

Day 3 - Test case solution document

NOTE: Change in test case



Re: Day 3 Test Case 📧
■ **Doniparthi Harisha** to: manjunath, ACADGILD
Cc: sweetaga, balajidinakaran1

Hi Manjunath,

The functionality of the below testcase is not working. There is no delete option in the actions of mail. The application needs to be corrected. Kindly let me know once that action is available, so that I can work on the below testcase.

For Day 3 - I am working on the deleting Coupon , as discussed in the session.

Thanks & Regards
Harisha D

▼ Notifications Acadgild	Hi, PFA day 3 test case:
From: Notifications Acadgild <notifications@acadgild.com> To: dharisha@in.ibm.com Date: 12/13/2018 06:52 PM Subject: Day 3 Test Case	

Hi,

PFA day 3 test case:

Process	Steps To Execute	Expected Results
Delete the mail	1. Login to admin portal	Validate the deleted mail details
	2. Delete anyone mail details	

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="DeleteCoupon")
    public void Day3TestCase() throws InterruptedException, IOException
    {
        LoginPage login=new LoginPage(driver,wait);
        DashboardPage dash=new DashboardPage(driver,wait);
        CouponPage coupon=new CouponPage(driver,wait);
        try
```

```

    {
        String url=hashData.get("url");
        String username=hashData.get("username");
        String password=hashData.get("password");
        String couponcode=hashData.get("couponcode");
        String expdeletemsg=hashData.get("expdeletemsg");
        //navigating to the Admin URL of atozgroceries
        driver.navigate().to(url);

        //login page is used to provide the input of username and password
        login.EnterEmail(username);
        login.EnterPassword(password);
        login.clickLogin();

        //Opening Marketing in dashboard
        dash.OpenMarketing();

        //Opening coupons in Marketing
        coupon.ClickCoupon();

        //Deleting coupon
        coupon.DeleteCoupon(couponcode);

        //getting the pagesource for validating
        String deleteSuccessMsg=coupon.pageSource();
        //Assertion on deleted coupon
        Assert.assertTrue(deleteSuccessMsg.contains(expdeletemsg),"Assertion on Deleted
Coupon!!");
    }
    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/Day3_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();
}
}

```

CouponPage.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.WebDriverWait;

public class CouponPage
{
    //Finding elements in Coupon page
    @FindBy(xpath="/html/body/div[1]/div[1]/div/ul/li[5]/ul/li[1]/a")
    WebElement couponEle;
    WebDriver driver;
    WebDriverWait wait;
    JavascriptExecutor js;
    //constructor
    public CouponPage(WebDriver driver,WebDriverWait wait)
    {
        this.driver=driver;
        this.wait=wait;
    }
}

