Name: Mayank Srivastava

Branch: CSE

Section: V B

Roll No: 2200290100102

Project Report: Simple Math Calculator

Project Overview

The Simple Math Calculator is a web-based application designed to perform basic arithmetic operations, including addition, subtraction, multiplication, and division. The calculator features a clean and modern interface, making it user-friendly for anyone needing quick calculations.

Technologies Used

- **HTML**: To structure the web page and create the layout of the calculator.

- **CSS**: For styling the calculator, ensuring it is visually appealing and responsive.
- **JavaScript**: To implement the calculator's functionality, enabling dynamic interactions and calculations.

Features

1. **Basic Operations**: The calculator supports addition, subtraction, multiplication, and division.
2. **Clear and Erase Functions**: Users can clear the entire input or erase the last character.
3. **Responsive Design**: The calculator adjusts to different screen sizes for accessibility on various devices.
4. **User-Friendly Interface**: The calculator is designed with intuitive

buttons and a display that shows the current input.

Implementation Details

HTML Structure

The HTML file (index.html) contains the following key elements:

- A display area for showing the current input and results.
- Buttons for numbers (0-9), arithmetic operations (+, -, *, /), and functional buttons (C for clear, ← for erase, and = for calculating the result).

CSS Styling

The CSS file (style.css) provides the following styles:

- A gradient background for visual appeal.

- A central calculator design with rounded corners and shadows.
- Responsive button styles that change color on hover, enhancing user experience.

JavaScript Functionality

The JavaScript file (script.js) handles the calculator's logic:

- **Event Listeners**: Each button is equipped with an event listener that triggers actions when clicked.
- **Input Handling**: The calculator displays input dynamically as buttons are pressed.
- **Calculation Logic**: The eval() function evaluates the expression entered by the

user. Proper error handling is included to alert users of invalid inputs.

- **State Management**: A boolean flag (isResultDisplayed) is used to manage the state of the calculator display, resetting the input after displaying a result.

Challenges Faced

- **Input Validation**: Initially, managing invalid inputs was challenging. Using try...catch blocks helped to handle errors effectively.
- **UI Responsiveness**: Ensuring the calculator looked good on various devices required careful attention to CSS styling.

Future Improvements

- **Advanced Functions**: Add more complex mathematical functions, such as square roots and exponents.
- **History Feature**: Implement a feature to keep track of previous calculations for user reference.
- **Theming Options**: Provide users with options to change the calculator's theme for better personalization.

Conclusion

The Simple Math Calculator project successfully demonstrates core web development skills, including HTML structure, CSS styling, and JavaScript functionality. It provides users with a practical tool for performing basic calculations while offering a pleasant user experience.

Project Report: To-Do-List

Project Overview

The "To-Do-List" application is a simple web-based task manager that allows users to add and remove daily tasks. This tool helps users stay organized and track their accomplishments, making it easier to manage daily activities.

Technologies Used

- **HTML**: For structuring the layout and elements of the web application.
- **CSS**: For styling the application and enhancing the user interface.
- **JavaScript**: For implementing functionality, including task management, user interactions, and dynamic updates to the task list.

Features

1. **Add Tasks**: Users can input tasks they need to accomplish and add them to a list.

2. **Remove Tasks**: Users can remove tasks from the list by clicking on them.
3. **Responsive Design**: The application is designed to be responsive and accessible on various devices.
4. **User-Friendly Interface**: The interface is intuitive, making it easy for users to navigate and manage tasks.

Implementation Details

HTML Structure

The HTML file (index.html) consists of:

- A container for the entire application.
- A heading with an icon and title.
- A form for inputting tasks.
- An unordered list to display tasks.

CSS Styling

The CSS file (styles.css) includes styles for:

- A visually appealing background and layout.

- Responsive design adjustments for different screen sizes.
- Form elements and buttons with hover effects for improved user experience.

JavaScript Functionality

The JavaScript file (script.js) handles the application's logic:

- **Event Listeners**: The form submission and task removal events are managed using event listeners.
- **Task Management**: Tasks are stored in an array (taskArray) and dynamically added to or removed from the DOM.
- **Unique Task IDs**: Each task is assigned a unique ID based on the current timestamp to differentiate them.

Challenges Faced

- **Dynamic Task Management**: Ensuring tasks could be added and removed seamlessly without page refreshes was a key challenge that was overcome by using JavaScript to manipulate the DOM.

- **User Experience**: Creating an intuitive user interface required careful consideration of design and responsiveness.

Future Improvements

- **Local Storage**: Implementing local storage to save tasks even after refreshing the page.
- **Edit Tasks**: Allowing users to edit existing tasks for better task management.
- **Priority Levels**: Adding functionality to categorize tasks by priority (e.g., high, medium, low).
- **Due Dates**: Incorporating due dates for tasks to enhance time management.

Conclusion

The "To-Do-List" application serves as a practical tool for managing daily responsibilities. It demonstrates essential web development skills, including structuring HTML, styling with CSS, and dynamic interactions using JavaScript. The application is a solid foundation that can be further developed with additional features to enhance functionality.