Test Objective:

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

Testcase: Tc06

Process: Edit a product

Steps to Execute:

Step1: Login to admin portal

Step2: Edit a product

Step3: Go to list all category option under show entries

Expected Results:

Validate the presence modified product.

Verify the presence of modified product on user page.

Note: Search for product using search bar.

Source code:

File1: `EditProduct.java`

```
package com.ibm.groceries;

import java.io.IOException;
import org.openqa.selenium.By;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Test;
import com.ibm.groceriespages.EditProductsPage;
import com.ibm.groceriespages.GroceriesUserPage;
import com.ibm.groceriespages.PageDashboard;
import com.ibm.groceriespages.PageLogin;
import com.ibm.groceriespages.PageProducts;
import com.ibm.initialization.WebDriverLaunch;

public class EditProduct extends WebDriverLaunch {


	@Test(priority=1, testName="EditProduct",groups="low")
	public void editProduct()throws IOException,InterruptedException
	{
		String url=data.get("url");
		String userName = data.get("username");
		String password = data.get("password");
		String expMessage=data.get("expectedEditProductMessage");
		String newNameProduct=data.get("prodNameNew");
```

```
             String newmodelProduct=data.get("modelNameNew");
             String productNotUpdatedMsg=data.get("productNotUpdated");
             String userPage=data.get("userPageUrl");
             String proudctFoundMessage=data.get("userPageMsg");

             //Launching the web site for atozgroceries
             driver.get(url);

             PageLogin login = new PageLogin(driver, wait);
             //To enter email address and password and clickon login button
             login.enterEmailAddress(userName);
             login.enterPassword(password);
             login.clickOnLogin();

      Assert.assertTrue(driver.findElement(By.partialLinkText("Logout")).isDisplayed
());

             PageDashboard dashboard=new PageDashboard(driver, wait);
             //To click on Catalog
             dashboard.clickOnCatalog();
             //To click on Products
             dashboard.clickOnProducts();


             PageProducts selProduct=new PageProducts(driver,wait);
             selProduct.selectProduct();

             // TO wait for the text box product name to be displayed.

      wait.until(ExpectedConditions.presenceOfElementLocated(By.id("pro_name")));

             EditProductsPage editPage=new EditProductsPage(driver,wait);
             String pageSource=editPage.editProduct(newNameProduct,newmodelProduct);

       try {

              if(pageSource.contains(expMessage))
                   {
                   Reporter.log(expMessage);
                   Assert.assertTrue(pageSource.contains(expMessage));

      wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//select[@nam
e='dataTableExample2_length']")));

                   //Calling method to select the entry dropdown as 'All'
                   String pageSource2=selProduct.selectEntry();

             //To check the  new name of the product is displayed
             Assert.assertTrue(pageSource2.contains(newNameProduct));

             //To check the new model of the product is displayed
             Assert.assertTrue(pageSource2.contains(newmodelProduct));
             }
      }
      catch(Exception e)
```

```
       {

             Assert.fail();
       }

       finally {
             //To click on on Logout button
             driver.findElement(By.partialLinkText("Logout")).click();

       }


       //To launch groceries user page
       driver.get(userPage);
       GroceriesUserPage userpage=new GroceriesUserPage(driver,wait);
       String userPageSource=userpage.searchProduct(newNameProduct);

       try {
              if(userPageSource.contains(newNameProduct))
                    {
                    System.out.println(proudctFoundMessage);
                    Assert.assertTrue(userPageSource.contains(newNameProduct));
                    }
             }
              catch(Exception e)
                    {
                     Assert.fail();
                    }

       }


}
```

File2: PageLogin.java

```
package com.ibm.groceriespages;
import org.openqa.selenium.By;
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.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class PageLogin {

       @FindBy(name="email")
       WebElement emailEle;

       @FindBy(name="pword")
       WebElement passEle;

       //@FindBy(className="btn btn-labeled btn-info m-b-5")
       @FindBy(xpath="//button[@class='btn btn-labeled btn-info m-b-5']")
```

```
        WebElement loginEle;

        WebDriverWait wait;
        WebDriver driver;

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

        //To enter email address
        public void enterEmailAddress(String userName)
        {
                emailEle.sendKeys(userName);
        }

        //To enter apssword
        public void enterPassword(String password)
        {
                passEle.sendKeys(password);
        }

        //To click on Login button
        public void clickOnLogin()
        {
                loginEle.click();

        }

}
```

File3: PageDashboard.java

```
package com.ibm.groceriespages;

import org.openqa.selenium.By;

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.ExpectedConditions;

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


public class PageDashboard {
```

```
//@FindBy(xpath="//a[@class='material-ripple']")
@FindBy(xpath="//a[text()=' Catalog']")
WebElement catalogEle;

@FindBy(xpath="//a[text()=' Products']")
WebElement productEle;

By prodElt=By.xpath("//a[text()=' Products']");
WebDriverWait wait;
WebDriver driver;

//Locating elements for Marketing link
@FindBy(xpath="//a[text()='  Marketing']")
WebElement marketEle;

//Locating Pushnotification link
@FindBy(xpath="//a[text()=' Push Notification']")
WebElement notificatonEle;



//Locating System link
@FindBy(xpath="//a[text()='   System']")
WebElement systemEle;

//Locating Returns link
@FindBy(xpath="//a[text()='  Returns']")
//@FindBy(xpath("//a[contains(text(),'Returns')]"))
WebElement returnsEle;
```

```
//Locating Shipping link
@FindBy(xpath="//a[text()='  Shipping']")
WebElement shippingEle;

//Locating ShippingLocatons link
@FindBy(xpath="//a[text()=' Shipping Locations']")
WebElement locatoinEle;

@FindBy(xpath="//a[text()=' Return Actions']")
WebElement actionsEle;

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

//To click on Catalog
public void clickOnCatalog()
{
	catalogEle.click();
	wait.until(ExpectedConditions.presenceOfElementLocated(prodElt));


	}
//To click on Products
public void clickOnProducts()
	{
	productEle.click();
```

```
}



//To click on Marketing link
public void clickOnMarketing()
{

	marketEle.click();

}

//To click on Push Notificatoin link
public void clickOnPushNotification()
{
	notificatonEle.click();

}

	//TO click on System link
public void clickOnsystem()
{
	systemEle.click();

}

//To click on Retuns link
public void clickOnReturns()
{
        driver.findElement(By.partialLinkText("Returns")).click();
        //returnsEle.click();
    }

    //To click on Return actoins link
    public void clickOnRetrunActions()

    {
        actionsEle.click();
    }

    public void clickOnShipping()
    {
        driver.findElement(By.partialLinkText("Shipping")).click();
        //shippingEle.click();

    }

    public void clickOnShippingLocations()
    {
        locatoinEle.click();

    }


}
```

