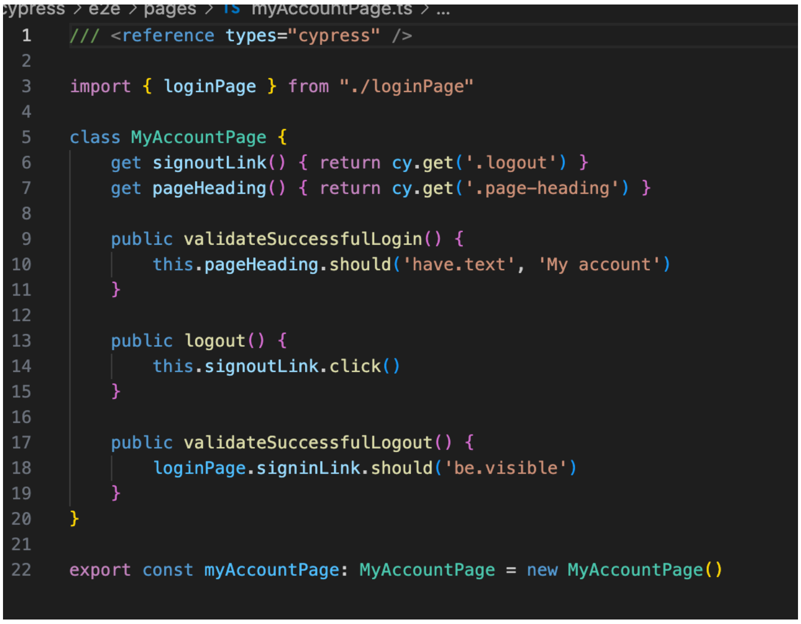
1. For testing fields using always the same testing data. Can use faker-js to generate invalid credentials

2. The “allure-results”, “docs” folders does not added to .gitignore so repo has many redundant files

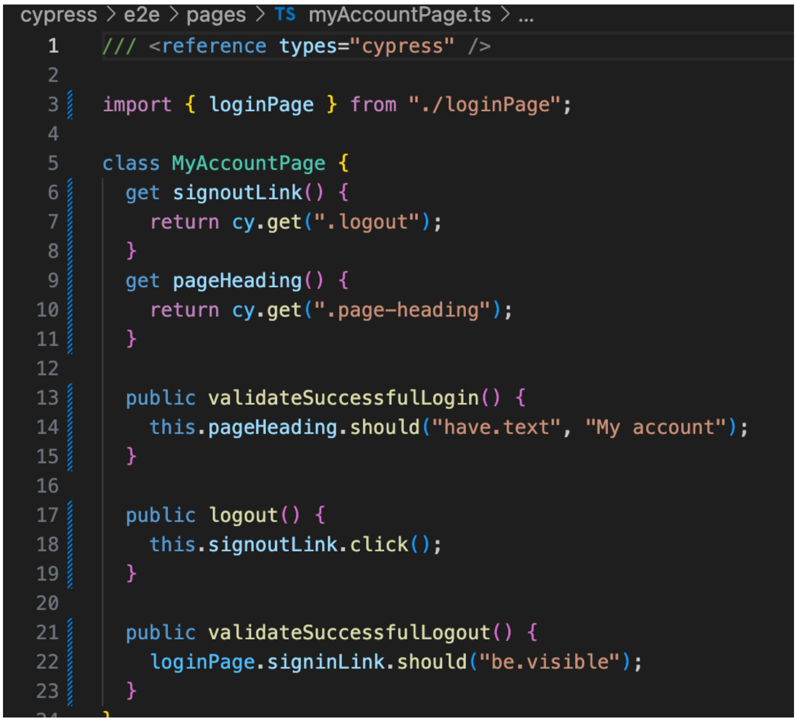
3. Code not formatted:

- extra spaces here



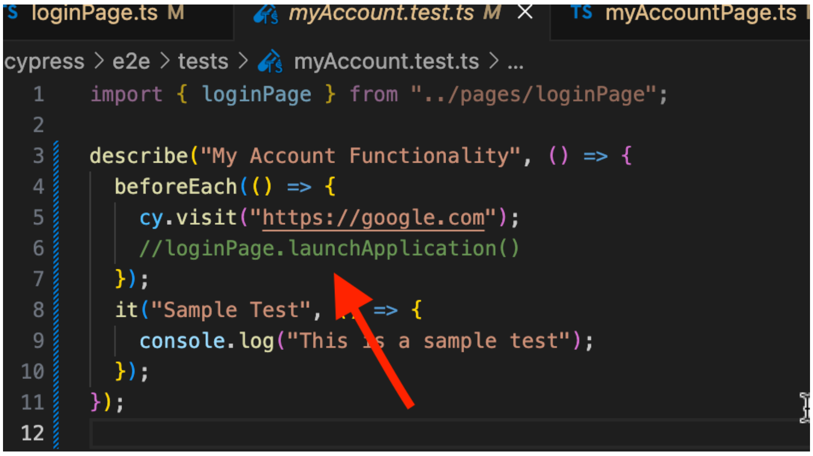
```
/// <reference types="cypress" />

import { loginPage } from "./loginPage"

class MyAccountPage {
    get signoutLink() { return cy.get('.logout') }
    get pageHeading() { return cy.get('.page-heading') }

    public validateSuccessfulLogin() {
        this.pageHeading.should('have.text', 'My account')
    }

    public logout() {
        this.signoutLink.click()
    }

    public validateSuccessfulLogout() {
        loginPage.signinLink.should('be.visible')
    }
}

export const myAccountPage: MyAccountPage = new MyAccountPage()
```

how looks formatted code

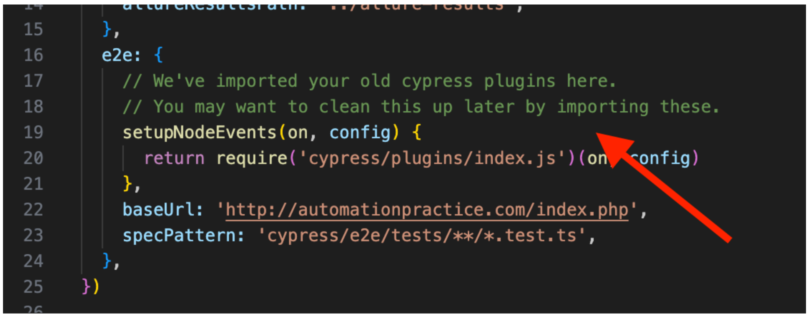


```
/// <reference types="cypress" />

import { loginPage } from "./loginPage";

class MyAccountPage {
  get signoutLink() {
    return cy.get(".logout");
  }
  get pageHeading() {
    return cy.get(".page-heading");
  }

  public validateSuccessfulLogin() {
    this.pageHeading.should("have.text", "My account");
  }

  public logout() {
    this.signoutLink.click();
  }

  public validateSuccessfulLogout() {
    loginPage.signinLink.should("be.visible");
  }
}
```

- entries, commented part of the code

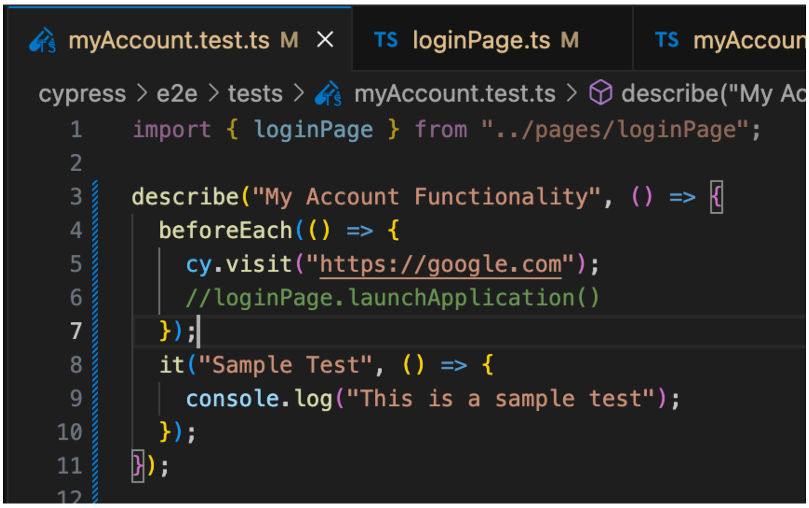


```
import { loginPage } from "../pages/loginPage";

describe("My Account Functionality", () => {
  beforeEach(() => {
    cy.visit("https://google.com");
    //loginPage.launchApplication()
  });
  it("Sample Test", () => {
    console.log("This is a sample test");
  });
});
```



