



Mini Project Report - 08

**Master of Computer Application – General
Semester – III**

Sub: Web Technologies

Topic: Calculator

By

Name: SANDRA B

Reg no.: 24110222500001

Faculty Name: VEERA RAGHAV K

Faculty Signature: _____

**Department of Computer Application
Alliance University
Chandapura - Anekal Main Road, Anekal
Bengaluru - 562 106**

August 2025

| Sno. | INDEX | Page No. |
|-------------|-----------------------------|-----------------|
| 1. | Introduction | 2 |
| 2. | Objective | 2 |
| 3. | Tools and Technologies Used | 2 |
| 4. | Description of Code | 3 |
| 5. | Working of the Project | 4 |
| 6. | Features | 4 |
| 7. | Output | 4 |
| 8. | Advantages | 5 |
| 9. | Future Scope | 5 |
| 10. | Conclusion | 5 |

Report on Calculator Project

1. Introduction

The project **Calculator using HTML and CSS** is a simple yet effective demonstration of how front-end technologies can be used to design a functional user interface.

A calculator is a basic tool used in day-to-day life for performing arithmetic operations such as addition, subtraction, multiplication, and division. While traditional calculators are built using hardware, this project shows how the same can be simulated digitally with the help of **web technologies**.

The calculator design focuses on simplicity, clarity, and user-friendliness. Although this version does not yet include JavaScript for performing operations, it provides the **structural (HTML)** and **styling (CSS)** parts, which are the backbone of any user interface.

2. Objectives

- To design a **basic calculator layout** using only **HTML and CSS**.
 - To practice **grid layouts in CSS** for arranging calculator buttons.
 - To understand **UI/UX principles** such as spacing, alignment, colors, and hover effects.
 - To provide a foundation that can later be extended with JavaScript for full functionality.
-

3. Tools & Technologies Used

- **HTML5** → For creating the structure of the calculator (buttons, display area, etc.).
 - **CSS3** → For styling (background colors, button shapes, hover effects, shadows, and grid layout).
 - **Web Browser** (Chrome/Edge/Firefox) → To run and test the calculator.
-

4. Description of Code

(a) HTML Structure

- `<div class="calculator">` → Main container for the calculator.
 - `<div class="display"></div>` → Placeholder screen for showing numbers.
 - `<div class="buttons">...</div>` → A grid that holds all buttons (MC, M+, ÷, X, 0–9, ., +, -, =).
 - `<button>` tags → Represent individual calculator keys.
 - Special classes:
 - `.equal` → Styles the = button differently (longer).
 - `.zero` → Makes the 0 button span two columns.
-

(b) CSS Styling

- **Body Styling**
 - Centering calculator with `display: flex; justify-content: center; align-items: center;`
 - Background color: light green (`#a4f1cb`).
 - **Calculator Box**
 - Dark background (`#2e2e2e`), padding, rounded corners, shadow.
 - Fixed width: 300px.
 - **Display Area**
 - Yellow background (`#d9d97c`).
 - Rounded rectangle look.
 - **Buttons Layout**
 - CSS Grid (`grid-template-columns: 22% 22% 22% 22%`).
 - Gap between buttons: 10px.
 - Hover effect for better interactivity.
 - **Special Buttons**
 - `.equal`: Red background (`#e74c3c`), spans **2 rows**.
 - `.zero`: Spans **2 columns**.
-

5. Working of the Project

Currently, this calculator design is **static** (only front-end).

- The **buttons** are clickable with hover effects.
 - The **layout and design** resembles a real calculator.
 - To make it functional, JavaScript can later be added for performing operations.
-

6. Features

- Clean and modern user interface.
 - Buttons aligned using **CSS Grid**.
 - Attractive hover effects.
 - Special styling for = and 0 buttons.
 - Responsive design (scales inside the fixed box).
-

7. Output

Through this project, the following concepts are learned:

- Practical use of **HTML5 tags**.
- Implementing **CSS Grid layout**.
- Importance of **color schemes and design** in UI.
- Basics of creating an **interactive calculator interface**.
- Understanding how front-end can later be extended with back-end logic (JavaScript).



8. Advantages

- Simple and easy-to-understand design.
 - Can be extended to add more features like memory operations, percentage, square root, etc.
 - Good starting project for beginners in web development.
-

9. Future Scope

- Adding **JavaScript** to perform calculations dynamically.
 - Improving **responsiveness** for different screen sizes.
 - Adding **scientific calculator functions** (sin, cos, tan, log, etc.).
 - Allowing **keyboard input** in addition to button clicks.
-

10. Conclusion

This **Calculator using HTML and CSS** project demonstrates the importance of **front-end design** in web applications. It gives a real-life application where HTML provides the structure and CSS provides the styling. Though not yet functional, it lays the foundation for building a **fully working calculator** with the addition of JavaScript.

The project has helped in understanding the **practical usage of grid systems, button design, UI aesthetics, and layout handling**. It is a great beginner-level project that can evolve into a complete application with further improvements.