

Day 2 Test Case: Test Cases	Process	Steps To Execute	Expected Results
TC02	Edit in Information tab	1. Login to admin portal 2. Edit anyone Information 3. Fill all data except Information name	Click the save icon and verify it is on the same page. Negative test case validation

```

package com.ibm.test;

import static org.testng.Assert.assertTrue;

import java.io.File;
import java.io.IOException;
import java.util.Date;
import java.util.HashMap;
import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;
import com.ibm.pages.AdminPage;
import com.ibm.pages.LoginPage;
import com.ibm.utilities.ExcelUtil;
import com.ibm.utilities.PropertiesFileHandler;

public class BaseTest {

    // initializing the driver, wait , profilehandler and HashMap
    WebDriver driver;
    WebDriverWait wait;
    PropertiesFileHandler propFileHandler;
    HashMap<String, String> data;

    // beforesuite method is used to declare the statement that are going to
execute
    // before the running of testcase

    @BeforeSuite

```

```

public void preSetForTest() throws IOException {

    String file = "./TestData/data.properties";
    propFileHandler = new PropertiesFileHandler();
    data = propFileHandler.getPropertiesAsMap(file);

}

// beforemethod is used to declare the statement that are going to run at the
// starting of all test cases
@BeforeMethod

public void Initialization() {

    System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    driver = new ChromeDriver();// opening the chrome browser
    wait = new WebDriverWait(driver, 60);// giving explicit wait
    driver.manage().window().maximize();// maximizing the window
    driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS); //
declaring implicit wait

}

// aftermethod is used to run once all the test cases are run

@AfterMethod
public void closeBrowser() {
    driver.quit();// close the browser
}

@Test
public void testcase2() throws InterruptedException, IOException {

    // all the data are declared in properties file
    String url = data.get("url");
    String username = data.get("user");
    String password = data.get("password");
    String expectededitMsg = data.get("expectedmsg");

    // open the url https://atozgroceries.com/admin

    driver.get(url);

    // calling login page method to enter the user name, password and click
on log

    // in button
    LoginPage login = new LoginPage(driver, wait);
    login.enterEmailAddress(username);
    login.enterPassword(password);
    login.clickOnLogin();

    //calling admin page method to perform the edit information action on
information tab

```

```

        AdminPage admin = new AdminPage(driver, wait);
        admin.clickOnCatalog();
        admin.clickOnInformationTab();
        admin.clickOnInformationAction();
        admin.clickOnEditButton();
        admin.clearTextOnMetaText();
        Thread.sleep(10000);
        admin.clickOnSave();

        // calling the method to verify the assertion
        String actualmsg = admin.getTexttoVerifyEditMsg();

        // Assertion to verify that the text Edit information is still visible
        on the screen when the save button is clicked without filling the
        // mandatory field.

        Assert.assertTrue(actualmsg.contains(expectededitMsg), "Assertion on edit
        information page");

        // To take 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/Error_" + currentDate +
        ".png"));
    }
}

```

Admin page :

```

package com.ibm.pages;

import java.util.concurrent.TimeUnit;

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.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class AdminPage {

    WebDriverWait wait;

```

```

WebDriver driver;
JavascriptExecutor js;

@FindBy(xpath="//*[@id=\"side-menu\"]/li[2]/ul/li[6]/a")
WebElement informationEle;

@FindBy(xpath="//*[@id=\"dataTableExample2\"]/tbody/tr[1]/td[3]/div/button")
WebElement informactionEle;

@FindBy(xpath="//*[@title='Edit']")
WebElement editEle;

@FindBy(xpath="//*[@id=\"meta_title\"]")
WebElement metaTextEle;

@FindBy(xpath="//*[@id=\"page-wrapper\"]/div/form/div[1]/button")
WebElement saveEle;

@FindBy(xpath="//*[contains(text(),' Edit Information')]")
WebElement editinfoMsgEle;

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

}

// method to click on information tab
public void clickOnInformationTab() {

    informationEle.click();
}

// method to click Action on information tab
public void clickOnInformationAction() {
    informactionEle.click();
}

// method to click on Edit button
public void clickOnEditButton() {

    editEle.click();
}

// methid to clear the text on meta title field
public void cleartextonMetaText() {

```

```

        metaTextEle.clear();
    }

    // method to click on save
    public void clickOnSave() {

        js.executeScript("arguments[0].click();", saveEle);
    }

    // this string is declared to verify the text we are getting after deleting
the
    // tab

    public String getTextToVerifyDeleteTabMsg() {

        wait.until(ExpectedConditions

            .presenceOfElementLocated(By.xpath("//*[contains(text(),'You have successfully
deleted data')]")));
        String actualmsg = deleteMsgEle.getText();
        return actualmsg;
    }

    // method to verify the assertion , edit information text should be present on
    // the window when save button is clicked without filling the mandatory
    // field.
    public String getTexttoverifyEditMsg() {

        String actualmsg = editinfoMsgEle.getText();

        return actualmsg;
    }
}

```

Screen Shot From console :

```
Console [X] Results of running class BaseTest Coverage
<terminated> BaseTest [TestNG] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (Dec 17, 2018, 11:00:44 AM)
[RemoteTestNG] detected TestNG version 6.14.3
Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on pc
Only local connections are allowed.
Dec 17, 2018 11:00:49 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
PASSED: testcase2

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

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

Marketing

System

Reports

Test Line

Meta Title

Meta Tag Description

Meta Tag Keywords

Please fill out this field.

I remember I used to add a piece of code the HTML text input field so that ... the suggestions showing then disable that option in your browser.

about us, about