File4: PageProducts.java

```
package com.ibm.groceriespages;
```

```
import java.io.IOException;
import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindAll;
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;
public class PageProducts {

	//Locating no.of products
	@FindAll(@FindBy(xpath="//table[@id='dataTableExample2']/tbody/tr"))
	List<WebElement> productRows;

	//Locating action button
	@FindBy(xpath="//table[@id='dataTableExample2']/tbody/tr[5]/td[9]/descendant::button")
	WebElement actionEle;

	//Locating delete web element
	@FindBy(xpath="//table[@id='dataTableExample2']/tbody/tr[1]/td[9]/descendant::a[2]")
	WebElement deleteEle;

	//Locating confirm button
	//@FindBy(xpath="//button[@className='confirm']")
```

```java
@FindBy(xpath="//button[text()='Yes, delete it!']")
WebElement confirmEle;
By confirmButton=By.xpath("//div[@class='sa-confirm-button-container']/descendant::button");



//Locating Edit web element
@FindBy(xpath="//table[@id='dataTableExample2']/tbody/tr[5]/td[9]/descendant::a[1]")
WebElement editEle;

@FindBy(xpath="//select[@name='dataTableExample2_length']")
WebElement entryEle;



WebDriverWait wait;
WebDriver driver;

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

//Method to delete the product
public String deleteProduct() throws InterruptedException

{
        //Verifying if number of products greater than zero
        if(productRows.size()>0)
        {
```

```
        System.out.println("The list of products are dipslayed");
        Thread.sleep(10000);
        //To click on action button
        actionEle.click();
        //To click on Delete button
        deleteEle.click();
        Thread.sleep(10000);

        wait.until(ExpectedConditions.presenceOfElementLocated(confirmButton));

        //To click on confirm button
        confirmEle.click();
        driver.manage().timeouts().implicitlyWait(60, TimeUnit.SECONDS);
    }
    return driver.getPageSource();
}

//Method to clikcon on edit button to edit product
public void selectProduct()
{
if(productRows.size()>0)
    {
    //to click on on action web element
    actionEle.click();

    //to click on Edit link
    editEle.click();

    }
```

```
	}

	public String selectEntry()
	{
		WebElement entryElt=driver.findElement(By.xpath("//select[@name='dataTableExample2_length']"));
		//Select selectEntry=new Select(entryEle);
		Select selectEntry=new Select(entryElt);
		selectEntry.selectByVisibleText("All");
		return driver.getPageSource();
	}

}
```

File5: EditProductsPage.java

```
package com.ibm.groceriespages;

import java.util.List;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindAll;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class EditProductsPage {
```

```
//To located product name

@FindBy(xpath = "//input[@id='pro_name']")

WebElement nameEle;


// To locate Data link

@FindBy(xpath = "//a[text()='Data']")

WebElement dataEle;


// To locate model text box

@FindBy(xpath = "//input[@id='model']")

WebElement modelEle;


// To locate save button

@FindBy(xpath = "//button[@title='Save']")

WebElement saveEle;


WebDriverWait wait;

WebDriver driver;


public EditProductsPage(WebDriver driver, WebDriverWait wait) {

        PageFactory.initElements(driver, this);

        this.driver = driver;

        this.wait = wait;

}


public String editProduct(String productNewname, String modelNewname) {

        // To enter the new name for product

        nameEle.clear();
        nameEle.sendKeys(productNewname);


        // To click on Data link
        dataEle.click();


        // To enter the new name for model
        modelEle.clear();
        modelEle.sendKeys(modelNewname);


        // To click on Save button
        saveEle.click();
        return driver.getPageSource();
    }


}
```

File6: GroceriesUserPage.java

```
package com.ibm.groceriespages;

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 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;

    WebDriverWait wait;
    WebDriver driver;
```

```java
    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();

    }
}
```

File7: WebDriverLaunch.java

```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;
    }
}
```

```
}
```

File8: 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();

	}


}
```

File9: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
```

File10:testing_EditProduct.xml

```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="EditingProduct">
  <parameter name="browser" value="firefox"></parameter>
    <classes>
      <class name="com.ibm.groceries.EditProduct"/>
    </classes>
  </test> <!-- EditingProduct -->
</suite> <!-- Suite -->
```

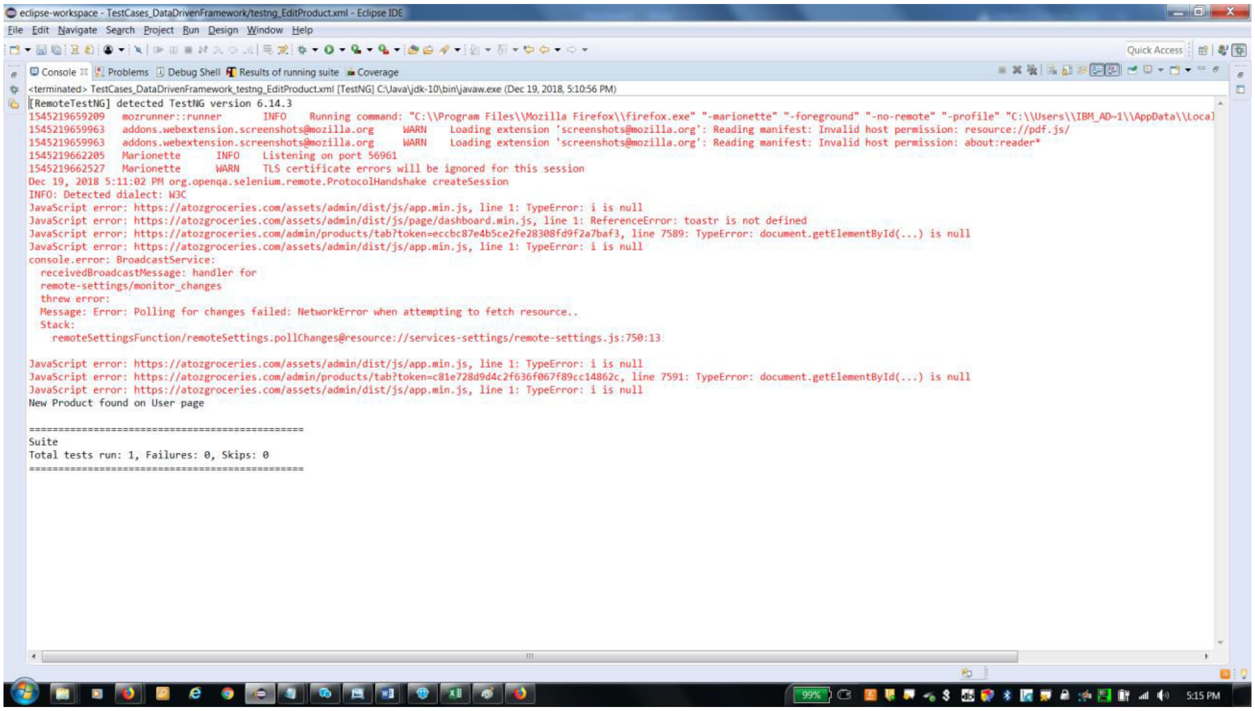
Test Result:

It is verified that the search is displayed data with “Discount” keyword when matching result found.

It is also verified for expected message ‘No matching records found’ when no matching result found.

Screenshots:

Console output:

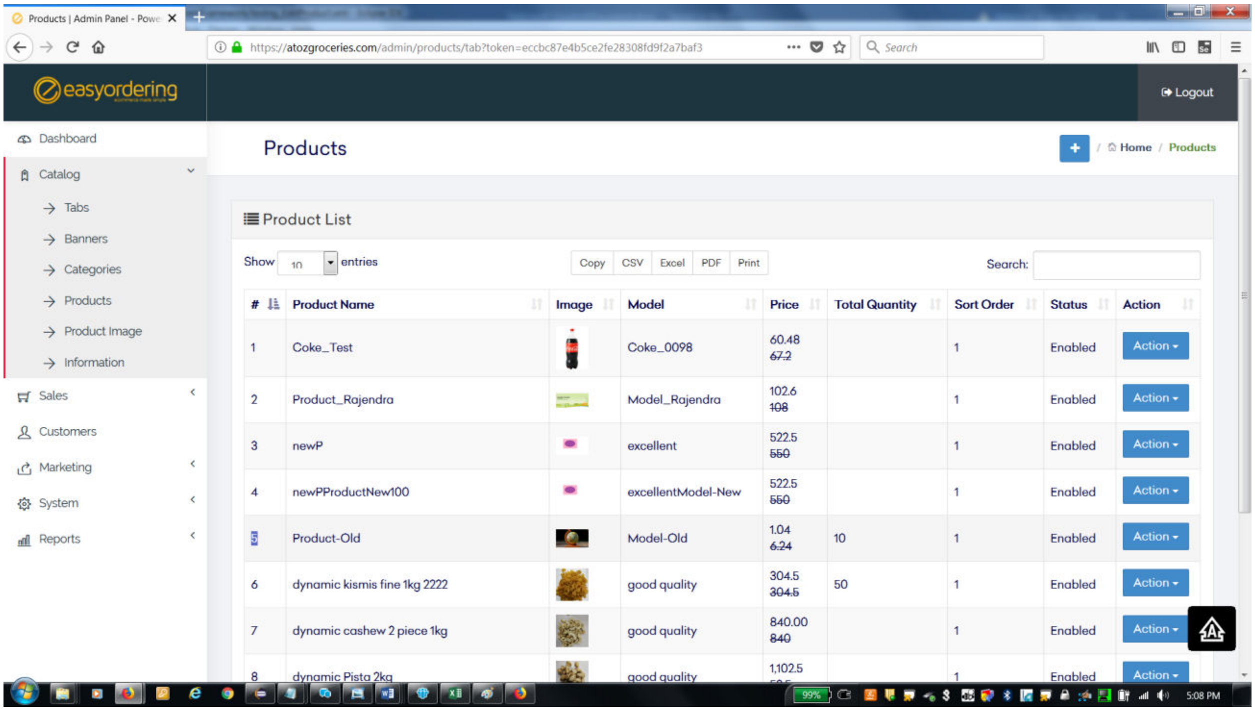

eclipse-workspace - TestCases_DataDrivenFramework/testng_EditProduct.xml - Eclipse IDE
File Edit Navigate Search Project Run Design Window Help
Quick Access
Console
Problems
Debug Shell
Results of running suite
Coverage
99%
5:15 PM

```
<terminated> TestCases_DataDrivenFramework_testng_EditProduct.xml [TestNG] C:\Java\jdk-10\bin\javaw.exe (Dec 19, 2018, 5:10:56 PM)
[RemoteTestNG] detected TestNG version 6.14.3
1545219659209   mozrunner::runner       INFO    Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "-marionette" "-foreground" "-no-remote" "-profile" "C:\\Users\\IBM_AD~1\\AppData\\Local
1545219659963   addons.webextension.screenshots@mozilla.org     WARN    Loading extension 'screenshots@mozilla.org': Reading manifest: Invalid host permission: resource://pdf.js/
1545219659963   addons.webextension.screenshots@mozilla.org     WARN    Loading extension 'screenshots@mozilla.org': Reading manifest: Invalid host permission: about:reader*
1545219662205   Marionette      INFO    Listening on port 56961
1545219662527   Marionette      WARN    TLS certificate errors will be ignored for this session
Dec 19, 2018 5:11:02 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/page/dashboard.min.js, line 1: ReferenceError: toastr is not defined
JavaScript error: https://atozgroceries.com/admin/products/tab?token=eccbc87e4b5ce2fe28308fd9f2a7baf3, line 7589: TypeError: document.getElementById(...) is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
console.error: BroadcastService:
  receivedBroadcastMessage: handler for
  remote-settings/monitor_changes
  threw error:
  Message: Error: Polling for changes failed: NetworkError when attempting to fetch resource..
  Stack:
    remoteSettingsFunction/remoteSettings.pollChanges@resource://services-settings/remote-settings.js:750:13

JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
JavaScript error: https://atozgroceries.com/admin/products/tab?token=c81e728d9d4c2f636f067f89cc14862c, line 7591: TypeError: document.getElementById(...) is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
New Product found on User page

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

Existing Products List:



Products

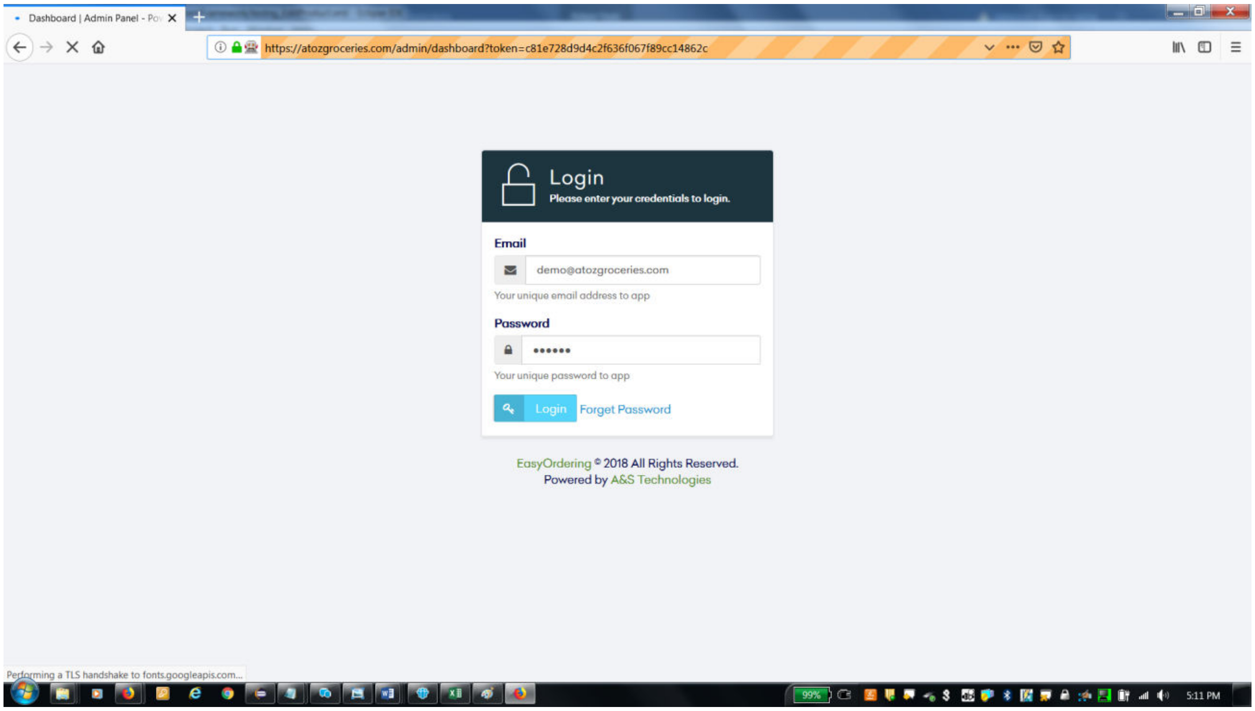
Product List

Show 10 entries

Search:

#	Product Name	Image	Model	Price	Total Quantity	Sort Order	Status	Action
1	Coke_Test		Coke_0098	60.48 ~~67.2~~		1	Enabled	Action
2	Product_Rajendra		Model_Rajendra	102.6 ~~108~~		1	Enabled	Action
3	newP		excellent	522.5 ~~550~~		1	Enabled	Action
4	newPProductNew100		excellentModel-New	522.5 ~~550~~		1	Enabled	Action
5	Product-Old		Model-Old	1.04 ~~6.24~~	10	1	Enabled	Action
6	dynamic kismis fine 1kg 2222		good quality	304.5 ~~304.5~~	50	1	Enabled	Action
7	dynamic cashew 2 piece 1kg		good quality	840.00 ~~840~~		1	Enabled	Action
8	dynamic Pista 2kg		good quality	1,102.5		1	Enabled	Action

Login page Is displayed with entered username and password



Dashboard | Admin Panel - Po
https://atozgroceries.com/admin/dashboard?token=c81e728d9d4c2f636f067f89cc14862c
Login
Forget Password
Performing a TLS handshake to fonts.googleapis.com...
5:11 PM

Login

Please enter your credentials to login.

Email

demo@atozgroceries.com

Your unique email address to app

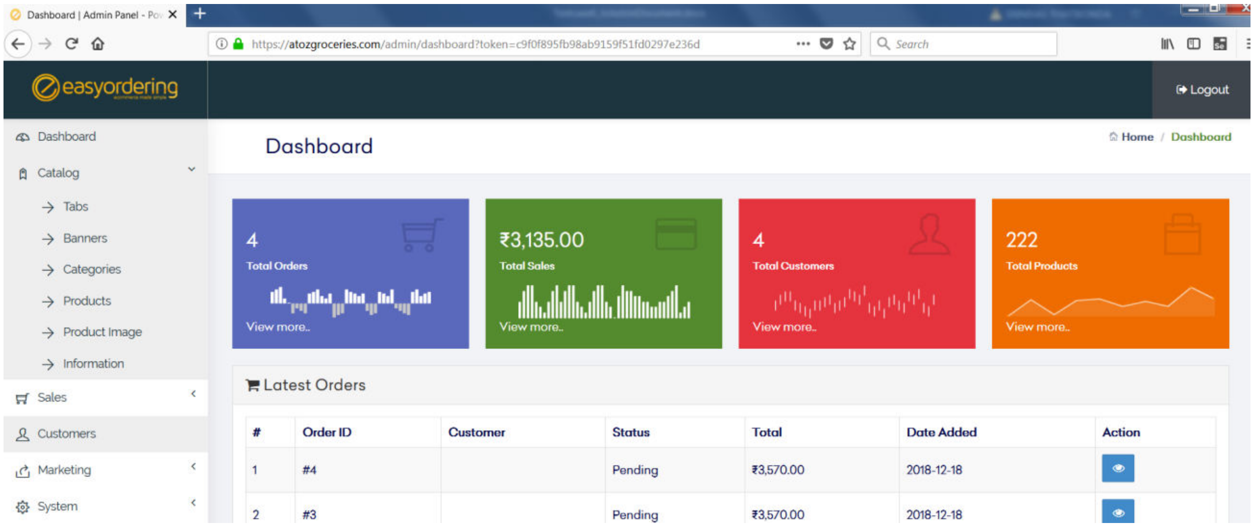
Password

