**Objective**

As part of a simulated role at Google, I was tasked with building an automated end-to-end Cypress test suite for the online calculator that appears when you search for "calculator" on Google.

Since CAPTCHA andbot detection protect the live Google calculator, I replicated the calculator UI and functionality using a public-facing alternative: [CalculatorSoup's Basic Calculator](https://www.calculatorsoup.com/calculators/math/basic.php" \t "_new). (https://www.calculatorsoup.com/calculators/math/basic.php)

**Assumptions**

* I assume we are testing an internal calculator component at Google, and are allowed to simulate or mock the experience as needed.
* Only the following buttons are required: 0–9, ., =, +, -, ×, ÷, AC, and CE.
* No scientific or extended operations (π, %, √, etc.) were required unless explicitly mentioned.
* User interactions are expected via button clicks rather than keyboard input unless stated.

**Rationale for Technology & Approach**

* **Cypress** was chosen for its reliability in handling browser-based UI automation.
* **Page Object Model (POM)** was implemented to keep the test code clean, maintainable, and scalable.
* A real, publicly available calculator (CalculatorSoup) was used to simulate real-user conditions in a browser.
* I avoided directly testing Google.com due to anti-automation limitations (e.g., CAPTCHA, fingerprinting).

**Steps Taken**

**1. Calculator Page Object Setup**

* Created a CalculatorPage.js class that abstracts all selectors and common user actions.
* The class includes utilities for chaining operations, verifying output, taking screenshots, and checking accessibility.

**2. E2E Test Suite Creation**

Test files are grouped logically to cover:

* **Display and Button Functionality** (numbers, decimal)
* **Basic Arithmetic** (+, -, ×, ÷)
* **Chaining Operations** (e.g., 1 + 2 + 3)
* **Edge Cases** (e.g., divide by zero, handle multiple decimal points, ignore leading zeros)
* **Long input cases and calculations** (e.g., scientific notation result)
* **Clear Functions** (AC, CE)
* **Equal Function** (Pressing equals multiple times)
* **Keyboard Input**
* **Accessibility Tests** (focusable buttons, screen reader labels)
* **Visual Regression** (before and after calculator state)

**3. Test Execution**

* Tests are executed locally using Cypress CLI (npx cypress open) and optionally integrated into CI via GitHub Actions.