

## Day 4 - Test case solution document

### Code Used:

#### BaseTest.java

```
package com.ibm.test;

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.testng.Assert;
import org.testng.annotations.Test;
import com.ibm.pages.CategoryPage;
import com.ibm.pages.CouponPage;
import com.ibm.pages.DashboardPage;
import com.ibm.pages.DriverLaunch;
import com.ibm.pages.LoginPage;
import com.ibm.pages>ReturnsPage;
import com.ibm.pages.TabsPage;

public class BaseTest extends DriverLaunch
{
    @Test(testName="DeleteReturnReason")
    public void Day4TestCase() throws InterruptedException
    {
        LoginPage login=new LoginPage(driver,wait);
        DashboardPage dash=new DashboardPage(driver,wait);
        ReturnsPage returns=new ReturnsPage(driver,wait);
        try
        {
            String url=hashData.get("url");
            String username=hashData.get("username");
            String password=hashData.get("password");
            String show=hashData.get("show");
            String returnreason=hashData.get("returnreason");
            //String expdeletemsg=hashData.get("expdeletemsg");
            driver.get(url);

            //Login into the Admin portal
            login.EnterEmail(username);
            login.EnterPassword(password);
            login.clickLogin();

            //Opening System from the dashboard
            dash.OpenSystem();

            //Opening Return Reasons
            returns.OpenReturnReasons();
            //Displaying all return reasons
            returns.displayReturnReasons(show);
        }
    }
}
```

```

        //Deleting the return reason
        returns.deleteReturnReason(returnreason);
        //getting the page source for validating the deleted reason
        String actualpgSrc=returns.validatingTableAfterDelete(returnreason);
        Assert.assertFalse(actualpgSrc.contains(returnreason), "Assertion on Deleted
Return Reason!!");
    }
catch(Exception e)// catching if any exception occurs and takes a screenshot of the
page
{
    TakesScreenshot ts=(TakesScreenshot) driver;
    File file = ts.getScreenshotAs(OutputType.FILE);
    Date date = new Date();
    String currentDate = date.toString().replace(":", "-");
    FileUtils.copyFile(file, new
File("./Screenshots/Day4_TestCase_Error_" + currentDate + ".jpg"));
    System.out.println(e.getMessage());
    Assert.fail();
}
finally
{
    login.ClickLogout();
}
}
}

```

## LoginPage.java

```

package com.ibm.pages;

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

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

    @FindBy(xpath="/html/body/div/div/div/div[2]/form/button")
    WebElement loginEle;

    @FindBy(xpath="//a[@title='Logout']")
    WebElement logoutEle;

    WebDriverWait wait;
    WebDriver driver;

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

```

```

        }
        public void EnterEmail(String username)
        {
            emailEle.sendKeys(username);
        }
        public void EnterPassword(String password)
        {
            passwordEle.sendKeys(password);
        }
        public void clickLogin()
        {
            loginEle.click();
        }
        public void ClickLogout()
        {
            logoutEle.click();
        }
    }
}

```

## **DashboardPage.java**

```

package com.ibm.pages;

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 DashboardPage {

    //Finding the WebElements
    @FindBy(partialLinkText="Catalog")
    WebElement catalogEle;

    @FindBy(xpath="/html/body/div[1]/div[1]/div/ul/li[5]/a")
    WebElement marketEle;

    @FindBy(xpath="//*[contains(text(),'System')]")
    WebElement systemEle;

    WebDriver driver;
    WebDriverWait wait;

    //catalog page constructor
    public DashboardPage(WebDriver driver, WebDriverWait wait)
    {
        this.driver=driver;
        this.wait=wait;
        PageFactory.initElements(driver, this);
    }

    //Opencatalog method to click on catalog
}

```

```

public void OpenCatalog()
{
    wait.until(ExpectedConditions.visibilityOf(catalogEle));
    catalogEle.click();
}

//method for opening marketing
public void OpenMarketing()
{
    marketEle.click();
}
//method for opening system
public void OpenSystem()
{
    systemEle.click();
}
}

```

### ReturnsPage.java

```

package com.ibm.pages;

import java.util.List;

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

public class ReturnsPage {

    //Finding elements of Returns Page
    //@FindBy(xpath="//*[contains(text(),'Returns')]")
    @FindBy(xpath="/html/body/div[1]/div[1]/div/ul/li[6]/ul/li[6]/a")
    WebElement returnEle;

    //@FindBy(xpath="//*[contains(text(),'Reason')]")
    @FindBy(xpath="/html/body/div[1]/div[1]/div/ul/li[6]/ul/li[6]/ul/li[3]/a")
    WebElement reasonEle;

    @FindBy(name="dataTableExample2_length")
    WebElement showEle;

    WebDriver driver;
    WebDriverWait wait;
    JavascriptExecutor js;
    //Constructor
    public ReturnsPage(WebDriver driver,WebDriverWait wait)
    {
        this.driver=driver;
    }
}

```

```

        this.wait=wait;
        js=(JavascriptExecutor) driver;
        PageFactory.initElements(driver, this);
    }
    //method to open Returns and Return Reasons
    public void OpenReturnReasons() throws InterruptedException
    {
        returnEle.click();
        reasonEle.click();
    }
    //Method to display all return reasons
    public void displayReturnReasons(String showvalue)
    {
        Select showSel=new Select(showEle);
        showSel.selectByVisibleText(showvalue);
    }

    //method to delete return reasons
    public void deleteReturnReason(String returnreason) throws
InterruptedException
    {
        //getting the row count
        List<WebElement>
rowEle=driver.findElements(By.xpath("//table[@id='dataTableExample2']/tbody/tr"));
        int rows=rowEle.size();
        for(int i=1;i<=rows;i++)
        {

            //getting the reason for deleting
            WebElement
reasonEle=driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+
"]/td[2]"));
            js.executeScript("arguments[0].scrollIntoView(true);",
reasonEle);
            String reasonTxt=reasonEle.getText();
            //going to the reason row for deleting
            if(reasonTxt.equals(returnreason))
            {
                //clicking on the action of the deleting row

                driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/
td[3]/descendant::button")).click();
                //clicking on the delete of the deleting row

                driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/
td[3]/descendant::ul/li)[2]")).click();
                Thread.sleep(5000);
                //clicking on the confirmation to delete the return
reason

                driver.findElement(By.xpath("//button[@class='confirm']")).click();
                break;
            }
        }
    }
}

```

```

        public String validatingTableAfterDelete(String returnreason) throws
InterruptedException
    {
        //getting the page source to validate the deleted return reason

        wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//div[contains(@class,'alert-danger')]")));
//validating the deleted return reason in table
        List<WebElement>
rowEle=driver.findElements(By.xpath("//table[@id='dataTableExample2']/tbody/tr"));

        int rows=rowEle.size();
        for(int i=1;i<=rows;i++)
        {

            //getting the reason for deleting
            WebElement
reasonEle=driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+
"]/td[2]"));
            js.executeScript("arguments[0].scrollIntoView(true);",
reasonEle);
            String reasonTxt=reasonEle.getText();
            //going to the reason row for deleting
            if(reasonTxt.equals(returnreason))
            {
                String fail="Deletion of Return Reason failed!";
                System.out.println(fail);
                break;
            }
        }

        return driver.getPageSource();
    }
}

```

## **DriverLaunch.java**

```

package com.ibm.pages;

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 DriverLaunch {

    public WebDriver driver;
    public WebDriverWait wait;
    public PropertiesFileHandler propFileHandle;
    public HashMap<String, String> hashData;

    @BeforeSuite
    public void preSetForTest() throws IOException
    {
        String file=".TestData/data.properties";
        propFileHandle =new PropertiesFileHandler();
        hashData=propFileHandle.getPropertiesAsMap(file);
    }
    @BeforeMethod
    @Parameters({"browser"})
    public void initialisation(@Optional("ch")String browser)
    {
        browserInitialization(browser);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        wait=new WebDriverWait(driver, 60);
    }
    @AfterMethod
    public void closingBrowser()
    {
        driver.quit();
    }

    public void browserInitialization(String browser)
    {
        switch(browser.toLowerCase())
        {
            case "ff":
                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;
        }
    }
}

```

```
    }
}
```

### PropertiesFileHandler.java

```
package com.ibm.utilities;

import java.io.FileInputStream;
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;
    }

}
```

