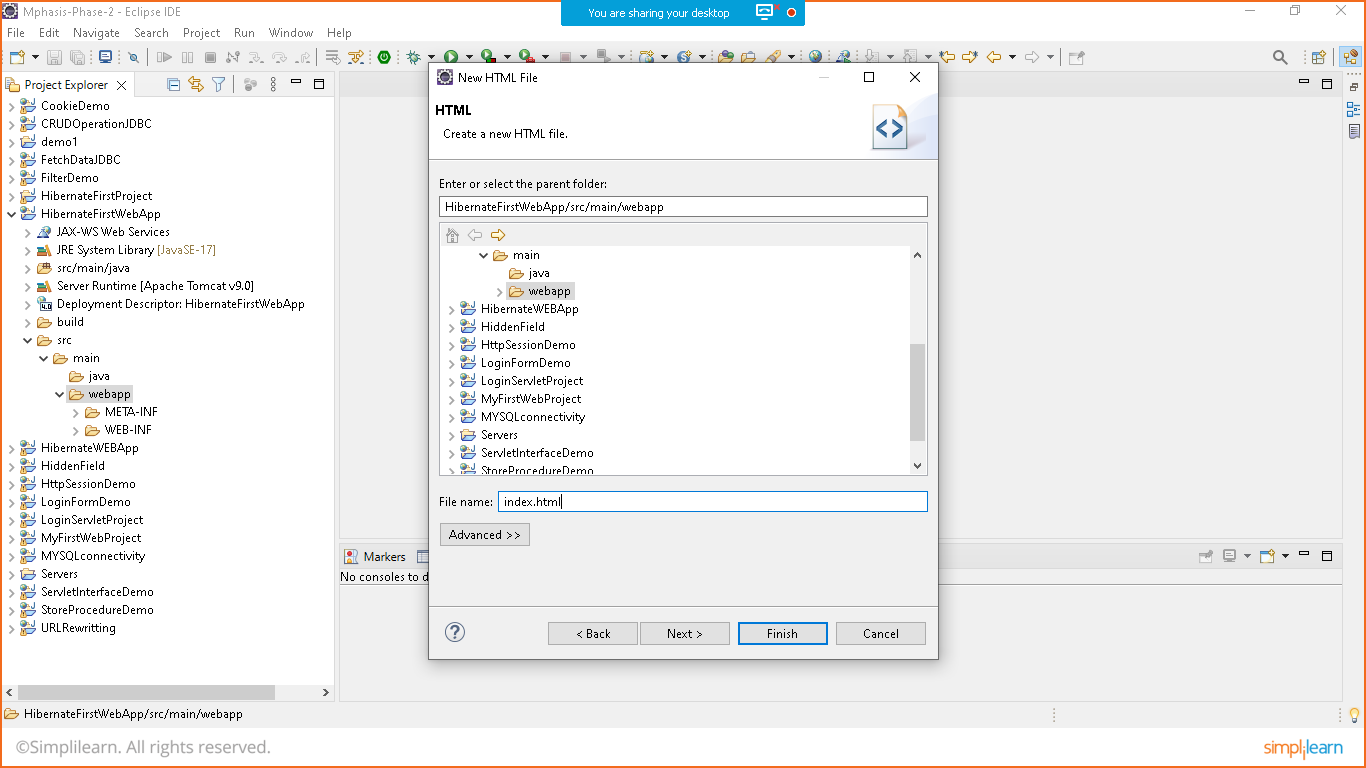
Dynamic Web App-Hibernate Project

Step:1 Prepare Dynamic web project

STEP:2 Create index.html file



Index.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<H1> WELCOME TO HIBERNATE WEB APP</H1>

