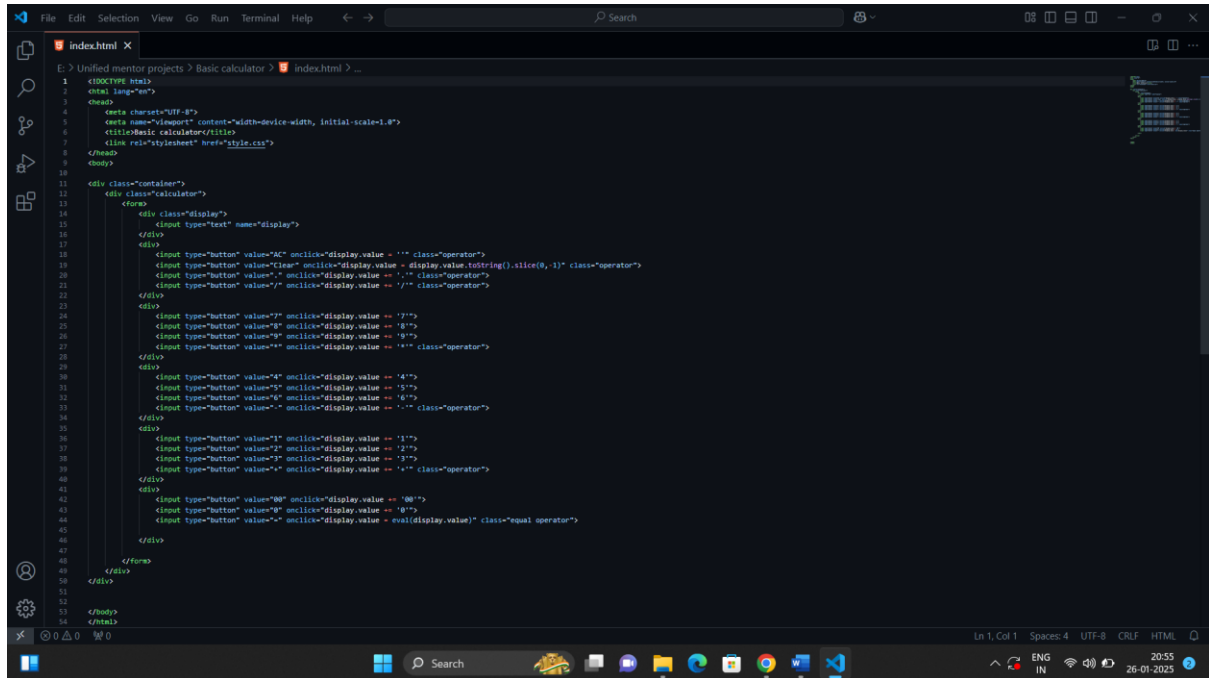


Project Report: Basic Calculator

Objective:

This project aims to create a simple and visually appealing calculator that performs basic arithmetic operations, including addition, subtraction, multiplication, and division.

