" value="/"
                    onclick="display.value += '/'"></td>
            <td><input type="button" value="tan("
                    onclick="display.value += 'tan('"></td>
        </tr>
        <tr>
            <td><input type="button" value="E"
                    onclick="e()"></td>
            <td><input type="button" value="pi"
                    onclick="pi()"></td>
            <td><input type="button" value="^"
                    onclick="power()"></td>
            <td><input type="button" value="("
                    onclick="display.value += '('"></td>
            <td><input type="button" value=")"
                    onclick="display.value += ')'"></td>
            <td><input type="button" value="log("
                    onclick="base10Log()"></td>
        </tr>
        <tr>
            <td><input type="button" value="sqrt("
                    onclick="squareRoot()"></td>
            <td><input type="button" value="ln2"
                    onclick="display.value += Math.LN2"></td>
            <td><input type="button" value="log10("
                    onclick="base10Log()"></td>
            <td><input type="button" value="l2e"
                    onclick="display.value += Math.LOG2E"></td>
            <td><input type="button" value="l10e"
                    onclick="display.value += Math.LOG10E"></td>
            <td><input type="button" value="exp("
                    onclick="display.value += 'exp('"></td>
        </tr>
    </table>
</body>
</html>
```

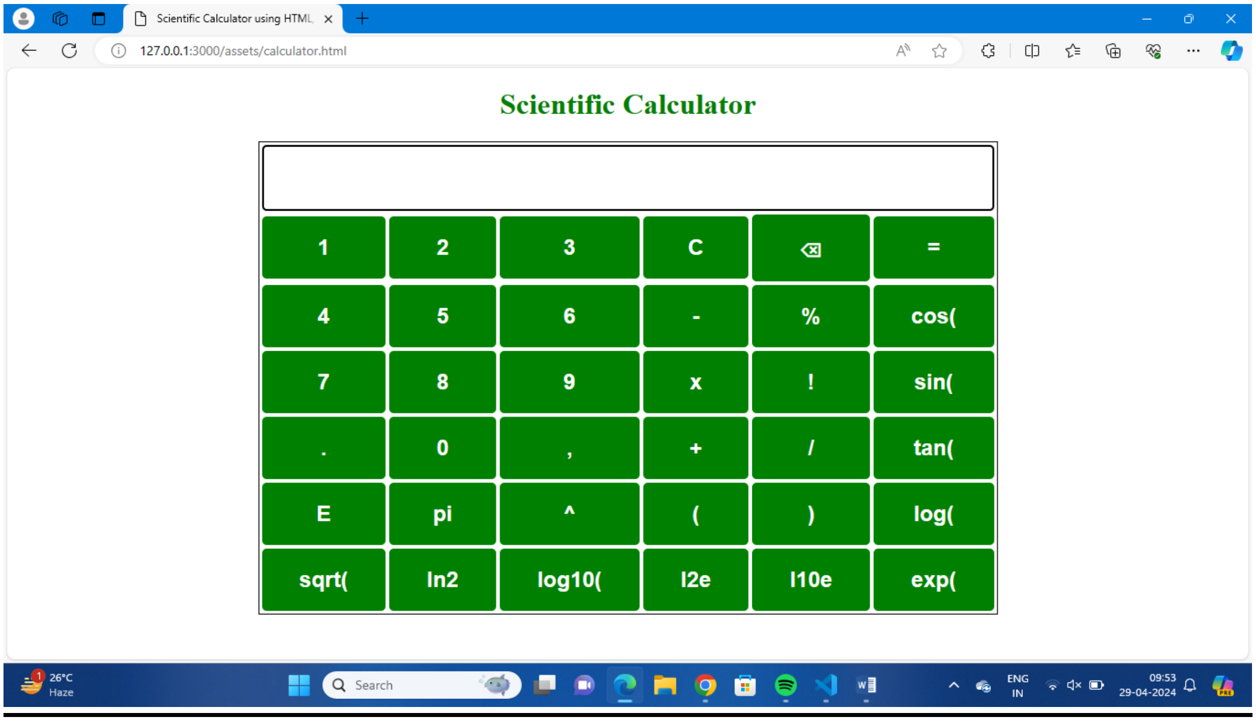
CSS CODE

```
table {
    border: 1px solid black;
    margin-left: auto;
    margin-right: auto;
}

input[type="button"] {
    width: 100%;
    padding: 20px 40px;
    background-color: green;
    color: white;
    font-size: 24px;
    font-weight: bold;
    border: none;
    border-radius: 5px;
}

input[type="text"] {
    padding: 20px 240px;
    font-size: 24px;
    font-weight: bold;
    border: none;
    border-radius: 5px;
    border: 2px solid black;
    text-align: left;
}

display {
    text-align: left;
}
```

JAVASCRI[PT CODE:-

```
function backspace() {
    let display = document.getElementById("display");
    display.value = display.value.slice(0, -1);
}

function calculate() {
    let display = document.getElementById("display");
    let expression = display.value;
    let result;

    try {
        // Convert trigonometric function inputs from degrees to radians
        expression = expression.replace(/sin\(/g, 'sin(' + Math.PI / 180 + '*');
        expression = expression.replace(/cos\(/g, 'cos(' + Math.PI / 180 + '*');
        expression = expression.replace(/tan\(/g, 'tan(' + Math.PI / 180 + '*');

        result = math.evaluate(expression);
        display.value = result;
    } catch (error) {
        display.value = "Error";
    }
}

function squareRoot() {
    let display = document.getElementById("display");
    display.value += "sqrt(";
}

function base10Log() {
    let display = document.getElementById("display");
    display.value += "log(";
}

function pi() {
    let display = document.getElementById("display");
    display.value += "pi";
}

function e() {
    let display = document.getElementById("display");
    display.value += "e";
}

function power() {
    let display = document.getElementById("display");
    display.value += "^(";
}
```

[redacted]



Scientific Calculator using HTML
127.0.0.1:3000/assets/calculator.html
Scientific Calculator
1 2 3 C ⌫ =
4 5 6 - % cos(
7 8 9 x ! sin(
. 0 , + / tan(
E pi ^ ( ) log(
sqrt(ln2 log10(l2e l10e exp(
26°C Haze
Search
ENG IN
09:53 29-04-2024

REFERENCES

https://www.geeksforgeeks.org

https://www.blackbox.ai/

https://chat.openai.com/