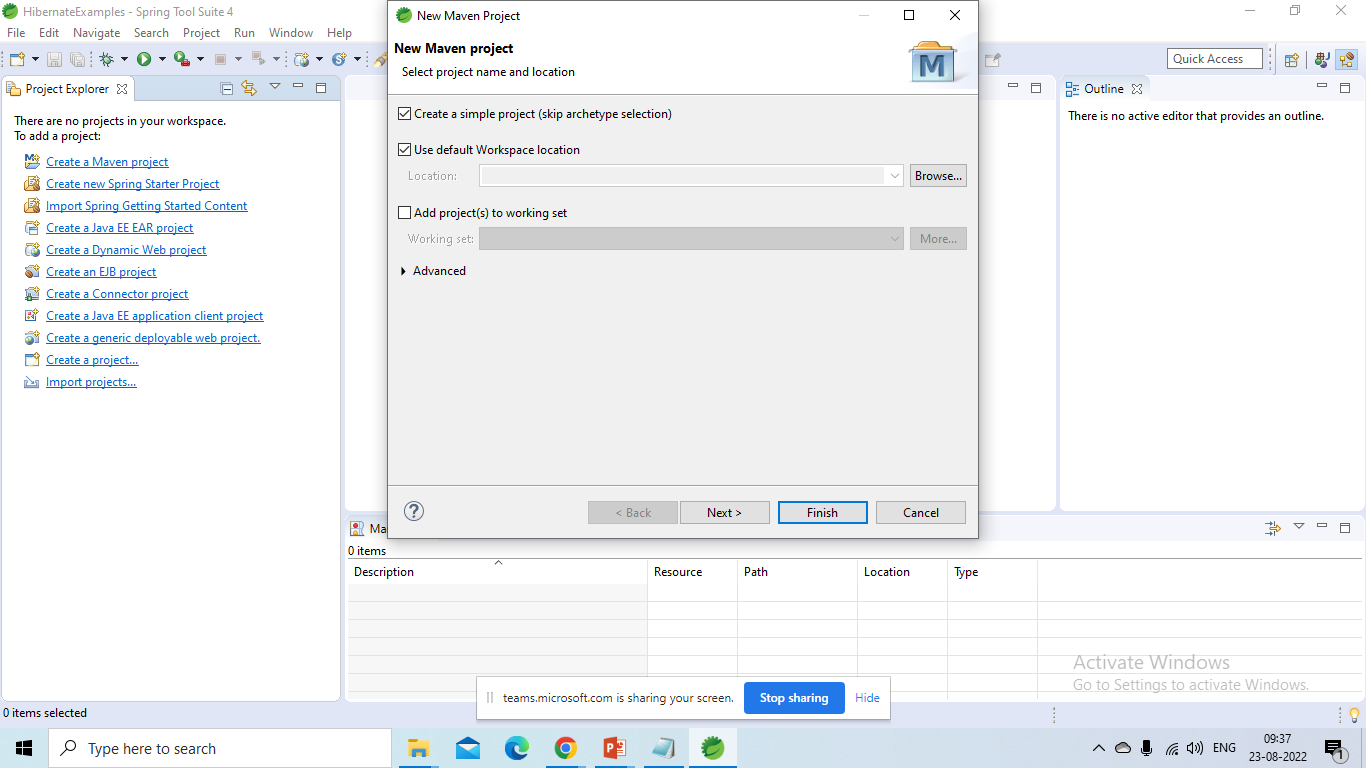
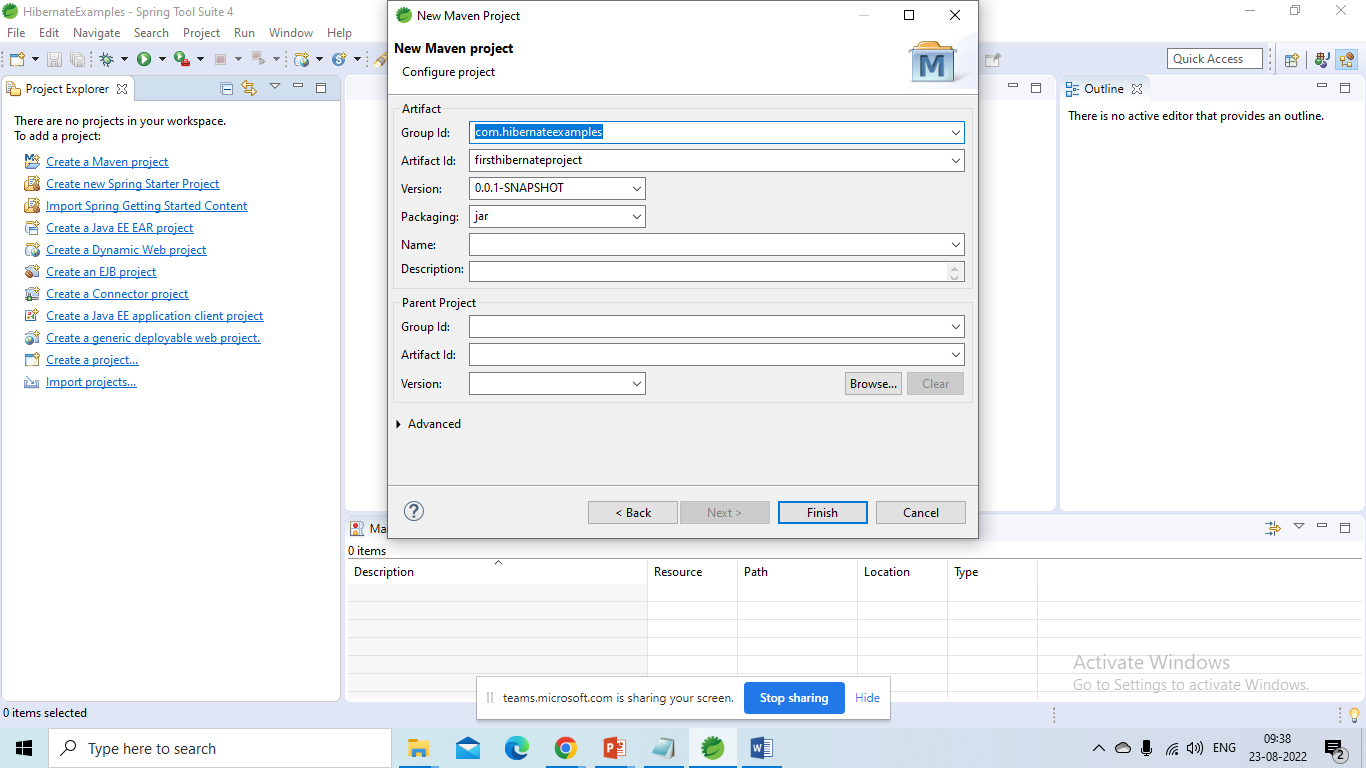
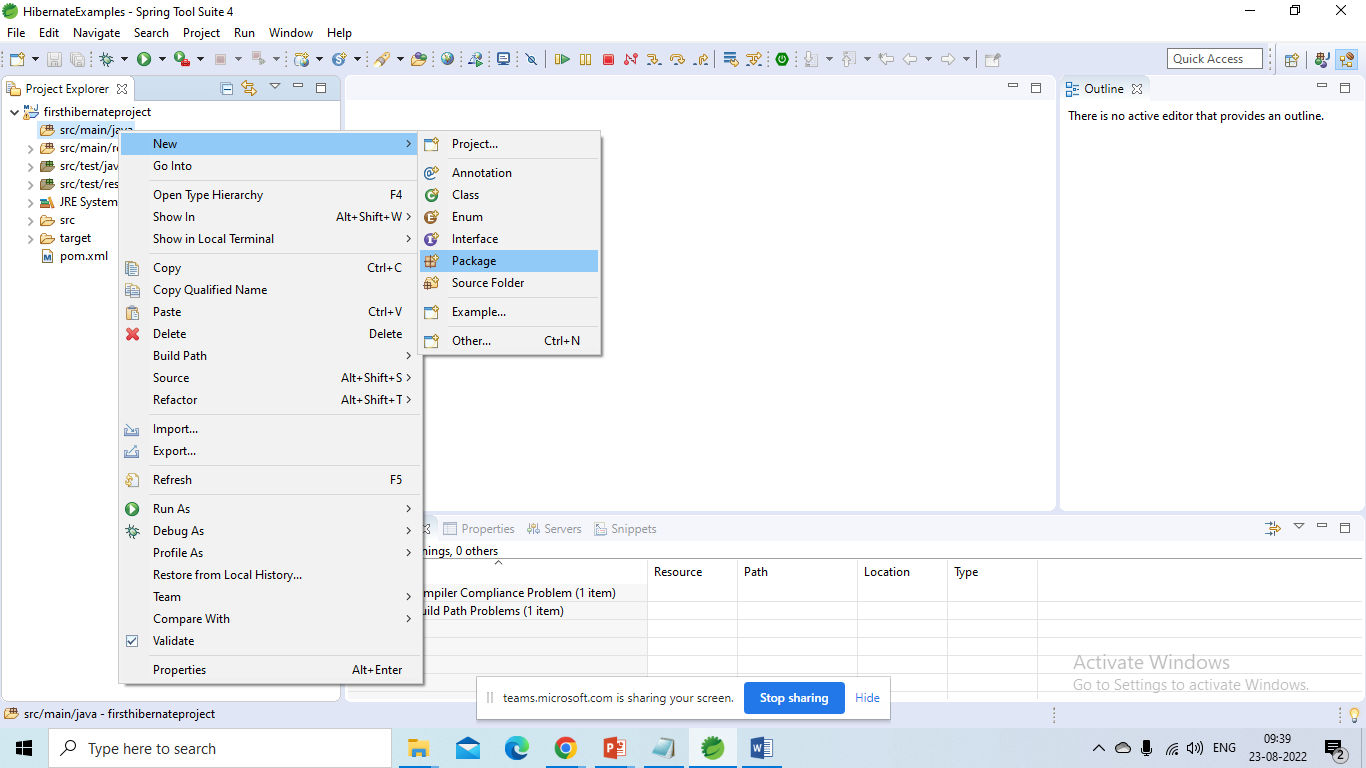
Step 1: create a simple maven project

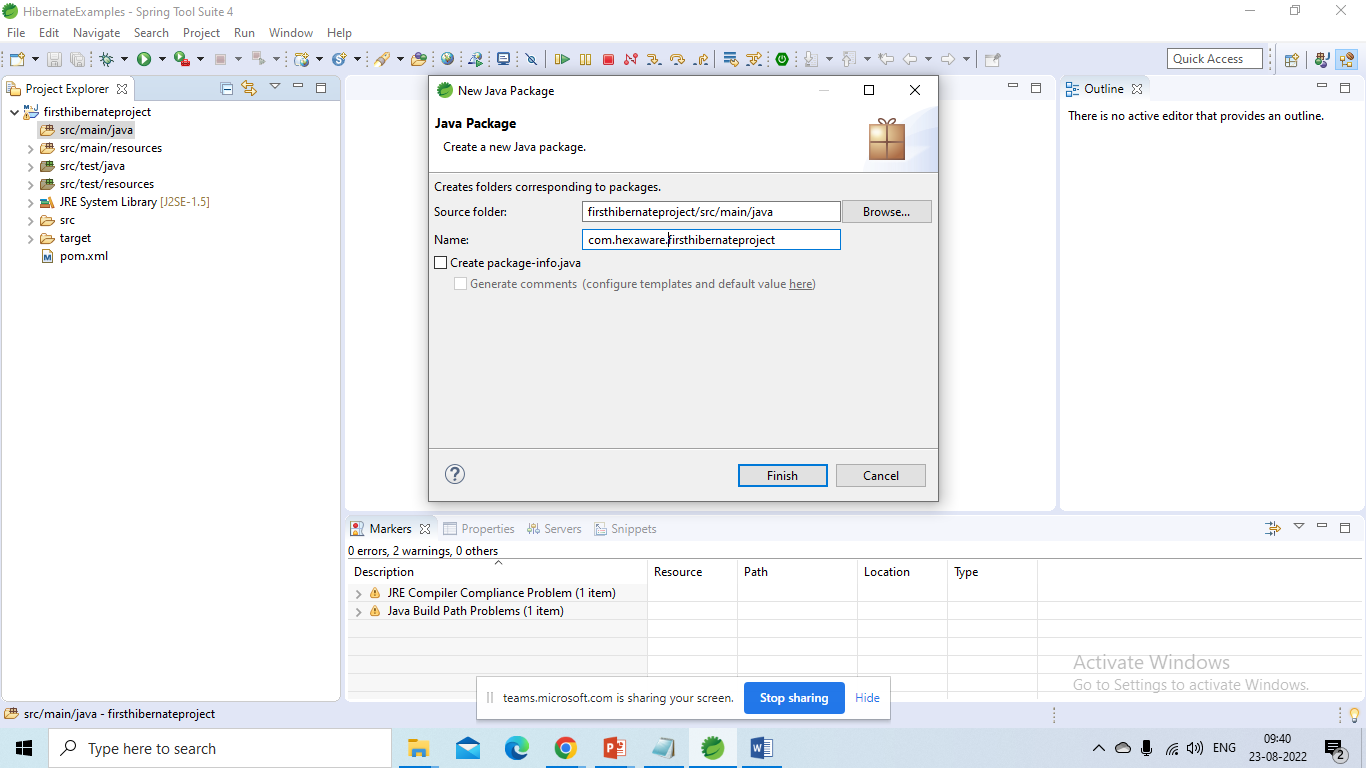




Step 2:

Create package structure for project



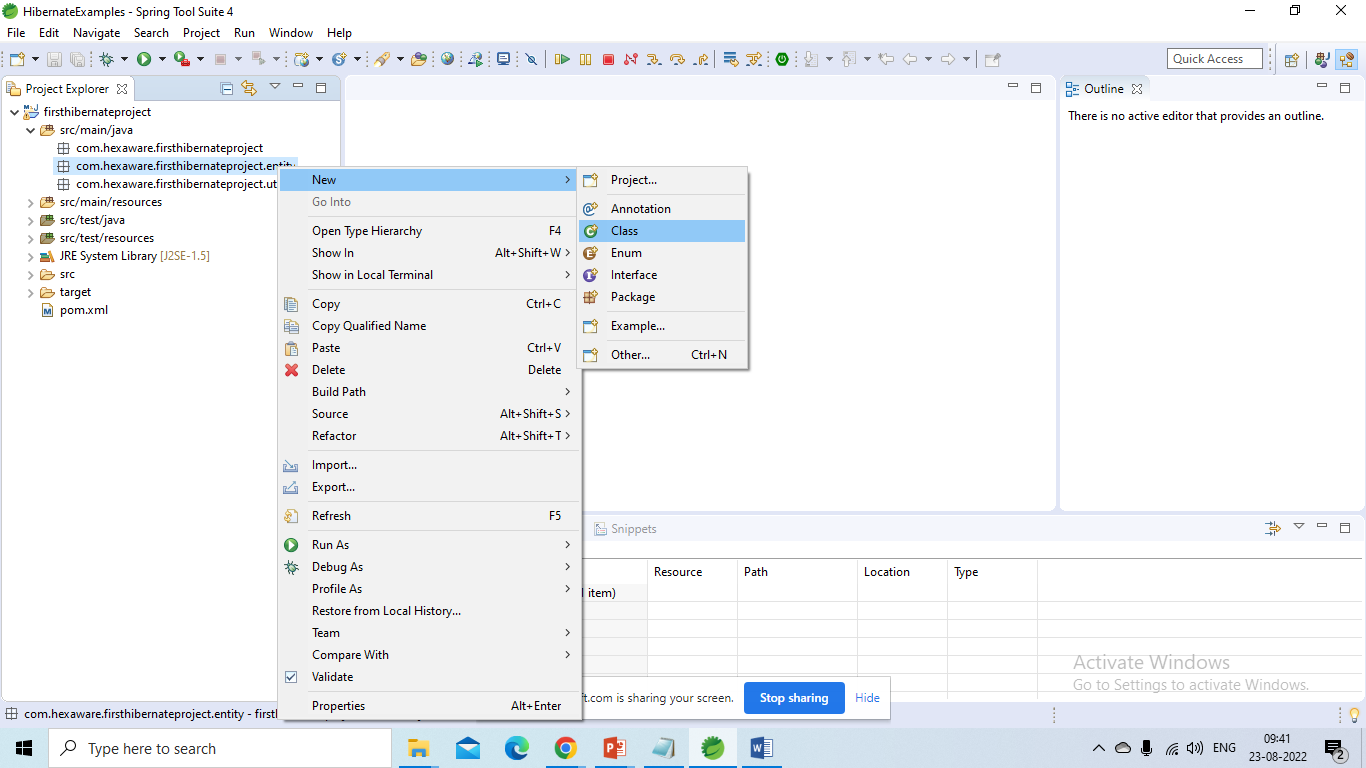


Step 3: update pom.xml with required dependencies to support hibernate

Hibernate-core

Mysql-connector-java

Step 4: Create an entity class



**package** com.hexaware.firsthibernateproject.entity;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

/\*

\* Specifies that the class is an entity.

\* This annotation is applied to the entity class

\*/

@Entity

@Table(name = "student\_info")

**public** **class** Student {

**public** Student() {

// default constructor

}

/\*

\* Specifies the primary key of an entity.

\* The field or property to which the Id annotation is applied

\* should be one of the following types: any Java primitive type;

\* any primitive wrapper type;

\* String;

\* java.util.Date;

\* java.sql.Date;

\* java.math.BigDecimal;

\* java.math.BigInteger.

The mapped column for the primary key of the entity is assumed

to be the primary key of the primary table.

If no Column annotationis specified,

the primary key column name is assumed to be the name

of the primary key property or field.

\*/

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name = "studentid")

**private** Long studentId;

@Column(name = "studentfirstname")

**private** String studentFirstName;

@Column(name = "studentlastname")

**private** String studentLastName;

@Column(name = "studentemail")

