```
package KitePOMUsingTestNG;


import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class KiteLoginPage

{

	// 1.data members
	@FindBy (id = "userid") private  WebElement userName;
	@FindBy (id = "password") private WebElement password;
	@FindBy (xpath = "//button[@type='submit']") private WebElement loginButton;



	//2. constructor

	public KiteLoginPage(WebDriver driver)
	{
		PageFactory.initElements(driver, this);
	}

	//3. methods

	public void sendUserName(String UserName)
	{
		userName.sendKeys(UserName);
	}

	public void sendPassword(String passWord)
```

```
        {
                password.sendKeys(passWord);
        }

        public void clickOnLoginButton()
        {
                loginButton.click();
        }
}
```

```
package KitePOMUsingTestNG;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class KitePinPage
{
        //1

        @FindBy(id = "pin") private WebElement PIN;
        @FindBy(xpath = "//button[@type='submit']")  private WebElement continueButton;

        //2

        public KitePinPage(WebDriver driver)
        {
                PageFactory.initElements(driver, this);

        }
```

```
	//3

	public void sendPin(String pin)
	{
		PIN.sendKeys(pin);
	}

	public void clickOnContinueButton()
	{
		continueButton.click();
	}
}
```

```
package KitePOMUsingTestNG;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class KiteHomePage
{

	//1
	@FindBy(xpath = "//span[@class='user-id']") private WebElement userName;
	@FindBy(xpath = "//a[@target='_self']") private WebElement logOutButton;
	//2

	public KiteHomePage(WebDriver driver)
	{
		PageFactory.initElements(driver, this);
	}

	//3

	public void validateUserName(String expextedUserID)
	{

		String expextedUserName=expextedUserID;
		String actualUserName = userName.getText();

		if(expextedUserName.equals(actualUserName))
		{
			System.out.println("Actual and Expected User Id are matching TC is 
passed");
		}

		else {
			System.out.println("Actual and Expected User Id are not matching TC is 
failed");

		}



	}
	//to get actual userName
	public String getActualUserName()
	{
		String actualUserName = userName.getText();
		return actualUserName;
	}

	public void logOut() throws InterruptedException
	{
		userName.click();

		Thread.sleep(200);

		logOutButton.click();

	}

	}
```

```
package KitePOMUsingTestNG;

import java.io.File;
import java.io.IOException;
import java.time.Duration;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

public class ValidateKiteAppUserName {

	WebDriver driver;
	Sheet mySheet;
	KiteLoginPage login;
	KitePinPage pin;
	KiteHomePage home;


	@BeforeClass
	public void launchBrowser() throws EncryptedDocumentException, IOException
	{
		System.setProperty("webdriver.chrome.driver", "D:\\Velocity\\Java Class\\26th
March B\\Selenium\\chromedriver.exe");

		ChromeOptions opt= new ChromeOptions();
		//opt.addArguments("--headless");
		//opt.addArguments("--disable-notifications");
		opt.addArguments("incognito");
		driver= new ChromeDriver(opt);
		driver.manage().window().maximize();
		driver.get("https://kite.zerodha.com/");
		Reporter.log("Launching browser",true);
		driver.manage().timeouts().implicitlyWait(Duration.ofMillis(1000));
		File myfile= new File("D:\\Velocity\\Java Class\\26th March
B\\Selenium\\Excel26thMarchB.xlsx");
		mySheet = WorkbookFactory.create(myfile).getSheet("Sheet2");

		login= new KiteLoginPage(driver);
		pin = new KitePinPage(driver);
		home= new KiteHomePage(driver);

	}
	@BeforeMethod
	public void loginToKiteApp()
	{
		String UN = mySheet.getRow(5).getCell(0).getStringCellValue();
		String PWD = mySheet.getRow(5).getCell(1).getStringCellValue();
		String PIN = mySheet.getRow(5).getCell(2).getStringCellValue();

		login.sendUserName(UN);
		Reporter.log("sending username",true);
		login.sendPassword(PWD);
