Test Objective:

The objective is to automate the below testcase using selenium WebDriver and framework concepts.

Testcase: Tc07

Process: Modify the address and verify the same on user page

Steps to Execute:

Step1: Login to admin portal

Step2: Go to System and then Settings

Step3: Change the address

Step4: Go to user page

Step5: Verify the same on user pge

Note: Search for the modified email-address

Expected Results:

Validate the presence of modified phone number under admin portal

Verify the address on the user page.

Note: Address presents on bottom of the user page.

Source code:

File1: `EditAddress.java`

```
package com.ibm.groceries;

import java.io.IOException;
import org.openqa.selenium.By;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Test;
import com.ibm.groceriespages.EditProductsPage;
import com.ibm.groceriespages.EditSettingPage;
import com.ibm.groceriespages.GroceriesUserPage;
import com.ibm.groceriespages.PageDashboard;
import com.ibm.groceriespages.PageLogin;
import com.ibm.groceriespages.PageProducts;
import com.ibm.initialization.WebDriverLaunch;

public class EditAddress extends WebDriverLaunch {

	@Test(priority = 1, testName = "EditAddress", groups = "low")
	public void editAddress() throws IOException, InterruptedException {
		String url = data.get("url");
		String userName = data.get("username");
		String password = data.get("password");
		String userPage = data.get("userPageUrl");
		String newAddress = data.get("addressNew");
		String newEmail = data.get("emailNew");
		String newPhone = data.get("phoneNew");
		String expMessage = data.get("expectedEditSettingMessage");
		String addressFoundMessage = data.get("addressFoundMsg");
		String emailFoundMessage = data.get("emailFoundMsg");
		String phoneFoundMessage = data.get("phoneFoundMsg");

		// Launching the web site for atozgroceries
		driver.get(url);

		PageLogin login = new PageLogin(driver, wait);
		// To enter email address and password and clickon login button
		login.enterEmailAddress(userName);
		login.enterPassword(password);
		login.clickOnLogin();

	Assert.assertTrue(driver.findElement(By.partialLinkText("Logout")).isDisplayed
());

		PageDashboard dashboard = new PageDashboard(driver, wait);

		// calling method to click on System link
		dashboard.clickOnsystem();

		// calling method to click on Settings link
		dashboard.clickOnSettings();

		// TO wait for the text area box address to be displayed.

	wait.until(ExpectedConditions.presenceOfElementLocated(By.id("address")));

		EditSettingPage editPage = new EditSettingPage(driver, wait);

		// calling method to edit address,email and phone
		String pageSource = editPage.editSettingInfo(newAddress, newEmail,
newPhone);
		Thread.sleep(10000);
		try {
			// Verifying for message whether the address,email,phone updated
or not on admin page
			if (pageSource.contains(expMessage)) {
				System.out.println(expMessage);
				Reporter.log(expMessage);


				// Assertion on expected message when settings updated
				Assert.assertTrue(pageSource.contains(expMessage));
				// Assertion on new email
				Assert.assertTrue(pageSource.contains(newEmail));
```

```
                    // Assertion new phone number
                    Assert.assertTrue(pageSource.contains(newPhone));
                }
            }

            catch (Exception e) {
                Assert.fail();
            }

            finally {
                // To click on on Logout button
                driver.findElement(By.partialLinkText("Logout")).click();

            }

            // To launch groceries user page
            driver.get(userPage);
            GroceriesUserPage userpage = new GroceriesUserPage(driver, wait);
            String userPageSource = userpage.verifyAddress();

            // Verifying for whether the address,email,phone updated or not on user
page
            try {
                if (userPageSource.contains(newAddress)) {
                    Assert.assertTrue(userPageSource.contains(newAddress));
                    System.out.println(addressFoundMessage);
                }
                if (userPageSource.contains(newEmail)) {
                    Assert.assertTrue(userpage.verifyEmail(newEmail));
                    System.out.println(emailFoundMessage);
                }
                if (userPageSource.contains(newPhone)) {
                    Assert.assertTrue(userpage.verifyPhone(newPhone));
                    System.out.println(phoneFoundMessage);
                }

            } catch (Exception e) {
                Assert.fail();
            }

        }
}
```

File2: PageLogin.java

```
package com.ibm.groceriespages;
import org.openqa.selenium.By;
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.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
```

```
public class PageLogin {

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

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

	//@FindBy(className="btn btn-labeled btn-info m-b-5")
	@FindBy(xpath="//button[@class='btn btn-labeled btn-info m-b-5']")
	WebElement loginEle;

	WebDriverWait wait;
	WebDriver driver;

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

	//To enter email address
	public void enterEmailAddress(String userName)
	{
		emailEle.sendKeys(userName);
	}

	//To enter apssword
	public void enterPassword(String password)
	{
		passEle.sendKeys(password);
	}

	//To click on Login button
	public void clickOnLogin()
	{
		loginEle.click();

	}

}
```

File3: PageDashboard.java

package com.ibm.groceriespages;

import org.openqa.selenium.By;

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.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;


public class PageDashboard {


        //@FindBy(xpath="//a[@class='material-ripple']")

        @FindBy(xpath="//a[text()=' Catalog']")

        WebElement catalogEle;


        @FindBy(xpath="//a[text()=' Products']")

        WebElement productEle;


        By prodElt=By.xpath("//a[text()=' Products']");

        WebDriverWait wait;

        WebDriver driver;


        //Locating elements for Marketing link

        @FindBy(xpath="//a[text()='  Marketing']")

        WebElement marketEle;


        //Locating Pushnotification link

        @FindBy(xpath="//a[text()=' Push Notification']")

        WebElement notificatonEle;




        //Locating System link

        @FindBy(xpath="//a[text()='   System']")
```

```
WebElement systemEle;

//Locating Returns link
@FindBy(xpath="//a[text()='  Returns']")
//@FindBy(xpath("//a[contains(text(),'Returns')]"))
WebElement returnsEle;

//Locating Shipping link
@FindBy(xpath="//a[text()='  Shipping']")
WebElement shippingEle;

//Locating ShippingLocatons link
@FindBy(xpath="//a[text()=' Shipping Locations']")
WebElement locatoinEle;

@FindBy(xpath="//a[text()=' Return Actions']")
WebElement actionsEle;

@FindBy(xpath="//a[text()=' Settings']")
WebElement settingsEle;

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

//To click on Catalog
public void clickOnCatalog()
```

```
{
        catalogEle.click();
        wait.until(ExpectedConditions.presenceOfElementLocated(prodElt));


        }
//To click on Products
public void clickOnProducts()
        {
        productEle.click();


}



//To click on Marketing link
public void clickOnMarketing()
{


        marketEle.click();


}





//To click on Push Notificatoin link
public void clickOnPushNotification()
{
        notificatonEle.click();


}
```

```
            //TO click on System link
public void clickOnsystem()
{
            systemEle.click();

}

//To click on Retuns link
public void clickOnReturns()
{
            driver.findElement(By.partialLinkText("Returns")).click();
            //returnsEle.click();
}

//To click on Return actoins link
public void clickOnRetrunActions()

{
            actionsEle.click();
}

public void clickOnShipping()
{
            driver.findElement(By.partialLinkText("Shipping")).click();
            //shippingEle.click();

}
```

```
	public void clickOnShippingLocations()
	{
		locatoinEle.click();

	}

	//To click on Settings link
	public void clickOnSettings()
	{
		settingsEle.click();
	}

}
```

File4: `EditSettingPage.java`

```
package com.ibm.groceriespages;

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;

public class EditSettingPage {

	//To locate Address text area box
	@FindBy(xpath="//textarea[@id='address']")
	WebElement addressEle;

	//To locate email text box
	@FindBy(xpath="//input[@id='email']")
	WebElement emailEle;

	//To locate phonenumber text box
	@FindBy(xpath="//input[@id='phonenumber']")
	WebElement phoneEle;

	// To locate save button
	@FindBy(xpath = "//button[@title='Save']")
	WebElement saveEle;
	WebDriverWait wait;
	WebDriver driver;

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


	public String editSettingInfo(String address,String email, String phone)
	{
		//To edit the address
		addressEle.clear();
		addressEle.sendKeys(address);


		//To edit email
		emailEle.clear();
		emailEle.sendKeys(email);

		//To edit phone number
		phoneEle.clear();
		phoneEle.sendKeys(phone);

		//To click on Save button
		saveEle.click();
		return driver.getPageSource();
	}



}
```

File5: GroceriesUserPage.java

```
package com.ibm.groceriespages;


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;


public class GroceriesUserPage {
```

```
@FindBy(xpath = "//input[@placeholder='Search for products...']")
WebElement searchEle;

@FindBy(xpath = "(//div[@class='input-group']/descendant::input[1])[2]")
WebElement searchEle2;

@FindBy(xpath = "//div[@id='searchproducts-div']/descendant::a[1]")
WebElement productNewLink;

// To locate email link on user page
@FindBy(xpath = "(//div[@class='header-mid-right-content']/descendant::a[1])[2]")
WebElement emailLink;

// To locate phone link on user page
@FindBy(xpath = "(//div[@class='header-mid-right-content']/descendant::a[1])[1]")
WebElement phoneLink;

WebDriverWait wait;
WebDriver driver;

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

public String searchProduct(String proudctName) throws InterruptedException {
		searchEle.sendKeys(proudctName);
		searchEle.click();

		// To enter product name in search text box of popup
		searchEle2.sendKeys(proudctName);
		Thread.sleep(10000);
		productNewLink.click();

		return driver.getPageSource();

	}

	// To verify the new address on user page
	public String verifyAddress() {
		return driver.getPageSource();
	}

	// To verify the new email link is on user page
	public boolean verifyEmail(String newemail) {

		if (emailLink.getText().contains(newemail))

			return true;
		else
			return false;
	}

	// To verify the new phone number is user page
	public boolean verifyPhone(String newphone) {
```

```
                    if (phoneLink.getText().contains(newphone))


                              return true;
                    else
                              return false;


          }


}
```

File6: WebDriverLaunch.java

```
package com.ibm.initialization;
import java.io.IOException;
import java.util.HashMap;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Optional;
import org.testng.annotations.Parameters;
import com.ibm.utilities.PropertiesFileHandler;
```

```
public class WebDriverLaunch {

        public WebDriver driver;

        public WebDriverWait wait;

        public PropertiesFileHandler propFileHandler;

        public HashMap<String, String> data;


        //Getting the keys from properties file

        //@BeforeSuite

        @BeforeSuite(groups= {"high","low"})

                public void preSetForTest() throws IOException {

                String file = "./TestData/groceries.properties";

                propFileHandler = new PropertiesFileHandler();

                data = propFileHandler.getPropertiesAsMap(file);

        }



        //@BeforeTest


        @BeforeMethod(groups= {"high","low"})

        @Parameters({"browser"})

        public void Initialization(@Optional("ff")String browser) {

                browserInitialization(browser);

                wait = new WebDriverWait(driver, 60);

                driver.manage().window().maximize();

                driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

        }



        //@AfterMethod
```

```
	//Closing driver
	@AfterMethod(groups= {"high","low"})
	public void closeBrowser() {
		driver.quit();
	}

	//Setting path for webdriver
	public void browserInitialization(String browser)
	{
		switch (browser.toLowerCase()) {
		case "ff":
		case "firefox":
			System.setProperty("webdriver.gecko.driver", "./drivers/geckodriver.exe");
			driver = new FirefoxDriver();
			break;
		case "ch":
			System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
			driver = new ChromeDriver();
			break;
		case "ie":
			System.setProperty("webdriver.ie.driver", "./drivers/IEDriverServer.exe");
			driver = new InternetExplorerDriver();
			break;
		default:
			System.out.println("No browser Available "+browser);
			break;
		}
	}
}
```

File7: 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 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();

        }


}
```

