

Expertlead QA Automation Engineer

Pytest-bdd solution has been used for creation of Automation Framework so as to use Cucumber+Gherkin feature files and to use full functionality of pytest. As suggested in the assignment, focus has been to reduce hard-coding and improve code reusability. The Gherkin steps have been reused multiple times for the above purpose.

Pytest Test Automation Framework

The Test Automation Suite will be having below folders.

- Features: Feature files will be created based on each feature. The different scenarios in the particular feature will be listed in the same feature file.
- Source: Source folder will be used to list the modules used to implement the source files for corresponding feature files. Emphasis will be kept to reuse the codes wherever possible.
- Pages: This folder will be used to save the locators for each page in case of Front End Automation and api's in case of Backend Automation. Each file will be used to save the locators belonging to each page in the UI.
- Drivers : Will be used to save driver details, docker configuration etc.
- Synchronization : Folder used to create and save custom wait or to import any wait libraries.

The file is hosted in the private repository <https://github.com/getarun4t/expertlead.git>

Test Execution

To run the framework, please follow the below steps.

1. Clone or download the project.
2. Go to project root in CLI.
3. Run the following
`pip install requirements.txt`
In windows systems, please make sure to add the above to Path.
4. Now we should start the containers by running the below step.
`docker-compose up`
We should be able to see selenium hub created and chrome node registered to the hub
5. Now we just need to run the test cases using the below command in *another CLI instance* or run the test cases in any python supported IDE by setting run configuration as pytest.
`pytest`
To view the execution, please install vnc player and connect to `vnc://localhost:5901`. Please note that vnc player (screen sharing) is pre-installed in Mac OS.

Alternatively the test cases can be executed by using the chromedriver and selenium installed in the local. Please follow the below steps for doing the same.

1. Clone or download the project.
2. Go to project root in CLI.
3. Install selenium webdriver and the latest version of chromedriver (assuming that chrome is pre-installed in the system).
4. Change the chromedriver file to run in the local machine and mention the chromedriver location. (details commented in the chromedriver file).
`..Expertlead/Drivers/chromedriver.py`
5. Run the following in CLI
`pip install requirements.txt`
In windows systems, please make sure to add the above to Path.
6. Now we just need to run the test cases using the below command in CLI or run the test cases in any python supported IDE by setting run configuration as pytest.
`pytest`
Execution can be viewed in chrome browser in the local itself.

For creation of a report, please use the below while running the test cases.

`pytest --html=report.html`