## Below library and framework are used:

- pytest for better control on test case execution

- pytest-html report for html report generation

- pytest-xdist for parallel test execution

- pytest-rerunfailures to rerun failed test cases

- selenium for ui automation

- webdriver-manager to get better control over web driver. if driver not installed it automatically installs it

- page object model to minimize code maintainance. With POM its easy to add/modify any element/component at single location

- custom fixtures and logger to log test case steps and function logs into HTML report

## Instruction on how to execute:

- Unzip the project zip file on local machine

- From terminal go to unzipped project location where you can find 'requirements.txt' file

- run command 'pip3 install -r requirements.txt' to install all required python libraries

- In the same directory 'driver.py' file is present, this is main python file which will execute all test cases.

- driver.py accepts various arguments which you can get by typing 'python3 driver.py --help'

```

usage: driver.py [-h] [--initApp INITAPP] [--quitApp QUITAPP] [--parallelTest PARALLELTEST] [--htmlReport HTMLREPORT] [--captureSysLog CAPTURESYSLOG] [--reruns RERUNS]

optional arguments:

-h, --help show this help message and exit

--initApp INITAPP Set True if need to start messenger container. Default is False.

--quitApp QUITAPP Set True if need to stop messenger container. Default is False.

--parallelTest PARALLELTEST

Number of tests to run in parallel. Default is 5.

--htmlReport HTMLREPORT

Set True if need to stop messenger container. Default is False.

--captureSysLog CAPTURESYSLOG

Set False if logger logs and console logs not required in html report. Default is True.

--reruns RERUNS Number of time failed tests to rerun. Default is 1.

```

**Example**: If you want to execute tests, 5 tests parallel at same time, capture logs and html report should be generated with 'report.html' name

**Command**: python3 driver.py --parallelTest=5 --reruns=1 --htmlReport=report.html --captureSysLog=True

## Prerequisite environment

- The `docker-engine` is required to be installed on your system.

- Python 3.8 and above

## Automated test cases:

1. Smoke test to test, 2 users can login to app, user2 can see messages sent by user1 and both user can logout successfully

2. Login feature test

- Valid login

- Invalid login

- Blank credentials

## Html reporting

Html report with default name 'report.html' will be generated at the end of test case execution. This report will automatically open in the default browser.

Sample report:

