Task:

- **Imagine you are testing an ordinary calculator. It is a web application that can perform 4 arithmetic operations. The UI has buttons with numbers, arithmetic actions, an expression input field and a result output field. There are also "equals" and "erase" buttons.**
- **You have no requirements for the application. The task is to cover the calculator with all possible tests.**

The output is a checklist with checks.

Checklist	
if we have access to the database and API, check the work of the calculator from these sides too. Check data types and boundary values in DB and API.	
Usability Test	Verify that all the buttons are present, work and text written on them is readable. Check whether the calculator contains 9 numeric digits.
Functional Test	Basic arithmetic operations:
	Addition: Test adding positive, negative, and decimal numbers.
	Subtraction: Test subtracting positive, negative, and decimal numbers.
	Multiplication: Test multiplying positive, negative, and decimal numbers.
	Division: Test dividing positive, negative, and decimal numbers, including division by zero.
Pairwise testing	check all different combinations of positive, negative, and decimal numbers for all different operations.
Order of operations:	Test whether the calculator follows the correct order of operations
	Verify that parentheses are handled correctly to override the default order.
Input validation:	Check that long results or results exceeding the display width are handled appropriately
	Test entering numbers via buttons and keyboard input.
	Verify that only valid characters and operations are accepted as input.
	Test entering numbers with decimals, negative numbers, and large numbers.
	Test whether it is possible to input wrong data from a buffer.
Equals and erase functionality:	Verify that pressing the "equals" button computes the correct result for the given expression.
	Test the "erase" button functionality to clear the input field or remove the last character entered.
Error handling:	Test division by zero and verify that the calculator displays an appropriate error message.
	Check how the calculator handles invalid input or incomplete expressions.
	Check whether the calculator doesn't allow characters.
Display output:	Verify that the result is displayed correctly in the output field.
	Test for proper formatting of results (decimal places).
	Check that long results or results exceeding the display width are handled appropriately
Cross-device and Cross-browser testing	Verify that the calculator interface is responsive and works well on various devices (desktops, tablets, mobile phones).
	Test the calculator's functionality across different web browsers
Performance testing:	Check for memory leaks or excessive resource usage during prolonged use.
	Test the calculator's performance with a large number of calculations to ensure it remains responsive.
Boundary Values and equivalence classes:	Check whether the limit for minvalue and maxvalue of the response value is.
	Test the calculator with a maxvalue, maxvalue -1 or minvalue, minvalue+1
	Test the calculator with a number between maxvalue and minvalue
	Check how the calculator handles edge cases like extremely large expressions or complex nested operations.
Localization and internationalization:	Whether it is possible to use a calculator with a different language to ensure all text is properly translated . (For example, for messages about mistakes.)
Accessibility testing:	Check that the calculator is accessible to users with disabilities
Load testing:	multiple requests to the calculator simultaneously