SOFTWARE TESTING PROJECT REPORT
AUTOMATION FRAMEWORK

Muhammad Umair Khan 18k-1292

Muhammad Mujtaba 18i-0623

Submitted To: Sir AMIR IMAAM

Table of Contents

ACKNOWLEDGMENT

We would like to thank Sir Amir Imam who gave us this opportunity to work on this project. I got to learn a lot from this project about testing and its scope which has inspired us to pursue a career in automation testing.

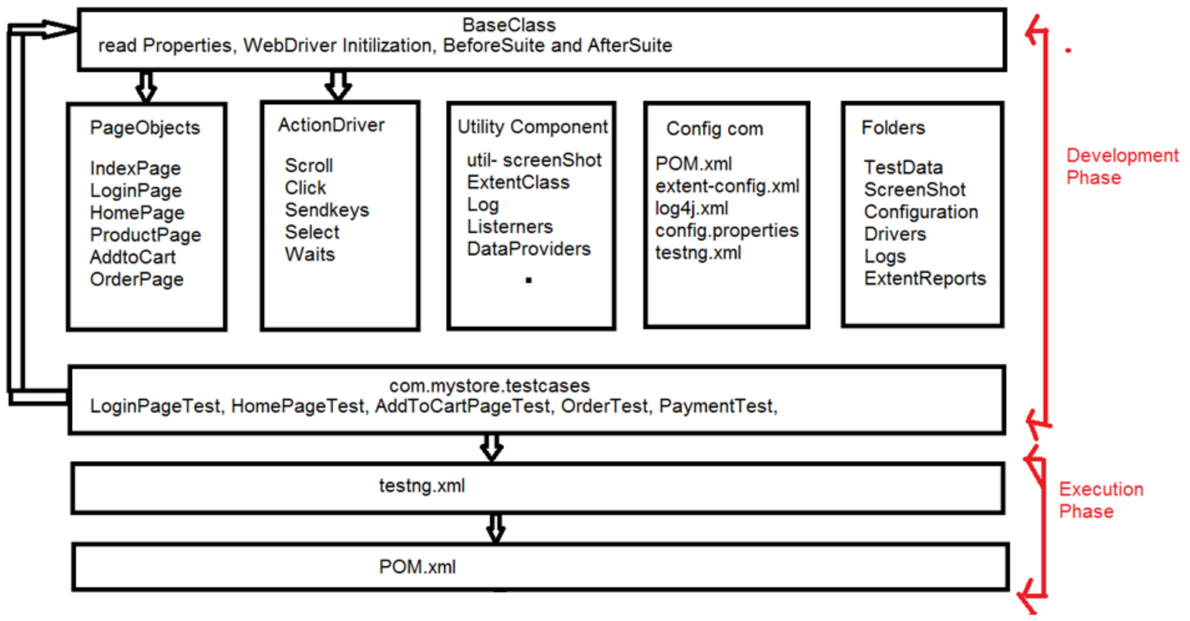
ABSTRACT

The automation framework focuses on website which conducts multiple type of testing such as cross browser testing, sanity testing, regression testing, smoke testing, performance testing and accessibility testing.

