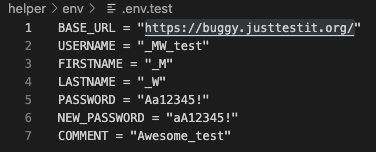
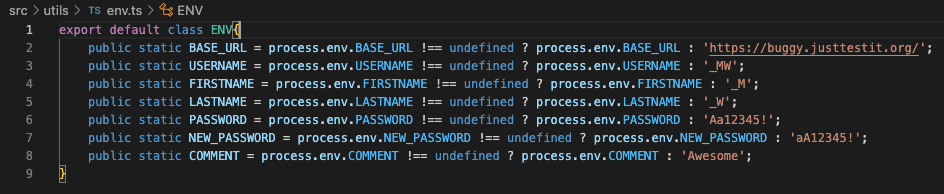
# Playwright buggy rating framework user guide

## How to add a new test case within this framework?

* 1. Add a child page object class which inherits from parent class Wrapper within folder /src/pages, add your config data into config file within folder helper/env/ and new vaiable into /src/utils/env.ts if needed.

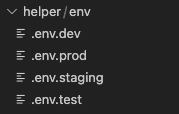




* 1. Add a facade class into /src/design\_patterns folder, you are able to implement actual test cases cross different page objects, just passing page object parameters into constructor function.
  2. Declare page object and facade variables within /src/tests/buggyrating.spc.ts, initize variables within test.beforeAll function, add a new test case within test.describe function.

## Multi environments running

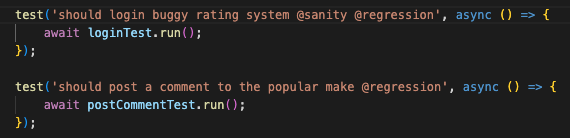
1. Config data against different environment are stored in corresponding config file within folder /helper/env



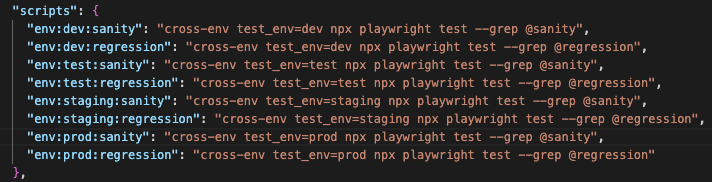
1. Command lines to run test cases against different environment are set within “scripts” field of package.json

## Organize test case by tags

1. You can organize test case by tags by adding tag name into test description, such as @sanity, @regression or @uat, of course, you can add multiple tags into a test case.



1. Only use cases with sanity tags are executed by add “--grep @sanity” into command line.



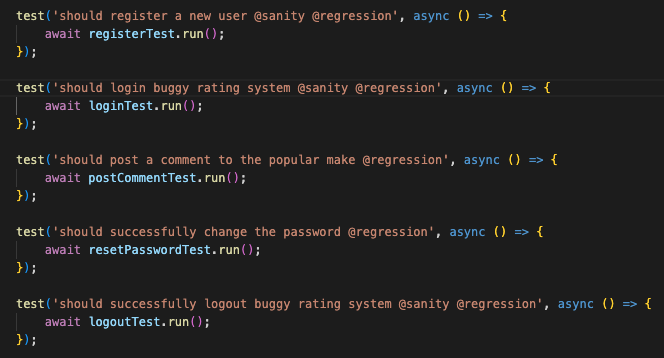
## Headless mode

1. You can change value of “headless” to “false” within file playwright.config.ts to shut down headless mode, which will be useful for code debugging. ￼



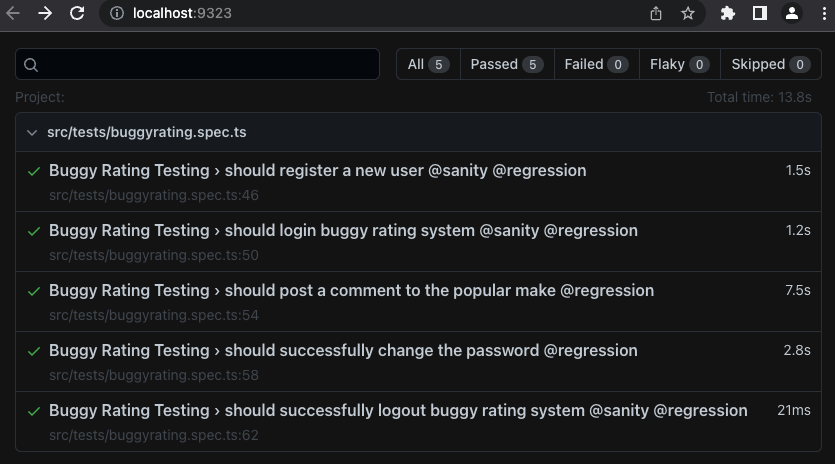
## Design Pattern

1. I use “facade” pattern within this framework to mask complex structural code which make the code of the spec file concise and clear



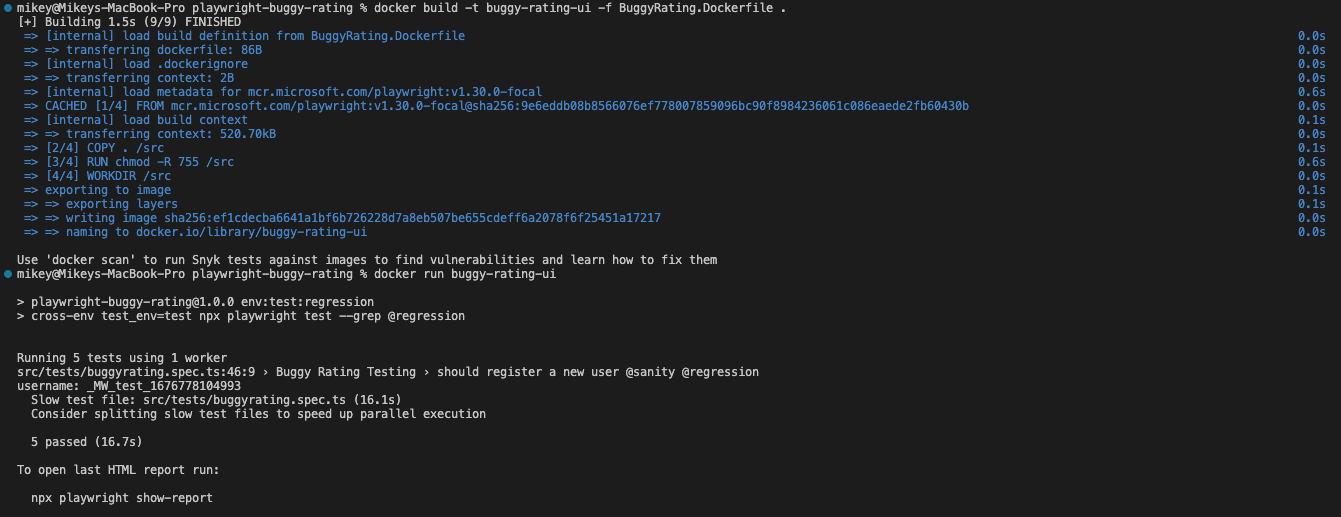
## View report

1. It’s easy to get report by running command " npx playwright show-report ", since the report is close to natural language, so it can be easily understood by non-technical stackholders and it is no need to apply separate BDD infrastructure.



## CI support

1. I integrate this framework with container running which make it easy to integrate framework to pipeline.



## Username and comment data naming tips

1. I attach timestamp to the end of username and comment which make it easier to track down issues, especially within pipeline.