

Part 2A: Code Review & Debugging

- **Issue Category:** Reliability
- **Severity:** Critical
- **Description:** Use of `cy.wait()` multiple times in the test. If app loads for a shorter amount of time, the test waits unnecessarily and if the app takes longer test fails.
- **Improved Code:** will wait as long as the server takes to respond
 - `cy.intercept('POST', '/api/login').as('loginRequest');`
`cy.get('button[type="submit"]').click();`
`cy.wait('@loginRequest');`
- **Issue Category:** Maintainability
- **Severity:** High
- **Description:** Use of `cy.get('button')` is not specific and the first button on the page will be selected, if button order changes test will break. Same with `cy.get('#question1')`, if the content of questions is switched it's no longer selecting the correct question.
- **Improved Code:** Using specific test attributes or text that the user sees to select is more stable
 - `cy.get('[data-testid="login-btn"]').click();`
 - `cy.contains('button', 'Log in').click();`
- **Issue Category:** Best Practice
- **Severity:** Medium
- **Description:** The login steps are inside the it block. If you add a second test case, you have to copy-paste the login code.
- **Improved Code:** Put login steps outside it block

```
describe('Post-Surgery Symptom Survey', () => {
  beforeEach(() => {
    cy.visit('/login');
    //login here...
  });
  it('should complete survey...', () => { ... })
});
```
- **Issue Category:** Reliability
- **Severity:** Medium
- **Description:** Test clicks elements immediately without checking if they are actionable. Submit button might be disabled until all questions are answered, leading to failure
- **Improved Code:** Check that the button is enabled first

```
cy.contains('button', 'Submit').should('not.be.disabled').click();
```