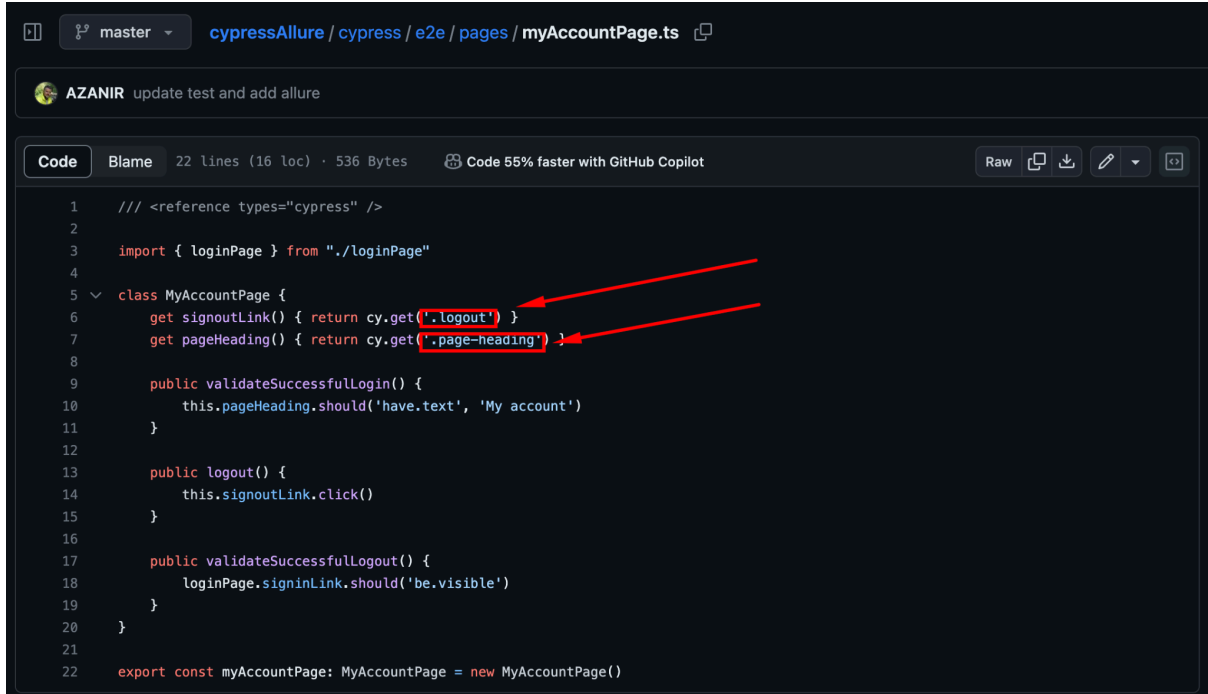
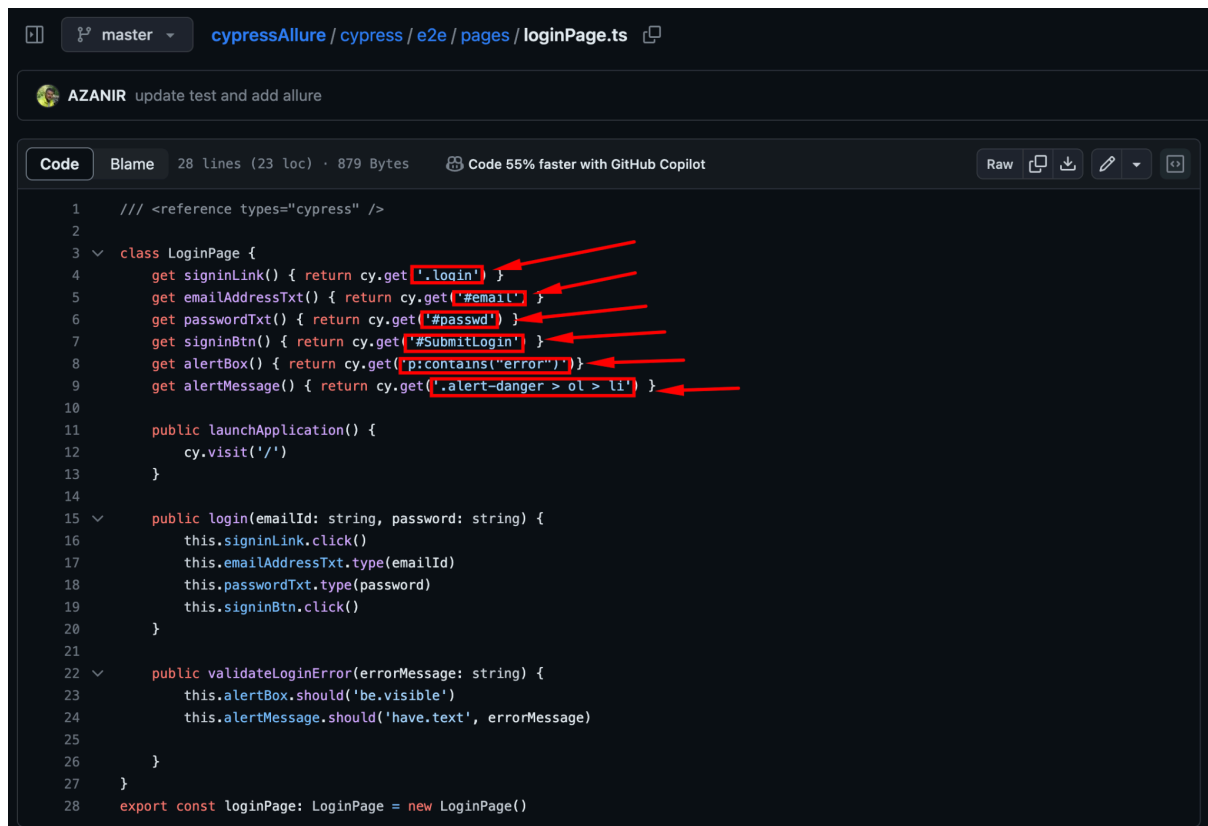