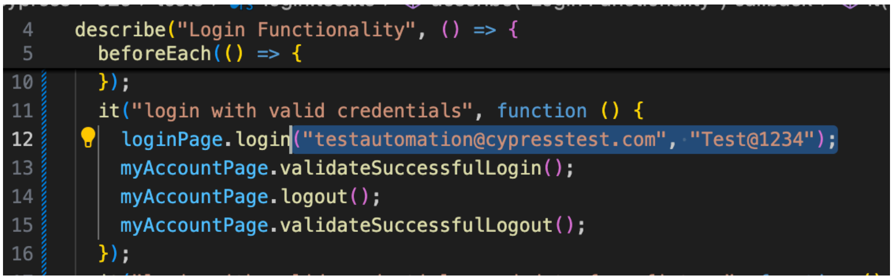
```
      allureResultsPath: '../allure-results',
    },
    e2e: {
      // We've imported your old cypress plugins here.
      // You may want to clean this up later by importing these.
      setupNodeEvents(on, config) {
        return require('cypress/plugins/index.js')(on, config)
      },
      baseUrl: 'http://automationpractice.com/index.php',
      specPattern: 'cypress/e2e/tests/**/*.test.ts',
    },
  })
```

4. Instead of verification used the "console log"



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

describe("My Account Functionality", () => {
  beforeEach(() => {
    cy.visit("https://google.com");
    //loginPage.launchApplication()
  });
  it("Sample Test", () => {
    console.log("This is a sample test");
  });
});
```

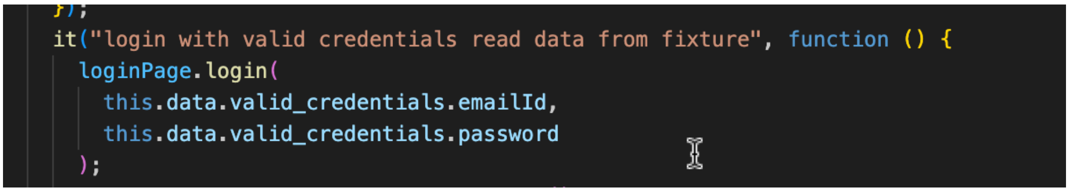
5. The login credentials were hardcoded directly in the test instead of being retrieved from the fixtures file



```
describe("Login Functionality", () => {
  beforeEach(() => {
  });
  it("login with valid credentials", function () {
    loginPage.login("testautomation@cypresstest.com", "Test@1234");
    myAccountPage.validateSuccessfulLogin();
    myAccountPage.logout();
    myAccountPage.validateSuccessfulLogout();
  });
```

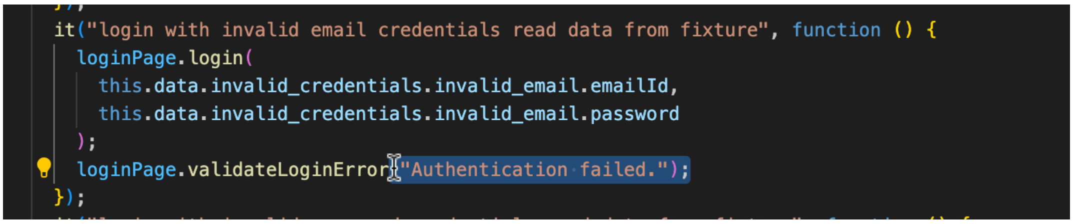
6. All tests doesn't use POM principles, takes data from fixtures in test, instead of take data in login method on LoginPage

like in this example



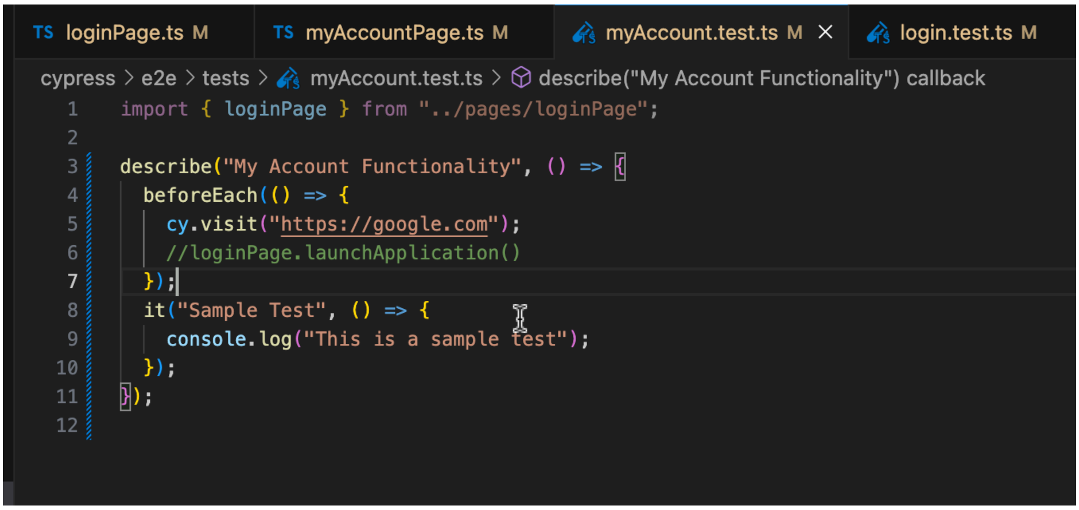
```
});
it("login with valid credentials read data from fixture", function () {
  loginPage.login(
    this.data.valid_credentials.emailId,
    this.data.valid_credentials.password
  );
```

7. Displaying an error message in tests instead of making a variable in the loginPage