```

```

        js=(JavascriptExecutor) driver;
        PageFactory.initElements(driver, this);
    }
    //opening Coupon under Marketing
    public void ClickCoupon()
    {
        couponEle.click();
    }
    //Method to delete coupon
    public void DeleteCoupon(String code) throws InterruptedException
    {
        //getting rows
        List<WebElement>
rowEle=driver.findElements(By.xpath("//table[@id='dataTableExample2']/tbody/tr"));
        //finding no of rows
        int rows=rowEle.size();
        for(int i=1;i<=rows;i++)
        {
            //getting the code of the coupon
            WebElement
codeEle=driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/td[3]"));
            String codeTxt=codeEle.getText();
            //getting the code row for deleting
            if(codeTxt.equals(code))
            {
                //clicking on Action button of deleting row

                driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/td[8]/descendant::button")).click();
                //clicking on Delete button to delete the row

                driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/td[8]/descendant::ul/li[2]")).click();
                // added thread.sleep for waiting till the
confirmation screen appears
                Thread.sleep(5000);
                //clicking on the delete confirmation

                driver.findElement(By.xpath("//button[@class='confirm']")).click();
                break;
            }
        }
    }
    public String pageSource() throws InterruptedException
    {
        //waiting till the Delete success message is displayed to get the
pagesource

        wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//div[contains(@class,'alert-danger')]")));
        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>

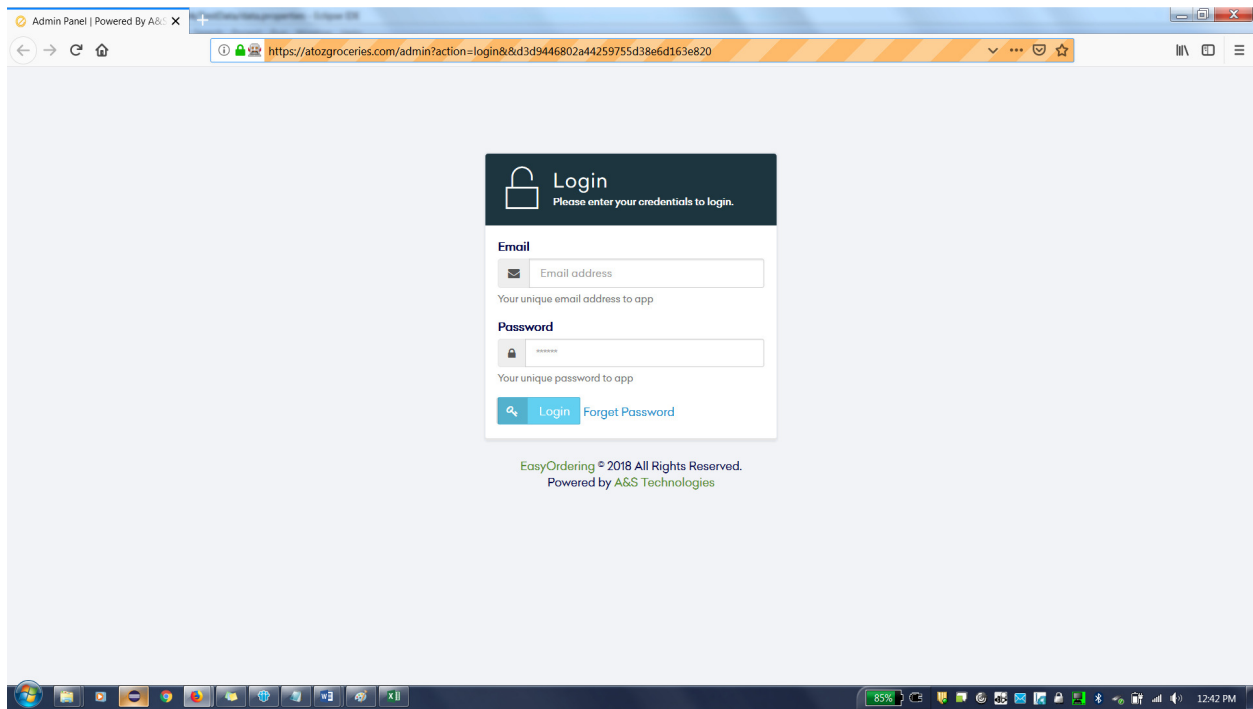
username=demo@atozgroceries.com
password=456789
categoryname=Milk Products
tagtitle=Milk Products
sortorder=12
status=Enabled
show=All
tagname=Milk
tagsort=10
tagstatus=Enabled
tagrequiredmsg=fill out this field
tagSelectReqMsg=select an item in the list
couponcode=150FF
expdeletemsg=successfully deleted

testing.xml

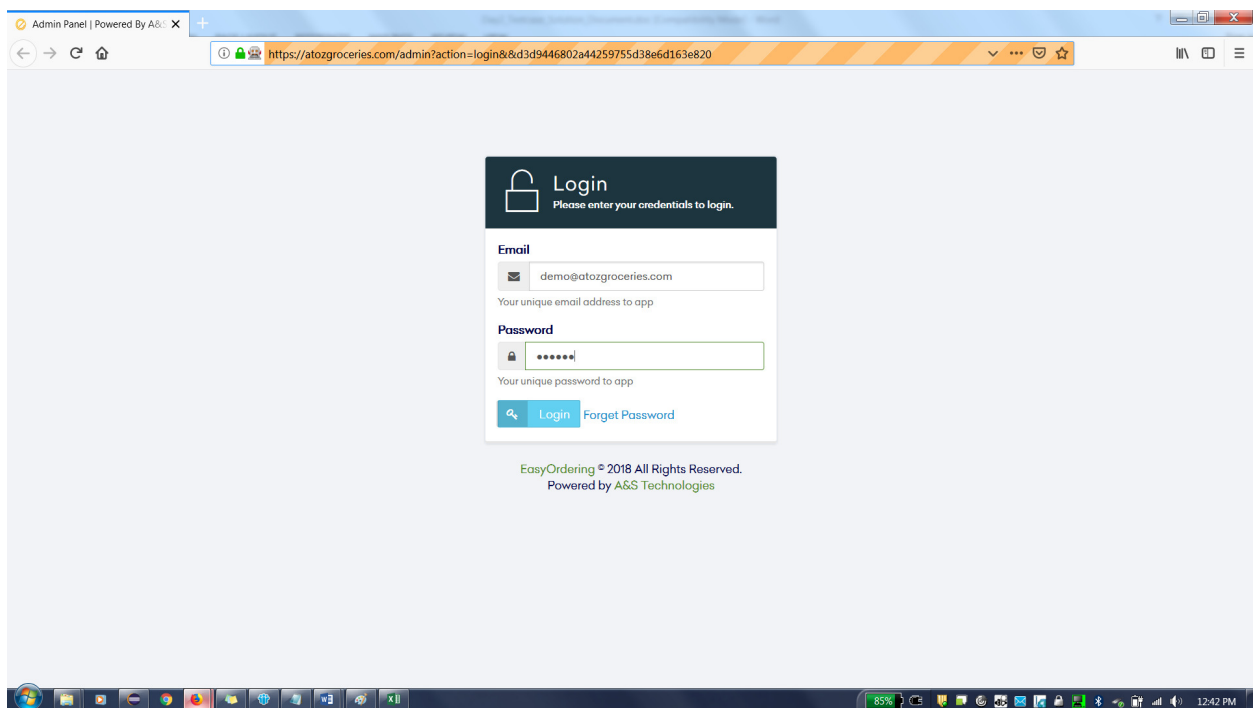
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
    <parameter name="browser" value="ff"></parameter>
    <classes>
