**Test Objective:**

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

**Testcase:** Tc09

**Process:** Edit a product.

**Steps to Execute:**

Step1: Login to admin portal

Step2: Edit Product

**Expected Results:**

Validate the presence of modified product

Verify the presence of modified product on the database

**Source code:**

**File1**: EditProductDbase.java

```
package com.ibm.groceries;


import java.io.IOException;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;


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;

import com.ibm.utilities.DatabaseConnection;

import com.ibm.utilities.GetScreenshot;


import java.sql.Connection;


public class EditProductDbase extends WebDriverLaunch {


	@Test(priority = 1, testName = "EditProductDbase", groups = "low")


	public void editProductDbase() throws IOException, InterruptedException, SQLException {

		String url = data.get("url");

		String userName = data.get("username");

		String password = data.get("password");

		String expMessage = data.get("expectedEditProductMessage");

		String product = data.get("productName");

		String newtitleProduct = data.get("titleNameNew");

		String newtagdescProduct = data.get("tagDescriptionNew");

		String newkeywordProduct = data.get("keywordNew");

		String newmodelProduct = data.get("modelNameNew");

		String newhsnProduct = data.get("hsnNew");

		String productNotUpdatedMsg = data.get("productNotUpdated");

		String proudctFoundMessage = data.get("userPageMsg");

		String productstablename = data.get("tableProducts");


		String modifiedMsg = data.get("modifiedProductConsole");

		// Launching the web site for atozgroceries
```

```
driver.get(url);
GetScreenshot screen = new GetScreenshot();

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

screen.takeScreenshot(driver);
PageProducts selProduct = new PageProducts(driver, wait);
// Calling method to select the product to edit
selProduct.selectProduct();
screen.takeScreenshot(driver);
// 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);

// Calling method to edit the product details
```

```
        String pageSource = editPage.editProductDbase(newtitleProduct, newtagdescProduct,
newkeywordProduct,
                newmodelProduct, newhsnProduct);
        screen.takeScreenshot(driver);

        // Verifying if modified product is present
        Assert.assertTrue(pageSource.contains(product));

        // Getting the details of modified product from database
        if (pageSource.contains(expMessage)) {
            System.out.println(expMessage);
            DatabaseConnection conn = new DatabaseConnection();
            Statement st = conn.connectDatabase();
            // ResultSet rs=st.executeQuery("select *from as_products where
            // name=\"Badam-p01\"");
            ResultSet rs = st.executeQuery("select *from " + productstablename + " where
name=" + "'" + product + "'");
            System.out.println(modifiedMsg);
            while (rs.next()) {
                System.out.println(rs.getString("meta_title"));
                Assert.assertEquals(rs.getString("meta_title"), newtitleProduct);
                System.out.println(rs.getString("meta_description"));
                Assert.assertEquals(rs.getString("meta_description"),
newtagdescProduct);
                System.out.println(rs.getString("meta_keyword"));
                Assert.assertEquals(rs.getString("meta_keyword"),
newkeywordProduct);
                System.out.println(rs.getString("model"));
                Assert.assertEquals(rs.getString("model"), newmodelProduct);
                System.out.println(rs.getString("hsn"));
                Assert.assertEquals(rs.getString("hsn"), newhsnProduct);

            }
            rs.close();
        }

    }

}
```

**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;

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

```
WebElement settingsEle;


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();

    }

    //To click on Settings link
    public void clickOnSettings()
    {
        settingsEle.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[1]/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']")
    @FindBy(xpath="//button[text()='Yes, delete it!']")
    WebElement confirmEle;
    By confirmButton=By.xpath("//div[@class='sa-confirm-button-container']/descendant::button");
```

```java
//Locating Edit web element
@FindBy(xpath="//table[@id='dataTableExample2']/tbody/tr[1]/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.io.IOException;
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;

import com.ibm.utilities.GetScreenshot;
```

```
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;

	//To locate Meta Title textbox
	@FindBy(xpath="//input[@id='meta_title']")
	WebElement titleEle;

	//To locate Meta Tag description text are field
	@FindBy(xpath="//textarea[@id='meta_tag_desc']")
	WebElement tagDescriptionEle;

	//To locate Meta Tag Keywords
	@FindBy(xpath="//textarea[@id='meta_tag_keyword']")
	WebElement keywordEle;
```

```
//To locate HSN textbox
@FindBy(xpath="//input[@id='hsn']")
WebElement hsnEle;



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();
        }


        public String editProductDbase(String titleNew,String tagdescNew,String keywordNew,String
modelNewname,String hsnNew) throws IOException, InterruptedException
        {
                // To enter the new name for product
                //      nameEle.clear();
                //      nameEle.sendKeys(productNewname);
                GetScreenshot screen = new GetScreenshot();


                titleEle.clear();
                titleEle.sendKeys(titleNew);


                tagDescriptionEle.clear();
                tagDescriptionEle.sendKeys(tagdescNew);


                keywordEle.clear();
                keywordEle.sendKeys(keywordNew);


                screen.takeScreenshot(driver);


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


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

```
		hsnEle.clear();
		hsnEle.sendKeys(hsnNew);


		screen.takeScreenshot(driver);
		// To click on Save button
		saveEle.click();


		return driver.getPageSource();


	}




}
```

**File6:** DatabaseConnection.java

```
package com.ibm.utilities;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;


public class DatabaseConnection {


	public Statement connectDatabase()throws SQLException
	{
	Connection c=DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atozgroceries","foodsonfinger_atoz","welcome@123");
	Statement stmt=c.createStatement();
```

```
    return stmt;
    }
}
```

**File7:** 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, InterruptedException
    {
        Thread.sleep(2000);
        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"));
    }

}
```

**File8:** PropertiesFileHandler.java

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: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;

        }

    }

}
```

**File10:**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!

titleNameNew=Title-New

tagDescriptionNew=Tag-New

keywordNew=keyword-New

hsnNew=Hsn-New

tableProducts=as_products

productName=Badam-p01

modifiedProductConsole=Modified details of product from DataBase:
```

