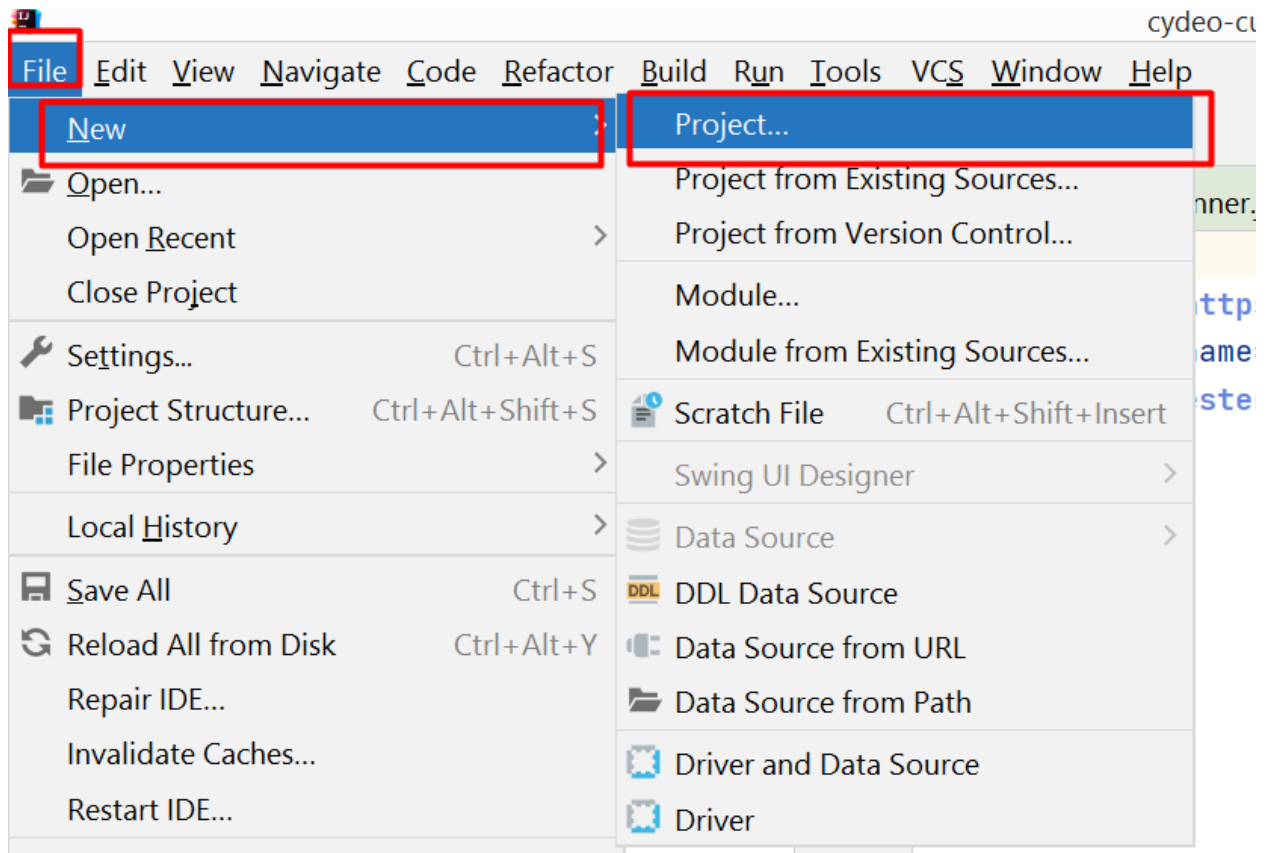
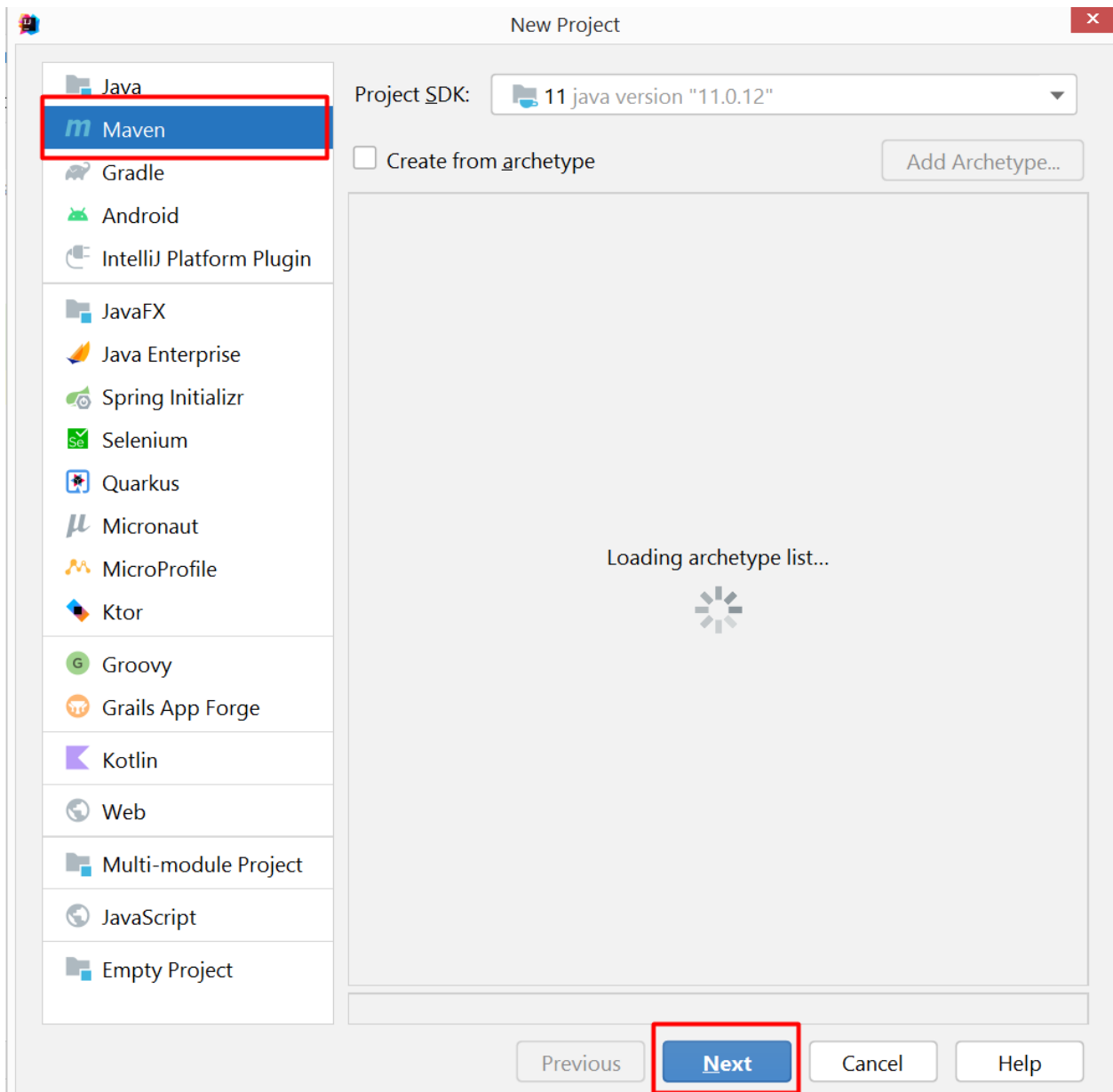


Create a framework.

