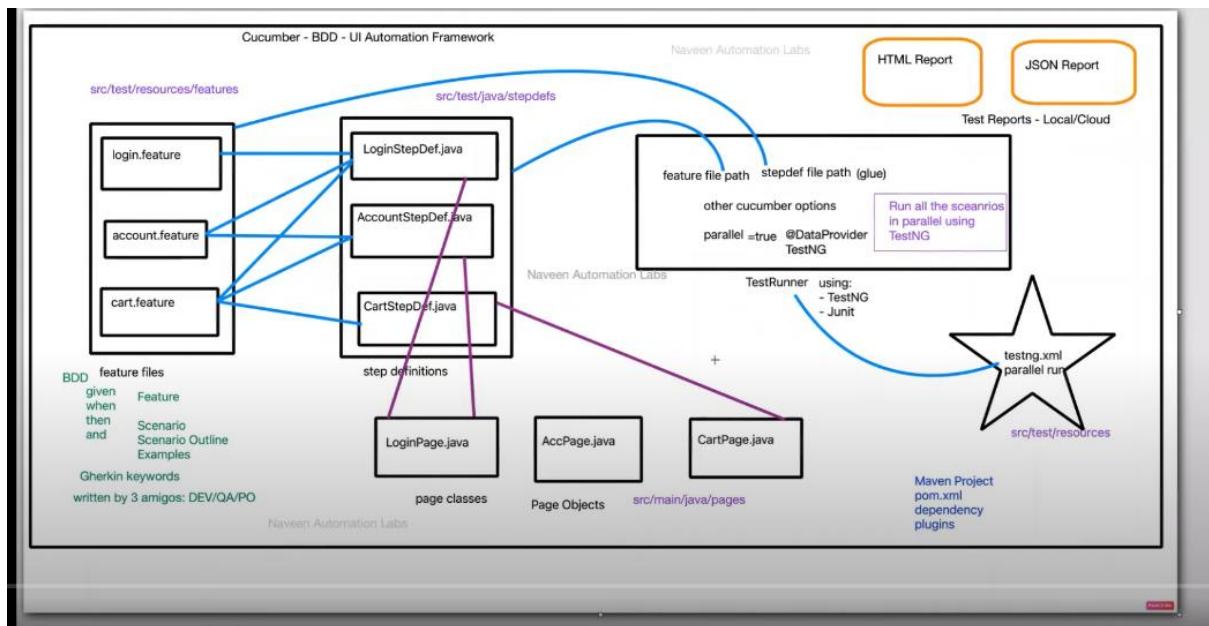


According to page object model, every page should have separate page class.

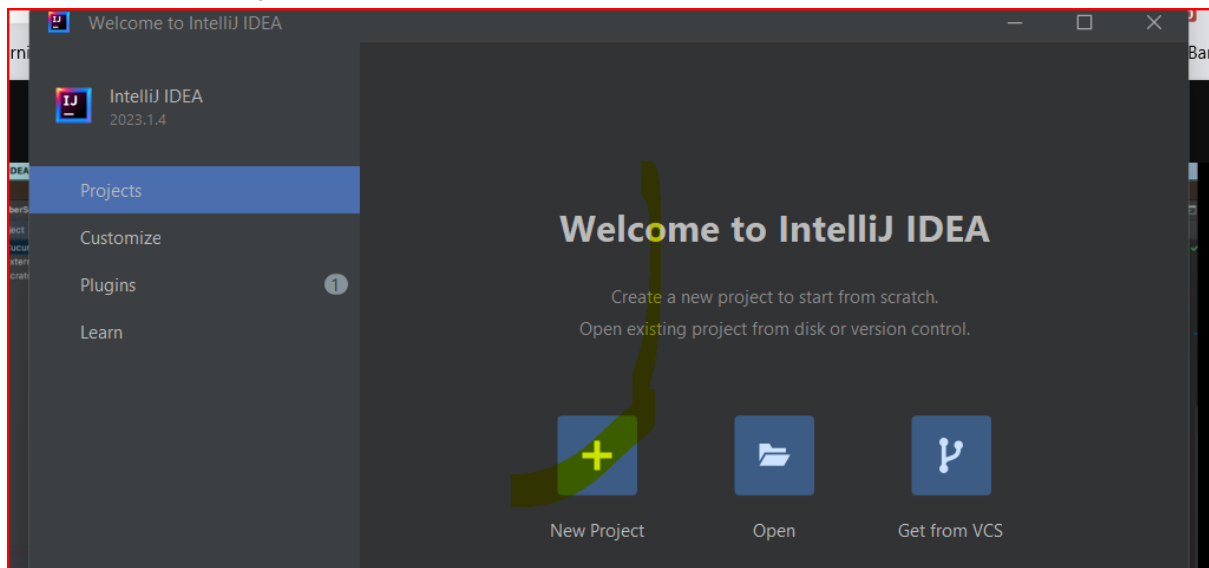
Selenium, playwright, cypress etc codes won't go into page step def, it will go into respective page class. In page class, we have the locators, ui actions, ui elements. Step def will call the page class. Features will call the step def.

Testng preferred because scenarios within feature file can run in parallel which is not possible in junit. Junit version 5 and above may support but not sure.

Runner files will be present in the same package as step def.