		Reporter.log("sending password",true);
		login.clickOnLoginButton();
		Reporter.log("clicking on login button",true);
		driver.manage().timeouts().implicitlyWait(Duration.ofMillis(1000));

		pin.sendPin(PIN);
		Reporter.log("sending PIN",true);
		pin.clickOnContinueButton();
		Reporter.log("clicking on continue button",true);
		driver.manage().timeouts().implicitlyWait(Duration.ofMillis(1000));

	}


	@Test
	public void validateUserName()
	{
		String expectedUN = mySheet.getRow(5).getCell(0).getStringCellValue();
		String actualUN = home.getActualUserName();
		Reporter.log("Validating UserName",true);
		Assert.assertEquals(actualUN, expectedUN,"Actual and Expected UN are not
matching TC failed");
		Reporter.log("Actual and Expected UN are matching TC PASSED",true);

	}

	@AfterMethod
	public void logoutFromKiteApp() throws InterruptedException
	{
		home.logOut();
		Reporter.log("logging out...",true);
	}

	@AfterClass
	public void closeBrowser() throws InterruptedException
	{
		Thread.sleep(2000);
		Reporter.log("Closing browser",true);
		driver.close();
	}
}
```

Kite Using Utility and Base

POM Classes will be same as previous

```
package KiteBase;

import java.time.Duration;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.testng.Reporter;



public class Base
{	protected WebDriver driver;
	public void openBrowser()
	{
		System.setProperty("webdriver.chrome.driver", "D:\\Velocity\\Java Class\\26th
March B\\Selenium\\chromedriver.exe");

		ChromeOptions opt= new ChromeOptions();
		opt.addArguments("--disable-notifications");
		opt.addArguments("incognito");
		driver= new ChromeDriver(opt);
		driver.manage().window().maximize();
		driver.get("https://kite.zerodha.com/");
		Reporter.log("Launching browser",true);
		driver.manage().timeouts().implicitlyWait(Duration.ofMillis(1000));
	}

}
```

```
package KiteUtility;

import java.io.File;
import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Utility
{
	//excel
	//screenshot
	//closing

	public static String readDataFromExcel(int row, int cell) throws
EncryptedDocumentException, IOException
	{
		File myfile= new File("D:\\Velocity\\Java Class\\26th March
B\\Selenium\\Excel26thMarchB.xlsx");
		Sheet mySheet = WorkbookFactory.create(myfile).getSheet("Sheet2");
		String value = mySheet.getRow(row).getCell(cell).getStringCellValue();
		return value;
	}


}
```

```
package KiteTest;

import java.io.IOException;
import java.time.Duration;

import org.apache.poi.EncryptedDocumentException;
import org.testng.Assert;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import KiteBase.Base;
import KitePOMnew.KiteHomePage;
import KitePOMnew.KiteLoginPage;
import KitePOMnew.KitePinPage;
import KiteUtility.Utility;

public class ValidateKiteUserID extends Base {

	KiteHomePage home;
	KiteLoginPage login;
	KitePinPage pin;

	@BeforeClass
	public void launchBrowser()
	{
		openBrowser();
		login= new KiteLoginPage(driver);
		pin= new KitePinPage(driver);
		home= new KiteHomePage(driver);
	}

	@BeforeMethod
	public void loginToKiteApp() throws EncryptedDocumentException, IOException
	{
		login.sendUserName(Utility.readDataFromExcel(5, 0));
		login.sendPassword(Utility.readDataFromExcel(5, 1));
		login.clickOnLoginButton();

		driver.manage().timeouts().implicitlyWait(Duration.ofMillis(1000));

		pin.sendPin(Utility.readDataFromExcel(5, 2));
		pin.clickOnContinueButton();
		driver.manage().timeouts().implicitlyWait(Duration.ofMillis(1000));

	}

	@Test
  public void validateUserID() throws EncryptedDocumentException, IOException
	{
		Assert.assertEquals(home.getActualUserName(), Utility.readDataFromExcel(5,
0),"Actual and Expected are not matching TC is failed");

	}

	@AfterMethod
	public void logOutFromKite() throws InterruptedException
	{
		home.logOut();
	}

	@AfterClass
	public void closeBrowser()
	{
		driver.close();
	}


}
```