<br>

<br>

<a href=*"init"*>Test Connection</a>

<br>

<br>

<a href=*"save"*>Save Data</a>

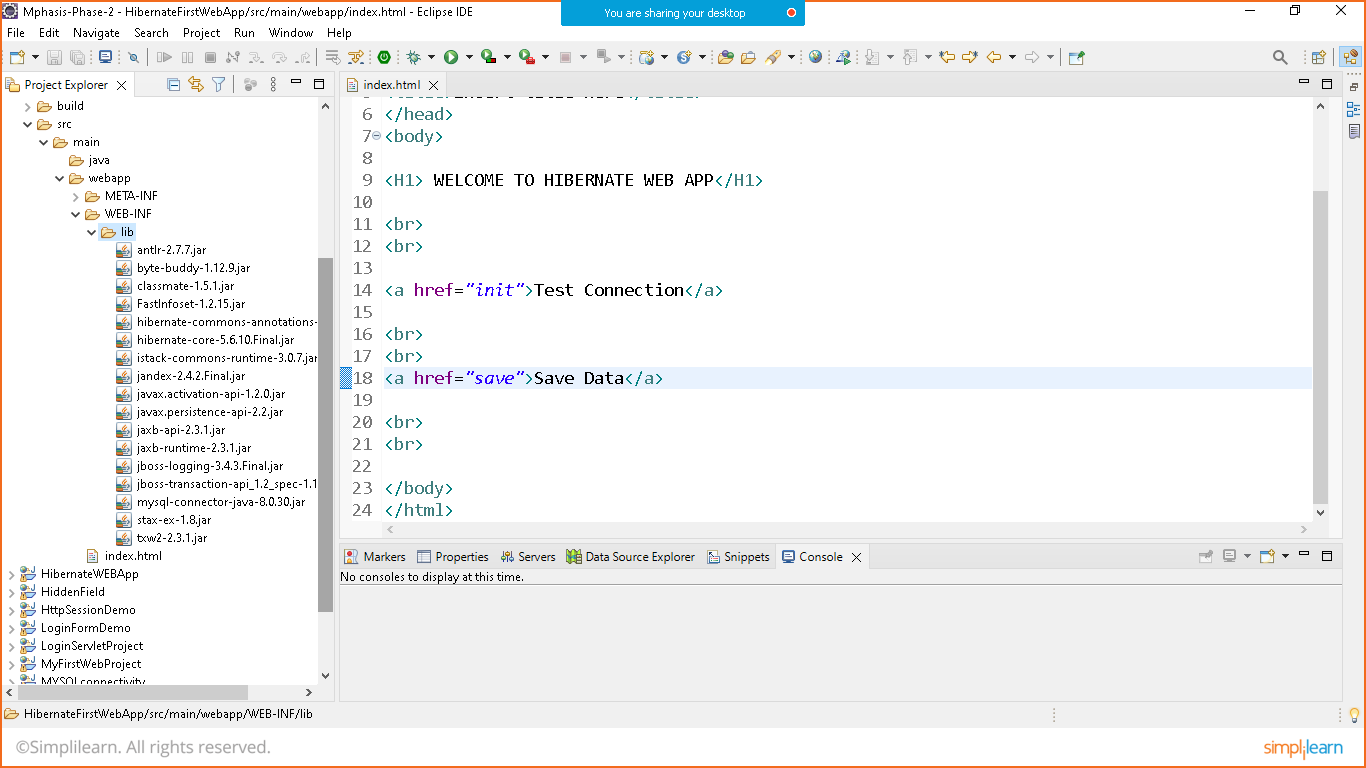
<br>

<br>

