UNIFIED MENTOR

YOUR SKILL, SUCCESS & JOURNEY

Name	:	Sivakumar M
UNID	:	UMIP9494
Batch	:	15/05/2024
Internship	:	Web development
Duration	:	1 Month
Project-1	:	Calculator
Project-2	:	Countdown Timer

1. BASIC CALCULATOR

INTRODUCTION

- The Basic Calculator project is a simple yet functional web application designed to perform basic arithmetic operations.
- It is developed using HTML, CSS, and JavaScript.
- The calculator supports standard operations like addition, subtraction, multiplication, and division, as well as memory functions and single input operations.

OBJECTIVES

The primary objectives of this project are:

- To create a user-friendly web-based calculator.
- To provide basic arithmetic functionalities.
- To implement memory functions for better usability.
- To ensure the application is responsive and works on various devices.

TECHNOLOGIES USED

- **HTML:** To structure the web pages.
- **CSS:** To style the web pages and ensure a responsive design.
- **JavaScript:** To implement the functionality of the calculator.

SYSTEM DESIGN

User Interface

The user interface of the calculator is designed to be intuitive and easy to use. It includes:

- A display area to show the current input and results.
- Buttons for numbers (0-9).
- Buttons for basic operations (addition, subtraction, multiplication, division).
- Memory function buttons (MC, MR, M+, M-).
- Buttons for single input operations (square root, percentage, reciprocal).
- Clear function buttons (CE, AC).

Functional Requirements

- The calculator should perform basic arithmetic operations.
- The calculator should handle memory functions.
- The calculator should provide single input operations like square root, percentage, and reciprocal.
- The calculator should support decimal numbers.
- The calculator should clear the display on demand.

Non-Functional Requirements

- The application should be responsive and work on various devices.
- The application should be user-friendly with a professional design.
- The application should have quick and accurate response times.

HOW THE BASIC CALCULATOR WORKS

User Interface:

- **Input Area:** Displays the numbers and operations entered by the user.
- **Buttons:** Includes numbers (0-9), basic operations (add, subtract, multiply, divide), memory functions (MC, MR, M+, M-), single input operations (square root, percentage, reciprocal), and clear functions (CE, AC).

Core Functionalities:

- **Basic Arithmetic Operations:** Allows addition, subtraction, multiplication, and division.
- **Memory Functions:** Enables storing, recalling, adding to, and subtracting from memory.
- **Single Input Operations**: Performs square root, percentage, and reciprocal calculations.

Implementation:

- **HTML:** Structures the calculator interface.
- **CSS:** Styles the calculator for a user-friendly experience.
- **JavaScript:** Implements the functionality for handling arithmetic operations and user interactions.

User Interaction:

- Step 1: User inputs numbers and selects operations using the buttons.
- Step 2: The display area updates to show the current input and results.
- Step 3: JavaScript functions process the inputs and perform calculations.
- Step 4: The result is displayed, and memory functions can be used as needed.

TESTING

TEST CASES:

1. **Addition Operation**:
   - Input: `2 + 3`
   - Expected Output: `5`

2. Subtraction Operation:
   - Input: `5 - 2`
   - Expected Output: `3`

3. Multiplication Operation:
   - Input: `4 * 2`
   - Expected Output: `8`

4. Division Operation:
   - Input: `8 / 4`
   - Expected Output: `2`

5. Memory Add and Recall:
   - Input: `5`, `M+`, `3`, `M+`, `MR`
   - Expected Output: `8`

6. Square Root Operation:
   - Input: `√9`
   - Expected Output: `3`

TESTING METHODOLOGY

- **Unit Testing :** Each function was tested individually to ensure it works as expected.

