

# Code Review of Cypress + Allure Project

## 1. Critical Issues

### 1.1. Test Website Not Working

The domain *automationpractice.com* no longer hosts a demo store and redirects to a standard InMotion Hosting page.

All automated tests using this site as the base URL fail because the page lacks elements needed for testing (e.g., login, registration, catalog).

Tests won't pass until the test website is changed.

### 1.2. Wrong Error Message Check in Login Test

In *login.test.ts*, the test "login with wrong email format credentials read data from fixture" expects the error message 'Invalid email addresssssss.' but the actual message is 'Invalid email address.'

This causes the test to fail even if the system works correctly.

### 1.3. Tests Depend Too Much on users.json Structure

Tests directly use fields in *users.json* (e.g., `this.data.valid_credentials.emailId`).

If the structure of *users.json* changes (e.g., keys are renamed), all tests will fail.

### 1.4. Incorrect beforeEach in myAccount.test.ts

The *beforeEach* uses `cy.visit('https://google.com')` instead of opening the tested application.

Tests don't check real functionality and just open an unrelated website.

## 2. Other Issues

- In *cypress/support/commands.js*, there are no custom commands, only template comments. The *commands.js* file is meant for creating custom Cypress commands to simplify and standardize repeated actions. An empty file with only comments means custom commands are either not used or not maintained.
- No *afterEach* to clean up state (e.g., logout after each test). Without cleaning up (e.g., not logging out), the next test may start in an incorrect state (e.g., user already logged in, session not cleared).
- In *myAccountPage.ts*, there's a dependency on *loginPage* (`loginPage.signinLink.should('be.visible')`), linking Page Object Model (POM) objects. POMs should be independent: each class should describe only its own page. If one POM uses elements from another, it creates a strong dependency. Changes in one POM can break another.
- *README.md* lacks full setup instructions (no *git clone* example, no details on installing Allure dependencies, no explanation of where to change test data).

*README.md* is the first thing a new team member or user reads. Missing instructions make it hard to set up, run, and maintain the project.

### 3. Recommendations

- Switch to a working demo store and update the base URL in Cypress.
- Split the large *login.test.ts* file into smaller tests based on scenario types or functionality.
- Consider generating random test data in tests for negative scenarios.  
This avoids duplication and dependency on fixture structure, and helps find more bugs related to validation (e.g., uniqueness, format).
- For positive scenarios or complex cases, use fixtures or create test data via API before running tests.
- Fix the expected error message in the login test to 'Invalid email address.'
- Document the *users.json* structure and add checks for data existence in tests.
- Update *beforeEach* in *myAccount.test.ts* to correctly start tests.
- Update *README.md*: add a *git clone* example, instructions for installing Allure CLI, and examples for running tests and generating reports.
- Make POMs independent of each other.

### 4. Summary

The most critical issues are the unavailable test website, incorrect error message check, heavy dependency on *users.json* structure, and wrong *beforeEach* in tests. Without fixing these, tests won't work or will give false results.

The project uses Page Object Model and integrates with Allure but needs fixes for critical issues, updated test data, better documentation, and improved test structure for better quality and easier maintenance.