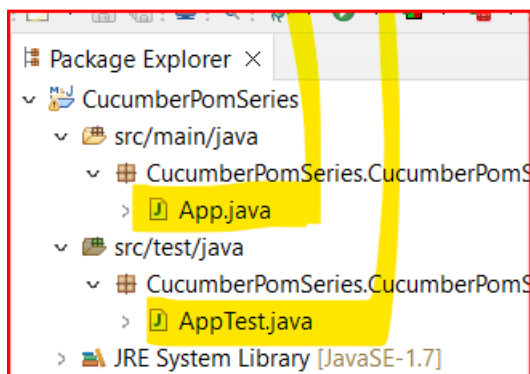


“Control+shift+f” is for formatting.

Delete the default files created-



First lets start with pom-

Remove all properties.

Replace with the ones shown below.

```

15
16 <properties>
17     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
18     <java.version>1.8</java.version>
19     <junit.version>4.13.1</junit.version>
20     <cucumber.version>6.9.0</cucumber.version>
21     <maven.compiler.version>3.8.1</maven.compiler.version>
22     <maven.surefire.version>2.22.2</maven.surefire.version>
23 </properties>

```

Remove existing dependencies and add the below ones-

```

<dependencies>
    <dependency>

```

```

        <groupId>io.cucumber</groupId>
        <artifactId>cucumber-java</artifactId>
        <version>${cucumber.version}</version>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>io.cucumber</groupId>
        <artifactId>cucumber-junit</artifactId>
        <version>${cucumber.version}</version>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>${junit.version}</version>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>3.141.59</version>
    </dependency>

    <dependency>
        <groupId>io.github.bonigarcia</groupId>
        <artifactId>webdrivermanager</artifactId>
        <version>4.2.2</version>
    </dependency>

    <dependency>
        <groupId>tech.grasshopper</groupId>
        <artifactId>extentreports-cucumber6-
adapter</artifactId>
        <version>2.8.0</version>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.testng</groupId>
        <artifactId>testng</artifactId>
        <version>6.14.3</version>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>io.cucumber</groupId>
        <artifactId>cucumber-testng</artifactId>
        <version>${cucumber.version}</version>
        <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.apache.poi/poi --
>
    <dependency>
        <groupId>org.apache.poi</groupId>
        <artifactId>poi</artifactId>
        <version>4.1.2</version>
    </dependency>


```


```































ooxml -->
    <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-
    <dependency>
        <groupId>org.apache.poi</groupId>
        <artifactId>poi-ooxml</artifactId>
        <version>4.1.2</version>
    </dependency>
</dependencies>

```

Pressing save will rebuild the project and dependencies get added to maven dependencies.

>  JRE System Library [JavaSE-1.7]

▼  Maven Dependencies

- >  animal-sniffer-annotations-1.14.jar - C:\Users\karan\.m2\repo
- >  animated-gif-lib-1.4.jar - C:\Users\karan\.m2\repository\com
- >  apiguardian-api-1.1.0.jar - C:\Users\karan\.m2\repository\org
- >  bsh-2.0b6.jar - C:\Users\karan\.m2\repository\org\apache-ex
- >  bson-3.6.4.jar - C:\Users\karan\.m2\repository\org\mongodb
- >  byte-buddy-1.8.15.jar - C:\Users\karan\.m2\repository\net\by
- >  checker-compat-qual-2.0.0.jar - C:\Users\karan\.m2\repositor
- >  commons-codec-1.13.jar - C:\Users\karan\.m2\repository\cor
- >  commons-collections4-4.4.jar - C:\Users\karan\.m2\repository
- >  commons-compress-1.19.jar - C:\Users\karan\.m2\repository\
- >  commons-exec-1.3.jar - C:\Users\karan\.m2\repository\org\aj
- >  commons-io-2.8.0.jar - C:\Users\karan\.m2\repository\comm
- >  commons-lang3-3.11.jar - C:\Users\karan\.m2\repository\org
- >  commons-logging-1.2.jar - C:\Users\karan\.m2\repository\co
- >  commons-math3-3.6.1.jar - C:\Users\karan\.m2\repository\or
- >  create-meta-2.0.4.jar - C:\Users\karan\.m2\repository\io\cucu
- >  cucumber-core-6.9.0.jar - C:\Users\karan\.m2\repository\io\ci
- >  cucumber-expressions-10.3.0.jar - C:\Users\karan\.m2\reposit
- >  cucumber-gherkin-6.9.0.jar - C:\Users\karan\.m2\repository\ic
- >  cucumber-gherkin-messages-6.9.0.jar - C:\Users\karan\.m2\re
- >  cucumber-java-6.9.0.jar - C:\Users\karan\.m2\repository\io\cu
- >  cucumber-junit-6.9.0.jar - C:\Users\karan\.m2\repository\io\ci
- >  cucumber-pdf-report-2.2.0.jar - C:\Users\karan\.m2\repositor
- >  cucumber-plugin-6.9.0.jar - C:\Users\karan\.m2\repository\io\
- >  cucumber-testng-6.9.0.jar - C:\Users\karan\.m2\repository\io\
- >  curvesapi-1.06.jar - C:\Users\karan\.m2\repository\com\githu
- >  datatable-3.5.0.jar - C:\Users\karan\.m2\repository\io\cucumi
- >  docstring-6.9.0.jar - C:\Users\karan\.m2\repository\io\cucumt
- >  error\_prone\_annotations-2.1.3.jar - C:\Users\karan\.m2\reposit
- >  extent-pdf-report-1.7.0.jar - C:\Users\karan\.m2\repository\te

similarly go to the plugin section

and remove everything.

