The Answers to the questions asked in the Mail

Question : Explain your approach to the implementation.

Answer : The approach was to follow the instructions given in the email. The first step was to install the application and test each functionality as part of exploratory testing. Based on the findings from the exploratory testing I started preparing skeleton of the test suite which will test the registration and login UI pages. The next step was to write robot test cases to test the UI pages. Then finally I used Postman to do exploratory testing on the API endpoints. From there, I formulated the final test cases for API. At last, use “coverage.py” to check the code coverage.

Question : How do you review code?

Answer : Code Review is an essential step to ensure the quality of the code before it hits the testing cycle. It consists of walkthrough and code inspection against the requirements, architectural guidelines and coding standards. It can prevent major bugs, manual errors penetrating into the later stages of testing. It is recommended to follow a standard checklist during the code review process, while in this particular case I did not have any so I relied on my knowledge and experience. Moreover, studies have found that, limiting the rate of code review (i.e. number of lines reviewed / hour) is helpful in increasing its effectiveness. This means that in one single sitting a reviewer must review only 200-400 lines of code per hour. While doing peer reviews I try to comment in a friendly and constructive manner.

Question : How do you enforce coding standards?

Answer : Coding standards are easy to follow when they are defined in the form of checklists. Code review also ensures that code is coding standard compliant.

Question : How do you plan what kind of approach you take for test automation - what libraries to use, how does it work in couple of years, how to make it easy to maintain, etc? What are the main points to consider?

Answer : In order to take right decisions related to the use of test automation libraries, I recommend doing a thorough research on the kind of system/application which is under test. For example, module based testing is specific type of test framework for systems with high modularity, whereas keyword based test framework is used for systems where quick turnaround is required with low expertise. There are several technology solutions which are available in the market which solve different testing related problems. Therefore, it is highly recommended that careful analysis of system requirement, test budget, test resources, test personnels should be done before introducing test automation as a solution to the testing troubles. This may also require careful study of the maintainability of the libraries in future.

Question : Code testability, how do you enforce it?

Answer : In a nutshell, the best way to keep the code testable is to follow Test Driven Development (TDD). This will ensure that the production has only that amount of code that is necessary to pass the test cases written early during the software development lifecycle. Apart from that, adding good amount of unit tests to test the code, test coverage improvement, code reviews and coding standards also improve code testability.

Question: How do you make sure that the product is testable?

Answer : Code properties like consistent naming conventions, modularity, maintainability increases testability of the system or application. Additionally, testing tools such as test management tools or defect tracking tools, eases the testing processes and as a result increases testability. Version management system to maintain test environment with right type of test data increases testability.

Question: Report of executed tests

Answer : The reports can be created by running the testcases executing from commandline the following commands ./run\_tests.sh and ./run\_api\_test.sh

Question : Report of found issues/bugs

Answer : Found a single bug. The screen shot is pasted in the github folder by the name "bug found - AttributeError- 'dict' object has no attribute iteritems.png"

Question : Exploratory Testing Report

Answer : It is available in the github folder by name Exploratory Testing - Demo App.docx

Question : Tell us what improvement would you propose for the app

Answer :

1. More interactive user interface
2. Possibility to update user records using web interface
3. User authentication using emailID to make things more interesting for the tester
4. http —> https
5. Domain name for the application
6. Possibility to setup and teardown database during API testing.

Question : If you would be given a week to do quality assurance for this product, briefly plan the tasks based on your skills, knowledge and expertise

Answer : It is available in the github folder by name Quality Assurance Plan.docx

Question : Instructions how to run it and short description of components, including external libraries

Answer :

1. From the directory Flasky, execute ./run\_tests.sh from commandline to execute the robot testcases.
2. From the directory Flasky, execute ./run\_api\_tests.sh from commandline to execute the api testcases.
3. Output reports will be updated after the execution in the same directory.

In addition, type “open demo\_app/htmlcov/index.html” in browser to see the code coverage.

Question : Description about taken approach and potential gaps in application

Answer : For the quick and easy solution, I used Robot framework and Python unittest to accomplish the given task. There is a lot of room for improvement and I assure that if given more time to learn I can improve the solution and find more bugs in the application.

Question : How much time it took, there is no time limit as such. Remember to return the task when agreed or let us know if you run late.

Answer : The whole process of testing and writing answers to the question took ~9 working days.