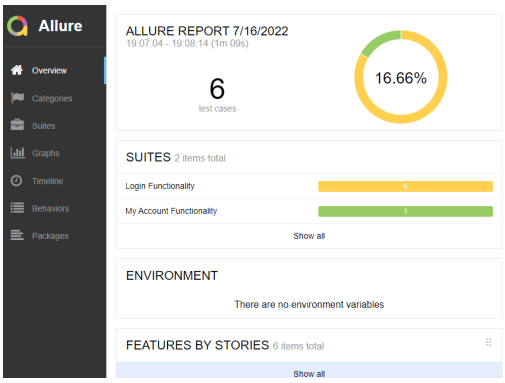
Issue #1: Allure report

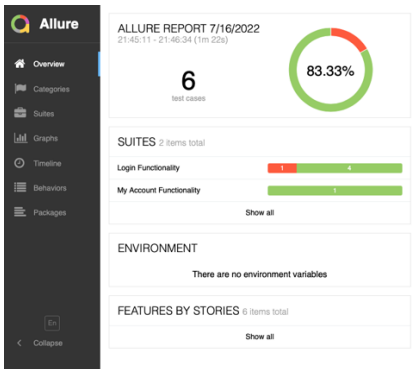
The README.md file states that all tests pass successfully (see Screenshot 1). However, when opening the Allure report, one test has actually failed.

Error message:
`Timed out retrying after 5000ms: expected '<li>' to have text 'Invalid email addressssss.', but the text was 'Invalid email address.'`
Reason: The expected error message in the assertion does not match the actual one.

Recommendation:

- Fix the test data if the extra "s" was a mistake.
- Ensure the Allure report link is reviewed before confirming test status in README.md.





cypressAllure/cypress/e2e/tests/login.test.ts

```
it('login with wrong email format credentials read data from fixture', function () {
    loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email
    loginPage.validateLoginError('Invalid email addressssss.')
})
```

Issue #2: Config File

1. Config.config.ts repeats the word "config".
2. Config file isn't located in the root directory.
3. Cypress already has built-in timeouts, so you don't have to use them all unless you have specific use cases.

```
defaultCommandTimeout: 5000,
execTimeout: 5000,
taskTimeout: 5000,
pageLoadTimeout: 30000,
requestTimeout: 5000,
responseTimeout: 30000,
```

Recommendation:

- Rename the file.
- Config file should be in root directory if it's not a custom config.
- If you don't have a reason to override all the timeouts, reduce to just what you need.

Issue #3: .gitignore

1. cypressAllure/docs wasn't added to .gitignore

Issue #4: scripts section in package.json

1. "cy:e2e:run" - the name suggests a simple Cypress test run, but it also generates and opens a report. Rename for clarity.
2. "cy:open" / "cy:run" - your config already defines env: { allureResultsPath: ... }, there's no need to pass --env again.
3. "clear" - it's not clearing— it's removing screenshots, videos, and allure-results. Rename for clarity.
4. In scripts npx is not necessary.

Issue #5: Pages

In all Pages move the text to a file like textConstants.ts

myAccountPage.ts

1. Recommend renaming to logoutLink for clarity and consistency.
   get signoutLink() { return cy.get('.logout') }
2. validateSuccessfulLogin() and validateSuccessfulLogout() – assertions should be in specs.

Issue #5: Tests

1. All tests don't have any assertions.

login.test.ts

2. Both those tests do the same verification (first with hardcoded values, second with fixture). Recommended to use the second one.

```
it('login with valid credentials', function () {
    loginPage.login("testautomation@cypresstest.com", "Test@1234")
    myAccountPage.validateSuccessfulLogin()
    myAccountPage.logout()
    myAccountPage.validateSuccessfulLogout()
})
it('login with valid credentials read data from fixture', function () {
    loginPage.login(this.data.valid_credentials.emailId, this.data.valid_credentials.password)
    myAccountPage.validateSuccessfulLogin()
    myAccountPage.logout()
    myAccountPage.validateSuccessfulLogout()
})
```

3. Tests contains hardcoded error messages, that should be moved to support folder like errorTexts.ts.

```
it('login with invalid email credentials read data from fixture', function () {
    loginPage.login(this.data.invalid_credentials.invalid_email.emailId,
        this.data.invalid_credentials.invalid_email.password)
    loginPage.validateLoginError('Authentication failed.')
})
it('login with invalid password credentials read data from fixture', function () {
    loginPage.login(this.data.invalid_credentials.invalid_password.emailId,
        this.data.invalid_credentials.invalid_password.password)
    loginPage.validateLoginError('Authentication failed.')
})
it('login with wrong email format credentials read data from fixture', function () {
    loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.
    loginPage.validateLoginError('Invalid email addresssss.')
})
```

users.json

4. Fixture “users.json” contains the same invalid data that may cause the pesticide paradox. Recommended to use random generated data.
5. Existing fixture too complicated, it’s better to avoid duplications and keep code dry.

You have: Duplicated data that cause the mistake in “wrong_email_format” that has invalid password.

```
{
    "valid_credentials": {
        "emailId": "testautomation@cypresstest.com",
        "password": "Test@1234"
    },
    "invalid_credentials": {
        "invalid_email": {
            "emailId": "invalidUser@cypresstest.com",
            "password": "Test@1234"
        },
        "invalid_password": {
            "emailId": "testautomation@cypresstest.com",
            "password": "test@12345"
        },
        "wrong_email_format": {
            "emailId": "testautomationresstest.com",
            "password": "test@12345"
        }
    }
}
```

Recommended: no duplications, use the same naming for valid / invalid data (without “wrong”).

```
{
    "valid_credentials": {
        "emailId": "testautomation@cypresstest.com",
        "password": "Test@1234"
    },
    "invalid_credentials": {
        "invalid_email": "invalidUser@cypresstest.com",
        "invalid_password": "test@12345",
        "invalid_email_format": "testautomationresstest.com"
    }
}
```

myAccount.test.ts

6. cy.visit('https://google.com') in the beforeEach() block is unrelated to the application. The correct method is hidden behind the comments.

```
import { loginPage } from "../pages/loginPage";

describe('My Account Functionality', () => {
    beforeEach(() => {
        cy.visit('https://google.com');
        //loginPage.launchApplication()
    })
```

7. Instead of verification used the "console log".

```
    it('Sample Test', () => {
        console.log("This is a sample test")
    })
})
```

Issue #6: Code is not formated.

The repository contains unnecessary comments.

Issue #7: Mixed usage of JavaScript and TypeScript files

Convert plugins/index.js, support/e2e.js and commands.js to TypeScript.

Issue #8: README.md

1. Missing Repo Requirements
2. Steps to install located in Steps to run
3. Steps to run has 1 option - npm run cy:run but in package.json there are many options.

Issue #9: .env

In the login test file, the email and password are hardcoded directly in the code and used in fixture. If these credentials are sensitive or should not be visible publicly, it's a bad practice to store them this way.

Instead, the credentials should be stored in an .env file.

The project should also include a .env.example file in the repository. The actual .env file should be added to .gitignore and not committed to version control.