**Selenium Hybrid Framework**

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

**Step1: Create new Project & install required packages**

* Selenium
* Pytest
* pytest-html
* pytest-xdist: run tests in paralel
* Openpyxl: python excel support library
* Allure-pytest

**Step2: Create Folder Structure**

**Project Name**

|

page\_objects(Package)

|

test\_cases(Package)

|

utilities(Package)

|

test\_data(Folder)

|

configurations(Folder)

|

logs(Folder)

|

screenshots(Folder)

|

reports(Folder)

|

run.bat

**Step3: Automation Login Test Case**

1. Create LoginPage object class under “page\_objects”
2. Create LoginTest under “test\_cases”
3. Create conftest.py under “test\_cases”

**Step4: Capture screenshots on failures**

Update LoginTest with screenshot under test\_cases

**Step5: Read common values from ini file**

1. Add “config.ini” file in “configurations” folder
2. Create “read\_properties.py” utility file under utilities package to read common data
3. Replace hard coded values in Login test case

**Step6: Add logs to test cases**

1. Add custom\_logger.py under utilities package
2. Add logs to Login test case

**Step7: Run tests on Desired Browser/Cross Browser/Parallel**

1. Update conftest.py with required Fixtures which will accept command line argument(browser)
2. Pass browser name as argument in command line

**To run tests on desired browser**

pytest -v test\_cases/test\_login.py –browser chrome

pytest -v test\_cases/test\_login.py –browser firefox

**To run tests in parallel**

pytest -v -n=2 test\_cases/test\_login.py –browser chrome

pytest -v -n=2 test\_cases/test\_login.py –browser firefox

**Step8: Generate pytest HTML reports**

1. Update conftest.py with pytest hooks
2. Generate HTML reports using command:

pytest -s -v –html=Reports\report.html --browser chrome

**Step9: Automating Data Driven Test Cases**

1. Prepare test data in Excel sheet, place de excel file inside test\_data folder
2. Create excel\_utils.py utility class file in utilities package
3. Create login\_data\_driven\_test under test\_cases
4. Run the test case

**Step10: Adding new test cases**

1. Add new customer (create add\_customer\_page.py under page\_objects package and test\_add\_customers.py under test\_cases)
2. Search customer by email (create search\_customer\_page.py under page\_objects package and test\_search\_customers\_by\_email.py under test\_cases)
3. Search customer by name (create test\_search\_customers\_by\_name.py under test\_cases)

**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]

markers=

sanity

regression

1. Select groups at run time

-m “sanity”

-m “sanity and regression”

-m “sanity or regression”

**Step12: Run tests in command prompt with run.bat file**

Make sure to install the python packages.

pushd C:\Python\Courses\Selenium with Python Hybrid Framework  
pytest -s -v test\_cases\ --browser chrome --html=.\reports\report.html

Run bat file with cmd (preferably as Administrator)

**Step13. Push the code to Git & Github Repository**

You can use right click Git bash here in Windows

Iinitial steps:

git init (create empty git local repository)

git remote add origin “<https://github.com/pavanoltraining/nopCommerceProject.git>” (connect your local repository with the global one)

git config --global user.name “vlad”

git config –global user.email “[vlad@gmail.com](mailto:vlad@gmail.com)”

git status (check status of files)

git add -A (add all files to staging area)

git commit -m “comment” (files are added to local repository)

git push -u origin master (files are added to global repository)

2nd Round:

git status

git add

git commit -m “comment”

git push -u origin master

git pull (pull files from github to local repository)

**Step 14. Install Jenkins (on windows 10)**

https://www.jenkins.io/download/

Download jenkins.war and put it in C:\Jenkins

Make sure you have java installed

Launch jenkins: java -jar jenkins.war

C:\Users\vladn\.jenkins\secrets\initialAdminPassword for initial password

[http://localhost:8080/](http://localhost:8080/login?from=%2F) and enter the initial password

Install suggested plugins

Create first admin user