What is a Selenium framework?
=============================

- Selenium framework is a code structure for making
  - Code maintenance is simpler.
  - Code readability is better.
- A framework involves breaking the entire code into smaller pieces of code, which test a particular functionality.
- The code is structured such that the “test data” is separated from the actual “test case” which will test the functionality of the web application.
- It can also be structured in a way where the test cases which need to be executed are called from an external source.

There are a number of frameworks out there, but 3 commonly used Selenium framework (s) are:

- Data Driven framework
- Keyword Driven framework
- Hybrid framework

Why do we need a Selenium framework?
==================================

Benefits of Selenium framework

- Increased code reuse
- Improved code readability
- Higher portability
- Reduced script maintenance
- Less maintenance
- Faster execution
- Better error handling

1. **Data Driven Framework**
   ===================

- A Data Driven framework in Selenium is the technique of separating the “test data” from the actual “test case” (code).
- This framework completely depends on the input test data.
- The test data is fed from external sources such as an excel file.
- Since the test case is separated from the test data, we can easily modify the test case of a particular functionality without making wholesale changes to your code.
- Besides this, we can also easily control how much data needs to be tested.
- You can easily increase the number of test parameters by adding more username and password fields to the excel file.

2. **Keyword Driven framework**
   ======================

- Keyword Driven framework is a technique in which all the operations & instructions to be performed are written separately from the actual test case like reusable methods.
- The similarity it has with the Data Driven framework is that the operations to be performed are again stored in an external file like Excel sheet.

3. **Hybrid framework**
   ===============

- Hybrid framework is a technique where we can make the best use of both Data Driven & Keyword Driven Selenium framework.
- We can build a Hybrid framework by storing the methods to execute in various java classes(keyword driven approach) and passing the data to this method(data driven approach)

============================

Framework design
===============

1. **Base Class**

- It is a supermost class
- In this class we define the properties related to the webdriver and browser.
- And also multi browser code.

2. **Test Class**

- In this class we keep all test case related code.

3. **Utility Class**

- In this class we write the repeated code in order to use it repeatedly in the execution.
- E.x taking a screenshot, fetching data from excel, wait actions.

4. **Extend Reporter**

- It is used for reporting purposes.
- We can capture screenshots of all failed test cases and stored it into extend report
- It displays time taken for test case execution
- It can be easily integrated with a framework.

5. **Use various folders**

- To keep browser driver
- To keep screenshots
- To keep reports

6. **POM Class**

- POM class also known as Page Object Model.
- It is a design pattern in Selenium that creates an object repository for storing all web elements.
- It is useful in reducing code duplication and improves test case maintenance.
- In Page Object Model, consider each web page of an application as a class file.
- Each class file will contain only corresponding web page elements.
- Using these elements, testers can perform operations on the website under test.
- POM strictly follows the encapsulation concepts as

    - All the data members are declared globally with access modifier private.
    - These data members are initialised by using a constructor with the help of pagefactory.
    - These data members are utilised within a method which is declared with access modifiers public.