# Code review for the “cypressAllure” project

1. Selectors should be declared as constants before the class declaration, rather than being placed within the getters.



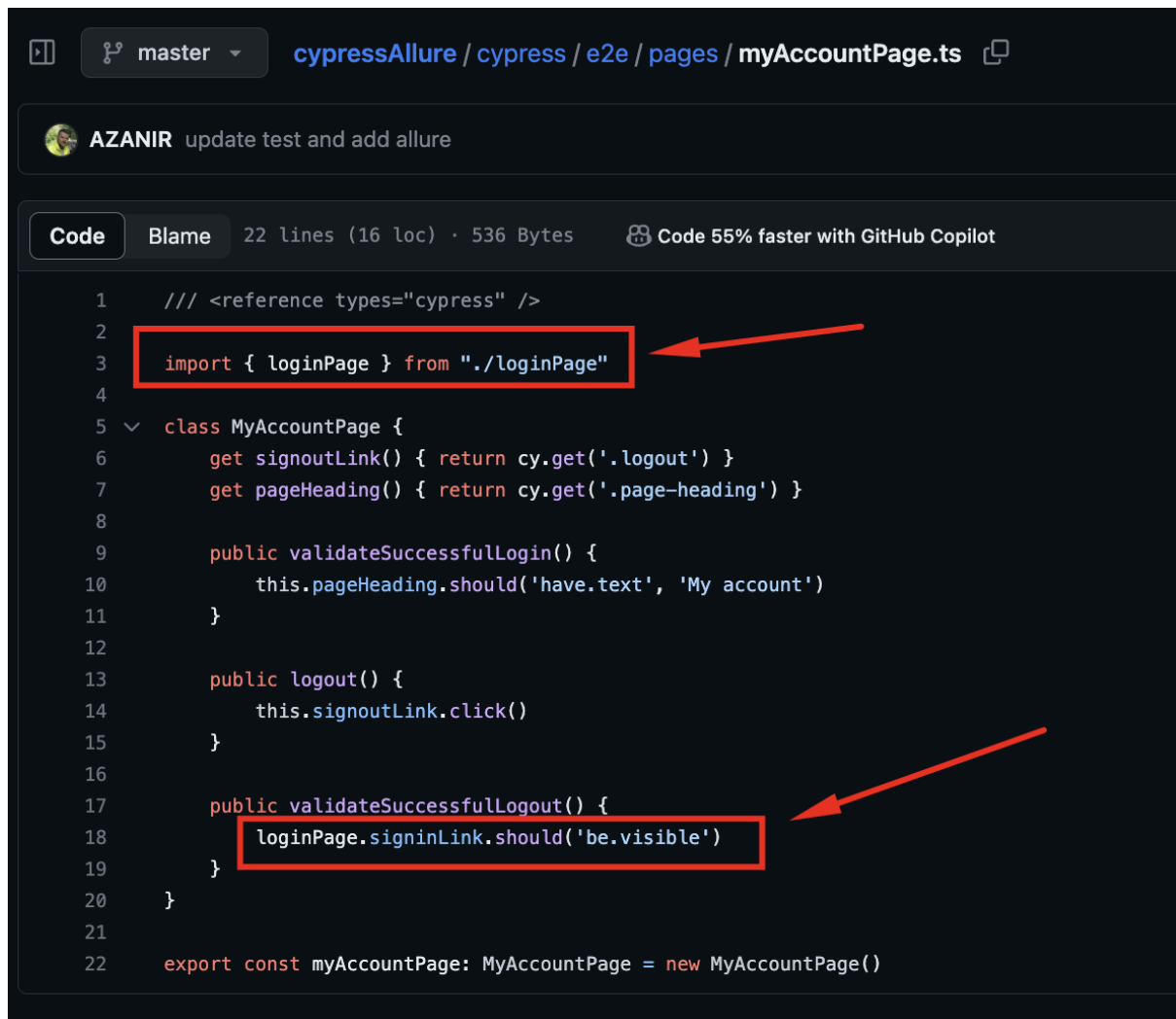
```
1  /// <reference types="cypress" />
2
3  import { loginPage } from "../loginPage"
4
5  class MyAccountPage {
6    get signoutLink() { return cy.get('logout') }
7    get pageHeading() { return cy.get('page-heading') }
8
9    public validateSuccessfulLogin() {
10      this.pageHeading.should('have.text',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



```
1  /// <reference types="cypress" />
2
3  class LoginPage {
4    get signinLink() { return cy.get('login') }
5    get emailAddressTxt() { return cy.get('#email') }
6    get passwordTxt() { return cy.get('#passwd') }
7    get signinBtn() { return cy.get('#SubmitLogin') }
8    get alertBox() { return cy.get('p:contains("error")') }
9    get alertMessage() { return cy.get('alert-danger > ol > li') }
10
11    public launchApplication() {
12      cy.visit('/')
13    }
14
15    public login(emailId: string, password: string) {
16      this.signinLink.click()
17      this.emailAddressTxt.type(emailId)
18      this.passwordTxt.type(password)
19      this.signinBtn.click()
20    }
21
22    public validateLoginError(errorMessage: string) {
23      this.alertBox.should('be.visible')
24      this.alertMessage.should('have.text', errorMessage)
25    }
26  }
27
28  export const loginPage: LoginPage = new LoginPage()
```

2. It is recommended to use a two-level page object pattern.

3. Page object models should contain getters and methods that correspond only to their respective pages.



