

## Day 10 - Test case solution document

### Code Used:

#### BaseTest.java

```
package com.ibm.test;

import java.io.File;
import java.io.IOException;
import java.sql.SQLException;
import java.util.Date;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Optional;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;
import com.ibm.pages.CategoryPage;
import com.ibm.pages.CouponPage;
import com.ibm.pages.DashboardPage;
import com.ibm.pages.DriverLaunch;
import com.ibm.pages.LoginPage;
import com.ibm.pages.ProductsPage;
import com.ibm.pages>ReturnsPage;
import com.ibm.pages.SettingsPage;
import com.ibm.pages.ShippingPage;
import com.ibm.pages.TabsPage;
import com.ibm.pages.UserPage;
import com.ibm.utilities.DBConnection;
import com.ibm.utilities.ExcelReader;

public class BaseTest extends DriverLaunch
{
    @DataProvider(name="EditCoupon")
    public Object[][] Coupon() throws IOException
    {
        return ExcelReader.DataTable("./TestData/TestData.xlsx", "Coupon");
    }
    @Test(testName="EditCoupon",dataProvider="EditCoupon")
    public void Day10TestCase(String oldCouponCode,String CouponName,String newCouponCode, String discount, String CpnMinAmnt, String CpnStartDate, String CpnEndDate, String usesPerCpn, String usesPerCust, String Cpnstatus) throws InterruptedException, IOException, SQLException
    {
        LoginPage login=new LoginPage(driver,wait);
        DashboardPage dash=new DashboardPage(driver,wait);
        CouponPage coupon=new CouponPage(driver,wait);
```

```

DBConnection dbconn=new DBConnection();
try
{
    String adminurl=hashData.get("adminurl");
    String username=hashData.get("username");
    String password=hashData.get("password");
    //navigating to the Admin URL of atozgroceries
    driver.navigate().to(adminurl);

    //login page is used to provide the input of username and
    password
    login.EnterEmail(username);
    login.EnterPassword(password);
    login.clickLogin();

    //Opening Marketing in dashboard
    dash.OpenMarketing();

    //Opening coupons in Marketing
    coupon.ClickCoupon();
    //getting the count of coupons before editing
    int CouponTotalAfter=dbconn.DBCouponCount();

    //getting the pagesource for validating
    String updateSuccessMsg=coupon.pageSource();
    //Assertion on edited coupon
    Assert.assertTrue(updateSuccessMsg.contains(newCouponCode),
"Assertion on edited coupon in admin page!!!");
    Reporter.Log("Assertion on Edited Coupon in Admin Page!!!");
    //Assertion on total coupons in DB before and after editing
    Assert.assertEquals(CouponTotalBfr,CouponTotalAfter);
    Reporter.Log("Assertion on total coupons before and after editing
coupon!!!");
    //calling the DB method to verify the Old and New coupon codes
    dbconn.DBCouponVerify(oldCouponCode, newCouponCode);
}
catch(Exception e)// catching if any exception occurs and takes a
screenshot of the page
{
    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/Day10_TestCase_Error_" + currentDate + ".jpg"));
    System.out.println(e.getMessage());
    Assert.fail();
}
finally
{
    login.ClickLogout();
}
}
}

```

## **LoginPage.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;

public class LoginPage {
    @FindBy(name="email")
    WebElement emailEle;

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

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

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

    WebDriverWait wait;
    WebDriver driver;

    public LoginPage(WebDriver driver, WebDriverWait wait)
    {
        PageFactory.initElements(driver, this);
    }
    public void EnterEmail(String username)
    {
        emailEle.sendKeys(username);
    }
    public void EnterPassword(String password)
    {
        passwordEle.sendKeys(password);
    }
    public void clickLogin()
    {
        loginEle.click();
    }
    public void ClickLogout()
    {
        logoutEle.click();
    }
}
```

## **DashboardPage.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.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class DashboardPage {

    //Finding the WebElements
    @FindBy(partialLinkText="Catalog")
    WebElement catalogEle;

    @FindBy(xpath="/html/body/div[1]/div[1]/div/ul/li[5]/a")
    WebElement marketEle;

    @FindBy(xpath="//*[contains(text(),'System')]")
    WebElement systemEle;

    @FindBy(xpath="//*[contains(text(),'Shipping')]")
    WebElement shipEle;

    WebDriver driver;
    WebDriverWait wait;

    //catalog page constructor
    public DashboardPage(WebDriver driver, WebDriverWait wait)
    {
        this.driver=driver;
        this.wait=wait;
        PageFactory.initElements(driver, this);
    }

    //Opencatalog method to click on catalog
    public void OpenCatalog()
    {
        wait.until(ExpectedConditions.visibilityOf(catalogEle));
        catalogEle.click();
    }

    //method for opening marketing
    public void OpenMarketing()
    {
        marketEle.click();
    }
    //method for opening system
    public void OpenSystem()
    {
        systemEle.click();
    }
    public void OpenShipping()
    {
        shipEle.click();
    }
}

```

## CouponPage.java

```
package com.ibm.pages;

import java.util.List;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
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.Select;
import org.openqa.selenium.support.ui.WebDriverWait;

public class CouponPage
{
    //Finding elements in Coupon page
    @FindBy(xpath="/html/body/div[1]/div[1]/div/ul/li[5]/ul/li[1]/a")
    WebElement couponEle;
    @FindBy(name="name")
    WebElement couponName;
    @FindBy(name="code")
    WebElement couponCode;
    @FindBy(name="discount")
    WebElement couponDisc;
    @FindBy(id="toTop")
    WebElement toTop;
    @FindBy(xpath="//button[@title='Save']")
    WebElement saveEle;
    @FindBy(name="min_amount")
    WebElement couponMinAmtEle;
    @FindBy(name="date_start")
    WebElement couponStrtEle;
    @FindBy(name="date_end")
    WebElement couponEndEle;
    @FindBy(name="uses_per_coupon")
    WebElement useCouponEle;
    @FindBy(name="uses_per_customer")
    WebElement useCustEle;
    @FindBy(name="status")
    WebElement statusEle;

    WebDriver driver;
    WebDriverWait wait;
    JavascriptExecutor js;
    //constructor
    public CouponPage(WebDriver driver,WebDriverWait wait)
    {
        this.driver=driver;
        this.wait=wait;
        js=(JavascriptExecutor) driver;
        PageFactory.initElements(driver, this);
    }
}
```

```

//opening Coupon under Marketing
public void ClickCoupon()
{
    couponEle.click();
}
//Method to delete coupon
public void DeleteCoupon(String code) throws InterruptedException
{
    //getting rows
    List<WebElement>
rowEle=driver.findElements(By.xpath("//table[@id='dataTableExample2']/tbody/tr"));
    //finding no of rows
    int rows=rowEle.size();
    for(int i=1;i<=rows;i++)
    {
        //getting the code of the coupon
        WebElement
codeEle=driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]"
/td[3]));
        String codeTxt=codeEle.getText();
        //getting the code row for deleting
        if(codeTxt.equals(code))
        {
            //clicking on Action button of deleting row

            driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/
td[8]/descendant::button")).click();
                //clicking on Delete button to delete the row

            driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/
td[8]/descendant::ul/li)[2]")).click();
                // added thread.sleep for waiting till the
confirmation screen appears
                Thread.sleep(5000);
                //clicking on the delete confirmation

            driver.findElement(By.xpath("//button[@class='confirm']")).click();
                break;
        }
    }
    public String pageSource() throws InterruptedException
    {
        //waiting till the Delete success message is displayed to get the
pagesource

        //wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//div[conta
ins(@class,'alert-danger')]")));
            //waiting till the Successfully updated message is displayed to get the
pagesource

        wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//div[contain
s(@class,'alert-success')]")));
            return driver.getPageSource();
    }
}

```

```

        }

    public void EditCoupon(String oldCode, String name, String newCode, String
discount, String minAmmt, String startdate, String enddate, String usesCoupon, String
usesCust, String status)
    {
        //getting rows
        List<WebElement>
rowEle=driver.findElements(By.xpath("//table[@id='dataTableExample2']/tbody/tr"));
        //finding no of rows
        int rows=rowEle.size();
        for(int i=1;i<=rows;i++)
        {
            //getting the code of the coupon
            WebElement
codeEle=driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/


```

```

        useCouponEle.sendKeys(usesCoupon);
        useCustEle.sendKeys(usesCust);
        Select selectStatus= new Select(statusEle);
        selectStatus.selectByVisibleText(status);
        //js.executeScript("arguments[0].scrollIntoView(true)",  

toTop);
        toTop.click();
        saveEle.click();

    }
}
}

DBConnection.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 org.testng.Assert;
import org.testng.Reporter;

public class DBConnection {

    public void DBCategoryVerify(String newCatName) throws SQLException
    {
        //Establishing the DB connection
        Connection c =
DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atozgr
oceries",
                "foodsonfinger_atoz","welcome@123");
        Statement s = c.createStatement();
        //Executing the required statement for getting the Edited category..
        sending the new catergory name in where condition
        ResultSet rs = s.executeQuery("SELECT * from as_category where
name=\""+newCatName+"\"");
        //the below while loop checks each and every line of the above query
result
        while(rs.next())
        {
            //doing assertion on New edited category name when matches the
name of the category in DB

            Assert.assertTrue(newCatName.equals(rs.getString("name")),"Assertion on Edited
Category in Database!!");
            Reporter.log("Assertion on Edited Category in Database!!");

        }
    }
    //method to count total no of coupons in DB
    public int DBCouponCount() throws SQLException

```

```

{
    int count=0;
    //Establishing the DB connection
    Connection
c=DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atoz
groceries",
                            "foodsonfinger_atoz","welcome@123");
    Statement s=c.createStatement();
    //Executing the required statement for counting the coupons
    ResultSet rs=s.executeQuery("select count(name) from
as_coupons");
    while(rs.next())
    {
        count=rs.getInt(1);
    }
    return count;
}
public void DBCouponVerify(String oldCode,String newCode) throws SQLException
{
    //Establishing the DB connection
    Connection
c=DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atoz
groceries",
                            "foodsonfinger_atoz","welcome@123");
    Statement s=c.createStatement();
    //Executing the required statement for validating the presence of edited
coupon code
    ResultSet rs=s.executeQuery("select * from as_coupons where
code=\""+newCode+"\"");
    while(rs.next())
    {
        String couponCode=rs.getString("code");
        //Assertion of new edited coupon code in DB!!
        Assert.assertEquals(couponCode,newCode);
        Reporter.log("Assertion on edited coupon in DB!!!");
        //Assertion of old coupon code in DB!!
        Assert.assertNotEquals(couponCode,oldCode);
        Reporter.log("Assertion on coupon before editing in
DB!!!");
    }
}
}

```

### **DriverLaunch.java**

```

package com.ibm.pages;

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.Optional;
import org.testng.annotations.Parameters;

import com.ibm.utilities.PropertiesFileHandler;

public class DriverLaunch {

    public WebDriver driver;
    public WebDriverWait wait;
    public PropertiesFileHandler propFileHandle;
    public HashMap<String, String> hashData;

    @BeforeSuite
    public void preSetForTest() throws IOException
    {
        String file=".TestData/data.properties";
        propFileHandle =new PropertiesFileHandler();
        hashData=propFileHandle.getPropertiesAsMap(file);
    }
    @BeforeMethod
    @Parameters({"browser"})
    public void initialisation(@Optional("ch")String browser)
    {
        browserInitialization(browser);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        wait=new WebDriverWait(driver, 60);
    }
    @AfterMethod
    public void closingBrowser()
    {
        driver.quit();
    }

    public void browserInitialization(String browser)
    {
        switch(browser.toLowerCase())
        {
            case "ff":
                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;
    }
}
```

## PropertiesFileHandler.java

```
package com.ibm.utilities;
```

```
import java.io.FileInputStream;
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;  
}  
}
```

## **data.properties**

```
adminurl=https://atozgroceries.com/admin  
userurl=https://atozgroceries.com  
username=demo@atozgroceries.com  
password=456789
```

testng.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">
        <parameter name="browser" value="ff"></parameter>
        <classes>
            <class name="com.ibm.test.BaseTest">
                <methods><include name="Day10TestCase"></include></methods>
            </class>
        </classes>
    </test> <!-- Test -->
```