1. Creating a maven project

“I am using maven in my framework”





New Project

Name:

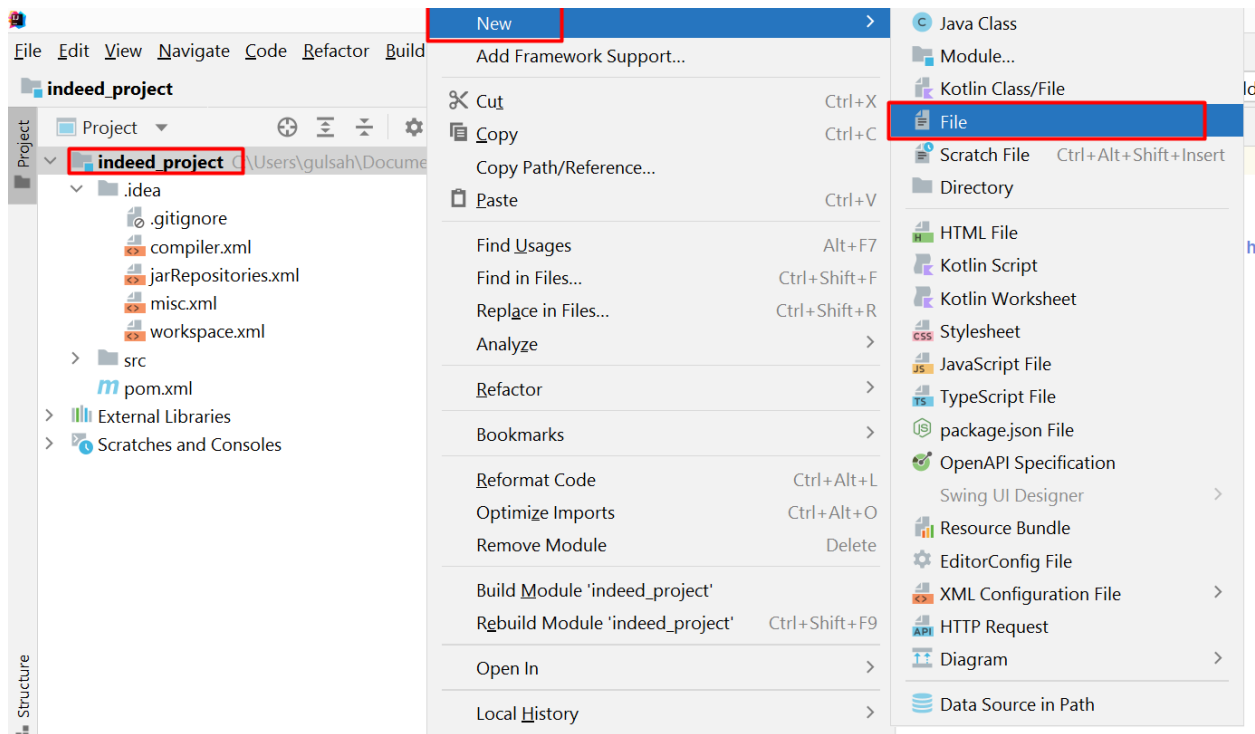
Location:

► Artifact Coordinates

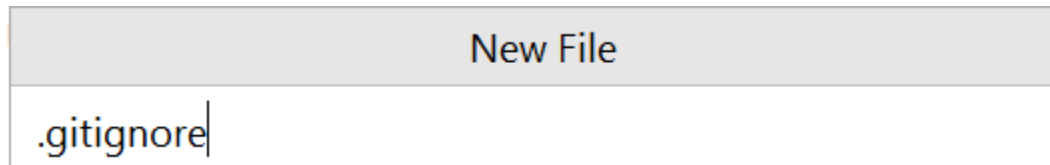
2. Creating Version Control- GIT

"I am using git as a version control tool"