Add the below ones. It is better to copy paste this stuff and remember what we need rather than typing.

```

<build>
    <plugins>
        <plugin>

```

```

        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>${maven.compiler.version}</version>
        <configuration>
            <encoding>UTF-8</encoding>
            <source>${java.version}</source>
            <target>${java.version}</target>
        </configuration>
    </plugin>
    <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>${maven.surefire.version}</version>
        <configuration>
            <includes>

<include>**/ParallelRun.java</include>
            </includes>
            <parallel>methods</parallel>
            <threadCount>4</threadCount>

            <useUnlimitedThreads>false</useUnlimitedThreads>
        </configuration>
    </plugin>

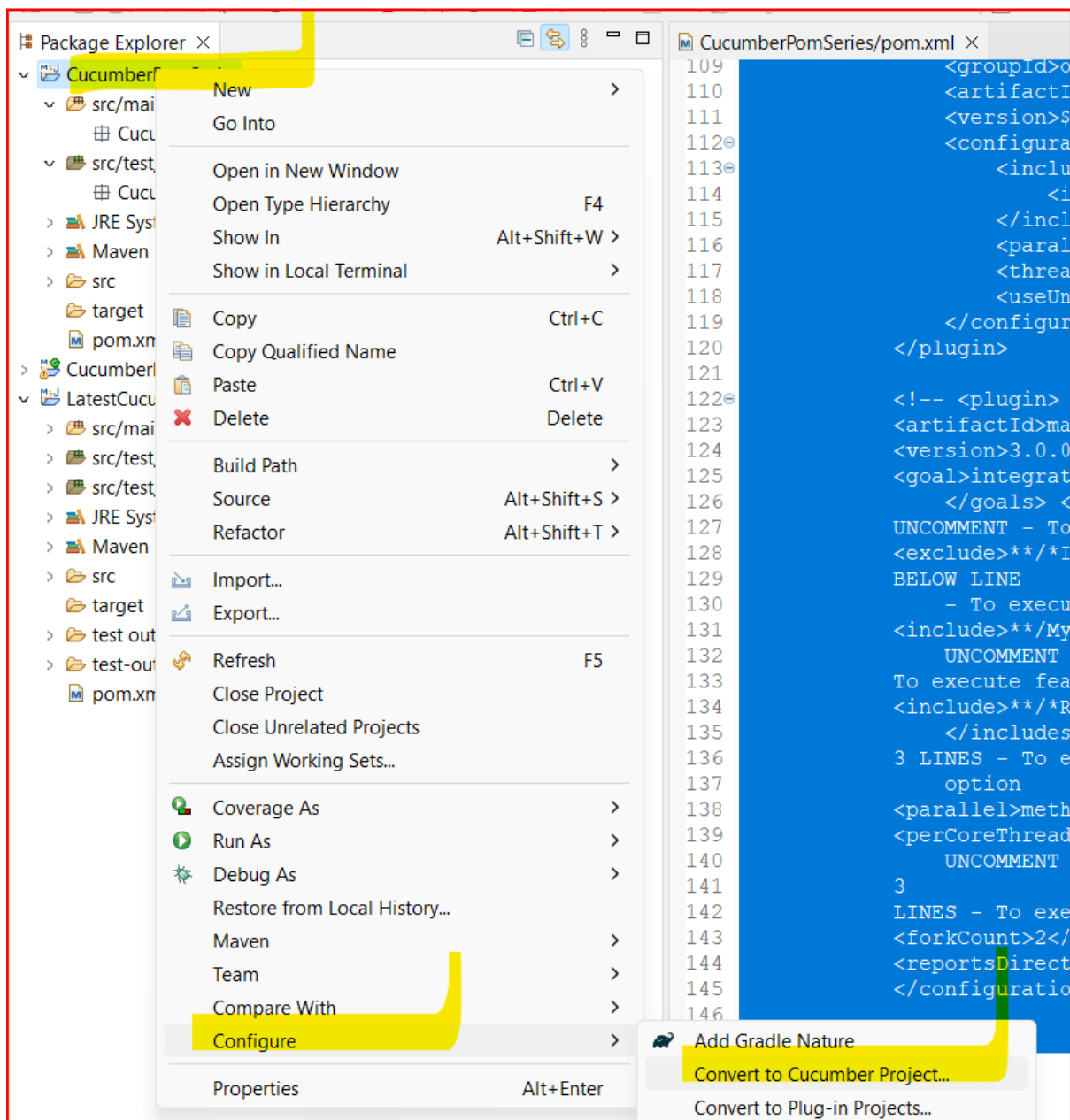
    <!-- <plugin>
<groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-failsafe-plugin</artifactId>
    <version>3.0.0-M3</version> <executions> <execution>
<goals>
    <goal>integration-test</goal>
    </goals> <configuration>
    UNCOMMENT - To add any exclusions if required
<excludes>
    <exclude>**/*IT*.java</exclude> </excludes> <includes>
UNCOMMENT
    BELOW LINE
    - To execute feature files with a single runner
    <include>**/MyTestRunner.java</include>
    UNCOMMENT BELOW LINE -
    To execute feature files with multiple runners
    <include>**/*Runner.java</include>
    </includes> UNCOMMENT BELOW
    3 LINES - To execute using parallel or combination
    option
    <parallel>methods</parallel>
<threadCount>4</threadCount>
    <perCoreThreadCount>true</perCoreThreadCount>
    UNCOMMENT BELOW
    3
    LINES - To execute using forking or combination option
    <forkCount>2</forkCount> <reuseForks>true</reuseForks>
    <reportsDirectory>${project.build.directory}/failsafe-
reports_${surefire.forkNumber}</reportsDirectory>
    </configuration> </execution> </executions> </plugin> -
->

    </plugins>
</build>

```

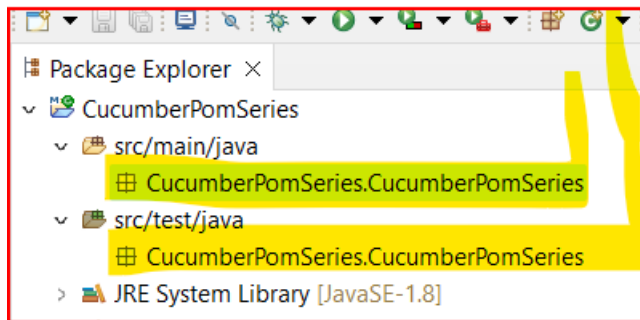
Once pom done, force update all the maven project.

Right click on project and convert to cucumber project.

