Selenium Hybrid Framework

(Python, selenium, PyTest, Page Object Model, HTML Reports)

**Step 1: Create new Project & install Required Packages/plugins**

* Selenium : Selenium Libraries
* Pytest: Python UnitTest framework
* Pytest-html : PyTest HTML Reports
* Pytest-xdist : run tests parallel
* Openpyxl : MS Excel Support
* Allure-pytest : to generate allure reports

**Step 2 : CreateFolder Structure**

**Project Name**

|

pageObjects (Package)

|

Testcase (Package)

|

Utilities (Package)

|

TestData (Folder)

|

Configurations (Folder)

|

Logs (Folder)

|

Sreenshots (Folder)

|

Reports (Folder)

|

Run.bat

**Step 3 : Automating Login Test Case**

1. Create LoginPage Object Class under **“PageObjects”**
2. Create LoginTest under **“testCases”**
3. Create conftest.py under **“testCases”**

**Step 4 : Capture screenshot on failures**

1. Update Login Test with screenshot under **“testCases”**

**Step 5 : Read common values from ini file**

1. Add “config.ini” file in “Configuration” folder
2. Create “readProperties.py” utility file under utilities package to read common data.
3. Replace har coded values in login test case

**Step 6 : Adding logs to test case**

1. Add “customLogger.py” under “utilities” package.
2. Add logs to Login test case.

**Step 7 : Run Tests on desired Browser/cross Browser/Parallel**

1. Update conftest.py with requires fixtures which will accept command line argument (browser name)
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**

1. Update confest.py with pytest hooks
2. ToGenerate HTML report run below command:

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

**Step 9 : Automating Data Driven Test Case**

1. Prepare test data in excel sheet, place the excel file inside the TestData folder
2. Create “ExcelUltils.py” utility class under utilities package
3. Create LoginDataDrivenTest under testCases
4. Run the test case

**Step 10 : Adding new testcases**

1. Add new customer
2. Search customer by email
3. Search customer by name

**Step 11: Grouping Tests**

1. Grouping Markers (Add markers to every test method)

@pytest.mark.sanity

@pytest.mark.regression

1. Add marker entries in pytest.ini file

Pytest.ini

[pytest]

Markers

Sanity

Regression

1. Select group 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**

1. Create run.bat file