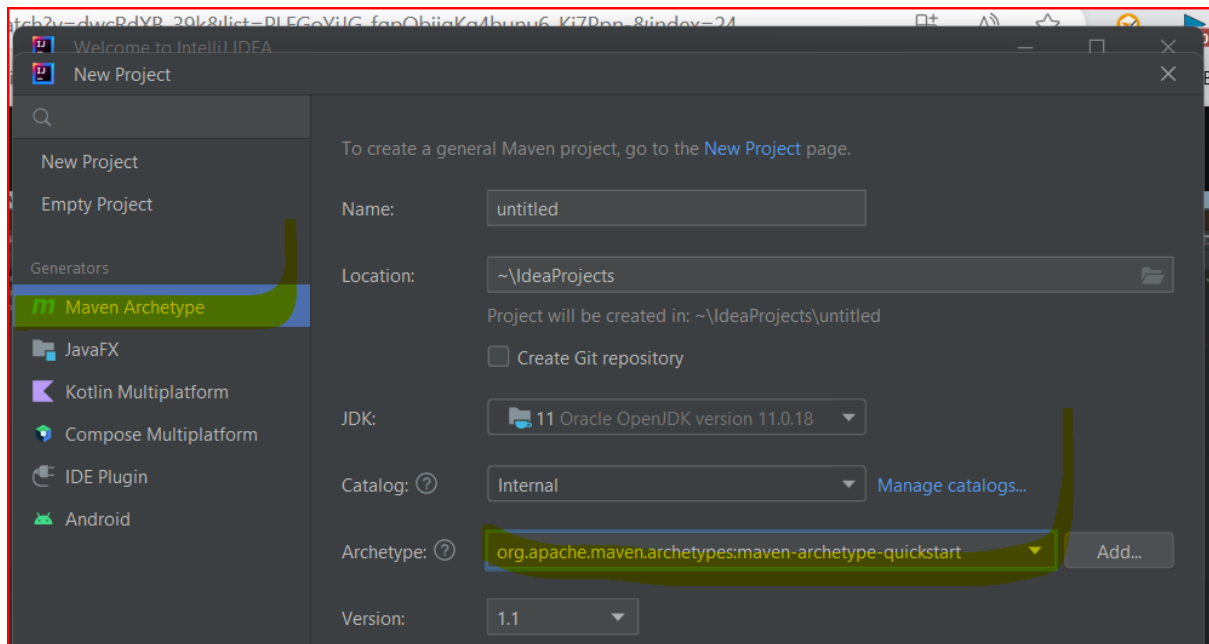


Let's use intellij-

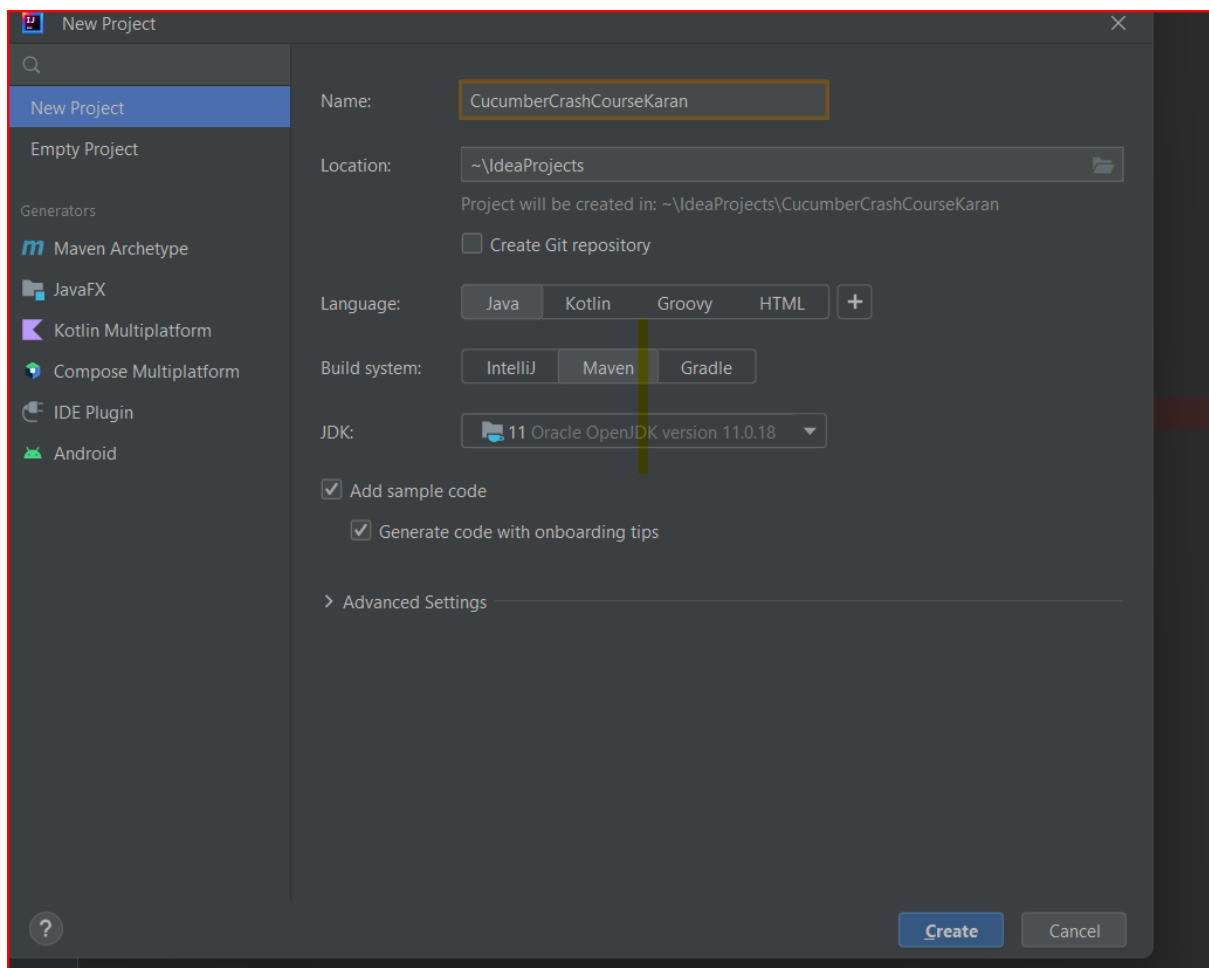


Select this maven archetype-

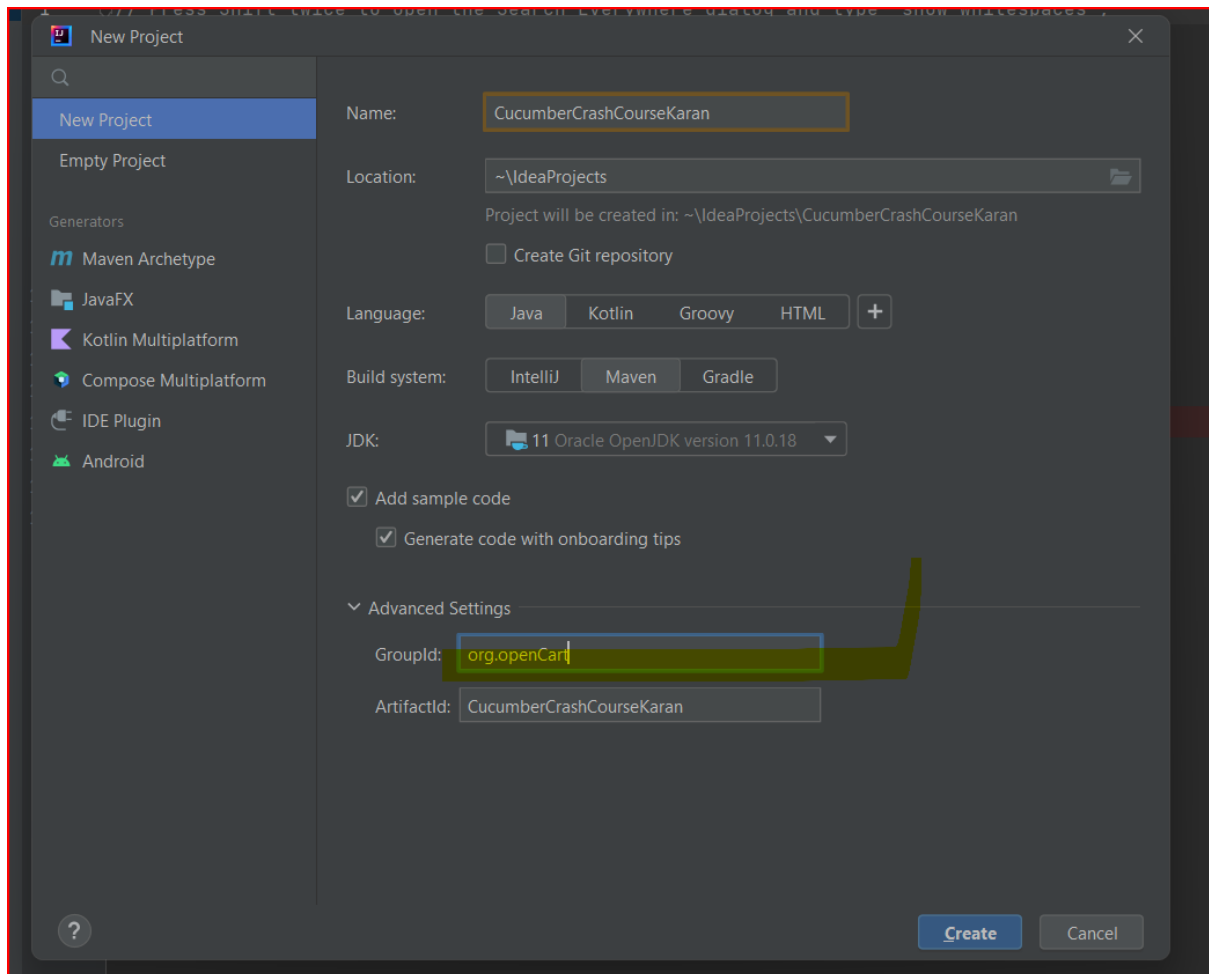
Give name here also, else it shows untitled.



Give name for project and select build system as maven-

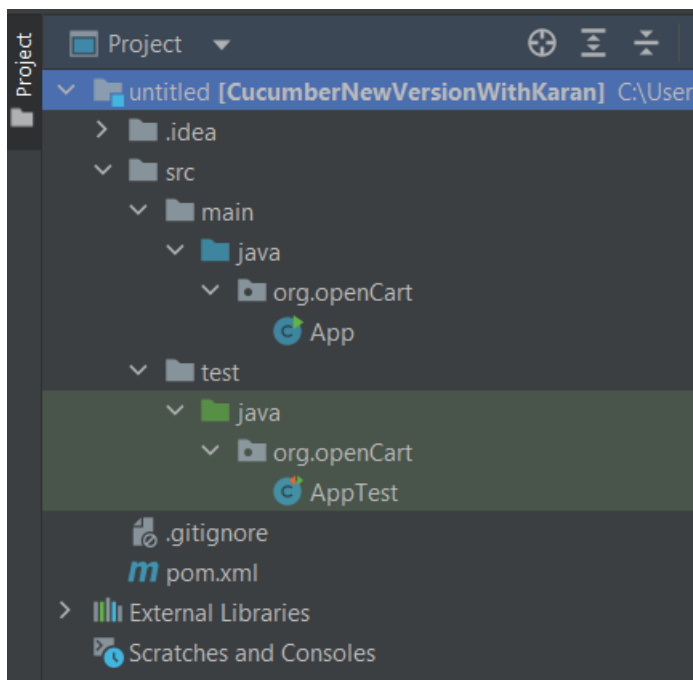


You can click advance settings. Then give group id and artifact id if you need.



We changed group id to the application name which is called as opencart.

Default skeleton created-



Delete the App and App test as seen above.

Now goto pom to add dependencies-

```

12
13     <properties>
14         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15         <maven.compiler.source>11</maven.compiler.source>
16         <maven.compiler.target>11</maven.compiler.target>
17     </properties>

```

Dependencies added for cucumber with java, cucumber with testng, testng, selenium java-

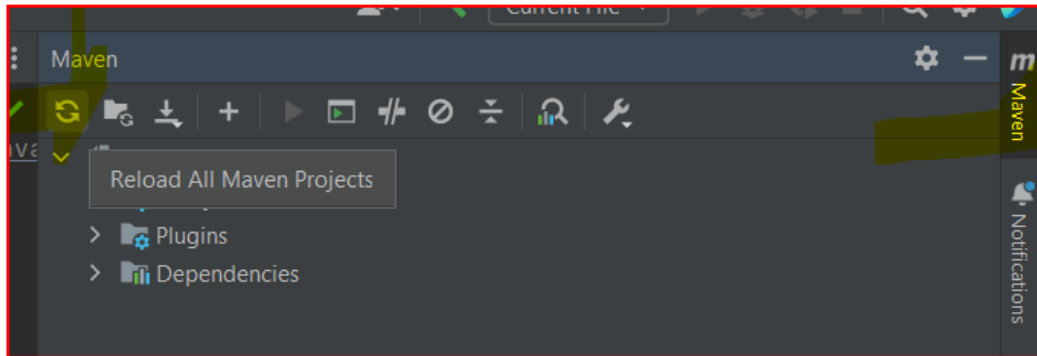
```

19     <dependencies>
20         <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
21         <dependency>
22             <groupId>io.cucumber</groupId>
23             <artifactId>cucumber-java</artifactId>
24             <version>7.13.0</version>
25         </dependency>
26         <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-testng -->
27         <dependency>
28             <groupId>io.cucumber</groupId>
29             <artifactId>cucumber-testng</artifactId>
30             <version>7.13.0</version>
31         </dependency>
32         <!-- https://mvnrepository.com/artifact/org.testng/testng -->
33         <dependency>
34             <groupId>org.testng</groupId>
35             <artifactId>testng</artifactId>
36             <version>7.8.0</version>
37             <scope>test</scope>
38         </dependency>
39         <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
40         <dependency>
41             <groupId>org.seleniumhq.selenium</groupId>
42             <artifactId>selenium-java</artifactId>
43             <version>4.10.0</version>
44         </dependency>
45     </dependencies>

```

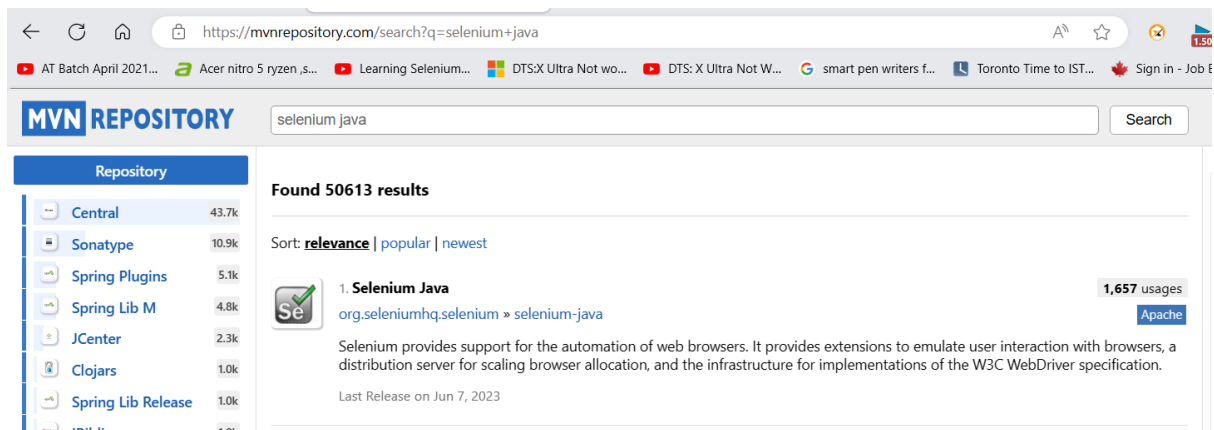
Once added in right hand side click maven. Reload the project. This is refresh in eclipse.

Then it will show you to install gherkin and cucumber java tool, tell yes and install and restart ide.



Note for searching inside maven repo-

Search inside mvn repo using artifact name. see few pictures above this, we can see selenium has artifact name as selenium java.

