Design Report:

1. Introduction:

This report describes the architectural and detailed design of the Calculator System.

2. Architectural Design:

A: Presentation Layer (HTML + CSS):

Responsible for:

- Creating calculator layout.
- Displaying values.
- Button arrangement and styling.

B: Logic Layer (JavaScript):

Responsible for:

- Handling button clicks.
- Managing user input.
- Performing calculations.
- Clearing the screen.
- Error handling.

3. Subsystem Design:

3.1 Presentation Subsystem:

Contains:

- Display field

- Button grid
- Operator buttons
  The UI is arranged using a CSS Grid-based layout.

3.2 Logic Subsystem:

Contains:

- Input handling logic
- Expression execution logic
- Clear/reset logic

4. Function Design:

- **appendToDisplay(input):**
  Adds numbers/operators to the display text.
- **calculate():**
  Evaluates the expression entered using a safe try-catch structure.
  If expression is invalid, "Error" is shown.
- **clearDisplay():**
  Clears the calculator display completely.
- **eval():**
  used for expression processing because it is the most straightforward

5. Design Decisions:

- A web-based application was chosen for accessibility.
- Simple UI grid layout improves usability.
- The display is read only to prevent manual typing errors.
- A dark modern theme was selected because I like darker colors and it looks nice.

6. Structure Explanation:

- responsibilities are clearly separated:

1. HTML only structures UI
2. CSS only handles appearance
3. JavaScript only controls behavior

- **This separation improves maintainability and clarity, plus the concept of OOP was not used because I wanted clear separated sections where each language has a particular job to handle.**