

Test Objective:

The objective is to automate the below testcase using selenium WebDriver and framework concepts.

Testcase: Tc08

Process: Add a product to cart and modify quantity

Steps to Execute:

Step1: SignUp on user page <https://atozgroceries.com>

Step2: Add anyone product to cart

Step3: Go to Cart

Step4: Change the quantity to 30 and validate the price

Expected Results:

Validate that the quantity is updated under cart.

Validate the price of the product should be unit price*30

Source code:

File1: AddtoCart.java

```
package com.ibm.groceries;

import java.io.IOException;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Test;

import com.ibm.groceriespages.GroceriesUserPage;
import com.ibm.groceriespages.ShoppingCartPage;
import com.ibm.initialization.WebDriverLaunch;
import com.ibm.utilities.GetScreenshot;

public class AddtoCart extends WebDriverLaunch {

    @Test(priority = 1, testName = "AddProductToCart", groups = "low")
    public void addProductToCart() throws IOException, InterruptedException {
        String userPage = data.get("userPageUrl");
        String fullName = data.get("fullName");
        String phoneNum = data.get("phoneNum");
        String passwd = data.get("passwd");
        String confirmPassword = data.get("confirmPassword");
        String expMessage = data.get("expectedShoppingCartMsg");
        String quantity = data.get("quantity");
        // To launch Groceries user page
        driver.get(userPage);

        GroceriesUserPage userpage = new GroceriesUserPage(driver, wait);
```

```

        userpage.signUp(fullName, phoneNum, passwd, confirmPassword);
        GetScreenshot screen = new GetScreenshot();
        screen.takeScreenshot(driver);
        userpage.addProductToCart();
        screen.takeScreenshot(driver);

        ShoppingCartPage shop = new ShoppingCartPage(driver, wait);
        String pageSource = shop.updateQuantity(quantity);
        screen.takeScreenshot(driver);

        try {
            if (pageSource.contains(expMessage)) {
                Reporter.Log(expMessage);
                // Assertion on updated quantity value
                Assert.assertTrue(shop.verifyQuantity(quantity));

                // Assertion on subtotal value
                Assert.assertTrue(shop.verifySubTotal());
            }
        }

        catch (Exception e) {
            Assert.fail();
        }
    }
}

```

File2: GroceriesUserPage.java

```

package com.ibm.groceriespages;

import java.io.IOException;
import org.openqa.selenium.Alert;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;
import com.ibm.utilities.GetScreenshot;

public class GroceriesUserPage {

    @FindBy(xpath = "//input[@placeholder='Search for products...']")
    WebElement searchEle;

    @FindBy(xpath = "(//div[@class='input-group']/descendant::input[1])[2]")
    WebElement searchEle2;
}

```

```

@FindBy(xpath = "//div[@id='searchproducts-div']/descendant::a[1]")
WebElement productNewLink;

// To locate email link on user page
@FindBy(xpath = "(//div[@class='header-mid-right-content']/descendant::a[1])[2]")
WebElement emailLink;

// To locate phone link on user page
@FindBy(xpath = "(//div[@class='header-mid-right-content']/descendant::a[1])[1]")
WebElement phoneLink;

// To locate SignUp link
@FindBy(xpath = "//a[text()='SignUp']")
WebElement signupEle;

// To locate fullname text box
@FindBy(xpath = "//input[@id='name']")
WebElement fullnameEle;

// To locate phonnumber text box on signupuser page
@FindBy(xpath = "//input[@id='pnum']")
WebElement phonenumEle;

// To locate password element on signup user page
@FindBy(xpath = "//input[@id='password']")
WebElement passwordEle;

// To locate confirm password text box
@FindBy(xpath = "//input[@id='cpassword']")
WebElement confirmpasswordEle;

// TO locate agree terms check box
@FindBy(xpath = "//input[@id='tccheckbox']")
WebElement agreeCheckEle;

// To locate sign up button
@FindBy(xpath = "//button[@id='mem_signup']")
WebElement signupButton;

// To locate addto cart link
@FindBy(xpath = "//a[@id='addtocart_cartbtn336']")
WebElement addcartEle;

// To locate Cart icon link
// @FindBy(xpath="//a[text()=' Cart']")
// WebElement cartEle;

@FindBy(xpath = "//div[@class='header-bottom-right']/descendant::a[1]")
WebElement cartEle;

// To locate Go to Cart icon link
@FindBy(xpath = "//a[text()='Go To Cart']")
WebElement gotocartEle;