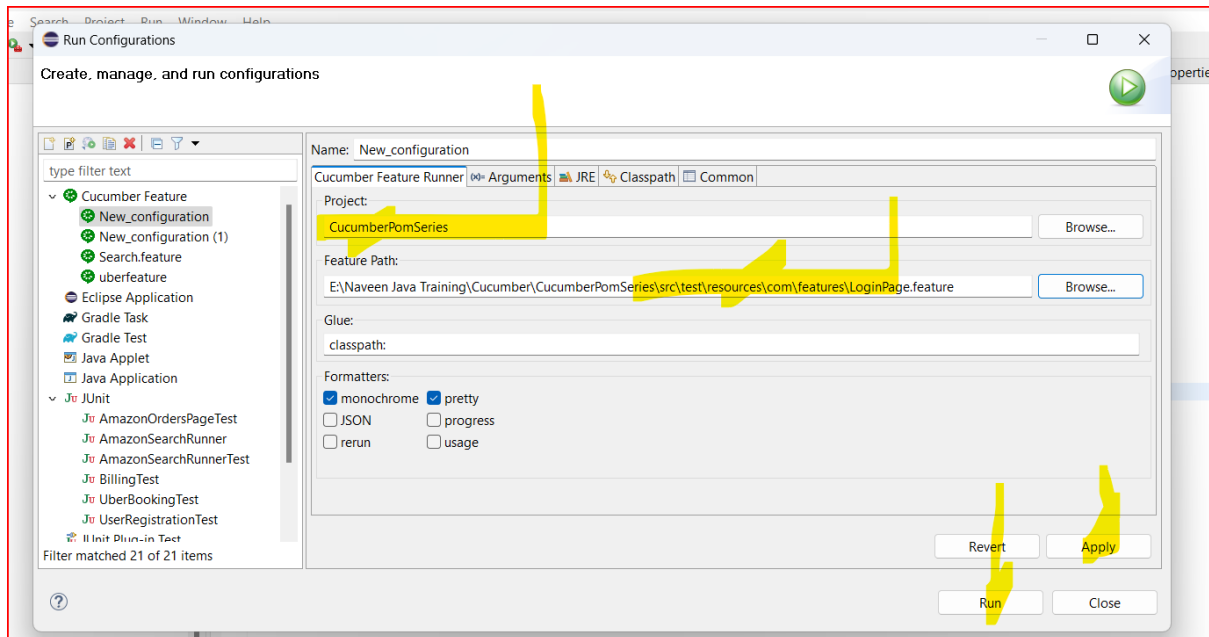


You can see “M and C” icon on project which means it’s a maven-cucumber project. This will come after you convert the project to cucumber project.

Even delete the default package which got created-

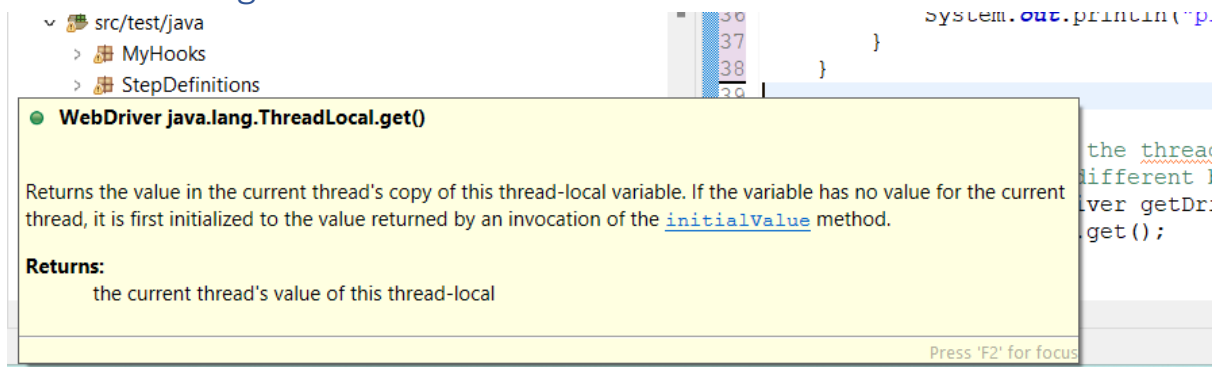


When running the new project, you need to select the project also-



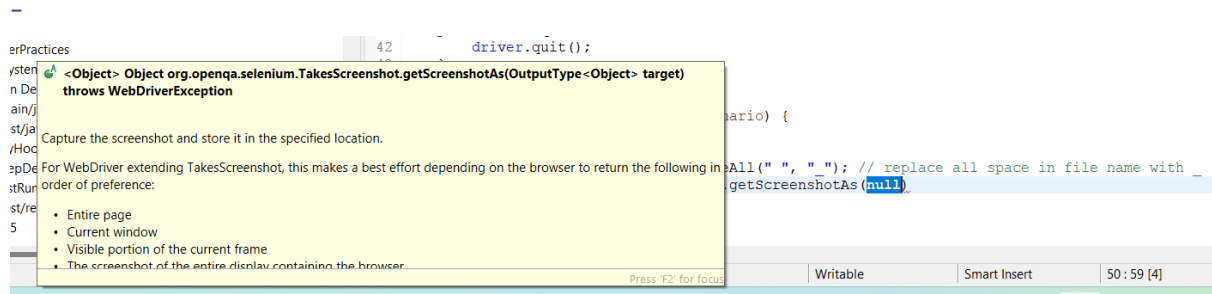
“Control+shift+O” for importing everything on page.