File9:groceries.properties

```
url=https://atozgroceries.com/admin
username=demo@atozgroceries.com
password=456789
expectedMessage=Success: You have successfully deleted data!
notifyName=DemoNotification
notifyMessage=Adding Notification
expNotificationMessage=Success: You have successfully sent push notification to your
app!
imagePath=C:\\Users\\IBM_ADMIN\\eclipse-
workspace\\TestCases_DataDrivenFramework\\TestData\\globe.jpg
searchForKeyword=refund
searchDispalyMessage=Search with keyword is successful
noMatchDisplayMessage=No matching records found
searchForShipKeyword=Discount
prodNameNew=ProductNew100
modelNameNew=Model-New
expectedEditProductMessage=Success: You have successfully updated product!
productNotUpdated=Product update failed
userPageUrl=https://atozgroceries.com
userPageMsg=New Product found on User page
addressNew=Koramangala,Bengaluru
emailNew=info@atozgroceries.com
phoneNew=9797979797
expectedEditSettingMessage=successfully updated Store!
addressFoundMsg=new Address found on user page.
emailFoundMsg=new Email found on user page.
```

```
phoneFoundMsg=new Phone number found on user page.
```

File9:testng_EditSettings.xml

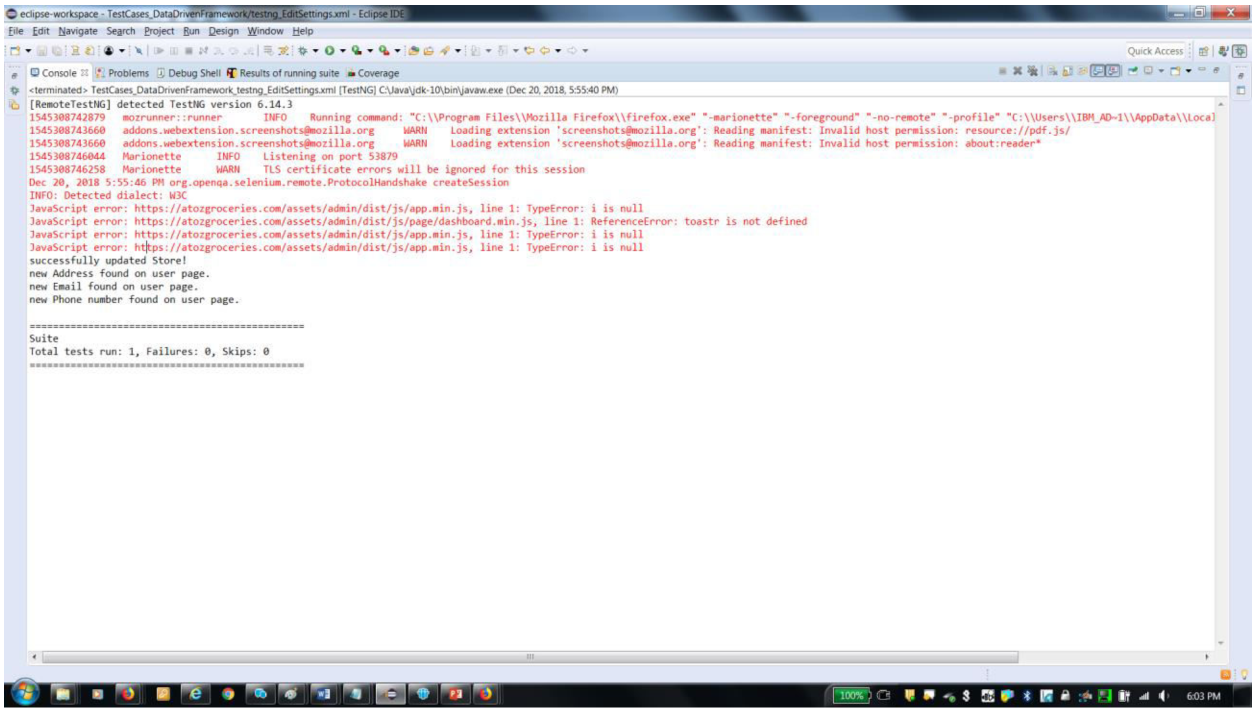
```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="EditingSettings">
  <parameter name="browser" value="firefox"></parameter>
    <classes>
      <class name="com.ibm.groceries.EditAddress"/>
    </classes>
  </test> <!-- EditSettings -->
</suite> <!-- Suite -->
```

Test Result:

It is validated that the new address,new phone number, new email is updated and displayed on user portal and also admin portal of atozgroceries site.

Screenshots:

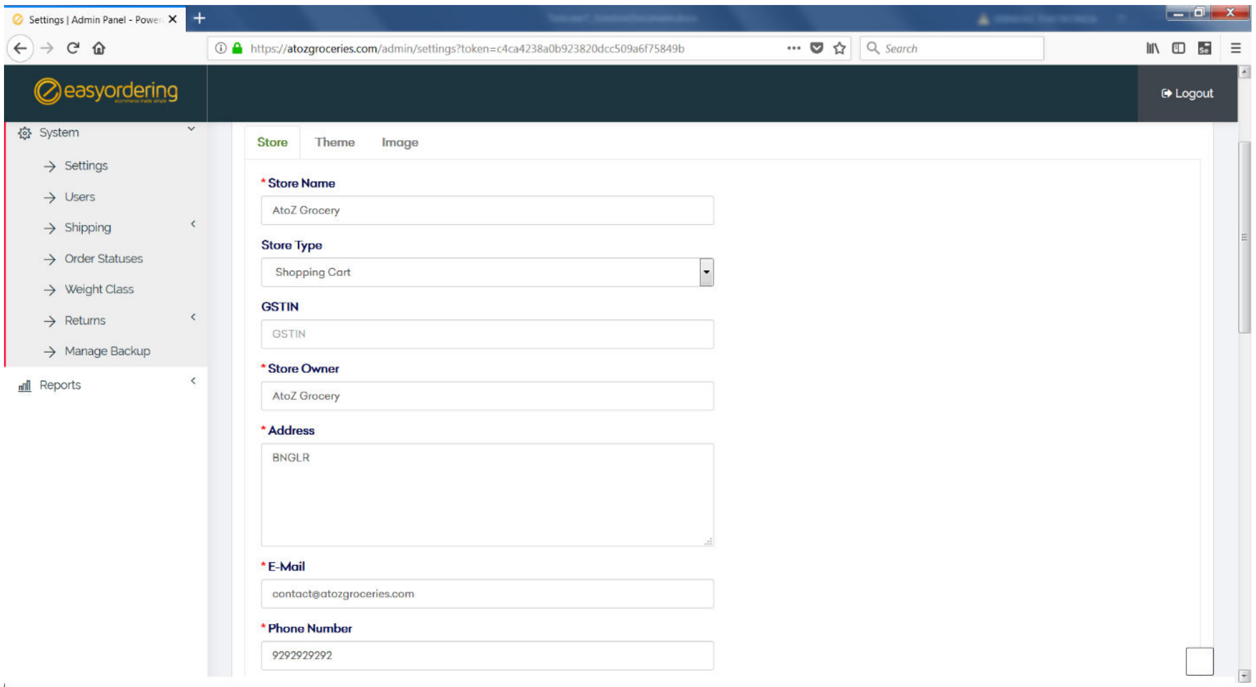
Console output:



```
eclipse-workspace - TestCases_DataDrivenFramework/testng_EditSettings.xml - Eclipse IDE
<terminated> TestCases_DataDrivenFramework_testng_EditSettings.xml [TestNG] C:\Java\jdk-10\bin\javaw.exe (Dec 20, 2018, 5:55:40 PM)
[RemoteTestNG] detected TestNG version 6.14.3
1545308742879   mozrunner::runner       INFO    Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "-marionette" "-foreground" "-no-remote" "-profile" "C:\\Users\\IBM_AD~1\\AppData\\Local
1545308743660   addons.webextension.screenshots@mozilla.org     WARN    Loading extension 'screenshots@mozilla.org': Reading manifest: Invalid host permission: resource://pdf.js/
1545308743660   addons.webextension.screenshots@mozilla.org     WARN    Loading extension 'screenshots@mozilla.org': Reading manifest: Invalid host permission: about:reader*
1545308746044   Marionette      INFO    Listening on port 53879
1545308746258   Marionette      WARN    TLS certificate errors will be ignored for this session
Dec 20, 2018 5:55:46 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/page/dashboard.min.js, line 1: ReferenceError: toastr is not defined
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
successfully updated Store!
new Address found on user page.
new Email found on user page.
new Phone number found on user page.

===============================================
Suite
Total tests run: 1, Failures: 0, Skips: 0
===============================================
```