```

```

WebDriverWait wait;
WebDriver driver;

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

public String searchProduct(String proudctName) throws InterruptedException {

    searchEle.sendKeys(proudctName);
    searchEle.click();

    // To enter product name in search text box of popup
    searchEle2.sendKeys(proudctName);
    Thread.sleep(10000);
    productNewLink.click();

    return driver.getPageSource();
}

// To verify the new address on user page
public String verifyAddress() {
    return driver.getPageSource();
}

// To verify the new email link is on user page
public boolean verifyEmail(String newemail) {

    if (emailLink.getText().contains(newemail))

        return true;
    else
        return false;
}

// To verify the new phone number is user page
public boolean verifyPhone(String newphone) {

    if (phoneLink.getText().contains(newphone))

        return true;
    else
        return false;
}

public void signUp(String fullName, String phoneNum, String password, String
confirmPassword)
    throws InterruptedException {

    // To click signUp link

```

```

        signupEle.click();
        // To enter the value for full name
        fullnameEle.sendKeys(fullName);
        // To enter the value for phone number
        phonenumberEle.sendKeys(phoneNum);
        // To enter the value for password
        passwordEle.sendKeys(password);
        // To enter the value for confirm password
        confirmpasswordEle.sendKeys(confirmPassword);

        // To click on check box for agree terms
        agreeCheckEle.click();

        // To click on SignUp button
        signupButton.click();

        // To click on Ok button of Alert box after signup
        Alert alertBox = driver.switchTo().alert();
        String text = alertBox.getText();
        Thread.sleep(2000);
        alertBox.accept();
    }

    public void addProductToCart() throws InterruptedException, IOException {
        GetScreenshot screenObj = new GetScreenshot();
        // To scroll down using key down
        Actions actions = new Actions(driver);
        for (int i = 1; i <= 4; i++)
            actions.sendKeys(Keys.ARROW_DOWN).build().perform();

        // To click on Add to Cart link
        addcartEle.click();

        // To click on Cart link at top right corner
        cartEle.click();

        screenObj.takeScreenshot(driver);

        // To click on Got TO Cart link
        gotocartEle.click();

        screenObj.takeScreenshot(driver);
    }
}

```

File3: ShoppingCartPage.java

```

package com.ibm.groceriespages;

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

```

```

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

public class ShoppingCartPage {

    @FindBy(xpath = "//input[@title='Qty']")
    WebElement qtyEle;

    @FindBy(xpath = "//button[text()='Update Shoping Cart']")
    WebElement updatecartEle;

    @FindBy(xpath = "//table[@class='shop_table cart']/tbody/tr[1]/td[3]/span")
    WebElement unitpriceEle;

    @FindBy(xpath = "//table[@class='shop_table cart']/tbody/tr[1]/td[5]/span")
    WebElement subtotalEle;

    WebDriverWait wait;
    WebDriver driver;

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

    public String updateQuantity(String quantity) {

        qtyEle.clear();
        // To assign quantity value
        qtyEle.sendKeys(quantity);

        // TO click on update shopping cart button
        updatecartEle.click();
        return driver.getPageSource();
    }

    public boolean verifyQuantity(String quantity) {
        String updatedqtyValue = qtyEle.getAttribute("value");
        if (updatedqtyValue.equals(quantity))
            return true;
        else
            return false;
    }

    // To verify the subtotal value
    public boolean verifySubTotal()

    {

        String unitprice = unitpriceEle.getText();
        String updatedqtyValue = qtyEle.getAttribute("value");
        String subTotal = subtotalEle.getText();
    }
}

```

```

        // To remove the first character ? and decimal places
        unitprice = unitprice.trim().substring(1).replace(".00", "");
        subTotal = subTotal.trim().substring(1).replace(".00", "");

        int intUnitPrice = Integer.parseInt(unitprice);
        int intSubTotal = Integer.parseInt(subTotal);
        int intQuantiy = Integer.parseInt(updatedqtyValue);

        System.out.println("unit price " + intUnitPrice);
        System.out.println("quantity " + intQuantiy);
        System.out.println("sub total " + intSubTotal);

        if (intSubTotal == (intQuantiy * intUnitPrice))
            return true;
        else
            return false;
    }
}

```

File4: WebDriverLaunch.java

```

package com.ibm.initialization;

import java.io.IOException;

import java.util.HashMap;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.ie.InternetExplorerDriver;

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.Optional;

import org.testng.annotations.Parameters;

import com.ibm.utilities.PropertiesFileHandler;

