Software QA Engineer – Challenge

**Overview**

You have an API that stores its data on an SQLite database. The API is a fake post management application backend. It allows creation of users; each user can create posts, list all the post on the application that are public, (including all the posts of the user), as well as update and delete their own posts. The database is stored within the container at location **/app/sql\_app.db;** it doesn’t require any sort of authentication to use it, just connect to it and you can use it as you want.

Tip: It’s important to mention that each action performed by the user requires authentication, for that reason you need to generate a ***jwt*** and pass it to each request. Use the **/token** endpoint to generate a token. Pass the username and password and you are good to go. Good luck!

To get started, follow the instructions on this [repository](https://hub.docker.com/r/farlahi00/qachallenge) to setup your environment.

**Challenge**

1. Define a test strategy that will allow you to test the API and find as many bugs as possible. Your test strategy should output a detailed strategy plan that includes a test plan.
2. We don’t run tests manually, we run automated tests. Implement your test strategy using any programming language that is easy for you.
3. Testing is about producing meaningful reports. Generate a report that will show the execution results of your automated tests.

**Evaluation Criteria**

1. Test Strategy
   1. A clear strategy that will be used to test the API. A docx file is expected.
   2. Test plan that includes test cases that are going to be used to test the API. An xlsx file is expected.
   3. Test cases design principles.
   4. Number of bugs identified.
2. Test Framework Implementation
   1. Clean code that is easy to maintain.
   2. Test case implementation matches test case design found in test plan.
3. Test Reports
   1. Clear and meaningful execution report.

You are required to submit a link of your test suite framework repository. The repository must also include all other documents that are expected in the evaluation criteria. These documents should be placed in the folder ***docs*** which must be in the root level of the repository.