A. .gitignore file



- Type .gitignore

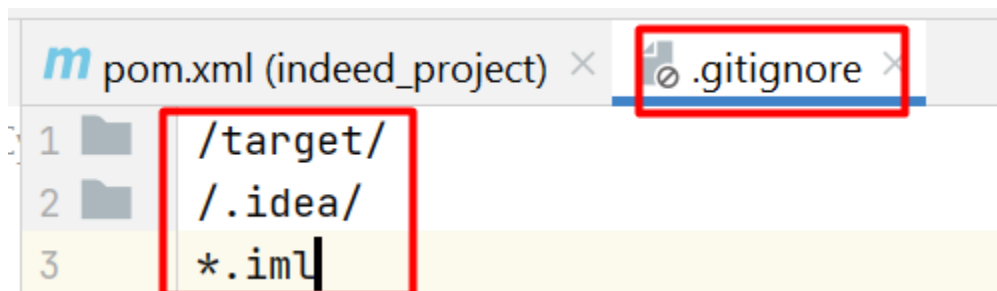


Copy and paste the following to the file

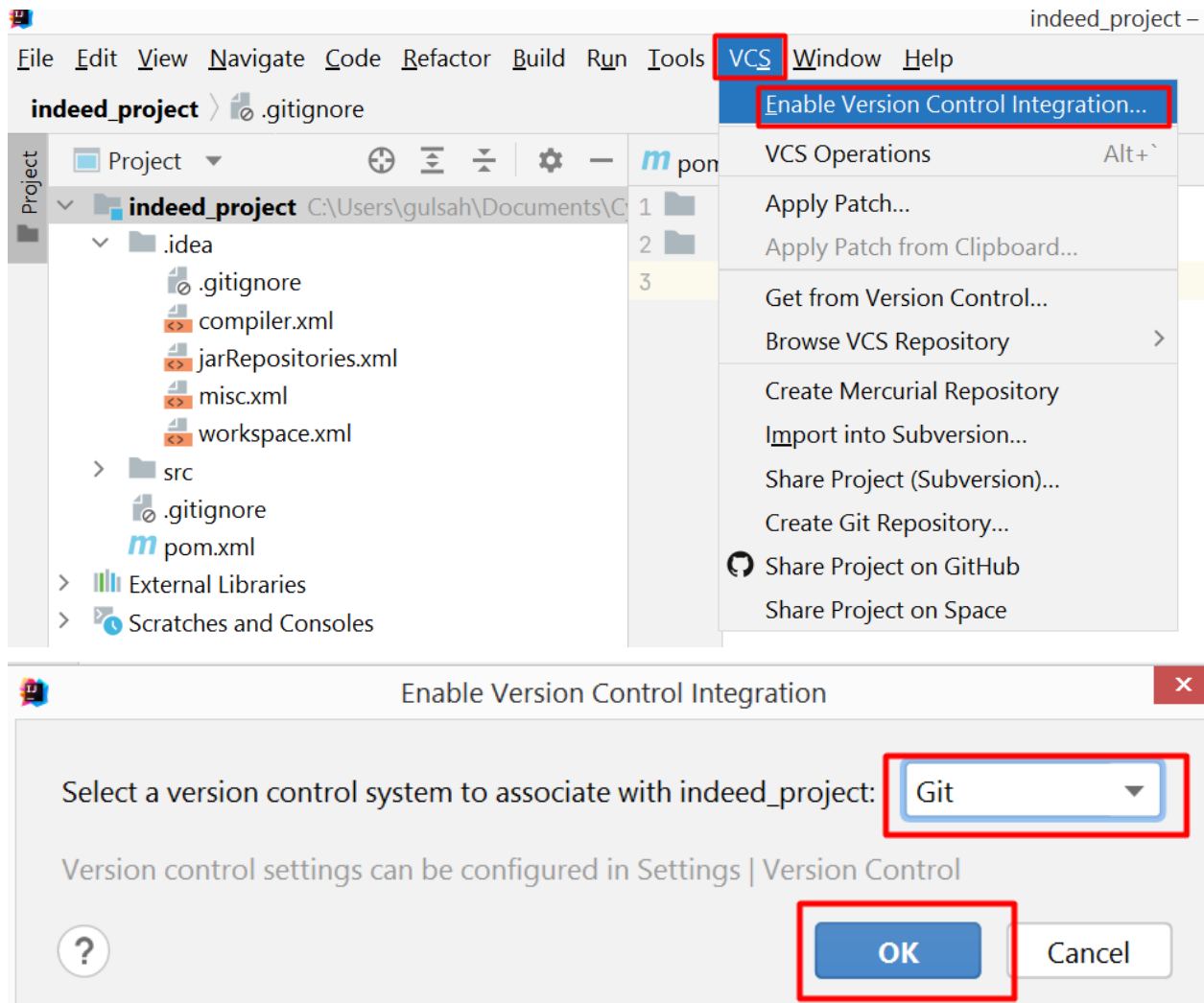
/target/

/.idea/

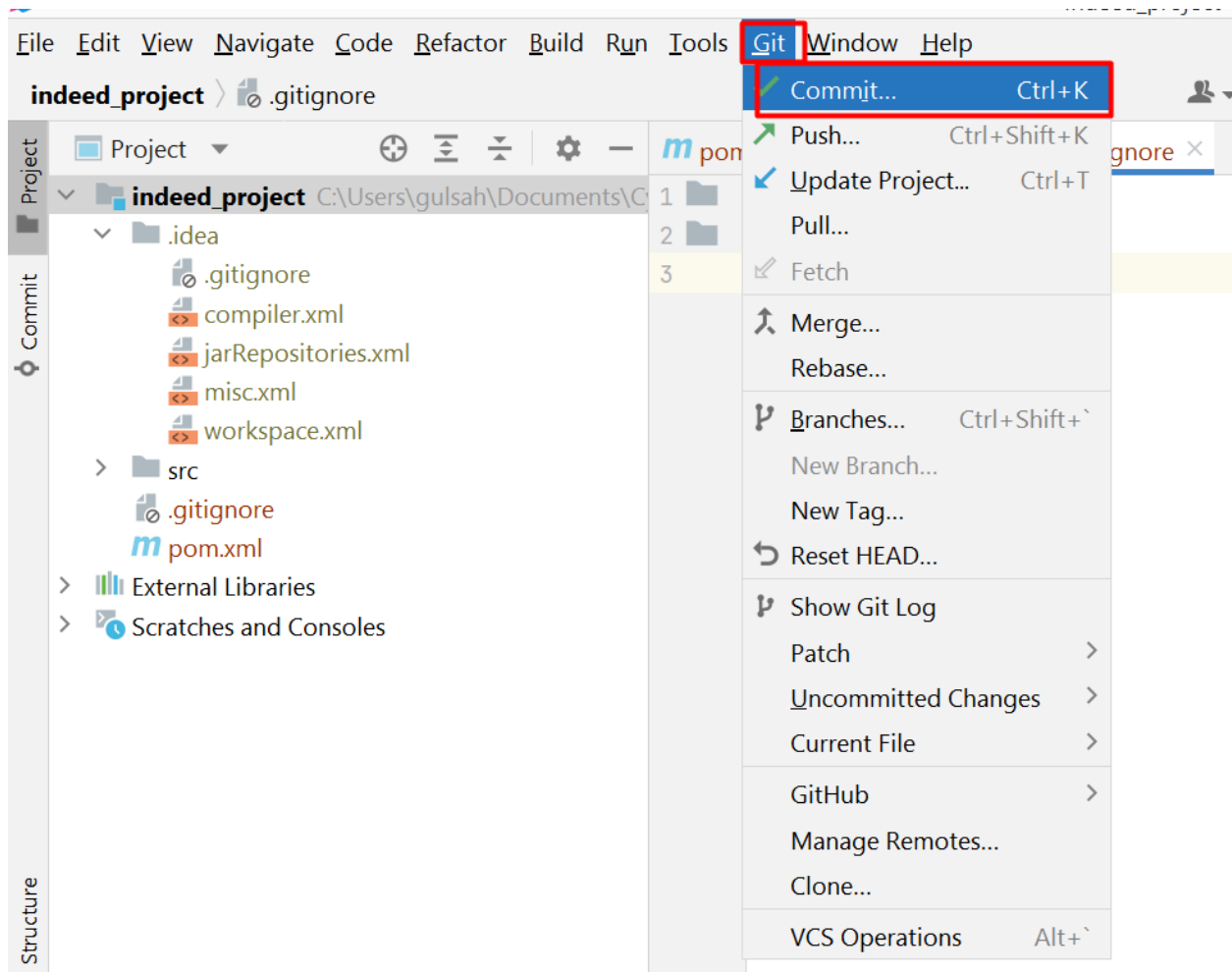
*.iml

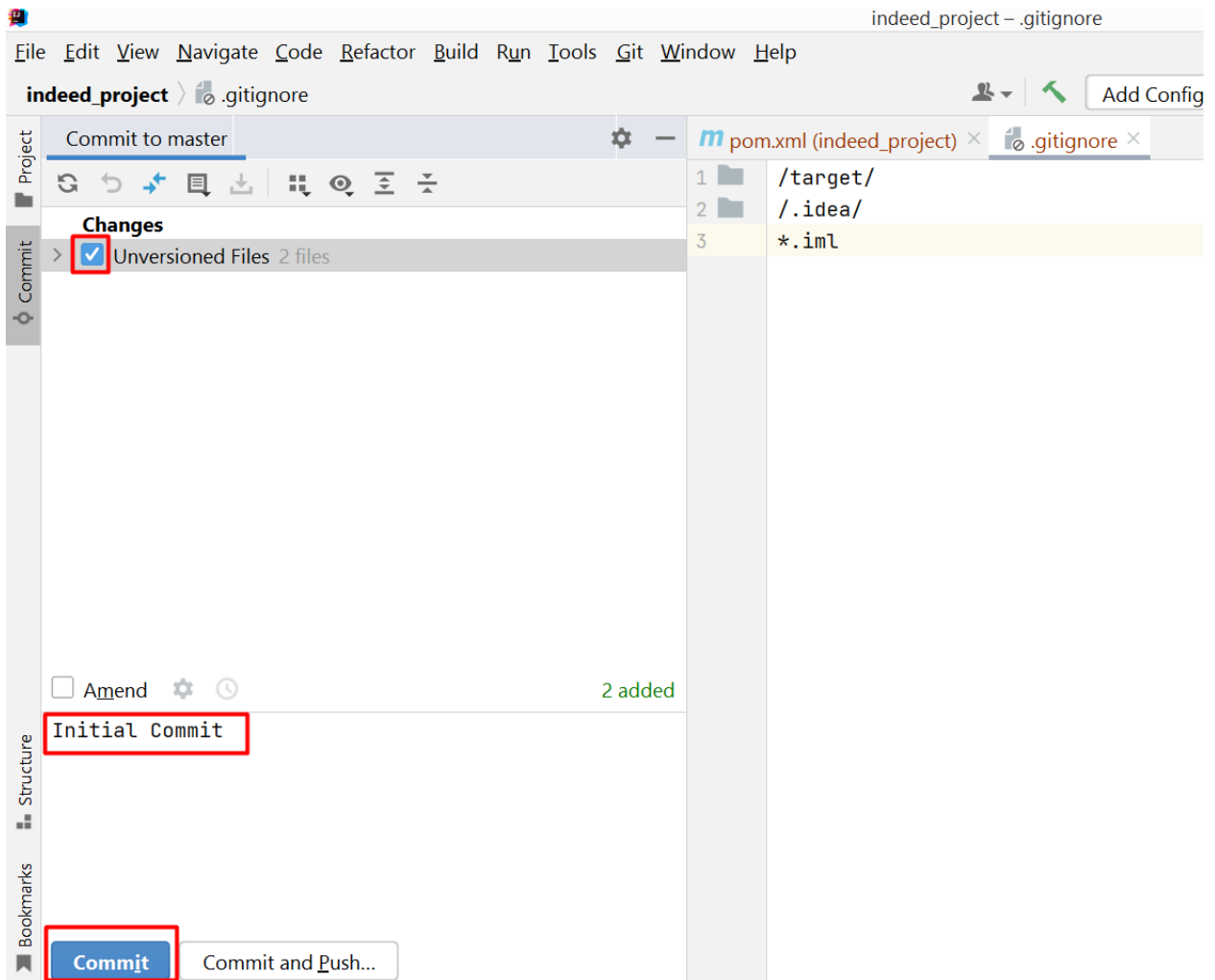


B. Enable Version Control Integration

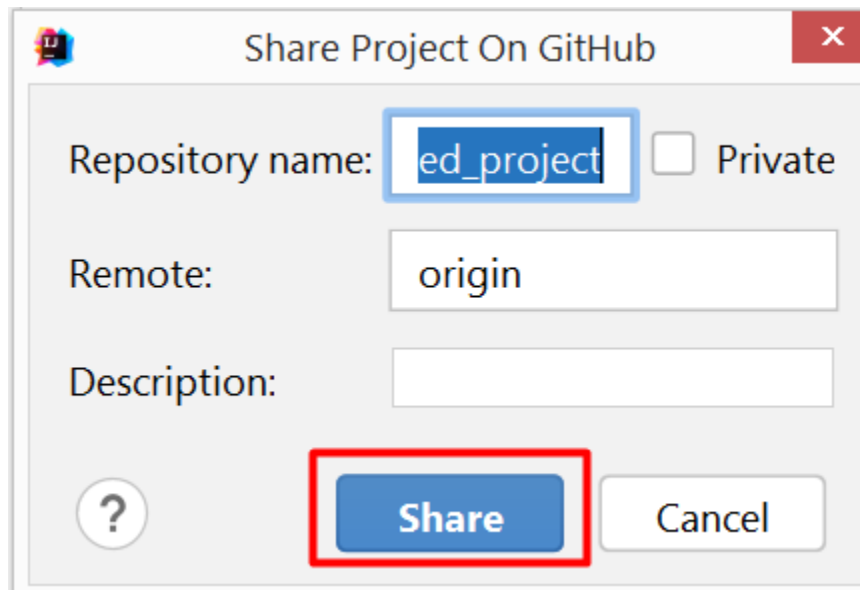
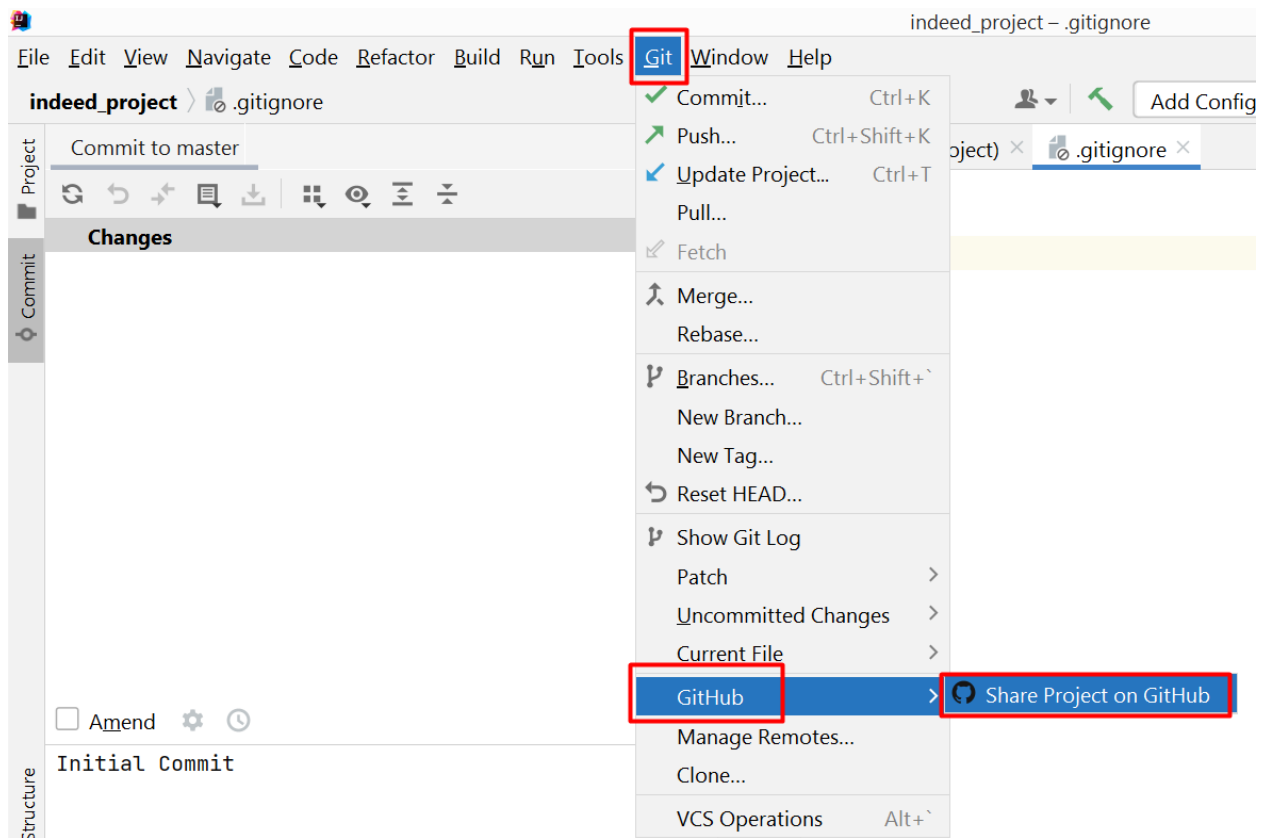


C. Initial Commit





D. Share my project on Github (Remote)



3. Updating pom.xml file

"In my project, I store my plug-ins and dependencies in pom.xml file such as bonigarcia webdriver"

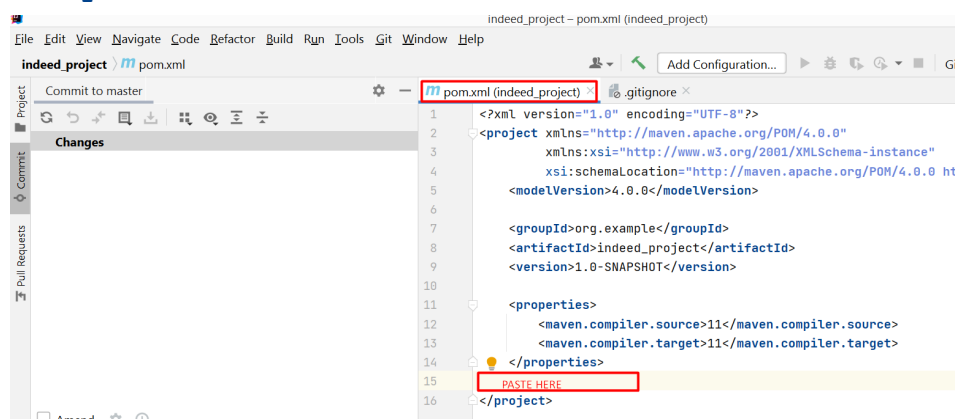
A. Copy and paste followings to the pom.xml file

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.0.0-M5</version>
      <configuration>
        <parallel>methods</parallel>
        <useUnlimitedThreads>true</useUnlimitedThreads>
        <!--
<threadCount>4</threadCount>-->
        <testFailureIgnore>true</testFailureIgnore>
        <includes>
          <include>*/CukesRunner*.java</include>
        </includes>
      </configuration>
    </plugin>
  </plugins>
</build>
<dependencies>
  <!--
https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.141.59</version>
  </dependency>
  <!--
https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
  <dependency>
    <groupId>io.github.bonigarcia</groupId>
    <artifactId>webdrivermanager</artifactId>
    <version>5.1.0</version>
  </dependency>
```