**File11:**testng_EditProductDbase.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="Editing Product Dbase">
  <parameter name="browser" value="firefox"></parameter>
    <classes>
      <class name="com.ibm.groceries.EditProductDbase"/>
    </classes>
  </test> <!-- Editing Product Dbase -->
</suite> <!-- Suite -->
```

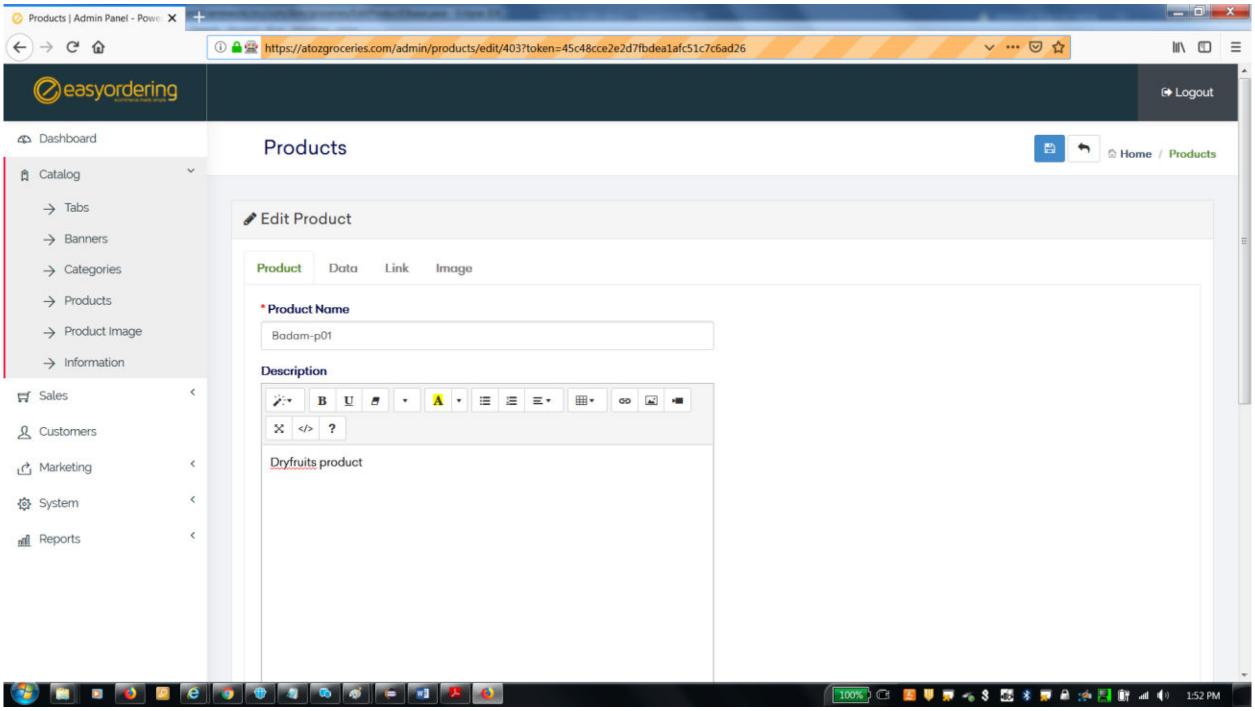
**Test Result:**

Validated the presence of modified product

Verified the presence of modified product on the database

**Screenshots:**

Existing product :



https://atozgroceries.com/admin/products/edit/403?token=45c48cce2e2d7fbdea1afc51c7c6ad26

Products

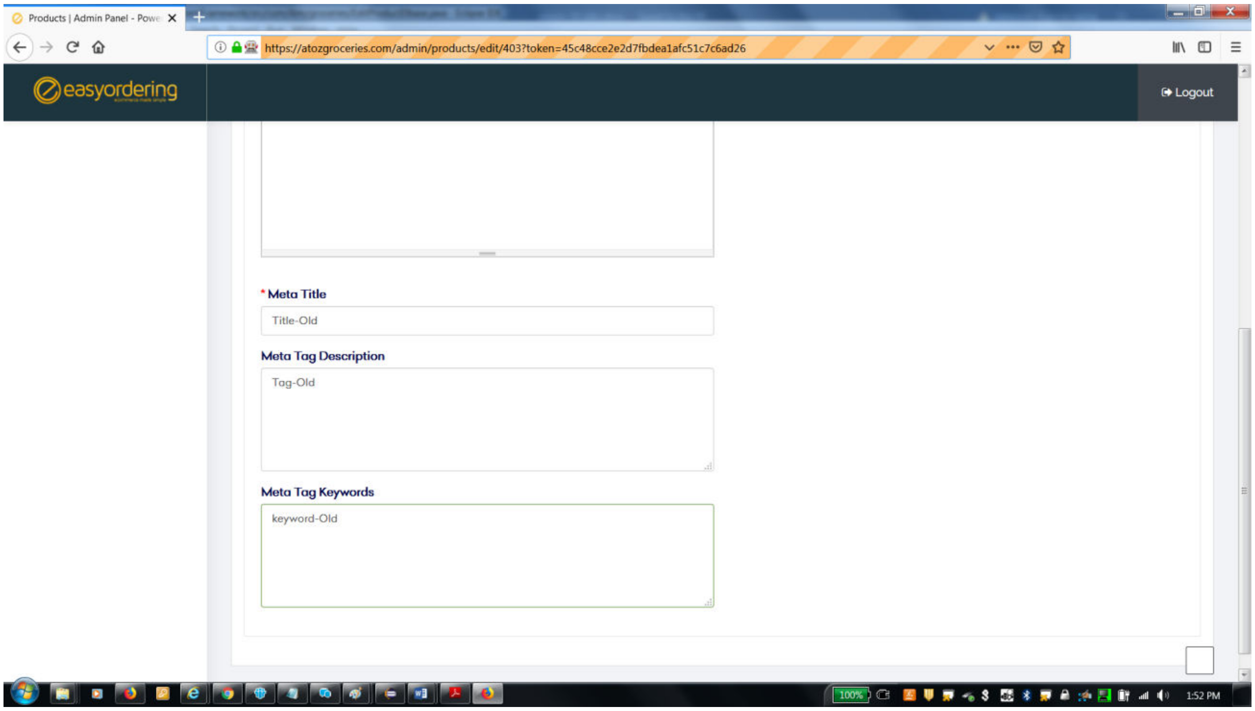
Edit Product

* Product Name

Badam-p01

Description

Dryfruits product



Products | Admin Panel - Powe
