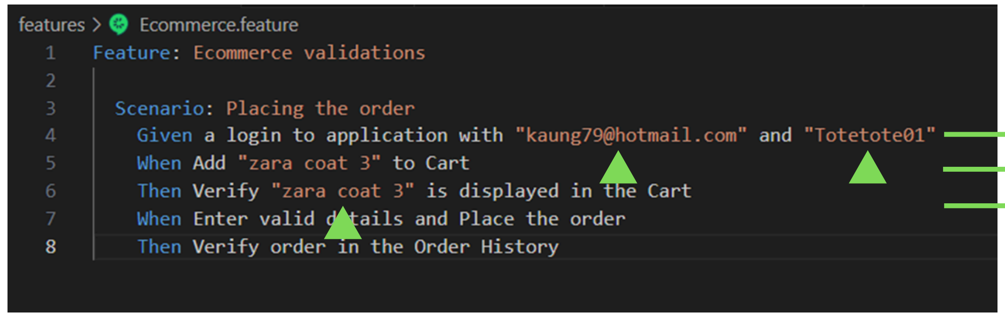
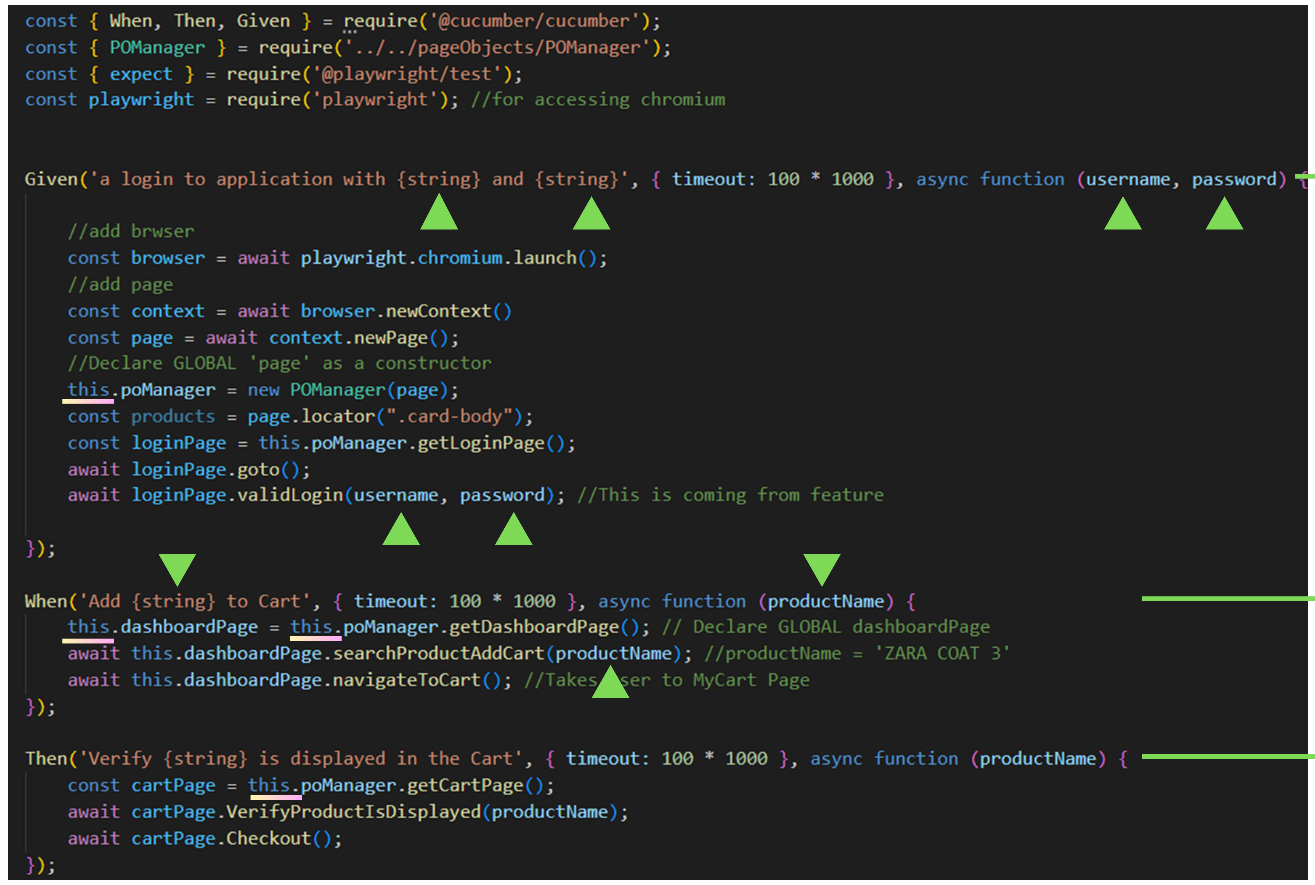
Cucumber features > Ecommerce.feature



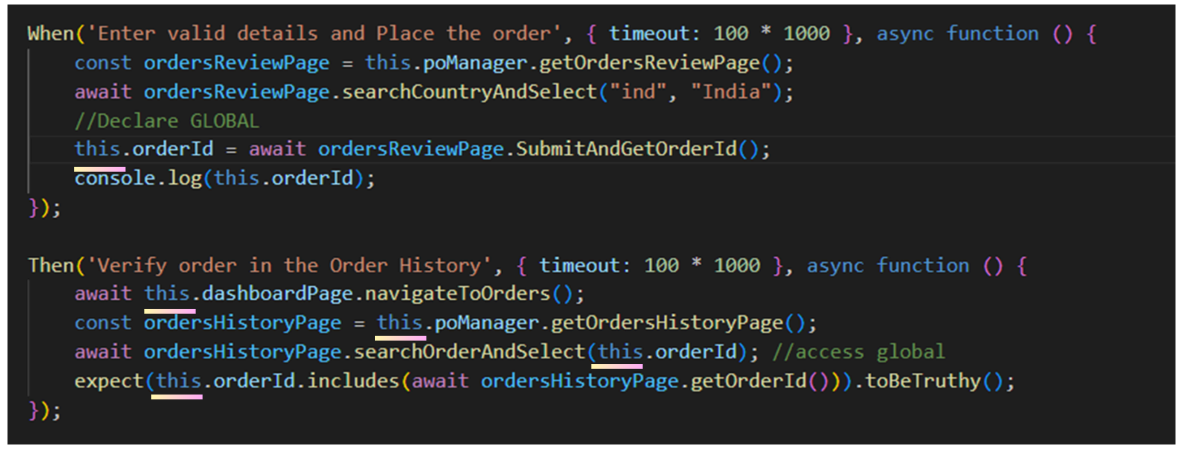
features > Ecommerce.feature

```
Feature: Ecommerce validations

  Scenario: Placing the order
    Given a login to application with "kaung79@hotmail.com" and "Totetote01"
    When Add "zara coat 3" to Cart
    Then Verify "zara coat 3" is displayed in the Cart
    When Enter valid details and Place the order
    Then Verify order in the Order History
```

features > step_definitions > steps.js



```
const { When, Then, Given } = require('@cucumber/cucumber');
const { POManager } = require('../../pageObjects/POManager');
const { expect } = require('@playwright/test');
const playwright = require('playwright'); //for accessing chromium


Given('a login to application with {string} and {string}', { timeout: 100 * 1000 }, async function (username, password) {

    //add brwser
    const browser = await playwright.chromium.launch();
    //add page
    const context = await browser.newContext()
    const page = await context.newPage();
    //Declare GLOBAL 'page' as a constructor
    this.poManager = new POManager(page);
    const products = page.locator(".card-body");
    const loginPage = this.poManager.getLoginPage();
    await loginPage.goto();
    await loginPage.validLogin(username, password); //This is coming from feature

