ABN AMRO UI Automation (E2E Tests)

Approach

- 1. I have used a hybrid framework which includes Playwright + JavaScript.
- 2. Benefits of this is it is very easy to read Scenario flow and easy to maintain.

Library used to automate Front End

- 1. I have chosen Playwright as a JS based framework to automate UI actions.
- 2. Playwright is an open source framework to automate user actions.
- 3. Playwright is compatible with a lot of different npm modules.
- 4. Playwright directly injects HTML for execution so it's fast in terms of performance.

Programming language

- 1. I have chosen JavaScript as a programming language.
- 2. Javascript is a modern world programming language widely used for modern web application development.
- 3. We can use the same programming language for UI as well as API Automation.

How to run

- 1. Clone the entire project on a local machine from GitHub.
- 2. Execute npm install to install all required packages from package.json
- 2. Execute npm command from command-line as per below screen-shot.

```
npx playwright show-report

PS C:\Users\Shrikant\eclipse-workspace\abnamrouitests> npx playwright test

Running 7 tests using 2 workers

Slow test file: [chrome] > loginusingui.spec.js (21s)

Slow test file: [chrome] > loginusingcookie.spec.js (18s)

Consider splitting slow test files to speed up parallel execution

7 passed (24s)

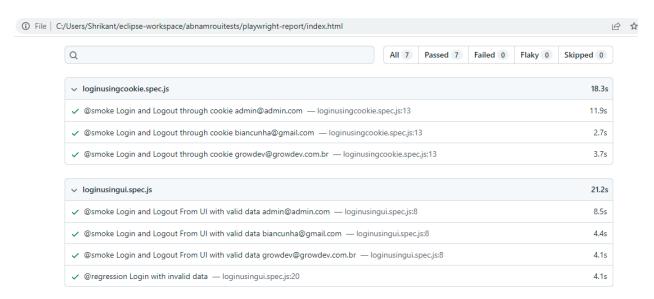
To open last HTML report run:

npx playwright show-report

PS C:\Users\Shrikant\eclipse-workspace\abnamrouitests> npx playwright test
```

Reporting

2. HTML Reporting



Debugging -

1. Assertions / Video recording / Screen shots on failures.

Framework features -

- 1. Execute Automation scripts in **headless mode** for faster execution.
- 2. Implemented **parallel execution** to reduce overall execution time.
- 3. Integrate Automation with Jenkins for continuous integration.
- 4. Pull code at runtime from **GitHub by Jenkins**, execute, and send emailable reports to relevant audiences.
- 5. Create **Automation dashboard** to monitor 24*7 Jenkins jobs status.
- 6. Store automation reports on cloud ex S3 bucket in AWS.
- 7. Manage most of Automation **configuration as command-line** arguments.
- 8. **Separate Object Repository**/Test Data on Cloud or any other Third Party tool for less maintenance.
- 9. Add Browser Compatibility feature to execute scripts on different browsers/OS etc.
- 10. Execute same scripts on different environments ex Integration/Staging/Preview/Production etc.
- 11. Integrate Automation with **different Third party tools** ex Slack for notification purpose / Integrate with JIRA to update automation results.