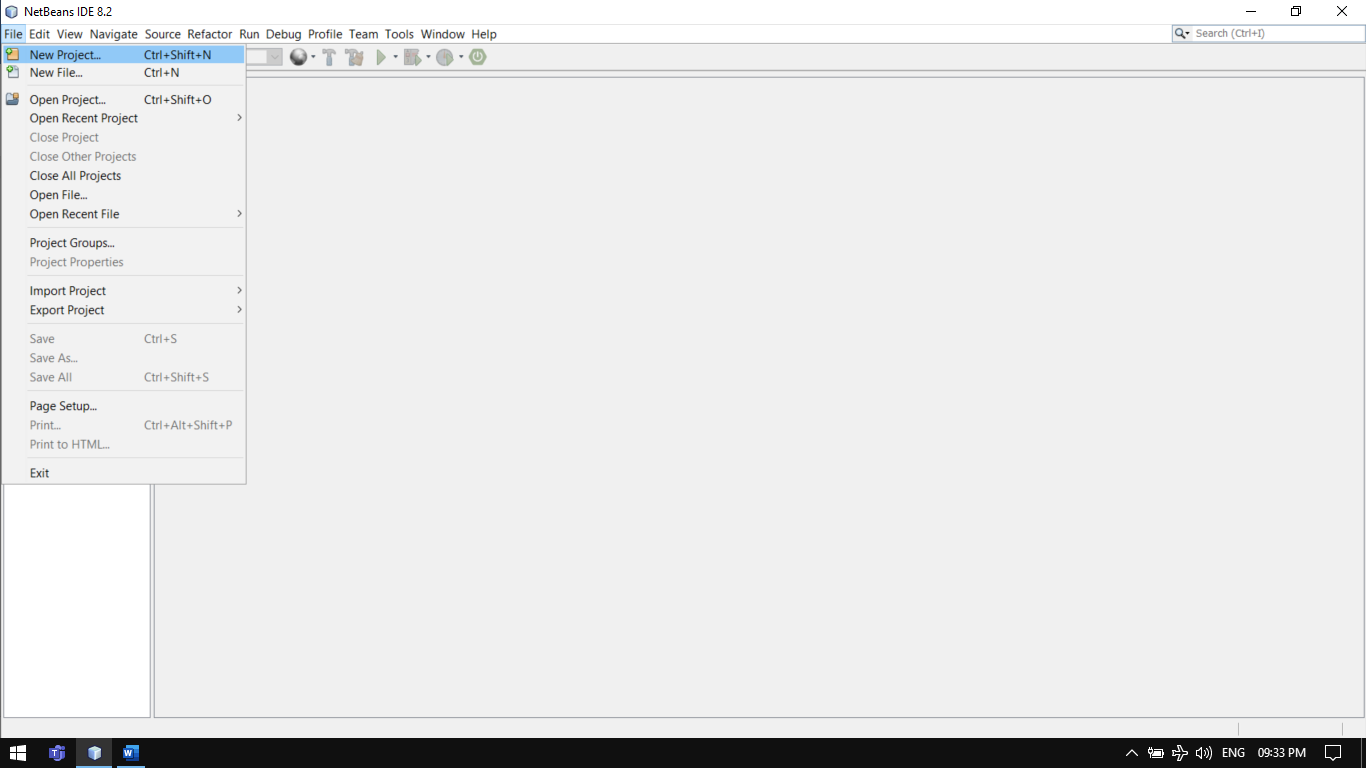
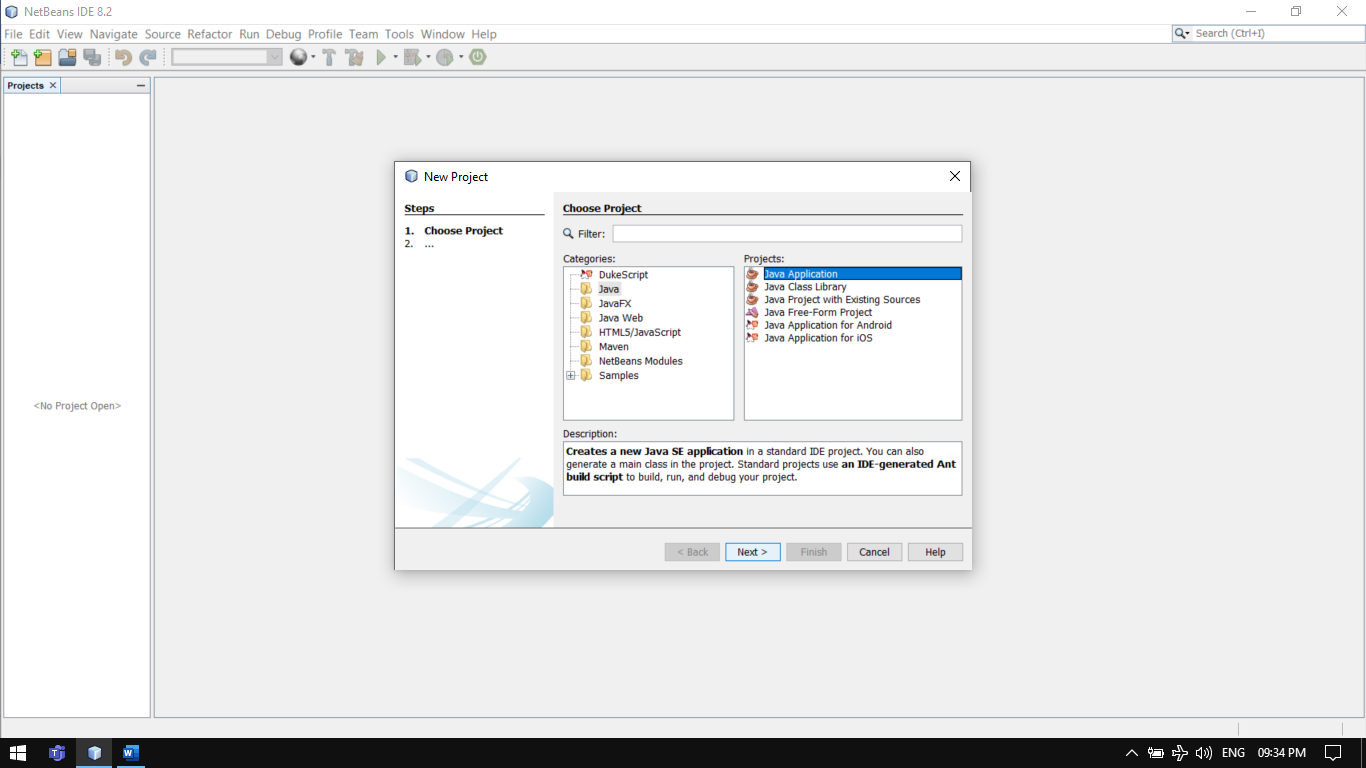
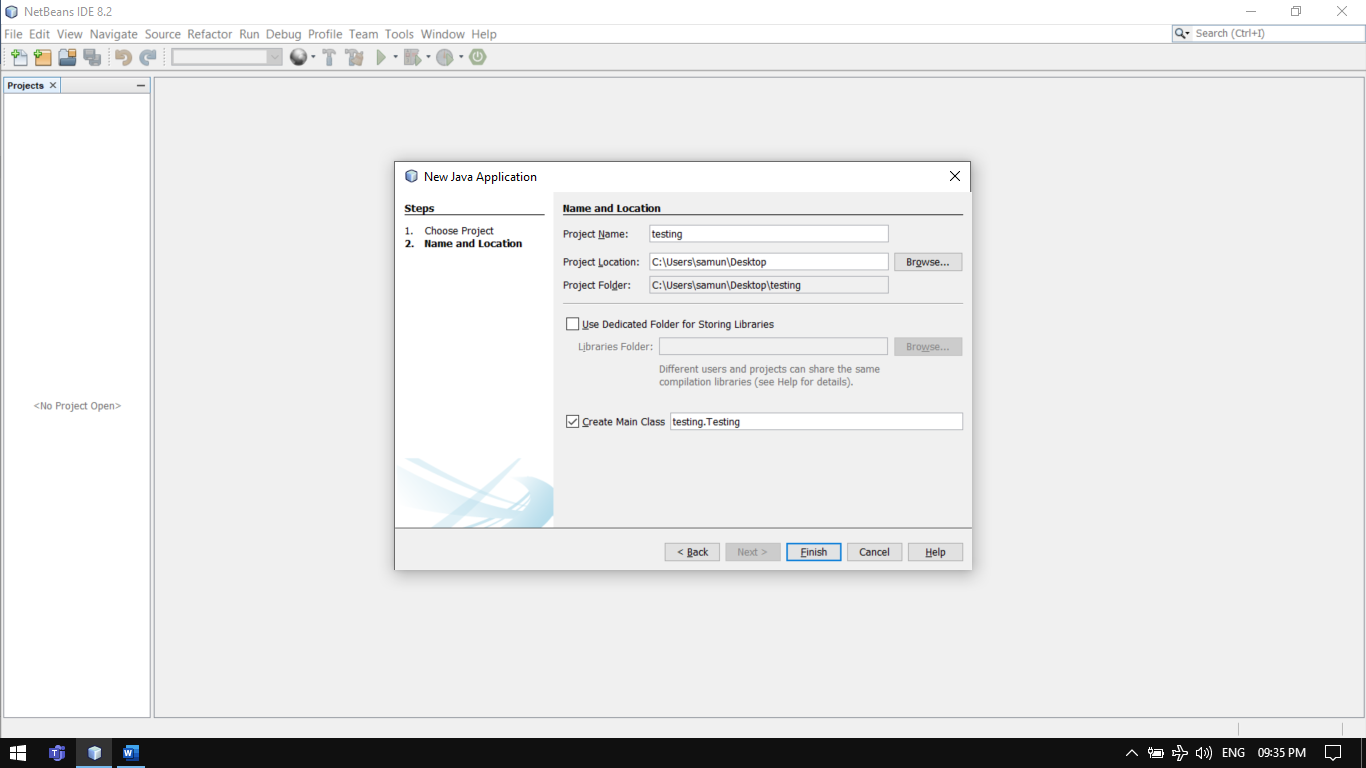
***Hibernate Framework***