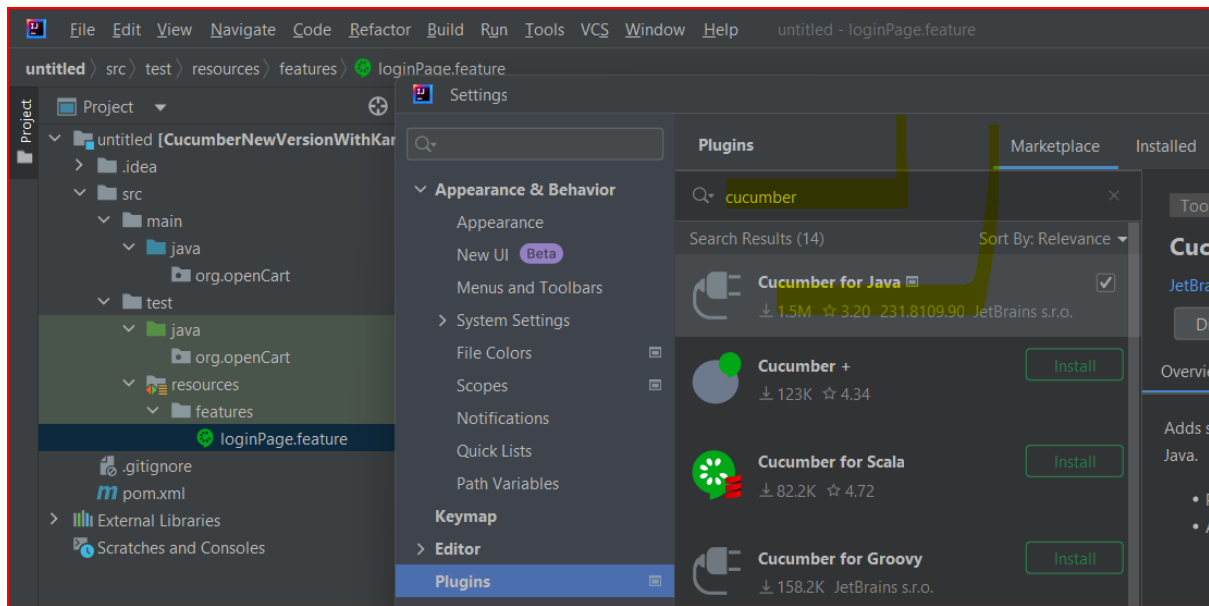


Note-

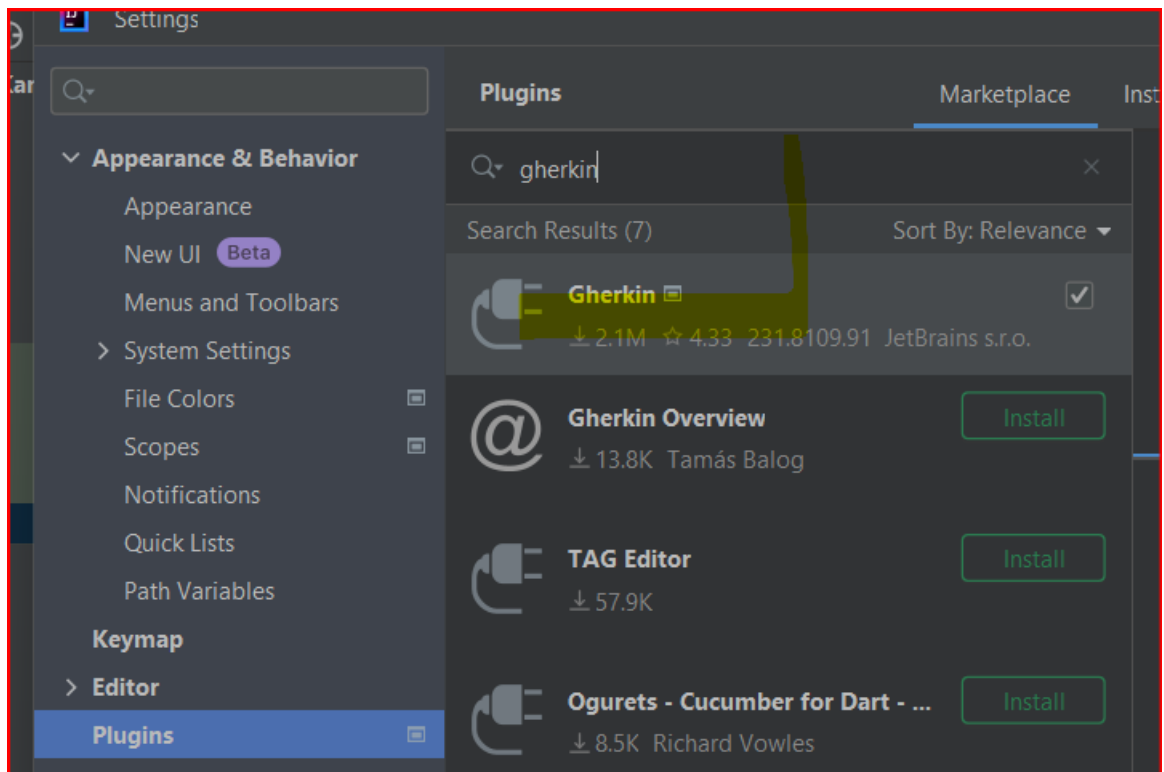
All file names should have camel case convention. Example loginPage.feature.

To get the auto suggestions for given when then etc-

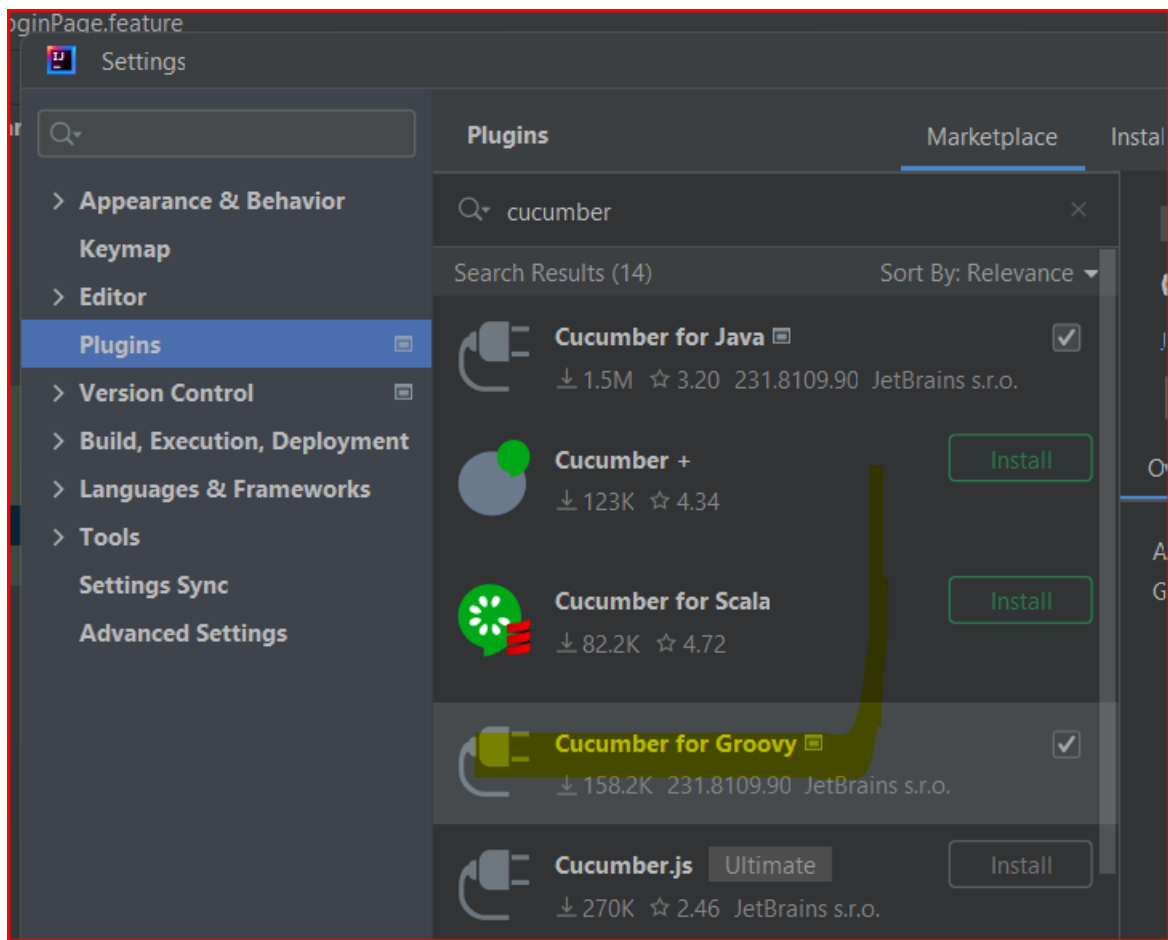
File -> settings -> search for cucumber -> install it.



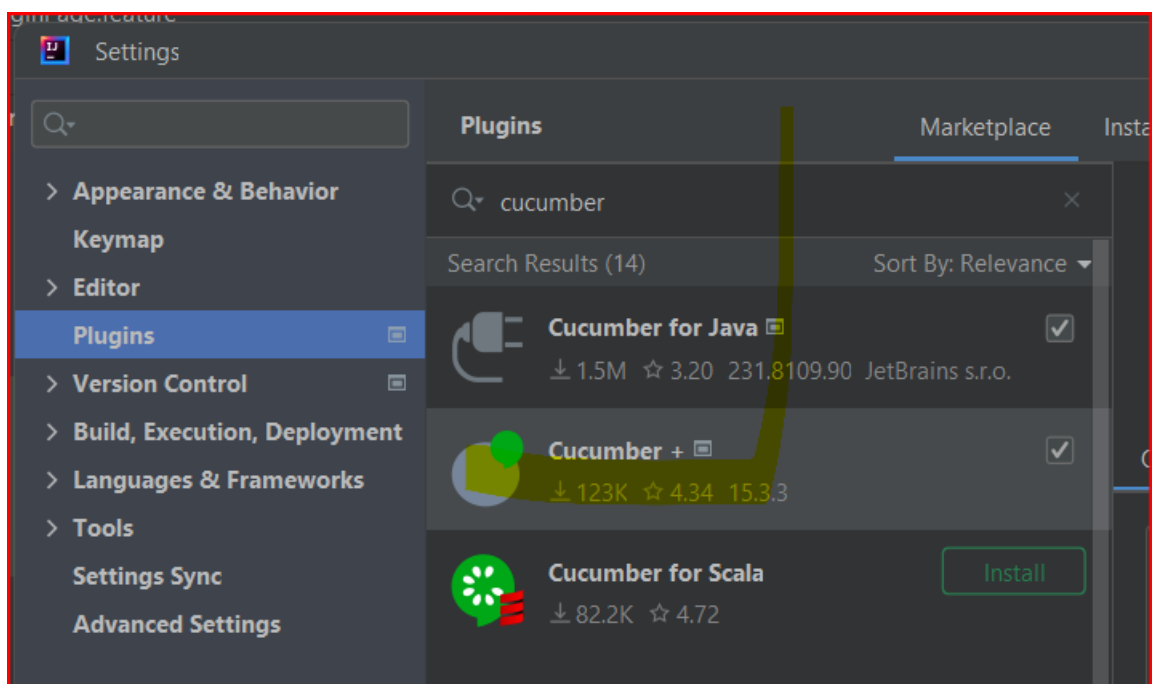
Install gherkin also.