```

```

public class WebDriverLaunch {

    public WebDriver driver;

    public WebDriverWait wait;

    public PropertiesFileHandler propFileHandler;

    public HashMap<String, String> data;

    //Getting the keys from properties file
    //@BeforeSuite
    @BeforeSuite(groups= {"high","low"})

        public void preSetForTest() throws IOException {
            String file = "./TestData/groceries.properties";
            propFileHandler = new PropertiesFileHandler();
            data = propFileHandler.getPropertiesAsMap(file);
        }

    //@BeforeTest

    @BeforeMethod(groups= {"high","low"})
    @Parameters({"browser"})
    public void Initialization(@Optional("ff")String browser) {
        browserInitialization(browser);
        wait = new WebDriverWait(driver, 60);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }
}

```



```

//@AfterMethod
//Closing driver
@AfterMethod(groups= {"high","low"})
public void closeBrowser() {
    driver.quit();
}

//Setting path for webdriver
public void browserInitialization(String browser)
{
    switch (browser.toLowerCase()) {
        case "ff":
        case "firefox":
            System.setProperty("webdriver.gecko.driver", "./drivers/geckodriver.exe");
            driver = new FirefoxDriver();
            break;
        case "ch":
            System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
            driver = new ChromeDriver();
            break;
        case "ie":
            System.setProperty("webdriver.ie.driver", "./drivers/IEDriverServer.exe");
            driver = new InternetExplorerDriver();
            break;
        default:
            System.out.println("No browser Available "+browser);
            break;
    }
}
}

```

```
}
```

File5: PropertiesFileHandler.java

```
package com.ibm.utilities;
```

```
import java.io.File;
```

```
import java.io.FileInputStream;
```

```
import java.io.FileOutputStream;
```

```
import java.io.IOException;
```

```
import java.util.HashMap;
```

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

```
import java.util.Set;
```

```
public class PropertiesFileHandler {
```

```
    public HashMap<String, String> getPropertiesAsMap(String file) throws IOException {
```

```
        HashMap<String, String> magentoMap = new HashMap<String, String>();
```

```
        FileInputStream fileIn = new FileInputStream(file);
```

```
        Properties prop = new Properties();
```

```
        prop.load(fileIn);
```

```
        Set<Object> keysProp = prop.keySet();
```

```
        for (Object key : keysProp) {
```

```
            magentoMap.put(key.toString(), prop.getProperty(key.toString()));
```

```
        }
```

```
        prop.clear();
```

```
        return magentoMap;
```

```
    }
```

```

    public void setKeyAndValue(String file,String key,String value) throws IOException
    {
        FileInputStream fileIn = new FileInputStream(file);
        Properties prop = new Properties();
        prop.load(fileIn);

        prop.setProperty(key, value);

        FileOutputStream fOut=new FileOutputStream(file);
        prop.store(fOut, "Test Result");
        fOut.close();
        fileIn.close();
    }
}

```

File6: GetScreenshot.java

```

package com.ibm.utilities;

import java.io.File;
import java.io.IOException;
import java.util.Date;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;

public class GetScreenshot {

```

```

public void takeScreenshot(WebDriver driver)throws IOException
{

    TakesScreenshot ts=(TakesScreenshot)driver;
    File file=ts.getScreenshotAs(OutputType.FILE);
    Date date=new Date();
    String currentDate=date.toString().replaceAll(":", "-");
    FileUtils.copyFile(file,new File("./screenshots/Error_"+currentDate+".png"));

}

}

```

File7:groceries.properties

```

url=https://atozgroceries.com/admin
username=demo@atozgroceries.com
password=456789
expectedMessage=Success: You have successfully deleted data!
notifyName=DemoNotification
notifyMessage=Adding Notification
expNotificationMessage=Success: You have successfully sent push notification to your app!
imagePath=C:\\Users\\IBM_ADMIN\\eclipse-
workspace\\TestCases_DataDrivenFramework\\TestData\\globe.jpg
searchForKeyword=refund
searchDispalyMessage=Search with keyword is successful
noMatchDisplayMessage=No matching records found
searchForShipKeyword=Discount

```

prodNameNew=ProductNew100
modelNameNew=Model-New
expectedEditProductMessage=Success: You have successfully updated product!
productNotUpdated=Product update failed
userPageUrl=https://atozgroceries.com
userPageMsg=New Product found on User page
addressNew=Koramangala,Bengaluru
emailNew=info@atozgroceries.com
phoneNew=9797979797
expectedEditSettingMessage=successfully updated Store!
addressFoundMsg=new Address found on user page.
emailFoundMsg=new Email found on user page.
phoneFoundMsg=new Phone number found on user page.
fullName=srinivast
phoneNum=9701934935
passwd=dasara
confirmPassword=dasara
quantity=30
expectedShopingCartMsg=You have successfully updated cart items!

