CUCUMBER:

- It's a Behavior Driven Development test tool (BDD).
- Cucumber is a software tool used by computer programmers used for testing the developed software's.
- It runs automated acceptance test written in BDD.

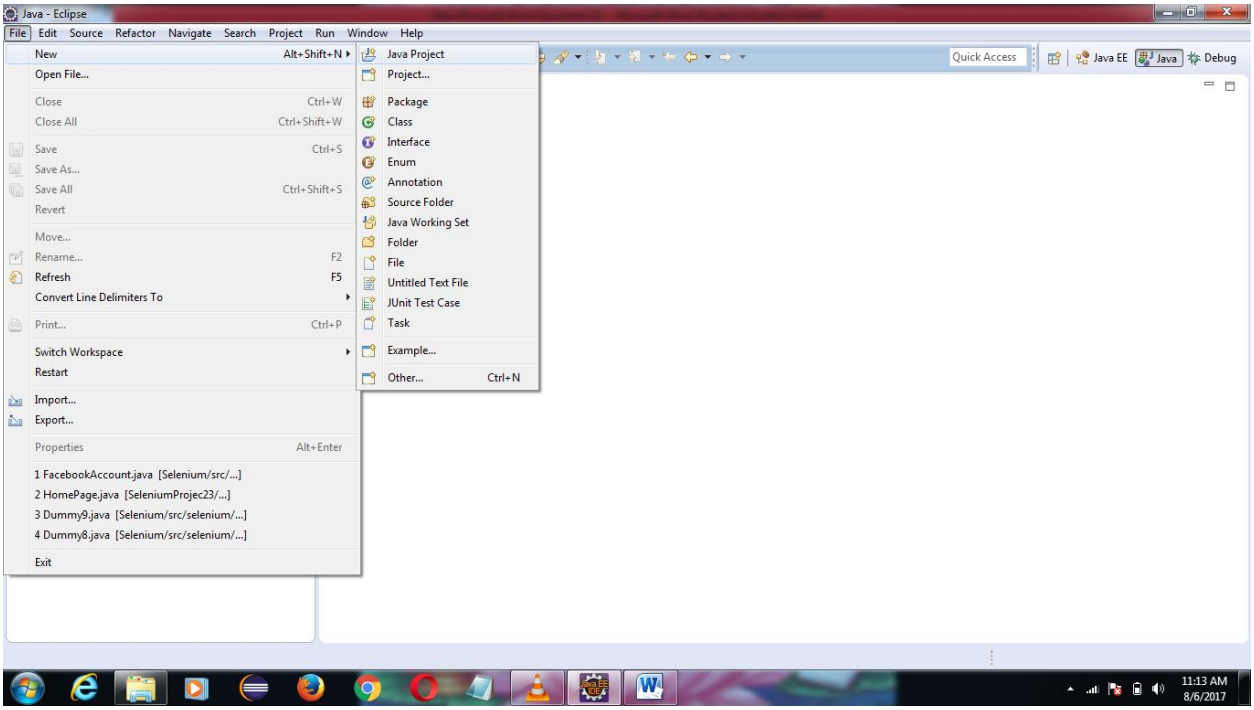
Things needed for Cucumber:

1. Feature file → It consists of scenarios.
2. Step definition file→ It consists of coding part (java +selenium).
3. Test runner class→ It consists of Feature files.

Simple program using cucumber:

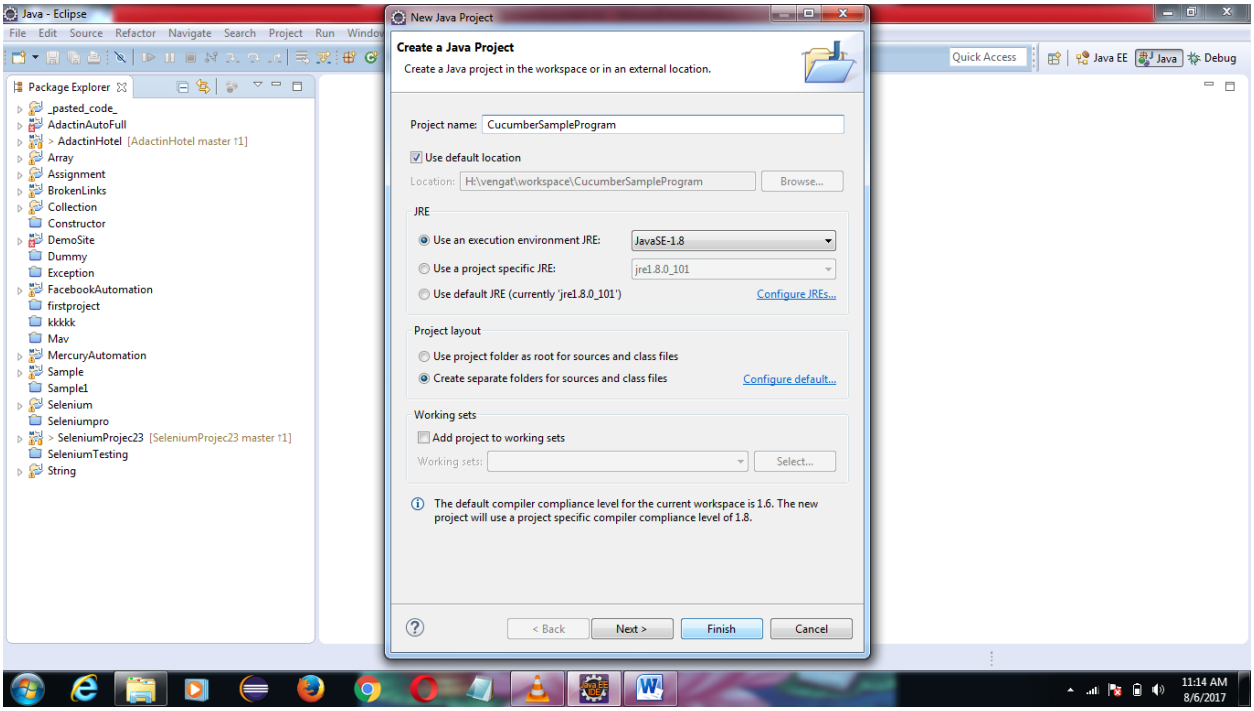
Step 1:

Create new java project.



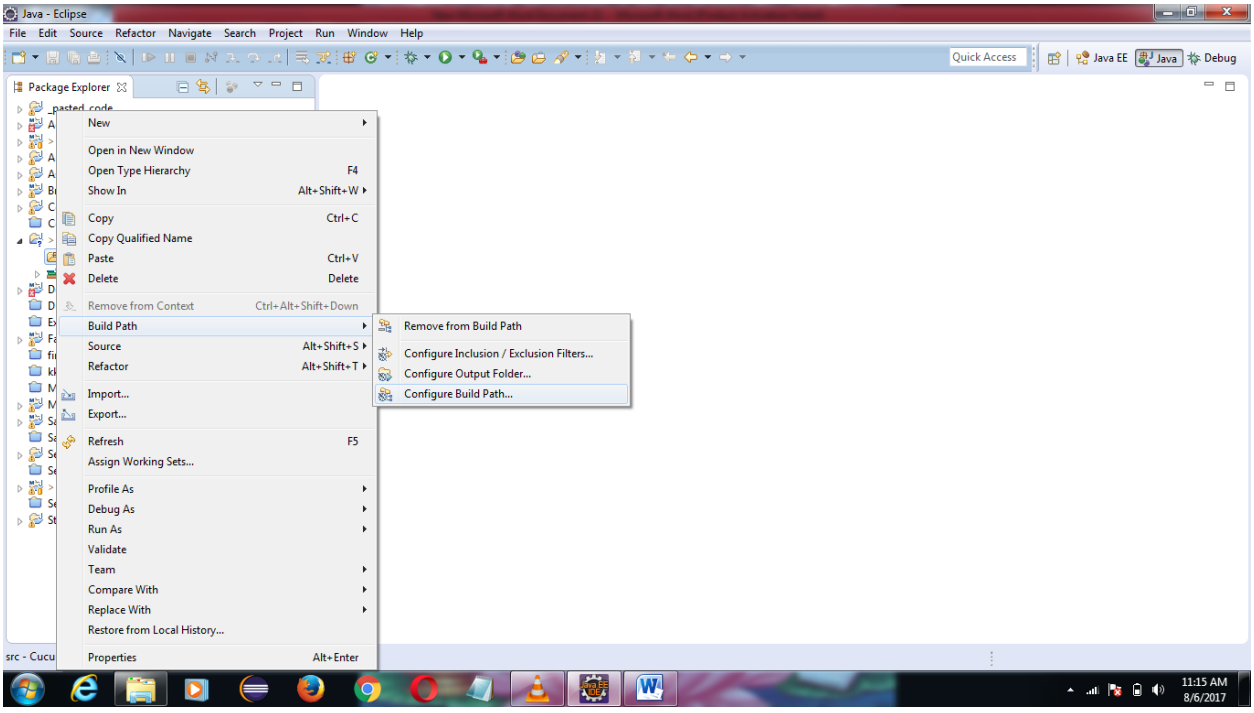
Step 2:

Enter the project name and then click finish.



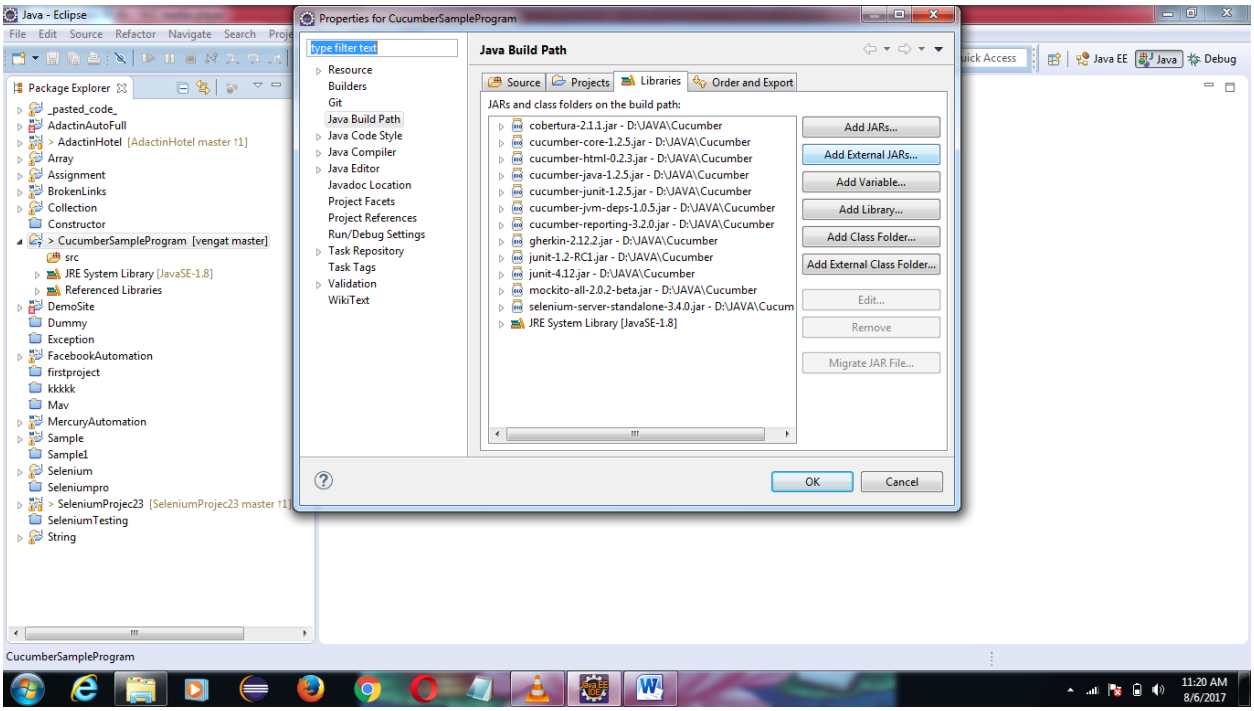
Step 3:

We need to configure jar files.

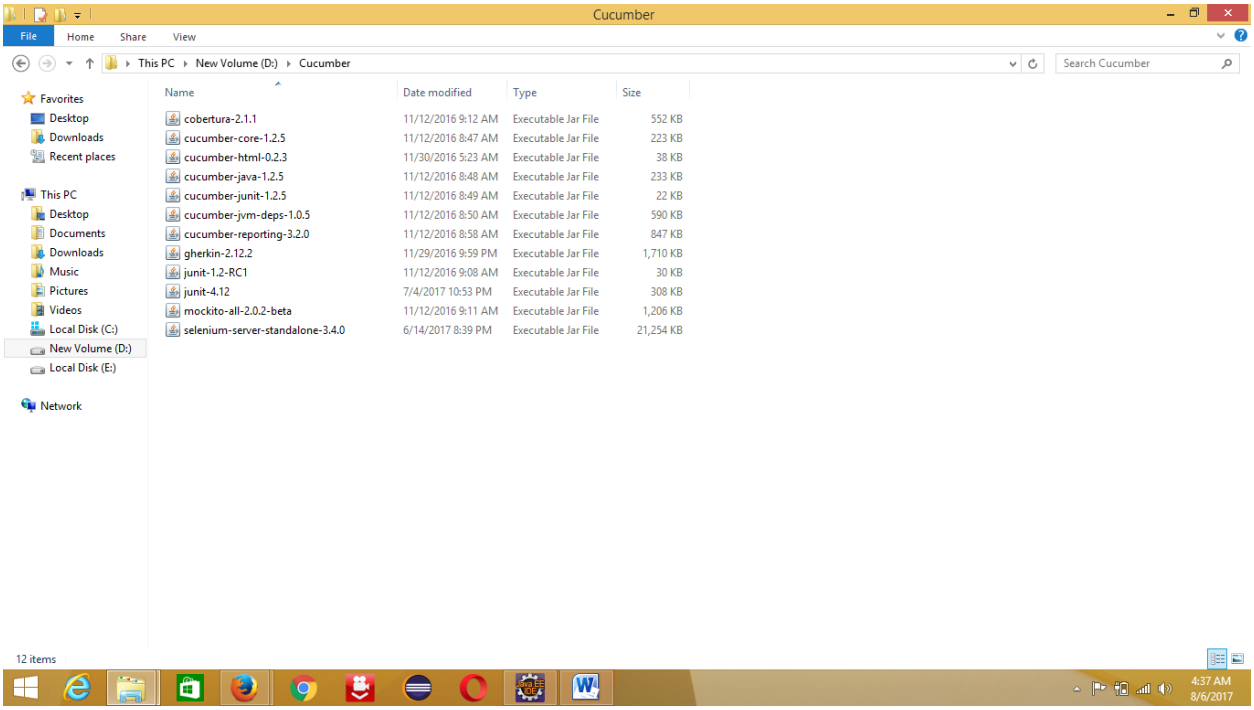


Step 4:

We need to configure our jar files by click Add External jars and then choose our location of the jar, then select the jars we want to add and click ok.

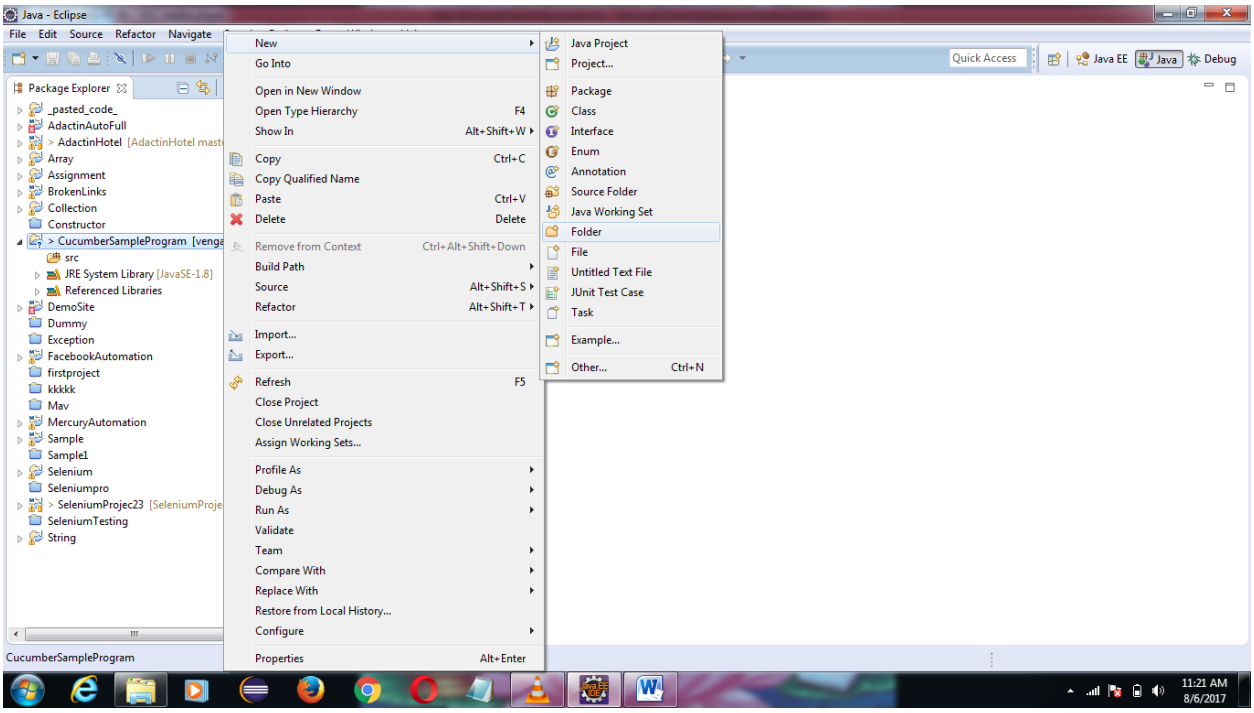


Download these jars:

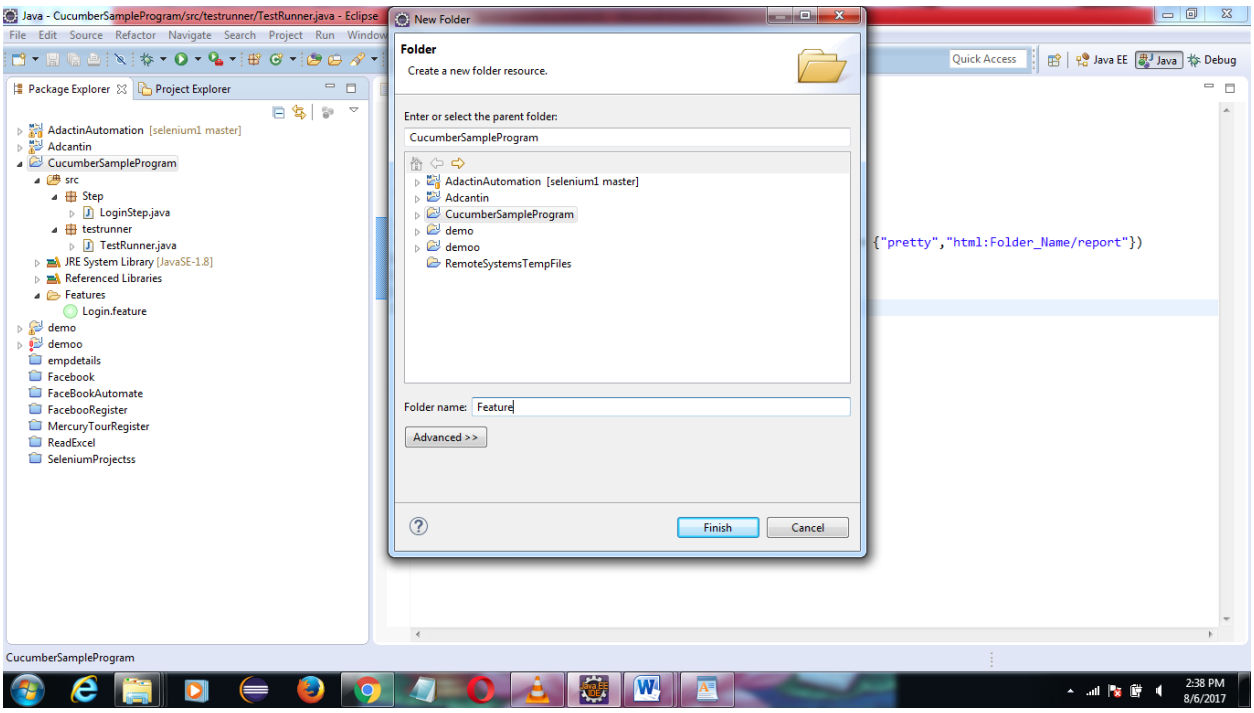


Step 5:

Create a new folder

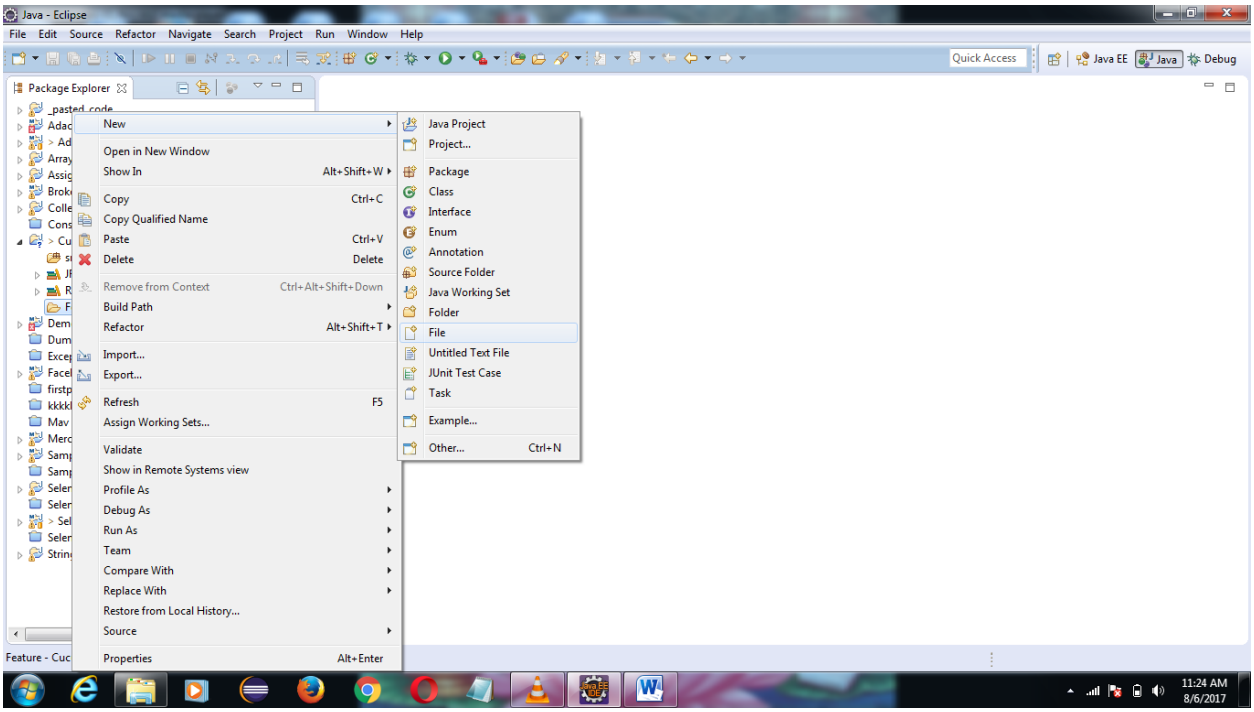


And enter the name as Feature and then click finish.

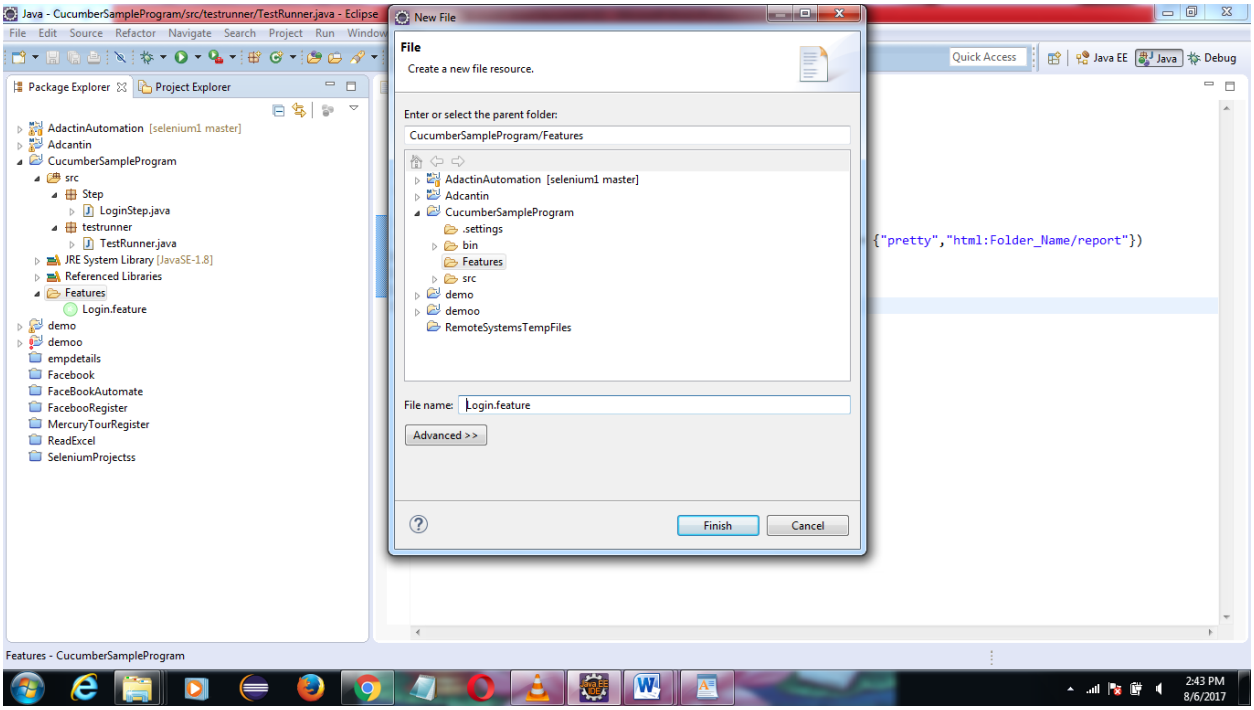


Step 6:

On the Feature folder we need to add File.

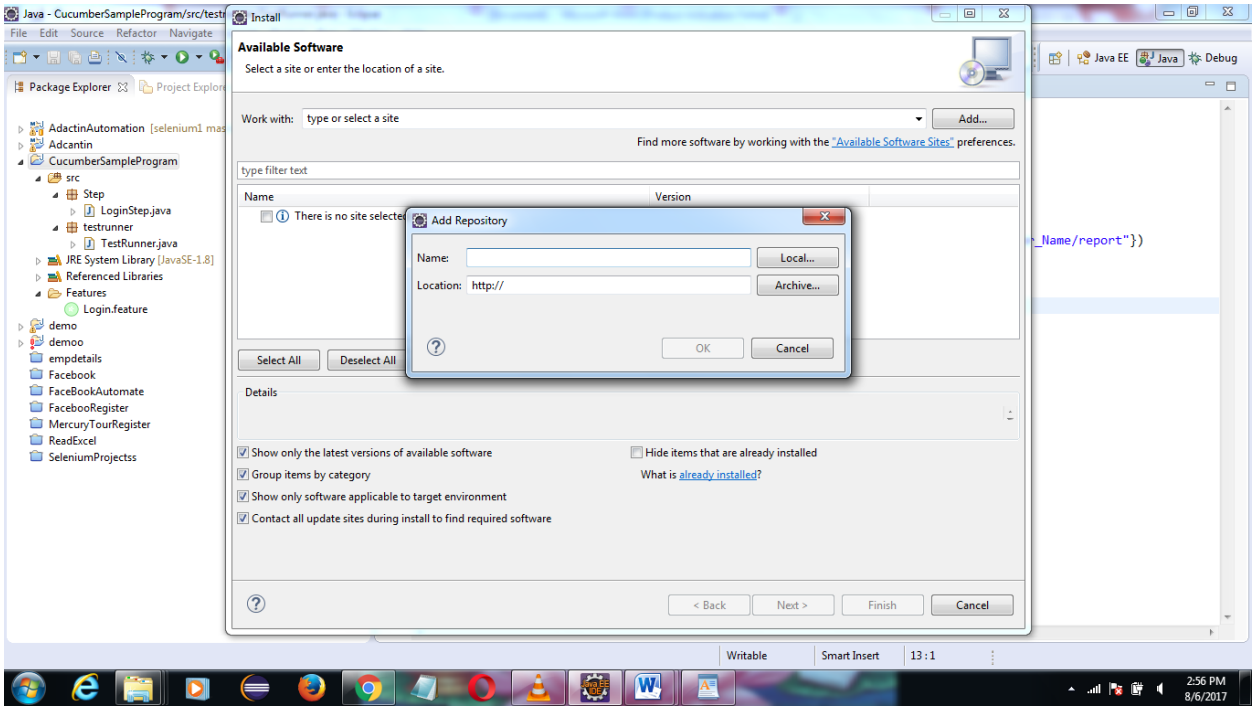


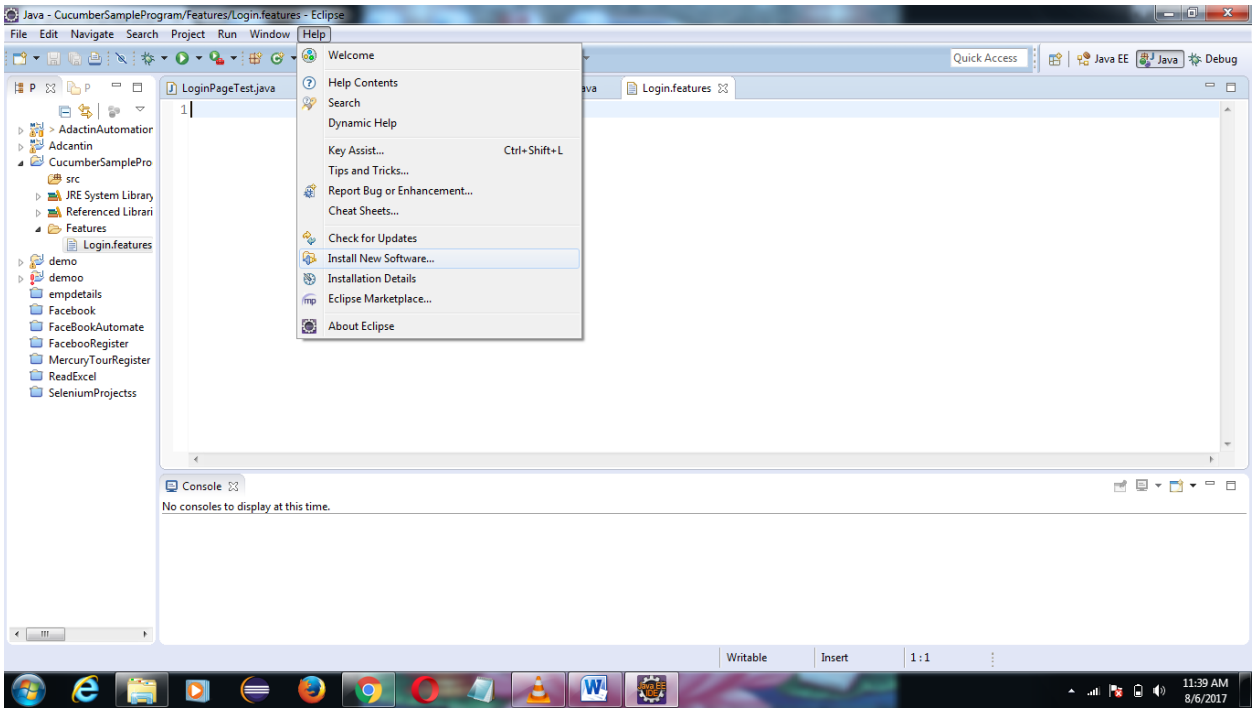
And the enter the file name it must ends with ”.feature” and Click Finish.



Step 7: Configure in two ways:

1. Now , we need configure Gherkin plugin in eclipse through, Go to Market place →Install New Software



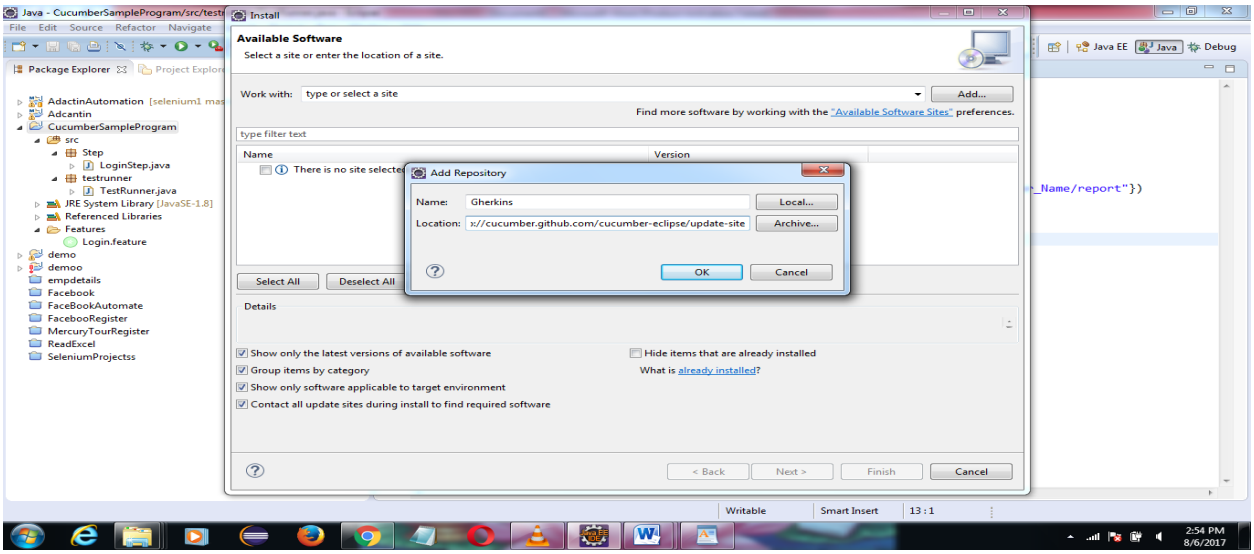


On that window click→ add .

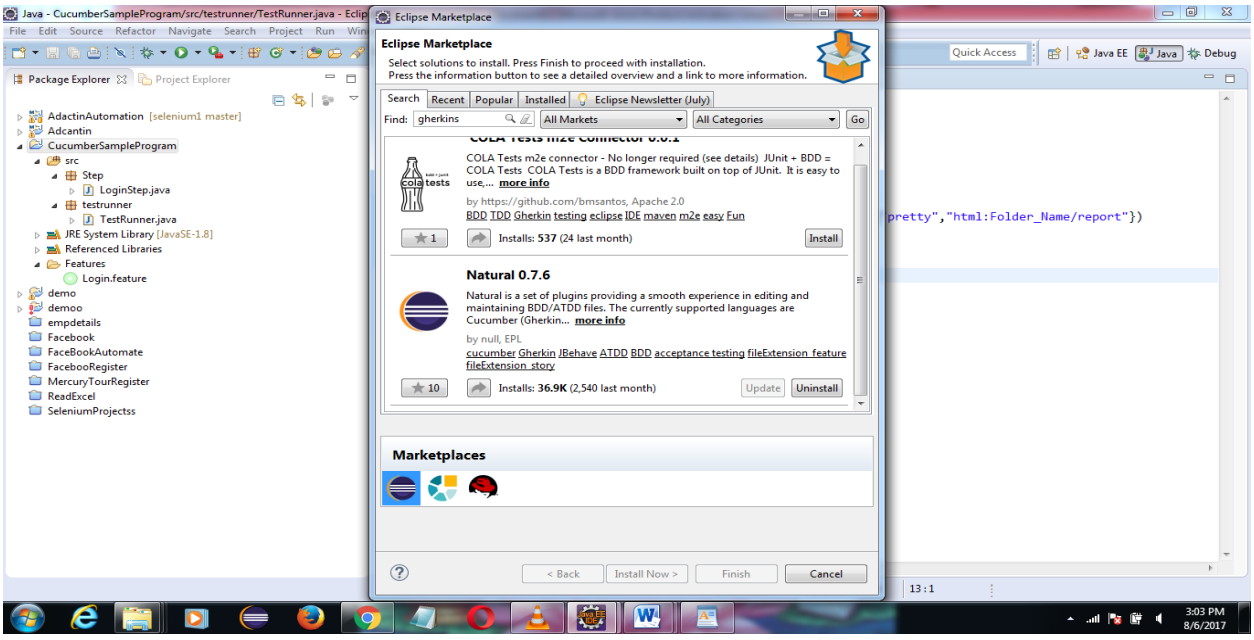
In that name: fill any name.

Address: use this address ” *http://cucumber.github.com/cucumber-eclipse/update-site*”

And click ok →Accept user agreement →Download plugins



2. In help → Eclipse market place → search Gherkins →install **Natural 0.7.6.**



By this way you download Gherkins plugin for cucumber.

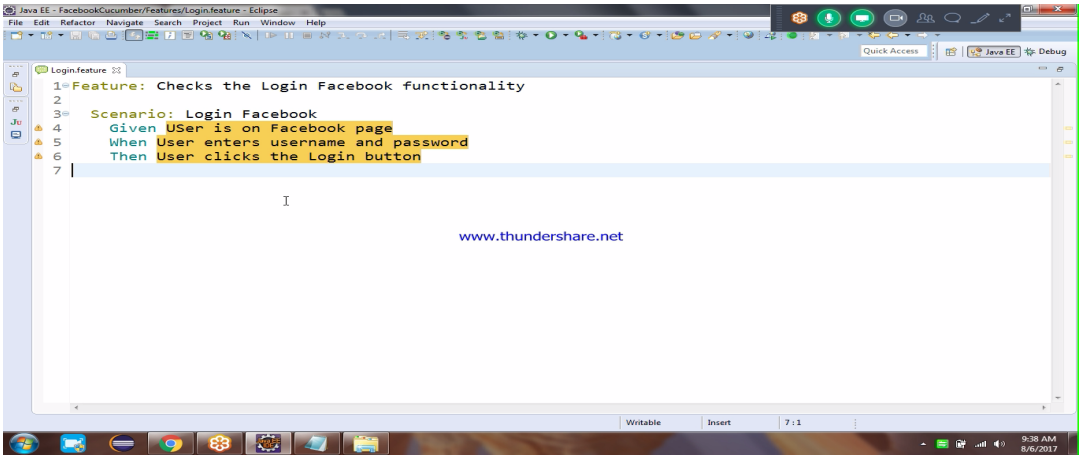
Step 8:

In our project ,we need to create three packages under src.

1. Testrunner → In this we add our test cases .
2. Steps → in this we create our methods .
3. pages → Here we add our Locators of web element and provide getters and setters.

Add future class on the file created on the Feature folder.

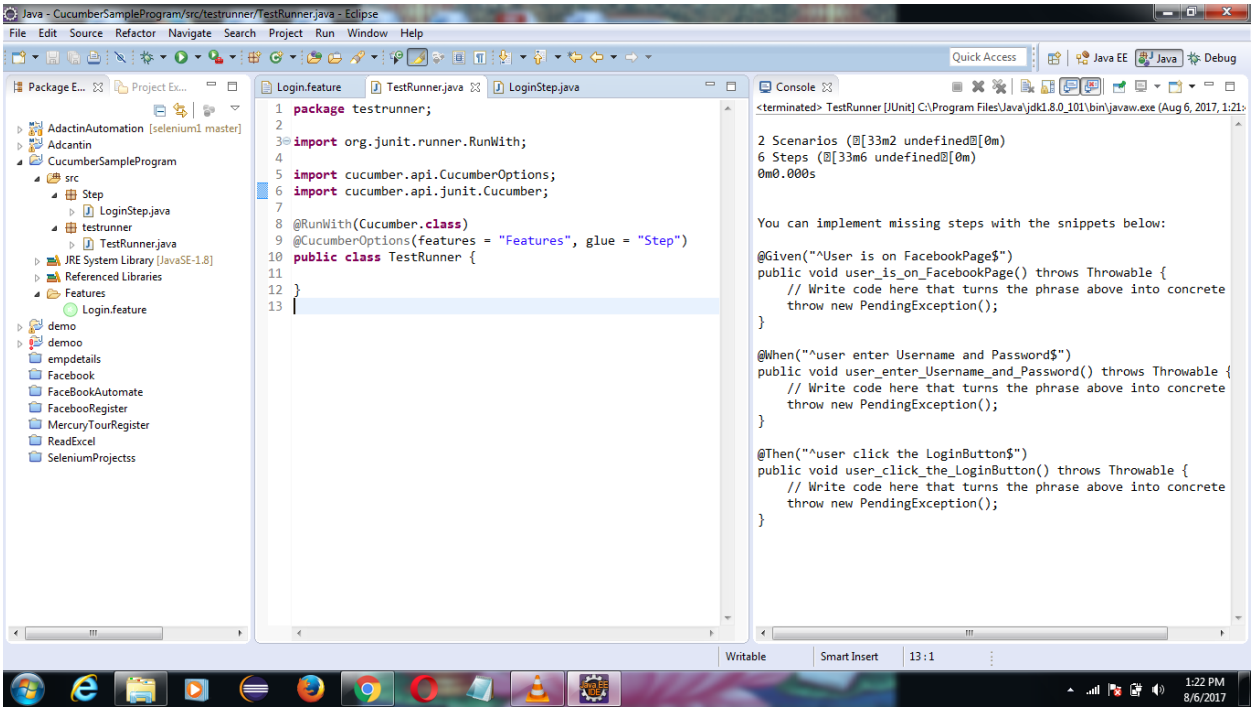
Create feature file like below



<u>Package: testrunner</u>

features:It contains Feature files.
Glue: It consists of steps of test runner class which obtained on console.

