

Objective:

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

Test Case:TC01

Process: Delete the product

Steps to execute:

Step1: Login to admin portal i.e. <https://atozgroceries.com/admin>

Step2: Delete product

Expected Results: Validate the deleted product

Source code:

File1: DeleteProduct.java

```
package com.ibm.groceries;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Date;
import java.util.HashMap;
import java.util.Properties;
import java.util.Set;
import java.util.concurrent.TimeUnit;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.By;
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.ie.InternetExplorerDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.AfterTest;
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 org.testng.annotations.Test;
import com.ibm.groceriespages.PageLogin;
import com.ibm.groceriespages.PageDashboard;
import com.ibm.groceriespages.PageProducts;
import com.ibm.initialization.WebDriverLaunch;
import com.ibm.utilities.PropertiesFileHandler;
import jdk.nashorn.internal.runtime.PrototypeObject;

public class DeleteProduct extends WebDriverLaunch{

    @Test(priority=1, testName="CheckDeleteProduct",groups="low")
    public void deleteProduct()throws IOException,InterruptedException
    {
        String url=data.get("url");
        String userName = data.get("username");
```

```

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

//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();

//To click on Delete button for a product
PageProducts product=new PageProducts(driver, wait);
String pageSource=product.deleteProduct();

//Verifying whether the expected message displayed after deleting product
if(pageSource.contains(expMessage))
System.out.println(expMessage);
Assert.assertTrue(pageSource.contains(expMessage),"Assertion on deletion of
prouduct");
}

```

```
}
```

File2: 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.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(groups= {"high", "low"})
    public void preSetForTest() throws IOException {
```

```

        String file = "./TestData/groceries.properties";
        propFileHandler = new PropertiesFileHandler();
        data = propFileHandler.getPropertiesAsMap(file);
    }

```

```

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

```

```

//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;
    }
}
}

```

File3: 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();
```

```
}
```

```
}
```

File4: 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();
    }
}

```

File5: 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.FindingBy;
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;

    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();

    }

}

```

File6: 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.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");

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 displayed");
        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();
}
}

```

File7:groceries.properties