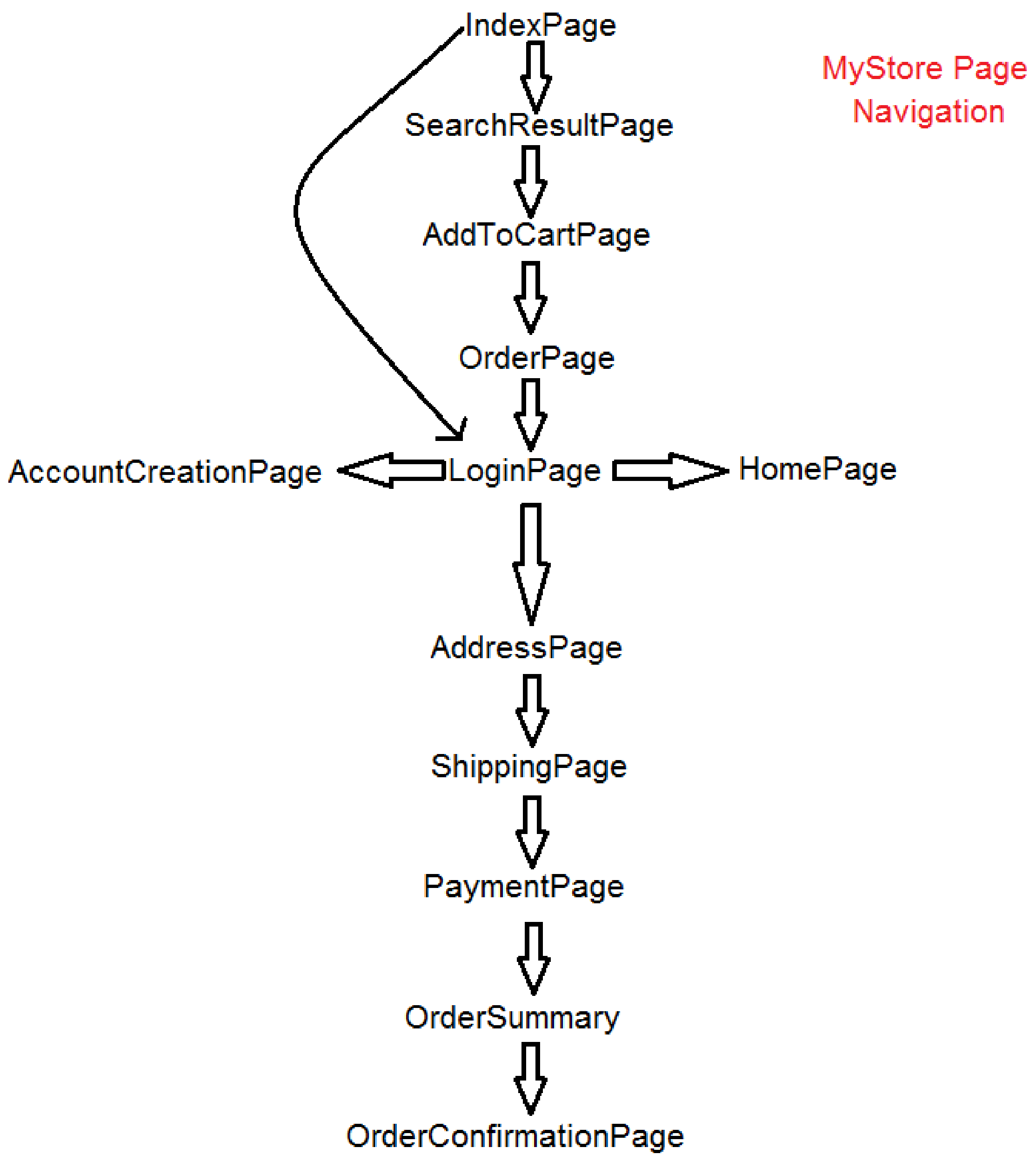
MOTIVATION

Every software development group tests its products, yet delivered software always has defects. Test engineers strive to catch them before the product is released but they always creep in and they often reappear, even with the best manual testing processes. Test Automation software is the best way to increase the effectiveness, efficiency and coverage of your software testing.

Framework Architecture:



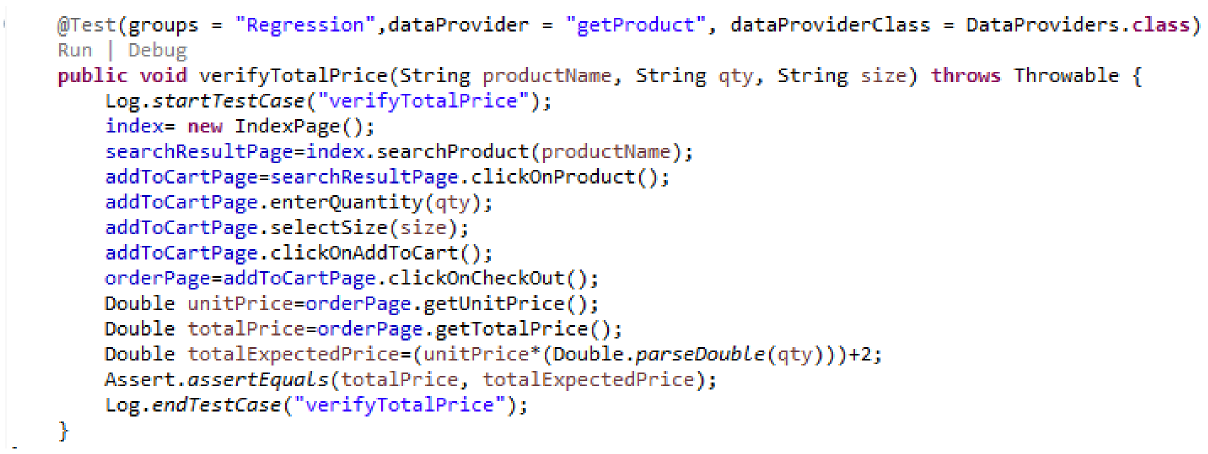
Testing Pages Navigation:



Various Testing:

Regression Testing:

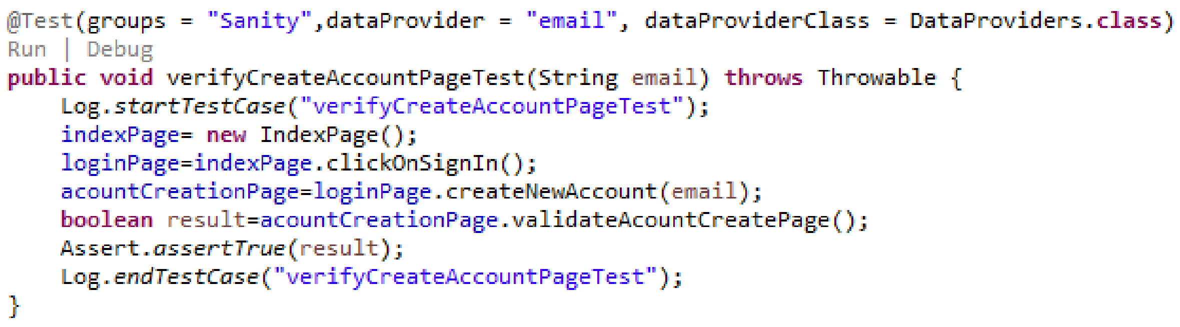
Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product.



```java
@Test(groups = "Regression",dataProvider = "getProduct", dataProviderClass = DataProviders.class)
Run | Debug
public void verifyTotalPrice(String productName, String qty, String size) throws Throwable {
    Log.startTestCase("verifyTotalPrice");
    index= new IndexPage();
    searchResultPage=index.searchProduct(productName);
    addToCartPage=searchResultPage.clickOnProduct();
    addToCartPage.enterQuantity(qty);
    addToCartPage.selectSize(size);
    addToCartPage.clickOnAddToCart();
    orderPage=addToCartPage.clickOnCheckOut();
    Double unitPrice=orderPage.getUnitPrice();
    Double totalPrice=orderPage.getTotalPrice();
    Double totalExpectedPrice=(unitPrice*(Double.parseDouble(qty)))+2;
    Assert.assertEquals(totalPrice, totalExpectedPrice);
    Log.endTestCase("verifyTotalPrice");
}
```

Sanity Testing:

A sanity check or sanity test is a basic test to quickly evaluate whether a claim or the result of a calculation can possibly be true

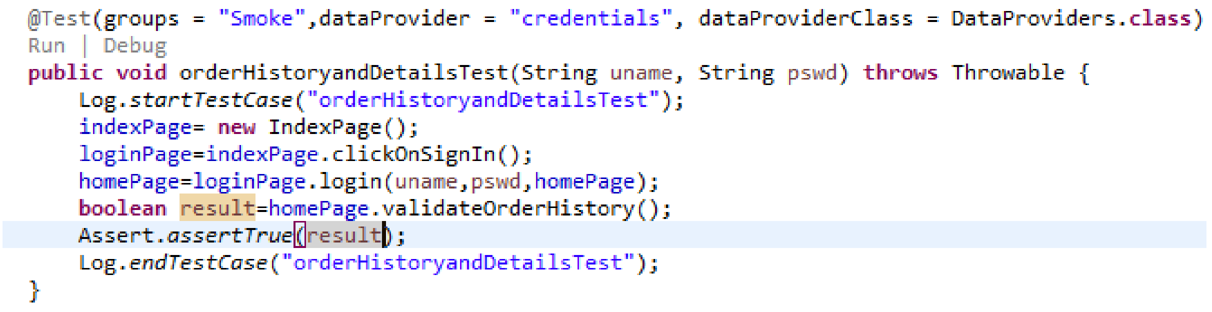


```java
@Test(groups = "Sanity",dataProvider = "email", dataProviderClass = DataProviders.class)
Run | Debug
public void verifyCreateAccountPageTest(String email) throws Throwable {
    Log.startTestCase("verifyCreateAccountPageTest");
    indexPage= new IndexPage();
    loginPage=indexPage.clickOnSignIn();
    acountCreationPage=loginPage.createNewAccount(email);
    boolean result=acountCreationPage.validateAcountCreatePage();
    Assert.assertTrue(result);
    Log.endTestCase("verifyCreateAccountPageTest");
}
```

