AIM: to use comments and tags in a Cucumber scenario.

CODE:

Code for features file

# this mean if this tag is called, all the scenario will run

# after the tag, we dont write anything

@all

Feature: Testing on NinaDemo application

Background:

Given I am on tutorialsNinja Homepage

@login @smoke

Scenario Outline: Test username and password for Login Page

Then I click on my account link

When I click on login link

Then I will be on the login page and I capture the title of the page

And I enter email id as '<EmailId>' on the page

And I enter password as '<password>' on the page

Then I click login button

And I close browser

Examples:

| EmailId | password |

| seleniumdemo@gmail.com | 123456 |

@search @regression

Scenario:

Then I enter Input for the search box

And I click on search button

And I close browser

@AddtoCart @regression

Scenario:

And I click on Add to Cart button

And I close browser

@checkout @regression

Scenario:

And I click on Checkout button

And I close browser

@logout @smoke

Scenario:

Then I click on my account link

And I click on logout link

And I close browser

Code for test cases

**package** steps;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** io.cucumber.java.en.Given;

**import** io.cucumber.java.en.Then;

**import** io.cucumber.java.en.When;

**public** **class** TestExamplesNinjaDemo {

WebDriver driver;

@Given("I am on tutorialsNinja Homepage")

**public** **void** i\_am\_on\_tutorials\_ninja\_homepage() {

driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("https://tutorialsninja.com/demo/index.php?route=common/home");

}

@Then("I click on my account link")

**public** **void** i\_click\_on\_my\_account\_link() {

driver.findElement(By.*xpath*("//span[text()='My Account']")).click();

}

@When("I click on login link")

**public** **void** i\_click\_on\_login\_link() {

driver.findElement(By.*xpath*("//a[text()='Login']")).click();

}

@Then("I will be on the login page and I capture the title of the page")

**public** **void** i\_will\_be\_on\_the\_login\_page\_and\_i\_capture\_the\_title\_of\_the\_page() {

System.***out***.println("title of the is page is : "+ driver.getTitle());

}

// Test data will not be hardcode in the step definition file

// We will always fetch the data from the feature file

@Then("I enter email id as {string} on the page")

**public** **void** i\_enter\_email\_id\_as\_seleniumdemo\_gmail\_com\_on\_the\_page(String EmailId) {

driver.findElement(By.*xpath*("//input[@id='input-email']")).sendKeys(EmailId);

}

@Then("I enter password as {string} on the page")

**public** **void** i\_enter\_password\_as\_on\_the\_page(String password) {

driver.findElement(By.*xpath*("//input[@id='input-password']")).sendKeys(password);

}

@Then("I click login button")

**public** **void** i\_click\_login\_button() {

driver.findElement(By.*xpath*("//input[@value='Login']")).click();

}

@Then("I close browser")

**public** **void** i\_close\_browser() {

driver.close();

}

@Then("I enter Input for the search box")

**public** **void** i\_enter\_input\_for\_the\_search\_box() {

driver.findElement(By.*xpath*("//input[@name='search']")).sendKeys("Mac");

}

@Then("I click on search button")

**public** **void** i\_click\_on\_search\_button() {

driver.findElement(By.*xpath*("//button[@class='btn btn-default btn-lg']")).click();

}

@Given("I click on Add to Cart button")

**public** **void** i\_click\_on\_add\_to\_cart\_button() {

driver.findElement(By.*linkText*("Shopping Cart")).click();

}

@Given("I click on Checkout button")

**public** **void** i\_click\_on\_checkout\_button() {

driver.findElement(By.*linkText*("Checkout")).click();

}

@Then("I click on logout link")

**public** **void** i\_click\_on\_logout\_link() {

driver.findElement(By.*linkText*("Logout")).click();

}

}

Code for Runner File

**package** runner;

**import** org.junit.runner.RunWith;

**import** io.cucumber.junit.Cucumber;

**import** io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.**class**)

@CucumberOptions(features="C:\\Users\\siva reddy\\eclipse-workspace\\Cucumber\\src\\test\\java\\features\\TagsDemo.feature",

glue= {"steps"},

dryRun=**false**,

plugin={"pretty",

"html:target/cucumberreport.html"},

tags="@all"

)

**public** **class** TestRunner {

}

OUTPUT:

