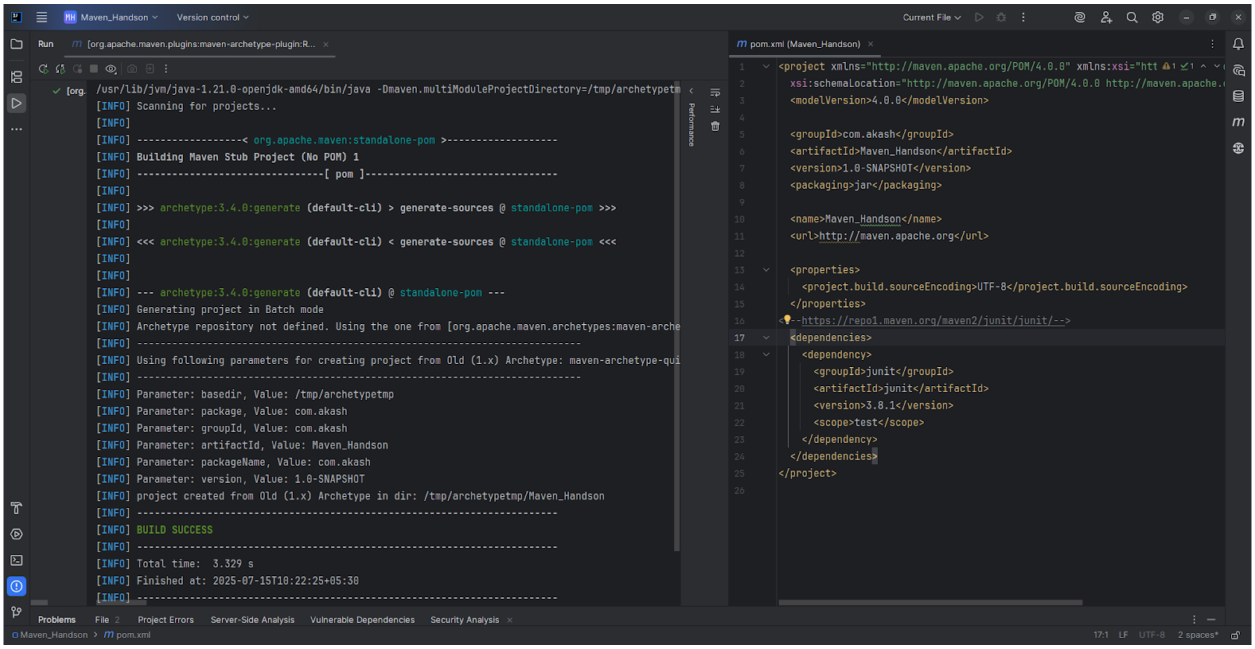
Assignment - Dependancy Management using Maven

Q1)Following parameters are to be considered for exercise submission:

- The complete project has to be submitted.
- .idea, target directories should be excluded
- A text file containing the relevant URLs which are referenced and any other documentation.

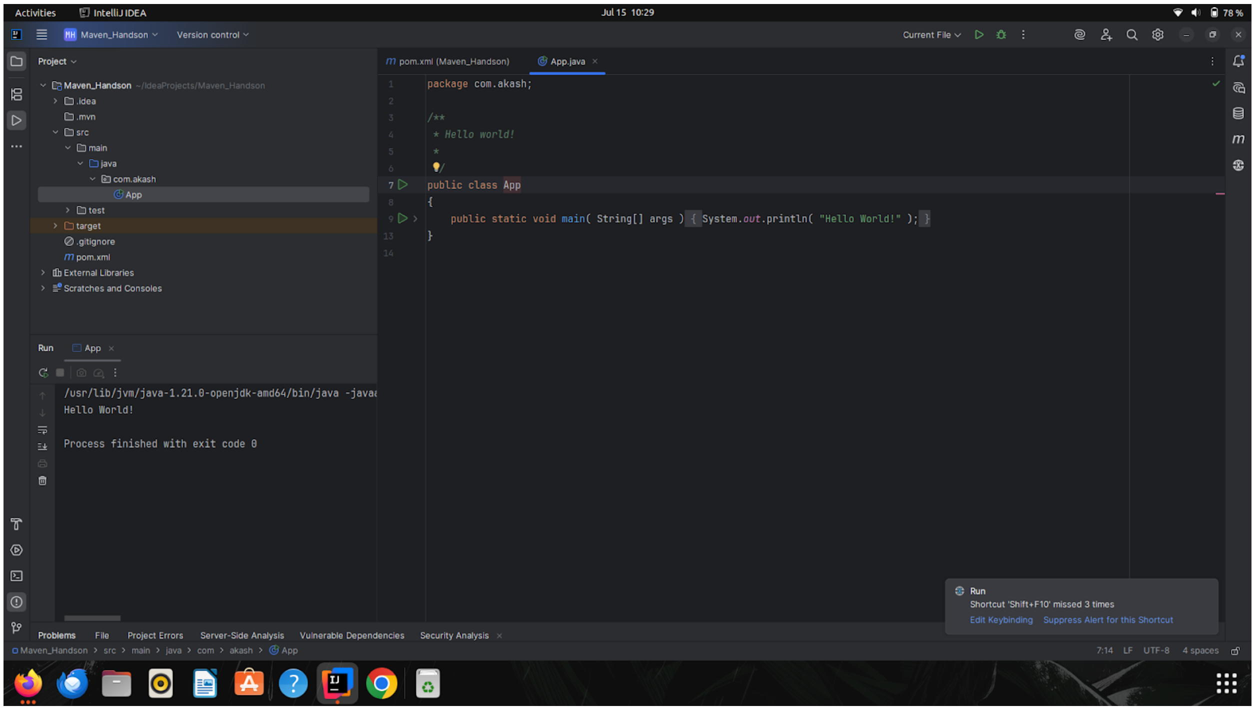
Q2)Add a maven dependency and its related repository URL.