```
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
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()
```

4. Verification steps should be placed in the spec files rather than defining verification methods within page object models.

```
master cyressAllure / cypress / e2e / tests / login.test.ts
AZANIR final update yml
Code Blame 37 lines (36 loc) · 1.75 KB Code 55% faster with GitHub Copilot

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    loginPage.login("testautomation@cypresstest.com", "Test@1234")
13    myAccountPage.validateSuccessfulLogin()
14    myAccountPage.logout()
15    myAccountPage.validateSuccessfulLogout()
16  })
17  it('login with valid credentials read data from fixture', function () {
18    loginPage.login(this.data.valid_credentials.emailId, this.data.valid_credentials.password)
19    myAccountPage.validateSuccessfulLogin()
20    myAccountPage.logout()
21    myAccountPage.validateSuccessfulLogout()
22  })
23  it('login with invalid email credentials read data from fixture', function () {
24    loginPage.login(this.data.invalid_credentials.invalid_email.emailId,
25      this.data.invalid_credentials.invalid_email.password)
26    loginPage.validateLoginError('Authentication failed.')
27  })
28  it('login with invalid password credentials read data from fixture', function () {
29    loginPage.login(this.data.invalid_credentials.invalid_password.emailId,
30      this.data.invalid_credentials.invalid_password.password)
31    loginPage.validateLoginError('Authentication failed.')
32  })
33  it('login with wrong email format credentials read data from fixture', function () {
34    loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email_format.password)
35    loginPage.validateLoginError('Invalid email addressssss.')
36  })
37 })
```

5. The test cases “login with valid credentials” and “login with valid credentials read data from fixture” are identical, except for reading the data. Therefore, the first one should be removed.

```
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    loginPage.login("testautomation@cypress.test.com", "Test@1234")
13    myAccountPage.validateSuccessfulLogin()
14    myAccountPage.logout()
15    myAccountPage.validateSuccessfulLogout()
16  })
17  it('login with valid credentials read data from fixture', function () {
18    loginPage.login(this.data.valid_credentials.emailId, this.data.valid_credentials.password)
19    myAccountPage.validateSuccessfulLogin()
20    myAccountPage.logout()
21    myAccountPage.validateSuccessfulLogout()
22  })
23  it('login with invalid email credentials read data from fixture', function () {
24    loginPage.login(this.data.invalid_credentials.invalid_email.emailId,
25      this.data.invalid_credentials.invalid_email.password)
26    loginPage.validateLoginError('Authentication failed.')
27  })
28  it('login with invalid password credentials read data from fixture', function () {
29    loginPage.login(this.data.invalid_credentials.invalid_password.emailId,
30      this.data.invalid_credentials.invalid_password.password)
31    loginPage.validateLoginError('Authentication failed.')
32  })
33  it('login with wrong email format credentials read data from fixture', function () {
34    loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email_format.password)
35    loginPage.validateLoginError('Invalid email addresssssss.')
36  })
37 })
```

6. The test “login with wrong email format credentials read data from fixture” is failed due to the typo in the error message argument provided to the validateLoginError method.

```
master cypressAllure / cypress / e2e / tests / login.test.ts
AZANIR final update.yml

Code Blame 37 lines (36 loc) · 1.75 KB Code 55% faster with GitHub Copilot

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    loginPage.login("testautomation@cyresstest.com", "Test@1234")
13    myAccountPage.validateSuccessfulLogin()
14    myAccountPage.logout()
15    myAccountPage.validateSuccessfulLogout()
16  })
17  it('login with valid credentials read data from fixture', function () {
18    loginPage.login(this.data.valid_credentials.emailId, this.data.valid_credentials.password)
19    myAccountPage.validateSuccessfulLogin()
20    myAccountPage.logout()
21    myAccountPage.validateSuccessfulLogout()
22  })
23  it('login with invalid email credentials read data from fixture', function () {
24    loginPage.login(this.data.invalid_credentials.invalid_email.emailId,
25      this.data.invalid_credentials.invalid_email.password)
26    loginPage.validateLoginError('Authentication failed.')
27  })
28  it('login with invalid password credentials read data from fixture', function () {
29    loginPage.login(this.data.invalid_credentials.invalid_password.emailId,
30      this.data.invalid_credentials.invalid_password.password)
31    loginPage.validateLoginError('Authentication failed.')
32  })
33  it('login with wrong email format credentials read data from fixture', function () {
34    loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email_format.password)
35    loginPage.validateLoginError('Invalid email addresssssss.')
36  })
37 })
```

azanir.github.io/cypressAllure/#suites/a9c8f11b4c0d702164a4731c0b55ffa0/57e1c9354671ca47/

Allure

Suites

order	name	duration	status
1	login with valid credentials	14s 300ms	Pass
2	login with valid credentials read data from fixture	10s 262ms	Pass
3	login with invalid email credentials read data from fixture	10s 064ms	Pass
4	login with invalid password credentials read data from fixture	6s 283ms	Pass
5	login with wrong email format credentials read data from fixture	12s 927ms	Failed

Failed login with wrong email format credentials read data from fixture

Overview History Retries

Timed out retrying after 5000ms: expected '<li>' to have text 'Invalid email addresssssss.', but the text was 'Invalid email address.'

Execution

Test body

- before each hook
- get login
- click
- page load --waiting for new page to load--
- GET https://www.facebook.com/x/oauth/status?ancestor\_origins=http%3A%2F%2F2Fautomationpractice.com&client\_id=334341610034299&input\_token&origin=1&redirect\_uri=http%3A%2F%2F2Fautomationpractice.com%2Findex.php&dk=joey&wants\_cookie\_data=false
- new url http://automationpractice.com/index.php?controller=authentication&back=my-account
- get #email
- type testautomationresstest.com
- get #passwd
- type test@12345

7. In certain test cases, such as “login with invalid email credentials” and “login with invalid password credentials”, the same testing data from a JSON file is being utilized, rather than generating realistic synthetic data with faker-js.