**Getting Started:**

**Step 1:** **Select new project** 

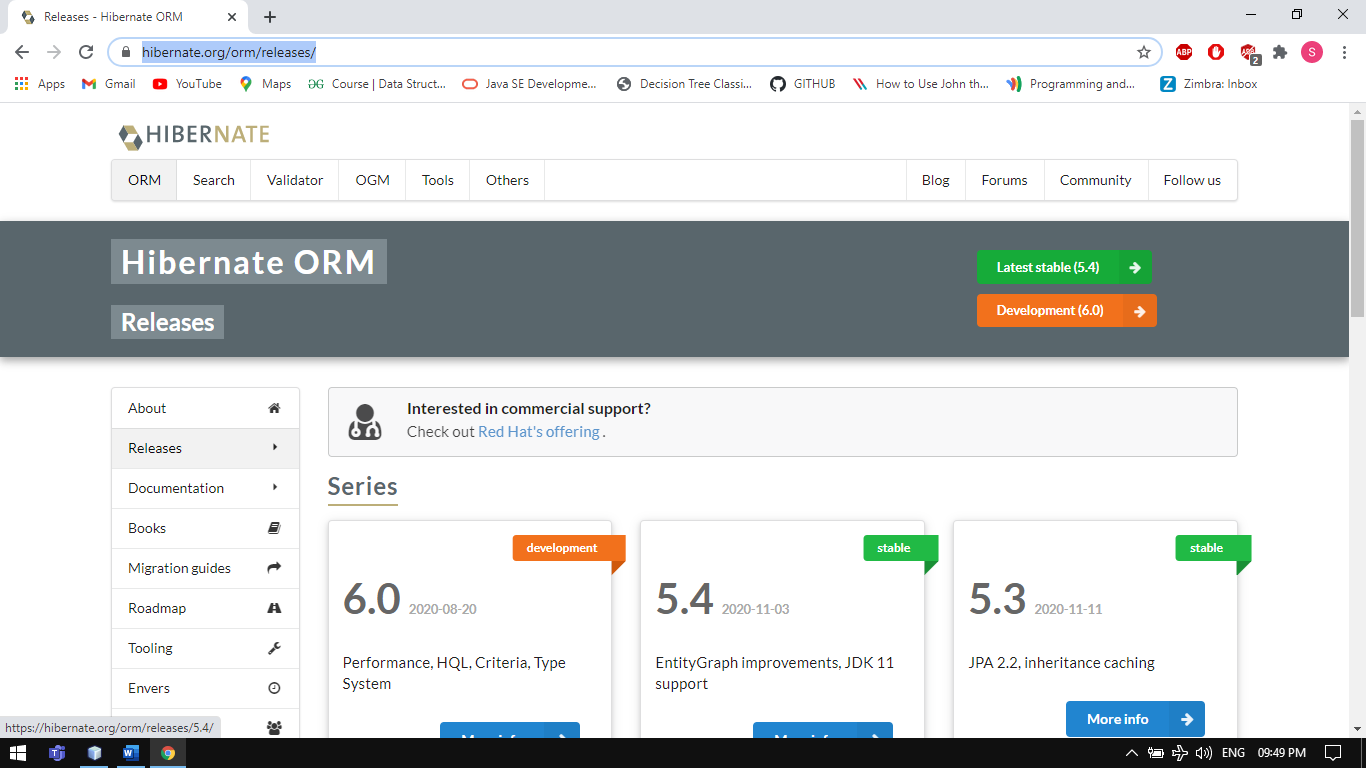
**Step 2: Choose Java Application.** 

**Step 3: Enter Name then Select finish.**



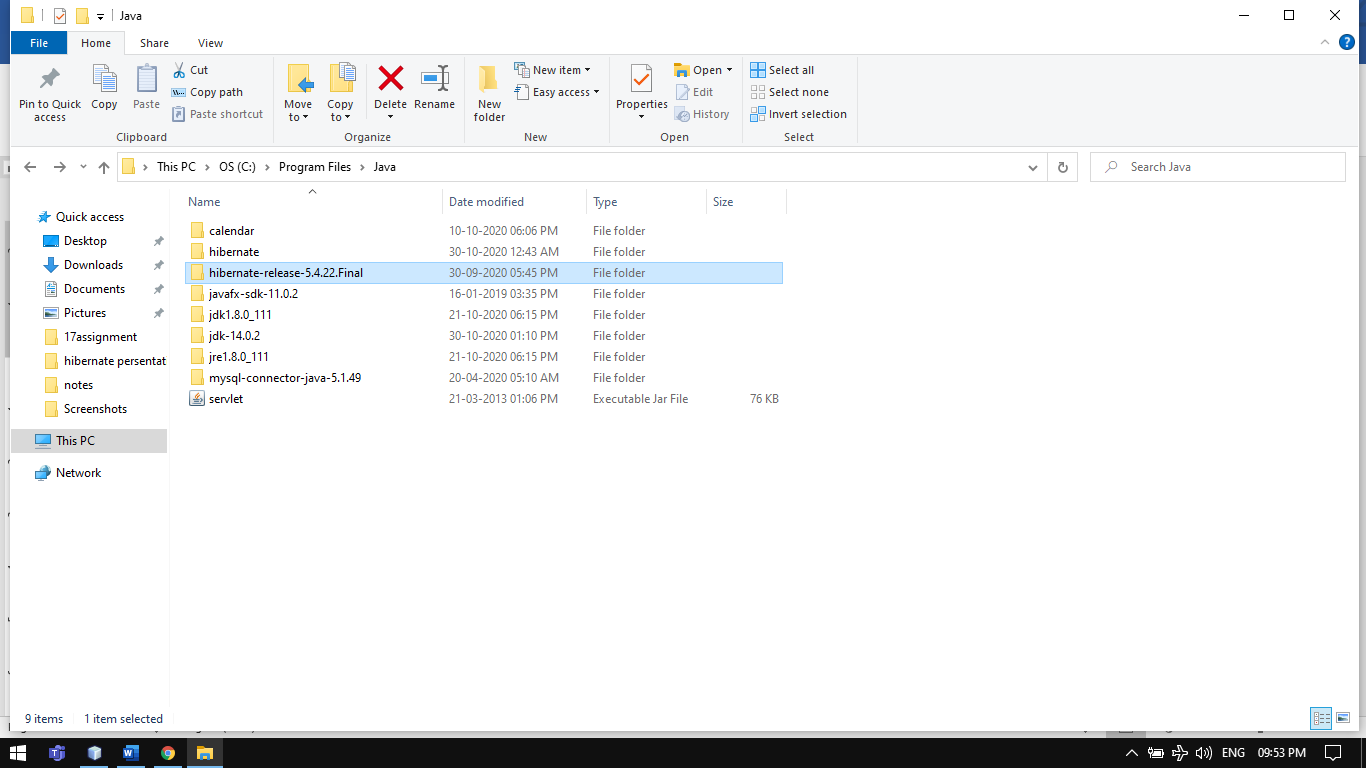
**Step 4: Here we have to download some jar files of hibernate.**

**Visit website** [**https://hibernate.org/orm/releases**](https://hibernate.org/orm/releases)



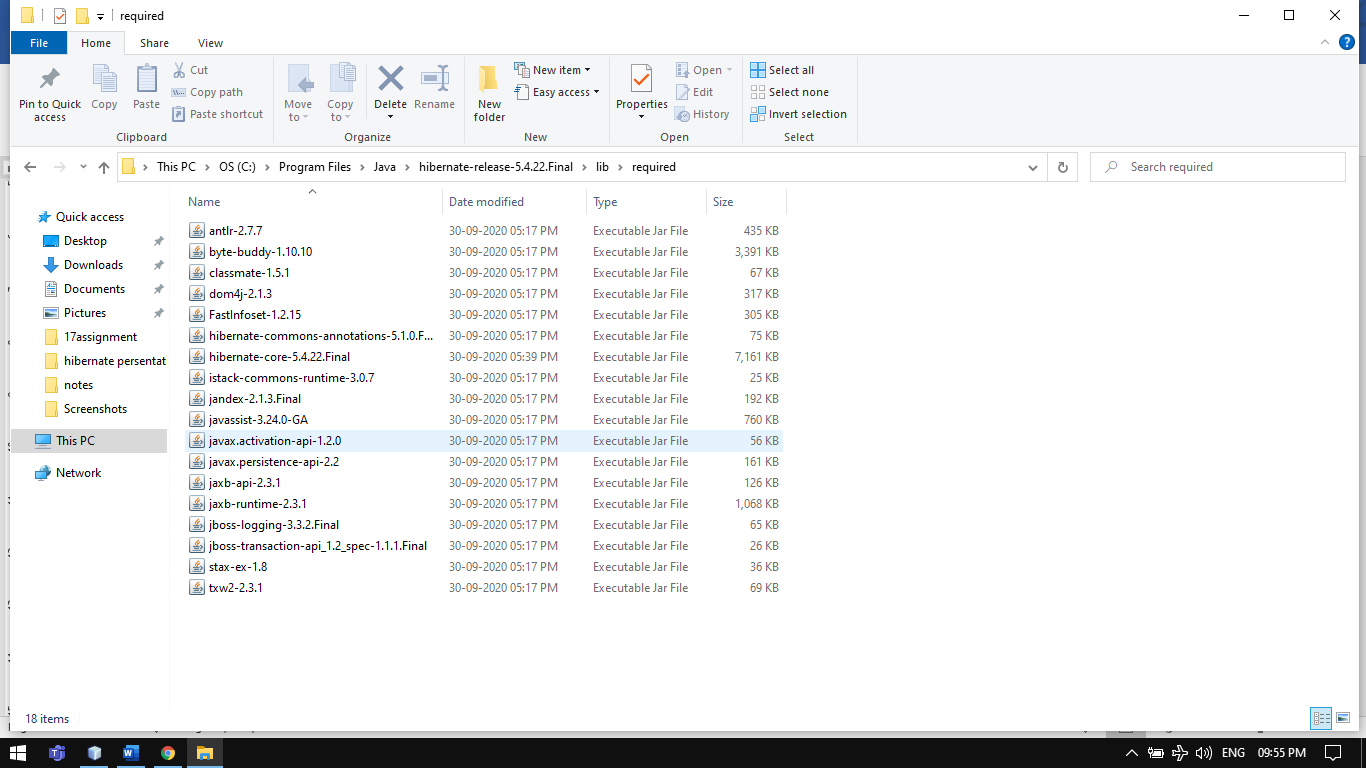
Choose the latest stable version

**Step 5: Download it and Unzip it will be shown like this**



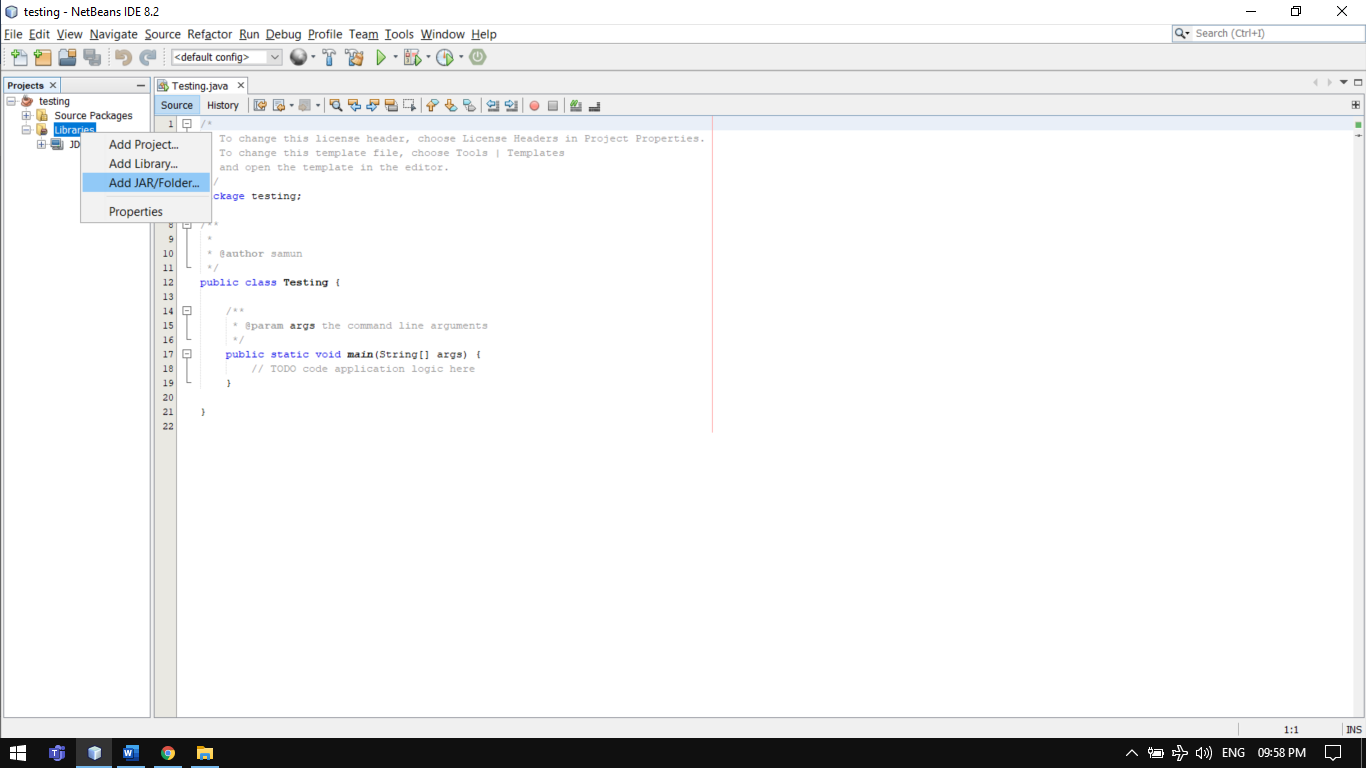
**For simple project we will need very few files so**

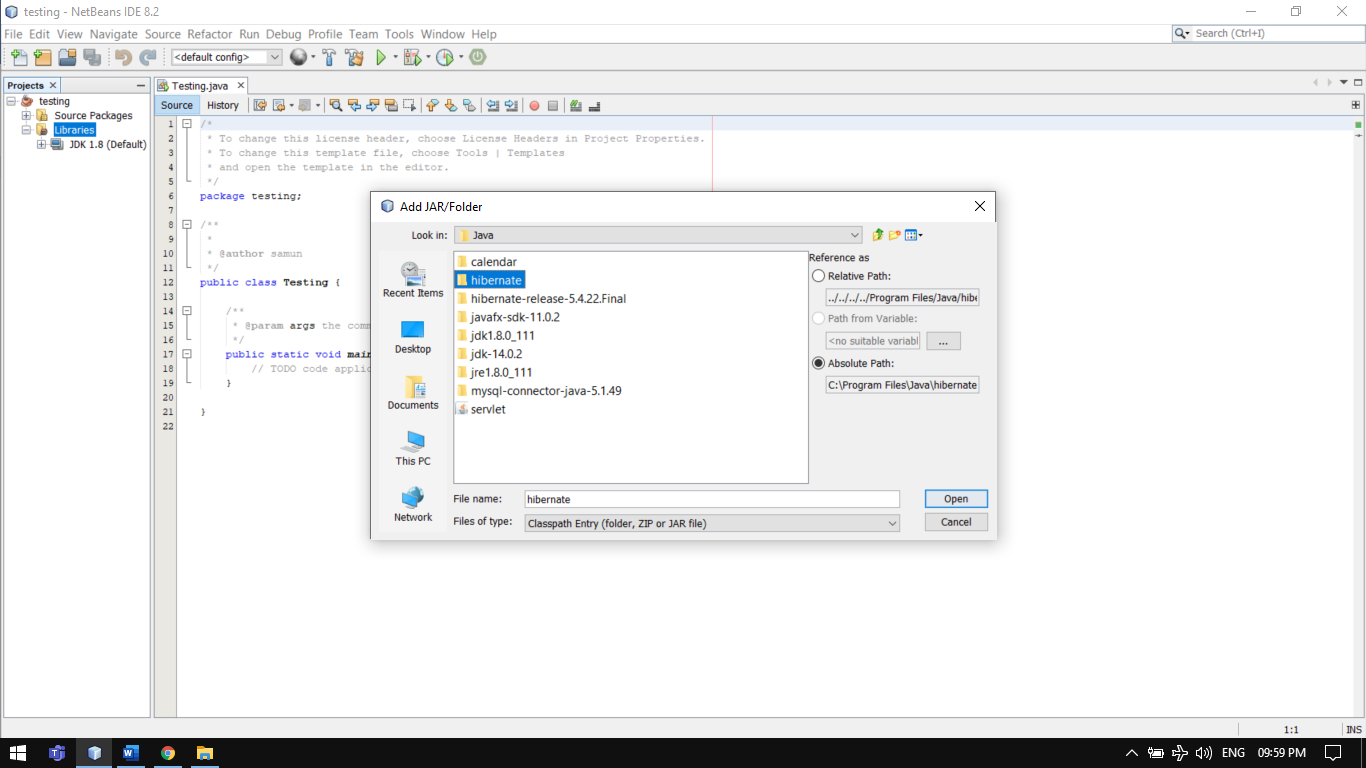
**Go :> hibernate-release-5.4.22.Final\lib\required**



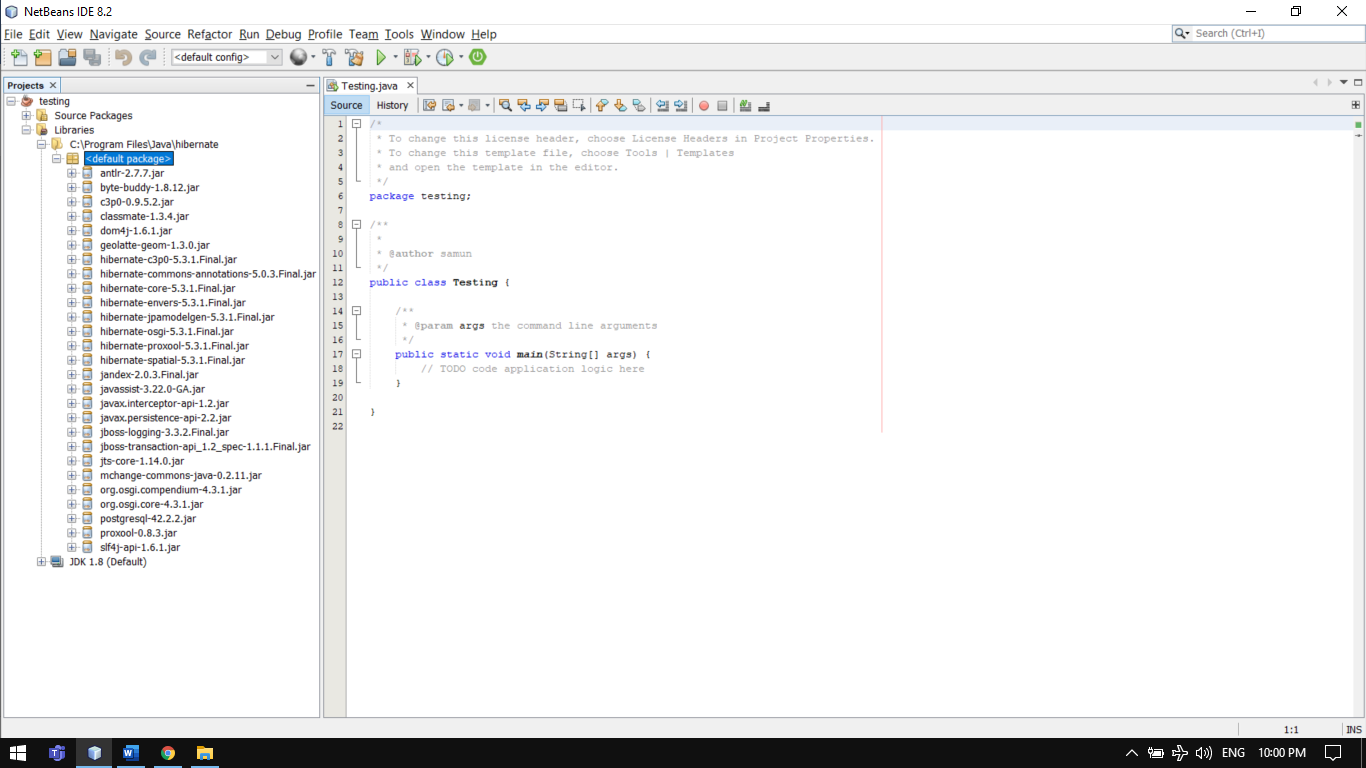
Here we are including only these file(as per requirement add optional jar file it’s totally upto you)