Repository URL - https://repo1.maven.org/maven2/junit/junit/



```
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -Dmaven.multiModuleProjectDirectory=/tmp/archetypetm
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< org.apache.maven:standalone-pom >-------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] --------------------------------[ pom ]---------------------------------
[INFO]
[INFO] >>> archetype:3.4.0:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< archetype:3.4.0:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO]
[INFO] --- archetype:3.4.0:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-arche
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-qui
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: basedir, Value: /tmp/archetypetmp
[INFO] Parameter: package, Value: com.akash
[INFO] Parameter: groupId, Value: com.akash
[INFO] Parameter: artifactId, Value: Maven_Handson
[INFO] Parameter: packageName, Value: com.akash
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /tmp/archetypetmp/Maven_Handson
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.329 s
[INFO] Finished at: 2025-07-15T10:22:25+05:30
[INFO] ------------------------------------------------------------------------
```

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="htt
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.akash</groupId>
  <artifactId>Maven_Handson</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>Maven_Handson</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
<!--https://repo1.maven.org/maven2/junit/junit/-->
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

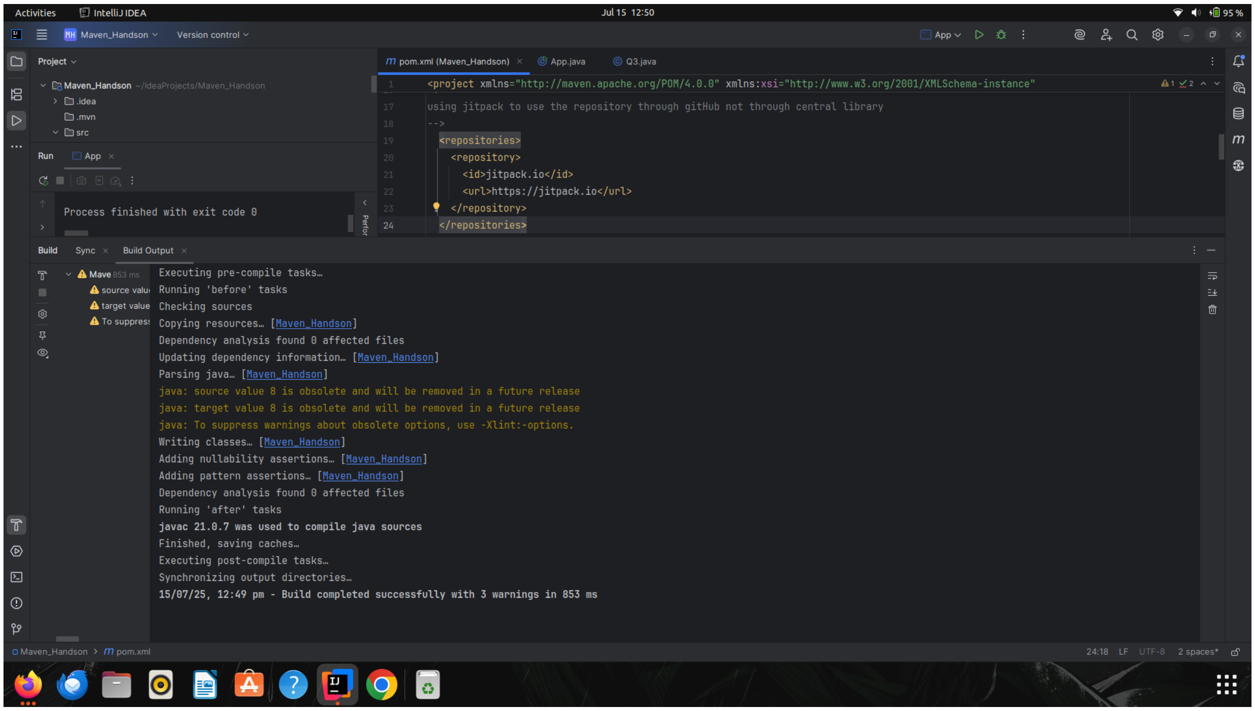


```java
package com.akash;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args ) { System.out.println( "Hello World!" ); }
}

```

```
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -javaa
Hello World!

Process finished with exit code 0
```

Run
Shortcut 'Shift+F10' missed 3 times
Edit Keybinding   Suppress Alert for this Shortcut

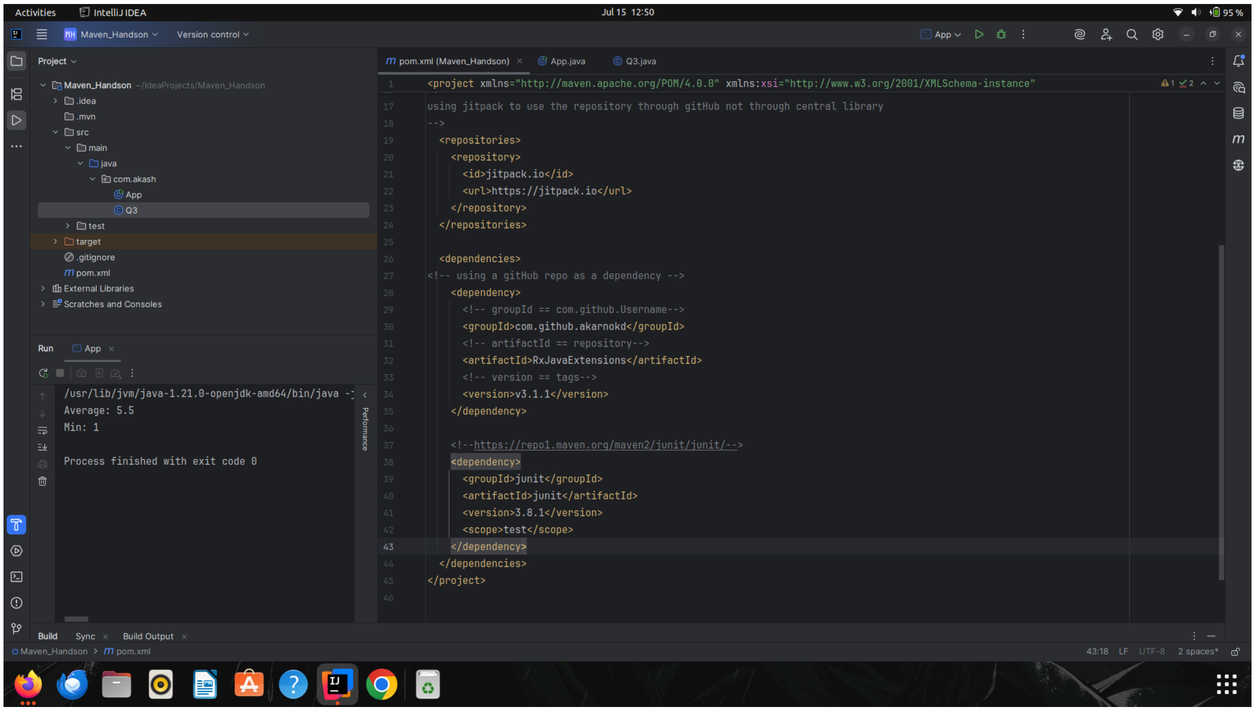
Q3)Add a new repository in the pom.xml and use its dependencies.



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
using jitpack to use the repository through gitHub not through central library
-->
  <repositories>
    <repository>
      <id>jitpack.io</id>
      <url>https://jitpack.io</url>
    </repository>
  </repositories>
```

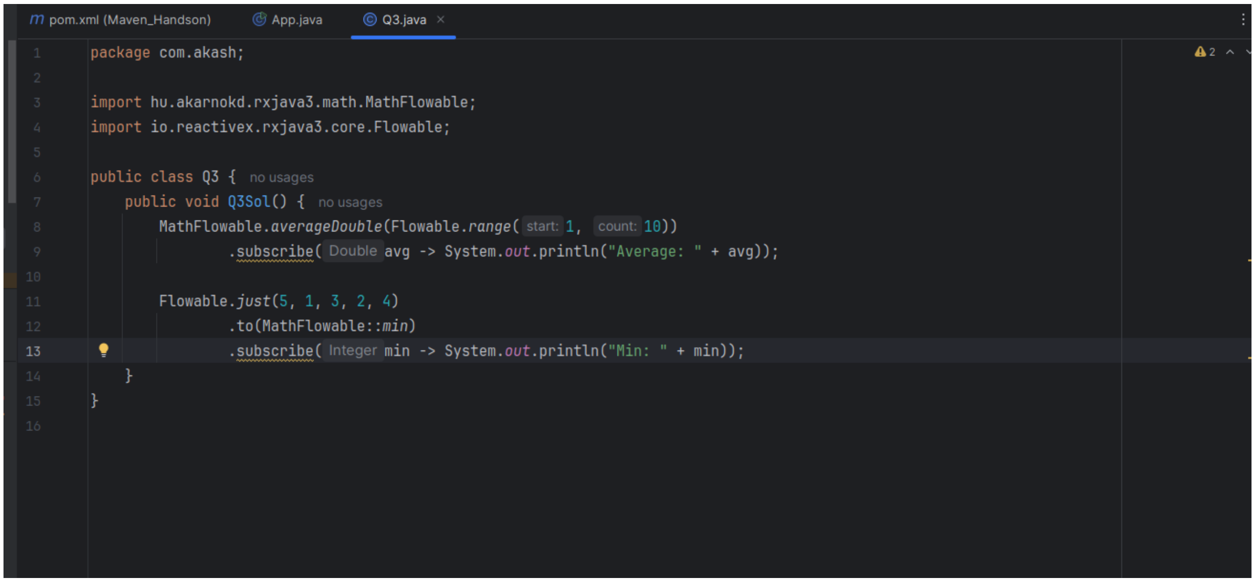
```
Process finished with exit code 0
```