### data.properties

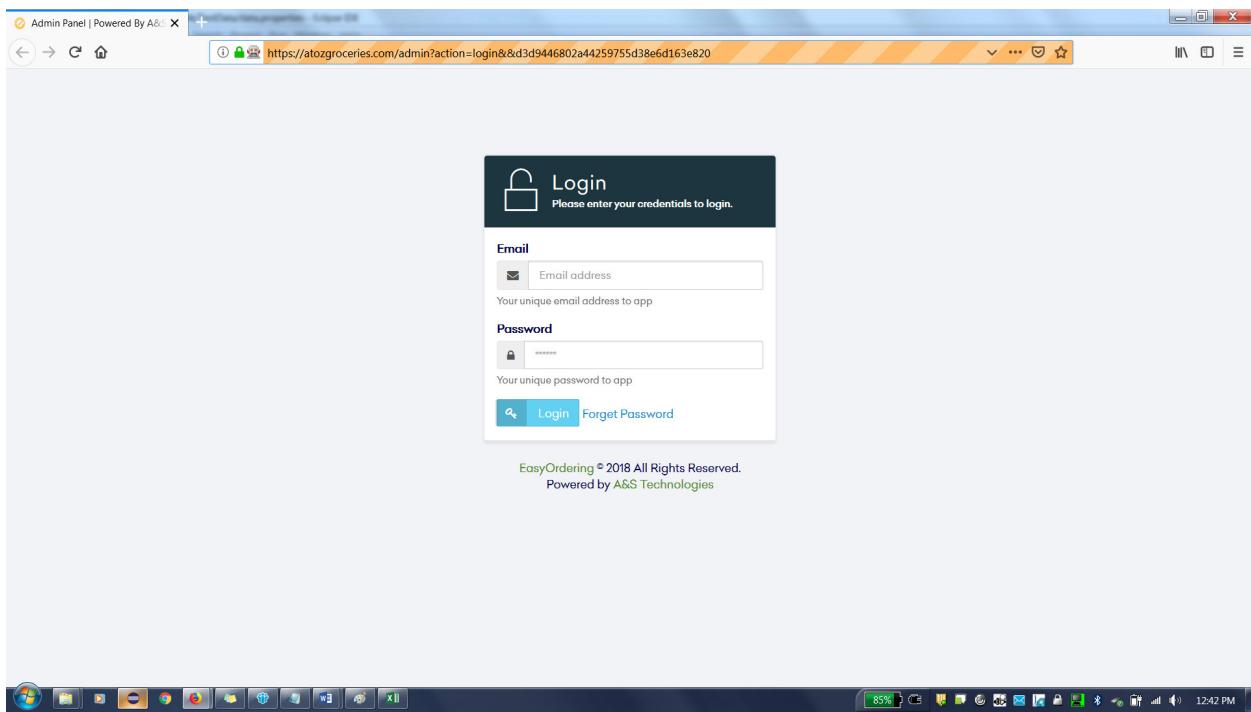
```
url=https://atozgroceries.com/admin
username=demo@atozgroceries.com
password=456789
categoryname=Milk Products
tagtitle=Milk Products
sortorder=12
status=Enabled
show>All
tagname=Milk
tagsort=10
tagstatus=Enabled
tagrequiredmsg=fill out this field
tagSelectReqMsg=select an item in the list
couponcode=15OFF
expdeletemsg=successfully deleted
returnreason=damage
```

### **testng.xml**

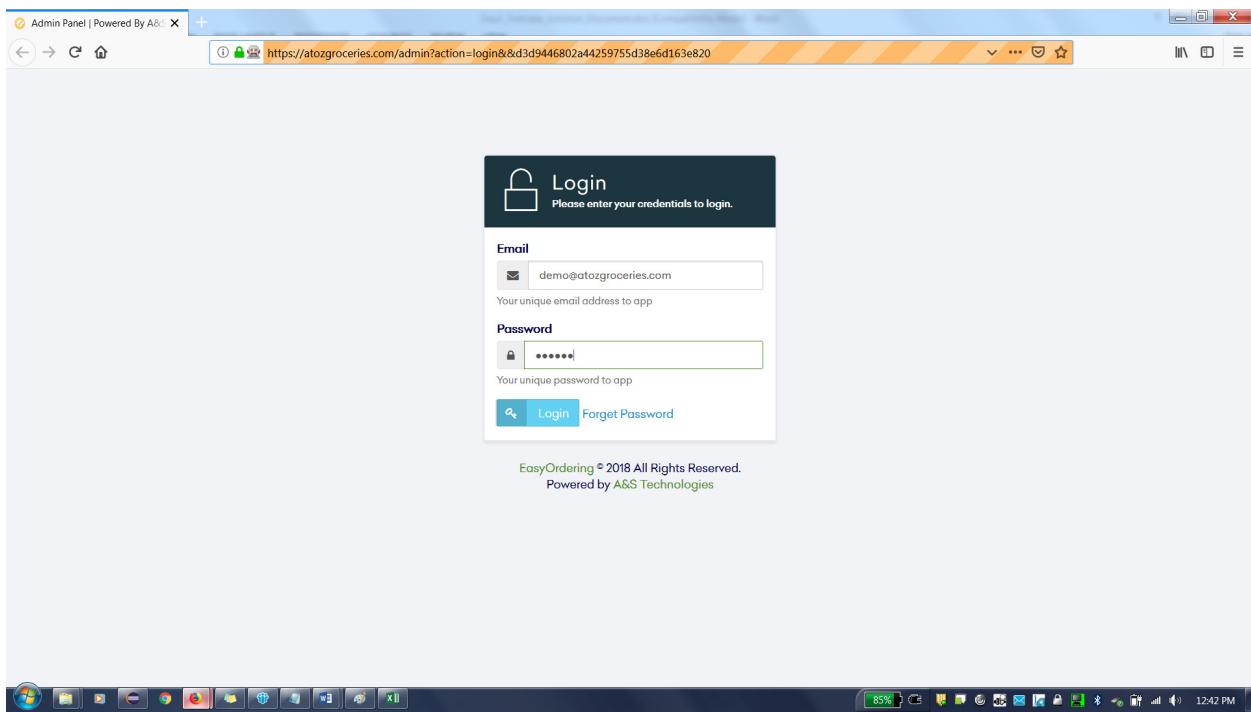
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
    <parameter name="browser" value="ff"></parameter>
    <classes>
      <class name="com.ibm.test.BaseTest"/>
    </classes>
  </test> <!--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

### **Step by step output:**

1. Navigating to the Admin Page of “**atozgroceries**”



## 2. Providing login details



## 3. After logging into the webpage

The screenshot shows the Admin Panel dashboard for 'atozgroceries.com'. The left sidebar includes links for Catalog, Sales, Customers, Marketing, System (with 'Settings' under it), and Reports. The main area displays four key metrics: 0 Total Orders (blue card), ₹698.25 Total Sales (green card), 0 Total Customers (red card), and 106 Total Products (orange card). Below these is a table titled 'Latest Orders' with one entry:

#	Order ID	Customer	Status	Total	Date Added	Action
1	#1		Pending	₹1,425.00	2018-12-10	

#### 4. Opening System

The screenshot shows the Admin Panel dashboard after the system has been opened. The left sidebar now includes 'Marketing' and 'Reports'. The main area displays updated metrics: 3 Total Orders (blue card), ₹2,565.00 Total Sales (green card), 1 Total Customers (red card), and 220 Total Products (orange card). The 'Latest Orders' table now shows three entries:

#	Order ID	Customer	Status	Total	Date Added	Action
1	#3		Pending	₹3,570.00	2018-12-18	
2	#2		Pending	₹3,570.00	2018-12-18	
3	#1		Pending	₹1,425.00	2018-12-10	

#### 5. Opening Returns

**Dashboard**

3 Total Orders

₹2,565.00 Total Sales

1 Total Customers

220 Total Products

#	Order ID	Customer	Status	Total	Date Added	Action
1	#3		Pending	₹3,570.00	2018-12-18	
2	#2		Pending	₹3,570.00	2018-12-18	
3	#1		Pending	₹1,425.00	2018-12-10	

## 6. Opening Return Reasons

**Return Reasons**

#	Return Reason Name	Action
1	Order Error	Action
2	Other, please supply details	Action
3	Received Wrong Item	Action
4	Test	Action
5	testing	Action
6	Not the same order	Action
7	damage	Action
8	small in size	Action

## 7. Selecting “ALL” in show for displaying all Return reasons

The screenshot shows the 'Return Reasons' section of the easyordering Admin Panel. The left sidebar has a 'Returns' section expanded, with 'Return Reasons' selected. The main area displays a table titled 'Return Reason List' with columns for '#', 'Return Reason Name', and 'Action'. The table contains 8 rows with names like 'Order Error', 'Other, please supply details', etc. Action buttons are available for each row.

