Assignment Test Automation Engineer

App Design Documentation / Requriments

The application displays a welcome page which shows a greeting
and implements an API `/finance/api/v1.0/payments` which supports below methods:

1. `GET /finance/api/v1.0/payments` - to retrieve all the payments from SQLite database as a list (allowed for admins only).
2. `PUT /finance/api/v1.0/payments` with json data in this form `{"purchase": <integer>, "amount": <float>, "currency": "<str>"}` where currency code is a 3-letter captalized string (e.g. `GBP` , `USD` , `EUR`) - to create a payment entry in the database (allowed for authorized users).
3. `POST /finance/api/v1.0/payments` - process all payments, where `completed` is still `0` : change the currency to `EUR` (convert amount to `EUR` according to the rates in the currency table) and set `completed` to `1` (allowed for admins only).
4. `DELETE /finance/api/v1.0/payments/<payment_id>` - to remove the payment by the given `payment_id` (allowed for admins to remove any payment and for the users to remove only their payments).

Notes:

- Currently, two users are registered in the system: `admin/admin` and `john/john` .
- At present time the application supports basic authentication only.

Deployment Instructions

1. Start the web application - in a terminal window, execute `python app.py` .
2. Application will start in localhost:5000

Note: you will need to install necessary Python dependencies.

Test Assignment

Note: Please complete and assignment and share the results (code & reports) in github or in zipfloder.

1. Explore the actual behavior of the application using `curl` or `Postman` or any other tool.
2. Design a test strategy for the application, create a checklist of potential test cases and prioritize them.
3. Automate test cases a)in Python: create test libraries,b)in Robot Framework: import these libraries and create test suites and test cases.

   Note: after each test consider to return the test target (app, data) into the same state.
4. Execute test cases locally using robot framework and from a docker container.
5. Provide a test report generated by Robot Framework. What issues have been discovered and what defects would you submit?
6. The application uses SQLite database, but what will change in your code if it gets replaced with MySQL or MS SQL or Oracle?
7. What would you improve in your approach and code if you have more time?