“Threadlocal.get” method returns webdriver-

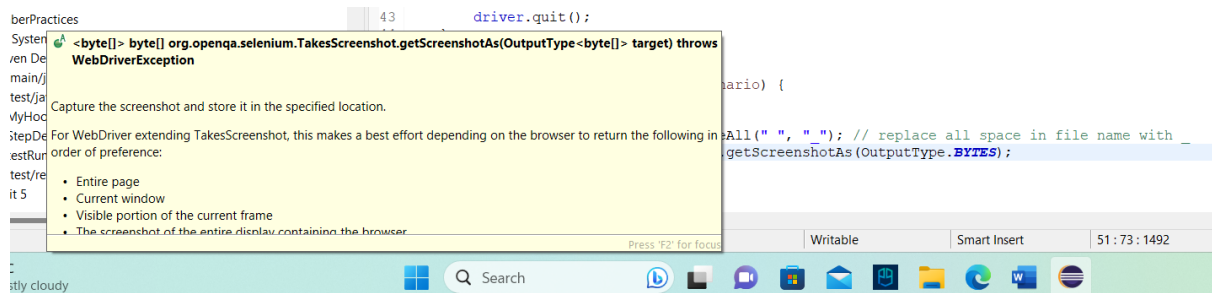




## See how to check what parameter to pass in takes screenshot method

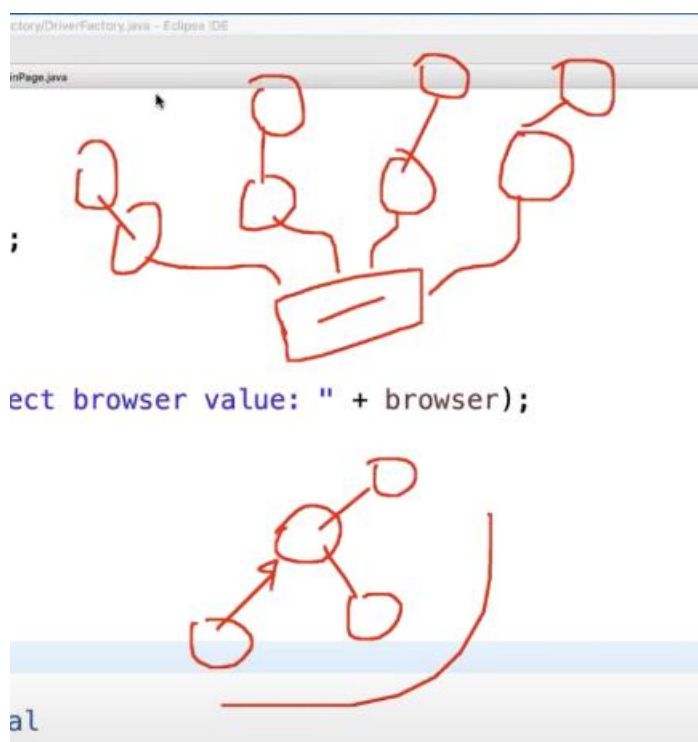


## See the output type of getscreenshot-



According to pom, we have to create separate pages for every page class.

See how thread local works-



The first picture is parallel execution using thread local. Same chrome browser being accessed by multiple threads at the same time. First test case attacked by first google chrome in first thread, second test case attacked by second google chrome in second thread and so on.

The second picture shows without thread local, at a time the browser can be accessed by only one thread. Other threads will have to wait.

“Threadlocal.get” will get the driver details at run time.

Since we are not using the concept of testing, threadlocal concept needed for parallel execution.

## Note on how to rename the file-

Click on the file name on project explorer.

Press F2.

Then rename.

## First big code as part of hybrid framework-

```
Driver factory-
package com.qa.factory;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.safari.SafariDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

//this class is for initialising the driver and returning it.
//this class will also help in parallel runs because we will use the
concept of threadlocal.
//thread local is java 8 feature.
public class DriverFactory {

    public WebDriver driver;
    // we want to initialize the webdriver with help of threadlocal so
we pass
    // WebDriver as generics.
    public static ThreadLocal<WebDriver> tlDriver = new
ThreadLocal<>();

    // pass the browser and webdriver will open that browser.
    // this method should also return webdriver.
    /**
     * this method is used to initialize the threadlocal driver on the
basis of
     * given browser.
     *
     * @param browser
     * @return this will return tlDriver.
     */
    public WebDriver init_driver(String browser) {
        System.out.println("browser value is" + browser);
        if (browser.equals("chrome")) {
            WebDriverManager.chromedriver().setup();
            // tl will give set and get method.
            // here chrome driver object is created and it is set
with thread local driver.
            // local copy of chrome driver object is created and
assigned to one thread.
            tlDriver.set(new ChromeDriver());
```

```

        } else if (browser.equals("firefox")) {
            WebDriverManager.firefoxdriver().setup();
            tldriver.set(new FirefoxDriver());
        } else if (browser.equals("safari")) {
            // for safari there is no binary so no web driver
manager.
            tldriver.set(new SafariDriver());
        } else {
            System.out.println("please pass the correct browser" +
browser);
        }
        getDriver().manage().deleteAllCookies();
        getDriver().manage().window().maximize();
        return getDriver();
    }

//get the driver.
// this will return the threadlocal instance.
// threadlocal has different browsers that driver will invoke.
// we are writing synchronized because multiple threads will call
this method at
// same time.
/**
 * this is used to get the driver with thread local.
 *
 * @return
 */
public static synchronized WebDriver getDriver() {
    return tldriver.get();
}
}

```

Config.properties-

browser = chrome

Config reader-

```

package com.qa.util;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Properties;

public class ConfigReader {

    private Properties prop;

    /**
     * this method is used to load the properties from
config.properties file.
     * @return it returns properties prop object.
     */

    public Properties initProp() {
        prop=new Properties(); // create object of properties class
        //we have to interact with config.properties which is file,
so we use file input stream class.
        try {
            FileInputStream ip=new
FileInputStream("./src/test/resources/config/config.properties");//give
path of config.properties.
            prop.load(ip); //load the properties for using them.
        } catch (FileNotFoundException e) {

```

```

        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return prop;
}
}

```

Hooks-

```
package appHooks;
```

```
import java.util.Properties;
```

```
import org.openqa.selenium.OutputType;
```

```
import org.openqa.selenium.TakesScreenshot;
```

```
import org.openqa.selenium.WebDriver;
```

```
import com.qa.factory.DriverFactory;
```

```
import com.qa.util.ConfigReader;
```

```
import io.cucumber.java.After;
```

```
import io.cucumber.java.Before;
```

```
import io.cucumber.java.Scenario;
```

```
public class ApplicationHooks {
```

```
    private DriverFactory driverFactory;
```

```
    private WebDriver driver;
```

```
    private ConfigReader configReader;
```

```
    private Properties prop;
```

```
    private String browserName;
```

```
    @Before(order=0)
```

```
        public void getProperty() {
```

```
            configReader=new ConfigReader();
```

```
            prop=configReader.initProp();
```

```
        }
```

```
    @Before(order=1)
```

```
    public void launchBrowser() {
```

```
        browserName=prop.getProperty("browser");
```

```
        driverFactory=new DriverFactory();
```

```
        driver=driverFactory.init_driver(browserName); // we need to assign driver here,
```

```
        else driver.quit will
```

```
        //give null pointer exception and the webdriver driver declared in step 16 is not
        pointing anywhere.
```

```
    }
```

```
    //note - in after the order works in reverse direction.
```

```
    //order = 2 is executed before order = 1 and order =0.
```

```
    @After(order=0)
```

```
    public void quitBrowser() {
```

```

        driver.quit();
    }

    @After(order=1)
    public void tearDown(Scenario scenario) {
        if(scenario.isFailed()) {
            //take screenshot
            String screenshotName=scenario.getName().replaceAll(" ", "_"); // replace
all space in file name with _
            byte []
sourcePath=((TakesScreenshot)driver).getScreenshotAs(OutputType.BYTES);// we keep the output
type as

            //bytes or base64 so that it is compatible with jenkins/ci/cd tool
scenario.attach(sourcePath, "image/png", screenshotName);
        }
    }
}

