

Table of Contents

Introduction:	1
Test Strategy:.....	1
1. Server-Side (Backend):	1
1.1. Objectives:.....	1
Functionality: Ensure that all server-side functionalities work as intended for each endpoint.	1
Data Integrity: Make sure the returned data is consistent and accurate.	1
Error Handling: Ensure appropriate errors are returned for invalid inputs or scenarios.....	1
1.2. Test Types:	1
1.2.1. API Functionality Testing:	1
Ensures endpoints operate correctly and return expected results.....	1
1.2.2. Boundary and Edge Case Testing:	1
2. Client-Side (Frontend):	2
3. Test Environment:	3
4. Test Schedule:	3
5. Risks and Mitigations:	3
REFERENCES	3

Introduction:

This test plan aims to ensure the quality of both the client-side and server-side interactions for the provided endpoints. It focuses on functionality, usability, and error-handling for the specific behaviors defined by these endpoints.

Test Strategy:

1. Server-Side (Backend):

1.1. Objectives:

Functionality: Ensure that all server-side functionalities work as intended for each endpoint.

Data Integrity: Make sure the returned data is consistent and accurate.

Error Handling: Ensure appropriate errors are returned for invalid inputs or scenarios.

1.2. Test Types:

1.2.1. API Functionality Testing:

Ensures endpoints operate correctly and return expected results.

- For endpoint1:
 - Test that the correct list of numbers is returned.
- For endpoint2:
 - Test with positive, negative, and zero numbers to validate squared results.
- For endpoint3:
 - Test with even numbers to validate error handling.
 - Test with odd numbers to validate correct division results.

1.2.2. Boundary and Edge Case Testing:

Endpoint behavior on the edge or limit might differ. This uncovers potential oversights in the code.

- For endpoint2:
 - Test with very large and very small numbers to ensure the squared value doesn't overflow or cause issues.
- For endpoint3:
 - Test with the smallest and largest possible odd and even integers.

1.1.1 Error Handling and Negative Testing:

Ensures the server responds gracefully to unexpected inputs or conditions and provides meaningful error messages.

- For endpoint2 and endpoint3:
 - Test with non-integer values.
 - Test with strings or other invalid inputs.

1.1.2 Test case definitions and details

[test\APITests\features\steps\test_app.feature](#)

2. Client-Side (Frontend):

2.1 Objectives:

User Experience: Ensure that the UI communicates with the backend correctly and displays results or errors clearly.

Input Validation: Ensure that the client-side validates and sanitizes inputs before sending them to the server.

2.2 Test Types:

Functional Testing:

Validates the frontend's capability to interact with the backend correctly and display the results.

For all endpoints:

- Ensure the returned data from the server is correctly displayed on the UI.
- Ensure error messages from the server are displayed appropriately on the UI.

Usability Testing:

- Ensures the interface is intuitive and user-friendly.

For all endpoints:

- Ensure it's clear to the user how to input data (for endpoint2 and endpoint3).
- Ensure results are presented in an understandable manner.

Input Validation Testing:

- Helps prevent invalid or malicious data from being sent to the server.

For endpoint2 and endpoint3:

- Ensure client-side validation is done for number inputs.
- Ensure meaningful feedback is provided for invalid inputs.

2.3 Test case definitions and details

[test\APITests\features\steps\test_app.feature](#)

3. Test Environment:

Detail out tools and platforms necessary for testing. This could include:

- Postman or similar tool for backend testing.
- Different browsers (Chrome, Firefox, Safari) for front-end testing.
- Any database setups if necessary.

4. Test Schedule:

Outline the sequence in which the tests will be executed, along with estimated timeframes.

5. Risks and Mitigations:

List potential challenges or risks during the testing phase and suggest possible solutions or preventive actions.

REFERENCES

Ref. #	Version	Date	Document Name
R1	1.0	9/4/2023	FrontEnd UI test_app.feature file
R2	1.0	9/4/2023	BackEndServerAPI test_app.feature file