</body>

</html>

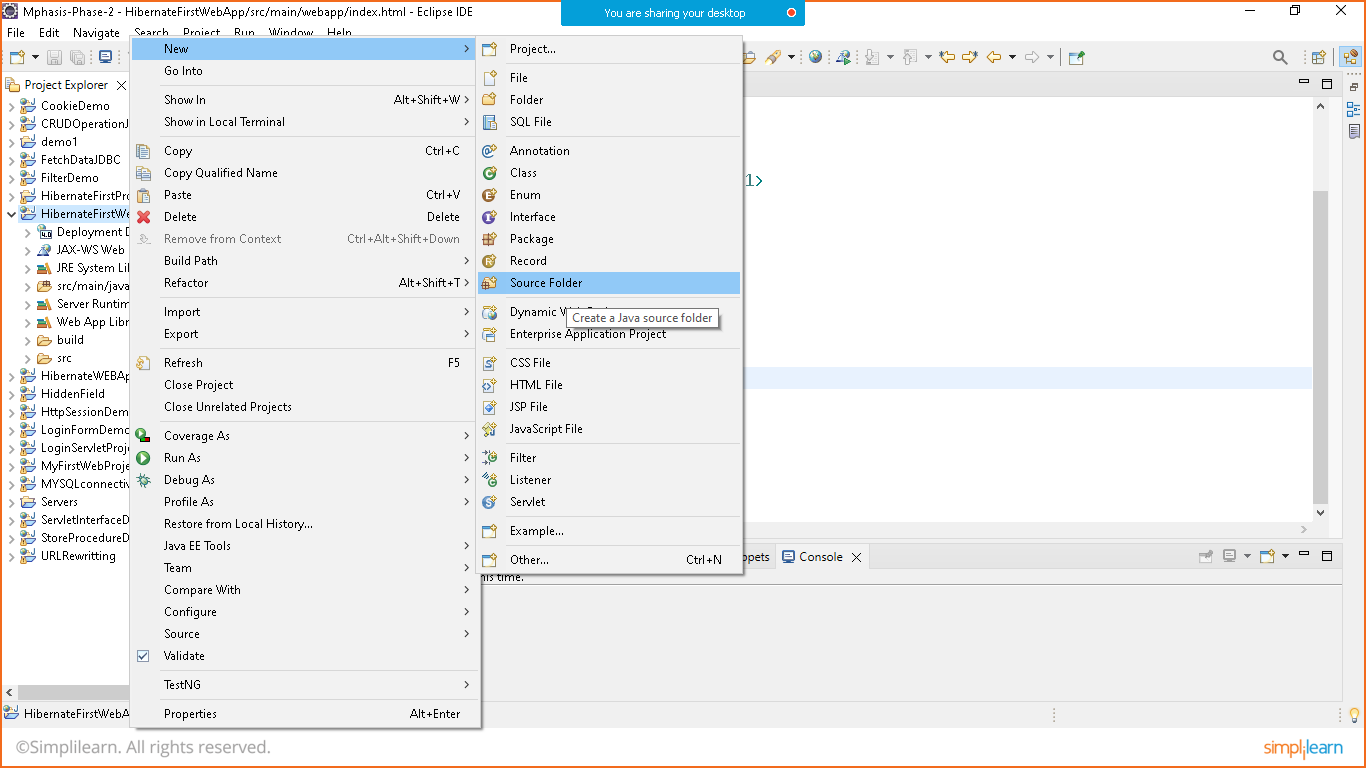
STEP:3 Copy all jar files of mysql-connector and hibernate to lib-folder in WEB-INF