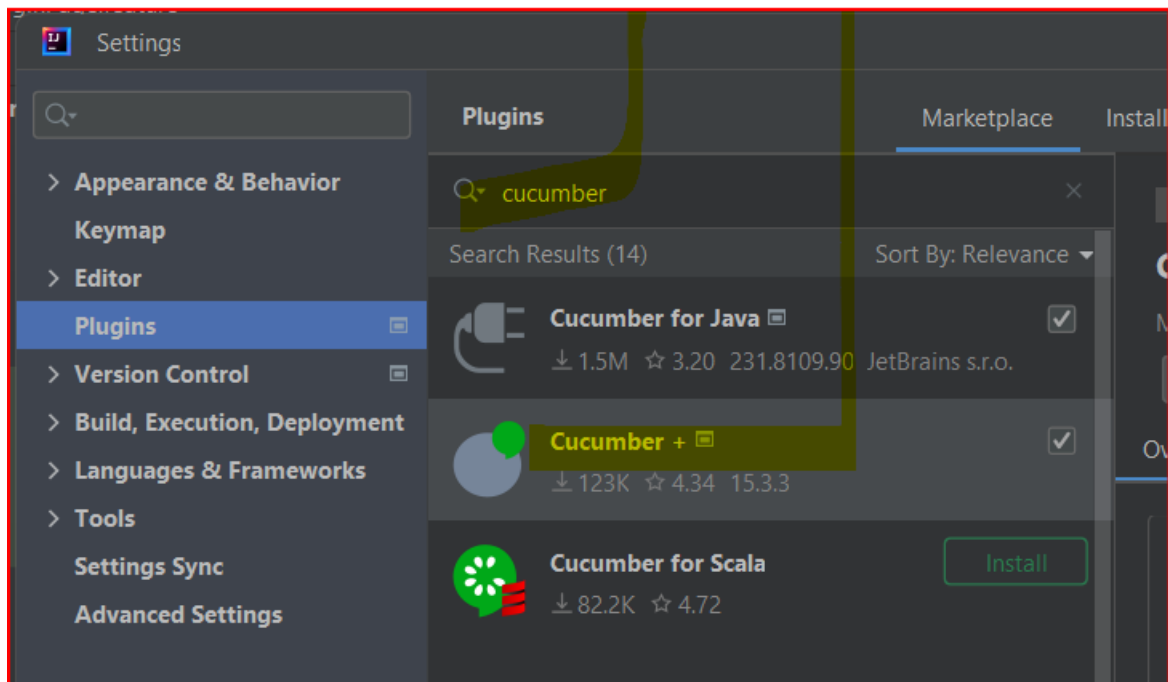
Cucumber for groovy also-



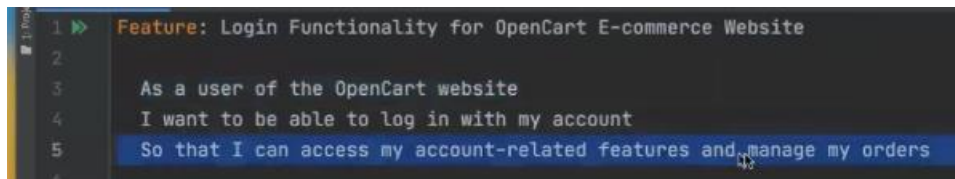
Install cucumber + -



Ocd-



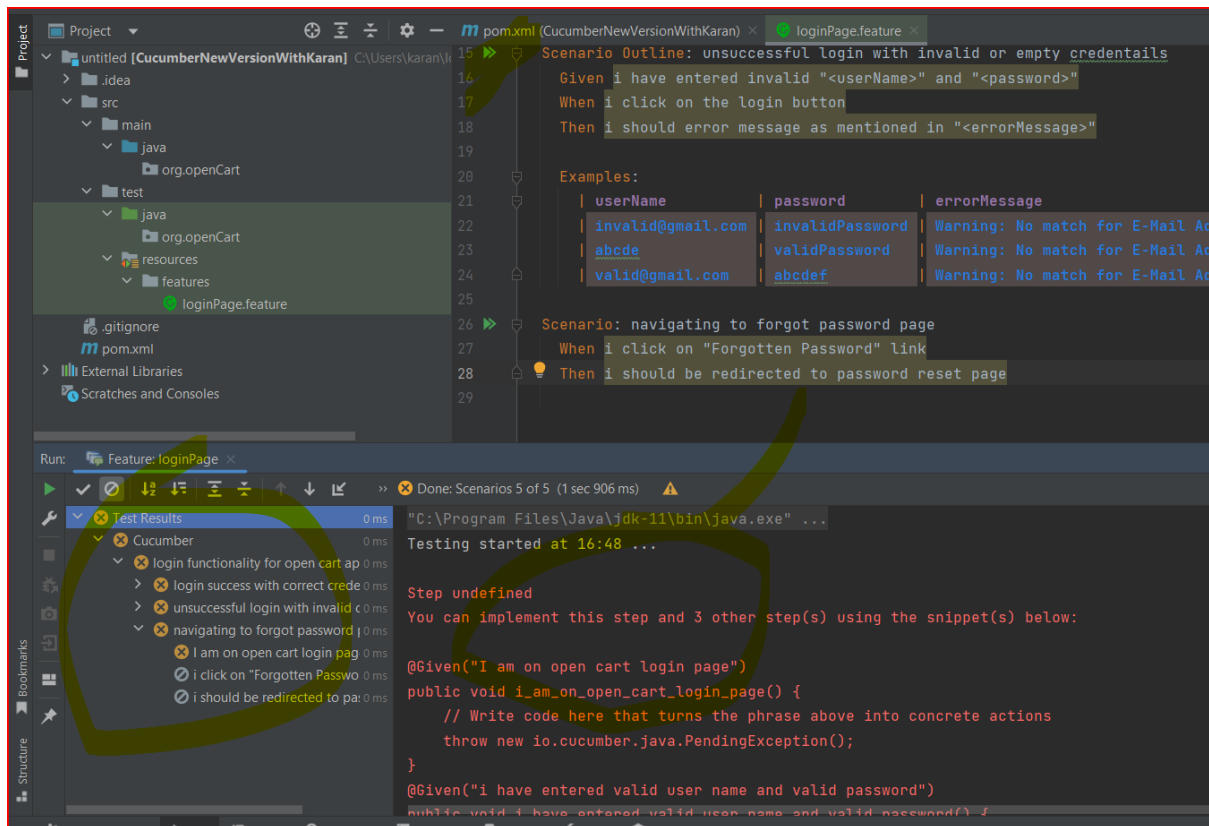
See the feature name and its description-



Scenario is normal scenario.

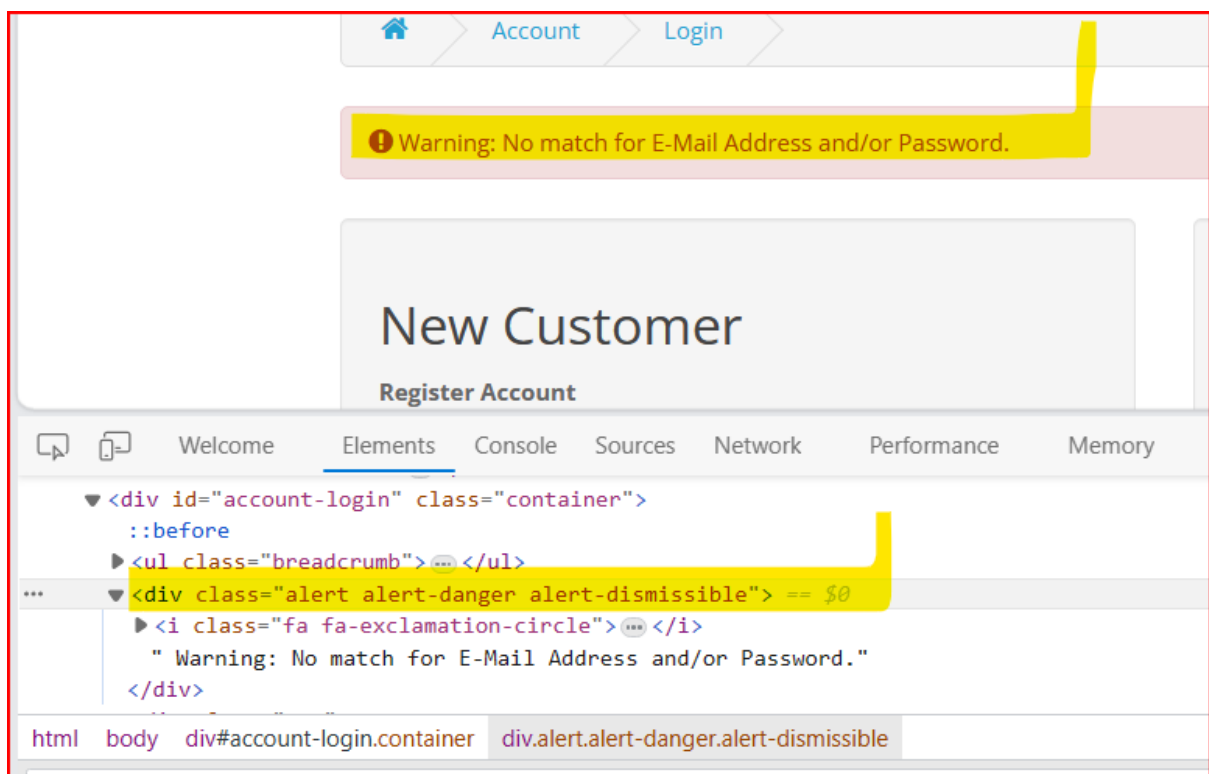
Scenario outline is used for parameterisation.

Run feature file and see in left side we get all test failed and in right hand side in console the error messages-

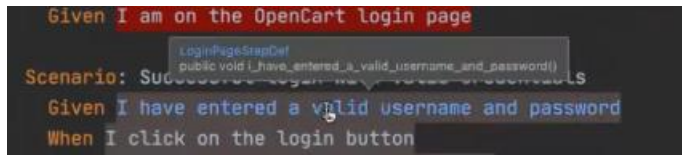


See the locator for the error message-

There are three classes and we can use any one which will return unique element.

