

Day 8 - 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.Reporter;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Optional;
import org.testng.annotations.Parameters;
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.ProductsPage;
import com.ibm.pages>ReturnsPage;
import com.ibm.pages.SettingsPage;
import com.ibm.pages.ShippingPage;
import com.ibm.pages.TabsPage;
import com.ibm.pages.UserPage;
import com.ibm.utilities.ExcelReader;

public class BaseTest extends DriverLaunch
{
    @Test(testName="EditProductValidation",dataProvider="EditProduct")
        public void Day8TestCase(String old_name,String ProNewName,String
ProMetaTitle,String ProModel,String ProGST,String ProPrice,String
ProSpDisc,String ProSort,String ProStatus,String ProTabs,String
        ProCategories,String ProImgPath) throws InterruptedException

    {
        LoginPage login=new LoginPage(driver,wait);
        DashboardPage dash = new DashboardPage(driver,wait);
        ProductsPage prod=new ProductsPage(driver,wait);
        UserPage user=new UserPage(driver,wait);
        String adminurl=hashData.get("adminurl");
        String username=hashData.get("username");
        String password=hashData.get("password");
        String userurl=hashData.get("userurl");
```

```

//Navigating to Admin page of atozgroceries.com
driver.get(adminurl);
//logging into the admin page
login.EnterEmail(username);
login.EnterPassword(password);
login.clickLogin();

dash.OpenCatalog();

prod.OpenProduct();

prod.EditProduct(old_name,ProNewName,ProMetaTitle,ProModel,ProGST,ProPrice,Pro
SpDisc,ProSort,ProStatus,ProTabs,ProCategories,ProImgPath);
prod.ProductValidation(ProNewName);
login.ClickLogout();
driver.get(userurl);
user.ProductsValidation(ProNewName);
}

@DataProvider(name="EditProduct")
public Object[][] Product() throws IOException
{
    return ExcelReader.DataTable("./TestData/TestData.xlsx", "Products");
}
}

```

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[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;

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

    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();
}
public void OpenShipping()
{
    shipEle.click();
}
}

```

ProductsPage.java

```

package com.ibm.pages;

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;
import org.testng.Assert;
import org.testng.Reporter;

public class ProductsPage {

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

    @FindBy(xpath="//*[contains(text(),'Product List')]")
    WebElement prodListEle;

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

    @FindBy(id="pro_name")
    WebElement proNameEle;
    @FindBy(id="meta_title")
    WebElement metaTitleEle;
    @FindBy(id="toTop")
    WebElement toTopEle;
}

```

```

//@FindBy(xpath="//a[contains(text()='Data')]")
@FindBy(xpath="/html/body/div[1]/div[3]/div/form/div[2]/div/div/div[2]/div/div
/ul/li[2]/a")
WebElement dataEle;
@FindBy(id="model")
WebElement modelEle;
@FindBy(id="gst")
WebElement gstEle;
@FindBy(id="price2")
WebElement priceEle;
@FindBy(id="special_dis")
WebElement spDisEle;
@FindBy(id="price_after_special_dis2")
WebElement SpPriceEle;
@FindBy(id="sort_order")
WebElement sortEle;
@FindBy(id="status")
WebElement statusEle;
//@FindBy(xpath="//a[contains(text()='Link')]")
@FindBy(xpath="/html/body/div[1]/div[3]/div/form/div[2]/div/div/div[2]/div/div
/ul/li[3]/a")
WebElement linkEle;
@FindBy(id="tabs")
WebElement tabsEle;
@FindBy(id="categories")
WebElement categoryEle;
//@FindBy(xpath="//a[contains(text()='Image')]")
@FindBy(xpath="/html/body/div[1]/div[3]/div/form/div[2]/div/div/div[2]/div/div
/ul/li[4]/a")
WebElement imgEle;
@FindBy(xpath="//input[@id='pro_image']")
WebElement proImgEle;

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

WebDriver driver;
WebDriverWait wait;
JavascriptExecutor js;

public ProductsPage(WebDriver driver, WebDriverWait wait)
{
    this.driver=driver;
    this.wait=wait;
    PageFactory.initElements(driver, this);
    js=(JavascriptExecutor) driver;
}
public void OpenProduct()
{
    prodEle.click();
}

public void EditProduct(String old_name,String new_name,String
meta_Title,String model,String gst,String price,String SplDiscount,String sort,String
Status,String tabs,String categories,String imgPath) throws InterruptedException

```

```

{
    wait.until(ExpectedConditions.visibilityOf(prodListEle));
    /*Select showSel=new Select(showEle);
    showSel.selectByVisibleText("25");*/
    wait.until(ExpectedConditions.visibilityOf(prodListEle));
    int
rowLen=driver.findElements(By.xpath("//table[@id='dataTableExample2']/tbody/tr")).size();
    for(int i=1;i<=rowLen;i++)
    {
        String
prod_name=driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/
td[2]").getText().trim();
        if(prod_name.equals(old_name))
        {
            //clicking on the Action button of the desired category to
edit
            driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/td[9]/descendant::button")).click();
            //clicking on the Edit button of the desired category to
edit
            driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/td[9]/descendant::ul/li)[1]").click();
            //Clears the existing data in product tab
            proNameEle.clear();

            js.executeScript("arguments[0].scrollIntoView(true);",metaTitleEle);
            metaTitleEle.clear();
            toTopEle.click();
            //sending new data in Product Tab
            proNameEle.sendKeys(new_name);

            js.executeScript("arguments[0].scrollIntoView(true);",metaTitleEle);
            metaTitleEle.sendKeys(meta_Title);
            toTopEle.click();
            //opening data tab
            js.executeScript("arguments[0].click()",dataEle);
            //clearing the existing data in Data tab

            js.executeScript("arguments[0].scrollIntoView(true);",modelEle);
            wait.until(ExpectedConditions.visibilityOf(modelEle));
            //Clears the existing data in Data tab
            modelEle.clear();
            gstEle.clear();
            priceEle.clear();
            spDisEle.clear();

            js.executeScript("arguments[0].scrollIntoView(true);",statusEle);
            SpPriceEle.clear();
            sortEle.clear();
            driver.findElement(By.id("toTop")).click();
            //sending new data to Data Tab
}

```

```

        js.executeScript("arguments[0].scrollIntoView(true);",modelEle);
        modelEle.sendKeys(model);
        gstEle.sendKeys(gst);
        priceEle.sendKeys(price);
        spDisEle.sendKeys(SplDiscount);

        js.executeScript("arguments[0].scrollIntoView(true);",statusEle);
        sortEle.sendKeys(sort);
        Select statusSel=new Select(statusEle);
        statusSel.selectByVisibleText(Status);
        driver.findElement(By.id("toTop")).click();
        //opening Link Tab
        js.executeScript("arguments[0].click()",linkEle);
        Thread.sleep(5000);//due to my browser prb, added this
        Select TabsSel=new Select(tabsEle);
        TabsSel.selectByVisibleText(tabs);
        Select CatSel=new Select(categoryEle);
        CatSel.selectByVisibleText("");
        CatSel.selectByVisibleText(categories);
        //opening Image tab
        js.executeScript("arguments[0].click()",imgEle);
        //wait.until(ExpectedConditions.visibilityOf(proImgEle));
        Thread.sleep(5000);//due to my browser prb, added this
        proImgEle.sendKeys(imgPath);
        //clicking on Save to save the product
        saveEle.click();
        break;
    }
}
}

public void ProductValidation(String pro_newname)
{
    wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//div[contains(text(),'successfully updated product')]")));
    /*Select showSel=new Select(showEle); showSel.selectByVisibleText("25");*/
    String actualMsg=driver.getPageSource();
    Assert.assertTrue(actualMsg.contains(pro_newname), "Assertion on Edited
Product in Admin Page!!!");
    Reporter.log("Assertion on Edited Product in Admin Page!!!");
}
}

```

UserPage.java

```

package com.ibm.pages;

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;
import org.testng.Assert;
import org.testng.Reporter;

public class UserPage
{
    //Finding the elements of the User Page
    @FindBy(xpath="(/span[@class='click-categories flaticon-bars'])[2]")
    WebElement catBarEle;
    @FindBy(xpath="//*[@class='category-drop-list-inner'][2]")
    WebElement CateList;
    @FindBy(xpath="/html/body/div[1]/div[5]/div/div[1]/div/nav/ul/li[2]/span")
    WebElement HdrEle;
    @FindBy(xpath="//input[@id='search-box']")
    WebElement mainSrchEle;

    WebDriver driver;
    WebDriverWait wait;
    JavascriptExecutor js;
    //Constructor
    public UserPage(WebDriver driver,WebDriverWait wait)
    {
        this.driver=driver;
        this.wait=wait;
        js=(JavascriptExecutor) driver;
        PageFactory.initElements(driver, this);
    }
    //this method validates the edited category presence in User Page under
    "Shop By Category"
    public void CatergoryValidation(String catename)
    {
        catBarEle.click();
        wait.until(ExpectedConditions.visibilityOf(CateList));
        WebElement newcateEle=driver.findElement(By.partialLinkText(catename));
        newcateEle.click();
        String validCate=HdrEle.getText().trim();
        Assert.assertTrue(validCate.contains(catename), "Assertion on Edited
Category in User Page!!!");
        Reporter.log("Assertion on Edited Category in User Page!!!");
    }
    //This method validates the edited Phone number present in the
    footer information in User Page
    public void footerPhnoValidation(String Phno) throws InterruptedException
    {
        WebElement
footPhno=driver.findElement(By.xpath("//p[contains(text(), '"+Phno+"')]"));
        js.executeScript("arguments[0].scrollIntoView(true);", footPhno); String
footerPh=footPhno.getText().trim();
        Assert.assertTrue(footerPh.contains(Phno), "Assertion on Phone number in
footer section!!!");
        Reporter.log("Assertion on Phone number in footer section!!!");
    }
    //This method validates the edited Phone number present in the
    header information in User Page
    public void headerPhnoValidation(String Phno) throws InterruptedException