**Step 6: Add the above jar file to your project .**

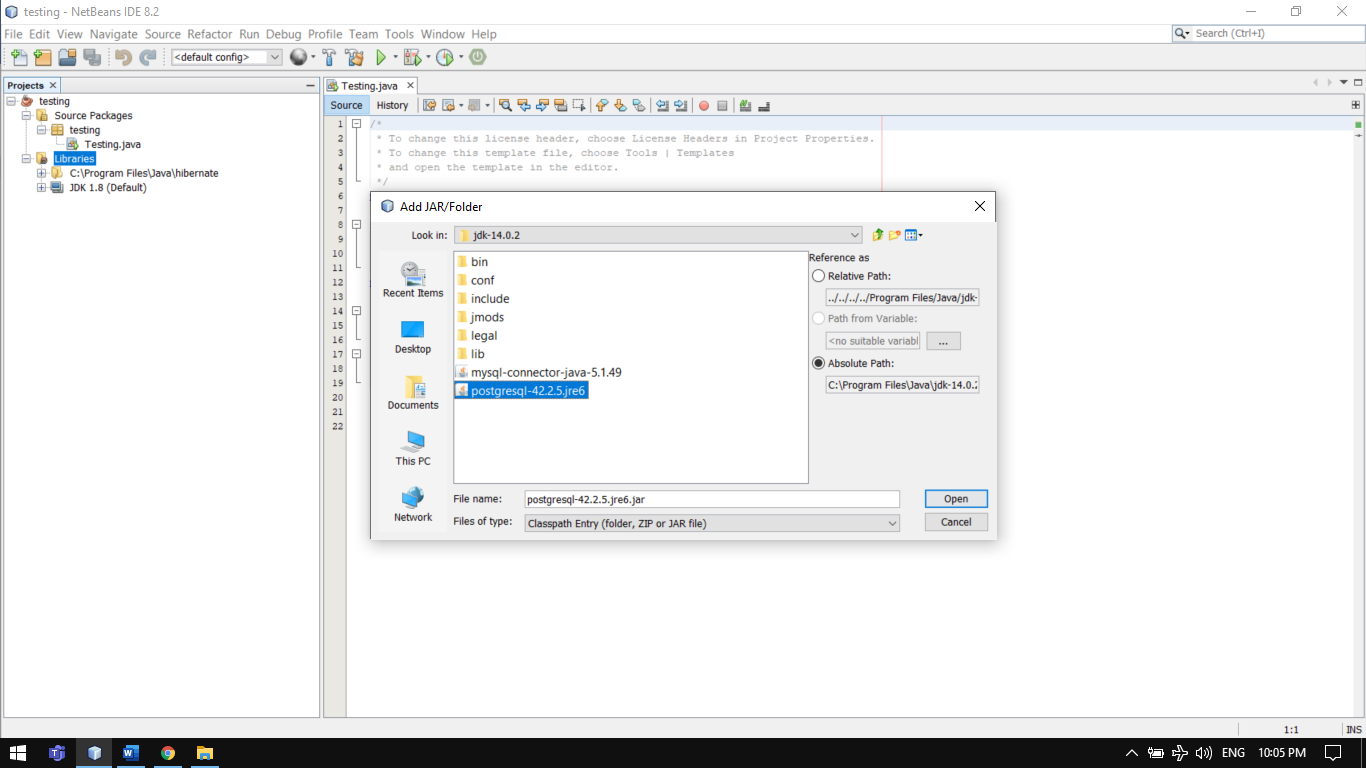


**Select open option.** 

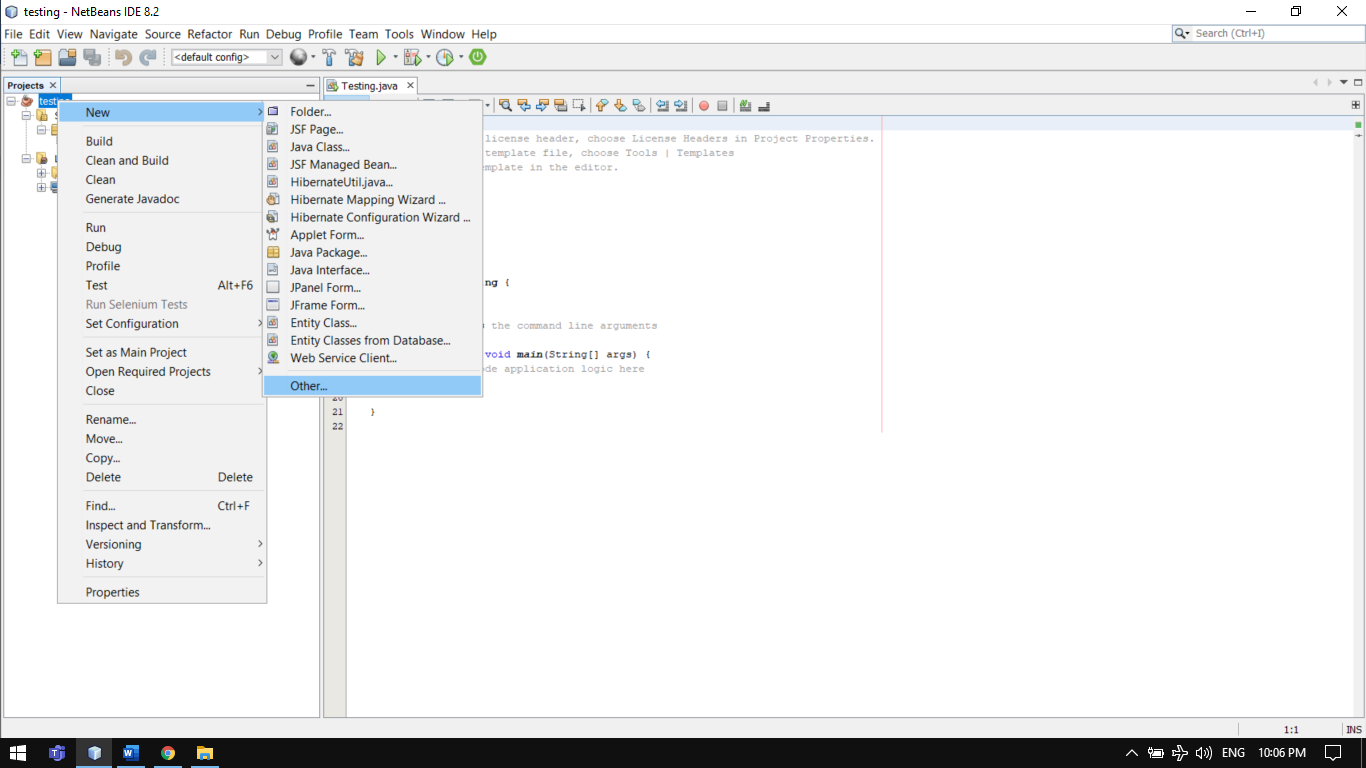
**After including all the jar file it will be shown like this.**



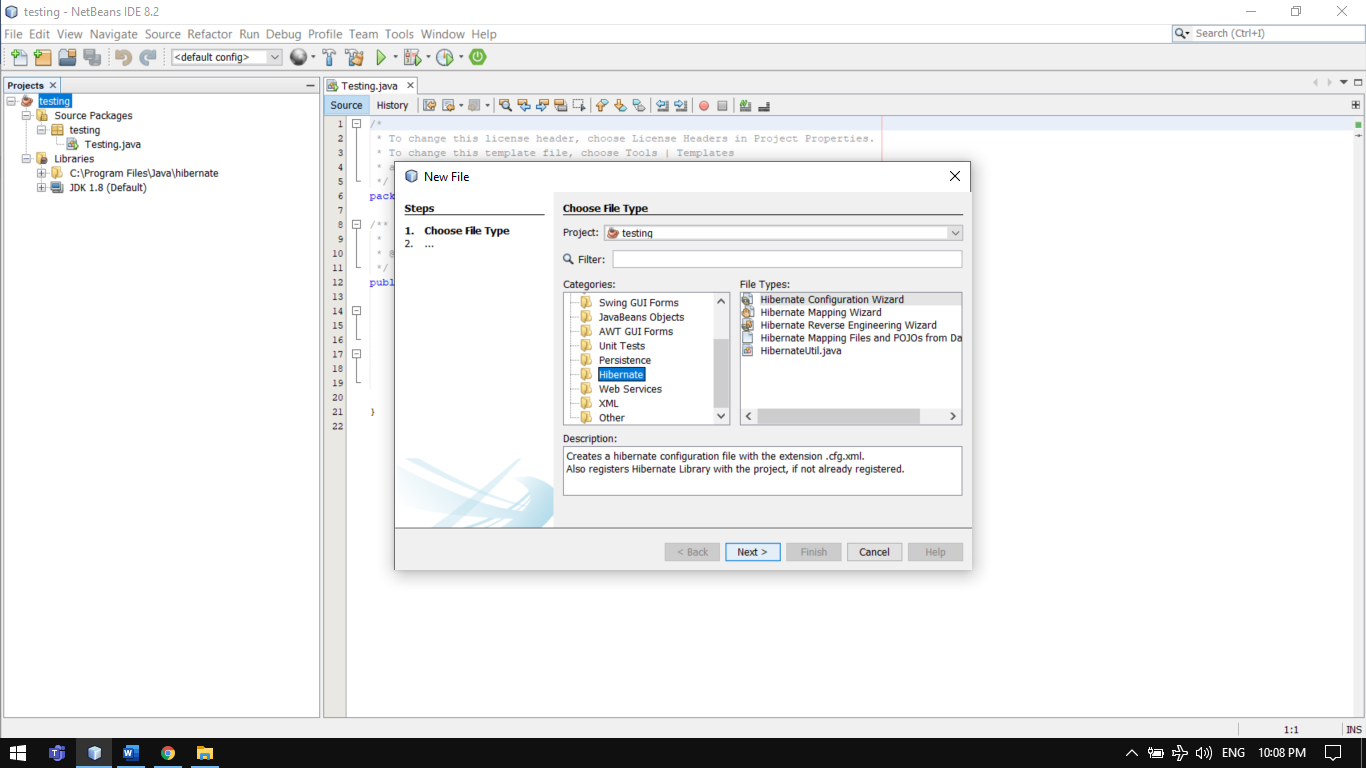
**Step 7: Download the jdbc jar file of the database that u are going to use and include it in same manner as we include the hibernate jar file which is mentioned above.**



