Test Plan for API: Finding the Second Largest Number in an Array:

Created By: **Prafulla Mishra(QA)**

1. Objective:

The purpose of this test plan is to validate the functionality of the API endpoint **POST /second-largest-number**-in-array, which is designed to find the second largest number in an array of integers. The test plan consist of approach, strategy, test scenarios, and criteria required to ensure the correct behavior of the API.

2. Scope:

- API Endpoint: POST http://13.202.76.215:3000/second-largest
- Input: JSON object containing an array of integers: {"arr": [1,2,3,4,5]}
- Functionality: The API returns the second largest number from the given array.
- Authentication: Requires a valid bearer token for authorization.

3. Testing Approach:

The testing approach will cover the following types of testing:

- **Functional Testing**: Verify the API returns the correct second largest number.
- **Boundary Testing**: Test with minimum, maximum, and edge values.
- **Negative Testing**: Test how the API handles invalid inputs and missing/incorrect data.
- **Performance Testing**: Validate the performance with large datasets.
- **Security Testing**: Ensure the API can only be accessed with valid authentication tokens.
- **Error Handling**: Verify the appropriate error messages are returned for invalid inputs.

4. Test Environment:

- **Base URL**: http://13.202.76.215:3000/second-largest
- **Authorization**: Bearer token (valid and invalid)
- **Tools**: Postman
- **Data**: JSON request body with an array of integers

5. Roles and Responsibilities:

- **Tester**: Responsible for test case design, execution, and reporting.

6. Entry Criteria:

- The API must be developed and deployed to the test environment.
- Documentation specifying API functionality and authentication must be available.
- Valid bearer tokens must be provided for testing.

7. Exit Criteria:

- All test cases must be executed.
- All high-priority defects must be resolved.
- Test reports must be submitted and reviewed.

8. Test Scenarios: Written In Excel Sheet

Functional Testing:

- Valid Input: Send an array of integers [1, 2, 3, 4, 5]. Expected output: second largest number 4.
- Array with Duplicate Numbers: [5, 5, 3, 2, 1]. Expected output: 3.
- Array with Negative Numbers: [-10, -20, -5, -15]. Expected output: -10.
- Array with Two Elements: [7, 3]. Expected output: 3.

Boundary Testing:

- **Single Element Array**: [10]. Expected output: Error message: "Array must contain at least two elements".
- **Empty Array**: []. Expected output: Error message: "Array cannot be empty".

Negative Testing:

- **Non-integer Values**: ["a", 3, 5]. Expected output: Error message: "Array must contain only integers".
- **All Elements Same**: [5, 5, 5]. Expected output: Error message: "Array must contain two unique numbers".
- **Invalid JSON Format**: Send a request with missing brackets. Expected output: Error message: "Invalid JSON format".

Performance Testing:

- **Large Array**: Test the API's response time and stability with an array of 100 elements.

Security Testing:

- **Invalid Bearer Token**: Send a valid request with an invalid bearer token. Expected output: 401 Unauthorized.

- **No Bearer Token**: Send a valid request without a token. Expected output: 401 Unauthorized.

9. **Test Data:** Written with Test Scenarios in Excel Sheet

10. Deliverables:

- Test cases in Excel format.
- Execution report with pass/fail status.
- Bug report for failed test cases.

11. Risks and Mitigations:

- **Risk**: API server unavailability during testing.
- **Mitigation**: Coordinate with the development team for uptime.
- **Risk**: High response time with large datasets.
- **Mitigation**: Conduct performance optimization after identifying bottlenecks.

12. Timeline:

- Test case preparation: 2 days
- Test execution: 3 days
- Bug fixing and retesting: 2 days
- Final reporting: 1 day

13. Conclusion:

This test plan provides an approach to ensure complete coverage of all functional and non-functional aspects of the API. Successful execution of this plan will ensure the API correctly identifies the second largest number in an array while handling errors and edge cases effectively.