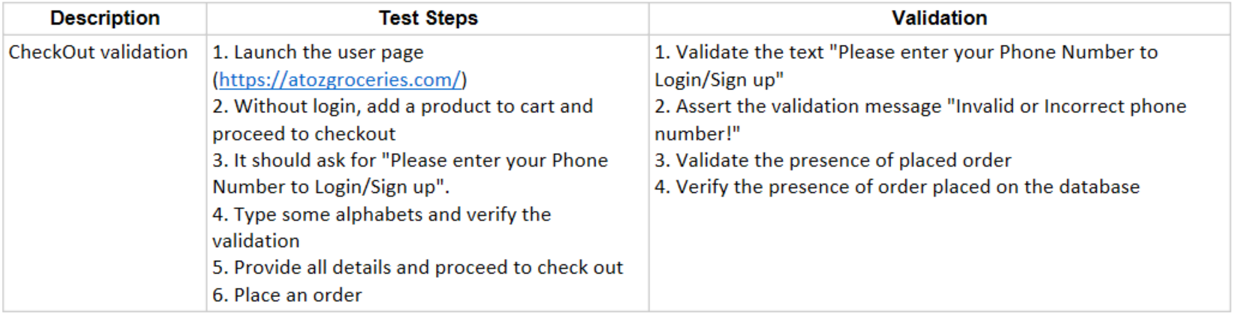
Test Case 13- Complex



| Description | Test Steps | Validation |
|---|---|---|
| CheckOut validation | 1. Launch the user page (https://atozgroceries.com/)<br>2. Without login, add a product to cart and proceed to checkout<br>3. It should ask for "Please enter your Phone Number to Login/Sign up".<br>4. Type some alphabets and verify the validation<br>5. Provide all details and proceed to check out<br>6. Place an order | 1. Validate the text "Please enter your Phone Number to Login/Sign up"<br>2. Assert the validation message "Invalid or Incorrect phone number!"<br>3. Validate the presence of placed order<br>4. Verify the presence of order placed on the database |

## Eclipse programs

BaseTest.java program

```
        package com.ibm.test;


import static org.testng.Assert.assertEquals;

import static org.testng.Assert.assertTrue;


import java.io.IOException;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.concurrent.TimeUnit;


import org.junit.Assert;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.BeforeTest;
//import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

import com.ibm.pages.AdminPage;
import com.ibm.utilities.DbConnectionUser;
//import com.ibm.utilities.ExcelUtil;
import com.ibm.utilities.PropertiesFileHandler;


public class BaseTest {
    WebDriver driver;
    WebDriverWait wait;
    PropertiesFileHandler propFileHandler;
    HashMap<String, String> data;
    DbConnectionUser dbConnect=new DbConnectionUser(driver, wait);
    //Use properties to solve the simple test cases.
    @BeforeSuite
    public void testcase1() throws IOException
    {
        //Locating property file
        String file="./TestData/data.properties";
        propFileHandler = new PropertiesFileHandler();
        data= propFileHandler.getPropertiesAsMap(file);
    }
```

```
	@BeforeMethod

	public void Initialization()

	{

		//Open chrome and wait

		System.setProperty("webdriver.gecko.driver","./drivers/geckodriver.exe");

		//System.setProperty("webdriver.chrome.driver","./drivers/chromedriver.exe");

		//driver=new ChromeDriver();

		driver=new FirefoxDriver();

		wait=new WebDriverWait(driver,60);

		driver.manage().window().maximize();

		driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);



	}



	@AfterMethod

	public void closeBrowser() {

		driver.quit();

	}



	//CheckOut validation from User page..
	@Test(priority=13)
	public void CheckOutValidationInUserPage() throws IOException, 
InterruptedException, SQLException
	{
		//using data from property file
				String url = data.get("userurl");
				String phoneNumErrMsg=data.get("phoneNumErrorMesg");
				String passwordErrorMesg=data.get("passwordErrorMesg");
				String phoneNumDummy=data.get("phoneNumDummy");
				String loginpwordDummy=data.get("loginpwordDummy");
				String phoneNum=data.get("phoneNum");
				String loginpword=data.get("loginpword");
				String checkOutName=data.get("checkOutName");
				String emailAddrInChek=data.get("emailAddrInChek");
				String enterAddrInCheck=data.get("enterAddrInCheck");
				String 
invalidpNumErrorMesg=data.get("invalidpNumErrorMesg");
```

```
                          String pincode=data.get("pincode");
                          String
OrderConfirmMessage=data.get("OrderConfirmMessage");
                          driver.get(url);

                          AdminPage adminPage=new AdminPage(driver, wait);

                          adminPage.CaptureScreenshot(driver);
                          adminPage.NewArrivalbutton();
                          Thread.sleep(1000);
                          adminPage.CaptureScreenshot(driver);

                          adminPage.ClickOnAddtoCardOne();
                          adminPage.ClickOnAddtoCardTwo();
                          adminPage.ClickOnAddtoCardButton();
                          Thread.sleep(1000);
                          adminPage.CaptureScreenshot(driver);

                          adminPage.ClickOnCheckOut();
                          adminPage.ClickOnNextButton();
                          Thread.sleep(1000);
                          adminPage.CaptureScreenshot(driver);

                          String pNumErrMsg = adminPage.PNumErrorMsg();
                          Assert.assertEquals(pNumErrMsg, phoneNumErrMsg);

                          String pwrdErrMsg=adminPage.PwordErrMsg();
                          Assert.assertEquals(pwrdErrMsg, passwordErrorMesg);

                          adminPage.EnterPhnNumdummy(phoneNumDummy);
                          adminPage.EnterPworddummy(loginpwordDummy);
                          adminPage.ClickOnNextButton();
                          Thread.sleep(1000);
                          adminPage.CaptureScreenshot(driver);


                          String invalidPhoneNum=adminPage.PNumErrorMsg();
                          Assert.assertEquals(invalidPhoneNum,
invalidpNumErrorMesg);

                          adminPage.EnterPhnNum(phoneNum);
                          adminPage.EnterPword(loginpword);
                          adminPage.ClickOnNextButton();
                          adminPage.CaptureScreenshot(driver);
                          Thread.sleep(1000);


                          adminPage.EnterNameinCheckOut(checkOutName);
                          adminPage.EnterEmailAddr(emailAddrInChek);
                          adminPage.enterAddrinChek(enterAddrInCheck);
                          adminPage.SelectCityInChek();
                          adminPage.EnterPincodeInChek(pincode);
                          adminPage.SelectResiType();
                          adminPage.CaptureScreenshot(driver);
                          adminPage.ClickOnConfirmPayment();
```

```
                Thread.sleep(1000);