.

Smoke Testing:

Smoke Testing is a software testing process that determines whether the deployed software build is stable or not. Smoke testing is a confirmation for QA team to proceed with further software testing.



```java
@Test(groups = "Smoke",dataProvider = "credentials", dataProviderClass = DataProviders.class)
Run | Debug
public void orderHistoryandDetailsTest(String uname, String pswd) throws Throwable {
    Log.startTestCase("orderHistoryandDetailsTest");
    indexPage= new IndexPage();
    loginPage=indexPage.clickOnSignIn();
    homePage=loginPage.login(uname,pswd,homePage);
    boolean result=homePage.validateOrderHistory();
    Assert.assertTrue(result);
    Log.endTestCase("orderHistoryandDetailsTest");
}
```

Cross Browser Testing:

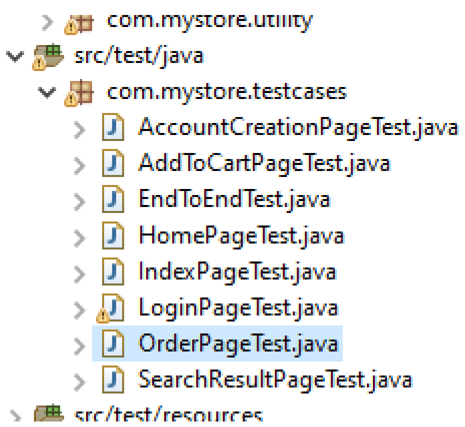
Cross Browser testing is a type of non-functional testing that lets you check whether your website works as intended when accessed through: Different Browser-OS used:

1) Chrome
2) Firefox
3) Internet Explorer

Implementation:

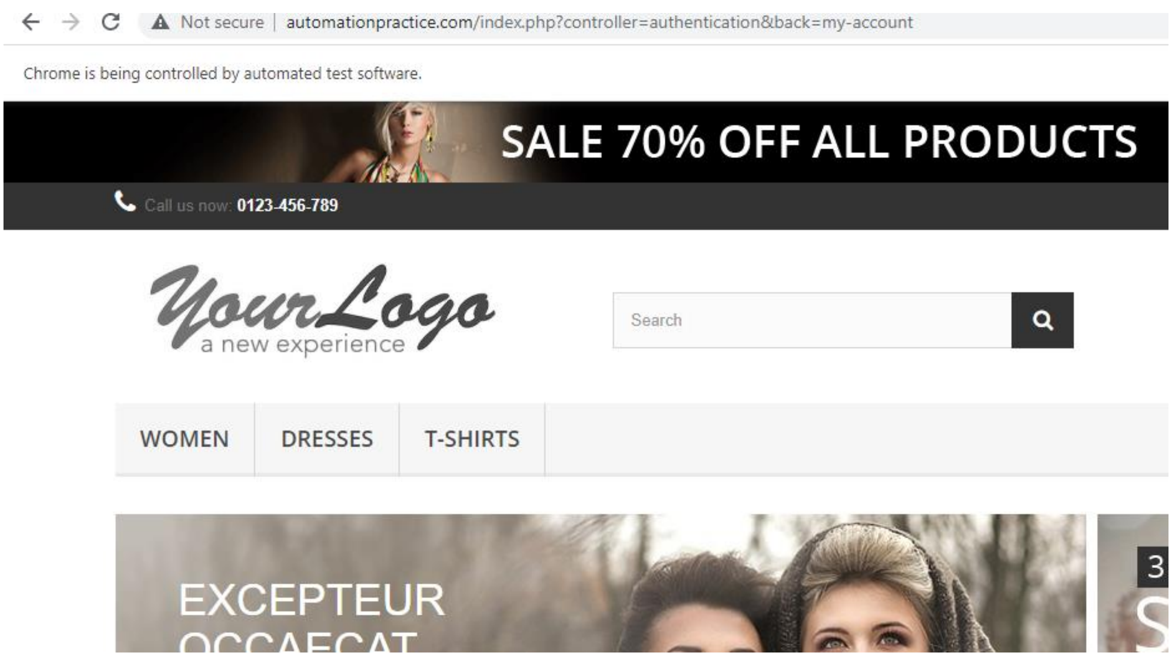
Test Cases

After running the framework on eclipse, all the page objects are tested through the test cases written in their respective files.



```
  > com.mystore.utility
v src/test/java
  v com.mystore.testcases
    > AccountCreationPageTest.java
    > AddToCartPageTest.java
    > EndToEndTest.java
    > HomePageTest.java
    > IndexPageTest.java
    > LoginPageTest.java
    > OrderPageTest.java
    > SearchResultPageTest.java
> src/test/resources
```

Driver opens the selected browser and executes the test cases one by one, **testing_all.xml** runs all the test cases.



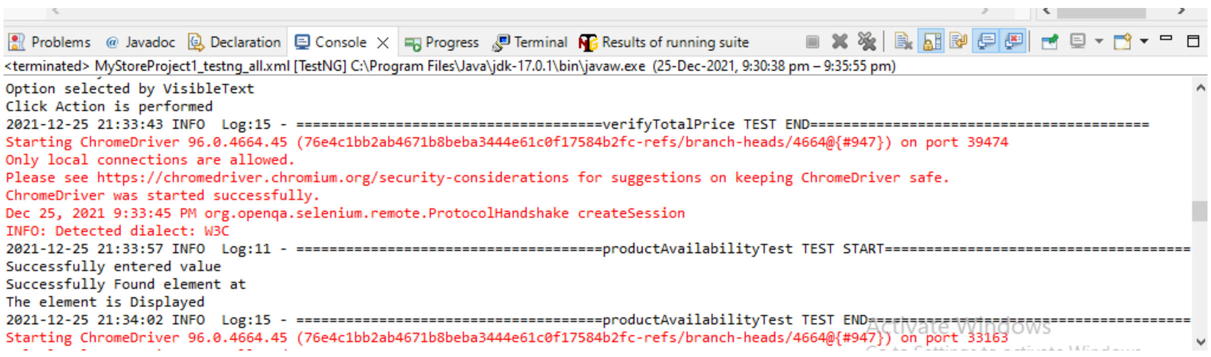
Not secure | automationpractice.com/index.php?controller=authentication&back=my-account

Chrome is being controlled by automated test software.

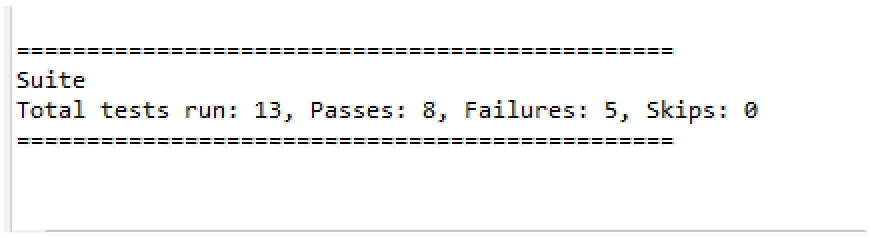
SALE 70% OFF ALL PRODUCTS
Call us now: 0123-456-789
YourLogo a new experience
Search
WOMEN
DRESSES
T-SHIRTS
EXCEPTEUR

Console display:

Console displays the test case status and which test case is being executed right now.



