Selenium with Excel

APACHI POI API

Selenium with Excel is Data Driven Testing

Before we start using with Framework lets understand the basics

In the below we get the file information and then use workbook for further operation

FileInputStream fis=**new** FileInputStream("E:\\html2\\test1.xlsx");

XSSFWorkbook workbook=**new** XSSFWorkbook(fis);

In below we will get the sheet information by giving the sheet name

XSSFSheet sheet =workbook.getSheet("sheetname");

Example: XSSFSheet sheet =workbook.getSheet("LoginInfo");

Please not row starts from zero and column also starts from one

To get the row count use below

**int** rowcount=sheet.getLastRowNum();

To get the column count of any row use below

**int** colcount=sheet.getRow(0).getLastCellNum();

To go into particular row use below

XSSFRow row=sheet.getRow(0);

To get certain value of 1st row and first column use below

XSSFCell cell=sheet.getRow(0).getCell(1);

To get certain value of 1st row and second column use below

XSSFCell cell=sheet.getRow(0).getCell(2);

To get certain value of 1st row and third column use below

XSSFCell cell=sheet.getRow(0).getCell(3);

And so on…

If you want to itereate from rows and get values for each row from different cells(i.e. column values) the use below

String[][] exceldatainfo =**null**;

exceldatainfo = **new** String[rowcount][colcount];

**for**(**int** i=1;i<=rowcount;i++) {

**for**(**int** j=0;j<1;j++) {

exceldatainfo[i-1][j]=sheet.getRow(i).getCell(0).getStringCellValue();

exceldatainfo[i-1][j+1]=sheet.getRow(i).getCell(1).getStringCellValue();

}

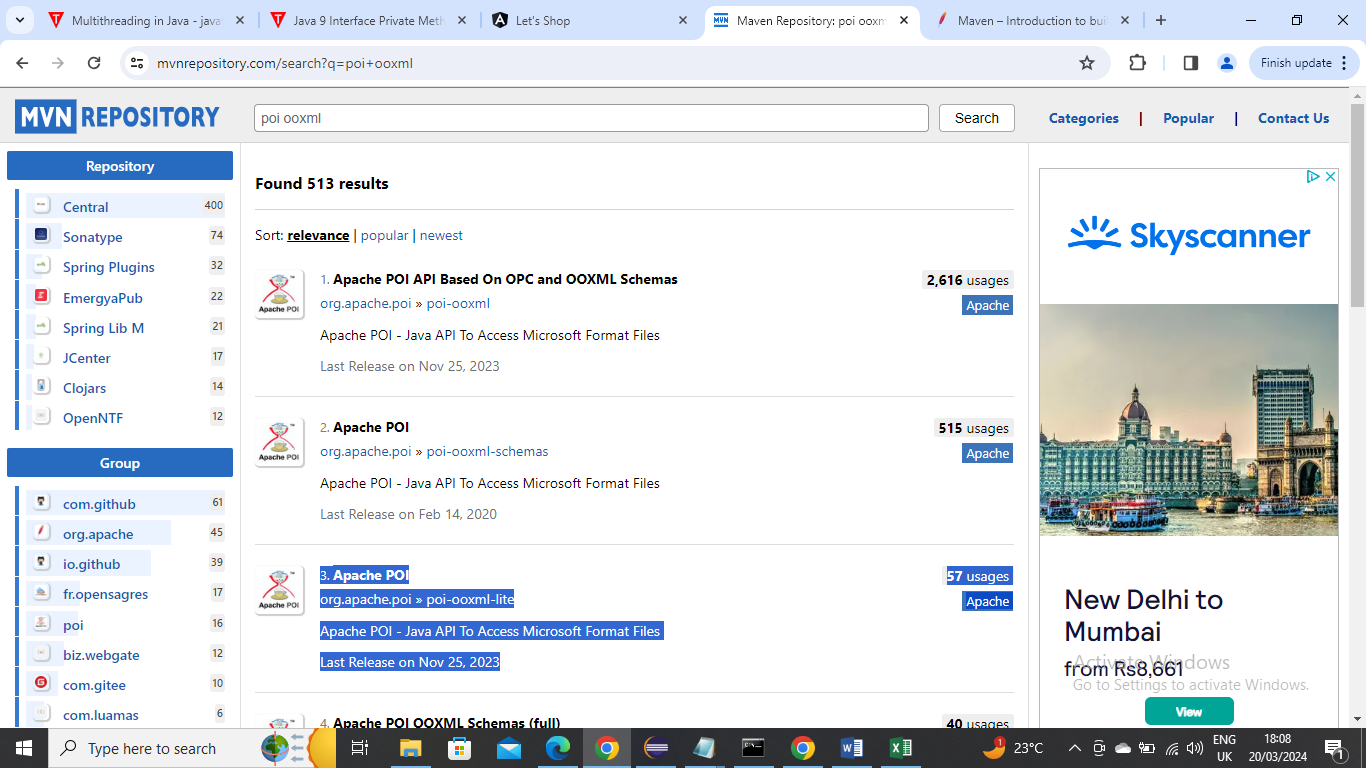
}

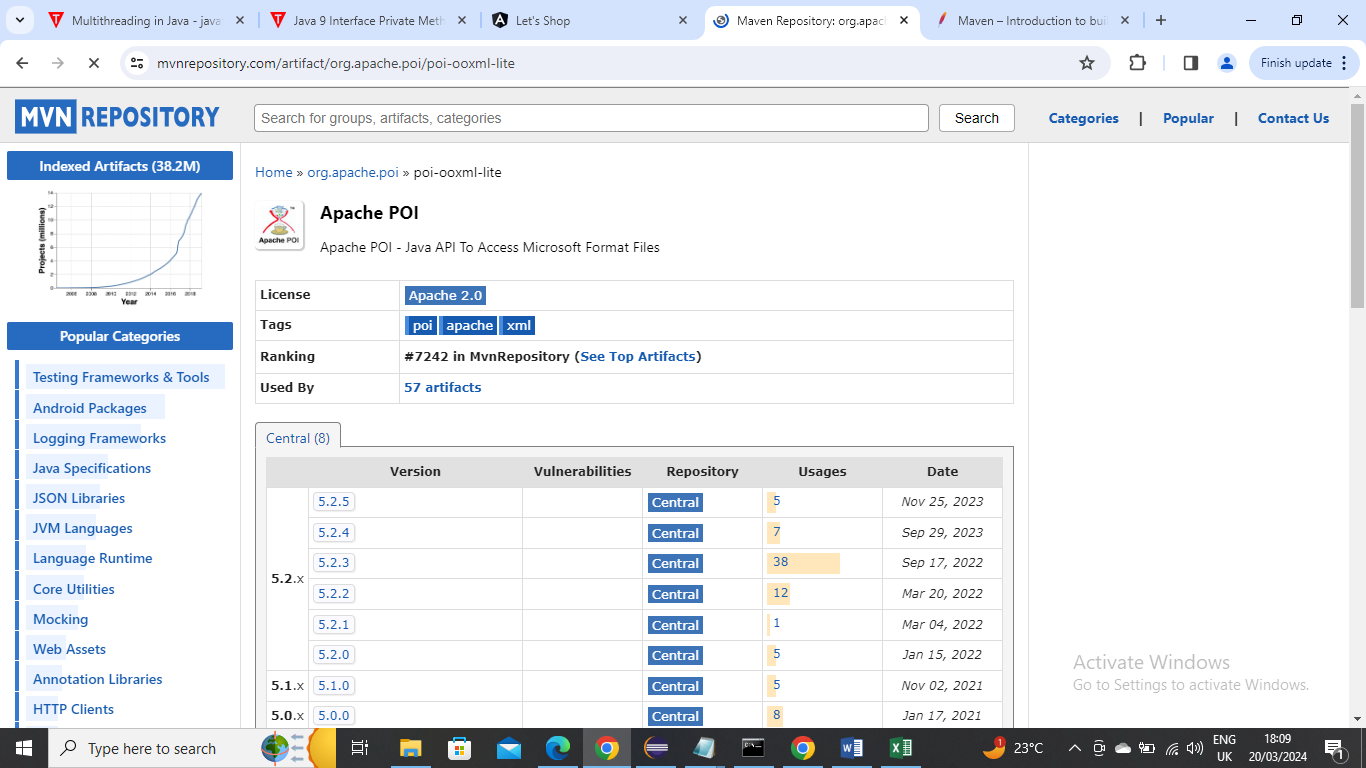
------------

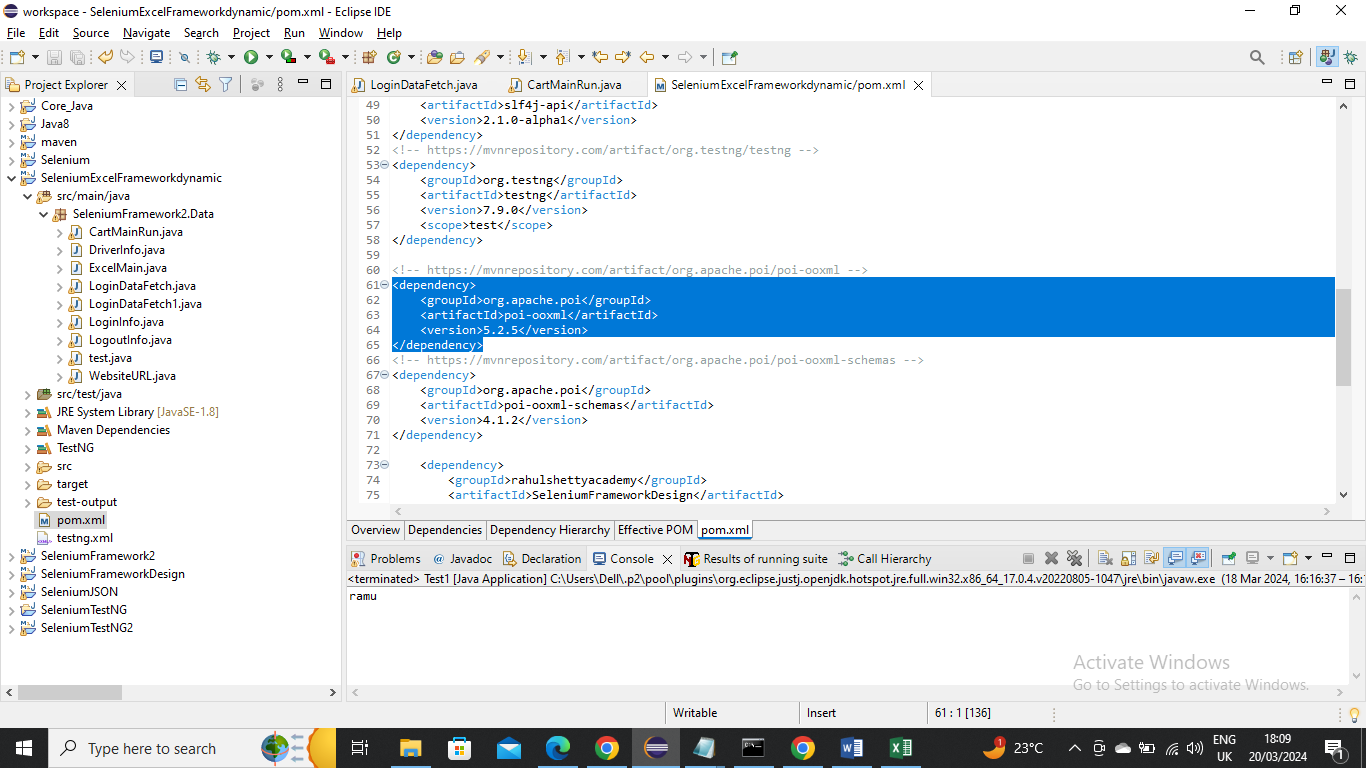
Maven dependencies for Excel

Poi-ooxml and poi

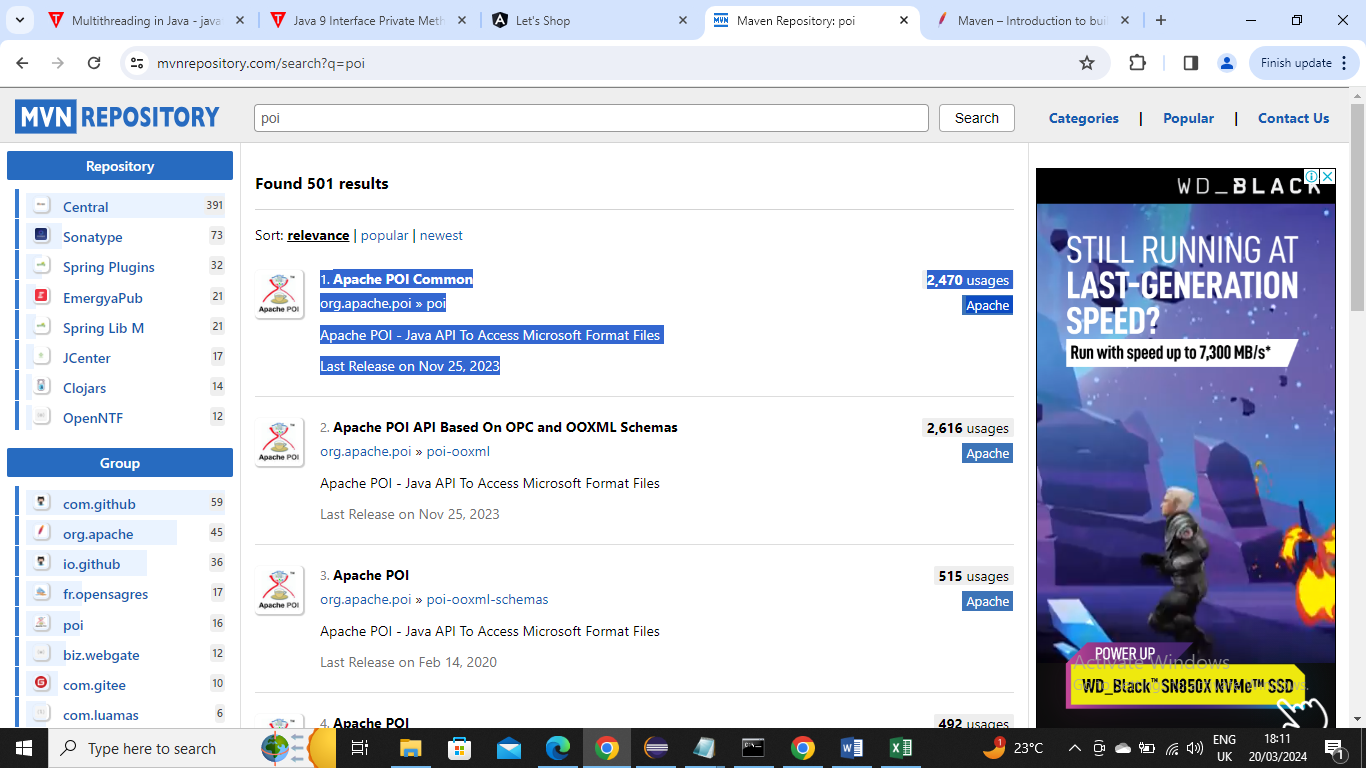
Lets add poi-ooxml dependency

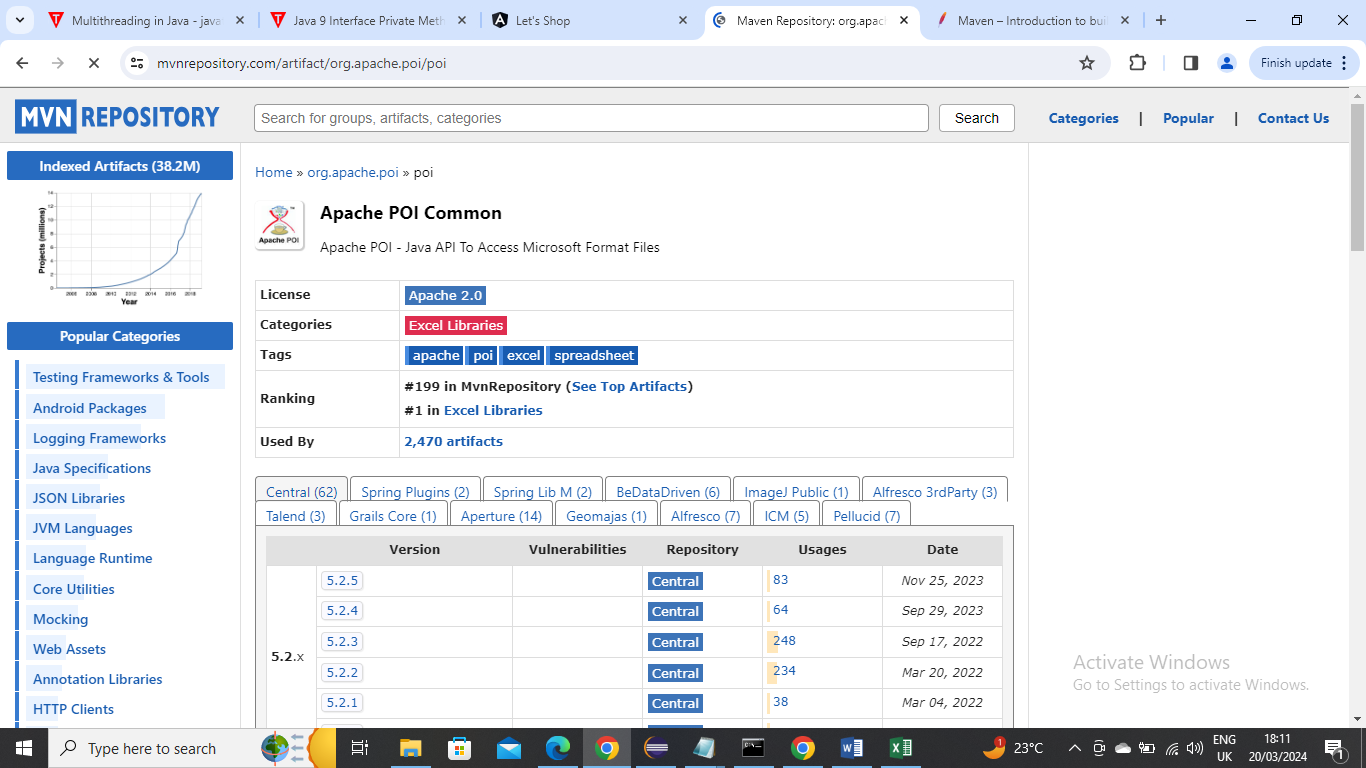


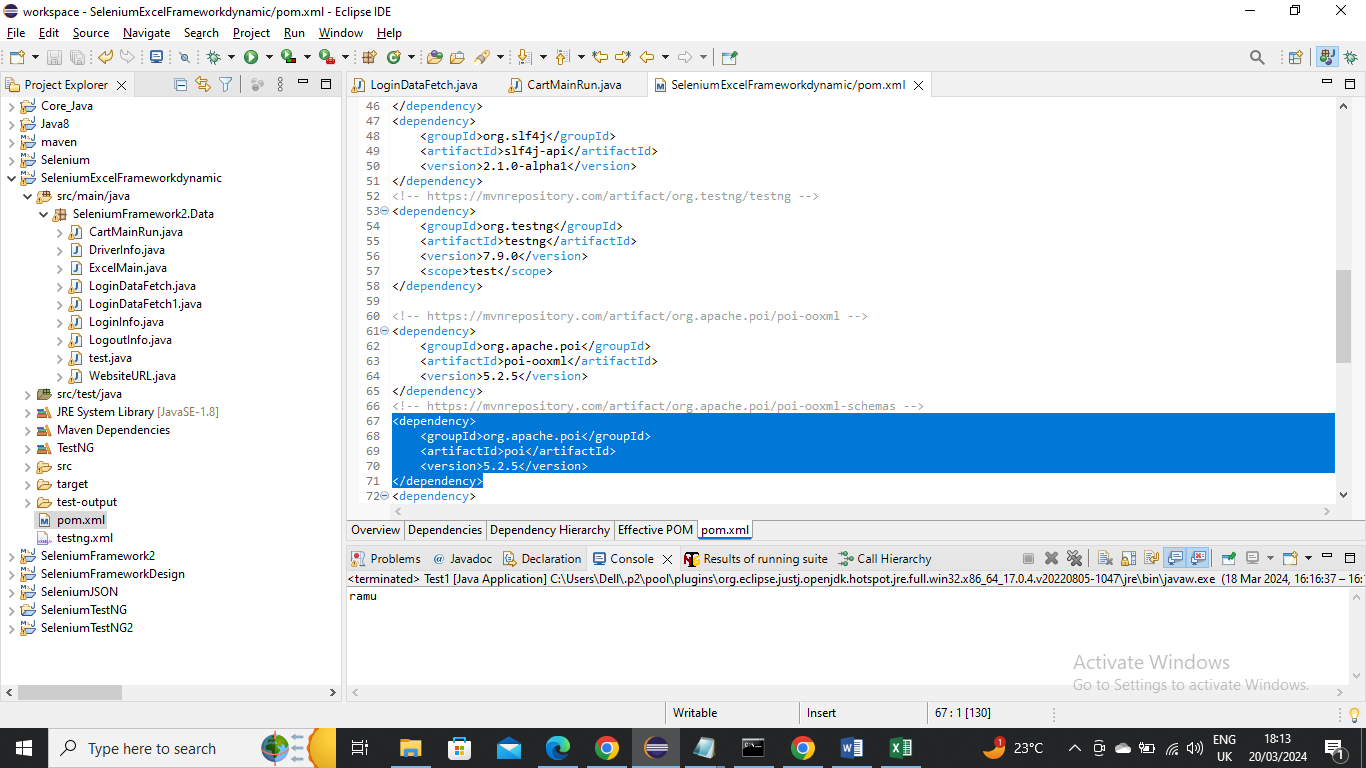




Now let’s add POI dependency





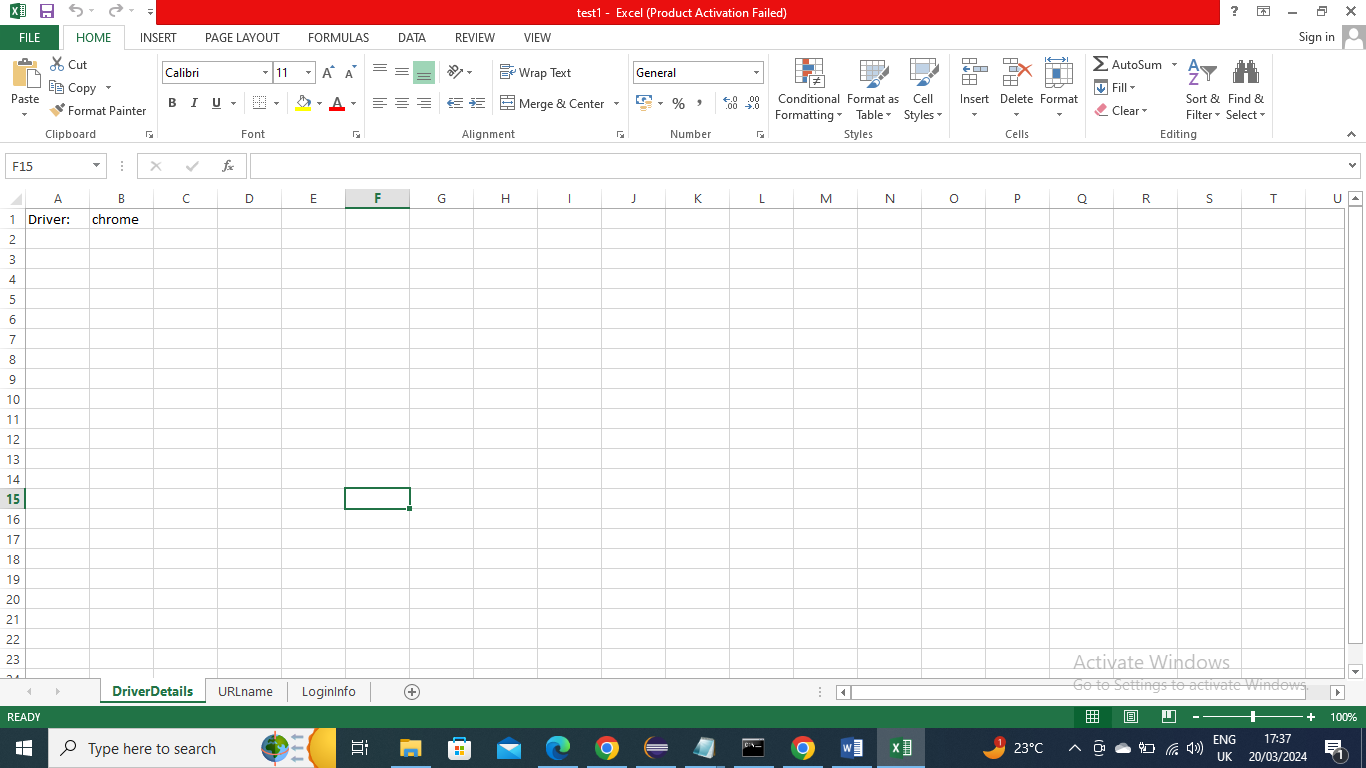


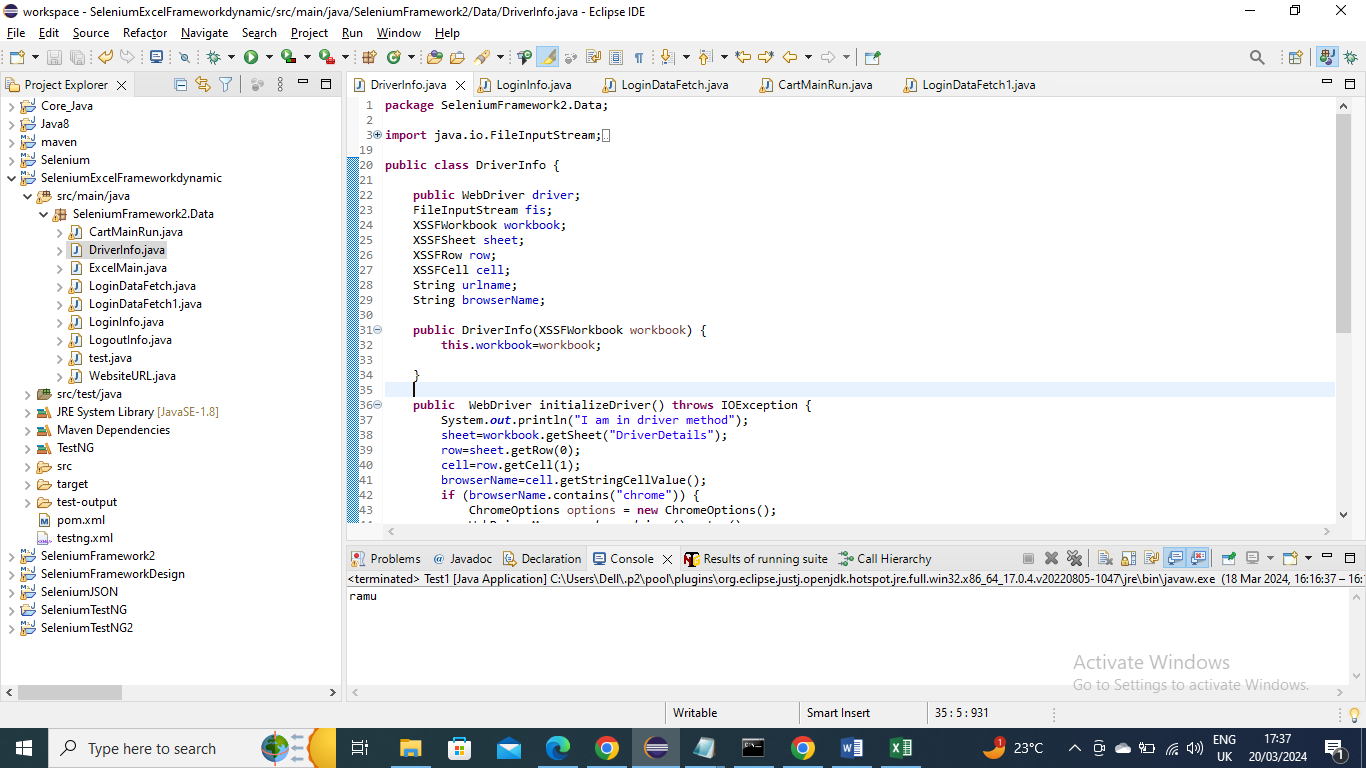
----------------

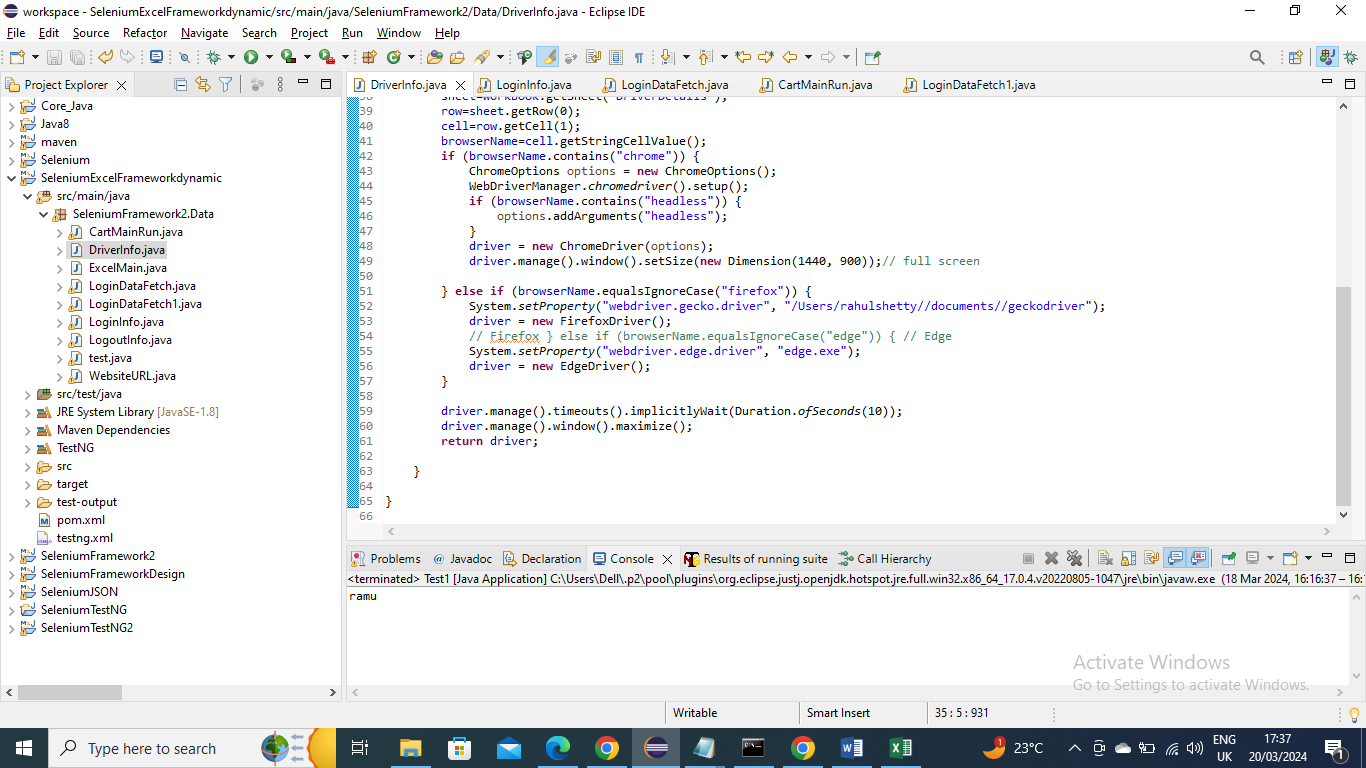
Now lets start using Excel in TestNG framework

The Objective is same like JSON where you need to login and logout using some credentials

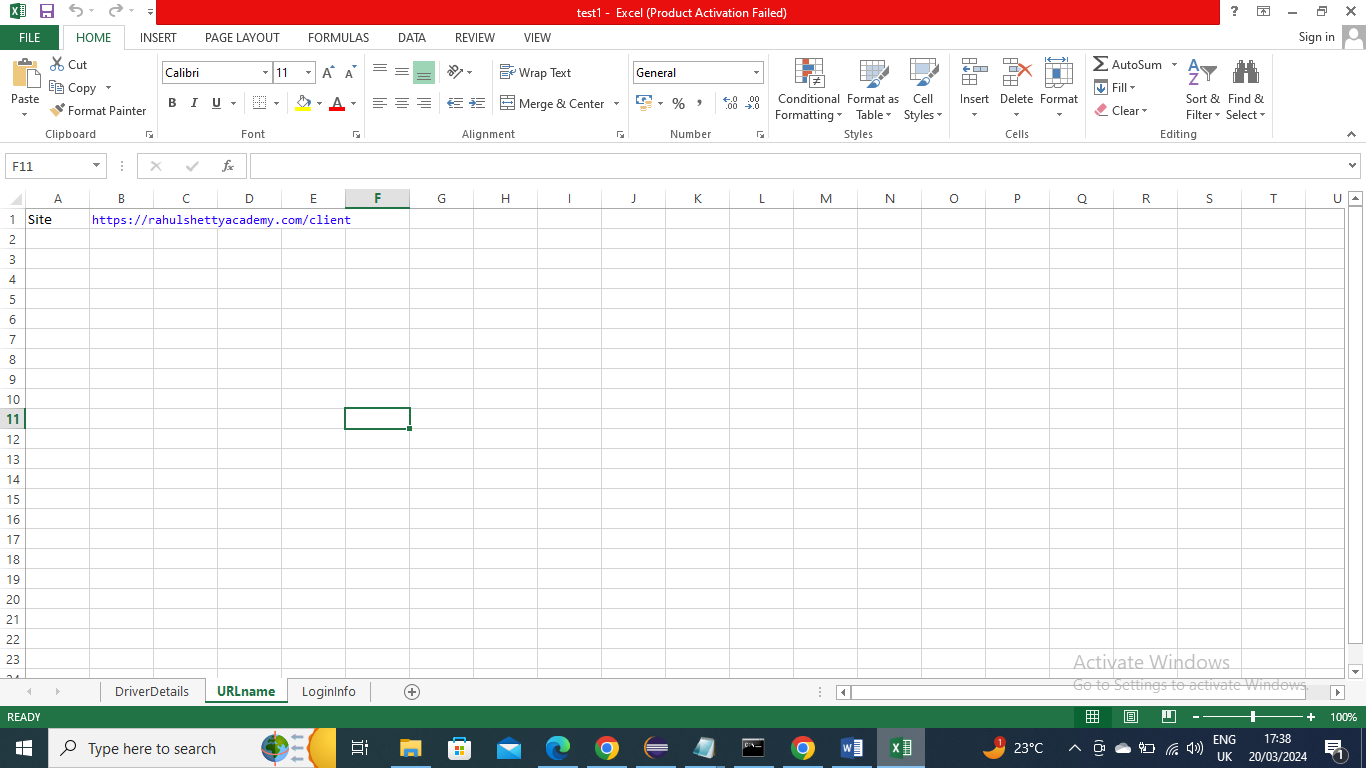
Driver – Using excel sheet that having driver information we are initializing driver in our program

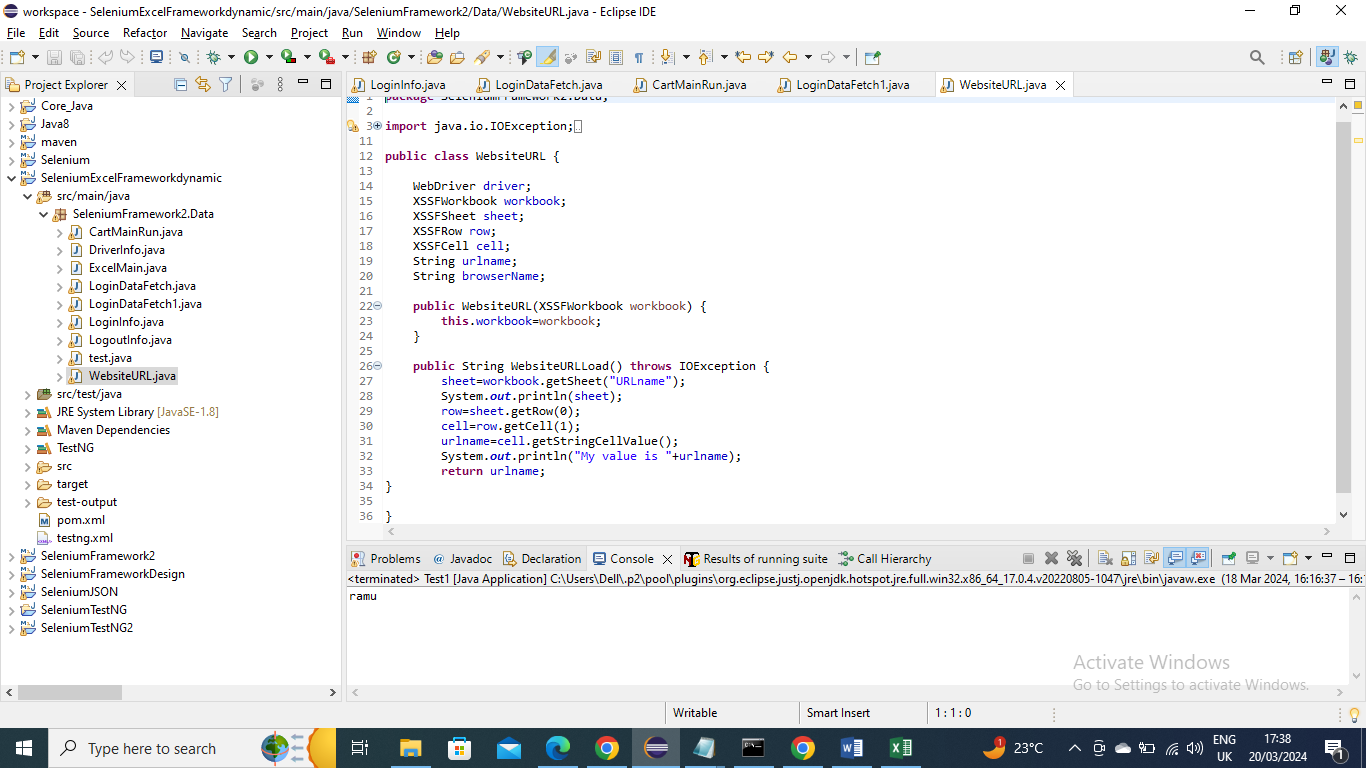






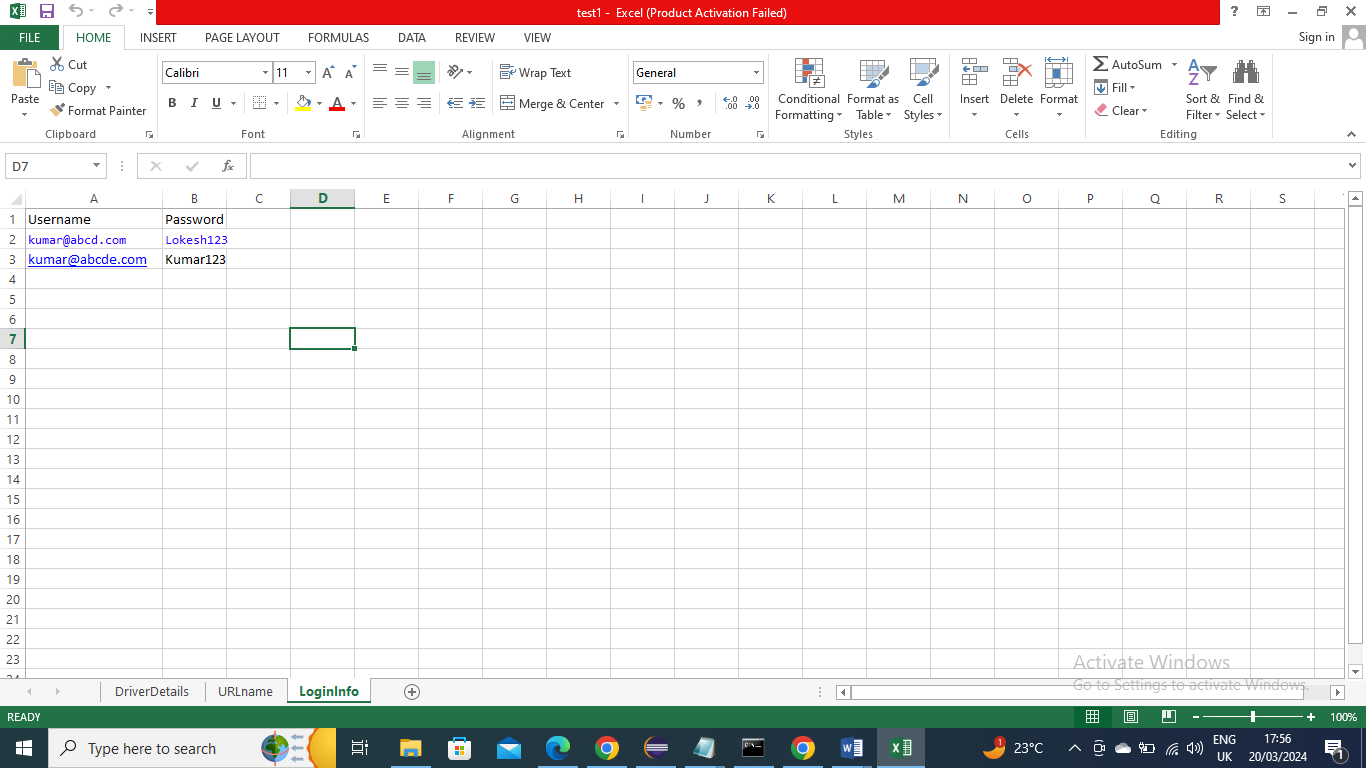
Website name – Get the website url from excel sheet

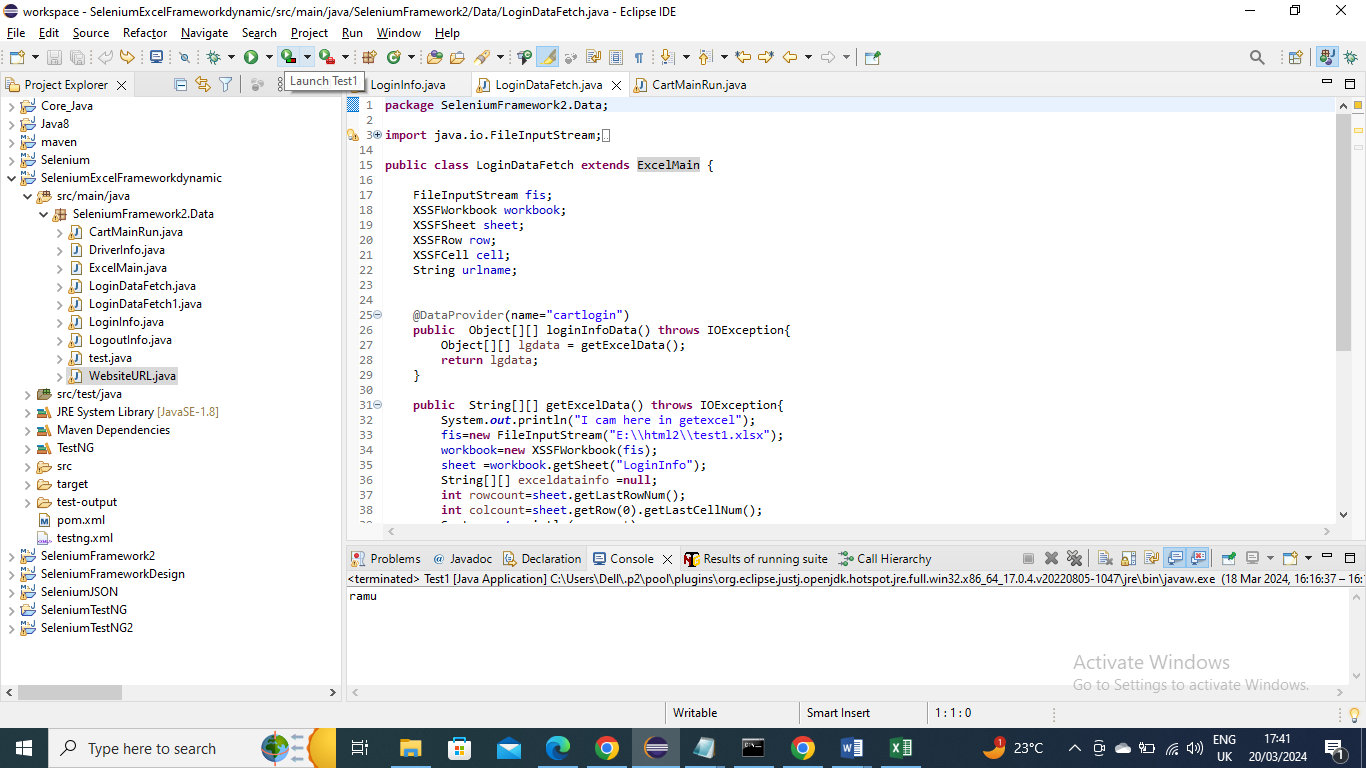


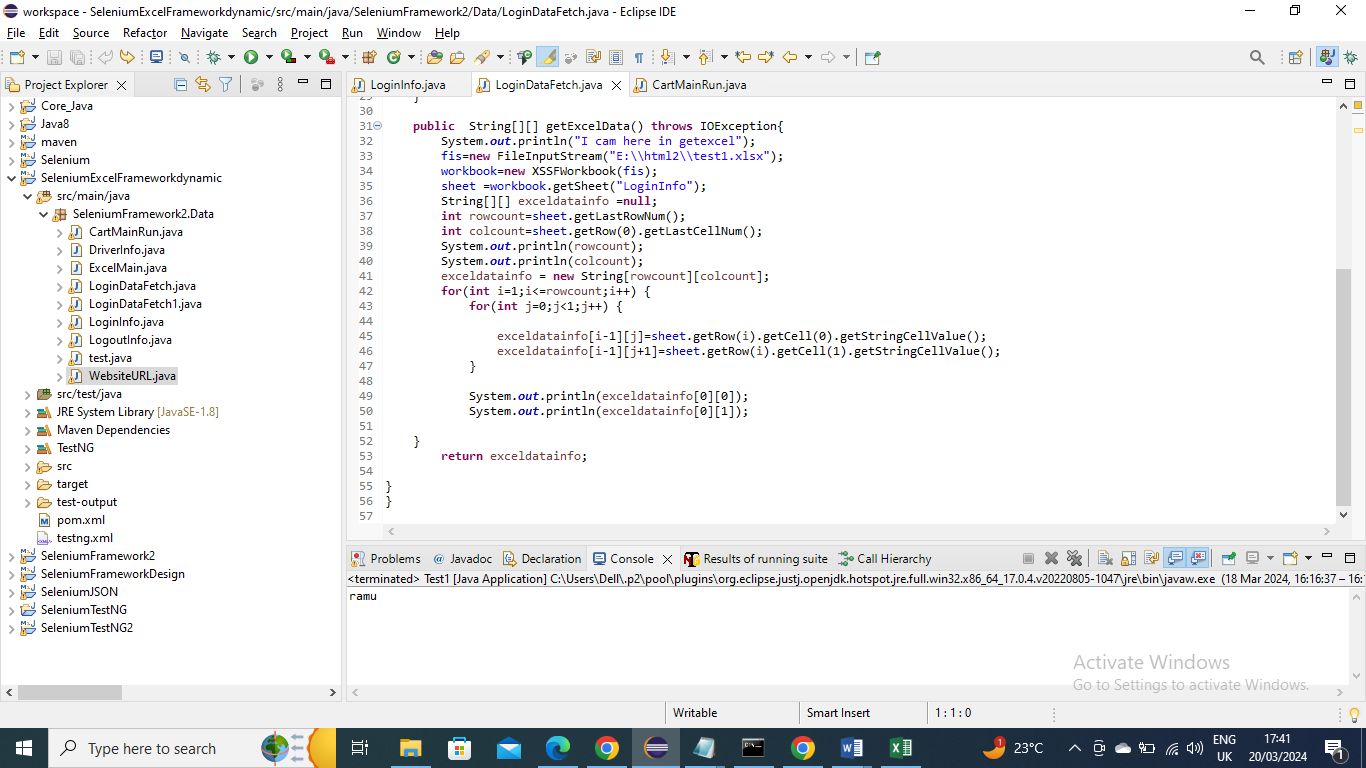


Login Credentials data fetch – the data is stored in Excel sheet file and program reads the data and returns back to called program

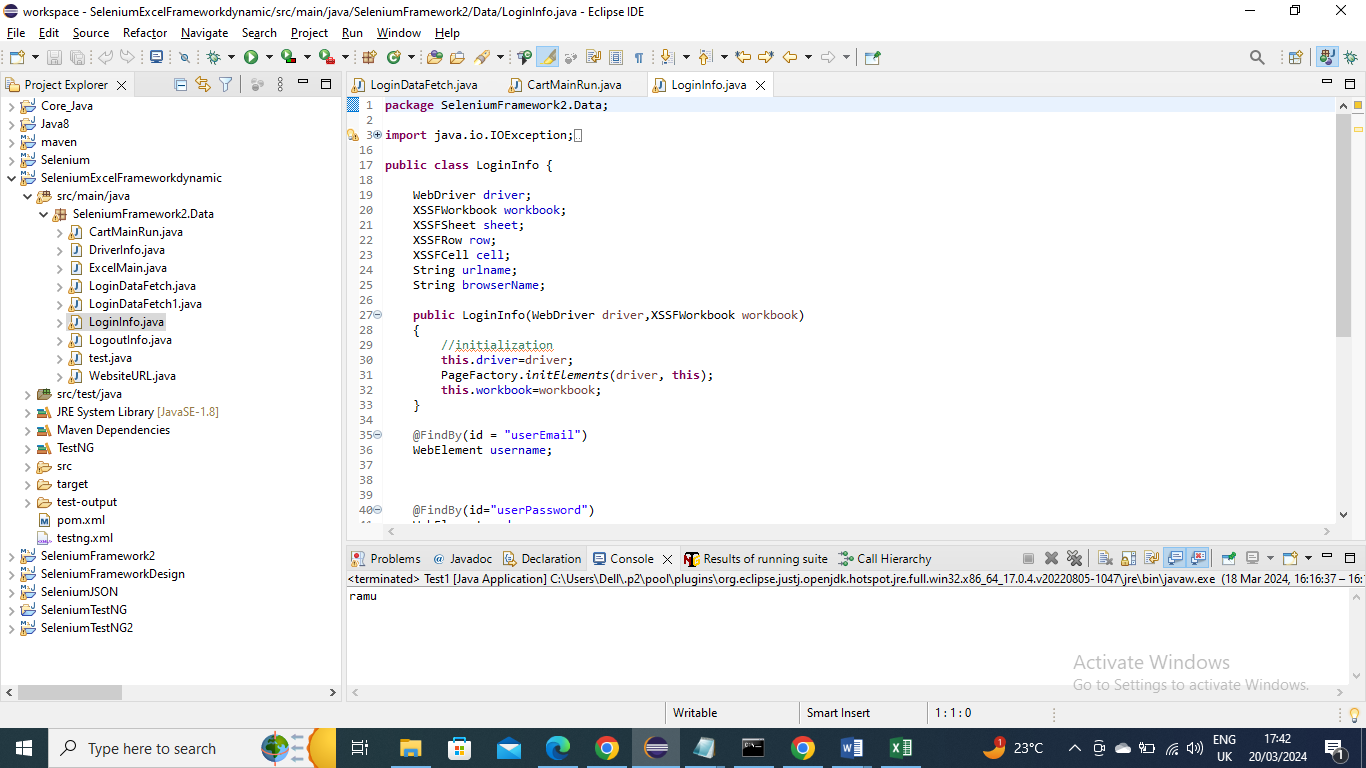
We need to provide @Dataprovider and create a name so that we need to pass this values to the method which wants to receive them that method also needs to say I want to receive data by giving Dataprovider =”cartlogin”

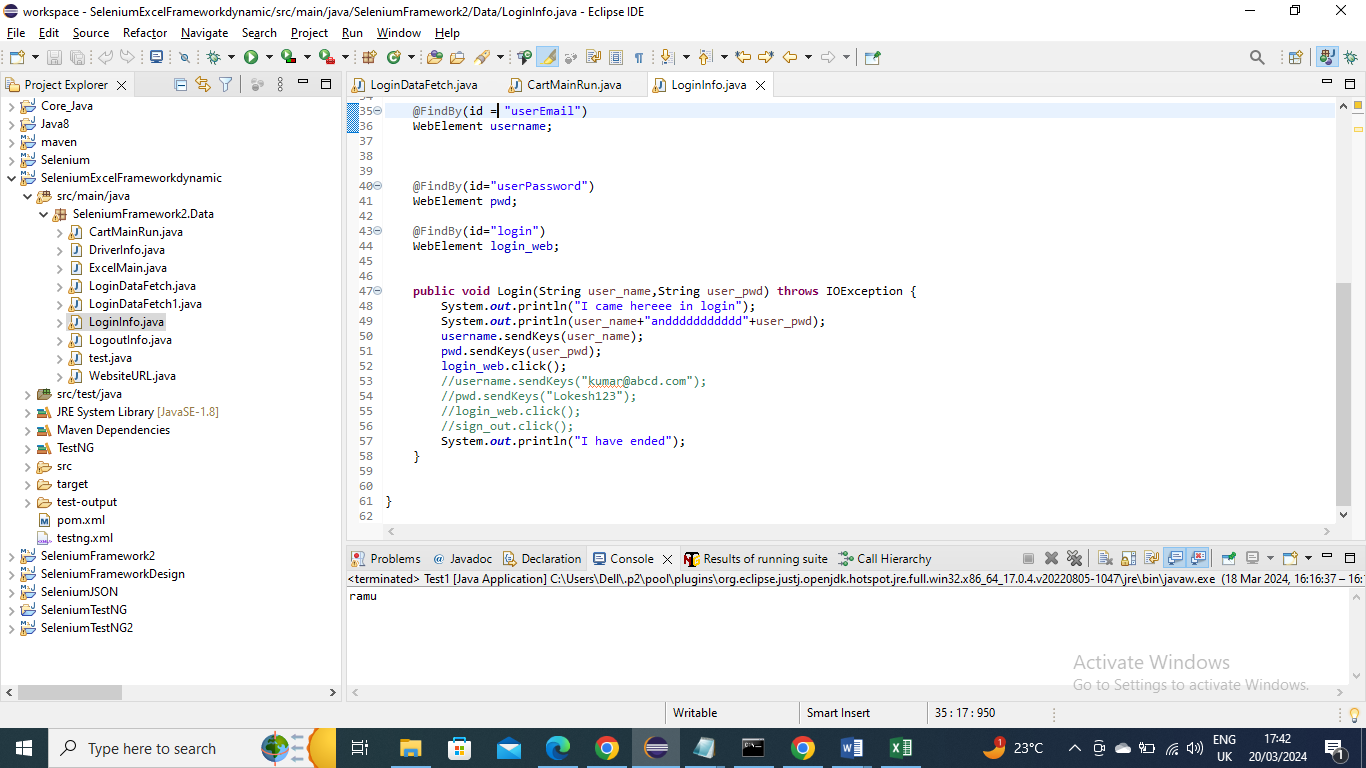




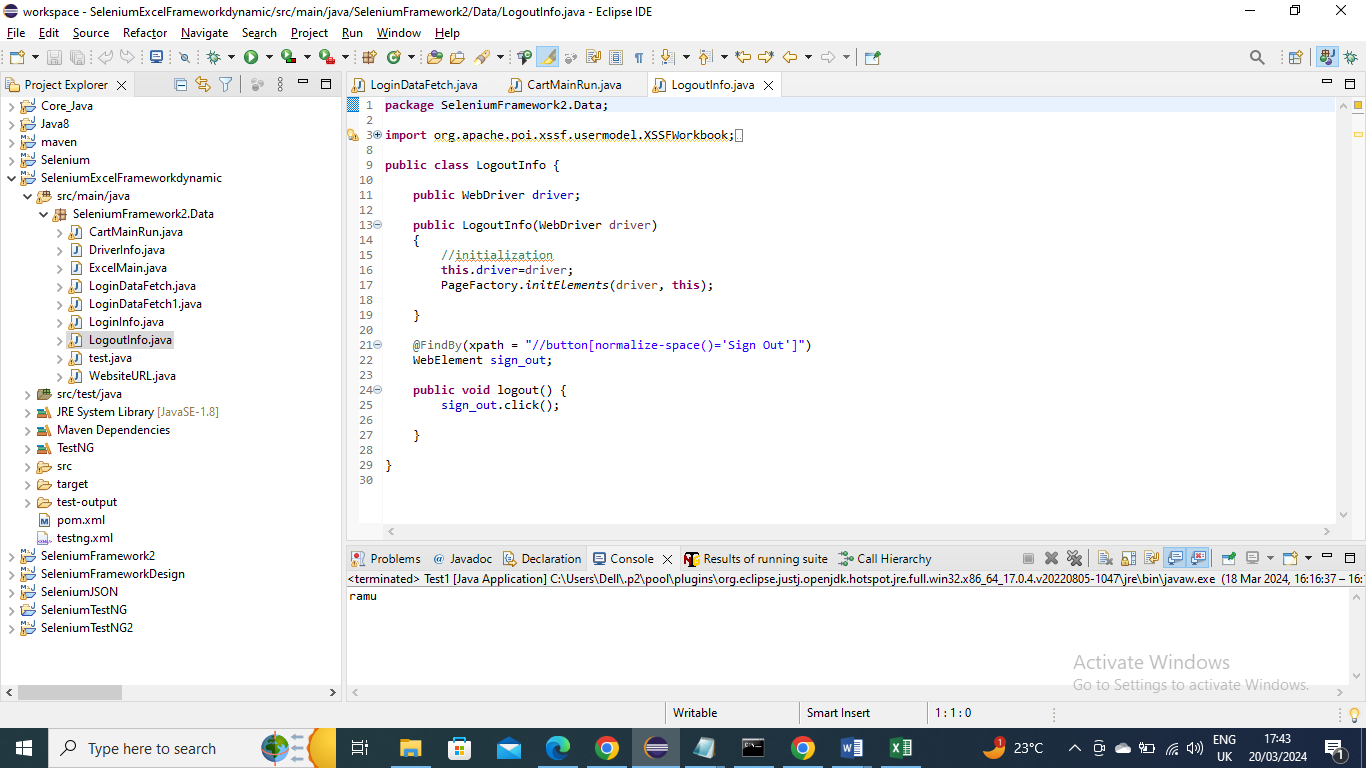


Login Information of paths (id’s) and clicking – In below program we provided the paths in @Findby and create a method Login where we give information on passing values and clicking it

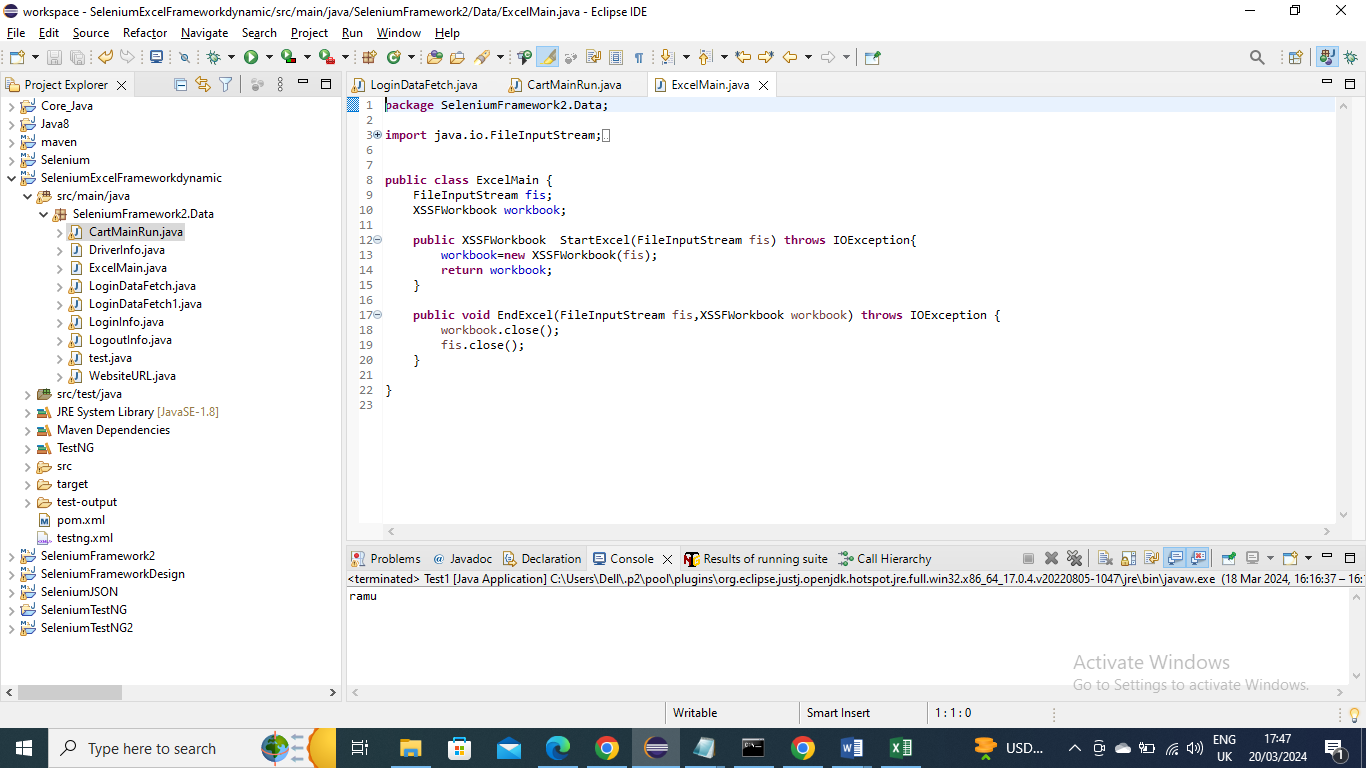




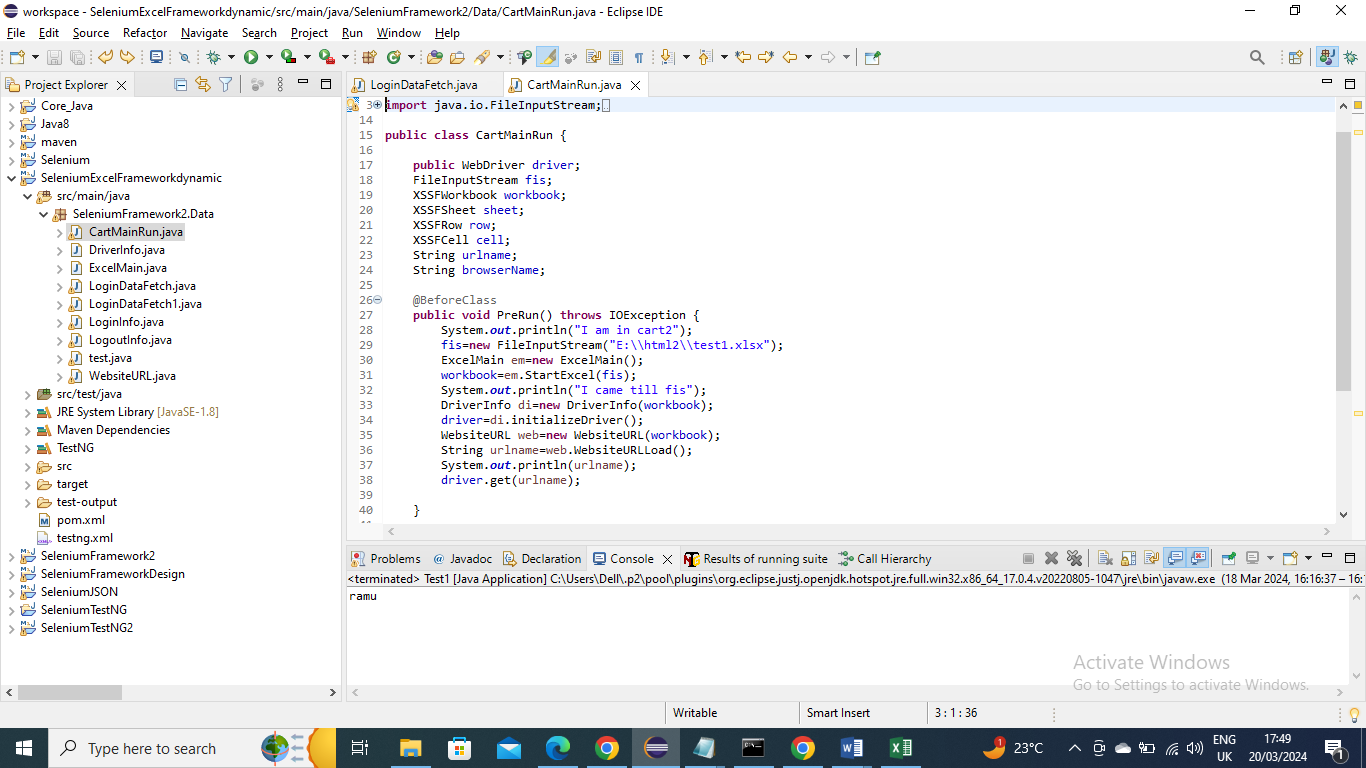
Logout Information – In below program we gave information on logout path(id) and method for sign out click

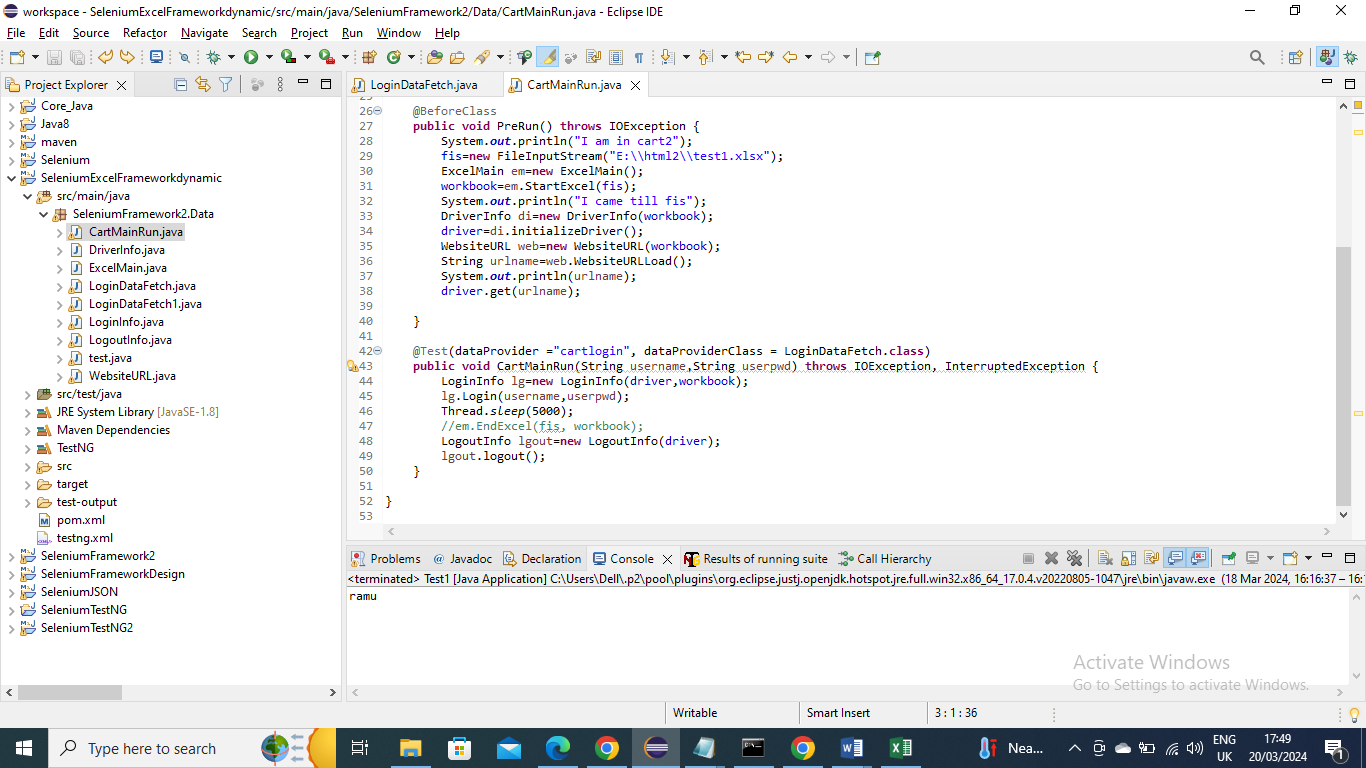


MainExcel – this program is one method just to initialize the workbook and later method is to close the workbook

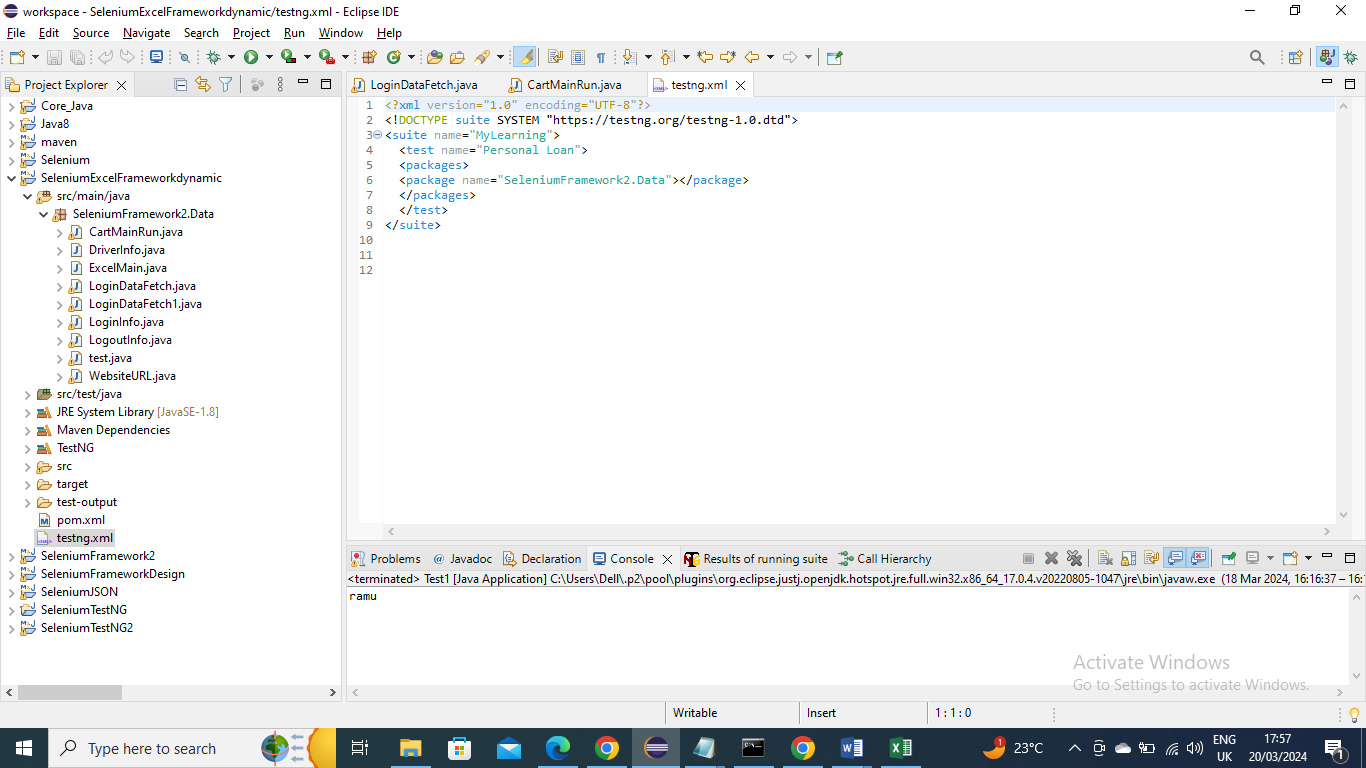


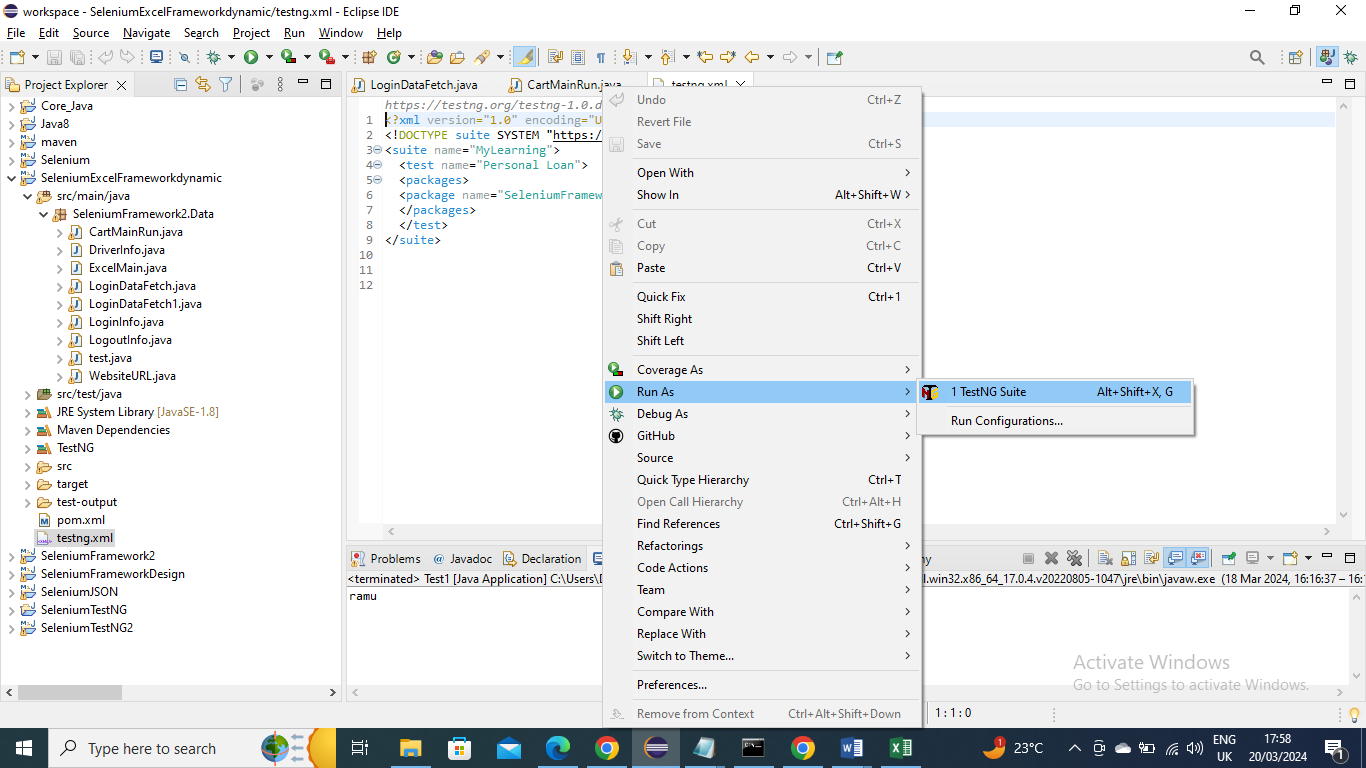
Main program (CartMainRun) – Now the below main program will instantiate wherever required and calls its methods to get the values and in another method to get the excel values it give in @Test annotation the Data provider name and in which class it is present