```
Executing pre-compile tasks…
Running 'before' tasks
Checking sources
Copying resources… [Maven_Handson]
Dependency analysis found 0 affected files
Updating dependency information… [Maven_Handson]
Parsing java… [Maven_Handson]
java: source value 8 is obsolete and will be removed in a future release
java: target value 8 is obsolete and will be removed in a future release
java: To suppress warnings about obsolete options, use -Xlint:-options.
Writing classes… [Maven_Handson]
Adding nullability assertions… [Maven_Handson]
Adding pattern assertions… [Maven_Handson]
Dependency analysis found 0 affected files
Running 'after' tasks
javac 21.0.7 was used to compile java sources
Finished, saving caches…
Executing post-compile tasks…
Synchronizing output directories…
15/07/25, 12:49 pm - Build completed successfully with 3 warnings in 853 ms
```



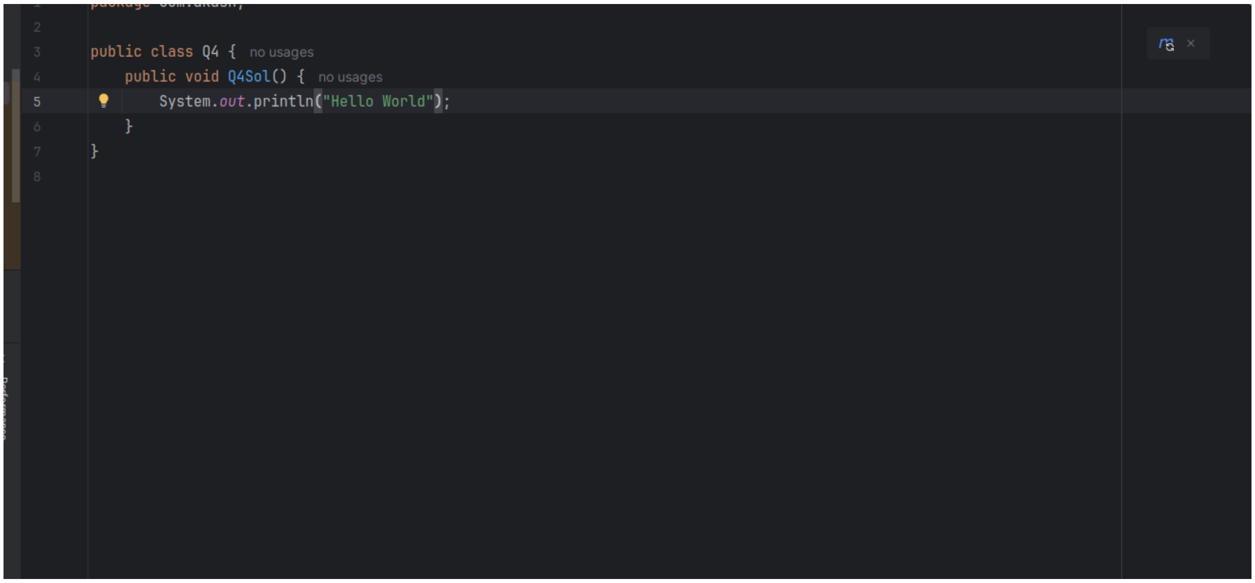
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
using jitpack to use the repository through gitHub not through central library
-->
  <repositories>
    <repository>
      <id>jitpack.io</id>
      <url>https://jitpack.io</url>
    </repository>
  </repositories>

  <dependencies>
<!-- using a gitHub repo as a dependency -->
    <dependency>
      <!-- groupId == com.github.Username-->
      <groupId>com.github.akarnokd</groupId>
      <!-- artifactId == repository-->
      <artifactId>RxJavaExtensions</artifactId>
      <!-- version == tags-->
      <version>v3.1.1</version>
    </dependency>

    <!--https://repo1.maven.org/maven2/junit/junit/-->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

```
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -j
Average: 5.5
Min: 1

Process finished with exit code 0
```

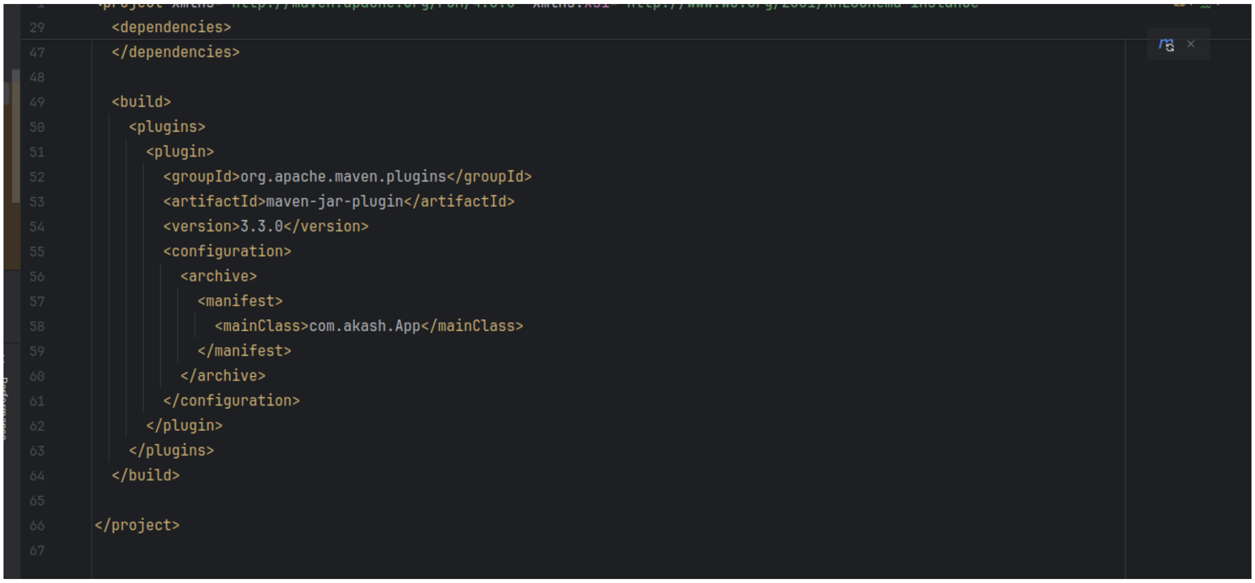


```java
package com.akash;

import hu.akarnokd.rxjava3.math.MathFlowable;
import io.reactivex.rxjava3.core.Flowable;

public class Q3 {
    public void Q3Sol() {
        MathFlowable.averageDouble(Flowable.range(1, 10))
                .subscribe(avg -> System.out.println("Average: " + avg));

