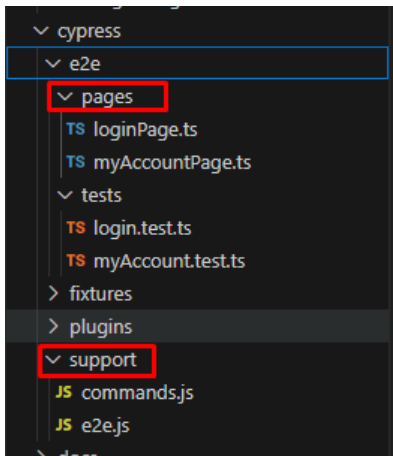
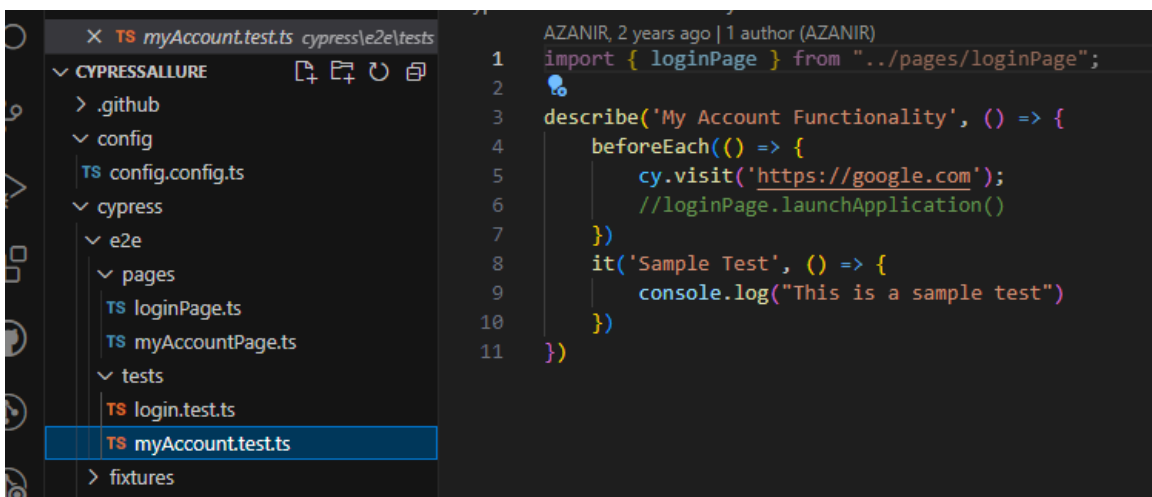


# Task 5 - Code review

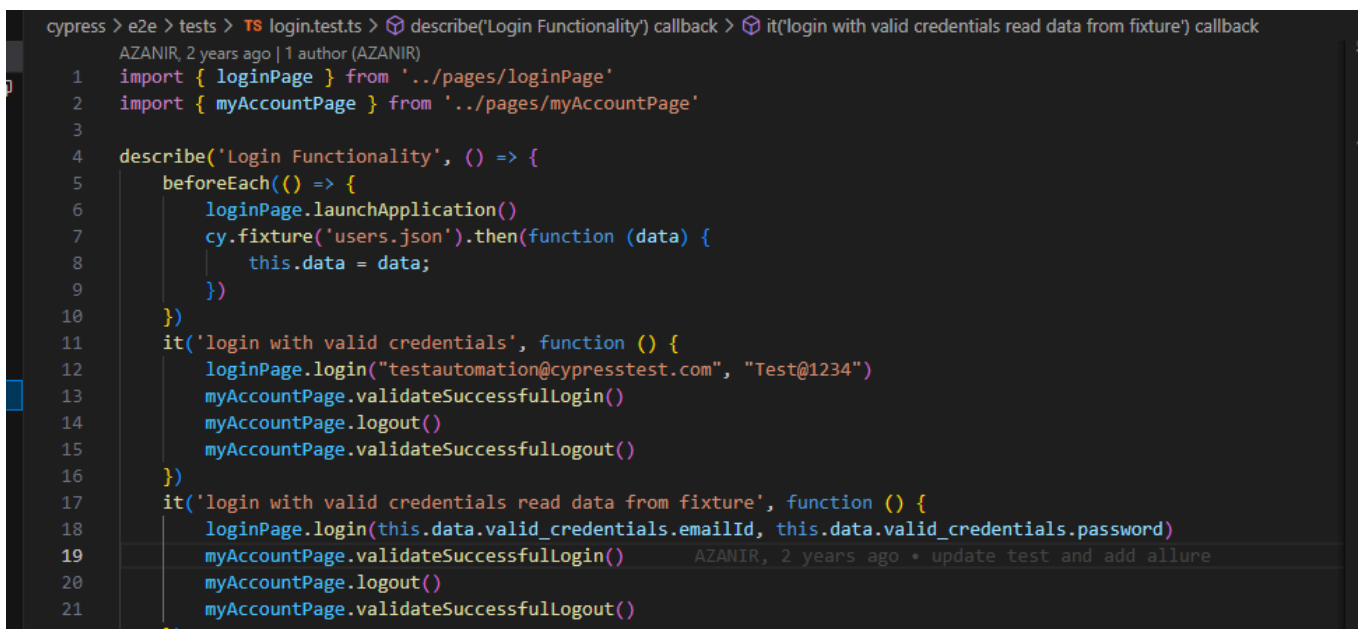
- 'Pages' folder should be on the same root level as the 'support' folder, or in the 'support' folder