```

<!--
https://mvnrepository.com/artifact/com.github.javafaker/javafake
r -->
<dependency>
  <groupId>com.github.javafaker</groupId>
  <artifactId>javafaker</artifactId>
  <version>1.0.2</version>
</dependency>
<!--
https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
<dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-java</artifactId>
  <version>7.2.3</version>
</dependency>
<!--
https://mvnrepository.com/artifact/io.cucumber/cucumber-junit
-->
<dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-junit</artifactId>
  <version>7.2.3</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>me.jvt.cucumber</groupId>
  <artifactId>reporting-plugin</artifactId>
  <version>7.2.0</version>
</dependency>
</dependencies>

```



B. Update maven



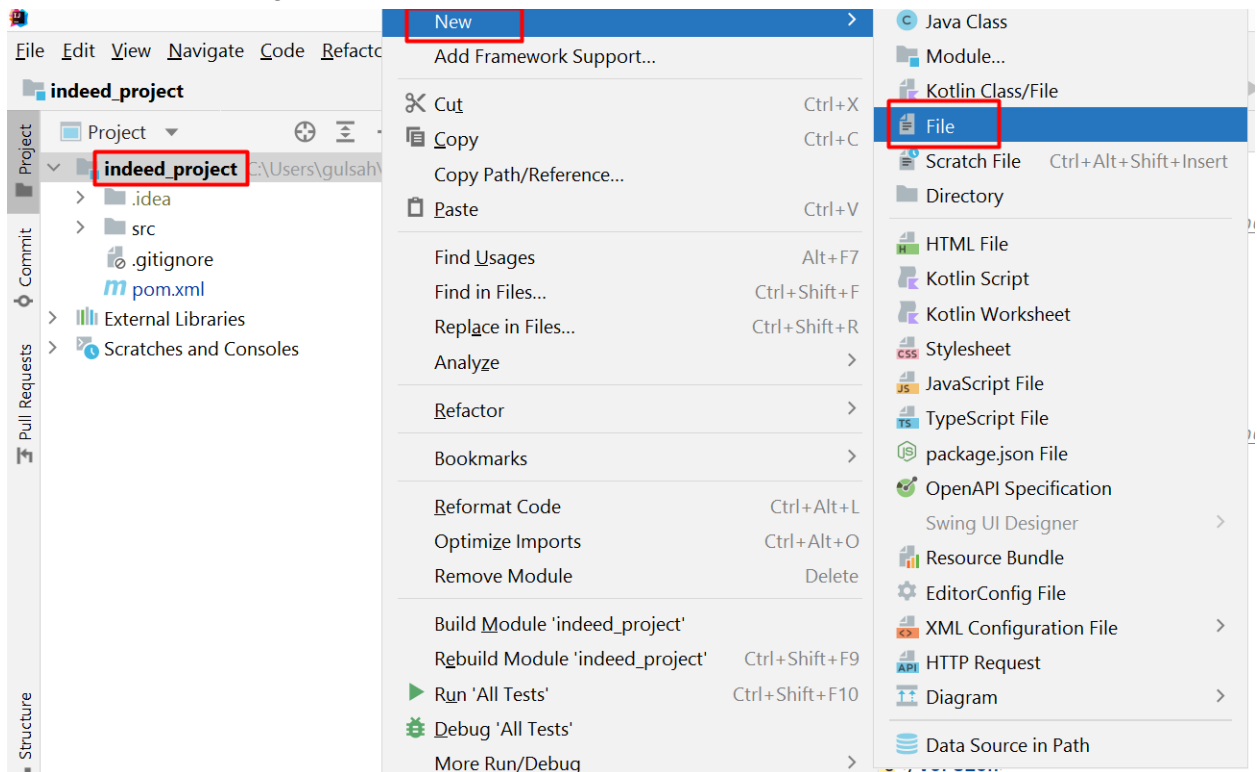
The screenshot shows an IDE window with a file named `dependencies` and a `.gitignore` file. The code in the `dependencies` file is as follows:

```
<dependencies>  
  </dependency>  
  
  <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->  
  <dependency>  
    <groupId>io.cucumber</groupId>  
    <artifactId>cucumber-java</artifactId>  
    <version>7.2.3</version>  
  </dependency>  
  
  <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-junit -->  
  <dependency>  
    <groupId>io.cucumber</groupId>  
    <artifactId>cucumber-junit</artifactId>  
    <version>7.2.3</version>  
    <scope>test</scope>  
  </dependency>  
  
  <dependency>  
    <groupId>me.jvt.cucumber</groupId>  
    <artifactId>reporting-plugin</artifactId>  
    <version>7.2.0</version>  
  </dependency>  
  
</dependencies>  
</project>
```

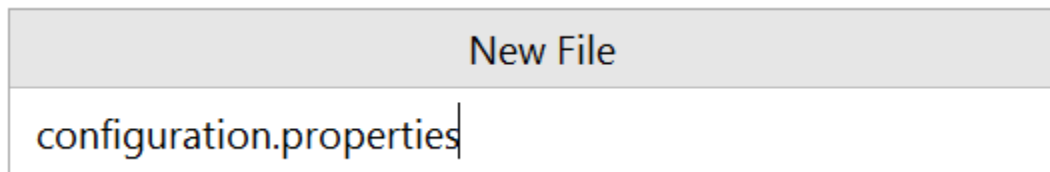
On the right side of the IDE, there is a Maven icon (a red square with a white 'm') and a red box around it. Below the icon, the word "UPDATE" is written in red. In the top right corner of the IDE window, there is a green checkmark icon with the number "1" next to it, and a small upward and downward arrow icon.

4. Create configuration.properties

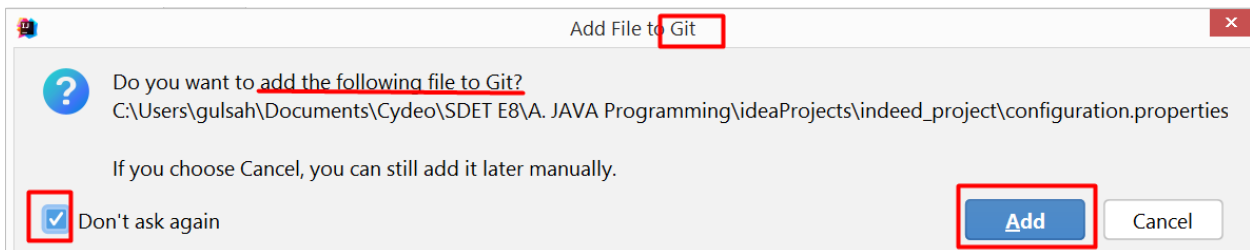
"I store important data in the configuration.properties file so I can easily change my test. So I avoid hardcoding."



A. Type configuration.properties



Note: I have added my configuration.properties files to git



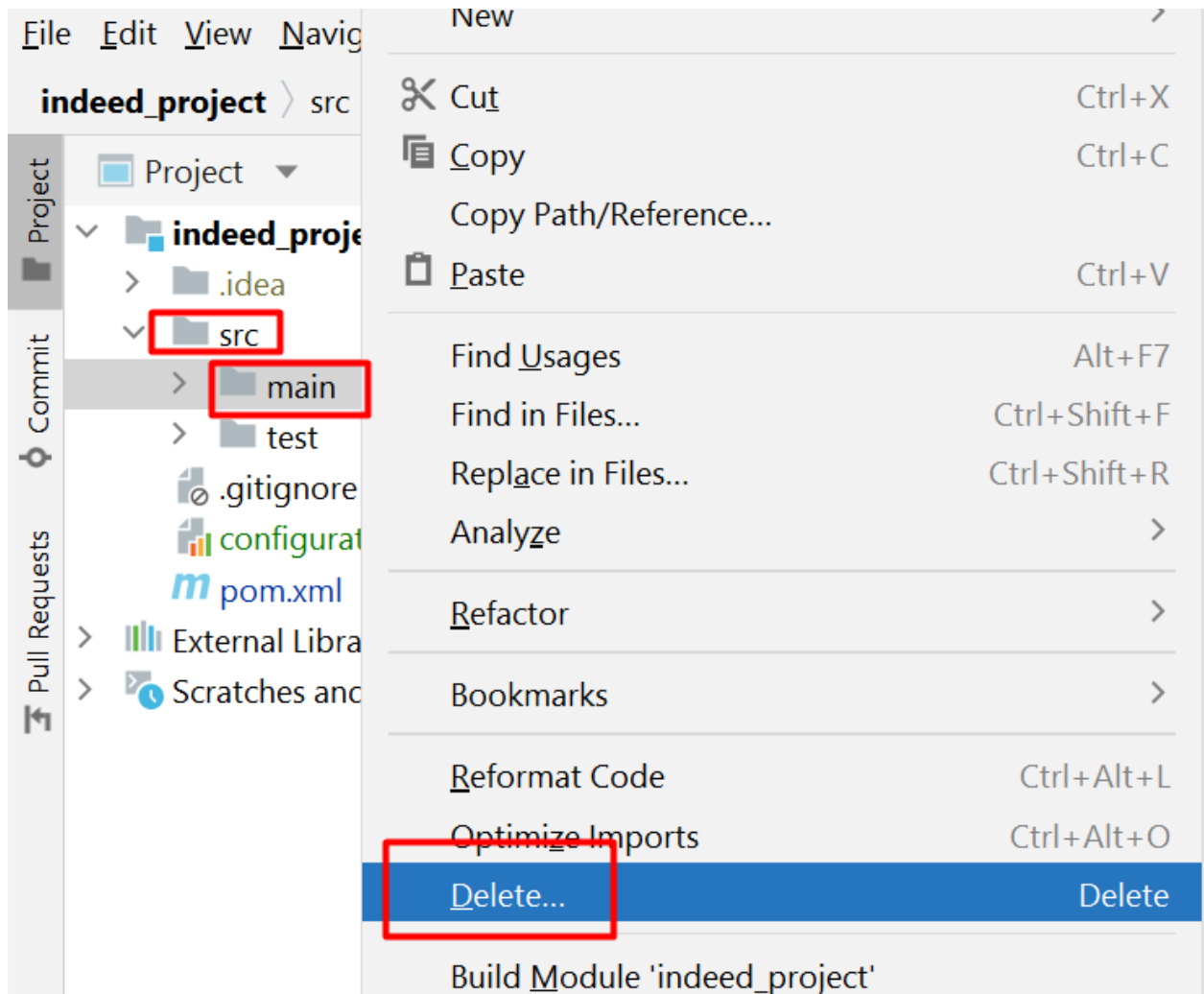
B. Copy and paste the following on configuration.properties

