**Practical :** Basic Math Operation Functionality and Usability Testing

White Box Testing

* Lab 01 DeskTop Math Cal Testing using Unittest, pytest
* Lab 02 DeskTop Math Cal Testing Using Robot

Grey Box Testing

* Lab 03 Local Web Math Cal App Tesing using Selenium
* Server.py
* test\_selenium\_operations.py

Black Box Testing

* Lab 04 Local Web Math Cal App using Robot Framework
* Server.py
* test\_robot\_web\_operations.robot

User stories capture various aspects of the user experience and functionality expected from the math application. They can guide the development process by ensuring that user needs are prioritized.

**As a [type of user], I want [some goal] so that [some reason].**

Here are some user stories related to the server.py application that performs basic mathematical operations via a web interface:

**User Stories**

Basic Addition

* As a user of the math application,
* I want to add two numbers through a web interface,
* So that I can quickly calculate the sum without using a calculator.

Basic Subtraction

* As a user of the math application,
* I want to subtract one number from another through a web interface,
* So that I can easily determine the difference between the two numbers.

Basic Multiplication

* As a user of the math application,
* I want to multiply two numbers using a web interface,
* So that I can perform multiplication calculations quickly and efficiently.

Basic Division

* As a user of the math application,
* I want to divide one number by another through a web interface,
* So that I can find the quotient without manual calculations.
* Error Handling for Division by Zero

As a user of the math application,

* I want to receive an error message when I attempt to divide by zero,
* So that I am informed of the invalid operation and can correct my input.

User-Friendly Interface

* As a user of the math application,
* I want a simple and intuitive web interface to perform calculations,
* So that I can use the application without confusion or difficulty.

View Results Clearly

* As a user of the math application,
* I want to see the result of my calculations displayed clearly on the screen,
* So that I can quickly understand the outcome of my operations.
* Cross-Browser Compatibility
* As a user of the math application,
* I want the application to work seamlessly in multiple web browsers,
* So that I can access it from any device without compatibility issues.

Responsive Design

* As a user of the math application,
* I want the web interface to be responsive and work well on mobile devices,
* So that I can perform calculations on the go using my smartphone or tablet.

Access via URL

* As a user of the math application,
* I want to access the application by entering a URL in my web browser,
* So that I can quickly get to the tool without navigating through multiple pages.

For testing the above user stories, we generate test cases which contain:

* **Test Number:** A unique identifier for each test case.
* **Test Steps**: Describes the actions to be performed in the test.
* **Inputs**: Specifies the input values for the operation.
* **Expected Output**: The expected result after performing the operation.

Here’s a structured representation of the test cases for the math operations application in a tabular format. This includes the test number, steps, inputs, and expected outputs.

**Test Cases Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Inputs** | **Expected Output** |
| TC01 | Open the math application | N/A | Application loads successfully |
| TC02 | Input numbers and click Add | x: 5, y: 3 | Result: 8 |
| TC03 | Input numbers and click Subtract | x: 10, y: 4 | Result: 6 |
| TC04 | Input numbers and click Multiply | x: 4, y: 5 | Result: 20 |
| TC05 | Input numbers and click Divide | x: 12, y: 4 | Result: 3 |
| TC06 | Input numbers and click Divide | x: 5, y: 0 | Error: Division by zero is undefined |
| TC07 | Input numbers and click Add | x: 0, y: 0 | Result: 0 |
| TC08 | Input numbers and click Subtract | x: -5, y: -3 | Result: -2 |
| TC09 | Input numbers and click Multiply | x: -2, y: 3 | Result: -6 |
| TC10 | Input numbers and click Divide | x: -10, y: -2 | Result: 5 |