```
</suite> <!-- Suite -->
```

## ExcelReader.java

```
package com.ibm.utilities;

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

import org.apache.poi.ss.usermodel.DataFormatter;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class ExcelReader {
    public static Object[][] DataTable(String WBLoc, String sheetName) throws
IOException
    {

        XSSFWorbook TestDataWB=new XSSFWorbook(new FileInputStream(WBLoc));
        XSSFSheet ProductsSheet=TestDataWB.getSheet(sheetName);
        Object[][] ProductInfo=new Object[ProductsSheet.getPhysicalNumberOfRows()-
1][ProductsSheet.getRow(0).getPhysicalNumberOfCells()];
        int rowcount=ProductsSheet.getPhysicalNumberOfRows();
        int cellcount=ProductsSheet.getRow(0).getPhysicalNumberOfCells();
        for(int i=1;i<rowcount;i++)
        {
            for(int j=0;j<cellcount;j++)
            {
                DataFormatter format=new DataFormatter();
                String
CellValue=format.formatCellValue(ProductsSheet.getRow(i).getCell(j));
                ProductInfo[i-1][j]=CellValue;
            }
        }
        return ProductInfo;
    }
}
```

## TestData.xlsx

The screenshot shows the Eclipse IDE interface with the title bar 'EclipseWorkSpace - SeleniumFramework/TestData/TestData.xlsx - Eclipse IDE'. The left sidebar displays the project structure under 'SamplePkg' and 'src'. The main area shows a Microsoft Excel spreadsheet titled 'TestData.xlsx' with the following data:

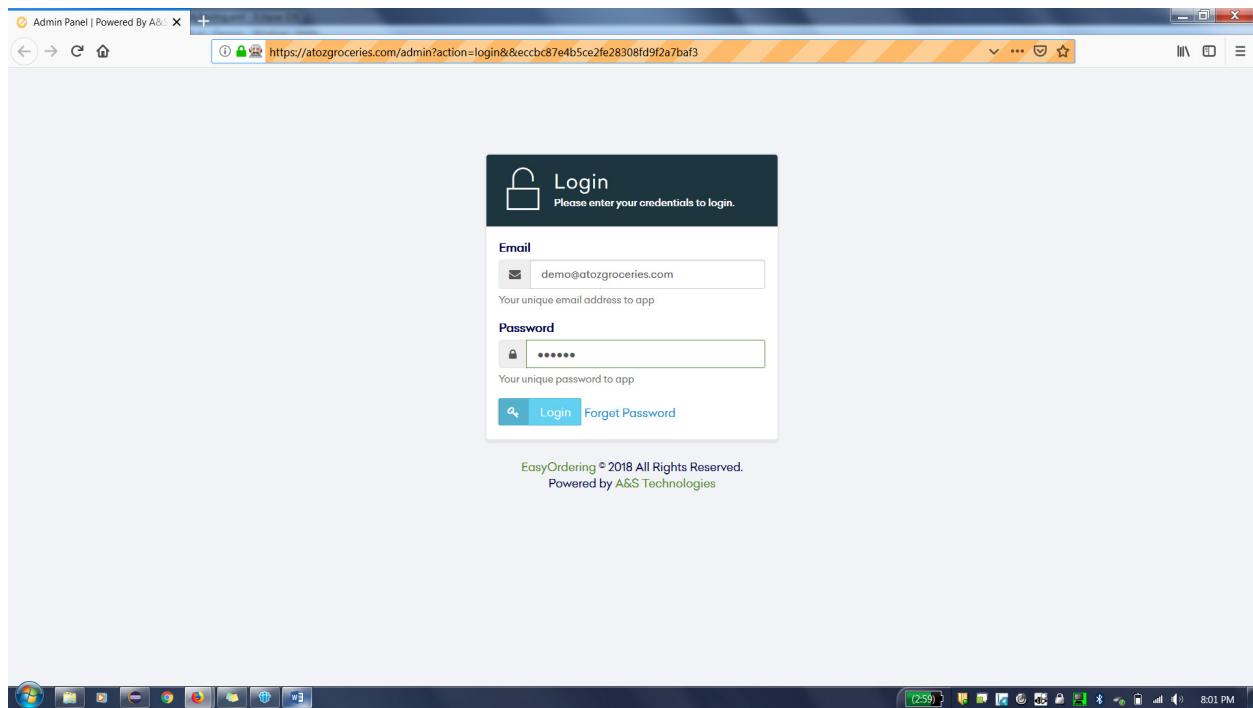
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	old_code	Cpname	new_code	discount	minAmt	StartDate	EndDate	usesPerCoupon	usesPerCust	status							
