

Code Review

LoginPage.ts

```
1  /// <reference types="cypress" />
2
3  class LoginPage {
4    get signinLink() { return cy.get('.login') }
5    //Code Review: Don't use shortcuts for naming variables or methods, it is better to use full name (email Address Text, signInButton etc.)
6    get emailAddressTxt() { return cy.get('#email') }
7    get passwordTxt() { return cy.get('#passwd') } //Code Review: This selector looks like invalid one, the correct one should looks like '#password'
8    get signInBtn() { return cy.get('#SubmitLogin') } //Code Review: The camel case style was not followed in this naming, the correct one is signInButton
9    get alertBox() { return cy.get('p:contains("error")') }
10   //^Code Review: Avoid such selectors since in case the language of the website is changed it will be not possible to find such element^
11   get alertMessage() { return cy.get('.alert-danger > ol > li') }
12
13   public launchApplication() {
14     cy.visit('/')
15   }
16
17   public login(emailId: string, password: string) {
18     this.signinLink.click()
19     this.emailAddressTxt.type(emailId)
20     this.passwordTxt.type(password)
21     this.signInBtn.click()
22   }
23
24   public validateLoginError(errorMessage: string) {
25     this.alertBox.should('be.visible')
26     this.alertMessage.should('have.text', errorMessage)
27   }
28 }
29
30 export const loginPage: LoginPage = new LoginPage()
```

myAccountPage.ts

```
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') "My account" text should be put in a fixture
  }

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

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

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

login.test.ts