```

url=https://atozgroceries.com/admin
username=demo@atozgroceries.com
password=456789
expectedMessage=Success: You have successfully deleted data!

```

File8:testing.xml

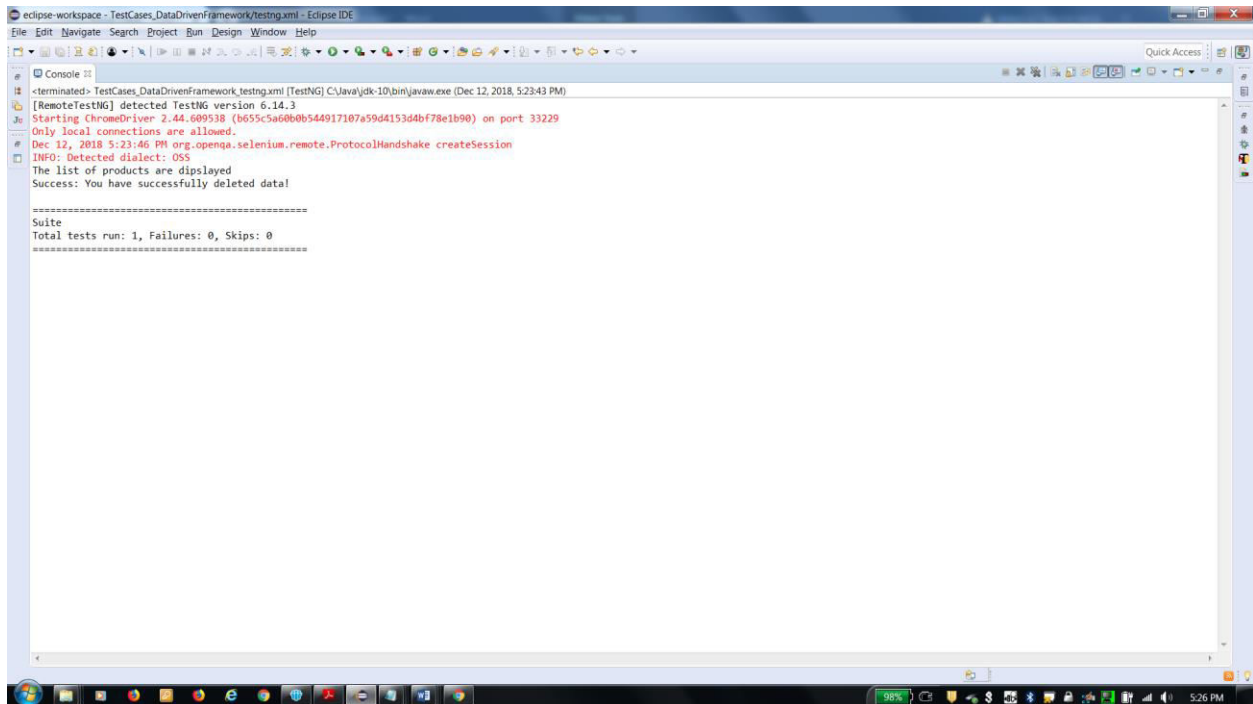
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <groups>
    <run>
      <exclude name="medium"></exclude>
    </run>
  </groups>
  <test thread-count="5" name="TestDeleteProduct">
    <parameter name="browser" value="ch"></parameter>
    <classes>
      <class name="com.ibm.groceries.DeleteProduct">
      </class>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

Test Result:

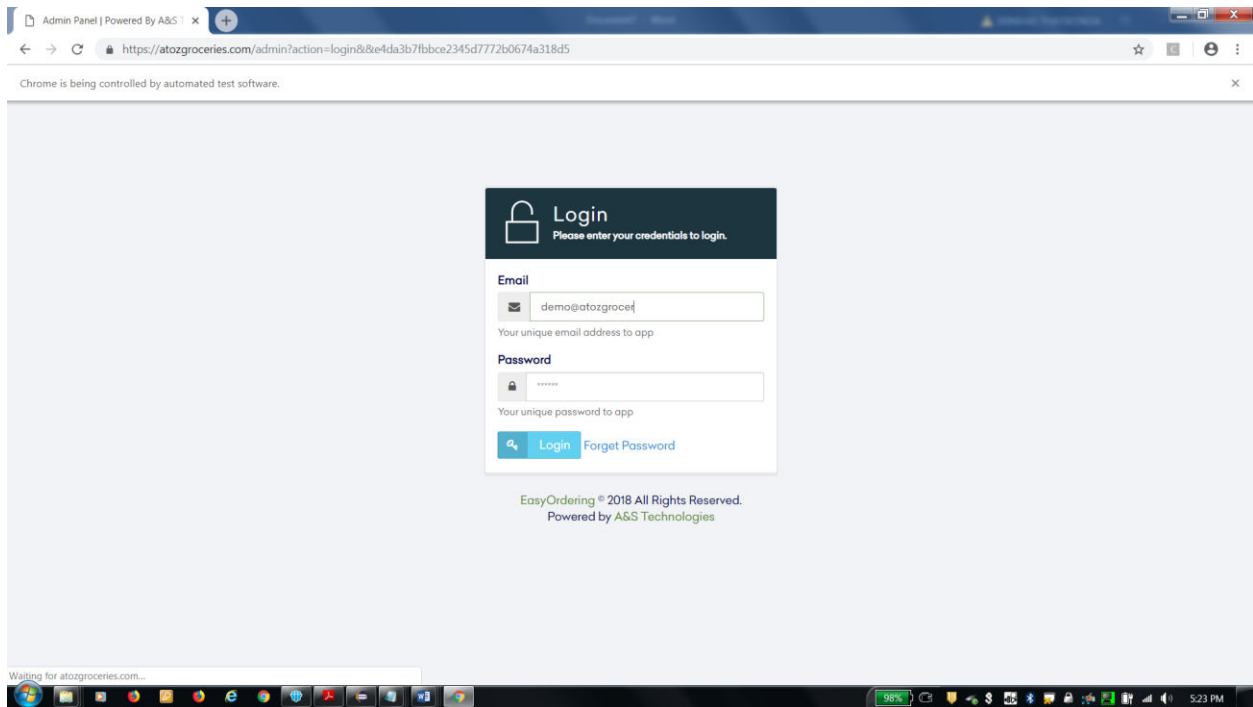
The product is deleted successfully and verified for the confirmation message of product deletion.

