

PFA day 11 test case:

Description	Pre-condition	Test Steps	Validation
Add 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 Tabs header 2. Validate the error message 3.1. Validate the success message 3.2. Validate the presence of record in the admin panel table 3.3 Validate the presence of record in the DB table 3.4 Validate the presence of newly added tab in first position on the user page

Below is the java file of “**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.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 Day11Testcase() 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");

        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 expectedcount = db.countQuery("SELECT count(tab_id) from as_tabs");
        System.out.println("The total count of records in DB for Tabs before
creating new record: " +expectedcount);

        AdminPage apage=new AdminPage(driver, wait);
        TabPage tp = new TabPage(driver, wait);
        tp.clickOnLinkCatalog();
        tp.clickOnLinkTabs();
        tp.clickOnIconAddNew();
        tp.getHeaderMessage();
        tp.clickOnIconSave();
        tp.validationOnMissingField();
        Thread.sleep(2000);
        tp.enterTabName(tabname);
        tp.enterSortOrder(sortorder);
        tp.selectStatus();
        tp.clickOnIconSave();
        Thread.sleep(2000);
        tp.validationOnSuccessMessage();
        tp.validationOnRecord();

        String actualname = db.singleDataQuery("SELECT name from as_tabs where
sort=1");
        System.out.println("The new tab added on DB as: " +actualname);
        Assert.assertEquals(tabname, actualname);

        int actualcount = db.countQuery("SELECT count(tab_id) from as_tabs");
        System.out.println("The total count of records in DB after creating new
tab: " +actualcount);
        Assert.assertEquals(expectedcount+1, actualcount);

        apage.clickOnLogOut();

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

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

```

Below is the java file of “**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 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[@class='alert alert-success alert-
dismissible']/text()[1]")

        @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 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 error message on missing mandatory field

    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 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().trim().replace("x",
""");

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

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

        //Assert.assertEquals(expectedtabmessage, actualtabmessage);
        return(actualtabmessage);
    }

//This is a method to validate that record is inserted into the list

    public void validationOnRecord() 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();  
        }  
    }  
}
```

Below is the java file of “**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.FindingBy;

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 verifytab() 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 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();
    }
}
}
}

```

Below is the java of “**DBUtil.java**”

```

package com.ibm.utilities;

import java.sql.Connection;

```

```
import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;


public class DBUtil {


    public int countQuery(String query) throws SQLException

    {

        int count=0;

        Connection con =

DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atozgroceries",

"foodsonfinger_atoz", "welcome@123");

        Statement s = con.createStatement();

        ResultSet rs = s.executeQuery(query);

        if(rs.next())

        {

            count = rs.getInt(1);

        }

        return count;

    }


    public String singleDataQuery(String query) throws SQLException

    {

        String text=null;
```

```

        Connection con =
DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atozgroceries",
"foodsonfinger_atoz", "welcome@123");

        Statement s = con.createStatement();

        ResultSet rs = s.executeQuery(query);

        if(rs.next())
        {
            text = rs.getString(1);
        }

        return text;
    }

    public static Object[] lineDataQuery(String query,int[] cols) throws SQLException
    {

        Object[] data = new Object[cols.length];

        Connection con =
DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atozgroceries",
"foodsonfinger_atoz", "welcome@123");

        Statement s = con.createStatement();

        ResultSet rs = s.executeQuery(query);

        int i=0;

        if(rs.next())
        {
            for(int col:cols)
            {
                data[i] = rs.getObject(col);

                i++;
            }
        }
    }

```



```
        }  
    }  
    return data;  
}  
  
}
```

Below is the java file “**PropertiesFileHandler.java**”

```
package com.ibm.utilities;  
  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.util.HashMap;  
import java.util.Properties;  
import java.util.Set;  
  
public class PropertiesFileHandler {  
  
    public String getValue(String file, String key) throws IOException {  
  
        FileInputStream fileIn = new FileInputStream(file);  
  
        Properties prop = new Properties();  
  
        prop.load(fileIn);  
  
        String value = null;  
  
        if (prop.containsKey(key)) {
```

```

        value = prop.getProperty(key);
    }
    prop.clear();
    return value;
}

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;
}

public void setKeyAndValue(String file,String key,String value) throws IOException
{
    FileInputStream fileIn = new FileInputStream(file);
    Properties prop = new Properties();

```

```
prop.load(fileIn);

prop.setProperty(key, value);

