The project follows the Page Object Model design pattern (approach). Which is basically Test-Driven Development framework.

There are main 2 folders, which are:

**1. src/main/java**

**2. src/test/java**

**src/main/java** contains several packages which have their own importance in the framework.

* **com.resolver.qa.base** package contains TestBase.java class which is the parent class of all other class. It is basically used to define all the prerequisites for the project. This can be initializing the wed driver, initializing the properties file, maximizing the window, implicitlyWait(), pageLoadTimeout(), get(“URL”) methods.
* **com.resolver.qa.config** package has config.properties file, which we use to define the Environment variables like URL, username, password, browser, etc.
* **com.resolver.qa.extentreportlistener** has extentreportlistener file which will create a fancy report with the help of third party tool.
* **com.resolver.qa.pages** has all the page file for each test scenarios. These files contain the Object repositories of Web Objects/ Web Elements and the Actions/ Methods which need to be performed. Each page has their own test.java class which performs related tests.
* **com.resolver.qa.util** is used for common utilities/ generic functions like page load timeout, implicit wait time, taking screenshots, sending an email, etc.

**src/test/java** has testcase package.

* **com.resolver.qa.testcases** have the files mapped with the number of Page.java defined in page package. All the testcases can be defined with TestNG annotations. All Test.java files inherit the TestBase file to use methods defined in that class file.

**pom.xml** is the most widely used to download the relevant dependencies required for our project and to run our project virtually without opening our IDE. This can be done through command prompt or CI/CD tool like Jenkins. Which helps manual testers to run the automated scripts.

**testng.xml** file is being used for creating the fancy reports and to run the test cases parallelly. TestNG is preferred over Junit because of the availability of more options.

**target-output** contains the various files including report. These reports are used to review the test and to share the results with peers.