**Selenium Python Framework Design**

1. Python Unit Testing Framework -pytest
2. Understand Logging and HTML Reports
3. Implement Selenium Python Framework from Scratch (Page Object Design Pattern)
4. Data Driven Framework with Excel to Selenium
5. Git Version Control and Jenkins CI/CD

**pytest Unit Testing Framework:**

1. Install pytest
2. Naming conventions to follow pytest
3. Running pytest(s) from command line and PyCharm IDE
4. Running selected test files using pytest
5. Running selected test methods based on matching keywords
6. pytest Tags mechanism to run tests based on functionality
7. Failing and Skipping tests with Annotations using pytest
8. What are fixtures and importance of their Hooks in pytest
9. How Fixtures can be configured in "conftest.py" file for better readability
10. Different scopes of fixtures and their related annotations to setup Pre and Post conditions of the test
11. How Parameterization can be achieved for Tests with multiple sets of data
12. How to pass command Line arguments into pytest
13. HTML report generation for pytest execution

**Selenium Python Framework Implementation from Scratch:**

1. Standards of writing selenium tests in framework
2. Creating Browser invocation Fixtures in conftest.py
3. Setting up Base class to hold all common Utilities.
4. Inheriting Base class to all tests to remove Fixture redundant code
5. Passing Command line options to select browser at run time
6. Implement Page Object Model pattern
7. Smarter way of returning Page Objects from Navigation methods
8. Creating Selenium WebDriver Custom Utilities in Base class
9. Parameteriging WebDriver tests with Multiple Data sets using pytest parameterization using fixture
10. Organizing Data from Separate Data files and injecting into fixture at run time
11. Implementing Logging feature to WebDriver tests
12. Test Execution with HTML Reporting
13. Automatic Screenshot Capture on Test Failures
14. Integrating Selenium Python Framework to Jenkins CI tool with Jenkins build parameterization
15. GitHub basics for Project version control

**To run tests:**

(.venv) C:\WorkspacePython\python-web-framework>python -m pytest -v -s --browser=chrome --html=report\report.html

python -m pytest -v -s tests\test\_e2e.py -- browser\_name=chrome --html=report\report.html

python -m pytest -v -s tests\test\_test\_home\_page.py -- browser\_name=chrome --html=report\report.html

(.venv) C:\WorkspacePython\python-web-framework>python -m pytest -v -s tests\test\_home\_page.py -- browser\_name=chrome --html=report\report.html

To generate reports:

(.venv) C:\WorkspacePython\python-web-framework>python -m pytest -v -s --browser\_name=chrome --html=report.html