Screenshots:

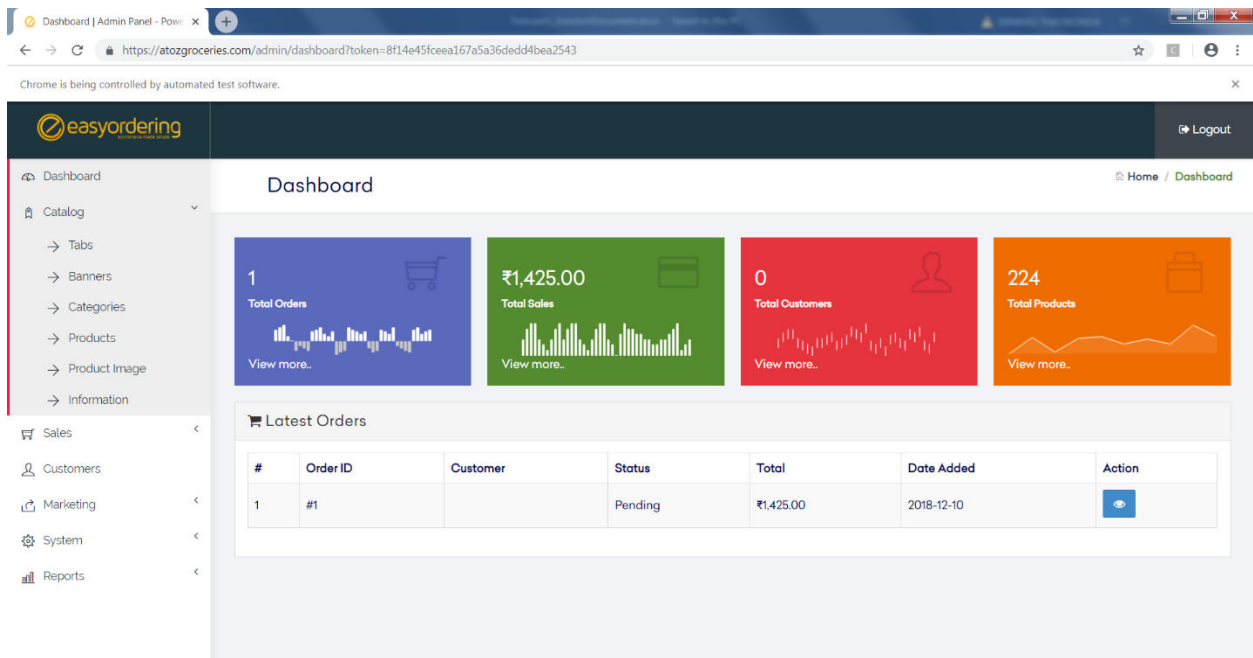
The out put console:



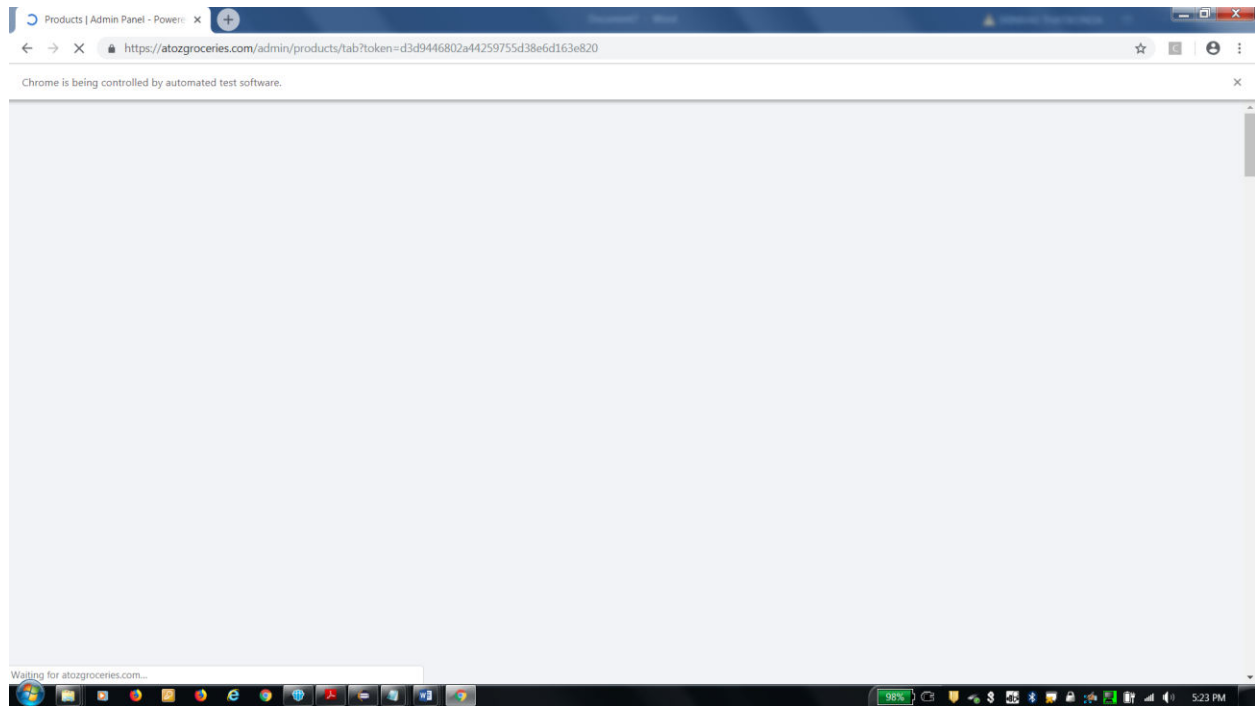
Entering username and password in admin portal . Clickin on Login button



The dashboard is displayed and Click on Catalog and then click on Products on left hand side menu



Displaying Products page.




The list of products displayed

Products | Admin Panel - Power: X

← → ↻ <https://atozgroceries.com/admin/products/tab?token=d3d9446802a44259755d38e6d163e820> ☆ ⓘ ⚙

Chrome is being controlled by automated test software.

 Logout

Dashboard

Catalog

- Tabs
- Banners
- Categories
- Products
- Product Image
- Information

Sales

Customers

Marketing

System

Reports

Products








+ / Home / Products

