

1. README.md file

1 - Incorrect line in the readme file

There is no explanation of where to change the «Username» and «Password».

```
## **Steps to run**  
  
- Clone the repository using "git clone "  
  
- Change "Username" and "Password"
```

2 – There is no link after the command “clone”

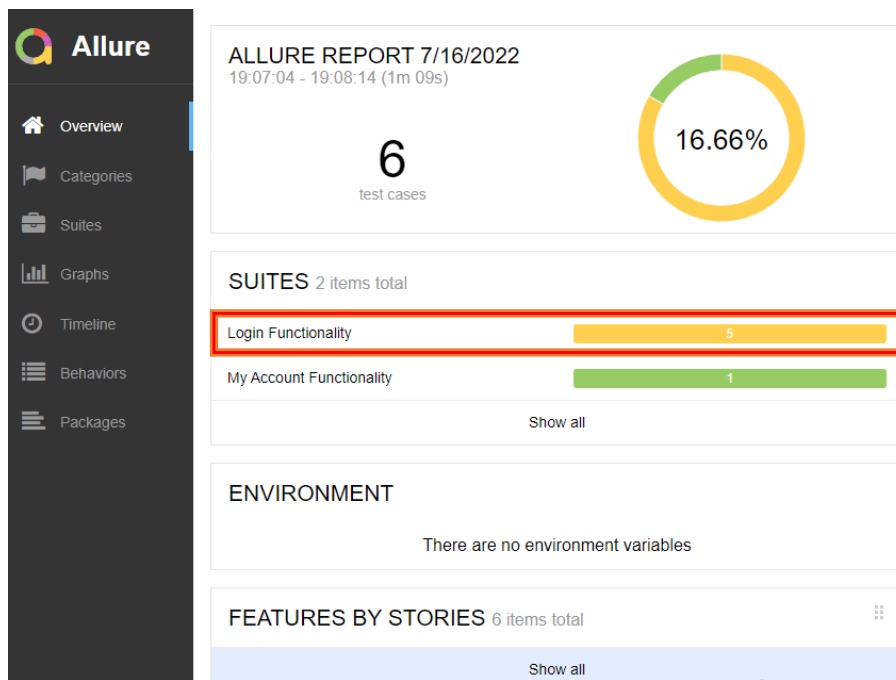
```
## **Steps to run**  
  
- Clone the repository using "git clone "  
  
- Change "Username" and "Password"
```

3 - Any information about GitHub Page in the readme file

GitHub Page with the allure report is connected, but it is not indicated anywhere

4 - Bad screenshot in the readme file

On the screenshot in the Redmi file, there are 5 tests with the status «Broken»



5 - Lack of information on how to view the allure report

The README file shows how to run the test and provides a screenshot of the report, but no information on how to view the report after running the tests locally.

6 – No information about Technologies

It's important to know which version of Node.js and Cypress was used.

2. Test-cases

1 - Lack of test cases

It is difficult to understand what the author of the code wanted to check without test cases

3. Config file

1 - A lot of timeouts in the config file

If possible, it is better not to use time delays.

```
import { defineConfig } from 'cypress'

export default defineConfig({
  video: true,
  defaultCommandTimeout: 5000,
  execTimeout: 5000,
  taskTimeout: 5000,
  pageLoadTimeout: 30000,
  requestTimeout: 5000,
  responseTimeout: 30000,
})
```

4. Pages

1 - Needed on mainPage or homePage

One of the elements in the loginPage doesn't belong to the login page but belongs to the main page of the site

```
class LoginPage {
  get signinLink() { return cy.get('.login') }
}
```

2 - No verification

There is no verification of going to another page and filling in the fields with the correct data.

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

3 - Unrealized inheritance

The MyAccountPage uses the LoginPage import but doesn't use the class extension. The inheritance implementation allows you to import only one page in a test.

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

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

5. Tests

1 - String too long

Moving part of the line to make the code easier to read is better.

```
it('login with wrong email format credentials read data from fixture', function () {
  loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email_format.password)
  loginPage.validateLoginError('Invalid email addresssssss.')
})
```

2 - Mistake in line

An error in the string comparing the expected and actual result (there must be an address, not addresssssss)

```
it('login with wrong email format credentials read data from fixture', function () {
  loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email_format.password)
  loginPage.validateLoginError('Invalid email addresssssss.')
})
```

3 - The description of the test doesn't correspond to it

In the MyAccount.test, the description doesn't correspond to the test.

```
cypress > e2e > tests > TS myAccount.test.ts > ...
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
```

4 - Unnecessary import

The loginPage is imported into the MyAccount.test, but it is not used

```
cypress > e2e > tests > TS myAccount.test.ts > ...
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 |
```

5 - MyAccount.test

The file is only a sample test. Its existence in the project makes no sense.

6 - Redundant actions in the tests

The description of the tests takes into account only the verification of the ability to log in but doesn't include the verification of logging out of the account.

```
it('login with valid credentials', function () {
  loginPage.login("testautomation@cypress.test.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()
})
```

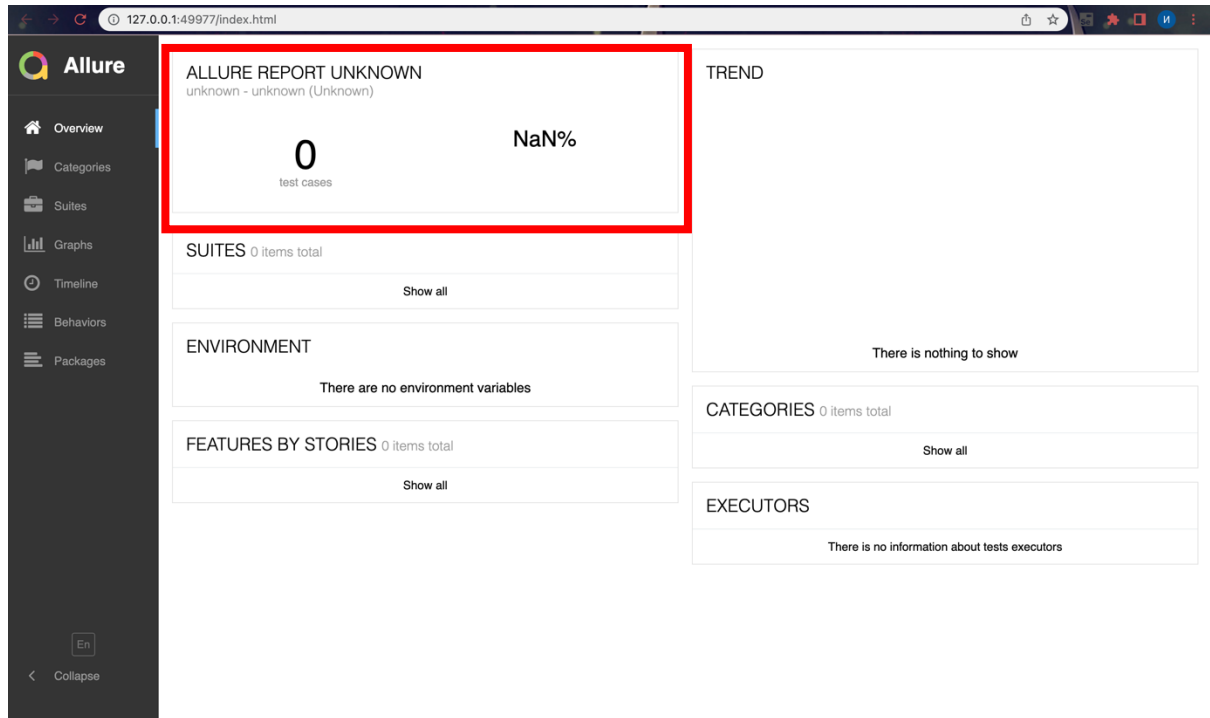
7 - There is no indexing of tests

It is difficult to find the required test without specifying the index if there are many tests in the project.

6. Reporter

1 - The allure report isn't generated

When using the command “npm run report”, the report returns an empty result.



7. Package.json file

1 - The command is incomplete

The command “cy:e2e:run” should generate and open the report, but nothing happens after running the tests.

```
"scripts": {  
  "cy:e2e:run": "npm run clear && npm run cy:run && npm run report",  
  "cy:open": "cypress open --config-file config/config.config.ts --env allure=true",  
  "cy:run": "cypress run --config-file config/config.config.ts --env allure=true",  
  "report": "npx allure generate reports/ui/allure-results --clean -o reports/ui/allure-report && allure open reports/ui/allure-report",  
  "allure:generate": "npx allure generate allure-results --clean",  
  "clear": "rm -r reports/** || true"  
},
```