

Day 12 - 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.AdminPage;
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
{
    @Test(testName="PlacingOrder")
    public void Day12TestCase() throws InterruptedException, SQLException,
IOException
    {
        LoginPage login=new LoginPage(driver,wait);
        UserPage user=new UserPage(driver,wait);
        AdminPage admin=new AdminPage(driver,wait);
        DBConnection dbconn=new DBConnection();
        try
        {
            //getting the data from the properties file using Hash map
            String adminurl=hashData.get("adminurl");
            String username=hashData.get("username");
            String password=hashData.get("password");
            String userurl=hashData.get("userurl");
```

```

String fullname=hashData.get("fullname");
String userphno=hashData.get("userphno");
String userpwd=hashData.get("userpwd");
String confirmpwd=hashData.get("confirmpwd");
String proname1=hashData.get("proname1");
String proname2=hashData.get("proname2");
String useremail=hashData.get("useremail");
String useradd=hashData.get("useradd");
String usercity=hashData.get("usercity");
String userpin=hashData.get("userpin");
//Navigating to User page
driver.get(userurl);

//Signing up to the User page
login.signUp(fullname, userphno, userpwd, confirmpwd);

//Adding Products to the cart
user.AddProductstoCart(proname1,proname2,useremail,useradd,usercity,userpin);

//Validating the orders in My Account -> My Orders
String orderid=user.validateMyOrders(fullname);

//Logging out from user page
user.ClickLogout();

//Navigating to the admin page
driver.get(adminurl);

//Logging into the admin page
login.EnterEmail(username);
login.EnterPassword(password);
login.clickLogin();

//Opening Sales
admin.openSales();

//Validating the order in Orders of Admin portal
String orderNum=admin.validateOrders(orderid, fullname);

//Validating the orders in Database
dbconn.DBOrderValidate(orderNum, fullname);
Reporter.Log("Order placed by "+fullname+" is available in Database!!");

}

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/Day12_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.ExpectedConditions;
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;

    @FindBy(xpath="//a[@class='register']")
    WebElement signUpEle;

    @FindBy(xpath="//input[@id='tccheckbox']")
    WebElement chkboxEle;

    @FindBy(id="name")
    WebElement fullnameEle;

    @FindBy(id="pnum")
    WebElement phoneEle;

    @FindBy(id="password")
    WebElement userPwdEle;

    @FindBy(id="cpassword")
    WebElement confirmPwdEle;

    @FindBy(xpath="//button[@id='mem_signup']")
    WebElement signButEle;
}

```

```

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();
}

//Signing up into the User Page
public void signUp(String name, String phno, String pwd, String conPwd)
throws InterruptedException
{
    signUpEle.click();
    fullnameEle.sendKeys(name);
    phoneEle.sendKeys(phno);
    userPwdEle.sendKeys(pwd);
    confirmPwdEle.sendKeys(conPwd);
    chkboxEle.click();
    signButEle.click();
    Thread.sleep(5000);
    wait.until(ExpectedConditions.alertIsPresent());
    driver.switchTo().alert().accept();
    Thread.sleep(5000);
}
}

```

AdminPage.java

```

package com.ibm.pages;

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.WebDriverWait;
import org.testng.Assert;

```

```

import org.testng.Reporter;

public class AdminPage {
    @FindBy(xpath="//a[contains(text(),'Sales')]")
    WebElement salesEle;
    @FindBy(xpath="//a[contains(text(),'Orders')]")
    WebElement ordersEle;

    WebDriver driver;
    WebDriverWait wait;
    JavascriptExecutor js;
    public AdminPage(WebDriver driver,WebDriverWait wait)
    {
        this.driver=driver;
        this.wait=wait;
        js=(JavascriptExecutor) driver;
        PageFactory.initElements(driver, this);
    }
    //Opening Sales
    public void openSales()
    {
        salesEle.click();
    }
    //Validating Order in Orders of Admin Portal
    public String validateOrders(String order,String uname)
    {
        String orderno=null;
        ordersEle.click();
        int rowcount=driver.findElements(By.xpath("//table[@class='table table-bordered table-striped table-hover']/tbody/tr")).size();
        for(int i=1;i<=rowcount;i++)
        {
            String orderid=driver.findElement(By.xpath("//table[@class='table table-bordered table-striped table-hover']/tbody/tr["+i+"]/td[2]")).getText().trim();
            if (orderid.equals(order))
            {
                orderno=driver.findElement(By.xpath("//table[@class='table table-bordered table-striped table-hover']/tbody/tr["+i+"]/td[3]")).getText().trim();
                Assert.assertEquals(orderid,order);
                Reporter.log("Order placed by "+uname+" is available in Admin Page -> Sales -> Orders with Order No "+orderno);
                break;
            }
        }
        return orderno;
    }
}

```

UserPage.java

```

package com.ibm.pages;

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

public class UserPage
{
    //Finding the elements of the User Page
    @FindBy(xpath="(/span[@class='click-categories flaticon-bars'])[2]")
    WebElement catBarEle;
    @FindBy(xpath="//*[@class='category-drop-list-inner'][2]")
    WebElement CateList;
    @FindBy(xpath="/html/body/div[1]/div[5]/div/div[1]/div/nav/ul/li[2]/span")
    WebElement HdrEle;
    @FindBy(xpath="//input[@id='search-box']")
    WebElement mainSrchEle;
    @FindBy(xpath="//a[contains(text(),'Continue to payment')]")
    WebElement continuePayEle;
    @FindBy(id="email")
    WebElement userEmailEle;
    @FindBy(id="address")
    WebElement userAddEle;
    @FindBy(id="city")
    WebElement userCityEle;
    @FindBy(id="pincode")
    WebElement userPinEle;
    @FindBy(xpath="//span[contains(text(),'order has been placed')]")
    WebElement orderPlacedEle;
    @FindBy(xpath="//input[@id='tc']")
    WebElement termsChkEle;
    //@FindBy(xpath="//a[@onclick='myAccount()']")
    @FindBy(xpath="/html/body/header/div[1]/div/div[2]/div[3]/li/a")
    WebElement myAccEle;
    @FindBy(xpath="(/a[contains(text(),'Log Out')])[1]")
    WebElement userLogout;
    WebDriver driver;
    WebDriverWait wait;
    JavascriptExecutor js;
    //Constructor
    public UserPage(WebDriver driver,WebDriverWait wait)
    {
        this.driver=driver;
        this.wait=wait;
        js=(JavascriptExecutor) driver;
        PageFactory.initElements(driver, this);
    }
    //this method validates the category presence in User Page under "Shop By Category"
    public void CateryValidation(String catename) throws InterruptedException
    {

```

```

        //clicking on the bars of Shop by category
        catBarEle.click();
        wait.until(ExpectedConditions.visibilityOf(CateList));
        //opening the category created
        WebElement newcateEle=driver.findElement(By.partialLinkText(catename));
        Thread.sleep(5000);
        newcateEle.click();
        //passing the header text to a string for validation
        String validCate=HdrEle.getText().trim();
        Assert.assertTrue(validCate.contains(catename), "Assertion on new
category in User Page!!!");
        Reporter.log("Validation of Category header in User Page!!!");
    }
    //This method validates the edited Phone number present in the footer
information in User Page
    public void footerPhnoValidation(String Phno) throws InterruptedException
    {
        WebElement
footPhno=driver.findElement(By.xpath("//p[contains(text(), '"+Phno+"')]"));
        js.executeScript("arguments[0].scrollIntoView(true);", footPhno);
        String footerPh=footPhno.getText().trim();
        Assert.assertTrue(footerPh.contains(Phno), "Assertion on Phone number in
footer section!!!");
        Reporter.log("Assertion on Phone number in footer section!!!");
    }
    //This method validates the edited Phone number present in the header
information in User Page
    public void headerPhnoValidation(String Phno) throws InterruptedException
    {
        WebElement hdrPhno=driver.findElement(By.xpath("(//div[@class='header-
mid-right-content'])[1]"));
        js.executeScript("arguments[0].scrollIntoView(true);", hdrPhno);
        String headerPh=hdrPhno.getText().trim();
        Assert.assertTrue(headerPh.contains(Phno), "Assertion on Phone number in
header section!!!");
        Reporter.log("Assertion on Phone number in header section!!!");
    }
    //this method validates the edited Product in User page
    public void ProductsValidation(String ProNewName) throws InterruptedException
    {

        mainSrchEle.click();
        //Thread.sleep(10000);
        //sending keys in the search

        driver.findElement(By.xpath("//*[@id='searchModal']/div/div/div[1]/input"))
.sendKeys(ProNewName);
        Thread.sleep(5000);
        //selecting the product from the dropdown

        driver.findElement(By.xpath("//span[text()='"+ProNewName+"' ]/ancestor::div[@id
='searchproducts-div']")).click();
        String validProduct=HdrEle.getText().trim();
        Assert.assertTrue(validProduct.contains(ProNewName), "Assertion on
Edited Product in User Page!!!");
    }

```