Product List

Show 10 entries

Copy CSV Excel PDF Print

Search:

#	Product Name	Image	Model	Price	Total Quantity	Sort Order	Status	Action
1	dynamics sabbakki fine 5 kg		good quality	236.25 236.25	10	1	Enabled	Action
2	dynamic cashew baby 1kg		good quality	420.00 420		1	Enabled	Action
3	dynamic white till 10 kg		good quality	1785.00 86		1	Enabled	Action
4	dynamic white till 5 kg		good quality	892.5 892.5		1	Enabled	Action
5	dynamic white till 1kg		good quality	189.00 189		1	Enabled	Action
6	dynamic pepper atom 1kg		good quality	451.5 451.5		1	Enabled	Action
7	dynamic pepper 1kg fine		good quality	462.00 462		1	Enabled	Action

98% 5:25 PM

Click on Action button->Select Delete button for the first product

Products | Admin Panel - Power: x

https://atozgroceries.com/admin/products/tab?token=a87ff679a2f3e71d9181a67b7542122c

Chrome is being controlled by automated test software.

easystore Logout

Dashboard

Catalog

- Tabs
- Banners
- Categories
- Products
- Product Image
- Information

Sales

Customers

Marketing

System

Reports

Products








+ Home / Products

Product List

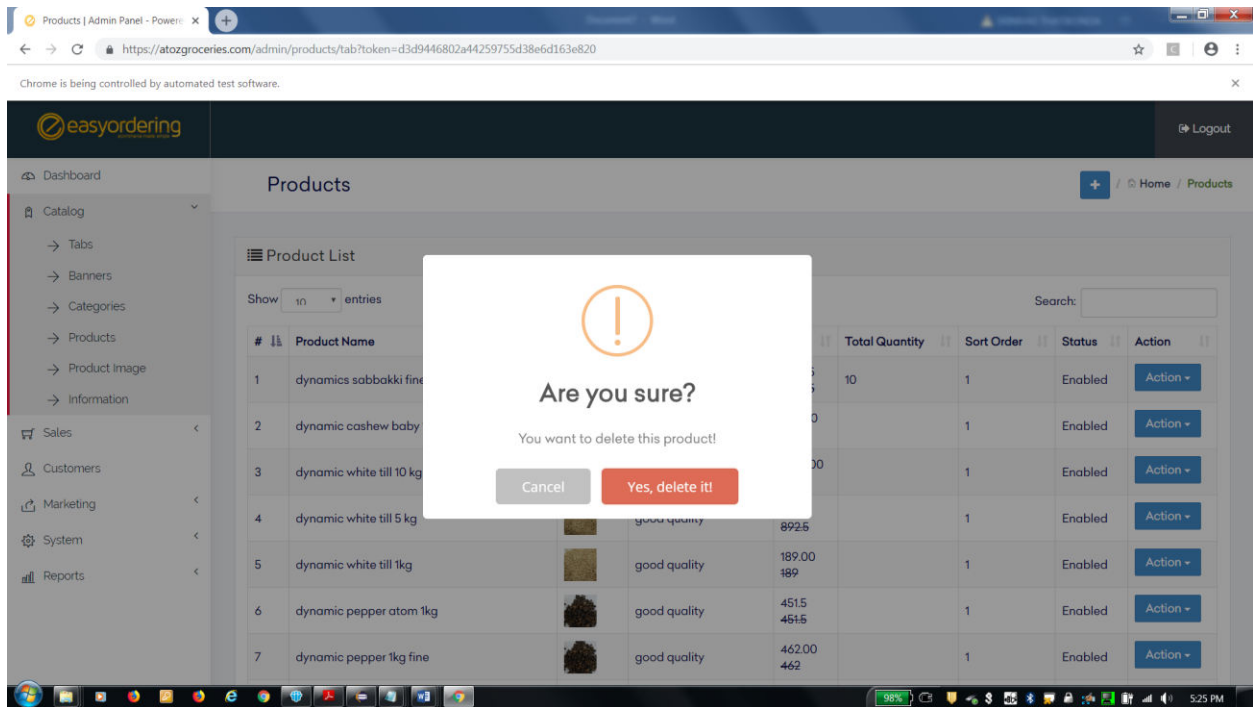
Show 10 entries

Copy CSV Excel PDF Print

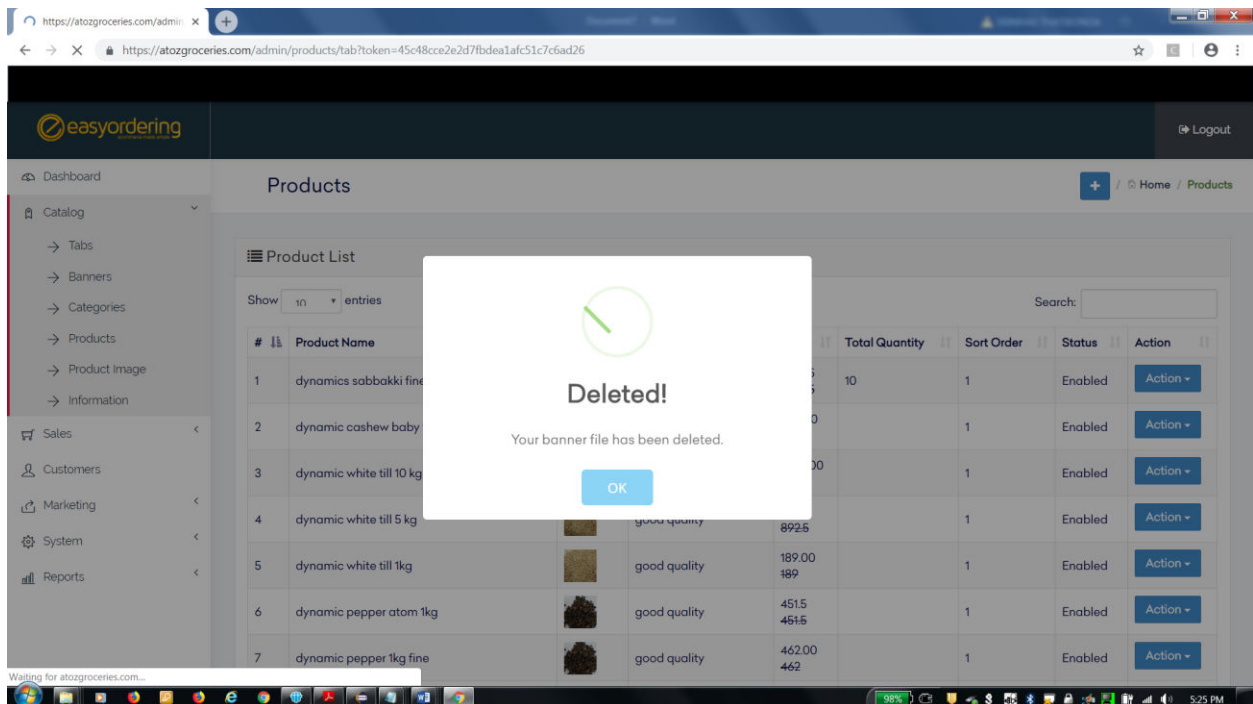
Search:

#	Product Name	Image	Model	Price	Total Quantity	Sort Order	Status	Action
1	dynamics sabbakki fine 5 kg		good quality	234.25 236.25	10	1	Enabled	Action ▾ Edit Delete
2	dynamic cashew baby 1kg		good quality	420.00 420		1	Enabled	Action ▾