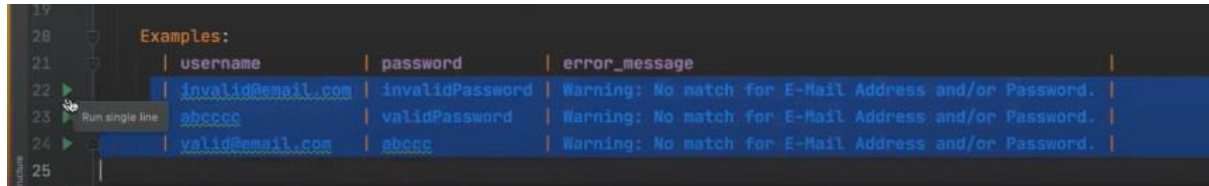


Press “control+click” on respective line in feature file and you go to the respective step def.



Good feature in IntelliJ-

We can run all lines or specific scenario also.



Note-

Assertions should be written inside step def only.

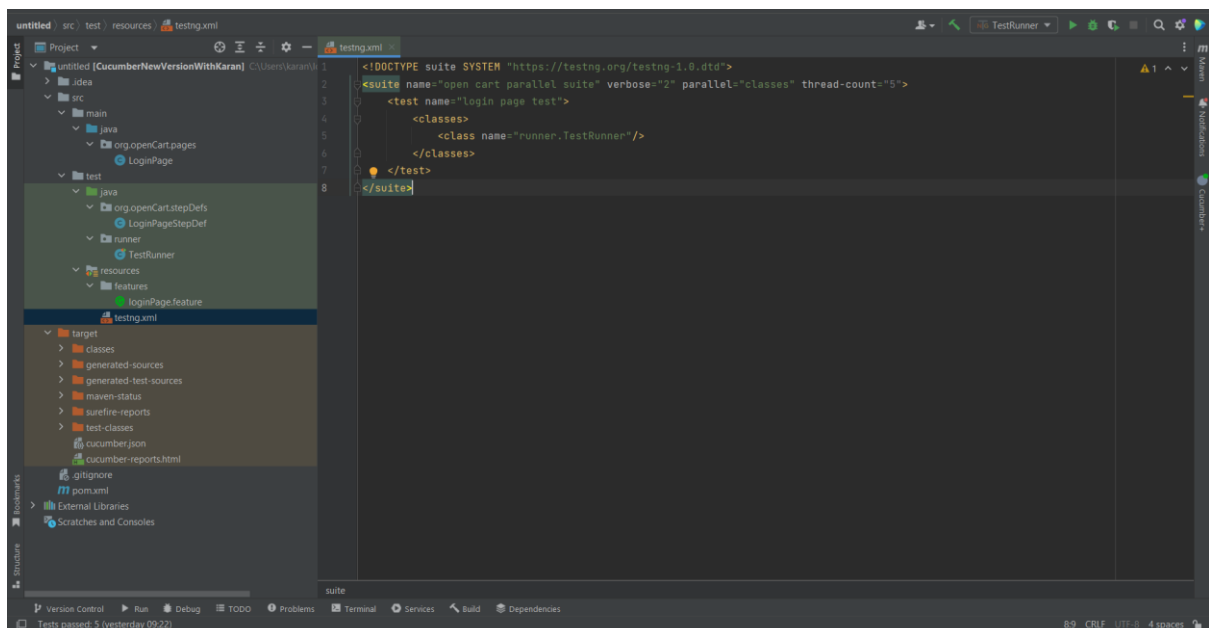
Page classes should not have any assertions.

Page classes should have page actions or page methods only, selenium methods, browser specific methods etc.

“@then” used generally used for assertion.

Testng xml file-

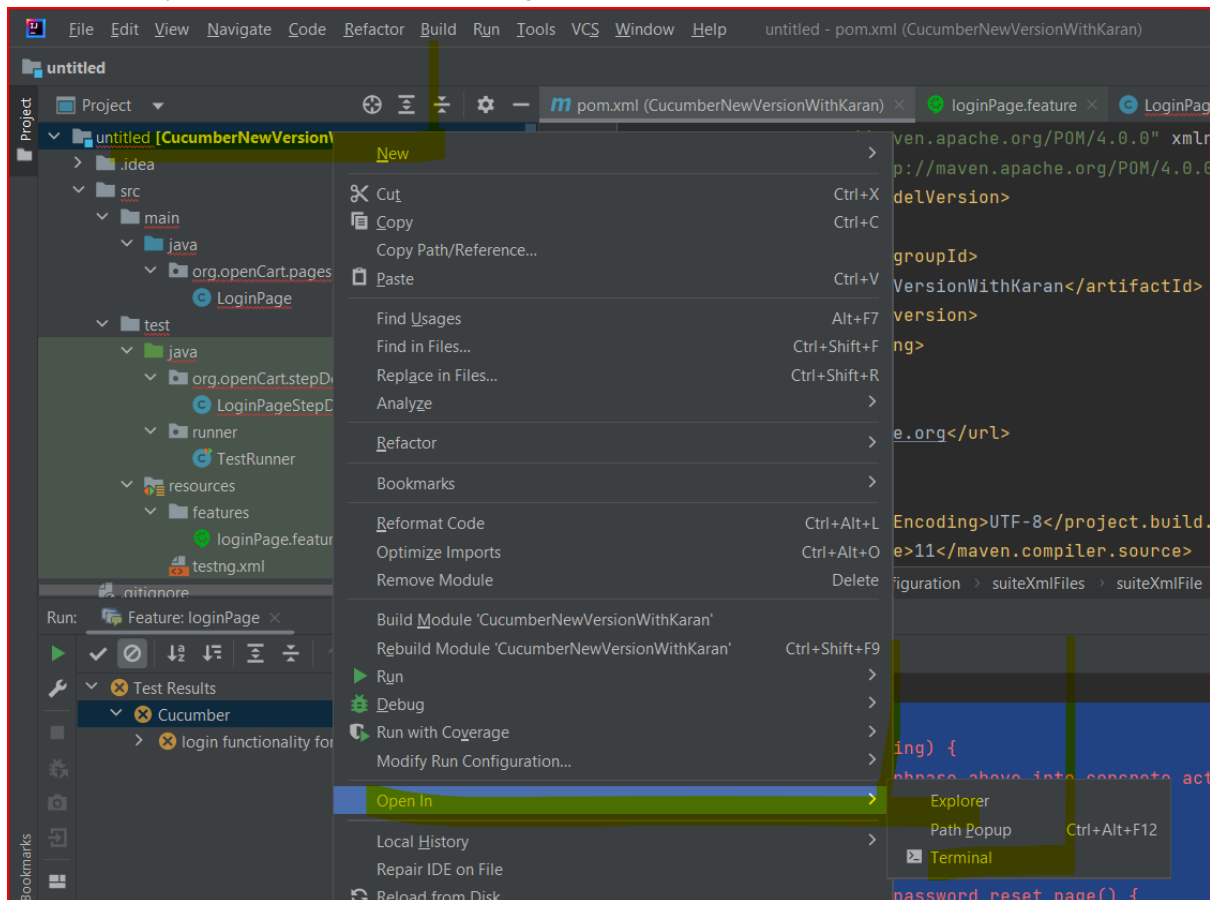
This is to give parallel configuration like how many threads to be used.



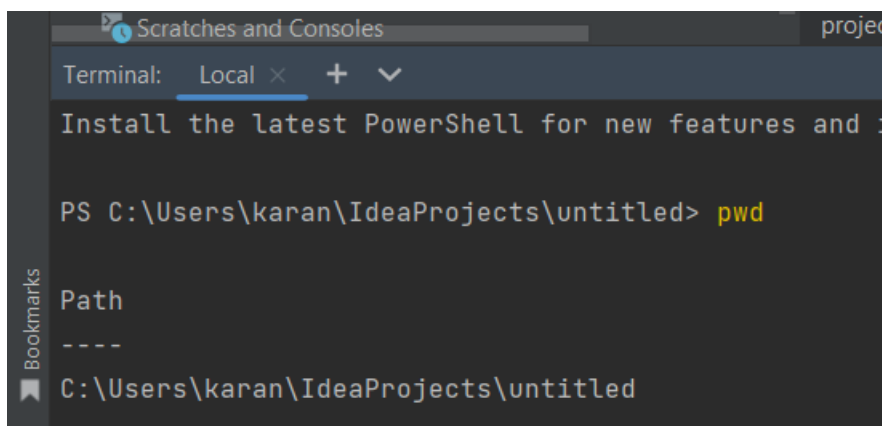
Verbose in testng xml file, will tell how much detailed logset we need to generate.

Now we can run the testng.xml file or directly run from maven.

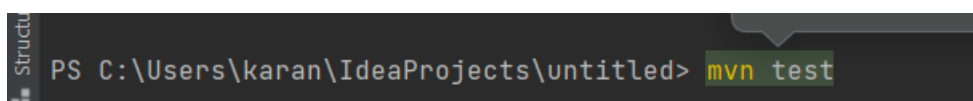
How to open terminal in intellij itself-



Tried pwd command-



Try the below command-



How to run specific scenario in cucumber-

Add tag in feature file.

```

25
26     @Test1
27     Scenario: navigating to forgot password page
28         When i click on "Forgotten Password" link
29         Then i should be redirected to password reset page
30

```

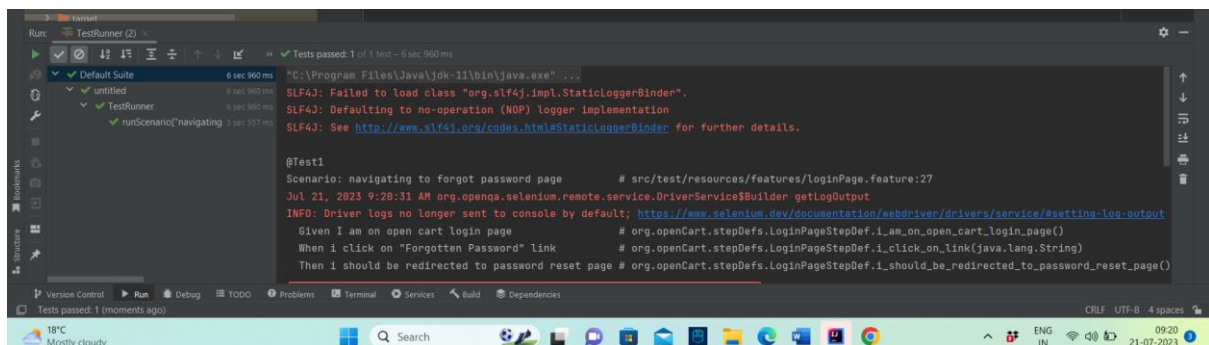
In runner file add the tags.

```

@CucumberOptions(features = "src/test/resources/features",
    glue={"org.openCart.stepDefs"},
    plugin={"pretty", "html:target/cucumber-reports", "json:target/cucumber.json"},
    tags = "@Test1",
    monochrome = true,
    publish = true)

```

Run it as it works.