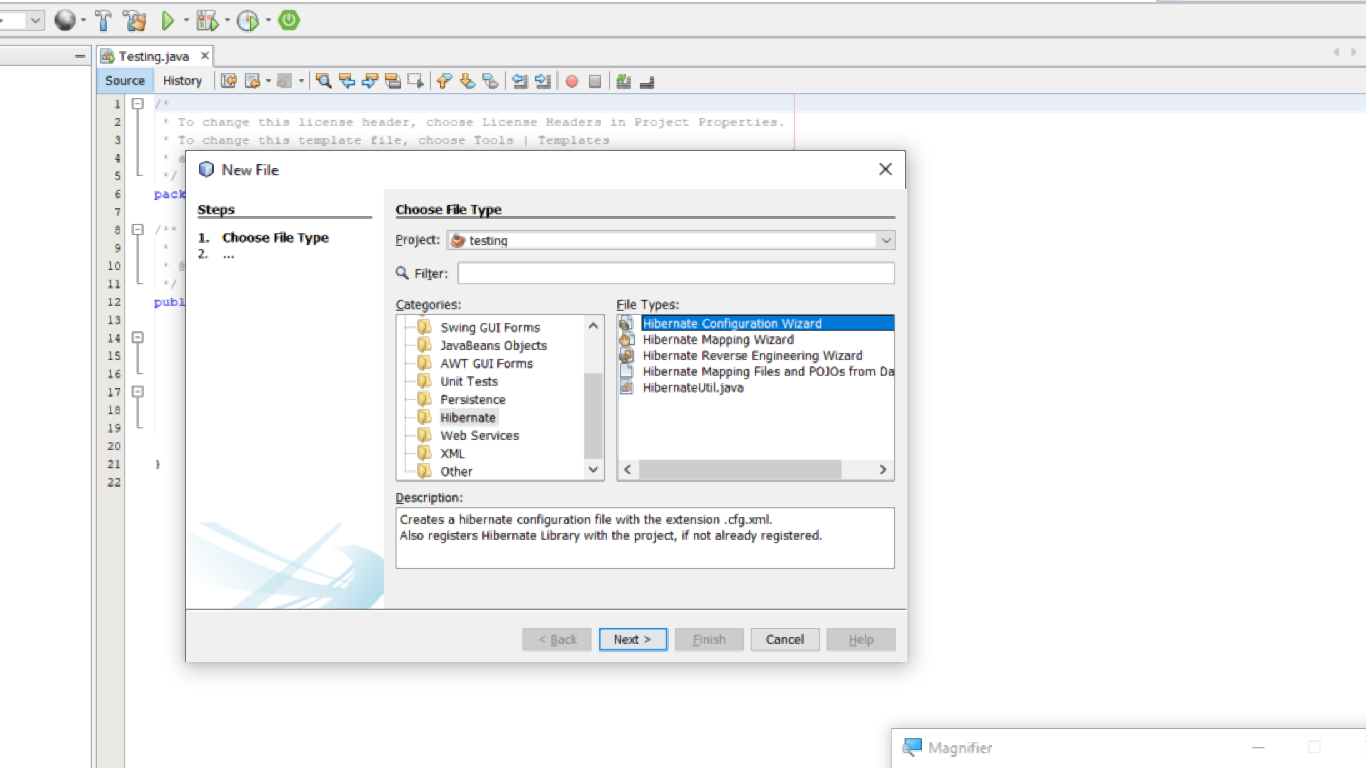
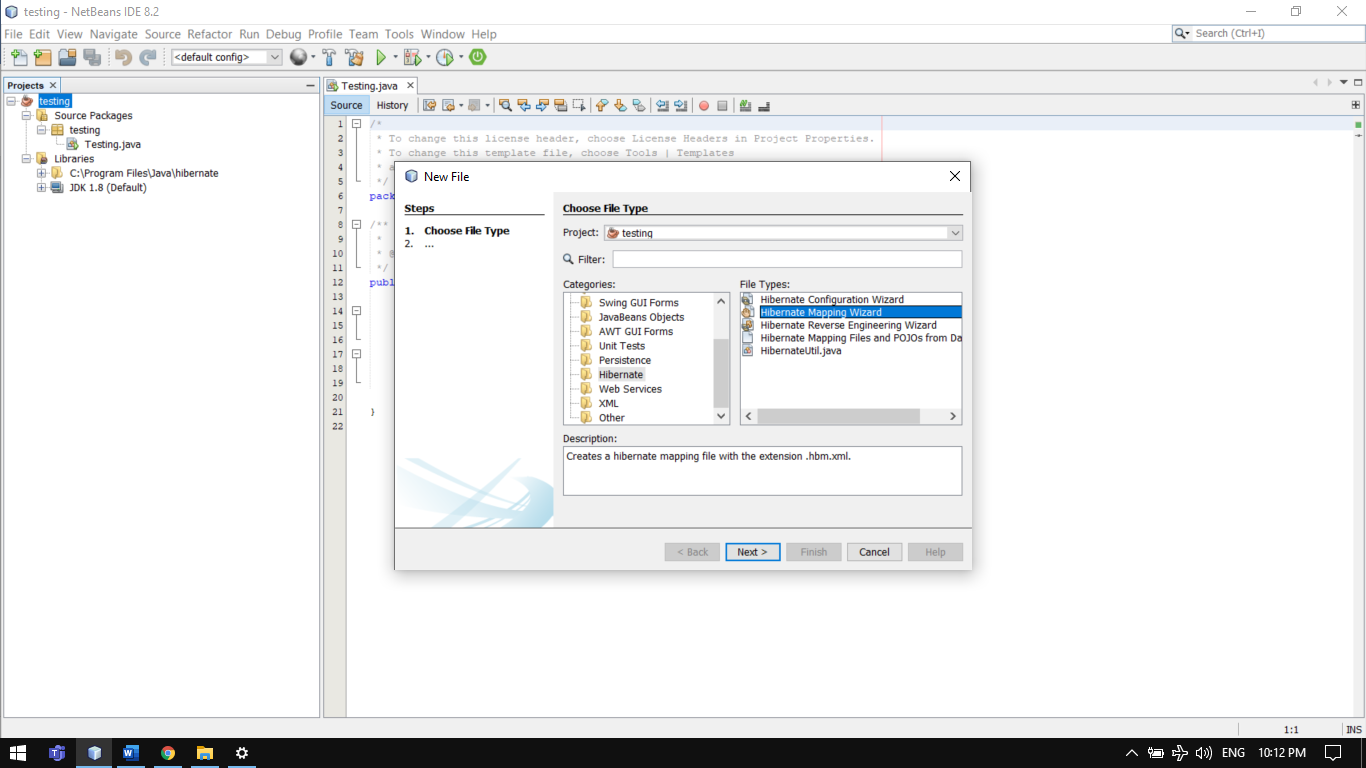
**Step 8: Now our project is configured Properly.to include other files we have to manually include it one by one.**

