Web Technology Laboratory Assignment 4

Name: Aakash A. Joshi

Roll No: 0077

Batch: T3

Topic: Practical Assignment 4

Title: Calculator UI using HTML, CSS and JS

Problem Statement:

Implement an application in Java Script using following:
a) Design UI of application using HTML, CSS etc.
b) Include Java script validation
c) Use of prompt and alert window using Java Script

e.g., Design and implement a simple calculator using Java Script for operations like addition, multiplication, subtraction, division, square of number etc.
a) Design calculator interface like text field for input and output, buttons for numbers and operators etc.
b) Validate input values
c) Prompt/alerts for invalid values etc.

Theory:

HTML (Hypertext Markup Language):

HTML is the standard markup language for creating web pages and web applications. It provides the structure and content of a webpage by using a variety of elements such as headings, paragraphs, lists, forms, etc.

CSS (Cascading Style Sheets):

CSS is used to style the HTML elements, defining how they should be displayed on the webpage. It allows you to control the layout, colors, fonts, and other visual aspects of your webpage.

JavaScript:

JavaScript is a high-level programming language that is commonly used to create interactive effects within web browsers. It can manipulate HTML and CSS, making web pages dynamic and interactive.

Implementation:

Designing the UI (HTML and CSS):

HTML Structure: Define the structure of the calculator interface using HTML elements such as <div>, <input>, <button>, etc.

CSS Styling: Apply CSS styles to the HTML elements to enhance the visual appearance of the calculator, including layout, colors, fonts, etc.

JavaScript Validation:

Input Validation: Implement JavaScript functions to validate user input for the calculator. Check for valid numeric inputs and handle invalid inputs appropriately.

Error Handling: Use JavaScript to display error messages using alert or custom error messages within the UI to prompt users to correct their input.

Using Prompt and Alert Windows:

Prompt Window: Use prompt to get input from the user, such as asking for numbers or operator selection.

Alert Window: Utilize alert to display messages to the user, such as error messages for invalid input or the result of calculations.

Calculator Operations:

Basic Operations: Implement JavaScript functions to perform basic calculator operations such as addition, subtraction, multiplication, and division.

Additional Operations: Include additional functionalities like calculating square of a number or other mathematical operations as per the requirements.

Code:

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-
scale=1.0">
   <title>Scientific Calculator</title>
   <link rel="stylesheet" href="styles.css">
</head>
<body>
   <form name="calculator">
```

```
        <table align="center">
            <tr>
                <td colspan="4">
                    <input type="text" name="result" placeholder="0"
style="text-align: right;">
                </td>
            </tr>
            <tr>
                <td><button type="button" value="sin"
onclick="sin()">sin</button></td>
                <td><button type="button" value="cos"
onclick="cos()">cos</button></td>
                <td><button type="button" value="tan"
onclick="tan()">tan</button></td>
                <td colspan="2"><button type="button" onclick="remove()"
class="clear">C</button></td>
            </tr>
            <tr>
                <td><button type="button" value="x^2"
onclick="square()">x<sup>2</sup></button></td>
                <td><button type="button" value="x^3"
onclick="cube()">x<sup>3</sup></button></td>
                <td><button type="button" value="sqrt2"
onclick="sqrt()">&radic;</button></td>
                <td><button type="button" value="cubrt"
onclick="cubrt()">&#8731;</button></td>
            </tr>
            <tr>
                <td><button type="button" value="7"
onclick="number(value)">7</button></td>
                <td><button type="button" value="8"
onclick="number(value)">8</button></td>
                <td><button type="button" value="9"
onclick="number(value)">9</button></td>
                <td><button type="button" value="BACKSPACE"
onclick="BACKSPACE()"><</button></td>
            </tr>
            <tr>
        <td><button type="button" value="4"
onclick="number(value)">4</button></td>
        <td><button type="button" value="5"
onclick="number(value)">5</button></td>
        <td><button type="button" value="6"
onclick="number(value)">6</button></td>
        <td><button type="button" value="-"
onclick="number(value)">-</button></td>
      </tr>
      <tr>
        <td><button type="button" value="1"
onclick="number(value)">1</button></td>
        <td><button type="button" value="2"
onclick="number(value)">2</button></td>
        <td><button type="button" value="3"
onclick="number(value)">3</button></td>
        <td><button type="button" value="/"
onclick="number(value)">/</button></td>
      </tr>
      <tr>
        <td><button type="button" value="."
onclick="number(value)">.</button></td>
        <td><button type="button" value="0"
onclick="number(value)">0</button></td>
        <td><button type="button" value="*"
onclick="number(value)">*</button></td>
        <td><button type="button" value="%"
onclick="number(value)">%</button></td>
      </tr>
      <tr>
        <td colspan="2"><button type="button" value="="
onclick="equal()" class="equal">=</button></td>
        <td><button type="button" value="+"
onclick="number(value)">+</button></td>
      </tr>
    </table>
  </form>
  <script src="main.js"></script>
```