```
1 import { loginPage } from '../pages/loginPage'
2 import { myAccountPage } from '../pages/myAccountPage'
3
4 describe('Login Functionality', () => {
5   beforeEach(() => {
6     loginPage.launchApplication()
7     cy.fixture('users.json').then(function (data) {
8       this.data = data;
9     })
10  })
11  it('login with valid credentials', function () {
12    // Code Review: The data below should be moved to a fixtures
13    loginPage.login("testautomation@cypressstest.com", "Test@1234")
14    myAccountPage.validateSuccessfulLogin()
15    myAccountPage.logout()
16    myAccountPage.validateSuccessfulLogout()
17  })
18  // Code Review: don't need to use 'read data from fixture' text in the naming of the following tests
19  it('login with valid credentials read data from fixture', function () {
20    loginPage.login(this.data.valid_credentials.emailId, this.data.valid_credentials.password)
21    myAccountPage.validateSuccessfulLogin()
22    myAccountPage.logout()
23    myAccountPage.validateSuccessfulLogout()
24  })
25  // Code Review: 'login with valid credentials' and 'login with valid credentials read data from fixture' are doing the same thing, the first test should be removed
26
27  it('login with invalid email credentials read data from fixture', function () {
28    loginPage.login(this.data.invalid_credentials.invalid_email.emailId,
29      this.data.invalid_credentials.invalid_email.password)
30    loginPage.validateLoginError('Authentication failed.') //Code Review: 'Authentication failed.' error message should ne moved to fixtures
31  })
32  it('login with invalid password credentials read data from fixture', function () {
33    loginPage.login(this.data.invalid_credentials.invalid_password.emailId,
34      this.data.invalid_credentials.invalid_password.password)
35    loginPage.validateLoginError('Authentication failed.') //Code Review: 'Authentication failed.' error message should ne moved to fixtures
36  })
37  it('login with wrong email format credentials read data from fixture', function () {
38    loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email_format.password)
39    loginPage.validateLoginError('Invalid email addresssssss.') //Code Review: 'Invalid email addresssssss.' error message should ne moved to fixtures
40  })
41 })
```

myAccount.tets.ts

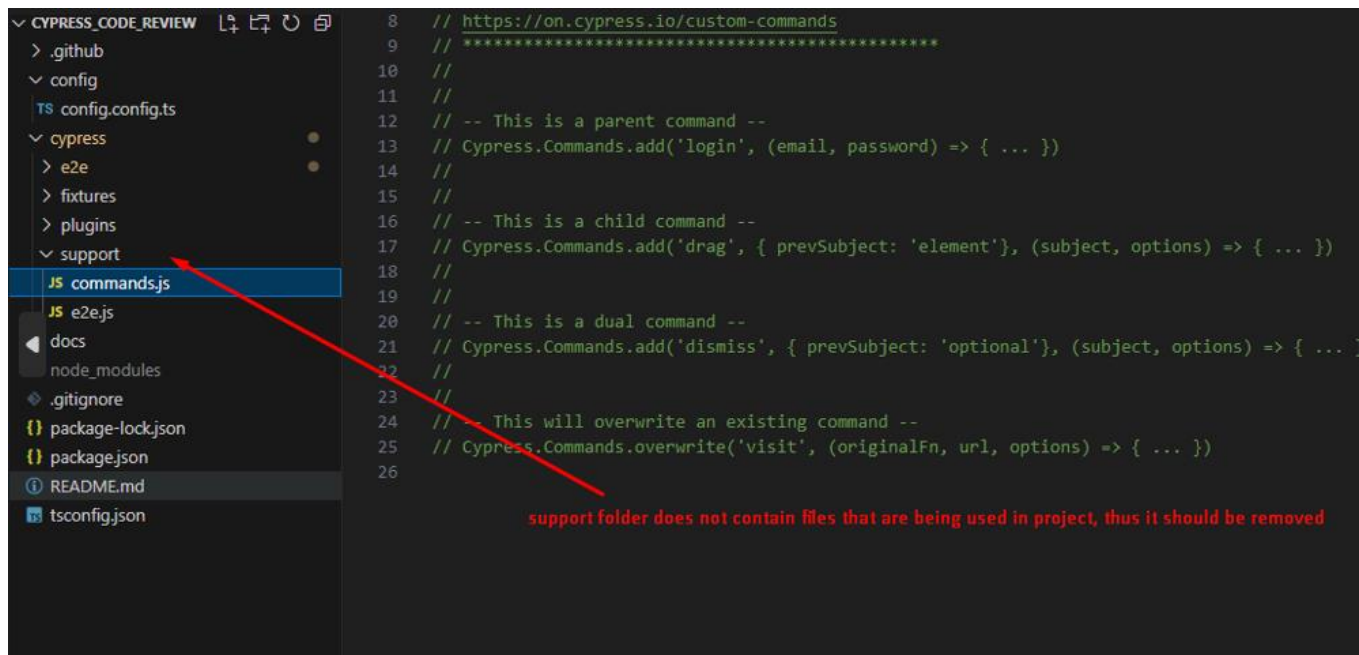
```
1 import { loginPage } from "../pages/loginPage"; // Code Review: this import is not used in this file, thus it can be removed
2
3 describe('My Account Functionality', () => {
4   // Code Review: beforeEach block can be removed since we have only one test here
5   beforeEach(() => {
6     cy.visit('https://google.com');
7     //loginPage.launchApplication() // Code Review: Your code should not contain commented code
8   })
9   it('Sample Test', () => {
10     console.log("This is a sample test")
11   })
12 })
```

users.json (fixtures)

```
{
  "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"
    }
  }
}
```

You should use only one style of naming variables.
- snake_case
- camelCase

'support' folder



The screenshot shows the VS Code interface with the Cypress project structure on the left and a code file on the right. The file explorer on the left shows the following structure:

- ▼ CYPRESS_CODE_REVIEW
 - > .github
 - ▼ config
 - TS config.config.ts
 - ▼ cypress
 - > e2e
 - > fixtures
 - > plugins
 - ▼ support
 - JS commands.js
 - JS e2e.js
 - ▼ docs
 - node_modules
 - ◆ .gitignore
 - { } package-lock.json
 - { } package.json
 - 📄 README.md
 - 📄 tsconfig.json

The code file on the right contains the following content:

```
8 // https://on.cypress.io/custom-commands
9 // *****
10 //
11 //
12 // -- This is a parent command --
13 // Cypress.Commands.add('login', (email, password) => { ... })
14 //
15 //
16 // -- This is a child command --
17 // Cypress.Commands.add('drag', { prevSubject: 'element'}, (subject, options) => { ... })
18 //
19 //
20 // -- This is a dual command --
21 // Cypress.Commands.add('dismiss', { prevSubject: 'optional'}, (subject, options) => { ... })
22 //
23 //
24 // -- This will overwrite an existing command --
25 // Cypress.Commands.overwrite('visit', (originalFn, url, options) => { ... })
26
```

support folder does not contain files that are being used in project, thus it should be removed