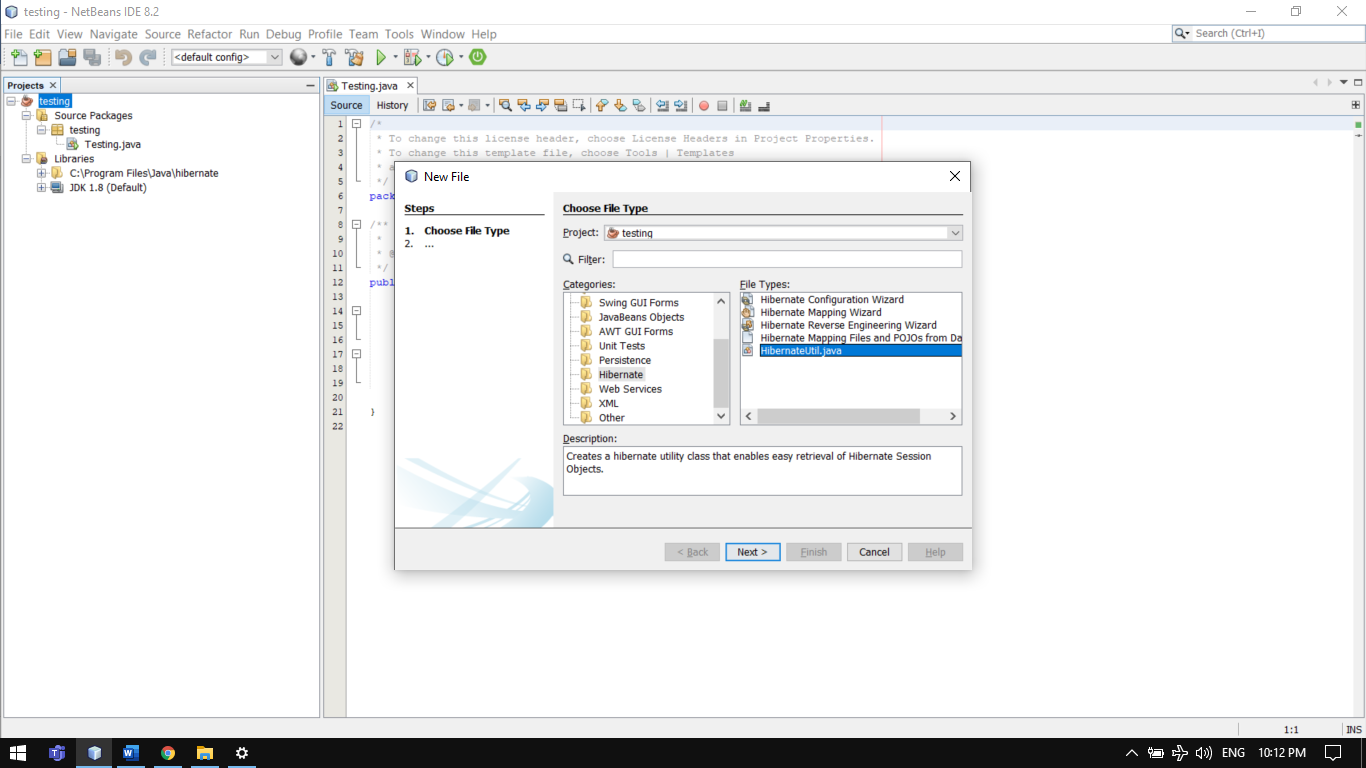


**Step 9: Chose hibernate from the menu.**

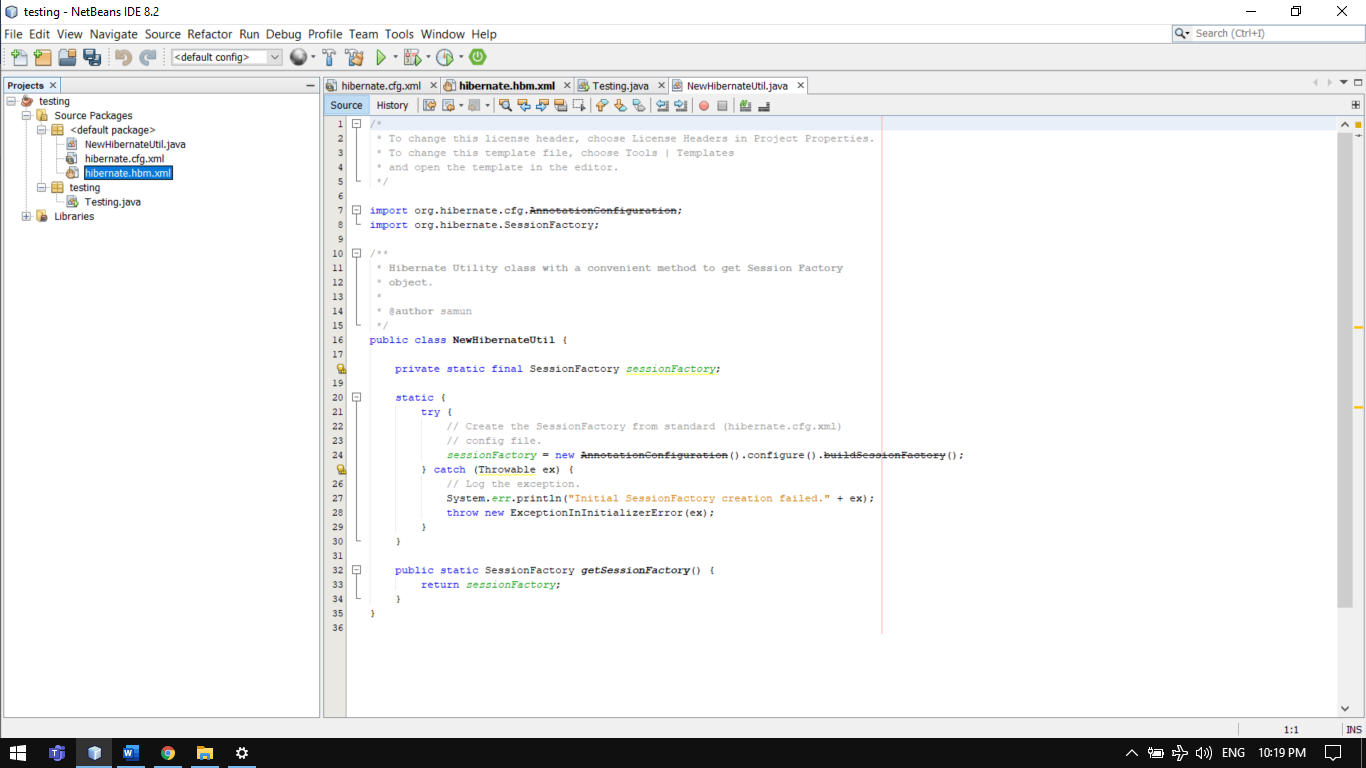


**Step 10: From here include the files one by one in project, the list of files are:**

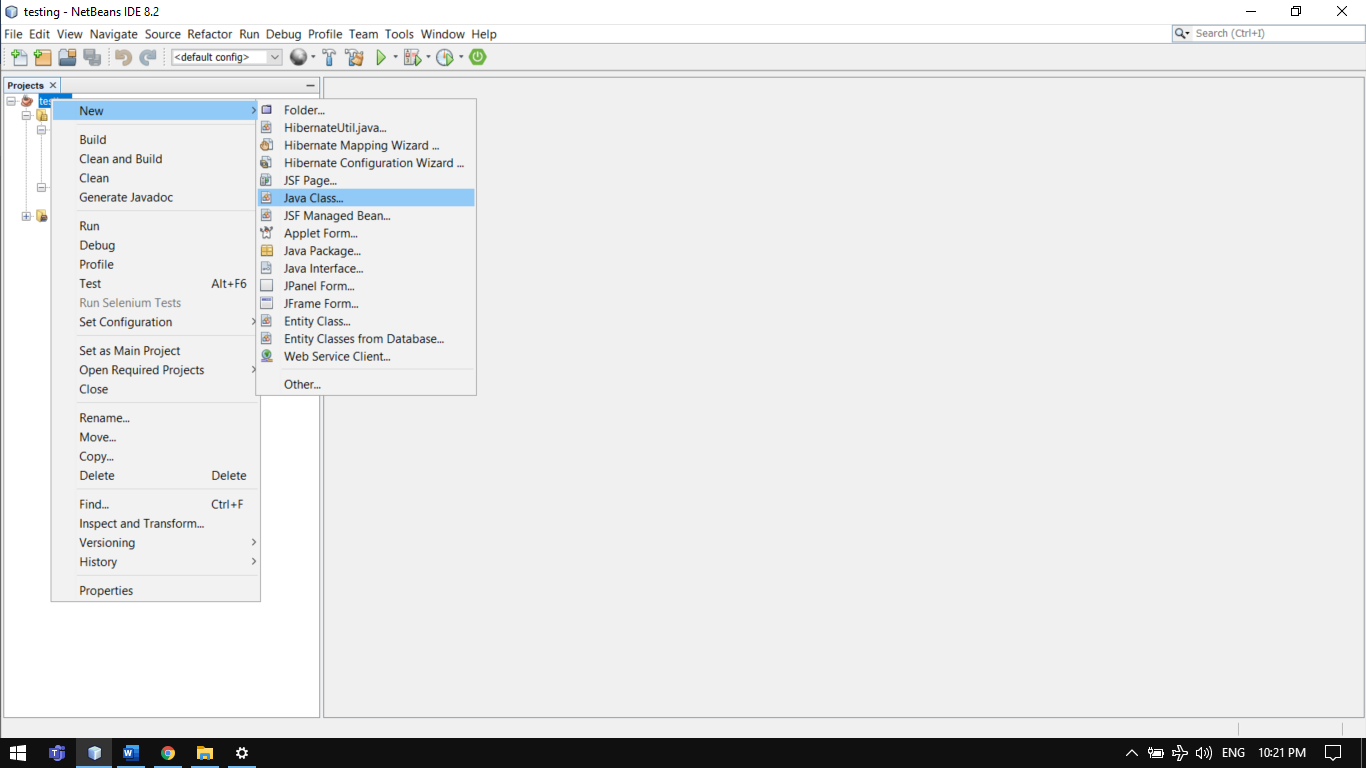
 



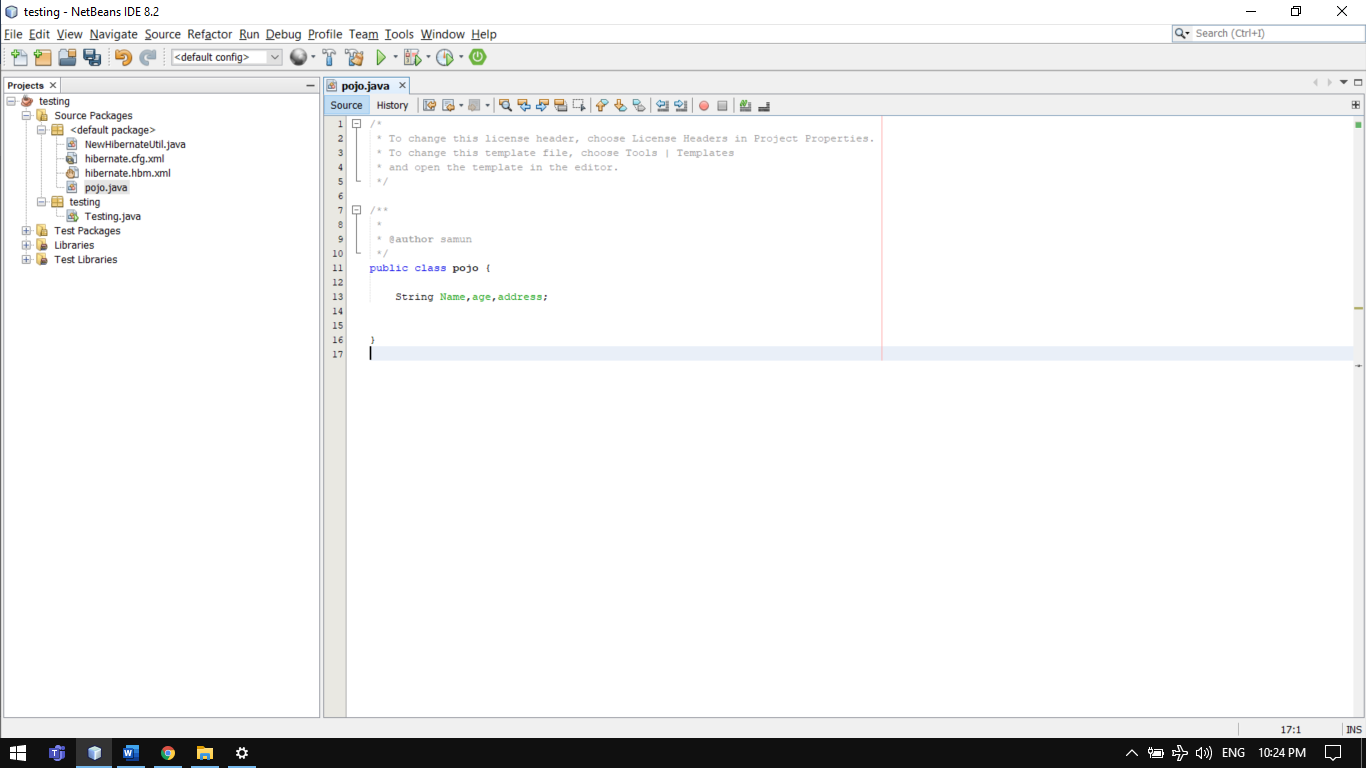
Go through all the setup Wizards and include them in project



