**TDD-Test Data Driven**

The traditional process of development is once the QA gets the requirement from PO/BA, they will first write the test cases.

From these test cases, developer will make the software. And these testcases are executed on the software. If there is any changes required, you will refractor it. These methods are too technical, so that other stake holders cannot understand the TDD process.

PO/BA does not know if all the requirements are implemented or not, there was lot of communication gap, there was no transparency and that’s why the result was not effective.

So, BDD was Introduced. In this QA, Developers and BA/PO sit together and create possible scenario’s out of the requirement. Everyone will be in the same page.

BDD was introduced to bridge gap between technical and non-technical stake holders.

These scenarios are written in the feature file using Gherkin Language, which is a simple English language.

**Example:**

**Scenario**: User Click the Link

**Given** User is on Homepage-precondition

**When** user click on link

**Then** user should see link click confirmation

These Scenario’s should be converted to code in the Step Definition file, which is technical. There is a mapping between the feature file and step definition file. For every scenario, there should be a code in the step definition file.

If there is multiple feature file, we have to use Test Runner class.

In Test Runner, we will define, which step definition is matched with which feature file.

BDD-Behavior Data Driven is an approach and Cucumber is a tool to implement it.

**Application under test:**

[**https://www.demoblaze.com/**](https://www.demoblaze.com/)