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Features", glue = "Step")
public class TestRunner {

}
```



After, we run the test case, we get console output use that steps for step package.

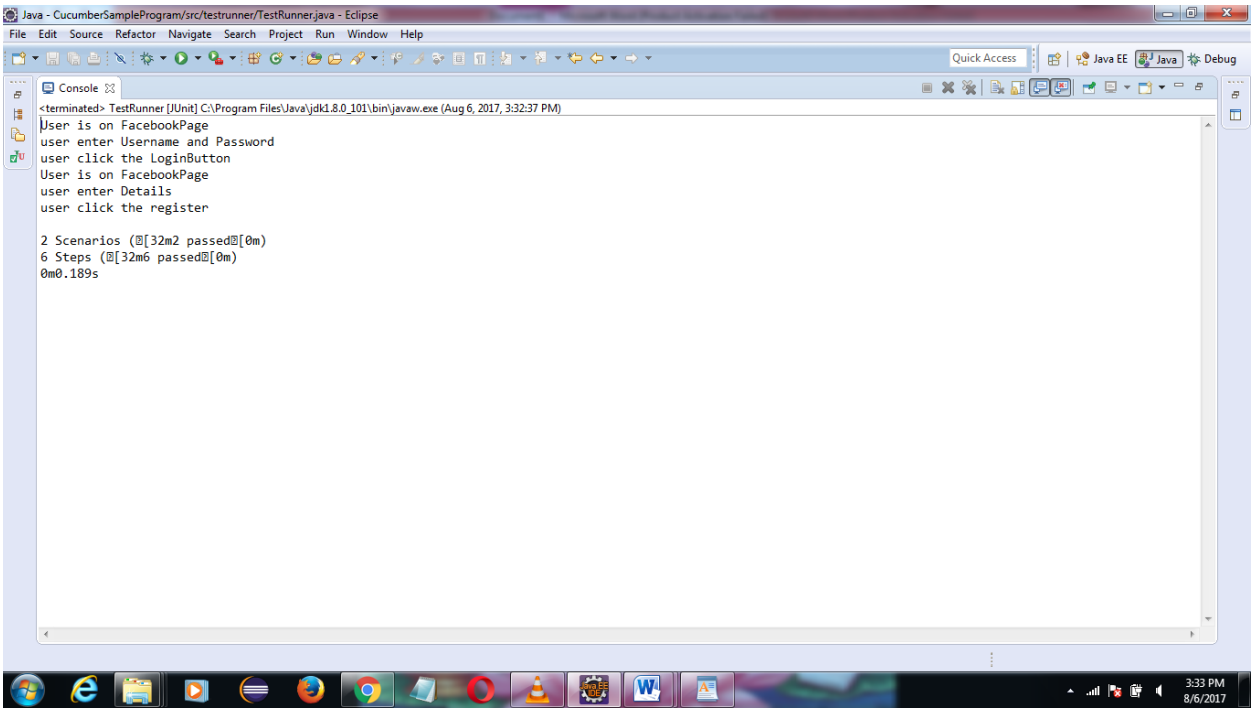
We create LoginSteps under the steps package and use that console output codes

Program of LoginStep class:

```
public class LoginStep {

@Given("^User is on FacebookPage$")
public void user_is_on_FacebookPage() throws Throwable {
 System.out.println("User is on FacebookPage");
}

@When("^user enter Username and Password$")
public void user_enter_Username_and_Password()  {
  System.out.println("user enter Username and Password");
}

@Then("^user click the LoginButton$")
public void user_click_the_LoginButton() {
        System.out.println("user click the LoginButton");

}
@When("^user enter Details$")
public void user_enter_Details() {
```

```
        System.out.println("user enter Details");
}

@Then("^user click the register$")
public void user_click_the_register()  {
        System.out.println("user click the register");
   }
}
```

Output:

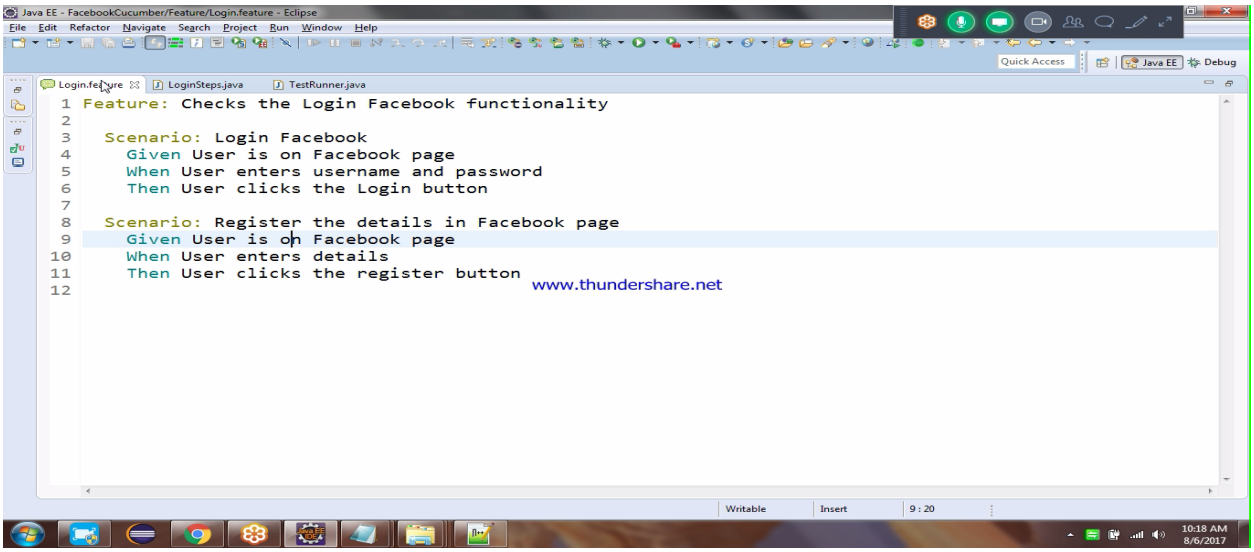


Formats of Report Using cucumber:

Cucumber supports various report formats. They are

1. **HTML REPORT FORMAT:**

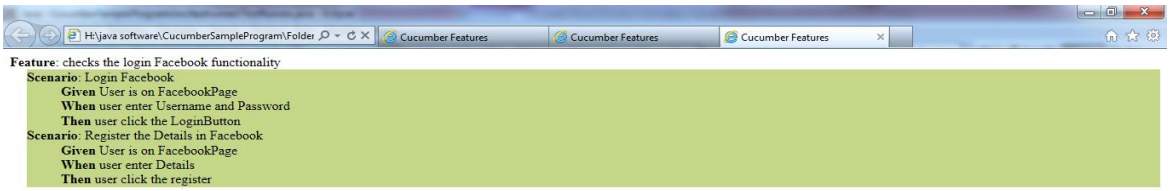
Feature file:



Package: Test runner

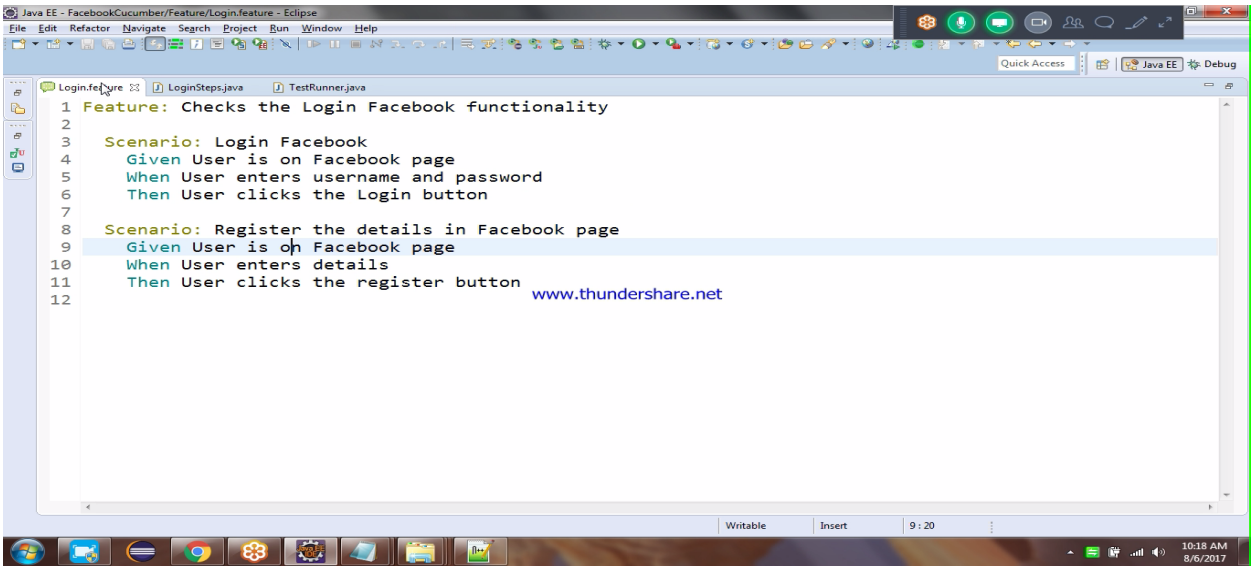
```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Features", glue = "Step", plugin= {"pretty","html:Folder name/report/cucumber"})
public class TestRunner {

}
```

Output:



2. JUNIT REPORT FORMAT:

Feature file:

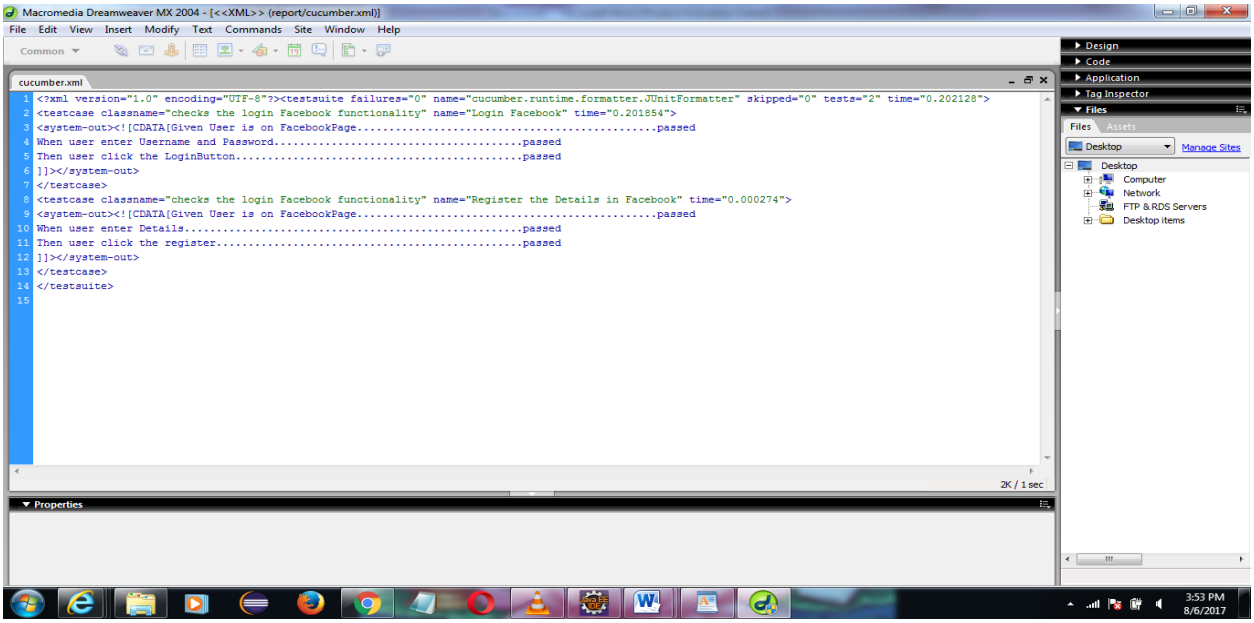


Package: Test runner

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Features", glue = "Step", plugin= {"pretty","junit:Folder name/report/cucumber.xml"})
```

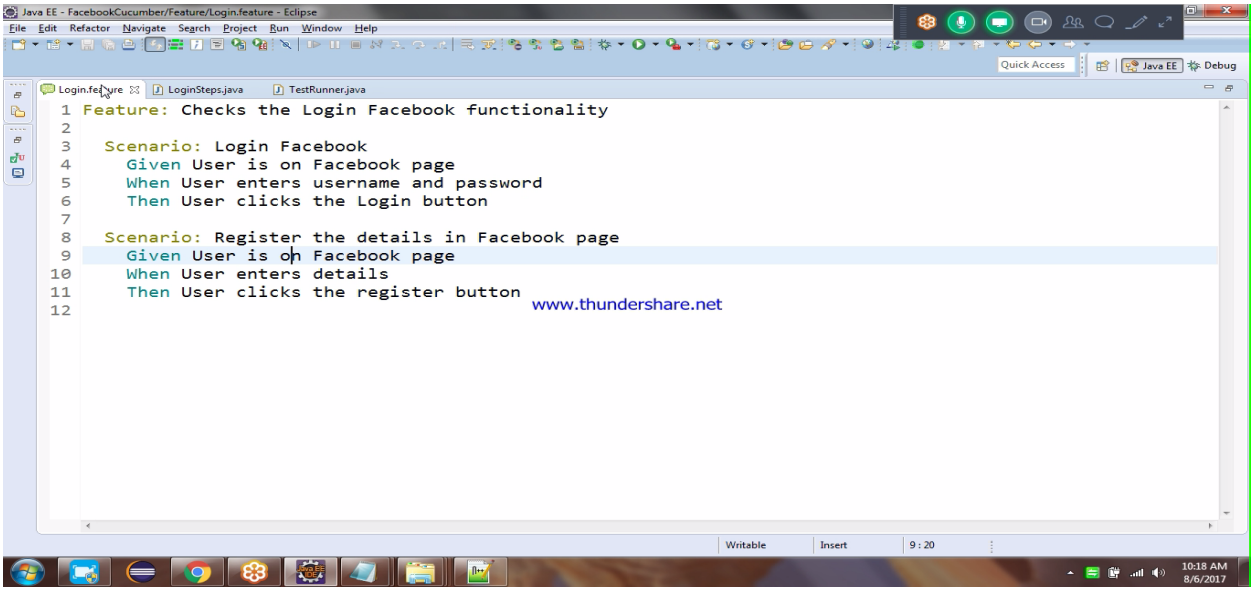
```
public class TestRunner {

}
```

Output:



3. JSON TYPE REPORT:

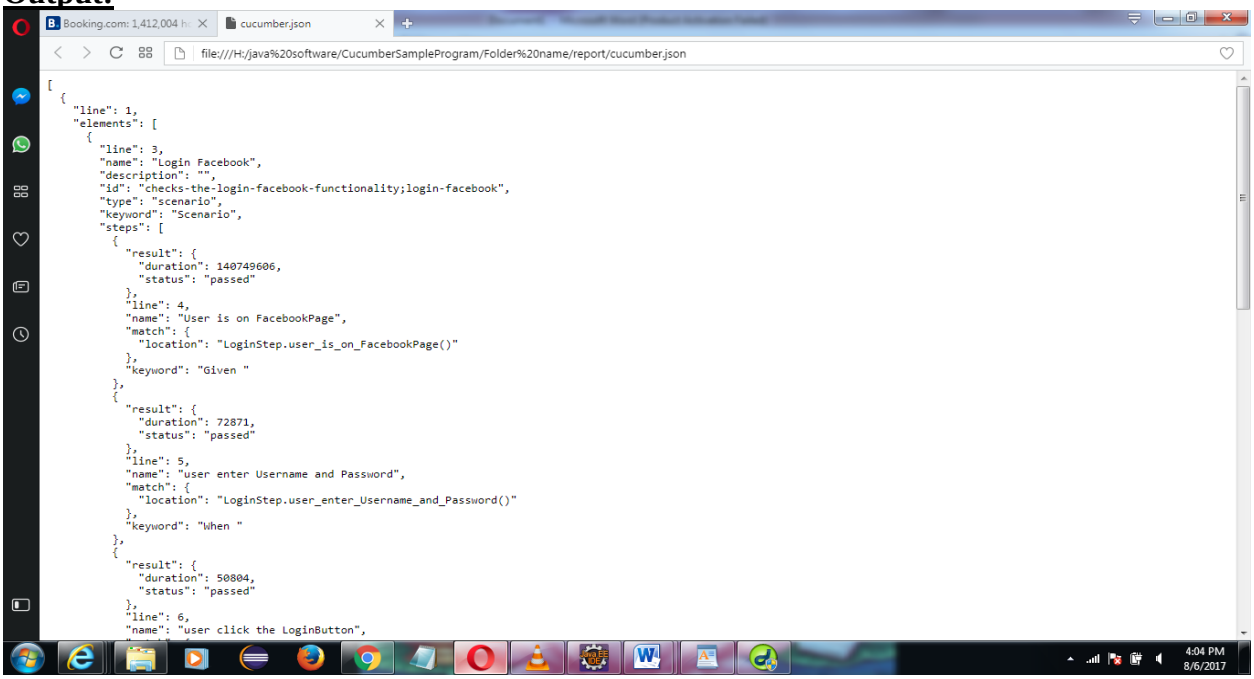
Feature file:



Package: Test runner

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Features", glue = "Step", plugin= {"pretty","json:Folder name/report/cucumber.json"})
public class TestRunner {

}
```

Output:



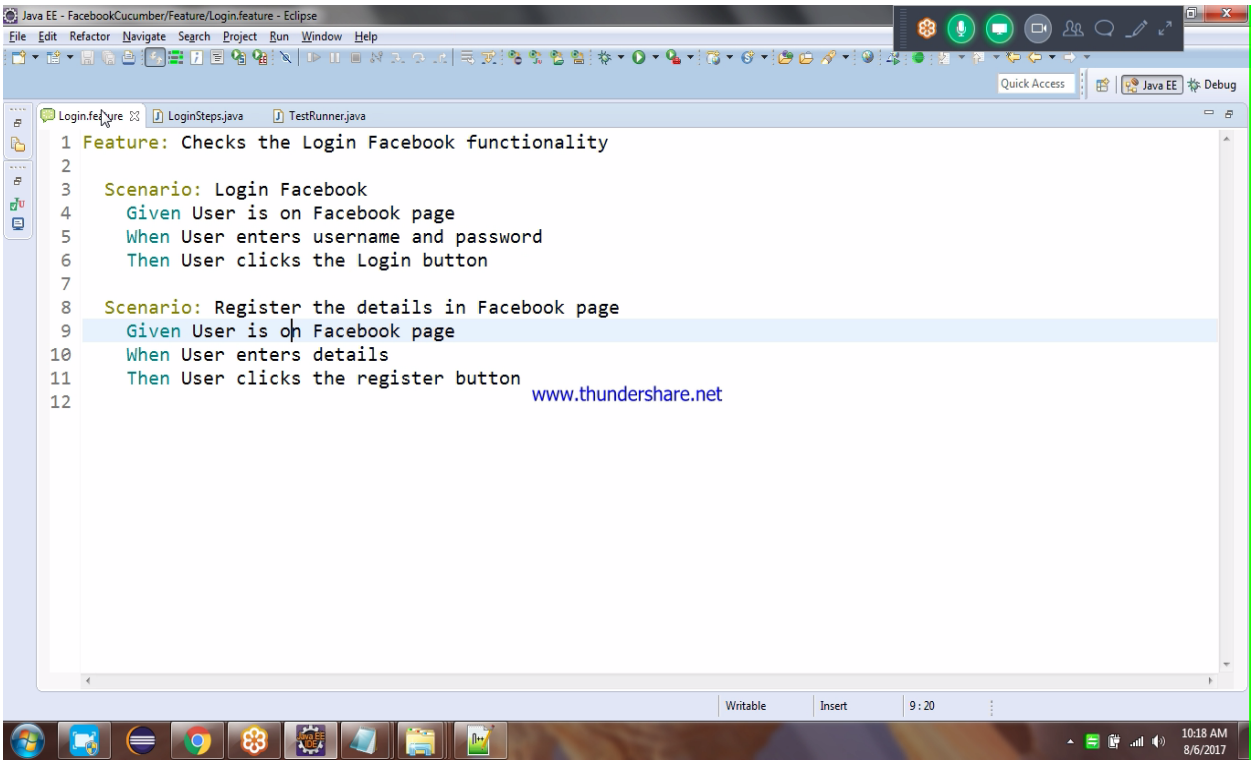
DRY RUN:

- This is used for check our formats on the Featured file.

  dryRun=true -→Validate our Featured file.

  dryRun=false → Not Validate our featured file

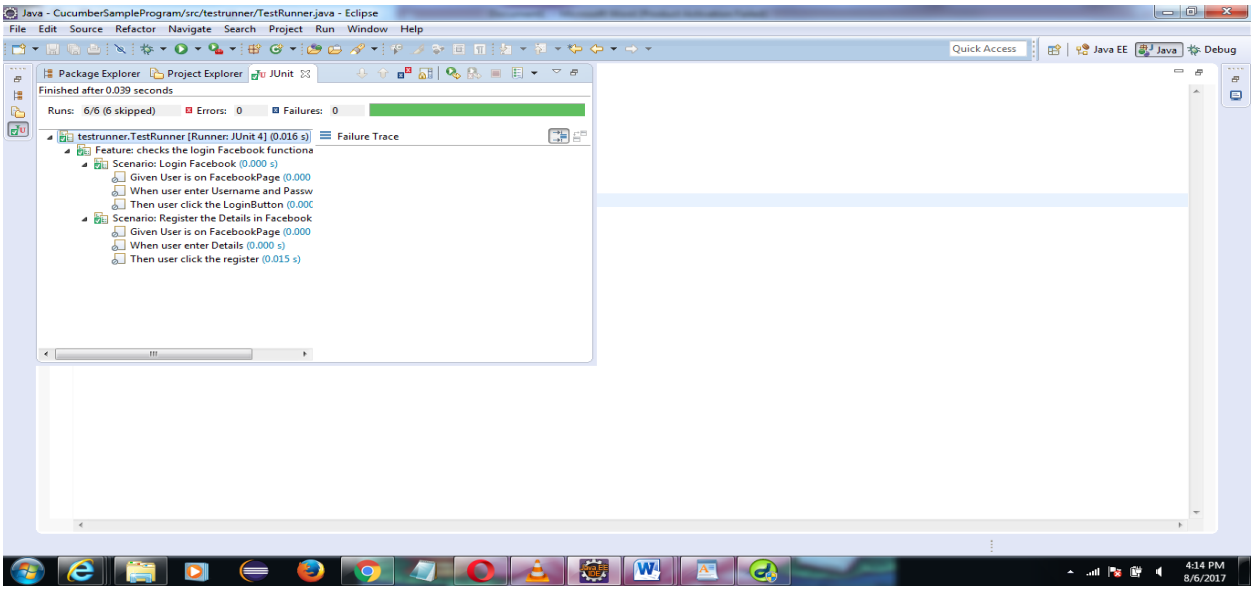
Feature file:



Package: Test runner

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Features", glue = "Step", dryRun=True)
public class TestRunner {

}
```

Output:



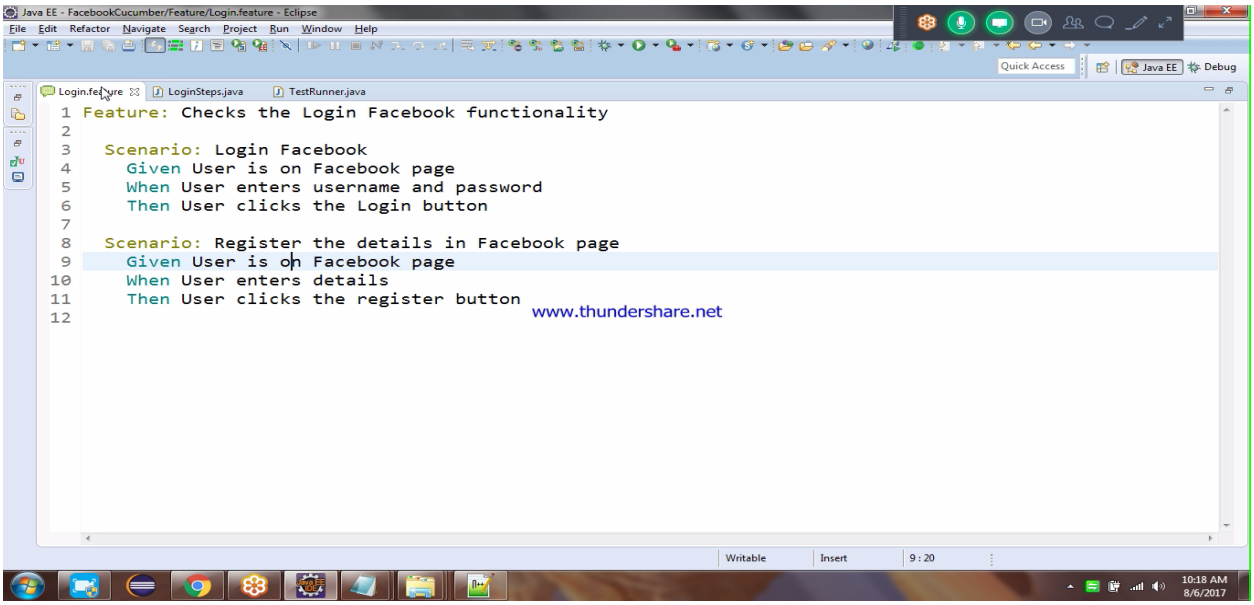
MONOCHROME:

- It gives report on console.

  monochrome = true → It a report.

  monochrome = false → It not display a report on console.