Curly brackets not mandatory for those fields in runner file, when there is only one value-

```

7     @CucumberOptions(features = "src/test/resources/features",
8         glue="org.openCart.stepDefs",
9         plugin={"pretty", "html:target/cucumber-reports", "json:target/cucumber.json"},
10        monochrome = true,
11        publish = true)
12

```

Run the first set of code using maven-

```

Pom -
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.openCart</groupId>
    <artifactId>CucumberNewVersionWithKaran</artifactId>
    <version>1.0-SNAPSHOT</version>

```

```

<packaging>jar</packaging>

<name>untitled</name>
<url>http://maven.apache.org</url>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>11</maven.compiler.source>
  <maven.compiler.target>11</maven.compiler.target>
</properties>

<dependencies>
  <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
  <dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-java</artifactId>
    <version>7.13.0</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-testng -->
  <dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-testng</artifactId>
    <version>7.13.0</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.testng/testng -->
  <dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>7.8.0</version>
    <scope>test</scope>
  </dependency>
  <!--
https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java
-->
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.10.0</version>
  </dependency>
</dependencies>

<build>
  <plugins>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.20</version>
  <configuration>
    <suiteXmlFiles>
      <suiteXmlFile>src/test/resources/testng.xml</suiteXmlFile>
    </suiteXmlFiles>
  </configuration>
</plugin>
  </plugins>
</build>

</project>

```

Login feature-

Feature: login functionality for open cart application

As a user of opencart
 I want to be able to login to my account
 So that I can access my account features and manage my orders

Background:

Given I am on open cart login page

Scenario: login success with correct credentials

Given i have entered valid user name and valid password

When i click on login button

Then i should be logged in successfully

Scenario Outline: unsuccessful login with invalid or empty credentials

Given i have entered invalid "<userName>" and "<password>"

When i click on login button

Then i should error message as mentioned in "<errorMessage>"

Examples:

userName	password	errorMessage
invalid@gmail.com	invalidPassword	Warning: No match for E-Mail Address and/or Password.
abcde	validPassword	Warning: No match for E-Mail Address and/or Password.
valid@gmail.com	abcdef	Warning: No match for E-Mail Address and/or Password.

Scenario: navigating to forgot password page

When i click on "Forgotten Password" link

Then i should be redirected to password reset page

Login page object-

```
package org.opencart.pages;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

/**
 * here we write page locators, page actions
 */
public class LoginPage {

    private WebDriver driver;

    //by locators
    private By emailInputLocator=By.name("email");
    private By passwordInputLocator=By.name("password");
    private By loginButtonLocator=By.xpath("//input[@type='submit']");
    private By forgotPasswordLinkLocator=By.linkText("Forgotten Password");
    private By logoutLinkLocator=By.linkText("Logout");

    //constructor
    public LoginPage(WebDriver driver){
        this.driver=driver;
    }

    //page methods or page actions
    //every page method public in nature
    //page objects classic example of encapsulation. private by locators
```

accessed and used by public methods.

```

public void enterEmail(String emailField){
    WebElement emailInput= driver.findElement(emailInputLocator);
    emailInput.sendKeys(emailField);
}

public void enterPassword(String passwordField){
    WebElement passwordInput=
driver.findElement(passwordInputLocator);
    passwordInput.sendKeys(passwordField);
}

public void clickLoginButton(){
    WebElement loginButton= driver.findElement(loginButtonLocator);
    loginButton.click();
}

public boolean checkForgotPasswordLink(){
    return
driver.findElement(forgotPasswordLinkLocator).isDisplayed();
}

public boolean checkLogoutLink(){
    return driver.findElement(logoutLinkLocator).isDisplayed();
}

public void clickForgotPasswordLink(){
    WebElement forgotPasswordLink=
driver.findElement(forgotPasswordLinkLocator);
    forgotPasswordLink.click();
}

public void login(String userName, String password){
    enterEmail(userName);
    enterPassword(password);
    clickLoginButton();
}

public String getForgotPasswordPageURL(){
    String forgotPasswordPageURL= driver.getCurrentUrl();
    return forgotPasswordPageURL;
}
}

```

Login page steps-

```

package org.openCart.stepDefs;

import io.cucumber.java.After;
import io.cucumber.java.Before;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import org.openCart.pages.LoginPage;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;

/**
 * here we will write step def code with help of cucumber annotations

```



```

* we write selenium/java code also.
* this will also call the page methods from the respective page class.
*/
public class LoginPageStepDef {

    private WebDriver driver;
    private LoginPage loginPage;

    //before and after annotations are known as hooks in cucumber.

    @Before
    public void setUp() {
        driver=new ChromeDriver();
    }

    @After
    public void tearDown() {
        if(driver!=null) {
            driver.quit();
        }
    }

    @Given("I am on open cart login page")
    public void i_am_on_open_cart_login_page() {

driver.get("https://naveenautomationlabs.com/opencart/index.php?route=acc
ount/login");
        loginPage=new LoginPage(driver); // we create object of this
login page class so we can access methods from the
        //page class.
    }

    @Given("i have entered valid user name and valid password")
    public void i_have_entered_valid_user_name_and_valid_password() {
        loginPage.enterEmail("cucumbertesting1960@gmail.com");
        loginPage.enterPassword("Malaravi@123");
    }

    @When("i click on login button")
    public void i_click_on_login_button() {
        loginPage.clickLoginButton();
    }

    @Then("i should be logged in successfully")
    public void i_should_be_logged_in_successfully() {
        Assert.assertEquals(loginPage.checkLogoutLink(), true);
    }

    /**
     * when a method being called returns boolean, then better to check
the validation with assert statement
     * in the calling class.
     * @param userName
     * @param password
     */

    @Given("i have entered invalid {string} and {string}")
    public void i_have_entered_invalid_and(String userName, String
password) {
        loginPage.enterEmail(userName);
        loginPage.enterPassword(password);
    }
}

```

```

        @Then("i should error message as mentioned in {string}")
        public void i_should_error_message_as_mentioned_in(String
errorMessage) {
            //Assert that error on page is equal to error we are expecting
            Assert.assertEquals(driver.findElement(By.cssSelector(".alert-
danger")).isDisplayed(), true);
        }

        @When("i click on {string} link")
        public void i_click_on_link(String string) {
            loginPage.clickForgotPasswordLink();
        }

        @Then("i should be redirected to password reset page")
        public void i_should_be_redirected_to_password_reset_page() {
            //System.out.println(loginPage.getForgotPasswordPageURL());

Assert.assertTrue(loginPage.getForgotPasswordPageURL().contains("account/
forgotten"));
            //one trick for remembering whether to use assert equals or
assert true is check return type.
            //the getforgotpasswordpageurl returns string so we check if the
string is present or not.
        }
    }
}

```

Test runner-

```

package runner;

import io.cucumber.testng.AbstractTestNGCucumberTests;
import io.cucumber.testng.CucumberOptions;
import org.testng.annotations.DataProvider;

@CucumberOptions(features = "src/test/resources/features",
glue="org.openCart.stepDefs",
plugin={"pretty", "html:target/cucumber-reports",
"json:target/cucumber.json"}, // we are generating json and html reports.
monochrome = true,
publish = true)

//monochrome will give good output format in console.
//publish will help to publish in cloud.

//to run test cases in parallel mode extend the class
abstracttestngcucumbertests.
public class TestRunner extends AbstractTestNGCucumberTests {

    //this will help in running all scenarios in parallel mode.
    //name of the method should exactly match. if we scenario instead of
scenarios then override throws error
    //"method does not override from its superclass"
    //this will collect all scenarios and then run them in parallel.
    @Override
    @DataProvider(parallel = true)
    public Object[][] scenarios() {
        return super.scenarios();
    }
}

```

Testng xml –

```

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="open cart parallel suite" verbose="2" parallel="classes"
thread-count="5">

```

```

    <test name="login page test">
      <classes>
        <class name="runner.TestRunner"/>
      </classes>
    </test>
  </suite>

```

Output-

```

Terminal: Local x Local (2) x Local (3) x + v
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 13.346 s
[INFO] Finished at: 2023-07-21T09:36:42+02:00
[INFO] -----
[WARNING]
[WARNING] Plugin validation issues were detected in 3 plugin(s)
[WARNING]
[WARNING] * org.apache.maven.plugins:maven-compiler-plugin:3.10.1
[WARNING] * org.apache.maven.plugins:maven-surefire-plugin:2.20
[WARNING] * org.apache.maven.plugins:maven-resources-plugin:3.3.0
[WARNING]
[WARNING] For more or less details, use 'maven.plugin.validation' property with one of the
[WARNING]
PS C:\Users\karan\IdeaProjects\untitled>
Version Control Run Debug TODO Problems Terminal Services Build Dependencies
Tests passed: 5 (yesterday 09:22)