**BDD Framework:**

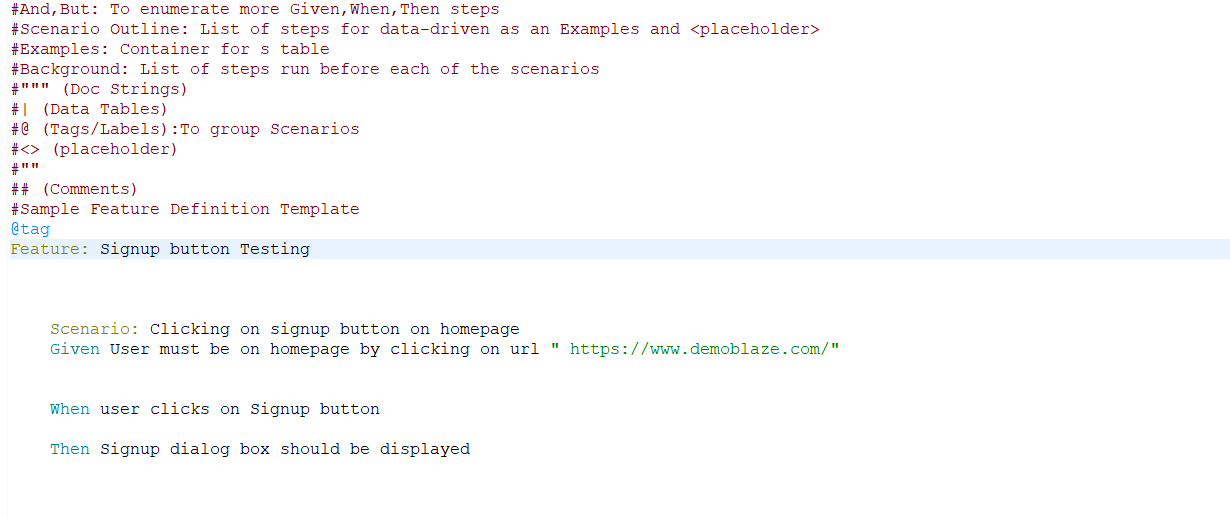
1.Create a Maven Project-BDD Framework

2. In the Pom.xml add selenium, add Cucumber JVM: Java (io. cucumber) and Junit.

3. In the src/test /java create a package as feature.

4.Create a file inside feature package, file name is Shopping. Feature

