Selenium Hybrid Framework

**STEP\_1: Create new Project & Install Required Packages/plugins**

* **Selenium** : Selenium Libraries
* **Pytest** : Python Unit-test Framework
* **Pytest-html** : Pytest HTML Reports
* **Pytest-xdist** : Run Tests Parallel
* **Openpyxl** : MS Excel Support
* **Allure pytest** : to generate allure reports

**STEP\_2: Create Folder Structure**

**Project Name**

pageObjects (Package)

|

testCases (Package)

|

Utilities (package)

|

TestData(Folder)

|

Configuration(Folder)

|

Logs (Folder)

|

Screenshots (Folder)

|

Reports (Folder)

|

Run.bat

**STEP\_3 : Automating Login Test Case**

**3.1 :** Create LoginPage Object Class under “pageObjects”

**3.2 :** Create LoginTest under “testCases”

**3.3 :** Create confest.py under “testCases”

**STEP\_4 : Capture screenshots on failures**

**4.1 :** Update Login Test with Screenshot under “testcases”

**STEP\_5 : Read common values from “ini” file**

**5.1 :** Add “config.ini” file in “Configurations” folder .

**5.2 :** Create “readProperties.py” utility file under utilities package to read the common data

**5.3 :** replace the hard coded values in Login test case

**STEP\_6 : Adding logs to test case**

**6.1 :** Add “customerLogger.py” under utilities package

**6.2 :** Add logs to login test case

**STEP\_7 : Run Tests on Desired Browser/Cross Browser/Parallel**

**7.1 :** Update contest.py with required Fixtures which will accept command line argument (browser)

**7.2 :** Pass browser name as argument in command line

* **To Run tests on desired browser**

pytest -s -v testCases/test\_login.py --browser chrome

pytest -s -v testCases/test\_login.py --browser Firefox

* **To Run tests parallel**

Pytest -s -v -n=3 testCases/test\_login.py --browser chrome

Pytest -s -v -n=3 testCases/test\_login.py --browser Firefox

**STEP\_8 : Generate pytest HTML Reports**

**8.1 :** Update conftest.py with pytest hooks

**8.2 :** To Generate HTML report run below command

**STEP\_9 : Automating Data Driven Test Case**

**9.1** : Prepare test data in Excel sheet, place the excel file inside the TestData folder

**9.2** : Create “Excelutils.py” uitility class under utilities package

**9.3** : Create LoginDataDrivenTest under testCases

**9.4** : Run the test case

**STEP\_10 : Adding New Testcases**

**10.1** : Adding new customer

**10.2** : Searching customer by email

**10.3** : Searching customer by Name (first name,last name)

**STEP\_11 : Grouping Tests**

**11.1** : Grouping markers(Add markers to every test method)

@pytest.mark.sanity

@pytest.mark.regression

**11.2** : Add Marker entries in pytest.ini.file

Pytest.ini

.……………..

[pytest]

Markers =

Sanity

Regression

**11.3** : Select groups at run time

-m “sanity”

-m “regression”

-m “sanity and regression”

-m “sanity or regression”

**Run command:**

Pytest -s -v -m “sanity or regression” --html=./Reports/report.html testCases/ -- browser chrome

**STEP\_12 : Run Tests in Command Prompt & run.bat file.**

**12.1** : Create run.bat file

Pytest -s -v -m “sanity” --html=./Reports/report.html testCases/ --browser chrome

**12.2** : Open command prompt as **Adminstrator** and then run **run.bat file**

**STEP\_13 : Push the Code to GIT and GITHUB Repository**

1) Create local git repository for your project

\* guide your project to the exact path

cd D:\Nandu\PYTHON\_PRACTISE\nopCommerceApp

\* git init

create an empty git local repository

\* git remote add origin https://github.com/Nandagopal2000/nopcommerceApp.git

Connect your local git(local repo) with Github(global repo)

\* Before doing commit first time we need to execute these below commands

git config --global user.name "Nandagopal2000"

git config --global user.email "nandugopaldevji@gmail.com"

\* git status

checking status of files(commited or not commited)

\* git add -A

Add all the files to stagging area

\* git commit -m "first commit"

Commit the code into git repository(local repo)

\* git push -u origin master

Push the code from git ---> github repository

2nd round

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

git status

git add -A --> add all the files in to stagging/indexing area

git commit -m "comment"

git push -u origin master

git pull --> pull all the files from github to local

**STEP\_14 : Run Tests using Jenkins**