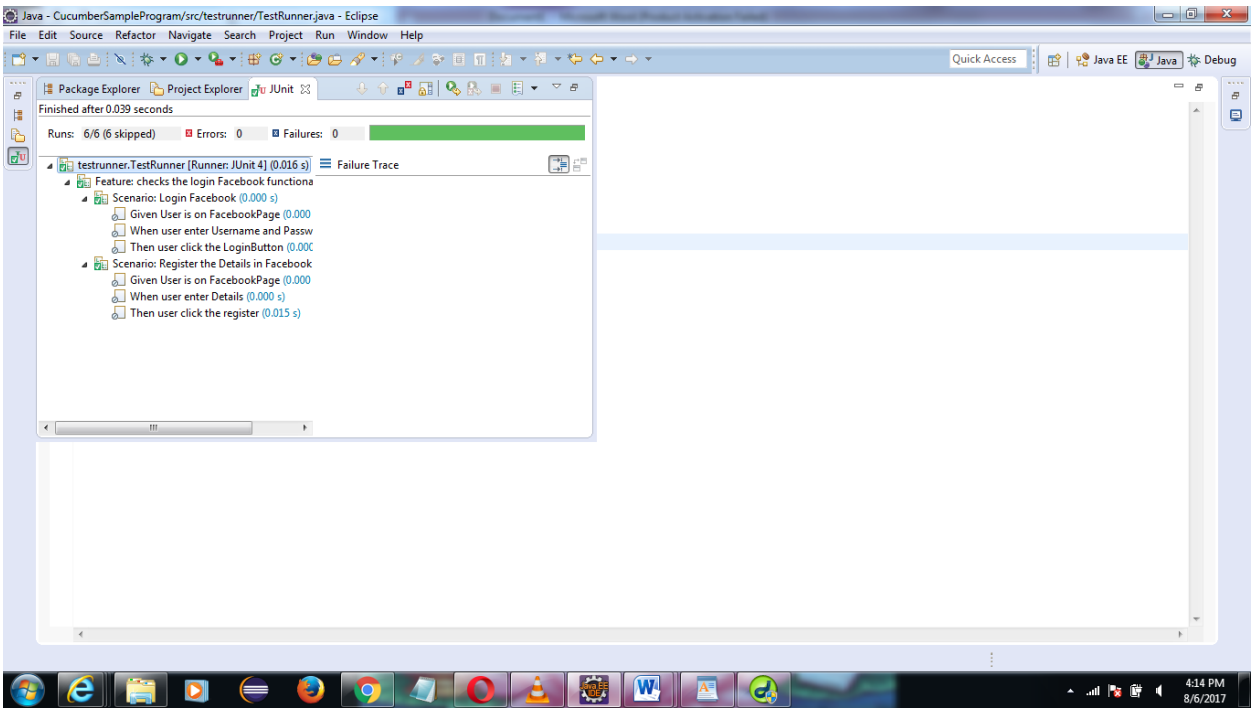
Feature file:



Package: Test runner

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Features", glue = "Step", dryRun=True)
public class TestRunner {

}
```

Output:

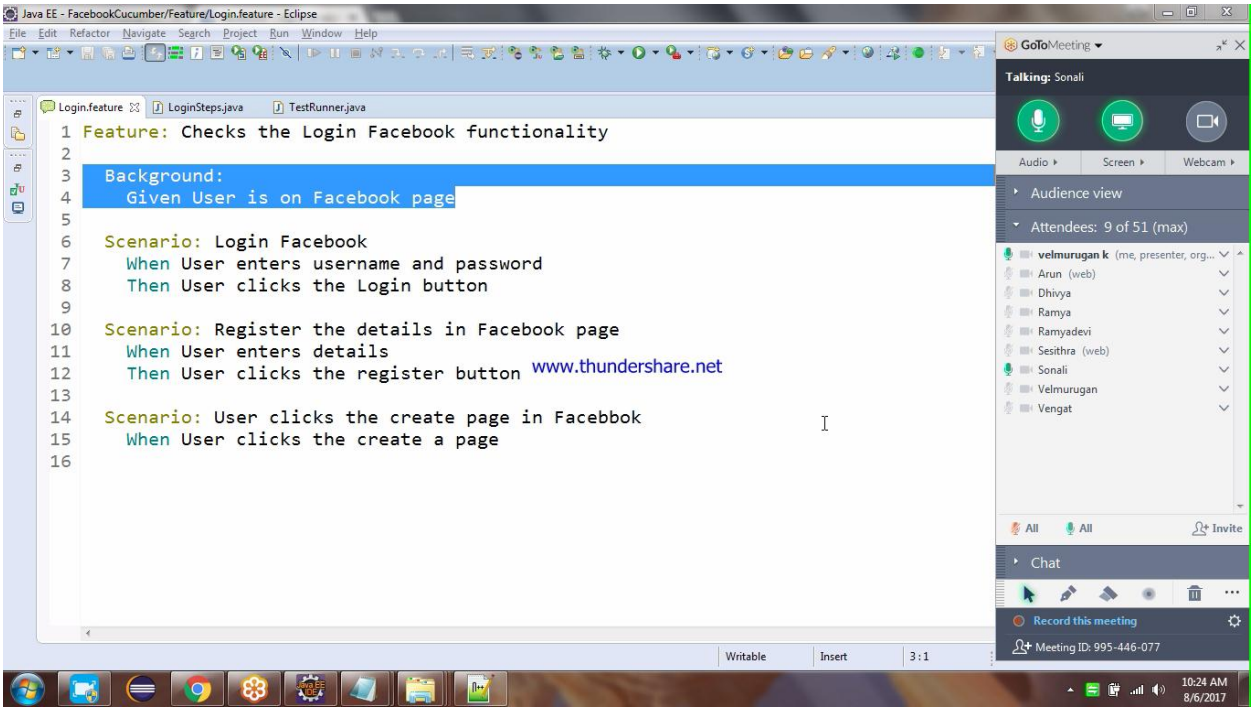


BACKGROUND:

- It's a keyword used in featured class for declare the common step as once.

Output:

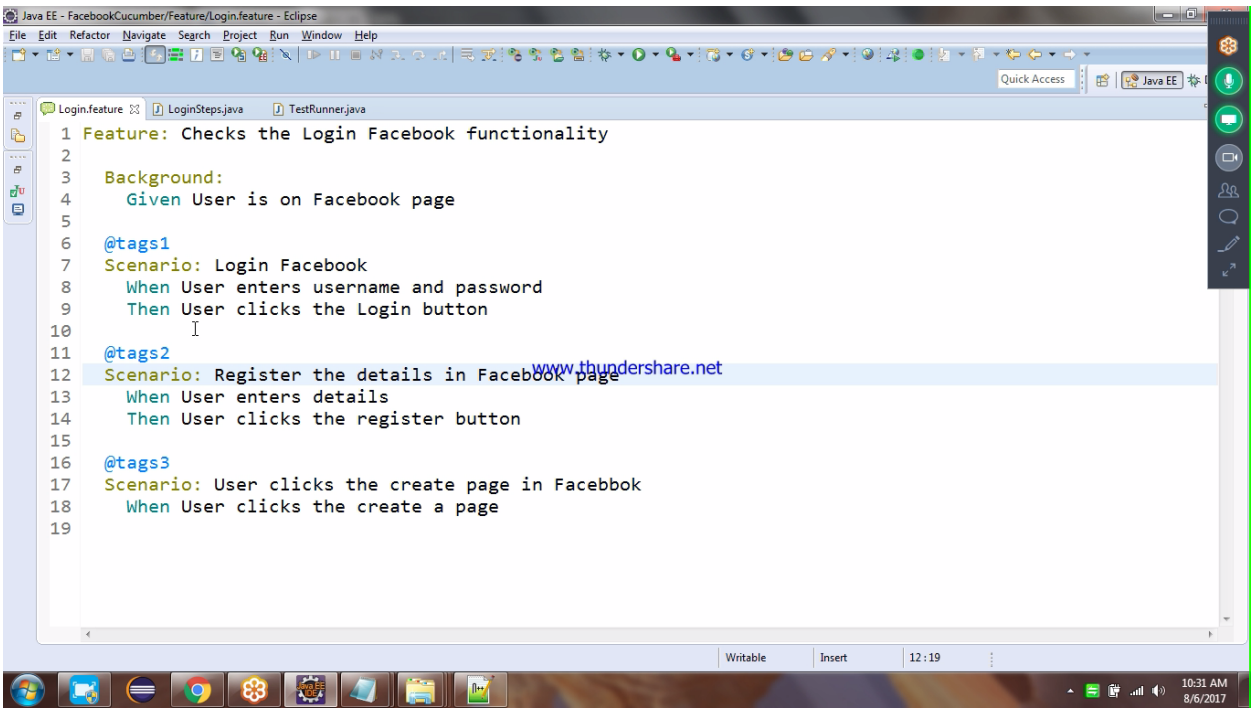
Here "Given" is common for all scenarios …so we declare under the background keyword.



TAGS:

- It's a command which is used for execute a particular scenario.

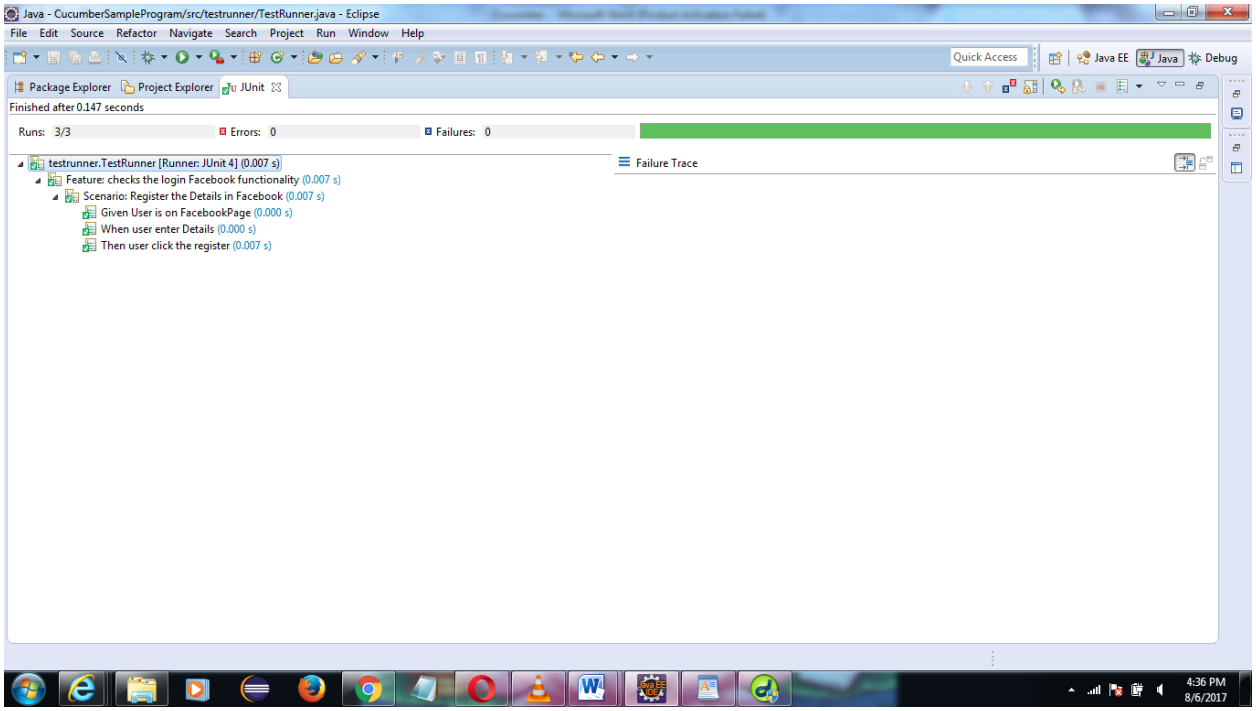
Feature file:



Package: Test runner

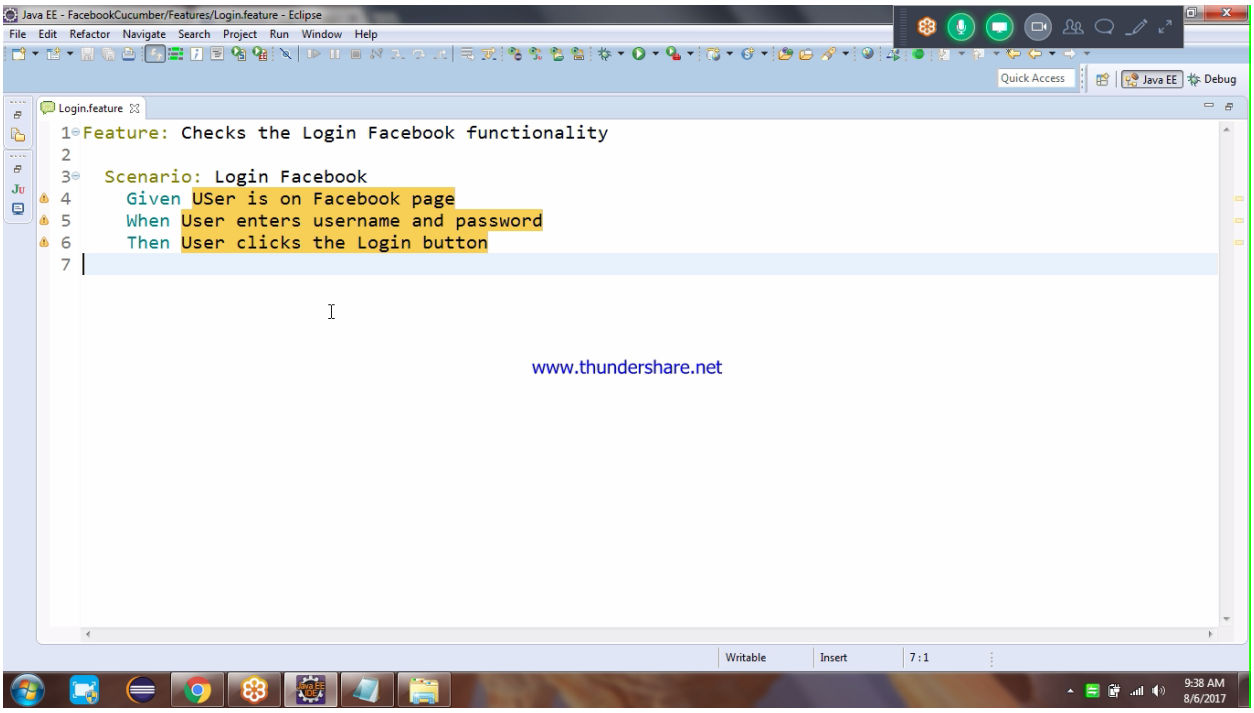
```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Features", glue = "Step", dryRun=True,
tags={"@tags1"})
public class TestRunner {

}
```

Output:



FB LOGIN USING CUCUMBER:

Feature file:



Package: testrunner

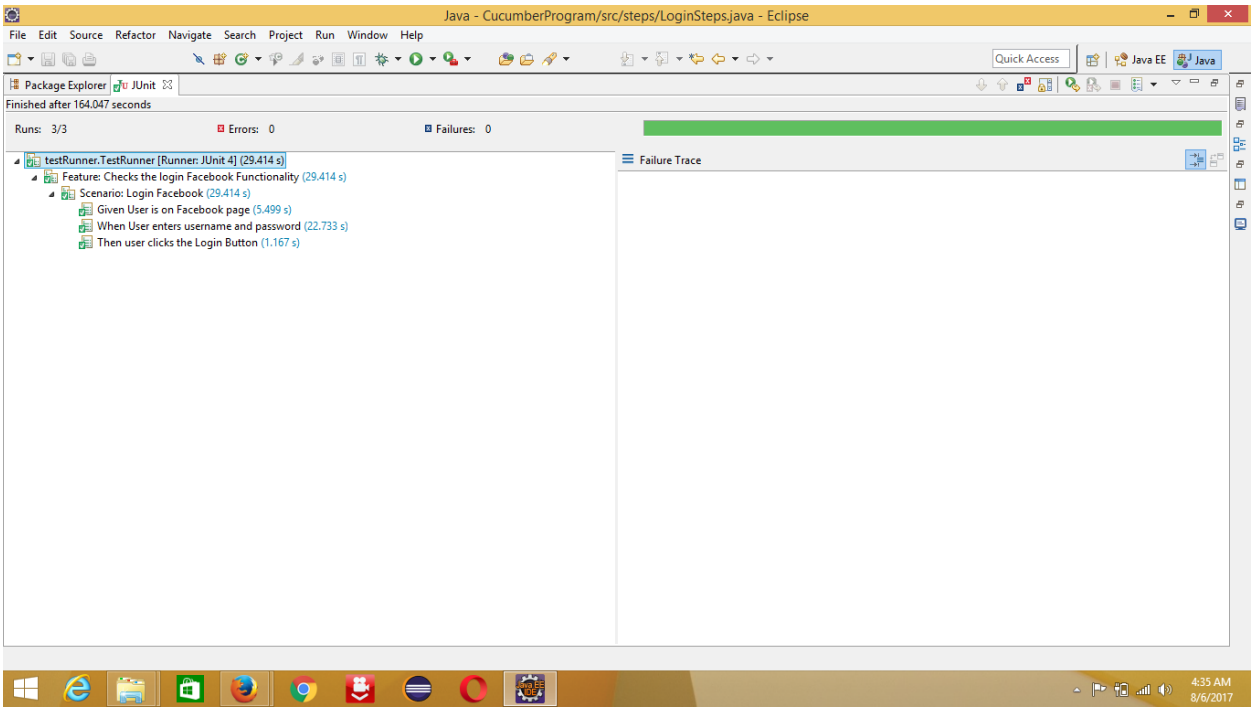
```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Feature", glue = "steps")
public class TestRunner {

}
```

Package: steps

```
public class LoginSteps {
	WebDriver driver;
	@Given("^User is on Facebook page$")
	public void user_is_on_Facebook_page()  {
		System.setProperty("webdriver.gecko.driver", "C:/Users/mohan
pc/Desktop/CucumberProgram/driver/geckodriver.exe");
		driver = new FirefoxDriver();
```

```
        driver.get("https://www.facebook.com/");
    }

    @When("^User enters username and password$")
    public void user_enters_username_and_password()  {
        driver.findElement(By.id("email")).sendKeys("mohanrajeee6@gmail.com");;
        driver.findElement(By.id("pass")).sendKeys("mohanrajeee6@gmail.com");;

    }

    @Then("^user clicks the Login Button$")
    public void user_clicks_the_Login_Button()  {
        driver.findElement(By.id("u_0_r")).click();
    }


}
```

Output:



ENTERING VALUE IN FEATURE FILE:

Feature file:

```
Feature: check the login adactin functionality
Scenario: Login Adactin
        Given User is an adactin page
        When User enter "vengat16" and "Karthick" and click login button
        Then Message displayed Login Successfully
```

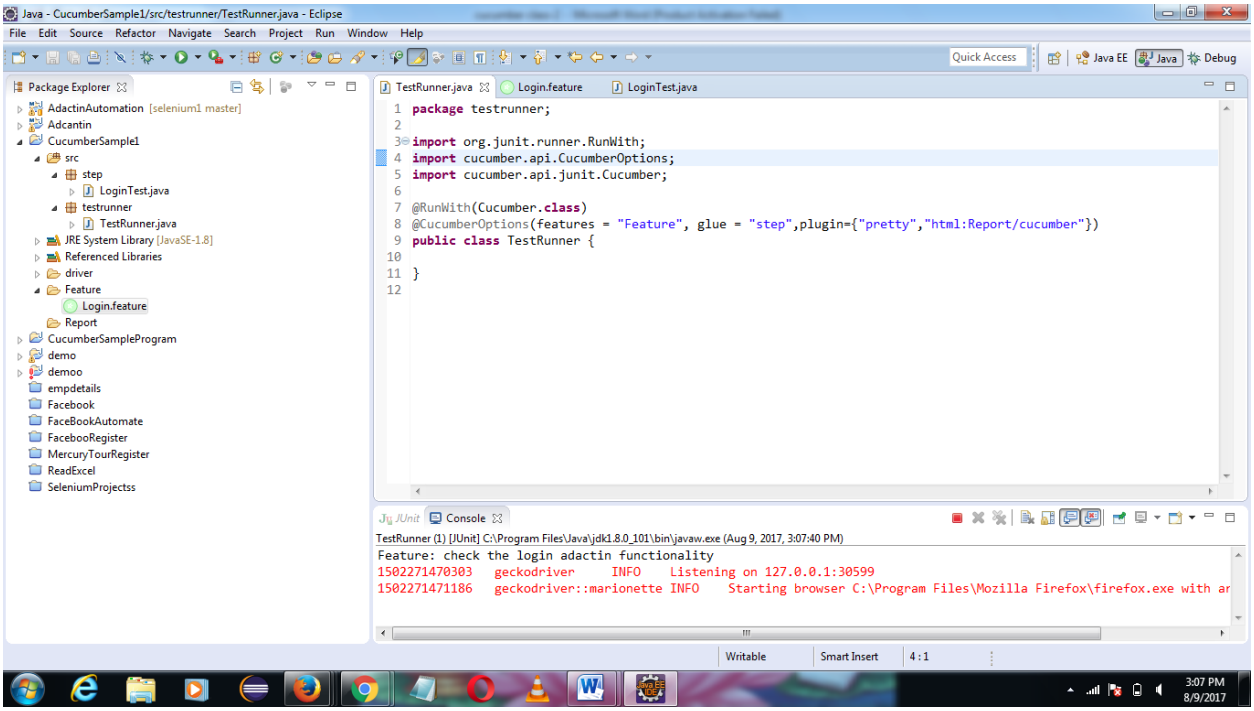
Step definition:

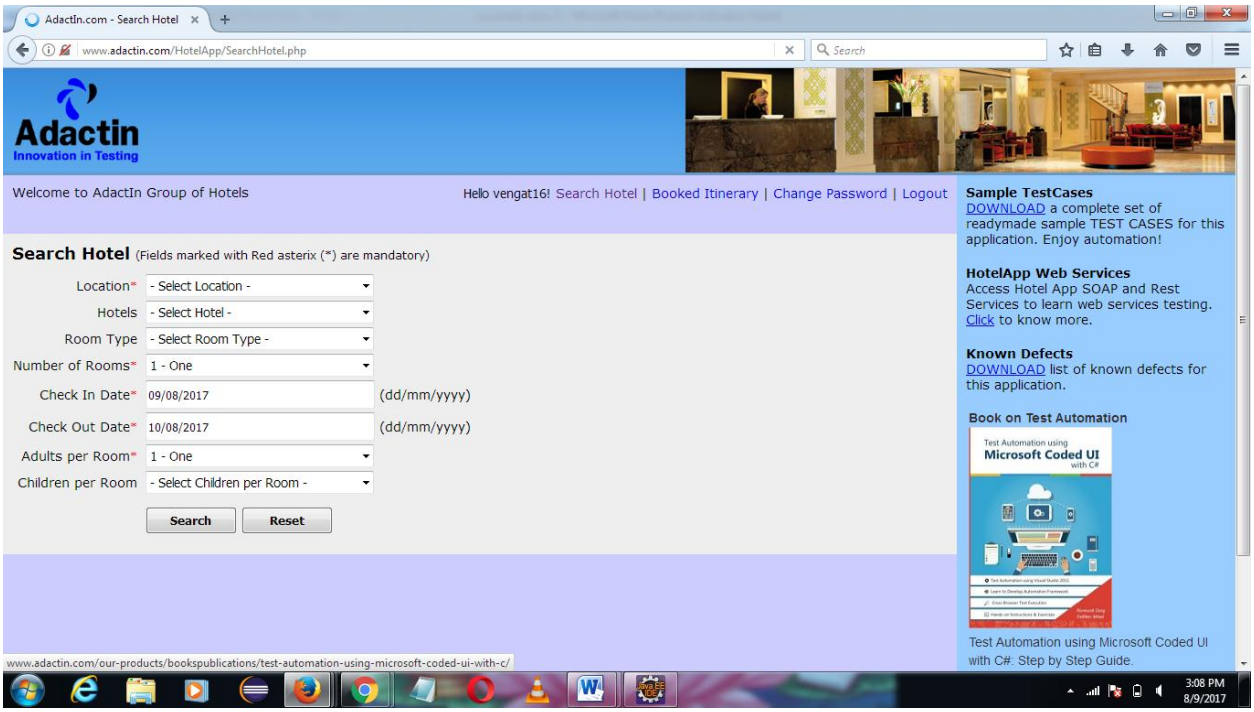
```
public class LoginTest {
        WebDriver driver;