2	15OFF	New Year 2019 Offer	30JANOFF	30	1499	2019-01-01	2019-01-15	1	2	Enabled							
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	

## Step by step output:

### 1. Navigating to the Admin Page of "atozgroceries"

The screenshot shows a web browser window with the title 'Admin Panel | Powered By A&S Technologies'. The URL in the address bar is 'https://atozgroceries.com/admin?action=login&&eccbc87e4b5ce2fe28308fd9f2a7baef'. The page displays a login form with fields for 'Email' and 'Password', both currently empty. Below the fields are 'Login' and 'Forgot Password' buttons. At the bottom of the form, it says 'EasyOrdering © 2018 All Rights Reserved.' and 'Powered by A&S Technologies'. The status bar at the bottom of the screen shows 'Transferring data from atozgroceries.com...'.

## 2. Providing login credentials



## 3. Dashboard after logging into the site

A screenshot of a web browser window showing the 'Dashboard' page for 'easyordering'. The URL is https://atozgroceries.com/admin/dashboard?token=45c48cce2e2d7fbdea1afc51c7c6ad26. The dashboard features a sidebar with links like 'Dashboard', 'Catalog', 'Sales', 'Customers', 'Marketing', 'System', and 'Reports'. The main area has a title 'Dashboard' and four summary cards: 'Total Orders' (1), 'Total Sales' (₹5,593.00), 'Total Customers' (6), and 'Total Products' (45). Below this is a section titled 'Latest Orders' with a table showing five recent orders. The table columns are '#', 'Order ID', 'Customer', 'Status', 'Total', 'Date Added', and 'Action'. Each row contains an order number, ID, customer name, status (Pending), total amount, date added, and a blue 'View more...' button.