                adminPage.SelectCheckBox();
                adminPage.ConfirmOrder();
                Thread.sleep(1000);
                adminPage.CaptureScreenshot(driver);
                String succMsgg=adminPage.VerifyOrderPlacetext().trim();

                Assert.assertTrue(succMsgg.contains("Thank you! Your order has been placed successfully."));

    }
```

AdminPage.java program

```
package com.ibm.pages;


import static org.testng.Assert.assertEquals;

import static org.testng.Assert.fail;


import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Date;

import java.util.List;

import java.util.Properties;

import java.util.concurrent.TimeUnit;


import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.Keys;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.Assert;

import org.testng.annotations.Test;


import com.ibm.test.LoginPage;


public class AdminPage {

	WebDriverWait wait;
	//WebDriver driver=new FirefoxDriver();
	WebDriver driver;
	JavascriptExecutor js=(JavascriptExecutor)driver;

	public AdminPage(WebDriver driver, WebDriverWait wait)
	{
		PageFactory.initElements(driver, this);
		this.driver=driver;
		this.wait=wait;
	}





@FindBy(xpath="(//a[text()='New Arrival'])[2]")
	WebElement newArrivalbutton;

	@FindBy(xpath="(//*[text()='Add To Cart'])[1]")
	WebElement addtoCartProductOne;
```

```
@FindBy(xpath="(//*[text()='Add To Cart'])[2]")
WebElement addtoCartProductTwo;

@FindBy(id="bigCart")
WebElement AddtoCartButton;

@FindBy(xpath="//*[text()='Check Out']")
WebElement clickOnCheckOut;

@FindBy(xpath="(//button[text()='Next'])[1]")
WebElement clickOnNextbutton;

@FindBy(xpath="(//*[@id='pnum_err'])[1]")
WebElement phoneNumErr;

@FindBy(xpath="(//*[@id='pword_err'])[1]")
WebElement pwordErr;

@FindBy(xpath="(//*[@id='pnum'])[1]")
WebElement enterPhonenum;

@FindBy(xpath="(//*[@id='pword'])[1]")
WebElement enterPword;

@FindBy(xpath="(//*[@id='name'])[1]")
WebElement enterName;

@FindBy(xpath="(//*[@id='email'])[1]")
WebElement enterEmailAddr;

@FindBy(id="address")
WebElement enterAddrInChek;

@FindBy(id="city")
WebElement selectCityInChek;

@FindBy(id="pincode")
WebElement enterPincode;

@FindBy(id="type")
WebElement selectType;

@FindBy(xpath="(//input[@type='checkbox'])[1]")
WebElement selectCheckBox;

@FindBy(xpath="(//*[text()='Confirm Order'])[1]")
WebElement clickOnConfirmOrder;

@FindBy(xpath="//*[text()='Continue to payment']")
WebElement clickOnConfirmPay;

@FindBy(xpath="//*[@class='col-md-12 text-center']")
WebElement verifyOrderPlacedText;

public void ClickOnConfirmPayment()
```

```
{
	clickOnConfirmPay.click();
}

public String VerifyOrderPlacetext()
{
	String stext=verifyOrderPlacedText.getText().trim();
	return stext;
}

public void ConfirmOrder()
{
	clickOnConfirmOrder.click();
}

public void SelectCheckBox()
{
	selectCheckBox.click();
}

public void SelectResiType()
{
	Select selType=new Select(selectType);
	selType.selectByValue("Home");
}

public void EnterPincodeInChek(String pincode)
{
	enterPincode.clear();
	enterPincode.sendKeys(pincode);
}

public void SelectCityInChek()
{
	Select selCity=new Select(selectCityInChek);
	selCity.selectByValue("KORAMANGALA");
}

public void enterAddrinChek(String enterAddrInCheck)
{
	enterAddrInChek.clear();
	enterAddrInChek.sendKeys(enterAddrInCheck);
}

public void EnterEmailAddr(String emailAddrInChek)
{
	enterEmailAddr.clear();
	enterEmailAddr.sendKeys(emailAddrInChek);
}

public void EnterNameinCheckOut(String checkOutName)
{
	enterName.clear();
	enterName.sendKeys(checkOutName);
}
```

```
public void EnterPword(String loginpword)
{
        enterPword.clear();
        enterPword.sendKeys(loginpword);
}

public void EnterPhnNum(String phoneNum)
{
        enterPhonenum.clear();
        enterPhonenum.sendKeys(phoneNum);
}

public void EnterPworddummy(String loginpwordDummy)
{
        enterPword.sendKeys(loginpwordDummy);
}

public void EnterPhnNumdummy(String phoneNumDummy)
{
        enterPhonenum.sendKeys(phoneNumDummy);
}

public String PwordErrMsg()
{
        String pwordErrMsgg=pwordErr.getText();
        return pwordErrMsgg;

}

public String PNumErrorMsg()
{
        String pnumErr=phoneNumErr.getText();
        return pnumErr;
}

public void ClickOnNextButton()
{
        clickOnNextbutton.click();
}

public void ClickOnCheckOut()
{
        clickOnCheckOut.click();
}

public void ClickOnAddtoCardButton()
{
        AddtoCartButton.click();
}

public void ClickOnAddtoCardOne()
{
        addtoCartProductOne.click();
}
```

```
    public void ClickOnAddtoCardTwo()
    {
        addtoCartProductTwo.click();
    }

    public void NewArrivalbutton()
    {
        newArrivalbutton.click();
    }
}
```

---

Data.properties file

```
url=https://atozgroceries.com/admin
userurl=https://atozgroceries.com/
username=demo@atozgroceries.com
password=456789
bannertitle=Banner | Admin Panel - Powered By A&S
editStoreNameMsg=Success: You have successfully updated Store!
newStoreName=ZtoA groceries Store
weightClassheader=Weight Class
validationMessage=Please fill out this field.
addWeightClassSuccessMsg=Success: You have successfully added weight class!
userNameinUser=PrivateUser
name=Customer
emailAddress=customerValue@gmail.com
passwordforUser=Dicaprio9
confirmpassword=Dicaprio9
addUserSuccMsg=Success: You have successfully added user!
deleteErrorMessage=You have successfully deleted data!
phoneNumErrorMesg=Phone Number is required!
passwordErrorMesg=Password is required!
invalidpNumErrorMesg=Invalid or Incorrect phone number!
phoneNumDummy=demouser
loginpwordDummy=castropic
phoneNum=9912312312
loginpword=123456789
checkOutName=Groceries
emailAddrInChek=customerCare@gmail.com
enterAddrInCheck=H.no 100, Patel Road, Secunderabad, Hyderabad.
pincode=560034
OrderConfirmMessage=Thank you! Your order has been placed successfully.
```

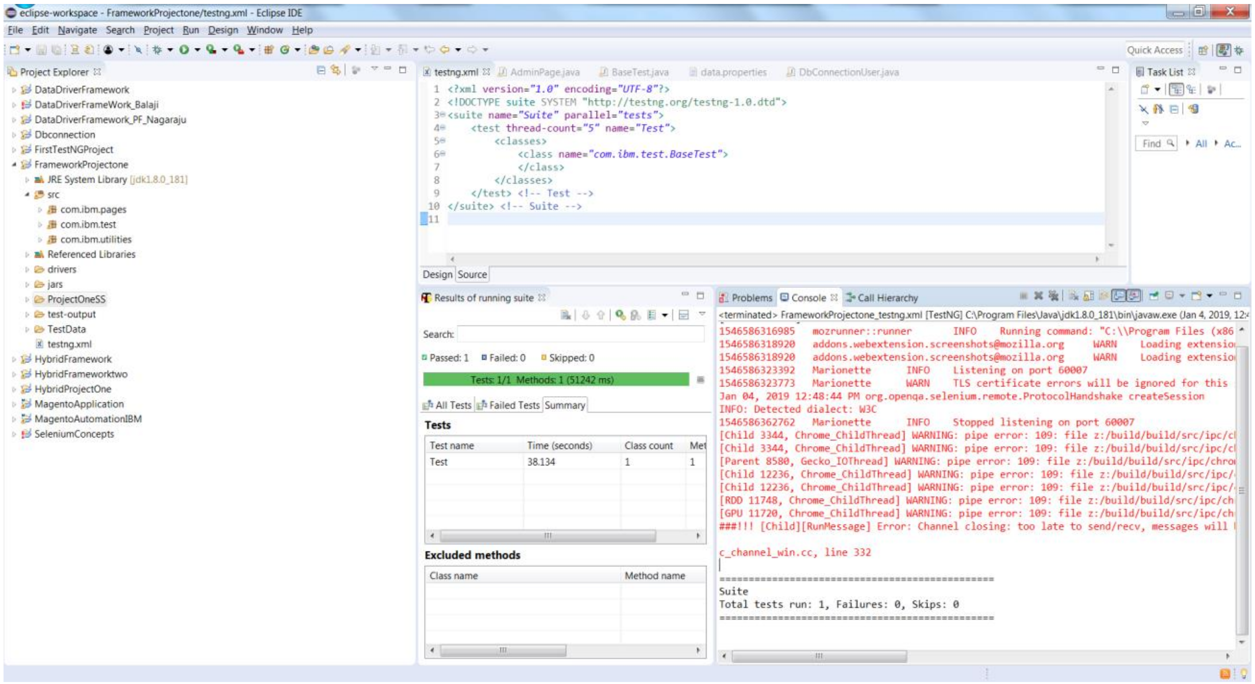
---

Testng.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite" parallel="tests">
    <test thread-count="5" name="Test">
```

```
            <classes>
                  <class name="com.ibm.test.BaseTest">
                  </class>
            </classes>
      </test> <!-- Test -->
</suite> <!-- Suite -->
```

Eclipse output screen shot



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite" parallel="tests">
    <test thread-count="5" name="Test">
        <classes>
            <class name="com.ibm.test.BaseTest">
            </class>
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

Search:

Passed: 1 Failed: 0 Skipped: 0

Tests: 1/1 Methods: 1 (51242 ms)

All Tests | Failed Tests | Summary

**Tests**

| Test name | Time (seconds) | Class count | Met |
|---|---|---|---|
| Test | 38.134 | 1 | 1 |

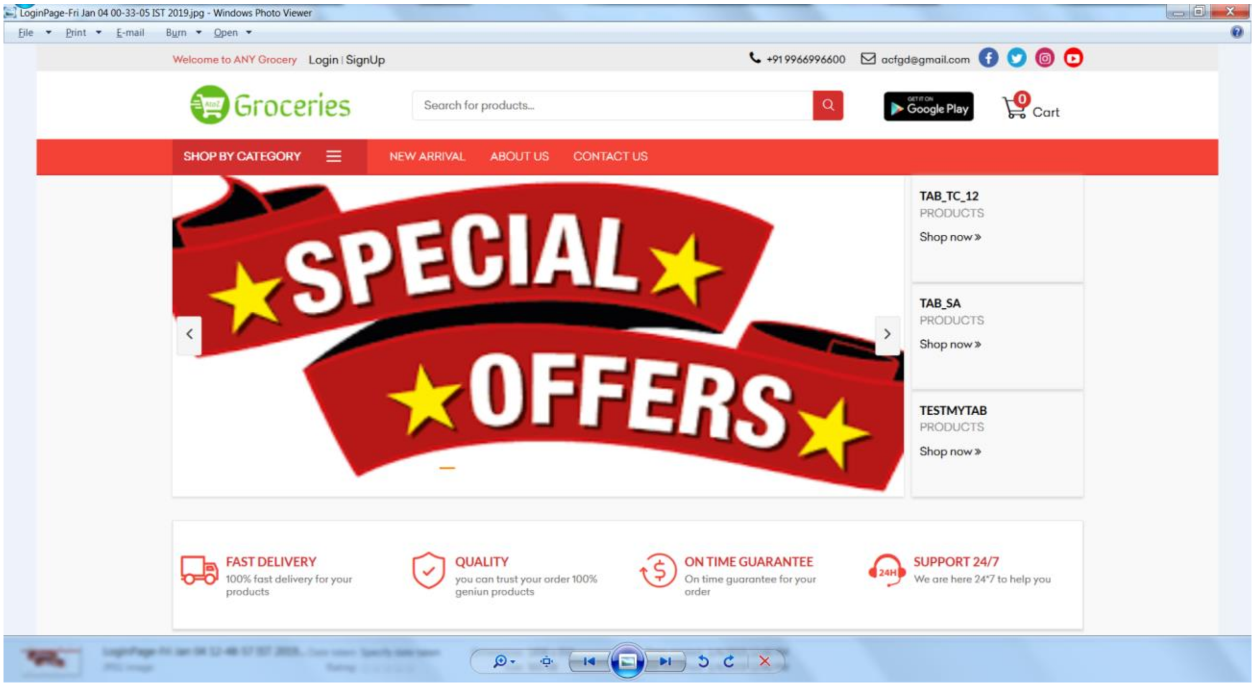
**Excluded methods**

| Class name | Method name |
|---|---|

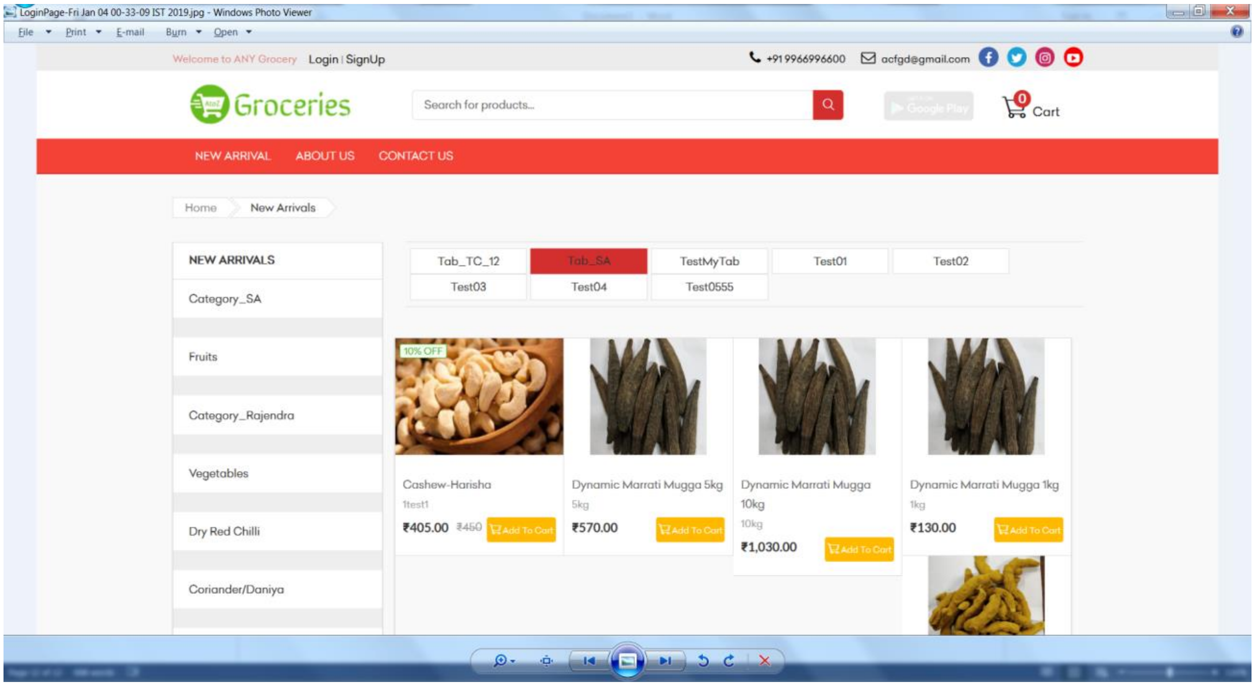
```
<terminated> FrameworkProjectone_testng.xml [TestNG] C:\Program Files\Java\jdk1.8.0_181\bin\javaw.exe (Jan 4, 2019, 12:
1546586316985   mozrunner::runner       INFO    Running command: "C:\\Program Files (x86
1546586318920   addons.webextension.screenshots@mozilla.org     WARN    Loading extensio
1546586318920   addons.webextension.screenshots@mozilla.org     WARN    Loading extensio