```

```

Login page steps-
package stepDefinitions;

import com.pages.LoginPageObject;
import com.qa.factory.DriverFactory;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import static org.junit.Assert.*;

import org.junit.Assert;

@SuppressWarnings("deprecation")
public class LoginPageSteps {

    private static String title;
    private LoginPageObject loginPageObject = new
LoginPageObject(DriverFactory.getDriver());

    @Given("user is on login page")
    public void user_is_on_login_page() {

DriverFactory.getDriver().get("http://www.automationpractice.pl/index.php?controller=authentica
tion&back=my-account");
        //getdriver from driver factory returns web driver and with this web driver we can use
all the
        //browser controls and actions
    }
}

```

```

    @When("user gets the title of the page")
    public void user_gets_the_title_of_the_page() throws InterruptedException {
        Thread.sleep(10000);
        title=loginPageObject.getLoginPageTitle();
        System.out.println("page title is " + title);
    }

    @Then("page title should be {string}")
    public void page_title_should_be(String expectedTitleName) {
        Assert.assertTrue(title.contains(expectedTitleName)); // assert has to come from
junit and not testng else we get
        //deprecated error.
    }

    @Then("forgot your password link should be displayed")
    public void forgot_your_password_link_should_be_displayed() {
        Assert.assertTrue(loginPageObject.isForgotPasswordLinkPresent());
    }

    @When("user enters username as {string}")
    public void user_enters_username_as(String userName) {
        loginPageObject.enterUserName(userName);
    }

    @When("user enters password as {string}")
    public void user_enters_password_as(String password) {
        loginPageObject.enterPassword(password);
    }

    @When("user clicks on login button")
    public void user_clicks_on_login_button() {
        loginPageObject.clickOnSignInButton();
    }

    //we always write assertions in then
    //when is for test actions
    //given is pre condition
}

```

Login page object-

```

package com.pages;

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

public class LoginPageObject {

    // every page object will have by locators, constructor and page
actions.

    private WebDriver driver; // every class will have this webdriver.

    // 1. by locators.
    //by locators are also known as object repositories.

```

```

private By emailID = By.id("email");
private By password = By.id("passwd");
private By signInButton = By.id("SubmitLogin");
private By forgotPasswordLink = By.linkText("Forgot your
password?");

// 2. constructor of the page class
public LoginPageObject(WebDriver driver) {
    this.driver = driver;
}

// page classes should not have assertion.
// assertion should be written in test class or step def class.

// 3. page actions: features (behaviour) of the page in the form of
methods.

public String getLoginPageTitle() {
    return driver.getTitle();
}

public boolean isForgotPasswordLinkPresent() {
    return driver.findElement(forgotPasswordLink).isDisplayed();
}

public void enterUserName(String userName) {
    driver.findElement(emailID).sendKeys(userName);
}

public void enterPassword(String pwd) {
    driver.findElement(password).sendKeys(pwd);
}

public void clickOnSignInButton() {
    driver.findElement(signInButton).click();
}

// in step def do not maintain by locators and page methods. its
ugly
// programming.
// in page object, selenium code should be written in page class.
}

```

```

Test runner-
package com.myTestRunner;

import org.junit.runner.RunWith;

import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)
@CucumberOptions(
    features = { "src/test/resources/appFeatures" },
    glue = { "stepDefinitions", "appHooks" },
    plugin = { "pretty" }
)

```

```
public class MyTestRunner {
}
```

#### Login feature-

**Feature:** login page feature

**Scenario:** login page title

**Given** user is on login page

**When** user gets the title of the page

**Then** page title should be "Login - My Shop"

**Scenario:** forgot password link

**Given** user is on login page

**Then** forgot your password link should be displayed

**Scenario:** login with correct credentials

**Given** user is on login page

**When** user enters username as "cucumbertesting1960@gmail.com"

**And** user enters password as "Malaravi@123"

**And** user clicks on login button

**Then** user gets the title of the page

**And** page title should be "My account - My Shop"

Run as junit and check console-

```
Scenario: login page title                                # src/test/
browser value ischrome
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerF
SLF4J: Defaulting to no-operation (NOP) logger implementa
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBi
Starting ChromeDriver 114.0.5735.90 (386bc09e8f4f2e025edc
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-con
ChromeDriver was started successfully.
Jun 29, 2023 7:49:17 AM org.openqa.selenium.remote.Protoc
INFO: Detected dialect: W3C
    Given user is on login page                            # stepDefir
page title is Login - My Shop
    When user gets the title of the page                    # stepDefir
    Then page title should be "Login - My Shop"            # stepDefir

Scenario: forgot password link                            # sr
browser value ischrome
Starting ChromeDriver 114.0.5735.90 (386bc09e8f4f2e025edc
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-con
ChromeDriver was started successfully.
Jun 29, 2023 7:49:31 AM org.openqa.selenium.remote.Protoc
INFO: Detected dialect: W3C
    Given user is on login page                            # st
    Then forgot your password link should be displayed    # st
```



```

Scenario: login with correct credentials # S
browser value ischrome
Starting ChromeDriver 114.0.5735.90 (386bc09e8f4f2e025eddae123f36f
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-consideratio
ChromeDriver was started successfully.
Jun 29, 2023 7:49:34 AM org.openqa.selenium.remote.ProtocolHandsha
INFO: Detected dialect: W3C
    Given user is on login page # S
    When user enters username as "cucumbertesting1960@gmail.com" # S
    And user enters password as "Malaravi@123" # S
    And user clicks on login button # S
page title is My account - My Shop
    Then user gets the title of the page # S
    And page title should be "My account - My Shop" # S

```

## Lets create ui report-

Created a properties file for reports.  
 Cucumber.properties in src/test/resources.  
 cucumber.publish.enabled=true

There should not be any space in between or after true else we get error as "cannot parse the property cucumber.enabled=true."

