Signant_assessment

Technical assessment is a simple login page which you should test with automation using Robot Framework for UI and Python for API.

The demo app can be found from: https://github.com/Interview-demoapp/Flasky

Basic instructions can be found from github page. Please create tests against the acceptance criteria.
The following is needed:

Write UI automation with Robot Framework
Delivery: User_Login.robot User_Registration.robot
I put them in two separate files. Because to avoid database entry conflicts, I used random created user name and password as implementation. I also want to avoid dependency between test cases, so I used separate test suites for login feature testing.

Write API automation – requests with Python (we know there are RF libraries for APIs, but we want to see your Python skills)
Delivery: api_GET.py api_POST.py api_DELLETE.py api.robot
I used two implementation. First I used Python, only GET case passed, POST and DELETE failed. Then I used RF, because I am familiar with it. The same situation. So my conclusion is the web site did not have POST and DELETE API. No PUT neither because there is not GUI show update is possible.

When using Python also write unit tests for Python implementation (=test the test code)
Delivery: I do not know how to write unit test. I will learn from colleagues.

Explain your approach to the implementation
Delivery:
see comment in files

Other questions to answer about testing and testability, from your point of view
How do you review code?
How do you enforce coding standards?
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?
Code testability, how do you enforce it?
How do you make sure that the product is testable?

Delivery: Testing and Testability.txt
I had put answers of above several questions into file Testing and Testability.txt.

Provide the following back as redistributable git repository.
Testing Task:
Report of executed tests
Delivery: Test Report.txt

Report of found issues/bugs
Delivery: Test Report.txt

Answers to the questions related testing and testability
Delivery: Testing and Testability.txt

Exploratory Testing Report
Delivery: Test Report.txt

Tell us what improvement would you propose for the app
Delivery: Improvement Idea.txt

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
Coding Task:
Delivery: Test Plan_Week.txt

Working stable application (Freely Distributable) of test automation implementation
Instructions how to run it and short description of components, including external libraries
Delivery: comment in each test files.

Description about taken approach and potential gaps in application
Delivery: The application is overly simple so it lacks of many features/functions that should be in such registration system. For example, lacking of feature to support update information is one. Publishing password without masking is another. Then other features can refer to Improvement Idea.txt. There could be much more but let's admit it is overly simple app.

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.
Delivery: 2h each day in three days