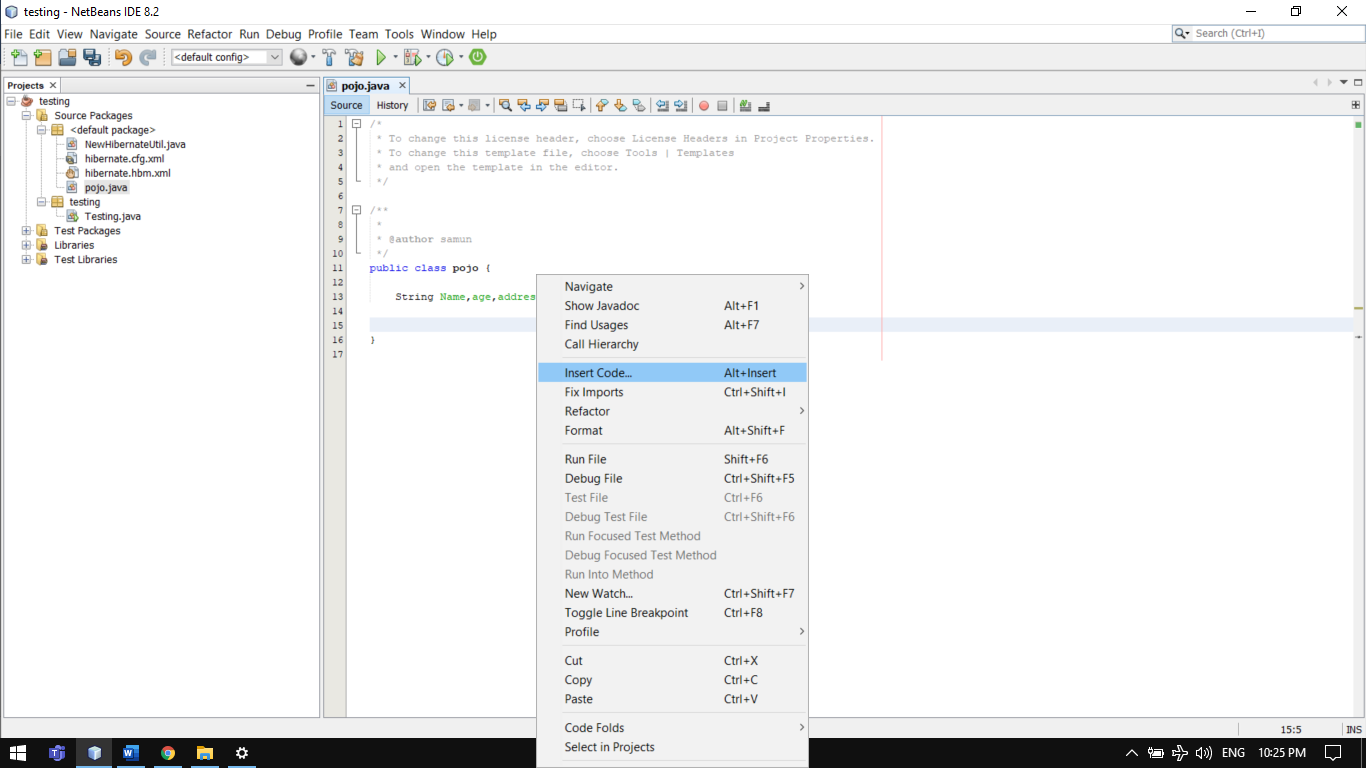
**Step 11: Here I am creating a POJO file manually**

