PRODIGY – SOFTWARE TESTING INTERNSHIP

TASK 1 – Test Cases For Calculator

Done By : Intern Lalitha B, Prodigy

(01/01/2026 – 31/01/2026)

1. Introduction :

This assignment is to test a simple web-based calculator application manually by designing and executing the test cases. The calculator application lets the users perform basic arithmetic operations such as addition, subtraction, multiplication, and division.

Software testing is a critical activity that ascertains an application behaves predictively and yields results accurately. It ensures that through this task, systematic test cases are written and all valid and invalid user inputs are checked, making the calculator reliable, accurate, and user-friendly.

2. Objective of Testing :

The objectives of testing the calculator application are:

- To verify the correctness of calculator operations
- To validate proper handling of user inputs
- To ensure appropriate error handling for invalid scenarios such as division by zero

3. Application Under Test (AUT) :

Attribute	Description
Application Name	Simple Calculator
URL	https://dunizb.github.io/sCalc/
Type	Web-based calculator
Supported Operations	Addition, Subtraction, Multiplication, Division

4. Test Environment :

Parameter	Details
OS	Windows / macOS
Browser	Google Chrome

Parameter	Details
Internet	Required
Testing Type	Manual Testing

5. Scope of Testing :

In-Scope

- Addition
- Subtraction
- Multiplication
- Division
- Decimal values
- BODMAS rule

Out-of-Scope

- Scientific functions
- Memory functions
- Performance testing

6. Assumptions & Constraints :

Assumptions

- Calculator follows BODMAS rule
- Calculator accepts valid numeric inputs
- User interacts using calculator buttons

Constraints

- Active internet connection is required
- Application behaviour may vary across browsers

7. Test Strategy :

The testing strategy adopted for this task includes:

- **Manual Testing Approach:** All test cases are executed manually
- **Black-Box Testing:** Testing is done without knowledge of internal code
- **Equivalence Class Testing:** Valid and invalid input classes are tested

- **Boundary Value Testing:** Edge cases such as zero and decimal values are tested

8. Test Scenarios :

- Verify arithmetic operations
- Verify decimal handling
- Verify invalid inputs
- Verify division by zero

9. Defect Report :

No defects were identified during the testing of the calculator application.

10. Tools Used :

- GitHub
- Web Browser (Google Chrome)
- Markdown

11. Conclusion :

Manual testing was successfully performed on the Simple Calculator application. All core arithmetic functionalities were verified and found to be working correctly. The application handles valid inputs efficiently and manages error scenarios such as division by zero effectively. Overall, the calculator application meets the expected functional requirements and is reliable for basic arithmetic operations.

12. Test Cases :

Test Case ID	Test Description	Test Steps	Expected Result
TC_ADD_01	Addition of two positive integers	5 + 3 =	Result is **8**
TC_ADD_02	Addition with a negative number	-5 + 10 =	Result is **5**
TC_ADD_03	Addition of decimal numbers	2.5 + 1.2 =	Result is **3.7**
TC_SUB_01	Subtraction of two numbers	10 − 4 =	Result is **6**
TC_SUB_02	Subtracting a negative number	5 − (-3) =	Result is **8**
TC_MUL_01	Multiplication of two numbers	6 × 4 =	Result is **24**

Test Case ID	Test Description	Test Steps	Expected Result
TC_MUL_02	Multiplication with zero	9 × 0 =	Result is **0**
TC_MUL_03	Multiplication of decimals	2.5 × 2 =	Result is **5**
TC_DIV_01	Division of two numbers	20 ÷ 4 =	Result is **5**
TC_DIV_02	Division resulting in decimal	5 ÷ 2 =	Result is **2.5**
TC_DIV_03	Division by zero	10 ÷ 0 =	Error message displayed
TC_BOD_01	Operator precedence (BODMAS)	2 + 3 × 4 =	Result is **14**
TC_BOD_02	Bracket precedence	(2 + 3) × 4 =	Result is **20**
TC_INV_01	Alphabet input	a =	Invalid input error
TC_INV_02	Special character input	@ =	Invalid input error
TC_INV_03	Empty input	Press =	No input error
TC_NEG_01	Two negative numbers	-4 + -6 =	Result is **-10**
TC_DEC_01	Complex decimal calculation	1.5 + 2.5 × 2 =	Result is **6.5**