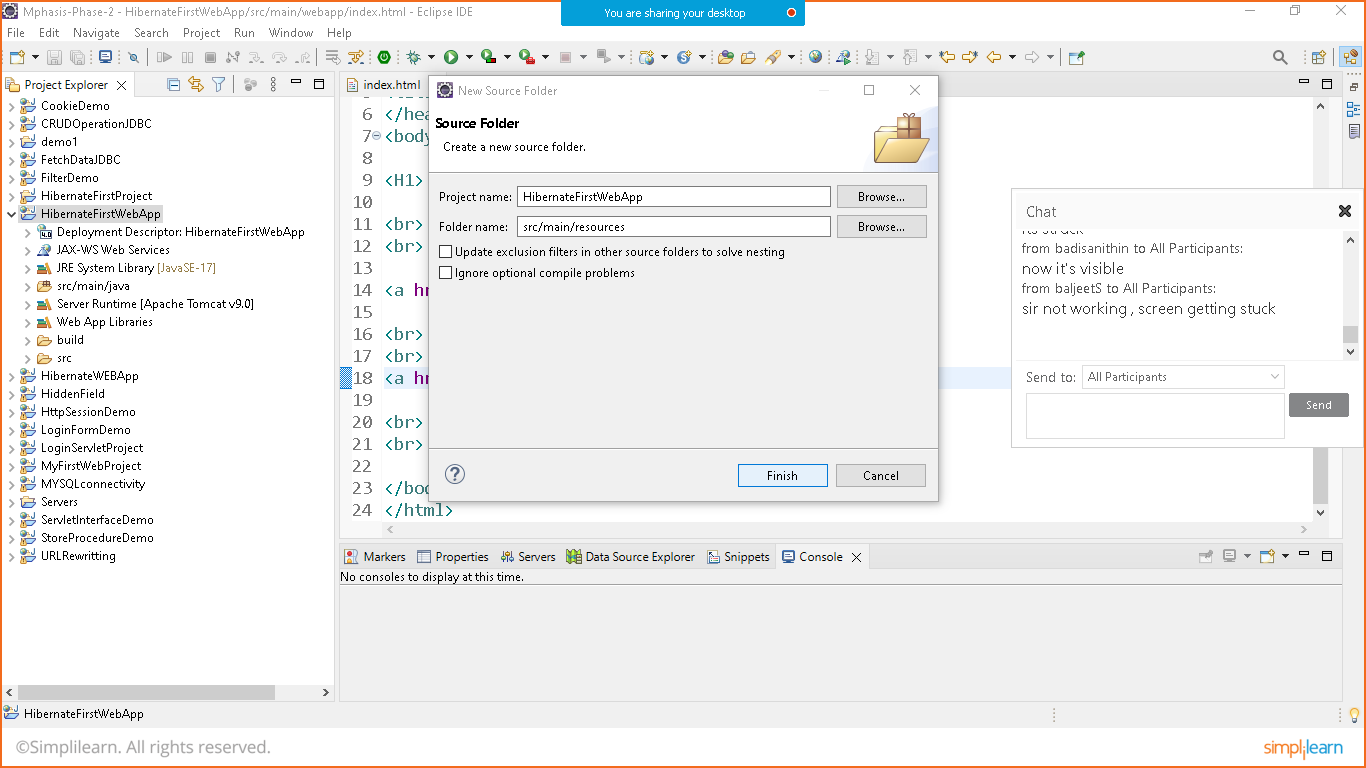


Step:4 Prepare hibernate.cfg.xml file by creating resource folder in app

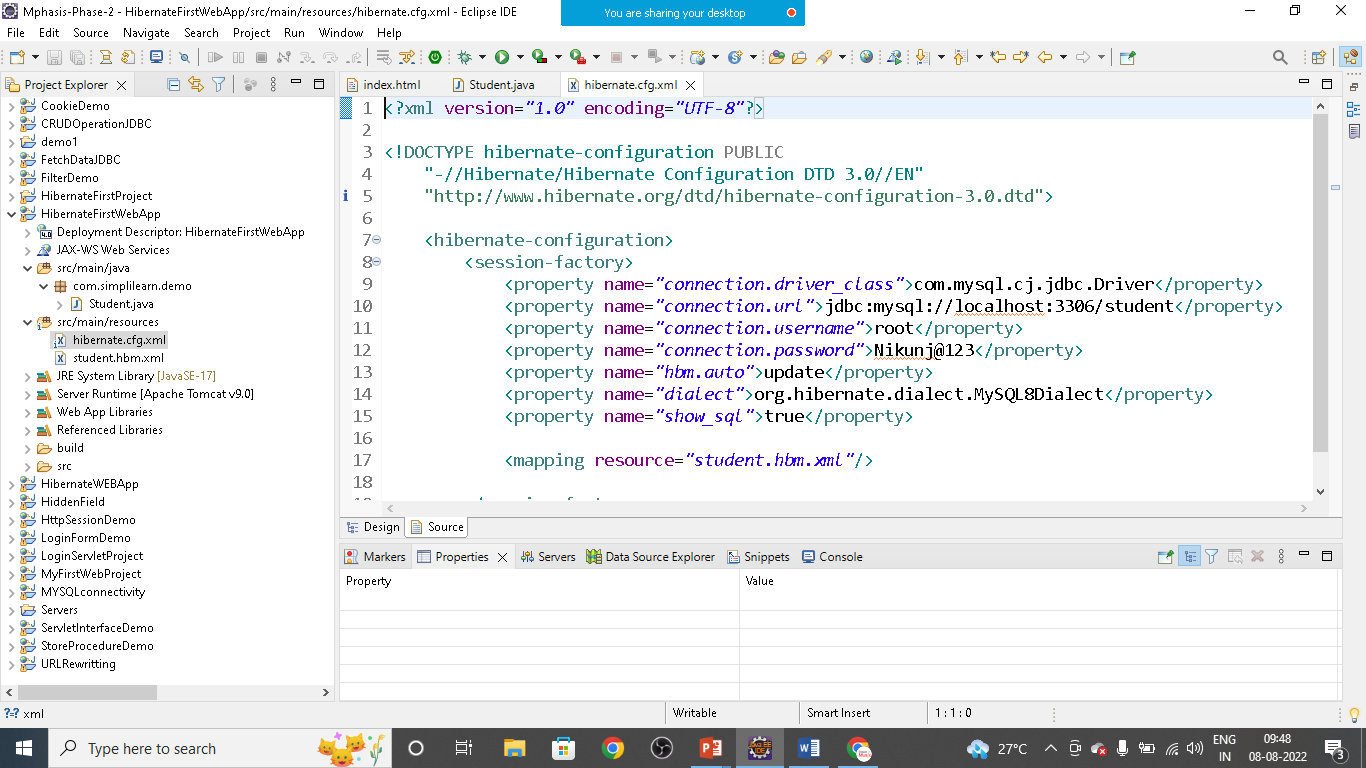
Rightclick project>new>resource folder>give the path

