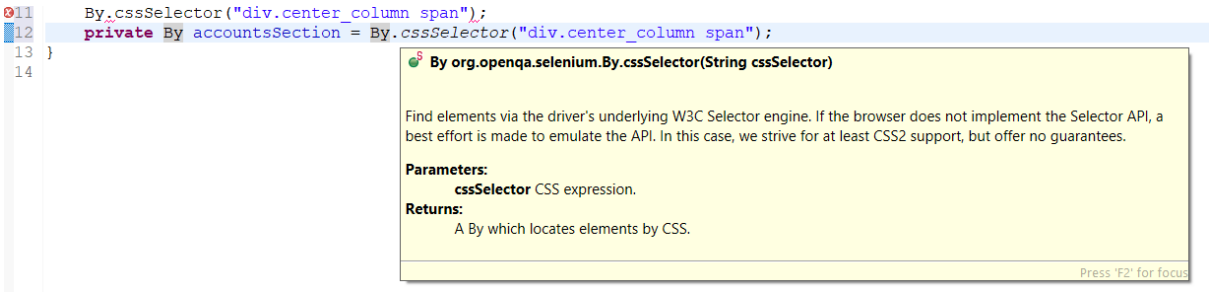
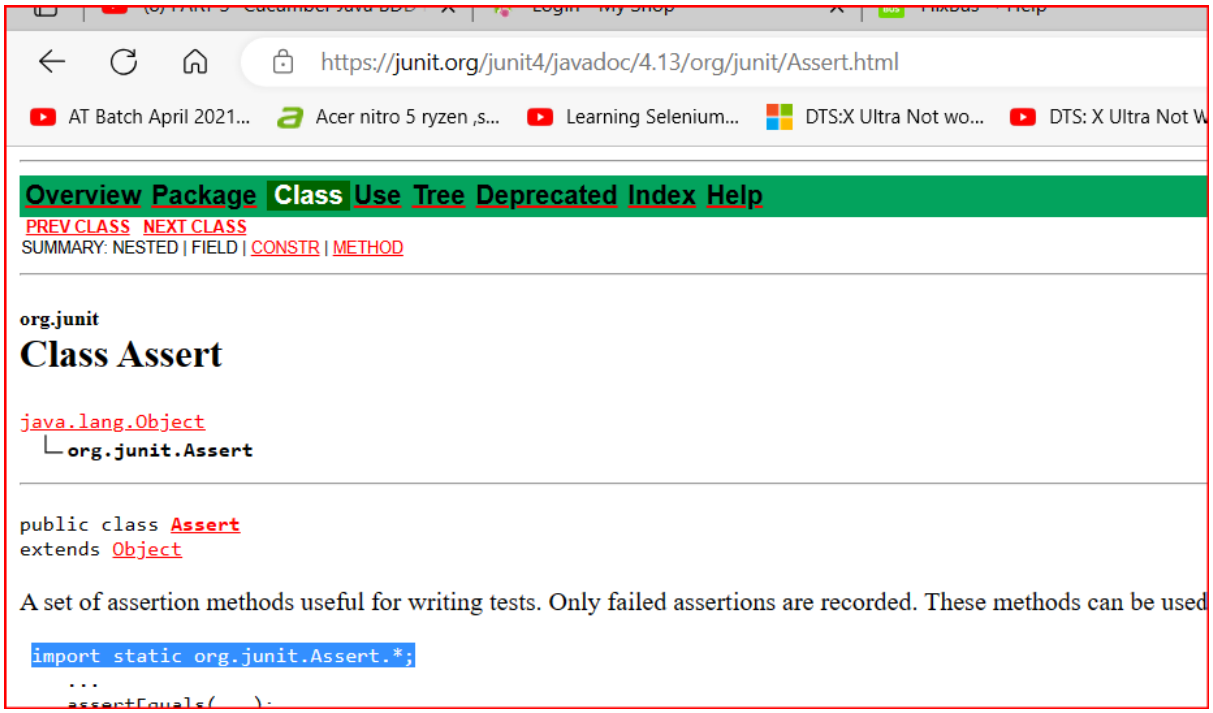
We are checking the return type and the parameter for By-



Sometimes common methods of all page class classes can be moved to a common package if needed.

Use this assert class as the testng one is deprecated-

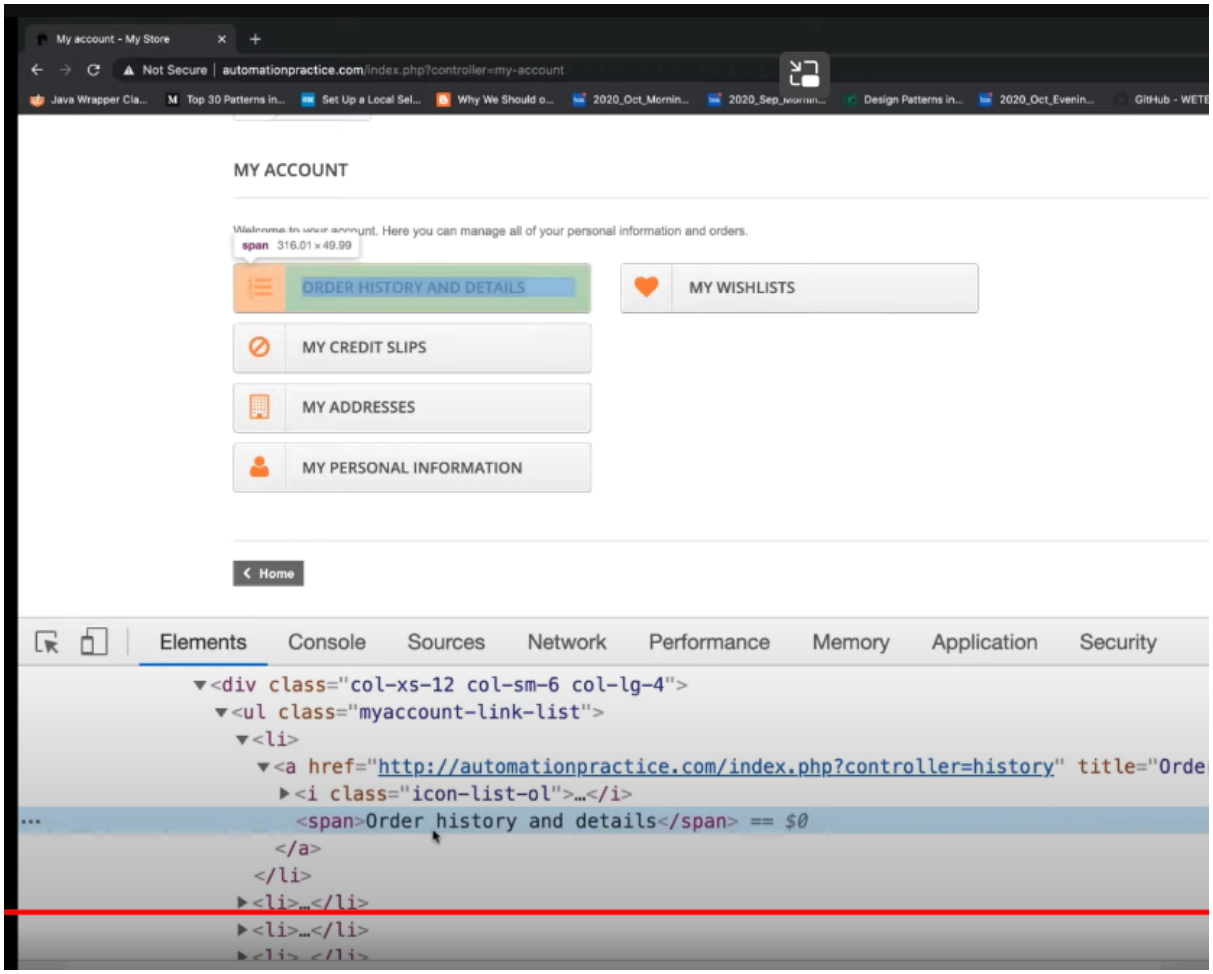


[Assert (JUnit API)]

When we run our test for first time we get assertion error-

Java is case sensitive. So expected and actual output not matching.

Selenium will always give preference to the visible text on the page and not the dom text, so we were getting the capitalization issue-



Accounts page code and its run-

```
Accounts feature file
Feature: account page feature

# we will use background here because we want the precondition that the
user should be logged in.
# if we change the hooks method and add the login there then out
loginpage.feature will be impacted.

Background:
Given user has already logged into application
|userName|password|
|cucumbertesting1960@gmail.com|Malaravi@123|

#since we dont want to parameterise the above background so we used data
table instead of examples.

Scenario: accounts page title
Given user is on accounts page
When user gets the title of the page
Then page title should be "My account - My Shop"

Scenario: account sections count
Given user is on accounts page
Then user gets accounts section on the page
#again we use data tables to pass in the data and check if all ok
# we dont want to parameterise, just verification of data
```

```
|ADD MY FIRST ADDRESS|
|ORDER HISTORY AND DETAILS|
|MY CREDIT SLIPS|
|MY ADDRESSES|
|MY PERSONAL INFORMATION|
|Home|
And account section page should have count as 6
```