```
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.");
});
```

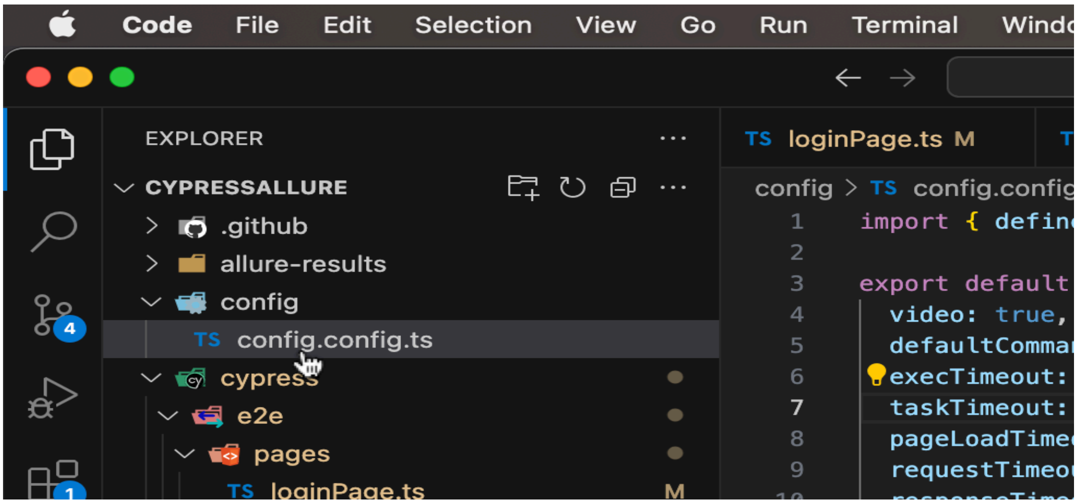
8. File "myAccount.test.ts" has no functionality for this project, looks like example file



TS loginPage.ts M
TS myAccountPage.ts M
myAccount.test.ts M ×
login.test.ts M
cypress > e2e > tests > myAccount.test.ts > describe("My Account Functionality") callback

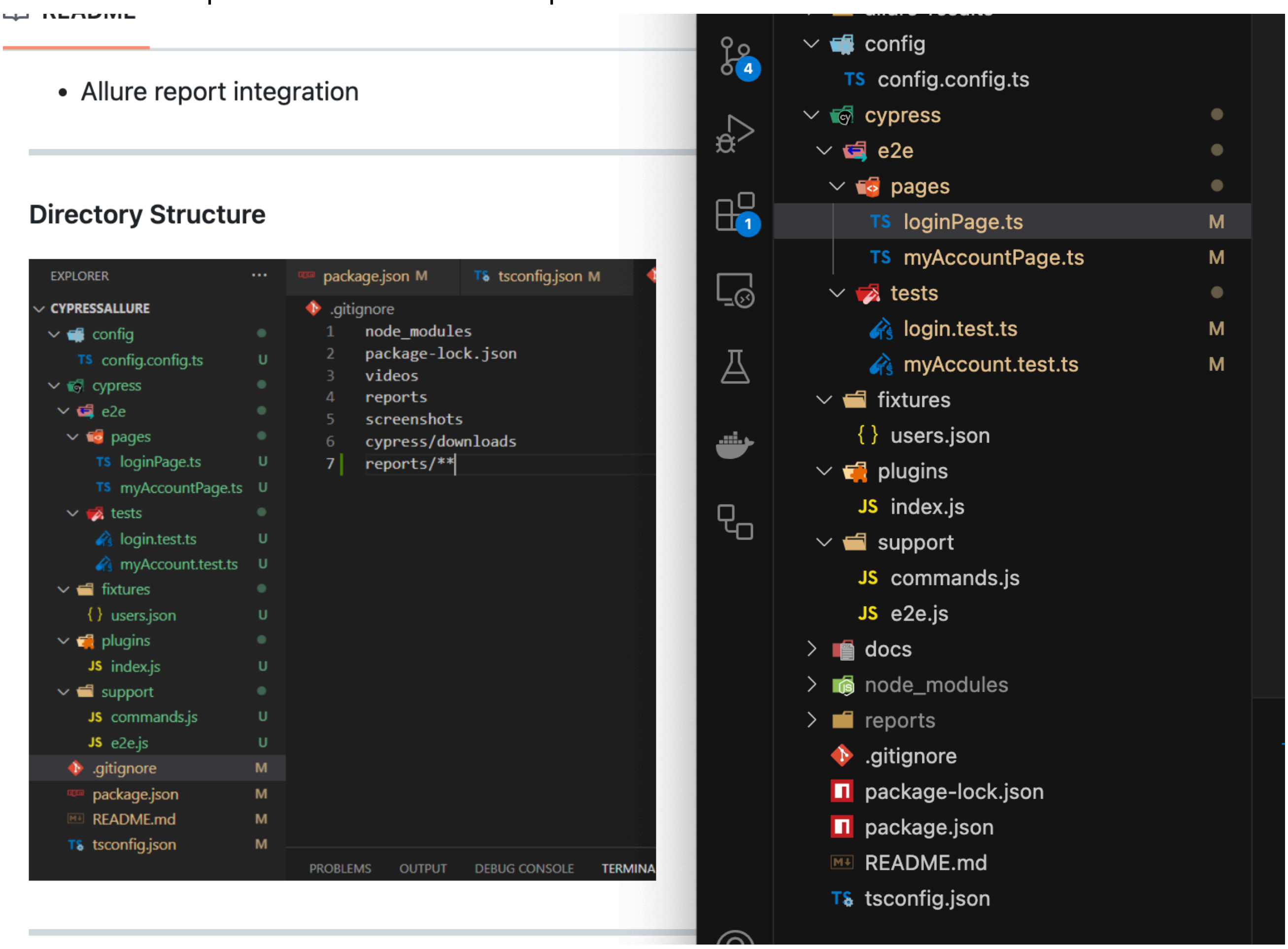
```
import { loginPage } from "../pages/loginPage";