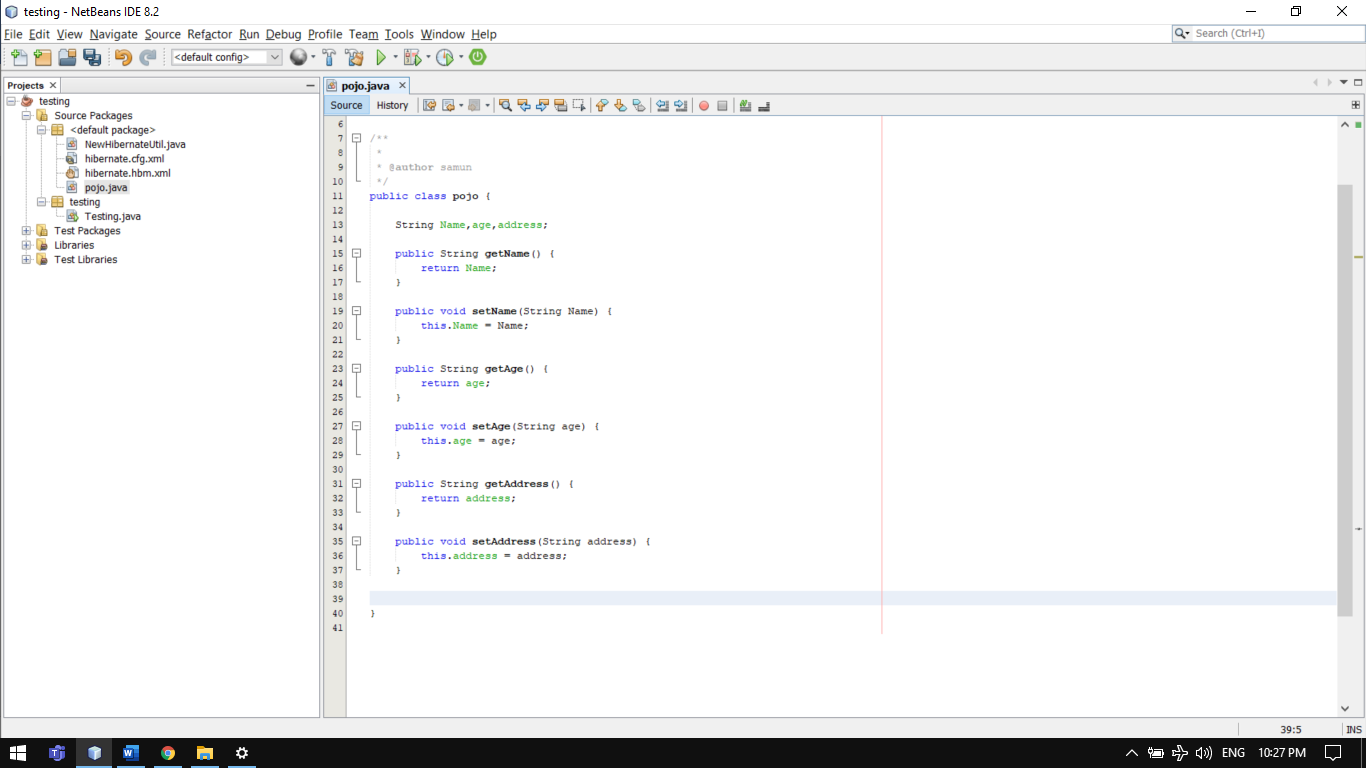


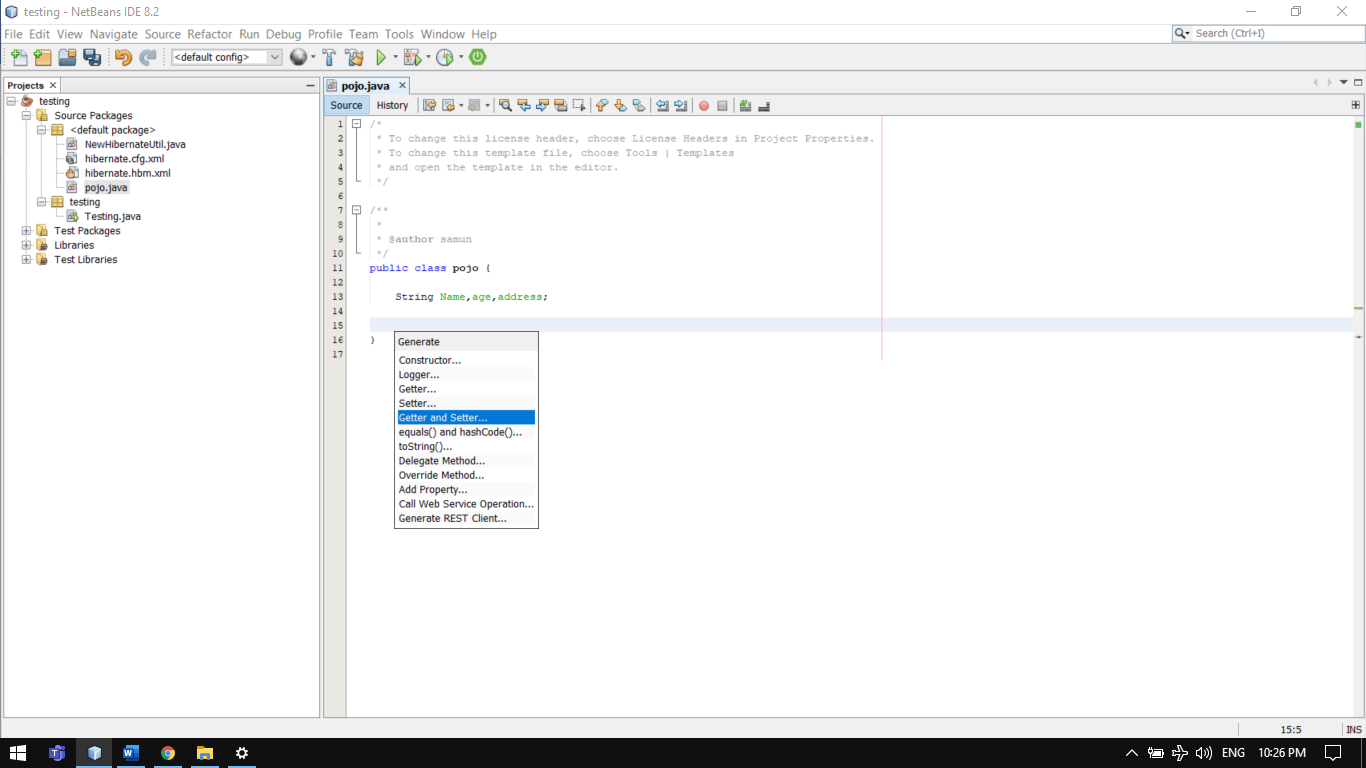
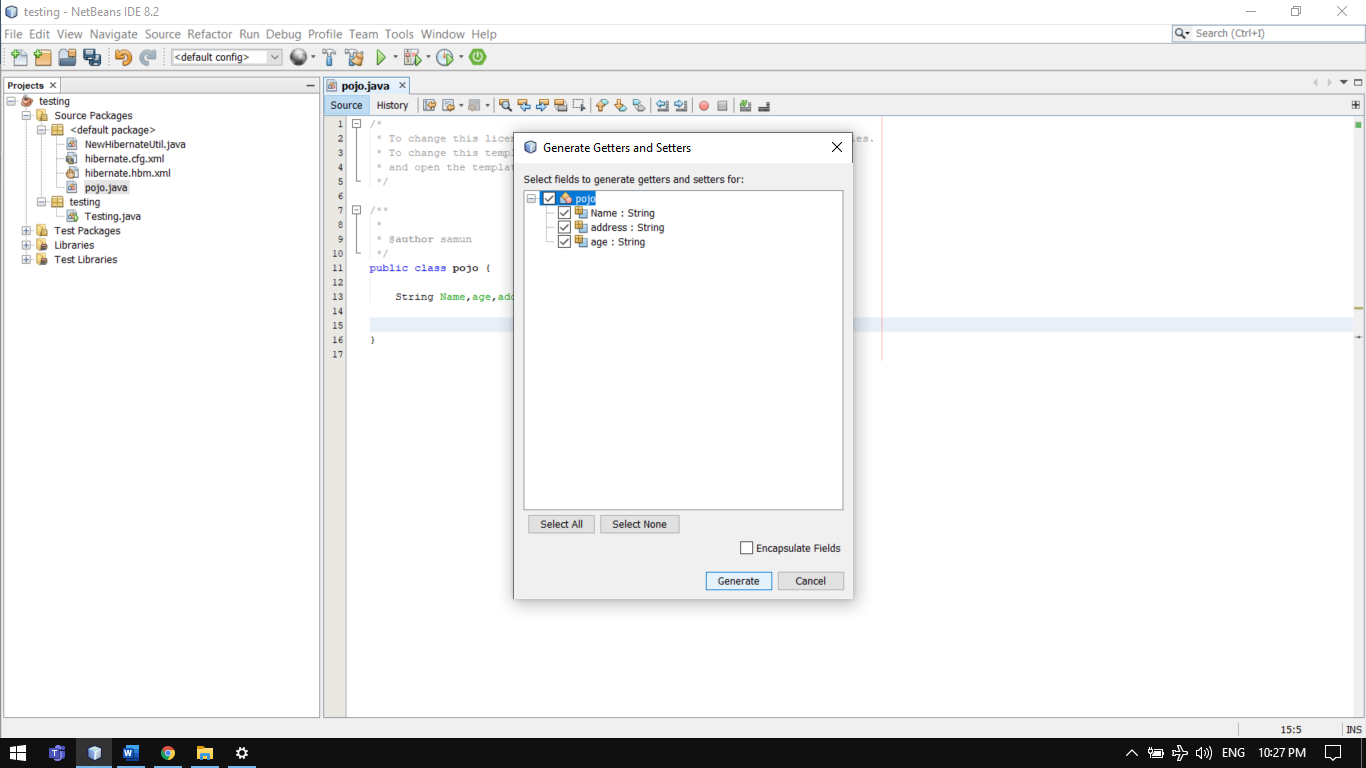
**Step 12: After creating file simply mentioned the data that you want to map with database**



**Step 12: After then simply right click choose insert code then getter and setter**





**Now your project is ready for the implementation All you need to do just writing some code and make some setting that is mentioned in ppt or in the given file .**

**Some important notes/terms:**

1. **What is hibernate?**

**Ans:** Hibernate is an open-source and lightweight ORM tool that is used to store, manipulate, and retrieve data from the database.

1. **What is ORM?**

**Ans: ORM** is an stands for Object/Relational mapping. It is a programming strategy to map object with the data stored in the database. It simplifies data creation, data manipulation, and data access.

1. **What are persistent classes?**

**Ans:** Classes whose objects are stored in a database table are called as persistent classes (POJO file).

1. **What is Session?**

**Ans:** It maintains a connection between the hibernate application and database. It provides methods to store, update, delete or fetch data from the database such as persist (), update (), delete (), load (), get () etc.

1. **What is HQL?**

**Ans:** HQL is the acronym of Hibernate Query Language. It is an Object-Oriented Query Language and is independent of the database.

1. **What do you mean by Hibernate Configuration File?**

**Ans:** Hibernate Configuration File mainly contains database-specific configurations and are used to initialize SessionFactory. Some important parts of the Hibernate Configuration File are Dialect information, so that hibernate knows the database type and mapping file or class details.

**Hibernate configuration Steps:**

1. First, identify the POJOs (Plain Old Java Objects) that have a database representation.
2. Identify which properties of POJOs need to be continued.
3. Annotate each of the POJOs in order to map the Java objects to columns in a database table.
4. Create a database schema using the schema export tool which uses an existing database, or you can create your own database schema.
5. Add Hibernate Java libraries to the application’s classpath.
6. Create a Hibernate *XML configuration file* that points to the database and the mapped classes.
7. In the Java application, you can create a Hibernate Configuration object that refers to your XML configuration file.
8. Also, build a Hibernate SessionFactory object from the Configuration object.
9. Retrieve the Hibernate Session objects from the SessionFactory and write down the data access logic for your application (create, retrieve, update, and delete).
10. **What do you mean by Hibernate mapping File?**

**Ans:** Hibernate mapping file is used to define the entity bean fields and database table column mappings. We know that JPA annotations can be used for mapping but sometimes XML mapping file comes handy when we are using third party classes and we can’t use annotations.

1. **What** **Elements of Hibernate Architecture?**

**Ans:**

SessionFactory

The SessionFactory is a factory of session and client of ConnectionProvider. It holds second level cache (optional) of data. The org.hibernate.SessionFactory interface provides factory method to get the object of Session.

Session

The session object provides an interface between the application and data stored in the database. It is a short-lived object and wraps the JDBC connection. It is factory of Transaction, Query and Criteria. It holds a first-level cache (mandatory) of data. The org.hibernate.Session interface provides methods to insert, update and delete the object. It also provides factory methods for Transaction, Query and Criteria.

Transaction

The transaction object specifies the atomic unit of work. It is optional. The org.hibernate.Transaction interface provides methods for transaction management.

ConnectionProvider

It is a factory of JDBC connections. It abstracts the application from DriverManager or DataSource. It is optional.

TransactionFactory

It is a factory of Transaction. It is optional.

1. **What** **Difference between first level and second level cache in Hibernate?**

V.V.I

**Ans:** The main difference between first level and second level cache in Hibernate is that the first level is maintained at the Session level and accessible only to the Session, while the second level cache is maintained at the SessionFactory level and available to all Sessions. This means, you can use the first level cache to store local data, i.e. the data which is needed by the Session, and you can use the second-level cache to store global data, i.e. something which can be shared across sessions.

**10.What is First Level Cache in Hibernate?**

As the name suggests, the first-level cache is the first cache hibernate consults before loading an object from the database. It is maintained at Session level, and it's by default enabled.

We know that Session is the interface between Hibernate and Database. We load objects using Session, e.g. by calling get() or load() method or by executing queries.

Example:

When we ask load method to return a Person (an object stored in database) with Id=1, first time it's loaded from database, that too lazily, when any method of that object is called other than getId(), next time if you load the same object then Hibernate doesn't go to [Database](https://dev.to/javinpaul/5-online-courses-to-learn-sql-and-database-for-beginners-best-of-lot-5533), instead it **returns the object from first level cache maintained at Session level**.

The data in the first level cache is maintained as long as Session is open, as soon as you close the Session, all data is lost. Next time even if you load the same object, e.g. Person with the same id, Hibernate will again go to Database to load that object, provided Second-level Cache is disabled

**10.What is Second Level Cache in Hibernate?**

**Ans:** Unlike first level cache which is accessible only to the session who maintains it, Second-level Cache is accessible to all Sessions.

This means if one Session loads an object, like Person with id=1 and Second session also loads the same object, only one database call will be made. Next session will get the data from the [Second-level Cache](http://javarevisited.blogspot.sg/2017/03/difference-between-first-and-second-level-cache-in-Hibernate.html).

Means no database call each time.

This also helps in interacting with different databases.