https://atozgroceries.com/admin/products/edit/403?token=45c48cce2e2d7fbdea1afc51c7c6ad26
easyordering
Logout
100%
1:52 PM

* Meta Title

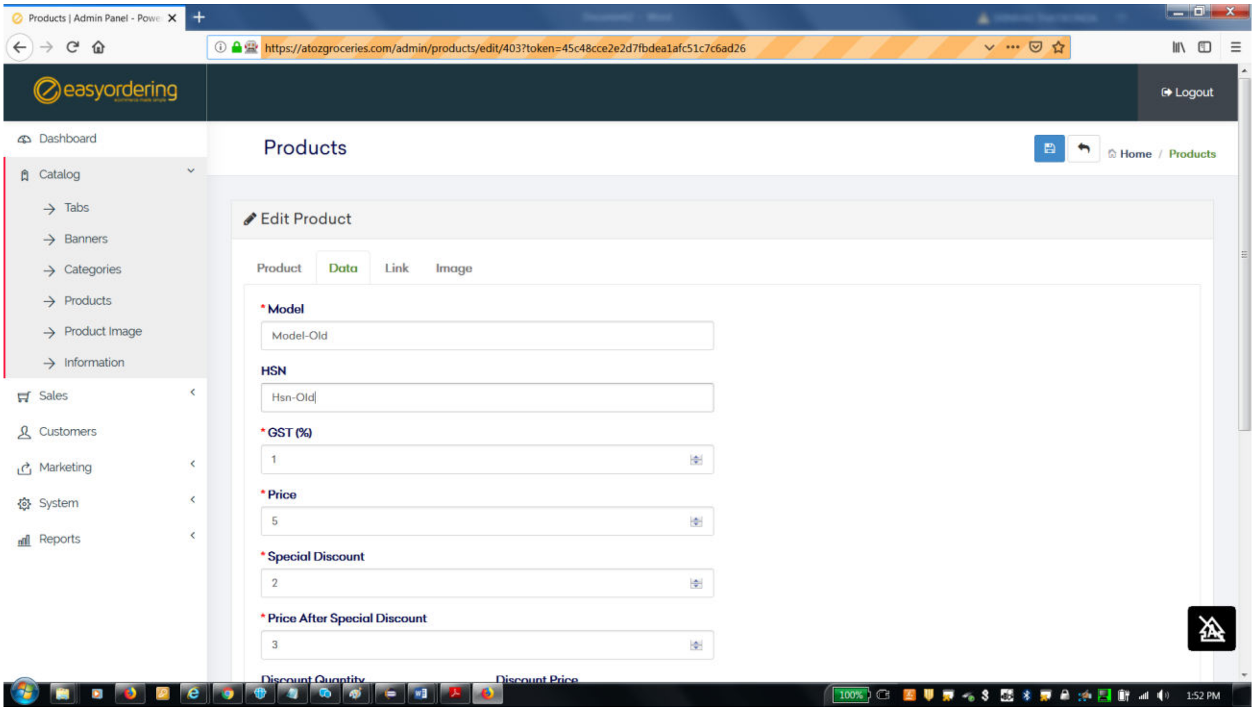
Title-Old

Meta Tag Description

Tag-Old

Meta Tag Keywords

keyword-Old



Products

Edit Product

* Model

Model-Old

HSN

Hsn-Old

* GST (%)

1

* Price

5

* Special Discount

2

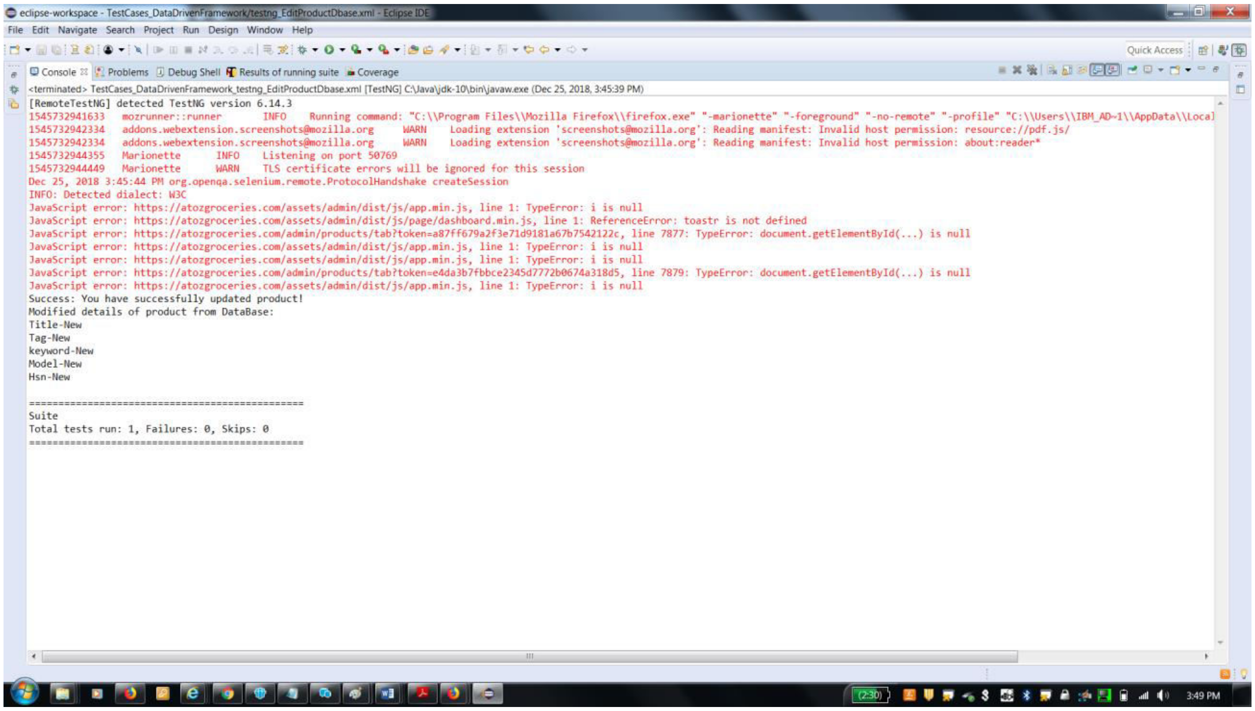
* Price After Special Discount

3

Discount Quantity

Discount Price

Console Output;

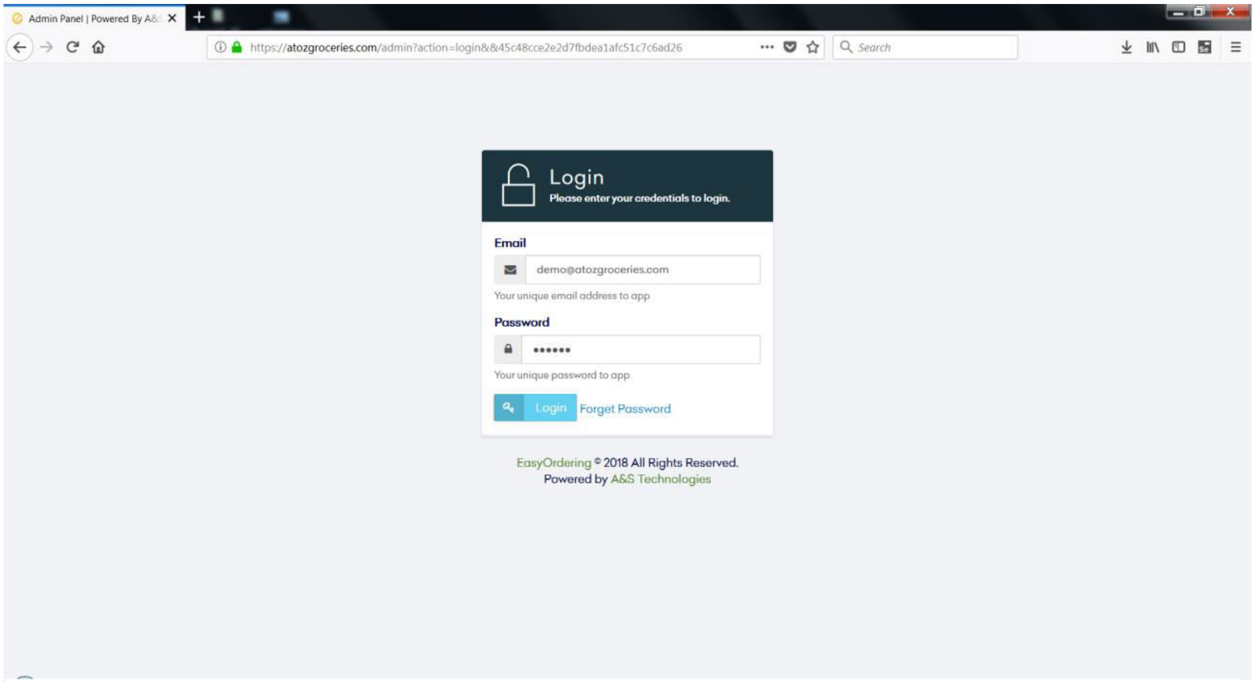


eclipse-workspace - TestCases_DataDrivenFramework/testng_EditProductDbase.xml - Eclipse IDE
File Edit Navigate Search Project Run Design Window Help
Quick Access
Console
Problems
Debug Shell
Results of running suite
Coverage
3:49 PM