#	Return Reason Name	Action
1	Order Error	Action ▾
2	Other, please supply details	Action ▾
3	Received Wrong Item	Action ▾
4	Test	Action ▾
5	testing	Action ▾
6	Not the same order	Action ▾
7	damage	Action ▾
8	small in size	Action ▾

## 8. Scrolling to view the delete Return Reason

This screenshot shows the same 'Return Reasons' list after a vertical scroll has been performed. The table now contains 12 rows, with the first 8 from the previous screenshot and additional rows labeled 9 through 12. The 'Action' button is present for each row.

#	Return Reason Name	Action
3	Received Wrong Item	Action ▾
4	Test	Action ▾
5	testing	Action ▾
6	Not the same order	Action ▾
7	damage	Action ▾
8	small in size	Action ▾
9	defect	Action ▾
10	Large in size	Action ▾
11	not satisfied	Action ▾
12	placed wrong item	Action ▾

## 9. Clicking on the Action button of deleting return reason

The screenshot shows the 'Return Reason | Admin Panel' interface. On the left, there's a sidebar with navigation links: Shipping, Order Statuses, Weight Class, Returns (with sub-links: Return Statuses, Return Actions, Return Reasons, Manage Backup), and Reports. The main area displays a table of return reasons with columns for ID, reason name, and an 'Action' dropdown menu. The 8th row, which contains the value 'small in size', has its 'Action' menu open, showing 'Edit' and 'Delete' options. The status bar at the bottom indicates '99%' battery and '1:56 PM'.

10. Clicking on delete button, it will redirect to confirmation page to delete the return reason

This screenshot shows the same Admin Panel interface after a user has clicked the 'Delete' button for the 8th row. A modal confirmation dialog is centered over the table. It features a large orange exclamation mark icon, the question 'Are you sure?', and a message below stating 'You want to delete this return action!'. There are two buttons at the bottom: 'Cancel' (gray) and 'Yes, delete it!' (red). The background table and sidebar are partially visible through the dialog.

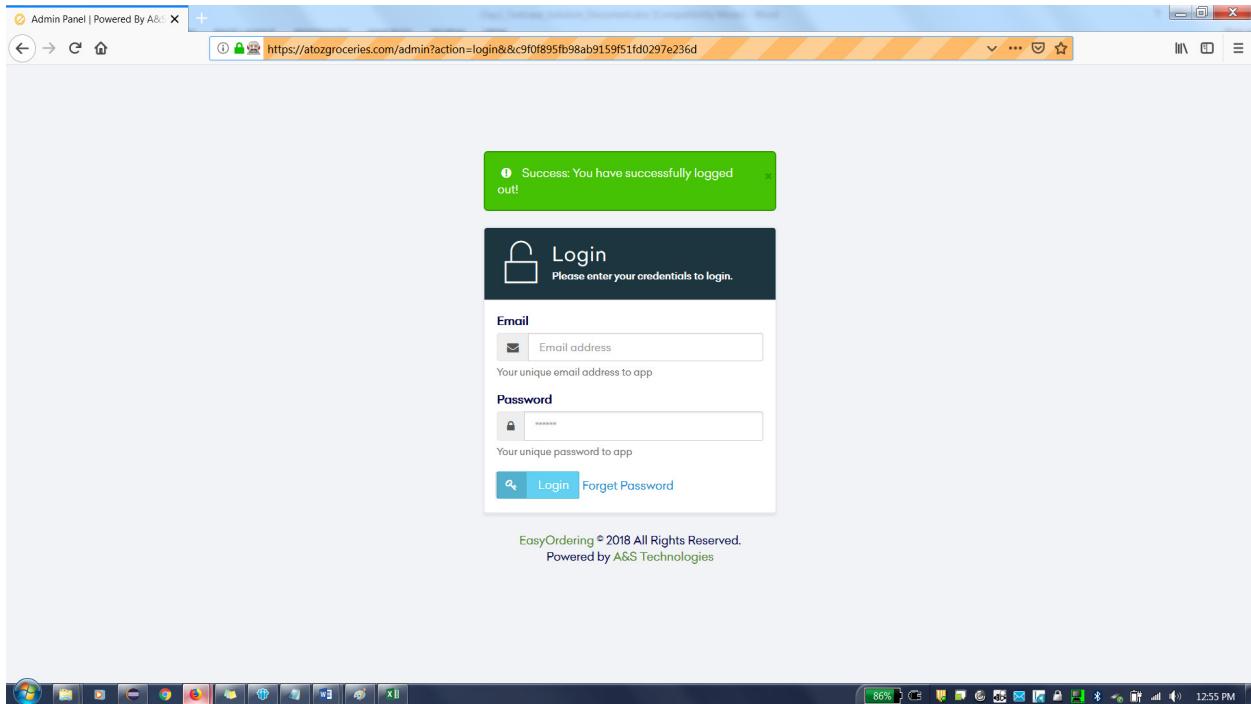
11. Clicking on "Yes, delete it"

