Registration Feature

one scenario is required

Prepare your files:

your feature file will be:

src >> main >> resources >> features >> F01_Register.feature (Remember to add @smoke)

your step definition class will be:

src >> test >> java >> org.example >> stepDefs >> D01_registerStepDef

your class to apply pom design pattern in your project will be:

src >> test >> java >> org.example >> pages >> P01_register

Description:

you need to create only one scenario in this feature file to verify that user could signup with valid account

Feature File:

src >> main >> resources >> features >> F01_Register.feature

@smoke

Feature: F01_Register | users could register with new accounts

Scenario: guest user could register with valid data successfully

Step 1 user go to register page

Step 2 user select gender type

Step 3 user enter first name "automation" and last name "tester"

Step 4 user enter date of birth

Step 5 user enter email "test@example.com" field

Step 6 user fills Password fields "P@ssw0rd" "P@ssw0rd"

Step 7 user clicks on register button

Step 8 success message is displayed

Replace Steps with the appropriate Keywords (Given, When, Then, And)

for example, you will replace Step 1 with Given and Step 2 with When and so one

Example on Step 1

* open src >> test >> java >> org.example >> pages >> P01_register

create WebElement method to register button

```
public WebElement registerlink()
{
return  Hooks.driver.findElement(By.cssSelector("a[class=\"ico-signUp\"]"));
}
```

* open src >> test >> java >> org.example >> stepDefs >> D01_registerStepDef

```
//Create new object from P01_register
P01_register register = new P01_register();
   @Given("user go to register page")
   public void go_to_register()
   {
          register.registerlink().click();
   }
```

* Follow same thing with all the remaining steps and don't forget to apply POM Design pattern on each step

* In step 4

you should check if the birthdate lists are static or dynamic then decide if you will use Select class in Selenium to handle the dropdown list or not

* In Step 8 (Assertions):

Use soft assertion to verify the following

- Success message is displayed "Your registration completed"

- the color of this message is green rgba(76, 177, 124, 1)

Note: we need to get the "color" not "background-color" using getCssValue command which is already explained in Selenium Commands Chapter

Reviewers will check the following

1- Student selects the correct Keywords in feature file

2- Student applied Select class to handle static dropdown lists

3- POM Design pattern is applied in step definition class correctly

4- Student applied Soft assertion as required for all scenarios

Login Feature

2 scenarios are required

Prepare your files:

your feature file will be:

src >> main >> resources >> features >> F02_Login.feature (Remember to add @smoke)

your step definition class will be:

src >> test >> java >> org.example >> stepDefs >> D02_loginStepDef

your class to apply pom design pattern in your project will be:

src >> test >> java >> org.example >> pages >> P02_login

in your feature file write the following

@smoke

Feature: F02_Login | users could use login functionality to use their accounts

Scenario: user could login with valid email and password

Step 1 user go to login page

Step 2 user login with "valid" "test@example.com" and "P@ssw0rd"

Step 3 user press on login button

Step 4 user login to the system successfully

Scenario: user could login with invalid email and password

Step 1 user go to login page

Step 2 user login with "invalid" "wrong@example.com" and "P@ssw0rd"

Step 3 user press on login button

Step 4 user could not login to the system

* Replace Steps with the appropriate Keywords (Given, When, Then, And)

* You should pass type, username, password values from feature file

What to assert exactly?

For Scenario 1

Use soft assertion to verify the following

1- getCurrentUrl and verify it equals https://demo.nopcommerce.com/

2- "My account" tab isDisplayed

don't forget assertAll()

For Scenario 2

Use soft assertion to verify the following

1- error message contains "Login was unsuccessful."

2- the color of this message is red "#e4434b"

don't forget assertAll()

Notes:

- the output of any color attribute is RGBA and could be converted to Hex value using this code

```
import org.openqa.selenium.support.Color;
Color.fromString("rgba(228, 67, 75, 1)").asHex();                    //output will be #e4434b
```

Reviewers will check the following

1- Student selects the correct Keywords in feature file

2- parameters are passed correctly from feature file

3- POM Design pattern is applied in step definition class correctly

4- Student applied Soft assertion as required for all scenarios

Currencies Feature

one scenario is required

Prepare your files:

your feature file will be:

src >> main >> resources >> features >> F03_currencies.feature (Remember to add @smoke)

your step definition class will be:

src >> test >> java >> org.example >> stepDefs >> D03_currenciesStepDef

your class to apply pom design pattern in your project will be:

src >> test >> java >> org.example >> pages >> P03_homePage

What you will learn?

