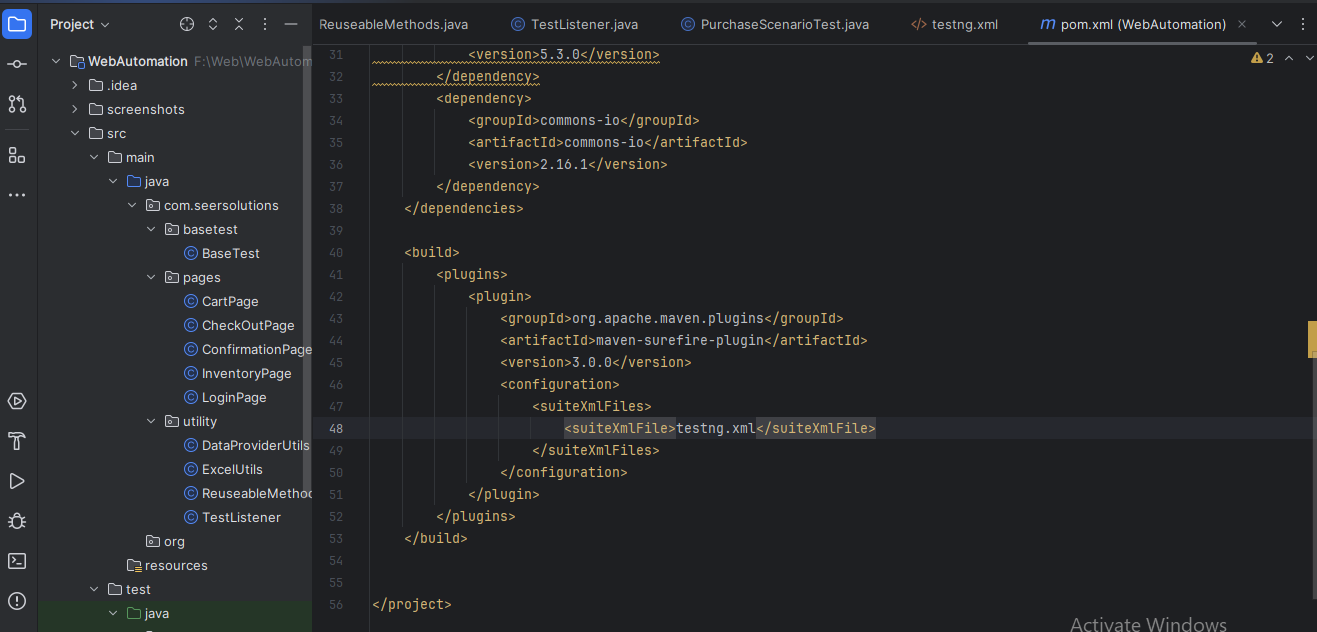
**Web Automation Report**

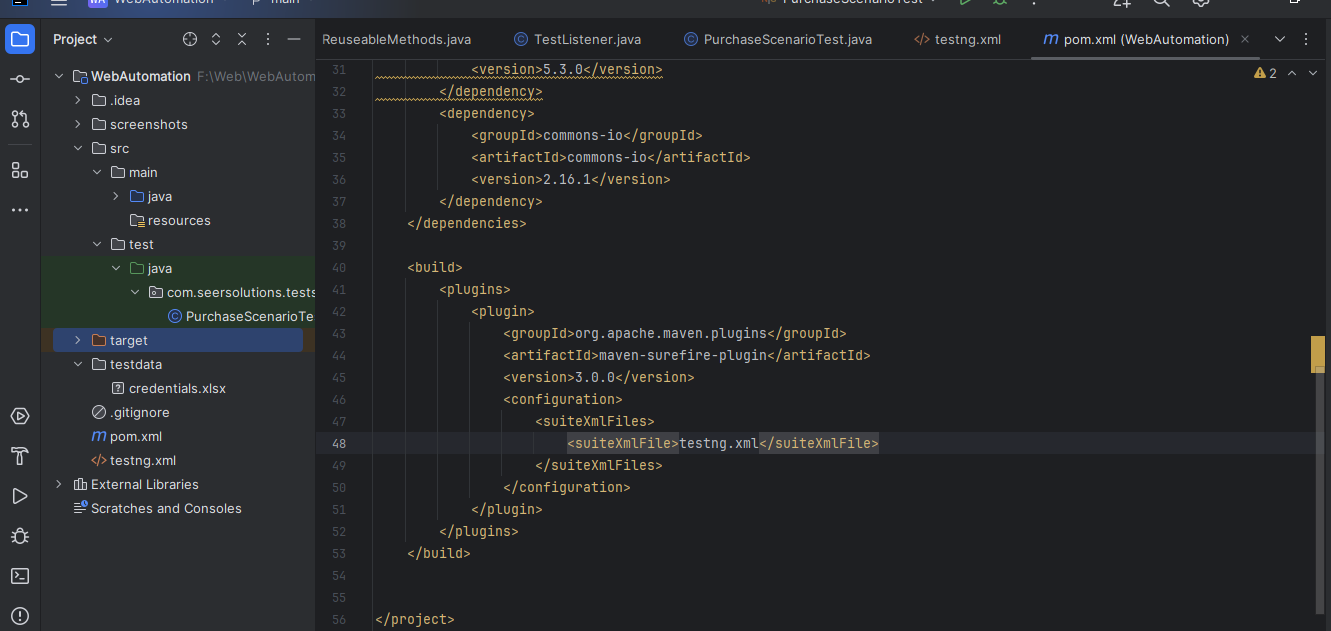
Git hub repository link : <https://github.com/Prasanthpsg/WebAutomationTest>

**Scenario 1:**

Verify the purchase flow.

Java + selenium webdriver + TestNG + Page factory





**BaseTest**

***src/main/java/com/seersolutions/basetest/BaseTest.java***

package com.seersolutions.basetest;  
  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.chrome.ChromeOptions;  
import org.testng.annotations.AfterClass;  
import org.testng.annotations.BeforeClass;  
  
import java.time.Duration;  
import java.util.Map;  
  
import com.seersolutions.utility.TestListener;  
import org.testng.annotations.Listeners;  
import org.testng.asserts.SoftAssert;  
  
@Listeners(TestListener.class)  
public class BaseTest {  
 public WebDriver driver;  
  
 @BeforeClass  
 public void createDriver(){  
 ChromeOptions options = new ChromeOptions();  
 options.addArguments("--incognito");  
 options.addArguments("--disable-notifications");  
 options.addArguments("--disable-popup-blocking");  
 options.addArguments("--disable-infobars");  
 options.addArguments("--disable-save-password-bubble");  
 options.addArguments("--disable-blink-features=AutomationControlled");  
 options.setExperimentalOption("prefs", Map.*of*(  
 "credentials\_enable\_service", false,  
 "profile.password\_manager\_enabled", false  
 ));  
 driver = new ChromeDriver(options);  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(10));  
 }  
  
 @AfterClass  
 public void tearDown() {  
 if (driver != null)  
 driver.quit();  
 }  
}

**Cart Page:**

***src/main/java/com/seersolutions/pages/CartPage.java***

package com.seersolutions.pages;  
  
import com.seersolutions.basetest.BaseTest;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.support.FindBy;  
import org.openqa.selenium.support.PageFactory;  
import org.testng.Assert;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class CartPage {  
  
 private WebDriver driver;  
  
 @FindBy(className = "inventory\_item\_name")  
 List<WebElement> cartItems;  
  
 @FindBy(xpath = "//a[text()='CHECKOUT']")  
 WebElement checkoutBtn;  
  
 @FindBy(xpath = "(//div[@class='cart\_item\_label']//a/div)[1]")  
 WebElement cartItems1;  
  
 @FindBy(xpath = "(//div[@class='cart\_item\_label']//a/div)[2]")  
 WebElement cartItems2;  
  
 public CartPage(WebDriver driver) {  
 this.driver = driver;  
 PageFactory.*initElements*(driver, this);  
 }  
  
 public void verifyItemsInCart(int expectedCount) {  
 Assert.*assertEquals*(cartItems.size(), expectedCount, "Cart item count mismatch!");  
 }  
  
 public List<String> getCartDetails(){  
 List<String> cartitems = new ArrayList<>();  
 cartitems.add(cartItems1.getText().trim());  
 cartitems.add(cartItems2.getText().trim());  
 return cartitems;  
 }  
  
 public void verifyAddedItems(List<String> actualList, List<String> expectedList){  
 Assert.*assertEquals*(actualList, expectedList, "Added items name does not match");  
 }  
  
 public void clickCheckout() {  
 checkoutBtn.click();  
 }  
}

**Check out page:**

***src/main/java/com/seersolutions/pages/CheckOutPage.java***

package com.seersolutions.pages;  
  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.support.FindBy;  
import org.openqa.selenium.support.PageFactory;  
  
public class CheckOutPage {  
 private WebDriver driver;  
  
 @FindBy(id = "first-name")  
 WebElement firstName;  
  
 @FindBy(id = "last-name")  
 WebElement lastName;  
  
 @FindBy(id = "postal-code")  
 WebElement postalCode;  
  
 @FindBy(xpath = "//input[@value='CONTINUE']")  
 WebElement continueBtn;  
  
 @FindBy(xpath = "//a[text()='FINISH']")  
 WebElement finishBtn;  
  
 public CheckOutPage(WebDriver driver) {  
 this.driver = driver;  
 PageFactory.*initElements*(driver, this);  
 }  
  
 public void fillDetailsAndFinish() {  
 firstName.sendKeys("Test firstname");  
 lastName.sendKeys("Test lastname");  
 postalCode.sendKeys("638503");  
 continueBtn.click();  
 finishBtn.click();  
 }  
}

**Confirmation page:**

***src/main/java/com/seersolutions/pages/ConfirmationPage.java***

package com.seersolutions.pages;  
  
import com.seersolutions.basetest.BaseTest;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.support.FindBy;  
import org.openqa.selenium.support.PageFactory;  
import org.testng.Assert;  
  
public class ConfirmationPage {  
  
 private WebDriver driver;  
  
 @FindBy(className = "complete-header")  
 WebElement confirmationMessage;  
  
 public ConfirmationPage(WebDriver driver) {  
 this.driver = driver;  
 PageFactory.*initElements*(driver, this);  
 }  
  
 public void verifyOrderMessage(String expectedmessage) {  
 Assert.*assertEquals*(confirmationMessage.getText().trim(), expectedmessage);  
 }  
}

**Inventory Page:**

***src/main/java/com/seersolutions/pages/InventoryPage.java***

package com.seersolutions.pages;  
  
import com.seersolutions.basetest.BaseTest;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.support.FindBy;  
import org.openqa.selenium.support.PageFactory;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class InventoryPage {  
  
  
 private WebDriver driver;  
  
 @FindBy(xpath = "(//div[@class='pricebar']//button)[1]")  
 WebElement firstProduct;  
  
 @FindBy(xpath = "(//div[@class='pricebar']//button)[2]")  
 WebElement secondProduct;  
  
 @FindBy(className = "shopping\_cart\_link")  
 WebElement cartIcon;  
  
 @FindBy(xpath = "(//div[@class='inventory\_item\_label']//a/div)[2]")  
 WebElement inventoryTitle2;  
  
 @FindBy(xpath = "(//div[@class='inventory\_item\_label']//a/div)[1]")  
 WebElement inventoryTitle1;  
  
  
 public InventoryPage(WebDriver driver) {  
 this.driver = driver;  
 PageFactory.*initElements*(driver, this);  
 }  
  
 public void addFirstTwoProducts() {  
 firstProduct.click();  
 secondProduct.click();  
 }  
  
 public void goToCart() {  
 cartIcon.click();  
 }  
  
 public List<String> getInventoryDetails(){  
 List<String> inventory = new ArrayList<>();  
 inventory.add(inventoryTitle1.getText().trim());  
 inventory.add(inventoryTitle2.getText().trim());  
 return inventory;  
 }  
}

**Login Page:**

***src/main/java/com/seersolutions/pages/LoginPage.java***

package com.seersolutions.pages;  
  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.support.FindBy;  
import org.openqa.selenium.support.PageFactory;  
  
public class LoginPage {  
  
 private WebDriver driver;  
  
 @FindBy(id = "user-name")  
 WebElement usernameInput;  
  
 @FindBy(id = "password")  
 WebElement passwordInput;  
  
 @FindBy(id = "login-button")  
 WebElement loginButton;  
  
 public LoginPage(WebDriver driver) {  
 this.driver = driver;  
 PageFactory.*initElements*(driver, this);  
 }  
  
 public void login(String user, String pass) {  
 usernameInput.sendKeys(user);  
 passwordInput.sendKeys(pass);  
 loginButton.click();  
 }  
}

**Utilities:**

***src/main/java/com/seersolutions/utility/DataProviderUtils.java***

package com.seersolutions.utility;  
  
import org.testng.annotations.DataProvider;  
  
public class DataProviderUtils {  
  
 @DataProvider(name = "loginData")  
 public Object[][] getLoginData() {  
 String excelPath = "testdata/credentials.xlsx";  
 String sheetName = "Sheet1";  
 return ExcelUtils.*readExcelData*(excelPath, sheetName);  
 }  
}

src/main/java/com/seersolutions/utility/ExcelUtils.java

package com.seersolutions.utility;  
  
import org.apache.poi.ss.usermodel.\*;  
import org.apache.poi.xssf.usermodel.XSSFWorkbook;  
  
import java.io.FileInputStream;  
import java.util.Arrays;  
  
public class ExcelUtils {  
  
 public static String getCellData(String filePath, String sheetName, int row, int col) {  
 try (FileInputStream fis = new FileInputStream(filePath);  
 Workbook workbook = new XSSFWorkbook(fis)) {  
 Sheet sheet = workbook.getSheet(sheetName);  
 return sheet.getRow(row).getCell(col).getStringCellValue();  
 } catch (Exception e) {  
 e.printStackTrace();  
 return "";  
 }  
 }  
  
 public static Object[][] readExcelData(String filePath, String sheetName) {  
 Object[][] data = null;  
  
 try (FileInputStream fis = new FileInputStream(filePath);  
 Workbook workbook = WorkbookFactory.*create*(fis)) {  
  
 Sheet sheet = workbook.getSheet(sheetName);  
 int rowCount = sheet.getLastRowNum();  
 int colCount = sheet.getRow(0).getPhysicalNumberOfCells();  
  
 data = new Object[rowCount - 1][colCount];  
  
 for (int i = 1; i < rowCount; i++) { // Skip header  
 Row row = sheet.getRow(i);  
 for (int j = 0; j < colCount; j++) {  
 Cell cell = row.getCell(j);  
 data[i - 1][j] = (cell != null) ? cell.toString().trim() : "";  
 }  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
// System.out.println("Excel Data (Object[][]):");  
// for (Object[] row : data) {  
// for (Object value : row) {  
// System.out.print(value + " | ");  
// }  
// System.out.println();  
// }  
 return data;  
 }  
}

**Listener:**

***src/main/java/com/seersolutions/utility/TestListener.java***

package com.seersolutions.utility;  
  
import org.testng.ITestListener;  
import org.testng.ITestResult;  
import org.openqa.selenium.\*;  
import org.apache.commons.io.FileUtils;  
  
  
import java.io.File;  
public class TestListener implements ITestListener {  
  
 @Override  
 public void onTestFailure(ITestResult result) {  
 Object currentClass = result.getInstance();  
 WebDriver driver = ((com.seersolutions.basetest.BaseTest) currentClass).driver;  
  
 try {  
 File src = ((TakesScreenshot) driver).getScreenshotAs(OutputType.*FILE*);  
 String dest = "screenshots/" + result.getName() + ".png";  
 FileUtils.*copyFile*(src, new File(dest));  
 System.*out*.println("Screenshot saved: " + dest);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
  
}

**Execution Flow:**

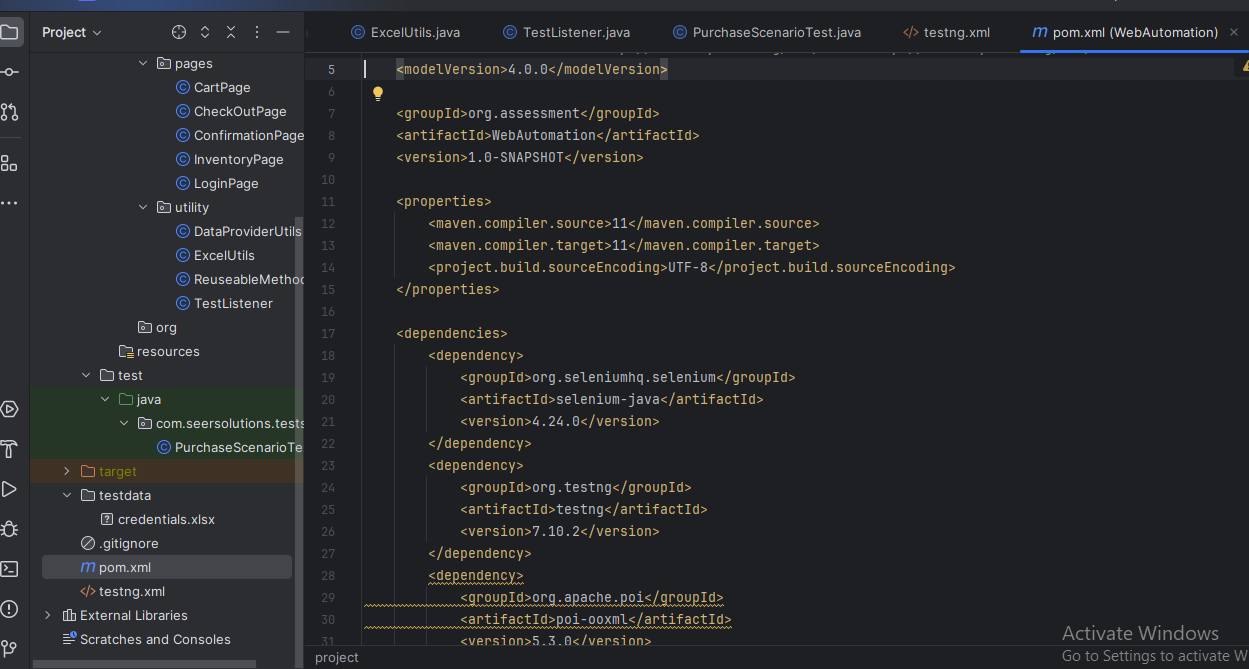
***src/test/java/com/seersolutions/tests/PurchaseScenarioTest.java***

package com.seersolutions.tests;  
  
import com.seersolutions.basetest.BaseTest;  
import com.seersolutions.pages.\*;  
import com.seersolutions.utility.DataProviderUtils;  
import com.seersolutions.utility.ReuseableMethods;  
import org.testng.annotations.DataProvider;  
import org.testng.annotations.Test;  
  
import java.util.List;  
  
public class PurchaseScenarioTest extends BaseTest {  
  
 @Test(dataProvider = "loginData" , dataProviderClass = DataProviderUtils.class)  
 public void verifyPurchaseFlow(String userName, String password) {  
 driver.navigate().to("https://www.saucedemo.com/v1/");  
  
 //Login using credentials stored in an external file  
 LoginPage login = new LoginPage(driver);  
 login.login(userName,password);  
  
 //Add the first two products to the cart  
 InventoryPage products = new InventoryPage(driver);  
 List<String> listoftwoproducts = products.getInventoryDetails();  
 products.addFirstTwoProducts();  
 products.goToCart();  
  
 //verify both products are listed and do checkout  
 CartPage cart = new CartPage(driver);  
 cart.verifyItemsInCart(2);  
 cart.verifyAddedItems(cart.getCartDetails(),listoftwoproducts);  
 cart.clickCheckout();  
  
 CheckOutPage checkout = new CheckOutPage(driver);  
 checkout.fillDetailsAndFinish();  
  
 //Assert the final confirmation message  
 ConfirmationPage confirmation = new ConfirmationPage(driver);  
 confirmation.verifyOrderMessage("THANK YOU FOR YOUR ORDER");  
  
 }  
  
 @Test(dataProvider = "loginData" , dataProviderClass = DataProviderUtils.class)  
 public void verifyPurchaseErrorFlow(String userName, String password) {  
 driver.navigate().to("https://www.saucedemo.com/v1/");  
  
 //Login using credentials stored in an external file  
 LoginPage login = new LoginPage(driver);  
 login.login(userName,password);  
  
 //Add the first two products to the cart  
 InventoryPage products = new InventoryPage(driver);  
 products.addFirstTwoProducts();  
 products.goToCart();  
  
 //verify both products are listed and do checkout  
 CartPage cart = new CartPage(driver);  
 cart.verifyItemsInCart(2);  
 cart.clickCheckout();  
  
 CheckOutPage checkout = new CheckOutPage(driver);  
 checkout.fillDetailsAndFinish();  
  
 //Assert the final confirmation message  
 ConfirmationPage confirmation = new ConfirmationPage(driver);  
 confirmation.verifyOrderMessage("THANK YOU FOR YOUR");  
 }  
}

**testng.xml**

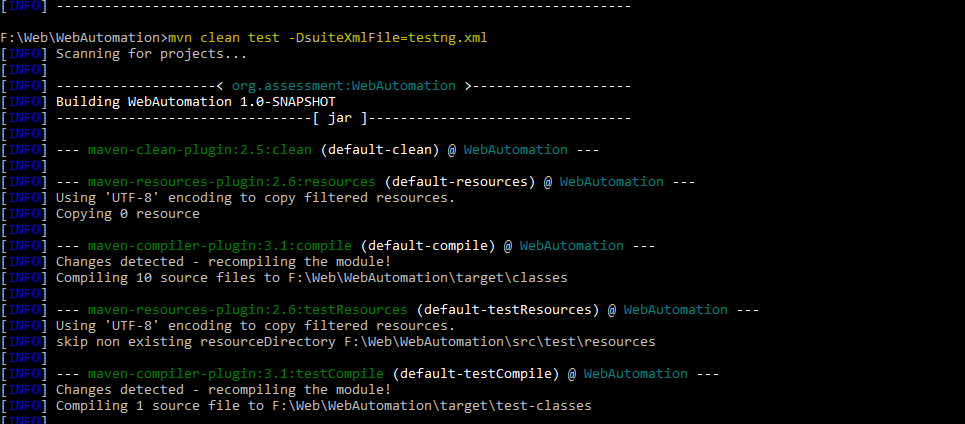
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">  
<suite name="All Test Suite">  
 <test verbose="2" preserve-order="true" name="PurchaseFlowTest">  
 <classes>  
 <class name="com.seersolutions.tests.PurchaseScenarioTest">  
 </class>  
 </classes>  
 </test>  
</suite>