1546586323392   Marionette      INFO    Listening on port 60007
1546586323773   Marionette      WARN    TLS certificate errors will be ignored for this
Jan 04, 2019 12:48:44 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
1546586362762   Marionette      INFO    Stopped listening on port 60007
[Child 3344, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/c
[Child 3344, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/c
[Parent 8580, Gecko_IOThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/chro
[Child 12236, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/
[Child 12236, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/
[RDD 11748, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/ch
[GPU 11720, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/ch
###!!! [Child][RunMessage] Error: Channel closing: too late to send/recv, messages will

c_channel_win.cc, line 332

===============================================
Suite
Total tests run: 1, Failures: 0, Skips: 0
===============================================
```

Browser screen shot for each step

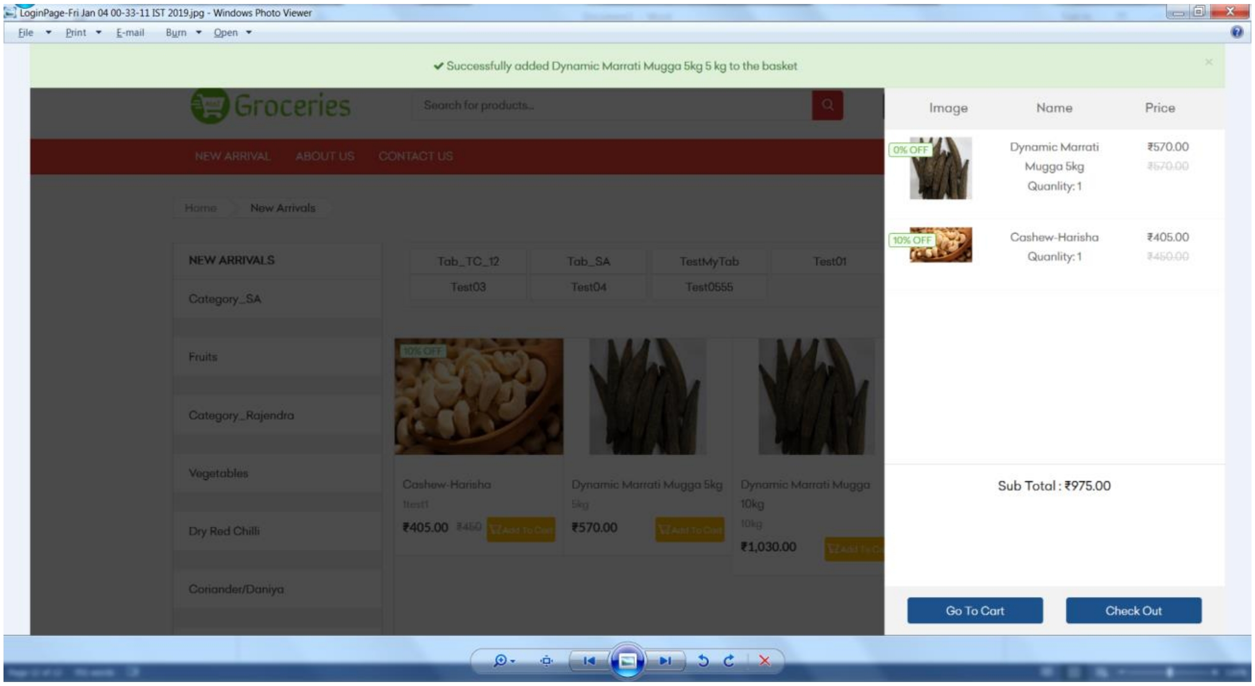
Main user screen


LoginPage-Fri Jan 04 00-33-05 IST 2019.jpg - Windows Photo Viewer
File Print E-mail Burn Open
Welcome to ANY Grocery Login | SignUp
+91 9966996600 acfgd@gmail.com
Groceries
Search for products...
Google Play
0 Cart
SHOP BY CATEGORY
NEW ARRIVAL
ABOUT US
CONTACT US
SPECIAL OFFERS
TAB_TC_12
PRODUCTS
Shop now »
TAB_SA
PRODUCTS
Shop now »
TESTMYTAB
PRODUCTS
Shop now »
FAST DELIVERY
100% fast delivery for your products
QUALITY
you can trust your order 100% geniun products
ON TIME GUARANTEE
On time guarantee for your order
SUPPORT 24/7
We are here 24*7 to help you

Selecting New Arrivals



Add products to cart

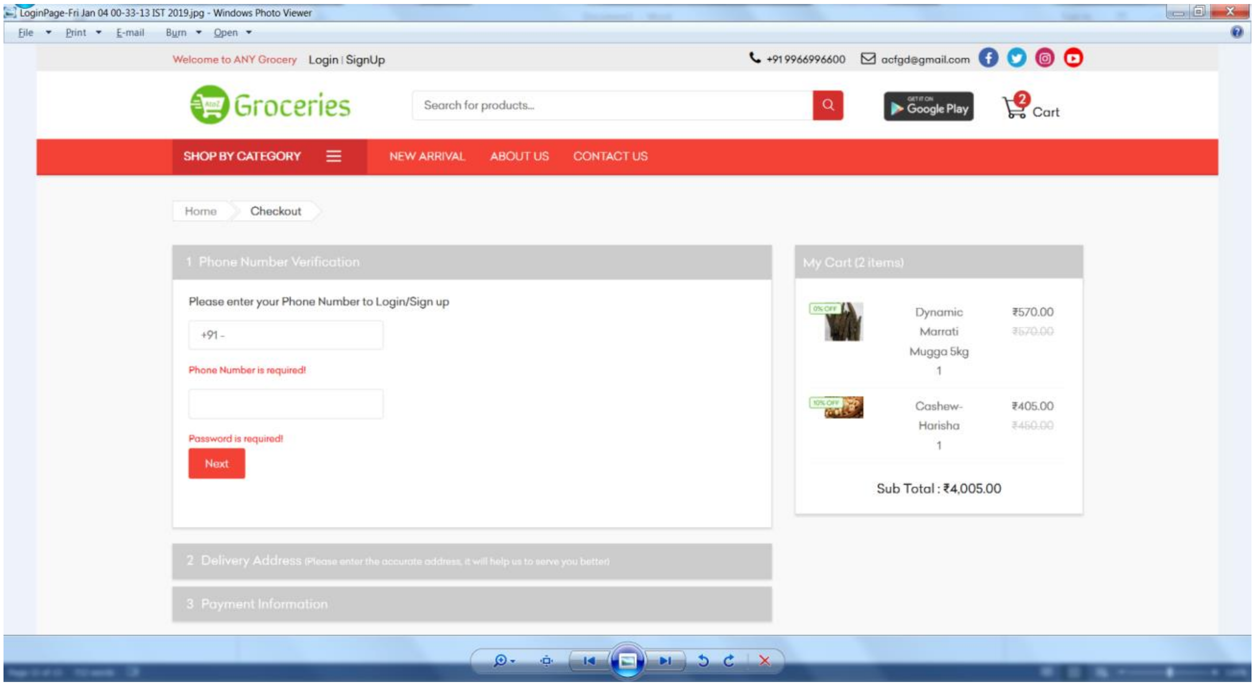

LoginPage-Fri Jan 04 00-33-11 IST 2019.jpg - Windows Photo Viewer

✔ Successfully added Dynamic Marrati Mugga 5kg 5 kg to the basket

| Image | Name | Price |
|---|---|---|
| 0% OFF | Dynamic Marrati Mugga 5kg<br>Quanlity: 1 | ₹570.00<br>~~₹570.00~~ |
| 10% OFF | Cashew-Harisha<br>Quanlity: 1 | ₹405.00<br>~~₹450.00~~ |

Sub Total : ₹975.00

Verifying validation on Phone And Password fields


LoginPage-Fri Jan 04 00-33-13 IST 2019.jpg - Windows Photo Viewer

Welcome to ANY Grocery Login | SignUp

+91 9966996600 acfgd@gmail.com

Home Checkout

1 Phone Number Verification

Please enter your Phone Number to Login/Sign up

+91 -

Phone Number is required!

Password is required!