••••••

Your unique password to app

EasyOrdering © 2018 All Rights Reserved.
Powered by A&S Technologies

Dashboard page is displayed with Catalog->Products

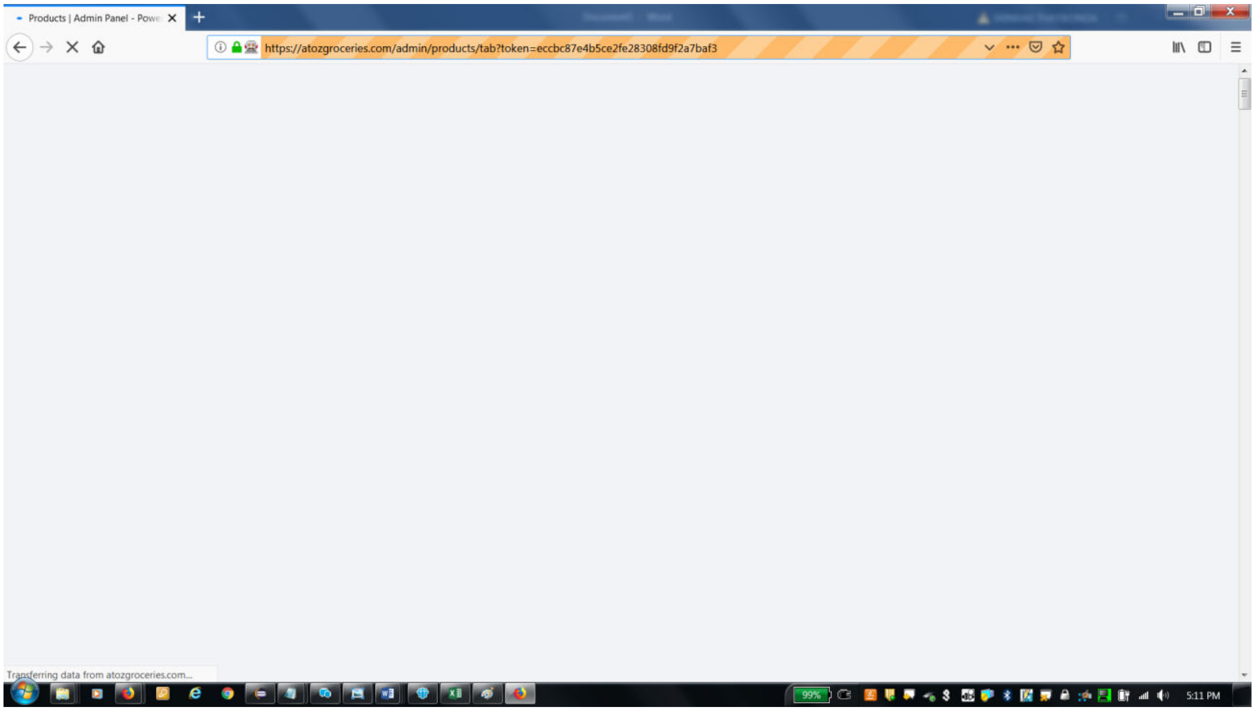


Dashboard

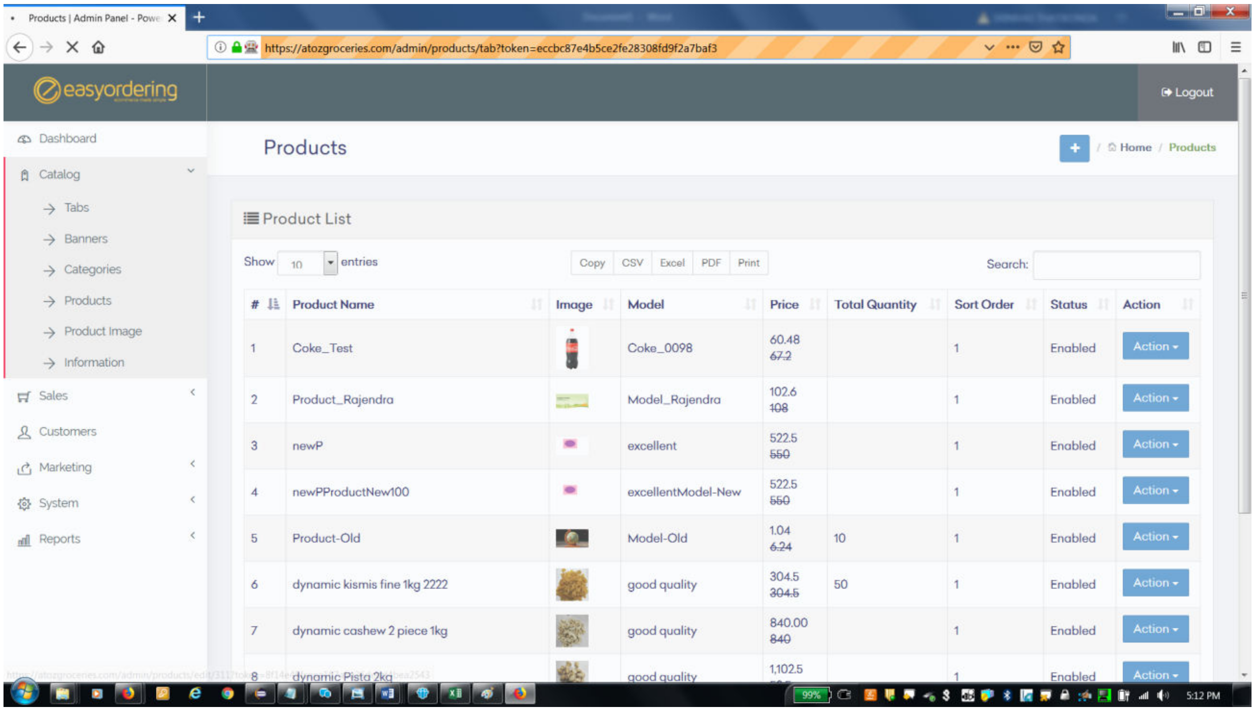
Latest Orders

#	Order ID	Customer	Status	Total	Date Added	Action
1	#4		Pending	₹3,570.00	2018-12-18	👁
2	#3		Pending	₹3,570.00	2018-12-18	👁

Products page is displaying

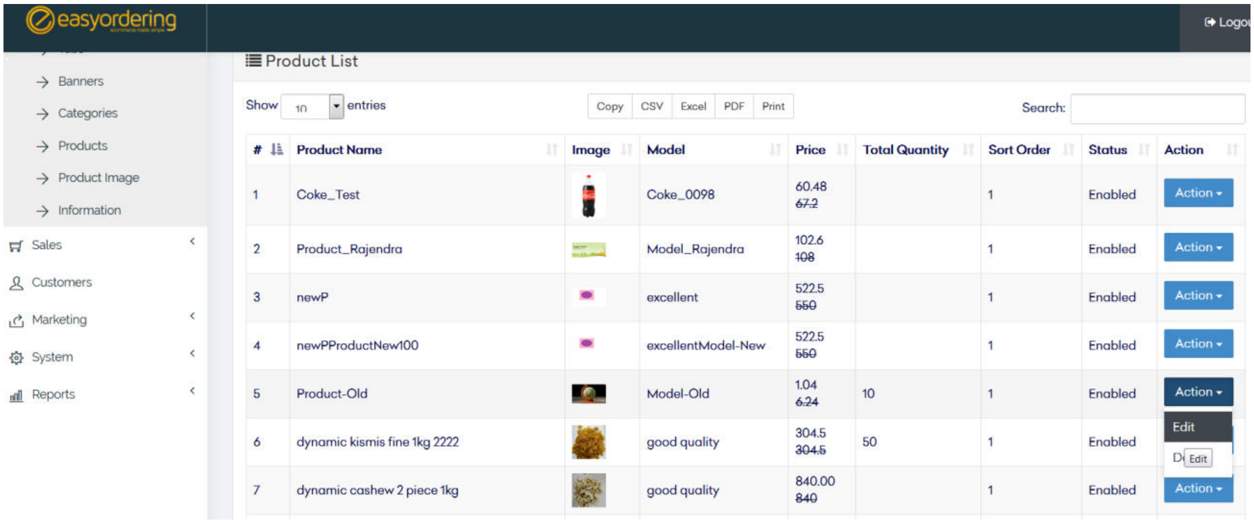

Products | Admin Panel - Powe
https://atozgroceries.com/admin/products/tab?token=eccbc87e4b5ce2fe28308fd9f2a7baf3
Transferring data from atozgroceries.com...
99%
5:11 PM

Product List is displayed



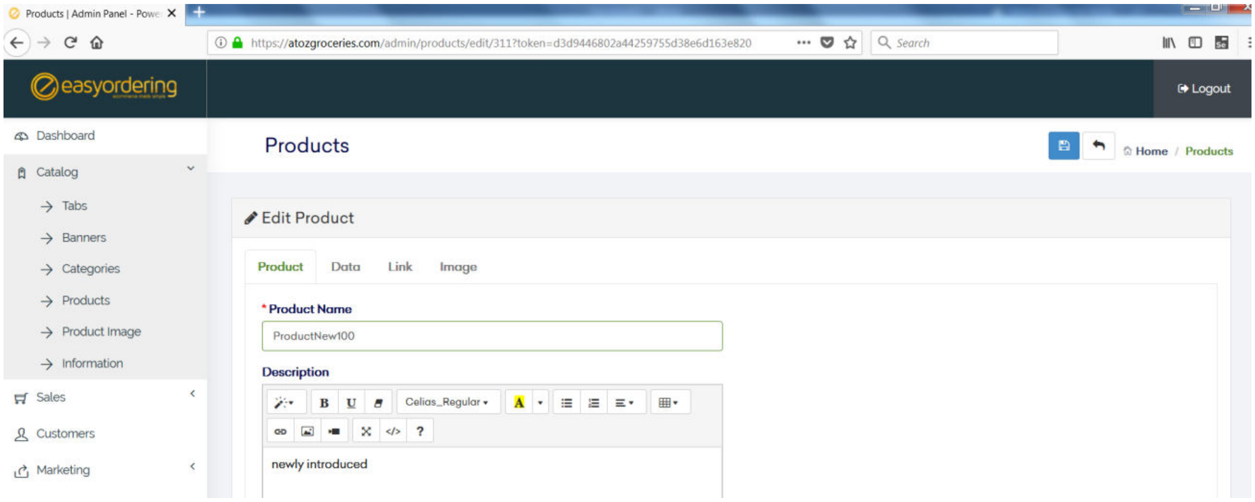
#	Product Name	Image	Model	Price	Total Quantity	Sort Order	Status	Action
1	Coke_Test		Coke_0098	60.48 ~~67.2~~		1	Enabled	Action ▾
2	Product_Rajendra		Model_Rajendra	102.6 ~~108~~		1	Enabled	Action ▾
3	newP		excellent	522.5 ~~550~~		1	Enabled	Action ▾
4	newPProductNew100		excellentModel-New	522.5 ~~550~~		1	Enabled	Action ▾
5	Product-Old		Model-Old	1.04 ~~6.24~~	10	1	Enabled	Action ▾
6	dynamic kismis fine 1kg 2222		good quality	304.5 ~~304.5~~	50	1	Enabled	Action ▾
7	dynamic cashew 2 piece 1kg		good quality	840.00 ~~840~~		1	Enabled	Action ▾
8	dynamic Pista 2kg		good quality	1,102.5		1	Enabled	Action ▾

The product is selected to edit.



#	Product Name	Image	Model	Price	Total Quantity	Sort Order	Status	Action
1	Coke_Test		Coke_0098	60.48 ~~67.2~~		1	Enabled	Action ▾
2	Product_Rajendra		Model_Rajendra	102.6 ~~108~~		1	Enabled	Action ▾
3	newP		excellent	522.5 ~~550~~		1	Enabled	Action ▾
4	newPProductNew100		excellentModel-New	522.5 ~~550~~		1	Enabled	Action ▾
