# CUCUMBER BDD (OR) FRAMEWORK EXPLANATION

In my project we are using Cucumber BDD framework. It is used to convert the exact requirements into plain English language using Gherkin Keywords.

Here the default execution starts from pom.xml where we used to maintain all our dependencies used in our project.

Then we have Test Runner class where we mention **@CucumberOption** Annotation which have different cucumber options like

- **features** for providing path of the feature files
- **glue** for providing path of the step definition class
- **dryRun** for generating updated snippets
- **monochrome** to print in a readable format
- **tags** for group the test cases
- **plugin** for generating different types of reports.

Then we have Hooks Class where we have two annotations

- **@Before** used to execute before each scenario
- **@After** used to execute after each scenario

Then we have Base Class where we used to maintain all the reusable methods required in my project.

Then we will create feature files in src/test/resources using Gherkin Keywords

- **Feature** where we mention what we are going to test in the particular requirement.
- **Scenario** where we mention we are going to test the feature.
- **Given** where we provide some precondition step.
- **When** where we provide some key actions.
- **And** where we add conditions in feature file
- **Then** used to verify or validate something
- **Scenario Outline** used to execute the same scenario with multiple times with different set of test datas.
- **Examples** used to pass the parameters/datas using the | (pipe symbol) | separate each data.
- **Background** where we mention steps which is are common to all the scenarios in the feature file.

Then we generate the snippets and write the business logics in Step Definition Class.

Then we have POJO classes in page wise where we used to maintain all the locators in page wise and call the locators by creating object in step definition class.

After scripting is done, we will execute in test runner and then generate reports like xml, html, json and JVM report. In my project we used to submit JVM report as test result.