**Shopping. Feature**

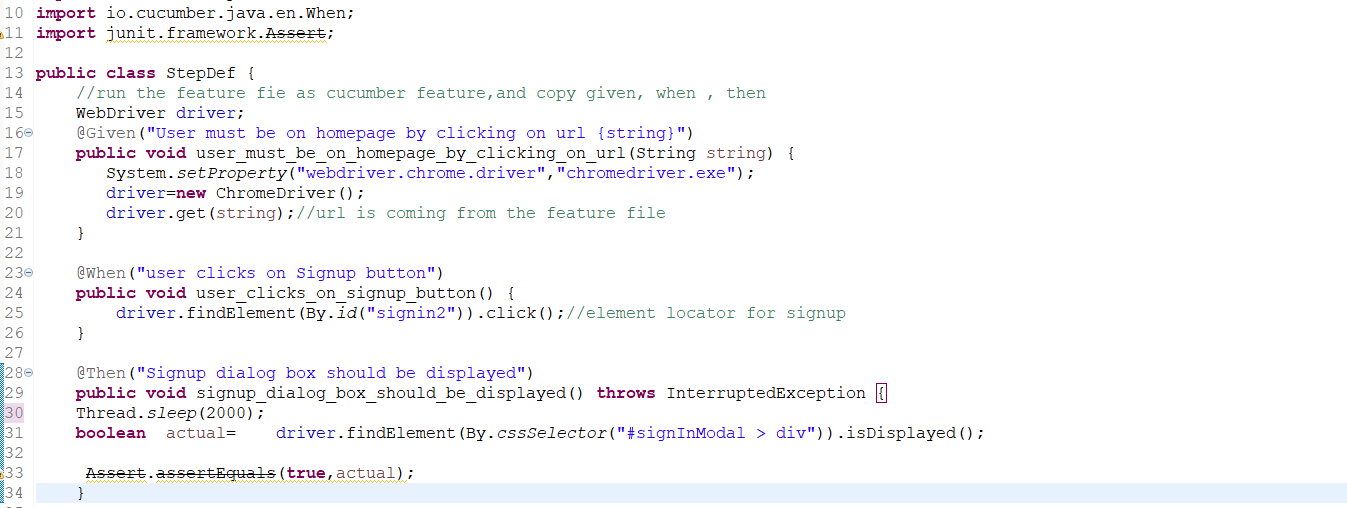


5. If the predefined template is not coming, install cucumber from the Eclipse marketplace

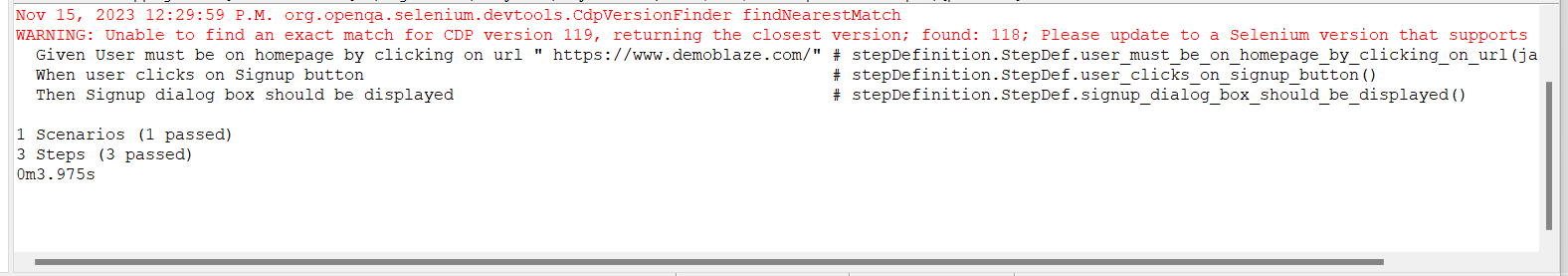
6. Create a package Step Definition and create a class Step Def