      <class name="com.ibm.test.BaseTest"/>
    </classes>
  </test> <!--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

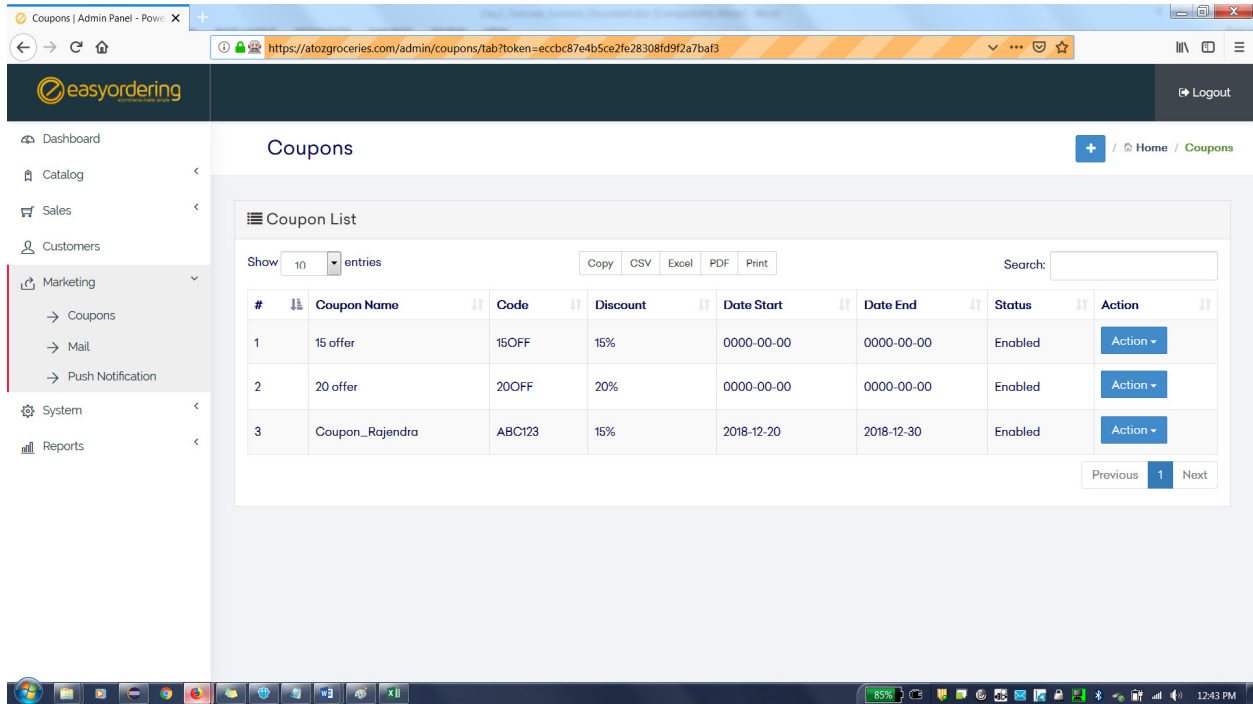
The screenshot shows the Easyordering Admin Dashboard. The left sidebar contains a menu with the following items: Dashboard, Catalog, Sales, Customers, Marketing, System, and Reports. The main content area is titled "Dashboard" and features four summary cards: Total Orders (0), Total Sales (₹698.25), Total Customers (0), and Total Products (106). Below these cards is a section titled "Latest Orders" containing a table with one order.

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

4. Opening Marketing

The screenshot shows the Easyordering Admin Dashboard with the "Marketing" menu item expanded in the left sidebar. The expanded menu shows three sub-items: Coupons, Mail, and Push Notification. The main content area remains the same as in the previous screenshot, showing the Dashboard summary cards and the Latest Orders table.

5. Opening Coupons

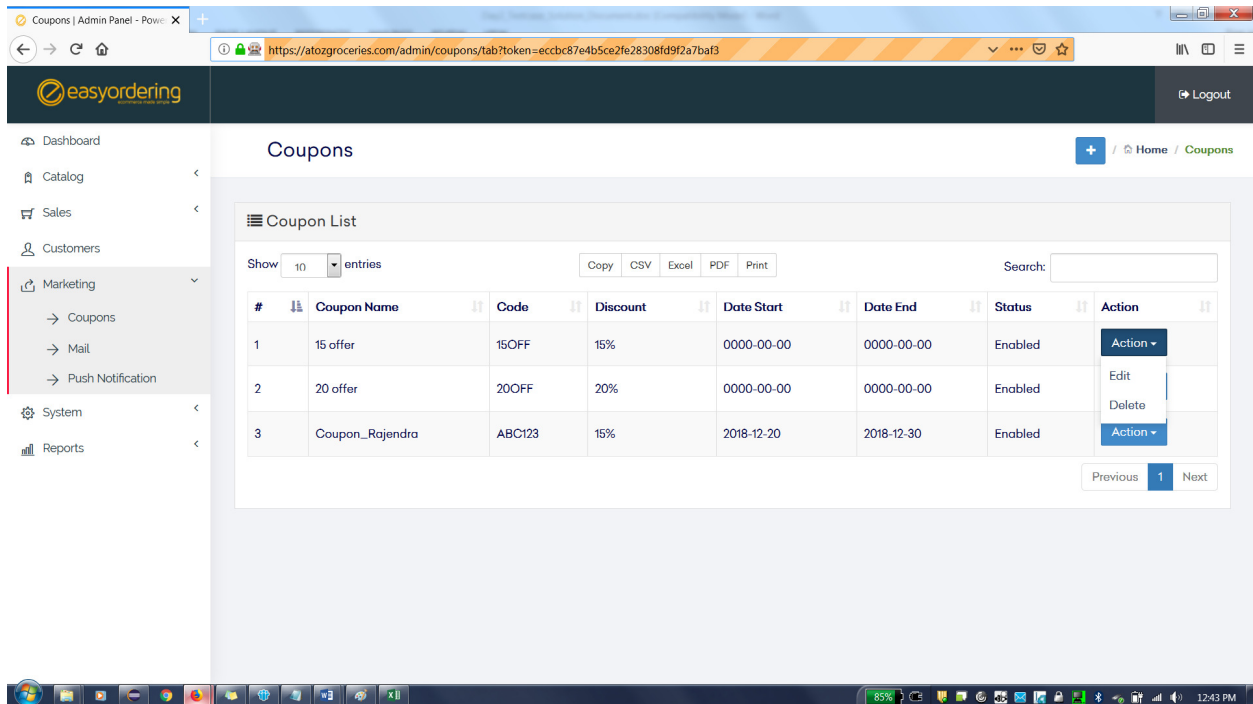