        Flowable.just(5, 1, 3, 2, 4)
                .to(MathFlowable::min)
                .subscribe(min -> System.out.println("Min: " + min));
    }
}
```

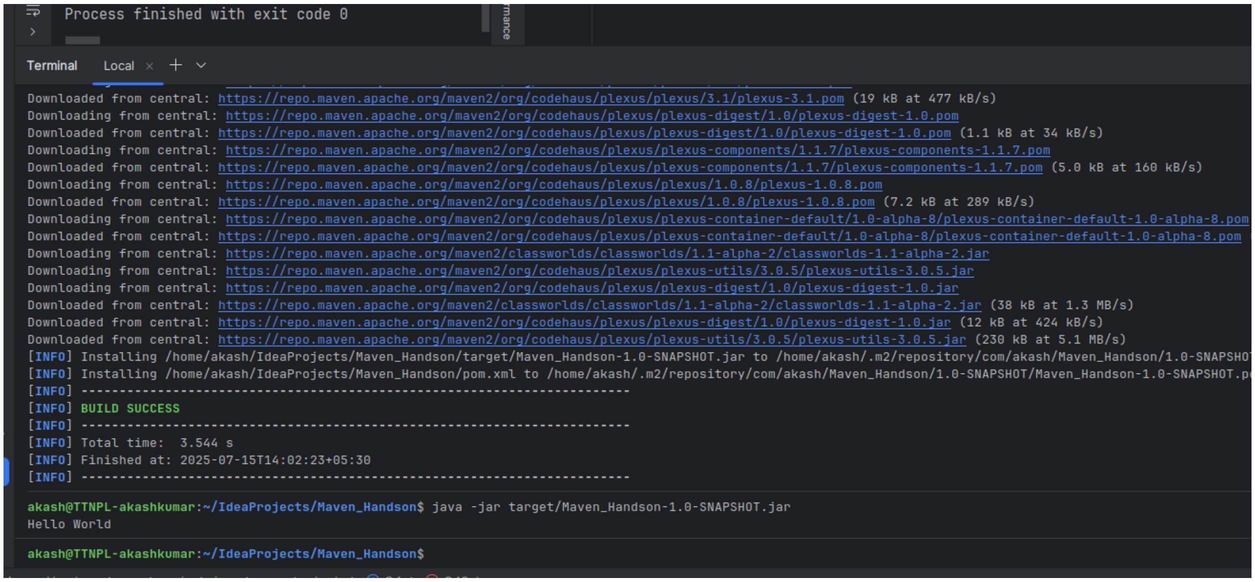
Q4)Using JAR plugin, make changes in the pom.xml to make the jar executable. Using java -jar JAR_NAME, the output should be printed as "Hello World"



```java
package com.akash;

public class Q4 {  no usages
    public void Q4Sol() {  no usages
        System.out.println("Hello World");
    }
}

```



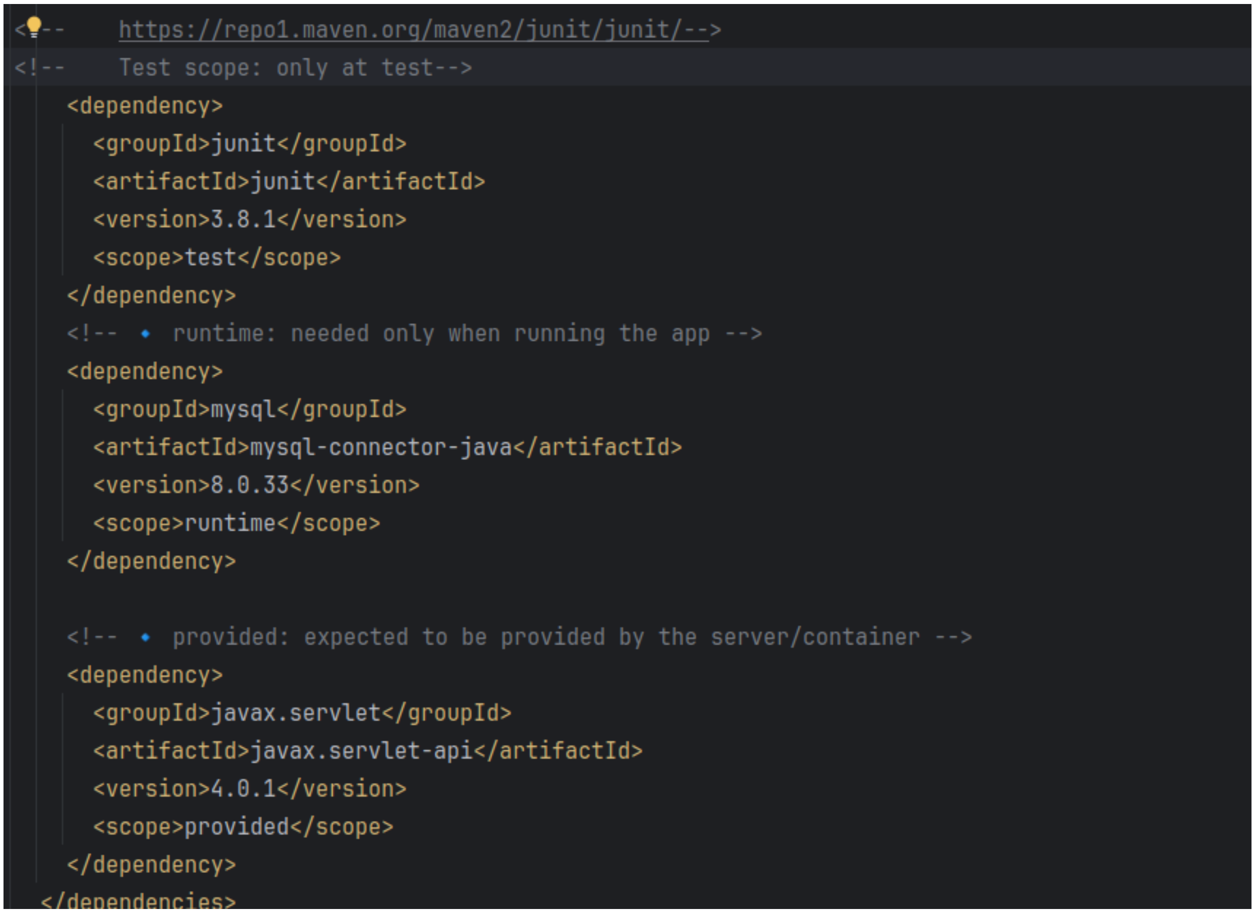
```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <dependencies>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>
        <version>3.3.0</version>
        <configuration>
          <archive>
            <manifest>
              <mainClass>com.akash.App</mainClass>
            </manifest>
          </archive>
        </configuration>
      </plugin>
    </plugins>
  </build>