Account steps-

```
package stepDefinitions;

import java.util.List;
import java.util.Map;

import org.junit.Assert;

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

import io.cucumber.datatable.DataTable;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;

public class AccountPageSteps {

	private LoginPageObject loginPageObject=new
LoginPageObject(DriverFactory.getDriver());
	private AccountPageObject accountPageObject;

	@Given("user has already logged into application")
	public void user_has_already_logged_into_application(DataTable
credTable) {
		List<Map<String, String>> credList=credTable.asMaps(); //as
maps returns list of map which contains two string
		//arguments.
		//get(0) will give the first map pair of strings. from that
first map pair we pass the key and get the value.
		//in our case the first map pair will be username and password.
		String userName=credList.get(0).get("userName");
		String password=credList.get(0).get("password");

DriverFactory.getDriver().get("http://www.automationpractice.pl/index.php
?controller=authentication&back=my-account");
		accountPageObject=loginPageObject.doLogin(userName, password);
	}

	@Given("user is on accounts page")
	public void user_is_on_accounts_page() {
	 String pageTitle=   accountPageObject.getAccountPageTitle();
	 System.out.println("accounts page title is " + pageTitle);
	}

	@Then("user gets accounts section on the page")
	public void user_gets_accounts_section_on_the_page(DataTable
sectionsList) {
		List<String> expectedAccountSectionList=
sectionsList.asList();
		System.out.println("expected account section list is " +
expectedAccountSectionList);
```

```
            List<String> actualAccountSectionList =
accountPageObject.getAccountsSectionList();
            System.out.println("actual account section list is " +
actualAccountSectionList);


      Assert.assertTrue(expectedAccountSectionList.containsAll(actualAcco
untSectionList));
      }

      @Then("account section page should have count as {int}")
      public void account_section_page_should_have_count_as(Integer
expectedAccountSectionCount) {

Assert.assertTrue(accountPageObject.getAccountSectionCount()==expectedAcc
ountSectionCount);
      }
}
```

Account page object-

```
package com.pages;

import java.util.ArrayList;
import java.util.List;

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

import com.qa.factory.DriverFactory;

public class AccountPageObject {

      private WebDriver driver;

      //we will create a list of objects and capture the text from the
account section.
      private By accountsSection = By.cssSelector("div.center_column
span");

      public AccountPageObject(WebDriver driver) {
            this.driver=driver;
      }

      public int getAccountSectionCount() {
            return driver.findElements(accountsSection).size();
      }

      public List<String> getAccountsSectionList() {
            List<String> accountsList=new ArrayList<>();
            List<WebElement>
accountsHeadersList=driver.findElements(accountsSection);
            //using for each we capture the text and store element in new
array list
            for(WebElement e:accountsHeadersList) {
                  String accountText=e.getText();
                  System.out.println(accountText);
                  accountsList.add(accountText);
            }
            return accountsList; //to return the list of strings which is
easy to manipulate, first we created
```

```
            //list of web elements and then captured the text,  store in
new list and return the new list.
      }

      /**
       * this will return the title of the accounts page
       * @return
       */
      public String getAccountPageTitle() {
            return driver.getTitle();
      }
}
```

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);
      }
```

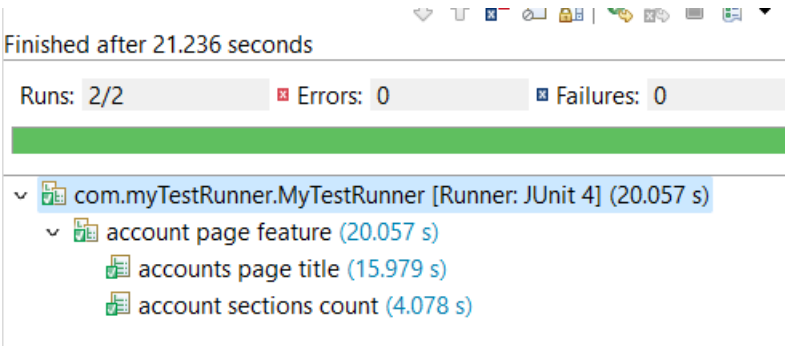
```
	public void enterPassword(String pwd) {
		driver.findElement(password).sendKeys(pwd);
	}

	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.

	// we will write one combine method for login which will take in username,
	// password and login button click

	public AccountPageObject doLogin(String un, String pwd) {
		System.out.println("login with " + un + "pwd " + pwd);
		driver.findElement(emailID).sendKeys(un);
		driver.findElement(password).sendKeys(pwd);
		driver.findElement(signInButton).click();
		return new AccountPageObject(driver); // this is called page chaining concept.
		//do login gives accounts page once logged in.
	}