POM.xml

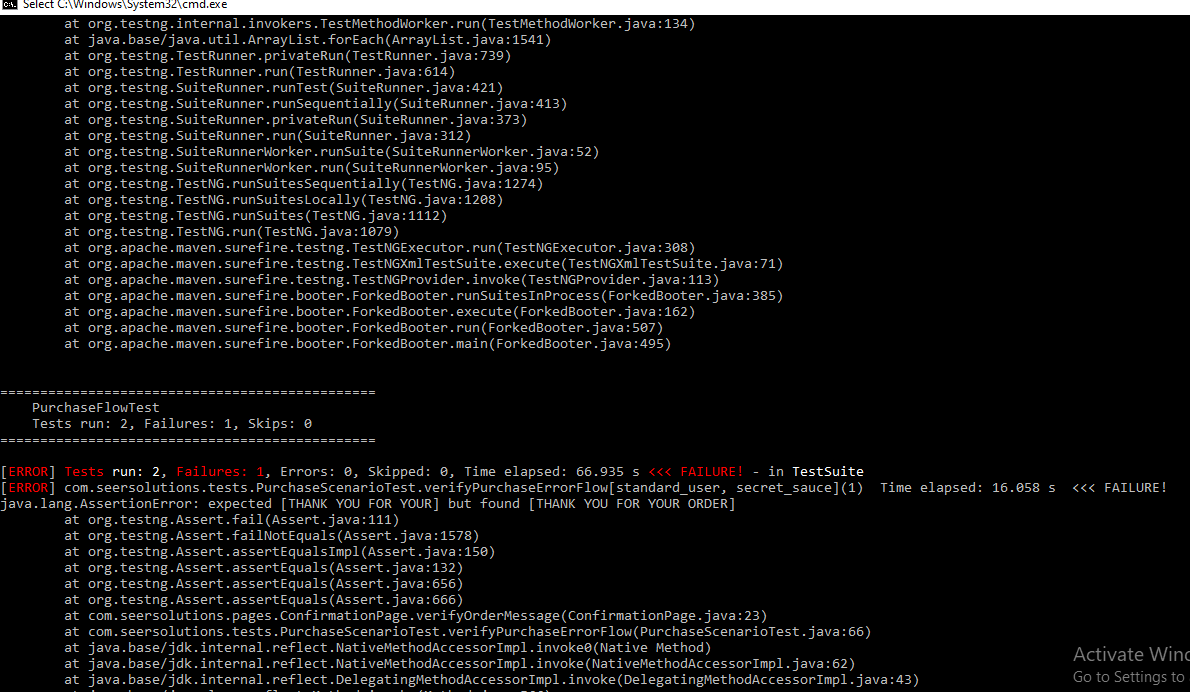


Execution via command prompt:

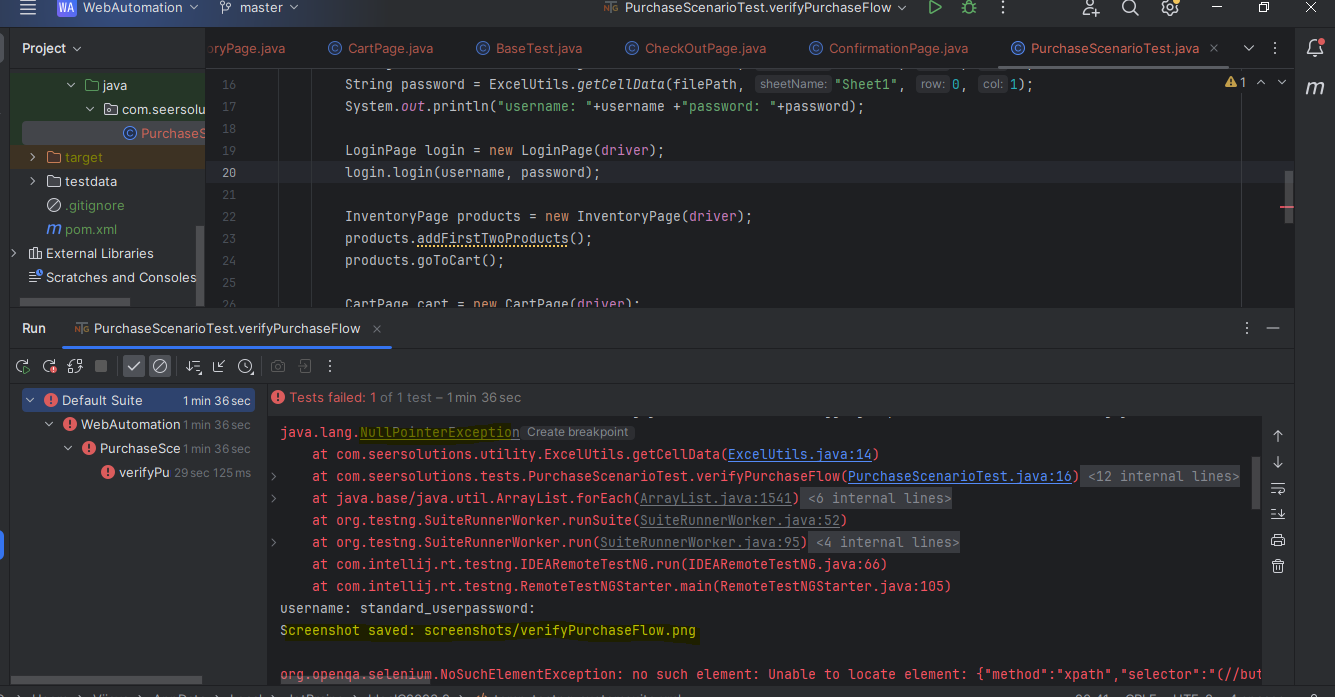
mvn clean test –DsuiteXmlFile=testing.xml



Due to assertion, 1 test case failed.



Scenario 2: Failure scenario – Validate Screenshot Capture via Listener.

Upon test failure, the ITestListener interface’s onTestFailure() method successfully captured a screenshot and saved it in the designated **screenshots** folder. The captured error details are highlighted below.

Screenshot folder: Highlighted below