```

        Reporter.log("Assertion on Edited Product in User Page!!");
    }
    //Adding 2 Products to the cart
    public void AddProductstoCart(String ProName1,String ProName2,String
userEmail,String userAdd,String userCity,String userPin) throws InterruptedException
    {
        mainSrchEle.click();
        Thread.sleep(5000);
        //Adding Product 1 to cart

        driver.findElement(By.xpath("//*[@id='searchModal']/div/div/div/div[1]/input"))
).sendKeys(ProName1);
        Thread.sleep(5000);

        driver.findElement(By.xpath("//span[text()='"+ProName1+"']/ancestor::div[@id='
searchproducts-div']")).click();
        driver.findElement(By.xpath("//a[contains(text(),'Add To
Cart')]")).click();
        //Adding Product 2 to cart
        mainSrchEle.click();
        Thread.sleep(5000);

        driver.findElement(By.xpath("//*[@id='searchModal']/div/div/div/div[1]/input"))
).sendKeys(ProName2);
        Thread.sleep(5000);

        driver.findElement(By.xpath("//span[text()='"+ProName2+"']/ancestor::div[@id='
searchproducts-div']")).click();
        driver.findElement(By.xpath("//a[contains(text(),'Add To
Cart')]")).click();
        //validation of products added in cart

        driver.findElement(By.xpath("/html/body/header/div[2]/div/div[3]/div/div/a
")).click();
        String
Pro1InCart=driver.findElement(By.xpath("//div[contains(text(),'"+ProName1+"')]")).get
Text().trim();
        String
Pro2InCart=driver.findElement(By.xpath("//div[contains(text(),'"+ProName2+"')]")).get
Text().trim();
        //Assertion on products in cart
        Assert.assertTrue(Pro1InCart.contains(ProName1),"Product 1 added in cart
is available!!!");
        Assert.assertTrue(Pro2InCart.contains(ProName2),"Product 1 added in cart
is available!!!");
        //printing the log
        Reporter.log("Products "+ProName1+" and "+ProName2+" are available in
cart!!!");
        //placing order
        driver.findElement(By.xpath("//a[contains(text(),'Check
Out')]")).click();
        userEmailEle.sendKeys(userEmail);
        userAddEle.sendKeys(userAdd);
        js.executeScript("arguments[0].scrollIntoView(true);",continuePayEle);
        Select citySel=new Select(userCityEle);

```

```

        citySel.selectByVisibleText(userCity);
        userPinEle.sendKeys(userPin);
        continuePayEle.click();
        js.executeScript("arguments[0].scrollIntoView(true);", termsChkEle);
        termsChkEle.click();
        driver.findElement(By.xpath("//a[@id='confirm-order-id']")).click();
        //js.executeScript("arguments[0].click()", myAccEle);

        Thread.sleep(5000);
    }
    //validating the order in My Account -> My Orders
    public String validateMyOrders(String uname)
    {
        String orderid=null;
        myAccEle.click();
        driver.findElement(By.xpath("//a[contains(text(),'My
Orders')]")).click();
        int rowLen=driver.findElements(By.xpath("//table[@class='table table-
bordered table-hover']/tbody/tr")).size();
        for(int i=1;i<=rowLen;i++)
        {
            String
userName=driver.findElement(By.xpath("//table[@class='table table-bordered table-
hover']/tbody/tr["+i+"]/td[2]")).getText().trim();
            if(userName.equals(uname))
            {
                orderid=driver.findElement(By.xpath("//table[@class='table
table-bordered table-hover']/tbody/tr["+i+"]/td[1]")).getText().trim();
                Assert.assertEquals(userName, uname);
                Reporter.log("Order placed by "+uname+" is available in My
Account -> My Orders with order id as "+orderid);
                break;
            }
        }
        return orderid;
    }

    //method for User Logout
    public void ClickLogout()
    {
        myAccEle.click();
        userLogout.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 {

    //method to validate category in Database
    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 new 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 category name when matches the name of
            the category in DB

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

//validating the order placed in Database
public void DBOrderValidate(String orderno,String uname) 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_order`"
WHERE order_no="""+orderno+"\"");
    while(rs.next())
    {

        Assert.assertEquals(orderno,rs.getString("order_no"));
    }
}
}

```

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
fullname=HarryTest3
userphno=9000090003
userpwd=welcome
confirmPWD=welcome
proname1=Bislery Water
proname2=Basmathi Rice
useremail=harisha.sunny@gmail.com
useradd=Bangalore
usercity=MADIWALA
userpin=560029
```

testng.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
```

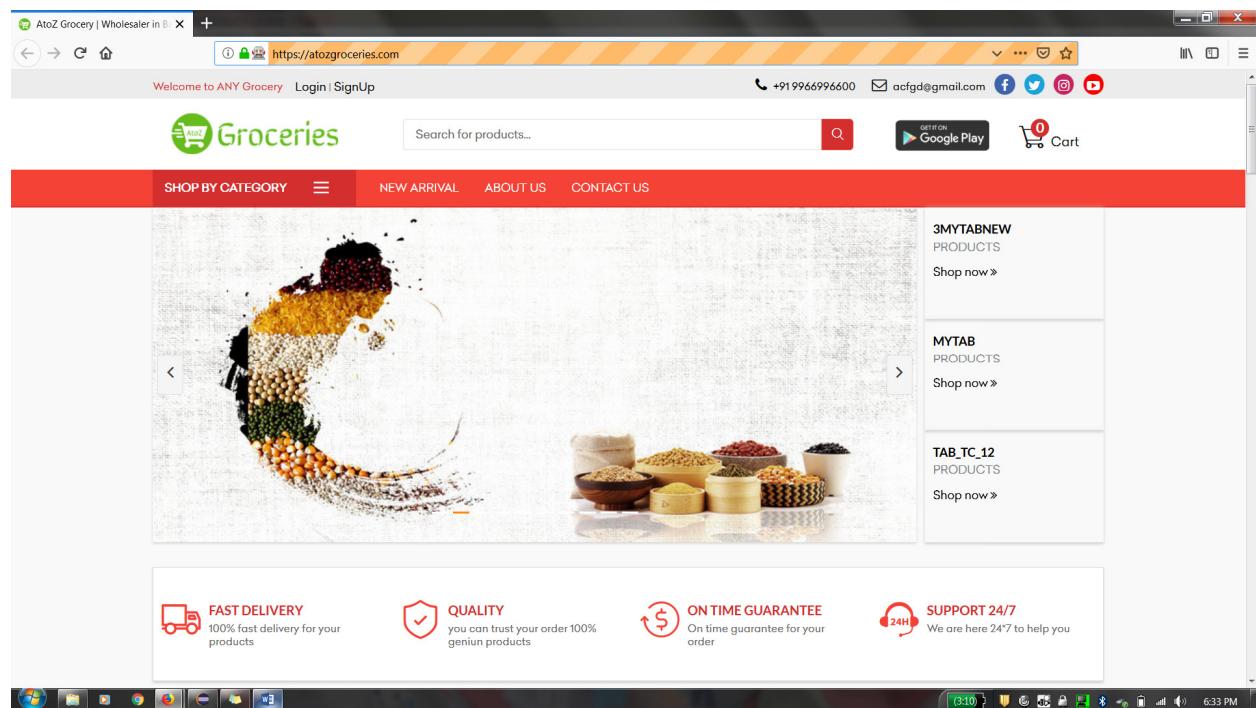
```

<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 -->

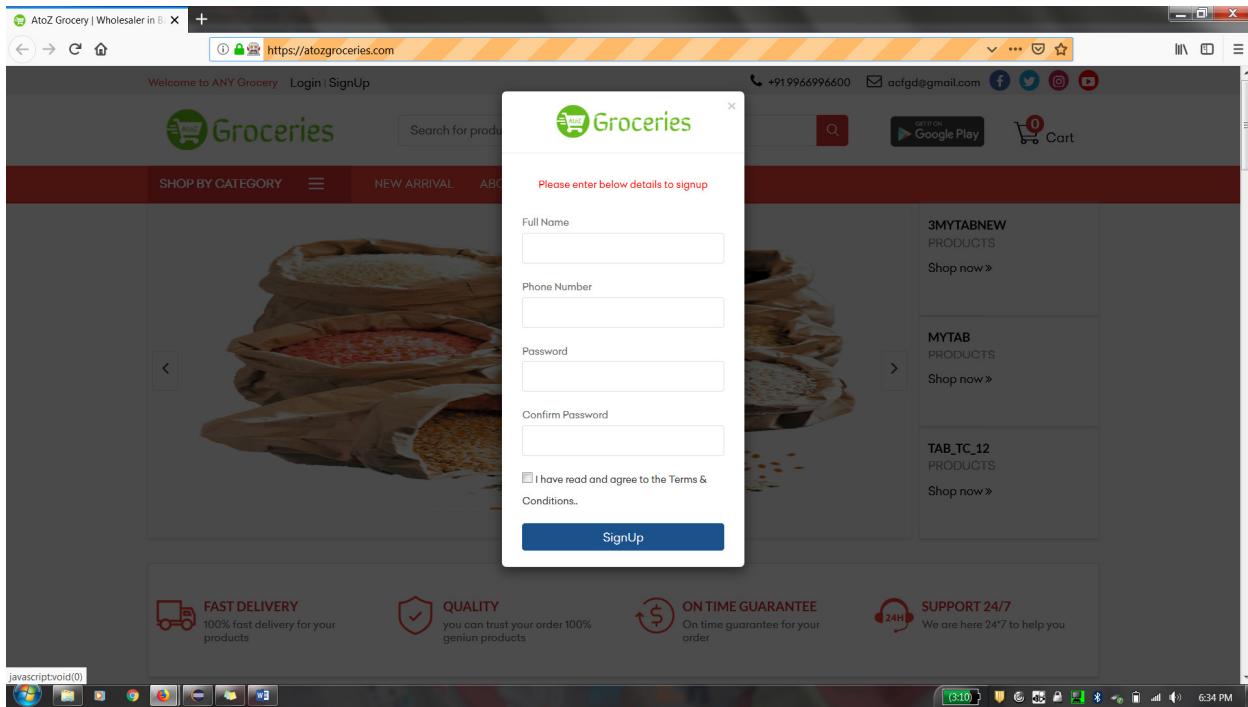
```

Step by step output:

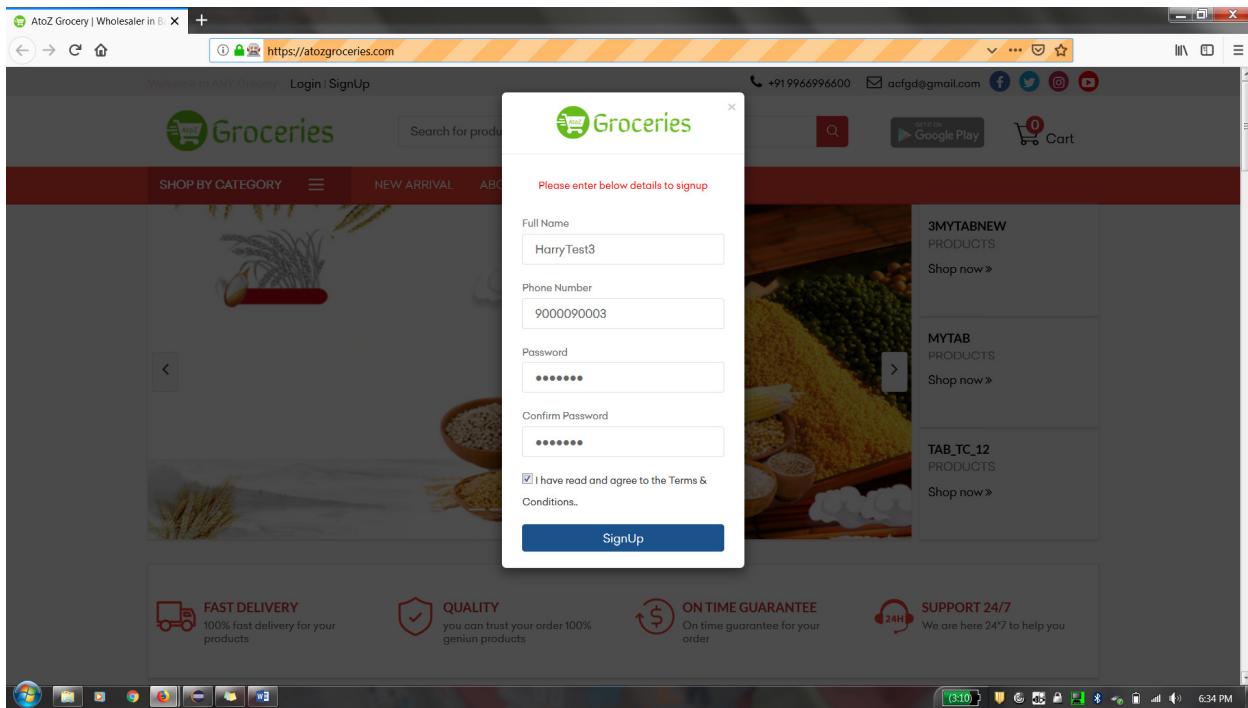
Opening User page



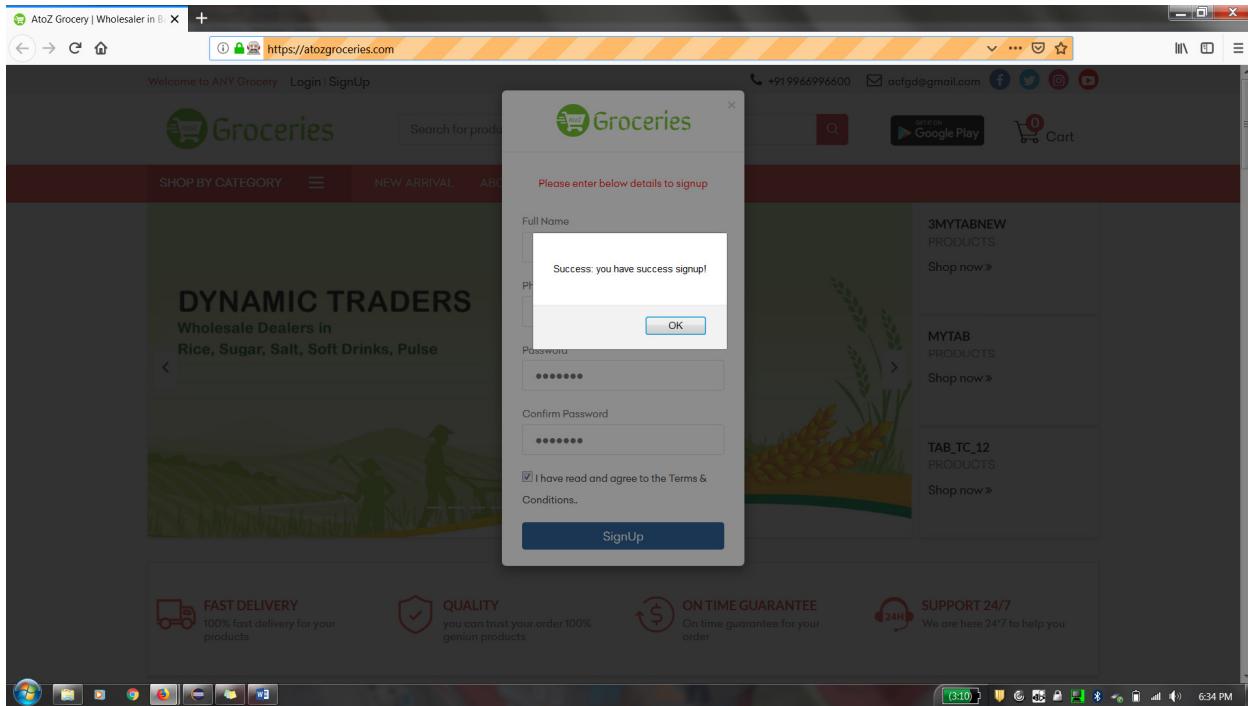
Clicking on Signup



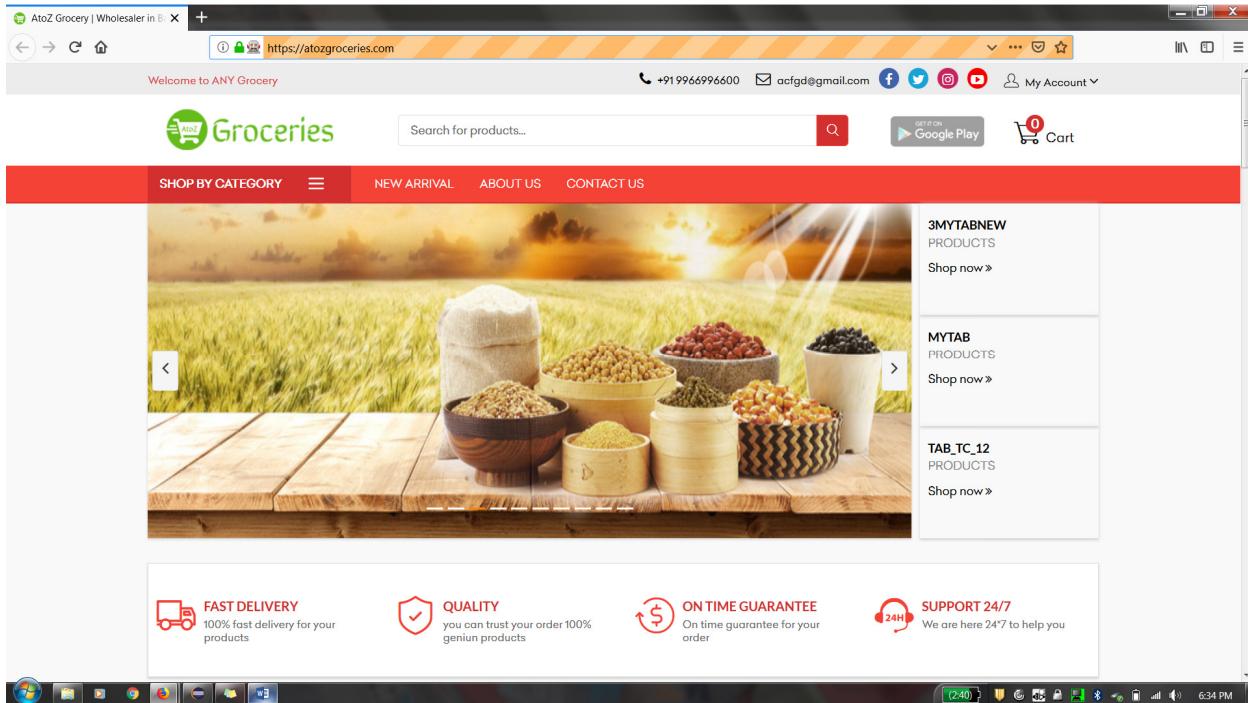
Providing input data of User for signup

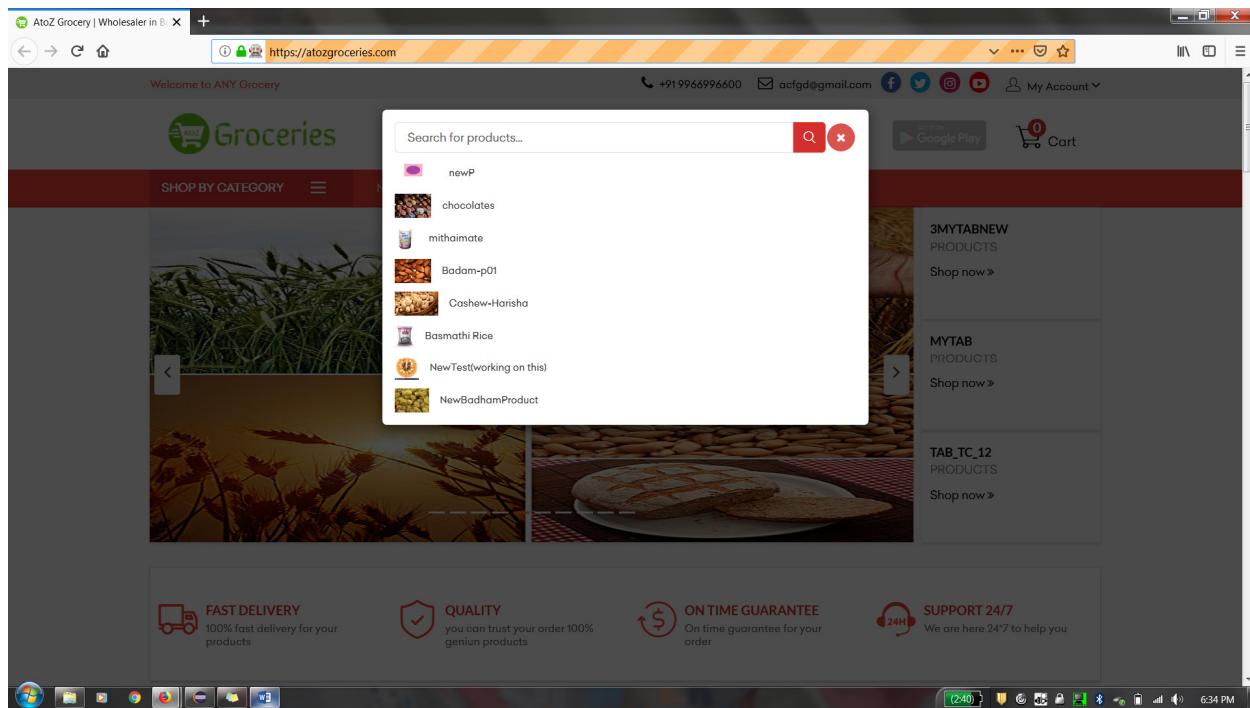


Success message after Signup

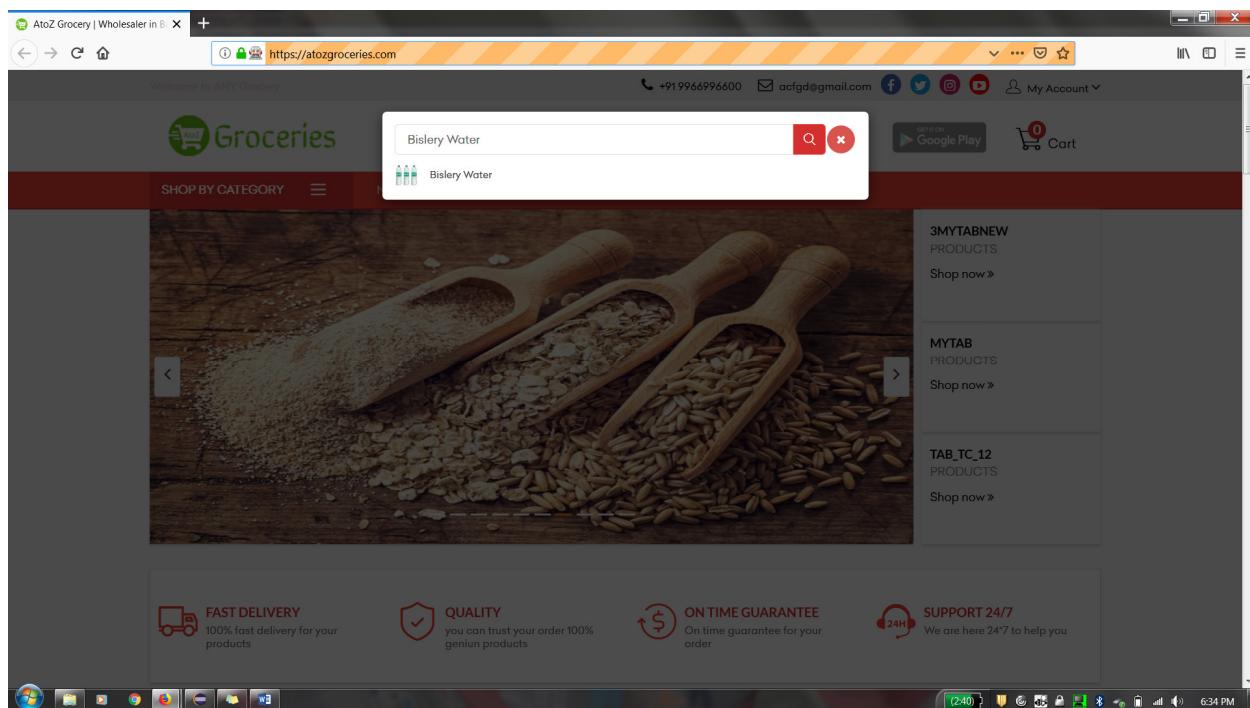


User page after signing up and clicking on Search for products





Searching for the Product 1 to add in cart



Opening the product to add in cart

The screenshot shows a web browser window for 'Bisley Water | AtoZ Grocery'. The URL is https://atozgroceries.com/common/productdetails/336. The page displays a product image of three Bisley water bottles. A green '6% OFF' badge is visible above the product name 'Bisley Water'. Below the badge, it says '25test1', 'Discounted price: ₹19.00', 'MRP: ₹20.00', and 'Availability: In stock'. There is an 'Add To Cart' button. The page also includes sections for 'QUICK OVERVIEW' and 'RELATED PRODUCTS'.

Success message after adding in cart

The screenshot shows the same web browser window after the product has been added to the cart. A green success message at the top of the page reads 'Successfully added Bisley Water 25 test1 to the basket'. The rest of the page content is identical to the first screenshot, showing the product details and related products.

Searching for product 2 to add in cart

The screenshot shows a web browser window for 'Bisley Water | AtoZ Grocery'. A search results overlay is displayed, listing various products under the heading 'Search for products...'. The products listed are: newP, chocolates, mithaimate, Badam-p01, Cashew-Harisha, Basmathi Rice, NewTest(working on this), and NewBadhamProduct. Each item has a small thumbnail image next to its name. The main page background shows three Bisley water bottles.