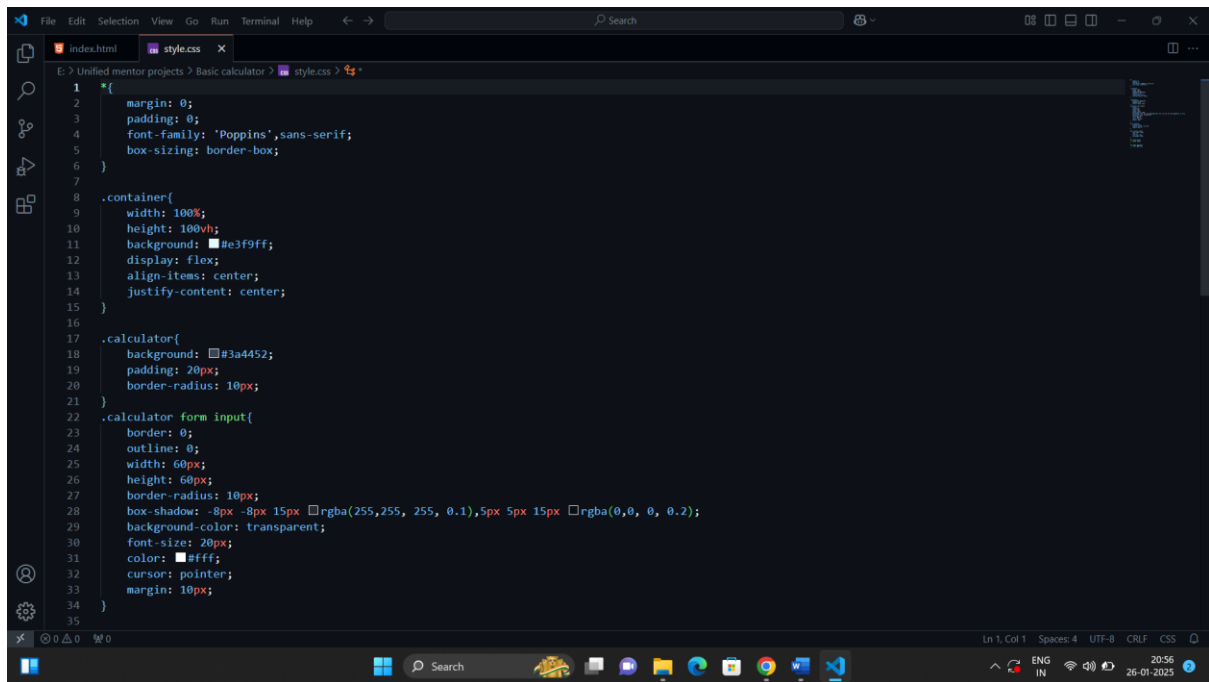
A screenshot of a code editor window showing the HTML structure of a basic calculator. The editor has a dark theme and a sidebar on the left with icons for Explorer, Search, and Run and Debug. The main area displays the code for index.html. The code starts with a DOCTYPE declaration, charset, viewport, and a title. It includes a link to a CSS file. The main content is wrapped in a <div> with class 'container'. Inside, there's another <div> with class 'calculator'. This contains a <div> with class 'display' for the input field, and a <form> containing buttons for numbers (0-9), operators (+, -, *, /), and functions (AC, Clear, %, =). Each button has an onclick attribute to update the display or perform a calculation using eval().

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Basic calculator</title>
7     <link rel="stylesheet" href="style.css">
8   </head>
9   <body>
10
11     <div class="container">
12       <div class="calculator">
13         <div class="display">
14           <input type="text" name="display">
15         </div>
16         <div>
17           <input type="button" value="AC" onclick="display.value = ''" class="operator">
18           <input type="button" value="Clear" onclick="display.value = display.value.toString().slice(0, 1)" class="operator">
19           <input type="button" value="%" onclick="display.value = '/'" class="operator">
20           <input type="button" value="+" onclick="display.value += '+'" class="operator">
21         </div>
22         <div>
23           <input type="button" value="7" onclick="display.value += '7'">
24           <input type="button" value="8" onclick="display.value += '8'">
25           <input type="button" value="9" onclick="display.value += '9'">
26           <input type="button" value="*" onclick="display.value += '*'" class="operator">
27         </div>
28         <div>
29           <input type="button" value="4" onclick="display.value += '4'">
30           <input type="button" value="5" onclick="display.value += '5'">
31           <input type="button" value="6" onclick="display.value += '6'">
32           <input type="button" value="-" onclick="display.value += '-'" class="operator">
33         </div>
34         <div>
35           <input type="button" value="1" onclick="display.value += '1'">
36           <input type="button" value="2" onclick="display.value += '2'">
37           <input type="button" value="3" onclick="display.value += '3'">
38           <input type="button" value="+" onclick="display.value += '+'" class="operator">
39         </div>
40         <div>
41           <input type="button" value="0" onclick="display.value += '0'">
42           <input type="button" value="00" onclick="display.value += '00'">
43           <input type="button" value="=" onclick="display.value = eval(display.value)" class="equal operator">
44         </div>
45       </div>
46     </div>
47   </body>
48 </html>
```

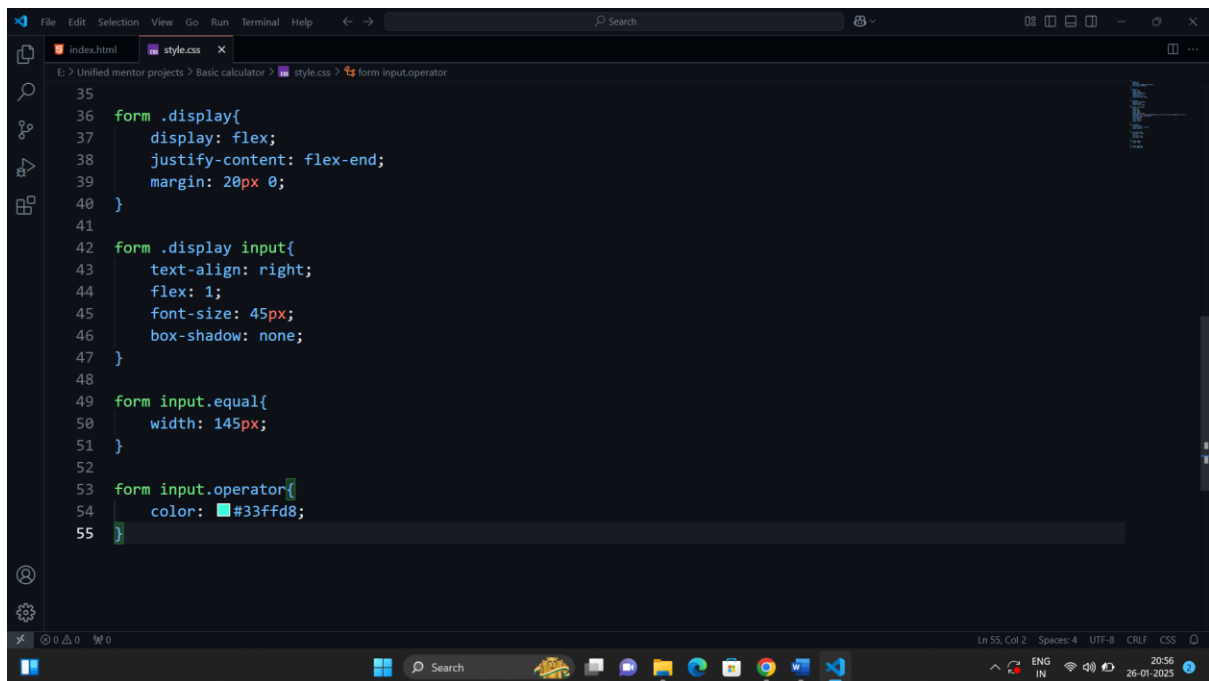
Implementation Details:

1. HTML Structure:

- The index.html file defines the calculator's layout within a <div> container.
- It includes an input field for displaying the calculation and buttons for numbers, operators, and functions like "AC" (clear all) and "Clear" (remove the last digit).
- The form uses onclick attributes in buttons to define behavior for updating the display or performing operations like eval() for calculations.



