**Testing Knowledge**

**PART 1 - Manual testing:**

***Testing an API involves a number of different activities, including planning, designing test cases, executing tests, and reporting and documenting results. Here is a free-form document outlining a possible test process for the https://jsonplaceholder.typicode.com/posts***

***API:***

***Planning: Determine the requirements and expectations for the API, including the expected behavior of the API, the types of requests and responses that are supported, and any performance or security considerations.***

***Identify the target audience for the API and any specific use cases that need to be tested.***

***Define the scope of testing, including which endpoints and parameters to test and which test environments to use.***

***Identify any necessary tools or resources, such as testing frameworks, test data, or test automation tools.***

***Define the test strategy, including the types of tests to be performed (such as functional, integration, performance, or security tests) and the level of testing (such as unit, system, or acceptance tests).***

***Designing test cases:***

***Identify the various API endpoints that need to be tested, including the inputs and outputs of each endpoint.***

***Develop test cases that cover the functional requirements of each endpoint, including edge cases, error handling, and boundary conditions.***

***Create test data and test scenarios that exercise the full range of inputs and outputs for each endpoint.***

***Design test cases that cover any security or performance requirements for the API.***

***Executing tests:***

***Run automated tests to check the functional requirements of the API and ensure that all endpoints are working as expected.***

***Manually execute tests to validate the results of the automated tests and identify any issues that were not caught by the automated tests.***

***Verify that error handling is working as expected, including testing edge cases and invalid inputs.***

***Validate the performance and security of the API by running load tests, stress tests, and security tests.***

***Document the results of each test and record any defects or issues that were found.***

***Reporting and documenting results: Generate a report of the test results, including any defects or issues that were identified and any recommendations for improving the API.***

***Document the test process and any test cases or test data that were used, to provide a reference for future testing.***

***Update any relevant documentation, such as user manuals or developer guides, to reflect any changes or improvements to the API based on the test results.***

***Work with the development team to address any issues that were identified and retest the API to ensure that the issues have been resolved.***

***In the context of the https://jsonplaceholder.typicode.com/posts API, the specific test activities may include:***

***Planning:***

***Identify the expected behavior of the API, such as the ability to return all posts for a given user ID or to handle invalid user IDs or requests. Determine which test environments to use, such as local or remote test environments. Identify any necessary testing tools, such as Postman or Swagger, to help design and execute tests.***

***Designing test cases:***

***API Develop test cases that exercise each of the API endpoints, including testing the /posts endpoint with different user IDs.***

***Test the input validation and error handling of the API, such as providing invalid user IDs or requesting non-existent endpoints.***

***Create test data that includes various types of posts and comments to ensure that all data is being returned correctly.***

***Executing tests:***

***Run automated tests to check the functional requirements of the API and ensure that all endpoints are working as expected.***

***Manually test the API by making requests to the /posts endpoint with different user IDs and verifying that the expected data is returned. Verify that they can handle high volumes.***

**PART 2 - Create a BDD feature file for API Testing**

***Feature: Search for available posts by title***

***Scenario Outline: Search for posts by title***

***Given I am a user of Example.com***

***When I search for posts with the title "<title>"***

***Then I should receive a list of available posts***

***Examples:***

***| title |***

***| First post |***

***| Another post |***

***| Example post |***

***Scenario: Search for non-existent post by title***

***Given I am a user of Example.com***

***When I search for a post with the title "Non-existent post"***

***Then I should receive an empty list***

***Scenario: Search for post with empty title***

***Given I am a user of Example.com***

***When I search for a post with an empty title***

***Then I should receive an error message***

***Scenario: Search for post with invalid title***

***Given I am a user of Example.com***

***When I search for a post with an invalid title, such as "%&$#@!"***

***Then I should receive an error messageunt***

***In this feature file, we have included four different scenarios to test the search functionality of Example.com:***

***• The first scenario outlines the basic functionality of searching for posts by title and receiving a list of available posts.***

***• The second scenario tests the scenario when the searched post is not available.***

***• The third scenario tests the scenario when an empty title is searched,***

***• The fourth scenario tests the scenario when an invalid title is searched. You can then write step definitions in your chosen programming language to execute these scenarios as part of your automated API testing process.***

**PART 3 – Create a BDD feature file for UI Testing**

***Feature: Sign-up Functionality***

***As a new user of www.w3schools.com, I want to be able to create a new account by providing my details using the sign-up form, so that I can access additional features and resources on the website.***

***Scenario: Successfully create a new account***

***Given I am on the w3schools sign-up page***

***When I fill in my details in the sign-up form***

***And I click on the sign-up button***

***Then I should see a message informing me that my account has been created***

***Scenario: Attempt to create a new account with incomplete details***

***Given I am on the w3schools sign-up page***

***When I fill in some but not all of my details in the sign-up form***

***And I click on the sign-up button***

***Then I should see a message informing me to complete all required fields***

***Scenario: Attempt to create a new account with an existing email address***

***Given I am on the w3schools sign-up page***

***When I enter an email address that has already been registered***

***And I click on the sign-up button***

***Then I should see a message informing me that the email address is already in use***

***Scenario: Attempt to create a new account with a weak password***

***Given I am on the w3schools sign-up page***

***When I enter a weak password in the password field***

***And I click on the sign-up button***

***Then I should see a message informing me to choose a stronger password***

***In this feature file, we have included four different scenarios to test the sign-up functionality of w3schools.com:***

***• The first scenario tests the basic functionality of successfully creating a new account by providing all required details in the sign-up form.***

***• The second scenario tests the scenario when some but not all the required details are entered in the sign-up form.***

***• The third scenario tests the scenario when an email address that has already been registered is entered.***

***• The fourth scenario tests the scenario when a weak password is entered in the password field.***