</project>

```



```
Process finished with exit code 0
```

```
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/3.1/plexus-3.1.pom (19 kB at 477 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom (1.1 kB at 34 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-components-1.1.7.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-components-1.1.7.pom (5.0 kB at 160 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/1.0.8/plexus-1.0.8.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/1.0.8/plexus-1.0.8.pom (7.2 kB at 289 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-8/plexus-container-default-1.0-alpha-8.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-8/plexus-container-default-1.0-alpha-8.pom
Downloading from central: https://repo.maven.apache.org/maven2/classworlds/classworlds/1.1-alpha-2/classworlds-1.1-alpha-2.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/classworlds/classworlds/1.1-alpha-2/classworlds-1.1-alpha-2.jar (38 kB at 1.3 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.jar (12 kB at 424 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar (230 kB at 5.1 MB/s)
[INFO] Installing /home/akash/IdeaProjects/Maven_Handson/target/Maven_Handson-1.0-SNAPSHOT.jar to /home/akash/.m2/repository/com/akash/Maven_Handson/1.0-SNAPSHOT
[INFO] Installing /home/akash/IdeaProjects/Maven_Handson/pom.xml to /home/akash/.m2/repository/com/akash/Maven_Handson/1.0-SNAPSHOT/Maven_Handson-1.0-SNAPSHOT.p
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.544 s
[INFO] Finished at: 2025-07-15T14:02:23+05:30
[INFO] ------------------------------------------------------------------------

akash@TTNPL-akashkumar:~/IdeaProjects/Maven_Handson$ java -jar target/Maven_Handson-1.0-SNAPSHOT.jar
Hello World

