

# Code review

Git repository link: <https://github.com/dreamquality/cypressAllure>

Author: Oleksandr Karpovych

Mentor: Alex

pages folder

## loginPage.ts

```
you, 46 minutes ago | 2 authors (you and others)
3 class LoginPage {
4   get signinLink() {
5     return cy.get(".login");
6   }
7   //REVIEW: Don't use shortcuts in variable naming. Should be
  emailAddressText
8   get emailAddressTxt() {
9     return cy.get("#email");
10  }
11  //REVIEW: Probably incorrect name of selector. Should be
  "#password". Also shortcut in the name - better will be if
  "passwordText"
12  get passwordTxt() {
13    return cy.get("#passwd");
14  }
15  get signinBtn() {
16    return cy.get("#SubmitLogin");
17  }
18  get alertBox() {
19    return cy.get('p:contains("error")');
20    //REVIEW: Bad selector - 1. If site will use another
  language - element won't be found. 2. :contains()
  pseudo-class is not part of the official CSS
  specification, and it has limited support in modern web
  browsers. To get element by its text better to use
  xpath selector. But better don't select elements by
  text at all.
21 }
```

1)

## MyAccountPage.ts

```
//REVIEW: 'My account' text should be moved to fixtures
public validateSuccessfulLogin() {
  this.pageHeading.should("have.text", "My account");
}
```

1)

tests folder

login.test.ts

```
it.only("login with valid credentials", function () {
  loginPage.login("testautomation@cypressstest.com", "Test@1234");
  myAccountPage.validateSuccessfulLogin();
  myAccountPage.logout();
  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();
});
```

These two blocks are doing the same thing. Testing data should be putted in fixtures and first "it" block should be removed

1)

```
//don't need "read data from fixture" text. All testing data come from fixtures so that text should be omitted
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.");
});
```

2)

```
24 it("login with invalid email credentials read data from fixture", function () {
25   loginPage.login(
26     this.data.invalid_credentials.invalid_email.emailId,
27     this.data.invalid_credentials.invalid_email.password
28   );
29   loginPage.validateLoginError("Authentication failed.");
30 });
31 it("login with invalid password credentials read data from fixture", function () {
32   loginPage.login(
33     this.data.invalid_credentials.invalid_password.emailId,
34     this.data.invalid_credentials.invalid_password.password
35   );
36   loginPage.validateLoginError("Authentication failed.");
37 });
38 it("login with wrong email format credentials read data from fixture", function () {
39   loginPage.login(
40     this.data.invalid_credentials.wrong_email_format.emailId,
41     this.data.invalid_credentials.wrong_email_format.password
42   );
43   loginPage.validateLoginError("Invalid email addresssssss.");
44 });
45 });
46
47
```

Put these texts in fixture please

3)

myAccount.test.ts

1)

```
1 import { loginPage } from "../pages/loginPage";
2 //REVIEW 1: import is not usable by code below. So it'll be better to remove this import
3 describe("My Account Functionality", () => {
4   //REVIEW 2: If there is only one 'it' block then there is no need for beforeEach
5   beforeEach(() => {
6     //REVIEW 3: Final product should not contain commented code
7     cy.visit("https://google.com");
8     //loginPage.launchApplication()
9   });
10  it("Sample Test", () => {
11    //REVIEW 4: console.log uses only for developing mode. In production mode it should be absent
12    console.log("This is a sample test");
13  });
14 });
```

fixtures folder

user.json

1)

```
1 { Magnifikate, 14 months ago • Init boilerplate
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": "testautomationpresstest.com",
17      "password": "test@12345"
18    }
19 }
20 }
21
```

1. You should use only one style of naming variables.  
2. This style should be camelCase

It's a snake\_case, not a camelCase

camelCase