7. Add gecko driver and chrome driver to the project

**StepDef.java**



**8. Run the feature file as cucumber. feature**



**Parametrization:**

1.Create another Feature file Login.Feature

2. Application under test: https://

3. If we need to pass multiple value to a testcase, put that element in conical brackets

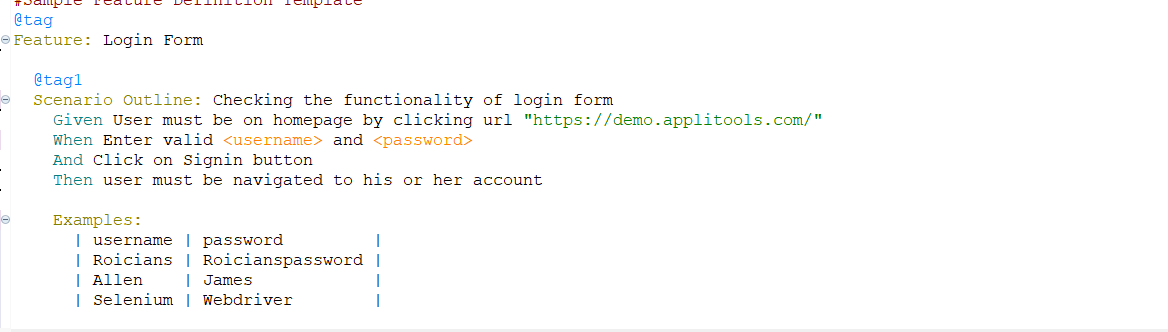
Example:<username> <password>

5. Also, instead of using Scenario, use **Scenario Outline, if wee need to do parametrization**

6. Parametrization can be done using the **keyword Examples**

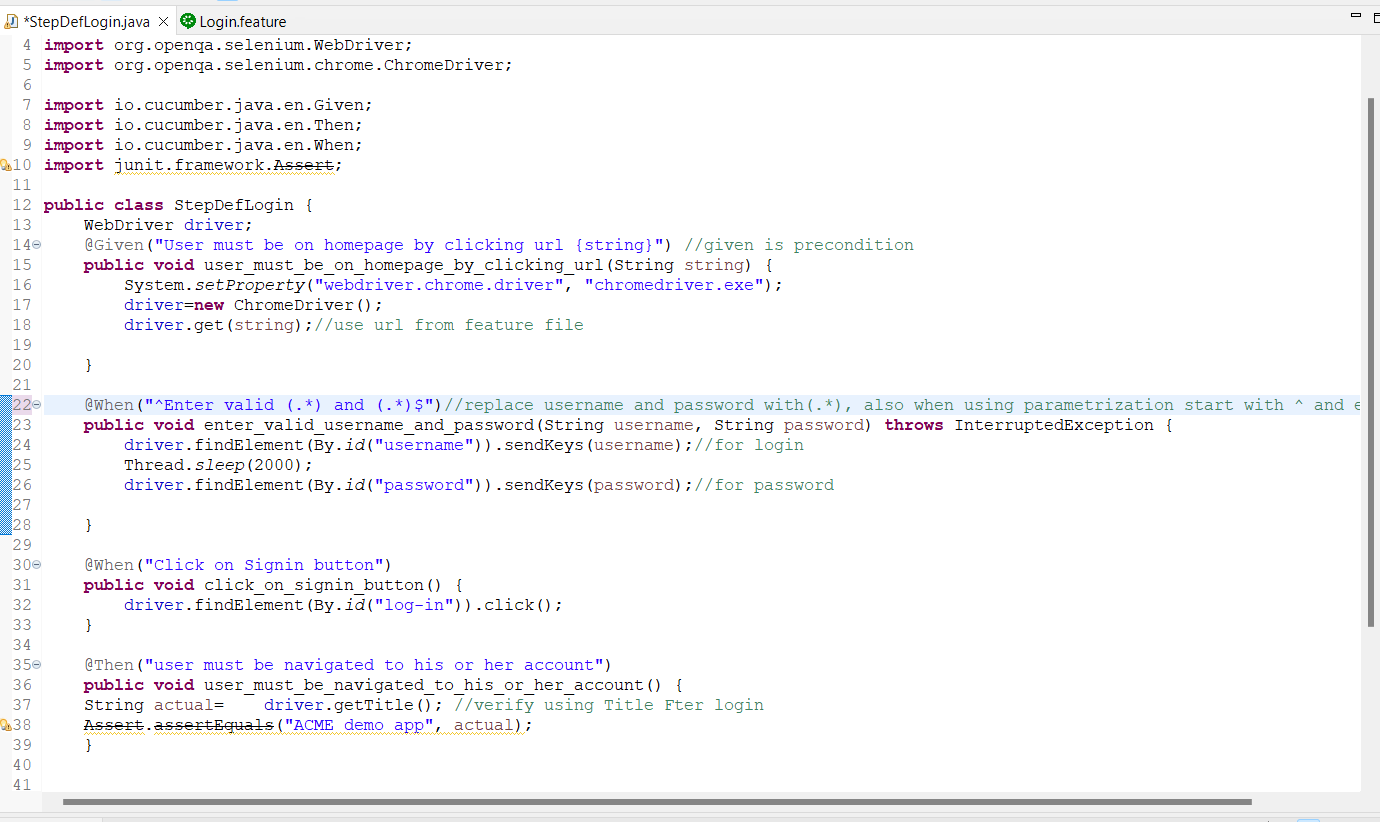
**7. Align the data by right click🡪pretty format**