FileOutputStream fOut=new FileOutputStream(file);
prop.store(fOut, "Test Result");
fOut.close();
fileIn.close();
}

public void setKeysAndValues(String file,HashMap<String, String> map) throws IOException
{
    FileInputStream fileIn = new FileInputStream(file);
    Properties prop = new Properties();
    prop.load(fileIn);

    Set<String> keys = map.keySet();
    for (String key : keys) {
        prop.setProperty(key, map.get(key));
    }

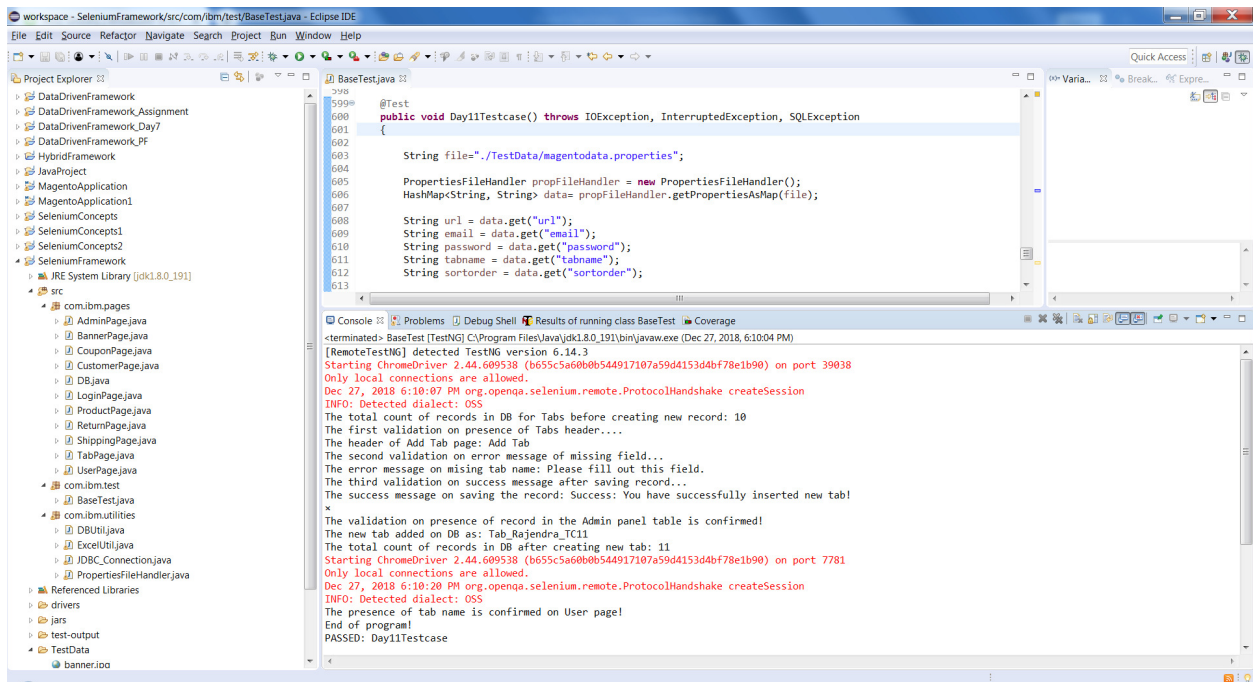
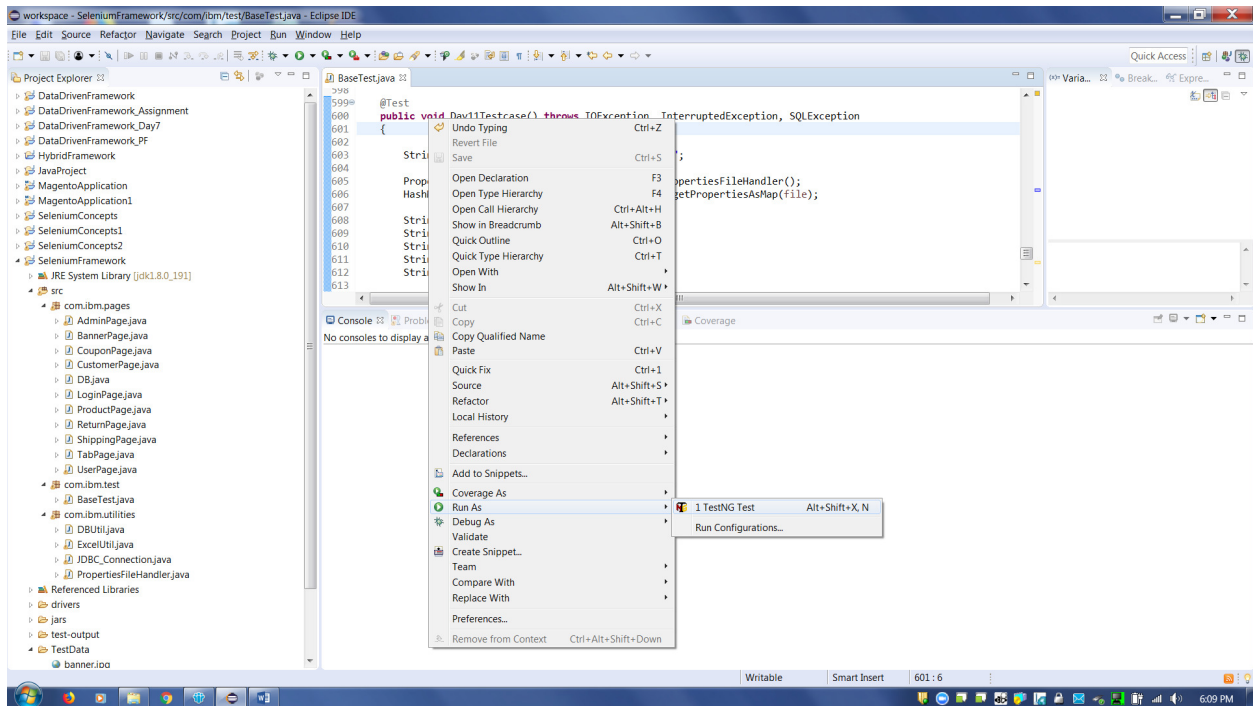
    FileOutputStream fOut=new FileOutputStream(file);
    prop.store(fOut, "Test Result");
    fOut.close();
    fileIn.close();
}
```

```
}  
  
}
```

Below is the Properties file “**magentodata.properties**”

```
expectedcouponmessage=Success\: You have successfully added new coupon\  
expectedpincode=Please fill out this field.  
expectedtabheader=Add Tab  
expectedmissingtabmessage=Please fill out this field.  
expectedtabmessage=Success: You have successfully inserted new tab!  
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  
userurl=http\://atozgroceries.com/  
pno=9810098100
```

Screen shot



Eclipse console:

[RemoteTestNG] detected TestNG version 6.14.3

Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port 39038

Only local connections are allowed.

```
Dec 27, 2018 6:10:07 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
The total count of records in DB for Tabs before creating new record: 10
The first validation on presence of Tabs header....
The header of Add Tab page: Add 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 inserted new
tab!
x
The validation on presence of record in the Admin panel table is confirmed!
The new tab added on DB as: Tab_Rajendra_TC11
The total count of records in DB after creating new tab: 11
Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port
7781
Only local connections are allowed.
Dec 27, 2018 6:10:20 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
The presence of tab name is confirmed on User page!
End of program!
PASSED: Day11Testcase
```

```
=====
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/tab?token=a87ff679a2f3e71d9181a67b7542122c

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 inserted new tab!

Tab List

Show 10 entries

Copy CSV Excel PDF Print

Search:

#	Tab Name	Sort Order	Status	Action
1	shobha	0	Enabled	Action
2	Tab_Rajendra_TC11	1	Enabled	Action
3	TestmyTab2	2	Enabled	Action
4	Tab_Rajendra	2	Enabled	Action
5	TestMyTab	12	Enabled	Action
6	Test01	111	Enabled	Action

AtoZ Grocery | Wholesaler in Bar... x

https://atozgroceries.com

Chrome is being controlled by automated test software.

Welcome to ANY Grocery Login | SignUp

+91 9966996600 acfgd@gmail.com

AtoZ Groceries Search for products...

GET IT ON Google Play

Cart

SHOP BY CATEGORY

NEW ARRIVAL ABOUT US CONTACT US

DYNAMIC TRADERS

Wholesale Dealers in Rice, Sugar, Salt, Soft Drinks, Pulse

SHOBHA PRODUCTS
Shop now »

TAB_RAJENDRA_TC11 PRODUCTS
Shop now »

TESTMYTAB2 PRODUCTS
Shop now »

FAST DELIVERY
100% fast delivery for your products

QUALITY
you can trust your order 100% genuine products

ON TIME GUARANTEE
On time guarantee for your order

SUPPORT 24/7
We are here 24/7 to help you