#### 4. Opening Marketing

The screenshot shows the Admin Panel Dashboard for 'easyordering'. The left sidebar has a 'Marketing' section selected, which includes 'Coupons', 'Mail', and 'Push Notification'. The main area displays four cards: 'Total Orders' (6), 'Total Sales' (₹27,965.00), 'Total Customers' (32), and 'Total Products' (230). Below these is a table titled 'Latest Orders' with columns: #, Order ID, Customer, Status, Total, Date Added, and Action. The table lists five pending orders.

#	Order ID	Customer	Status	Total	Date Added	Action
1	#10	Rajendra Sahoo	Pending	₹21,750.00	2018-12-21	
2	#6	TestNew	Pending	₹3,950.00	2018-12-20	
3	#5		Pending	₹8,700.00	2018-12-20	
4	#3		Pending	₹3,570.00	2018-12-18	
5	#2		Pending	₹3,570.00	2018-12-18	

#### 5. Opening Coupons

The screenshot shows the Admin Panel Coupons page for 'easyordering'. The left sidebar has a 'Marketing' section selected, which includes 'Coupons', 'Mail', and 'Push Notification'. The main area is titled 'Coupon List' and shows a table of existing coupons with columns: #, Coupon Name, Code, Discount, Date Start, Date End, Status, and Action. The table lists seven coupons, all currently enabled.

#	Coupon Name	Code	Discount	Date Start	Date End	Status	Action
1	25 Offer	25OFF	25%	0000-00-00	0000-00-00	Enabled	Action ▾
2	12 Offer	12OFF	12%	0000-00-00	0000-00-00	Enabled	Action ▾
3	15 Offer	15OFF	15%	2019-01-01	2019-01-15	Enabled	Action ▾
4	ForTest	Test	10%	0000-00-00	0000-00-00	Enabled	Action ▾
5	Coupon_Rajendra	ABC123	15%	2018-12-20	2018-12-30	Enabled	Action ▾
6	10% OFF on GROC 2000	GROC	15%	2018-12-20	2018-12-30	Enabled	Action ▾
7	20 offer	20OFF	20%	0000-00-00	0000-00-00	Enabled	Action ▾

