**Personalised Selenium Hybrid Framework- soumya kanti de**

(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: Create Folder Structure

* Project Name
* pageObjects (Package)
* testCases (Package)
* utilities (Package)
* TestData (Folder)
* Configurations (Folder)
* Logs (Folder)
* Screenshots (Folder)
* Reports (Folder)

Step 3: Automating Login Test Case

* Create LoginPage Object Class under "pageObjects"
* Create LoginTest under "testCases"
* Create conftest.py under "testCases"

Step 4: capture screenshot on failures

* Update Login Test with Screenshot under "testCases"

Step 5: Read common values from ini file.

* Add "config.ini file in "Configurations" folder.
* "readProperties.py" utility file under utilities package to read common data.
* Replace hard coded values in Login test case.

Step 6: Adding logs to test case

* Add customLogger.py under utilities package.
* Add logs to Login test case.

Step 7: Run Tests on Desired Browser/Cross Browser/Parallel

* update contest.py with required Fixtures which will accept command line argument (browser).
* Pass browser name as argument in command line
* To Run tests parallel-

pytests-v-3 testCases/test\_login.py-browser chrome pytests testCases/test\_login.py -browser firefox

Step 8: Generate pytest HTML Reports

* Update conftest.py with pytest hooks
* To Generate HTML report run below command:
* pytest-s-v-n-3--html-Reports\report.html testCases/test\_login.py

Step 9: Automating Data Driven Test Case

* Prepare test data in Excel sheet, place the excel file inside the TestData folder.
* Create "ExcelUtils.py utility class under utilities package.
* Create LoginDataDriven Test under testCases
* Run the test case

Step 10: Adding new test cases

Step 11: Grouping Tests

* Grouping markers Add markers to every test method)-@pytest.mark.sanity and @pytest.mark.regression
* Select groups at run time
* -m " sanity" -m "regression"
* -m "sanity and regression"
* -m "sanity or regression"

Step 12: Run Tests in Command Prompt & run.bat file.

* Create run.bat file
* pytest -s -v "sanity! -html-/Reports/report.html testcases/-browser chrome
* Open command prompt as Administrator and then run run.bat file

Step 13: Push the Code to Git & GitHub Repository

**Creation of framework took time, framework is ready, only need to add test cases**