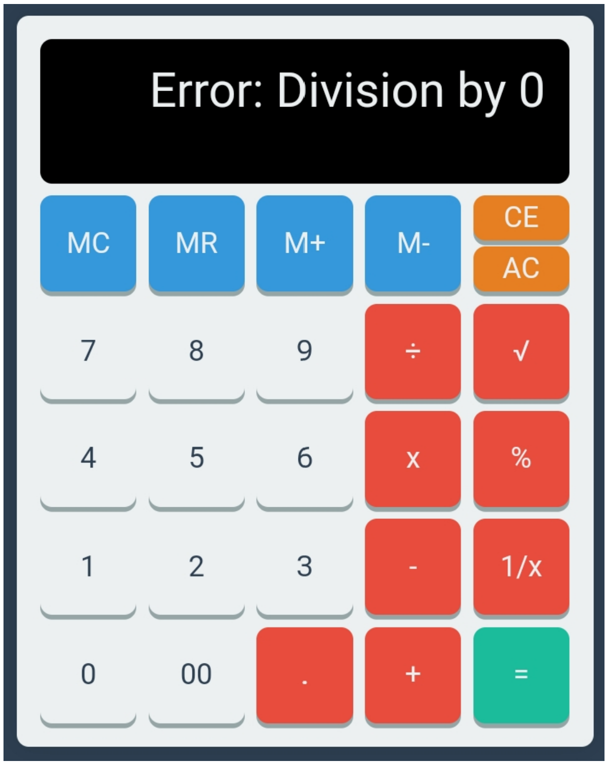
- **Integration Testing:** The overall functionality of the calculator was tested by combining all functions and ensuring they work together seamlessly.

- **User Testing**:The calculator was tested by multiple users to ensure the interface is user-friendly and the functionality meets user expectations.

RESULTS

- The Basic Calculator web application successfully performs all the specified arithmetic operations, memory functions, and single input operations.
- The user interface is intuitive, and the application is responsive across various devices.

OUTPUT



Error: Division by 0

CONCLUSION

- The Basic Calculator project demonstrates a simple yet effective use of HTML, CSS, and JavaScript to create a functional web application.
- It meets the objectives and functional requirements set out at the beginning of the project.
- The project serves as a foundation for more complex calculator applications.

FUTURE ENHANCEMENTS

- **Advanced Mathematical Functions :** Implement trigonometric functions, logarithms, and exponentiation.
- **History Feature :** Add a history feature to keep track of previous calculations.
- **Scientific Calculator Mode :** Introduce a scientific mode with more advanced features.
- **Improved Design :** Enhance the user interface with better aesthetics and animations.

REFERENCES

- [Mozilla Developer Network (MDN) Web Docs](https://developer.mozilla.org/)
- [W3Schools Online Web Tutorials](https://www.w3schools.com/)
- [Stack Overflow Community](https://stackoverflow.com/

2. COUNTDOWN TIMER

INTRODUCTION

- The Countdown Timer project is a web application designed to provide users with a visually appealing and interactive way to count down to a specific date and time.

- The application includes a preloader with a spinner animation, a user-friendly interface to input the target date and time, and animated transitions for the countdown numbers.

HOW IT WORKS

Preloader:

- When the webpage loads, a preloader with a spinner animation is displayed.
- This preloader hides once the page has fully loaded.

Setting the Target Date and Time:

- Users input the target date and time using a datetime-local input field.
- The countdown timer starts when the "Start" button is clicked.

Countdown Display:

- The countdown timer dynamically calculates the time remaining until the target date and time.
- It updates the display every second, showing the days, hours, minutes, and seconds remaining.

Animations:

Animated transitions are used to smoothly update the countdown numbers, enhancing the visual appeal.

TECHNIQUES AND METHODS USED

HTML and CSS:

- Structured the webpage using HTML elements.
- Applied CSS for styling, including flexbox for layout, animations for transitions, and responsive design principles.

JavaScript:

- Implemented event listeners for user interactions (e.g., clicking the "Start" button).
- Used setInterval to update the countdown every second.
- Calculated the remaining time by comparing the current time to the target date and time.
- Applied conditional logic to handle edge cases (e.g., invalid date input).

Animation:

- CSS animations (@keyframes) for the spinner in the preloader.
- JavaScript animations for smooth number transitions in the countdown timer.