Existing address



Settings | Admin Panel - Power

https://atozgroceries.com/admin/settings?token=c4ca4238a0b923820dcc509a6f75849b

* Store Name: AtoZ Grocery

Store Type: Shopping Cart

GSTIN: GSTIN

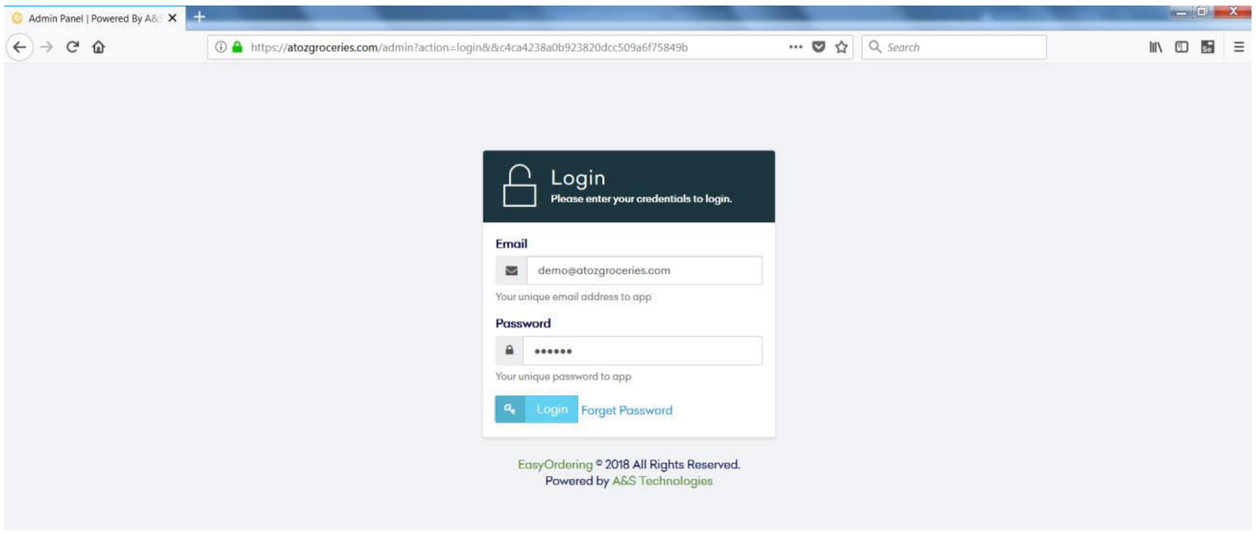
* Store Owner: AtoZ Grocery

* Address: BNGLR

* E-Mail: contact@atozgroceries.com

* Phone Number: 9292929292

Login page is displayed



Admin Panel | Powered By A&

https://atozgroceries.com/admin?action=login&&c4ca4238a0b923820dcc509a6f75849b

Login

Please enter your credentials to login.