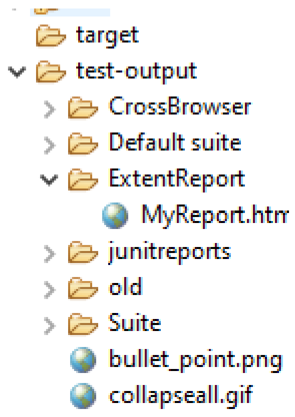
```
<terminated> MyStoreProject1_testng_all.xml [TestNG] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe  (25-Dec-2021, 9:30:38 pm – 9:35:55 pm)
Option selected by VisibleText
Click Action is performed
2021-12-25 21:33:43 INFO  Log:15 - =====================================verifyTotalPrice TEST END=========================================
Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8beba3444e61c0f17584b2fc-refs/branch-heads/4664@{#947}) on port 39474
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Dec 25, 2021 9:33:45 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
2021-12-25 21:33:57 INFO  Log:11 - =====================================productAvailabilityTest TEST START=====================================
Successfully entered value
Successfully Found element at
The element is Displayed
2021-12-25 21:34:02 INFO  Log:15 - =====================================productAvailabilityTest TEST END=======================================
Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8beba3444e61c0f17584b2fc-refs/branch-heads/4664@{#947}) on port 33163
```



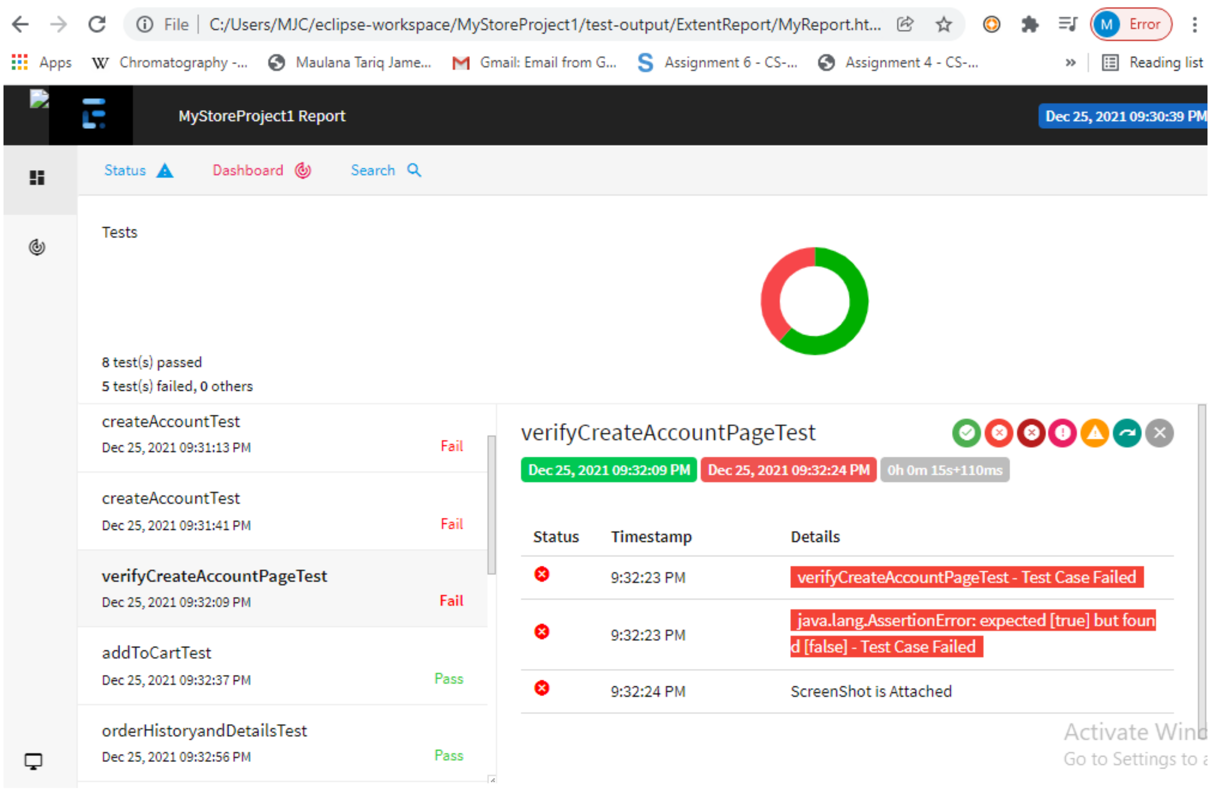
```
===============================================
Suite
Total tests run: 13, Passes: 8, Failures: 5, Skips: 0
===============================================
```

Test Cases Report:

After the test cases have been executed, a reported is generated in html format in the **Extent Report** folder.



```
target
test-output
  CrossBrowser
  Default suite
  ExtentReport
    MyReport.htm
  junitreports
  old
  Suite
  bullet_point.png
  collapseall.gif
```