The screenshot shows a web browser window for 'Bisley Water | AtoZ Grocery'. The product details page for 'Bisley Water' is displayed. The product is shown in a large image with a '6% OFF' badge. The product name is 'Bisley Water' and the discounted price is ₹19.00. The MRP is ₹20.00. Availability status is 'In stock'. An 'Add To Cart' button is visible. The main page background shows three Bisley water bottles.

Opening the product 2 to add in cart

The screenshot shows a web browser window for 'Basmati Rice | AtoZ Grocery'. The URL is https://atozgroceries.com/common/productdetails/353. The page displays a product image of 'INDIA GATE BASMATI RICE Teat Dazzles', a discounted price of ₹162.00 (from ₹180.00), and a '10% OFF' badge. A user named '1test1' has added the item to the cart. The 'Add To Cart' button is visible. Below the main product, there's a 'QUICK OVERVIEW' section and a 'RELATED PRODUCTS' section with thumbnails for other items like '10% OFF' and '5% OFF' offers.

Success message after adding the product in cart

The screenshot shows the same product details page for Basmati Rice. A green success message banner at the top states 'Successfully added Basmati Rice 1 test1 to the basket'. The rest of the page content is identical to the previous screenshot, including the product image, price, discount badge, user information, and related products section.

Opening the cart to validate the 2 added products

The screenshot shows a web browser displaying the AtoZ Groceries website. The main content area shows a product detail page for 'Basmati Rice' with a discounted price of ₹162.00. Below this, there is a section for 'RELATED PRODUCTS' featuring other grocery items like 'INDIA GATE BASMATI RICE' and 'DYNAMIC TRADERS'. To the right, a modal window titled 'My Cart (26 items)' is open, showing a list of items with their names, quantities, and prices. The cart contains two items: 'Bisleri Water' (Quantity: 25) and 'Basmati Rice' (Quantity: 1). The total sub-total for the cart is ₹637.00.

Image	Name	Price
	Bisleri Water Quantity: 25	₹19.00 ₹20.00
	Basmati Rice Quantity: 1	₹162.00 ₹180.00

Doing checkout