```
<terminated> TestCases_DataDrivenFramework_testng_EditProductDbase.xml [TestNG] C:\Java\jdk-10\bin\javaw.exe (Dec 25, 2018, 3:45:39 PM)
[RemoteTestNG] detected TestNG version 6.14.3
1545732941633	mozrunner::runner	INFO	Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "-marionette" "-foreground" "-no-remote" "-profile" "C:\\Users\\IBM_AD~1\\AppData\\Local
1545732942334	addons.webextension.screenshots@mozilla.org	WARN	Loading extension 'screenshots@mozilla.org': Reading manifest: Invalid host permission: resource://pdf.js/
1545732942334	addons.webextension.screenshots@mozilla.org	WARN	Loading extension 'screenshots@mozilla.org': Reading manifest: Invalid host permission: about:reader*
1545732944355	Marionette	INFO	Listening on port 50769
1545732944449	Marionette	WARN	TLS certificate errors will be ignored for this session
Dec 25, 2018 3:45:44 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=a87ff679a2f3e71d9181a67b7542122c, line 7877: TypeError: document.getElementById(...) is null
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/app.min.js, line 1: TypeError: i is null
JavaScript error: https://atozgroceries.com/admin/products/tab?token=e4da3b7fbbce2345d7772b0674a318d5, line 7879: TypeError: document.getElementById(...) is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
Success: You have successfully updated product!
Modified details of product from DataBase:
Title-New
Tag-New
keyword-New
Model-New
Hsn-New

