1.Problem statement:

Day 11 test case,

| Description | Test Steps | Validation |
|---|---|---|
| Add a banner | 1. Navigate onto admin panel -> Catalog -> Banners<br>2. Populate Banner name and Save<br>3. Attach a file and Save | 1. Validate the presence of Banners header<br>2. Validate the error message<br>3.1. Validate the success message<br>3.2. Validate the presence of record in the admin panel table<br>3.3 Validate the presence of record in the DB table |

2. code with comments:

BaseTest.java:

```
package com.ibm.test;


import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.lang.*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.HashMap;

import java.util.Properties;

import java.util.concurrent.TimeUnit;


import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

import com.ibm.pages.UserPage;
import com.ibm.utilities.DBUtil;
import com.ibm.utilities.ExcelReader;
import com.ibm.utilities.ExcelUtil;
import com.ibm.utilities.PropertiesFileHandler;

public class BaseTest extends ExcelReader{
        WebDriverWait wait;
        WebDriver driver;
   @Test()

   public void testcase11() throws InterruptedException, IOException, SQLException{

        FileInputStream file = new FileInputStream("./TestData/data.properties");
        Properties prop = new Properties();
        prop.load(file);
        String url = prop.getProperty("url");
```

```
String username = prop.getProperty("user");
String password = prop.getProperty("password");
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        driver = new ChromeDriver();
        wait = new WebDriverWait(driver, 60);



        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        Login login = new Login(driver, wait);
        driver.get(url);

        login.enterEmailAddress(username);
        login.enterPassword(password);
        login.clickOnLogin();

        WebElement catalogEle=driver.findElement(By.linkText("Catalog"));
        catalogEle.click();

        WebElement bannersEle=driver.findElement(By.linkText("Banners"));
        bannersEle.click();

        int expCount= DBUtil.countQuery("SELECT count(name) from as_banner");
        System.out.println("Count beofre adding banner : "+expCount);
        Thread.sleep(3000);

        WebElement newBannerEle=driver.findElement(By.cssSelector("i.fa.fa-plus"));
        newBannerEle.click();
```

```
WebElement bannerName=driver.findElement(By.name("name"));

bannerName.sendKeys("Test_Banner_Test");


WebElement codeEle=driver.findElement(By.xpath("//*[@id=\"page-
wrapper\"]/div/form/div[2]/div/div/div[2]/div[4]/span"));

codeEle.click();


String winHandleBefore = driver.getWindowHandle();


WebElement frame=driver.switchTo().activeElement();


frame.sendKeys("C:\\Users\\Public\\Pictures\\Sample Pictures\\Chrysanthemum.jpg");

//codeEle.sendKeys("C:/Users/Public/Pictures/Sample Pictures/Chrysanthemum.jpg");

Thread.sleep(3000);

driver.switchTo().window(winHandleBefore);


WebElement saveEle=driver.findElement(By.cssSelector("i.fa.fa-save"));

saveEle.click();

Thread.sleep(3000);


String exp="Test_Banner_Test";

String act= DBUtil.singleDataQuery("SELECT name from as_banner where
name=\"Test_Banner_Test\"");


//validation of banner header

Assert.assertEquals(act,exp);

Reporter.log("Assertion on banner added present in database");


//validation in database
```

```
                int actCount=DBUtil.countQuery("SELECT count(name) from as_banner");
                System.out.println("COunt after adding banner"+actCount);

                Assert.assertEquals(actCount,expCount+1);
                Reporter.log("Assertion on banner added present in database");

                Thread.sleep(3000);
    }

}
```

Login.java:

```
package com.ibm.test;

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

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.Assert;
import org.testng.annotations.BeforeSuite;
```

```
import org.testng.annotations.Test;

import com.ibm.pages.AdminPage;
import com.ibm.utilities.PropertiesFileHandler;

public class Login extends BaseTest{

	@FindBy(name="email")
	WebElement emailEle;

	@FindBy(name="pword")
	WebElement passEle;

	@FindBy(xpath="/html/body/div/div/div/div[2]/form/button")
	WebElement loginEle;
	WebDriverWait wait;
	WebDriver driver;

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

	public void enterEmailAddress(String user)
	{
		emailEle.sendKeys(user);
	}
	public void enterPassword(String password)
```

```
	{
		passEle.sendKeys(password);
	}
	public void clickOnLogin()
	{
		loginEle.click();
	}

}
```

ExceUtil.java:

```
package com.ibm.utilities;
//We will understand the construction of this page
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import com.ibm.test.BaseTest;

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

public class ExcelUtil{
   public static Object[][] DataTable(String Workbook, String Sheet) throws FileNotFoundException,
IOException{
      XSSFWorkbook wb = new XSSFWorkbook(new FileInputStream(Workbook));
      XSSFSheet sh = wb.getSheet(Sheet);
      Object[][] x = new Object[sh.getPhysicalNumberOfRows()-
1][sh.getRow(0).getPhysicalNumberOfCells()];
      for(int i=1;i<sh.getPhysicalNumberOfRows();i++)
```

```
    {
        for(int j=0;j<sh.getRow(0).getPhysicalNumberOfCells();j++)
        {
            x[i-1][j] = sh.getRow(i).getCell(j).getStringCellValue();
        }
    }
    return x;
  }

  public static Object[][] DataTable(String Workbook, String Sheet, int rows) throws
FileNotFoundException, IOException {
    XSSFWorkbook wb = new XSSFWorkbook(new FileInputStream(Workbook));
    XSSFSheet sh = wb.getSheet(Sheet);
    Object[][] x = new Object[rows][sh.getRow(0).getPhysicalNumberOfCells()];
    for(int i=1;i<rows+1;i++)
    {
        for(int j=0;j<sh.getRow(0).getPhysicalNumberOfCells();j++)
        {
            x[i-1][j] = sh.getRow(i).getCell(j).getStringCellValue();
        }
    }
    return x;
  }

  public static Object[][] DataTable(String Workbook, String Sheet, int rows, int col) throws
FileNotFoundException, IOException{
    XSSFWorkbook wb = new XSSFWorkbook(new FileInputStream(Workbook));
    XSSFSheet sh = wb.getSheet(Sheet);
    Object[][] x = new Object[rows][col];
```

```
    for(int i=1;i<rows+1;i++)
    {
       for(int j=0;j<col;j++)
       {
          x[i-1][j] = sh.getRow(i).getCell(j).getStringCellValue();
       }
    }
    return x;
  }
}
```

Data.properties file

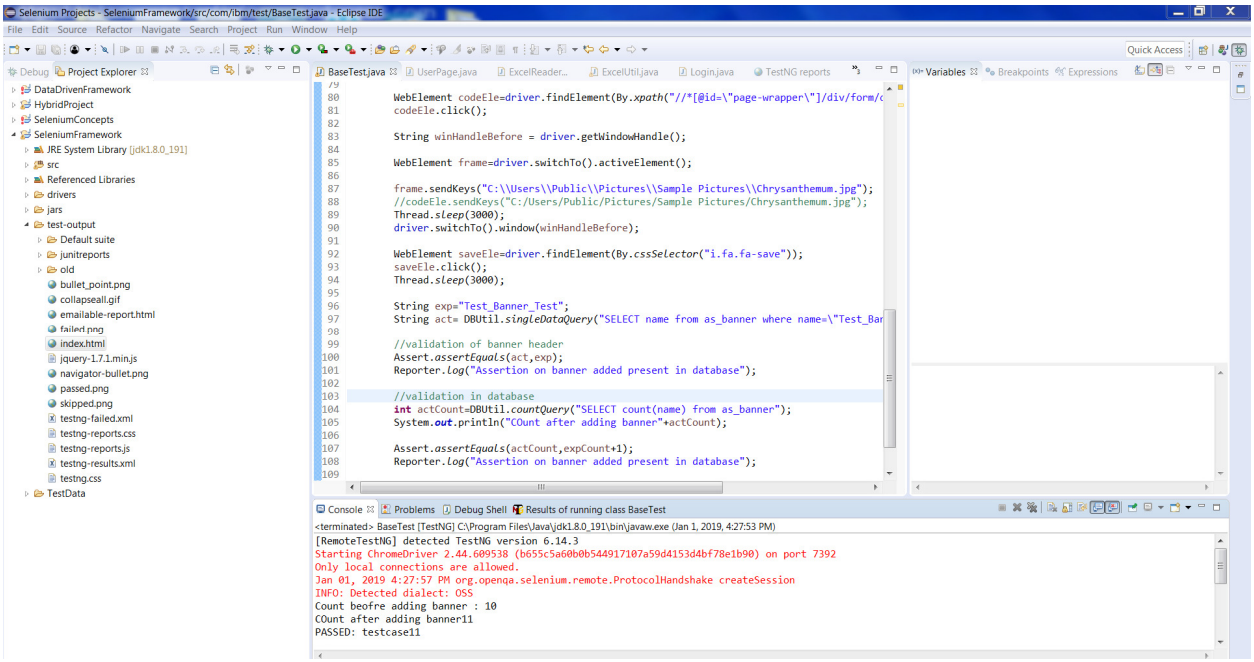
```
url=https://atozgroceries.com/admin
user=demo@atozgroceries.com
password=456789
url1=https://atozgroceries.com
user1=7675058941
password1=456789
```

3. Explanation of the code: DBUtil class has functions for DB validations which is imported in basetest.java prg. Getting login details from data.properties file. ExcelUtil class can also be used for creating banner header but since only one value is needed in this scenario used only direct send keys to focus on DB validations.

4. Result flow in detail:

DBUtil class has functions for DB validations which is imported in basetest.java prg. Getting login details from data.properties file. ExcelUtil class can also be used for creating banner header but since only one value is needed in this scenario used only direct send keys to focus on DB validations.

5. Output screenshot:


Selenium Projects - SeleniumFramework/src/com/ibm/test/BaseTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Debug
Project Explorer
DataDrivenFramework
HybridProject
SeleniumConcepts
SeleniumFramework
JRE System Library [jdk1.8.0_191]
src
Referenced Libraries
drivers
jars
test-output
Default suite
junitreports
old
bullet_point.png
collapseall.gif
emailable-report.html
failed.png
index.html
jquery-1.7.1.min.js
navigator-bullet.png
passed.png
skipped.png
testng-failed.xml
testng-reports.css
testng-reports.js
testng-results.xml
testng.css
TestData
BaseTest.java
UserPage.java
ExcelReader...
ExcelUtil.java
Login.java
TestNG reports
Variables
Breakpoints
Expressions
WebElement codeEle=driver.findElement(By.xpath("//*[@id=\"page-wrapper\"]/div/form/c
codeEle.click();
String winHandleBefore = driver.getWindowHandle();
WebElement frame=driver.switchTo().activeElement();
frame.sendKeys("C:\\Users\\Public\\Pictures\\Sample Pictures\\Chrysanthemum.jpg");
//codeEle.sendKeys("C:/Users/Public/Pictures/Sample Pictures/Chrysanthemum.jpg");
Thread.sleep(3000);
driver.switchTo().window(winHandleBefore);
WebElement saveEle=driver.findElement(By.cssSelector("i.fa.fa-save"));
saveEle.click();
Thread.sleep(3000);
String exp="Test_Banner_Test";
String act= DBUtil.singleDataQuery("SELECT name from as_banner where name=\"Test_Bar
//validation of banner header
Assert.assertEquals(act,exp);
Reporter.log("Assertion on banner added present in database");
//validation in database
int actCount=DBUtil.countQuery("SELECT count(name) from as_banner");
System.out.println("COunt after adding banner"+actCount);
Assert.assertEquals(actCount,expCount+1);
Reporter.log("Assertion on banner added present in database");
Console
Problems
Debug Shell
Results of running class BaseTest
<terminated> BaseTest [TestNG] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (Jan 1, 2019, 4:27:53 PM)
[RemoteTestNG] detected TestNG version 6.14.3
Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port 7392
Only local connections are allowed.
Jan 01, 2019 4:27:57 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
Count beofre adding banner : 10
COunt after adding banner11
PASSED: testcase11