

1.Problem statement:

Description	Test Steps	Validation
Modify the setting tabs and verify contact us on userpage	1. Login to admin portal 2. Goto System and then settings 3. Change the email address, phone number, address 4. Goto user page. 5. Click on contact us. 5. Verify the below contact us on user page. Note: Search for the modified email address.	1. Validate the presence of modified data 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 record on the user page

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.FindBy;
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 testcase13() 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 systemEle=driver.findElement(By.linkText("System"));
    systemEle.click();


    WebElement settingsEle=driver.findElement(By.linkText("Settings"));
    settingsEle.click();


    Thread.sleep(3000);


    WebElement addressEle=driver.findElement(By.id("address"));
    addressEle.clear();
```

```
addressEle.sendKeys("Change123");
```

```
WebElement emailEle=driver.findElement(By.id("email"));
```

```
emailEle.clear();
```

```
emailEle.sendKeys("test@gmail.com");
```

```
WebElement phoneNoEle=driver.findElement(By.id("phonenumber"));
```

```
phoneNoEle.clear();
```

```
phoneNoEle.sendKeys("1234567890");
```

```
WebElement goToTopEle=driver.findElement(By.cssSelector("span.ti-arrow-up"));
```

```
goToTopEle.click();
```

```
Thread.sleep(3000);
```

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

```
saveEle.click();
```

```
Thread.sleep(2000);
```

```
//Assertion in the admin page
```

```
String successMsg=login.AddWeightClassSuccMsg().trim();
```

```
Assert.assertEquals(successMsg,"Success: You have successfully updated Store!");
```

```
System.out.println("Updated address, Phone number and email is reflected in the  
Admin Page");
```

```
WebElement logOutEle=driver.findElement(By.cssSelector("i.fa.fa-sign-out"));
```

```
logOutEle.click();
```

```
Thread.sleep(3000);
```

```
driver.get("https://atozgroceries.com");  
driver.manage().window().maximize();  
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);  
UserPage user = new UserPage(driver, wait);
```

```
WebElement loginEle=driver.findElement(By.linkText("Login"));  
loginEle.click();
```

```
user.userName("7675058941");  
user.enterPassword("456789");  
user.clickOnLogin();  
Thread.sleep(3000);  
//Assertion in the user page
```

```
Assert.assertTrue(driver.findElement(By.partialLinkText("test@gmail.com")).isDisplayed());  
System.out.println("Modified settings from admin Panel are reflected in User page");
```

```
String exp="1234567890";  
String act= DBUtil.singleDataQuery("SELECT pnum from as_store");
```

```
//validation of database  
Assert.assertEquals(act,exp);  
Reporter.log("Assertion on phone number change present in database");
```

```
}
```

```
}
```

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;
import java.util.ArrayList;
import java.util.List;

public class DBUtil {

    public static String singleDataQuery(String query) throws SQLException {

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

        Statement s=c.createStatement();

        String text=null;

        //SELECT name from as_category where name="Fruits" - if data exists value else null.

        //SELECT tab_id,name FROM as_tabs ORDER BY tab_id DESC

        //SELECT * from as_coupons

        ResultSet rs=s.executeQuery(query);

        while(rs.next()) {

            text=rs.getString(1);//validating name alone

            //System.out.println("database check for added
element:"+rs.getString("name"));

        }

        return text;

        //Assert.assertEquals("Grains", rs.getString("name"));

    }
}

```

```

public static Object[] lineDataQuery(String query, int[] cols) throws SQLException {
    Object[] data=new Object[cols.length];

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

"foodsonfinger_atoz",

"welcome@123");

    Statement s=c.createStatement();
    //SELECT * from as_banner where name="Pavithra"
    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
c=DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atozgroceries","
foodsonfinger_atoz","welcome@123");

    Statement s=c.createStatement();
    //select count(name) from as_coupons

```

```

        ResultSet rs=s.executeQuery(query);

        while(rs.next()) {

            count=rs.getInt(1);

        }

        return count;

        //Assert.assertEquals("Grains", rs.getString("name"));

    }

    /*

    public ArrayList<ArrayList<Object>> tableDataQuery(String query) throws SQLException{

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

        Statement s=c.createStatement();

        ArrayList<ArrayList<Object>> data=new ArrayList<ArrayList<Object>>();

        String[] queries=query.split(",");

        ResultSet rs=s.executeQuery(query);

        ArrayList<Object> temp = null;

        while(rs.next()) {

            //data.add(rs.getString(1), rs.getString(2), rs.getString(3));

            for(int i=0;i<queries.length;i++)

            {

                temp.add(rs.getObject(i));

            }

            data.add(temp);

            temp.clear();

        }

        return data;

        //Assert.assertEquals("Grains", rs.getString("name"));

```



```
    }  
*/  
}
```

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;
```

```
@FindBy(xpath="//*[@class='alert alert-success alert-dismissible']")
```

```
WebElement addWeightSuccMsg;
```

```
WebDriverWait wait;
```

```
WebDriver driver;
```

```
public Login(WebDriver driver,WebDriverWait wait) {
```

```
    PageFactory.initElements(driver, this);
```

```
    this.driver=driver;
```

```
    this.wait=wait;
```

```
}
```

```
public String AddWeightClassSuccMsg()
```

```
{
```

```
String succMsg=addWeightSuccMsg.getText().trim().replace("x", "");
```

```
//System.out.println(succMsg);
```

```
return succMsg;
```

```
}
```

```
public void enterEmailAddress(String user)
```

```

        {
            emailEle.sendKeys(user);
        }
        public void enterPassword(String password)
        {
            passEle.sendKeys(password);
        }
        public void clickOnLogin()
        {
            loginEle.click();
        }
    }
}

UserPage.java

package com.ibm.pages;

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.WebDriverWait;

import com.ibm.test.BaseTest;

public class UserPage extends BaseTest{

    @FindBy(xpath="//*[@id=\"pnum2\"]")
    WebElement userEle;

```

```
@FindBy(id="pword2")
```

```
WebElement passEle;
```

```
@FindBy(id="mem_login")
```

```
WebElement loginEle;
```

```
WebDriverWait wait;
```

```
WebDriver driver;
```

```
public UserPage(WebDriver driver,WebDriverWait wait) {
```

```
    PageFactory.initElements(driver, this);
```

```
    this.driver=driver;
```

```
    this.wait=wait;
```

```
}
```

```
public void userName(String user1)
```

```
{
```

```
    userEle.sendKeys(user1);
```

```
}
```

```
public void enterPassword(String password1)
```

```
{
```

```
    passEle.sendKeys(password1);
```

```
}
```

```
public void clickOnLogin()
```

```
{
```

```
    loginEle.click();
```

```
}
```

```
/* @FindBy(xpath="//*cvdcg")
```

```
WebElement tabNotPresentEle;
```

```

        public boolean getTabNotPresent()
        {
            boolean deletedTab=tabNotPresentEle.isDisplayed();

            return deletedTab;
        }
    }
}

```

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:

Made changes in the settings in Basetest.java and used DBUtil class for database check, Login.java for admin page code and userPage.java for userpage code and used data.properties file for login details for admin and user logins.

4. Result flow in detail:

Made changes in the settings in Basetest.java and used DBUtil class for database check, Login.java for admin page code and userPage.java for userpage code and used data.properties file for login details for admin and user logins.

5. Output screenshot:

