**🧮 Intern Task Document: Build a Calculator using HTML, CSS, and JavaScript**

**📌 Task Title:**

**Create a Web-Based Calculator**

**🎯 Objective:**

To develop a simple and interactive calculator web application using core web technologies: HTML, CSS, and JavaScript. This task will help you understand DOM manipulation, event handling, styling, and functional logic implementation in JavaScript.

**🛠️ Tools & Technologies:**

* **HTML** – for structure
* **CSS** – for styling
* **JavaScript** – for logic and interactivity

**✅ Task Requirements:**

You need to create a calculator with the following functionalities:

**Basic UI:**

* Input/display screen (to show current input and results)
* Buttons for:
  + Digits: 0–9
  + Operators: +, −, ×, ÷
  + Decimal point .
  + Equal button =
  + Clear/Reset C

**JavaScript Functionalities:**

* Perform the 4 basic arithmetic operations
* Allow continuous calculations (e.g., 2 + 3 \* 4)
* Handle invalid input or divide-by-zero gracefully
* Clear all input when user clicks the C button

**Styling:**

* Responsive layout using **Flexbox or CSS Grid**
* Buttons should look clickable with hover effects
* The calculator should work well on both desktop and mobile devices

**💡 Bonus (Optional but Recommended):**

* Add dark/light theme toggle
* Add keyboard input support (e.g., user can type numbers/operators)
* Add animation on button click
* Add sound effects (optional)
* Add percentage (%) and square root (√) operations

### ✅ ****Student Checklist Before Submission:****

#### 🔧 Core Features

* Calculator performs addition, subtraction, multiplication, and division
* Buttons for digits (0-9) and operators are functional
* Equal button (=) shows the result
* Clear (C) button resets the calculator
* Decimal point (.) works correctly

#### 🎨 Design & Layout

* Display area for input and result
* Buttons are arranged in a grid or flex layout
* Responsive design for mobile and desktop
* Buttons have hover effects or animations

#### 📜 Code Quality

* JavaScript code is well-structured and commented
* CSS is organized (use classes, avoid inline styles)
* HTML is semantic and clean

#### 🧠 Bonus Features (Optional)

* Dark/Light theme toggle
* Keyboard input support
* Advanced operations (%, √, ^, etc.)
* Button animations or sound effects
* Calculation history

|  |
| --- |
| **NOTE:-**  **First go through simple creating Calculator after finishing this go through more enhance and updated styled and functionality** |