API Test Strategy Document

# 1. Introduction

The purpose of this document is to outline the test strategy for the API available at Swagger API Documentation (https://fakerestapi.azurewebsites.net/index.html). The test plan ensures the API meets its functional and non-functional requirements and works as expected under various conditions.

# 2. Scope

In Scope:   
- Functional testing of all endpoints.  
- Integration testing between endpoints.  
- End-to-End validation of critical flows.  
- Performance testing for response times and load handling.  
  
Out of Scope:   
- UI-related testing (if no UI is directly linked).

# 3. Testing Approach

The following levels of testing will be conducted:  
1. Unit Testing: Validation of individual endpoints.  
2. Integration Testing: Verification of interactions between API endpoints.  
3. System Testing: Testing the complete API as an independent system.  
4. End-to-End Testing: Validation of workflows involving multiple endpoints.  
5. Performance Testing: Assess response times and system stability under load.

# 4. Priority Levels

1. High Priority: Critical functionalities and positive test cases ensuring basic API operability.  
2. Medium Priority: Negative test cases for robustness and error handling.  
3. Low Priority: Performance and scalability tests under extreme conditions.

# 5. Test Cases

## 5.1 Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Test Case | Priority | Type | Expected Result |
| UT-001 | Validate GET /api/v1/Books with valid data | High | Positive | Returns 200 and a list of books. |
| UT-002 | Validate POST /api/v1/Books with valid data | High | Positive | Returns 200 and the newly created book. |
| UT-003 | Validate GET /api/v1/Books/{id} with invalid ID | High | Negative | Returns 404 error. |
| UT-004 | Validate DELETE /api/v1/Books/{id} | Medium | Positive | Returns 204 after successful deletion. |
| UT-005 | Validate PUT /api/v1/Books/{id} with empty payload | Medium | Negative | Returns 400 error for bad request. |
| UT-006 | Validate GET /api/v1/Activities with valid data | High | Positive | Returns 200 and a list of activities. |
| UT-007 | Validate POST /api/v1/Activities with valid data | High | Positive | Returns 200 and the newly created activity. |
| UT-008 | Validate GET /api/v1/Activities /{id} with a valid ID | High | Positive | Returns 200 with the activity matching the sent id |
| UT-009 | Validate GET /api/v1/Activities /{id} with invalid ID | High | Negative | Returns 404 error. |
| UT-010 | Validate DELETE /api/v1/Activities /{id} | Medium | Positive | Returns 204 after successful deletion. |
| UT-011 | Validate PUT /api/v1/Activities /{id} with empty payload | Medium | Negative | Returns 400 error for bad request. |

NB: To include testcases for all endpoints

# 6. Test Execution and Reporting

Test Execution Summary:  
- High-priority test cases executed first, followed by medium and low.  
Evidence Collection:  
- Screenshots, logs, and automated test reports.  
Defect Management:  
- Log and track defects for failures, ensuring resolution before deployment.

# 7. Conclusion and Next Steps

Test Execution Summary:   
10 of the 12 test cases passed successfully. Two test cases failed, primarily due to error handling issues with the PUT and POST endpoints.  
Next Steps:  
1. Review and fix the error handling for invalid inputs on the PUT and POST endpoints.  
2. Re-test after fixes are applied.  
3. Continue with performance and integration testing for load/stress handling.