```
browser=chrome  
web.url=https://uk.indeed.com/
```

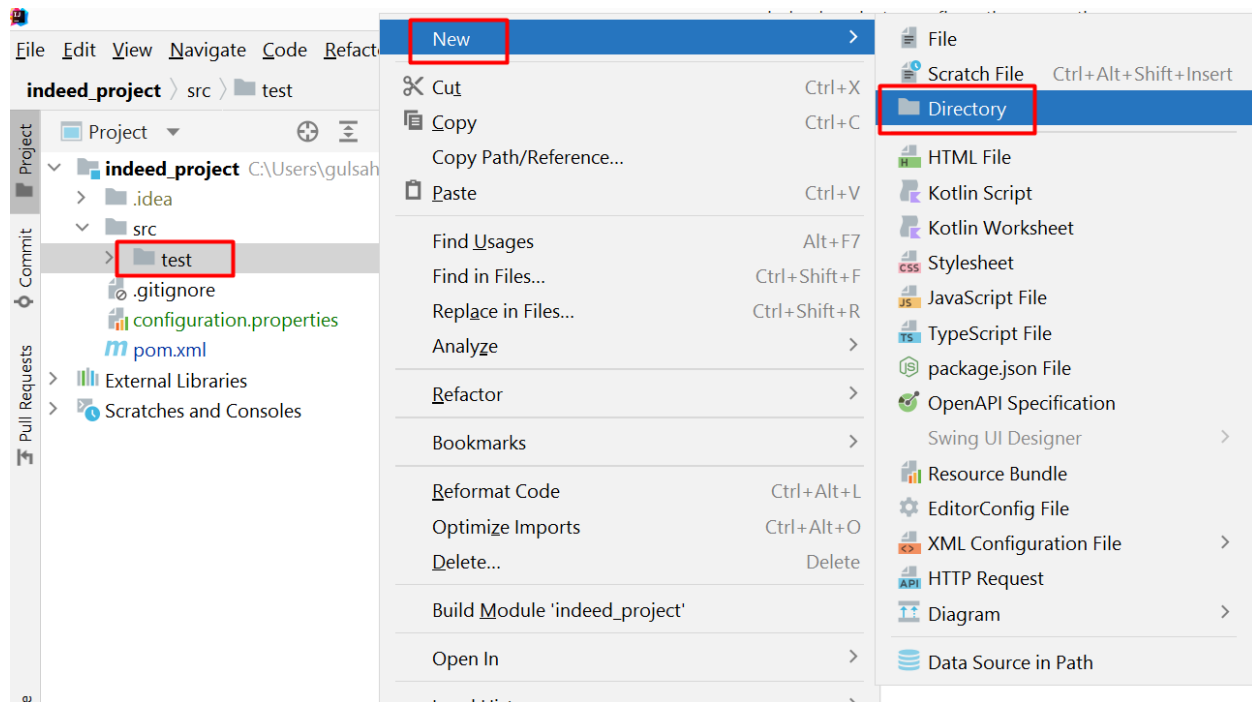


5. Create the folder structure in my framework

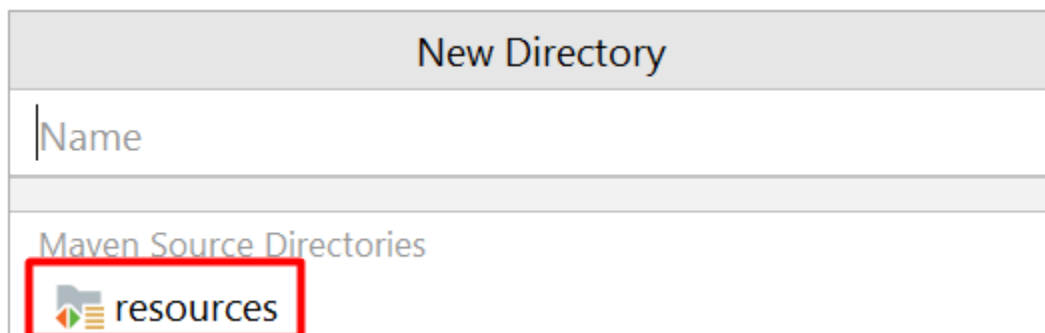
A. Delete main under src folder



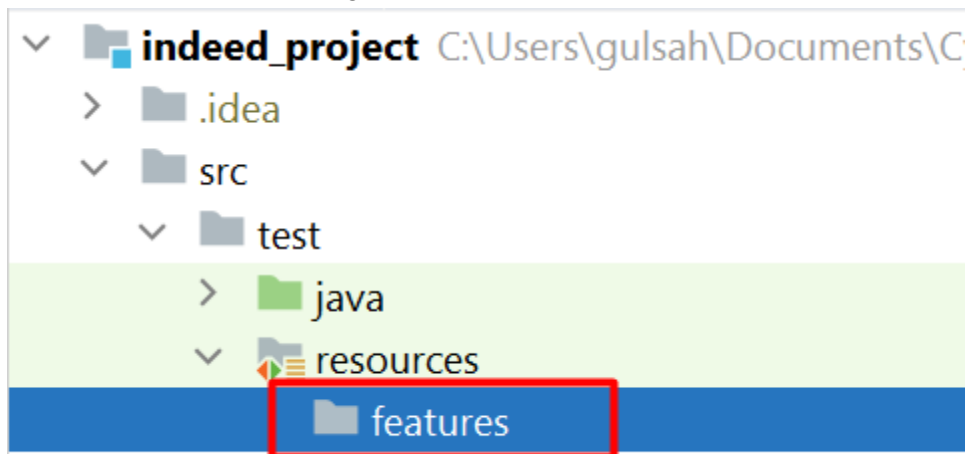
B. Create resources directory under test