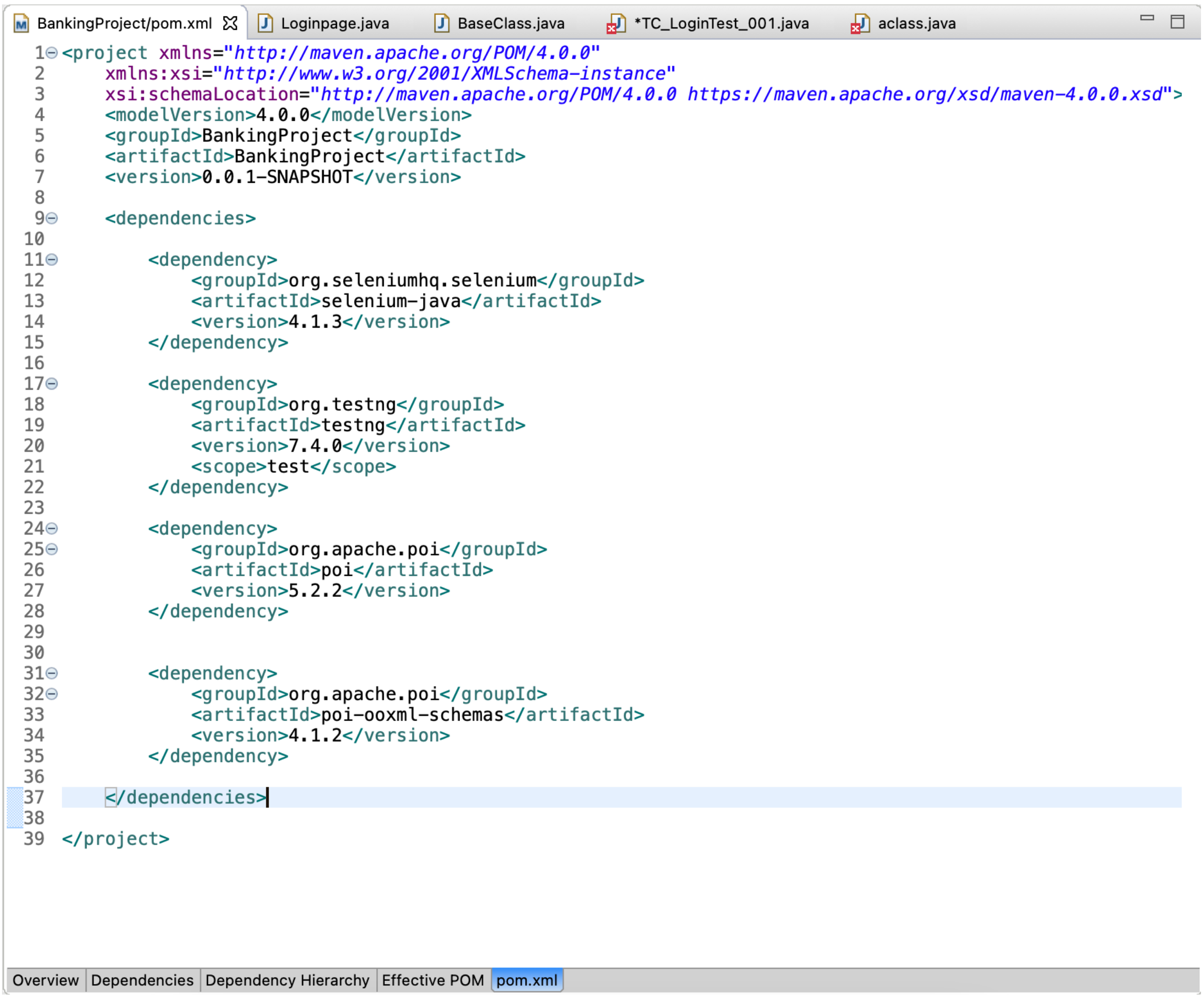
==================

Implementation

1. **Create a maven project**

    - Go to File > New > Other > Maven
    - Inside maven folder > select maven project
    - Click on create simple project
    - Enter Group id and Artifact id
    - Click on finish

2. **Add dependencies in pom.xml file**
    - Go to https://mvnrepository.com/
    - Search for selenium, testng, apache poi and add the dependencies into the pom.xml file.



```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>BankingProject</groupId>
    <artifactId>BankingProject</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <dependencies>

        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>4.1.3</version>
        </dependency>

        <dependency>
            <groupId>org.testng</groupId>
            <artifactId>testng</artifactId>
            <version>7.4.0</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.apache.poi</groupId>
            <artifactId>poi</artifactId>
            <version>5.2.2</version>
        </dependency>


        <dependency>
            <groupId>org.apache.poi</groupId>
            <artifactId>poi-ooxml-schemas</artifactId>
            <version>4.1.2</version>
        </dependency>

    </dependencies>

</project>
```