5	Product-Old		Model-Old	1.04 ~~6.24~~	10	1	Enabled	Action ▾
6	dynamic kismis fine 1kg 2222		good quality	304.5 ~~304.5~~	50	1	Enabled	
7	dynamic cashew 2 piece 1kg		good quality	840.00 ~~840~~		1	Enabled	Action ▾

Edit Product page is displayed and Product Name is edited as ‘ProductNew100’ for existing product ‘Product-Old’



Products

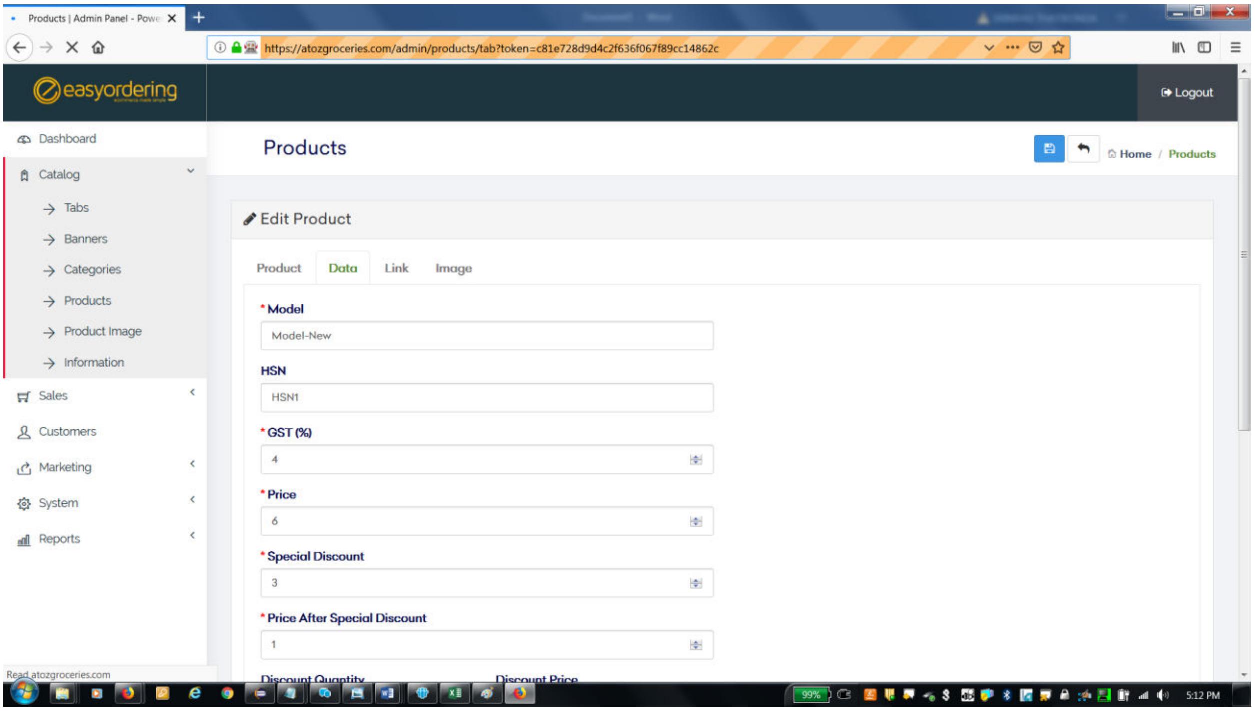
Edit Product

Product | Data | Link | Image

* Product Name: ProductNew100

Description: newly introduced

After clicking Data link, Model is changed to Model-New



Products

Edit Product

Product | Data | Link | Image

* Model: Model-New

HSN: HSN1

* GST (%): 4

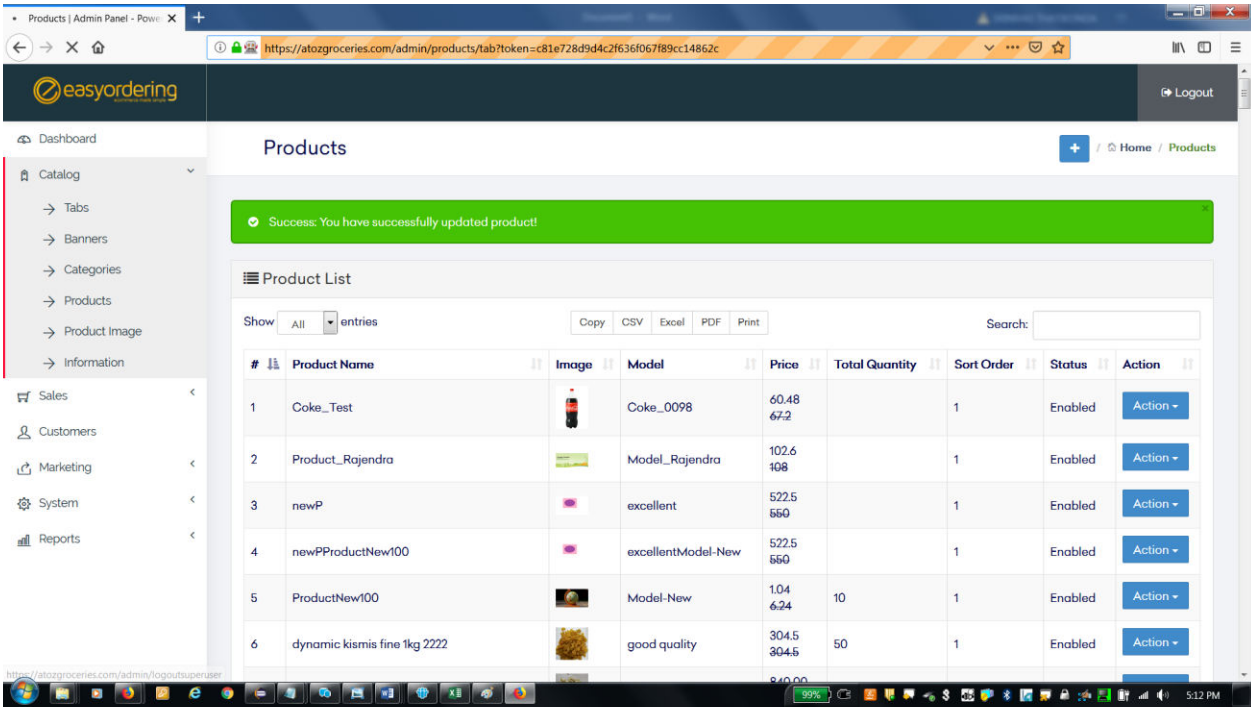
* Price: 6

* Special Discount: 3

* Price After Special Discount: 1

Discount Quantity | Discount Price

After clicking on Save button, the below page is displayed



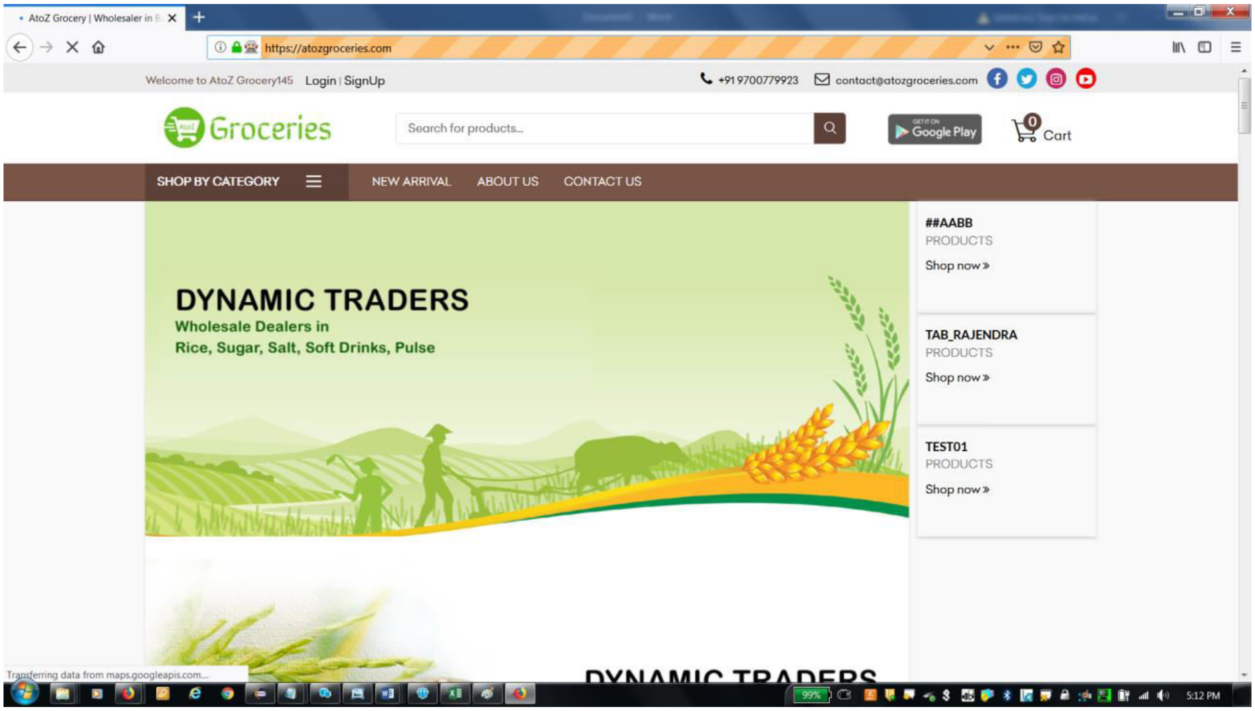
Products

Success: You have successfully updated product!

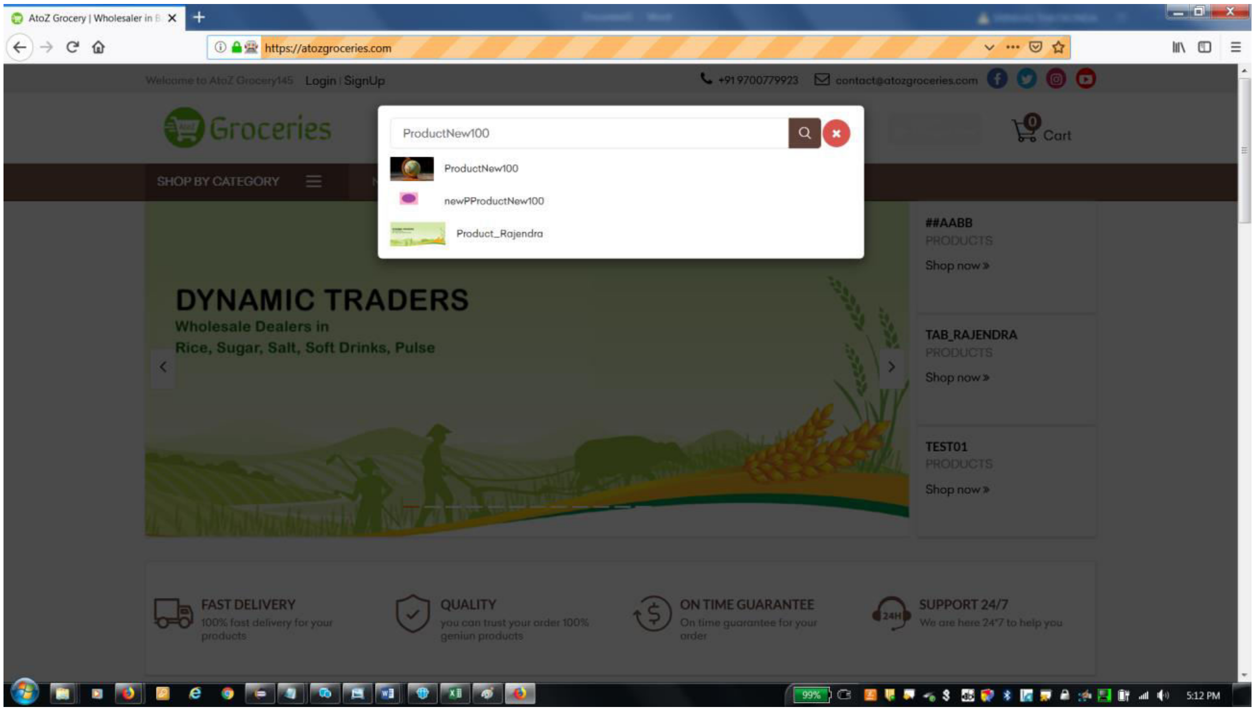
Product List

#	Product Name	Image	Model	Price	Total Quantity	Sort Order	Status	Action
1	Coke_Test		Coke_0098	60.48 ~~67.2~~		1	Enabled	Action
2	Product_Rajendra		Model_Rajendra	102.6 ~~108~~		1	Enabled	Action
3	newP		excellent	522.5 ~~550~~		1	Enabled	Action
4	newPProductNew100		excellentModel-New	522.5 ~~550~~		1	Enabled	Action
5	ProductNew100		Model-New	1.04 ~~6.24~~	10	1	Enabled	Action
6	dynamic kismis fine 1kg 2222		good quality	304.5 ~~304.5~~	50	1	Enabled	Action

User page is launched for www.atozgrocerries.com site.



Search text box is populated with new name of the product to search


ProductNew100

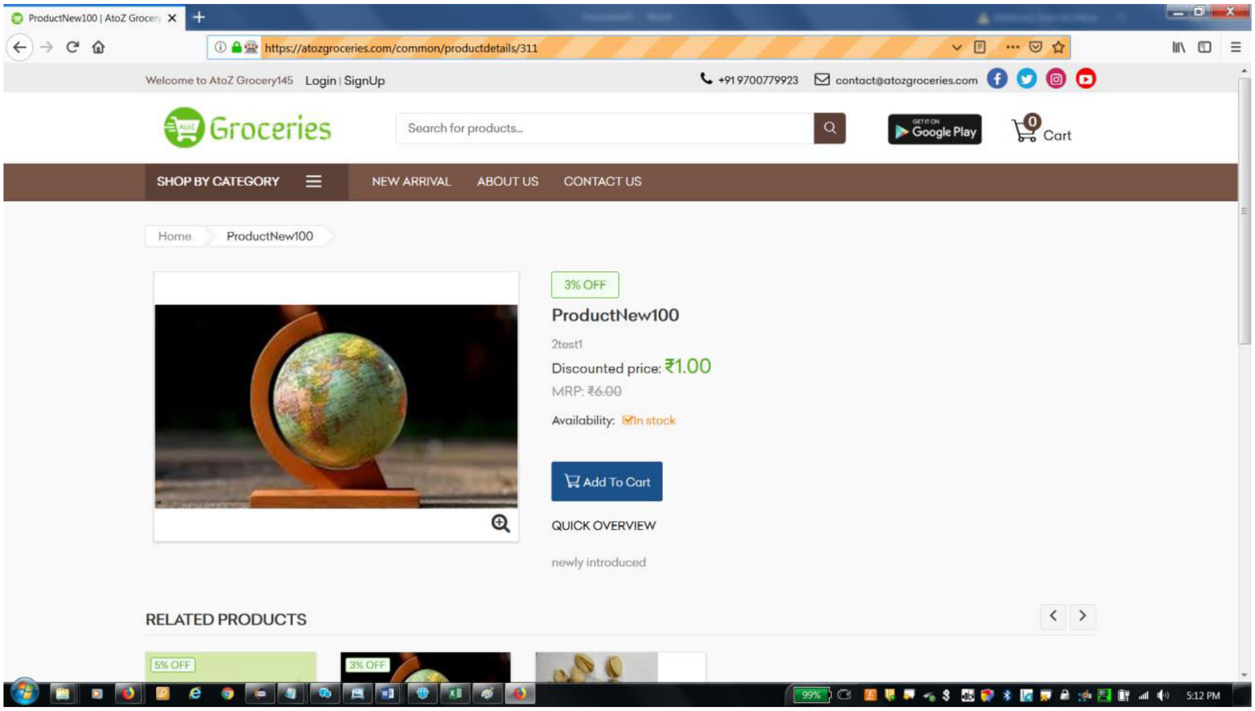