6. Selecting and clicking on the “Action” of the coupon to edit

The screenshot shows the 'Coupons' section of the easyordering Admin Panel. On the left, there's a sidebar with navigation links like Dashboard, Catalog, Sales, Customers, Marketing (with sub-links for Coupons, Mail, Push Notification), System, and Reports. The main area is titled 'Coupons' and contains a table titled 'Coupon List'. The table has columns for #, Coupon Name, Code, Discount, Date Start, Date End, Status, and Action. There are 10 entries listed:

#	Coupon Name	Code	Discount	Date Start	Date End	Status	Action
1	25 Offer	25OFF	25%	0000-00-00	0000-00-00	Enabled	Action
2	12 Offer	12OFF	12%	0000-00-00	0000-00-00	Enabled	Action
3	15 Offer	15OFF	15%	2019-01-01	2019-01-15	Enabled	Action
4	ForTest	Test	10%	0000-00-00	0000-00-00	Enabled	Edit Delete
5	Coupon_Rajendra	ABC123	15%	2018-12-20	2018-12-30	Enabled	Action
6	10% OFF on GROC 2000	GROC	15%	2018-12-20	2018-12-30	Enabled	Action
7	20 offer	20OFF	20%	0000-00-00	0000-00-00	Enabled	Action

At the bottom right of the table, there are buttons for 'Previous', '1', and 'Next'. The status bar at the bottom shows system information like battery level (25%), signal strength, and time (8:02 PM).

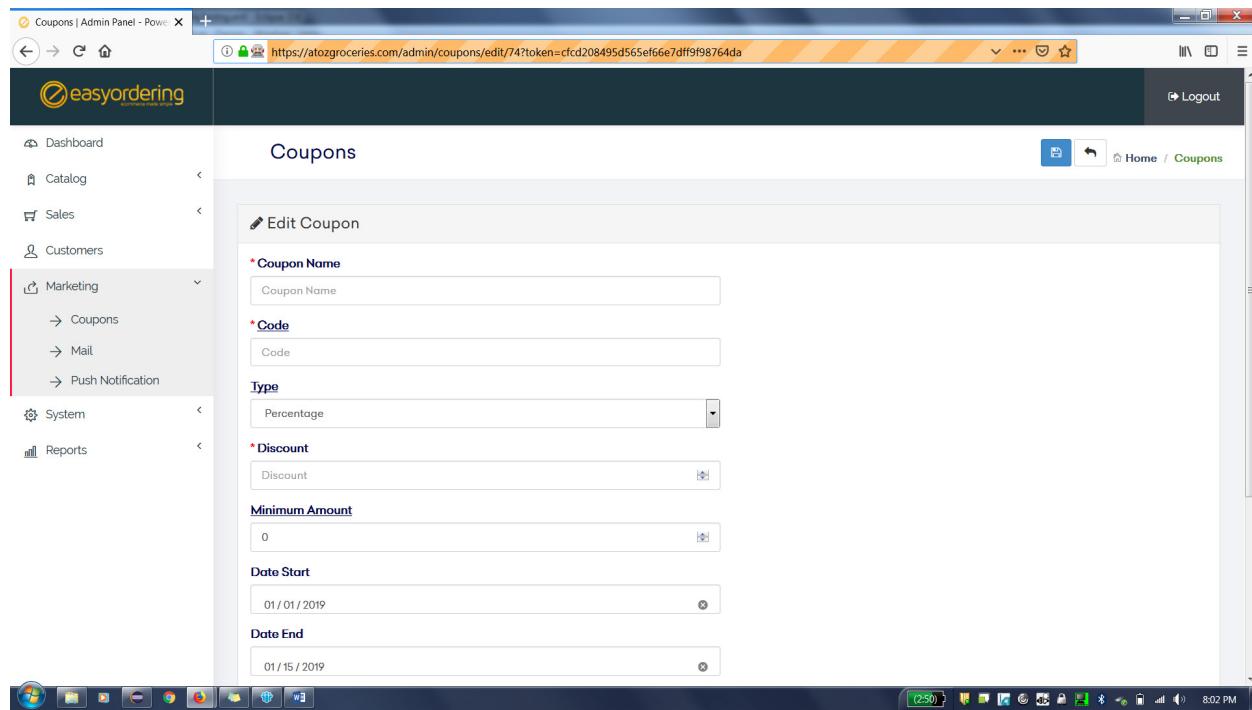
7. Clicking on Edit button will open the fields of the coupon

The screenshot shows the 'Edit Coupon' page. The sidebar on the left is identical to the previous screenshot. The main area is titled 'Edit Coupon' and contains form fields for editing a coupon. The fields include:

- Coupon Name:** 15 Offer
- Code:** 15OFF
- Type:** Percentage
- Discount:** 15
- Minimum Amount:** 0
- Date Start:** 01/01/2019