The screenshot shows the AtoZ Groceries website at the checkout stage. Step 1, 'Phone Number Verification', has been completed successfully, as indicated by a green checkmark next to the entered phone number '+919000090003'. Step 2, 'Delivery Address', is the current step, prompting the user to enter their full name ('HarryTest3') and email address. To the right, the shopping cart is visible, showing the same two items: Bisleri Water and Basmati Rice, with a total sub-total of ₹3,657.00.

Providing input for address

The screenshot shows the AtoZ Groceries website's checkout process. Step 1: Phone Number Verification is completed with the number +919000090003. Step 2: Delivery Address is being filled out with Full Name (HarryTest3) and Email Address (harisha.sunny@gmail.com). The address field contains Bangalore. To the right, the cart summary shows 26 items with a subtotal of ₹3,657.00.

Item	Quantity	Price
Bisley Water	25	₹19.00 ₹20.00
Basmati Rice	1	₹162.00 ₹180.00

Sub Total: ₹3,657.00

Clicking on “Continue to payment”

The screenshot shows the continuation of the checkout process. The delivery address information has been submitted. The cart summary still shows 26 items with a subtotal of ₹3,657.00. Below the address fields, City (MADIWALA) and Pincode (560029) are entered. A red "Continue to payment" button is visible at the bottom left. Step 3: Payment Information is indicated at the bottom.

Item	Quantity	Price
Rice	1	₹180.00

Sub Total: ₹3,657.00

Making payment and clicking on comfirm order

The screenshot shows the payment information step of the checkout process. It offers two payment methods: COD (Cash On Delivery) and PayU money. Below these options is a checkbox for accepting terms and conditions. A red 'Confirm Order' button is at the bottom.

3 Payment Information

COD (Cash On Delivery)
or
 Debit / Credit Card

I have read and agree to the Terms & Conditions.

Confirm Order

This screenshot is identical to the one above, showing the payment information step with COD and PayU money options, a terms and conditions checkbox, and a 'Confirm Order' button.

3 Payment Information

COD (Cash On Delivery)
or
 Debit / Credit Card

I have read and agree to the Terms & Conditions.

Confirm Order

This screenshot is identical to the previous ones, showing the payment information step with COD and PayU money options, a terms and conditions checkbox, and a 'Confirm Order' button.

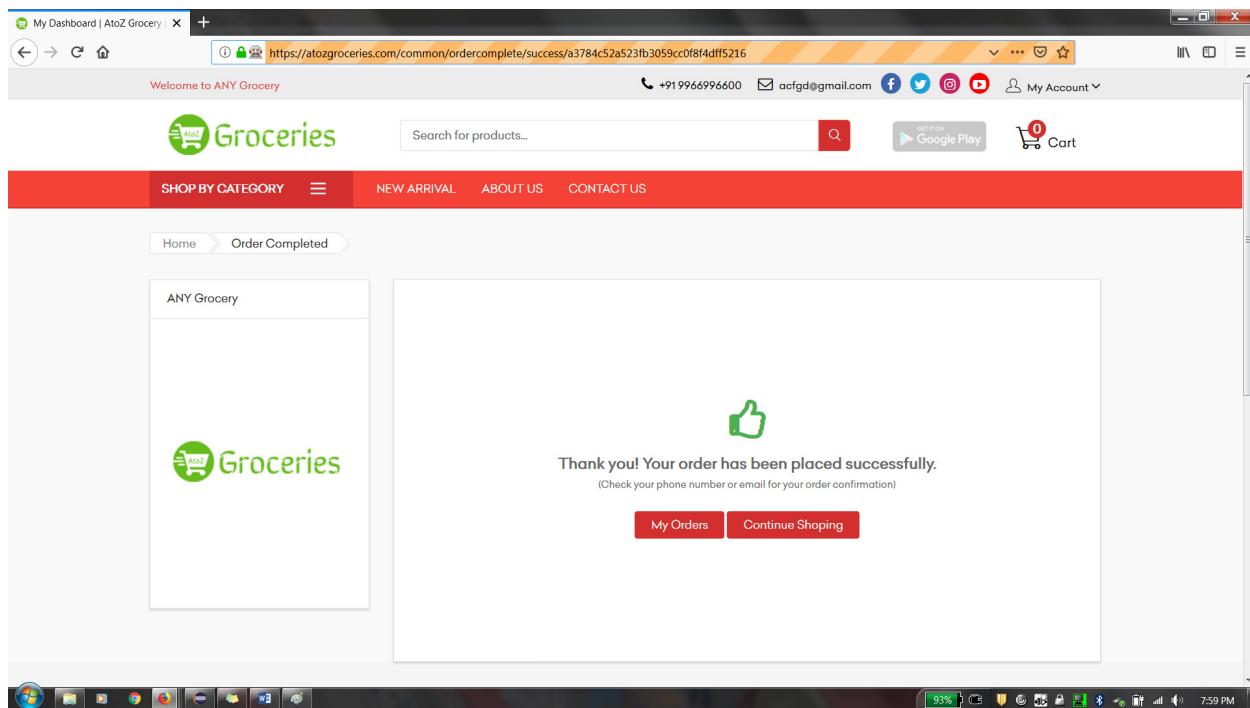
3 Payment Information

COD (Cash On Delivery)
or
 Debit / Credit Card

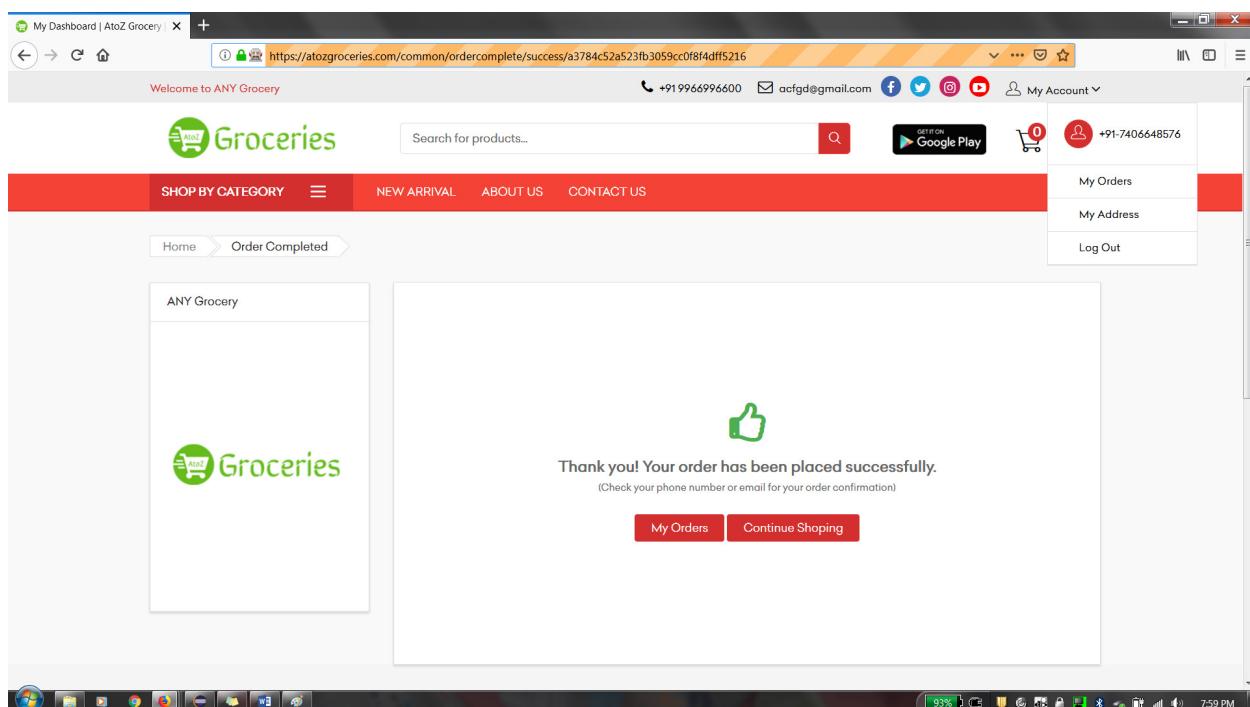
I have read and agree to the Terms & Conditions.

Confirm Order

Confirmation after placing order



Validating order placed under My Account -> My Orders



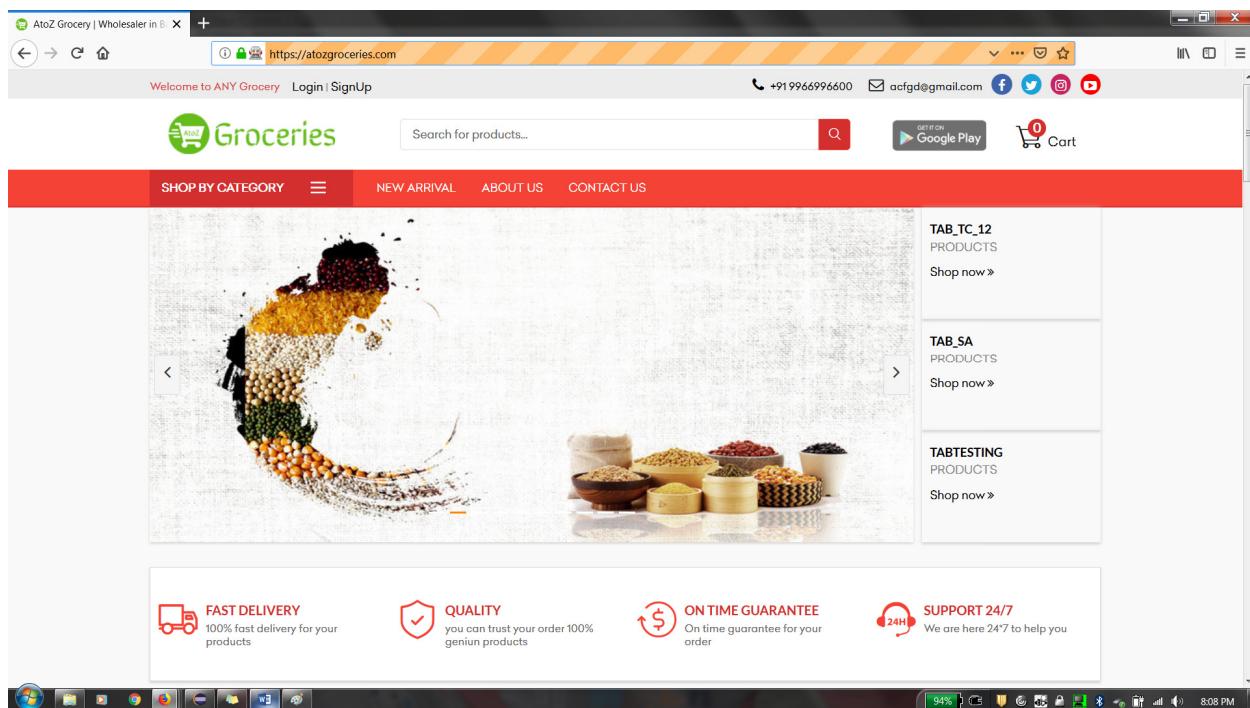
Validating the order in My Orders

The screenshot shows the AtoZ Grocery user dashboard. At the top, there's a header with a phone number (+91 9966996600), an email (acfgd@gmail.com), and social media links. Below the header is a navigation bar with 'SHOP BY CATEGORY', 'NEW ARRIVAL', 'ABOUT US', and 'CONTACT US'. The main content area has a breadcrumb trail 'Home > My Dashboard / My Orders'. On the left, a sidebar shows contact information (+91 9000090003) and links for 'My Orders' (highlighted in green), 'My Address', and 'Log Out'. The right side displays a table of orders. One order is listed: Order ID #49, Customer HarryTest3, 2 products, Status Pending, Total ₹3,637.00, Date Added 2019-01-03. There's also a 'View' button next to the row.

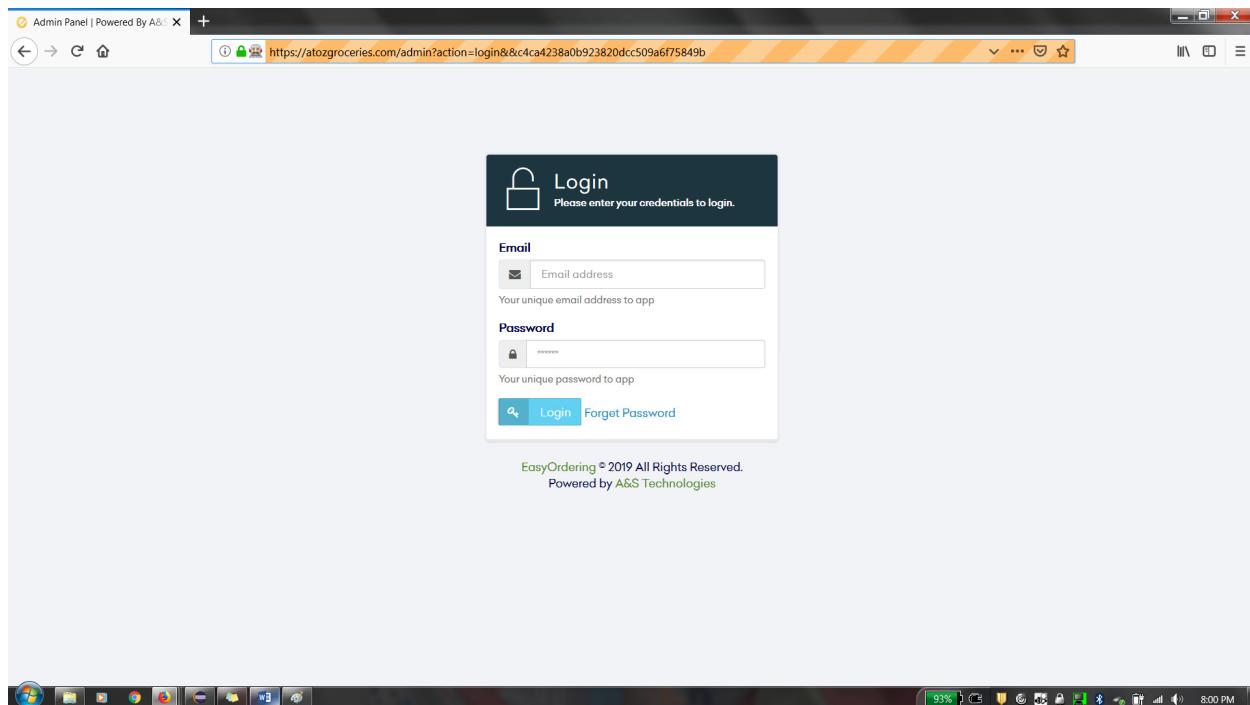
Logging out from the user page

This screenshot shows the same user dashboard after logging out. The contact information in the sidebar now shows a different phone number: +91 7406648576. The order table also shows a different order: Order ID #50, Customer HarryTest3, 2 products, Status Pending, Total ₹3,637.00, Date Added 2019-01-03. The rest of the interface remains the same, including the header, navigation bar, and footer map.

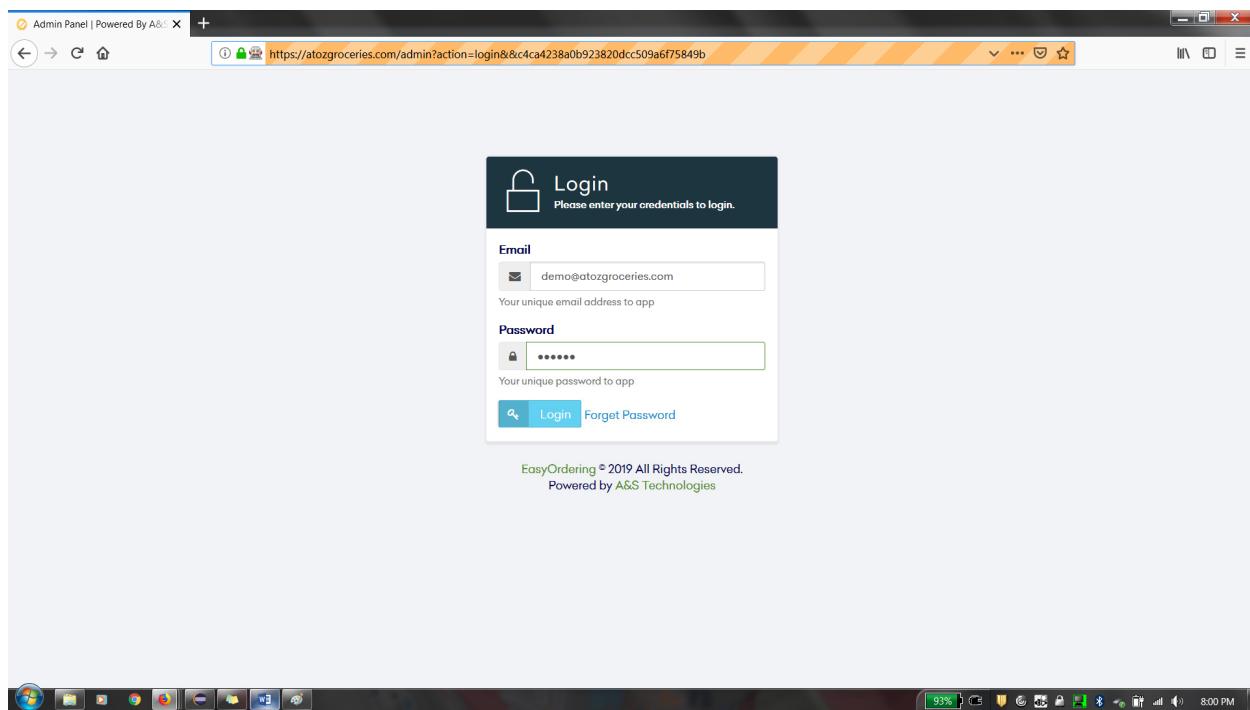
User page after logging out



Navigating to Admin portal



Providing login details



Dashboard after logging into the admin portal

