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: PyTests 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/Directory)

**|**

Configurations (Folder/Directory)

**|**

Logs (Folder/Directory)

**|**

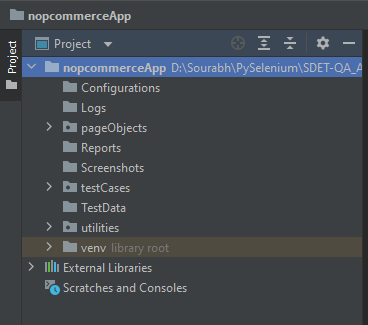
Screenshots (Folder/Directory)

**|**

Reports (Folder/Directory)

|

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 conftest.py under “testCases”

**Step 4: Capture Screenshot On Failures**

4.1: Update Login Test with Screenshot under “testCases”

**Step 5: Read Common Values From ini File (Initialization File)**

5.1: Add “config.ini” file in “Configurations” Folder

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

5.3: Replace hard coded values in Login TestCase

**Step 6: Adding Logs To Test Cases**

6.1: Add “customLogger.py” under “utilities” package

6.2: Add “logs” to the “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 arguments (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 In Parallel (pytest-xdist package is required)**

pytest –s –v –n=3 testCases/test\_login.py –browser chrome

pytest –s –v –n=3 testCases/test\_login.py –browser firefox

Note:- -n = number of test methods, Don’t run more than 3 methods at a time, otherwise slower execution

**Step 8: Generate pytest HTML Reports (pytest-html package is required)**

8.1: Update “conftest.py” with pytest hooks

8.2: To Generate HTML Report run below command:

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

------------------- No Log Output Captured ---------------------------

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

0R

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

**Step 9: Automation 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” utility class under “utilities” package.

9.3: Create “LoginDataDrivenTest” under “testCases”

9.4: Run the Test Case

**Step 10: Adding new Test Cases**

10.1: Add New Customer

10.2: Search Customer by email

10.3: Search Customer by 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 **Administrator** and then run **run.bat** file

**Step 13: Push the Code in Git & GitHub Repository**

1st Round

Initial Steps (only one time)

1. git init -> Create an empty Git Repository (Local Repository)
2. Git remote add origin \*

<https://github.com/pavanoltraining/nopCommerce_Hybrid.git> \*

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

Git config –global user.name “pavan”

Git config –global user.email [pavanoltraining@gmail.com](mailto:pavanoltraining@gmail.com)

1. Git status
2. Git add –A -> add all the files into staging/indexing area
3. Git commit –m “comment”
4. Git push –u origin master

2nd Round