- Date End:** 01/15/2019

The status bar at the bottom shows system information like battery level (25%), signal strength, and time (8:02 PM).

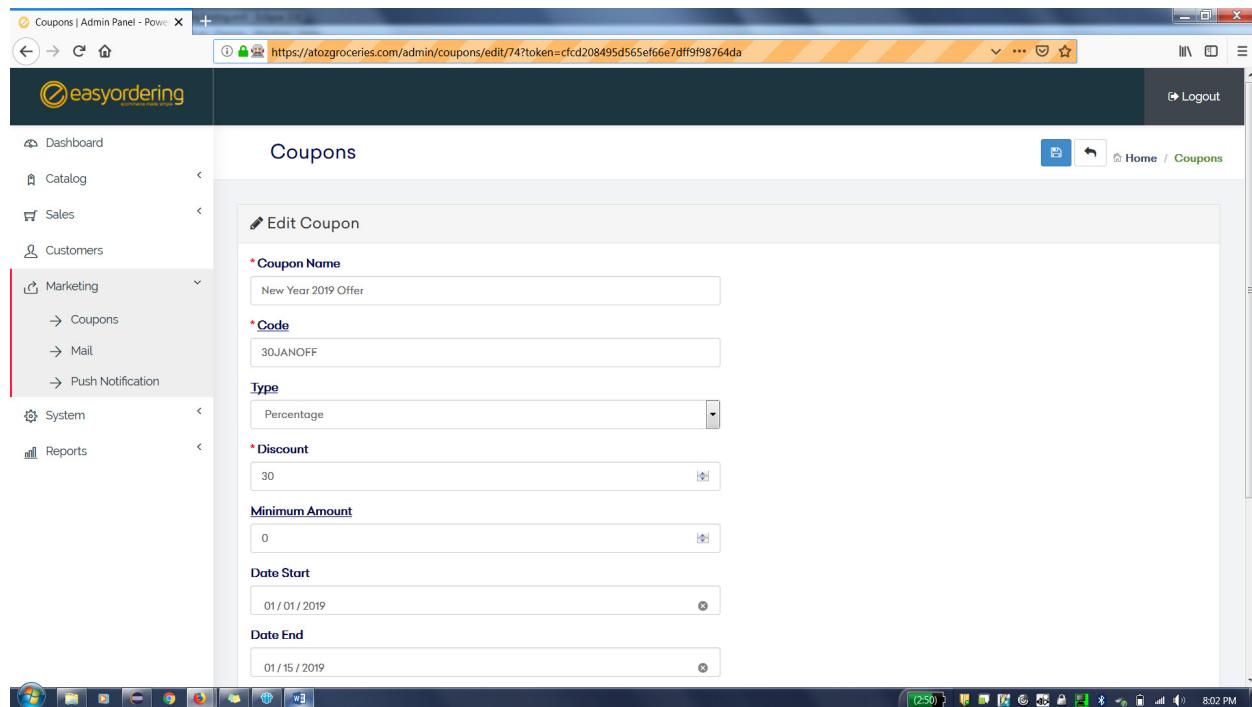
## 8. Clearing the Existing data



The screenshot shows the 'Edit Coupon' page in the easyordering Admin Panel. The coupon details are as follows:

- Coupon Name:** (empty)
- Code:** (empty)
- Type:** Percentage
- Discount:** (empty)
- Minimum Amount:** 0
- Date Start:** 01/01/2019
- Date End:** 01/15/2019

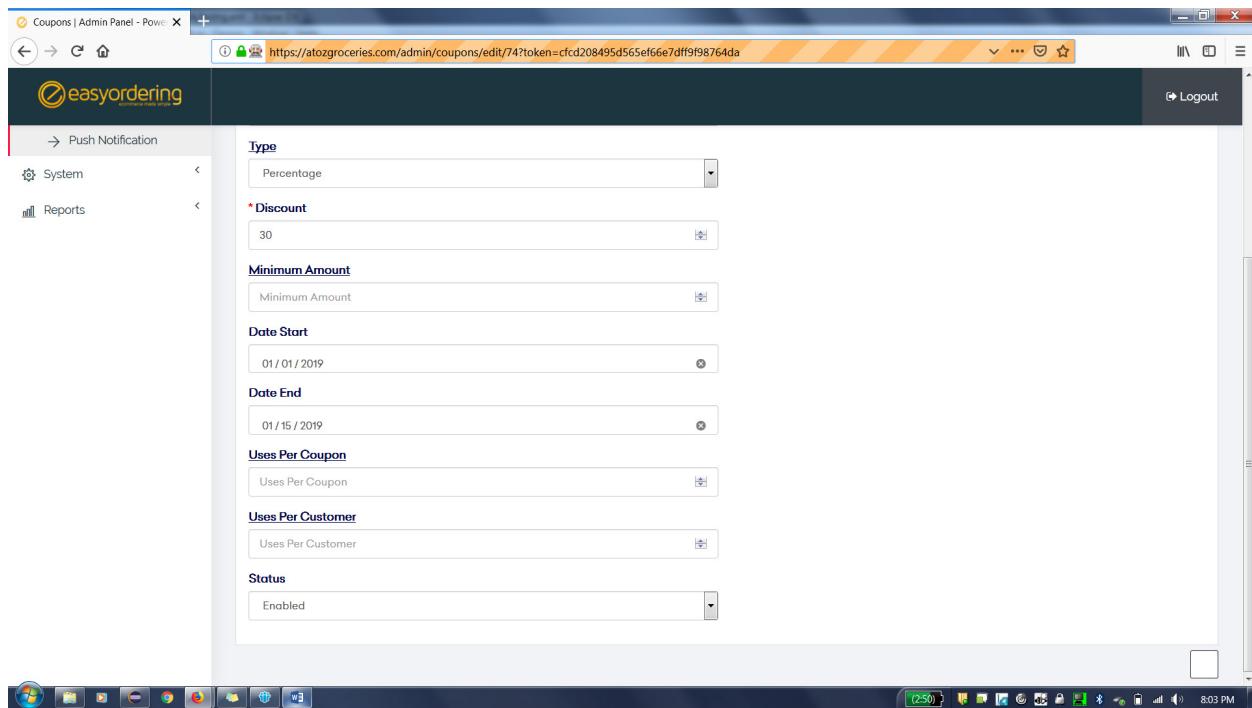
## 9. Sending new data



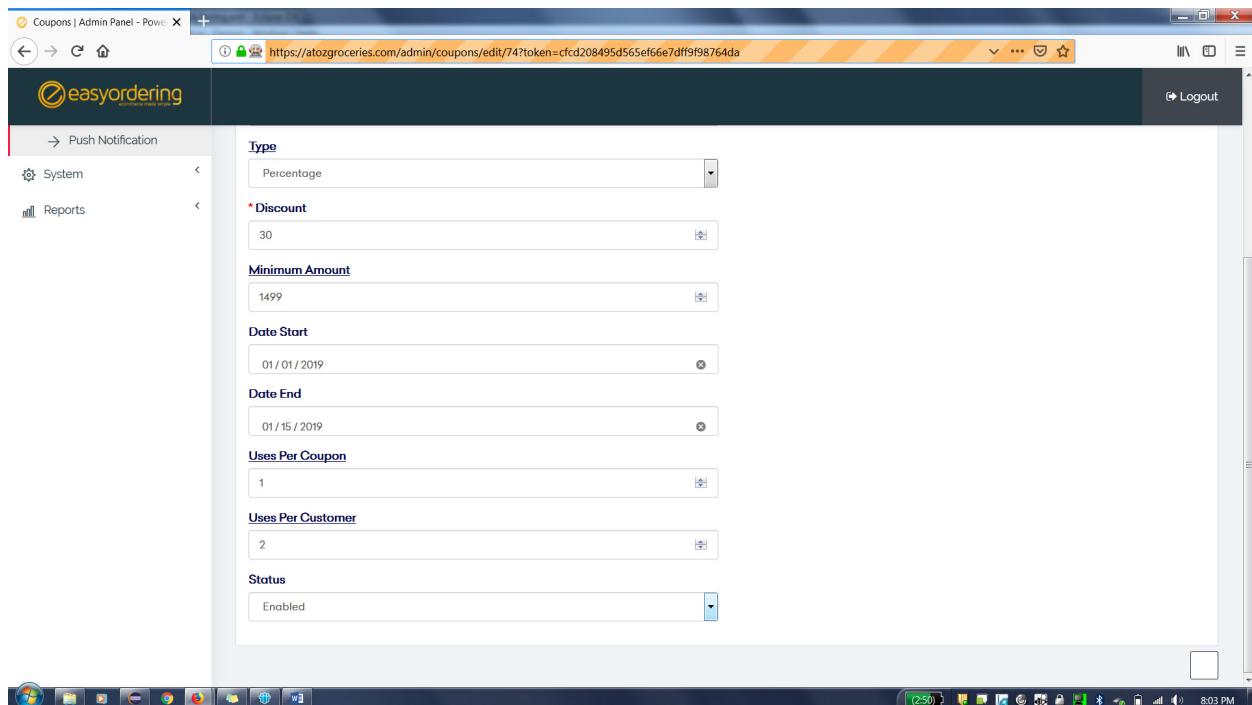
The screenshot shows the 'Edit Coupon' page in the easyordering Admin Panel with the following updated coupon details:

- Coupon Name:** New Year 2019 Offer
- Code:** 30JANOFF
- Type:** Percentage
- Discount:** 30
- Minimum Amount:** 0
- Date Start:** 01/01/2019
- Date End:** 01/15/2019

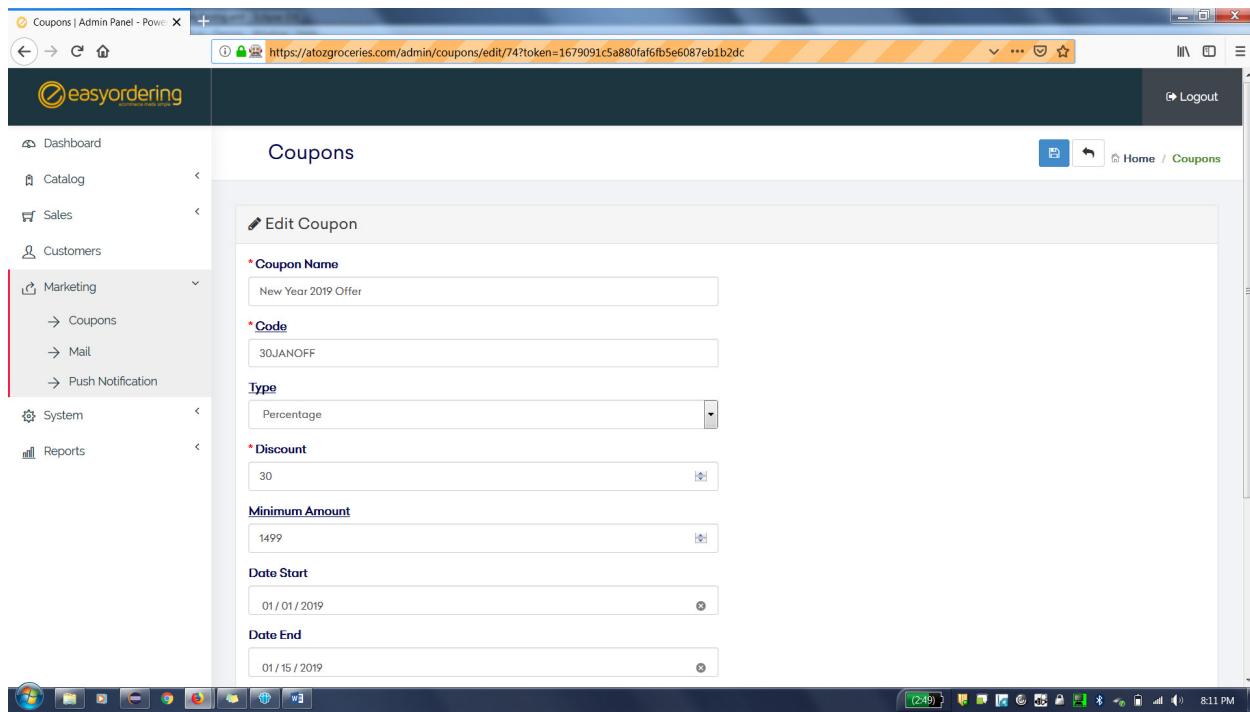
## 10. Scrolling to view Status field for clearing the data of bewlo fields



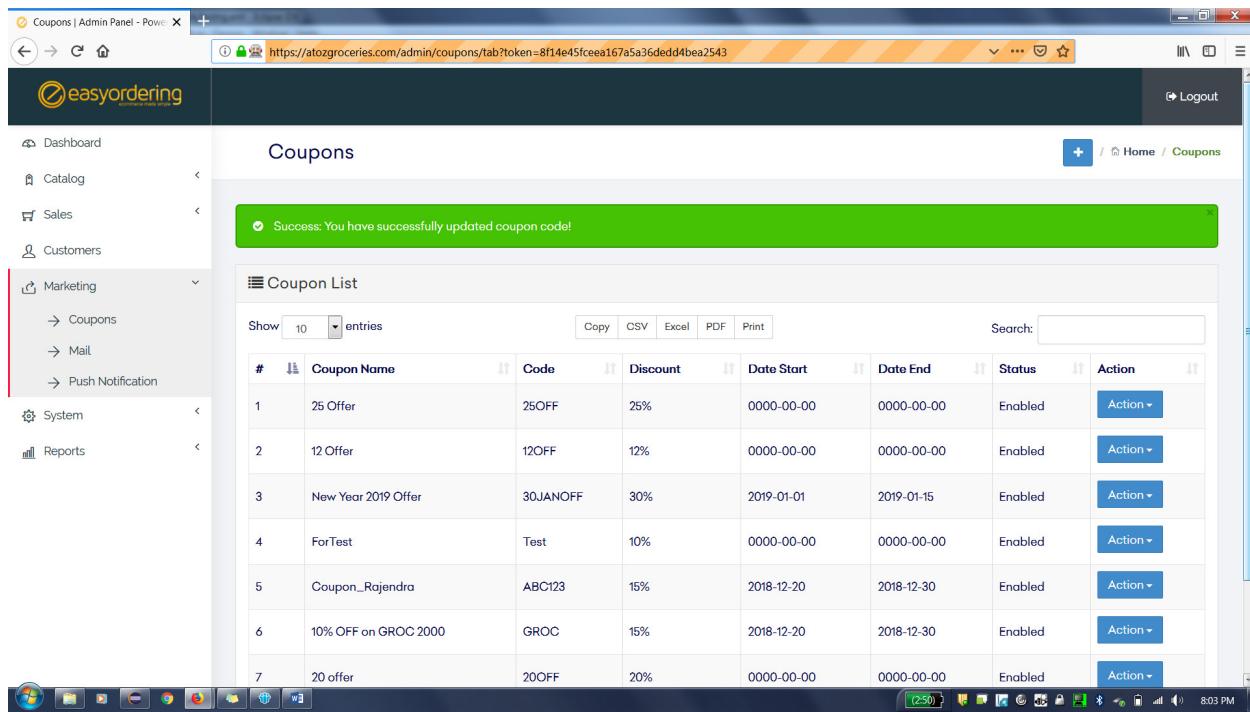
## 11. Sending new data



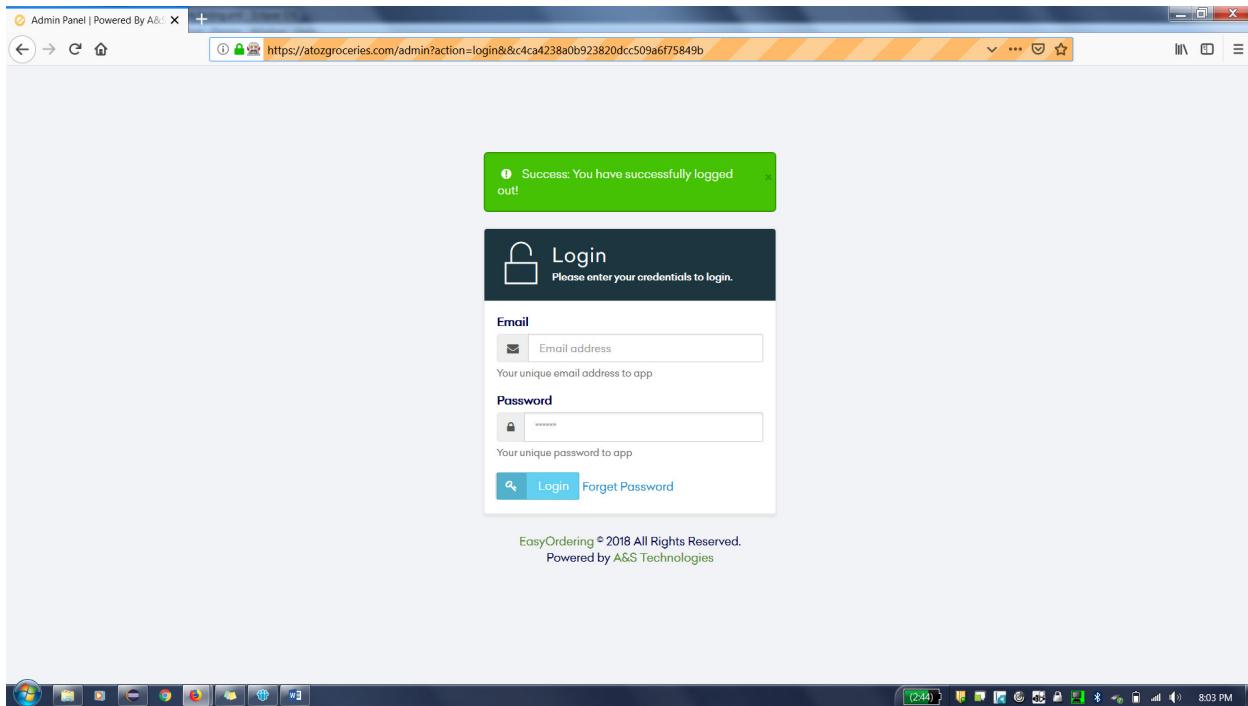
12. Clicking on “To-Top” button for clicking on Save



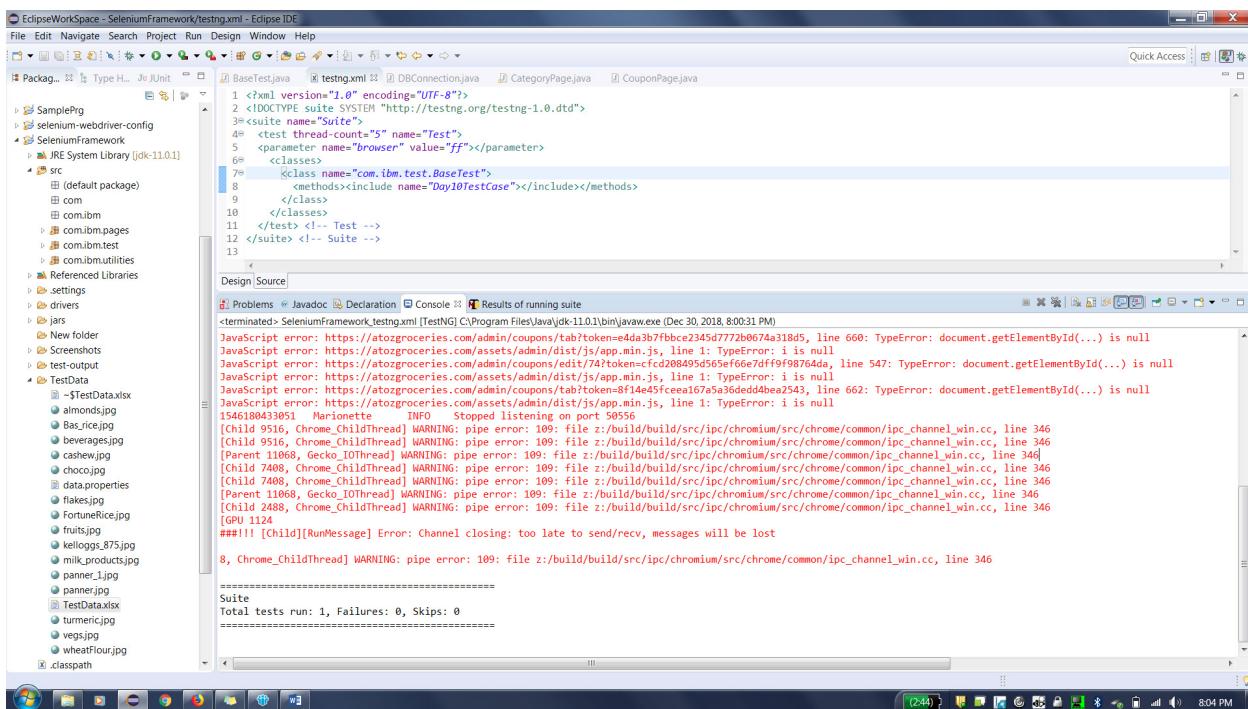
13. Clicking on “Save”, it will redirect to coupons page with successful updation message. Here we are validating the presence of new Coupon also.



## 14. Logging out from the Admin Page



## 15. Output from Console



## 16. Automated Index.html report

The screenshot shows a web browser window with the title "TestNG reports". The URL is "file:///C:/Harisha/Selenium\_Training/EclipseWorkSpace/SeleniumFramework/test-output/index.html#". The page content is a TestNG report for a suite named "Suite".  
