**1. Test Strategy**

|  |  |
| --- | --- |
| Introduction | The test strategy outlines our approach to testing the API of the fake post management application. The goal is to ensure the API is functioning correctly, find as many bugs as possible, and provide meaningful reports on the testing results. |
| Objectives | * Validate the functionality of all API endpoints. * Ensure data integrity in the SQLite database. * Verify user authentication using JWT. * Test edge cases and error handling. |
| Scope | API Endpoints:   * User creation * Post creation * List all posts (including public and user-specific) * Update and delete user posts   Authentication via JWT |
| Test Types | * Functional Testing: Verify that each endpoint works as expected. * Integration Testing: Ensure the API integrates correctly with the SQLite database. * Security Testing: Check authentication mechanisms using JWT. * Negative Testing: Test with invalid inputs to ensure the API handles errors gracefully. |
| Test Environment | * Docker container with the SQLite database located at /app/sql\_app.db. |
| Tools | * Programming Language: Python, using libraries such as requests and unittest/pytest for test automation. * Reports: Using pytest-html to generate detailed reports. |
| Testes case | * Detailed test cases and scenarios for each feature will be documented and executed. |

2. Test Plan

Each test case will have a unique ID.

All steps will be detailed

Cover typical user actions, edge cases, and invalid input.

2.1. Test Case Design

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Steps | Expected Result | Actual Result |  |
| TC01 | User Creation | 1. Send POST request to /create-user with valid data. | 201 Created, user data in response |  |  |
| TC02 | User Creation (Invalid Data) | 1. Send POST request to /create-user with invalid data (e.g., missing fields). | 400 Bad Request, error message in response |  |  |
| TC03 | Generate JWT | 1. Send POST request to /token with valid username and password. | 200 OK, JWT token in response |  |  |
| TC04 | Generate JWT (Invalid Credentials) | 1. Send POST request to /token with invalid username or password. | 401 Unauthorized, error message in response |  |  |
| TC05 | Create Post | 1. Send POST request to /create-post with valid data and JWT in header. | 201 Created, post data in response |  |  |
| TC06 | List All Posts | 1. Send GET request to /list-posts. | 200 OK, list of posts in response |  |  |
| TC07 | Update Post | 1. Send PUT request to /update-post/{id} with valid data and JWT in header. | 200 OK, updated post data in response |  |  |
| TC08 | Delete Post | 1. Send DELETE request to /delete-post/{id} with JWT in header. | 204 No Content |  |  |
| TC09 | Create Post (Invalid JWT) | 1. Send POST request to /create-post with valid data and invalid JWT in header. | 401 Unauthorized, error message in response |  |  |
| TC10 | Update Post (Invalid User) | 1. Send PUT request to /update-post/{id} with valid data and JWT for a different user. | 403 Forbidden, error message in response |  |  |