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

Admin portal page is displayed with user name and password



Admin Panel | Powered By A&S
https://atozgroceries.com/admin?action=login&&45c48cce2e2d7fbdea1afc51c7c6ad26
Search
Login
Login
Forget Password

Please enter your credentials to login.

Email: demo@atozgroceries.com

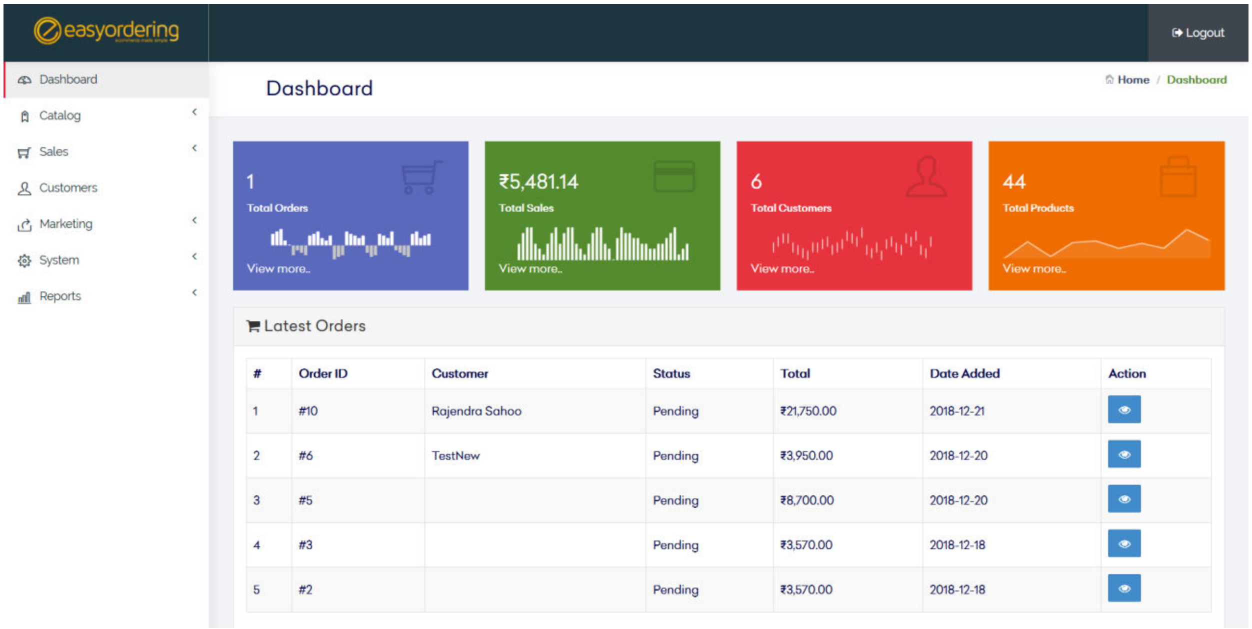
Your unique email address to app

Password: ••••••

Your unique password to app

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

Dashboard page is displayed



## Dashboard

1 Total Orders

₹5,481.14 Total Sales

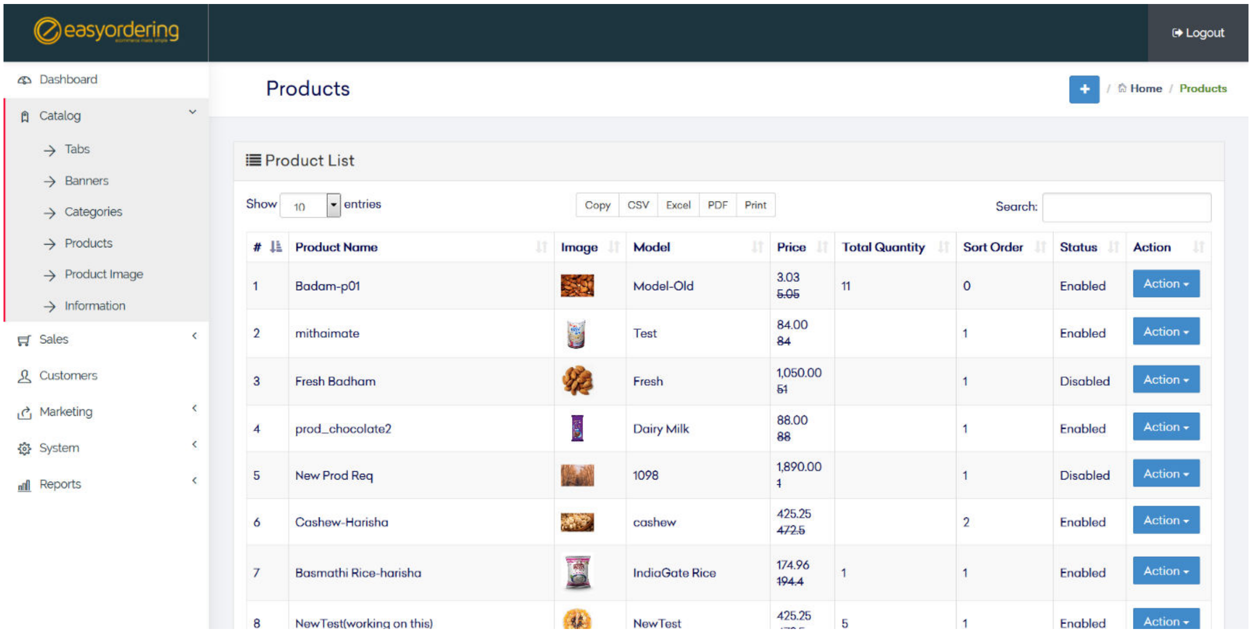
6 Total Customers

44 Total Products

### Latest Orders

| # | Order ID | Customer | Status | Total | Date Added | Action |
|---|---|---|---|---|---|---|
| 1 | #10 | Rajendra Sahoo | Pending | ₹21,750.00 | 2018-12-21 | |
| 2 | #6 | TestNew | Pending | ₹3,950.00 | 2018-12-20 | |
| 3 | #5 | | Pending | ₹8,700.00 | 2018-12-20 | |
| 4 | #3 | | Pending | ₹3,570.00 | 2018-12-18 | |
| 5 | #2 | | Pending | ₹3,570.00 | 2018-12-18 | |

Products list is displayed



## Products

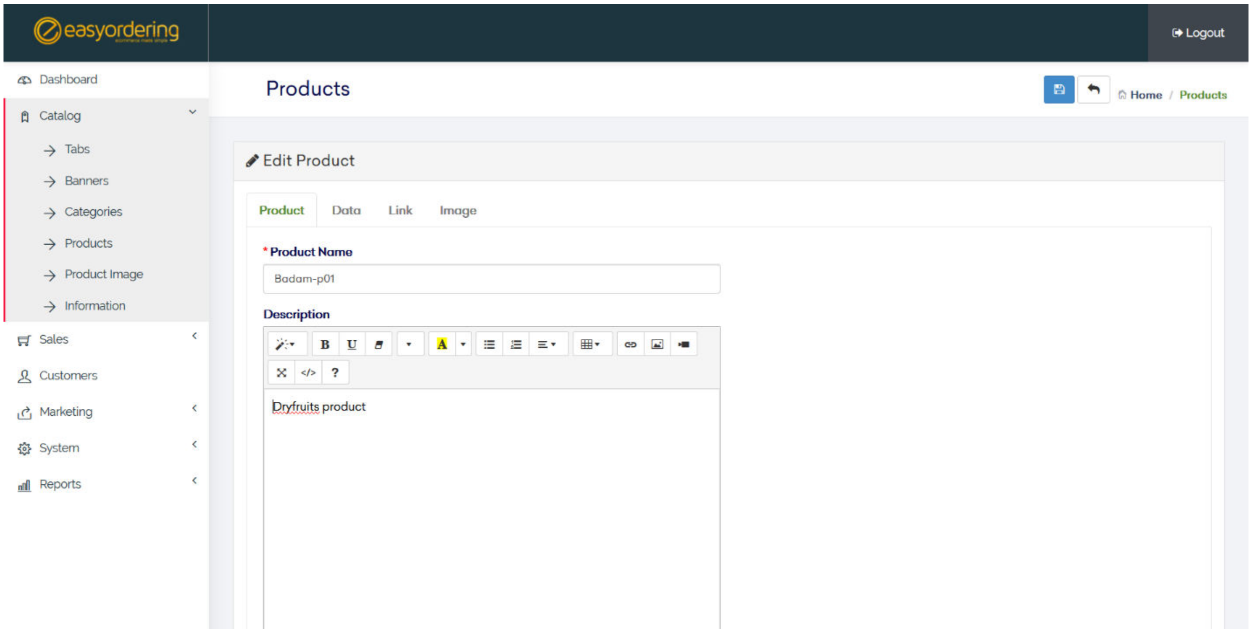
### Product List

Show 10 entries

Search:

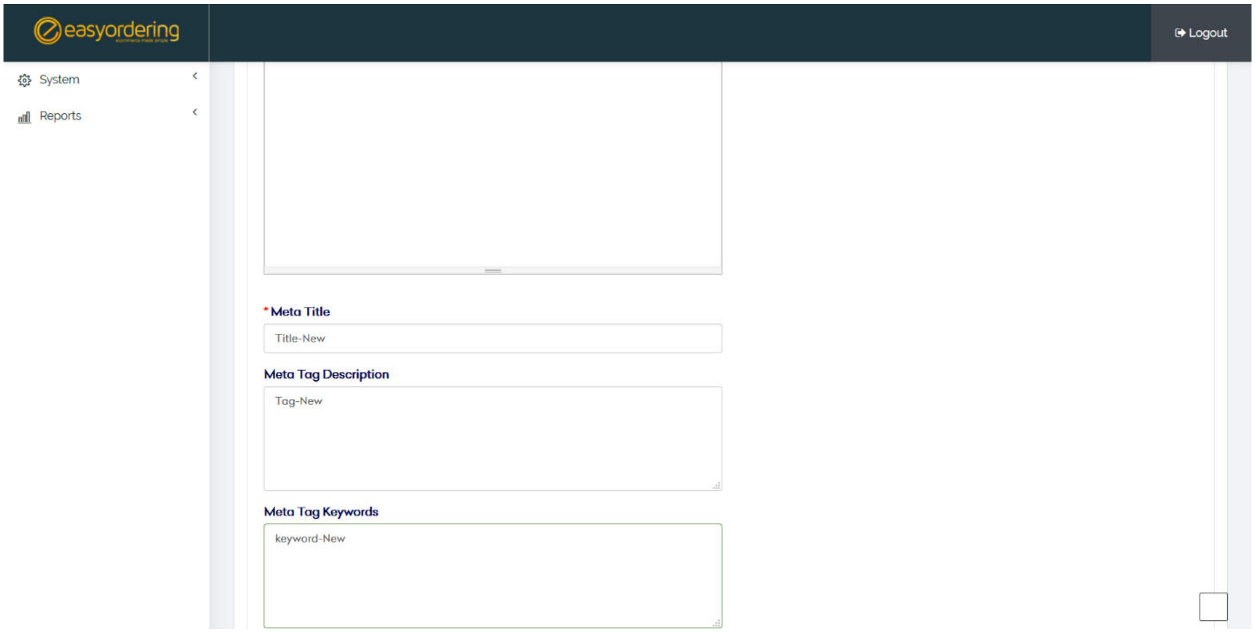
| # | Product Name | Image | Model | Price | Total Quantity | Sort Order | Status | Action |
|---|---|---|---|---|---|---|---|---|
| 1 | Badam-p01 | | Model-Old | 3.03<br>~~5.05~~ | 11 | 0 | Enabled | Action |
| 2 | mithaimate | | Test | 84.00<br>~~84~~ | | 1 | Enabled | Action |
| 3 | Fresh Badham | | Fresh | 1,050.00<br>~~51~~ | | 1 | Disabled | Action |
| 4 | prod_chocolate2 | | Dairy Milk | 88.00<br>~~88~~ | | 1 | Enabled | Action |
| 5 | New Prod Req | | 1098 | 1,890.00<br>~~1~~ | | 1 | Disabled | Action |
| 6 | Cashew-Harisha | | cashew | 425.25<br>~~472.5~~ | | 2 | Enabled | Action |
| 7 | Basmathi Rice-harisha | | IndiaGate Rice | 174.96<br>~~194.4~~ | 1 | 1 | Enabled | Action |
| 8 | NewTest(working on this) | | NewTest | 425.25 | 5 | 1 | Enabled | Action |

Edit Product page is displayed with new values for meta title,meta tag description,Meta tag keywords,model,Hsn



* Product Name: Badam-p01

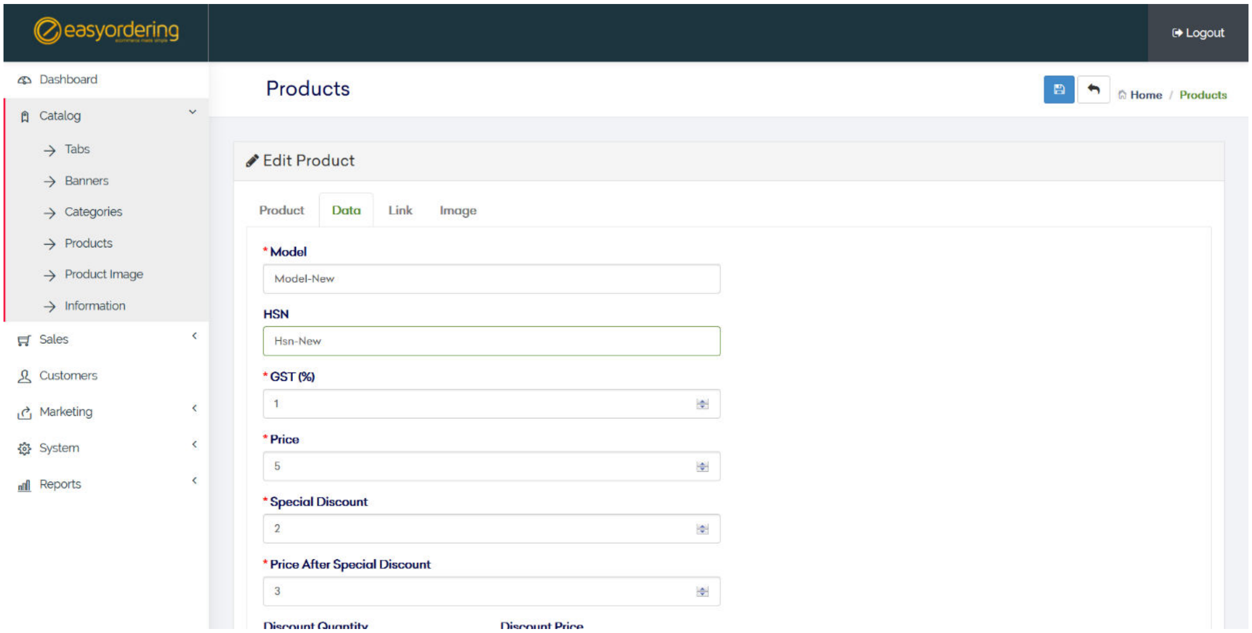
Description: Dryfruits product



* Meta Title: Title-New

Meta Tag Description: Tag-New

Meta Tag Keywords: keyword-New



## Products

Edit Product

* Model

Model-New

HSN

Hsn-New

* GST (%)

