**Assumptions and its implementation**

**Language used :** java

**Project Setup:**

Test Framework – Cucumber: why?

Cucumber is a collaboration tool supports BDD approach, where any individual can write an executable testcase. Gherkin is an important feature that gives the transparency that business also understands the functionalities that has been automated. With Jenkins, the living documentation and cucumber reports facilitates easy report generation.

**Page Object Model:**

**The structure is:**

src/test/java - contains all .java class

Actions – is a package that contains classes that classes in other packages are dependent of

* ActionLib.class - contains all methods which is basically needed for a selenium automation like, waits for page to load, wait till the element is present/displayed, is the element clickable.
* ExtentReporter.class - is for creations of automation test report and taking snapshot of the screen
* TestDataExcelRead.class - read the input from the excel that is required for the test run, here for excel read and wrote I have used Apache poi api.

JunitCucumber – package that contains runner class

* RunCucumberTest.class - is the starting point for the test run contains the feature/scenario to be executed

Manager – package that contains page object manager

* PageObjectManager.class - Have used this to create objects for the classes created as in step definition instead of creating objects for each and every step of feature, page object manager lets us create object for the first time and reuse the object for later steps.

Pages – contains all the classes where actual functionalities are performed here, I create class for each page the applications contains, all the functionalities to be performed in that page will be maintained in single class, say here I have automated the first page, so I have created one which is SearchPage

* SearchPage – Contains methods that performs, selection of category , entering item , Click on search and fetching results

StepDefinition - package contains step definition class

* StepDefinition.class - For each step I have called the method that performs that functionality

TestBase – package that contains TestBase class

* TestBase.class - is initialized before the feature run, this class contains methods for selection of browser, launching url, opening and closing of browser

src/test/resources - contains feature files and some properties file

Resources – extent-config.xml - For generation of extent report

TestFeature – contains all the feature files required for the execution

**Page factory:**

In all the classes page factory has been used, as it gives greater advantage of creating objects using @FIndBy annotation.

**Locators are declared and parametrised to improve reusability**

**Assertions:**

Assert.fail - have used in all catch blocks as I want to stop the execution when there is an exception

Assert.assertTrue - where I want to check a condition and continue the execution only if it's true

Assert.assertNotNull - where am afraid that the string might be null

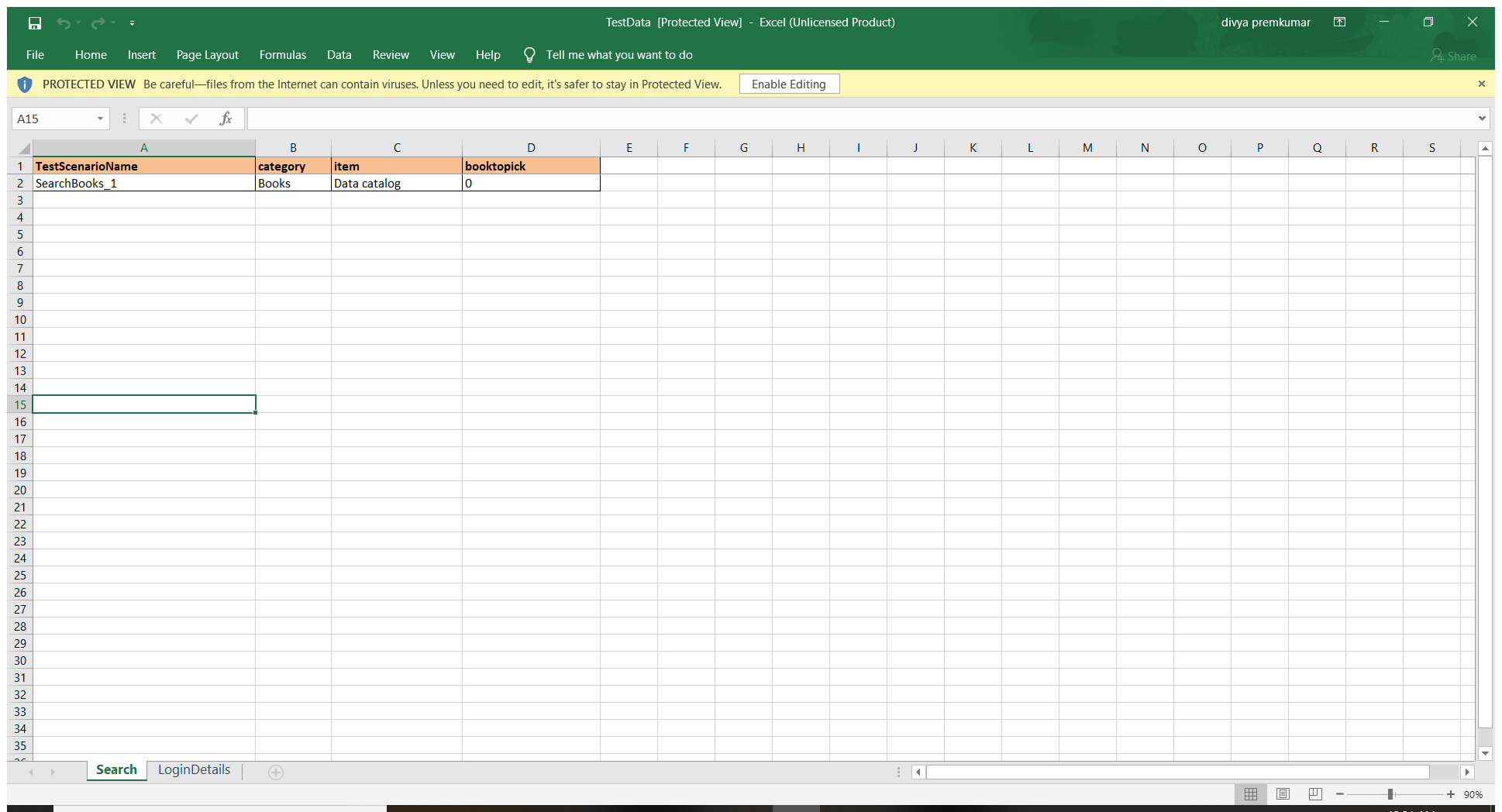
Not much assertions I was able to use as the functionality that has been automated doesn’t do much comparisons or check any conditions. Necessary assertions have been inserted.

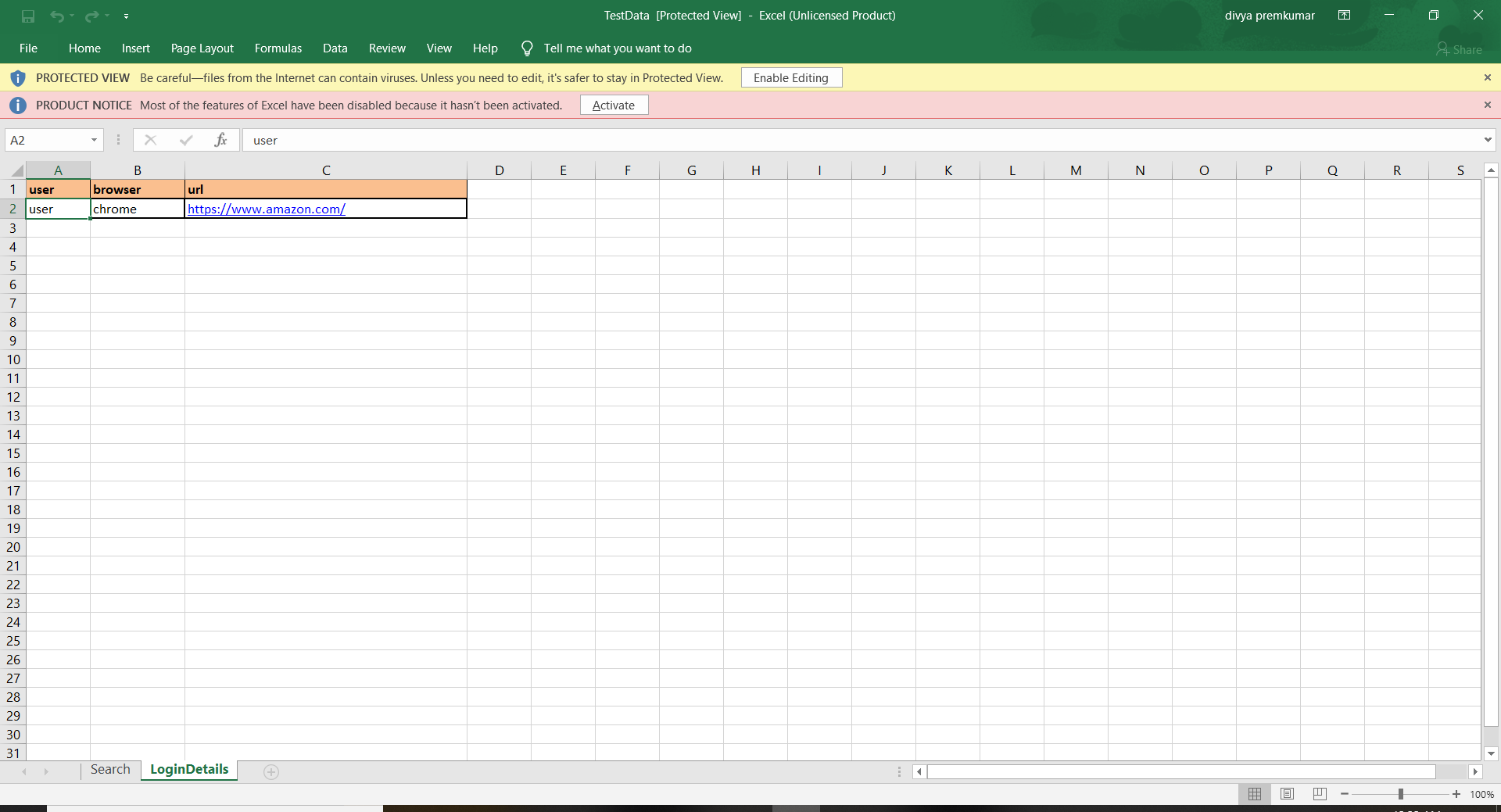
**Loggers:**

Have used loggers wherever I want to keep me informed, that actions have been performed and to see what values have been entered or selected. Makes it easier while debugging. Logger.info served this purpose well.

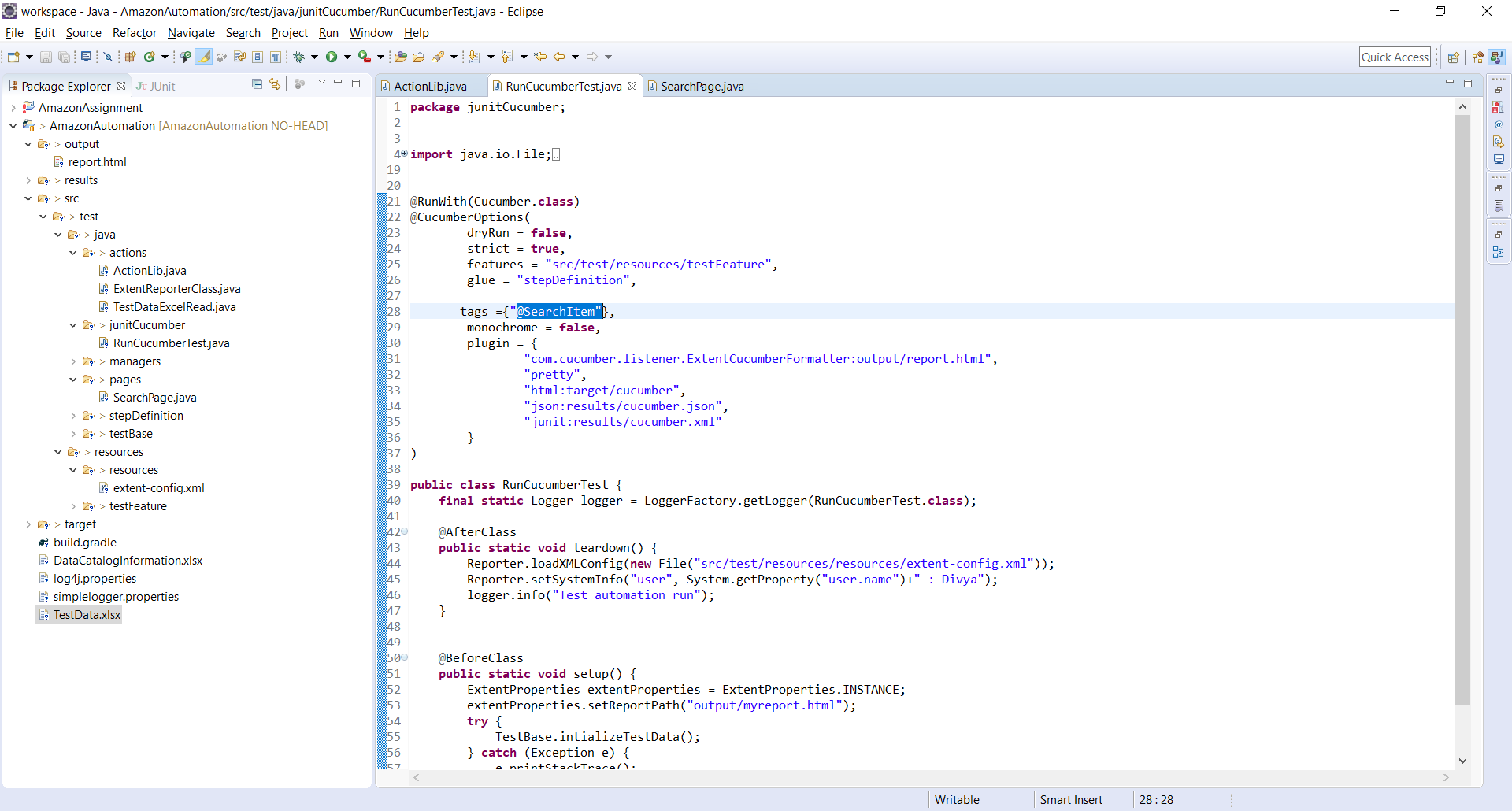
**Execution:**

1. In **TestData.xlsx, Search sheet** Input the Category, item and the number of results for which info needs to be fetched and Provide browser and url details in Login details sheet

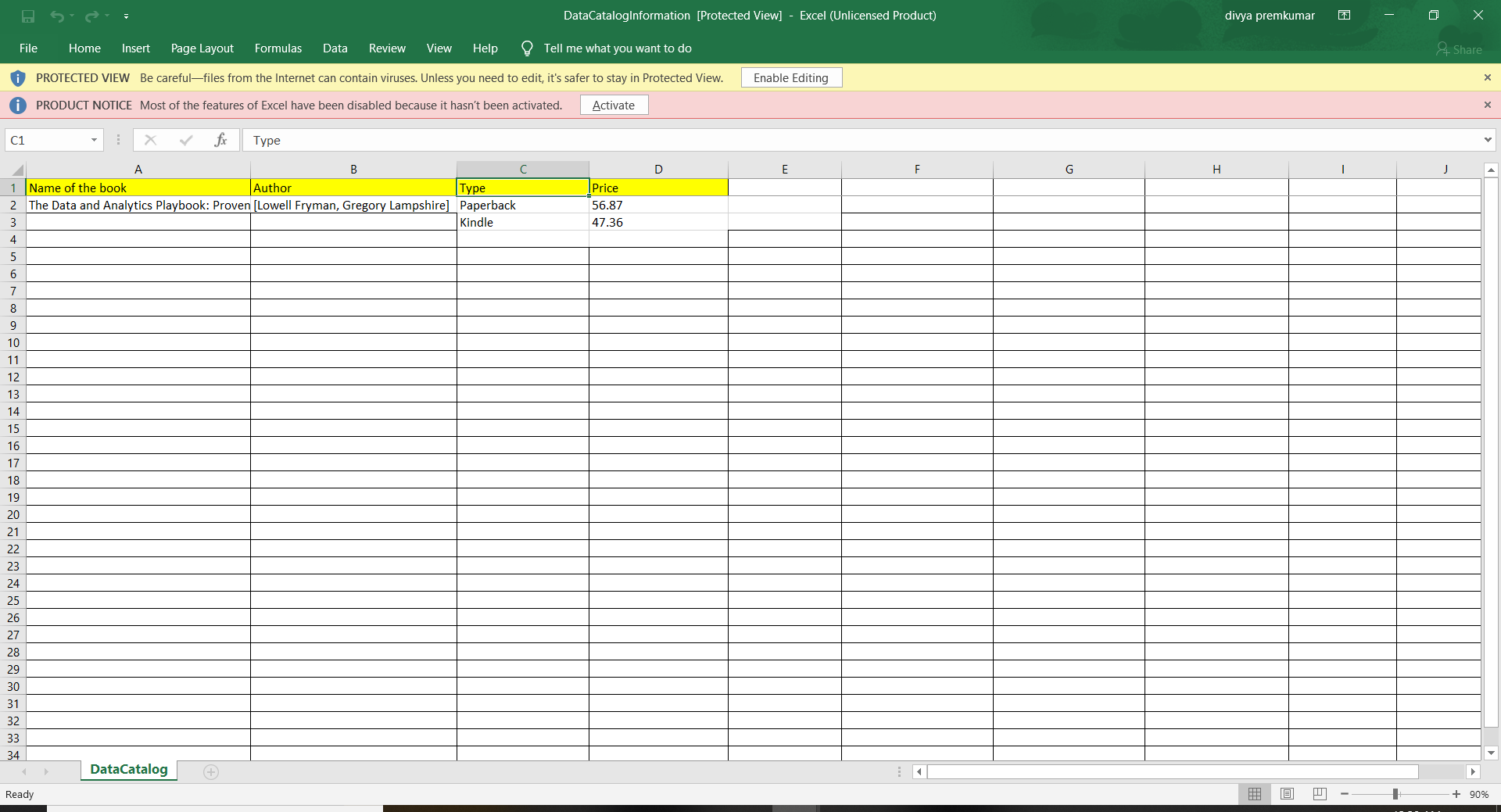




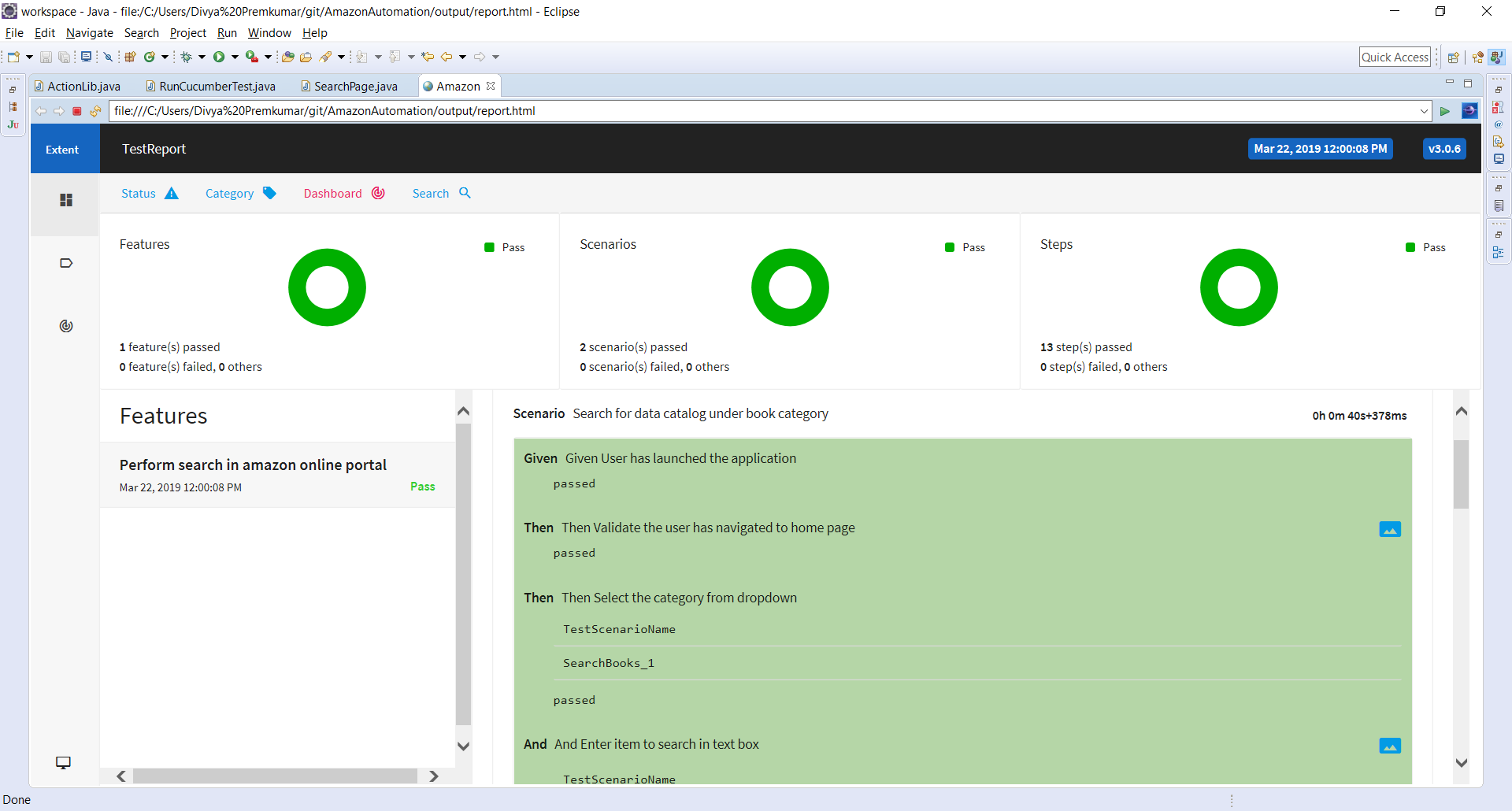
1. Provide the name of the feature/scenario that needs to be executed in RunCucumberTest.java
2. Right Click and run it as junit



1. The info of the book is stored in DataCatalogInformation.xlsx



1. After execution verify the report under, output/report.html , under each step the screenshots will be loaded.



**Instructions to execute in Mac environment**

* Driver for Mac OS should be used
* Rest is same