akash@TTNPL-akashkumar:~/IdeaProjects/Maven_Handson$
```

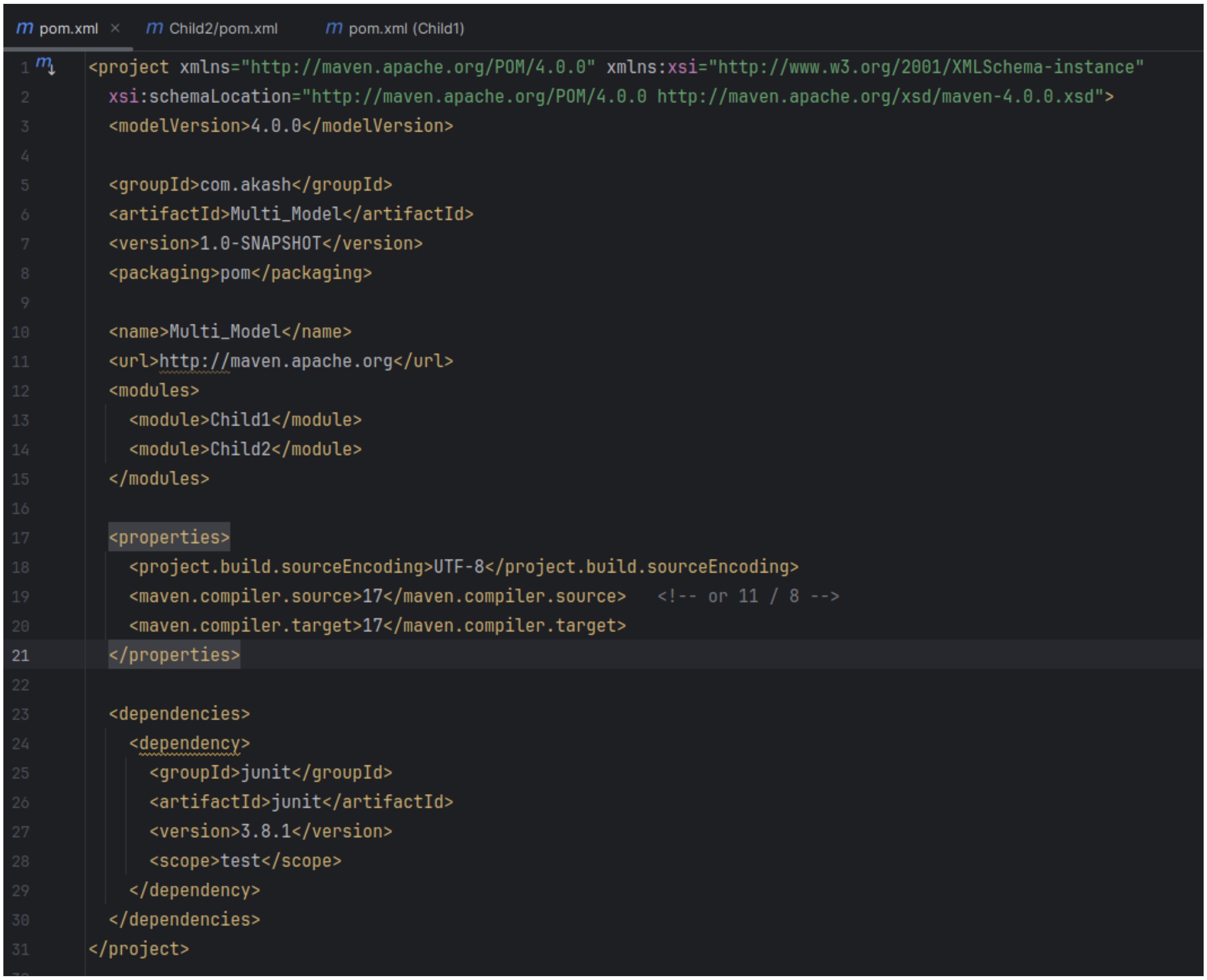
Q5)Differentiate between the different dependency scopes: compile, runtime, test, provided using different dependencies being defined in your pom.xml.



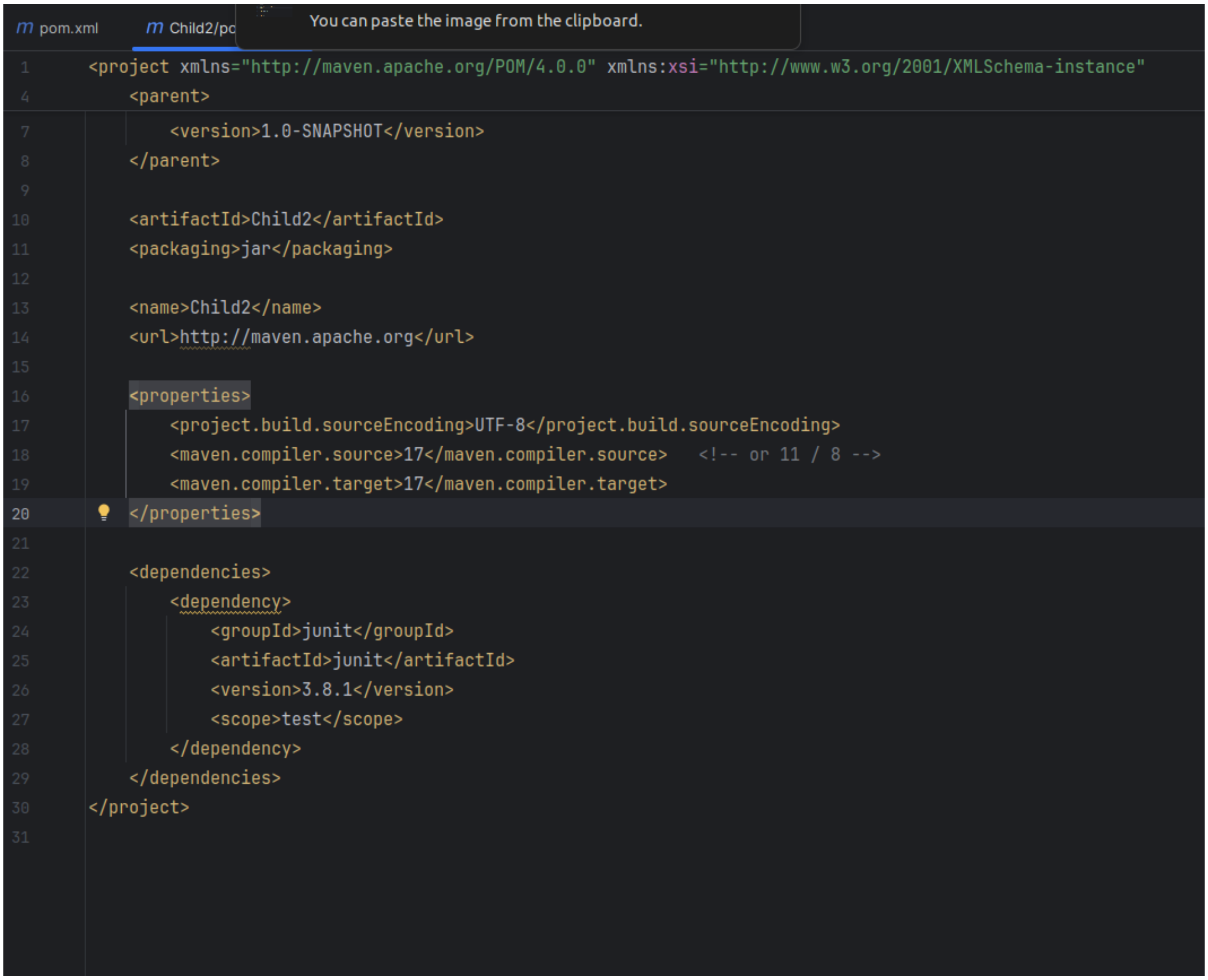
```
<!--    https://repo1.maven.org/maven2/junit/junit/-->
<!--    Test scope: only at test-->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <!--  • runtime: needed only when running the app -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.33</version>
      <scope>runtime</scope>
    </dependency>