File8:testing_AddToCart.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="AddingProducttoCart">
    <parameter name="browser" value="firefox"></parameter>
    <classes>
      <class name="com.ibm.groceries.AddtoCart"/>
    </classes>
  </test> <!-- AddingProducttoCart -->
</suite> <!-- Suite -->
```

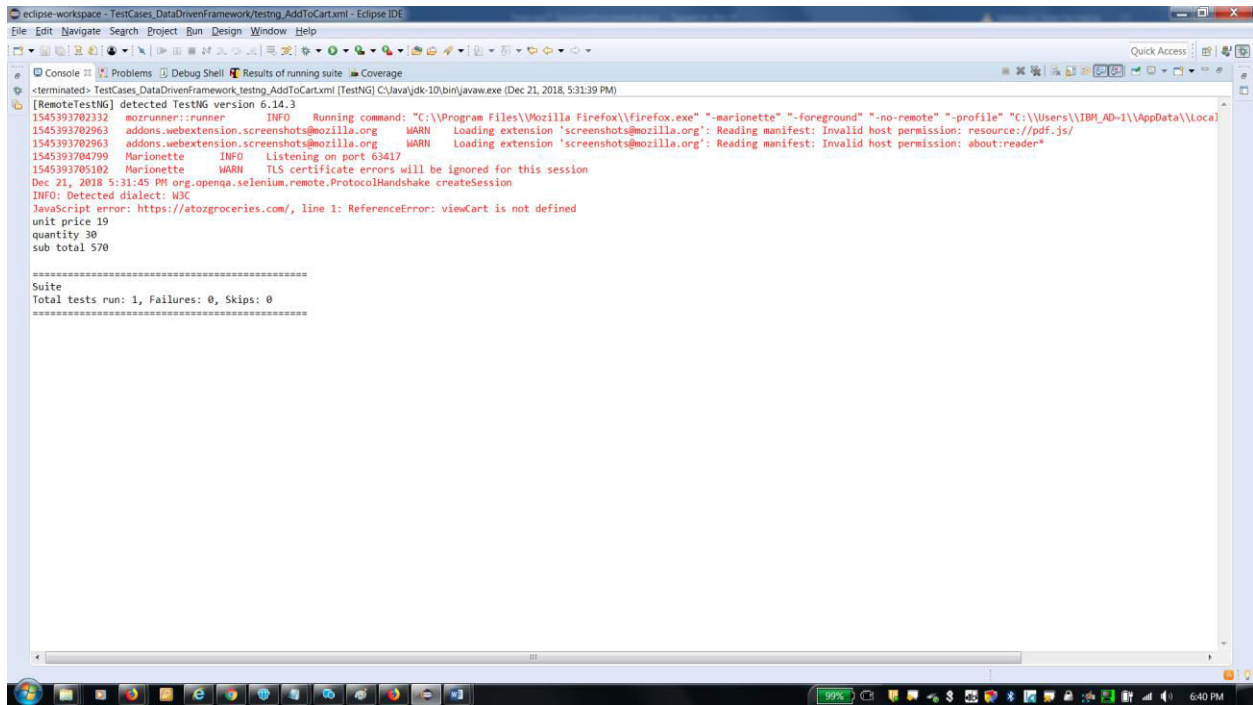
Test Result:

It is validated that quantity is updated under cart.

Validated the price of the product as unitprice * 30.

Screenshots:

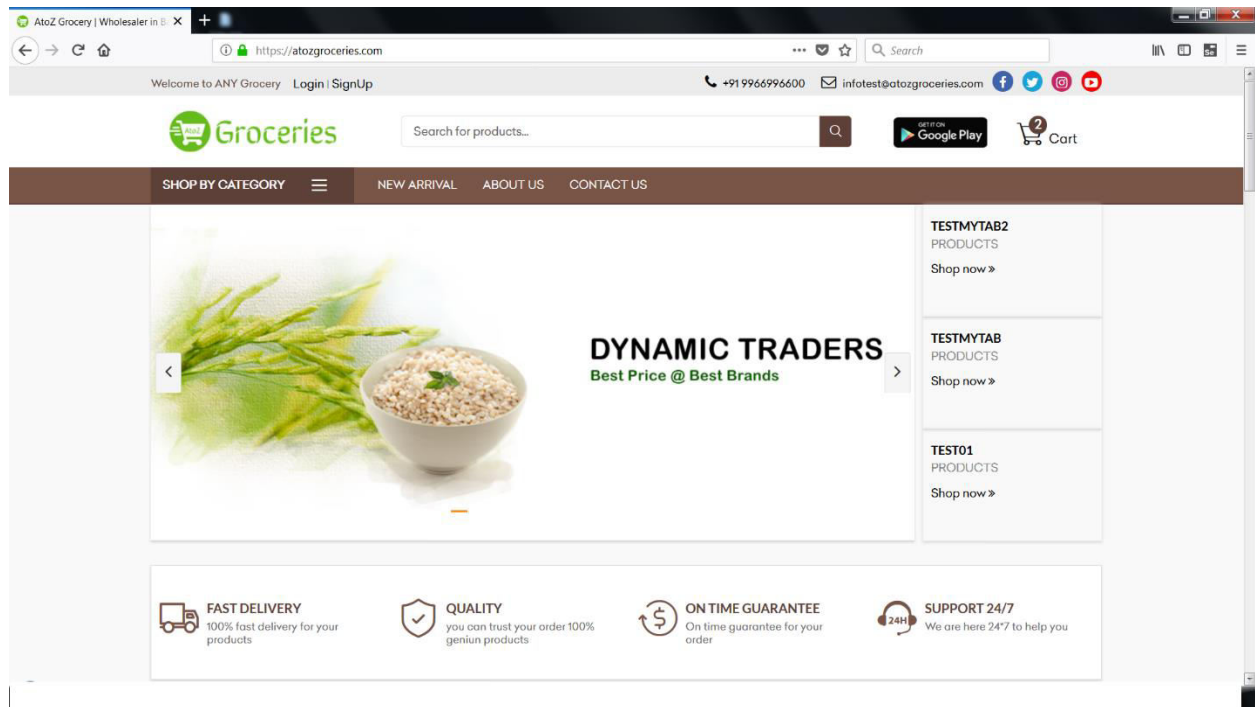
Console output:



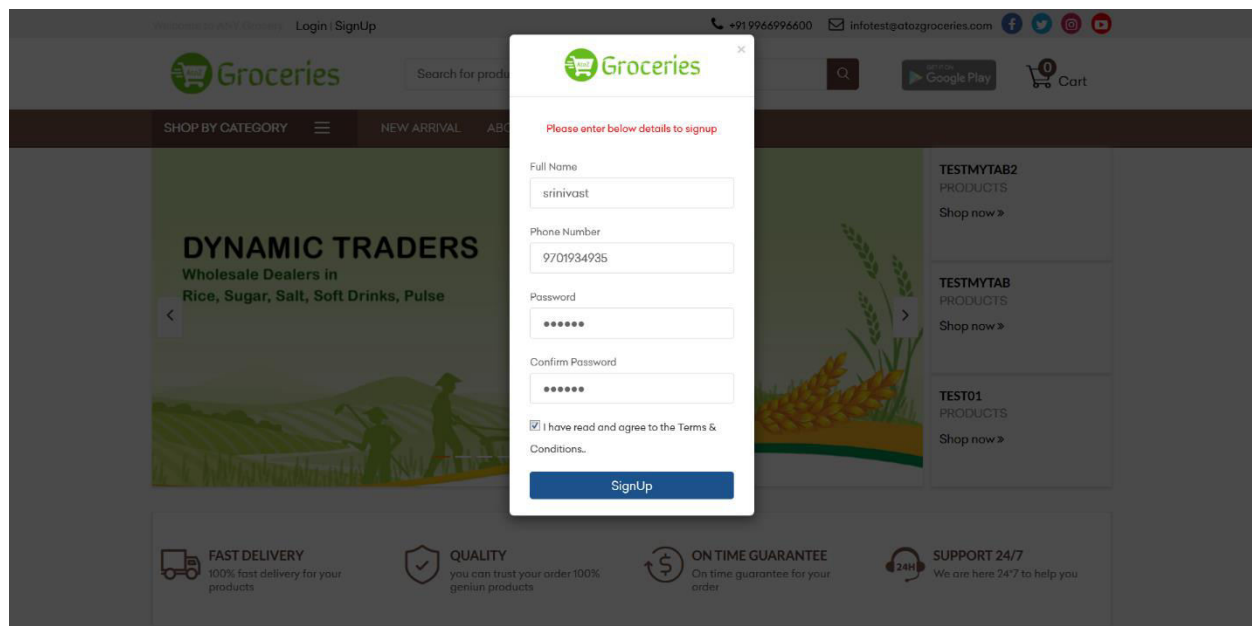
```
eclipse-workspace - TestCases_DataDrivenFramework/testing_AddToCart.xml - Eclipse IDE
File Edit Navigate Search Project Run Design Window Help
[RemoteTestNG] detected TestNG version 6.14.3
1545393702332 mozrunner:runner INFO Running command: "C:\Program Files\Mozilla Firefox\firefox.exe" "-marionette" "-foreground" "-no-remote" "-profile" "C:\Users\IBM_AD-1\AppData\Local
1545393702963 addons.webextension.screenshots@mozilla.org WARN Loading extension 'screenshots@mozilla.org': Reading manifest: Invalid host permission: resource://pdf.js/
1545393702963 addons.webextension.screenshots@mozilla.org WARN Loading extension 'screenshots@mozilla.org': Reading manifest: Invalid host permission: about:reader*
1545393704799 Marionette INFO Listening on port 63417
1545393705102 Marionette WARN TLS certificate errors will be ignored for this session
Dec 21, 2018 5:31:45 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
JavaScript error: https://atozgroceries.com/, line 1: ReferenceError: viewCart is not defined
unit price 19
quantity 30
sub total 570