```

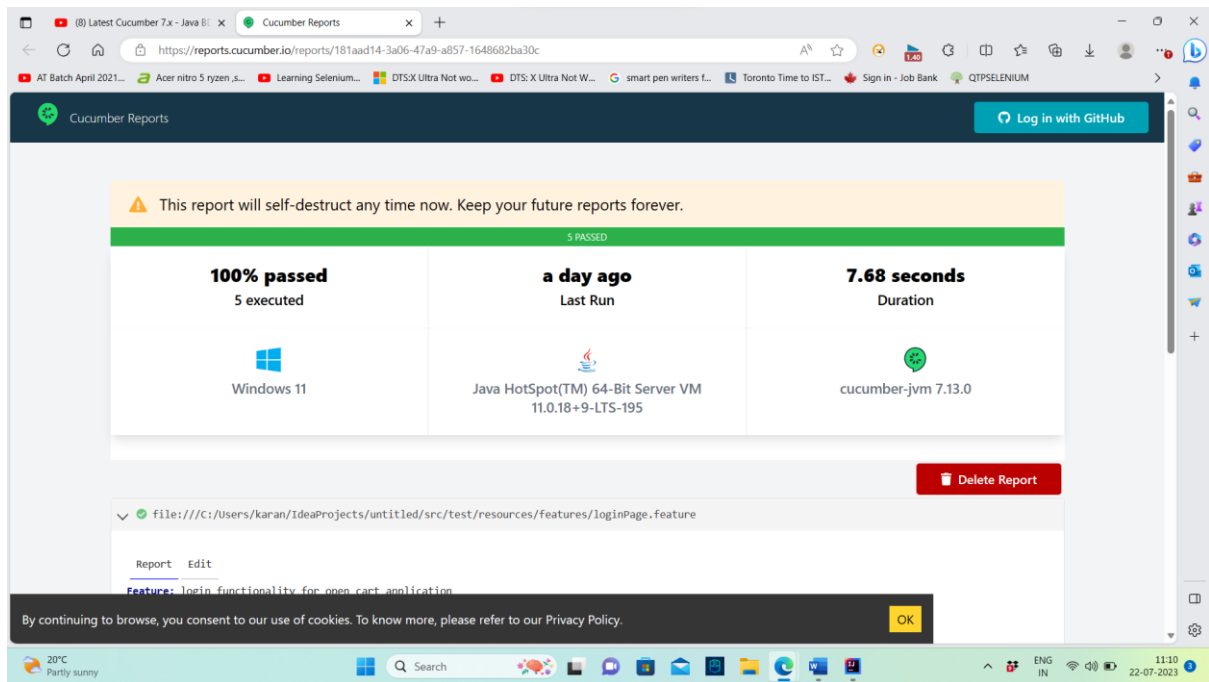
This is the cucumber cloud report which we are talking-

```

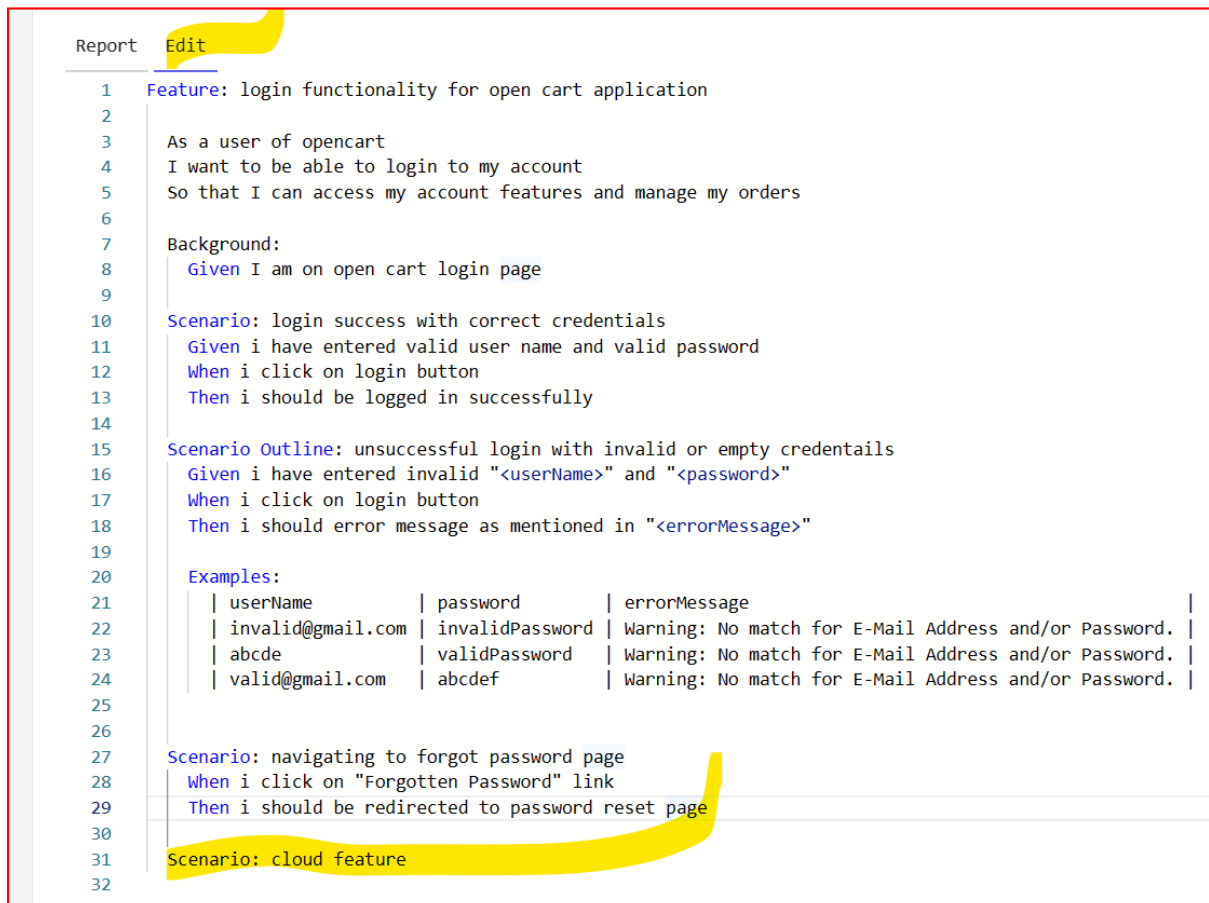
????????????????????????????????????????????????????????????????????????????????????
? View your Cucumber Report at: ?
? https://reports.cucumber.io/reports/181aad14-3a86-47a9-a897-1a48682ba38c ?
?
? This report will self-destruct in 24h. ?
? Keep reports forever: https://reports.cucumber.io/profile ?
????????????????????????????????????????????????????????????????????????????????????[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 9.768 s - in TestSuite
[success]

```

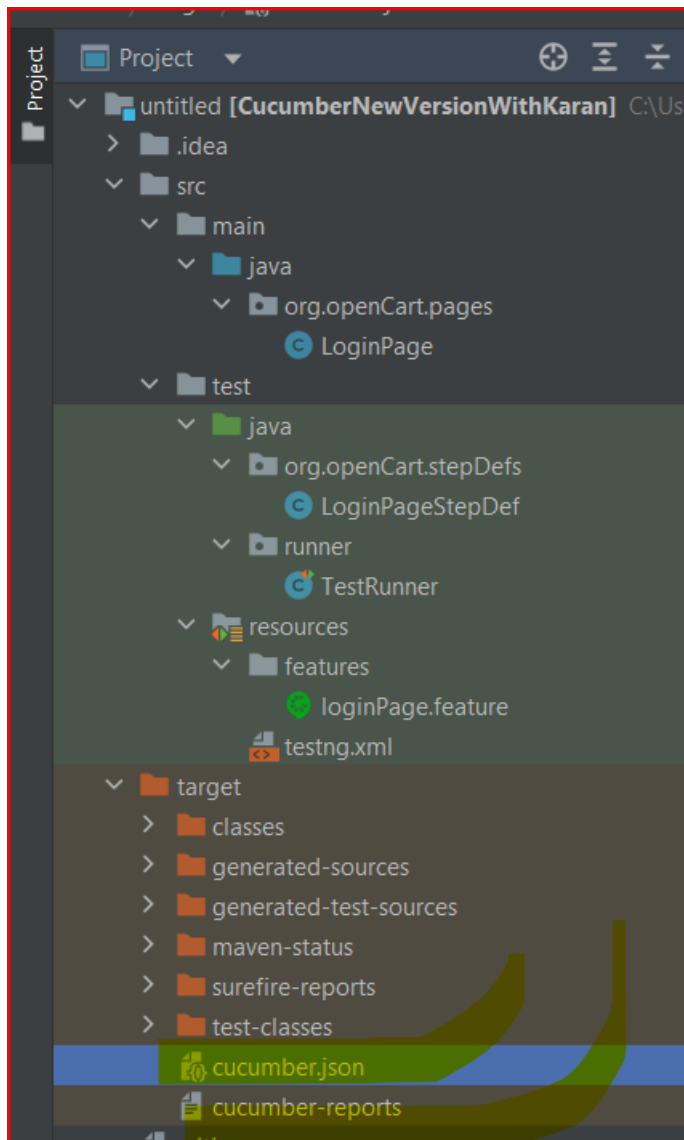
See how report looks when opened-



We can edit the feature files on cloud also and write scenarios if required-

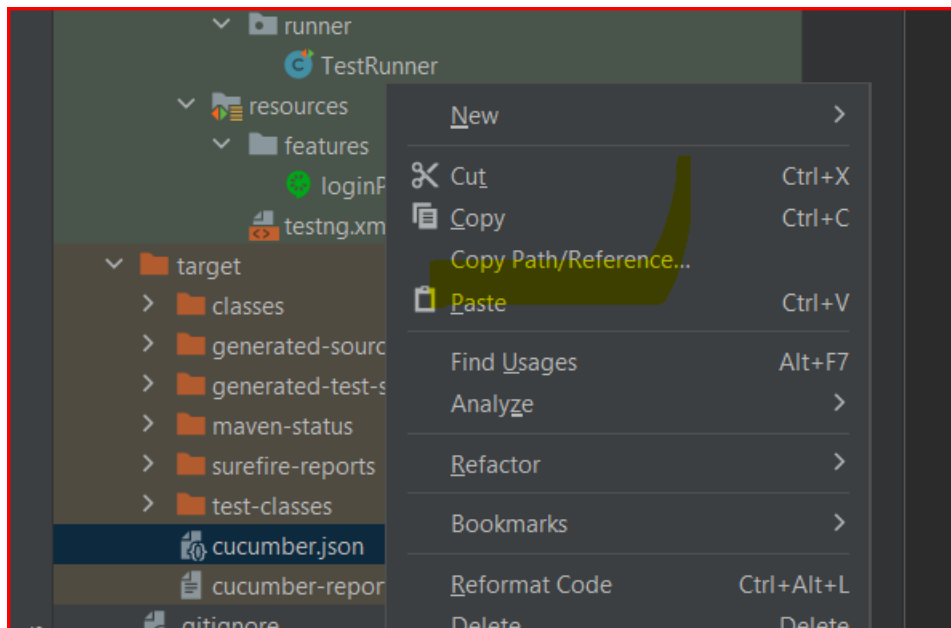


Let's see if json report and html report generated in target folder-

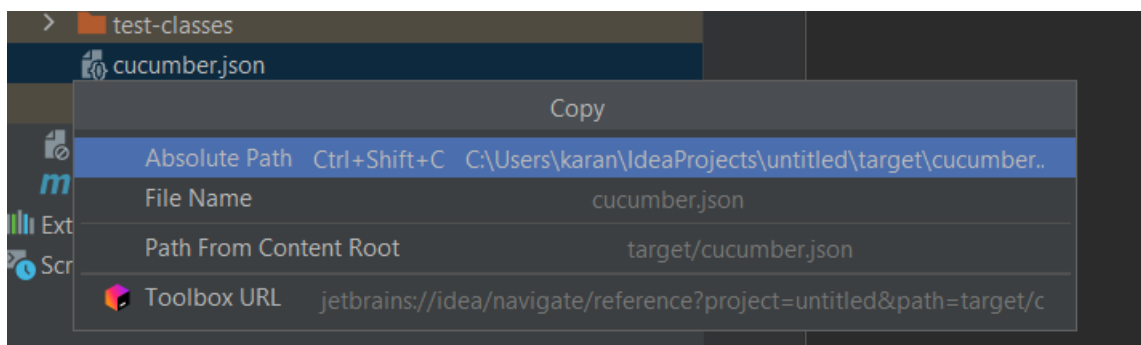


Right click on report.