});

When('Add {string} to Cart', { timeout: 100 * 1000 }, async function (productName) {
    this.dashboardPage = this.poManager.getDashboardPage(); // Declare GLOBAL dashboardPage
    await this.dashboardPage.searchProductAddCart(productName); //productName = 'ZARA COAT 3'
    await this.dashboardPage.navigateToCart(); //Takes user to MyCart Page
});

Then('Verify {string} is displayed in the Cart', { timeout: 100 * 1000 }, async function (productName) {
    const cartPage = this.poManager.getCartPage();
    await cartPage.VerifyProductIsDisplayed(productName);
    await cartPage.Checkout();
});
```



```
When('Enter valid details and Place the order', { timeout: 100 * 1000 }, async function () {
    const ordersReviewPage = this.poManager.getOrdersReviewPage();
    await ordersReviewPage.searchCountryAndSelect("ind", "India");
    //Declare GLOBAL
    this.orderId = await ordersReviewPage.SubmitAndGetOrderId();
    console.log(this.orderId);
});

Then('Verify order in the Order History', { timeout: 100 * 1000 }, async function () {
    await this.dashboardPage.navigateToOrders();
    const ordersHistoryPage = this.poManager.getOrdersHistoryPage();
    await ordersHistoryPage.searchOrderAndSelect(this.orderId); //access global
    expect(this.orderId.includes(await ordersHistoryPage.getOrderId())).toBeTruthy();
});
```