        @Given("^User is an adactin page$")
        public void user_is_an_adactin_page() {
                System.setProperty("webdriver.gecko.driver",
                                "H:\\java software\\CucumberSample1\\driver\\geckodriver.exe");
                driver = new FirefoxDriver();
                driver.get("http://www.adactin.com/HotelApp/");
        }

        @When("^User enter \"([^\"]*)\" and \"([^\"]*)\" and click login button$")
        public void user_enter_UserName_and_Password(String Username,
                        String Password) {
                driver.findElement(By.id("username")).sendKeys(Username);
                ;
                driver.findElement(By.id("password")).sendKeys(Password);
                ;
                driver.findElement(By.id("login")).click();

        }

        @Then("^Message displayed Login Successfully$")
        public void message_displayed_Login_Successfully() {
        driver.quit();
        }

}
```

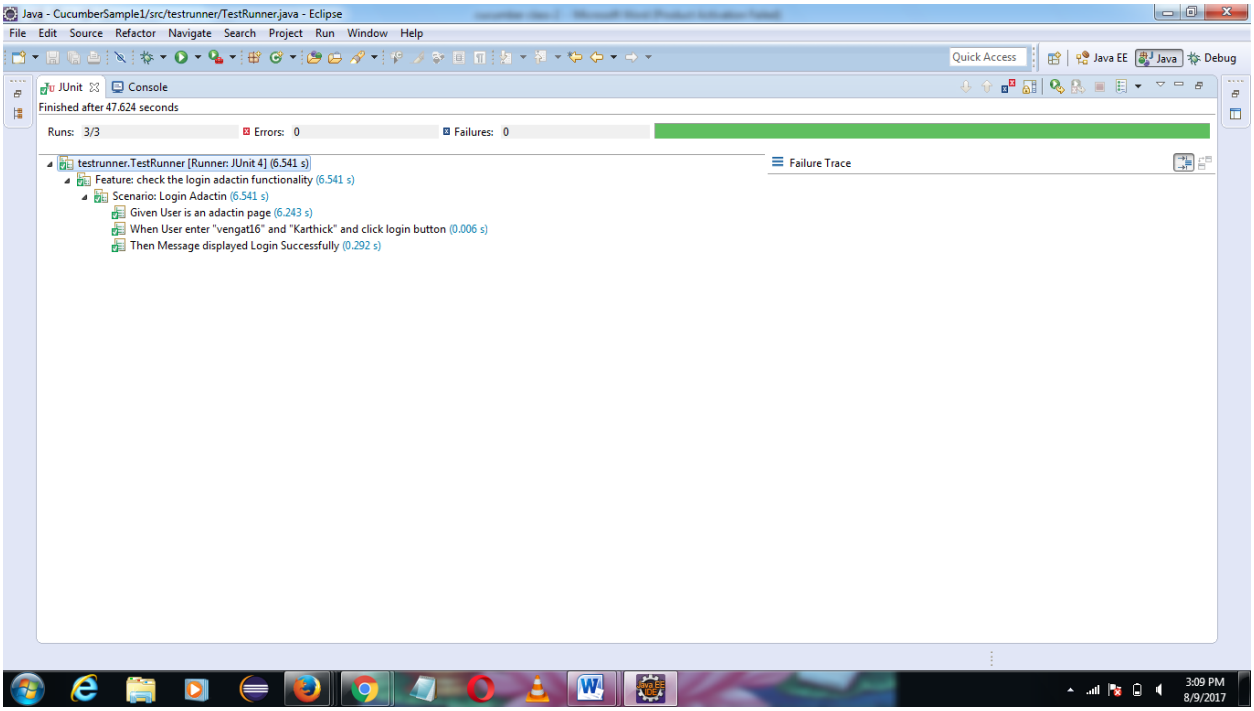
Test runner class:

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Feature", glue =
"step",plugin={"pretty","html:Report/cucumber"})
public class TestRunner {

}
```

Output:


Java - CucumberSample1/src/testrunner/TestRunner.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
AdactinAutomation [selenium1 master]
Adcantin
CucumberSample1
src
step
LoginTest.java
testrunner
TestRunner.java
JRE System Library [JavaSE-1.8]
Referenced Libraries
driver
Feature
Login.feature
Report
CucumberSampleProgram
demo
demoo
empdetails
Facebook
FaceBookAutomate
FacebooRegister
MercuryTourRegister
ReadExcel
SeleniumProjectss
TestRunner.java
Login.feature
LoginTest.java
1 package testrunner;
2
3 import org.junit.runner.RunWith;
4 import cucumber.api.CucumberOptions;
5 import cucumber.api.junit.Cucumber;
6
7 @RunWith(Cucumber.class)
8 @CucumberOptions(features = "Feature", glue = "step",plugin={"pretty","html:Report/cucumber"})
9 public class TestRunner {
10
11 }
12
Console
TestRunner (1) [JUnit] C:\Program Files\Java\jdk1.8.0_101\bin\javaw.exe (Aug 9, 2017, 3:07:40 PM)
Feature: check the login adactin functionality
1502271470303 geckodriver INFO Listening on 127.0.0.1:30599
1502271471186 geckodriver::marionette INFO Starting browser C:\Program Files\Mozilla Firefox\firefox.exe with ar
Writable
Smart Insert
4 : 1
3:07 PM
8/9/2017


AdactIn.com - Search Hotel
www.adactin.com/HotelApp/SearchHotel.php
Adactin
Innovation in Testing
Welcome to AdactIn Group of Hotels
Hello vengat16! Search Hotel | Booked Itinerary | Change Password | Logout
Search Hotel (Fields marked with Red asterix (*) are mandatory)
Location* - Select Location -
Hotels - Select Hotel -
Room Type - Select Room Type -
Number of Rooms* 1 - One
Check In Date* 09/08/2017 (dd/mm/yyyy)
Check Out Date* 10/08/2017 (dd/mm/yyyy)
Adults per Room* 1 - One
Children per Room - Select Children per Room -
Search
Reset
Sample TestCases
DOWNLOAD a complete set of readymade sample TEST CASES for this application. Enjoy automation!
HotelApp Web Services
Access Hotel App SOAP and Rest Services to learn web services testing. Click to know more.
Known Defects
DOWNLOAD list of known defects for this application.
Book on Test Automation
Test Automation using Microsoft Coded UI with C#
Test Automation using Microsoft Coded UI with C#: Step by Step Guide.
www.adactin.com/our-products/bookspublications/test-automation-using-microsoft-coded-ui-with-c/
3:08 PM
8/9/2017



To check multiple values:

- Using this method we can check the login function with multiple values without using iteration
- This is the easy way to check multiple values
  In this method we use "Examples" one keyword in feature file.
- If we use "Examples" , we must give "Scenario Outline" instead of "Scenario"

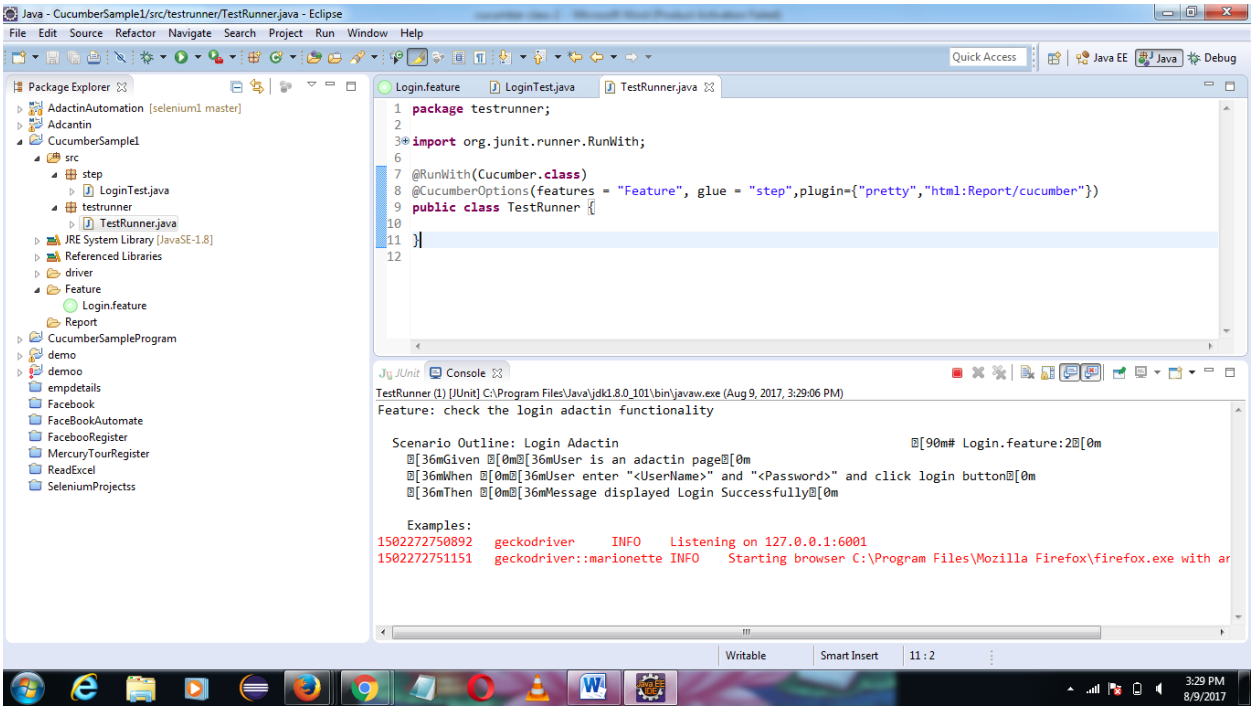
Feature file:

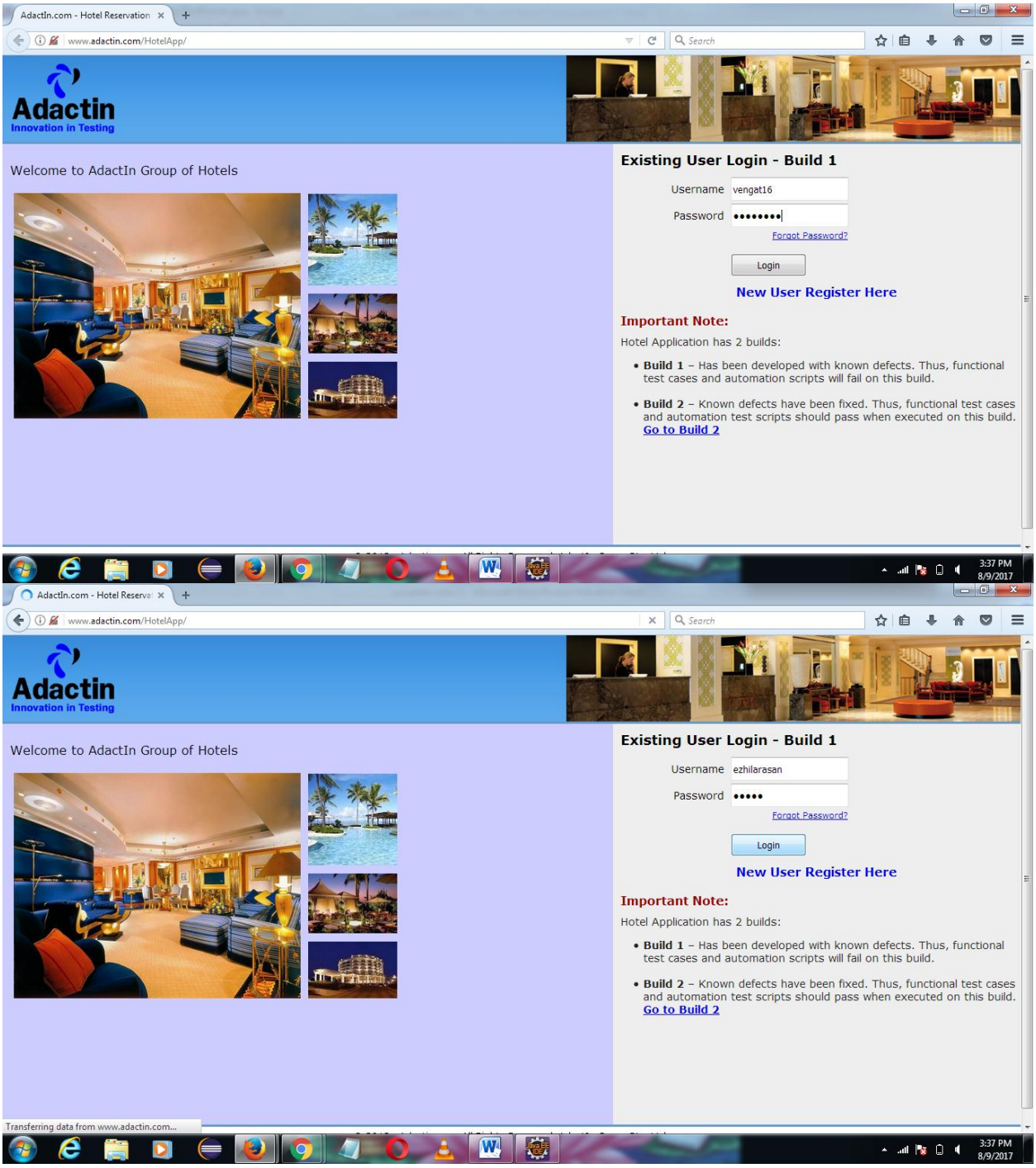
```
Feature: check the login adactin functionality
Scenario Outline: Login Adactin
      Given User is an adactin page
      When User enter "<UserName>" and "<Password>" and click login button
      Then Message displayed Login Successfully
      Examples:
      |UserName|Password|
      |vengat16|Karthick|
      |ezhilarasan|12345|
```

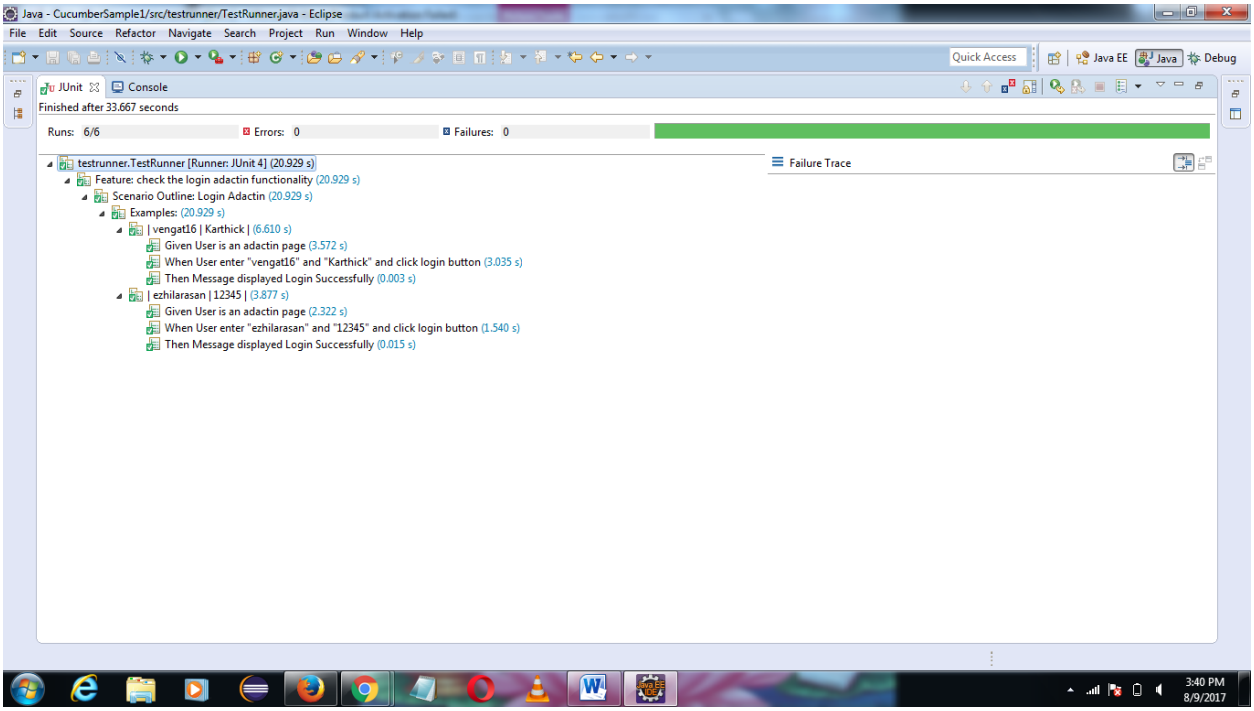
Step definition :

```
public class LoginTest {
	WebDriver driver;

	@Given("^User is an adactin page$")
	public void user_is_an_adactin_page() {
		System.setProperty("webdriver.gecko.driver",
				"H:\\java
software\\CucumberSample1\\driver\\geckodriver.exe");
		driver = new FirefoxDriver();
		driver.get("http://www.adactin.com/HotelApp/");
	}

	@When("^User enter \"([^\"]*)\" and \"([^\"]*)\" and click login
button$")
	public void user_enter_UserName_and_Password(String Username,
			String Password) {
		driver.findElement(By.id("username")).sendKeys(Username);

		driver.findElement(By.id("password")).sendKeys(Password);

		driver.findElement(By.id("login")).click();

	}

	@Then("^Message displayed Login Successfully$")
	public void message_displayed_Login_Successfully() {
		driver.quit();
	}

}
```

Output:


Java - CucumberSample1/src/testrunner/TestRunner.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java EE
Java
Debug
Package Explorer
AdactinAutomation [selenium1 master]
Adcantin
CucumberSample1
src
step
LoginTest.java
testrunner
TestRunner.java
JRE System Library [JavaSE-1.8]
Referenced Libraries
driver
Feature
Login.feature
Report
CucumberSampleProgram
demo
demoo
empdetails
Facebook
FaceBookAutomate
FacebooRegister
MercuryTourRegister
ReadExcel
SeleniumProjectss
Login.feature
LoginTest.java
TestRunner.java
1 package testrunner;
2
3 import org.junit.runner.RunWith;
6
7 @RunWith(Cucumber.class)
8 @CucumberOptions(features = "Feature", glue = "step",plugin={"pretty","html:Report/cucumber"})
9 public class TestRunner {
10
11 }
12
JUnit
Console
TestRunner (1) [JUnit] C:\Program Files\Java\jdk1.8.0_101\bin\javaw.exe (Aug 9, 2017, 3:29:06 PM)
Feature: check the login adactin functionality
Scenario Outline: Login Adactin
[36mGiven [0m[36mUser is an adactin page[0m
[36mWhen [0m[36mUser enter "<UserName>" and "<Password>" and click login button[0m
[36mThen [0m[36mMessage displayed Login Successfully[0m
Examples:
1502272750892 geckodriver INFO Listening on 127.0.0.1:6001
1502272751151 geckodriver::marionette INFO Starting browser C:\Program Files\Mozilla Firefox\firefox.exe with ar
Writable
Smart Insert
11 : 2
3:29 PM
8/9/2017


AdactIn.com - Hotel Reservation
www.adactin.com/HotelApp/
Search
Adactin
Innovation in Testing
Welcome to AdactIn Group of Hotels
Existing User Login - Build 1
Username
vengat16
Password
Forgot Password?
Login
New User Register Here
Important Note:
Hotel Application has 2 builds:
Build 1 – Has been developed with known defects. Thus, functional test cases and automation scripts will fail on this build.
Build 2 – Known defects have been fixed. Thus, functional test cases and automation test scripts should pass when executed on this build.
Go to Build 2
3:37 PM
8/9/2017
AdactIn.com - Hotel Reserva
www.adactin.com/HotelApp/
Search
Adactin
Innovation in Testing
Welcome to AdactIn Group of Hotels
Existing User Login - Build 1
Username
ezhilarasan
Password
Forgot Password?
Login
New User Register Here
Important Note:
Hotel Application has 2 builds:
Build 1 – Has been developed with known defects. Thus, functional test cases and automation scripts will fail on this build.
Build 2 – Known defects have been fixed. Thus, functional test cases and automation test scripts should pass when executed on this build.
Go to Build 2
Transferring data from www.adactin.com...
3:37 PM
8/9/2017



- If we enter more than one value it will automatically execute one by one.