A screenshot of the Admin Panel dashboard. The URL is https://atozgroceries.com/admin/dashboard?token=d3d9446802a44259755d38e6d163e820. The dashboard has a dark header with the 'easyordering' logo and a 'Logout' button. On the left is a sidebar with links for Dashboard, Catalog, Sales, Customers, Marketing, System, and Reports. The main area is titled 'Dashboard' and contains four cards: 'Total Orders' (4), 'Total Sales' (₹5,475.51), 'Total Customers' (8), and 'Total Products' (38). Below these 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 has an 'Action' button. The taskbar at the bottom shows various application icons and system status indicators.

Clicking on Sales

Dashboard

- Dashboard
- Catalog
- Sales
 - Orders
 - Returns
- Customers
- Marketing
- System
- Reports

Latest Orders

#	Order ID	Customer	Status	Total	Date Added	Action
1	#49	HarryTest3	Pending	₹3,637.00	2019-01-03	
2	#48	Amutha	Pending	₹3,030.00	2019-01-03	
3	#35	HarryTest2	Pending	₹3,637.00	2019-01-03	
4	#34	HarryTest1	Pending	₹3,475.00	2019-01-02	
5	#33	HarryTest	Pending	₹3,475.00	2019-01-02	

Opening Orders for validating order placed in admin portal

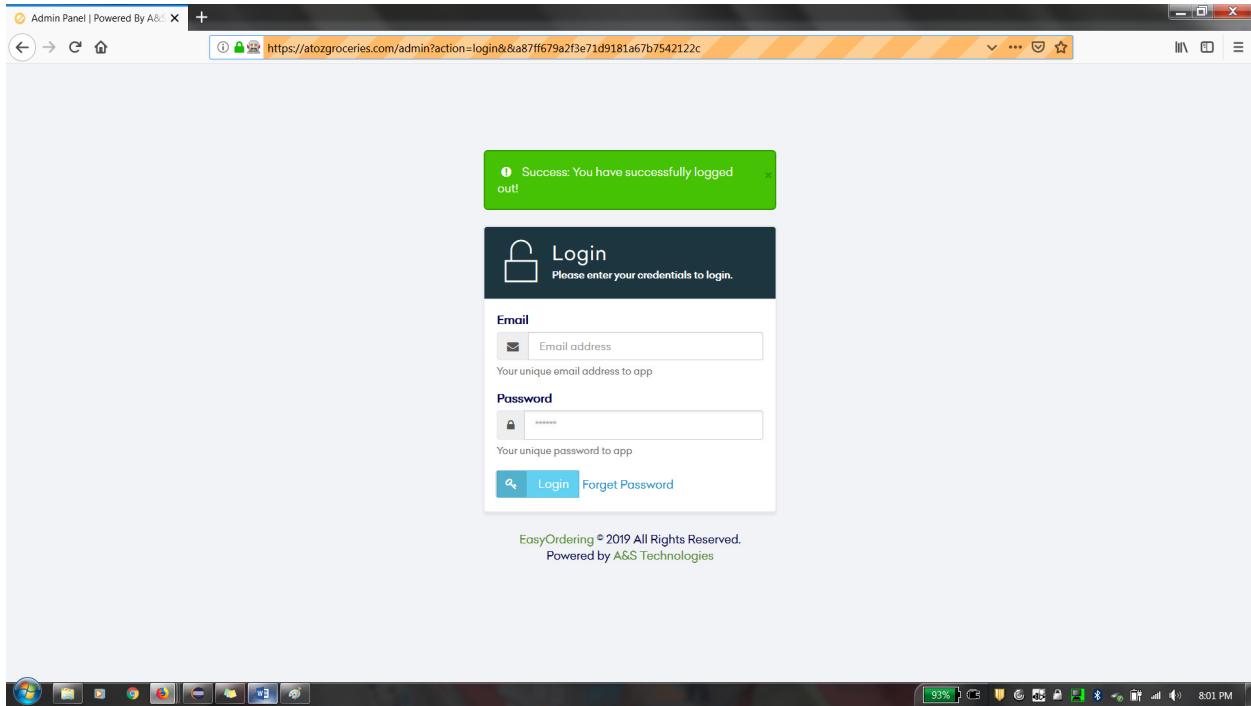
Orders

- Dashboard
- Catalog
- Sales
 - Orders
 - Returns
- Customers
- Marketing
- System
- Reports

Order List

	Order ID	Order No	Customer	Status	Total	Date Added	Action
	#49	#6234214618	HarryTest3	Pending	₹3,673.71	2019-01-03	Action ▾
	#48	#7336447961	Amutha	Pending	₹3,030.3	2019-01-03	Action ▾
	#35	#2505930375	HarryTest2	Pending	₹3,673.71	2019-01-03	Action ▾
	#34	#5517219865	HarryTest1	Pending	₹3,498.75	2019-01-02	Action ▾
	#33	#7012548633	HarryTest	Pending	₹3,498.75	2019-01-02	Action ▾
	#31	#6195266881	HarryTest12	Pending	₹3,174.96	2019-01-02	Action ▾
	#24	#1514564080	HarryTest12	Pending	₹3,498.75	2019-01-02	Action ▾
	#23	#3851988554	HarryTest12	Pending	₹3,498.75	2019-01-02	Action ▾
	#22	#1600952940	HarryTest12	Pending	₹3,600.21	2019-01-02	Action ▾

Logging out from admin portal after validating in database



Output from console

EclipseWorkSpace - SeleniumFramework/testng.xml - Eclipse IDE