```
master cypressAllure / cypress / e2e / tests / login.test.ts
AZANIR final update.yml
Code Blame 37 lines (36 loc) · 1.75 KB Code 55% faster with GitHub Copilot

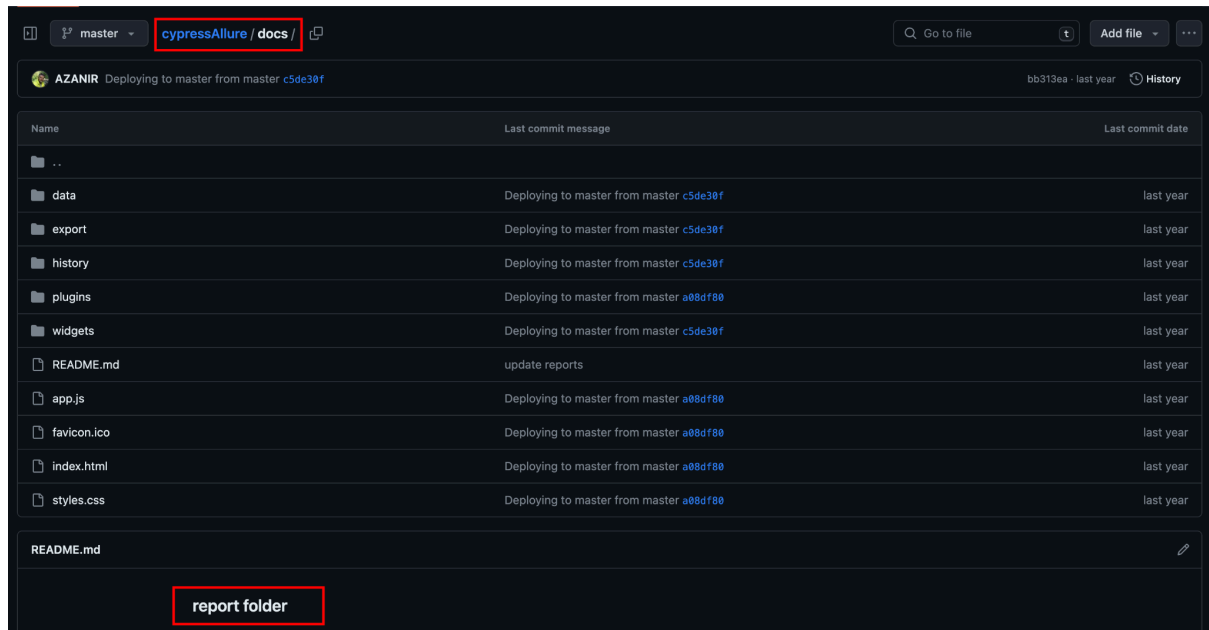
1 import { loginPage } from '../pages/loginPage'
2 import { myAccountPage } from '../pages/myAccountPage'
3
4 describe('Login Functionality', () => {
5   beforeEach(() => {
6     loginPage.launchApplication()
7     cy.fixture('users.json').then(function (data) {
8       this.data = data;
9     })
10  })
11  it('login with valid credentials', function () {
12    loginPage.login("testautomation@cypresstest.com", "Test@1234")
13    myAccountPage.validateSuccessfulLogin()
14    myAccountPage.logout()
15    myAccountPage.validateSuccessfulLogout()
16  })
17  it('login with valid credentials read data from fixture', function () {
18    loginPage.login(this.data.valid_credentials.emailId, this.data.valid_credentials.password)
19    myAccountPage.validateSuccessfulLogin()
20    myAccountPage.logout()
21    myAccountPage.validateSuccessfulLogout()
22  })
23  it('login with invalid email credentials read data from fixture', function () {
24    loginPage.login(this.data.invalid_credentials.invalid_email.emailId,
25      this.data.invalid_credentials.invalid_email.password)
26    loginPage.validateLoginError('Authentication failed.')
27  })
28  it('login with invalid password credentials read data from fixture', function () {
29    loginPage.login(this.data.invalid_credentials.invalid_password.emailId,
30      this.data.invalid_credentials.invalid_password.password)
31    loginPage.validateLoginError('Authentication failed.')
32  })
33  it('login with wrong email format credentials read data from fixture', function () {
34    loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email_format.password)
35    loginPage.validateLoginError('Invalid email addresssssss.')
36  })
37 })
```

8. Since there is only one test case in the “myAccount.test.ts”, using beforeEach is unnecessary. Additionally, the test case in this file does not correspond to the checking of the my account functionality.