The screenshot shows the 'Return Reason' tab in the Admin Panel. A modal dialog box is centered on the screen with the title 'Deleted!' and the message 'Your banner file has been deleted.' An 'OK' button is at the bottom of the dialog. In the background, a table lists 12 return reasons with their names and action buttons. The table has columns for '#', 'Return Reason Name', and 'Action'. The names include 'Received Wrong Item', 'Test', 'testing', 'Not the same order', 'damaged', 'small in size', 'defect', 'Large', 'not satisfied', and 'placed wrong item'. At the bottom right of the table, there are 'Previous' and 'Next' buttons.

## 12. Getting successful message after deleting.

The screenshot shows the 'Return Reasons' page in the Admin Panel. A red success message banner at the top says 'You have successfully deleted data!'. Below it is a table titled 'Return Reason List' with columns for '#', 'Return Reason Name', and 'Action'. The table contains 7 entries: 'Order Error', 'Other, please supply details', 'Received Wrong Item', 'Test', 'testing', 'Not the same order', and 'small in size'. The 'Action' column for each row contains a blue 'Action' button. The left sidebar shows a navigation menu with categories like Dashboard, Catalog, Sales, Customers, Marketing, System, and Reports. The 'Reports' category is currently selected. The status bar at the bottom shows 'Waiting for atozgroceries.com...' and the system time as 1:56 PM.

## 13. Logging off from the webpage



## 14. Output from console

EclipseWorkSpace - SeleniumFramework/testng.xml - Eclipse IDE

```

<?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="Test">
<parameter name="browser" value="ff"/>
<classes>
<class name="com.ibm.test.BaseTest"/>
</classes>
</test> <!-- Test -->
</suite> <!-- Suite -->

```

Dec 18, 2018 1:58:30 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/returnreason/tabs?token=1679891ca5a6fa6fb6e08eb1b2dc, line 689: TypeError: document.getElementById(...)  
JavaScript error: javascript:script(), line 1: ReferenceError: script is not defined  
JavaScript error: https://atozgroceries.com/admin/returnreason/tabs?token=a5c48cce2ed7fbde1afcfc1c7c6ad26, line 671: TypeError: document.getElementById(...)  
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null  
1545121795372 Marionette INFO Stopped listening on port 61963  
[Child 10412, Chrome\_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/chromium/src/common/ipc\_channel\_win.cc, line 346  
[Child 10412, Chrome\_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/chromium/src/common/ipc\_channel\_win.cc, line 346  
[Child 13612, Chrome\_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/chromium/src/common/ipc\_channel\_win.cc, line 346  
[Child 13612, Chrome\_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/chromium/src/common/ipc\_channel\_win.cc, line 346  
[Parent 12672, Gecko\_IOThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/chromium/src/common/ipc\_channel\_win.cc, line 346  
[Child 13676, Chrome\_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/chromium/src/common/ipc\_channel\_win.cc, line 346  
[Child 13676, Chrome\_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/ipc/chromium/src/common/ipc\_channel\_win.cc, line 346  
#####
[Child][MessageChannel::SendAndWait] Error: Channel error: cannot send/recv

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

**\*\*\* End of the document \*\*\***