**src/main/resources**





**Right click on resources folder> new>xmlfile> hibernate.cfg.xml**



**hibernate.cfg.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!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>

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

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/student</property>

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

<property name=*"connection.password"*>Nikunj@123</property>

<property name=*"hbm.auto"*>update</property>

<property name=*"dialect"*>org.hibernate.dialect.MySQL8Dialect</property>

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

<mapping resource=*"student.hbm.xml"*/>

</session-factory>

</hibernate-configuration>

**Similarly prepare student.hbm.xml file**

**student.hbm.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

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

<hibernate-mapping package=*"com.simplilearn.demo"*>

<class name=*"Student"* table=*"student"*>

<id name=*"id"* column=*"id"*>

<generator class=*"increment"*></generator>

</id>

<property name=*"name"* type=*"string"* />

<property name=*"email"* type=*"string"* />

<property name=*"password"* type=*"string"* />

</class>

</hibernate-mapping>

STEP:5 Creating Student Bean(Pojo) class and prepare Getter Setter

Student.java

**package** com.simplilearn.demo;

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

**private** **int** id;

**private** String name;

**private** String email;

**private** String password;

**public** Student() {

}

**public** Student(String name, String email, String password) {

**this**.name = name;

**this**.email = email;

**this**.password = password;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

}

STEP:6 PREPARE SERVLET TO TEST THE CONNECTION

Create hibernate .util class to use session factory

Hibernate.utils

**package** com.simplilearn.demo;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.boot.Metadata;

**import** org.hibernate.boot.MetadataSources;

**import** org.hibernate.boot.registry.StandardServiceRegistry;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**public** **class** HibernateUtils {

**private** **static** SessionFactory *sessionFactory*;

**static** {

**try** {

StandardServiceRegistry registry=**new** StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();

Metadata metadata= **new** MetadataSources(registry).getMetadataBuilder().build();

*sessionFactory*= metadata.getSessionFactoryBuilder().build();

}**catch** (Exception e) {

// **TODO**: handle exception

}

}

**public** **static** SessionFactory getSessionFactory()

{

**return** *sessionFactory*;

}

}

Prepare InitializeHibernate Servlet

**package** com.simplilearn.demo;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

@WebServlet("/init")

**public** **class** Intializehibernate **extends** HttpServlet {

@Override

**protected** **void** doGet(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

PrintWriter out= resp.getWriter();

SessionFactory factory= HibernateUtils.*getSessionFactory*();

Session session= factory.openSession();

out.print("Session is Opened");

session.close();

out.print("Session is closed");