```
master cypressAllure / cypress / e2e / tests / myAccount.test.ts
AZANIR update test
Code Blame 11 lines (10 loc) · 288 Bytes Code 55% faster with GitHub Copilot

1 import { loginPage } from "../pages/loginPage";
2
3 describe('My Account Functionality', () => {
4   beforeEach(() => {
5     cy.visit('https://google.com');
6     //loginPage.launchApplication()
7   })
8   it('Sample Test', () => {
9     console.log("This is a sample test")
10  })
11 })
```

9. The repository contains a folder with redundant files. The folder containing reports should either be placed in a separate branch within the same repository or in a different repository or added to the .gitignore file.



10. The repository has files with unnecessary comments.





master ▾

cypressAllure / cypress / support / e2e.js



AZANIR update test and add allure

Code

Blame

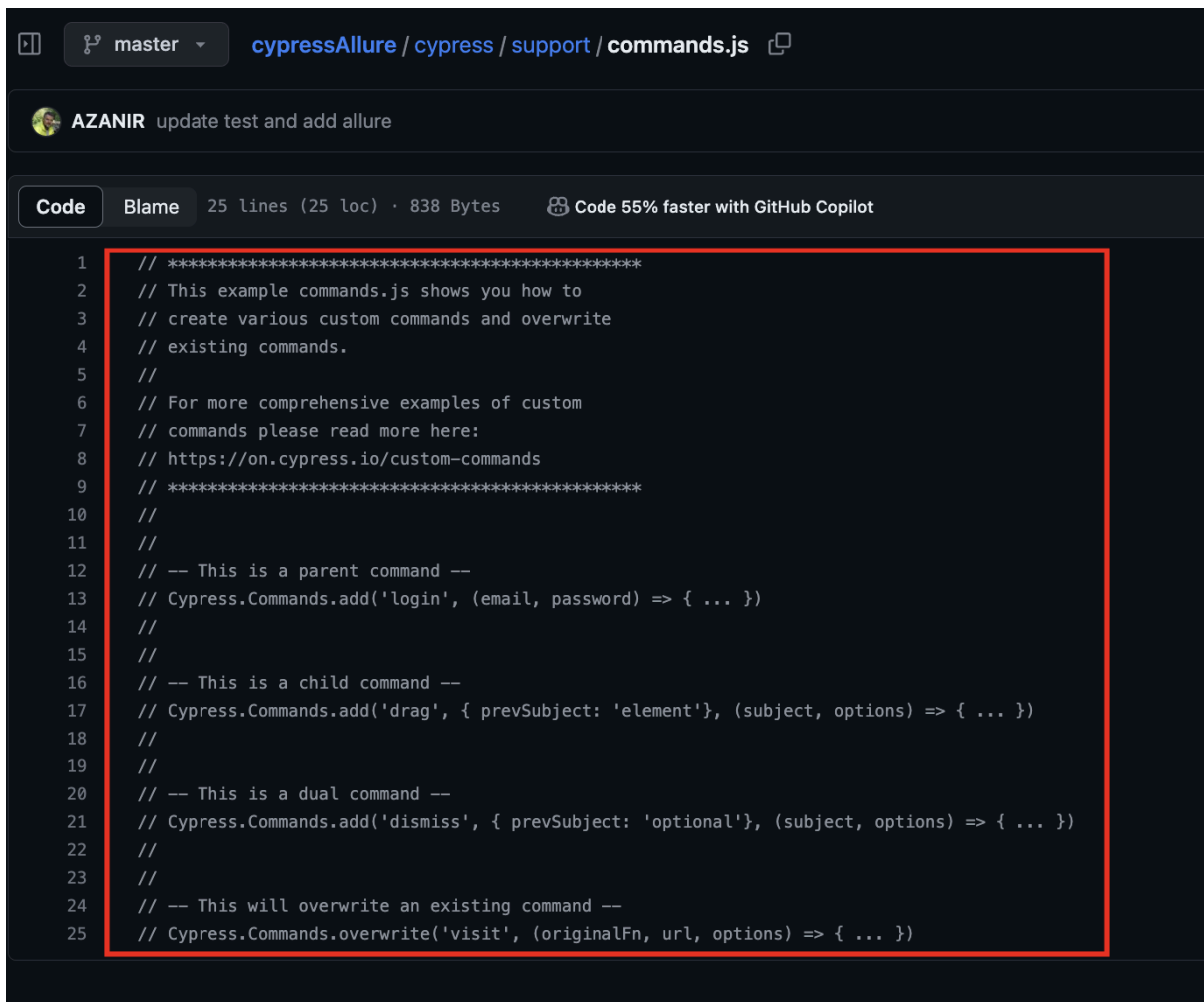
21 lines (19 loc) · 709 Bytes



Code 55% faster with GitHub Copilot


```
1 // *****
2 // This example support/index.js is processed and
3 // loaded automatically before your test files.
4 //
5 // This is a great place to put global configuration and
6 // behavior that modifies Cypress.
7 //
8 // You can change the location of this file or turn off
9 // automatically serving support files with the
10 // 'supportFile' configuration option.
11 //
12 // You can read more here:
13 // https://on.cypress.io/configuration
14 // *****
15
16 // Import commands.js using ES2015 syntax:
17 import './commands'
18 import '@shelex/cypress-allure-plugin'
19
20 // Alternatively you can use CommonJS syntax:
21 // require('./commands')
```





```
1 // *****
2 // This example commands.js shows you how to
3 // create various custom commands and overwrite
4 // existing commands.
5 //
6 // For more comprehensive examples of custom
7 // commands please read more here:
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) => { ... })
```

11. Used “console.log” instead of verification rather than using verification with “expect” or “assert”.



```
1 import { loginPage } from "../pages/loginPage";
2
3 describe('My Account Functionality', () => {
4   beforeEach(() => {
5     cy.visit('https://google.com');
6     //loginPage.launchApplication()
7   })
8   it('Sample Test', () => {
9     console.log("This is a sample test")
10  })
11 })
```

12. In the include paths, incorrect entries are used. The paths should be corrected from “../node\_modules/...” to “./node\_modules/...” since the include and exclude paths are relative to the location of the tsconfig.json file.

As the “cypress” package is already included in the “types” field, it is unnecessary to include it again in the “include” field.