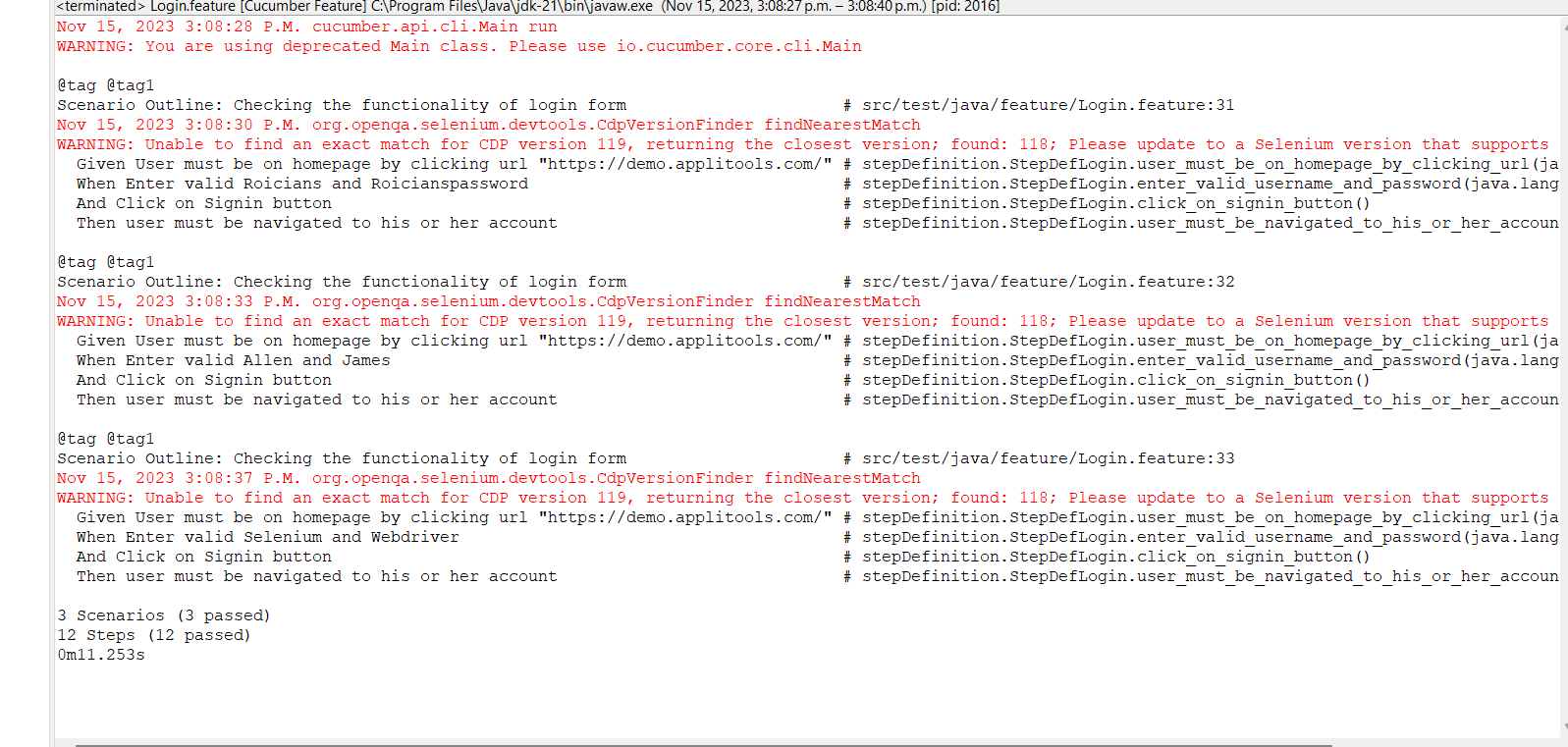
**Login. Feature:**



8. Replace username and password in Stef Definition file with **(. \*)**

9. When using parametrization, in Step definition file, start with, **^ and end with $**



**Output:**

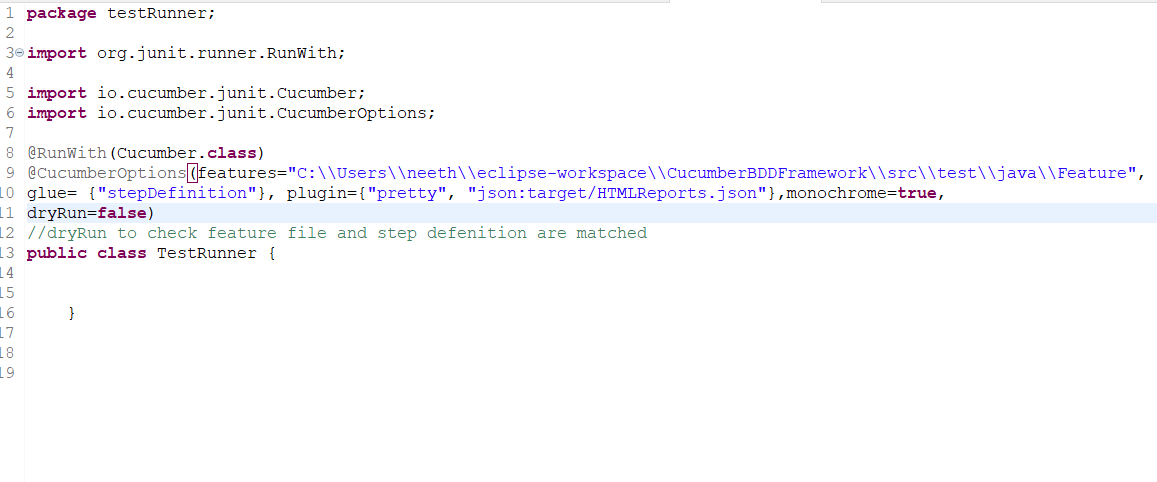
**Test Runner:**

In order to glue 2 step definition files, and also in order to map them, we need Test Runner class.

Also reports can be generated**.**

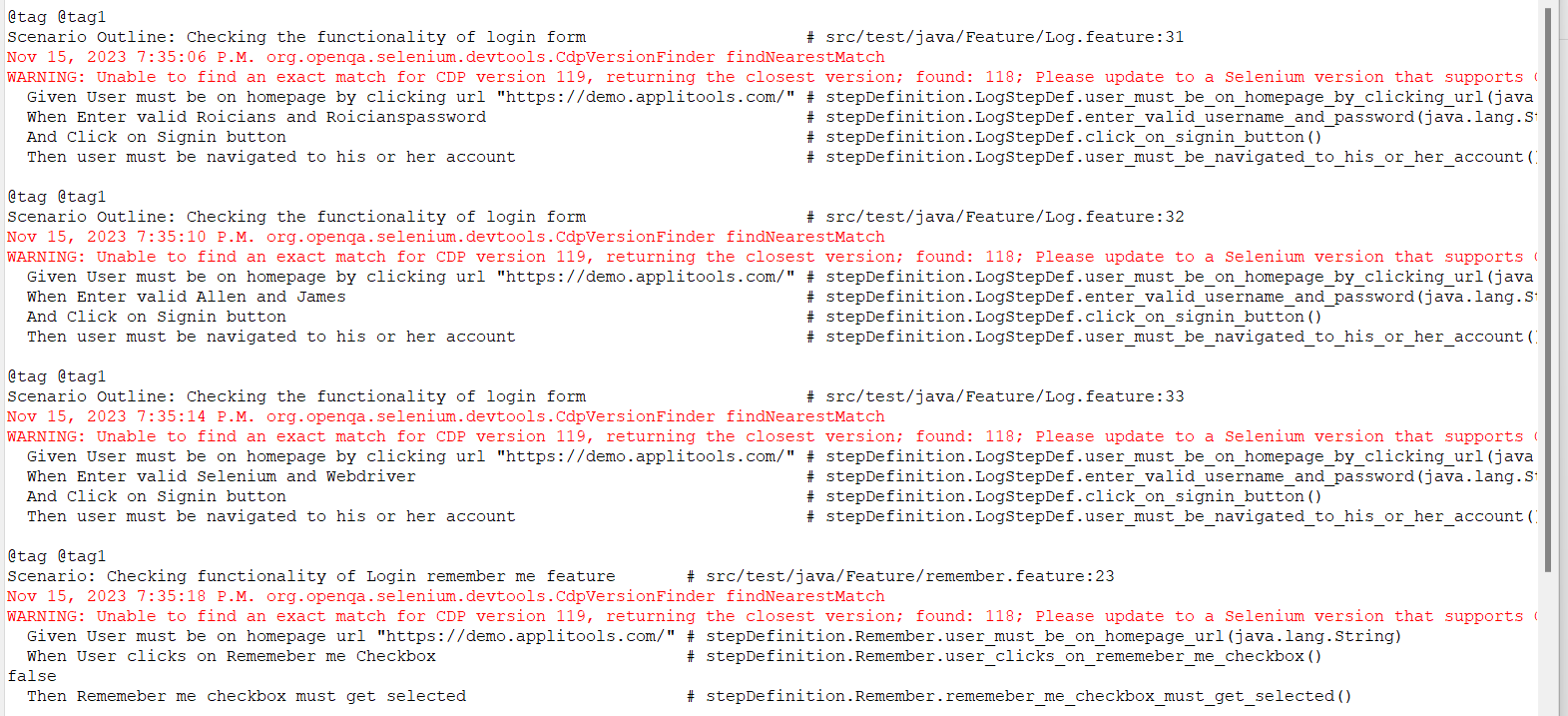
**1.Create a Package test Runner and create a class TestRunner.java**

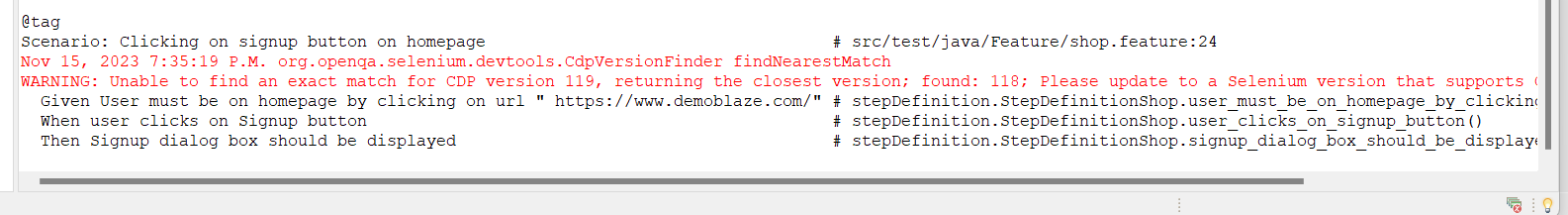
**2. Add Cucumber JVM: JUnit and Cucumber JVM: core to pom.xml**