So remove all spaces and update maven project and then run and you should be good.

Report link is opened.

The screenshot displays the Cucumber Reports web interface in a browser. The URL is <https://reports.cucumber.io/reports/fc5a5695-7c5f-402f-aa27-681bebddeee9>. The interface shows a summary of the test run with the following details:

- 100% passed** (3 executed)
- a minute ago** (Last Run)
- 32.69 seconds** (Duration)
- Environment: Windows 10
- Configuration: Java HotSpot(TM) 64-Bit Server VM 25.202-b08
- Version: cucumber-jvm 6.9.0

A warning message at the top states: "This report will self-destruct in a day. Keep your future reports forever." Below the summary, there is a "Delete Report" button. The feature being tested is "login page feature". The interface also includes a "Log in with GitHub" button and a cookie consent banner at the bottom.

Lets fail a scenario and then see if screenshot is taken-

```

5
6 public class LoginPageObject {
7
8     // every page object will have by locators, constructor and page actions.
9
10    private WebDriver driver; // every class will have this webdriver.
11
12    // 1. by locators.
13    //by locators are also known as object repositories.
14
15    private By emailID = By.id("email");
16    private By password = By.id("passwd");
17    private By signInButton = By.id("SubmitLogin");
18    private By forgotPasswordLink = By.linkText("Forgot your password?1111");
19

```

Console-

```

Scenario: forgot password link # src/test/resources/appFeatures/LoginPage.feature:8
browser value ischrome
Starting ChromeDriver 114.0.5735.90 (386bc09e8f4f2e025eddae123f36f6263096ae49--refs/branch-heads/5735@{#1052}) on port 9722
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jun 29, 2023 8:04:42 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Given user is on login page # stepDefinitions.LoginPageSteps.user_is_on_login_page()
Then forgot your password link should be displayed # stepDefinitions.LoginPageSteps.forgot_your_password_link_should_be_displayed()
31morg.openqa.selenium.NoSuchElementException: no such element: Unable to locate element: {"method":"link text","selector":"Forgot
your password?1111"}
(Session info: chrome=114.0.5735.199)
For documentation on this error, please visit: https://www.seleniumhq.org/exceptions/no_such_element.html
Build info: version: '3.141.59', revision: 'e82be7d358', time: '2018-11-14T08:17:03'
System info: host: 'LAPTOP-H56OVJTV', ip: '192.168.0.16', os.name: 'Windows 10', os.arch: 'amd64', os.version: '10.0', java.version:
'1.8.0_202'
Driver info: org.openqa.selenium.chrome.ChromeDriver
Capabilities {acceptInsecureCerts: false, browserName: chrome, browserVersion: 114.0.5735.199, chrome: {chromedriverVersion: 114.0.5735.90
(386bc09e8f4f..., userDataDir: C:\Users\karan\AppData\Loca...), goog:chromeOptions: {debuggerAddress: localhost:57441}, javascriptEnabled:
true, networkConnectionEnabled: false, pageLoadStrategy: normal, platform: WINDOWS, platformName: WINDOWS, proxy: Proxy(), setWindowRect:
true, strictFileInteractability: false, timeouts: {implicit: 0, pageLoad: 300000, script: 30000}, unhandledPromptBehavior: dismiss and
notify, webauthn:extension:credBlob: true, webauthn:extension:largeBlob: true, webauthn:extension:minPinLength: true,
webauthn:extension:prf: true, webauthn:virtualAuthenticators: true}
Session ID: a34alf518b8c5e0ead7c0fa3a7f7205f
*** Element info: {Using=link text, value=Forgot your password?1111}
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)




```

Report-

https://reports.cucumber.io/reports/c0090768-82df-465d-b551-30a187d5354b

Cucumber Reports [Log in with GitHub](#)

⚠ This report will self-destruct in a day. Keep your future reports forever.

2 PASSED		1 FAILED
<b>67% passed</b> 3 executed	<b>2 minutes ago</b> Last Run	<b>33.11 seconds</b> Duration
 Windows 10	 Java HotSpot(TM) 64-Bit Server VM 25.202-b08	 cucumber-jvm 6.9.0

[Delete Report](#)

file:///E:/Naveen%20Java%20Training/Cucumber/CucumberPomSeries/src/test/resources/appFeatures/LoginPage.feature

Report Edit

**Feature:** login page feature

**Scenario:** login page title

- Given user is on login page
- When user gets the title of the page
- Then page title should be "Login - My Shop"

**Scenario:** forgot password link

- Given user is on login page
- Then forgot your password link should be displayed

```
org.openqa.selenium.NoSuchElementException: no such element: Unable to locate element: ("method":"link text","selector":"Forgot your password?1111")
(Session info: chrome=114.0.5735.199)
For documentation on this error, please visit: https://www.seleniumhq.org/exceptions/no_such_element.html
Build info: version: '3.141.59', revision: 'e82be7d358', time: '2018-11-14T08:17:03'
System info: host: 'LAPTOP-H56OVJTV', ip: '192.168.0.16', os.name: 'Windows 10', os.arch: 'amd64', os.version: '10.0', java.version: '1.8.0_202'
Driver info: org.openqa.selenium.chrome.ChromeDriver
Capabilities {acceptInsecureCerts: false, browserName: chrome, browserVersion: 114.0.5735.199, chrome: {chromedriverVersion: 114.0.5735.90 (386c09e8f4f..., userDataDir: C:\Users\karan\AppData\Loca..., goog:chromeOptions: {debuggerAddress: localhost:57441}}, javascriptEnabled: true, networkConnectionEnabled: false, pageLoadStrategy: normal, platform: WINDOWS, platformName: WINDOWS, proxy: Proxy(), setWindowRect: true, strictFileInteractability: false, timeouts: {implicit: 0, pageLoad: 300000, script: 300000}, unhandledPromptBehavior: dismiss and notify, webauthn:extension:credBlob: true, webauthn:extension:largeBlob: true,
```

ng to browse, you consent to our use of cookies. To know more, please refer to our Privacy Policy. OK

cloudy Search ENG IN 08:07 29-06-2023

Screenshot also taken-

Attached Image

▼ Attached Image

Call us now 0123-456-789

SALE 70% OFF ALL PRODUCTS GET SAVINGS NOW

Contact us Sign in

Your Logo a new experience

Search

Cart (empty)

WOMEN DRESSES T-SHIRTS BLOG

Authentication

AUTHENTICATION

CREATE AN ACCOUNT

Please enter your email address to create an account.