The screenshot shows the 'Coupons' page in the Easyordering Admin Panel. The page title is 'Coupons'. Below the title, there is a 'Coupon List' section. It includes a 'Show' dropdown set to '10' and 'entries', and buttons for 'Copy', 'CSV', 'Excel', 'PDF', and 'Print'. A search bar is also present. The table below lists the coupons:

#	Coupon Name	Code	Discount	Date Start	Date End	Status	Action
1	15 offer	15OFF	15%	0000-00-00	0000-00-00	Enabled	Action
2	20 offer	20OFF	20%	0000-00-00	0000-00-00	Enabled	Action
3	Coupon_Rajendra	ABC123	15%	2018-12-20	2018-12-30	Enabled	Action

At the bottom of the table, there are 'Previous', '1', and 'Next' navigation links.

6. Clicking on the action of the coupon we want to delete.

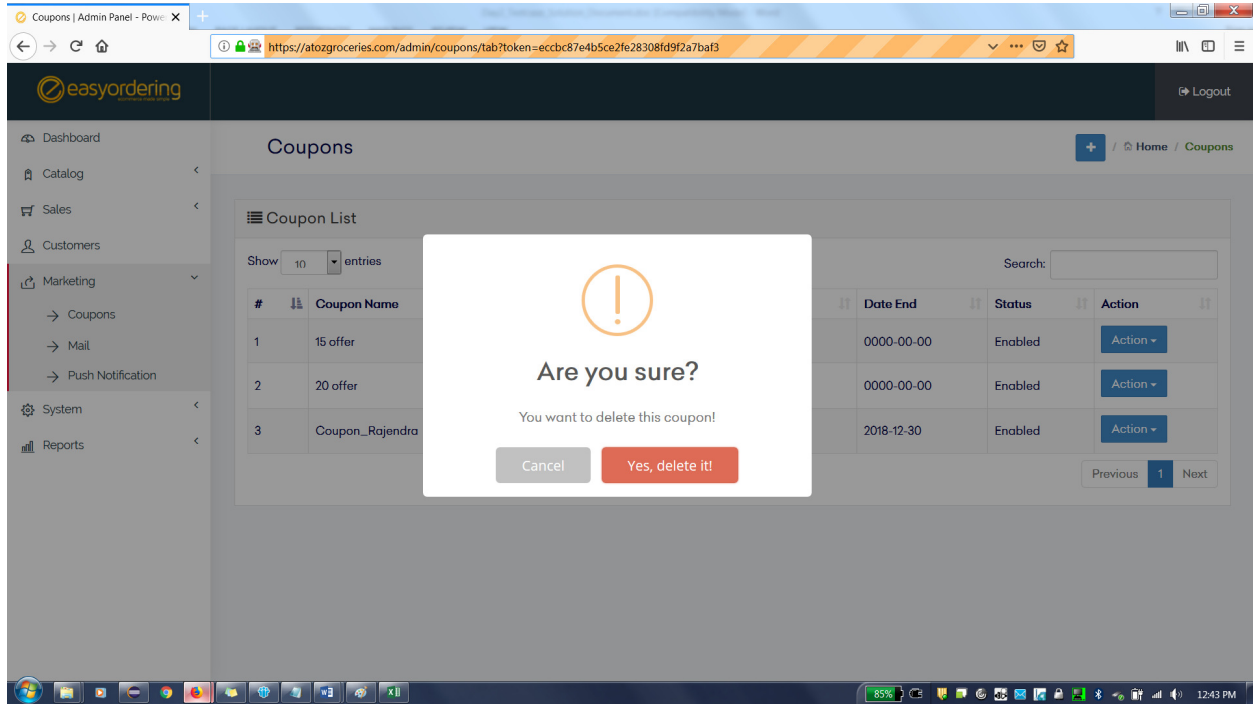


The screenshot shows the 'Coupons' page in the Easyordering Admin Panel. The page title is 'Coupons'. Below the title, there is a 'Coupon List' section. It includes a 'Show' dropdown set to '10' and 'entries', and buttons for 'Copy', 'CSV', 'Excel', 'PDF', and 'Print'. A search bar is also present. The table below lists the coupons:

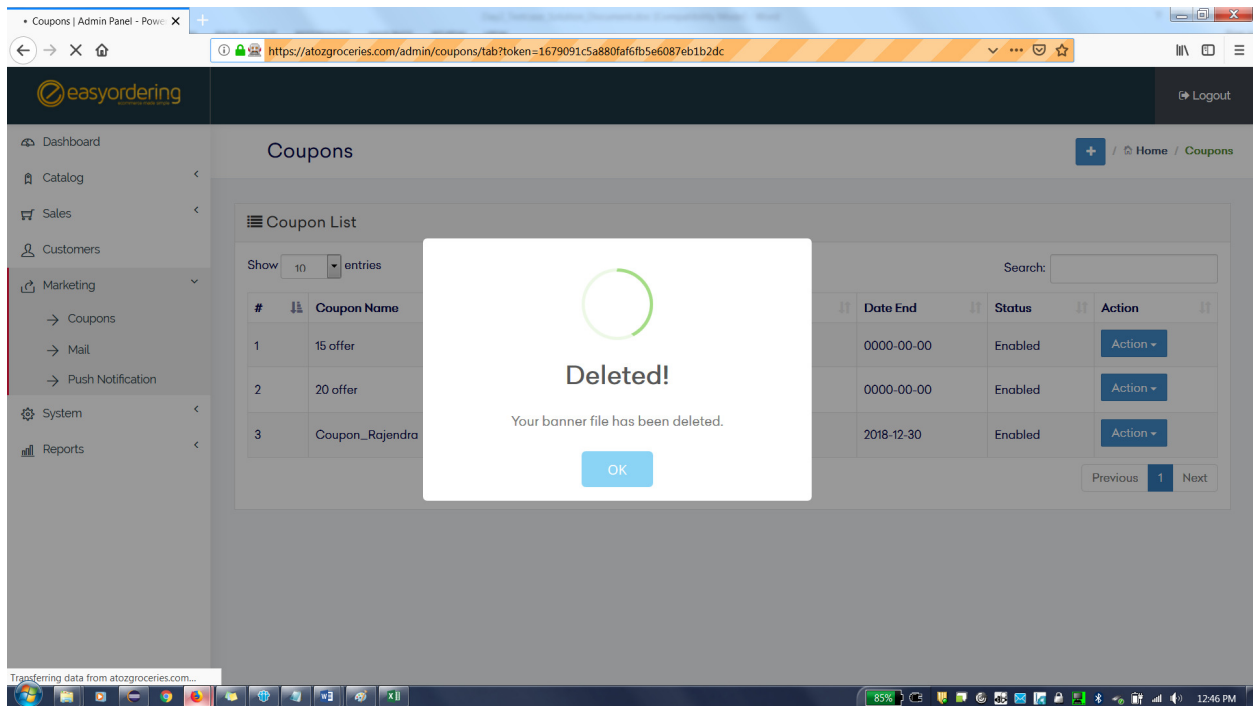
#	Coupon Name	Code	Discount	Date Start	Date End	Status	Action
1	15 offer	15OFF	15%	0000-00-00	0000-00-00	Enabled	Action