**private** String studentEmail;

**public** Student(String studentFirstName, String studentLastName, String studentEmail) {

**super**();

**this**.studentFirstName = studentFirstName;

**this**.studentLastName = studentLastName;

**this**.studentEmail = studentEmail;

}

**public** Long getStudentId() {

**return** studentId;

}

**public** **void** setStudentId(Long studentId) {

**this**.studentId = studentId;

}

**public** String getStudentFirstName() {

**return** studentFirstName;

}

**public** **void** setStudentFirstName(String studentFirstName) {

**this**.studentFirstName = studentFirstName;

}

**public** String getStudentLastName() {

**return** studentLastName;

}

**public** **void** setStudentLastName(String studentLastName) {

**this**.studentLastName = studentLastName;

}

**public** String getStudentEmail() {

**return** studentEmail;

}

**public** **void** setStudentEmail(String studentEmail) {

**this**.studentEmail = studentEmail;

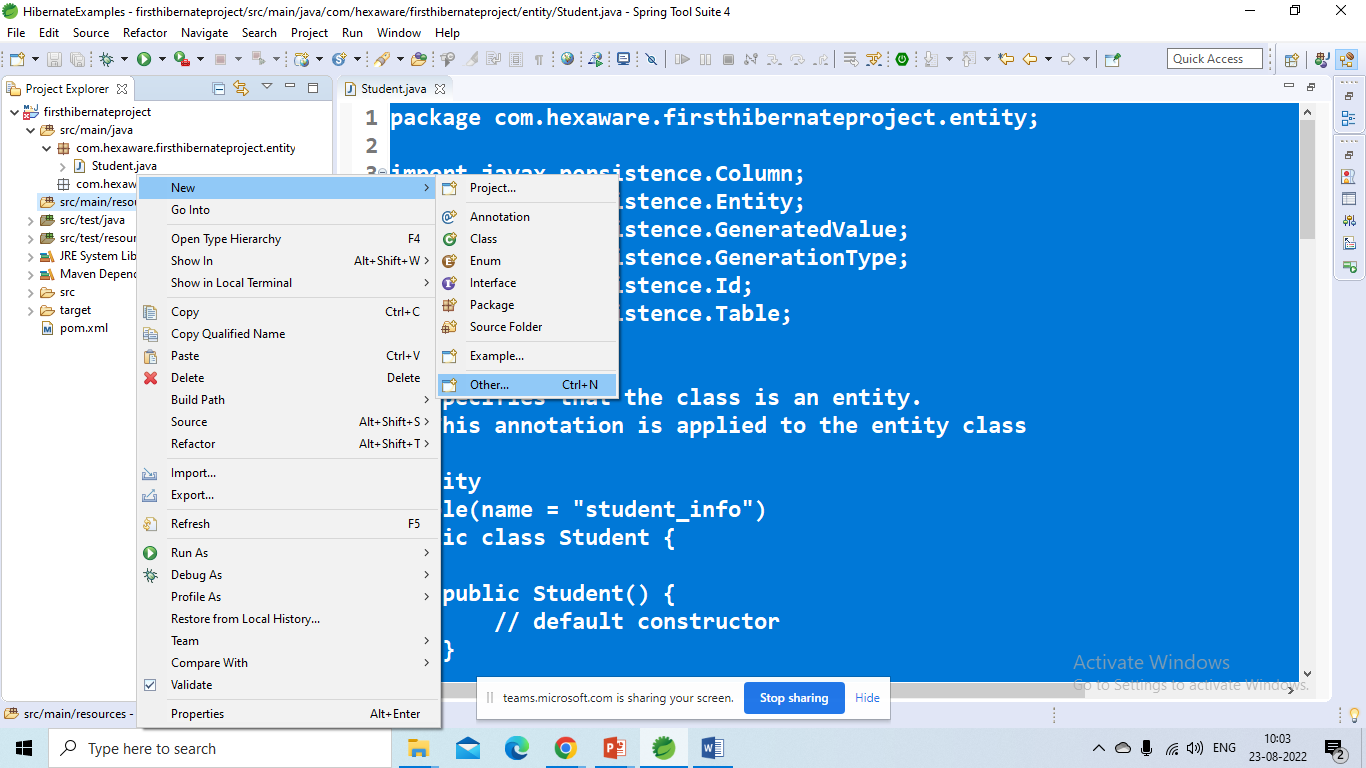
}

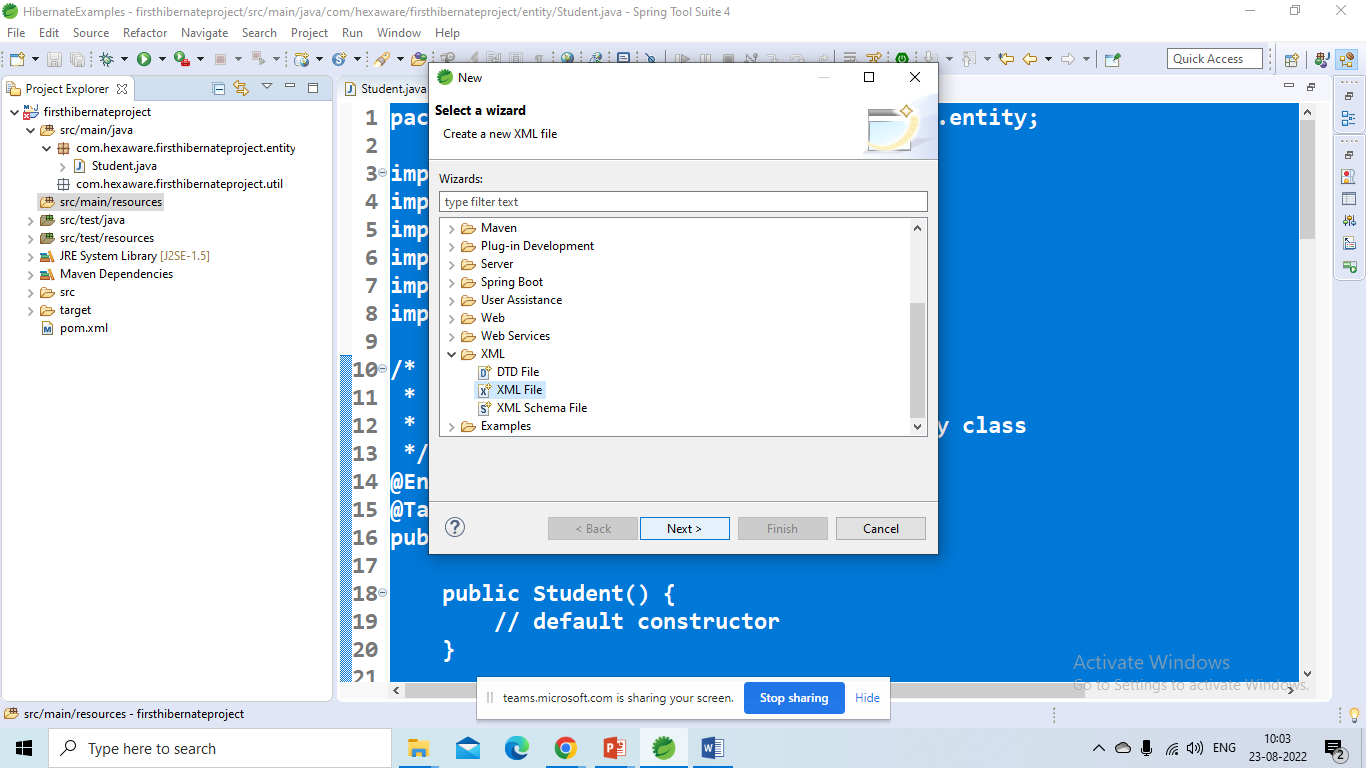
}

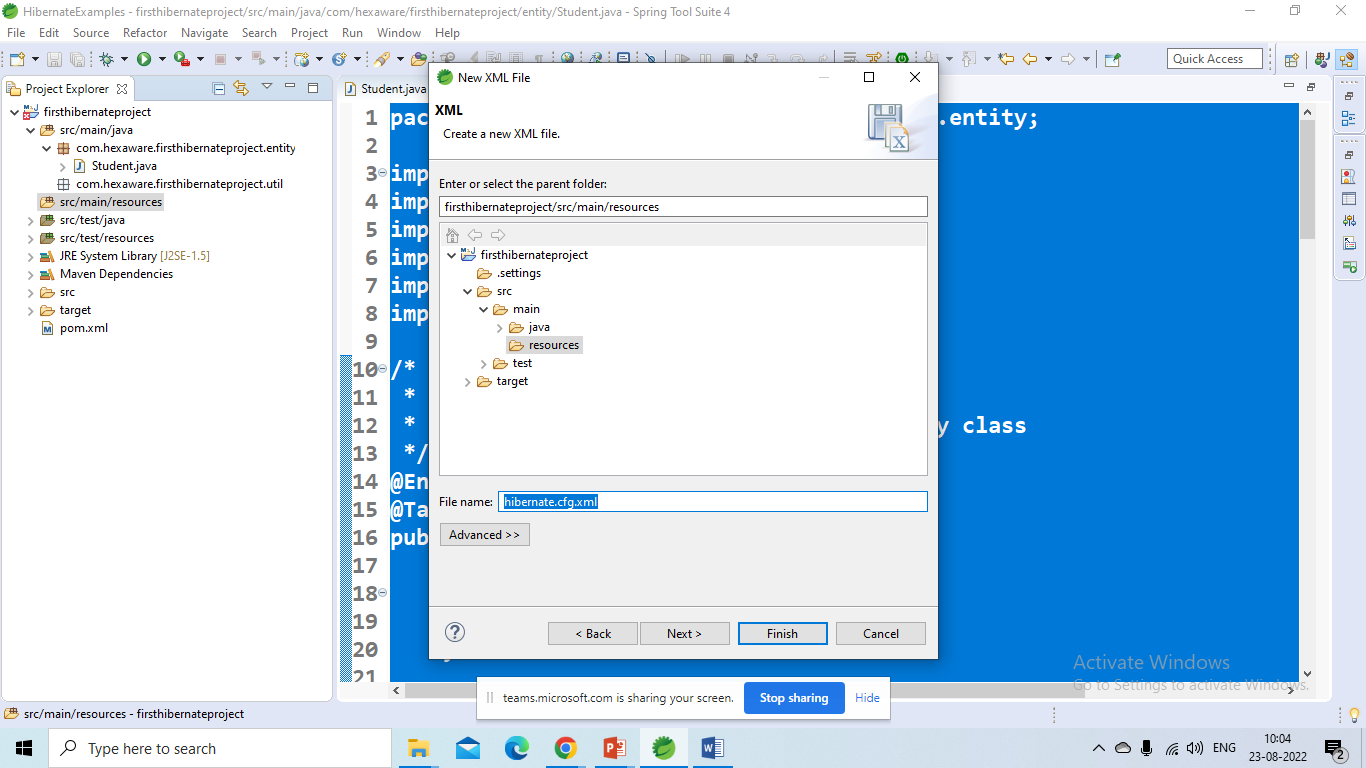
Step 5:

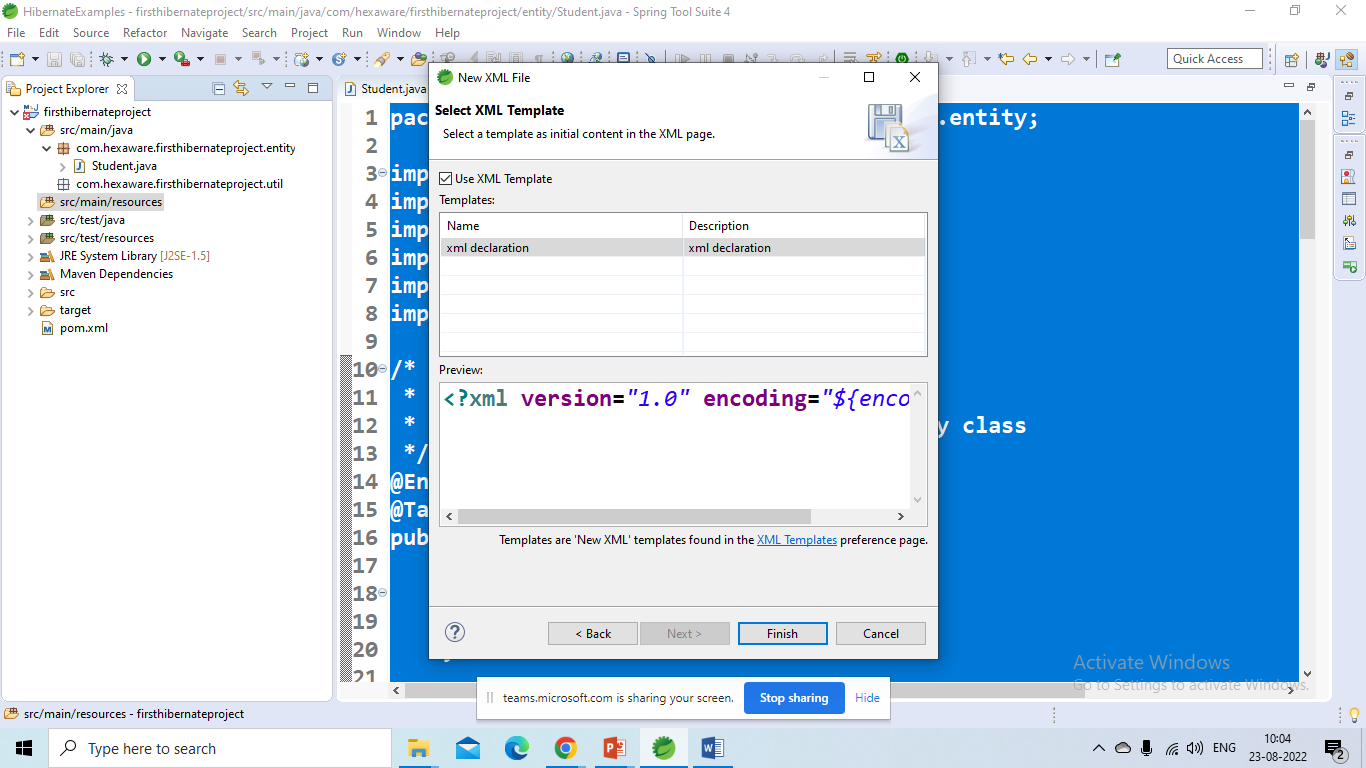
Create hibernate.cfg.xml file for mapping database and entity classes

resources/ hibernate.cfg.xml









<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- JDBC Database connection settings -->

<property name=*"connection.driver\_class"*>com.mysql.cj.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/sample?useSSL=false</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>root</property>