ProductNew100

newPProductNew100

Product_Rajendra

AtoZ Grocery | Wholesaler in
https://atozgroceries.com
Welcome to AtoZ Grocery145 Login | SignUp
+91 9700779923
contact@atozgroceries.com
Groceries
0 Cart
SHOP BY CATEGORY
DYNAMIC TRADERS
Wholesale Dealers in
Rice, Sugar, Salt, Soft Drinks, Pulse
##AABB
PRODUCTS
Shop now »
TAB_RAJENDRA
PRODUCTS
Shop now »
TEST01
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
99%
5:12 PM

Product with new name is displayed on user page.



3% OFF

ProductNew100

2test1

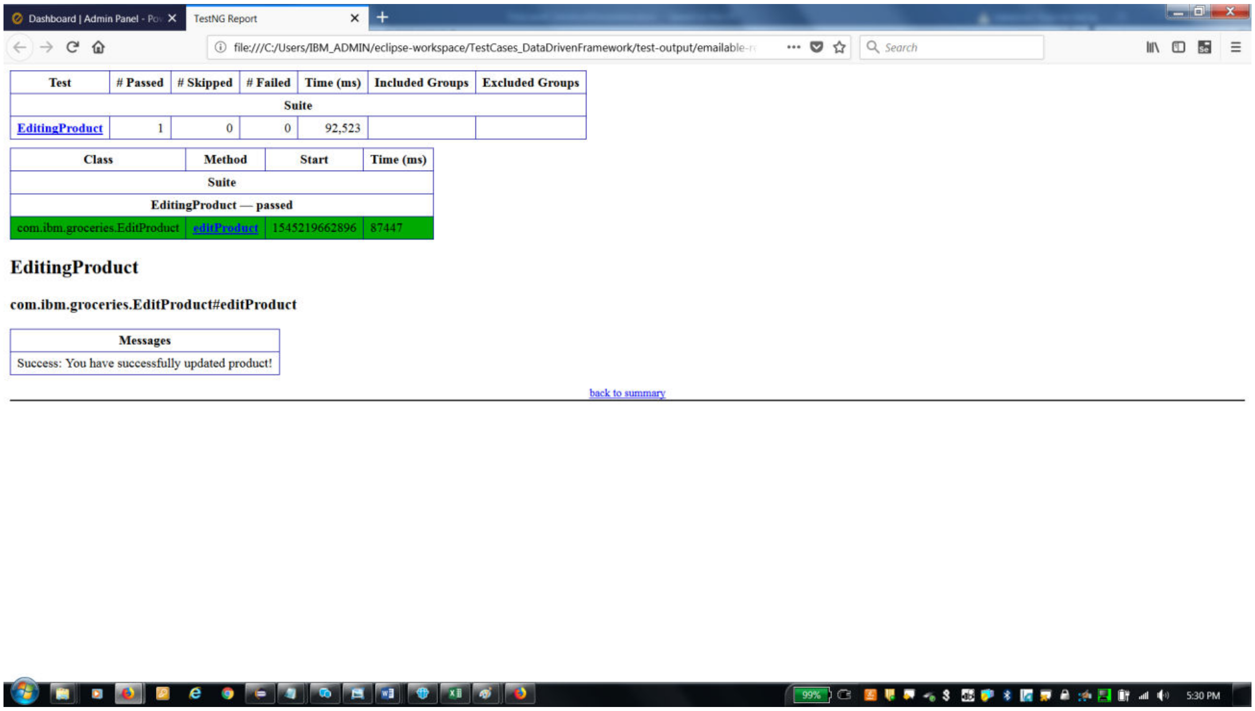
Discounted price: ₹1.00

MRP: ~~₹6.00~~

Availability: ☑In stock

QUICK OVERVIEW

newly introduced


Dashboard | Admin Panel - Po
TestNG Report
file:///C:/Users/IBM_ADMIN/eclipse-workspace/TestCases_DataDrivenFramework/test-output/emailable-r
Search
back to summary
5:30 PM

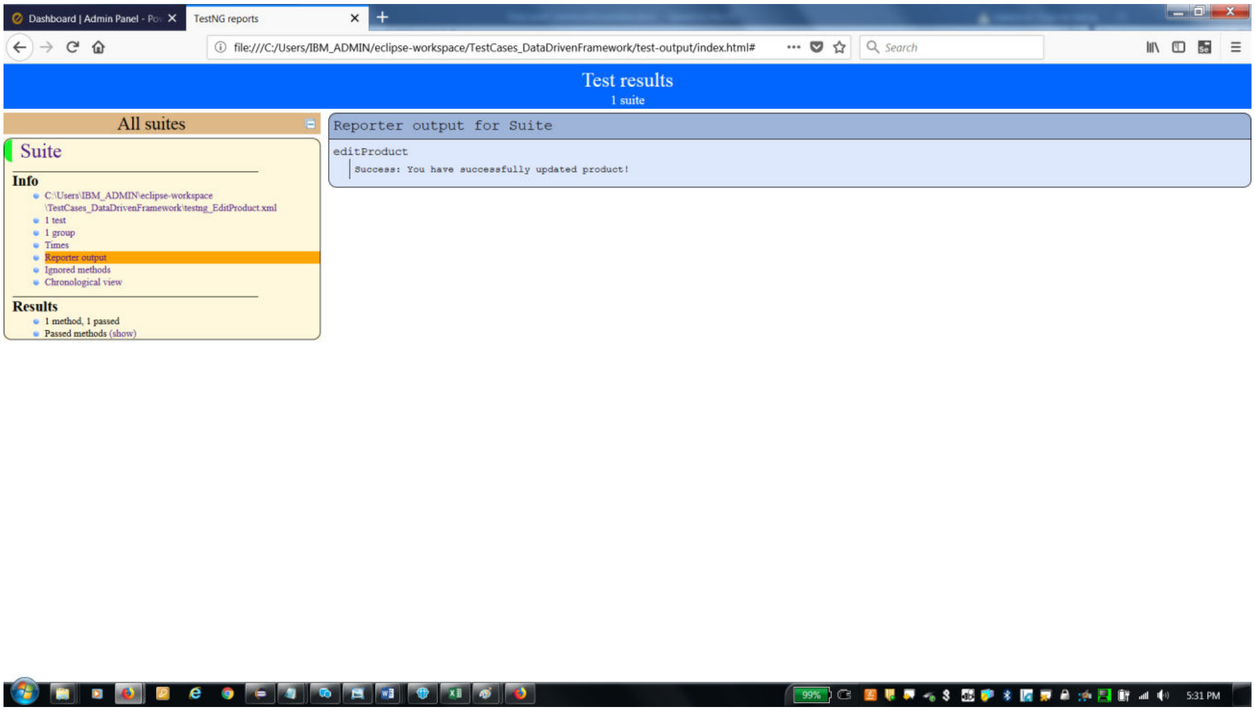
Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
EditingProduct	1	0	0	92,523		

Class	Method	Start	Time (ms)
Suite			
EditingProduct — passed			
com.ibm.groceries.EditProduct	editProduct	1545219662896	87447

EditingProduct

com.ibm.groceries.EditProduct#editProduct

Messages
Success: You have successfully updated product!


Dashboard | Admin Panel - Po
TestNG reports
file:///C:/Users/IBM_ADMIN/eclipse-workspace/TestCases_DataDrivenFramework/test-output/index.html#
Search
5:31 PM

Test results
1 suite

All suites

Suite

Info
- C:\Users\IBM_ADMIN\eclipse-workspace \TestCases_DataDrivenFramework\testng_EditProduct.xml
- 1 test
- 1 group
- Times
- Reporter output
- Ignored methods
- Chronological view

Results
- 1 method, 1 passed
- Passed methods (show)

Reporter output for Suite