MyStoreProject1 Report

Dec 25, 2021 09:30:39 PM

Tests

8 test(s) passed
5 test(s) failed, 0 others

createAccountTest
Dec 25, 2021 09:31:13 PM Fail

createAccountTest
Dec 25, 2021 09:31:41 PM Fail

verifyCreateAccountPageTest
Dec 25, 2021 09:32:09 PM Fail

addToCartTest
Dec 25, 2021 09:32:37 PM Pass

orderHistoryandDetailsTest
Dec 25, 2021 09:32:56 PM Pass

verifyCreateAccountPageTest

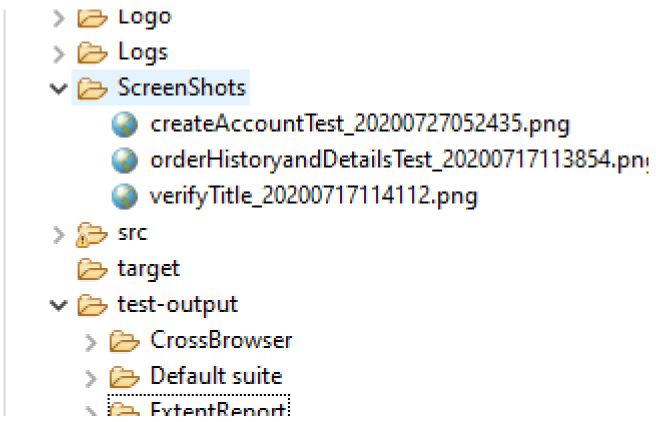
Dec 25, 2021 09:32:09 PM | Dec 25, 2021 09:32:24 PM | 0h 0m 15s+110ms

Status	Timestamp	Details
✗	9:32:23 PM	verifyCreateAccountPageTest - Test Case Failed
✗	9:32:23 PM	java.lang.AssertionError: expected [true] but found [false] - Test Case Failed
✗	9:32:24 PM	ScreenShot is Attached

A dashboard is generated which displays all the details about each test case individually and displays the reason if a test case is not working along with its screenshot.

Screenshot:

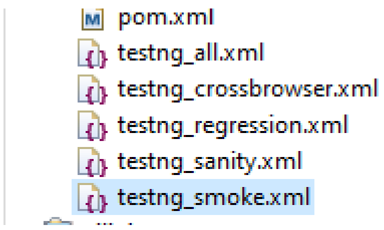
Screenshots folder stores the snap of those test cases which failed to work.



```
> Logo
> Logs
v ScreenShots
    createAccountTest_20200727052435.png
    orderHistoryandDetailsTest_20200717113854.pn
    verifyTitle_20200717114112.png
> src
  target
v test-output
  > CrossBrowser
  > Default suite
  > ExtentReport
```

Running Configuration:

TestNG is a testing framework for the Java programming language created by Cédric Beust and inspired by JUnit and NUnit which has been used to run this framework. Frameworks can be run through 4 different files, where each file has a different purpose defined below.



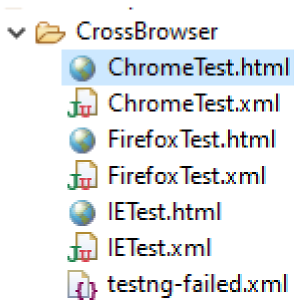
pom.xml
testng_all.xml
testng_crossbrowser.xml
testng_regression.xml
testng_sanity.xml
testng_smoke.xml

testing_all.xml

Runs all the test cases present in project.

testng_crossbrowser.xml

Runs the test cases in three different browsers (Chrome, Firefox and Internet Explorer) simultaneously and report of testcases is also generated of each browser in a separate file.



CrossBrowser
ChromeTest.html
ChromeTest.xml
FirefoxTest.html
FirefoxTest.xml
IETest.html
IETest.xml
testng-failed.xml

testng_regression.xml

This file will only run the test cases of regression testing present inside the project.

testng_sanity.xml

This file will only run the test cases of sanity testing present inside the project.

testng_smoke.xml

This file will only run the test cases of smoke testing present inside the project.

Conclusion:

This automation framework is designed to ease the pressure on testers and software houses won’t have to hire so many testers to test their application, a simple automation framework will do their job.