**User Requirements:**

1. **Core Functionality**:
   * **Basic Operations**: The calculator must be able to perform addition, subtraction, multiplication, and division.
   * **Input Method**: Users should be able to input numbers and operations via an on-screen interface.
   * **Execution**: The calculation must execute upon user command (e.g., pressing an "=" button).
2. **User Interface**:
   * **Display Area**: There must be a display area where the numbers and results are shown.
   * **Buttons**:
     + Numeric buttons for digits 0-9.
     + Buttons for each arithmetic operation (+, -, \*, /).
     + An "=" button to compute operations.
     + A "Clear" button (C) to reset the display.
   * **Layout**: Buttons should be arranged in a grid layout similar to traditional calculators.
3. **Performance Requirements**:
   * **Speed**: Calculations should complete instantaneously from the user’s perspective (no noticeable delay).
   * **Precision**: The calculator should handle operations on integers and floating-point numbers up to two decimal places.
4. **Usability Requirements**:
   * **Accessibility**: Font sizes and button sizes should be large enough to be easily readable and clickable.
   * **Responsive Design**: The calculator should be usable on both desktop and mobile browsers, adjusting layout as necessary.
5. **Technical Constraints**:
   * **Technologies Used**: HTML, CSS for styling, and JavaScript for functionality.
   * **Browser Compatibility**: Must be compatible with the latest versions of Chrome, Firefox, and Edge.
6. **Stretch Goals (Optional)**:
   * **Advanced Mathematical Operations**: Include functions for calculating percentages, square roots, or powers.
   * **Memory Functions**: Add buttons to allow users to store and recall numbers.
   * **Error Handling**: Display user-friendly error messages for invalid operations (e.g., division by zero).