**Testing Task:**

**1.How do you review code?**

**Code formatting**

1. I make sure to check the code formatting to improve readability.
2. I Ensure that there is no blocker.
3. I check the indentation, proper naming conventions.

**Architecture**

1. I make sure that code is in sync with existing code patterns/technologies.
2. I ensure to code should look compact.
3. Code should look organised.
4. I use appropriate design pattern, after completely understanding the problem and context.

**Coding**

1. I follow no hard coding, use configuration values.
2. I use framework features, wherever possible instead of writing custom code.
3. I make sure code is bug free.
4. Execution is stable.
5. Avoid creating unnecessary variables.
6. Avoid abbreviations.

**Code should have all these properties**

1. Maintainability
2. Testability
3. Debuggability
4. Configurability
5. Reusability
6. Security

**2.How do you enforce coding standards?**

A Code standard enforced by automated rule checks improve the readability and maintainability of code also reduce the number of bugs. I used indentation, proper naming conventions.

I follow no hard coding, use configuration values.

Enforcement of coding standards too often depends on programmers already under deadline pressure to be disciplined while they code or to make time to perform peer code reviews.

To ensure your selected coding standard is followed, and thus effective, your team should find as many automated ways to enforce as many of its rules as possible.

**3.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?**

For Test Automation planning I used the following steps

1. Analysis and Planning-

This is the first step where the Analysis is done on the requirement and planning for execution is started. It is the first step to understand the product.

1. Identify Tools and Technology

This is the second step where the tools should be decided to start with the automation based on the technology on with application was developed.

1. Select and Design Framework

This is the third step where frameworks is decided as per the required application.

1. Define Scope of tests

Here the scope for developing and testing the application on the desired tool and framework is decided that what is the workflow of the code.

1. Prepare the Testbed

On this step all the environments set up was done to make sure no blockage should be fetched while implementing or testing of the code.

1. Identify common functionality/methods

Here we identify the common methods and functionality of code to keep it separate so that no duplication will take place.

1. Implement the test cases

After the environment setup was done, here the test cases must be written and executed to see the success results.

1. Review the test cases

Review the code to check the bugs not generated.

1. Maintenance

Finally, it should go to the maintenance stage to keep it more productive.

**4. Code testability, how do you enforce it?**

Code testability is the testing of the code if the additional feature is added to the web application and the same needs to be implemented in the framework to the existing code so to make sure that the code should not be deprecated and align proper, we use code testability approach.

**5. How do you make sure that the product is testable?**

**To make sure product is testable,**

**Idea-Based Product Testing**

Product testing is the process of gathering feedback directly from customers or potential customers about a product. The feedback can be through informal conversation, formal surveys, or even indirectly through data about clicks on an online advertisement or traffic to a website.

**Prototype-Based Product Testing**

Product testing can happen both before and after the official launch, all the way from the initial idea stage to the prototype to the finished product. In the beginning, product testing helps businesses to determine the viability of the product.

**Market Testing**

Moving closer to the launch-ready version, product testing identifies ways to further improve on what has already been built.

**6. Tell us what improvement you would propose for the app?**

a. This app is used to register the user with the user details.

b. After registration, user can login and check the dashboard.

**Suggestions:**

I have noticed while testing the application that it is **allowing all the duplicates** value while registering the user only username is unique other details have no authentications.

I have noticed that the UI of the application could be made **more interactive** to the user.

Some important **features** should be added that like **alert windows**, **Pictures** to figure out what the web page is all about just by looking at it.

There must be one **automatic robot** chat generated on the web application with interact with the user to understand why user is present on the web page.

We can improve a lot of features in this app because after login the user we can use this app as per the requirement we can develop this app for creating any website.

**Coding Task:**

**1.Working stable application (Freely Distributable) of test automation implementation.**

Yes, the application is working stable so, it is easy to automate it**.**

**2.Instructions how to run it and short description of components, including external libraries.**

I have used the PyCharm as the IDE for Robot Framework for UI Automation and Python for API.

**Libraries used**:

Requests

Selenium

Robot framework

Robot framework-selenium library

Json

Json path

HTTPBasicAuth

**Plugins**

Intellibot

Intellibot@ selenium library patched

Robot framework support

**Commands to install packages:**

>pip

>pip install robotframework

>pip install seleniumlibrary

>pip install requests

>pip install jsonpath

**Instructions to run**

a. To run Robot file, Go to the project folder location and type >robot Test.robot

b. To run Python file, Go to the project folder location and type>python Test.py

c. Also right click and run file

**3. Description about taken approach and potential gaps in application.**

a. First, I have started performing the **manual steps** which helps to navigate from one web page to another shows the workflow of the application.

b. I have gone through the documentations for further instructions which helps to understand which action needs to perform with respect to the Automation Testing and what verifications needs to be done.

c. I have **installed all the required** libraries and packages step by step to set up the environment for Automating UI with Robotframework and API Testing with python.

d. I have started creating the **project folder** structure which needs to store all the files on the desired locations.

e. I have started writing the code as per the required steps mentioned in the documentations and implemented all the methods also **added plugins** to the IDE to make sure there will be no blocks during the test run.

f. Once I have completed implementing the code, when error is shown I have **fixed** it by carefully looking on the issue.

**Note**: I have noticed while testing the application that it is **allowing all the duplicates** value while registering the user only username is unique other details have no authentications.

I have noticed that the UI of the application could be made **more interactive** to the user.

Some important **features** should be added that like **alert windows**, **Pictures** to figure out what the web page is all about jus by looking at it.

There must be one **automatic robot** chat generated on the web application with interact with the user to understand why user is present on the web page.

**4. How much time it took.**

**Day 1**- Installed **Python,** Installed **PyCharm,** Installed **pip** commands like, UI Automation folder

>pip

>pip install robotframework

>pip install seleniumlibrary

* I have **Cloned the Project** from GitHub repository in to PyCharm.
* I have **installed** all the required **plugins to integrate robotframework with PyCharm**.
* I have **created the user** on the web page manually and checked the workflow or navigations from one page to another.
* I have located the web elements into the **Variable.robot** file.
* I have started writing the **Resource page** where action methods are created because I have used **POM** (Page Object Model) for designing the automation so that any changed in future can be easily made.
* I have written Register Test case to make sure the **user is registered** and **authorized** to login and his details must be shown on the dashboard.
* I have separated the variables file so that the code looks clean and tidy.
* After implementing the code with keywords because **robotframework is a keyword driven framework,** so I have initialized local and global keywords.
* To run the test, use the command from the test file >robot Register.robot
* All the tests got passed.

**Day 2** – API with Python, Installed requests libraries, Json ,API Automation folder

* First, I have created a **batch file** for the following commands shown to reduce the manual steps for putting **server up and running** to run the application.
* **set FLASK\_APP=demo\_app**
* **flask init-db**
* **flask run --host=0.0.0.0 --port=8080**
* I have created separate folder for API automation using python. After importing requests. I have started requesting the server with the GET, POST, PUT requests with the python code.
* I have implemented all the requests till token is generated with valid credentials.

**Day 3** -API Automation Successful, user is not updating with valid token authentication.

* I was facing some errors while fetching the requests and after tons of attempt finally it worked at the end of the day till token authorization and getting user details.
* To put the user or update any request with token available is not updating the user with the given token even after authenticating it correctly. I have considered it as bug to discuss further.

**Day 4**- I have started on preparing reports.

**Day 5**- **Submitted**