Entering value in future file (another method):

By using List:

Feature file:

```
Feature: check the login adactin functionality
Scenario: Login Adactin
	Given User is an adactin page
	When User enters credentials and click login button
	|vengat16|Karthick|
	|ezhilarasan|12345|
	Then Message displayed Login Successfully
```

Step definition:

```
public class LoginTest {
	WebDriver driver;

	@Given("^User is an adactin page$")
	public void user_is_an_adactin_page() {
		System.setProperty("webdriver.gecko.driver",
```

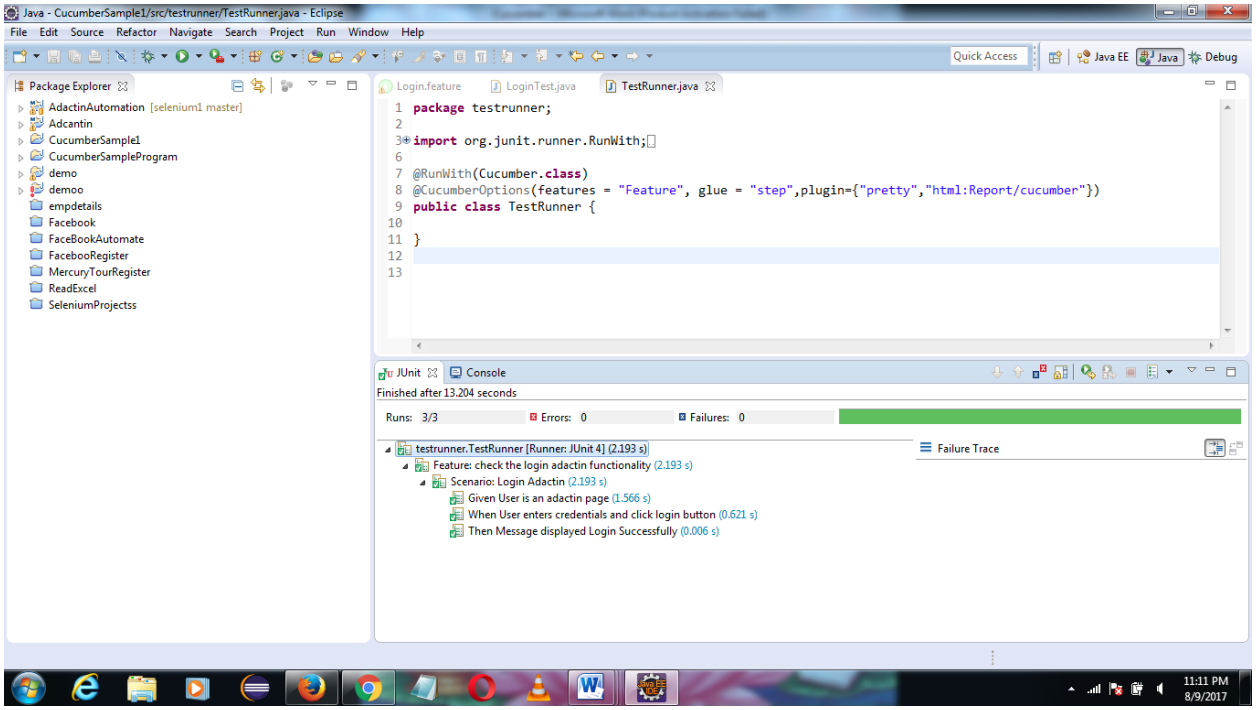
```
                        "H:\\java
software\\CucumberSample1\\driver\\geckodriver.exe");
            driver = new FirefoxDriver();
            driver.get("http://www.adactin.com/HotelApp/");
      }
      @When("^User enters credentials and click login button$")
      public void user_enters_credentials_and_click_login_button(DataTable
data) {
            List<List<String>> dataList = data.raw();
driver.findElement(By.id("username")).sendKeys(dataList.get(1).get(0));


      driver.findElement(By.id("password")).sendKeys(dataList.get(1).get(1));

            driver.findElement(By.id("login")).click();
      }


      @Then("^Message displayed Login Successfully$")
      public void message_displayed_Login_Successfully() {
            driver.quit();
      }

}
```

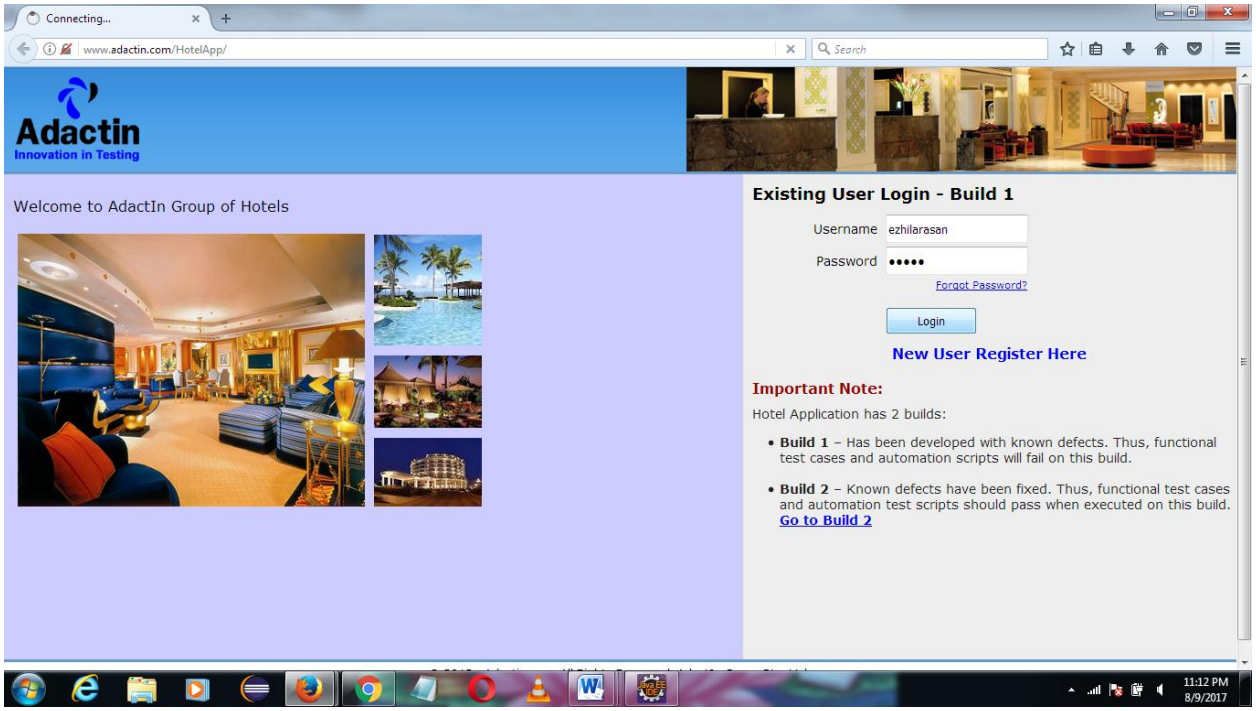
- This is the data table concept
- Here we input the data based on index

Test runner class:

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Feature", glue =
"step",plugin={"pretty","html:Report/cucumber"})
public class TestRunner {

}
```

Output:





- Here we given first index, so it takes first index value.
- If we give 0^{th} index, it will take 0^{th} value

By using map:

Feature file:

```
Feature: check the login adactin functionality
Scenario: Login Adactin
	Given User is an adactin page
	When User enters credentials and click login button
	|userName|Password|
	|vengat16|Karthick|
	|ezhilarasan|12345|
	Then Message displayed Login Successfully
```

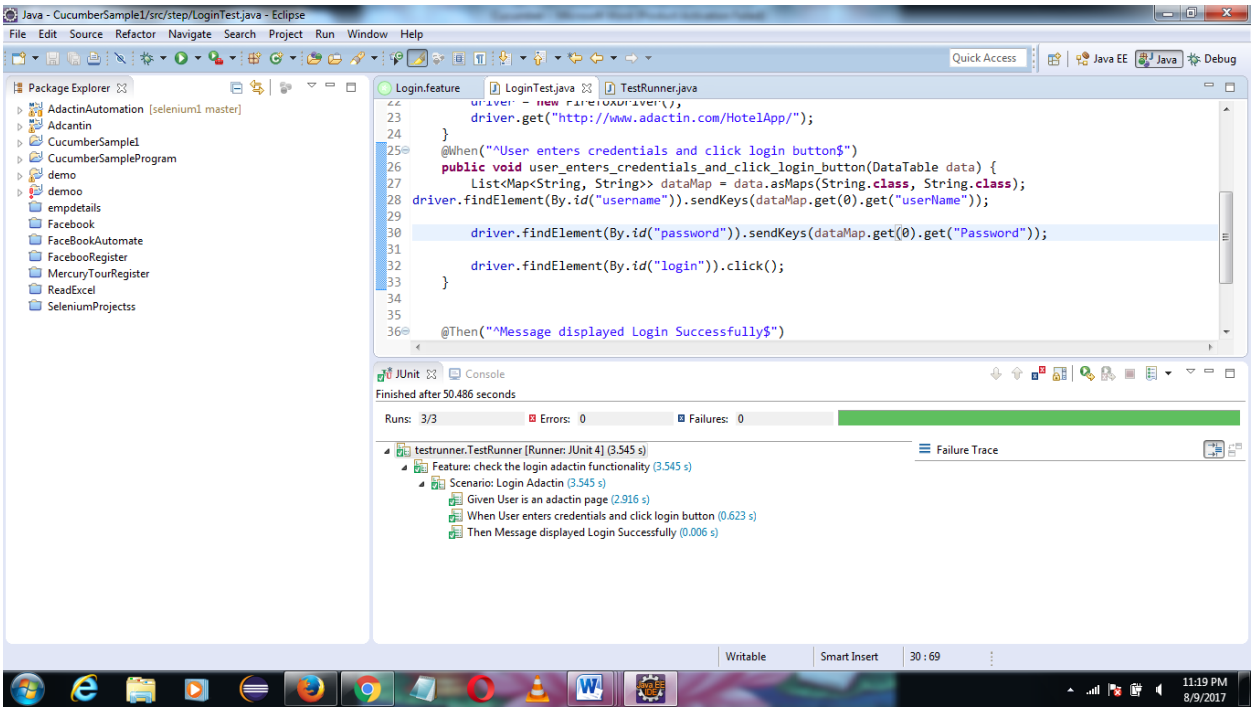
Step definition :

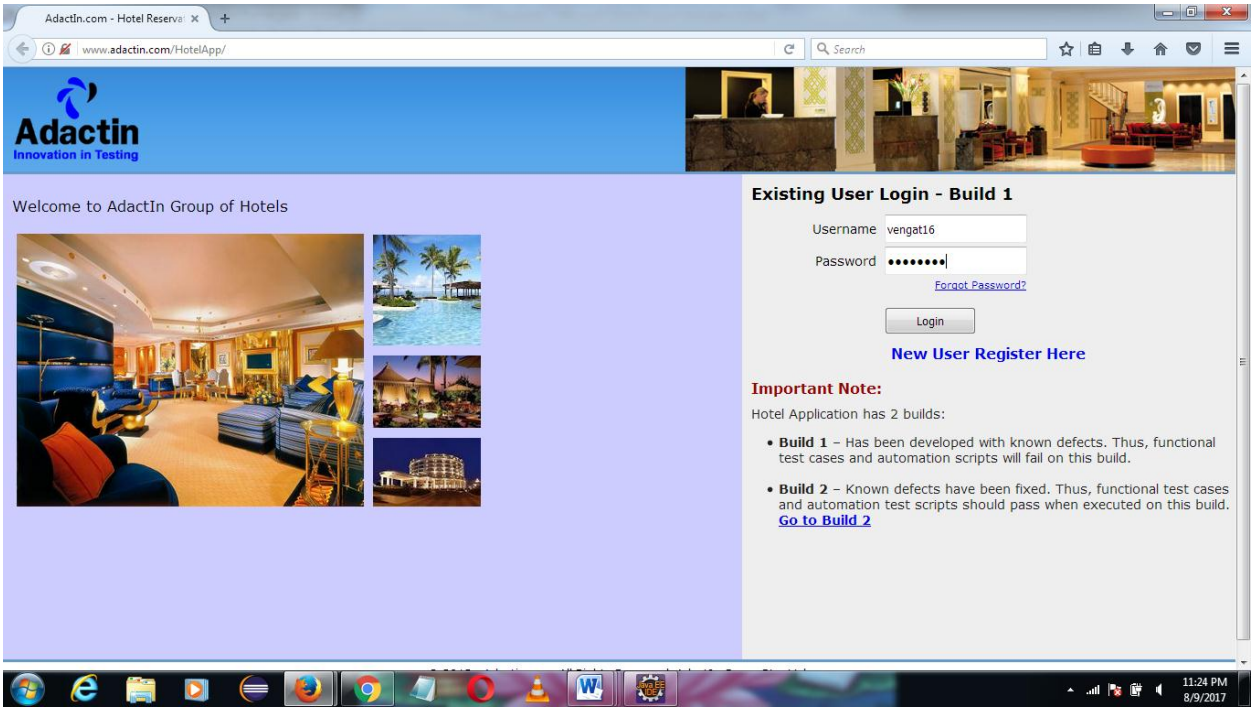
```
public class LoginTest {
	WebDriver driver;

	@Given("^User is an adactin page$")
	public void user_is_an_adactin_page() {
		System.setProperty("webdriver.gecko.driver",
				"H:\\java
software\\CucumberSample1\\driver\\geckodriver.exe");
		driver = new FirefoxDriver();
		driver.get("http://www.adactin.com/HotelApp/");
	}
	@When("^User enters credentials and click login button$")
	public void user_enters_credentials_and_click_login_button(DataTable
data) {
		List<Map<String, String>> dataMap = data.asMaps(String.class,
String.class);
driver.findElement(By.id("username")).sendKeys(dataMap.get(0).get("userName")
);


	driver.findElement(By.id("password")).sendKeys(dataMap.get(0).get("Pass
word"));

		driver.findElement(By.id("login")).click();
	}


	@Then("^Message displayed Login Successfully$")
	public void message_displayed_Login_Successfully() {
		driver.quit();
	}

}
```

Test runner class:

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Feature", glue =
"step",plugin={"pretty","html:Report/cucumber"})
public class TestRunner {

}
```

Output :





- Here we using map, so we pass the key and it will enter relevent values of the particular index

<u>To check random input values</u>

- Here for example , one unique username is there, using this user name we need to check multiple times. That time we get error(i.e) “user name already registered”, these kind of error will show.
- To avoid these, we use **RandomStringUtils** class and append the value in the user name
- So we get the random input
- Syntax :
  - To append alphabet value

    ```
    String s = RandomStringUtils.randomAlphabetic(4);
    Here, 4 means 4 alphabet letters will add randomly
    ```

  - To append alphabet value

    ```
    String s = RandomStringUtils.randomAlphanumeric (4);
    Here, 4 means 4 alphanumeric letters will add randomly
    ```

- To append alphabet value

  ```
  String s = RandomStringUtils.randomNumeric (4);
  Here, 4 means 4 numeric values will add randomly
  ```

Feature file:

```
Feature: check the login adactin functionality
Scenario: Login Adactin
	Given User is an adactin page
	When User enters credentials and click login button
	|userName|Password|
	|vengat16|Karthick|
	|ezhilarasan|12345|
	Then Message displayed Login Successfully
```

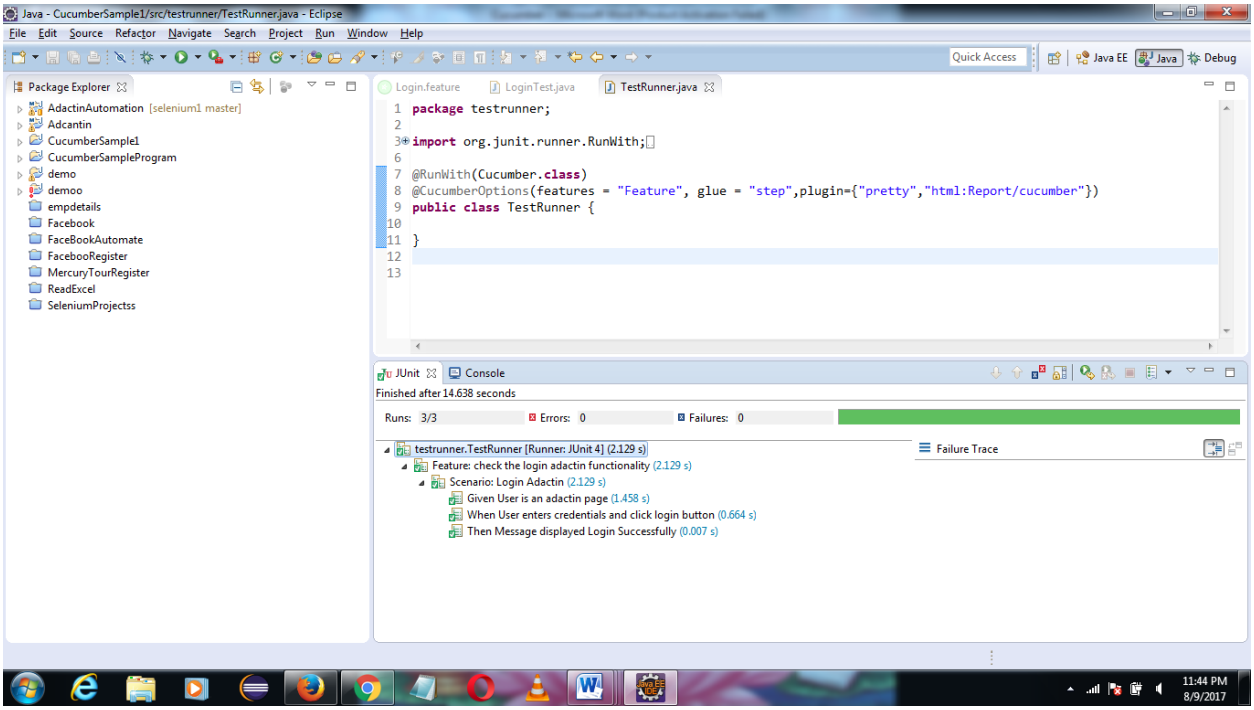
Step definition :

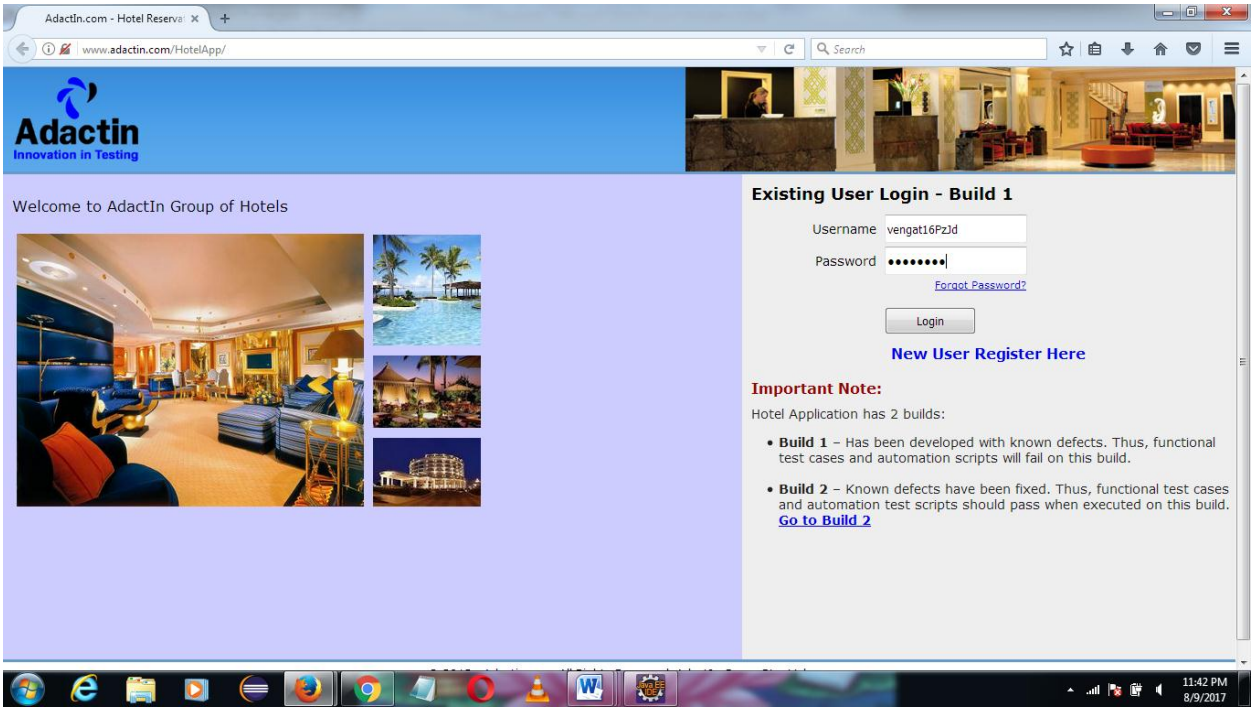
```
public class LoginTest {
	WebDriver driver;

	@Given("^User is an adactin page$")
	public void user_is_an_adactin_page() {
		System.setProperty("webdriver.gecko.driver",
				"H:\\java
software\\CucumberSample1\\driver\\geckodriver.exe");
		driver = new FirefoxDriver();
		driver.get("http://www.adactin.com/HotelApp/");
	}
	@When("^User enters credentials and click login button$")
	public void user_enters_credentials_and_click_login_button(DataTable
data) {
		String s = RandomStringUtils.randomAlphanumeric(4);
		List<Map<String, String>> dataMap = data.asMaps(String.class,
String.class);
driver.findElement(By.id("username")).sendKeys(dataMap.get(0).get("userName")
+s);


	driver.findElement(By.id("password")).sendKeys(dataMap.get(0).get("Pass
word"));

		driver.findElement(By.id("login")).click();
	}

	@Then("^Message displayed Login Successfully$")
	public void message_displayed_Login_Successfully() {
		driver.quit();
	}}
```

Test runner class:

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Feature", glue =
"step",plugin={"pretty","html:Report/cucumber"})
public class TestRunner {

}
```

Output :



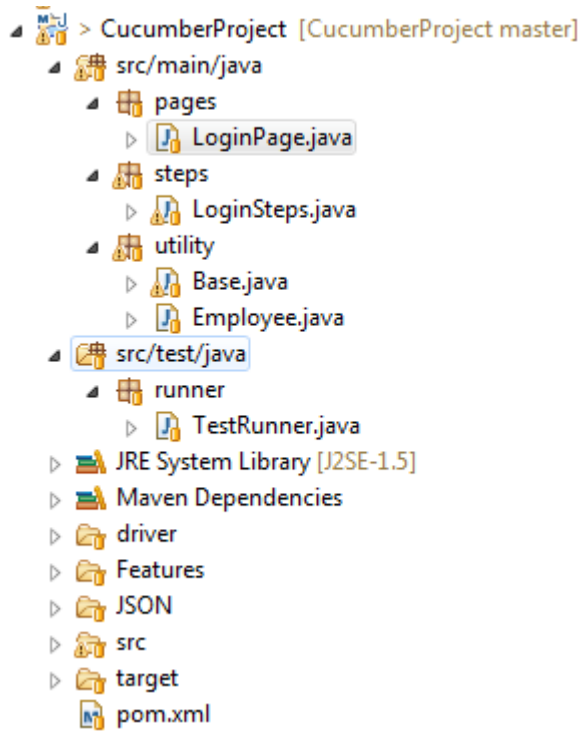


- Here 4 alphanumeric letters added in the user name
- It will randomly change for the next iteration

Cucumber with maven and POM:

- Here we use maven, POM and cucumber combination

Framework structure:



POM class:

```
public class LoginPage extends Base {

	@FindBy(id = "email")
	private WebElement txtUserName;

	@FindBy(id = "pass")
	private WebElement txtPassword;

	@FindBy(xpath = "//*[text()='Log In']")
	private WebElement btnLogin;