USAGE

User Interaction:

- Open the webpage and wait for the preloader to disappear.
- Enter the target date and time using the provided input field.
- Click the "Start" button to begin the countdown.
- Observe the countdown display updating every second.
- Listen for the alarm sound when the countdown reaches zero.
- Practical Applications:

Event reminders:

Users can set countdowns for events such as meetings, deadlines, or personal milestones.

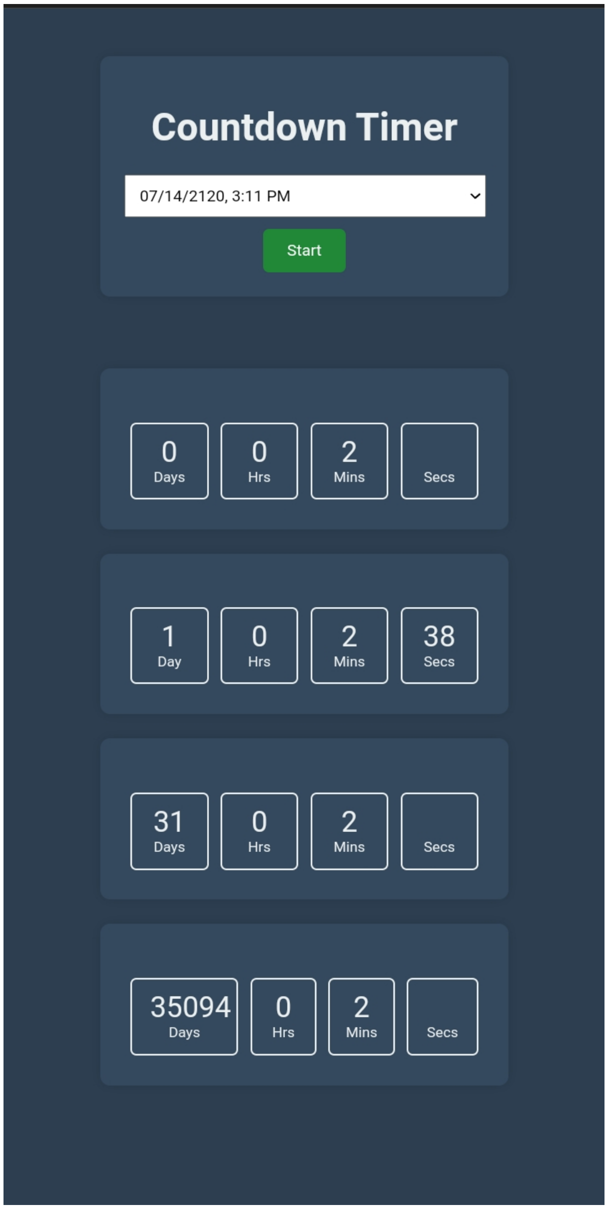
Time management:

Helps users track time remaining for tasks or activities.

User engagement:

Enhances user experience on websites by adding interactive elements.

OUTPUT



Countdown Timer

07/14/2120, 3:11 PM

0 Days | 0 Hrs | 2 Mins | Secs

1 Day | 0 Hrs | 2 Mins | 38 Secs

31 Days | 0 Hrs | 2 Mins | Secs

35094 Days | 0 Hrs | 2 Mins | Secs

CONCLUSION

- The Countdown Timer project successfully combines a preloader, a user-friendly input interface, animated transitions, and an alarm notification to create an engaging and functional web application.
- The techniques and methods used ensure smooth performance and visual appeal, making it a useful tool for various practical applications.
- Future enhancements could include additional customization options, multiple countdown timers, and more interactive features to further improve the user experience.

REFERENCES

- **HTML Documentation:** MDN Web Docs - HTML
- **CSS Documentation:** MDN Web Docs - CSS
- **JavaScript Documentation**: MDN Web Docs - JavaScript
- **W3Schools:** HTML, CSS, and JavaScript Tutorials