Then copy path reference.



Copy absolute path.



Entire path will be copied.

Paste in browser.

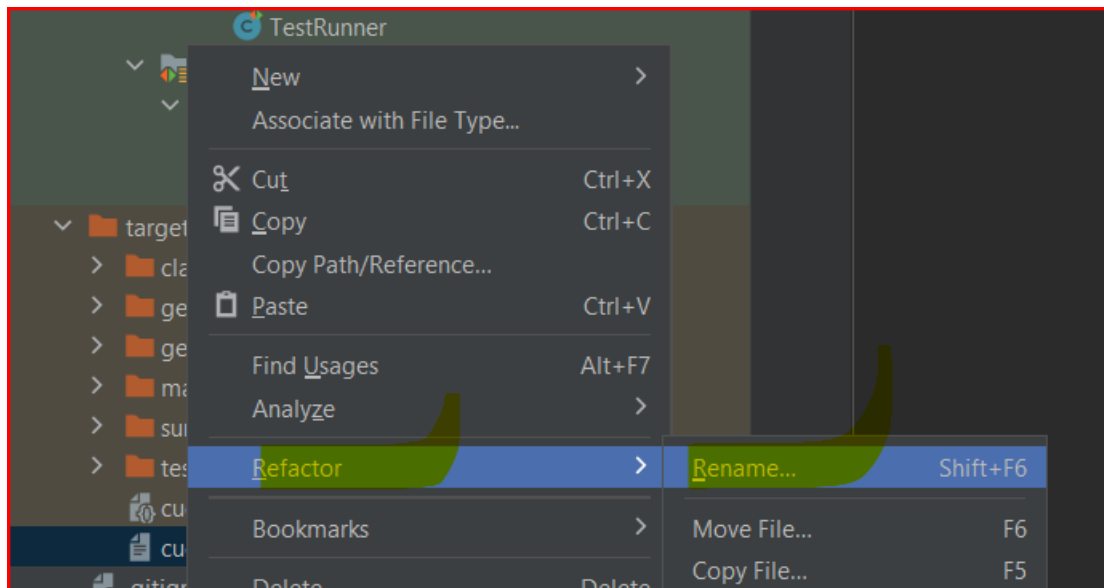
Json report in json format-

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Cucumber</title>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="icon" type="image/svg+xml" id="cucumber" data-bbox="102 102 903 400">
```

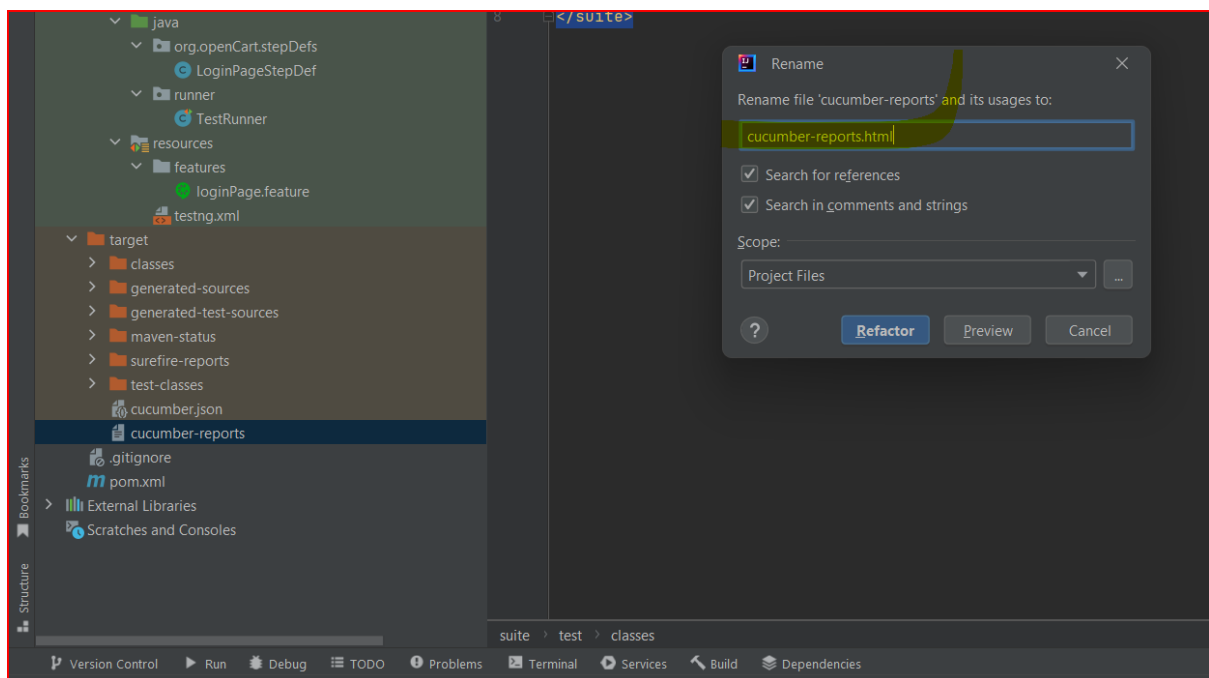
We want in nice html.

Refactor and rename the report.

Refactor and rename the report.



Rename the report and add “.html”



Done.

Reopen the report in browser and voila.

5 PASSED

100% passed 5 executed	1 day ago last run	7.69 seconds duration
Windows 11	Java HotSpot(TM) 64-Bit Server VM 11.0.18+9-LTS-195	cucumber-jvm 7.13.0

Search with text or @tags
You can search with plain text or [Cucumber Tag Expressions](#) to filter the output

file:///C:/Users/karan/IdeaProjects/untitled/src/test/resources/features/loginPage.feature

Feature: login functionality for open cart application

As a user of open cart
I want to be able to login to my account
So that I can access my account features and manage my orders

Background:

- Given I am on open cart login page

Scenario: login success with correct credentials

- Given i have entered valid user name and valid password
- When i click on login button
- Then i should be logged in successfully

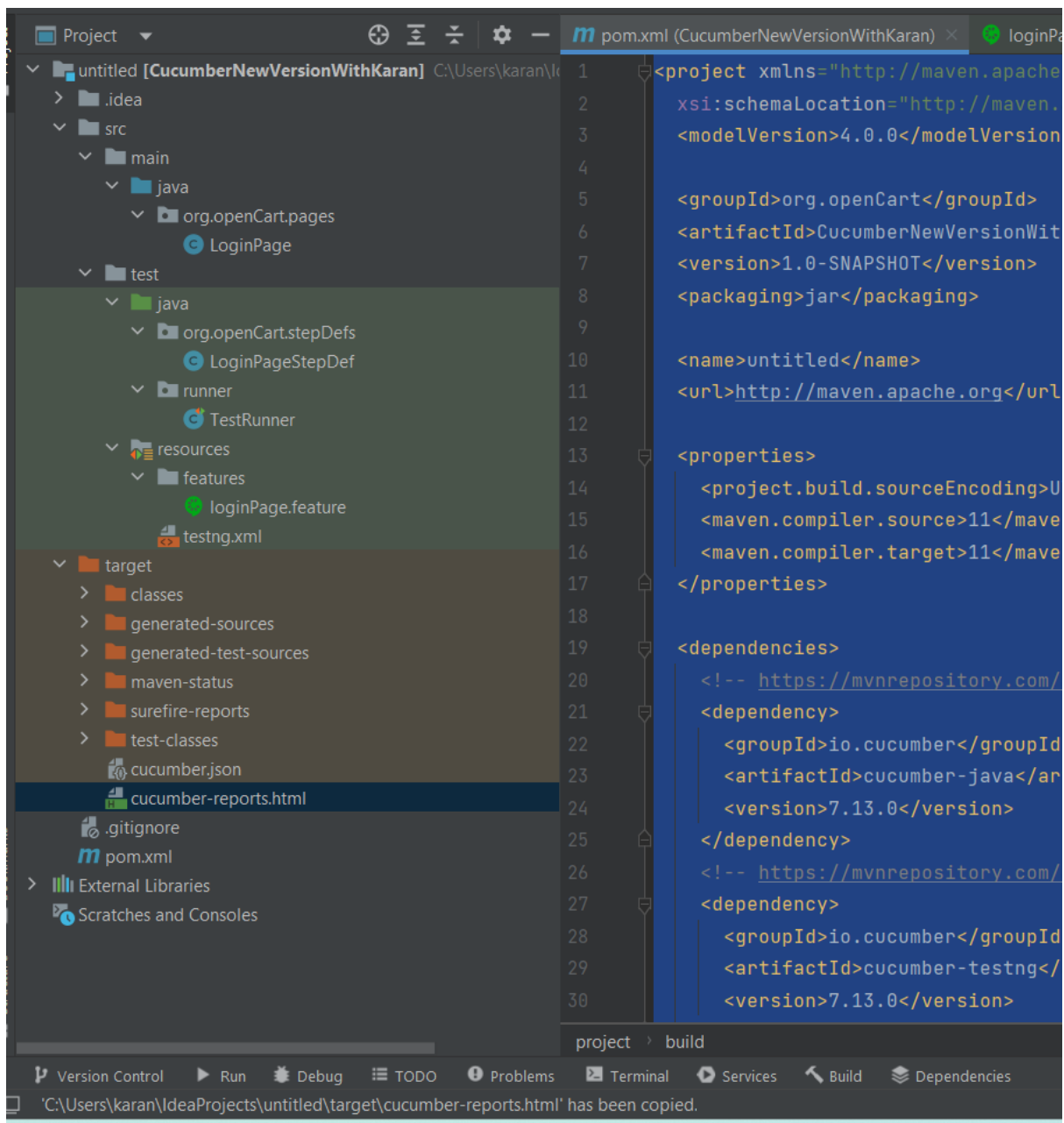
Project structure-

Ocd-

```

<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>org.openCart</groupId>
<artifactId>CucumberNewVersionWithKaran</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>
<name>untitled</name>
<url>http://maven.apache.org</url>
<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<maven.compiler.source>11</maven.compiler.source>
<maven.compiler.target>11</maven.compiler.target>
</properties>
<dependencies>
<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
<dependency>
<groupId>io.cucumber</groupId>
<artifactId>cucumber-java</artifactId>
<version>7.13.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-testing -->
<dependency>
<groupId>io.cucumber</groupId>
<artifactId>cucumber-testing</artifactId>
<version>7.13.0</version>
</dependency>
</dependencies>

```



Codes from this chapter –

Already pasted up.