Email address

Create an account

ALREADY REGISTERED?

Email address

Password

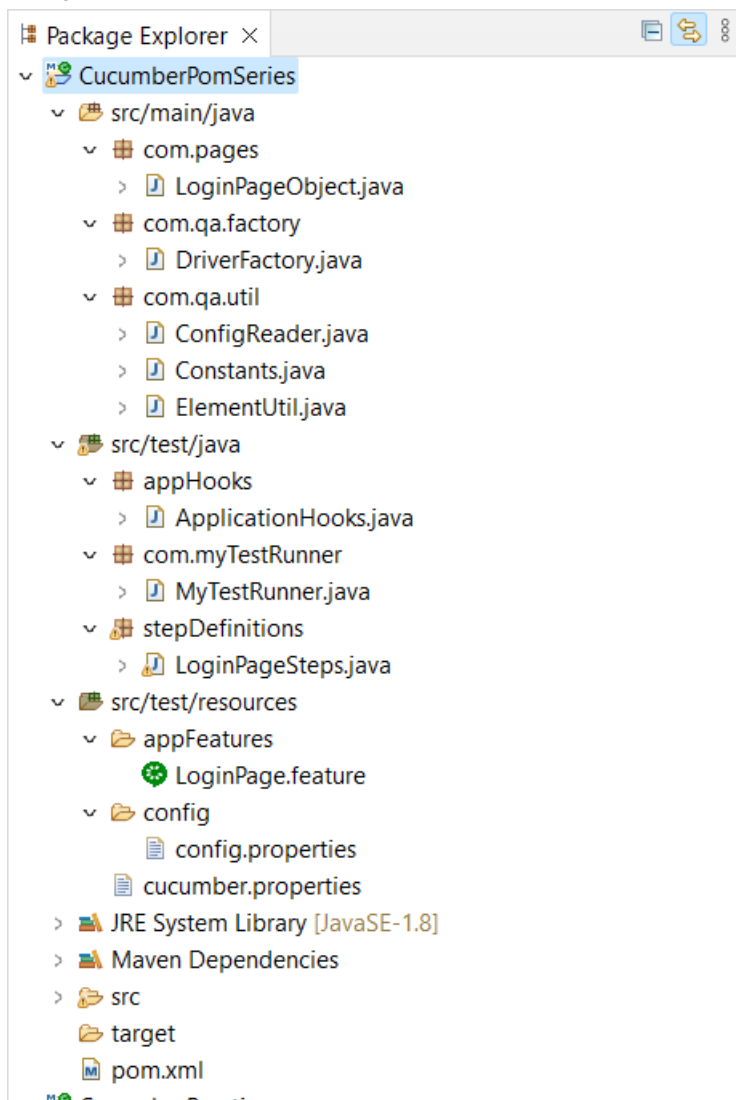
Forgot your password?

Scenario: login with correct credentials

- Given user is on login page

ng to browse, you consent to our use of cookies. To know more, please refer to our Privacy Policy. OK

## Project structure-



## Codes used-

### Driver factory-

```
package com.qa.factory;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.safari.SafariDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

//this class is for initialising the driver and returning it.
//this class will also help in parallel runs because we will use the
concept of threadlocal.
//thread local is java 8 feature.
public class DriverFactory {

    public WebDriver driver;
    // we want to initialize the webdriver with help of threadlocal so
we pass
    // WebDriver as generics.
```

```

    public static ThreadLocal<WebDriver> tldriver = new
ThreadLocal<>();

    // pass the browser and webdriver will open that browser.
    // this method should also return webdriver.
    /**
     * this method is used to initialize the threadlocal driver on the
basis of
     * given browser.
     *
     * @param browser
     * @return this will return tldriver.
     */
    public WebDriver init_driver(String browser) {
        System.out.println("browser value is" + browser);
        if (browser.equals("chrome")) {
            WebDriverManager.chromedriver().setup();
            // tl will give set and get method.
            // here chrome driver object is created and it is set
with thread local driver.
            // local copy of chrome driver object is created and
assigned to one thread.
            tldriver.set(new ChromeDriver());
        } else if (browser.equals("firefox")) {
            WebDriverManager.firefoxdriver().setup();
            tldriver.set(new FirefoxDriver());
        } else if (browser.equals("safari")) {
            // for safari there is no binary so no web driver
manager.

            tldriver.set(new SafariDriver());
        } else {
            System.out.println("please pass the correct browser" +
browser);
        }
        getDriver().manage().deleteAllCookies();
        getDriver().manage().window().maximize();
        return getDriver();
    }

    //get the driver.
    // this will return the threadlocal instance.
    // threadlocal has different browsers that driver will invoke.
    // we are writing synchronized because multiple threads will call
this method at
    // same time.
    /**
     * this is used to get the driver with thread local.
     *
     * @return
     */
    public static synchronized WebDriver getDriver() {
        return tldriver.get();
    }
}

```

Config properties-

browser = chrome

Config reader-

**package** com.qa.util;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

```

import java.io.IOException;
import java.util.Properties;

public class ConfigReader {

    private Properties prop;

    /**
     * this method is used to load the properties from
    config.properties file.
     * @return it returns properties prop object.
     */

    public Properties initProp() {
        prop=new Properties(); // create object of properties class
        //we have to interact with config.properties which is file,
        so we use file input stream class.
        try {
            FileInputStream ip=new
FileInputStream("./src/test/resources/config/config.properties");//give
path of config.properties.
            prop.load(ip); //load the properties for using them.
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return prop;
    }
}