Email: demo@atozgroceries.com

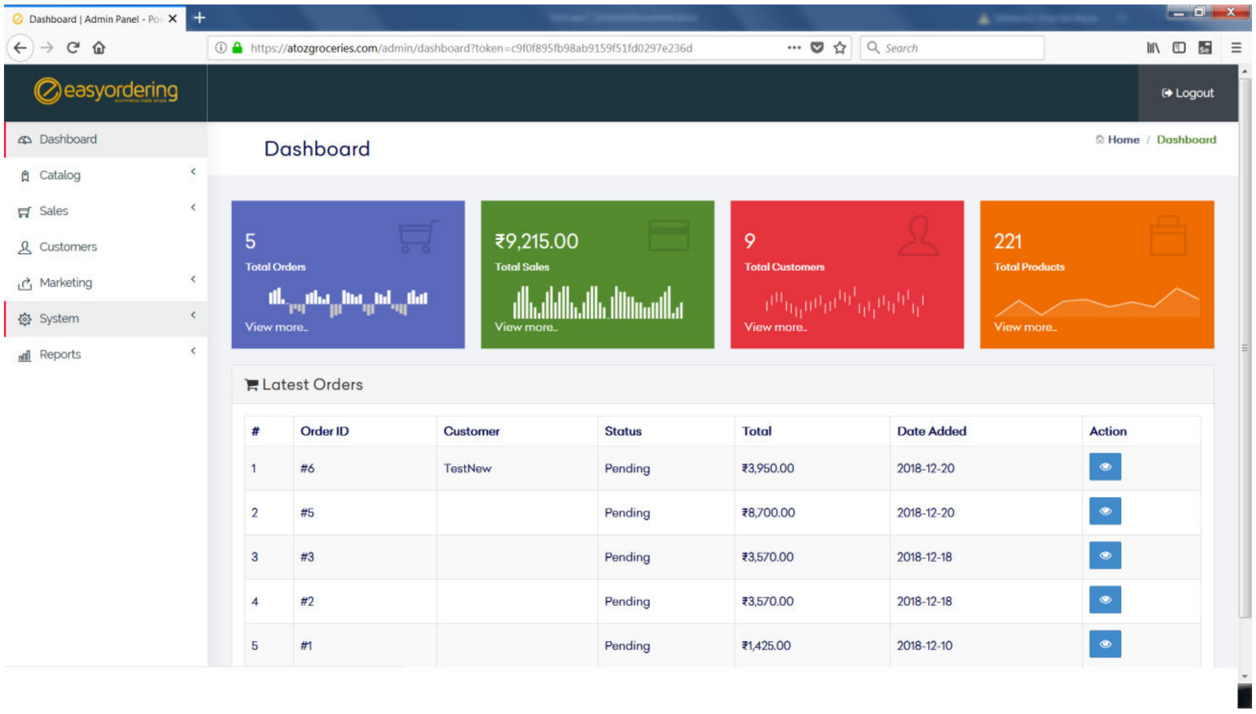
Your unique email address to app

Password: ••••••

Your unique password to app

EasyOrdering © 2018 All Rights Reserved.
Powered by A&S Technologies

Dashboard page is displayed


Dashboard | Admin Panel - Po
https://atozgroceries.com/admin/dashboard?token=c9f0f895fb98ab9159f51fd0297e236d
easyordering
Logout
Dashboard
Catalog
Sales
Customers
Marketing
System
Reports
Home / Dashboard
View more..

Dashboard

5 Total Orders

₹9,215.00 Total Sales

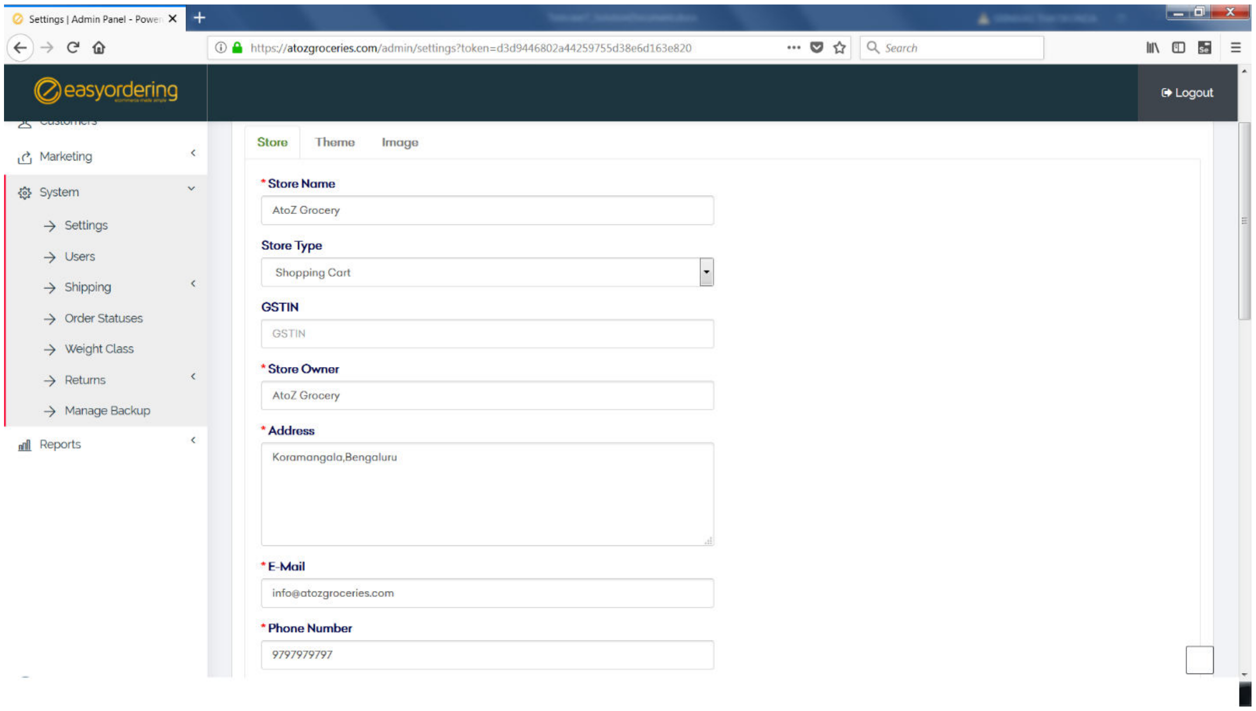
9 Total Customers

221 Total Products

Latest Orders

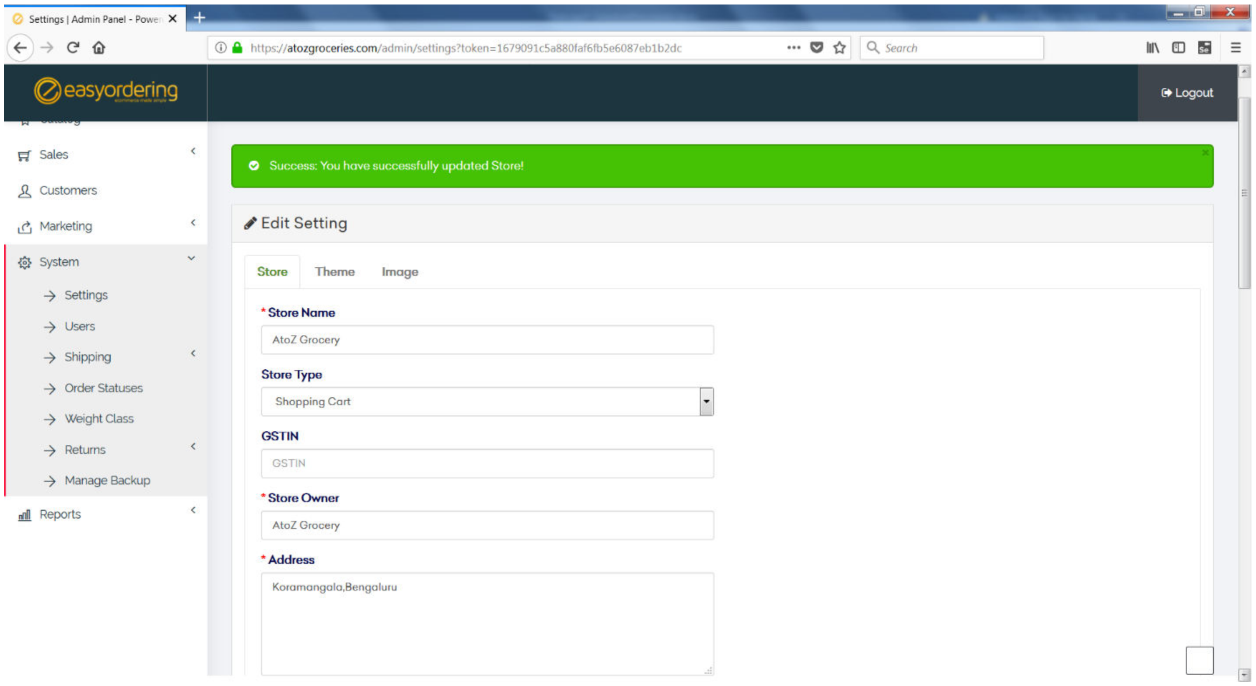
#	Order ID	Customer	Status	Total	Date Added	Action
1	#6	TestNew	Pending	₹3,950.00	2018-12-20	
2	#5		Pending	₹8,700.00	2018-12-20	
3	#3		Pending	₹3,570.00	2018-12-18	
4	#2		Pending	₹3,570.00	2018-12-18	
5	#1		Pending	₹1,425.00	2018-12-10	

Edit setting page is displayed



* Store Name	AtoZ Grocery
Store Type	Shopping Cart
GSTIN	GSTIN
* Store Owner	AtoZ Grocery
* Address	Koramangala,Bengaluru
* E-Mail	info@atozgroceries.com
* Phone Number	9797979797

Expected message is displayed after updating the address.



Settings | Admin Panel - Power

https://atozgroceries.com/admin/settings?token=1679091c5a880faf6fb5e6087eb1b2dc

Success: You have successfully updated Store!

Edit Setting

Store | Theme | Image

* Store Name: AtoZ Grocery

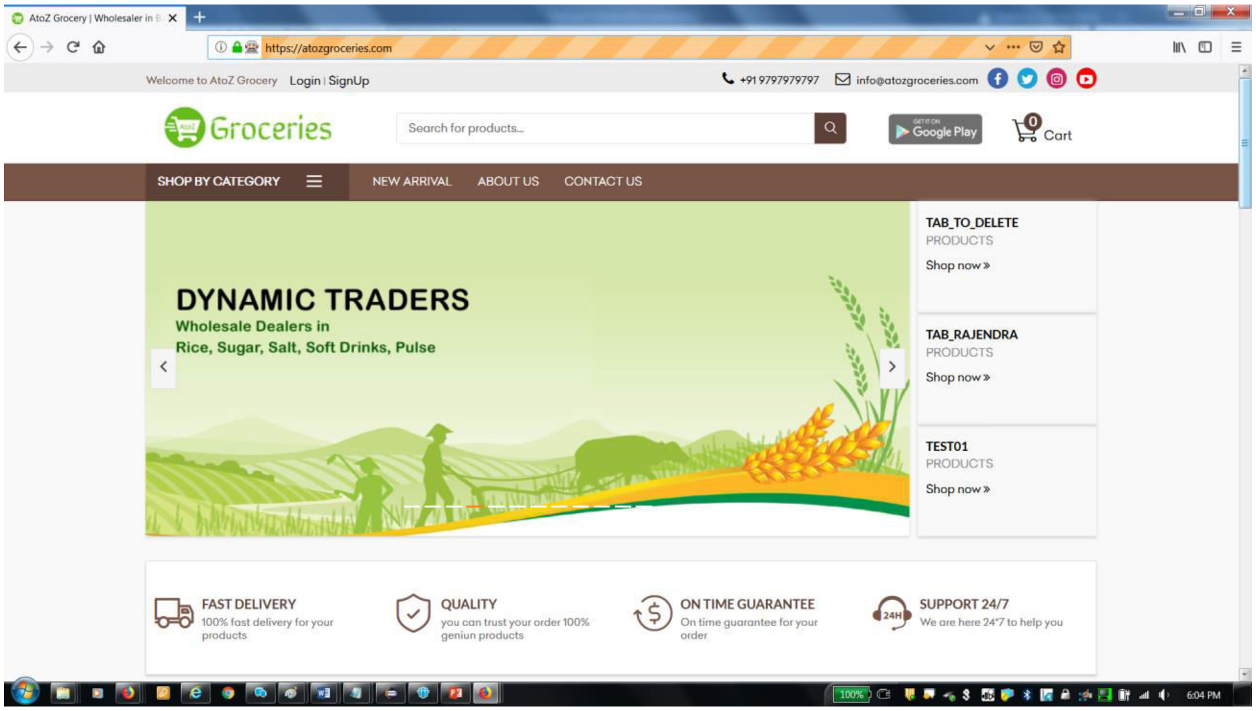
Store Type: Shopping Cart

GSTIN: GSTIN

* Store Owner: AtoZ Grocery

* Address: Koramangala,Bengaluru

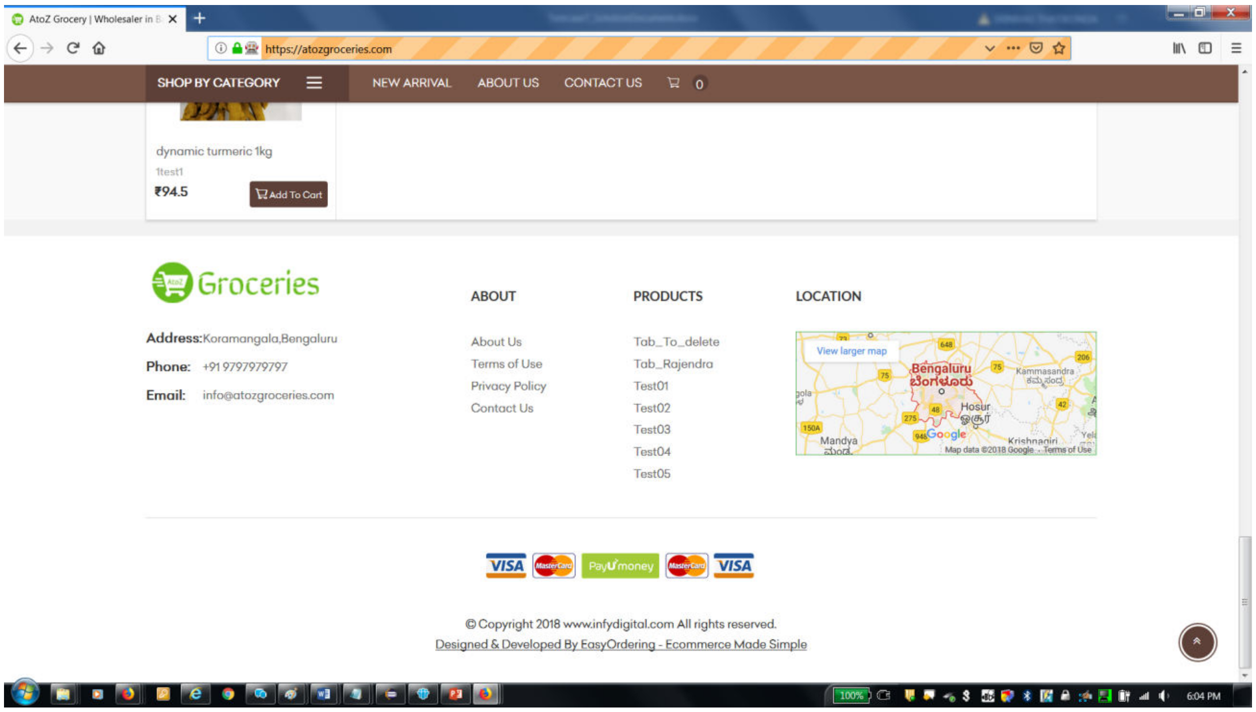
User portal is displayed with newly updated address ,email and phone number



AtoZ Grocery | Wholesaler in B

https://atozgroceries.com

Welcome to AtoZ Grocery Login | SignUp +91 9797979797 info@atozgroceries.com



dynamic turmeric 1kg

1test1

₹94.5 Add To Cart

Groceries

Address: Koramangala,Bengaluru

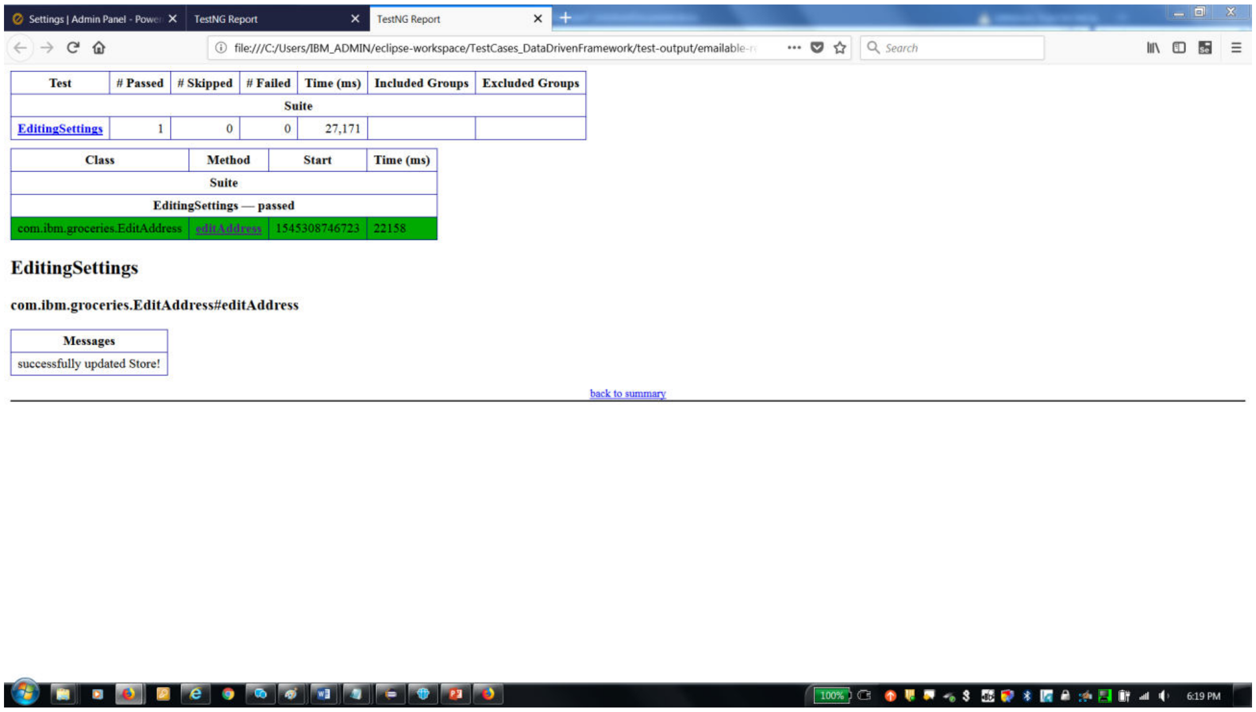
Phone: +91 9797979797

Email: info@atozgroceries.com

ABOUT	PRODUCTS	LOCATION
About Us	Tab_To_delete	
Terms of Use	Tab_Rajendra	
Privacy Policy	Test01	
Contact Us	Test02	
	Test03	
	Test04	
	Test05	

© Copyright 2018 www.infydigital.com All rights reserved.

