PFB day 13 test case:

Description	Pre-condition	Test Steps	Validation
Edit a Category	Manually set the priorities of all other tabs to value above 2	1. Navigate onto admin panel -> Catalog -> Category 2. Populate Only Category Name and Save 3. Populate the other fields with priority set to 1 and Save	1. Validate the presence of modified Category 2. Validate the error message 3.1. Validate the success message 3.2. Validate the presence of modified record in the admin panel table 3.3 Validate the presence of modified record in the DB table 3.4 Validate the presence of modified Category on the user page

Solution:

I have created new java file as “**CategoryPage.java**”

```
package com.ibm.pages;



import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.HashMap;

import java.util.Properties;



import org.junit.Assert;

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;

import com.ibm.utilities.PropertiesFileHandler;

public class CategoryPage {

    WebDriver driver;
    WebDriverWait wait;

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

    //WebElement for link Catalog
            @FindBy(xpath="//a[contains(text(),'Catalog')]")
            WebElement catalogEle;

    //WebElement for link Categories
            @FindBy(xpath="//a[contains(text(),'Categories')]")
            WebElement catEle;
```

```
//WebElement for icon Add New
            @FindBy(xpath="//a[@title='Add New']")
            WebElement addnewEle;

//WebElement for field Category Name
            @FindBy(xpath="//input[@name='name']")
            WebElement catnameEle;

//WebElement for field Meta Tag Title
            @FindBy(xpath="//input[@name='tag_title']")
            WebElement mtitleEle;

//WebElement for field Sort Order
            @FindBy(xpath="//input[@name='sort']")
            WebElement sortEle;

//WebElement for field Status
            @FindBy(xpath="//select[@name='status']")
            WebElement statusddEle;

//WebElement for Image
            @FindBy(xpath="//input[@id='files']")
            WebElement imageEle;
```

```
//WebElement for icon Top Arrow

                @FindBy(xpath="//div[@id='toTop']")

                WebElement topEle;


//WebElement for icon Save

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

                WebElement saveEle;


//WebElement for Search text box

                @FindBy(xpath="//input[@type='search']")

                WebElement searchEle;


//WebElement for button Action

                @FindBy(xpath="(//button[contains(text(), 'Action')])[1]")

                WebElement actionEle;


//WebElement for Edit

                @FindBy(xpath="//a[@title='Edit']")

                WebElement editEle;


//WebElement for title of panel - Edit Category

                @FindBy(xpath="//div[@class='panel-title']/h4")

                WebElement headerEle;


//WebElement for success message after editing a record
```

```
		@FindBy(xpath="//div[contains(text(),'Success')]")
		WebElement successmsgEle;


//This is a method to click on link Catalog on Admin page
		public void clickOnLinkCatalog()
		{
			catalogEle.click();
		}
//This is a method to click on link Categories under Catalog on Admin page
		public void clickOnLinkCategories()
		{
			catEle.click();
		}
//This is a method to click on icon Add New on Category List of Admin page
		public void clickOnIconAddNew()
		{
			addnewEle.click();
		}
//This is a method to enter value on field Category Name using properties file
		public void enterCategoryName(String catname)
		{
			catnameEle.sendKeys(catname);
		}
//This is a method to enter value on field Meta Tag Title using properties file
		public void enterMetaTagTitle(String metatitle)
```

```
        {
            mtitleEle.sendKeys(metatitle);
        }
//This is a method to enter value on field Sort Order using properties file
        public void enterSortOrder(String sortorder)
        {
            sortEle.sendKeys(sortorder);
        }
//This is a method to select value on field Status
        public void selectStatus()
        {
            statusddEle.click();
            Select statusEle = new Select(statusddEle);
            statusEle.selectByVisibleText("Enabled");
        }
//This is a method to select image using properties file
        public void enterImage(String imagepath)
        {
            imageEle.sendKeys(imagepath);
        }
//This is a method to perform a Search
        public void performSearch(String catname)
        {
            searchEle.sendKeys(catname);
            searchEle.click();
        }
//This is a method to select Action icon
        public void clickOnAction()
        {
            actionEle.click();
        }
//This is a method to select Edit option
        public void clickOnEdit()
        {
            editEle.click();
        }
//This is a method to clear value on field Meta Tag Title
        public void clearMetaTagTitle()
        {
            mtitleEle.clear();
        }
//This is a method to clear value on field Sort Order
        public void clearSortOrder()
        {
            sortEle.clear();
        }
//This is a method to clear value on field Status
        public void clearStatus()
        {
            statusddEle.click();
```

```
                Select statusEle = new Select(statusddEle);
                statusEle.selectByVisibleText("");
            }
    //This is a method to click on icon Tpo Arrow
            public void clickOnTopArrow()
            {
                topEle.click();
            }

    //This is a method to click on icon Save
            public void clickOnIconSave()
            {
                saveEle.click();
            }

    //This is a method to validate the header of Add Tab page
            public String getHeaderMessage() throws IOException
            {
                String actualheadermsg = headerEle.getText();
                System.out.println("The first validation on presence of Category
header....");
                System.out.println("The header of Add Category page: "
+actualheadermsg);
                Assert.assertEquals("Add Category", actualheadermsg);
                return(actualheadermsg);
            }
    //This is a method to validate the header of Edit Tab page

            public String getEditHeaderMessage() throws IOException

            {

                String actualheadermsg = headerEle.getText();

                System.out.println("The first validation on presence of modified
Category header....");

                System.out.println("The header of Edit Category page: "
+actualheadermsg);

                Assert.assertEquals("Add Category", actualheadermsg);

                return(actualheadermsg);

            }


    //This is a method to validate the error message on missing mandatory field - Tab Name

            public String validationOnMisingField() throws IOException

            {

                String file="./TestData/magentodata.properties";

                PropertiesFileHandler propFileHandler = new PropertiesFileHandler();

                HashMap<String, String> data=
propFileHandler.getPropertiesAsMap(file);



                String expectedmissingtabmessage =
data.get("expectedmissingtabmessage");



                JavascriptExecutor js=(JavascriptExecutor) driver;

                String actualmissingtitlemessage = (String) js.executeScript("return
document.getElementsByName('tag_title')[0].validationMessage;");
```

```
                System.out.println("The second validation on error message of missing
field...");

                System.out.println("The error message on missing Mega Tag Title: "
+actualmissingtitlemessage);

                Assert.assertEquals(expectedmissingtabmessage,
actualmissingtitlemessage);

                return(actualmissingtitlemessage);

            }

    //

    //This is a method to validate the success message after saving the record

            public String validationOnSuccessMessage() throws IOException

            {

                String file="./TestData/magentodata.properties";

                PropertiesFileHandler propFileHandler = new PropertiesFileHandler();

                HashMap<String, String> data=
propFileHandler.getPropertiesAsMap(file);



                String expectedcatmessage = data.get("expectedcatmessage");



                String actualcatmessage = successmsgEle.getText();

                    if (actualcatmessage.contains(expectedcatmessage))

                    {

                    System.out.println("The third validation on success message
after saving record...");

                    System.out.println("The success message on saving the record:
" +actualcatmessage);

                    Assert.assertTrue(true);

                    }
                    else
                    {
                        Assert.fail();
                    }
                return(actualcatmessage);
            }

    //This is a method to validate the success message after saving the record
            public String validationOnEditSuccessMessage() throws IOException
            {
                String actualsuccessmessage = successmsgEle.getText();
                    if (actualsuccessmessage.contains("Success: You have
successfully updated category!"))
                    {
                    System.out.println("The third validation on success message
after editing record...");
                    System.out.println("The success message on saving the record:
" +actualsuccessmessage);
                    Assert.assertTrue(true);
                    }
                    else
                    {
                        Assert.fail();
                    }
                return(actualsuccessmessage);
            }
    //This is a method to validate that record is inserted into the list
            public void validationOnAddRecord() throws FileNotFoundException,
IOException
            {
                Properties p = new Properties();
                p.load(new FileInputStream("./TestData/magentodata.properties"));


                String pagesource = driver.getPageSource();


                if(pagesource.contains(p.getProperty("catname"))) {
                    System.out.println("The validation on presence of record in the
Admin panel table is confirmed!");
                    Assert.assertTrue(true);
                }
                else {
                    System.out.println("The category name is not added to this
list");
                    Assert.fail();
                }
            }
    //This is a method to validate that record is updated into the list
            public void validationOnEditRecord() throws FileNotFoundException,
IOException
            {
                Properties p = new Properties();
                p.load(new FileInputStream("./TestData/magentodata.properties"));
                String pagesource = driver.getPageSource();


                if(pagesource.contains(p.getProperty("catname"))) {
                    System.out.println("The validation on presence of record in the
Admin panel table is confirmed!");
                    Assert.assertTrue(true);
                }
                else {
                    System.out.println("The category name is not added to this
list");
                    Assert.fail();
                }
            }


}
```

I have edited existing java file as “**UserPage.java**”

```
package com.ibm.pages;


import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Properties;
import java.util.concurrent.TimeUnit;

import org.junit.Assert;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Reporter;

import com.ibm.utilities.PropertiesFileHandler;

public class UserPage
{
        WebDriver driver;
        WebDriverWait wait;

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

```

```
public void launchUserPage () throws IOException, InterruptedException
{

        String file="./TestData/magentodata.properties";

        PropertiesFileHandler propFileHandler = new PropertiesFileHandler();
        HashMap<String, String> data= propFileHandler.getPropertiesAsMap(file);

        String userurl = data.get("userurl");

        //WebDriverWait wait=new WebDriverWait(driver, 60);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30,TimeUnit.SECONDS);
        driver.get(userurl);

}

public void verifyaddtab() throws IOException, InterruptedException
{

        String file="./TestData/magentodata.properties";

        PropertiesFileHandler propFileHandler = new PropertiesFileHandler();
        HashMap<String, String> data= propFileHandler.getPropertiesAsMap(file);
```

```
                String userurl = data.get("userurl");


                System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

                WebDriver driver = new ChromeDriver();

                WebDriverWait wait = new WebDriverWait(driver, 60);

                driver.manage().window().maximize();

                driver.manage().timeouts().implicitlyWait(30,TimeUnit.SECONDS);

                driver.get(userurl);


                Properties p = new Properties();

                p.load(new FileInputStream("./TestData/magentodata.properties"));


                String title = driver.getTitle();

                Assert.assertEquals("AtoZ Grocery | Wholesaler in Bangalore", title);


                String pagesource = driver.getPageSource();

                if(pagesource.contains(p.getProperty("tabname")))

                {

                        System.out.println("The presence of newly added tab name is confirmed on
User page!");

                        Assert.assertTrue(true);

                }

                else {

                        System.out.println("The tab name doesn't exist on User page");

                        Assert.fail();
```

```
        }



}

public void verifyedittab() throws IOException, InterruptedException

{


        String file="./TestData/magentodata.properties";


        PropertiesFileHandler propFileHandler = new PropertiesFileHandler();

        HashMap<String, String> data= propFileHandler.getPropertiesAsMap(file);


        String userurl = data.get("userurl");


        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        WebDriverWait wait = new WebDriverWait(driver, 60);

        driver.manage().window().maximize();

        driver.manage().timeouts().implicitlyWait(30,TimeUnit.SECONDS);

        driver.get(userurl);


        Properties p = new Properties();

        p.load(new FileInputStream("./TestData/magentodata.properties"));
        String title = driver.getTitle();

        Assert.assertEquals("AtoZ Grocery | Wholesaler in Bangalore", title);



        String pagesource = driver.getPageSource();

        if(pagesource.contains(p.getProperty("edittabname")))

        {

            System.out.println("The presence of edited tab name is confirmed on User page!");

            Assert.assertTrue(true);

        }

        else {

            System.out.println("The tab name doesn't exist on User page");

            Assert.fail();

        }




    }


    //WebElement for link SignUp

        @FindBy(xpath="//a[contains(text(),'SignUp')]")

        WebElement signupEle;

    //WebElement for field Full Name

        @FindBy(xpath="//input[@id='name']")

        WebElement nameEle;

    //WebElement for field Phone Number

        @FindBy(xpath="//input[@id='pnum']")
```

```
            WebElement pnumEle;
//WebElement for field Password
            @FindBy(xpath="//input[@id='password']")
            WebElement pwdEle;
//WebElement for field Confirm Password
            @FindBy(xpath="//input[@id='cpassword']")
            WebElement cpwdEle;
//WebElement for checkbox Term & Conditions
            @FindBy(xpath="//input[@id='tccheckbox']")
            WebElement tccEle;
//WebElement for button SignUp
            @FindBy(xpath="//button[@id='mem_signup']")
            WebElement bsignupEle;
//WebElement for link My Account
            @FindBy(xpath="(//i[@class='flaticon-user-outline'])[1]")
            WebElement myacctEle;
//WebElement for link Logout
            @FindBy(xpath="//a[contains(text(),'Log Out')]")
            WebElement logoutEle;

//WebElement for link Login
            @FindBy(xpath="//a[contains(text(),'Login')]")
            WebElement loginlinkEle;
//WebElement for field Phone Number
            @FindBy(xpath="//input[@id='pnum2']")
```

```
        WebElement mobileEle;

//WebElement for field Password

        @FindBy(xpath="//input[@id='pword2']")

        WebElement passwordEle;

//WebElement for button Login

        @FindBy(xpath="//button[@id='mem_login']")

        WebElement loginbuttonEle;



//WebElement for field 1st Search

        @FindBy(xpath="//input[@id='search-box']")

        WebElement searchEle;



//WebElement for icon Add to Cart

        @FindBy(xpath="//a[contains(text(),'Add To Cart')]")

        WebElement addcartEle;

//WebElement for icon Cart

        @FindBy(xpath="//p[@id='bigCart']")

        WebElement cartEle;

//WebElement for icon Checkout

        @FindBy(xpath="//a[contains(text(),'Check Out')]")

        WebElement checkoutEle;

//WebElement for icon Continue to Payment

        /*@FindBy(xpath="(//a[@class='btn orange'])[1]")

        WebElement ctopaymentEle;*/

//WebElement for icon Confirm Orders
```

```
    }
    @FindBy(xpath="//a[@id='confirm-order-id']")
    WebElement corderEle;
//WebElement for icon My Order
    @FindBy(xpath="(//a[contains(text(),'My Orders')])[2]")
    WebElement myorderEle;
//WebElement for Customer name on My Order
    @FindBy(xpath="//*[@class='table table-bordered table-hover']/tbody/tr/td[2]")
    WebElement custnameEle;

    public void clickOnLinkSignUp()
    {
        signupEle.click();
    }
    public void enterFullName(String fullname)
    {
        nameEle.sendKeys(fullname);
    }
    public void enterPhoneNo(String pno)
    {
        pnumEle.sendKeys(pno);
    }
    public void enterPassword(String cpassword)
    {
        pwdEle.sendKeys(cpassword);
    }
```

```
public void enterConfirmPassword(String confpassword)

{

        cpwdEle.sendKeys(confpassword);

}

public void clickOnTermCondition()

{

        tccEle.click();

}

public void clickOnButtonSignUp()

{

        bsignupEle.click();

}

public void clickOnAlertBox()

{

        Alert alertBox = driver.switchTo().alert();

        String alertmsg = alertBox.getText();

        System.out.println(alertmsg);

        alertBox.accept();

}

public void clickOnLogout()

{

        myacctEle.click();

        logoutEle.click();

}
```

```
		public void clickOnLinkLogin()

		{

			loginlinkEle.click();

		}

		public void enterMobileNo(String cpno)

		{

			mobileEle.sendKeys(cpno);

		}

		public void enterUserPassword(String cpassword)

		{

			passwordEle.sendKeys(cpassword);

		}

		public void clickOnButtonLogin()

		{

			loginbuttonEle.click();

		}

		public void performSearch() throws InterruptedException

		{

			//driver.findElement(By.xpath("(//*[@placeholder='Search for
products...'])[1]")).click();

			searchEle.click();

			Thread.sleep(3000);

			driver.findElement(By.xpath("(//*[@placeholder='Search for
products...'])[2]")).sendKeys("rice");

			Thread.sleep(3000);

			List<WebElement> searchProductLinks=
driver.findElements(By.xpath("//*[@id='searchproducts-div']/a"));
```

```
        int linkCount=searchProductLinks.size();

        System.out.println(linkCount);

        Reporter.log("Links available"+ linkCount);

        if(linkCount != 0)

        {

            System.out.println("Product Available");

            Reporter.log("Product Available");

        }


        WebElement orangeRiceEle=null;

        for(WebElement link : searchProductLinks)

        {

            System.out.println(link.getText());

            if(link.getText().toLowerCase().contains("orange"))

            {

                orangeRiceEle=link;

            }

        }

        Thread.sleep(5000);

        orangeRiceEle.click();

    }

    public void clickOnAddToCart()

    {

        addcartEle.click();

    }
```

```
public void clickOnCart()

{

    cartEle.click();

}

public void clickOnCheckout()

{

    checkoutEle.click();

}

public void clickOnContinuePayment()

{

    driver.findElement(By.xpath("//input[@id='pincode']")).click();

    driver.findElement(By.xpath("(//a[@class='btn orange'])[1]")).click();

}

public void clickOnConfirmOrder()

{

    driver.findElement(By.xpath("//input[@id='tc']")).click();

    corderEle.click();

}

public void clickOnMyOrder()

{

    myorderEle.click();

}

public void validationOnOrder() throws FileNotFoundException, IOException

{

    Properties p = new Properties();
```

```
        p.load(new FileInputStream("./TestData/magentodata.properties"));


        String customername = custnameEle.getText();
        if(customername.equalsIgnoreCase(p.getProperty("fullname"))) {
            System.out.println("The placed order is available on the list!");
            Assert.assertTrue(true);
        }
        else {
            System.out.println("The placed order is not available on the list!");
            Assert.fail();
        }
    }


    public void verifycategory() throws IOException, InterruptedException
    {


        String file="./TestData/magentodata.properties";


        PropertiesFileHandler propFileHandler = new PropertiesFileHandler();
        HashMap<String, String> data= propFileHandler.getPropertiesAsMap(file);


        String userurl = data.get("userurl");
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        WebDriverWait wait = new WebDriverWait(driver, 60);
```

```
driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(30,TimeUnit.SECONDS);

driver.get(userurl);


Properties p = new Properties();

p.load(new FileInputStream("./TestData/magentodata.properties"));


String title = driver.getTitle();

Assert.assertEquals("AtoZ Grocery | Wholesaler in Bangalore", title);


driver.findElement(By.xpath("(//span[@class='click-categories flaticon-bars'])[2]")).click();


String pagesource = driver.getPageSource();

if(pagesource.contains(p.getProperty("catname")))

{

	System.out.println("The presence of edited Category name is confirmed on User page!");

	Assert.assertTrue(true);

}

else {

	System.out.println("The tab name doesn't exist on User page");

	Assert.fail();

}
```

```
                    }


}
```

I have edited existing java file as “**BaseTest.java**”

```
package com.ibm.test;


import java.io.FileInputStream;

import java.io.IOException;

import java.sql.SQLException;

import java.util.HashMap;

import java.util.concurrent.TimeUnit;


import org.apache.poi.xssf.usermodel.XSSFCell;

import org.apache.poi.xssf.usermodel.XSSFRow;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import org.junit.Assert;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

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

import org.testng.annotations.Test;

import com.ibm.pages.AdminPage;

import com.ibm.pages.BannerPage;

import com.ibm.pages.CategoryPage;

import com.ibm.pages.CouponPage;
import com.ibm.pages.CustomerPage;

import com.ibm.pages.DB;

import com.ibm.pages.LoginPage;

import com.ibm.pages.OrderPage;

import com.ibm.pages.ProductPage;

import com.ibm.pages.ReturnPage;

import com.ibm.pages.ShippingPage;

import com.ibm.pages.TabPage;

import com.ibm.pages.UserPage;

import com.ibm.utilities.DBUtil;

import com.ibm.utilities.PropertiesFileHandler;




public class BaseTest {

	@Test
	public void Day13Testcase() throws IOException, InterruptedException,
SQLException
	{

		String file="./TestData/magentodata.properties";

		PropertiesFileHandler propFileHandler = new PropertiesFileHandler();
		HashMap<String, String> data= propFileHandler.getPropertiesAsMap(file);

		String url = data.get("url");
		String email = data.get("email");
		String password = data.get("password");
		String catname = data.get("catname");
		String metatitle = data.get("metatitle");
		String sortorder = data.get("sortorder");
		String imagepath = data.get("imagepath");


		System.setProperty("webdriver.chrome.driver",
"./drivers/chromedriver.exe");
		WebDriver driver = new ChromeDriver();
		WebDriverWait wait=new WebDriverWait(driver, 60);
		driver.manage().window().maximize();
```

```
             driver.manage().timeouts().implicitlyWait(30,TimeUnit.SECONDS);
             driver.get(url);

             LoginPage login = new LoginPage(driver);
             login.enterEmailAddress(email);
             login.enterPassword(password);
             login.clickOnLogin();

             DBUtil db = new DBUtil();
             int beforecount = db.countQuery("SELECT count(category_id) from
as_category");
             System.out.println("The total count of records in DB for Category before
creating record: " +beforecount);

             // Creating a new record for Category
             AdminPage apage=new AdminPage(driver, wait);
             CategoryPage cp = new CategoryPage(driver, wait);
             cp.clickOnLinkCatalog();
             cp.clickOnLinkCategories();
             cp.clickOnIconAddNew();
             cp.enterCategoryName(catname);
             cp.enterMetaTagTitle(metatitle);
             cp.selectStatus();
             cp.enterSortOrder(sortorder);
             cp.enterImage(imagepath);
             cp.clickOnTopArrow();
             Thread.sleep(2000);
             cp.clickOnIconSave();

             String categoryname = db.singleDataQuery("SELECT name from as_category
where sort=1");
             System.out.println("The created category on DB as: " +categoryname);
             Assert.assertEquals(categoryname, catname);

             int aftercount = db.countQuery("SELECT count(category_id) from
as_category");
             System.out.println("The total count of records in DB after creating new
category: " +aftercount);
             Assert.assertEquals(beforecount+1, aftercount);

             // Editing the newly created Category record
             Thread.sleep(2000);
             cp.performSearch(catname);
             cp.clickOnAction();
             cp.clickOnEdit();
             cp.clearMetaTagTitle();
             cp.clearSortOrder();
             cp.clearStatus();
             cp.clickOnTopArrow();
             Thread.sleep(1000);
             cp.clickOnIconSave();
             Thread.sleep(2000);
             cp.getEditHeaderMessage();
             cp.validationOnMisingField();
             cp.enterMetaTagTitle(metatitle);
```

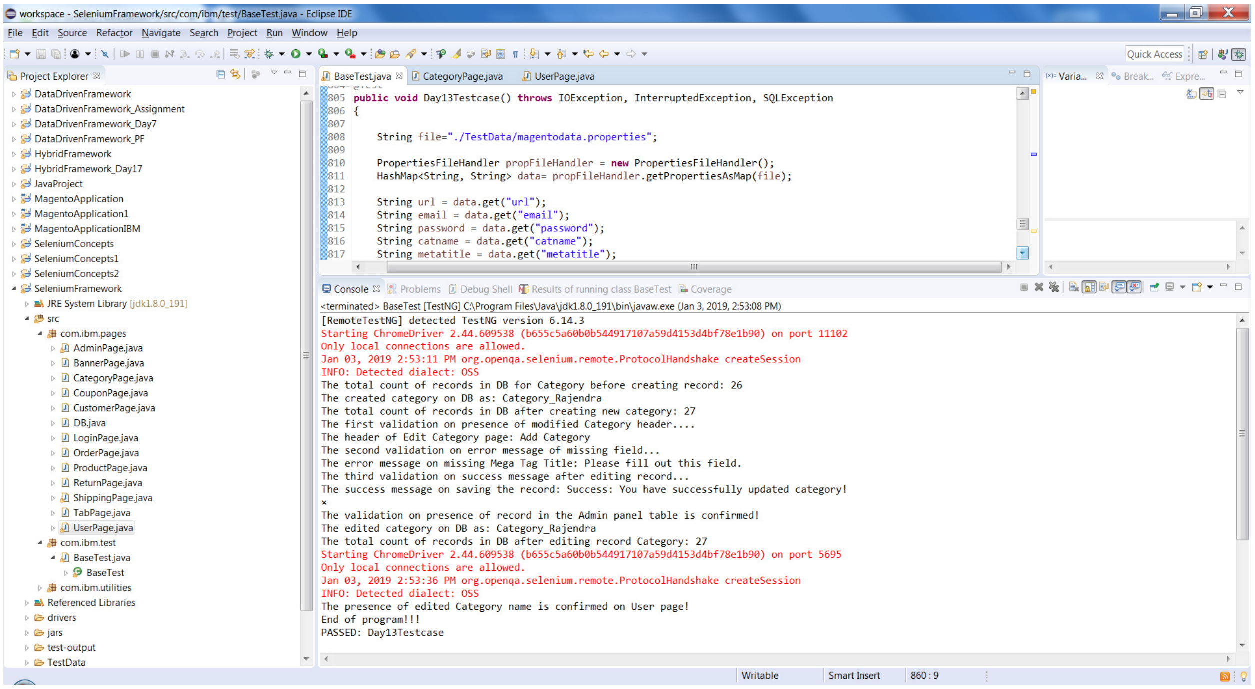
```
            cp.enterSortOrder(sortorder);
            cp.selectStatus();
            cp.clickOnTopArrow();
            Thread.sleep(1000);
            cp.clickOnIconSave();
            cp.validationOnEditSuccessMessage();
            cp.validationOnEditRecord();

            String editedname = db.singleDataQuery("SELECT name from as_category
where sort=1");
            System.out.println("The edited category on DB as: " +editedname);
            Assert.assertEquals(catname, editedname);

            int aftereditrecordcount = db.countQuery("SELECT count(category_id) from
as_category");
            System.out.println("The total count of records in DB after editing
record Category: " +aftereditrecordcount);
            Assert.assertEquals(aftercount, aftereditrecordcount);


            apage.clickOnLogOut();

            UserPage up = new UserPage (driver, wait);
            up.verifycategory();

            driver.quit();
            System.out.println("End of program!!!");

      }

}
```

I have edited properties file “**magentodata.properties**”

```
expectedcouponmessage=Success\: You have successfully added new coupon\!
expectedpincodemessage=Please fill out this field.
expectedtabheader=Add Tab
expectedmissingtabmessage=Please fill out this field.
expectedmissingordstatusmessage=Please fill out this field.
expectedtabmessage=Success: You have successfully inserted new tab!
expectedcatmessage=Success: You have successfully inserted new category!
model=Model_Rajendra
expectedtitle=Dashboard | Admin Panel - Powered By A&S
usespercustomer=1
code=ABC123
url=http\://atozgroceries.com/admin/
amount=1000
IPaddress=100.1.0.11
cpincode=560045
ccity=MADIWALA
discount=15
usespercoupon=1
metatitle=MetaTitle_Rajendra
startdate=12/20/2018
```

```
gst=8
expectedbannermessage=Success\: You have successfully added banner\!
specialdiscount=5
email=demo@atozgroceries.com
sortorder=1
returnactionname=new_refunded
cemail=test@mail.com
enddate=12/30/2018
caddress=IBM India Pvt Ltd, Manyata Tech Park, Nagawara, Outer Ring Road
cpno=9810098101
price=100
TestResult=Test Passed
password=456789
productname=Product_Rajendra
discountprice=950
confpassword=123456
fullname=Rajendra Sahoo
couponname=Coupon_Rajendra
bannername=Banner_Rajendra
imagepath=C\:\\Users\\IBM_ADMIN\\Documents\\Training\\Selenium\\Proj
Day1\\SeleniumFramework\\TestData\\banner.jpg
cpassword=123456
tabname=Tab_Rajendra_TC11
edittabname=Tab_Rajendra_TC12
userurl=http\://atozgroceries.com/
pno=9810098100
catname=Category_Rajendra
```

Screen shot of Output

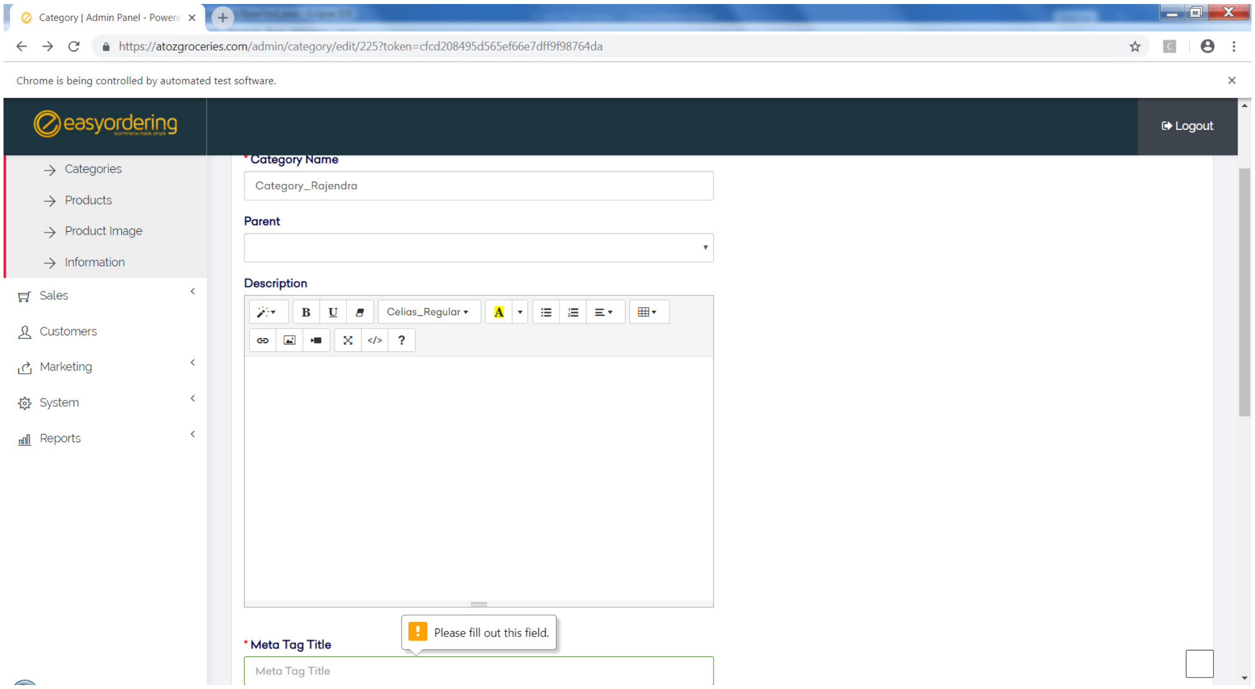


```
public void Day13Testcase() throws IOException, InterruptedException, SQLException
{

    String file="./TestData/magentodata.properties";

    PropertiesFileHandler propFileHandler = new PropertiesFileHandler();
    HashMap<String, String> data= propFileHandler.getPropertiesAsMap(file);

    String url = data.get("url");
    String email = data.get("email");
    String password = data.get("password");
    String catname = data.get("catname");
    String metatitle = data.get("metatitle");
```

```
<terminated> BaseTest [TestNG] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (Jan 3, 2019, 2:53:08 PM)
[RemoteTestNG] detected TestNG version 6.14.3
Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port 11102
Only local connections are allowed.
Jan 03, 2019 2:53:11 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
The total count of records in DB for Category before creating record: 26
The created category on DB as: Category_Rajendra
The total count of records in DB after creating new category: 27
The first validation on presence of modified Category header....
The header of Edit Category page: Add Category
The second validation on error message of missing field...
The error message on missing Mega Tag Title: Please fill out this field.
The third validation on success message after editing record...
The success message on saving the record: Success: You have successfully updated category!
×
The validation on presence of record in the Admin panel table is confirmed!
The edited category on DB as: Category_Rajendra
The total count of records in DB after editing record Category: 27
Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port 5695
Only local connections are allowed.
Jan 03, 2019 2:53:36 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
The presence of edited Category name is confirmed on User page!
End of program!!!
PASSED: Day13Testcase
```

Eclipse console:

```
[RemoteTestNG] detected TestNG version 6.14.3
Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port
11102
Only local connections are allowed.
Jan 03, 2019 2:53:11 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
The total count of records in DB for Category before creating record: 26
The created category on DB as: Category_Rajendra
The total count of records in DB after creating new category: 27
The first validation on presence of modified Category header....
The header of Edit Category page: Add Category
The second validation on error message of missing field...
The error message on missing Mega Tag Title: Please fill out this field.
The third validation on success message after editing record...
The success message on saving the record: Success: You have successfully updated
category!
×
The validation on presence of record in the Admin panel table is confirmed!
The edited category on DB as: Category_Rajendra
The total count of records in DB after editing record Category: 27
Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port
5695
Only local connections are allowed.
Jan 03, 2019 2:53:36 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
The presence of edited Category name is confirmed on User page!
End of program!!!
PASSED: Day13Testcase

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 1, Failures: 0, Skips: 0
===============================================
```



https://atozgroceries.com/admin/category/edit/225?token=cfcd208495d565ef66e7dff9f98764da

Chrome is being controlled by automated test software.

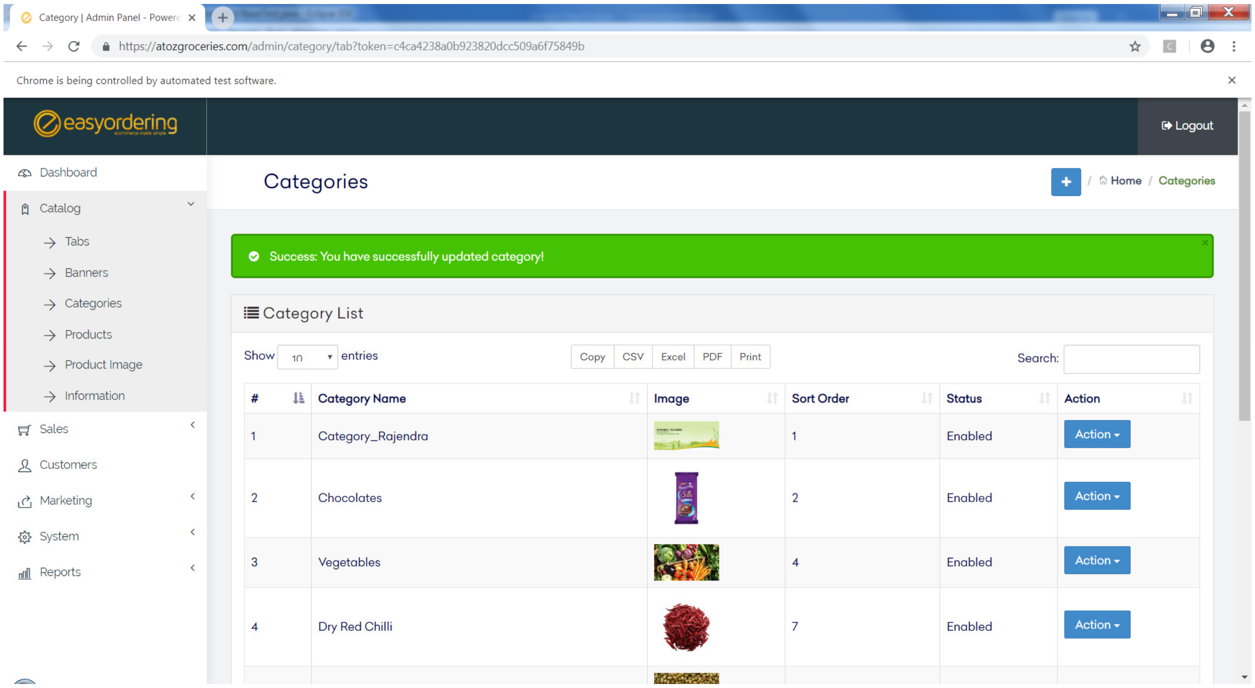
* Category Name: Category_Rajendra

Parent:

Description:

* Meta Tag Title: Meta Tag Title

Please fill out this field.



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

Chrome is being controlled by automated test software.

Categories

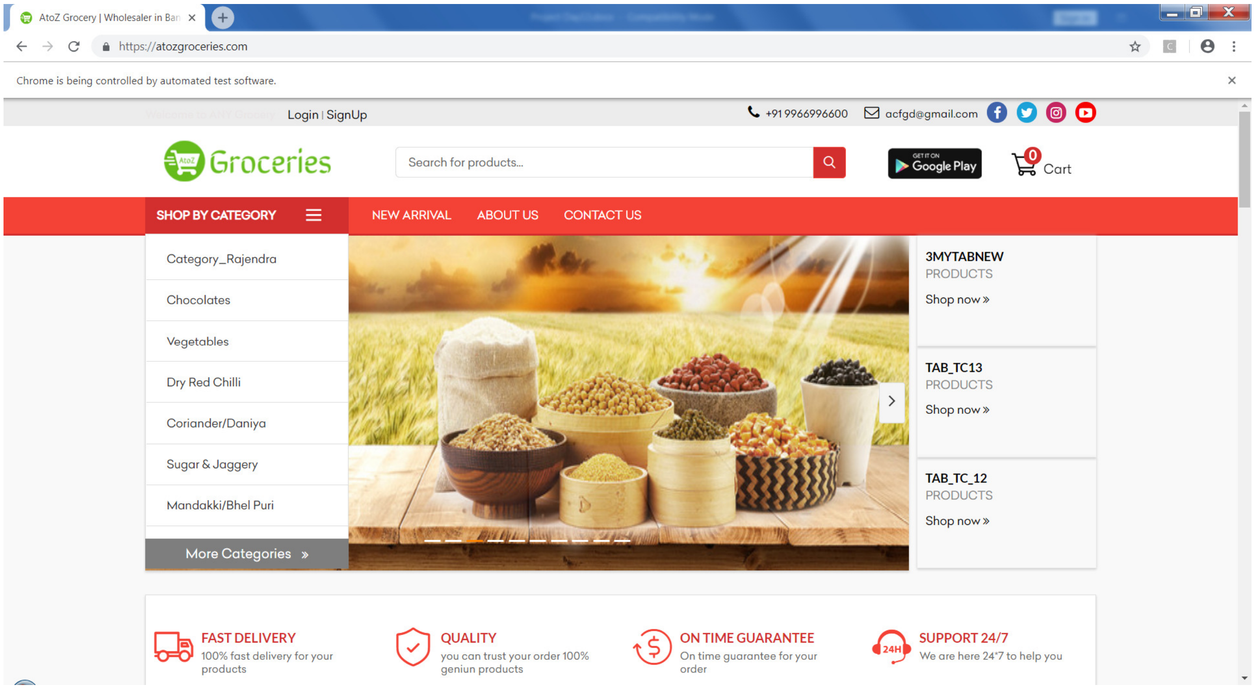
Success: You have successfully updated category!

Category List

Show 10 entries

Search:

#	Category Name	Image	Sort Order	Status	Action
1	Category_Rajendra		1	Enabled	Action
2	Chocolates		2	Enabled	Action
3	Vegetables		4	Enabled	Action
4	Dry Red Chilli		7	Enabled	Action



Chrome is being controlled by automated test software.