	//according to page object model, when a method is landing you to next page from the
current one
	//then it is that methods responsibility to give you the object of that landing (new) page.
	//this is page chaining concept used in frameworks.
}
```

Test runner-

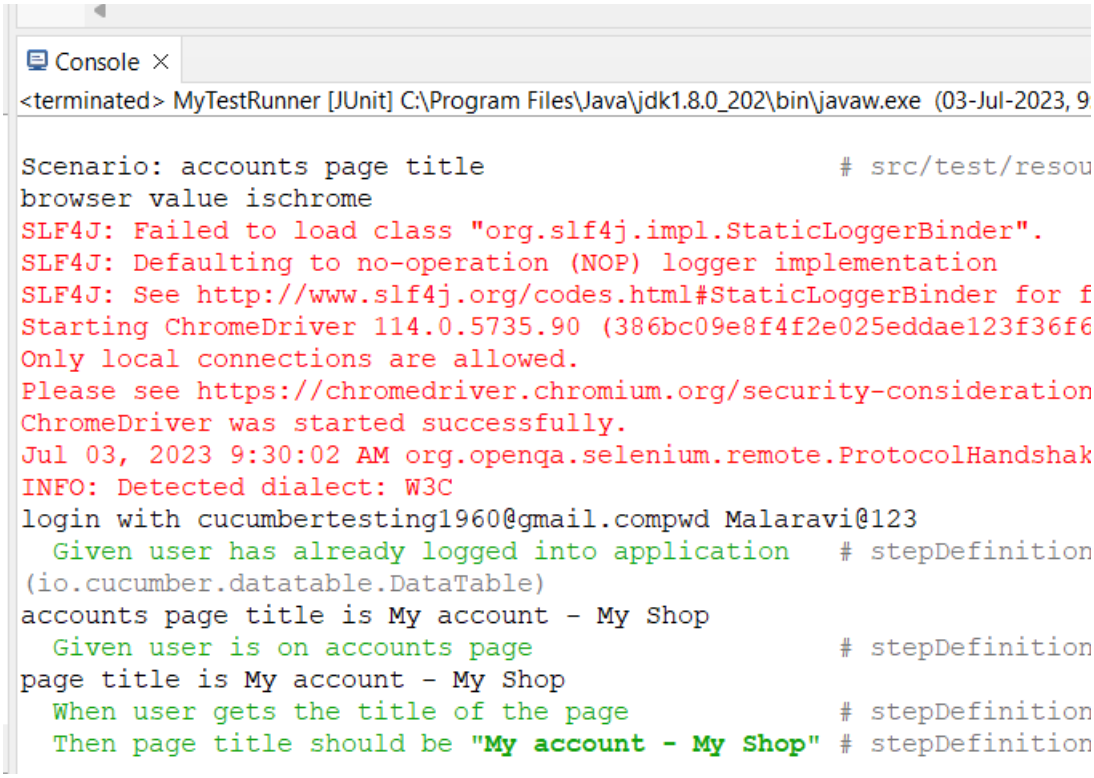
```
package com.myTestRunner;

import org.junit.runner.RunWith;

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

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

)
public class MyTestRunner {

}
```

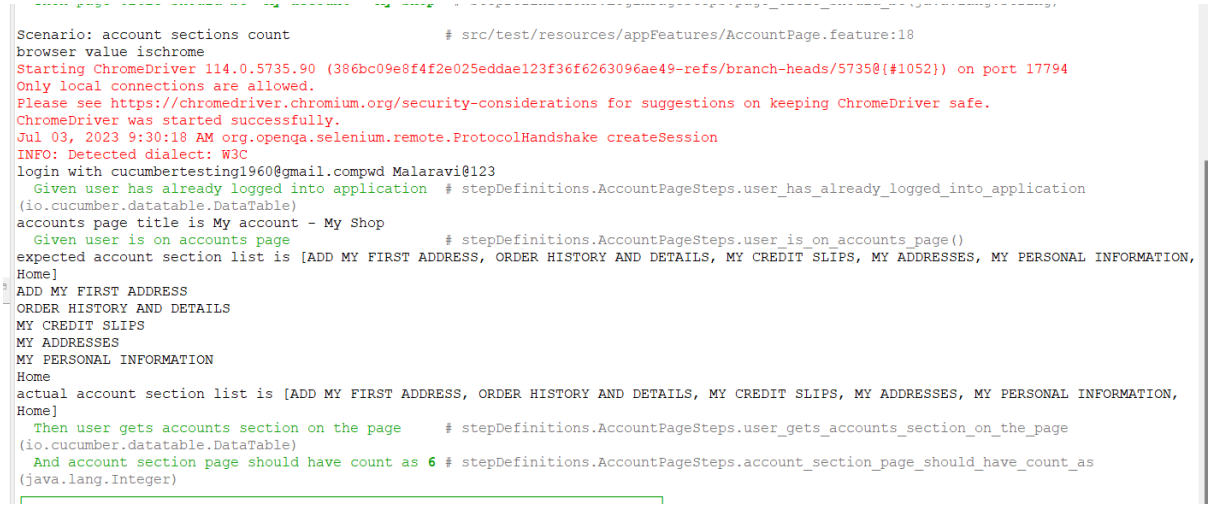
output-

Junit output-

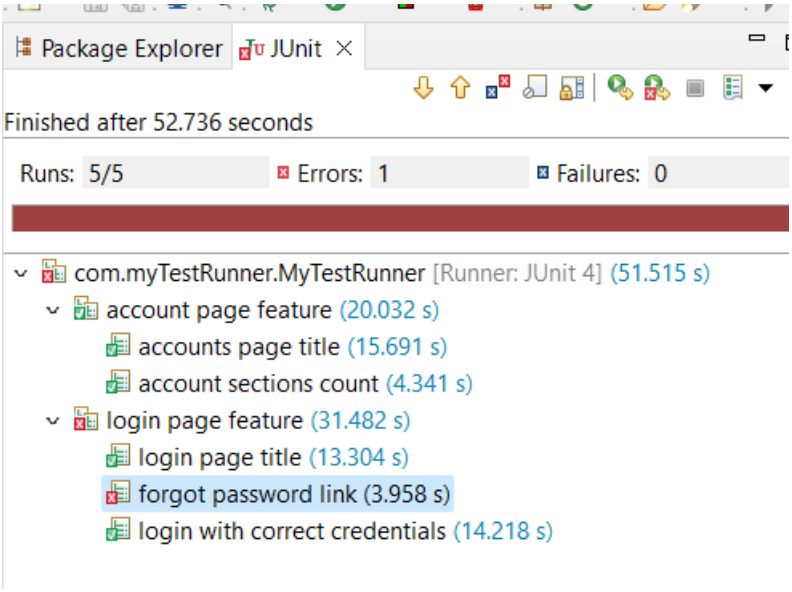


Console output-





Running login and accounts page together-



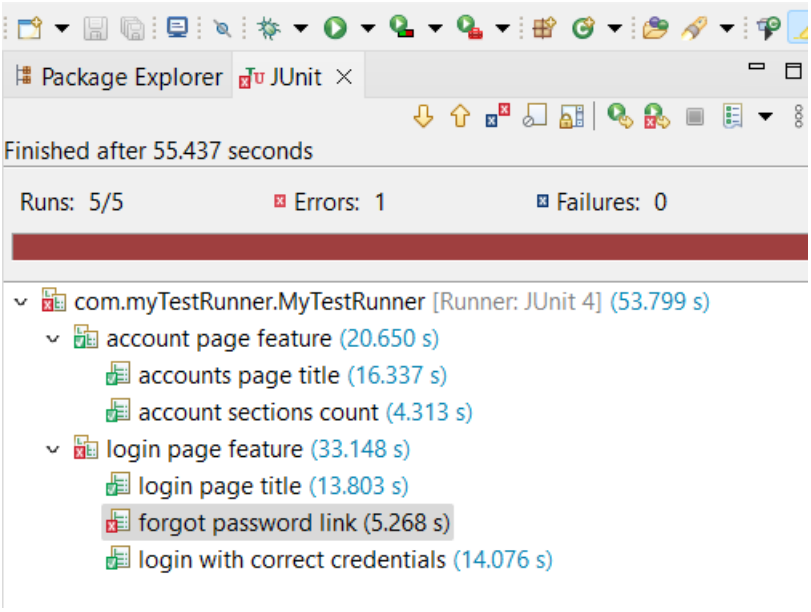
It runs in alphabetical order of page names. So, account page will run first followed by login page.

Now let's run login and accounts page together-

In runner just modify the feature file line.

