

This is a test project by playwright and robot framework with page object model.

You can find test scenarios in tests/question\_page.robot. and find pages in pages folder (pages/Question page)

There is vars folder that you can find configs.py in it, it contains configs of project. You can set configs by three ways: environment variable, the command that you run tests with or set in the configs file.

And there is Base.py file in libs folder that has general functions that pages need to use.

You can install the robot framework plugin from here:

<https://plugins.jetbrains.com/plugin/16382-hyper-robotframework-support>

### Installation:

```
Pip install requirments.txt -r
```

### Running:

For running tests, you have many options such as sequential, parallel, by tags, etc.

For example:

- **Parallel execution with tags:**

```
pabot --testlevelsplrit --processes 8 --pythonpath . -d results -o Output.xml --include textsORrequired . .
```

--testlevelsplrit: splits the suits by test cases

--processes 8: number of threads to run in parallel

--include textsORrequired : just run tests that contain tags "texts" or "required"











- **Parallel execution with all tests:**

```
pabot --testlevelsplrit --processes 8 --pythonpath . -d results -o Output.xml . .
```

- **specific test case execution:**

```
python.exe -m robot.run --test "qa-engineer-assignment.tests.question_page.check static texts" .
```

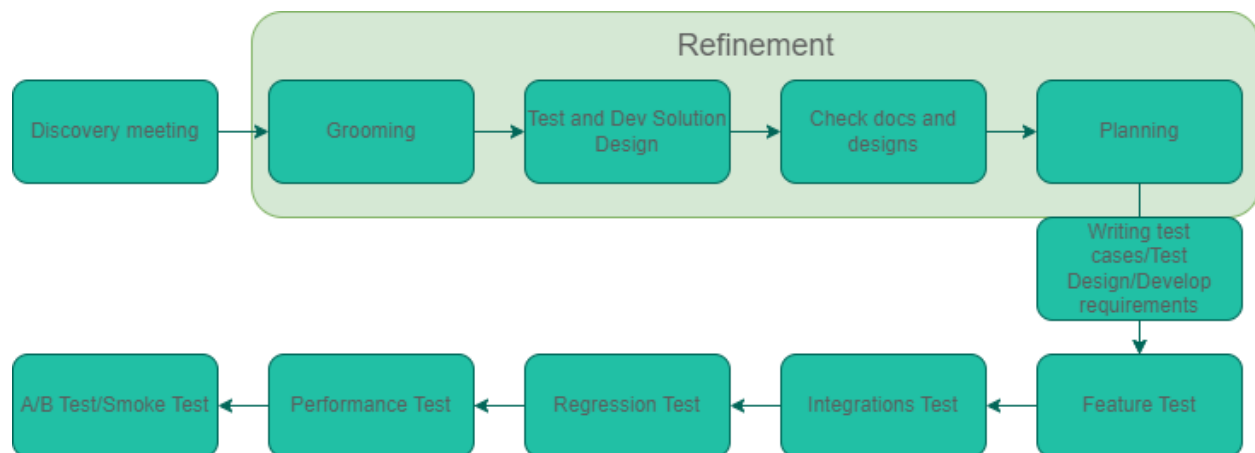
You can find the results in the results folder with two formats, XML and html.

Test Statistics							
Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip	
All Tests	11	8	3	0	00:01:27		
Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip	
Answer	2	1	1	0	00:00:15		
answers	1	0	1	0	00:00:10		
questions	5	3	2	0	00:00:57		
required	2	1	1	0	00:00:39		
sidebar	1	1	0	0	00:00:05		
sort	4	3	1	0	00:00:23		
studocu	11	8	3	0	00:01:27		
texts	2	2	0	0	00:00:08		
visibility	2	1	1	0	00:00:15		

As you can see the result of this project is 3 fail and 8 pass with 1:27 elapsed time.

# Insights:

- things to do before start the test process
  - Investigate about the technology that you should test.
  - Investigate about how you can automate it or is it automatable?
  - To develop automated test, consider the ROI and the maintenance cost
  - Make sure the solution that you choose is compatible with repeatable tests and doesn't affect the other tests.



1. **Discovery Meeting:** At the first step, QA engineers are participating in Discovery meetings. In these meetings, the expectation is:
  - a. Understanding the new features
  - b. Thinking about them from different perspectives to find the dependencies or some points that PMs should consider to define better Acceptance Criteria
2. **Refinement:** In these meetings, we already read the documentation of the features in detail. In case of any issues or incomplete Acceptance criteria, we should raise the matter to PMs to resolve.  
Also giving ETA for each task should be done in planning sessions.
3. **Writing test cases/Test Design/Develop requirements:** When the sprint starts, QA engineers should start the test design and write the test cases (Based on the Product's documentation and domain dominance)
4. **Feature Testing:** At the end of the development and code review, the test cases are ready and feature testing is started.  
At this step we should consider these points:
  - a. All features should be tested before RC. After testing and fixing the bugs, the feature can be inserted into the RC.

- b. All features should be tested on different OS versions, different display sizes, different manufacturers, different browsers, etc.
- c. Getting approval from the design team.

**5. Integration test:**

After finishing feature tests, it's the time for integration testing. Checking the integrations between services.

**6. Regression test:**

Next step is the regression test. This step is consist of critical part of product plus the parts that may be affected. (Should be designed)

**7. Performance test:**

This part is not just for back-end services, it also should be about client performance. Memory, CPU, Network, etc.

**8. A/B Test/Smoke Test:**

And finally, the last step, we should enable the new feature flag for the specific users to make sure everything is ok and then enable it for all the users and also check the main functionality on the production.