```
editProduct
    Success: You have successfully updated product!
```

Flow Information:

Initially a property file ‘groceries.properties’ is created in Test Data folder of the project and list of key, values for admin page url, username,password ,expected validation message, user page url 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 PageLogin class is defined with methods to locate the web elements email,password and login buttons. PageFactory is used in the constructor of this class. The method sendKeys is used to enter the email address and password. The method click is used to click on login button

The class ‘PageDashboard’ is defined with methods to locate the web elements Catalog,Products. PageFactory is used in the constructor of this class.The method click is used to click on Catalog, Products links.

The class ‘PageProducts’ is defined with method selectProduct to click on action link,Edit link which are located using ‘FindBy’ annotation . PageFactory method is used in the in the constructor of this class. The method selectEntry is defined to select the value as ‘All’ for entry dropwdown and pagesource is returned.

The class ‘EditProductsPage’ uses FindBy annotation to locate the elements product name,model ,data link,Save link.The method editProduct is defined to click edit the new name for product name, new value for model using sendkeys method.Click method is used to clik on Save button and pasgesource is returned.

The annotation @Test is used with method editProduct in the class ‘EditProduct’. This method is used to get the values for username,password , new name for product,new name for model ,valiation message etc. The get method is called to launch the web url for admin portal as well as user page. The objects for classes PageLogin,PageDashboard, PageProducts,EditProductsPage ,GroceriesUserPage are created and the above necessary methods are called on these objects. Finally the Assert is used to verify presence of the new name of product, new value of the model and expected message when product updated in admin page. The product with new name is also searched and validated in user page too. The testng_EditProduct xml file is having the parameter ‘firefox’ to launch Firefox browser.