```
1 {
2   "compilerOptions": {
3     "target": "es2016",
4     "module": "commonjs",
5     "lib": ["dom", "es6", "es7"],
6     "types": ["node", "cypress"],
7     "strict": true,
8     "skipLibCheck": true,
9     "baseUrl": "./",
10  },
11  "exclude": ["node_modules"],
12  "include": ["**/*.ts",
13    "../node_modules/@shelex/cypress-allure-plugin/reporter",
14    "../node_modules/cypress"]
15 }
```

13. This repository is incorrectly labeled as a framework in the “README.md” file.

# cypress-e2e-allure# Cypress-e2e-Allure Automation-Framework with allure reports

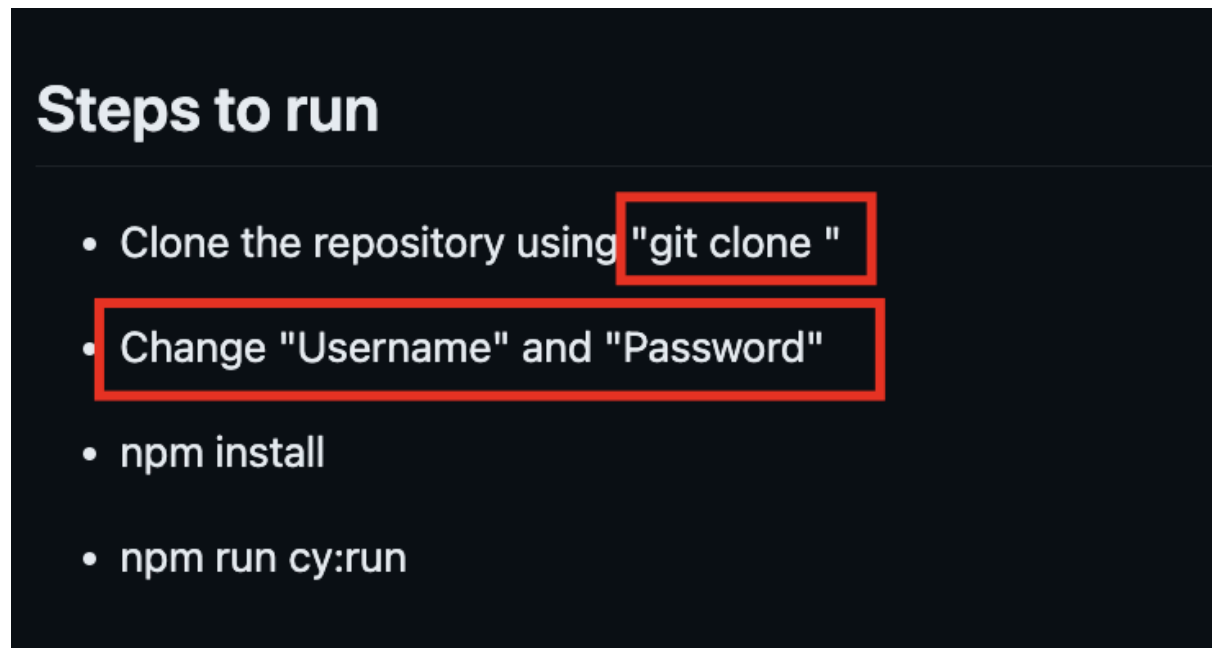
This **Framework** contains sample code for # Cypress-e2e-Allure Automation-Framework

- Page Object Model
- Reading test data from json file
- Simple login scenario with valid credentials and with invalid credentials
- Allure report integration

14. The provided steps to run are incomplete, to the “git clone” it is preferable to add the url to the repository.

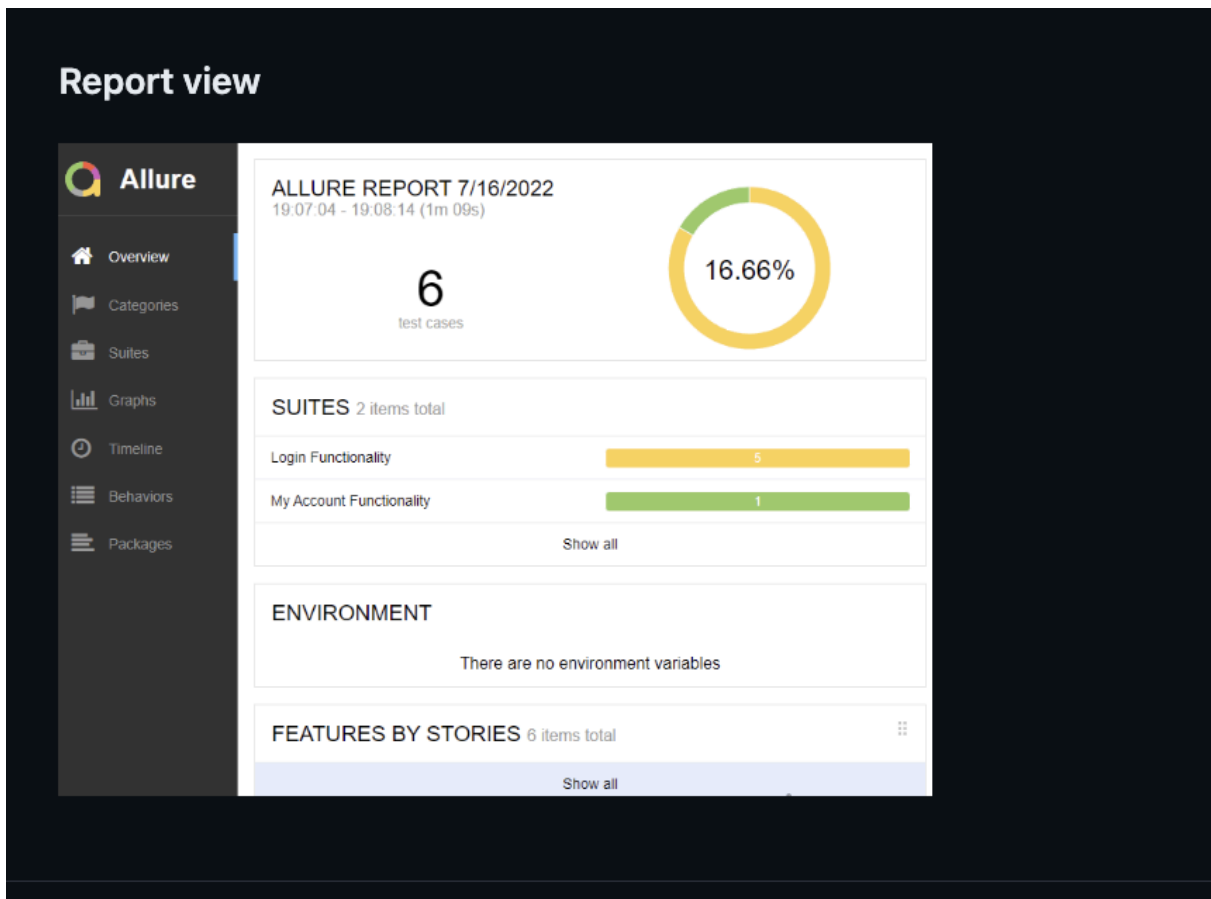
The provided details in the step about changing “Username” and “Password” are insufficient, more specific information is needed.

