Automation Framework

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

1. Installable Packages, Libraries, and plugins

* Python
* Selenium: Selenium Libraries
* Pytest: Python unit test framework
* pytest-html: For pytest html reports
* pytest-xdist: Run test in parallel
* openpyxl: For MS Excel support

2.Folder Structure of framework (POM )

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Details about packages and folders

* pageObjects-This package having classes according to pages of the application under test. That classes has Web Elements and methods to perform actions on them for that class.
* testCases-This package having all the test cases for application under test.
* utilities -This package having classes with reusable methods. Which we can use all over the framework.
* TestData- This is folder which consist of test data XL file for data driven test cases
* Reports- This is folder which will have test execution reports in HTML format.
* Logs- This folder will have logs captured while running test case.
* Screenshots -This is folder where screenshots of failure will get stored.
* Configurations- This folder is having config.ini file which is having commonly used data for testing.

1. Framework Features

* Test cases can run from command line.

Command: pytest -s -v --html=./Reports/report.html testCases/ --browser chrome

* Test cases can run on multiple browsers which are chrome, Firefox, and opera. Need to pass browser name from command line while running as follow.

Command: pytest -s -v testCases/ --browser chrome

* In framework html report will get generated with all logs in report folder.

Command-pytest -s -v --html=./Reports/report.html testCases/ --browser chrome.

* Test Can run in parallel using below command. In command ‘n’ will represent number of instances of browser.

Command: pytest -s -v n=2 --html=./Reports/report.html testCases/ --browser chrome.

* In framework can test application for data driven test cases , for data driven added XL sheet of data which will provide data to test cases for different scenarios.
* In framework screenshots will get captured on failure.
* Can run test suite by grouping test cases on functionality. Added group tags as smoke and regression.

Command: pytest -s -v -m "smoke" testCases/ --browser chrome

5.Steps to Run Test cases from Command Prompt with .bat file

1. Install Python to your machine

2.Install pytest (Open CMD type pip install pytest)

3.Add selenium (In CMD type pip install selenium)

4.Add webdriver\_manager (On CMD type pip install webdriver\_manager)

5. Add pytest-html - (On CMD type pip install pytest-html)

6.Add openpyxl – (On CMD type pip install openpyxl)

7.Double click on .bat file it will start execution

6. Test cases added to automate application

|  |
| --- |
| 1. test\_homepage\_title |
| 1. test\_login\_valid 2. test\_login\_invalid 3. test\_login\_ddt 4. test\_order\_product 5. test\_remove\_product\_from\_cart 6. test\_add\_all\_products\_to\_cart 7. test\_validate\_logout\_application  |  |  |  | | --- | --- | --- | | Sr. No | Test Case name | Status | | 1 | test\_homepage\_title | Passed | | 2 | test\_login\_valid | Passed | | 3 | test\_login\_invalid | Passed | | 4 | test\_login\_ddt | Passed | | 5 | test\_order\_product | Passed | | 6 | test\_remove\_product\_from\_cart | Passed | | 7 | test\_add\_all\_products\_to\_cart | Passed | | 8 | test\_validate\_logout\_application | Passed |   Note: All the required software and libraries are open source. |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |