Day 10 test case_Programs

Base test:

```
@Test(priority=10)
    public void Testcase10() throws IOException, InterruptedException,
SQLException {
              try {

              String url = data.get("url");
                     String userName = data.get("user");
                     String passWord = data.get("password");
                     String customerName= data.get("customername");
                     String noOfRecords= data.get("noofrecordsfound");


                     driver.get(url);

                     LoginPage login = new LoginPage(driver, wait); // Login to
Admin Page
                     login.enterEmailId(userName);
                     login.enterPassWord(passWord);
                     login.clickOnLogin();

                     AdminPage admin = new AdminPage(driver, wait); // To Select
System Tab
              admin.clickOnCustomers();
              admin.customerNameSearch(customerName);
                     String expCustomername =admin.getTextToVerifyCustomerName();
                  System.out.println(expCustomername);

                     //To validate Presence of a Customer Record on Database

                   String actualCustomerName =  DBUtil.singleDataQuery("SELECT name
FROM as_customer WHERE name='shobha' ");
                   System.out.println(actualCustomerName);
                   Assert.assertEquals(actualCustomerName.trim(),expCustomername);
                   Reporter.log("Assertion on customer record  present in
Database");
                   System.out.println("Customer Name = shobha is  available");

              // Navigate to admin panel and delete a customer record
                admin.clickOnCustomers();
                admin.customerNameSearch(customerName);
                admin.clickOnCustomersAction();
                admin.clickOnCustomerDelete();
               Thread.sleep(90000); // without Thread, its not clicking DleteIt
button
               admin.clickOnDeleteIt();

              //To Verify the deleted customer under the Customer  list
              Assert.assertTrue(driver.getPageSource().contains("You have
successfully deleted data!"));
              Reporter.log("Assertion on Deleted Customer not present in Tab
List");
              System.out.println("message: You have successfully deleted data!");


              admin.customerNameSearch(customerName);
              String expDeletedCustomer = admin.getTextToVerifyDeletedCustomer();
```

```
            System.out.println(expDeletedCustomer);
            Assert.assertEquals(noOfRecords, expDeletedCustomer);
            Reporter.log("Assertion on Deleted Customer");

            //To validate Deleted Customer is not present on Database

          int countOfActualCustomerName =  DBUtil.countQuery("SELECT count(*)
FROM as_customer WHERE name='shobha' "); //use count query
          System.out.println(countOfActualCustomerName);
          Assert.assertEquals(0,countOfActualCustomerName);
          Reporter.log("Assertion on Deleted Customer name not present in
Database");
          System.out.println("Customer Name = shobha is not available");

          //To validate Customer Record count is Decreased by 1

          int countOFTotalCustomerRecord= DBUtil.countQuery("SELECT count(*) FROM
as_customer");
          System.out.println(countOFTotalCustomerRecord);
          Assert.assertEquals(31,countOFTotalCustomerRecord);
          Reporter.log("Assertion on Customer Record count in Database");
          System.out.println("Customer name= shobha not present in Database");



            } catch (Exception e) { // TODO: handle exception
                    System.out.println(e.getMessage());
                    Reporter.log("Negataive Credentials Failed"); //screenshot
                    TakesScreenshot ts=(TakesScreenshot) driver;
                    File file= ts.getScreenshotAs(OutputType.FILE);
                    Date date =new Date(); String
currentDate=date.toString().replace(":", "-");
                    FileUtils.copyFile(file, new

File("./screenshots/Day_10_Test_case_Error_"+currentDate+".jpg"));
                    System.out.println(e.getMessage());
                    Assert.fail();
                    }

    }
```

---

url=https://atozgroceries.com/admin
user=demo@atozgroceries.com
password=456789
customername=shobha
noofrecordsfound=No matching records found

Admin page
---
package com.ibm.pages;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

```
import org.openqa.selenium.remote.server.handler.GetPageSource;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.How;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class AdminPage {


	// To select Catalog

			@FindBy(xpath = "//a[@href='#']")
			WebElement catalogEle;

			//To select Tabs
			@FindBy(xpath="//a[contains(text(),'Tabs')]")
			WebElement tabsEle;

			//To select Action button for a Tab
			@FindBy(xpath="(//button[contains(text(),'Action')])[2]")
			WebElement tabActionEle;

			//To Delete Tab
			@FindBy(linkText ="Delete")
			WebElement tabDeleteEle;

			//To click on Delete It
			//@FindBy(xpath="//button[@class='confirm']")
			@FindBy(xpath="/html/body/div[4]/div[7]/div/button")
			WebElement DeleteItEle;

			//Message for Deleted data in tab List
			@FindBy(xpath="//*[@id='page-wrapper']/div/div[2]")
			WebElement deleteMsgEle;

			//Logout
			@FindBy(xpath="//a[@title='Logout']")
			WebElement logoutEle;


		//Search for Deleted Tab

		@FindBy(xpath="//*[@id='dataTableExample2_filter']/label/input")
		WebElement searchTabEle;

		@FindBy(xpath="//*[@id='dataTableExample2']/tbody/tr/td")
		WebElement deletedTabNotPresentEle;


		//To select Customers
```

```
	@FindBy(xpath = "//*[@id='side-menu']/li[4]/a")
	WebElement customersEle;

	@FindBy(xpath="//*[@id='dataTableExample2_filter']/label/input")
	WebElement customerNamesearchTabEle;


	@FindBy(xpath="//*[@id='dataTableExample2']/tbody/tr/td[2]")
	WebElement presenceCustomerNameEle;

	//To select Action for Customer
	@FindBy(xpath = "//button[contains(text(),'Action')]")
	WebElement customersActionEle;

	//To Delete customer
	@FindBy(linkText ="Delete")
	WebElement customerDeleteEle;

	//To serach for deleted customer
	@FindBy(xpath="//td[text()='No matching records found']")
	WebElement deletedCustomerEle;



WebDriverWait wait;
WebDriver driver;

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

public void clickOnCatalog() {
	catalogEle.click();
}

public void clickOnTabs()
{
	tabsEle.click();
}

public void clickOnTabAction()
{
	tabActionEle.click();
}

public void clickOnTabDelete()
{
	tabDeleteEle.click();
```

```
	}

	public void clickOnDeleteIt()
	{
		DeleteItEle.click();
	}

	public String getTextToVerifyDeletedTabMsg() {

	wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//*[contains(text(),'Yo
u have successfully deleted data!')]")));
		String deletedmsg = deleteMsgEle.getText();
		return deletedmsg ;
	}

	public void clickOnLogout()
	{
		logoutEle.click();
	}

	public String searchForTabName(String deletedTabName)
	{
		searchTabEle.sendKeys(deletedTabName);
		return deletedTabName;
	}

	public String getTextToVerifyDeletedTab() {
		String expectedTabName = deletedTabNotPresentEle.getText();
		return expectedTabName ;
	}


	public void clickOnCustomers()
	{
		customersEle.click();
	}

	public String customerNameSearch (String customername)
	{
		customerNamesearchTabEle.sendKeys(customername);
		return customername;

	}

	public String getTextToVerifyCustomerName() {
		String expectedCustomerName = presenceCustomerNameEle.getText().trim();
		return expectedCustomerName ;
	}

	public void clickOnCustomersAction()
	{
```

```
		customersActionEle.click();
	}

	public void clickOnCustomerDelete()
	{
		customerDeleteEle.click();
	}

	public String getTextToVerifyDeletedCustomer()
	{
				String expDeletedCustomer = deletedCustomerEle.getText();
		return expDeletedCustomer ;
	}
}
```

Loginpage:

```
package com.ibm.pages;

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.annotations.Test;

public class LoginPage {

	@FindBy(xpath="//*[@name='email']")
	WebElement emailEle;

	@FindBy(xpath="//*[@name='pword']")
	WebElement passwordEle;

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

	WebDriverWait wait;
	WebDriver driver;


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

	public void  enterEmailId(String user)
	{
```

```
		emailEle.sendKeys(user);

	}

	public void enterPassWord(String password)
	{
		passwordEle.sendKeys(password);

	}

	public void clickOnLogin()
	{
	  loginEle.click();
	}

	public void enterUserEmail(String usermailid)
	{
		emailEle.sendKeys(usermailid);
	}

	public void enterUserPassword(String userpassword)
	{
		passwordEle.sendKeys(userpassword);
	}
}
```

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  static 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;
	}


	public static 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;
	}

}
```

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> BaseMap = 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) {
		BaseMap.put(key.toString(), prop.getProperty(key.toString()));
	}
	prop.clear();
	return BaseMap;
}

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();
        }
}
```

Testng2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
    <classes>
      <class name="com.ibm.test.BaseTest">
      <methods>
      <include name=".*10"></include>
      </methods>
        <!-- <methods><include name="TestCase10"></include></methods> -->
      </class>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```