The purpose of this scenario is to learn how to use findElements

Steps:

1- Select Euro currency from the dropdown list on the top left of home page

2- Use hard assertion to verify Euro Symbol (€) is shown on the 4 products displayed in Home page

in this step you should create for loop

2.1- use findElements with get(i) method then use getText()

2.2- save the output inside variable

2.3- do hard assert to verify that the variable contains "€"

Note when applying POM Design with findElements >> you should make your method List<WebElement> not WebElement

Remember:

The difference between findElement & findElements commands

findElement >> click(), sendKeys(""), isDisplayed()

findElements >> size(), get(), isEmpty()

Reviewers will check the following

1- Student selects the correct Keywords in feature file

2- POM Design pattern is applied in step definition class correctly

2.1- List<WebElement> method is used in POM Design Pattern

3- Student applies for loop correctly

4- Student applies Soft assertion as required for all scenarios

Search Feature

2 scenarios are required

Prepare your files:

your feature file will be:

src >> main >> resources >> features >> F04_Search.feature (Remember to add @smoke)

your step definition class will be:

src >> test >> java >> org.example >> stepDefs >> D04_searchStepDef

your class to apply pom design pattern in your project will be:

src >> test >> java >> org.example >> pages >> P03_homePage

What you should do?

You will learn how to apply Scenario Outline in cucumber

Explanation:

Scenario outline basically replaces variable/keywords with the value from the table.

Each row in the table is considered to be a scenario.

```
Scenario Outline: Login functionality for a social networking site.
Given user navigates to Facebook
When I enter Username as "<username>" and Password as "<password>"
Then login should be unsuccessful
Examples:
| username | password |
| Test1 | Pass1 |
| Test2 | Pass2 |
| Test3 | Pass3 |
```

In this example, the scenario will run 3 times.

First Time will use Test1, Pass1 as inputs and Second time will be Test2, Pass2 and so on

Note: username & password are the variable name and not considered as one of the inputs

Scenarios:

Scenario Outline 1 user could search using product name

Examples: book, laptop, nike

Scenario Outline 2 user could search for product using sku

Examples: SCI_FAITH, APPLE_CAM, SF_PRO_11

Note: sku means serial number of the product

for example open this link https://demo.nopcommerce.com/apple-macbook-pro-13-inch you will find sku is "AP_MBP_13"

Scenario Outline 1

Use soft assertion to verify the following

1- url contains https://demo.nopcommerce.com/search?q=

2- search shows relevant results

2.1- Count number of search results using findElements & size()

2.2- Create for loop and verify each result contains the search word

You should use toLowerCase() method

Scenario Outline 2

1- search with sku (You should pass parameter from Feature File in this step)

2- After searching, you need to click on the product in search result

3- Use hard assertion to verify the following

3.1- get the sku shown in this page then make sure it contains the sku that you are using in search

Reviewers will check the following

1- Student selects the correct Keywords in feature file

2- Student uses scenario outline as explained

3- POM Design pattern is applied in step definition class correctly

    3.1- List<WebElement> method is used in POM Design Pattern

4- Student applies for loop correctly

5- assertions are implemented

hoverCategories Feature

one scenario is required

Prepare your files:

your feature file will be:

src >> main >> resources >> features >> F05_hoverCategories.feature (Remember to add @smoke)

your step definition class will be:

src >> test >> java >> org.example >> stepDefs >> D05_hoverCategoriesStepDef

your class to apply pom design pattern in your project will be:

src >> test >> java >> org.example >> pages >> P03_homePage

What you need to do?

In this test scenario you will use Actions class in Selenium framework to hover on one of main categories then click on sub category

Scenario steps:

1- in Home page, there are 3 main categories contains sub-categories

[Computers, Electronics, Apparel]

2- you need to select random one of the three main categories

3- you should hover on the selected category

Note: there's a difference between click and hover command

How to do Hover action? you could get help from this project

https://github.com/KhaledAMRS/SeleniumCommands

4- after hovering, you need to select random one of the three sub cateogries like [Desktops, Notebooks, Software]

4.1- get the text of this sub-category

4.2- click on it

5- after opening sub-category page, you need to get text of the page title

div[class="page-title"] h1

6- Assert that the sub-category title is (equal or contains) the one you get it while selecting random sub-category

Note: you are supposed to use below functions

toLowerCase().trim(); // this will change "Desktops " to "desktops"

How to automate this test case in advanced method

Note: it's not mandatory to do the advanced method in the project, it's just to learn something new in Selenium course

1- Select any random category whether it contains sub cateogry or not

2- do if condition

if (selected category contains sub-category)

{

click on random sub-category then assert the opened page contains sub-category name

}

else if (selected category doesn't contains sub-category)

{

click on the main category itself then assert the opened page contains main category name

}

Note: before doing if condition you supposed to reduce implicitlyWait to 2 seconds

then after the if condition return implicitlyWait time again

Try to figure out why it's recommended to do something like this

Reviewers will check the following

1- Student selects the correct Keywords in feature file

2- Hover action is being used in the code

3- toLowerCase() & trim() are used in the code

4- POM Design pattern is applied in step definition class correctly

5- Student is doing assertion correctly