<!-- JDBC connection pool settings ... using built-in test pool -->

<property name=*"connection.pool\_size"*>1</property>

<!-- Select our SQL dialect -->

<property name=*"dialect"*>org.hibernate.dialect.MySQL5Dialect</property>

<!-- Echo the SQL to stdout -->

<property name=*"show\_sql"*>true</property>

<!-- Set the current session context -->

<property name=*"current\_session\_context\_class"*>thread</property>

<!-- create the database schema on startup -->

<!-- create,create-drop,update,none,validate -->

<property name=*"hbm2ddl.auto"*>create</property>

<!-- dbcp connection pool configuration -->

<property name=*"hibernate.dbcp.initialSize"*>5</property>

<property name=*"hibernate.dbcp.maxTotal"*>20</property>

<property name=*"hibernate.dbcp.maxIdle"*>10</property>

<property name=*"hibernate.dbcp.minIdle"*>5</property>

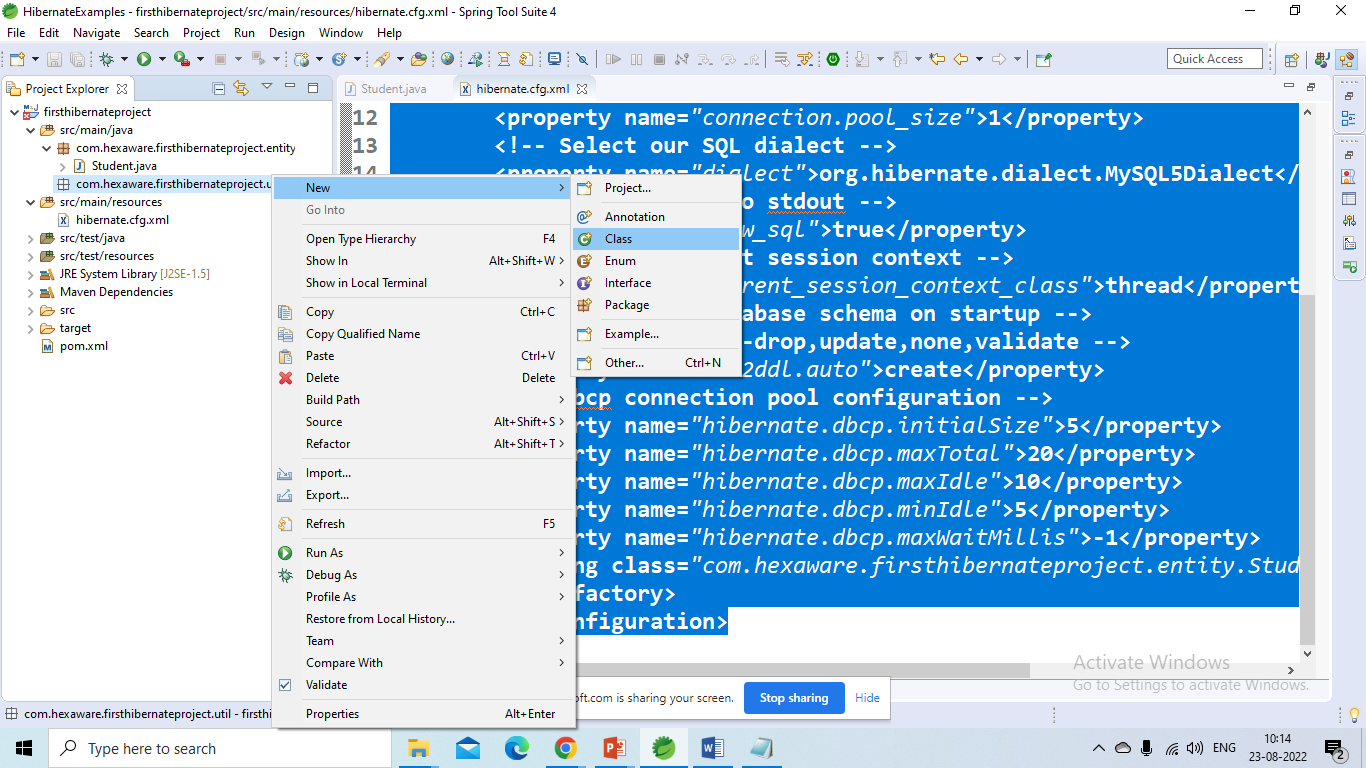
<property name=*"hibernate.dbcp.maxWaitMillis"*>-1</property>

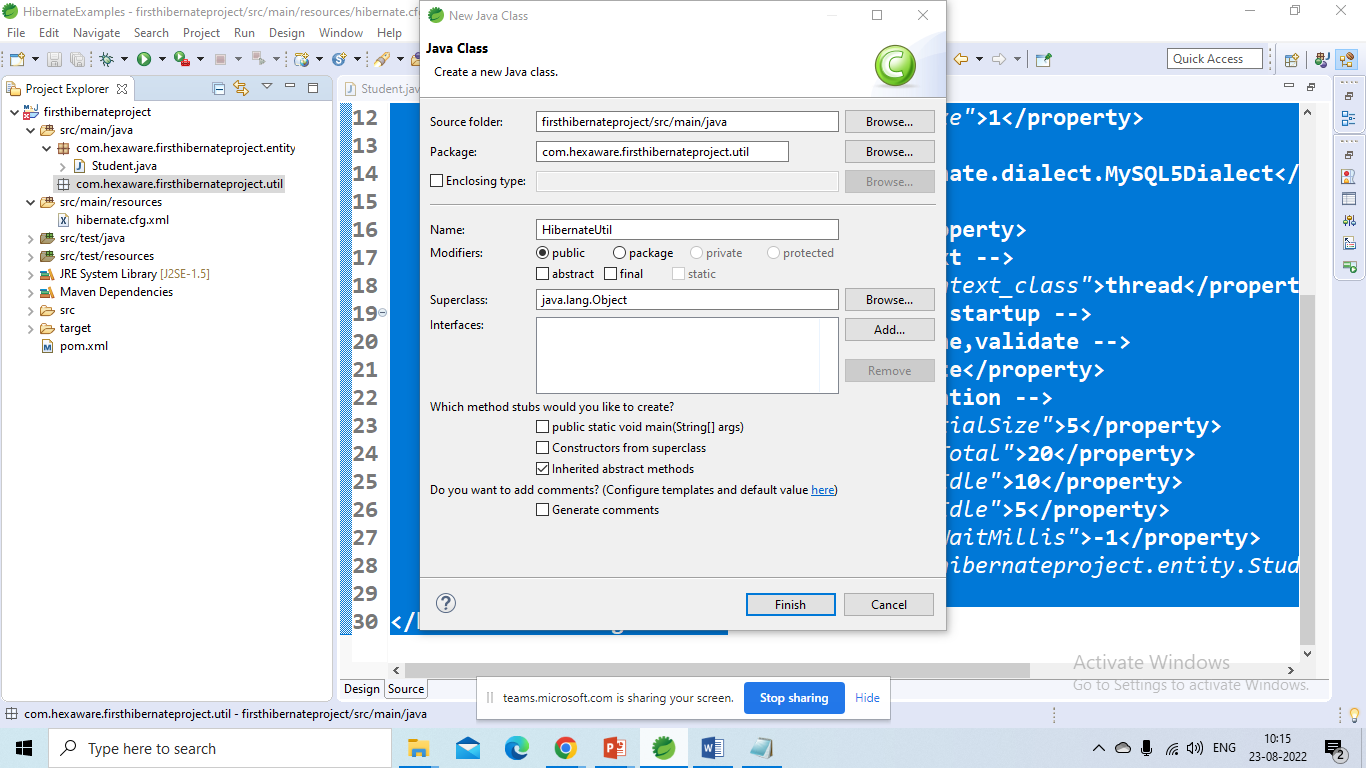
<mapping class=*"com.hexaware.firsthibernateproject.entity.Student"* />