```
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 {
}
```

Junit console-



Console-

```
Scenario: accounts page title                          # src/test/
browser value ischrome
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder'
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder
Starting ChromeDriver 114.0.5735.90 (386bc09e8f4f2e025eddae123
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considera
ChromeDriver was started successfully.
Jul 04, 2023 7:43:25 AM org.openqa.selenium.remote.ProtocolHand
INFO: Detected dialect: W3C
login with cucumbertesting1960@gmail.compwd Malaravi@123
  Given user has already logged into application    # stepDefin
(io.cucumber.datatable.DataTable)
accounts page title is My account - My Shop
  Given user is on accounts page                    # stepDefin
page title is My account - My Shop
  When user gets the title of the page              # stepDefin
  Then page title should be "My account - My Shop"  # stepDefin
```

```
Scenario: account sections count                # src/test/resources/appFeatures/AccountPage.feature:18
browser value ischrome
Starting ChromeDriver 114.0.5735.90 (386bc09e8f4f2e025eddae123f36f6263096ae49-refs/branch-heads/5735@{#1052}) on port 34611
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 04, 2023 7:43:40 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
login with cucumbertesting1960@gmail.compwd Malaravi@123
  Given user has already logged into application  # stepDefinitions.AccountPageSteps.user_has_already_logged_into_application
(io.cucumber.datatable.DataTable)
accounts page title is My account - My Shop
  Given user is on accounts page                # stepDefinitions.AccountPageSteps.user_is_on_accounts_page()
expected account section list is [ADD MY FIRST ADDRESS, ORDER HISTORY AND DETAILS, MY CREDIT SLIPS, MY ADDRESSES, MY PERSONAL INFORMATION,
Home]
ADD MY FIRST ADDRESS
ORDER HISTORY AND DETAILS
MY CREDIT SLIPS
MY ADDRESSES
MY PERSONAL INFORMATION
Home
actual account section list is [ADD MY FIRST ADDRESS, ORDER HISTORY AND DETAILS, MY CREDIT SLIPS, MY ADDRESSES, MY PERSONAL INFORMATION,
Home]
  Then user gets accounts section on the page     # stepDefinitions.AccountPageSteps.user_gets_accounts_section_on_the_page
(io.cucumber.datatable.DataTable)
  And account section page should have count as 6 # stepDefinitions.AccountPageSteps.account_section_page_should_have_count_as
(java.lang.Integer)
```