```

File Edit Source Navigate Project Run Window Help
Package Explorer JUnit LoginPage.java AdminPage.java UserPage.java DBConnection.java data.properties testing.xml DriverLaunch.java
  > AdminPage.java
  > CategoryPage.java
  > CouponPage.java
  > DashboardPage.java
  > DriverLaunch.java
  > LoginPage.java
  > ProductsPage.java
  > ReturnsPage.java
  > SettingsPage.java
  > ShippingPage.java
  > TabsPage.java
  > UserPage.java
  com.ibm.test
    > BaseTest.java
    > JTest.java
  com.ibmutilities
    > DBConnection.java
    > ExcelReader.java
    > FileReader.java
    > JDBC_Connection.java
    > PropertiesFileHandler.java
  Referenced Libraries
  > settings
  > drivers
  > jars
  > Screenshots
  > Test-output
  > TestData
    > classpath
    > project
    > testing.xml
  > SetGet
  > Shape
  > SumOffTwo
  > SuperDemo
  > Test
  > Timer
  BaseTestJava LoginPage.java AdminPage.java UserPage.java DBConnection.java data.properties testing.xml DriverLaunch.java
  <?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"/>
<classes>
<class name="com.ibm.test.BaseTest">
<methods><include name="Day12TestCase"></include></methods>
</class>
</classes>
</test> <!-- Test -->
</suite> <!-- Suite -->
<!--
Design Source
Problems Javadoc Declaration Console Results of running suite
<terminated> SeleniumFramework_testing.xml [TestNG] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (Jan 3, 2019, 7:58:10 PM)
Message: Error: Polling for changes failed: NetworkError when attempting to fetch resource..
Stack:
remoteSettingsFunction/remoteSettings.pollChanges@resource://services-settings/remote-settings.js:750:13
JavaScript error: https://atozgroceries.com/assets/admin/dist/jss/app.min.js, line 1: TypeError: i is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/jss/page/dashboard.min.js, line 1: ReferenceError: toastr is not defined
JavaScript error: https://atozgroceries.com/admin/orders/tab/token/a87fff679a2f3e71d9181a67b7542122c/0, line 1061: TypeError: document.getElementById(...) is null
1546525992508 Marionette INFO Stopped listening on port 52380
[Parent 132, Gecko_IOThread] WARNING: pipe error: 109: file z:/build/build/src/IPC/chromium/src/chrome/common/ipc_channel_win.cc, line 346
[Parent 132, Gecko_IOThread] WARNING: pipe error: 109: file z:/build/build/src/IPC/chromium/src/chrome/common/ipc_channel_win.cc, line 346
[Child 9724, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/IPC/chromium/src/chrome/common/ipc_channel_win.cc, line 346
[Child 9688, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/IPC/chromium/src/chrome/common/ipc_channel_win.cc, line 346
[Child 9688, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/IPC/chromium/src/chrome/common/ipc_channel_win.cc, line 346
[Parent 132, Gecko_IOThread] WARNING: pipe error: 109: file z:/build/build/src/IPC/chromium/src/chrome/common/ipc_channel_win.cc, line 346
[Child 8904, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/IPC/chromium/src/chrome/common/ipc_channel_win.cc, line 346
[GPU 6404, Chrome_ChildThread] WARNING: pipe error: 109: file z:/build/build/src/IPC/chromium/src/chrome/common/ipc_channel_win.cc, line 346
#####
[Child][MessageChannel:SendAndWait] Error: Channel error: cannot send/recv
=====
Suite
Total tests run: 1, Failures: 0, Skips: 0
=====

Writable Smart Insert 13:1 93% available (plugged in, charging) 8:01 PM

```

Automated output of index.html

Test results
1 suite

All suites

Suite

Info

- C:\Harisha\Selenium_Training\EclipseWorkSpace\SeleniumFramework\testing.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 1 method, 1 passed
- Passed methods (hide)
- Day12TestCase

Reporter output for Suite

Day12TestCase

Products Bisley Water and Basmati Rice are available in cart!! Order placed by HarryTest3 is available in My Account -> My Orders with order id as #49 Order placed by HarryTest3 is available in Admin Page -> Sales -> Orders with Order No #6234214618 Order placed by HarryTest3 is available in Database!!

Automated output of emailable report

TestNG Report

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
Test	1	0	0	201,851		

Class	Method	Start	Time (ms)
Suite			
Test — passed			
com.ibm.test.BaseTest	Day12TestCase	[1546525698254]	195459

Test

com.ibm.test.BaseTest#Day12TestCase

Messages

Products Bisley Water and Basmati Rice are available in cart!!
Order placed by HarryTest3 is available in My Account -> My Orders with order id as #49
Order placed by HarryTest3 is available in Admin Page -> Sales -> Orders with Order No #6234214618
Order placed by HarryTest3 is available in Database!!

[back to summary](#)

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