1

* Price

5

* Special Discount

2

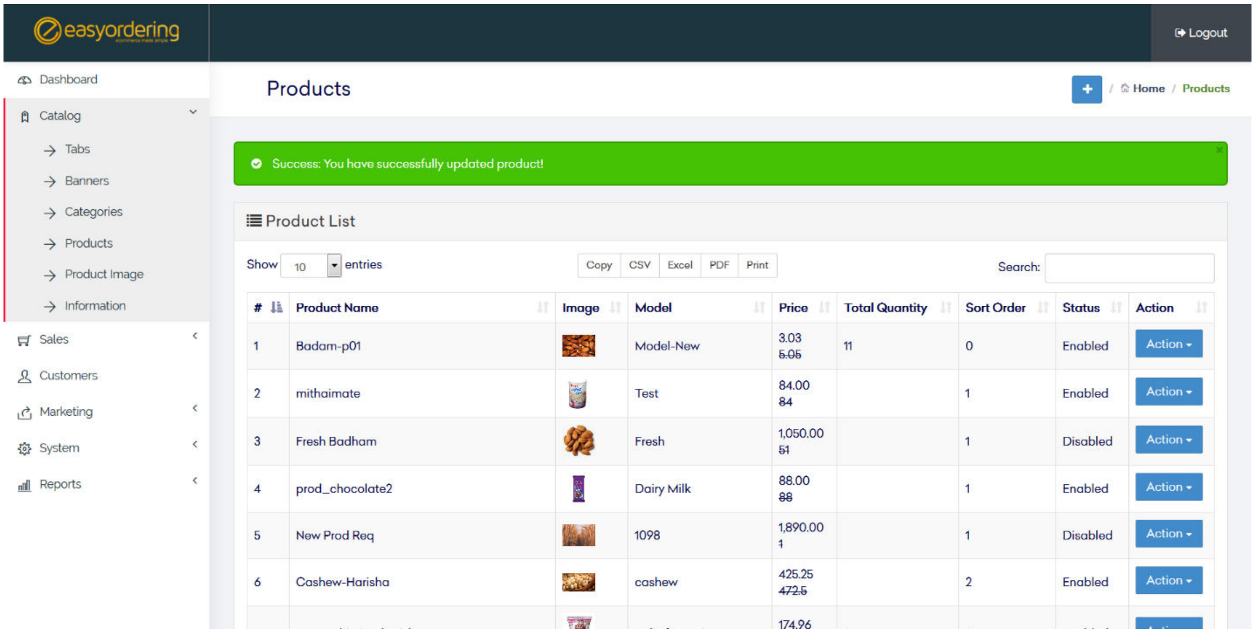
* Price After Special Discount

3

Discount Quantity

Discount Price

Confirmation message is displayed after product is updated.



## Products

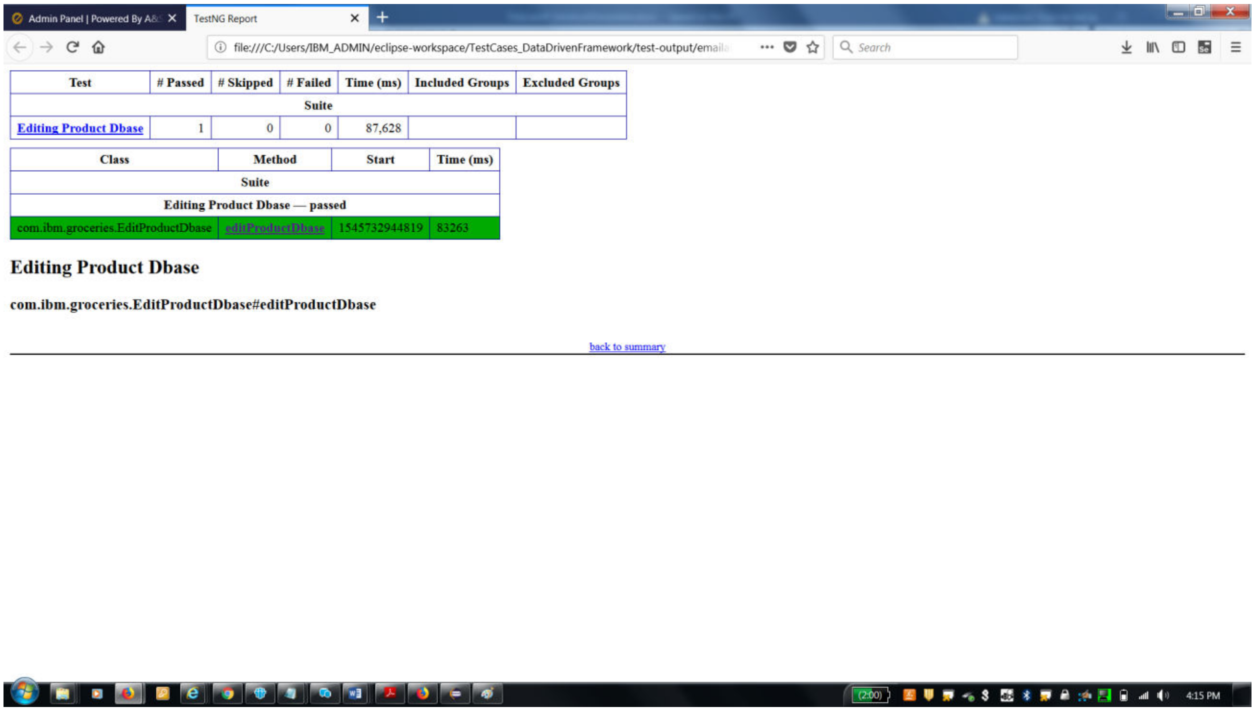
Success: You have successfully updated product!