Designed & Developed By EasyOrdering - Ecommerce Made Simple


Settings | Admin Panel - Power
TestNG Report
TestNG Report
file:///C:/Users/IBM_ADMIN/eclipse-workspace/TestCases_DataDrivenFramework/test-output/emailable-r
Search

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
EditingSettings	1	0	0	27,171		

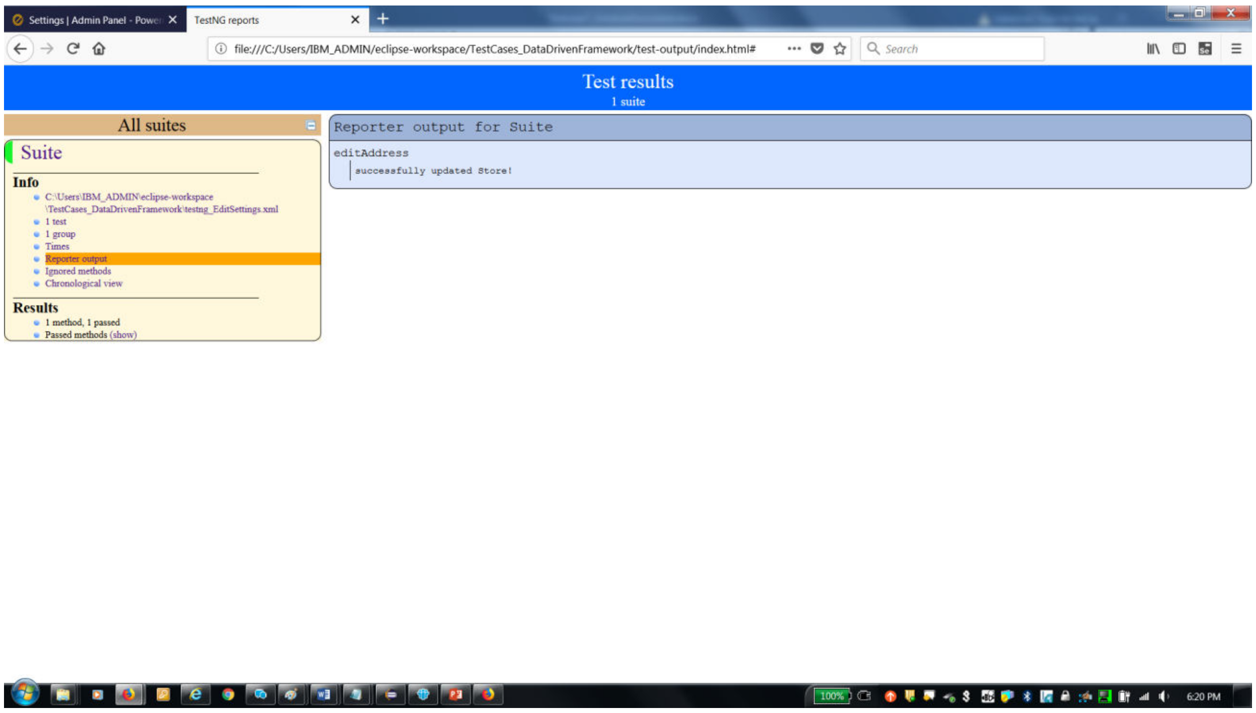
Class	Method	Start	Time (ms)
Suite			
EditingSettings — passed			
com.ibm.groceries.EditAddress	editAddress	1545308746723	22158

EditingSettings

com.ibm.groceries.EditAddress#editAddress

Messages
successfully updated Store!

back to summary


Settings | Admin Panel - Power
TestNG reports
file:///C:/Users/IBM_ADMIN/eclipse-workspace/TestCases_DataDrivenFramework/test-output/index.html#
Search
6:20 PM

Test results
1 suite

All suites

Suite

Info
- C:\Users\IBM_ADMIN\eclipse-workspace\TestCases_DataDrivenFramework\testng_EditSettings.xml
- 1 test
- 1 group
- Times
- Reporter output
- Ignored methods
- Chronological view

Results
- 1 method, 1 passed
- Passed methods (show)

```
Reporter output for Suite
editAddress
  successfully updated Store!
```

Flow Information:

Initially a property file 'groceries.properties' is created in Test Data folder of the project and list of key, values for admin page url, username,password ,expected validation message, user page url,new address,new email,new phone number etc. are stored in this property file.

Under the class 'WebDriverLaunch' ,a Test NG annotation 'Before Suite' is used to invoke the method 'preSetFortTest' which is used to instantiate object for class PropertiesFileHandler to return the list of these key,values in to a HashMap.

The Test NG annotation 'Before Method' is used by passing the parameter for browser to select and to invoke the method 'Initialization' which is used to call another method and to create webdriver object for the firefoxdriver or chromedriver or internetexplorer by using switch statement and setting path for webdriver exe.

The PageLogin class is defined with methods to locate the web elements email,password and login buttons. PageFactory is used in the constructor of this class. The method sendKeys is used to enter the email address and password. The method click is used to click on login button

The class 'PageDashboard' is defined with methods to locate the web elements System,Settings. PageFactory is used in the constructor of this class.The method click is used to click on System, Settings links.

The class ‘EditSettingPage’ is defined with method editSettingInfo to enter new values for address,email and phonenumber whiich are located using ‘FindBy’ annotation . Click method is used to click on Save button.PageFactory method is used in the in the constructor of this class and pagesource is returned.

The class ‘GroceriesUserPage’ uses FindBy annotation to locate the elements email and phonenumber links.The method verifyemail is defined to check the presence of new email link,the method verifyPhone is defined to check the presence of new phonenumber link in user portal and verifyAddress is used to return the pagesource.

The testng_EditSetting xml file is having the parameter ‘firefox’ to launch Firefox browser.

The annotation @Test is used with method editAddress in the class ‘EditAddress’. This method is used to get the values for admin page url, username,password ,expected validation message, user page url, new address,new email,new phone number etc. The get method is called to launch the web url for admin portal as well as user page. The objects for classes PageLogin,PageDashboard, EditSettingPage,GroceriesUserPage are created and the above necessary methods are called on these objects. Finally the Assert is used   by verifying presence of the new address,new email,new phone number in user portal as well as admin portal when updated.