```
Scenario: login page title                      # src/test/res
browser value ischrome
Starting ChromeDriver 114.0.5735.90 (386bc09e8f4f2e025eddae1
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-consic
ChromeDriver was started successfully.
Jul 04, 2023 7:43:44 AM org.openqa.selenium.remote.ProtocolH
INFO: Detected dialect: W3C
  Given user is on login page                   # stepDefiniti
page title is Login - My Shop
  When user gets the title of the page          # stepDefiniti
  Then page title should be "Login - My Shop"   # stepDefiniti
```

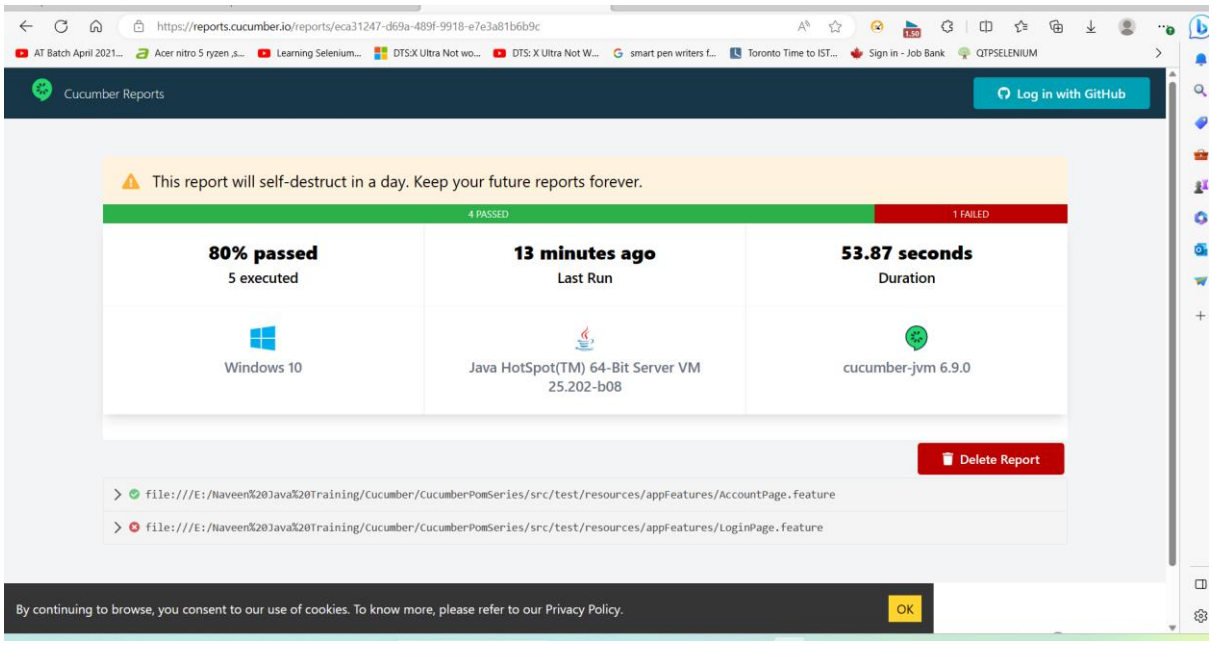
```
Scenario: forgot password link                  # src/test/resources/appFeatures/LoginPage.feature:8
browser value ischrome
Starting ChromeDriver 114.0.5735.90 (386bc09e8f4f2e025eddae123f36f6263096ae49-refs/branch-heads/5735@{#1052}) on port 9578
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 04, 2023 7:43:58 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:65029}, 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: 8d5682c74f3ba2e08824808fbe80f404
*** 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)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
        at org.openqa.selenium.remote.http.W3CHttpResponseCodec.createException(W3CHttpResponseCodec.java:187)
        at org.openqa.selenium.remote.http.W3CHttpResponseCodec.decode(W3CHttpResponseCodec.java:122)
        at org.openqa.selenium.remote.http.W3CHttpResponseCodec.decode(W3CHttpResponseCodec.java:49)
        at org.openqa.selenium.remote.HttpCommandExecutor.execute(HttpCommandExecutor.java:158)
```

Writable | Smart Insert | 18 : 1 [351]

```
    Embedding forgot_password_link [image/png 158345 bytes]


Scenario: login with correct credentials                    # src/test/resources/appFeat
browser value ischrome
Starting ChromeDriver 114.0.5735.90 (386bc09e8f4f2e025eddae123f36f6263096ae49-refs/branch-h
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on kee
ChromeDriver was started successfully.
Jul 04, 2023 7:44:03 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
  Given user is on login page                               # stepDefinitions.LoginPageS
  When user enters username as "cucumbertesting1960@gmail.com" # stepDefinitions.LoginPageS
  And user enters password as "Malaravi@123"                # stepDefinitions.LoginPageS
  And user clicks on login button                           # stepDefinitions.LoginPageS
page title is My account - My Shop
  Then user gets the title of the page                      # stepDefinitions.LoginPageS
  And page title should be "My account - My Shop"           # stepDefinitions.LoginPageS

View your Cucumber Report at:
https://reports.cucumber.io/reports/eca31247-d69a-489f-9918-e7e3a81b6b9c