describe("My Account Functionality", () => {
  beforeEach(() => {
    cy.visit("https://google.com");
    //loginPage.launchApplication()
  });
  it("Sample Test", () => {
    console.log("This is a sample test");
  });
});
```

9. The name of the config file is incorrect - must be "cypress.config.ts" and its location is not correct must be located at the root of the project



```
import { defin

export default
  video: true,
  defaultComma
  execTimeout:
  taskTimeout:
  pageLoadTime
  requestTimeo
```

10. Directory Structure not like in ReadMe file
for example folders “docs” are not specified



README

- Allure report integration

Directory Structure

```
EXPLORER
CYPRESSALLURE
  config
    config.config.ts        U
  cypress
    e2e
      pages
        loginPage.ts        U
        myAccountPage.ts    U
      tests
        login.test.ts       U
        myAccount.test.ts   U
    fixtures
      users.json            U
    plugins
      index.js              U
    support
      commands.js           U
      e2e.js                U
  .gitignore                M
  package.json              M
  README.md                 M
  tsconfig.json             M
```

package.json M   tsconfig.json M

.gitignore

```
node_modules
package-lock.json
videos
reports
screenshots
cypress/downloads
reports/**
```

```
  config
    config.config.ts
  cypress
    e2e
      pages
        loginPage.ts        M
        myAccountPage.ts    M
      tests
        login.test.ts       M
        myAccount.test.ts   M
    fixtures
      users.json
    plugins
      index.js
    support
      commands.js
      e2e.js
  docs
  node_modules
  reports
  .gitignore
  package-lock.json
  package.json
  README.md
  tsconfig.json
```