3	dynamic white till 10 kg		good quality	1,785.00 86		1	Enabled	Action ▾
4	dynamic white till 5 kg		good quality	892.5 892.5		1	Enabled	Action ▾
5	dynamic white till 1kg		good quality	189.00 189		1	Enabled	Action ▾
6	dynamic pepper atom 1kg		good quality	451.5 451.5		1	Enabled	Action ▾
7	dynamic pepper 1kg fine		good quality	462.00 462		1	Enabled	Action ▾

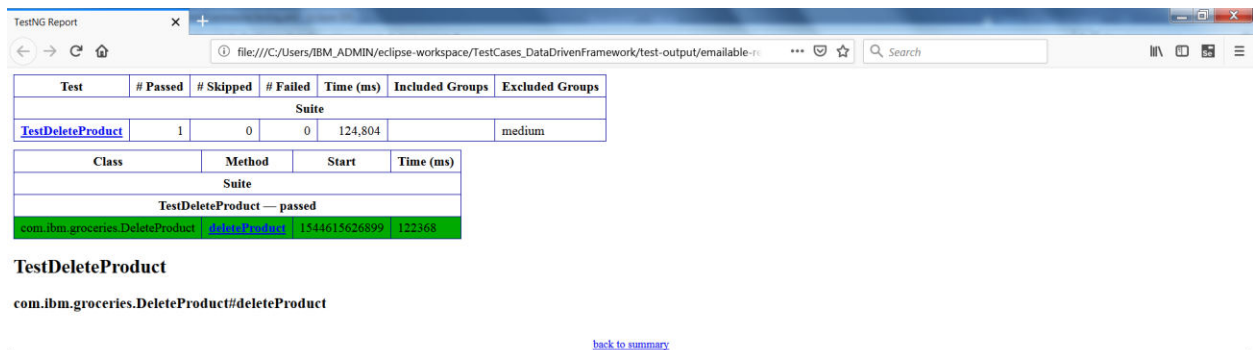
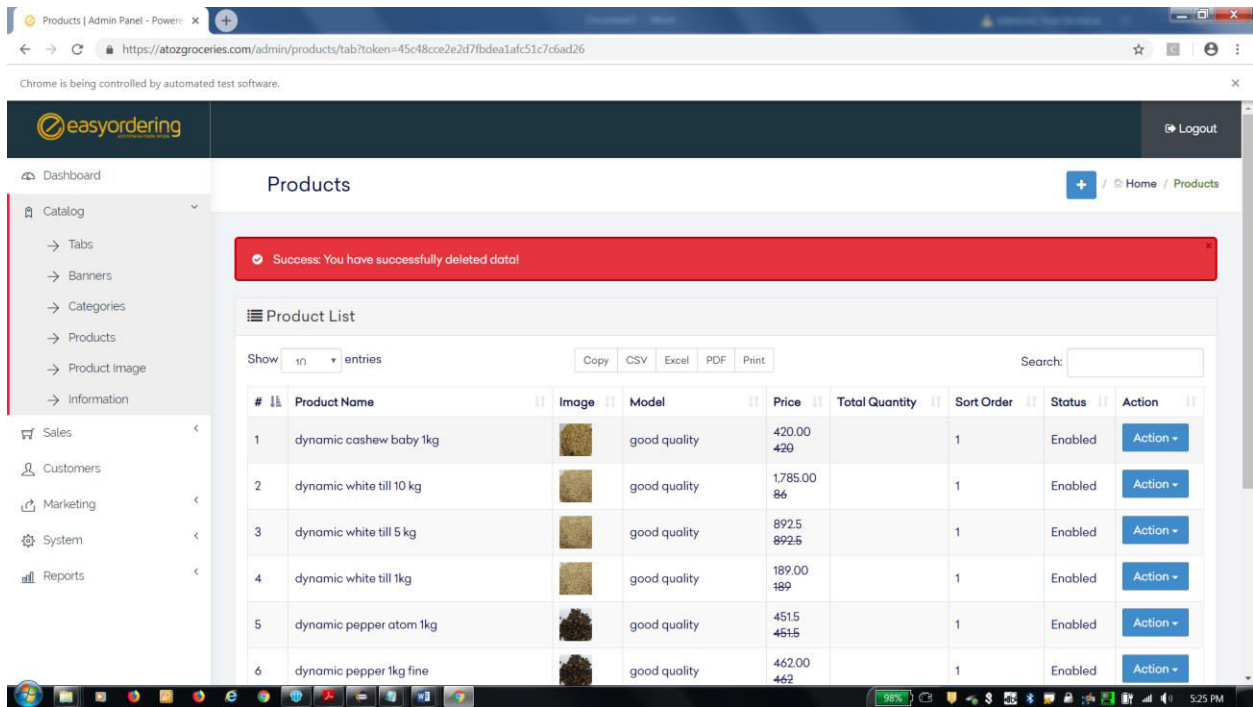
A confirmation is displayed with two buttons to delete the product .Click on 'Yes delete it' button

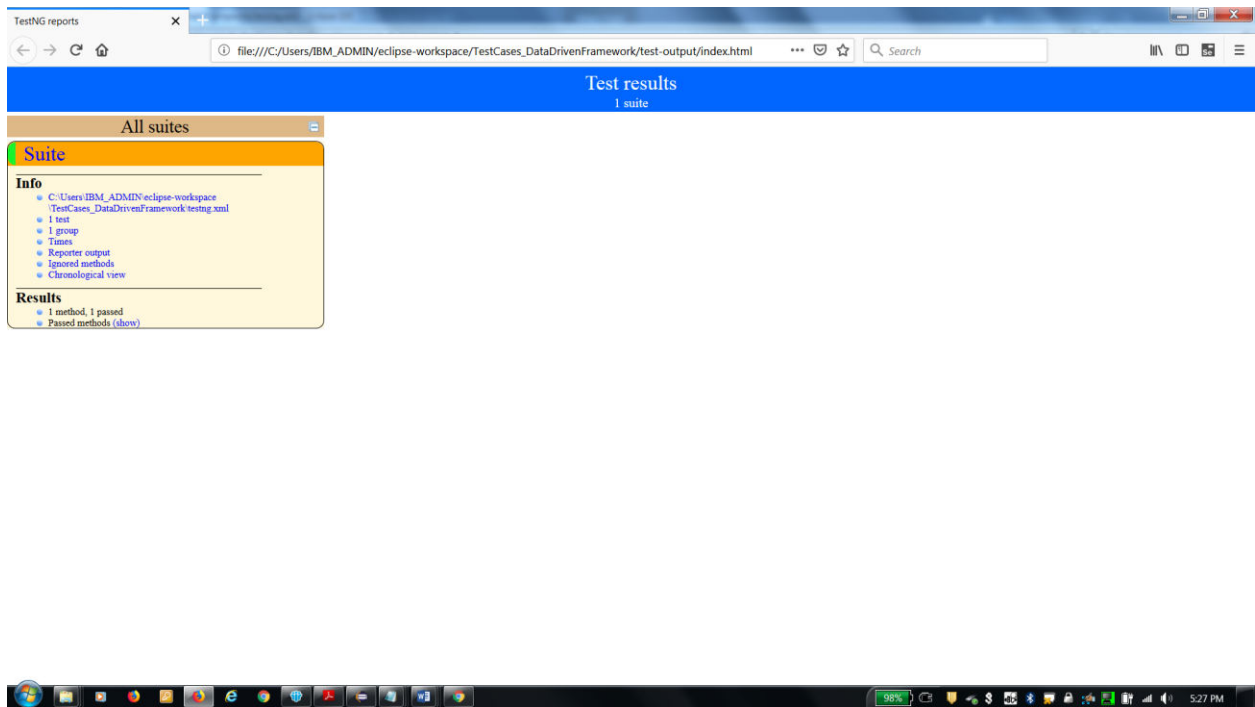


A popup is displayed indicating product is deleted



The expected message “ Success: You have successfully deleted data!” is displayed on the page.





Flow information:

Initially a property file 'groceries.properties' is created in Test Data folder of the project and list of key, values for url, username,password ,expected 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 method is used in the 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,product. PageFactory method is used in the in the constructor of this class.The method click is used to click on Catalog and Products web elements.

The class 'PageProducts' is defined with methods to locate the list of products displayed,action web element,delete button, confirm button using 'Find By'. PageFactory method is used in the in the constructor of this class. The xpath is defined to locate the first product to delete in the table. The method deleteProduct is defined to check if number of products displayed is greater than zero and then click method is used to click on action button,delete button and confirm button.

The annotation @Test is used with method deleteProduct in the class 'DeleteProduct'. This method is used to get the values for username,password and the expected message to validate. The get method is called to launch the web url for admin portal.The objects for classes PageLogin,PageDashboard,PageProducts are created and the above necessary methods are called on these objects. Finally the Assert is used to verify the expected message for successful deletion of product.The testing xml file is having the parameter 'ch' to launch chrome browser.