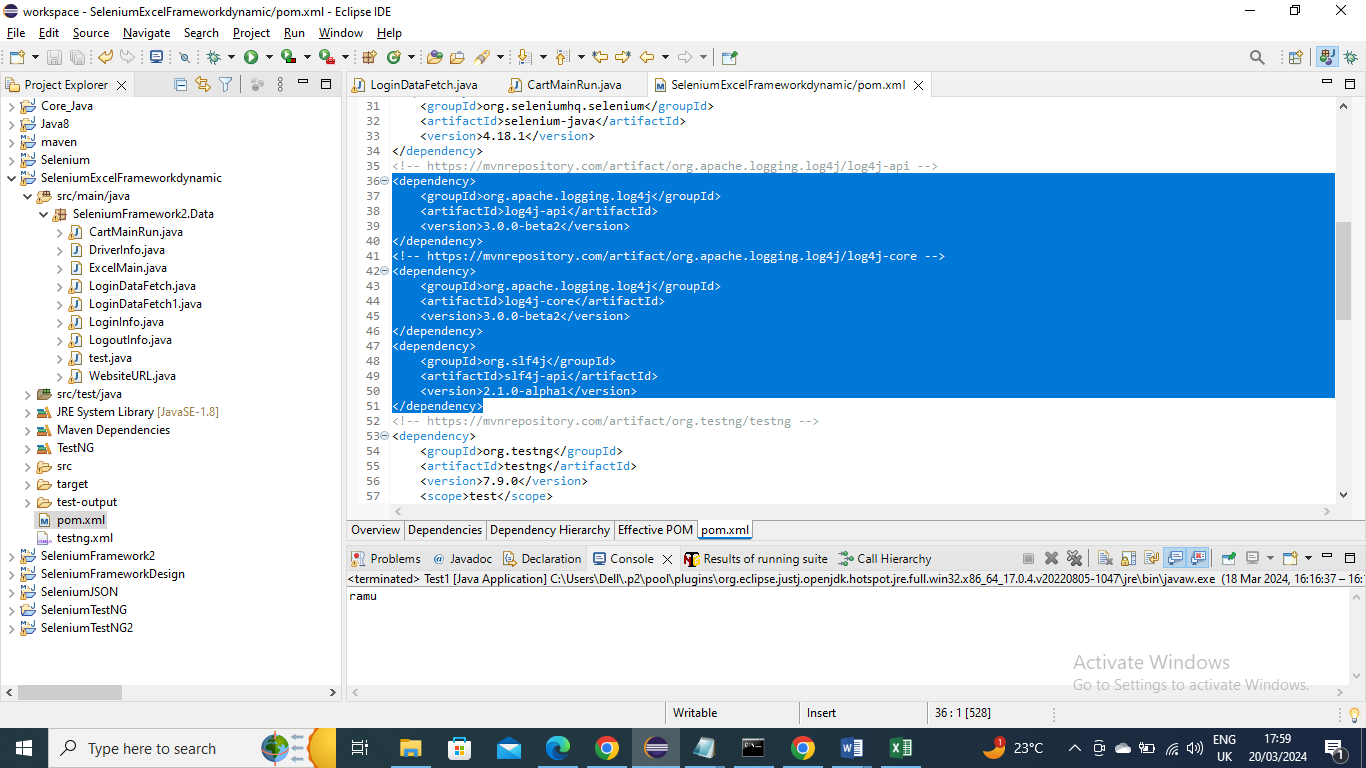


Run the Test – provide the Suite name and test name and give the class or package that you want to run and then right click and run as “TestNG Suite”

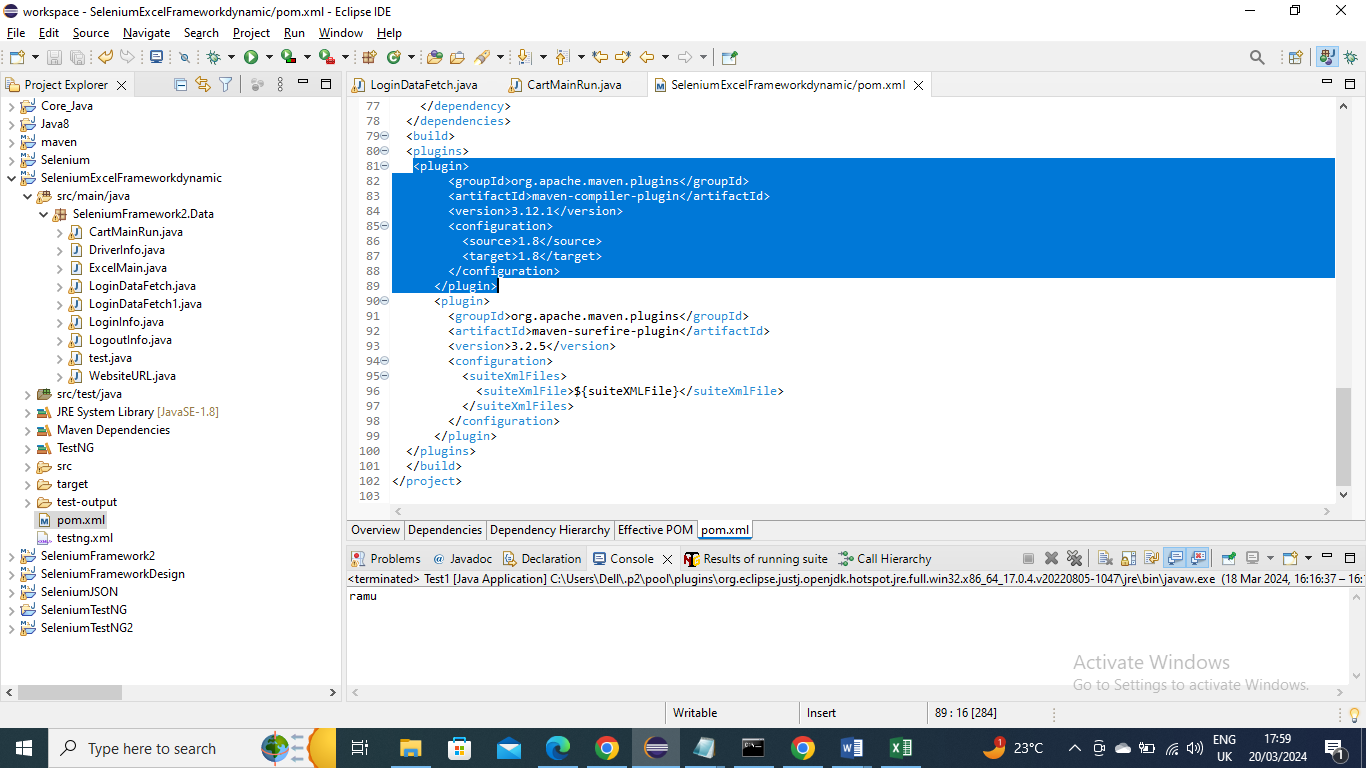




Getting errors - If getting errors of log4j or slf4j issues then add below dependencies



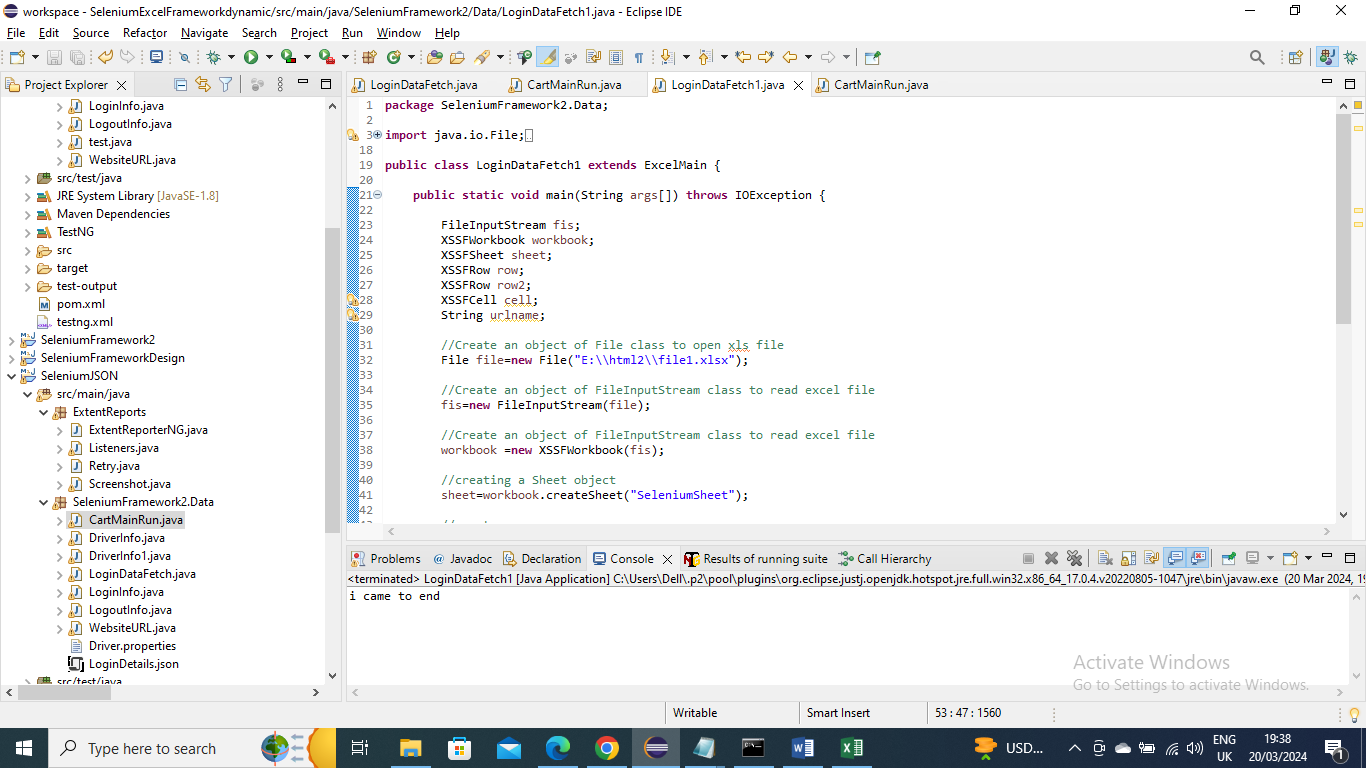
If getting error for Java 1.8 then add below plugin in pom file

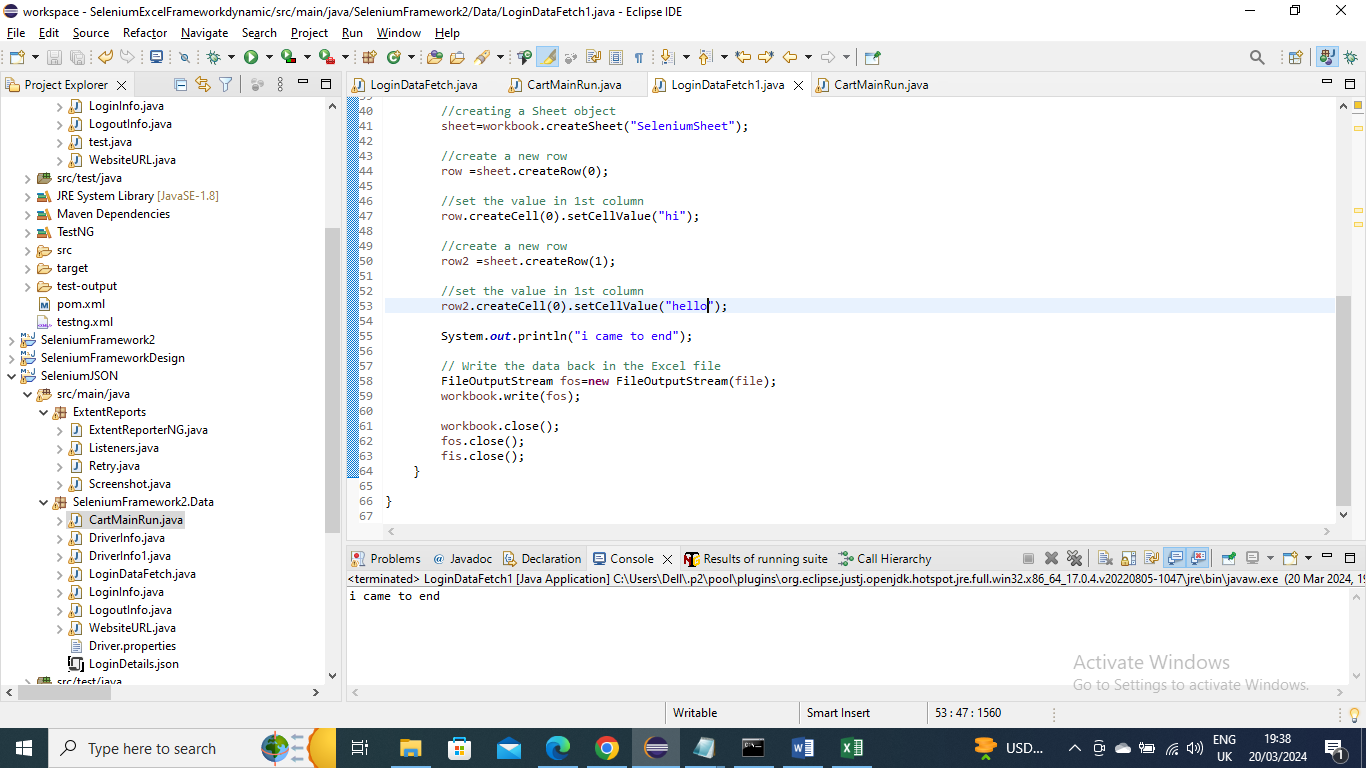


Now for retry or screenshot or extent reports and how on run on Maven you can follow JSON and maven document since its not excel specific it is generic

---------------

Below is an example to write data into excel newly please note you have to create an excel in given specific location





Output-

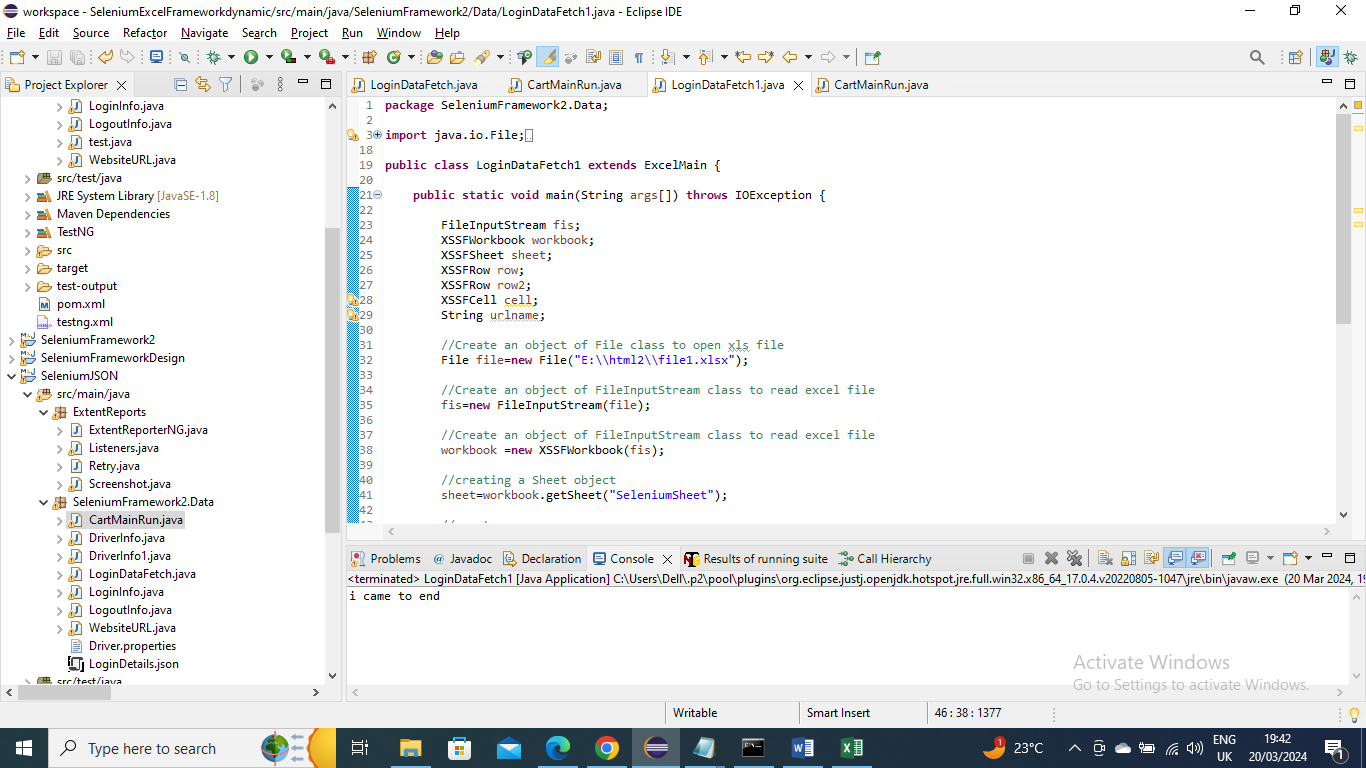


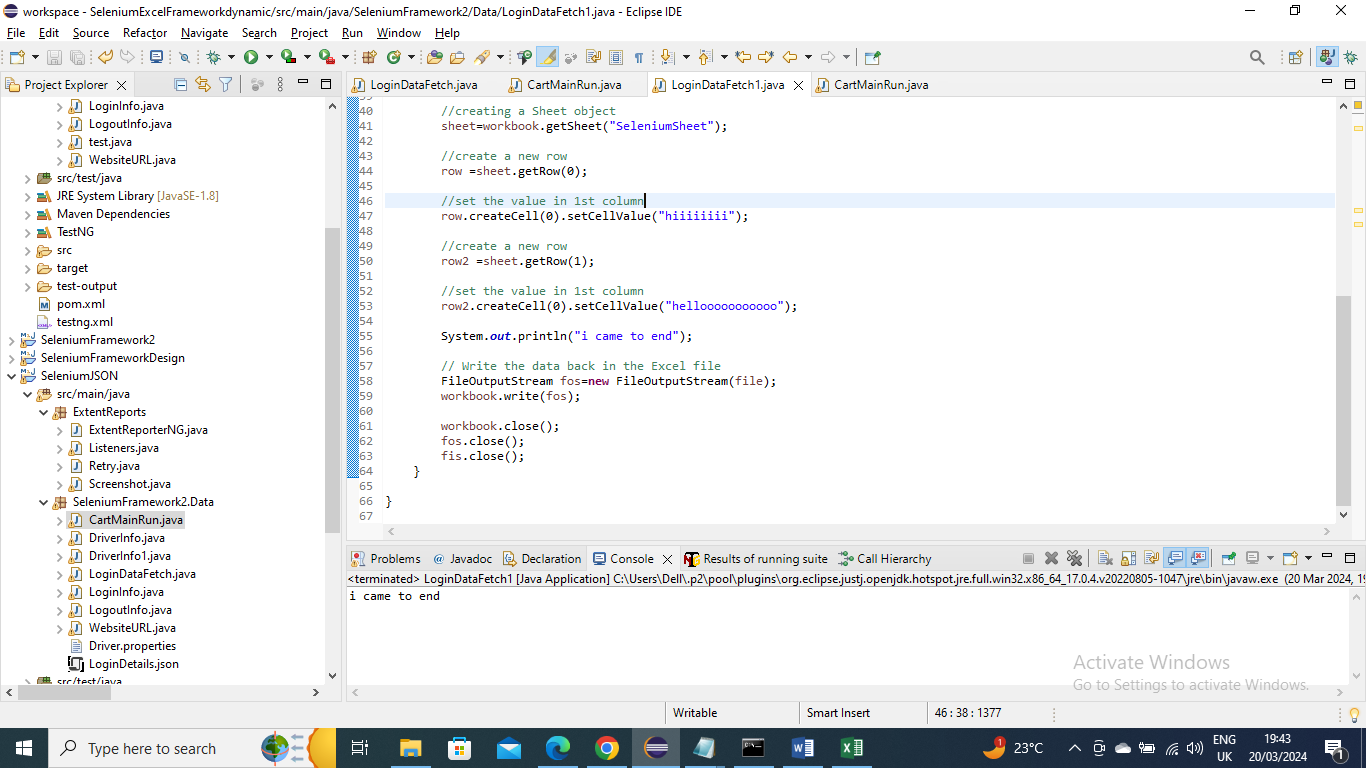
Below is an example to edit data in excel

Before -

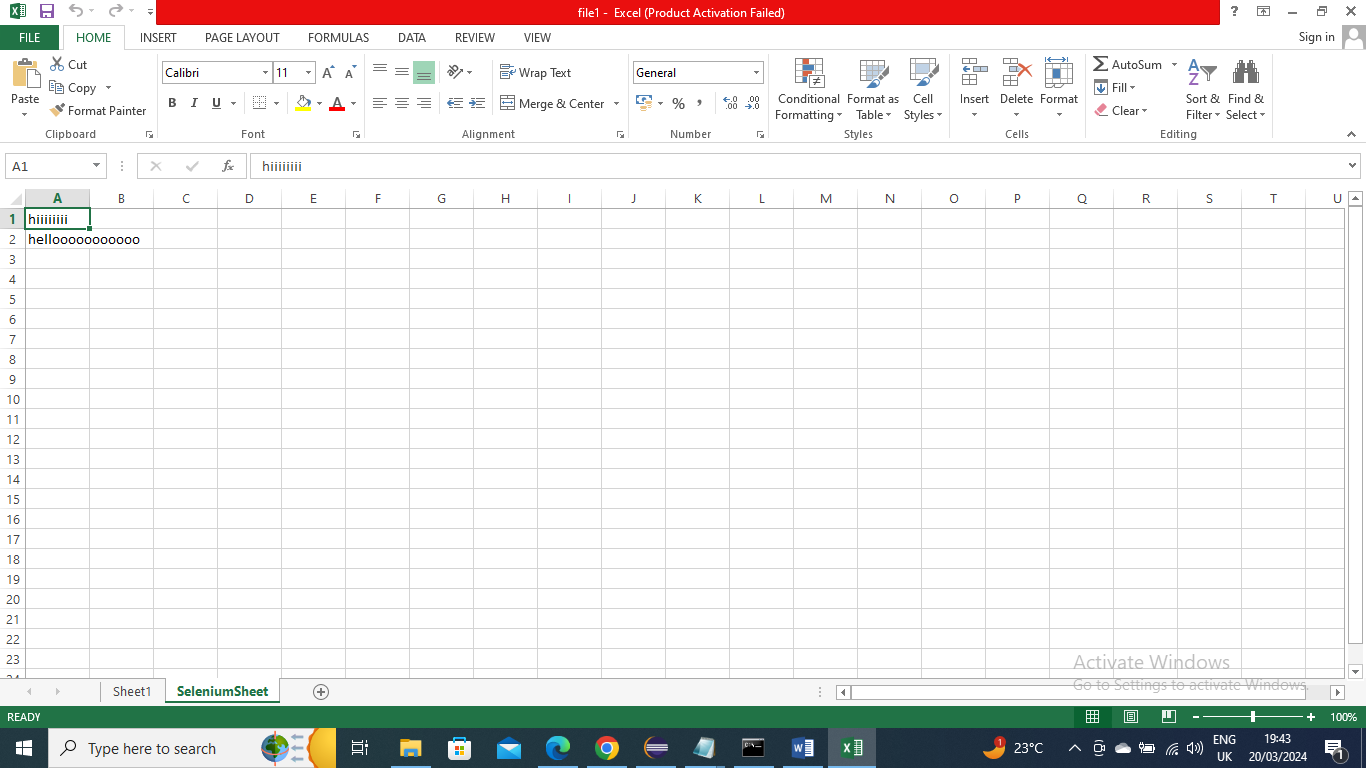


Program –





After (output) –



------------------

If you want to upload excel you can do something like below

driver.findElement(By.id("Upload id/xpath or anything")).sendKeys("E:\\html2\\file1.xlsx");

here E:\\html2\\file1.xlsx – is the path for example