This report will self-destruct in 24h.
Keep reports forever: https://reports.cucumber.io/profile
```

Writable | Smart Insert | 18 : 1

Report link-



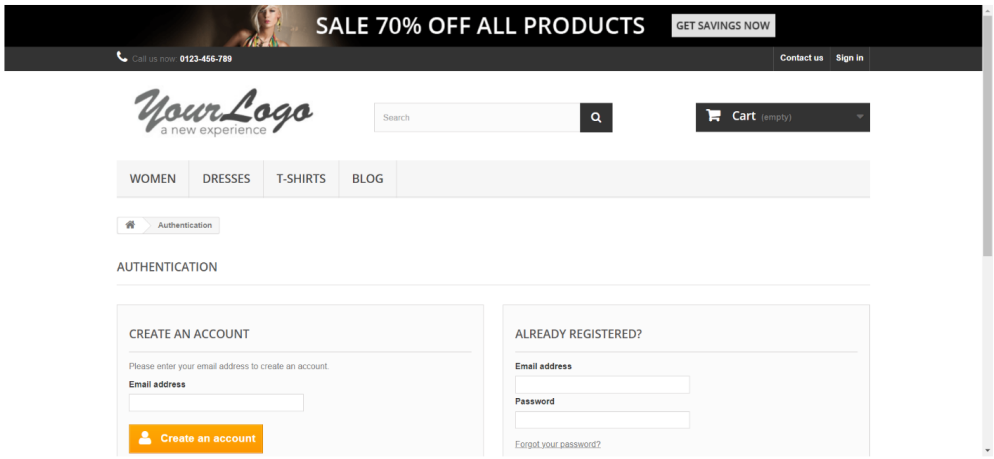
Scenario: forgot password link

✔ **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 (386bc09e8f4f...,
userDataDir: C:\Users\karan\AppData\Loca...}, goog:chromeOptions: {debuggerAddress: localhost:65029}, 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: 8d5682c74f3ba2e08824808fbe80f404
*** 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)
    at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
    at org.openqa.selenium.remote.http.W3CHttpResponseCodec.createException(W3CHttpResponseCodec.java:187)
    at org.openqa.selenium.remote.http.W3CHttpResponseCodec.decode(W3CHttpResponseCodec.java:122)
    at org.openqa.selenium.remote.http.W3CHttpResponseCodec.decode(W3CHttpResponseCodec.java:49)
    at org.openqa.selenium.remote.HttpCommandExecutor.execute(HttpCommandExecutor.java:158)
    at org.openqa.selenium.remote.service.DriverCommandExecutor.execute(DriverCommandExecutor.java:83)
    at org.openqa.selenium.remote.RemoteWebDriver.execute(RemoteWebDriver.java:552)
    at org.openqa.selenium.remote.RemoteWebDriver.findElement(RemoteWebDriver.java:323)
    at org.openqa.selenium.remote.RemoteWebDriver.findElementByLinkText(RemoteWebDriver.java:380)
```