```
1  *{
2    margin: 0;
3    padding: 0;
4    font-family: 'Poppins', sans-serif;
5    box-sizing: border-box;
6  }
7
8  .container{
9    width: 100%;
10   height: 100vh;
11   background: #e3f9ff;
12   display: flex;
13   align-items: center;
14   justify-content: center;
15 }
16
17 .calculator{
18   background: #3a4452;
19   padding: 20px;
20   border-radius: 10px;
21 }
22 .calculator form input{
23   border: 0;
24   outline: 0;
25   width: 60px;
26   height: 60px;
27   border-radius: 10px;
28   box-shadow: -8px -8px 15px rgba(255,255, 255, 0.1),5px 5px 15px rgba(0,0, 0, 0.2);
29   background-color: transparent;
30   font-size: 20px;
31   color: #fff;
32   cursor: pointer;
33   margin: 10px;
34 }
35
```



```
35
36 form .display{
37   display: flex;
38   justify-content: flex-end;
39   margin: 20px 0;
40 }
41
42 form .display input{
43   text-align: right;
44   flex: 1;
45   font-size: 45px;
46   box-shadow: none;
47 }
48
49 form input.equal{
50   width: 145px;
51 }
52
53 form input.operator{
54   color: #33ffd8;
55 }
```

1. CSS Styling:

- The style.css file provides styling for the calculator.
- Key design features include:
 - A full-page, centered layout (display: flex; align-items: center; justify-content: center).
 - A dark background for the calculator (#3a4452) with soft shadows for a modern look.

- Buttons are styled with rounded corners, hover effects, and a neon-inspired color palette for operators.

2. JavaScript Integration:

- Inline JavaScript handles all operations through button clicks.
- The `eval()` function is used for calculating expressions, and custom onclick events manage clearing or updating the input.

Key Features:

- **User-Friendly Interface:** The calculator is responsive and provides a clean layout with readable fonts and appropriately sized buttons.
- **Basic Functionality:** Supports addition, subtraction, multiplication, division, and handling decimal points.
- **Error Handling:** The clear button and AC function allow easy correction of input errors.

Explanation:

When a button is clicked, its value is appended to the display field. Special buttons like "AC" and "Clear" perform actions like clearing or trimming input. The "=" button evaluates the input using `eval()`. The CSS enhances usability and visual appeal by aligning the calculator centrally with a sleek, shadowed design.