	@FindBy(xpath = "//a[@title='Go to Facebook home']")
	private WebElement imgFbLogo;

	public WebElement getImgFbLogo() {
		return imgFbLogo;
	}
```

```
public void setImgFbLogo(WebElement imgFbLogo) {
        this.imgFbLogo = imgFbLogo;
}

public void setTxtUserName(WebElement txtUserName) {
        this.txtUserName = txtUserName;
}

public void setTxtPassword(WebElement txtPassword) {
        this.txtPassword = txtPassword;
}

public LoginPage() {
        PageFactory.initElements(driver, this);

}

public WebElement getTxtUserName() {
        return txtUserName;
}

public WebElement getTxtPassword() {
        return txtPassword;
}

public void setTxtPassword(String txtPassword) {
        this.txtPassword.sendKeys(txtPassword);
}

public WebElement getBtnLogin() {
        return btnLogin;
}

public void setBtnLogin(WebElement btnLogin) {
        this.btnLogin = btnLogin;
}



        }
```

Step definition:

```
public class LoginSteps extends Base {
      WebDriver driver;
      LoginPage loginPage;
```

```
	@Given("^User is on facebook Page$")
	public void user_is_on_Home_Page() {
		driver = getDriver();

	}

	@When("^User enters \"([^\"]*)\",\"([^\"]*)\" and click the login
button$")
	public void user_enters_and_click_the_login_button(String userName,
			String Password)  {
		loginPage = new LoginPage();

		setText(loginPage.getTxtUserName(), userName);
		setText(loginPage.getTxtPassword(), Password);
		clickBtn(loginPage.getBtnLogin());

	}

	@Then("^Message displayed Login Successfully$")
	public void message_displayed_Login_Successfully() {

		driver.quit();

	}

}
```

Feature file:

Feature: check the login adactin functionality
Scenario: Login Adactin
Given User is an adactin page
When User enter "vengat16" and "Karthick" and click login button
Then Message displayed Login Successfully

Utility package:

Base:

```
public class Base {
	public static WebDriver driver;
	static WebDriverWait wait;
	static File f1 = new File("./JSON/Configuration.json");

	public static WebDriver getDriver() {
		JSONObject jsonObject = JSONReadFromFile();
		String browser = (String) jsonObject.get("browser");
```

```
            File f = new File("./driver");
            if (browser.equals("chrome")) {
                  System.setProperty("webdriver.chrome.driver",
f.getAbsolutePath()
                              + "/chromedriver.exe");
                  driver = new ChromeDriver();

            } else if (browser.equals("firefox")) {
                  System.setProperty("webdriver.gecko.driver",
f.getAbsolutePath()
                              + "/geckodriver.exe");
                  driver = new FirefoxDriver();

            } else if (browser.equals("ie")) {
                  System.setProperty("webdriver.ie.driver",
f.getAbsolutePath()
                              + "/IEDriverServer.exe");
                  driver = new InternetExplorerDriver();

            }

            driver.manage().window().maximize();
            driver.get((String) jsonObject.get("url"));
            return driver;
      }

      public static boolean elementToBeVisible(WebDriver driver, int time,
                  WebElement element) {
            boolean flag = false;
            try {
                  wait = new WebDriverWait(driver, time);
                  wait.until(ExpectedConditions.visibilityOf(element));
                  flag = true;
            } catch (Exception e) {
                  e.printStackTrace();
            }
            return flag;
      }

      public static boolean alertIsPresent(WebDriver driver, int time) {
            boolean flag = false;
            try {
                  wait = new WebDriverWait(driver, time);
                  wait.until(ExpectedConditions.alertIsPresent());
                  flag = true;
            } catch (Exception e) {
                  e.printStackTrace();
            }
            return flag;
      }
```

```
	public static boolean elementToBeClickable(WebDriver driver, int time,
			WebElement element) {
		boolean flag = false;
		try {
			wait = new WebDriverWait(driver, time);

	wait.until(ExpectedConditions.elementToBeClickable(element));
			flag = true;
		} catch (Exception e) {
			e.printStackTrace();
		}
		return flag;
	}

	public static boolean elementFound(WebDriver driver, int time,
			WebElement element) {
		boolean res = false;
		driver.manage().timeouts().implicitlyWait(time,
TimeUnit.SECONDS);
		try {

			res = element.isDisplayed();
		} catch (Exception e) {
			e.printStackTrace();

		}
		return res;

	}

	public static boolean elementFound(WebElement element) {
		boolean res = false;
		try {
			res = element.isDisplayed();
		} catch (Exception e) {
			e.printStackTrace();

		}
		return res;
	}

	public static void setText(WebElement element, String name) {
		if (name != null && elementFound(element)) {
			element.clear();
			element.sendKeys(name);
		}

	}
```

```
	public static String getText(WebElement element) {
		String name = null;
		if (elementFound(element)) {
			name = element.getAttribute("value");
		}
		return name;
	}

	public static void clickBtn(WebElement element) {
	if (elementFound(element)) {
			element.click();
		}
	}

	public static JSONObject JSONReadFromFile() {
		JSONParser parser = new JSONParser();
		JSONObject jsonObject = null;
		try {

			Object obj = parser.parse(new
FileReader(f1.getAbsoluteFile()));

			jsonObject = (JSONObject) obj;

		} catch (Exception e) {
			e.printStackTrace();
		}
		return jsonObject;
	}

	public static void getScreenShot(String screenShotFileName) {
		File screenShotLocation = new File("./screenshot/" +
screenShotFileName
				+ ".png");
		TakesScreenshot screenshot = (TakesScreenshot) driver;
		File file = screenshot.getScreenshotAs(OutputType.FILE);
		try {
			FileUtils.copyFile(file, screenShotLocation);
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
	}

	public static void uploadFiles(File path) {
		try {
			Robot robot = new Robot();
			robot.setAutoDelay(3000);
			StringSelection selection = new StringSelection(
					path.getAbsolutePath());
```

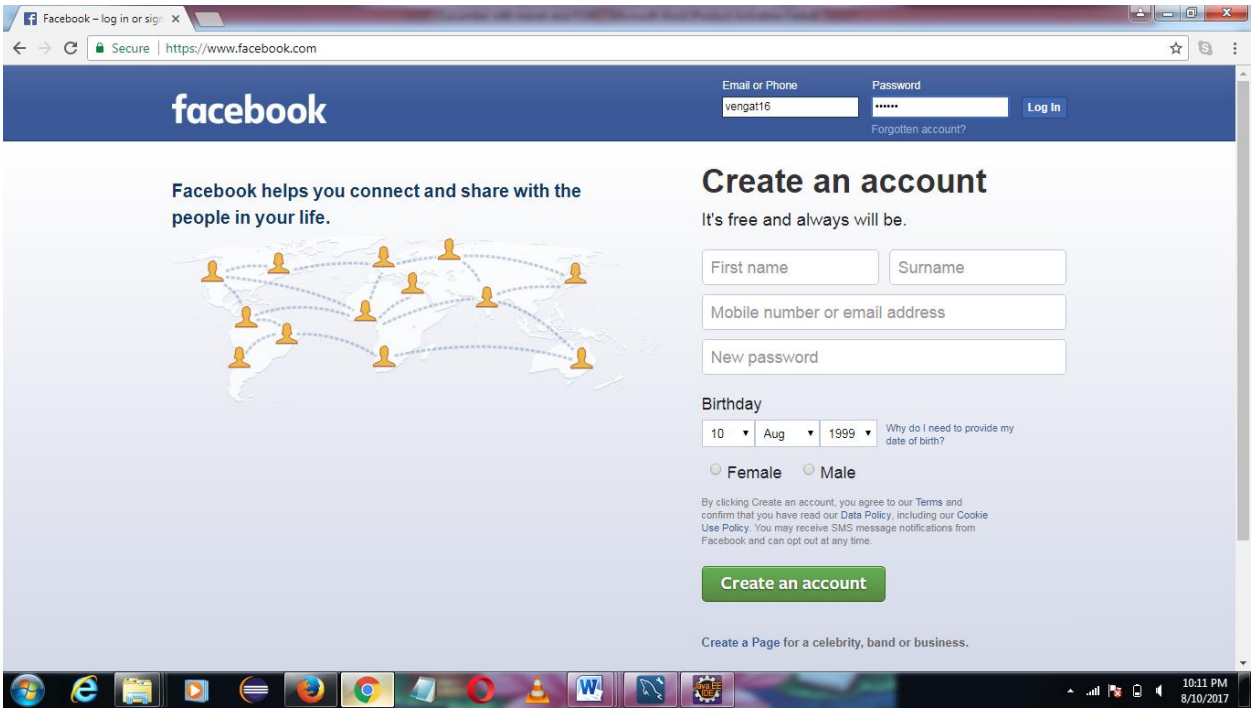
```
                Toolkit.getDefaultToolkit().getSystemClipboard()
                            .setContents(selection, null);
                // press control+v
                robot.keyPress(KeyEvent.VK_CONTROL);
                robot.keyPress(KeyEvent.VK_V);
                robot.setAutoDelay(3000);
                // release control+v
                robot.keyRelease(KeyEvent.VK_CONTROL);
                robot.keyRelease(KeyEvent.VK_V);
                // press enter
                robot.setAutoDelay(3000);
                robot.keyPress(KeyEvent.VK_ENTER);
                robot.keyRelease(KeyEvent.VK_ENTER);

          } catch (AWTException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
          }

    }

    public static List<HashMap<String, String>> readValueFromExcelSheet() {
          List<HashMap<String, String>> mapDatasList = new ArrayList();
          try {
                File excelLocaltion = new File("./Excel/Facebook.xlsx");

                String sheetName = "Datas";

                FileInputStream f = new FileInputStream(
                            excelLocaltion.getAbsolutePath());
                Workbook w = new XSSFWorkbook(f);
                Sheet sheet = w.getSheet(sheetName);
                Row headerRow = sheet.getRow(0);
                for (int i = 0; i < sheet.getPhysicalNumberOfRows(); i++) {
                      Row currentRow = sheet.getRow(i);
                      HashMap<String, String> mapDatas = new
HashMap<String, String>();
                      for (int j = 0; j <
headerRow.getPhysicalNumberOfCells(); j++) {
                            Cell currentCell = currentRow.getCell(j);

                            switch (currentCell.getCellType()) {
                            case Cell.CELL_TYPE_STRING:

    mapDatas.put(headerRow.getCell(j).getStringCellValue(),

    currentCell.getStringCellValue());
                                  break;
                            case Cell.CELL_TYPE_NUMERIC:
```

```
		mapDatas.put(headerRow.getCell(j).getStringCellValue(),
                                                String.valueOf(currentCell

		.getNumericCellValue()));

                                    break;

                              }
                        }

                        mapDatasList.add(mapDatas);
                  }

            } catch (Throwable e) {
                  e.printStackTrace();
            }
            return mapDatasList;

      }

      public static List<Employee> retriveValueFromDataBase() {
            ResultSet rs = null;
            List<Employee> emp = new ArrayList<Employee>();
            try {
                  Class.forName("com.mysql.jdbc.Driver");
                  Connection con = DriverManager.getConnection(
                              "jdbc:mysql://127.0.0.1:3306/selenum_schema",
"root", "");
                  PreparedStatement ps = con
                              .prepareStatement("SELECT * FROM
selenium_schema.emp_table where roll=3;");

                  rs = ps.executeQuery();

                  while (rs.next()) {
                        Employee e = new Employee();
                        e.setName(rs.getString("name"));
                        e.setPassword(rs.getString("password"));
                        emp.add(e);

                  }
            } catch (ClassNotFoundException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }
            return emp;}}
```

Test runner class:

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Features", glue = "steps")
public class TestRunner {

}
```

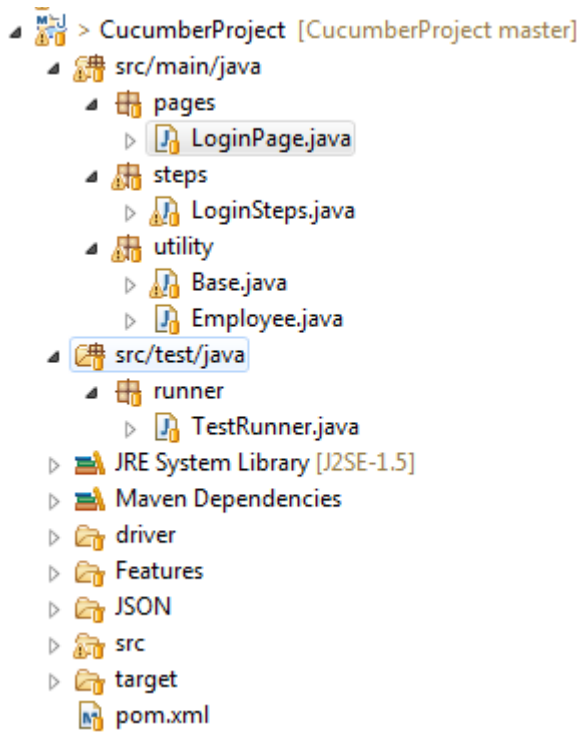
Output:



Cucumber with POM and data base:

- In this method, we taken all the input data's from data base

Framework structure:



Step definition:

```
public class LoginSteps extends Base {
	WebDriver driver;
	LoginPage loginPage;

	@Given("^User is on facebook Page$")
	public void user_is_on_Home_Page() {
		driver = getDriver();

	}

	@When("^User enters \"([^\"]*)\",\"([^\"]*)\" and click the login
button$")
	public void user_enters_and_click_the_login_button(String userName,
			String Password) throws SQLException {
		loginPage = new LoginPage();
		List<Employee> emp = retriveValueFromDataBase();

		setText(loginPage.getTxtUserName(), emp.get(0).getName());
		setText(loginPage.getTxtPassword(), emp.get(0).getPassword());
		clickBtn(loginPage.getBtnLogin());
```

```
	}

	@Then("^Message displayed Login Successfully$")
	public void message_displayed_Login_Successfully() {

		driver.quit();

	}

}
```

POM class:

```
public class LoginPage extends Base {

	@FindBy(id = "email")
	private WebElement txtUserName;

	@FindBy(id = "pass")
	private WebElement txtPassword;

	@FindBy(xpath = "//*[text()='Log In']")
	private WebElement btnLogin;

	@FindBy(xpath = "//a[@title='Go to Facebook home']")
	private WebElement imgFbLogo;

	public WebElement getImgFbLogo() {
		return imgFbLogo;
	}

	public void setImgFbLogo(WebElement imgFbLogo) {
		this.imgFbLogo = imgFbLogo;
	}

	public void setTxtUserName(WebElement txtUserName) {
		this.txtUserName = txtUserName;
	}

	public void setTxtPassword(WebElement txtPassword) {
		this.txtPassword = txtPassword;
	}

	public LoginPage() {
		PageFactory.initElements(driver, this);

	}
```

```
	public WebElement getTxtUserName() {
		return txtUserName;
	}

	public WebElement getTxtPassword() {
		return txtPassword;
	}

	public void setTxtPassword(String txtPassword) {
		this.txtPassword.sendKeys(txtPassword);
	}

	public WebElement getBtnLogin() {
		return btnLogin;
	}

	public void setBtnLogin(WebElement btnLogin) {
		this.btnLogin = btnLogin;
	}
	}
```

Utility package:

Base:

```
public class Base {
	public static WebDriver driver;
	static WebDriverWait wait;
	static File f1 = new File("./JSON/Configuration.json");

	public static WebDriver getDriver() {
		JSONObject jsonObject = JSONReadFromFile();
		String browser = (String) jsonObject.get("browser");

		File f = new File("./driver");
		if (browser.equals("chrome")) {
			System.setProperty("webdriver.chrome.driver",
f.getAbsolutePath()
					+ "/chromedriver.exe");
			driver = new ChromeDriver();

		} else if (browser.equals("firefox")) {
			System.setProperty("webdriver.gecko.driver",
f.getAbsolutePath()
					+ "/geckodriver.exe");
			driver = new FirefoxDriver();

		} else if (browser.equals("ie")) {
```

```
                  System.setProperty("webdriver.ie.driver",
f.getAbsolutePath()
                              + "/IEDriverServer.exe");
                  driver = new InternetExplorerDriver();

            }

            driver.manage().window().maximize();
            driver.get((String) jsonObject.get("url"));
            return driver;
      }

      public static boolean elementToBeVisible(WebDriver driver, int time,
                  WebElement element) {
            boolean flag = false;
            try {
                  wait = new WebDriverWait(driver, time);
                  wait.until(ExpectedConditions.visibilityOf(element));
                  flag = true;
            } catch (Exception e) {
                  e.printStackTrace();
            }
            return flag;
      }

      public static boolean alertIsPresent(WebDriver driver, int time) {
            boolean flag = false;
            try {
                  wait = new WebDriverWait(driver, time);
                  wait.until(ExpectedConditions.alertIsPresent());
                  flag = true;
            } catch (Exception e) {
                  e.printStackTrace();
            }
            return flag;
      }

      public static boolean elementToBeClickable(WebDriver driver, int time,
                  WebElement element) {
            boolean flag = false;
            try {
                  wait = new WebDriverWait(driver, time);

      wait.until(ExpectedConditions.elementToBeClickable(element));
                  flag = true;
            } catch (Exception e) {
                  e.printStackTrace();
            }
            return flag;
      }
```

```
	public static boolean elementFound(WebDriver driver, int time,
				WebElement element) {
			boolean res = false;
			driver.manage().timeouts().implicitlyWait(time,
TimeUnit.SECONDS);
			try {

				res = element.isDisplayed();
			} catch (Exception e) {
				e.printStackTrace();

			}
			return res;

	}

	public static boolean elementFound(WebElement element) {
			boolean res = false;
			try {
				res = element.isDisplayed();
			} catch (Exception e) {
				e.printStackTrace();

			}
			return res;
	}

	public static void setText(WebElement element, String name) {
			if (name != null && elementFound(element)) {
				element.clear();
				element.sendKeys(name);
			}

	}

	public static String getText(WebElement element) {
			String name = null;
			if (elementFound(element)) {
				name = element.getAttribute("value");
			}
			return name;
	}

	public static void clickBtn(WebElement element) {
	if (elementFound(element)) {
				element.click();
			}
	}
```

```
	public static JSONObject JSONReadFromFile() {
		JSONParser parser = new JSONParser();
		JSONObject jsonObject = null;
		try {

			Object obj = parser.parse(new
FileReader(f1.getAbsoluteFile()));

			jsonObject = (JSONObject) obj;

		} catch (Exception e) {
			e.printStackTrace();
		}
		return jsonObject;
	}

	public static void getScreenShot(String screenShotFileName) {
		File screenShotLocation = new File("./screenshot/" +
screenShotFileName
				+ ".png");
		TakesScreenshot screenshot = (TakesScreenshot) driver;
		File file = screenshot.getScreenshotAs(OutputType.FILE);
		try {
			FileUtils.copyFile(file, screenShotLocation);
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
	}

	public static void uploadFiles(File path) {
		try {
			Robot robot = new Robot();
			robot.setAutoDelay(3000);
			StringSelection selection = new StringSelection(
					path.getAbsolutePath());
			Toolkit.getDefaultToolkit().getSystemClipboard()
					.setContents(selection, null);
			// press control+v
			robot.keyPress(KeyEvent.VK_CONTROL);
			robot.keyPress(KeyEvent.VK_V);
			robot.setAutoDelay(3000);
			// release control+v
			robot.keyRelease(KeyEvent.VK_CONTROL);
			robot.keyRelease(KeyEvent.VK_V);
			// press enter
			robot.setAutoDelay(3000);
			robot.keyPress(KeyEvent.VK_ENTER);
			robot.keyRelease(KeyEvent.VK_ENTER);
```

```
		} catch (AWTException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

	}

	public static List<HashMap<String, String>> readValueFromExcelSheet() {
		List<HashMap<String, String>> mapDatasList = new ArrayList();
		try {
			File excelLocaltion = new File("./Excel/Facebook.xlsx");

			String sheetName = "Datas";

			FileInputStream f = new FileInputStream(
					excelLocaltion.getAbsolutePath());
			Workbook w = new XSSFWorkbook(f);
			Sheet sheet = w.getSheet(sheetName);
			Row headerRow = sheet.getRow(0);
			for (int i = 0; i < sheet.getPhysicalNumberOfRows(); i++) {
				Row currentRow = sheet.getRow(i);
				HashMap<String, String> mapDatas = new
HashMap<String, String>();
				for (int j = 0; j <
headerRow.getPhysicalNumberOfCells(); j++) {
					Cell currentCell = currentRow.getCell(j);

					switch (currentCell.getCellType()) {
					case Cell.CELL_TYPE_STRING:

	mapDatas.put(headerRow.getCell(j).getStringCellValue(),

	currentCell.getStringCellValue());
						break;
					case Cell.CELL_TYPE_NUMERIC:

	mapDatas.put(headerRow.getCell(j).getStringCellValue(),
							String.valueOf(currentCell

	.getNumericCellValue()));

						break;

					}
				}

				mapDatasList.add(mapDatas);
			}

		} catch (Throwable e) {
```

```
                e.printStackTrace();
            }
            return mapDatasList;

      }

      public static List<Employee> retriveValueFromDataBase() {
            ResultSet rs = null;
            List<Employee> emp = new ArrayList<Employee>();
            try {
                  Class.forName("com.mysql.jdbc.Driver");
                  Connection con = DriverManager.getConnection(
                              "jdbc:mysql://127.0.0.1:3306/selenum_schema",
"root", "");
                  PreparedStatement ps = con
                              .prepareStatement("SELECT * FROM
selenium_schema.emp_table where roll=3;");

                  rs = ps.executeQuery();

                  while (rs.next()) {
                        Employee e = new Employee();
                        e.setName(rs.getString("name"));
                        e.setPassword(rs.getString("password"));
                        emp.add(e);

                  }
            } catch (ClassNotFoundException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }
            return emp;

      }

}
```

Employee:

```
public class Employee {
      private String name;
      private String password;

      public String getName() {
            return name;
      }
```

```
	public void setName(String name) {
		this.name = name;
	}

	public String getPassword() {
		return password;
	}

	public void setPassword(String password) {
		this.password = password;
	}

}
```

Test runner class:

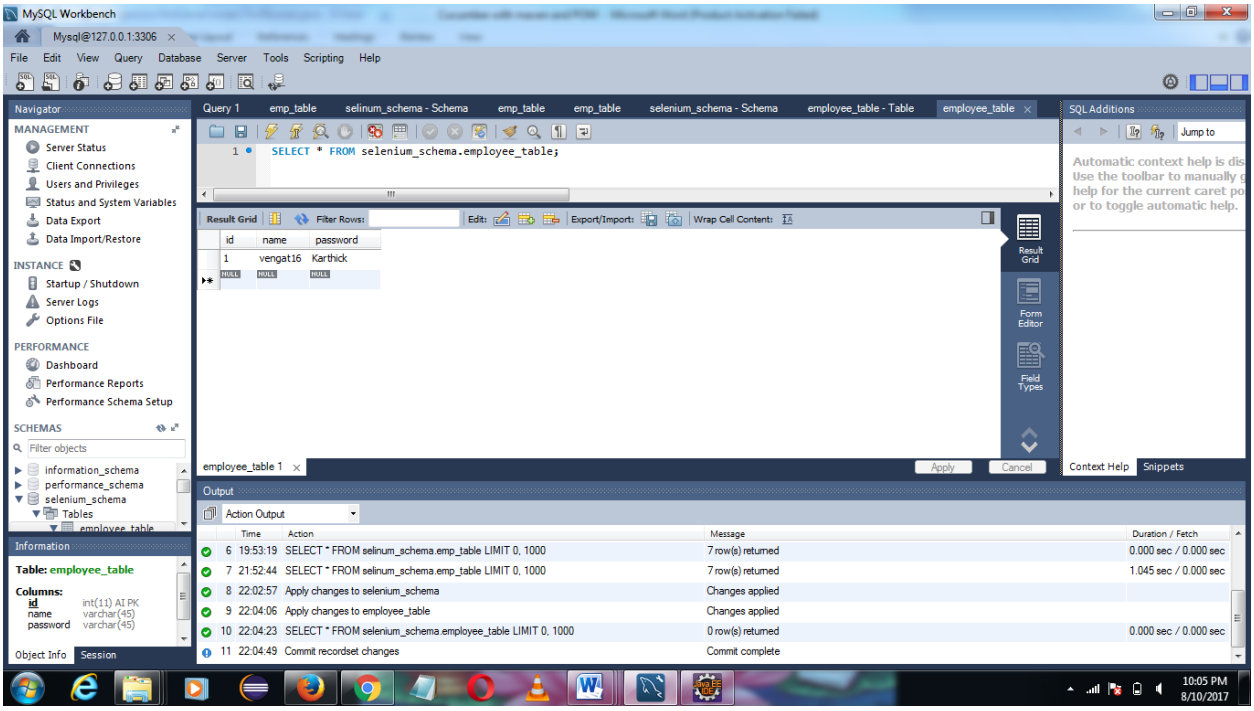
```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Features", glue = "steps")
public class TestRunner {

}
```

Feature file:

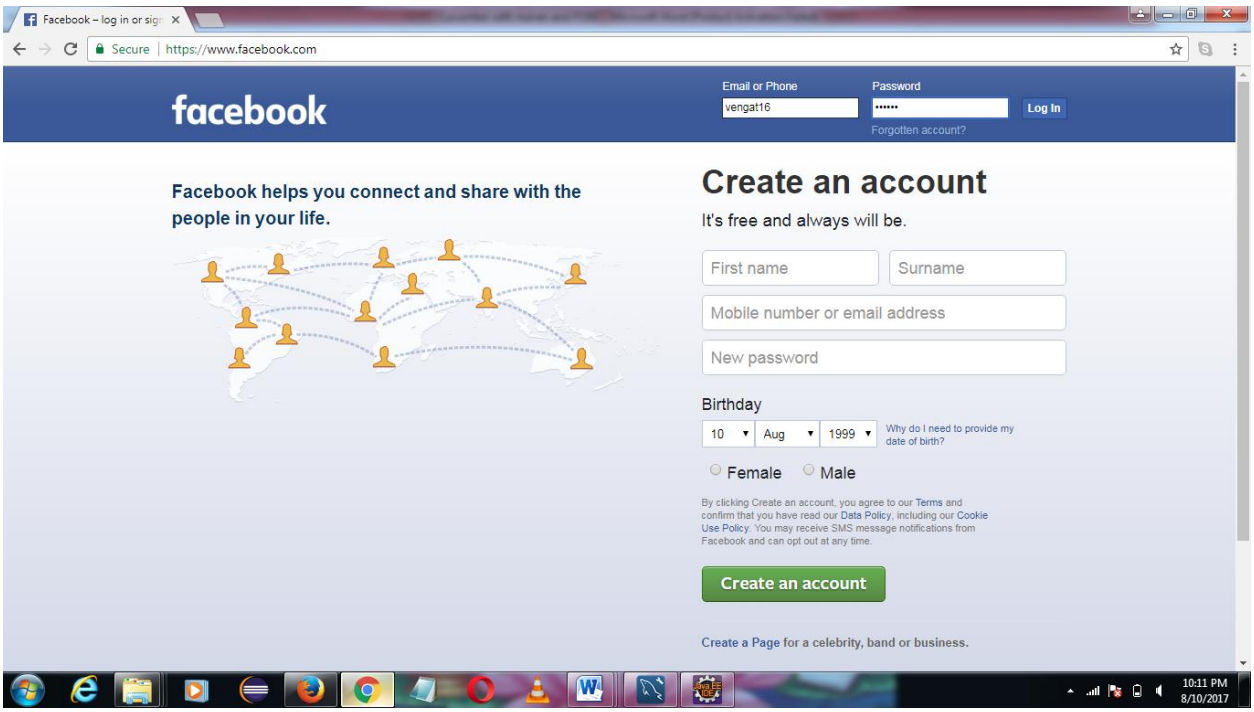
```
Scenario Outline: Successful Login with Valid Credentials
    Given User is on facebook Page
    When User enters "<userName>","<password>" and click the login  button
    Then Message displayed Login Successfully

    Examples:
      | userName | password |
      | ganesh   | Java65   |
```

Input data:

- This is a input data from data base



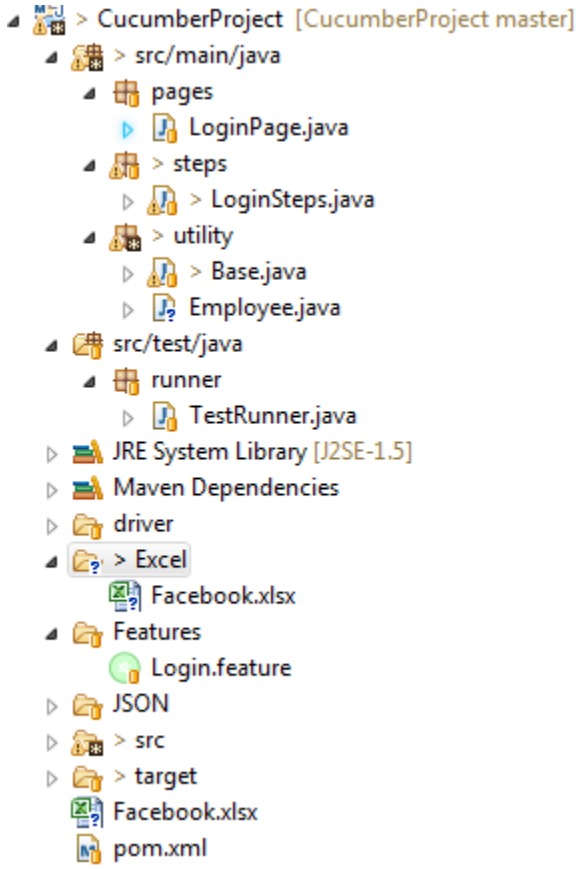
Output :



Cucumber with POM and Data driven:

- Here we use data driven to read the input data's from excel sheet

Framework structure:



- Here one excel folder will added and in this folder have one excel sheet contains all the input datas

POM class:

```
public class LoginPage extends Base {

	@FindBy(id = "email")
	private WebElement txtUserName;

	@FindBy(id = "pass")
	private WebElement txtPassword;

	@FindBy(xpath = "//*[text()='Log In']")
	private WebElement btnLogin;
```

```
@FindBy(xpath = "//a[@title='Go to Facebook home']")
private WebElement imgFbLogo;

public WebElement getImgFbLogo() {
	return imgFbLogo;
}

public void setImgFbLogo(WebElement imgFbLogo) {
	this.imgFbLogo = imgFbLogo;
}

public void setTxtUserName(WebElement txtUserName) {
	this.txtUserName = txtUserName;
}

public void setTxtPassword(WebElement txtPassword) {
	this.txtPassword = txtPassword;
}

public LoginPage() {
	PageFactory.initElements(driver, this);

}

public WebElement getTxtUserName() {
	return txtUserName;
}

public WebElement getTxtPassword() {
	return txtPassword;
}

public void setTxtPassword(String txtPassword) {
	this.txtPassword.sendKeys(txtPassword);
}

public WebElement getBtnLogin() {
	return btnLogin;
}

public void setBtnLogin(WebElement btnLogin) {
	this.btnLogin = btnLogin;
}
}
```

Utility package:

Base class:

```
public class Base {
	public static WebDriver driver;
	static WebDriverWait wait;
	static File f1 = new File("./JSON/Configuration.json");

	public static WebDriver getDriver() {
		JSONObject jsonObject = JSONReadFromFile();
		String browser = (String) jsonObject.get("browser");

		File f = new File("./driver");
		if (browser.equals("chrome")) {
			System.setProperty("webdriver.chrome.driver",
f.getAbsolutePath()
					+ "/chromedriver.exe");
			driver = new ChromeDriver();

		} else if (browser.equals("firefox")) {
			System.setProperty("webdriver.gecko.driver",
f.getAbsolutePath()
					+ "/geckodriver.exe");
			driver = new FirefoxDriver();

		} else if (browser.equals("ie")) {
			System.setProperty("webdriver.ie.driver",
f.getAbsolutePath()
					+ "/IEDriverServer.exe");
			driver = new InternetExplorerDriver();

		}

		driver.manage().window().maximize();
		driver.get((String) jsonObject.get("url"));
		return driver;
	}

	public static boolean elementToBeVisible(WebDriver driver, int time,
			WebElement element) {
		boolean flag = false;
		try {
			wait = new WebDriverWait(driver, time);
			wait.until(ExpectedConditions.visibilityOf(element));
			flag = true;
		} catch (Exception e) {
			e.printStackTrace();
		}
		return flag;
```

```
	}

	public static boolean alertIsPresent(WebDriver driver, int time) {
		boolean flag = false;
		try {
			wait = new WebDriverWait(driver, time);
			wait.until(ExpectedConditions.alertIsPresent());
			flag = true;
		} catch (Exception e) {
			e.printStackTrace();
		}
		return flag;
	}

	public static boolean elementToBeClickable(WebDriver driver, int time,
			WebElement element) {
		boolean flag = false;
		try {
			wait = new WebDriverWait(driver, time);

	wait.until(ExpectedConditions.elementToBeClickable(element));
			flag = true;
		} catch (Exception e) {
			e.printStackTrace();
		}
		return flag;
	}

	public static boolean elementFound(WebDriver driver, int time,
			WebElement element) {
		boolean res = false;
		driver.manage().timeouts().implicitlyWait(time,
TimeUnit.SECONDS);
		try {

			res = element.isDisplayed();
		} catch (Exception e) {
			e.printStackTrace();

		}
		return res;

	}

	public static boolean elementFound(WebElement element) {
		boolean res = false;
		try {
			res = element.isDisplayed();
		} catch (Exception e) {
			e.printStackTrace();
```

```
		}
		return res;
	}

	public static void setText(WebElement element, String name) {
		if (name != null && elementFound(element)) {
			element.clear();
			element.sendKeys(name);
		}

	}

	public static String getText(WebElement element) {
		String name = null;
		if (elementFound(element)) {
			name = element.getAttribute("value");
		}
		return name;
	}

	public static void clickBtn(WebElement element) {
	if (elementFound(element)) {
			element.click();
		}
	}

	public static JSONObject JSONReadFromFile() {
		JSONParser parser = new JSONParser();
		JSONObject jsonObject = null;
		try {

			Object obj = parser.parse(new
FileReader(f1.getAbsoluteFile()));

			jsonObject = (JSONObject) obj;

		} catch (Exception e) {
			e.printStackTrace();
		}
		return jsonObject;
	}

	public static void getScreenShot(String screenShotFileName) {
		File screenShotLocation = new File("./screenshot/" +
screenShotFileName
				+ ".png");
		TakesScreenshot screenshot = (TakesScreenshot) driver;
		File file = screenshot.getScreenshotAs(OutputType.FILE);
		try {
```

```
                FileUtils.copyFile(file, screenShotLocation);
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }

        public static void uploadFiles(File path) {
            try {
                Robot robot = new Robot();
                robot.setAutoDelay(3000);
                StringSelection selection = new StringSelection(
                        path.getAbsolutePath());
                Toolkit.getDefaultToolkit().getSystemClipboard()
                        .setContents(selection, null);
                // press control+v
                robot.keyPress(KeyEvent.VK_CONTROL);
                robot.keyPress(KeyEvent.VK_V);
                robot.setAutoDelay(3000);
                // release control+v
                robot.keyRelease(KeyEvent.VK_CONTROL);
                robot.keyRelease(KeyEvent.VK_V);
                // press enter
                robot.setAutoDelay(3000);
                robot.keyPress(KeyEvent.VK_ENTER);
                robot.keyRelease(KeyEvent.VK_ENTER);

            } catch (AWTException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }

        public static List<HashMap<String, String>> readValueFromExcelSheet() {
            List<HashMap<String, String>> mapDatasList = new ArrayList();
            try {
                File excelLocaltion = new File("./Excel/Facebook.xlsx");

                String sheetName = "Face";

                FileInputStream f = new FileInputStream(
                        excelLocaltion.getAbsolutePath());
                Workbook w = new XSSFWorkbook(f);
                Sheet sheet = w.getSheet(sheetName);
                Row headerRow = sheet.getRow(0);
                for (int i = 0; i < sheet.getPhysicalNumberOfRows(); i++) {
                    Row currentRow = sheet.getRow(i);
                    HashMap<String, String> mapDatas = new
HashMap<String, String>();
```

```
                        for (int j = 0; j <
headerRow.getPhysicalNumberOfCells(); j++) {
                              Cell currentCell = currentRow.getCell(j);

                              switch (currentCell.getCellType()) {
                              case Cell.CELL_TYPE_STRING:

      mapDatas.put(headerRow.getCell(j).getStringCellValue(),

      currentCell.getStringCellValue());
                                    break;
                              case Cell.CELL_TYPE_NUMERIC:

      mapDatas.put(headerRow.getCell(j).getStringCellValue(),
                                                String.valueOf(currentCell

      .getNumericCellValue()));

                                    break;

                              }
                        }

                        mapDatasList.add(mapDatas);
                  }

            } catch (Throwable e) {
                  e.printStackTrace();
            }
            return mapDatasList;

      }

      public static List<Employee> retriveValueFromDataBase() {
            ResultSet rs = null;
            List<Employee> emp = new ArrayList<Employee>();
            try {
                  Class.forName("com.mysql.jdbc.Driver");
                  Connection con = DriverManager.getConnection(
                              "jdbc:mysql://127.0.0.1:3306/selenum_schema",
"root", "");
                  PreparedStatement ps = con
                              .prepareStatement("SELECT * FROM
selenium_schema.emp_table where roll=3;");

                  rs = ps.executeQuery();

                  while (rs.next()) {
                        Employee e = new Employee();
                        e.setName(rs.getString("name"));
```

```
                        e.setPassword(rs.getString("password"));
                        emp.add(e);

                  }
            } catch (ClassNotFoundException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }
            return emp;

      }

}
```

Test runner class:

```
@RunWith(Cucumber.class)
@CucumberOptions(features = "Features", glue = "steps")
public class TestRunner {

}
```

Feature file:

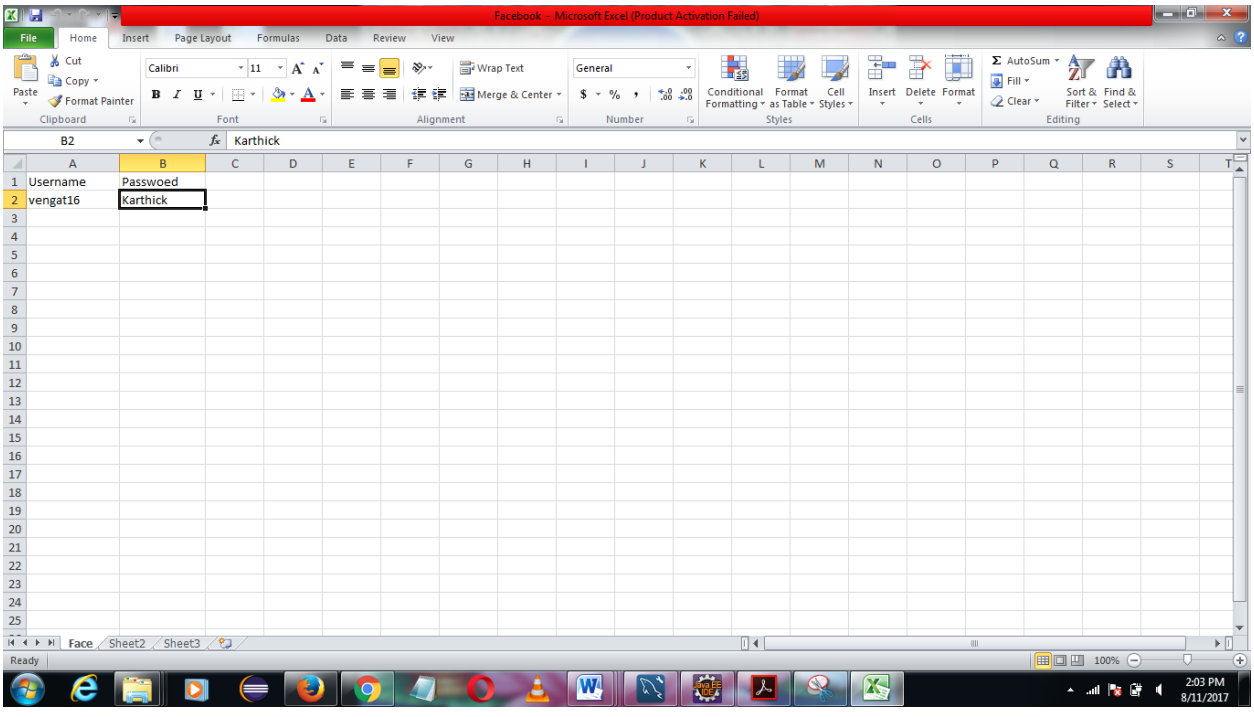
```
Scenario Outline: Successful Login with Valid Credentials
    Given User is on facebook Page
    When User enters "<userName>","<password>" and click the login  button
    Then Message displayed Login Successfully

    Examples:
      | userName | password |
      | ganesh   | Java65   |
```

Step definition file:

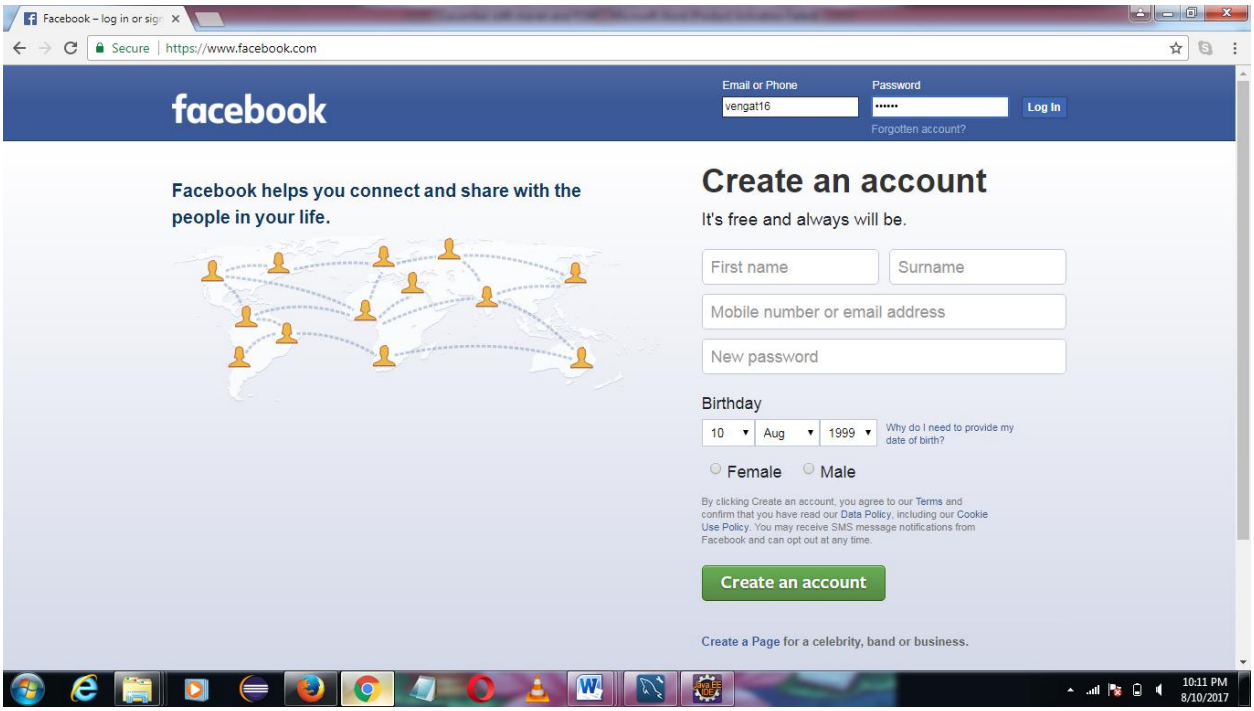
```
public class LoginSteps extends Base {
      WebDriver driver;
      LoginPage loginPage;

      @Given("^User is on facebook Page$")
      public void user_is_on_Home_Page() {
            driver = getDriver();

      }
```

```
	@When("^User enters \"([^\"]*)\",\"([^\"]*)\" and click the login
button$")
	public void user_enters_and_click_the_login_button(String userName,
			String Password) {
		loginPage = new LoginPage();


		setText(loginPage.getTxtUserName(),readValueFromExcelSheet().get(1)
				.get("Username"));

		setText(loginPage.getTxtPassword(),readValueFromExcelSheet().get(1)
				.get("Password"));

		clickBtn(loginPage.getBtnLogin());

	}

	@Then("^Message displayed Login Successfully$")
	public void message_displayed_Login_Successfully() {

		driver.quit();

	}
```

Excel input:



Output :


Facebook – log in or sign
Secure | https://www.facebook.com
facebook
Email or Phone
vengat16
Password
Log In
Forgotten account?
Facebook helps you connect and share with the people in your life.
Create an account
It's free and always will be.
First name
Surname
Mobile number or email address
New password
Birthday
10
Aug
1999
Why do I need to provide my date of birth?
Female
Male
By clicking Create an account, you agree to our Terms and confirm that you have read our Data Policy, including our Cookie Use Policy. You may receive SMS message notifications from Facebook and can opt out at any time.
Create an account
Create a Page for a celebrity, band or business.
10:11 PM
8/10/2017