}

@Override

**protected** **void** doPost(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {

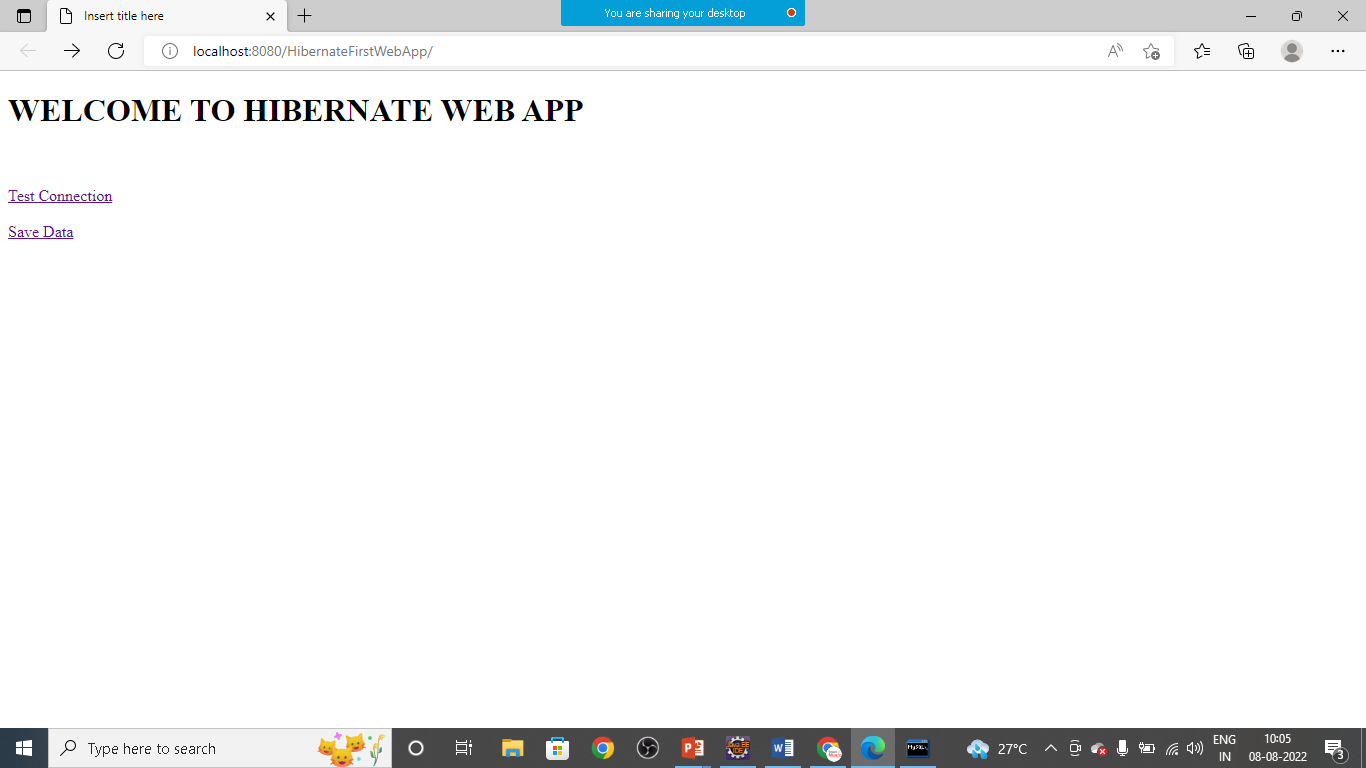
// **TODO** Auto-generated method stub

doGet(req, resp);

}

}

Right click on project and run it



Prepare Save Servlet

package com.simplilearn.demo;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

@WebServlet("/save")

public class SaveServlet extends HttpServlet {

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

// TODO Auto-generated method stub

PrintWriter out= resp.getWriter();

//call hibernate session factory

SessionFactory factory=HibernateUtils.getSessionFactory();

Session session= factory.openSession();

Transaction tx= session.beginTransaction();

Student s1=new Student("Nikunj", "test@gmail.com", "12345");

session.save(s1);

tx.commit();

session.close();

out.println("Data Inserted Successfully");

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(req, resp);

}

}