Next

2 Delivery Address (Please enter the accurate address, it will help us to serve you better)

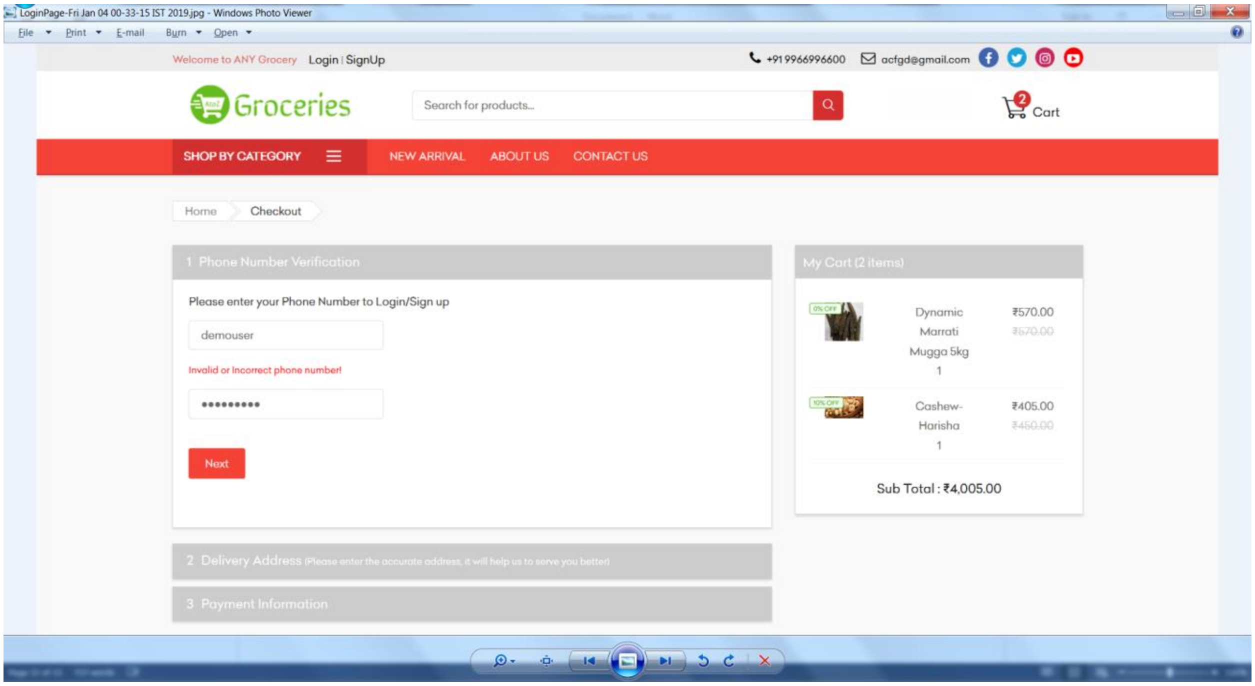
3 Payment Information

My Cart (2 items)

| | | |
|---|---|---|
| 0% OFF | Dynamic Marrati Mugga 5kg<br>1 | ₹570.00<br>~~₹570.00~~ |
| 10% OFF | Cashew-Harisha<br>1 | ₹405.00<br>~~₹450.00~~ |

Sub Total : ₹4,005.00

Verifying incorrect Phone num validation



1 Phone Number Verification

Please enter your Phone Number to Login/Sign up

demouser

Invalid or Incorrect phone number!

•••••••••

2 Delivery Address (Please enter the accurate address, it will help us to serve you better)

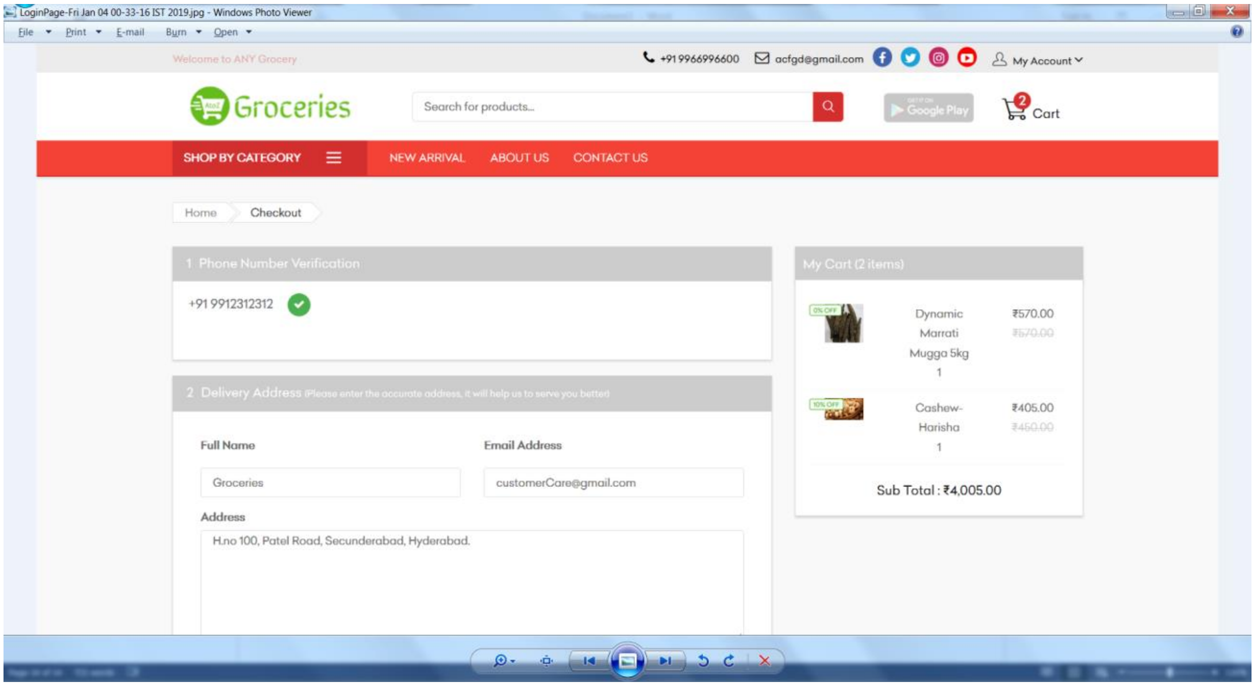
3 Payment Information

My Cart (2 items)

| | | |
|---|---|---|
| 0% OFF | Dynamic Marrati Mugga 5kg<br>1 | ₹570.00<br>~~₹570.00~~ |
| 10% OFF | Cashew-Harisha<br>1 | ₹405.00<br>~~₹450.00~~ |

Sub Total : ₹4,005.00

Login with valid credentials



1 Phone Number Verification

+91 9912312312 ✓

2 Delivery Address (Please enter the accurate address, it will help us to serve you better)

| | |
|---|---|
