TEST PLAN

1. Objective

This section defines the purpose of the testing.

- **Objective:** The goal is to test the functionality of the API endpoint `https://ec2-15-207-3-97.ap-south-1.compute.amazonaws.com:3000/second-largest` to ensure it correctly returns the second-largest number from an array provided in the request body.

2. Scope

Define what will be tested and what will not be covered.

- **In Scope:**
  - Validation of the API's response for correct input.
  - Error handling for invalid inputs, missing or incorrect authorization tokens.
  - Testing various input scenarios like positive numbers, negative numbers, duplicate numbers, and arrays with insufficient elements.
- **Out of Scope:**
  - Performance testing, load testing, and any testing not related to functional validation.

3. Test Scenarios

Outline the different scenarios that will be tested.

Scenario ID	Test Scenario	Description
TS1	Valid Array Input	Test the API with a valid array input (e.g., `[1, 2, 3, 4, 5]`) and verify that it returns the second-largest number (e.g., `4`).
TS2	Missing Authorization Token	Send a request without the authorization token and verify that the API returns an unauthorized access error.
TS3	Wrong Authorization Token	Send a request wrong authorization token and verify that the API returns an unauthorized access error.
TS4	Empty Array Input	Send a request with an empty array (e.g., `[]`) and verify that the API returns an appropriate error message.
TS5	Single Element Array	Send a request with an array containing only one element (e.g., `[10]`) and verify that the API returns an appropriate error message.
TS6	Non-Array Input	Send a request with a non-array input (e.g., `"arr": "string"`) and verify that the API returns an appropriate error message.
TS7	Array with Duplicates	Test the API with an array containing duplicate numbers (e.g., `[2, 3, 3, 5, 5]`) and verify that it correctly identifies the second-largest value.
TS8	Array with Negative Numbers	Test the API with an array containing negative numbers (e.g., `[-1, -2, -3, -4, -5]`) and verify the correct second-largest value is returned.

TS9	Array With Zero Numbers Only	Test the API with an array containing negative numbers (e.g., `[0, 0, 0, 0, 0]`) and verify the correct second-largest value is returned.
TS10	Array with decimal numbers only	Test the API with an array containing negative numbers (e.g., `[1.1, 1.2, 1.3, 1.4, 1.5]`) and verify the correct second-largest value is returned.
TS11	Array with decimal and integer numbers	Test the API with an array containing negative numbers (e.g., `[1, 2, 3, 1.4, 1.5]`) and verify the correct second-largest value is returned.

4. Test Data

List the data that will be used for testing, including both valid and invalid data.

Test Data ID	Test Data	Purpose
TD1	`[1, 2, 3, 4, 5]`	Valid input
TD2	`[10]`	Insufficient elements
TD3	`[]`	Empty array
TD4	`{"arr": "string"}`	Non-array input
TD5	`[2, 3, 3, 5, 5]`	Duplicate numbers
TD6	`[-1, -2, -3, -4, -5]`	Negative numbers
TD7	`[0,0,0,0,0]`	Zero Numbers
TD8	`[1.1, 1.2, 1.3, 1.4, 1.5]`	Decimal Numbers
TD9	`[1, 2, 3, 1.4, 1.5]`	Decimal And Integer numbers

5. Tools

Mention the tools that will be used.

- **Tools:**
  - Postman for sending and validating API requests.
  - Excel for documenting test cases and results.

6. Expected Outcomes

For each scenario, define what the expected result should be.

- **Example:**
  - For input `[1, 2, 3, 4, 5]`, the expected result is `4`.
  - For input `[-1, -2, -3, -4, -5]`, the expected result is `-2`.
  -

7. Execution Plan

Define how you plan to execute the tests.

- **Execution Steps:**
    1. Configure Postman with the API endpoint and authorization token.
    2. Execute each test scenario by sending requests with the respective test data.
    3. Document the actual results and compare them with the expected results.