**Info:**  
C:\Harisha\Selenium\_Training\EclipseWorkSpace\SeleniumFramework\testing.xml  
test  
0 groups  
Times  
Reporter output  
Ignored methods  
Chronological view  
**Results:**  
1 method, 1 passed  
Passed methods (hide)  
Day10TestCase(15OFF, New Year 2019 Offer, 30JANOFF, 30, 1499, 2019-01-01, 2019-01-15, 1, 2, Enabled)  
  
**Reporter output for Suite**  
Day10TestCase(15OFF, New Year 2019 Offer, 30JANOFF, 30, 1499, 2019-01-01, 2019-01-15, 1, 2, Enabled)  
Assertion on Edited Coupon in Admin Page!! Assertion on total coupons before and after editing coupon!! Assertion on edited coupon in DB!! Assertion on coupon before editing in DB!!



## 17. Automated emailable report

The screenshot shows a web browser window with the title "TestNG Report". The URL is "file:///C:/Harisha/Selenium\_Training/EclipseWorkSpace/SeleniumFramework/test-output/emailable-report.html". The page content is a TestNG report for a suite named "Suite".  
**Test**  
# Passed: 1, # Skipped: 0, # Failed: 0, Time (ms): 50,080  
  
**Suite**  
Test — passed  
com.ibm.test.BaseTest Day10TestCase 1546180847056 41117  
  
**Test**  
com.ibm.test.BaseTest#Day10TestCase  

Parameter #1	Parameter #2	Parameter #3	Parameter #4	Parameter #5	Parameter #6	Parameter #7	Parameter #8	Parameter #9	Parameter #10
15OFF	New Year 2019 Offer	30JANOFF	30	1499	2019-01-01	2019-01-15	1	2	Enabled

  
**Messages**  
Assertion on Edited Coupon in Admin Page!!  
Assertion on total coupons before and after editing coupon!!  
Assertion on edited coupon in DB!!  
Assertion on coupon before editing in DB!!



\*\*\* End of the document \*\*\*