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

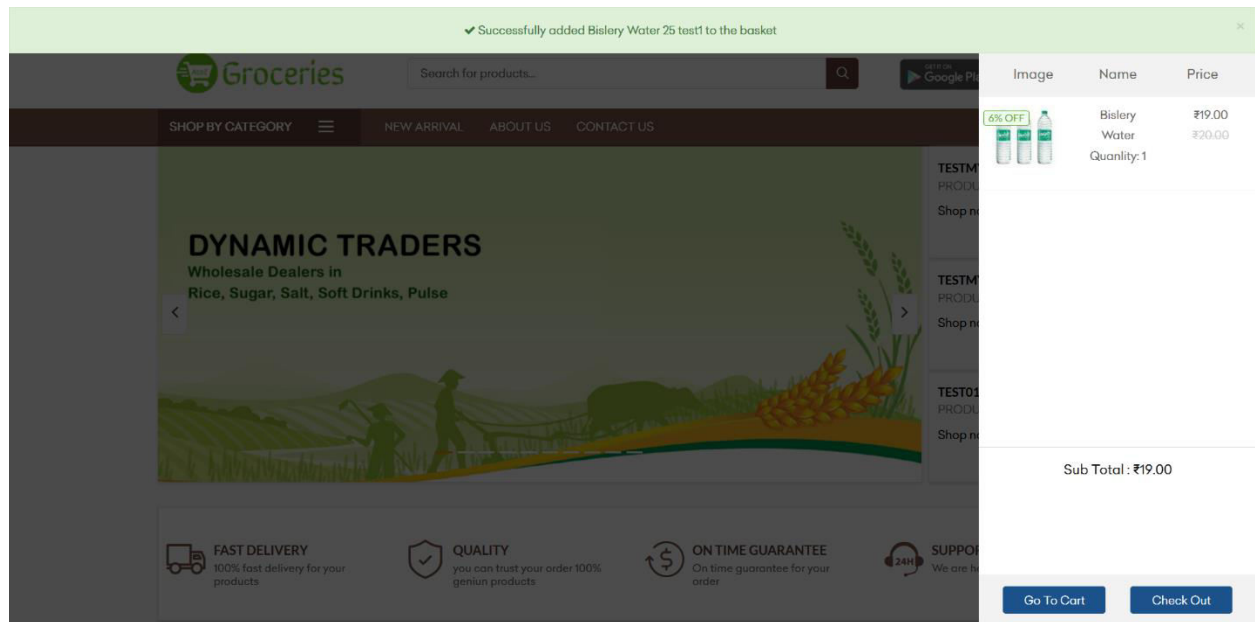
User portal is launched



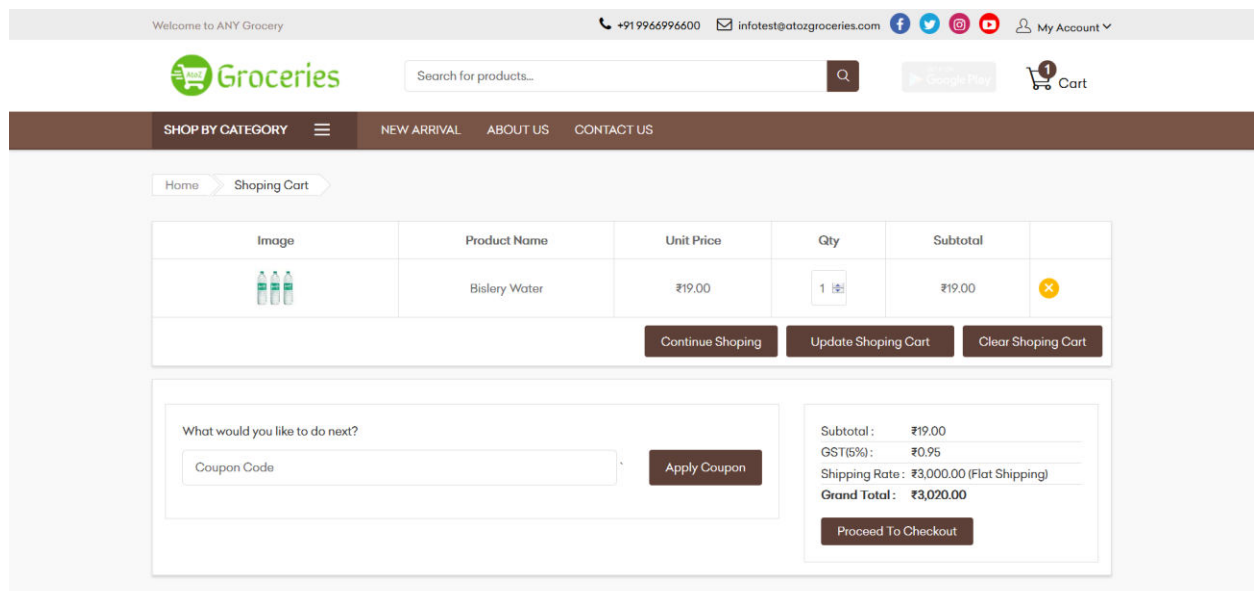
SignUp page is displayed with entered details.



Product is added to cart



Shopping cart page is displayed .



Quantity is updated and subtotal is displayed



Welcome to ANY Grocery +91 9966996600 infotest@atozgroceries.com Facebook Twitter Instagram YouTube My Account

Groceries Search for products... Google Play Cart

SHOP BY CATEGORY **NEW ARRIVAL** **ABOUT US** **CONTACT US**

Home > Shopping Cart

You have successfully updated cart items!

Image	Product Name	Unit Price	Qty	Subtotal	
	Bislery Water	₹19.00	<input type="text" value="30"/>	₹570.00	

[Continue Shopping](#) [Update Shopping Cart](#) [Clear Shopping Cart](#)

What would you like to do next?

[Apply Coupon](#)

Subtotal: ₹570.00
GST(5%): ₹28.5
Shipping Rate: ₹3,000.00 (Flat Shipping)
Grand Total: ₹3,599.00

[Proceed To Checkout](#)

Atoz Grocery | Wholesaler in | TestNG Report

file:///C:/Users/IBM_ADMIN/eclipse-workspace/TestCases_DataDrivenFramework/test-output/emailable-...

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
AddingProducttoCart	1	0	0	15,969		

