**Project Report: Basic Calculator**

* **Overview**

This project focuses on developing a functional calculator application using web technologies. The calculator allows users to perform essential arithmetic operations, thus refining their front-end development skills while demonstrating their understanding of JavaScript logic and design principles.

* **Features**

**User Interface**

* A visually appealing and intuitive interface designed using HTML and CSS.
* Includes:
  + Buttons for digits (0-9), decimal point, and operators (+, -, \*, /).
  + Special buttons for "Clear" and "Equals."
  + A display panel for showing input and results.

**Functionality**

* Implemented using JavaScript:
  + Supports basic arithmetic operations: addition, subtraction, multiplication, and division.
  + Ensures operations follow the **BODMAS/BIDMAS** rules for correct order.
  + Accepts both button clicks and keyboard inputs.

**Error Handling**

* Proper error handling included:
  + Division by zero gracefully displays an "Error" message.
  + Handles invalid input scenarios and ensures accurate responses.

**Optional Enhancements**

This project can be extended by:

* Adding square root calculations.
* Implementing percentage functionality.
* Introducing memory functions (M+, M-, MR, MC).
* **Code Structure**

**HTML**

* Provides the structure for the calculator:
  + Divides the calculator into display and button sections.
  + Uses grid layout to organize buttons.

**CSS**

* Styles the calculator:
  + Utilizes a dark theme for a professional appearance.
  + Includes hover effects and smooth transitions for enhanced user experience.

**JavaScript**

* Controls the functionality of the calculator:
  + **appendToDisplay(value)**: Adds numbers and operators to the display, while preventing invalid inputs (e.g., multiple decimals).
  + **clearDisplay()**: Resets the display and clears calculations.
  + **calculate()**: Evaluates input using eval() safely, handling errors like division by zero.

**Keyboard Interaction**

* Enables calculator use via keyboard inputs:
  + Supports digits, operators, Enter, Escape, and Backspace keys for efficient input handling.
* **Testing**

**Functional Testing**

* Validate basic arithmetic operations to ensure accurate results.
* Test buttons and keyboard inputs for responsiveness.

**Error Testing**

* Confirm division by zero is handled properly.
* Verify the system displays "Error" for unsupported operations.

**Submission Guidelines**

1. **Repository Contents**:
   * index.html: HTML structure for the calculator.
   * styles.css: CSS styling for layout and design.
   * script.js: JavaScript logic for functionality.
   * README.md: Usage instructions and description of additional features.
2. **Code Comments**:
   * Clear and concise explanations for code sections.
3. **README File**:
   * Include:
     + Project description.
     + Instructions on calculator usage.
     + Information about optional enhancements (if implemented).

**Assessment Criteria**

1. **Functionality**:
   * Performs arithmetic operations correctly.
   * Follows BODMAS/BIDMAS rules.
2. **User Interface**:
   * Visually appealing and user-friendly layout.
3. **Code Quality**:
   * Clean, structured, and adheres to best practices.
4. **Error Handling**:
   * Displays appropriate responses for invalid inputs.
5. **Optional Features**:
   * Additional functionalities implemented successfully.

OUTPUT:

