

PFB day 12 test case:

Description	Pre-condition	Test Steps	Validation
Edit a tab	Manually set the priorities of all other tabs to value above 2	1. Navigate onto admin panel -> Catalog -> Tabs 2. Populate Only Tab Name and Save 3. Populate the other fields with priority set to 1 and Save	1. Validate the presence of modified Tabs header 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 added tab in first position on the user page

Solution:

Edited existing java file "**TabPage.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 TabPage {

    WebDriver driver;

    WebDriverWait wait;


    public TabPage(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 Tabs

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

        WebElement tabsEle;
```

//WebElement for icon Add New

```
@FindBy(xpath="//a[@title='Add New']")
```

```
WebElement addnewEle;
```

//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 field Tab Name

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

```
WebElement tabnameEle;
```

//WebElement for field Sort

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

```
WebElement sortEle;
```

```
//WebElement for field Status
```

```
@FindBy(xpath="//select[@name='status']")
```

```
WebElement statusddEle;
```

```
//WebElement for icon Save
```

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

```
WebElement saveEle;
```

```
//WebElement for title of panel - Add Tab
```

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

```
WebElement headerEle;
```

```
//WebElement for success message after saving a new 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 Tabs under Catalog on Admin page
```

```
public void clickOnLinkTabs()
```

```
{
```

```
    tabsEle.click();
```

```
    }

//This is a method to click on icon Add New on Tabs List of Admin page

    public void clickOnIconAddNew()

    {

        addnewEle.click();

    }

//This is a method to enter value on field Tab Name using properties file

    public void enterTabName(String tabname)

    {

        tabnameEle.sendKeys(tabname);

    }

//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 perform a Search

    public void performSearch(String tabname)
```

```
        {  
            searchEle.sendKeys(tabname);  
            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 Tab Name  
    public void clearTabName()  
    {  
        tabnameEle.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 Save

        public void clickOnIconSave()
        {
            saveEle.click();
        }

//This is a method to validate the header of Add Tab page

        public String getHeaderMessage() throws IOException
        {
            String file="./TestData/magentodata.properties";

            PropertiesFileHandler propFileHandler = new PropertiesFileHandler();

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

            String expectedtabheadermsg = data.get("expectedtabheader");

            String actualtabheadermsg = headerEle.getText();

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

            System.out.println("The header of Add Tab page: "
+actualtabheadermsg);

```

```

        Assert.assertEquals(expectedtabheadermsg, actualtabheadermsg);

        return(actualtabheadermsg);
    }

//This is a method to validate the header of Edit Tab page

    public String getEditHeaderMessage() throws IOException
    {
        String actualtabheadermsg = headerEle.getText();

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

        System.out.println("The header of Edit Tab page: "
+actualtabheadermsg);

        Assert.assertEquals("Edit Tab", actualtabheadermsg);

        return(actualtabheadermsg);
    }

//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 actualmissingtabmessage = (String) js.executeScript("return
document.getElementsByName('name')[0].validationMessage;");

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

        System.out.println("The error message on mising tab name: "
+actualmissingtabmessage);

        Assert.assertEquals(expectedmissingtabmessage,
actualmissingtabmessage);

        return(actualmissingtabmessage);
    }

    //This is a method to validate the error message on missing mandatory field - Sort Order
    public String validationOnMisingSortField() 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 actualmissingmessage = (String) js.executeScript("return
document.getElementsByName('sort')[0].validationMessage;");

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

        System.out.println("The error message on mising tab name: "
+actualmissingmessage);

```

```

        Assert.assertEquals(expectedmissingtabmessage,
actualmissingmessage);

        return(actualmissingmessage);

    }

    //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 expectedtabmessage = data.get("expectedtabmessage");

        String actualtabmessage = successmsgEle.getText();

        if (actualtabmessage.contains(expectedtabmessage))

        {

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

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

            Assert.assertTrue(true);

        }

        else

        {

            Assert.fail();

        }

    }

```

```

        return(actualtabmessage);
    }

    //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 tab!"))
        {
            System.out.println("The third validation on success message
after saving 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("tabname"))) {

            System.out.println("The validation on presence of record in the
Admin panel table is confirmed!");

            Assert.assertTrue(true);

        }

        else {

            System.out.println("The tab name is not added to this list");

            Assert.fail();

        }

    }

    //This is a method to validate that record is inserted 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("edittabname"))) {

            System.out.println("The validation on presence of record in the
Admin panel table is confirmed!");

```

```
        Assert.assertTrue(true);
    }
    else {
        System.out.println("The tab name is not added to this list");
        Assert.fail();
    }
}
}
```

Edited existing java file “**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 verifyadddtab() 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 verifiedittab() 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();
    }
}
}
```

Edited existing java file “**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.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 Day12Testcase() 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 tabname = data.get("tabname");
```

```

String sortorder = data.get("sortorder");
String edittabname = data.get("edittabname");

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(tab_id) from as_tabs");
System.out.println("The total count of records in DB for Tabs before
creating record: " +beforecount);

AdminPage apage=new AdminPage(driver, wait);
TabPage tp = new TabPage(driver, wait);
tp.clickOnLinkCatalog();
tp.clickOnLinkTabs();
tp.performSearch(tabname);
tp.clickOnAction();
tp.clickOnEdit();
tp.clearSortOrder();
tp.clearStatus();
tp.clickOnIconSave();
tp.getEditHeaderMessage();
tp.clickOnIconSave();
tp.validationOnMissingSortField();
Thread.sleep(1000);
tp.clearTabName();
tp.enterTabName(edittabname);
tp.enterSortOrder(sortorder);
tp.selectStatus();
tp.clickOnIconSave();
Thread.sleep(1000);
tp.validationOnEditSuccessMessage();
tp.validationOnEditRecord();

String editedname = db.singleDataQuery("SELECT name from as_tabs where
sort=1");

System.out.println("The edited tab on DB as: " +editedname);
Assert.assertEquals(edittabname, editedname);

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

apage.clickOnLogout();

```

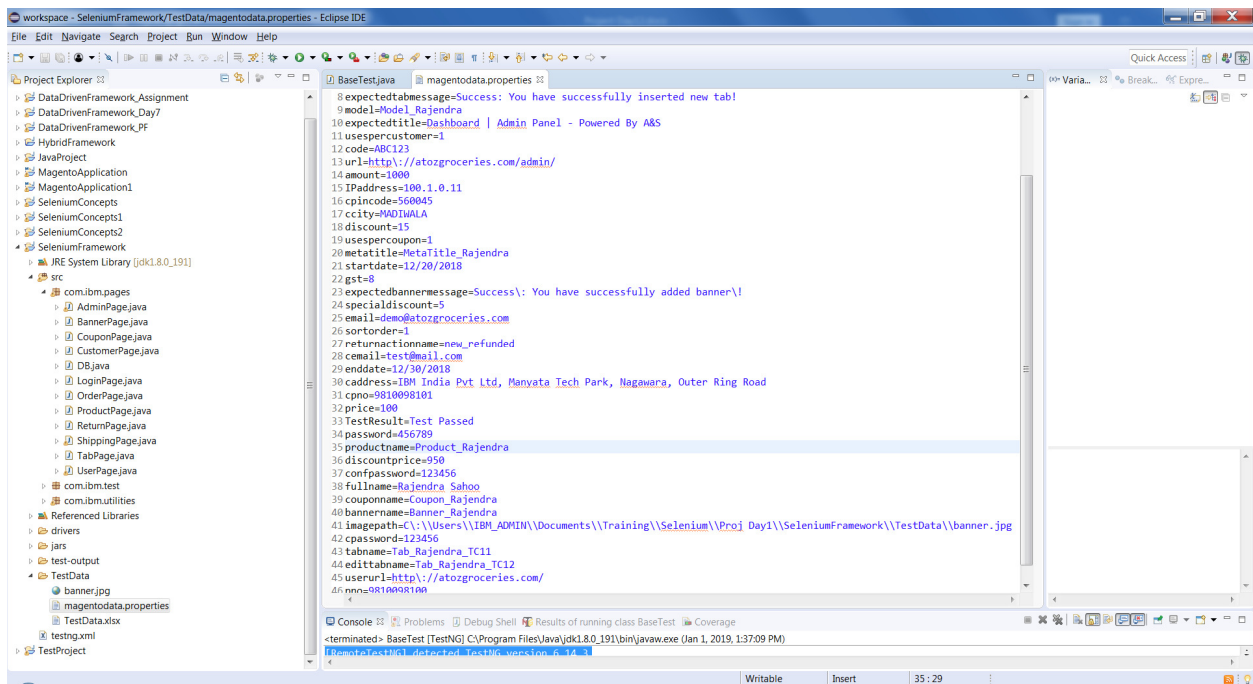
```

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

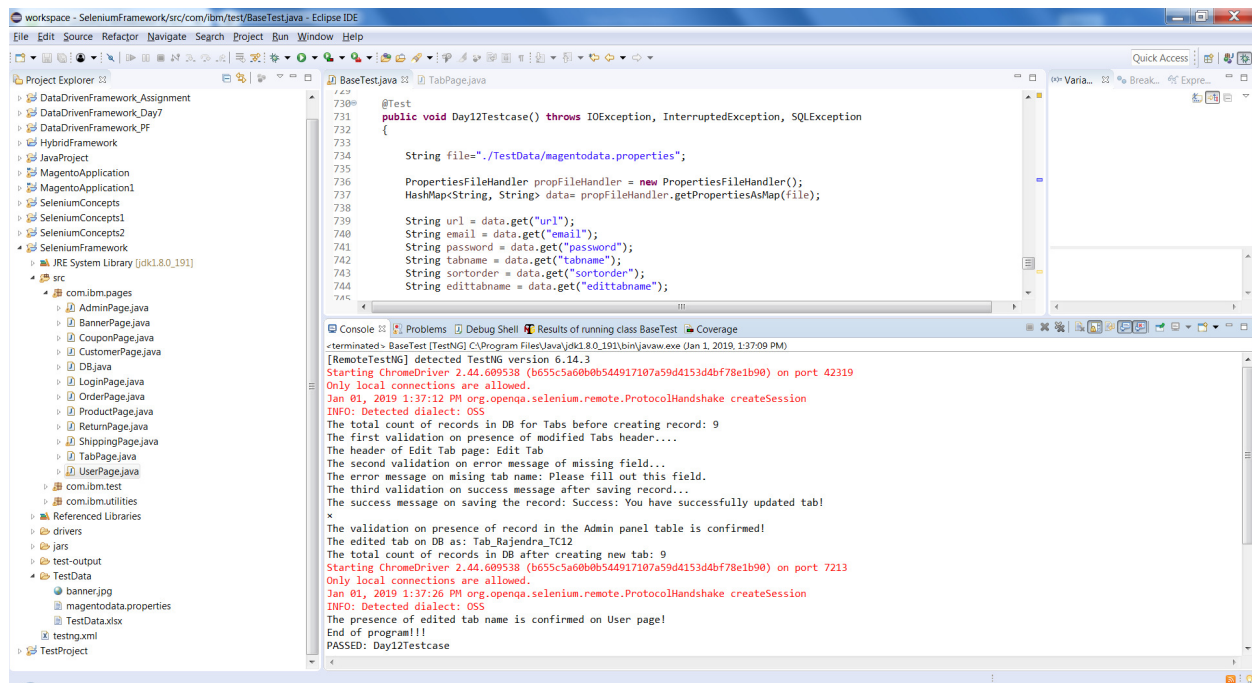
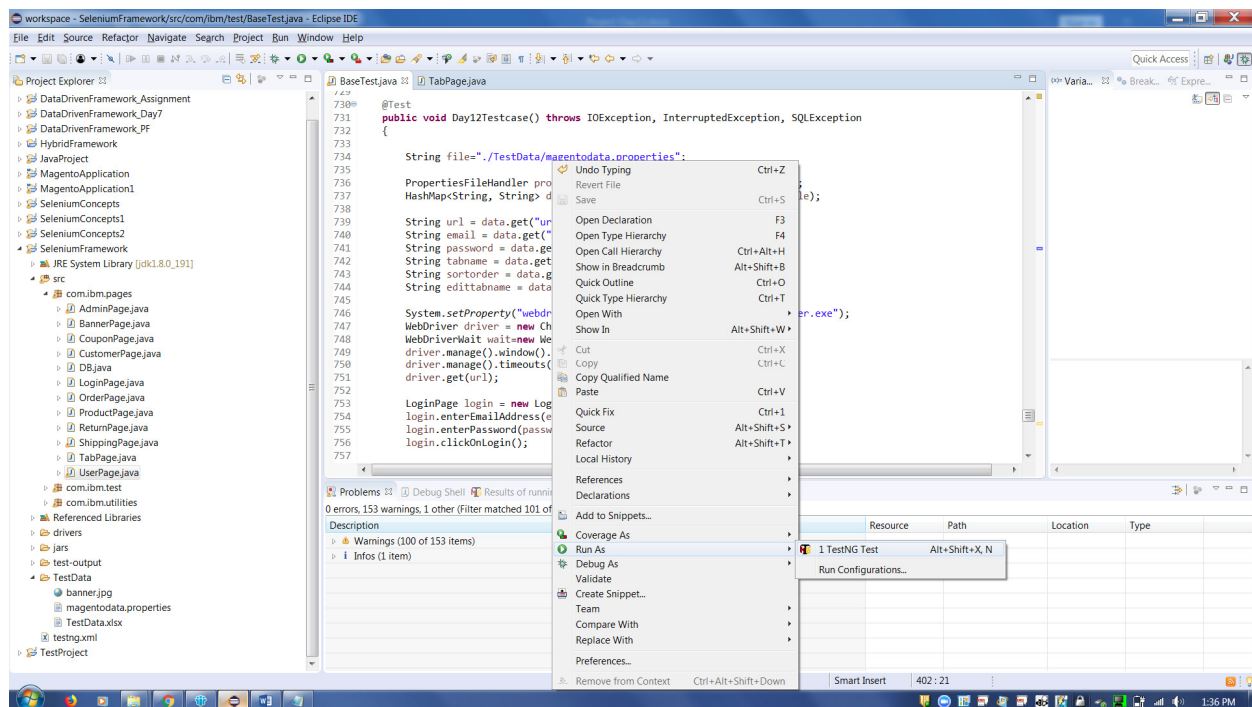
        driver.quit();
        System.out.println("End of program!!!");
    }
}

```

Edited existing properties file “magentodata.properties”



Output screenshot:



Eclipse console:

[RemoteTestNG] detected TestNG version 6.14.3

Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port 42319

Only local connections are allowed.

Jan 01, 2019 1:37:12 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: OSS

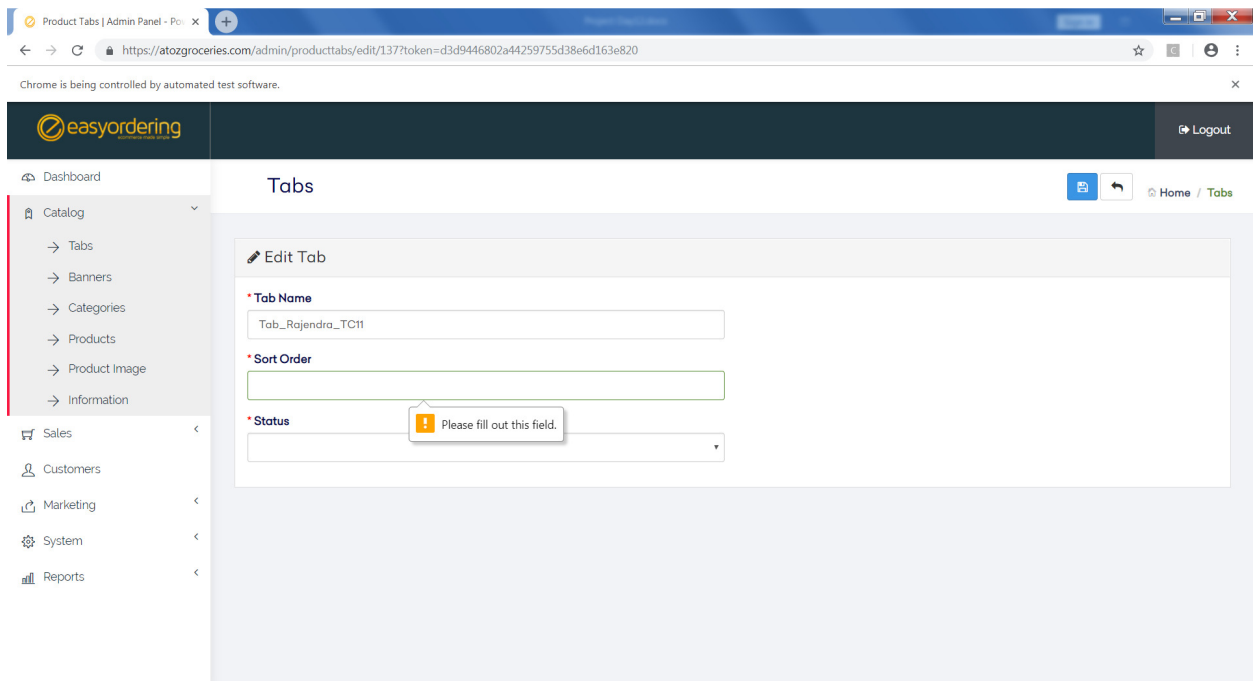
The total count of records in DB for Tabs before creating record: 9
 The first validation on presence of modified Tabs header....
 The header of Edit Tab page: Edit Tab
 The second validation on error message of missing field...
 The error message on missing tab name: Please fill out this field.
 The third validation on success message after saving record...
 The success message on saving the record: Success: You have successfully updated tab!
 x
 The validation on presence of record in the Admin panel table is confirmed!
 The edited tab on DB as: Tab_Rajendra_TC12
 The total count of records in DB after creating new tab: 9
 Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port 7213
 Only local connections are allowed.
 Jan 01, 2019 1:37:26 PM org.openqa.selenium.remote.ProtocolHandshake createSession
 INFO: Detected dialect: OSS
 The presence of edited tab name is confirmed on User page!
 End of program!!!
 PASSED: Day12Testcase

```

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

```

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



Product Tabs | Admin Panel - Po... x

https://atozgroceries.com/admin/producttabs/edit/137?token=d3d9446802a44259755d38e6d163e820

Chrome is being controlled by automated test software.

easyordering [Logout](#)

Dashboard

Catalog

- Tabs
- Banners
- Categories
- Products
- Product Image
- Information

Sales

Customers

Marketing

System

Reports

Tabs

[Home](#) / [Tabs](#)

Edit Tab

* **Tab Name**

Tab_Rajendra_TC12

* **Sort Order**

1

* **Status**

Enabled

Product Tabs | Admin Panel - Po... x

https://atozgroceries.com/admin/producttabs/tab?token=e4da3b7fbce2345d7772b0674a318d5

Chrome is being controlled by automated test software.

easyordering [Logout](#)

Dashboard

Catalog

- Tabs
- Banners
- Categories
- Products
- Product Image
- Information

Sales

Customers

Marketing

System

Reports

Tabs

[Home](#) / [Tabs](#)

Success: You have successfully updated tab!

Tab List

Show 10 entries

[Copy](#) [CSV](#) [Excel](#) [PDF](#) [Print](#)

Search:

#	Tab Name	Sort Order	Status	Action
1	MyTabNew	0	Enabled	Action
2	Tab_Rajendra_TC12	1	Enabled	Action
3	TestMyTab	12	Enabled	Action
4	Test01	111	Enabled	Action
5	Test02	112	Enabled	Action
6	Test03	113	Enabled	Action