Product List

Show 10 entries

Search:

| # | Product Name | Image | Model | Price | Total Quantity | Sort Order | Status | Action |
|---|---|---|---|---|---|---|---|---|
| 1 | Badam-p01 | | Model-New | 3.03<br>~~5.05~~ | 11 | 0 | Enabled | Action ▾ |
| 2 | mithaimate | | Test | 84.00<br>~~84~~ | | 1 | Enabled | Action ▾ |
| 3 | Fresh Badham | | Fresh | 1,050.00<br>~~51~~ | | 1 | Disabled | Action ▾ |
| 4 | prod_chocolate2 | | Dairy Milk | 88.00<br>~~88~~ | | 1 | Enabled | Action ▾ |
| 5 | New Prod Req | | 1098 | 1,890.00<br>~~1~~ | | 1 | Disabled | Action ▾ |
| 6 | Cashew-Harisha | | cashew | 425.25<br>~~472.5~~ | | 2 | Enabled | Action ▾ |
| | | | | 174.96 | | | | |



Admin Panel | Powered By A&
TestNG Report
file:///C:/Users/IBM_ADMIN/eclipse-workspace/TestCases_DataDrivenFramework/test-output/emaila
Search
4:15 PM

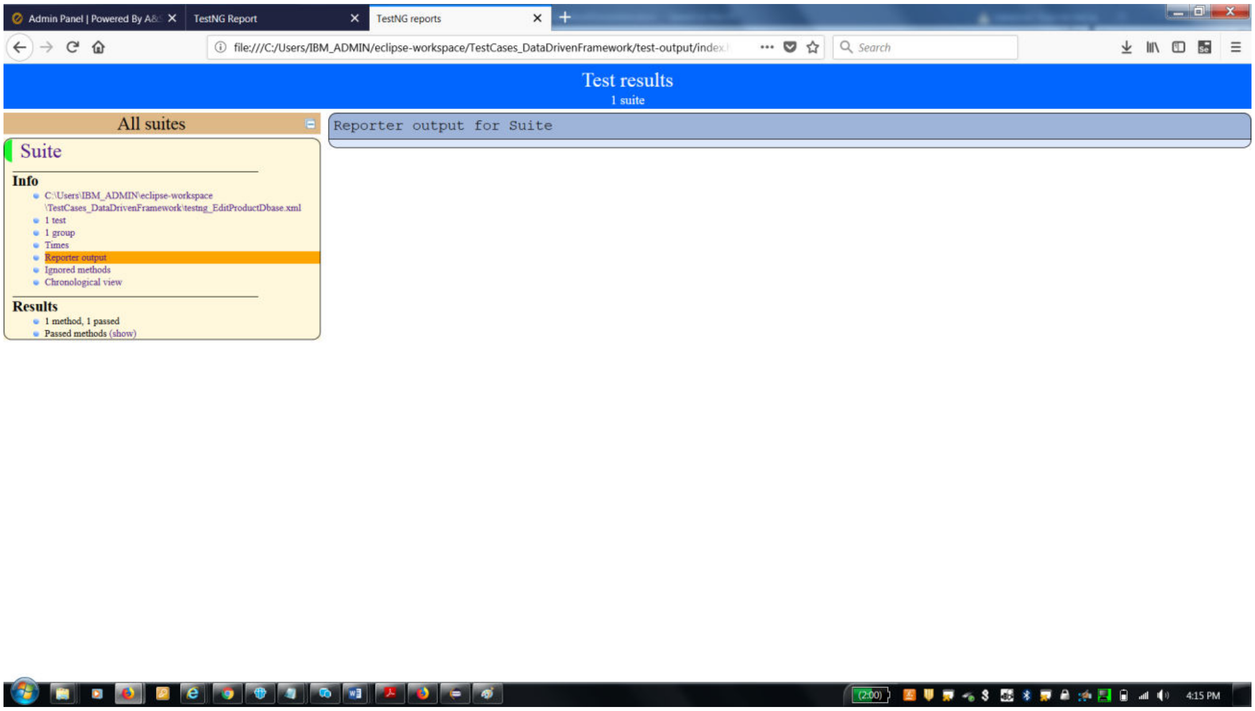
| Test | # Passed | # Skipped | # Failed | Time (ms) | Included Groups | Excluded Groups |
|---|---|---|---|---|---|---|
| Suite | | | | | | |
| Editing Product Dbase | 1 | 0 | 0 | 87,628 | | |

| Class | Method | Start | Time (ms) |
|---|---|---|---|
| Suite | | | |
| Editing Product Dbase — passed | | | |
| com.ibm.groceries.EditProductDbase | editProductDbase | 1545732944819 | 83263 |

### Editing Product Dbase

**com.ibm.groceries.EditProductDbase#editProductDbase**

back to summary



Admin Panel | Powered By A&
TestNG Report
TestNG reports
file:///C:/Users/IBM_ADMIN/eclipse-workspace/TestCases_DataDrivenFramework/test-output/index.
Search
4:15 PM

Test results

1 suite

All suites

Suite

**Info**

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

**Results**

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

```
Reporter output for Suite
```

**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,user name,password,product name to edit,new name for meta title,meta tag description,meta tag keyword,model,Hsn and 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 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 class 'EditProductsPage' uses FindBy annotation to locate the elements product name,Meta Title,Meta Tag Description,Meta Tag keywords, data link, model ,HSN and Save link.The method editProductDbase is defined to enter the new values Meta Title, Meta Tag Description, Meta Tag keywords , model,HSN using sendkeys method.Click method is used to click on Save button and pasgesource is returned.

The class DatabaseConnection defines the method 'connectDatabase' to connect with database and returns the statement.

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

The annotation @Test is used with method editProductDbase in the class 'EditProductDbase'. This method is used to get the values for username,password , product name to edit,new name for meta title,meta tag description,meta tag keyword,model,Hsn and validation message etc. The get method is called to launch the web url for admin portal . The objects for classes PageLogin,PageDashboard, PageProducts,EditProductsPage ,DatabaseConnection are created and the above necessary methods are called on these objects. ResultSet is used to execute the query to retrieve the new values of modified product based on product name and next method used on result set object

Finally the Assert is used to verify presence of the modified product, new value of the meta title,meta tag description,meta tag keyword,model,Hsn and expected message when product updated in admin page.