    <!--  • provided: expected to be provided by the server/container -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
```

Scope	At Compile time	At Runtime	Finally at jar	Used for
compile	Yes	Yes	Yes	default
test	No	No	No	Only for test
runtime	No	Yes	Yes	Needed only at runtime
provided	Yes	No	No	Provided by container

Q6)Create a multi-module project. Run package command at the top level to make jar of every module.

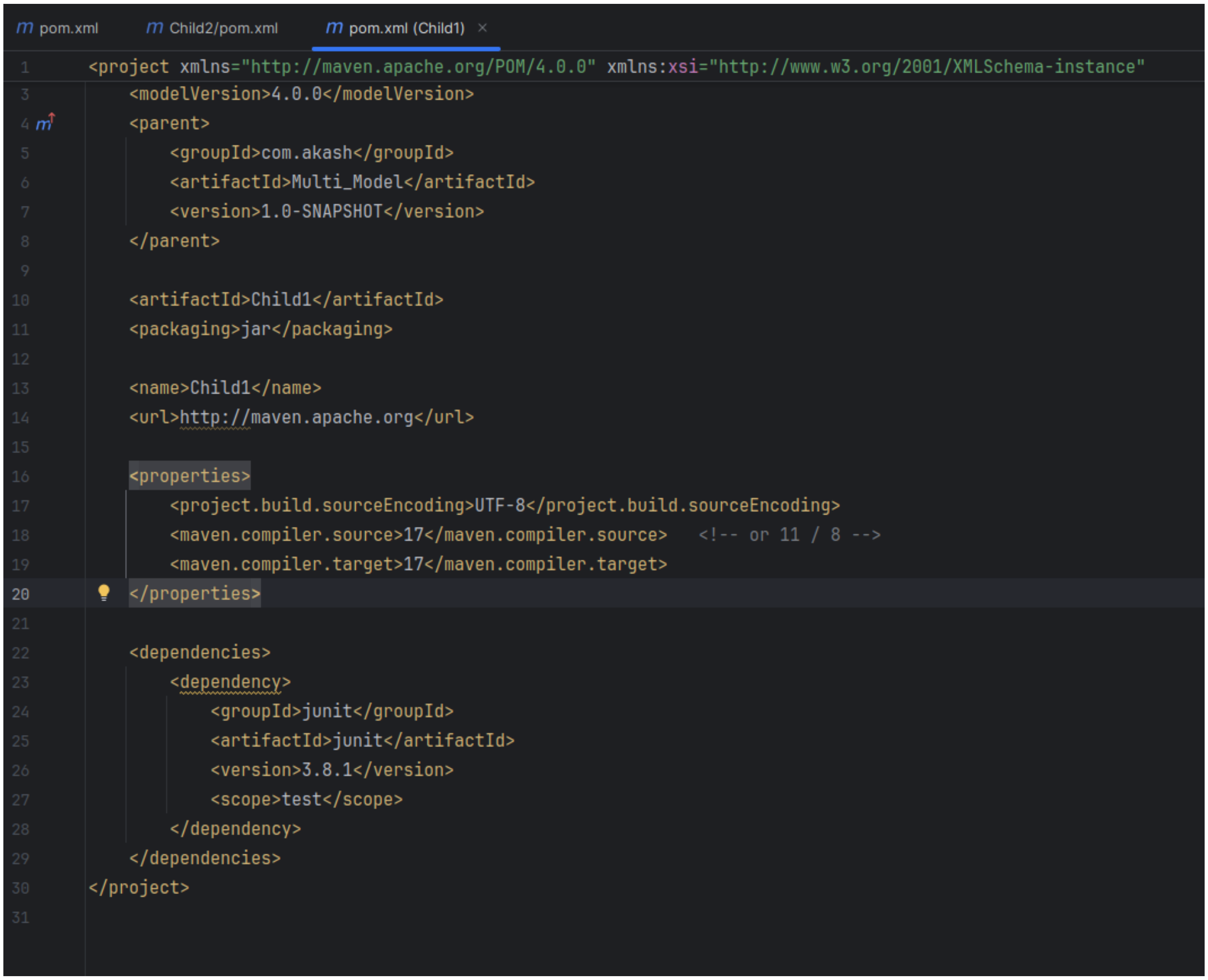


```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.akash</groupId>
  <artifactId>Multi_Model</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>pom</packaging>

  <name>Multi_Model</name>
  <url>http://maven.apache.org</url>
  <modules>
    <module>Child1</module>
    <module>Child2</module>
  </modules>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>17</maven.compiler.source>   <!-- or 11 / 8 -->
    <maven.compiler.target>17</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```



You can paste the image from the clipboard.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <parent>
        <version>1.0-SNAPSHOT</version>
    </parent>

    <artifactId>Child2</artifactId>
    <packaging>jar</packaging>

    <name>Child2</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>17</maven.compiler.source>   <!-- or 11 / 8 -->
        <maven.compiler.target>17</maven.compiler.target>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```



```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>com.akash</groupId>
        <artifactId>Multi_Model</artifactId>
        <version>1.0-SNAPSHOT</version>
    </parent>

    <artifactId>Child1</artifactId>
    <packaging>jar</packaging>

    <name>Child1</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>17</maven.compiler.source>   <!-- or 11 / 8 -->
        <maven.compiler.target>17</maven.compiler.target>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```