2	20 offer	20OFF	20%	0000-00-00	0000-00-00	Enabled	Edit
3	Coupon_Rajendra	ABC123	15%	2018-12-20	2018-12-30	Enabled	Delete

At the bottom of the table, there are 'Previous', '1', and 'Next' navigation links.

7. Clicking on delete, we will get confirmation



8. Clicking on "Yes, delete it"



9. Getting successful message after deleting.

The screenshot shows the EasyOrdering Admin Panel. The left sidebar contains navigation links: Dashboard, Catalog, Sales, Customers, Marketing (selected), System, and Reports. The Marketing section is expanded, showing sub-links for Coupons, Mail, and Push Notification. The main content area is titled 'Coupons' and displays a success message: 'You have successfully deleted data!'. Below the message is a 'Coupon List' table with columns: #, Coupon Name, Code, Discount, Date Start, Date End, Status, and Action. The table contains two entries. Entry 1 is '20 offer' with code '20OFF', a 20% discount, and dates '0000-00-00'. Entry 2 is 'Coupon_Rajendra' with code 'ABC123', a 15% discount, and dates '2018-12-20' to '2018-12-30'. Both entries have a status of 'Enabled' and an 'Action' button. The table has a search bar and pagination controls showing 'Previous 1 Next'.

#	Coupon Name	Code	Discount	Date Start	Date End	Status	Action
1	20 offer	20OFF	20%	0000-00-00	0000-00-00	Enabled	Action
2	Coupon_Rajendra	ABC123	15%	2018-12-20	2018-12-30	Enabled	Action

10. Logging off from the webpage

The screenshot shows the EasyOrdering Admin Panel after logging out. The left sidebar is visible, and the main content area displays a success message: 'Success: You have successfully logged out!'. Below the message is a 'Login' form with fields for 'Email' and 'Password'. The 'Email' field is labeled 'Email address' and the 'Password' field is labeled 'Your unique password to app'. There are 'Login' and 'Forgot Password' buttons. At the bottom, it says 'EasyOrdering © 2018 All Rights Reserved. Powered by A&S Technologies'.

Success: You have successfully logged out!

Login
Please enter your credentials to login.

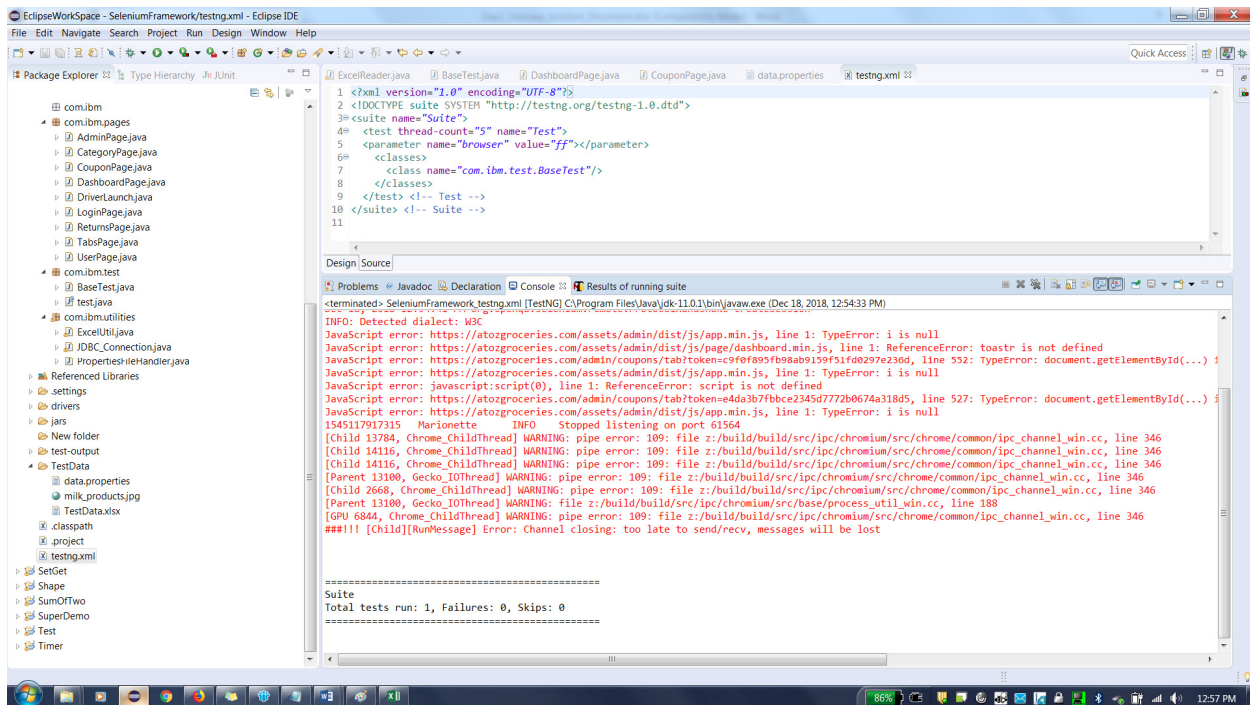
Email
Email address
Your unique email address to app

Password
Your unique password to app

[Login](#) [Forgot Password](#)

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

11. Output from console



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