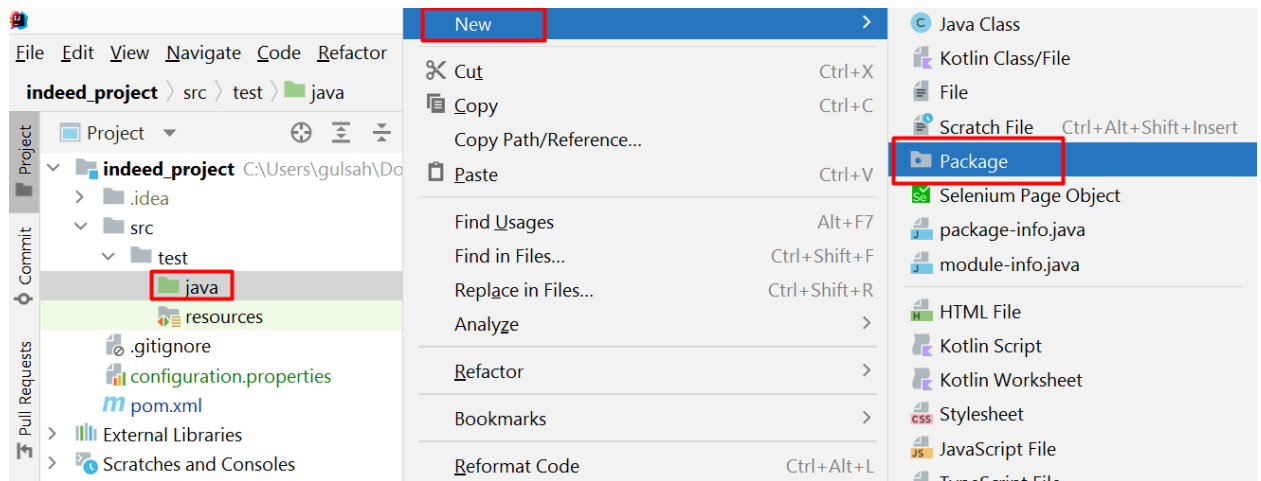
Type resources or click on it



Then create feature package under resources

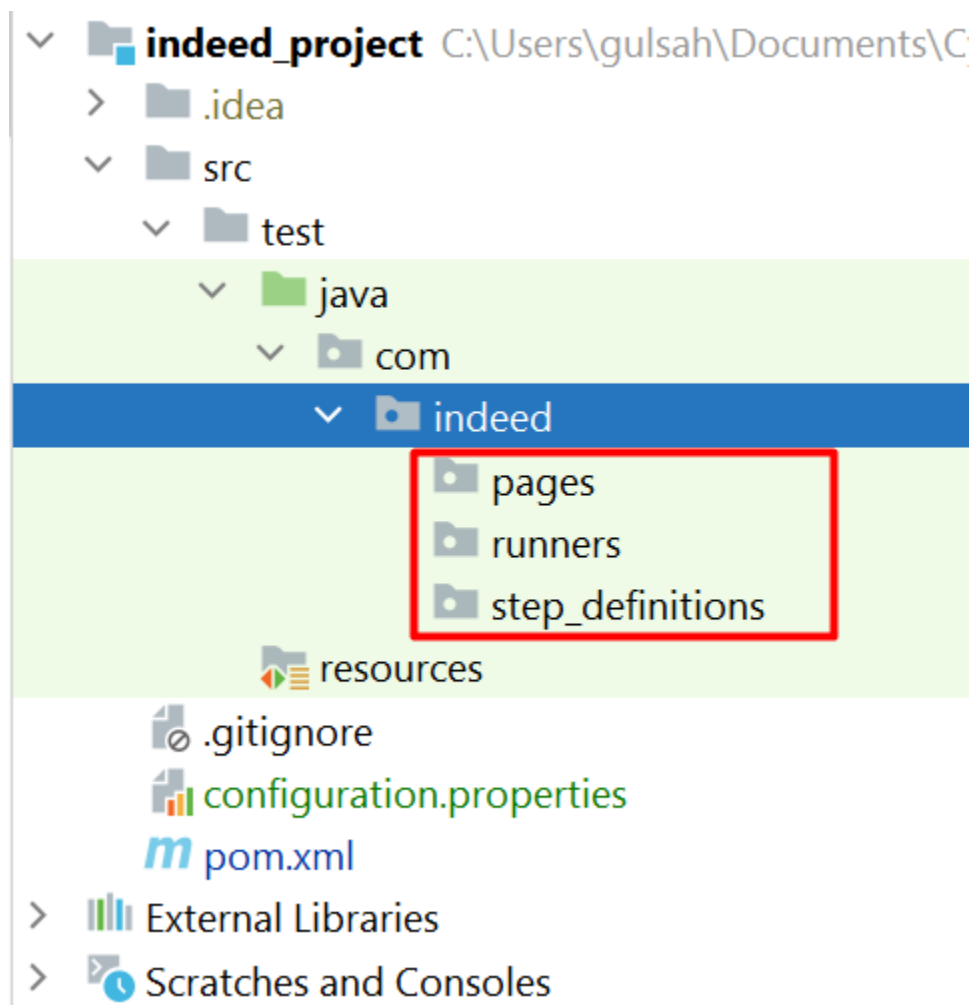


C. Create com.indeed package under java

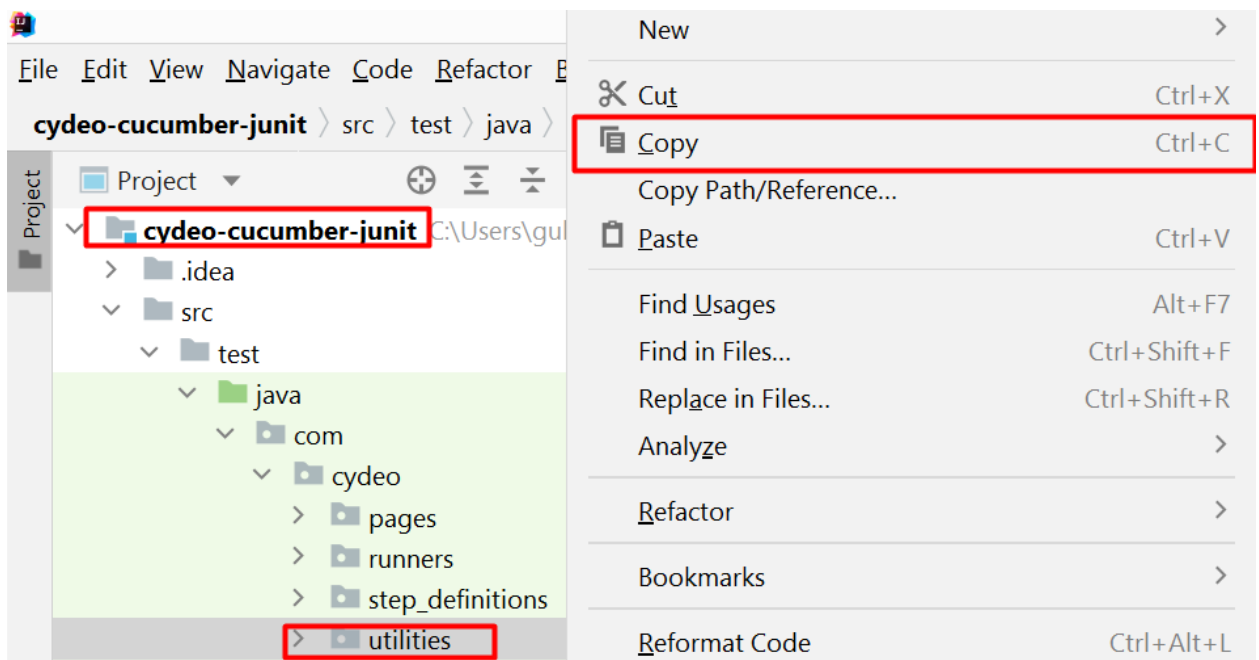


Type com.indeed