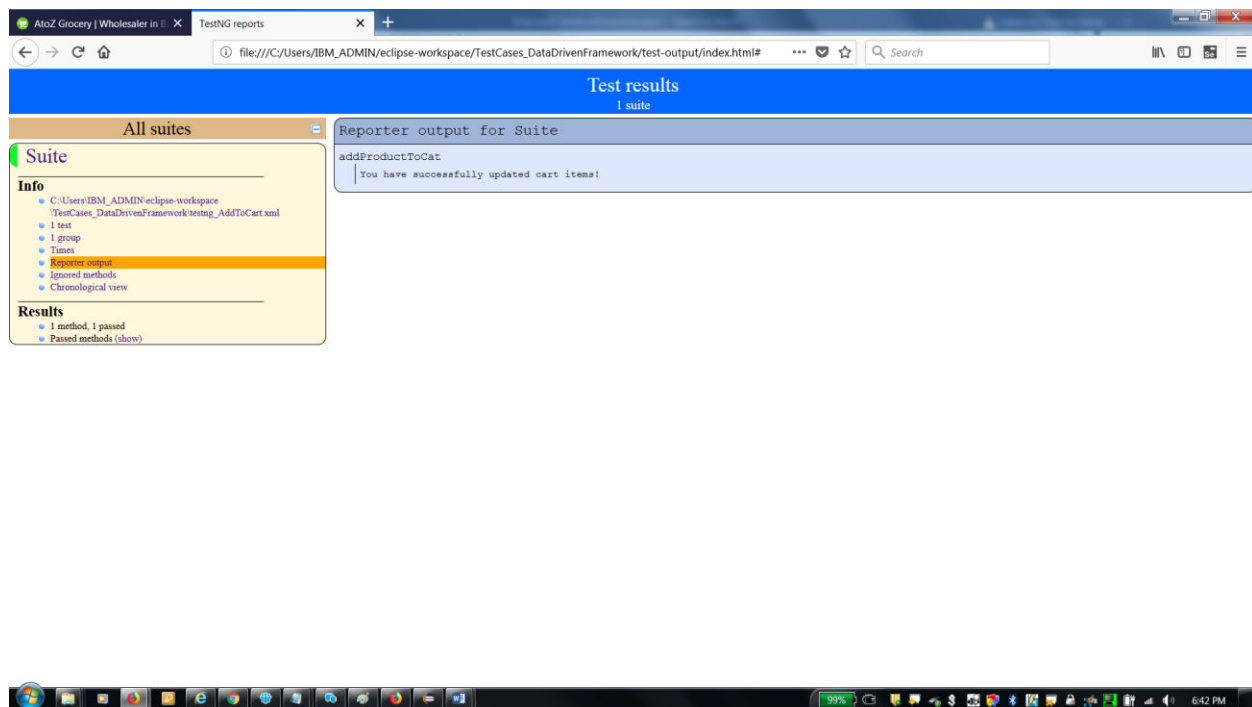
Class	Method	Start	Time (ms)
Suite			
AddingProducttoCart — passed			
com.ibm.groceries.AddtoCart	addProductToCart	1543393705407	11752

AddingProducttoCart

com.ibm.groceries.AddtoCart#addProductToCart

Messages
You have successfully updated cart items!

[back to summary](#)



Flow Information:

Initially a property file 'groceries.properties' is created in Test Data folder of the project and list of key, values for userpage url,full name,phone number, password,confirm password,quantity, validation message etc. are stored in this property file.

Under the class 'WebDriverLaunch' ,a Test NG annotation 'Before Suite' is used to invoke the method 'preSetFortTest' which is used to instantiate object for class PropertiesFileHandler to return the list of these key,values in to a HashMap.

The Test NG annotation 'Before Method' is used by passing the parameter for browser to select and to invoke the method 'Initialization' which is used to call another method and to create webdriver object for the firefoxdriver or chromedriver or internetexplorer by using switch statement and setting path for webdriver exe.

The class 'GroceriesUserPage' uses FindBy annotation to locate the elements signup link,fullname,password,phone number,confirm password,agree check box an signUp button. PageFactory is used in the constructor of this class.The method signUp is defined to enter the values for fullname,phone number,password,confirm password using sendKeys method. Click method is used to click on signup link,signup button.The method addProductToCart is defined to click on Addto Cart link, Cart link,Go to Cart link.

The class ShoppingCartPage uses FindBy annotation to locate elements quantity,Update shopping cart button,unit price,sub total value. PageFactory is used in the constructor of this class.The method updateQuantity is used to enter the value for quantity and to click on update shopping cart button.It

returns pagesource. The method verifyQuantity is defined to verify the updated value of quantity. The method verifySubTotal is used to verify that subtotal value is equal to quantity*unitprice and boolean value is returned.

The testng_AddToCart.xml file is having the parameter 'firefox' to launch Firefox browser.

The class GetScreenshot defines method takeScreenshot to capture screenshots.

The annotation @Test is used with method addProductToCart in the class 'AddtoCart'. This method is used to get the values for userpage url, fullname, password, phone number, confirm password etc, expected validation message, quantity. The get method is called to launch the web url for user portal. The objects for classes GroceriesUserPage, GetScreenShot, ShoppingCartPage are created and the above necessary methods are called on these objects. Finally the Assert is used by calling method for verifying the updated value of quantity and also verifying SubTotal value is equal to quantity*unitprice.