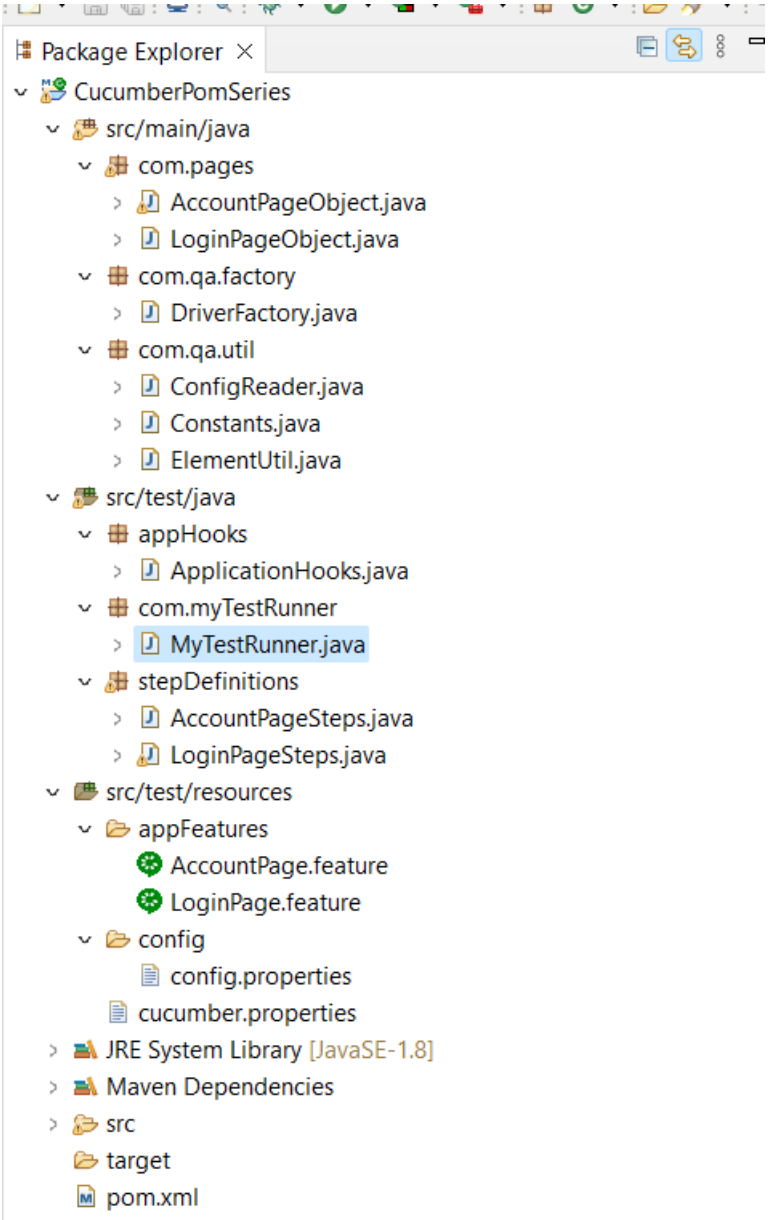

▼ Attached Image



enario: login with correct credentials

✔ Given user is on login page

Project structure-



Codes from this lecture-

Account page feature-

```
Feature: account page feature

# we will use background here because we want the precondition that the
user should be logged in.
# if we change the hooks method and add the login there then out
loginpage.feature will be impacted.

Background:
Given user has already logged into application
|userName|password|
|cucumbertesting1960@gmail.com|Malaravi@123|

#since we dont want to parameterise the above background so we used data
table instead of examples.
```

```
Scenario: accounts page title
Given user is on accounts page
When user gets the title of the page
Then page title should be "My account - My Shop"

Scenario: account sections count
Given user is on accounts page
Then user gets accounts section on the page
#again we use data tables to pass in the data and check if all ok
# we dont want to parameterise, just verification of data
|ADD MY FIRST ADDRESS|
|ORDER HISTORY AND DETAILS|
|MY CREDIT SLIPS|
|MY ADDRESSES|
|MY PERSONAL INFORMATION|
|Home|
And account section page should have count as 6
```

Account page steps-

```
package stepDefinitions;

import java.util.List;
import java.util.Map;

import org.junit.Assert;

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

import io.cucumber.datatable.DataTable;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;

public class AccountPageSteps {

	private LoginPageObject loginPageObject=new
LoginPageObject(DriverFactory.getDriver());
	private AccountPageObject accountPageObject;

	@Given("user has already logged into application")
	public void user_has_already_logged_into_application(DataTable
credTable) {
		List<Map<String, String>> credList=credTable.asMaps(); //as
maps returns list of map which contains two string
		//arguments.
		//get(0) will give the first map pair of strings. from that
first map pair we pass the key and get the value.
		//in our case the first map pair will be username and password.
		String userName=credList.get(0).get("userName");
		String password=credList.get(0).get("password");

DriverFactory.getDriver().get("http://www.automationpractice.pl/index.php
?controller=authentication&back=my-account");
		accountPageObject=loginPageObject.doLogin(userName, password);
	}

	@Given("user is on accounts page")
	public void user_is_on_accounts_page() {
	 String pageTitle=   accountPageObject.getAccountPageTitle();
	 System.out.println("accounts page title is " + pageTitle);
	}
```

```
	@Then("user gets accounts section on the page")
	public void user_gets_accounts_section_on_the_page(DataTable
sectionsList) {
		List<String> expectedAccountSectionList=
sectionsList.asList();
		System.out.println("expected account section list is " +
expectedAccountSectionList);

		List<String> actualAccountSectionList =
accountPageObject.getAccountsSectionList();
		System.out.println("actual account section list is " +
actualAccountSectionList);


	Assert.assertTrue(expectedAccountSectionList.containsAll(actualAcco
untSectionList));
	}

	@Then("account section page should have count as {int}")
	public void account_section_page_should_have_count_as(Integer
expectedAccountSectionCount) {

Assert.assertTrue(accountPageObject.getAccountSectionCount()==expectedAcc
ountSectionCount);
	}
}
```

Accounts page object-

```
package com.pages;

import java.util.ArrayList;
import java.util.List;

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

import com.qa.factory.DriverFactory;

public class AccountPageObject {

	private WebDriver driver;

	//we will create a list of objects and capture the text from the
account section.
	private By accountsSection = By.cssSelector("div.center_column
span");

	public AccountPageObject(WebDriver driver) {
		this.driver=driver;
	}

	public int getAccountSectionCount() {
		return driver.findElements(accountsSection).size();
	}

	public List<String> getAccountsSectionList() {
		List<String> accountsList=new ArrayList<>();
		List<WebElement>
accountsHeadersList=driver.findElements(accountsSection);
```

```
            //using for each we capture the text and store element in new
array list
            for(WebElement e:accountsHeadersList) {
                  String accountText=e.getText();
                  System.out.println(accountText);
                  accountsList.add(accountText);
            }
            return accountsList; //to return the list of strings which is
easy to manipulate, first we created
            //list of web elements and then captured the text,  store in
new list and return the new list.
      }

      /**
       * this will return the title of the accounts page
       * @return
       */
      public String getAccountPageTitle() {
            return driver.getTitle();
      }
}
```

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.

	// we will write one combine method for login which will take in username,
	// password and login button click

	public AccountPageObject doLogin(String un, String pwd) {
		System.out.println("login with " + un + "pwd " + pwd);
		driver.findElement(emailID).sendKeys(un);
		driver.findElement(password).sendKeys(pwd);
		driver.findElement(signInButton).click();
		return new AccountPageObject(driver); // this is called page chaining concept.
		//do login gives accounts page once logged in.
	}

	//according to page object model, when a method is landing you to next page from the
current one
	//then it is that methods responsibility to give you the object of that landing (new) page.
	//this is page chaining concept used in frameworks.
}
```

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 {
}
```