```

```

{
    WebElement hdrPhno=driver.findElement(By.xpath("//div[@class='header-mid-right-content']")[1]));
    js.executeScript("arguments[0].scrollIntoView(true);", hdrPhno); String
    headerPh=hdrPhno.getText().trim();
    Assert.assertTrue(headerPh.contains(Phno), "Assertion on Phone number in
header section!!");
    Reporter.log("Assertion on Phone number in header section!!");
}
//this method validates the edited Product in User page
public void ProductsValidation(String ProNewName) throws InterruptedException
{
    mainSrchEle.click();
    //sending keys in the search

    driver.findElement(By.xpath("//*[@id=\"searchModal\"]//div/div/div/div[1]/input
")).sendKeys(ProNewName);
    //Thread.sleep(5000);
    //selecting the product from the dropdown

    driver.findElement(By.xpath("//span[text()='"+ProNewName+"']/ancestor::div[@id
='searchproducts-div']")).click();
    String validProduct=HdrEle.getText().trim();
    Assert.assertTrue(validProduct.contains(ProNewName), "Assertion on
Edited Product in User Page!!!");
    Reporter.log("Assertion on Edited Product in User Page!!!");
}
}

```

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

```

adminurl=https://atozgroceries.com/admin
userurl=https://atozgroceries.com
username=demo@atozgroceries.com
password=456789

```

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">
                <methods><include name="Day8TestCase"></include></methods>
            </class>
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->

```

ExcelReader.java

```

package com.ibm.utilities;

import java.io.FileInputStream;
import java.io.IOException;

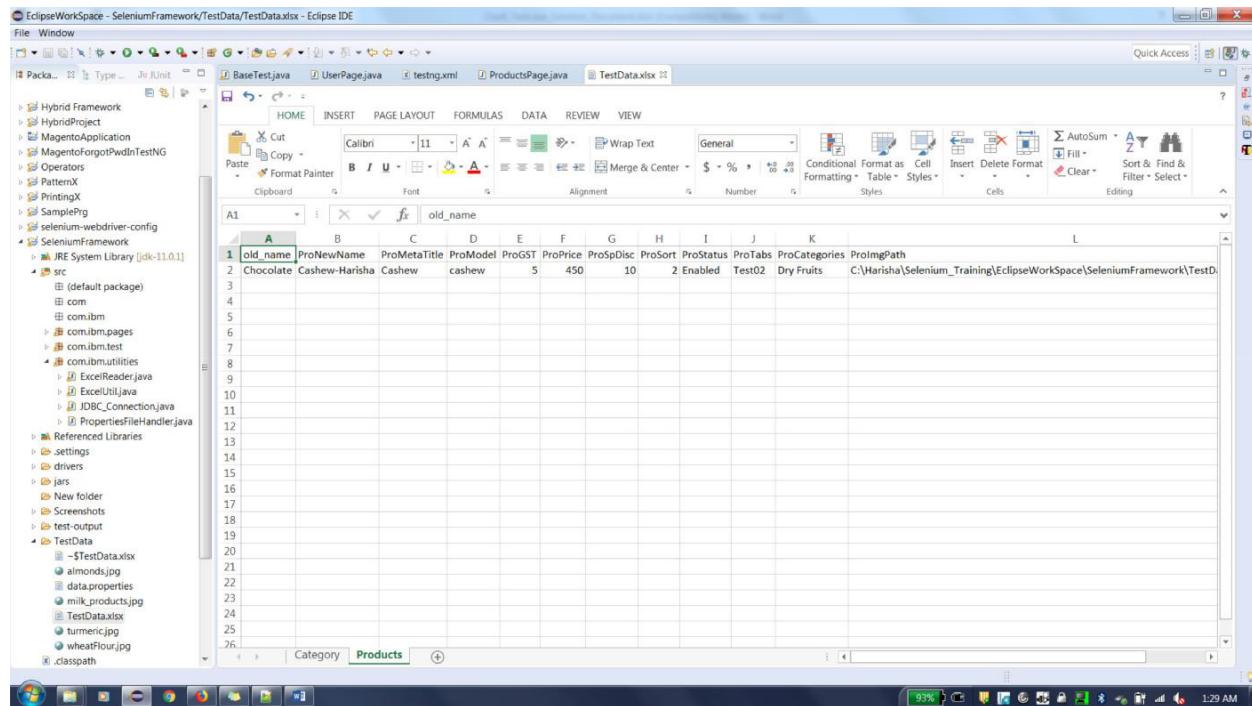
import org.apache.poi.ss.usermodel.DataFormatter;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class ExcelReader {
    public static Object[][] DataTable(String WBLoc, String sheetName)
throws IOException

```

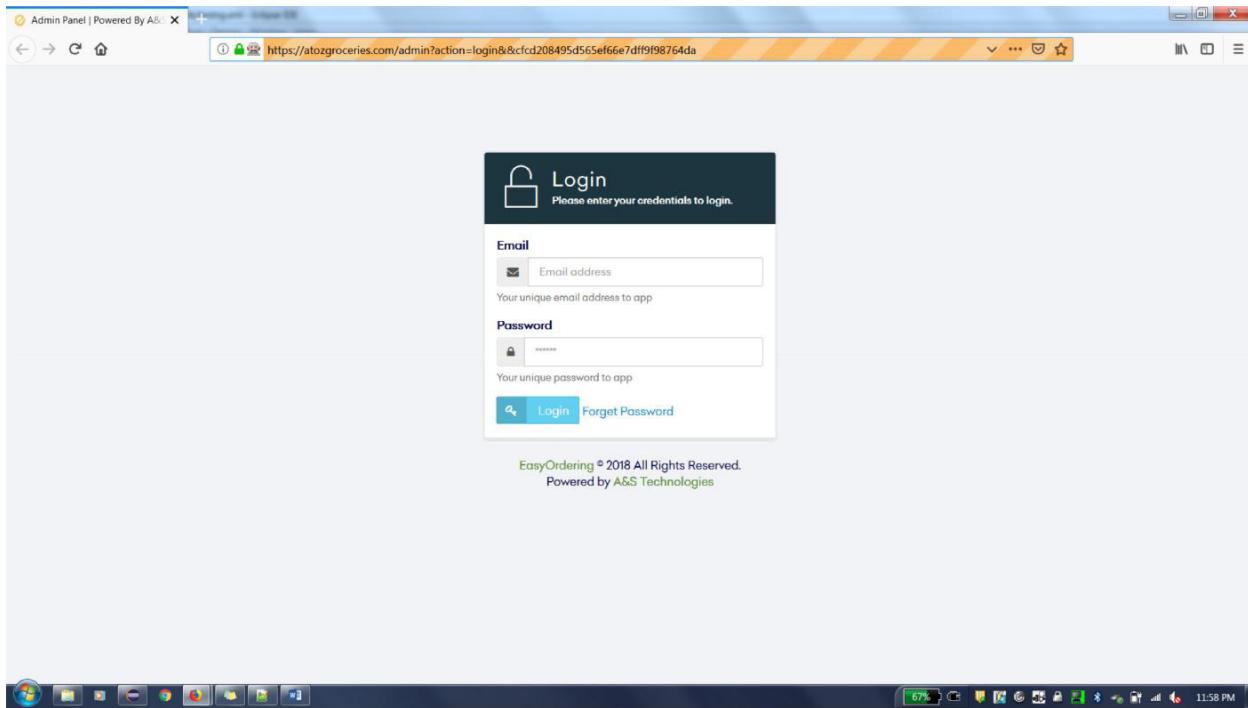
```
XSSFWorkbook TestDataWB=new XSSFWorkbook(new FileInputStream(WBLoc));
XSSFSheet ProductsSheet=TestDataWB.getSheet(sheetName);
Object[][][] ProductInfo=new Object[ProductsSheet.getPhysicalNumberOfRows()-
1][ProductsSheet.getRow(0).getPhysicalNumberOfCells()];
int rowcount=ProductsSheet.getPhysicalNumberOfRows();
int cellcount=ProductsSheet.getRow(0).getPhysicalNumberOfCells();
for(int i=1;i<rowcount;i++)
{
    for(int j=0;j<cellcount;j++)
    {
        DataFormatter format=new DataFormatter();
        String
CellValue=format.formatCellValue(ProductsSheet.getRow(i).getCell(j));
        ProductInfo[i-1][j]=CellValue;
    }
}
return ProductInfo;
}
```

TestData.xlsx

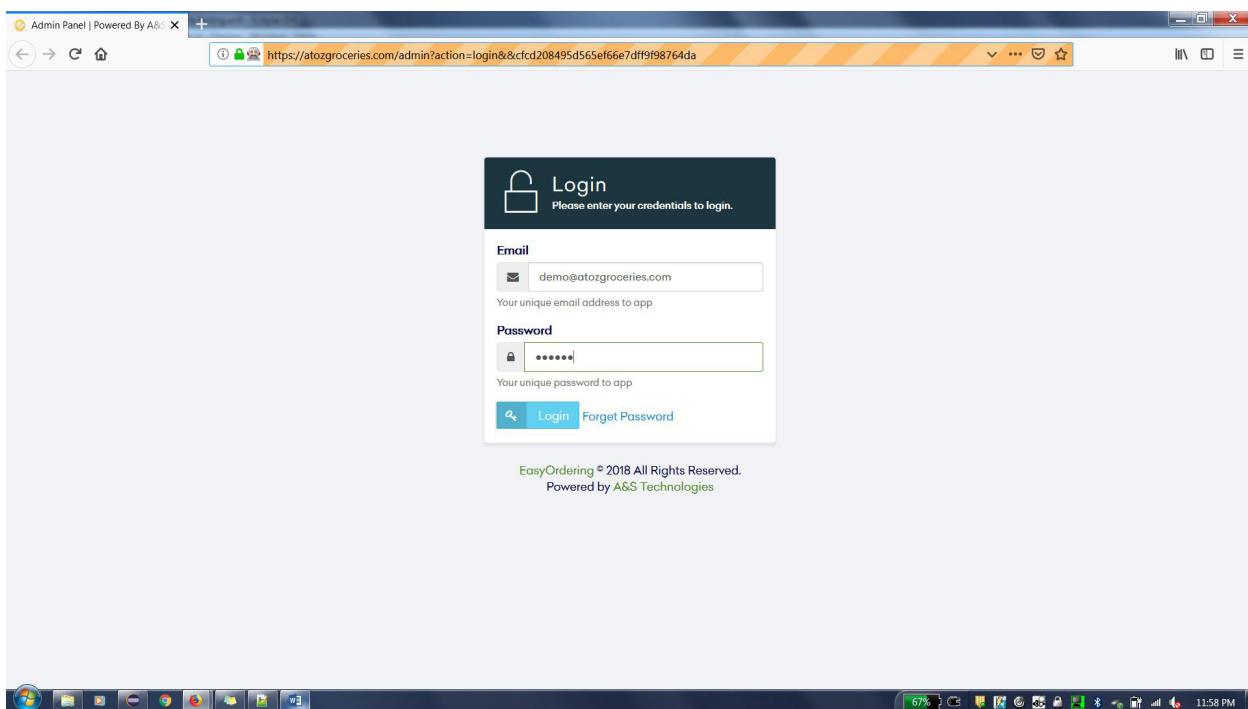


Step by step output:

1. Navigating to the Admin Page of “atozgroceries”



2. Providing login credentials



3. Dashboard after logging into the site

The screenshot shows the Admin Panel Dashboard for 'atozgroceries.com'. The top navigation bar includes links for 'Logout' and 'Home / Dashboard'. The left sidebar menu lists 'Dashboard', 'Catalog', 'Sales', 'Customers', 'Marketing', 'System', and 'Reports'. The main dashboard area features four cards: 'Total Orders' (1), 'Total Sales' (₹5,537.07), 'Total Customers' (3), and 'Total Products' (44). Below these is a section titled 'Latest Orders' with a table:

#	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	

4. Opening Catalog

The screenshot shows the Admin Panel Catalog Dashboard for 'atozgroceries.com'. The top navigation bar includes links for 'Logout' and 'Home / Dashboard'. The left sidebar menu lists 'Dashboard', 'Catalog' (with sub-options: Tabs, Banners, Categories, Products, Product Image, Information), 'Sales', 'Customers', 'Marketing', 'System', and 'Reports'. The main dashboard area features four cards: 'Total Orders' (6), 'Total Sales' (₹27,965.00), 'Total Customers' (19), and 'Total Products' (227). Below these is a section titled 'Latest Orders' with a table:

#	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	

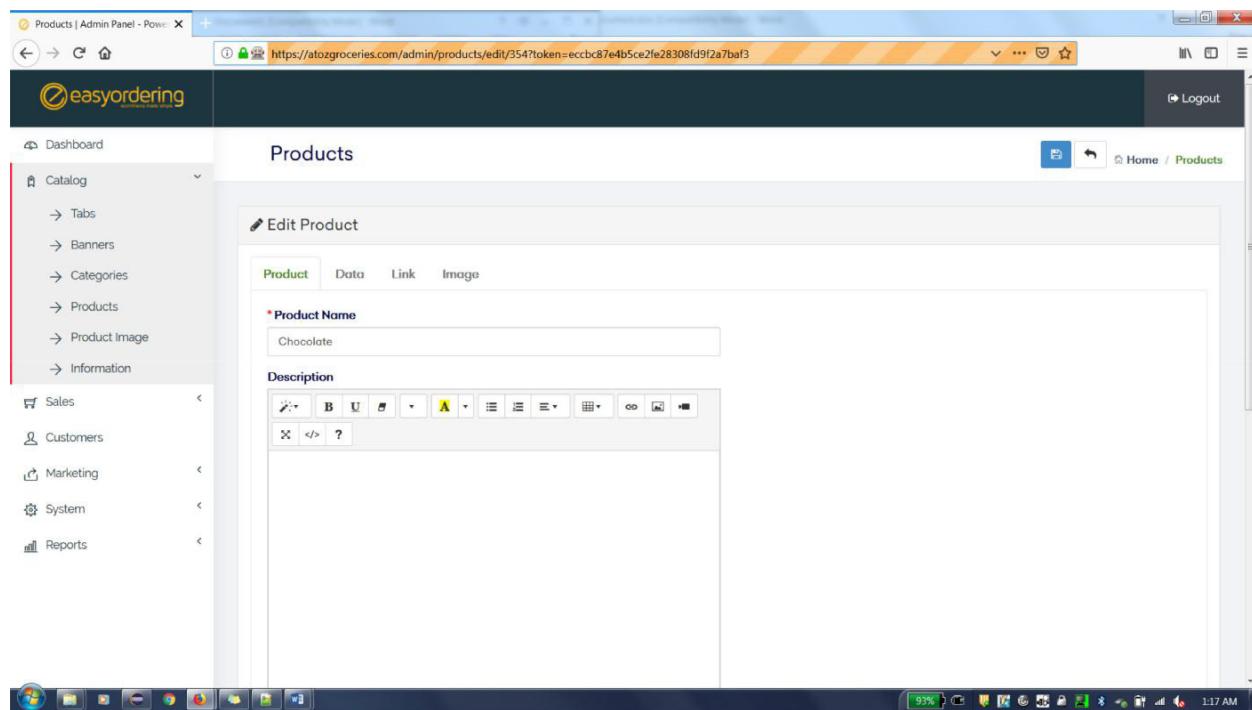
5. Opening Products and clicking on the action of desired product to edit

The screenshot shows the 'Products' page in the Admin Panel. The left sidebar is collapsed, and the main area displays a 'Product List' table. The table has columns for #, Product Name, Image, Model, Price, Total Quantity, Sort Order, Status, and Action. There are 10 entries shown. The second entry, 'Chocolate', has its 'Action' button highlighted.

#	Product Name	Image	Model	Price	Total Quantity	Sort Order	Status	Action
1	New Prod Req		1098	1,890.00 1		1	Enabled	Action
2	Chocolate		Chocolates	203.3 214		2	Disabled	Action
3	Basmathi Rice-harisha		IndiaGate Rice	174.96 194.4	1	1	Enabled	Action
4	NewTest(working on this)		NewTest	425.25 472.5	5	1	Enabled	Action
5	Bisleri Water		Model-W	19.95 21	25	0	Enabled	Action
6	NewBadhamProduct		New	1,320.00 121		1	Enabled	Action
7	DoNotDelete		excellent	1,025.00 1025		1	Enabled	Action

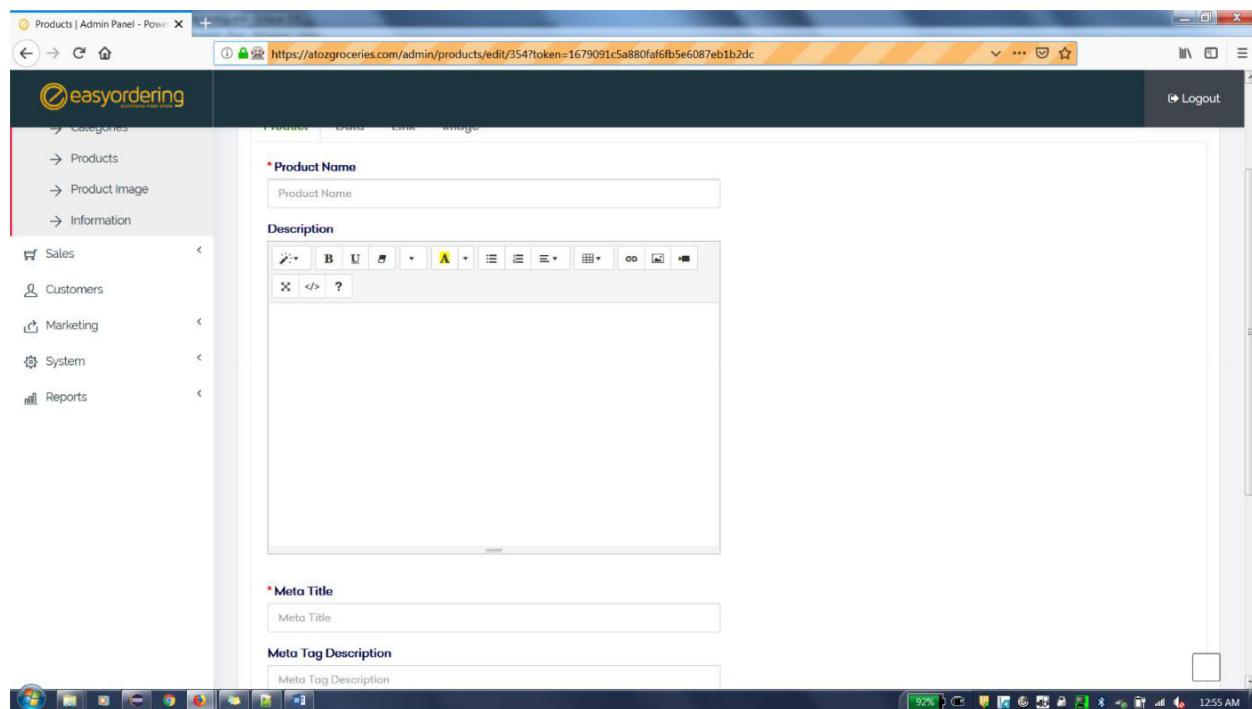
This screenshot shows the same 'Products' page as the previous one, but the second product, 'Chocolate', has its 'Action' button clicked, revealing a dropdown menu. The 'Edit' option is highlighted in the menu.

6. After clicking on Edit



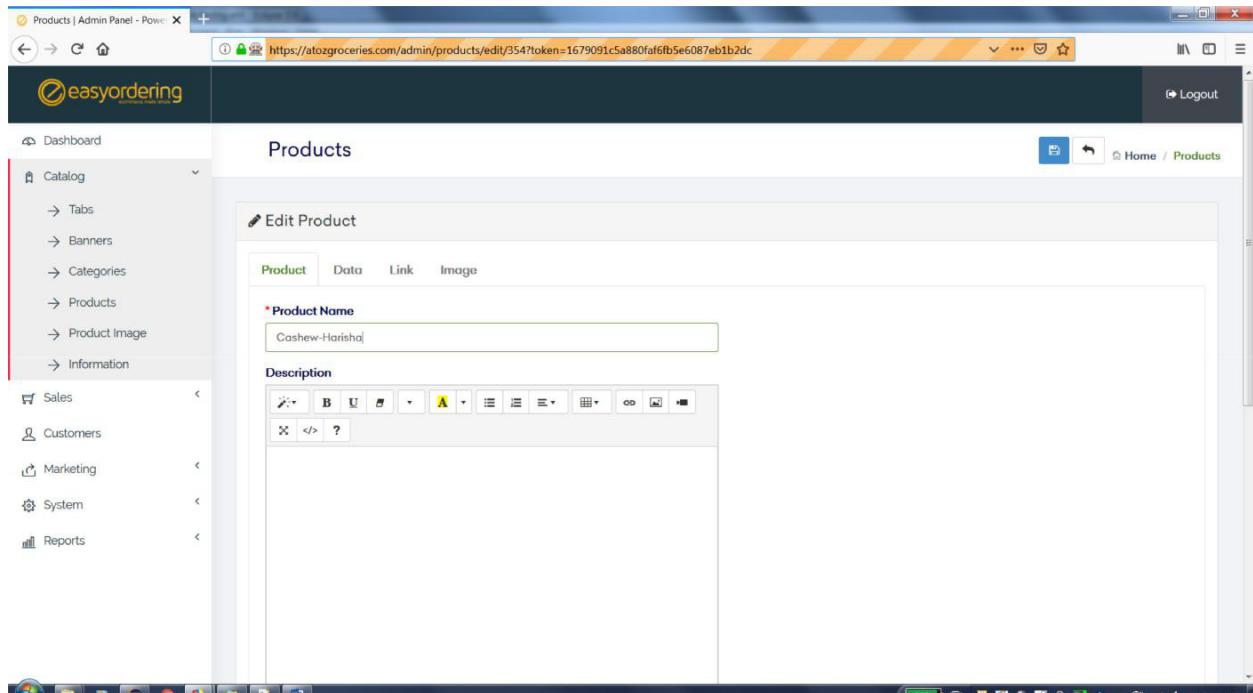
The screenshot shows the 'Products' section of the easyordering Admin Panel. On the left, there's a sidebar with categories like Dashboard, Catalog, Sales, Customers, Marketing, System, and Reports. The 'Catalog' section is expanded, showing sub-options: Tabs, Banners, Categories, Products (which is selected and highlighted in red), Product Image, and Information. The main content area is titled 'Edit Product' and has tabs for Product, Data, Link, and Image. The 'Product' tab is active. It contains a field labeled 'Product Name' with the value 'Chocolate' and a 'Description' field with a rich text editor interface.

7. Clearing the existing values in Product tab

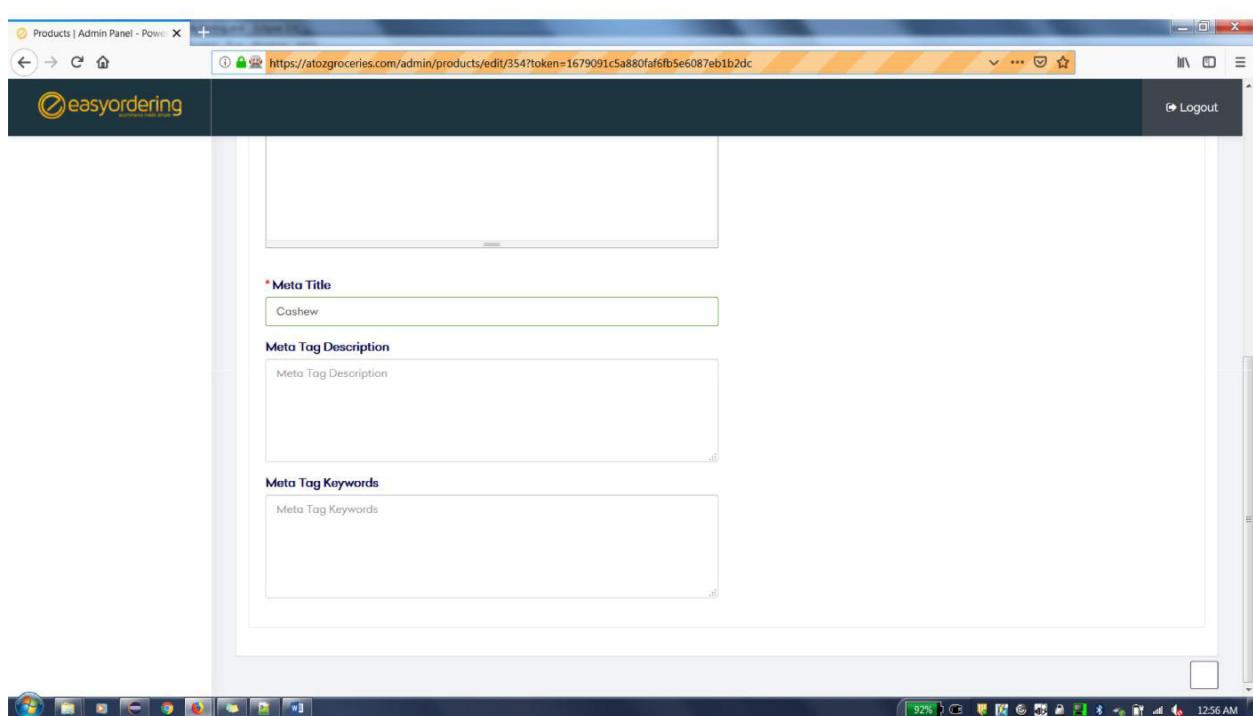


This screenshot shows the same 'Edit Product' page as the previous one, but the 'Product Name' field now contains the placeholder text 'Product Name' instead of 'Chocolate'. The 'Description' field is also empty. The rest of the interface remains the same, with the 'Product' tab selected and the sidebar showing the 'Products' option under 'Catalog'.

8. Sending the new values in product tab

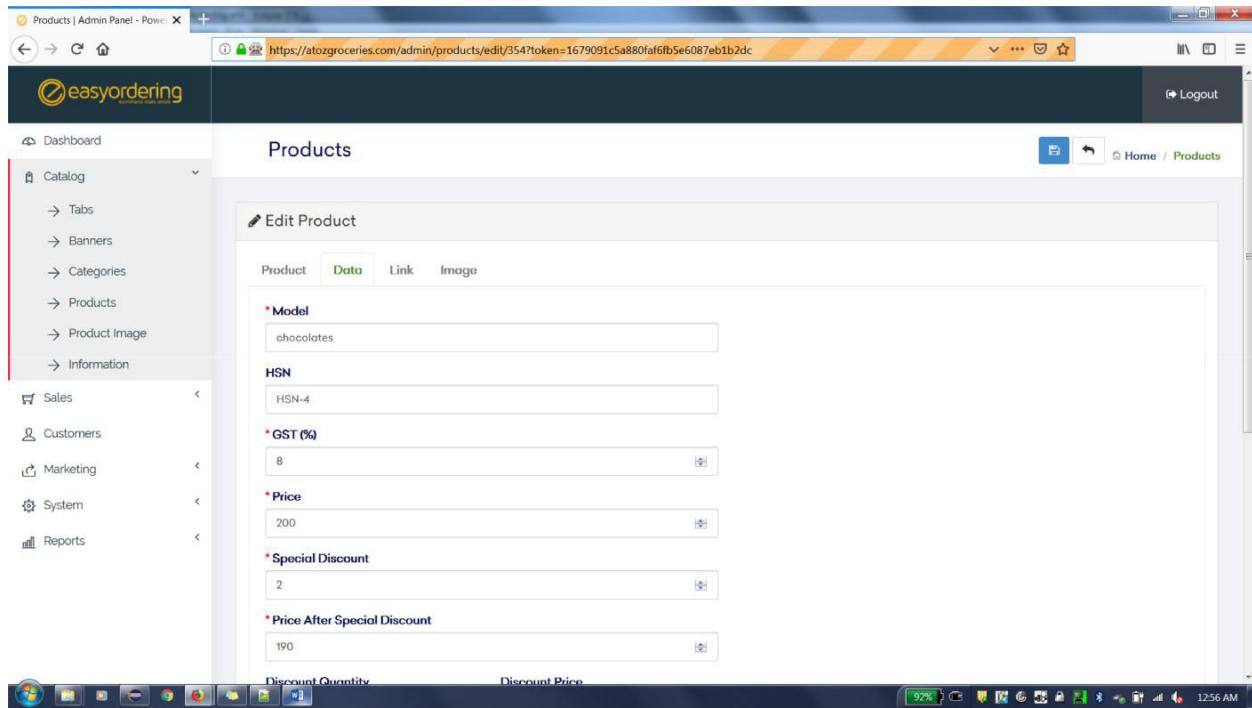


The screenshot shows the 'Edit Product' screen in the easyordering Admin Panel. The left sidebar is collapsed, and the main area displays the 'Edit Product' form. The 'Product' tab is selected, showing a field for 'Product Name' containing 'Casew-Harisha'. Below it is a rich text editor for 'Description'. The URL in the browser is <https://atozgroceries.com/admin/products/edit/354?token=1679091c5a880faf6fb5e6087eb1b2dc>.



The screenshot shows the 'Edit Product' screen with the 'Meta Tag' section expanded. It includes fields for 'Meta Title' (containing 'Casew'), 'Meta Tag Description' (containing 'Meta Tag Description'), and 'Meta Tag Keywords' (containing 'Meta Tag Keywords'). The URL in the browser is the same as the previous screenshot.

9. Opening Data tab

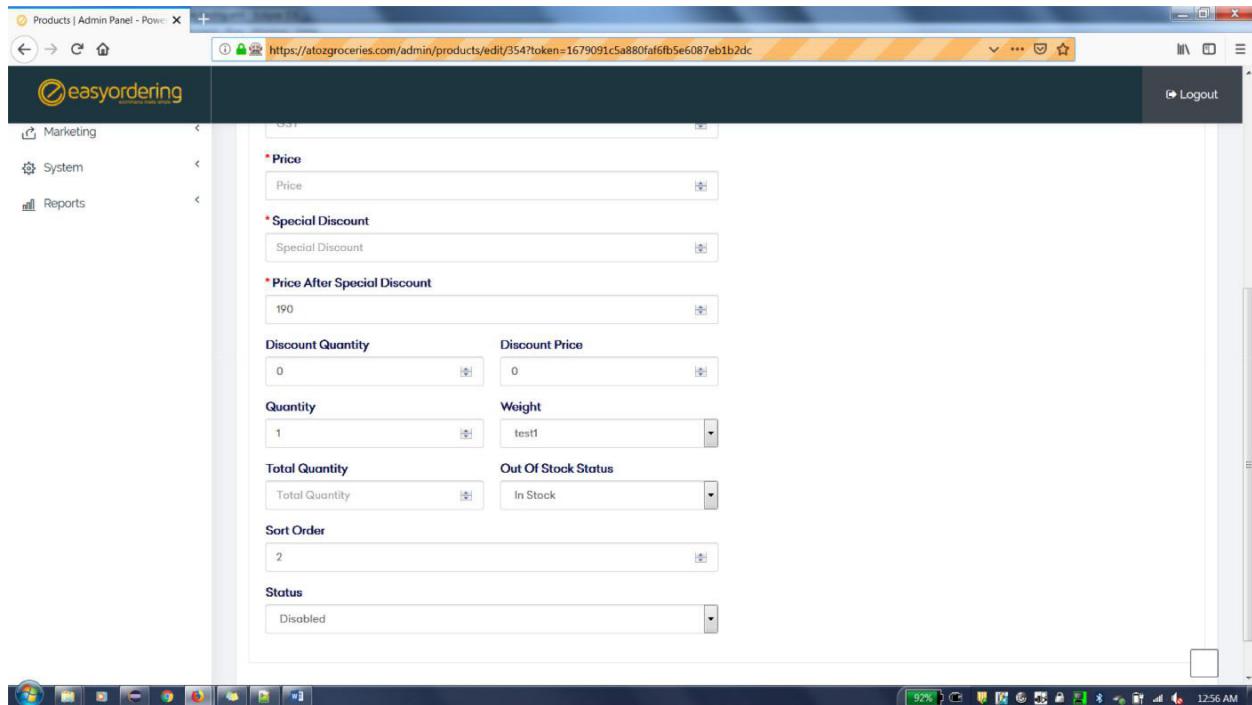


The screenshot shows the 'Edit Product' form in the easyordering Admin Panel. The left sidebar has 'Catalog' expanded, with 'Products' selected. The main area is titled 'Products' and contains the 'Edit Product' form. The 'Data' tab is active. The form fields include:

- *Model:** chocolates
- HSN:** HSN-4
- *GST (%):** 8
- *Price:** 200
- *Special Discount:** 2
- *Price After Special Discount:** 190

Below the form are two buttons: 'Discount Quantity' and 'Discount Price'. The status bar at the bottom shows '92%' and '12:56 AM'.

10. Clearing the existing values in Data tab

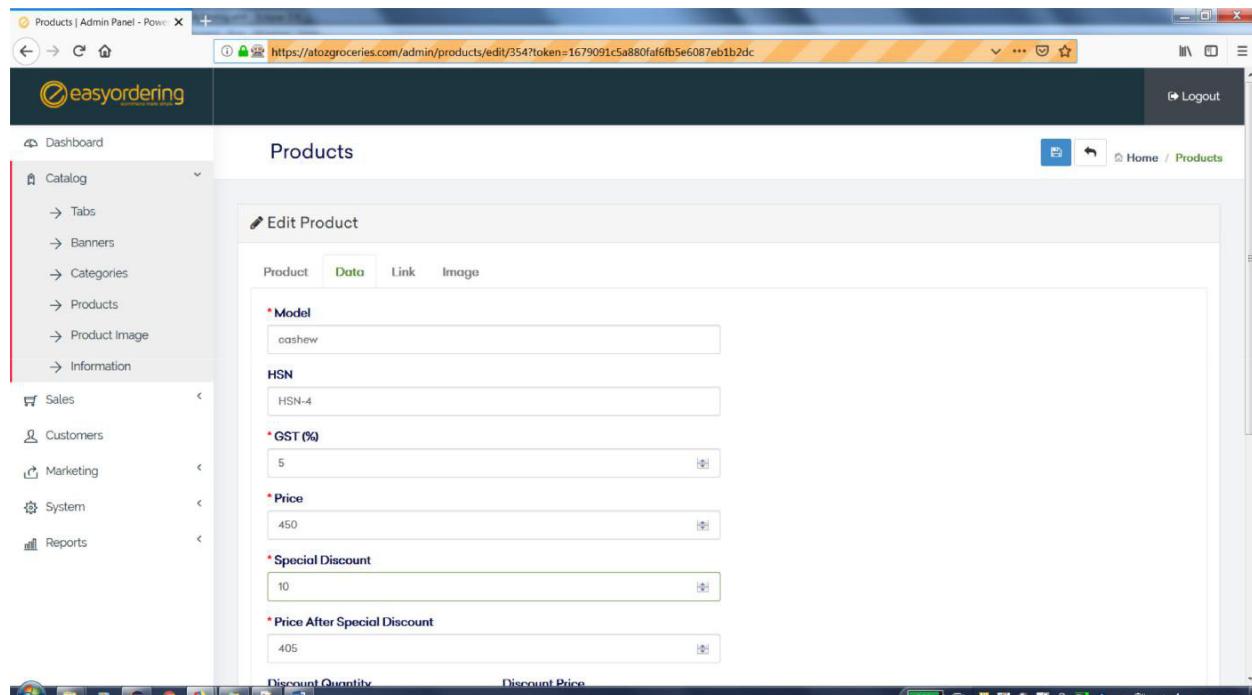


The screenshot shows the same 'Edit Product' form after clearing the values. The fields now contain:

- *Price:** Price
- *Special Discount:** Special Discount
- *Price After Special Discount:** 190
- Discount Quantity:** 0
- Discount Price:** 0
- Quantity:** 1
- Weight:** test1
- Total Quantity:** Total Quantity
- Out Of Stock Status:** In Stock
- Sort Order:** 2
- Status:** Disabled

The status bar at the bottom shows '92%' and '12:56 AM'.

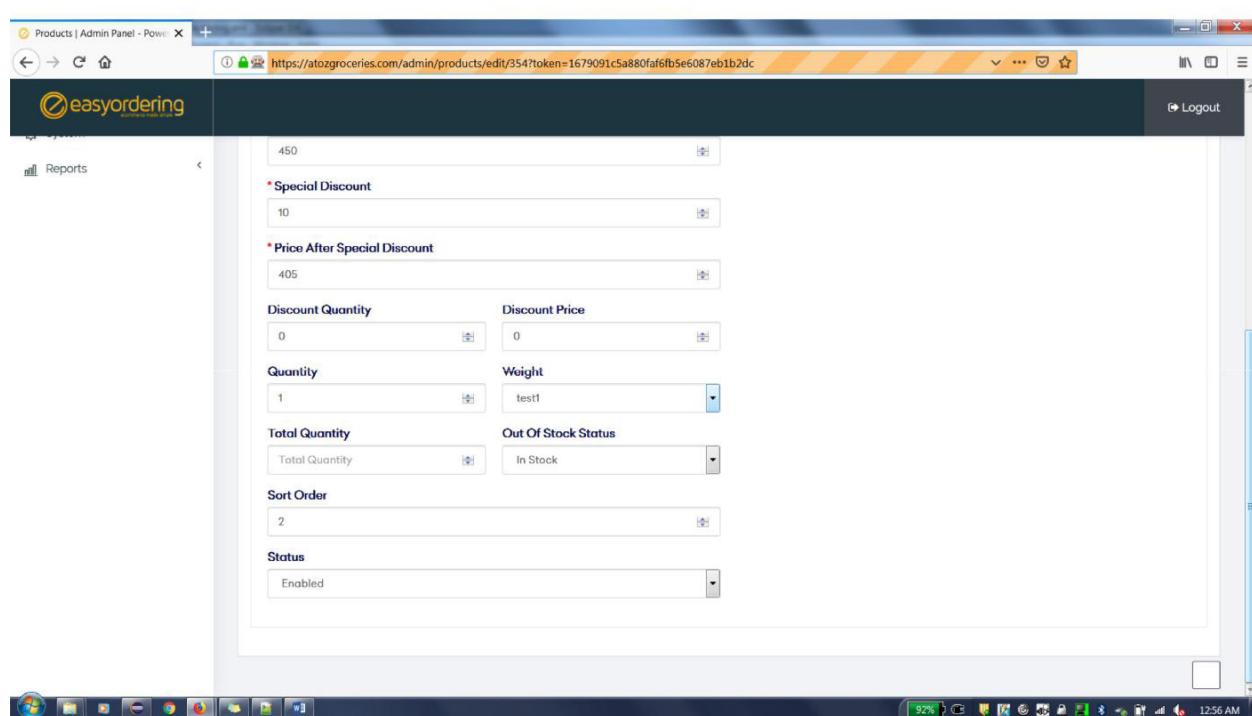
11. Sending new values to data tab



The screenshot shows the 'Edit Product' form in the 'Data' tab. The product details are as follows:

- Model:** cashew
- HSN:** HSN-4
- GST (%):** 5
- Price:** 450
- Special Discount:** 10
- Price After Special Discount:** 405

Below the form, there are two sections: 'Discount Quantity' and 'Discount Price'. Both fields are currently empty.

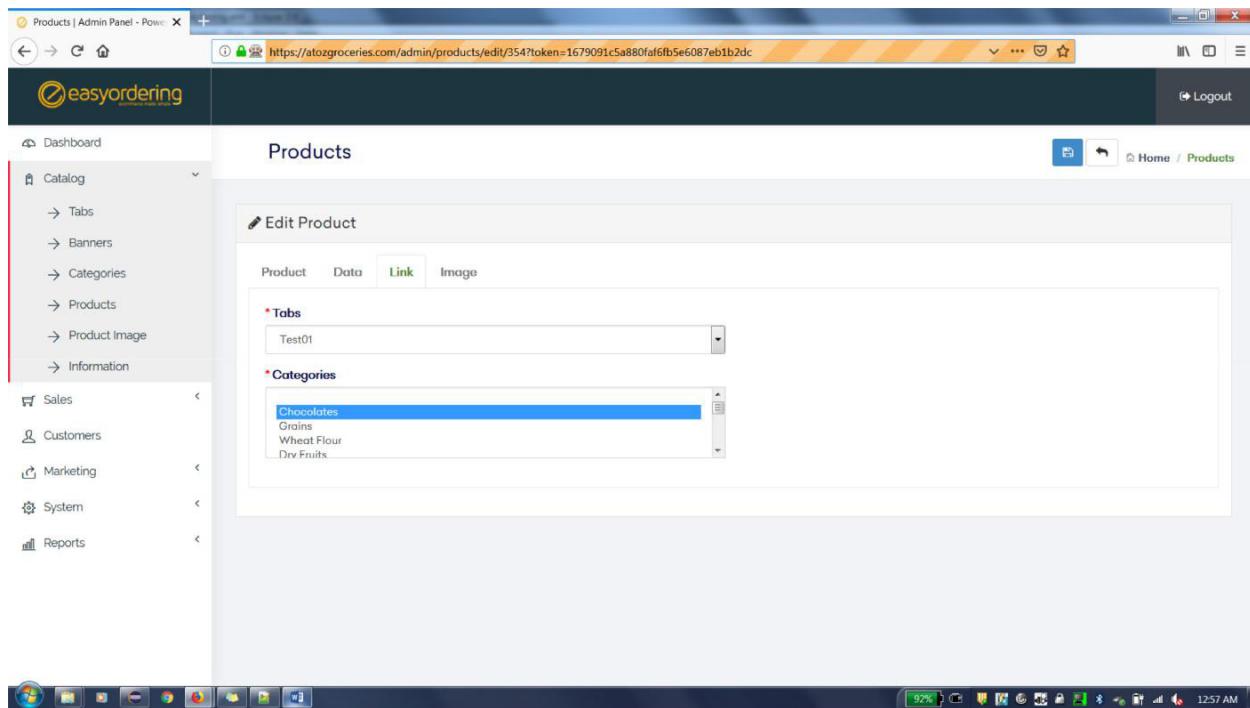


The screenshot shows the same 'Edit Product' form after changes were made. The updated values are:

- Special Discount:** 10
- Price After Special Discount:** 405

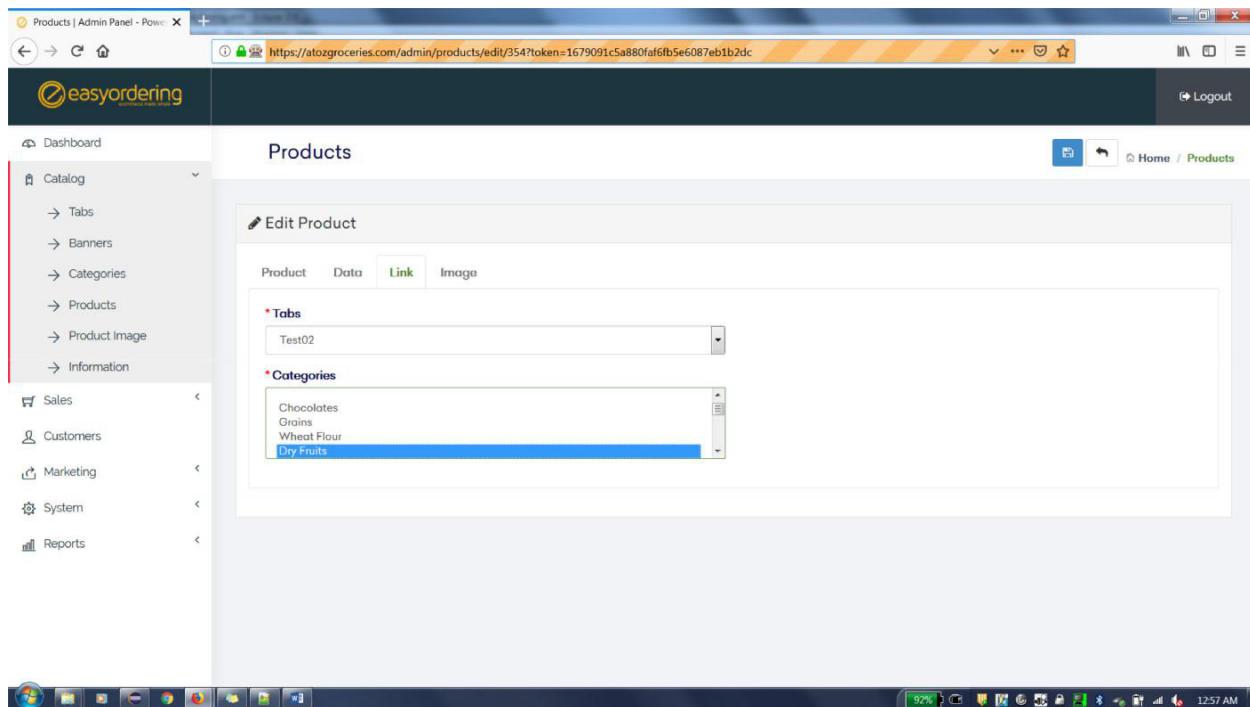
Below the form, the 'Discount Quantity' field contains the value 450, and the 'Discount Price' field contains the value 0. Other fields like 'Quantity', 'Weight', 'Total Quantity', 'Out Of Stock Status', 'Sort Order', and 'Status' remain unchanged from the initial state.

12. Opening Link Tab



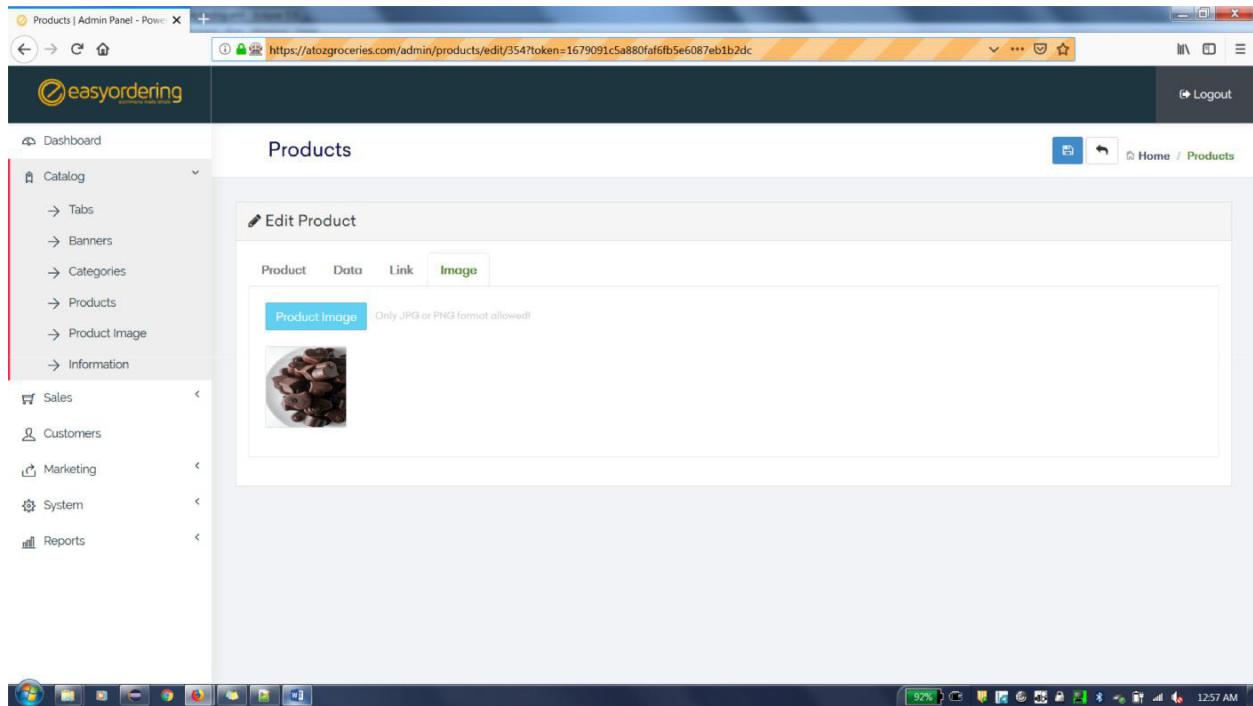
The screenshot shows a web browser window titled "Products | Admin Panel - Powe". The URL is <https://atozgroceries.com/admin/products/edit/354?token=1679091c5a880faf6fb5e6087eb1b2dc>. The main content area is titled "Products" and contains a sub-form titled "Edit Product". The sub-form has tabs: "Product", "Data", "Link" (which is highlighted in green), and "Image". Under the "Link" tab, there are two sections: "Tabs" and "Categories". The "Tabs" section contains a dropdown menu with "Test01" selected. The "Categories" section contains a dropdown menu with "Chocolates" selected, followed by "Grains", "Wheat Flour", and "Dry Fruits". The left sidebar of the admin panel includes links for Dashboard, Catalog (with sub-links for Tabs, Banners, Categories, Products, Product Image, and Information), Sales, Customers, Marketing, System, and Reports.

13. Selecting new values in Link Tab

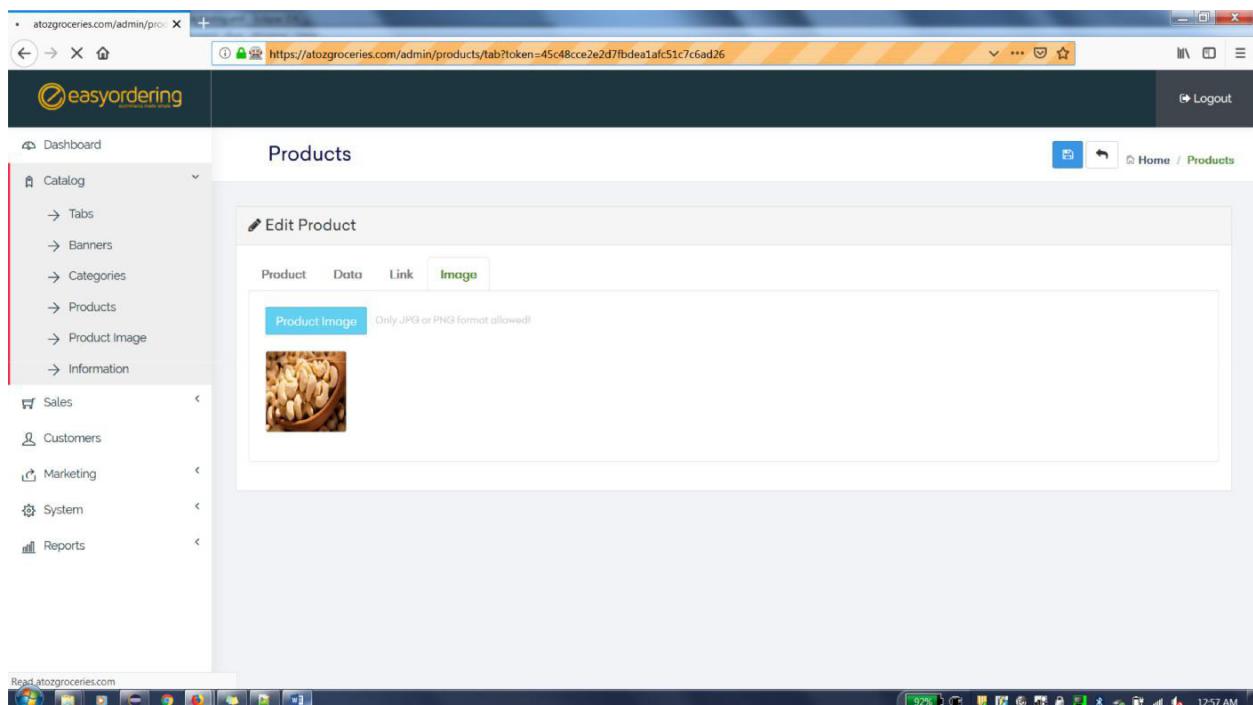


This screenshot is identical to the one above, showing the "Edit Product" form with the "Link" tab selected. However, in the "Categories" dropdown, the selection has changed from "Chocolates" to "Dry Fruits". The rest of the interface, including the sidebar and the overall layout, remains the same.

14. Opening Image Tab



15. Selecting new image and Clicking on Save



16. Validation of edited product in Admin Page

The screenshot shows a Windows desktop environment with a browser window titled "Products | Admin Panel - Powe". The URL is https://atozgroceries.com/admin/products/lab?token=45c48cce2e2d7fbdea1afc51c7c6ad26. The page displays a success message: "Success: You have successfully updated product!". Below this is a "Product List" table with the following data:

#	Product Name	Image	Model	Price	Total Quantity	Sort Order	Status	Action
1	New Prod Req		1098	1,890.00 1		1	Enabled	Action
2	Cashew-Harisha		cashew	425.25 472.5		2	Enabled	Action
3	Basmathi Rice-harisha		IndiaGate Rice	174.96 194.4	1	1	Enabled	Action
4	NewTest(working on this)		NewTest	425.25 472.5	5	1	Enabled	Action
5	Bisley Water		Model-W	19.95 21	25	0	Enabled	Action
6	NewBadhamProduct		New	1,320.00 125		1	Enabled	Action

17. Logging out from Admin Page

The screenshot shows a Windows desktop environment with a browser window titled "Admin Panel | Powered By A&S Technologies". The URL is https://atozgroceries.com/admin?action=login&&1679091c5a880faf6fb5e6087eb1b2dc. The page displays a success message: "Success: You have successfully logged out!". Below this is a "Login" form with the following fields:

Login
Please enter your credentials to login.

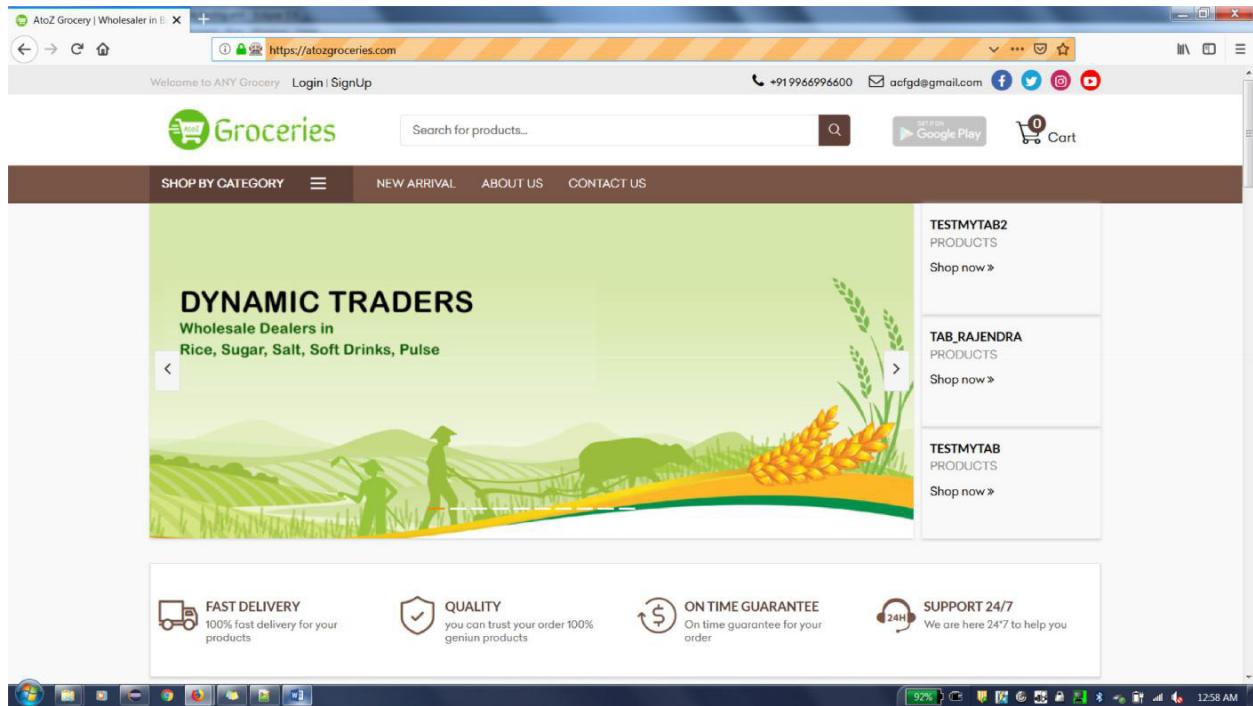
Email
 Email address
Your unique email address to app

Password
 Password
Your unique password to app

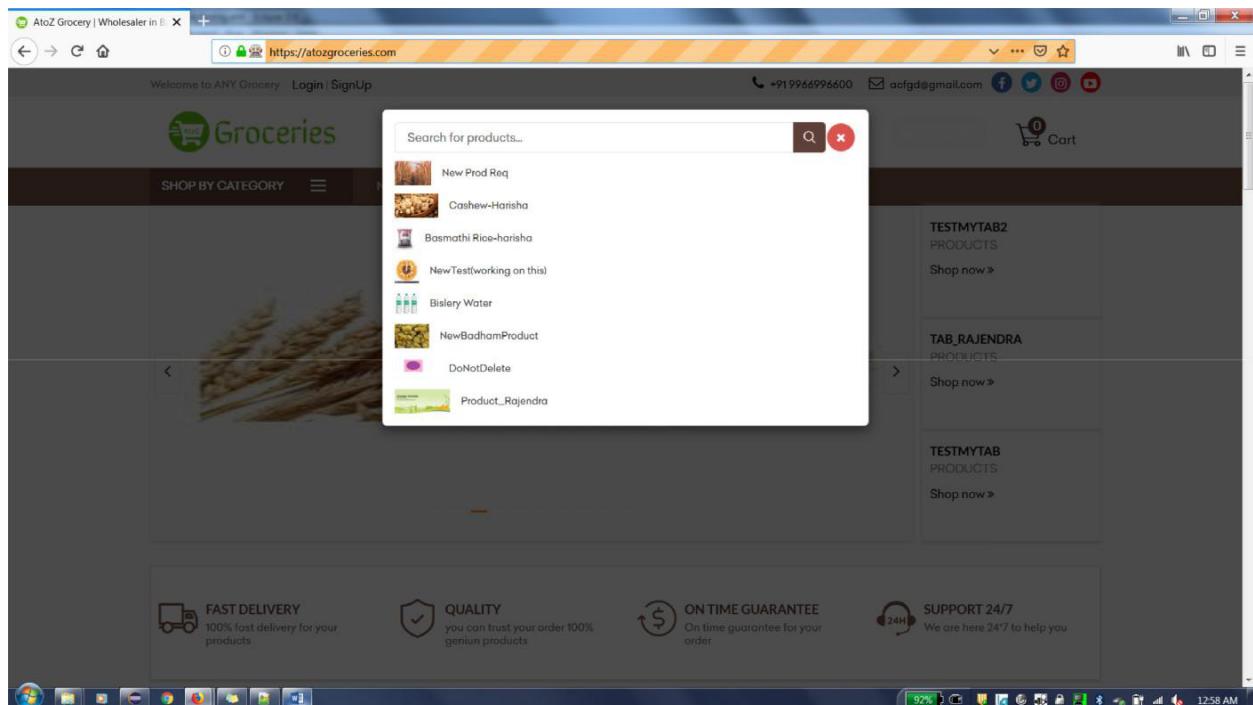
Login [Forgot Password](#)

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

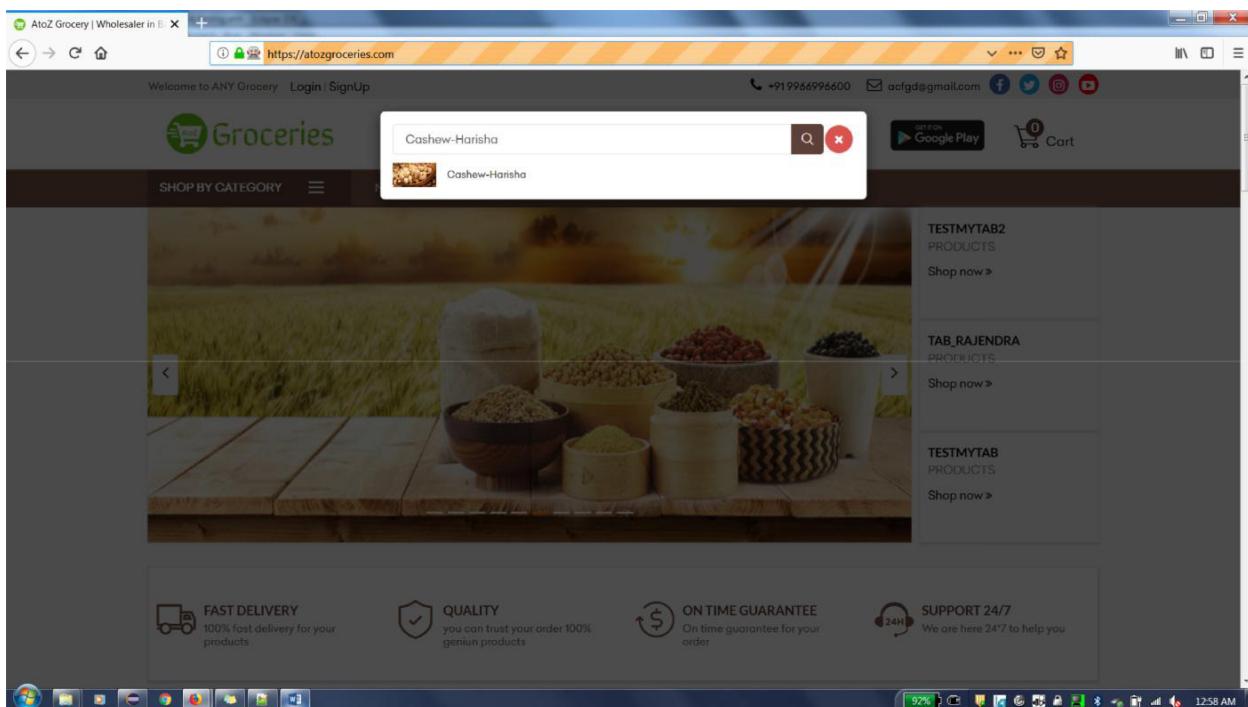
18. Navigating to User Page and Clicking on the Search for products



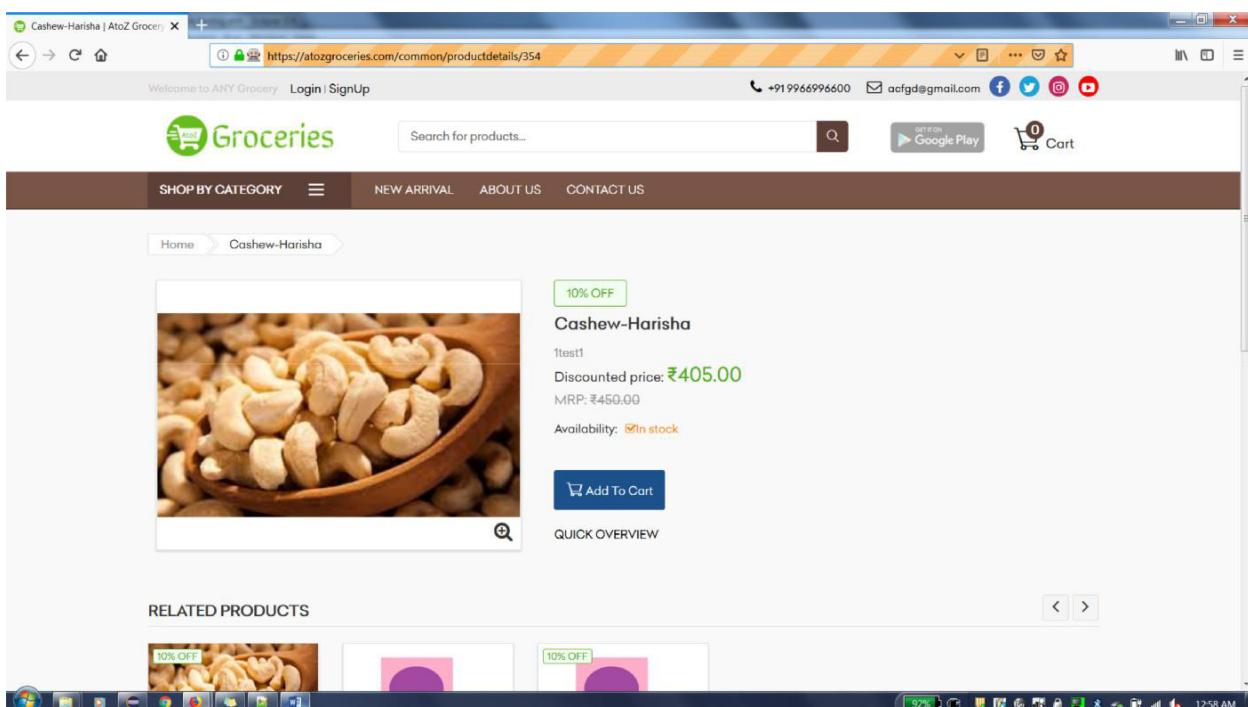
19. Clicking on the Search for products will open another search for products



20. Sending the value of edited product for validation and clicking on the desired product



21. Validation of Edited Product in User Page



22. Output from console

The screenshot shows the Eclipse IDE interface with the title bar "EclipseWorkSpace - SeleniumFramework/testng.xml - Eclipse IDE". The left sidebar displays a package structure for "com.ibm.test" containing various test classes like AgeException, Collection, DataDrivenFramework, etc. The main editor area shows the content of the "testng.xml" file:

```

<?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">
<methods><include name="Day8Testcase"></include></methods>
</class>
</classes>
</test> <!-- Test -->
</suite> <!-- Suite -->

```

Below the code, the status bar indicates "<terminated> SeleniumFramework_testng.xml [TestNG] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (Dec 24, 2018, 12:54:14 AM)". The bottom of the screen shows the Windows taskbar with icons for Start, Task View, File Explorer, Edge, and others.

23. Automated Index.html report

The screenshot shows a web browser window titled "TestNG reports" with the URL "file:///C/Harisha/Selenium_Training/EclipseWorkSpace/SeleniumFramework/test-output/index.html#". The page content is the TestNG report for the "Suite" test case:

Test results
1 suite

Reporter output for Suite	
Day8TestCase(Chocolate, Cashew-Harisha, Cashew, cashew, 5, 450, 10, 2, Enabled, Test02, Dry Fruits, C:\Harisha\Se...)	
Assertion on Edited Product in Admin Page!! Assertion on Edited Product in User Page!!	

Suite

- Info
 - C:\Harisha\Selenium_Training\EclipseWorkSpace\SeleniumFramework\testng.xml
 - 1 test
 - 0 groups
 - 1 Times
 - 0 Failed methods
 - 0 Ignored methods
 - Chronological view

Results

- method, 1 passed
 - Passed methods (hide)
 - Day8TestCaseChocolate, Cashew-Harisha, Cashew, cashew, 5, 450, 10, 2, Enabled, Test02, Dry Fruits, C:\Harisha\Se...)

24. Automated Emailable report

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "TestNG Report" and displays an "emailable-report.html" file. The report summary shows 1 passed test, 0 skipped, and 0 failed. The total time taken is 286,383 ms. Below this, a detailed table lists the test case, its class, method, start time, and duration. The test case is "com.ibm.test.BaseTest#Day8TestCase" and it passed. The parameters for this test are listed in a table:

Parameter #1	Parameter #2	Parameter #3	Parameter #4	Parameter #5	Parameter #6	Parameter #7	Parameter #8	Parameter #9	Parameter #10	Parameter #11	Parameter #12
Chocolate	Cashew-Harisha	Cashew	cashew	5	450	10	2	Enabled	Test02	Dry Fruits	C:\Harisha\Selenium_Training\EclipseWorkSpace\SeleniumFramework\ TestData\cashew.jpg

Below the parameter table, there is a "Messages" section containing two assertions:

- Assertion on Edited Product in Admin Page!!
- Assertion on Edited Product in User Page!!

At the bottom of the report, there is a link to "back to summary".

*** End of the document ***