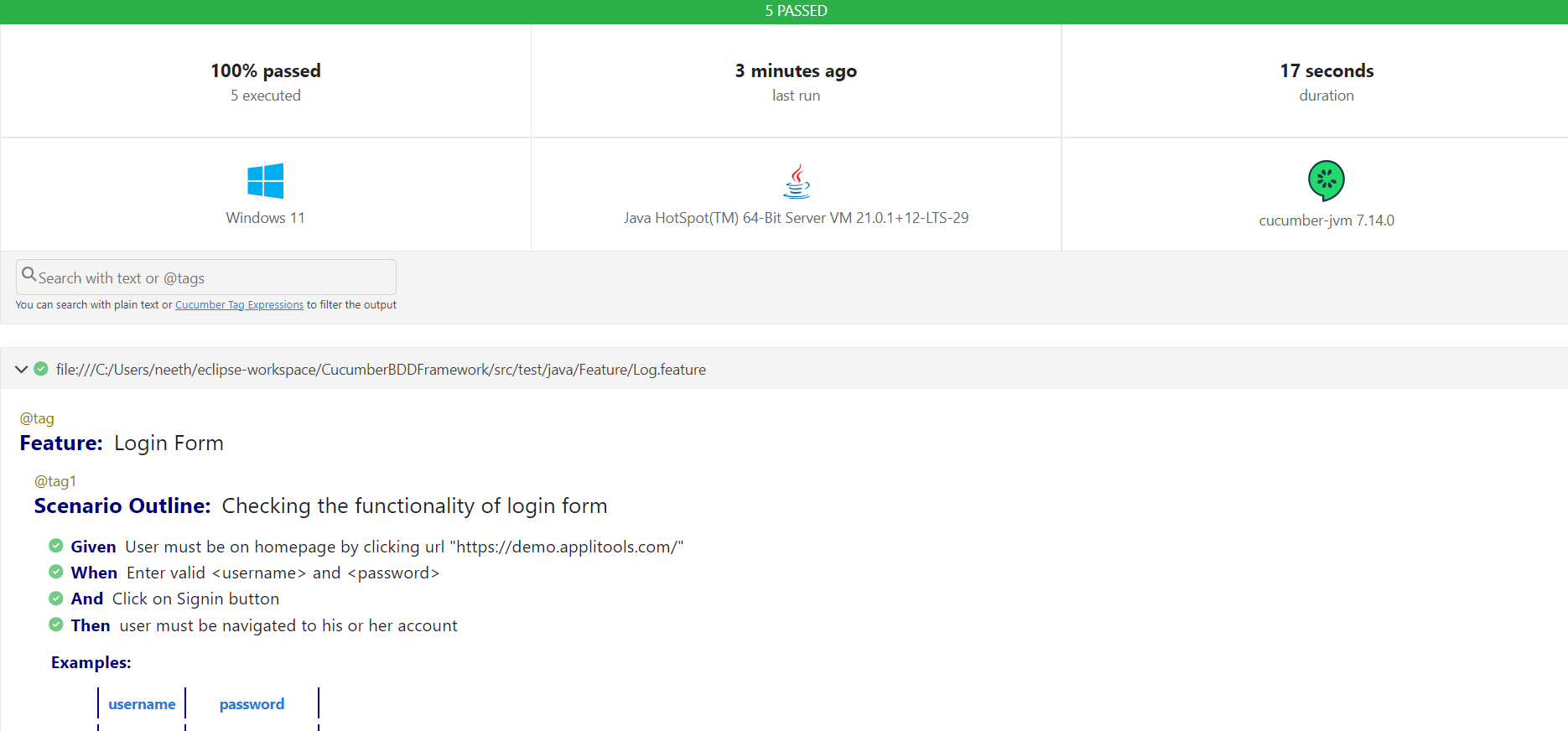
****

dryRun=True to check if the feature file and Step Definition are mapped

**Output:**

****

****

****

**Cucumber Regular expression cheat sheet:**