```

Hooks-

```
package appHooks;
```

```
import java.util.Properties;
```

```
import org.openqa.selenium.OutputType;
```

```
import org.openqa.selenium.TakesScreenshot;
```

```
import org.openqa.selenium.WebDriver;
```

```
import com.qa.factory.DriverFactory;
```

```
import com.qa.util.ConfigReader;
```

```
import io.cucumber.java.After;
```

```
import io.cucumber.java.Before;
```

```
import io.cucumber.java.Scenario;
```

```
public class ApplicationHooks {
```

```
    private DriverFactory driverFactory;
```

```
    private WebDriver driver;
```

```
    private ConfigReader configReader;
```

```
    private Properties prop;
```

```
    private String browserName;
```

```
    @Before(order=0)
```

```

        public void getProperty() {
            configReader=new ConfigReader();
            prop=configReader.initProp();
        }

    @Before(order=1)
    public void launchBrowser() {
        browserName=prop.getProperty("browser");
        driverFactory=new DriverFactory();
        driver=driverFactory.init_driver(browserName); // we need to assign driver here,
else driver.quit will
        //give null pointer exception and the webdriver driver declared in step 16 is not
pointing anywhere.
    }

    //note - in after the order works in reverse direction.
    //order = 2 is executed before order = 1 and order =0.

    @After(order=0)
    public void quitBrowser() {
        driver.quit();
    }

    @After(order=1)
    public void tearDown(Scenario scenario) {
        if(scenario.isFailed()) {
            //take screenshot
            String screenshotName=scenario.getName().replaceAll(" ", "_"); // replace
all space in file name with _
            byte []
sourcePath=((TakesScreenshot)driver).getScreenshotAs(OutputType.BYTES);// we keep the output
type as

            //bytes or base64 so that it is compatible with jenkins/ci/cd tool
scenario.attach(sourcePath, "image/png", screenshotName);
        }
    }
}

```

Login page steps-

package stepDefinitions;

import com.pages.LoginPageObject;

import com.qa.factory.DriverFactory;

import io.cucumber.java.en.Given;

import io.cucumber.java.en.Then;

import io.cucumber.java.en.When;

import static org.junit.Assert.\*;

```

import org.junit.Assert;

@SuppressWarnings("deprecation")
public class LoginPageSteps {

    private static String title;
    private LoginPageObject loginPageObject = new
LoginPageObject(DriverFactory.getDriver());

    @Given("user is on login page")
    public void user_is_on_login_page() {

DriverFactory.getDriver().get("http://www.automationpractice.pl/index.php?controller=authentica
tion&back=my-account");
        //getdriver from driver factory returns web driver and with this web driver we can use
all the
        //browser controls and actions
    }

    @When("user gets the title of the page")
    public void user_gets_the_title_of_the_page() throws InterruptedException {
        Thread.sleep(10000);
        title=loginPageObject.getLoginPageTitle();
        System.out.println("page title is " + title);
    }

    @Then("page title should be {string}")
    public void page_title_should_be(String expectedTitleName) {
        Assert.assertTrue(title.contains(expectedTitleName)); // assert has to come from
junit and not testng else we get
        //deprecated error.
    }

    @Then("forgot your password link should be displayed")
    public void forgot_your_password_link_should_be_displayed() {
        Assert.assertTrue(loginPageObject.isForgotPasswordLinkPresent());
    }

    @When("user enters username as {string}")
    public void user_enters_username_as(String userName) {
        loginPageObject.enterUserName(userName);
    }

    @When("user enters password as {string}")
    public void user_enters_password_as(String password) {
        loginPageObject.enterPassword(password);
    }

    @When("user clicks on login button")
    public void user_clicks_on_login_button() {

```



```

        loginPageObject.clickOnSignInButton();
    }

    //we always write assertions in then
    //when is for test actions
    //given is pre condition
}

Login page object-
package com.pages;

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

public class LoginPageObject {

    // every page object will have by locators, constructor and page
    actions.

    private WebDriver driver; // every class will have this webdriver.

    // 1. by locators.
    //by locators are also known as object repositories.

    private By emailID = By.id("email");
    private By password = By.id("passwd");
    private By signInButton = By.id("SubmitLogin");
    private By forgotPasswordLink = By.linkText("Forgot your
password?1111");

    // 2.constructor of the page class
    public LoginPageObject(WebDriver driver) {
        this.driver = driver;
    }

    // page classes should not have assertion.
    // assertion should be written in test class or step def class.

    // 3. page actions: features (behaviour) of the page in the form of
    methods.

    public String getLoginPageTitle() {
        return driver.getTitle();
    }

    public boolean isForgotPasswordLinkPresent() {
        return driver.findElement(forgotPasswordLink).isDisplayed();
    }

    public void enterUserName(String userName) {
        driver.findElement(emailID).sendKeys(userName);
    }

    public void enterPassword(String pwd) {
        driver.findElement(password).sendKeys(pwd);
    }

    public void clickOnSignInButton() {
        driver.findElement(signInButton).click();
    }
}

```

```

        // in step def do not maintain by locators and page methods. its
ugly
        // programming.
        // in page object, selenium code should be written in page class.
    }

```

Test runner-

```
package com.myTestRunner;
```

```
import org.junit.runner.RunWith;
```

```
import io.cucumber.junit.Cucumber;
```

```
import io.cucumber.junit.CucumberOptions;
```

```
@RunWith(Cucumber.class)
```

```
@CucumberOptions(
```

```
    features = { "src/test/resources/appFeatures" },
```

```
    glue = { "stepDefinitions", "appHooks" },
```

```
    plugin = { "pretty" }

```

```
)
```

```
public class MyTestRunner {
```

```
}
```

Login page feature-

```
Feature: login page feature
```

```
Scenario: login page title
```

```
Given user is on login page
```

```
When user gets the title of the page
```

```
Then page title should be "Login - My Shop"
```

```
Scenario: forgot password link
```

```
Given user is on login page
```

```
Then forgot your password link should be displayed
```

```
Scenario: login with correct credentials
```

```
Given user is on login page
```

```
When user enters username as "cucumbertesting1960@gmail.com"
```

```
And user enters password as "Malaravi@123"
```

```
And user clicks on login button
```

```
Then user gets the title of the page
```

```
And page title should be "My account - My Shop"
```

Cucumber properties-

```
cucumber.publish.enabled=true
```