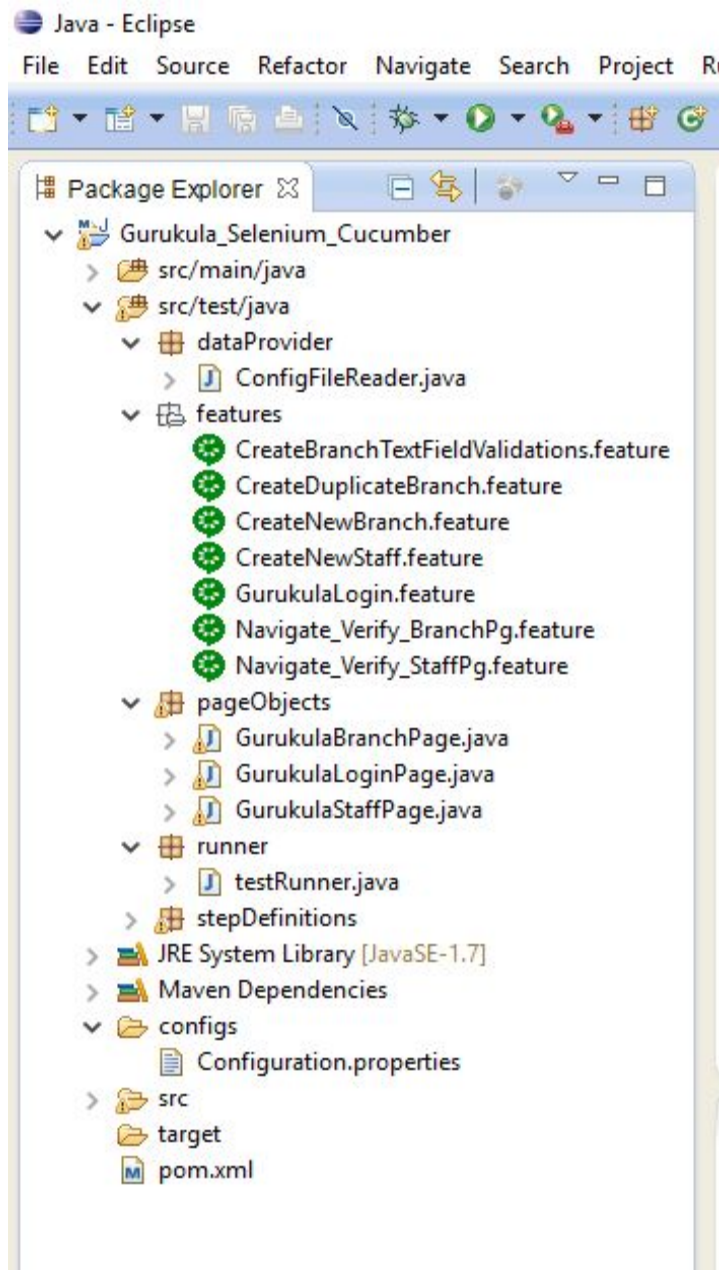


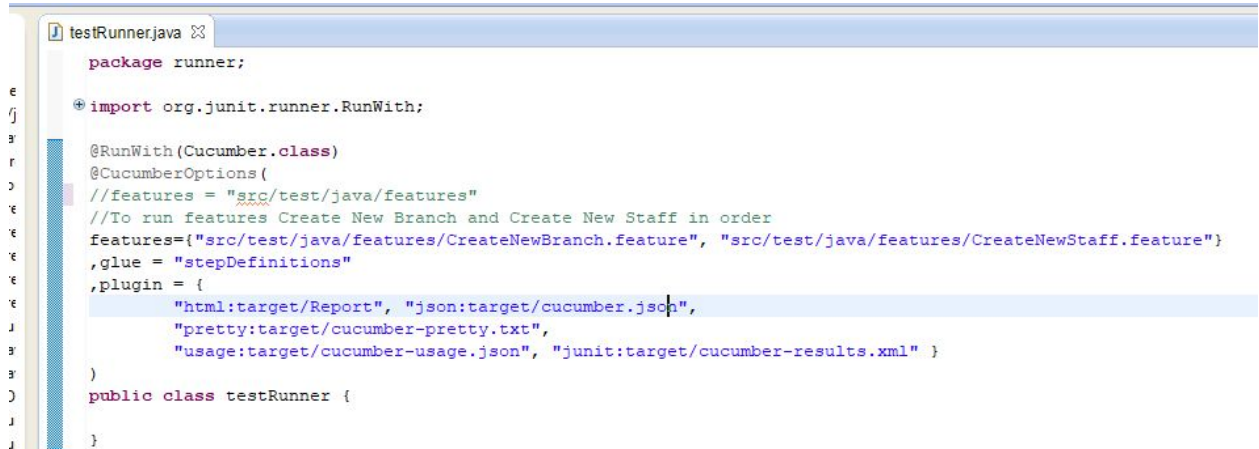
Selenium Cucumber Framework (BDD)