</session-factory>

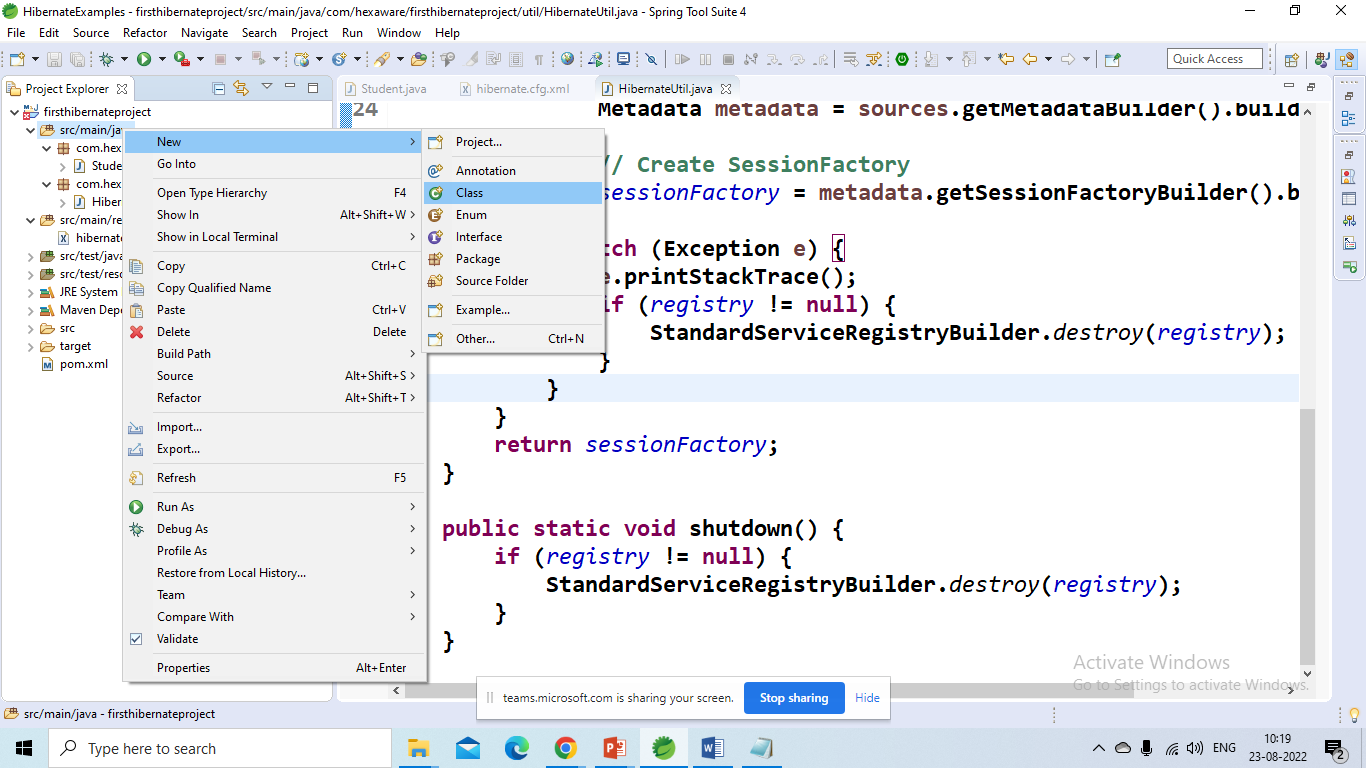
</hibernate-configuration>

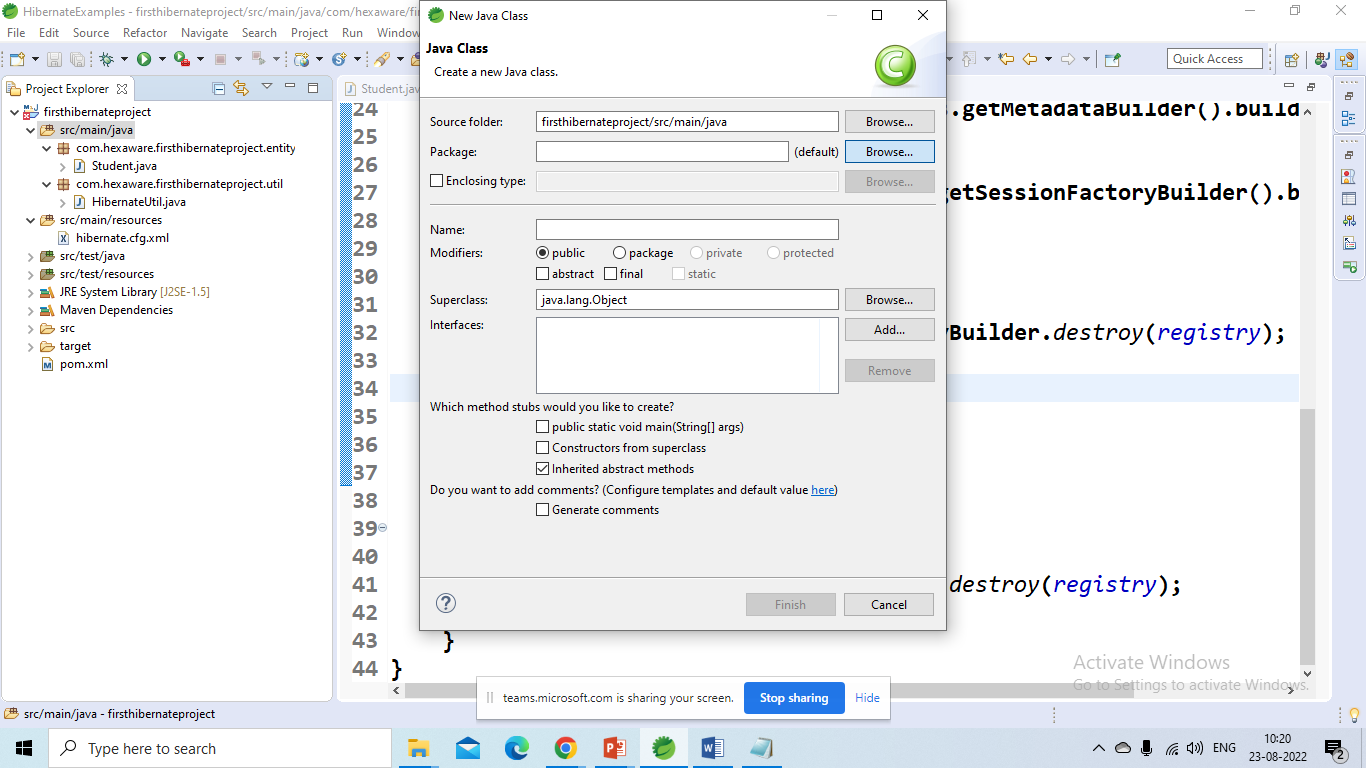
Step 6: create a hibernate util class

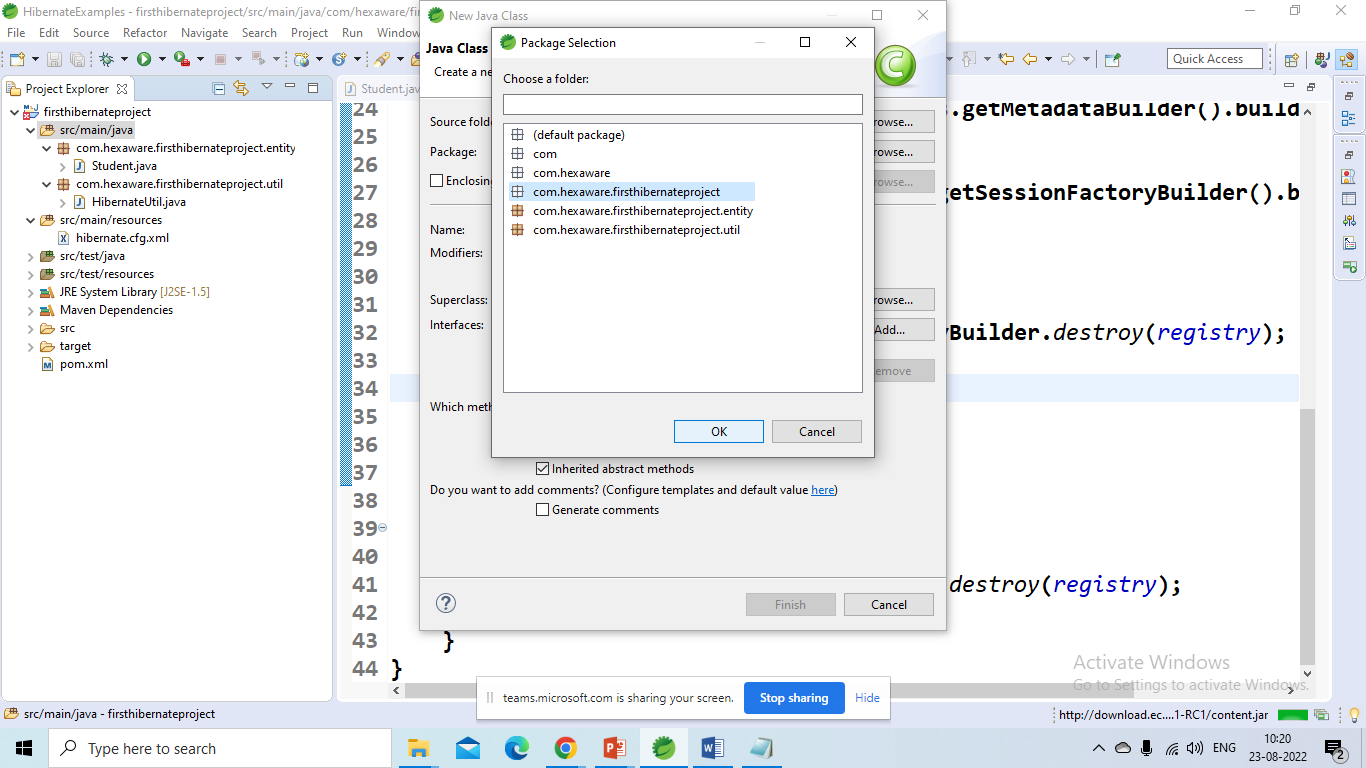


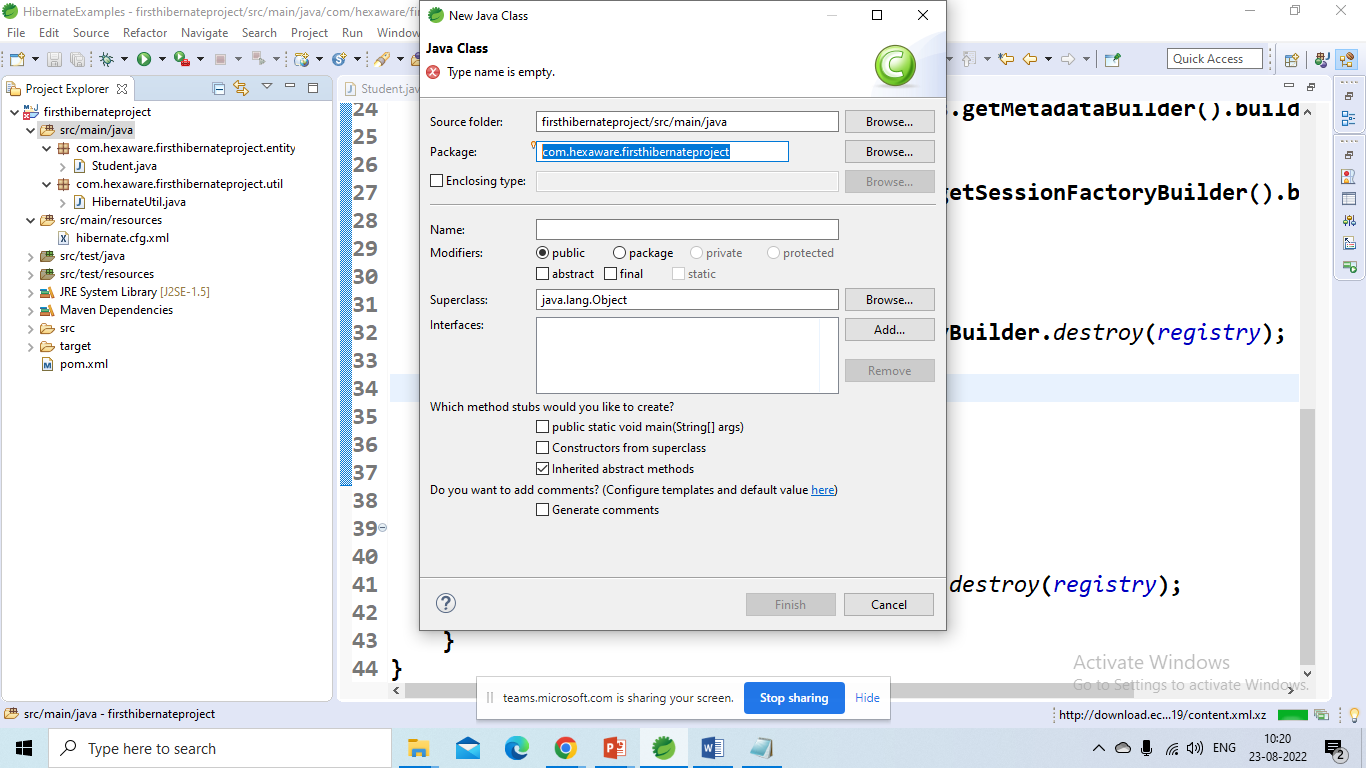


Step 7: Create a Main class for the above application









Name class as “MainAPP”

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.Transaction;

**import** com.hexaware.firsthibernateproject.entity.Student;

**import** com.hexaware.firsthibernateproject.util.HibernateUtil;

**public** **class** MainApp {

**public** **static** **void** main(String[] args) {

Student st = **new** Student("Rama","Krishna","rk@gmail.com");

Student st1 = **new** Student("Raja","Ram","rr@gmail.com");

Transaction transaction = **null**;

Session session = **null**;

**try** {

session = HibernateUtil.*getSessionFactory*().openSession();

//start a transaction

transaction = session.beginTransaction();

//save the student objects to db

session.save(st1);

session.save(st);

//commit the transaction

transaction.commit();

}**catch** (Exception e) {

**if**(transaction != **null**) {

transaction.rollback();

}

e.printStackTrace();

}**finally** {

**if**(session != **null**) {

session.close();

}

}

**try** {

session = HibernateUtil.*getSessionFactory*().openSession();

List<Student> students = session.createQuery("from Student", Student.**class**).list();

students.forEach(s -> System.***out***.println(s.getStudentFirstName()));

}**catch** (Exception e) {

**if**(transaction != **null**) {

transaction.rollback();

}

e.printStackTrace();

}**finally** {

**if**(session != **null**) {

session.close();

}

}

}

}