| Full Name | Email Address |
| Groceries | customerCare@gmail.com |

Address

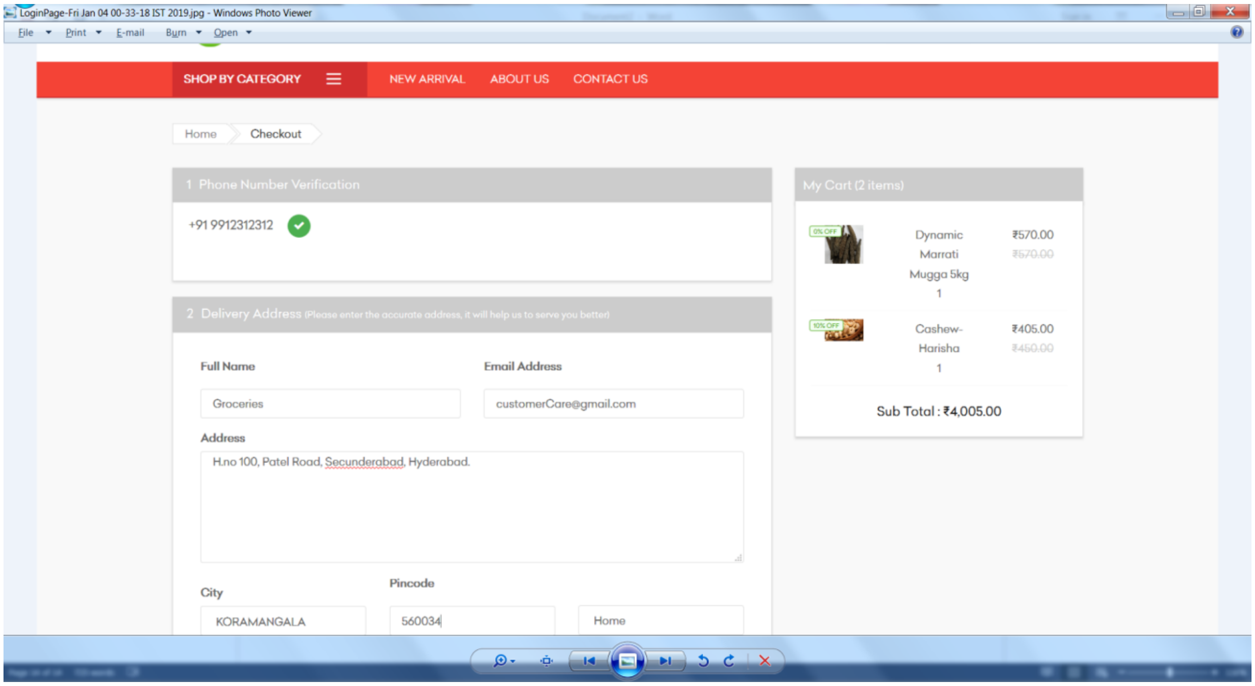
H.no 100, Patel Road, Secunderabad, Hyderabad.

My Cart (2 items)

| | | |
|---|---|---|
| 0% OFF | Dynamic Marrati Mugga 5kg<br>1 | ₹570.00<br>~~₹570.00~~ |
| 10% OFF | Cashew-Harisha<br>1 | ₹405.00<br>~~₹450.00~~ |

Sub Total : ₹4,005.00

Enter details


LoginPage-Fri Jan 04 00-33-18 IST 2019.jpg - Windows Photo Viewer
File Print E-mail Burn Open
SHOP BY CATEGORY
NEW ARRIVAL
ABOUT US
CONTACT US
Home
Checkout

1 Phone Number Verification

+91 9912312312 ✓

2 Delivery Address (Please enter the accurate address, it will help us to serve you better)

Full Name: Groceries

Email Address: customerCare@gmail.com

Address: H.no 100, Patel Road, Secunderabad, Hyderabad.

City: KORAMANGALA

Pincode: 560034

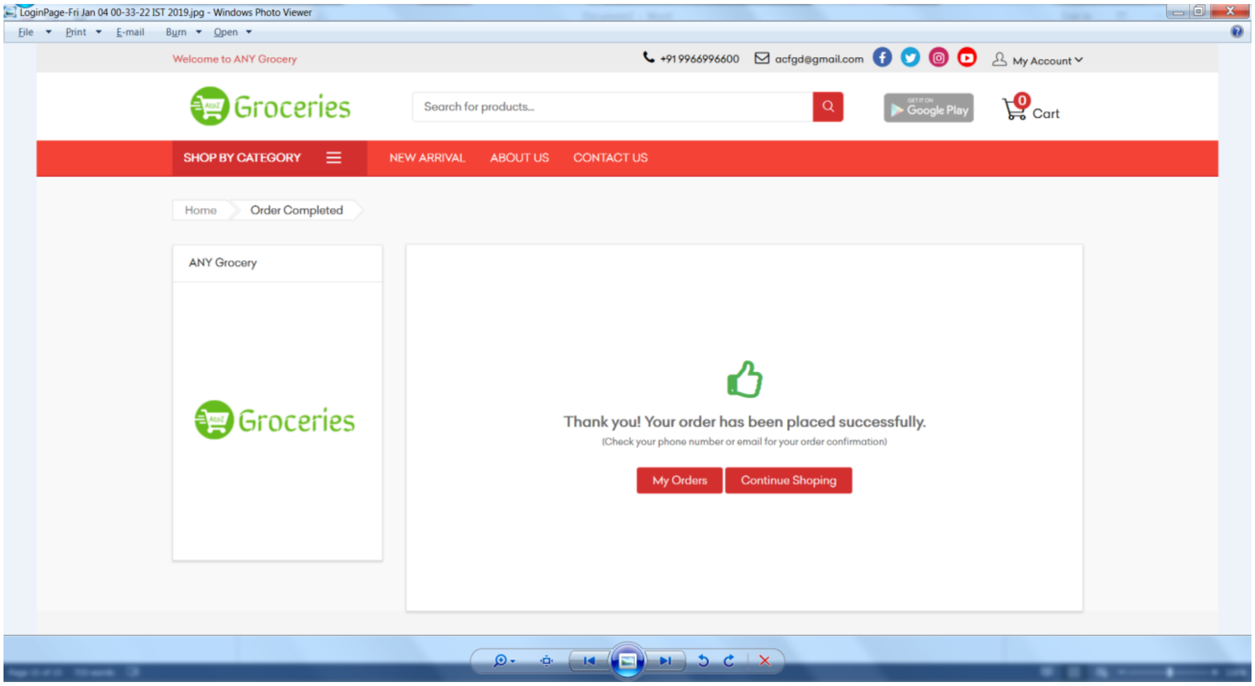
Home

My Cart (2 items)

| | | |
|---|---|---|
| 0% OFF | Dynamic Marrati Mugga 5kg<br>1 | ₹570.00<br>~~₹570.00~~ |
| 10% OFF | Cashew-Harisha<br>1 | ₹405.00<br>~~₹450.00~~ |

Sub Total : ₹4,005.00