- JUnit, Maven, Gherkin, Page Object Model, Selenium and Java
- Framework Folder Structure



1. **src/test/java - dataProvider : ConfigFileReader** - Java class to read the config file

2. **src/test/java - features** : Entry point to cucumber tests. Test cases are described in english language
3. **src/test/java -pageObjects** : Object oriented class that is the interface to all the pages of the AUT (Application Under Test). To enhance code maintenance and reducing code duplication.
4. **src/test/java -runner** : **TestRunner class is used to run our test cases.**



```
testRunner.java
package runner;

import org.junit.runner.RunWith;

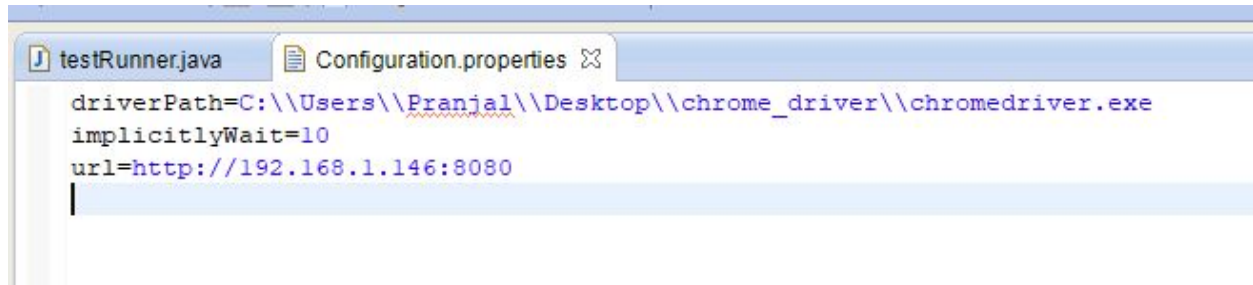
@RunWith(Cucumber.class)
@CucumberOptions(
    //features = "src/test/java/features"
    //To run features Create New Branch and Create New Staff in order
    features={"src/test/java/features/CreateNewBranch.feature", "src/test/java/features/CreateNewStaff.feature"},
    glue = "stepDefinitions"
    ,plugin = {
        "html:target/Report", "json:target/cucumber.json",
        "pretty:target/cucumber-pretty.txt",
        "usage:target/cucumber-usage.json", "junit:target/cucumber-results.xml" }
)
public class testRunner {
}
```

To run the features CreateNewBranch and CreateNewStaff, mention the sequence as in above image. To execute, open the TestRunner file, right click 'Run As' JUnit
Rest all other features can be executed independently. To execute each feature click on the feature file and right click 'Run As' Cucumber
Cucumber reporting format is provided

5. src/test/java - stepDefinitions : Java class that contains java methods that links to the steps (Given,When,Then) of the Gherkin language in the Scenario present in feature file

6. Maven Dependencies : Contains all the jar files of JUnit, Cucumber, Selenium etc. provided in the Maven file : pom.xml

7. Configs - Configuration.properties : Properties file to maintain project configuration data like application url, browser exe, data-base configurations etc.



```
testRunner.java Configuration.properties
driverPath=C:\\Users\\Pranjal\\Desktop\\chrome_driver\\chromedriver.exe
implicitlyWait=10
url=http://192.168.1.146:8080
```

To execute the test cases, provide the driverPath of the browser driver exe
And provide the url of the application

8. Target : Contains cucumber, JUnit reports

9. Pom.xml : Maven (pom.xml) configures dependencies for building testing and running code.

- Steps to open the Gurukula application :

Gurukula application is written in Java and available in **staff** folder

Launch the application using: java -jar .war

Use Java 1.8 to access the application. Username: admin Password: admin

Features of the application:

1. The application is used to maintain Branches and Staff information (can be seen from Entities Menu).
2. You can view, edit, delete and query both Staff and Branches.
3. Further, Pagination is enabled when viewing the Staff/Branch.
4. The logged in account information can be viewed/edited from the Account Menu.
5. Login/Logout as existing user
6. Register a new User