- myAccount.test.ts is a sample file, not needed in repository



- better practice is to have IDs in the names of test cases



- the title of the test cases will not match the documentation, as it states “data from fixture”

```

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()
})

```

- using same test data for invalid login is not a good practice

```

1  {
2    "valid_credentials": {
3      "emailId": "testautomation@cypressstest.com",
4      "password": "Test@1234"
5    },
6    "invalid_credentials": {
7      "invalid_email": {
8        "emailId": "invalidUser@cypressstest.com",
9        "password": "Test@1234"
10     },
11     "invalid_password": {
12       "emailId": "testautomation@cypressstest.com",
13       "password": "test@12345"
14     },
15     "wrong_email_format": {
16       "emailId": "testautomationresstest.com",
17       "password": "test@12345"
18     }
19   }
20 }

```

- it is better to use “eq” to validate error message, as it may also contain not valid text

```

public validateLoginError(errorMessage: string) {
  this.alertBox.should('be visible')
  this.alertMessage.should('have.text', errorMessage)
}

```

- is it correct to use methods of other .page file in current .page file?

```
...  <  >  cypressAllure

TS login.test.ts  TS loginPage.ts  TS myAccountPage.ts X

cypress > e2e > pages > TS myAccountPage.ts > [?] myAccountPage
  AZANIR, 2 years ago | 1 author (AZANIR)
1  /// <reference types="cypress" />
2
3  import { loginPage } from "../loginPage"
4
  AZANIR, 2 years ago | 1 author (AZANIR)
5  class MyAccountPage {
6    get signoutLink() { return cy.get('.logout') }
7    get pageHeading() { return cy.get('.page-heading') }
8
9    public validateSuccessfulLogin() {
10     |   this.pageHeading.should('have.text', 'My account')
11   | }
12
13   public logout() {
14     |   this.signoutLink.click()
15   | }
16
17   public validateSuccessfulLogout() {
18     |   loginPage.signinLink.should('be.visible')
19   | }
20 }
21
22 export const myAccountPage: MyAccountPage = new MyAccountPage()
```

- formatting is incorrect

AZANIR, 2 years ago | 1 author (AZANIR)

```
class LoginPage {
  get signinLink() { return cy.get('.login') }
  get emailAddressTxt() { return cy.get('#email') }
  get passwordTxt() { return cy.get('#passwd') }
  get signinBtn() { return cy.get('#SubmitLogin') }
  get alertBox() { return cy.get('p:contains("error")') }
  get alertMessage() { return cy.get('.alert-danger > ol > li') }

  public launchApplication() {
    cy.visit('/')
  }

  public login(emailId: string, password: string) {
    this.signinLink.click()
    this.emailAddressTxt.type(emailId)
    this.passwordTxt.type(password)
    this.signinBtn.click()
  }

  public validateLoginError(errorMessage: string) {
    this.alertBox.should('be.visible')
    this.alertMessage.should('have.text', errorMessage)
  }
}

export const loginPage: LoginPage = new LoginPage()
```

AZANIR, 2 years ago

cypress > e2e > pages > TS myAccountPage.ts > MyAccountPage > validateSuccessfulLogout

AZANIR, 2 years ago | 1 author (AZANIR)

```
1  /// <reference types="cypress" />
2
3  import { loginPage } from "../loginPage"
4
5  class MyAccountPage {
6    get signoutLink() { return cy.get('.logout') }
7    get pageHeading() { return cy.get('.page-heading') }
8  }
```