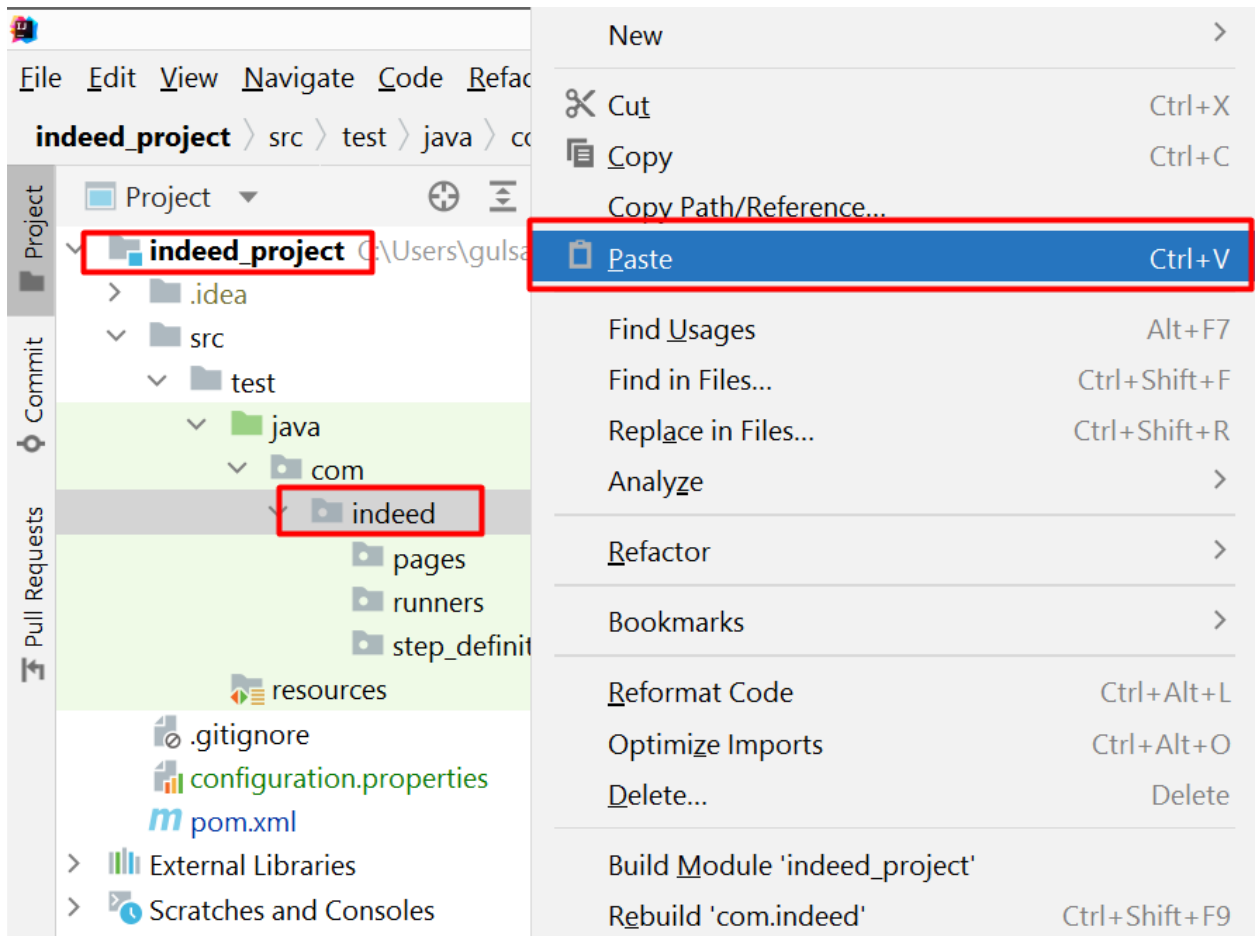
D. Create pages, runners, step_definitions packages under indeed



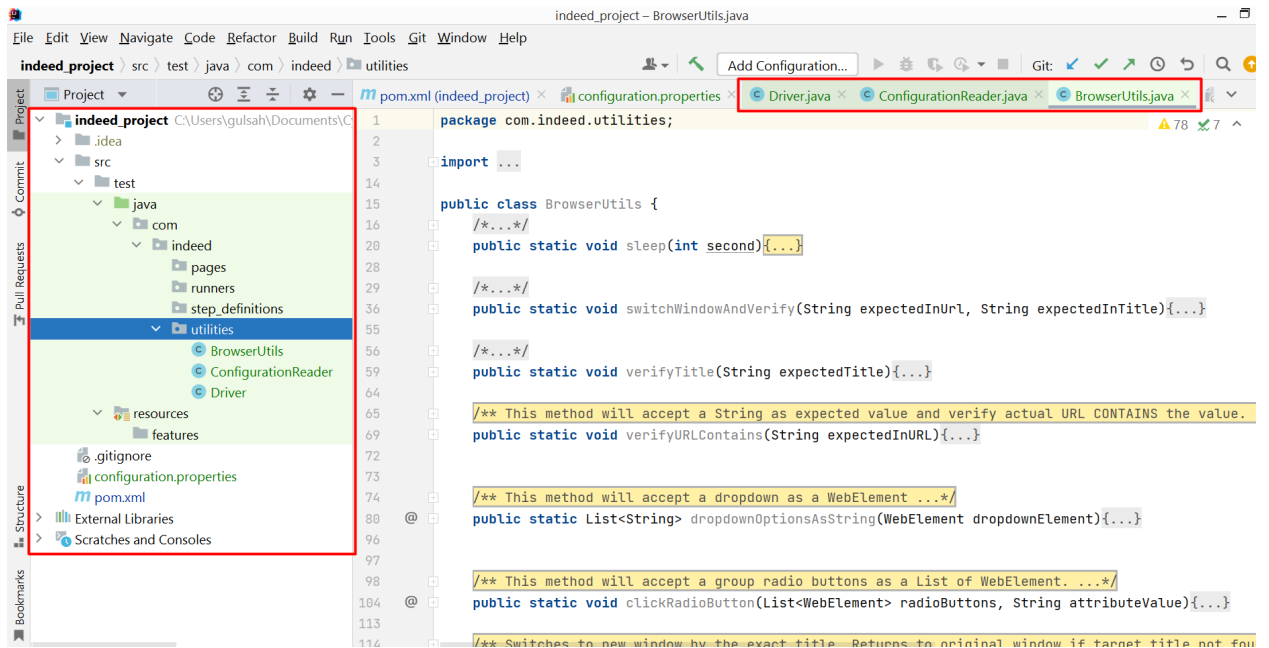
E. Copy utilities from the last project



And paste under indeed package



5. Finally our framework is ready



Note: Please check that auto import is enabled