Complete transaction to confirm order

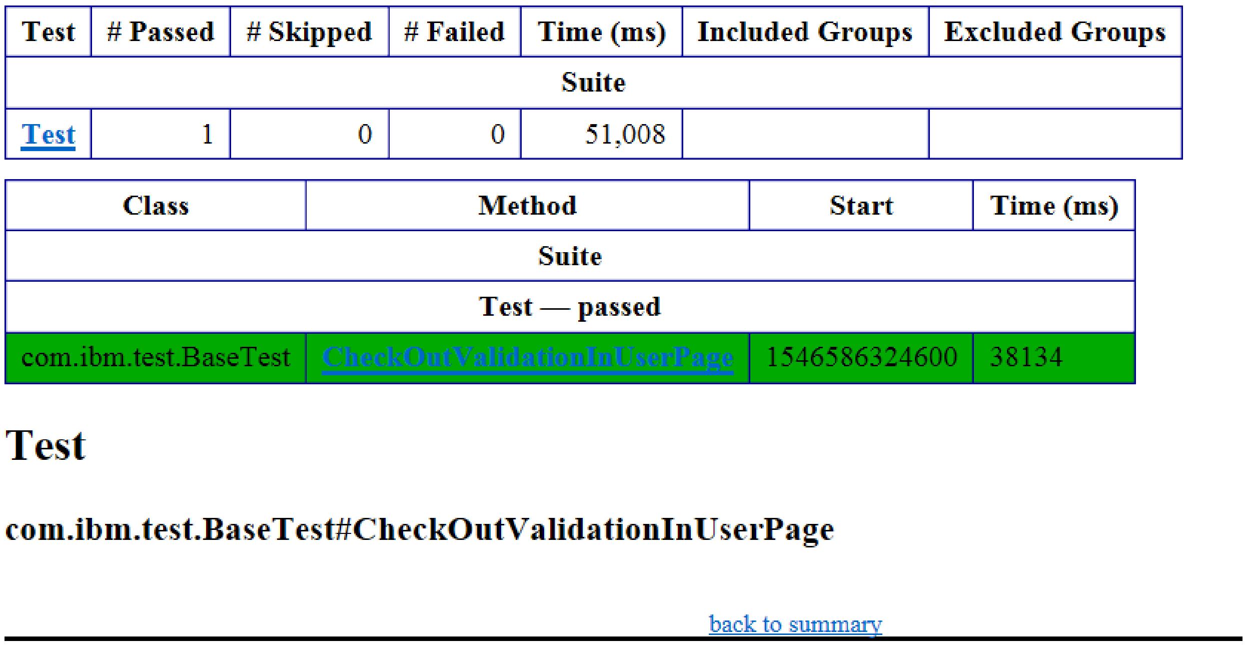


Thank you! Your order has been placed successfully.

(Check your phone number or email for your order confirmation)

Test Case passed

Emailable report



| Test | # Passed | # Skipped | # Failed | Time (ms) | Included Groups | Excluded Groups |
|---|---|---|---|---|---|---|
| Suite | | | | | | |
| Test | 1 | 0 | 0 | 51,008 | | |

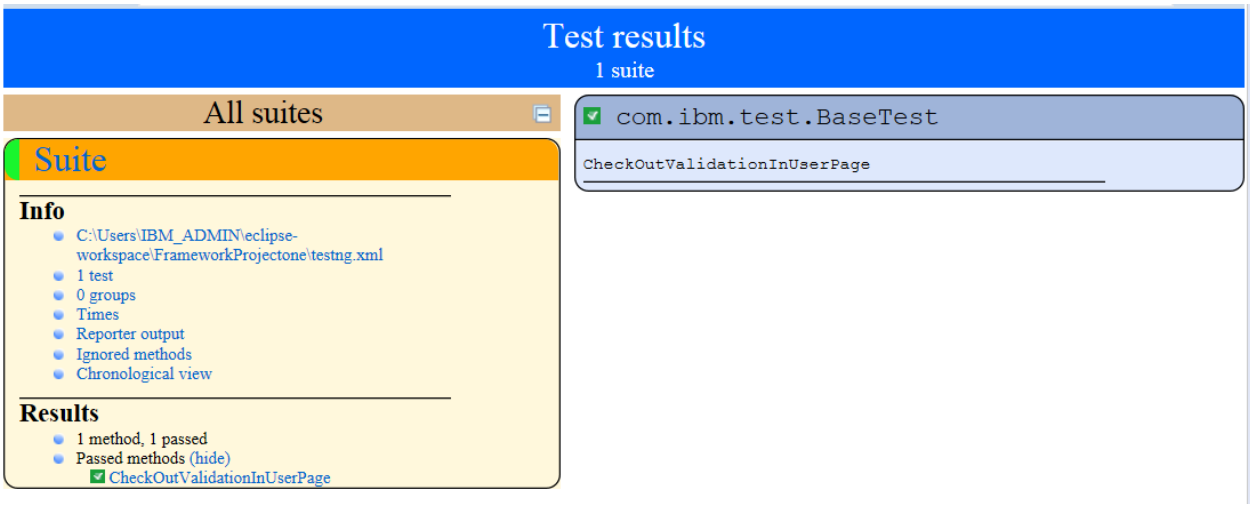
| Class | Method | Start | Time (ms) |
|---|---|---|---|
| Suite | | | |
| Test — passed | | | |
| com.ibm.test.BaseTest | CheckOutValidationInUserPage | 1546586324600 | 38134 |

# Test

### com.ibm.test.BaseTest#CheckOutValidationInUserPage

back to summary

Index.html report



Test results

1 suite

All suites

Suite

**Info**

- C:\Users\IBM_ADMIN\eclipse-workspace\FrameworkProjectone\testng.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

**Results**

- 1 method, 1 passed
- Passed methods (hide)
  - ☑ CheckOutValidationInUserPage

☑ com.ibm.test.BaseTest

CheckOutValidationInUserPage

Working as expected and it is pass