

Day 1 - 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.CatalogPage;
import com.ibm.pages.CategoryPage;
import com.ibm.pages.DriverLaunch;
import com.ibm.pages.LoginPage;

//jars would be in the link
// https://drive.google.com/open?id=10a7AzFN8nQapvatjRToU_bwDgpf18QKI
public class BaseTest extends DriverLaunch
{
    @Test
    public void Category() throws InterruptedException, IOException
    {
        LoginPage login=new LoginPage(driver,wait);
        CatalogPage catalog=new CatalogPage(driver,wait);
        CategoryPage category=new CategoryPage(driver,wait);

        try {

            //getting the data from the properties file using Hash map
            String url=hashData.get("url");
            String username=hashData.get("username");
            String password=hashData.get("password");
            String categoryname=hashData.get("categoryname");
            String tagtitle=hashData.get("tagtitle");
            String sortorder=hashData.get("sortorder");
            String status=hashData.get("status");
            String showdata=hashData.get("show");
            //String cateImg=hashData.get("image");

            //navigating to the Admin URL of atozgroceries
            driver.get(url);

            //login page is used to provide the input of username and password

            login.EnterEmail(username);
            login.EnterPassword(password);
            login.clickLogin();
        }
    }
}
```

```

    //In catalog page we will find the catalog element and click on that to
open categories

    catalog.OpenCatalog();

    //category page

    category.clickCategory();
    //Sending input for the new category
    category.addNewCategory(categoryname,tagtitle,sortorder,status);

    String cateTitle= category.getCategoriesTitle();
    Assert.assertTrue(cateTitle.contains("Category"), "Assertion on
categories page!!");

    //Selecting to show All in categories
    category.displayAllCategories(showdata);

    //validating the newly added category
    String actualMsg=category.dataPageSource();
    Assert.assertTrue(actualMsg.contains(categoryname), "Assertion on newly
added category!!");
    System.out.println("Successful assertion on newly added category!!");
}
catch(Exception e)
{
    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/Day1_TestCase1_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();
}
}

```

CatalogPage.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 CatalogPage {

    @FindBy(partialLinkText="Catalog")
    WebElement catalogEle;

    WebDriver driver;
    WebDriverWait wait;

    public CatalogPage(WebDriver driver, WebDriverWait wait)

```

```

    {
        this.driver=driver;
        this.wait=wait;
        PageFactory.initElements(driver, this);
    }
    public void OpenCatalog()
    {
        wait.until(ExpectedConditions.visibilityOf(catalogEle));
        catalogEle.click();
    }
}

```

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

public class CategoryPage
{
    //Finding the Elements
    @FindBy(xpath="/html/body/div[1]/div[1]/div/ul/li[2]/ul/li[3]/a")
    WebElement categoryEle;

    @FindBy(xpath="//a[@title='Add New']")
    WebElement addNewEle;

    @FindBy(name="name")
    WebElement cateNameEle;

    @FindBy(name="tag_title")
    WebElement tagTitleEle;

    @FindBy(name="sort")
    WebElement sortEle;

    @FindBy(name="status")
    WebElement statusEle;

    @FindBy(xpath="//*[@id='files']")
    WebElement imageEle;

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

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

```

```

    WebDriver driver;
    WebDriverWait wait;

    //Constructor
    public CategoryPage(WebDriver driver,WebDriverWait wait)
    {
        this.driver=driver;
        PageFactory.initElements(driver, this);
    }

    //this method is used to click on the Categories under the catalog
    public void clickCategory()
    {
        categoryEle.click();
    }

    // This method is used to add New Category to the catalog
    public void addNewCategory(String name,String tagtitle,String sortorder,String
status)
    {
        // Javascript Executer is used to scroll the window
        JavascriptExecutor js=(JavascriptExecutor) driver;
        //clicking on + sign for adding new category
        addNewEle.click();
        //providing the new category details
        cateNameEle.click();
        cateNameEle.sendKeys(name);
        tagTitleEle.sendKeys(tagtitle);
        js.executeScript("window.scrollBy(0,500)");
        sortEle.sendKeys(sortorder);
        js.executeScript("window.scrollBy(500,1000)");
        Select selectStatus= new Select(statusEle);
        selectStatus.selectByVisibleText(status);

        imageEle.sendKeys("C:\\\\Harisha\\\\Selenium_Training\\\\EclipseWorkSpace\\\\SeleniumF
ramework\\\\TestData\\\\milk_products.jpg");
        driver.findElement(By.id("toTop")).click();
        saveEle.click();
    }
    public String getCategoriesTitle()
    {
        String title= driver.getTitle();
        return title;
    }
    // After saving this will redirect to categories page.

    //showSel will select "ALL" items or display
    public void displayAllCategories(String showvalue)
    {
        JavascriptExecutor js=(JavascriptExecutor) driver;
        Select showSel=new Select(showEle);
        showSel.selectByVisibleText(showvalue);
        //scrolling the window to see the products
    }
}

```

```

        js.executeScript("window.scrollBy(0,500)");
        js.executeScript("window.scrollBy(500,1000)");
        js.executeScript("window.scrollBy(1000,2000)");
    }
    public String dataPageSource()
    {
        String dataPgSrc=driver.getPageSource();
        return dataPgSrc;
    }
}

```

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

```

testing.xml

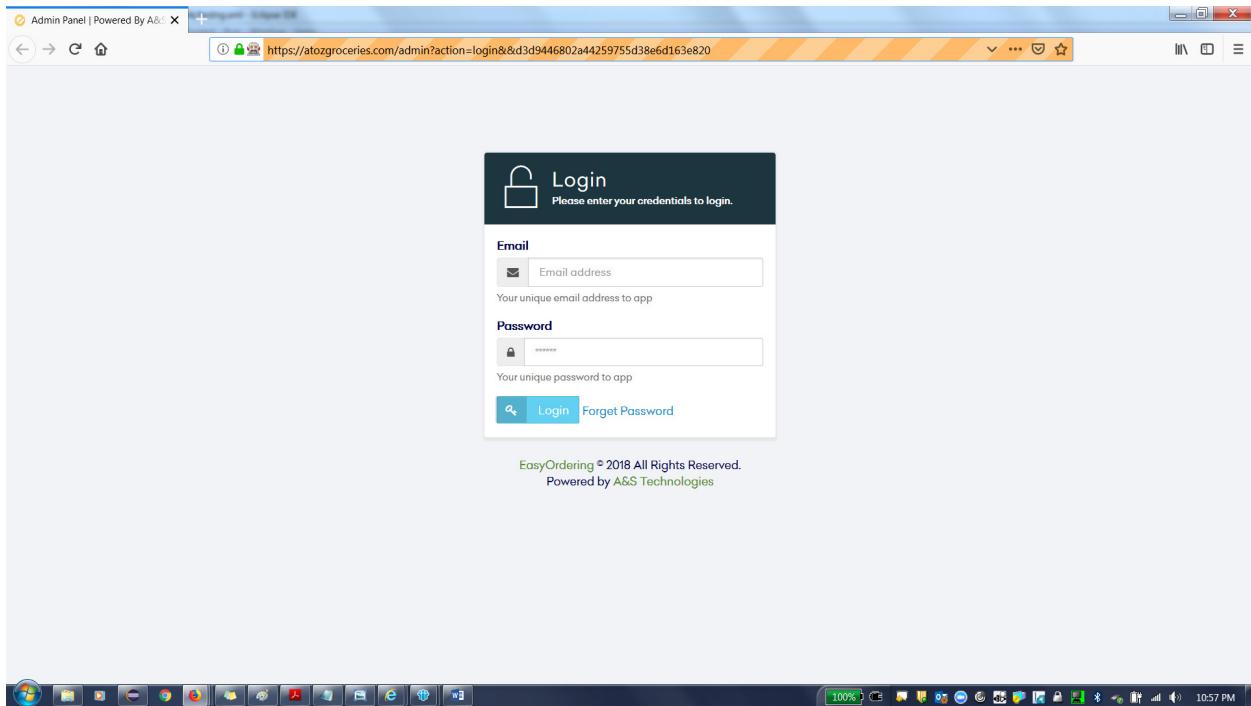
```

<?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 -->

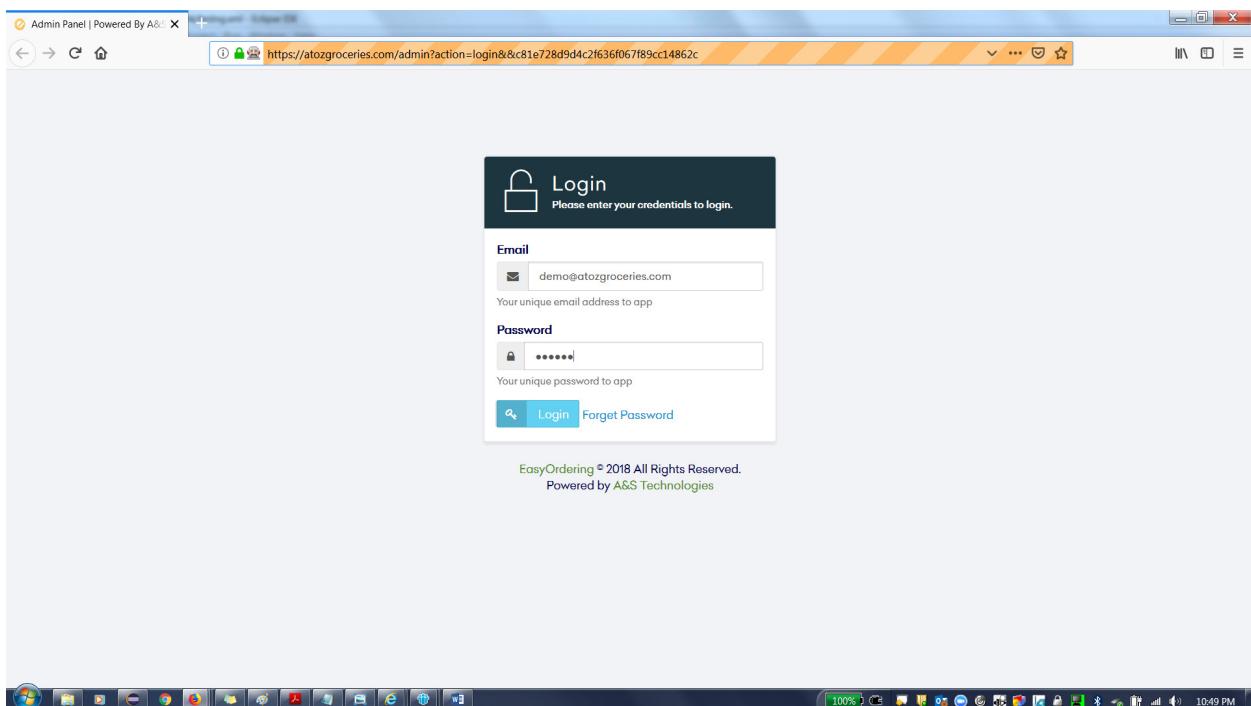
```

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 Admin Panel dashboard for easyordering. On the left, a sidebar menu includes 'Catalog', 'Sales', 'Customers', 'Marketing', 'System', and 'Reports'. The main area features four large cards: 'Total Orders' (0), 'Total Sales' (₹299.25), 'Total Customers' (0), and 'Total Products' (47). Below these is a section titled 'Latest Orders' with a table:

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

4. Clicking on Catalog

The screenshot shows the Admin Panel dashboard with the 'Catalog' option selected in the sidebar. The sidebar now lists 'Catalog' with sub-options: 'Tabs', 'Banners', 'Categories', 'Products', 'Product Image', and 'Information'. The main area remains the same as in the previous screenshot, displaying the dashboard cards and the 'Latest Orders' table.

5. Clicking on Categories

The screenshot shows the 'Categories' page in the easyordering Admin Panel. The left sidebar has a 'Catalog' section with 'Categories' selected. The main area displays a table titled 'Category List' with columns: #, Category Name, Image, Sort Order, Status, and Action. The table contains six rows, each representing a category with a corresponding image:

#	Category Name	Image	Sort Order	Status	Action
1	Rice Items		0	Enabled	Action ▾
2	Pulses Items		2	Enabled	Action ▾
3	Whole Grains		3	Enabled	Action ▾
4	Spices		4	Enabled	Action ▾
5	Flours		5	Enabled	Action ▾
6	Dry Fruits		6	Enabled	Action ▾

6. Clicking on "+" Add New category

The screenshot shows the 'Add Category' form in the easyordering Admin Panel. The left sidebar has a 'Catalog' section with 'Categories' selected. The main area has a title 'Add Category' and fields for 'Category Name' (with a placeholder 'Category Name') and 'Parent' (a dropdown menu). Below these is a 'Description' rich text editor with various formatting options.

7. Providing new category information

The screenshot shows the 'Category | Admin Panel' interface. On the left, a sidebar lists navigation items: Banners, Categories, Products, Product Image, Information, Sales, Customers, Marketing, System, and Reports. The main content area is titled 'Category Name' and contains a text input field with the value 'Milk Products'. Below it is a 'Parent' dropdown menu, which is currently empty. A rich text editor is provided for 'Description'. At the bottom of the form, there is a 'Meta Tag Title' field containing 'Milk Products'. The browser's address bar shows the URL: <https://atozgroceries.com/admin/category/new?token=e4da3b7fbce2345d7772b0674a318d5>. The taskbar at the bottom of the screen shows various application icons.

8. After providing input clicking on save

The screenshot shows the same 'Category | Admin Panel' interface after saving the category. The 'Category Name' field now contains 'Milk Products'. The 'Meta Tag Title' field also contains 'Milk Products'. Below these fields, there are sections for 'Meta Tag Keywords' (containing 'Meta Tag Keywords'), 'Top' (with a checked checkbox), 'Sort Order' (set to '12'), 'Status' (set to 'Enabled'), and 'Category Image' (which displays a thumbnail image of various milk products). The browser's address bar and taskbar are visible at the top and bottom of the screen respectively.

9. Success message after adding new category

The screenshot shows the 'Categories' page in the easyordering Admin Panel. A green success message at the top right of the main content area reads: 'Success: You have successfully added category!'. Below this message is a table titled 'Category List' with the following columns: #, Category Name, Image, Sort Order, Status, and Action. The table contains five rows of data:

#	Category Name	Image	Sort Order	Status	Action
1	Rice Items		0	Enabled	Action ▾
2	Pulses Items		2	Enabled	Action ▾
3	Whole Grains		3	Enabled	Action ▾
4	Spices		4	Enabled	Action ▾
5	Flours		5	Enabled	Action ▾

10. Selecting “ALL” from show to list all categories

The screenshot shows the 'Categories' page in the easyordering Admin Panel. A green success message at the top right of the main content area reads: 'Success: You have successfully added category!'. Below this message is a table titled 'Category List' with the following columns: #, Category Name, Image, Sort Order, Status, and Action. The table contains five rows of data, identical to the previous screenshot:

#	Category Name	Image	Sort Order	Status	Action
1	Rice Items		0	Enabled	Action ▾
2	Pulses Items		2	Enabled	Action ▾
3	Whole Grains		3	Enabled	Action ▾
4	Spices		4	Enabled	Action ▾
5	Flours		5	Enabled	Action ▾

11. All the available categories are displayed

Sr No.	Category Name	Image	Category ID	Status	Action
4	Spices		4	Enabled	Action ▾
5	Flours		5	Enabled	Action ▾
6	Dry Fruits		6	Enabled	Action ▾
7	Dry Red Chilli		7	Enabled	Action ▾
8	Coriander/Daniya		8	Enabled	Action ▾
9	Sugar & Jaggery		9	Enabled	Action ▾
10	Mandakkki/Bhel puri		10	Enabled	Action ▾
11	Salt (iodised)		11	Enabled	Action ▾

Sr No.	Category Name	Image	Category ID	Status	Action
11	Salt (iodised)		11	Enabled	Action ▾
12	Avalakki		12	Enabled	Action ▾
13	Soap Products		12	Enabled	Action ▾
14	Milk Products		12	Enabled	Action ▾
15	Ground Nuts		13	Enabled	Action ▾
16	Softdrinks Coca-Cola Items		14	Enabled	Action ▾
17	Softdrinks Pepsi Items		15	Enabled	Action ▾
18	Softdrinks Other Items		16	Enabled	Action ▾
19	Mineral Water Items		17	Enabled	Action ▾

Category | Admin Panel - Power

<https://atozgroceries.com/admin/category/tab?token=c4ca4238a0b923820dcc509a6f75849b>

Logout

17	Softdrinks Coca-Cola Items		14	Enabled	Action ▾
18	Softdrinks Pepsi Items		15	Enabled	Action ▾
19	Softdrinks Other Items		16	Enabled	Action ▾
20	Mineral Water Items		17	Enabled	Action ▾
21	Oils (edible) Items		18	Enabled	Action ▾
22	Oils (Non-edible) Items		19	Enabled	Action ▾
23	Biscuit Items		20	Enabled	Action ▾
24	Chocolate Items		21	Enabled	Action ▾

Admin Panel | Powered By AtozGroceries

<https://atozgroceries.com/admin?action=login&eccbc87e4b5ce2fe28308fd9f2a7baf>

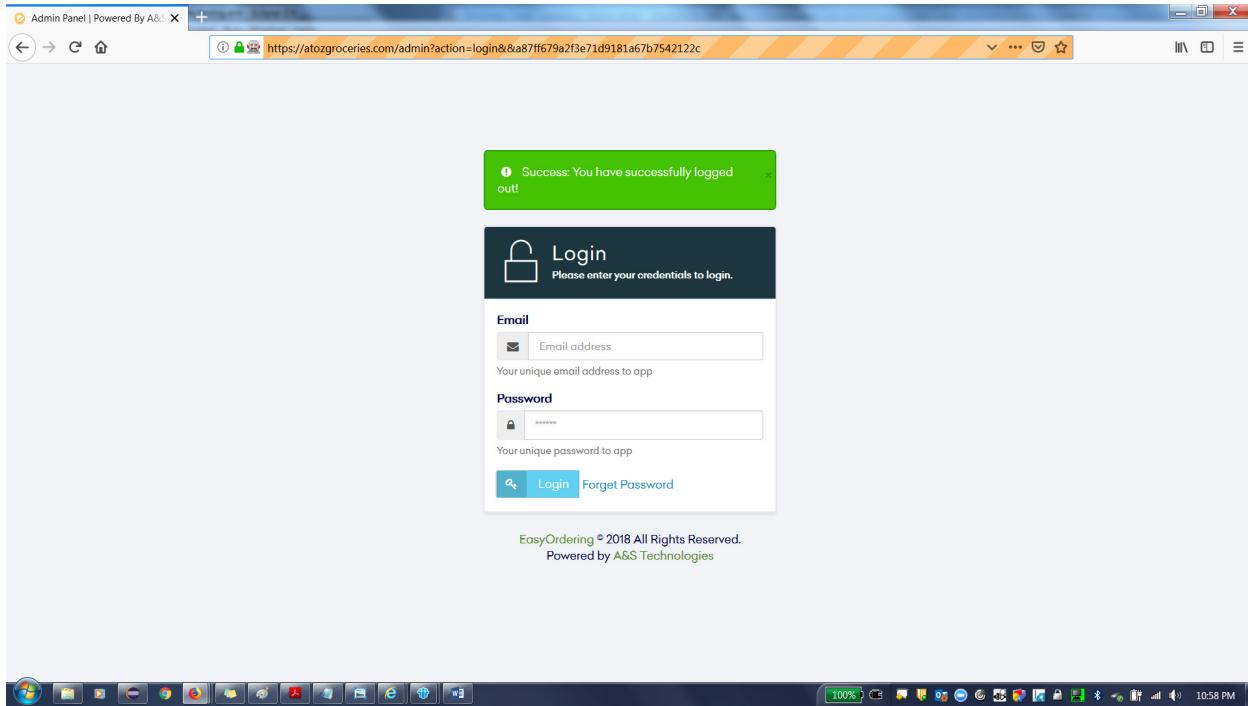
Logout

25	Vegetables		22	Enabled	Action ▾
26	Sovereigns(Mixtures) & Sweets		23	Enabled	Action ▾
27	Masala Items		24	Enabled	Action ▾
28	Papad all types		25	Enabled	Action ▾
29	Vermicel,maggi all fast cookings		25	Enabled	Action ▾
30	Masala , Powders& Sauce		27	Enabled	Action ▾
31	Colours(Edible)		28	Enabled	Action ▾

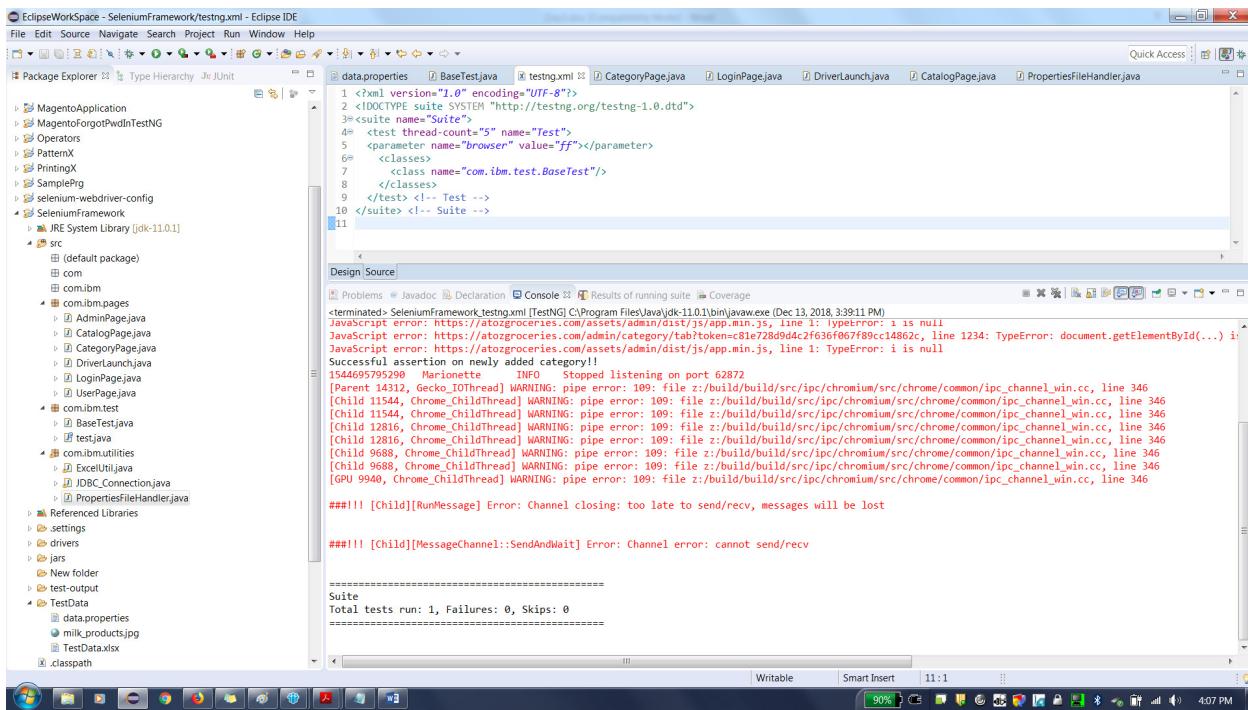
Previous 1 Next

Transferring data from atozgroceries.com...

12. After validating the newly added product, clicking on Logout



13. After quitting from the browser, the below is the console output



*** End of the document ***