It would be beneficial to expand the steps to run to include details about other scripts defined in the “package.json” file, not just limited to “npm run cy:run”.

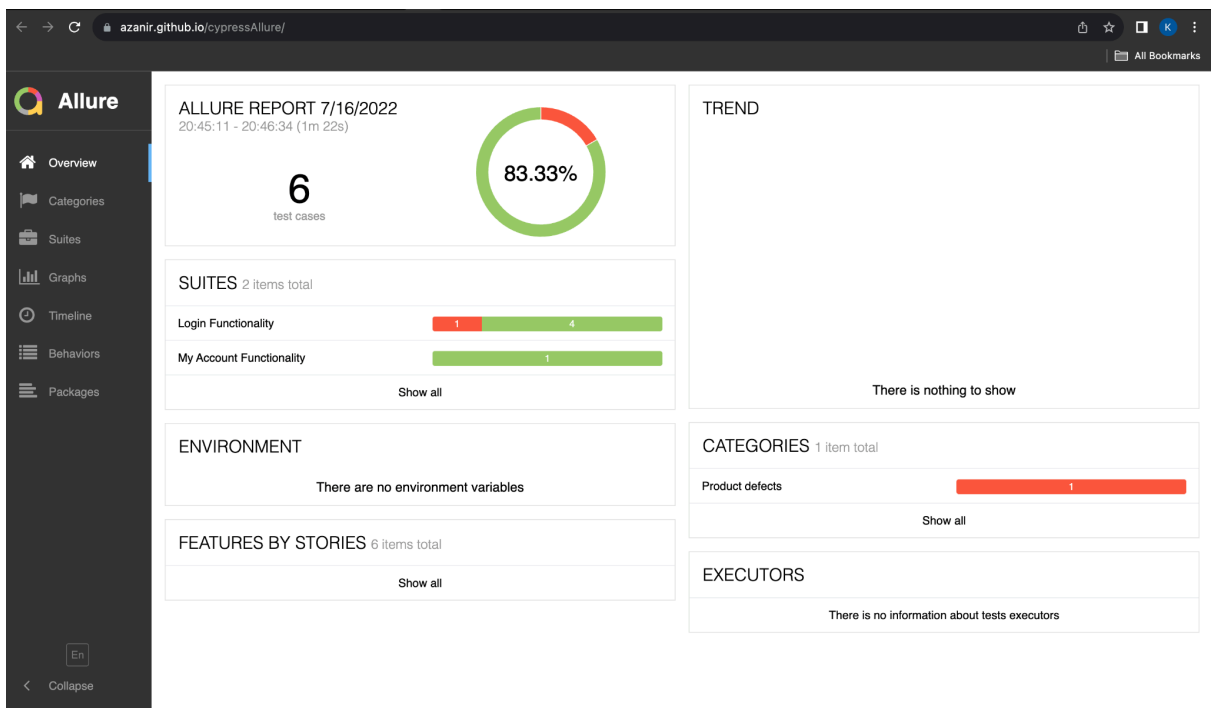


15. Instead of including a screenshot of the report in the “Report view” section of the “README.md” file, it is preferable to provide a link to the deployed report on GitHub Pages. This ensures that users can access the most up-to-date report, considering potential changes in the repository. Also, the deployed report and the screenshot of the report are already different.

// screenshot of the report in the “README.md”



// the deployed report to the GitHub Pages

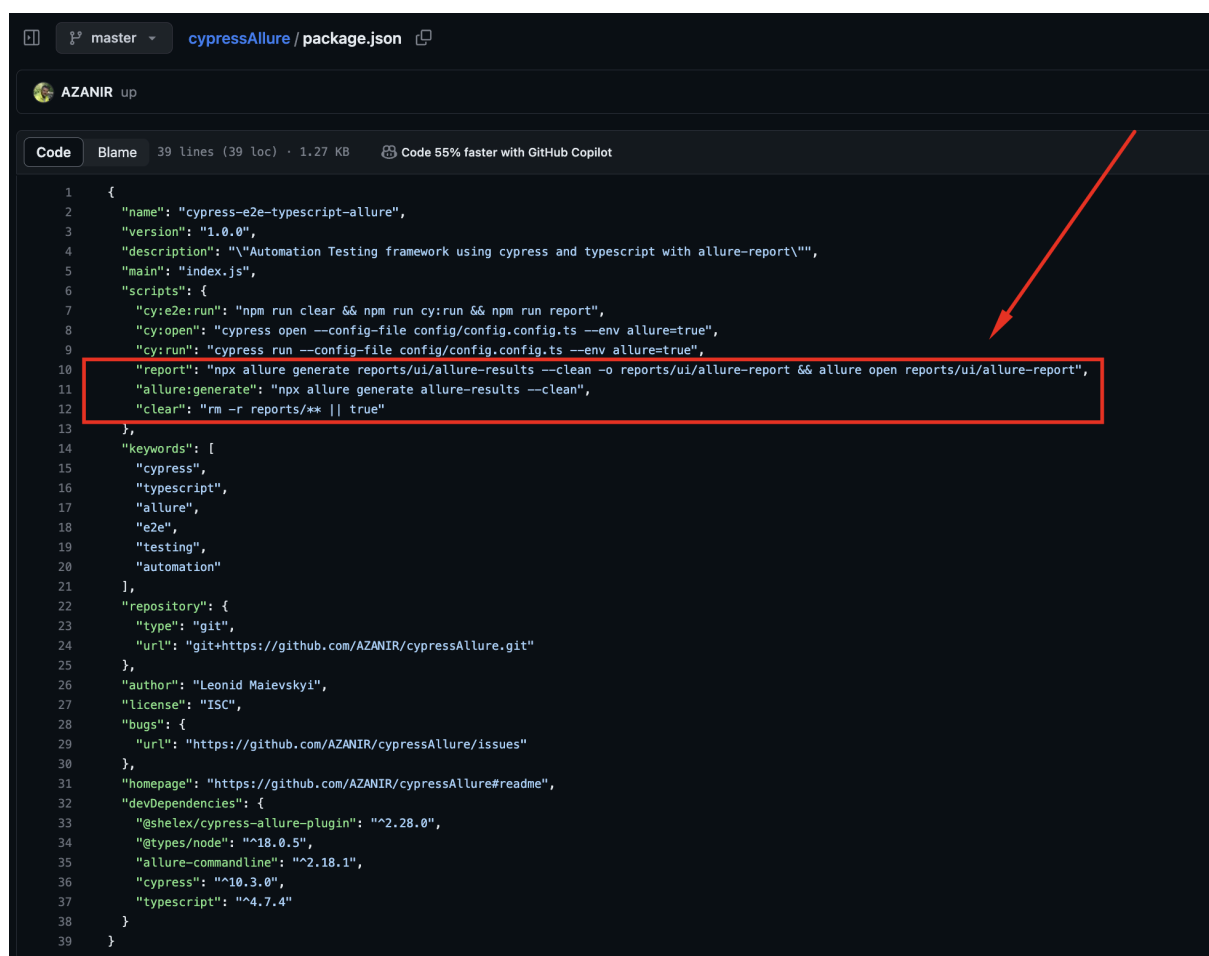


16. The “README.md” file lacks information about requirements, specifically, there is no mention of the need to have Node.js and Java (8 or higher) installed.

17. The “report” and “allure:generate” scripts in the “package.json” file are facing issues due to misaligned path, and the “clear” script failed to remove all content from directories used for allure report. According to the “config.config.ts” file, these scripts can be updated to the following:

```
"report": "npx allure generate allure-results --clean -o reports/allure-report  
&& allure open reports/allure-report",  
  "allure:generate": "npx allure generate allure-results --clean -o  
reports/allure-report",  
  "clear": "rm -r reports/** allure-results/* || true"
```

But it’s advisable to maintain all the data related to allure reports in one folder.



```
1 {  
2   "name": "cypress-e2e-typescript-allure",  
3   "version": "1.0.0",  
4   "description": "\"Automation Testing framework using cypress and typescript with allure-report\"",  
5   "main": "index.js",  
6   "scripts": {  
7     "cy:e2e:run": "npm run clear && npm run cy:run && npm run report",  
8     "cy:open": "cypress open --config-file config/config.config.ts --env allure=true",  
9     "cy:run": "cypress run --config-file config/config.config.ts --env allure=true",  
10    "report": "npx allure generate reports/ui/allure-results --clean -o reports/ui/allure-report && allure open reports/ui/allure-report",  
11    "allure:generate": "npx allure generate allure-results --clean",  
12    "clear": "rm -r reports/** || true"  
13  },  
14  "keywords": [  
15    "cypress",  
16    "typescript",  
17    "allure",  
18    "e2e",  
19    "testing",  
20    "automation"  
21  ],  
22  "repository": {  
23    "type": "git",  
24    "url": "git+https://github.com/AZANIR/cypressAllure.git"  
25  },  
26  "author": "Leonid Maievskiy",  
27  "license": "ISC",  
28  "bugs": {  
29    "url": "https://github.com/AZANIR/cypressAllure/issues"  
30  },  
31  "homepage": "https://github.com/AZANIR/cypressAllure#readme",  
32  "devDependencies": {  
33    "@shelex/cypress-allure-plugin": "^2.28.0",  
34    "@types/node": "^18.0.5",  
35    "allure-commandline": "^2.18.1",  
36    "cypress": "^10.3.0",  
37    "typescript": "^4.7.4"  
38  }  
39 }
```

18. The “.gitignore” file contains directory names that are already part of another directory, which is already included in the “.gitignore” file.

If the scripts in the “package.json” are corrected as provided earlier, the allure-report directory can also be removed from the “.gitignore” file.



master ▾

cypressAllure / .gitignore



AZANIR up

Code

Blame

8 lines (8 loc) · 103 Bytes



Code 55% faster with GitHub Copilot

```
1  node_modules
2  videos
3  reports
4  screenshots
5  cypress/downloads
6  reports/**
7  allure-report/**
8  allure-results/**
```