```
</body>
</html>
```

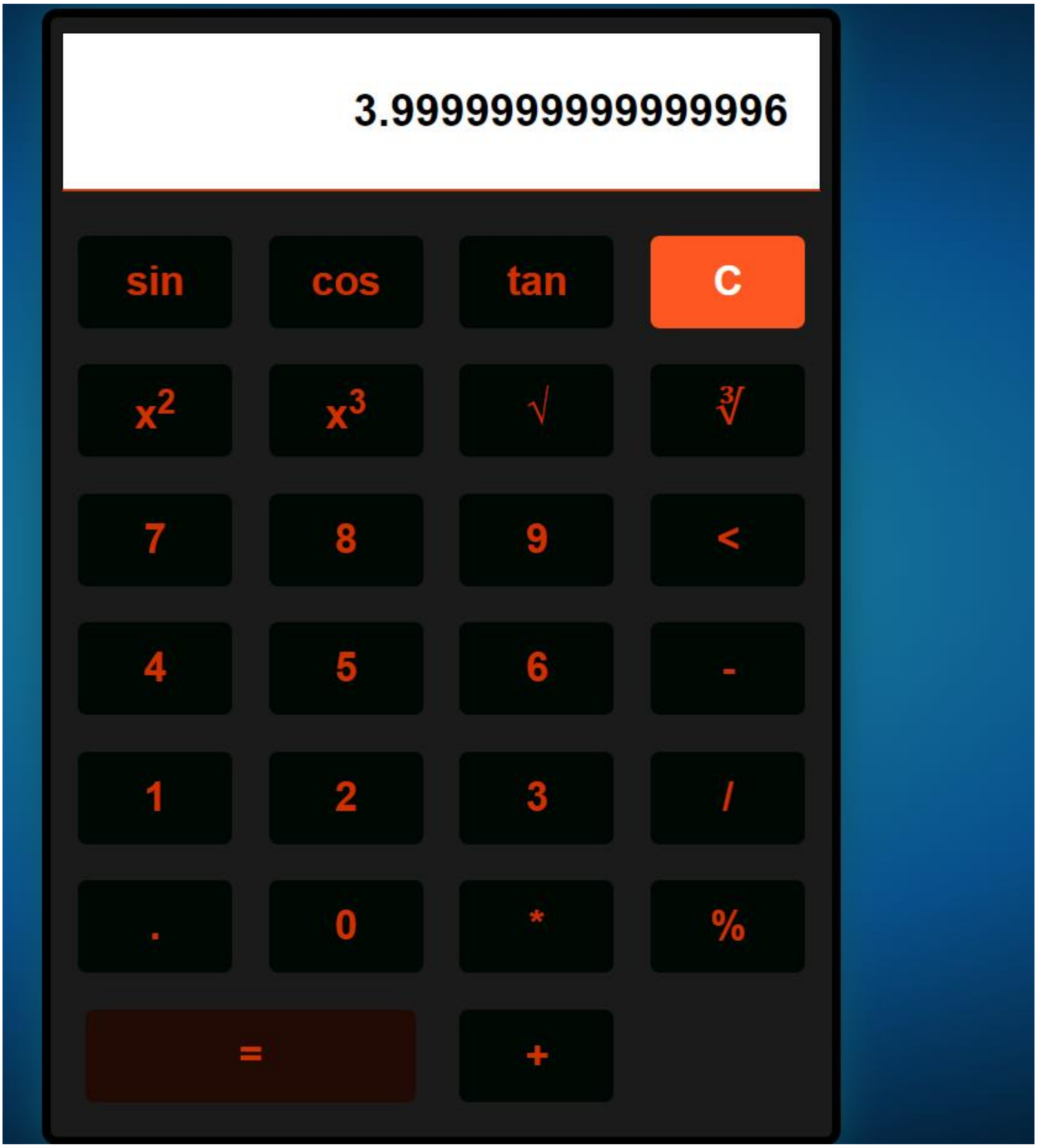
CSS:

```
body {
   margin: 0;
   padding: 50px, 0px;
   background: radial-gradient(#1e8fa5, #08508b, black)no-repeat
fixed;
}

button.clear{
   background: #FF5722;
   color:#fff;
}

table {
   background: #1b1b1b;
   padding: 7px 5px;
   border: 6px solid black;
   text-align: center;
   padding-bottom: 0px;
   box-shadow: 0px 0px 27px #0d5569;
   border-radius: 10px;
   margin-top: 90px;
}

button.equal{
   width: 88%;
   background: #210b04;
}

button.equal:hover{
   background: #000803;
}

input[type="text"]{
   width: 450px;
   height:81px;
   border: 1px solid #000;
   padding: 10px 20px;
   font-size: 29px;
   font-weight: bold;
   margin-bottom: 25px;
   font-family: Helvetica, sans-serif;
   margin-top: #dedede;
   border-bottom: 2px solid #cc4014;
   color: #000;
}

button {
   padding: 10px 20px;
   width: 100px;
   height: 60px;
   margin-bottom: 20px;
   font-size: 26px;
   font-weight: bold;
   font-family: Helvetica, sans-serif;
   background: #000803;
   border: none;
   color: #d23200;
   border-radius: 6px;
   cursor: pointer;
}

button:hover{
   background: #210b04;
}
```

Javascript

```
function sin(){
   document.calculator.result.value =
Math.sin(document.calculator.result.value);
}

function cos(){
   document.calculator.result.value =
Math.cos(document.calculator.result.value);
}

function tan(){
   document.calculator.result.value =
Math.tan(document.calculator.result.value);
}

function BACKSPACE(){
   var a = document.calculator.result.value;
   document.calculator.result.value = a.substr(0, a.length-1);
}

function square(){
   document.calculator.result.value =
Math.pow(document.calculator.result.value, 2);
}

function cube(){
   document.calculator.result.value =
Math.pow(document.calculator.result.value, 3);
}

function sqrt(){
   document.calculator.result.value =
Math.pow(document.calculator.result.value, 1/2);
}

function cubrt(){
   document.calculator.result.value =
Math.pow(document.calculator.result.value, 1/3);
}

function number(value){
   document.calculator.result.value += value;
}

function remove(){
   document.calculator.result.value = " ";
}

function equal(){
   document.calculator.result.value =
eval(document.calculator.result.value);
}
```

Output:



3.9999999999999996

Conclusion:

- This project demonstrates fundamental principles of web development, utilizing HTML for structure and CSS for styling to create an intuitive user interface.
- JavaScript adds interactivity to the application, enabling input validation, error handling, and calculation functionalities, while prompt and alert windows facilitate user interaction and feedback.