**Exercise 2 - Hands on 2 - Hibernate XML Config implementation walk through**

**CODE:**

**Country Class: -**

package com.cognizant.ormlearn.model;  
  
public class Country {  
 private String code;  
 private String name;  
  
 public String getCode() { return code; }  
 public void setCode(String code) { this.code = code; }  
  
 public String getName() { return name; }  
 public void setName(String name) { this.name = name; }  
  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

**hibernate.cfg.xml file: -**

<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">  
  
<hibernate-configuration>  
 <session-factory>  
 <property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>  
 <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/ormlearn</property>  
 <property name="hibernate.connection.username">root</property>  
 <property name="hibernate.connection.password">Sreddy@11</property>  
  
 <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>  
  
 <property name="show\_sql">true</property>  
  
 <mapping resource="Country.hbm.xml"/>  
 </session-factory>  
</hibernate-configuration>

**Country.hbm.xml file: -**

<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">  
  
<hibernate-configuration>  
 <session-factory>  
 <property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>  
 <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/ormlearn</property>  
 <property name="hibernate.connection.username">root</property>  
 <property name="hibernate.connection.password">Sreddy@11</property>  
  
 <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>  
  
 <property name="show\_sql">true</property>  
  
 <mapping resource="Country.hbm.xml"/>  
 </session-factory>  
</hibernate-configuration>

**HibernateUtil Class: -**

package com.cognizant.ormlearn.util;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
  
public class HibernateUtil {  
 private static final SessionFactory *sessionFactory* = *buildSessionFactory*();  
  
 private static SessionFactory buildSessionFactory() {  
 try {  
 return new Configuration().configure("hibernate.cfg.xml").buildSessionFactory();  
 } catch (Throwable ex) {  
 throw new ExceptionInInitializerError("Initial SessionFactory failed" + ex);  
 }  
 }  
  
 public static SessionFactory getSessionFactory() {  
 return *sessionFactory*;  
 }  
  
 public static void shutdown() {  
 *getSessionFactory*().close();  
 }  
}

**OrmlearnApplication Class(Main Class): -**

package com.cognizant.ormlearn;  
  
import com.cognizant.ormlearn.model.Country;  
import com.cognizant.ormlearn.util.HibernateUtil;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.Transaction;  
import org.hibernate.query.Query;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class OrmlearnApplication {  
  
 public static void main(String[] args) {  
 SessionFactory sessionFactory = HibernateUtil.*getSessionFactory*();  
 Session session = sessionFactory.openSession();  
 Transaction tx = null;  
  
 try {  
 tx = session.beginTransaction();  
  
 Country country = new Country();  
 country.setCode("JP");  
 country.setName("Japan");  
 session.save(country);  
  
 Country c = session.get(Country.class, "JP");  
 System.*out*.println("Fetched: " + c);  
  
 Query<Country> query = session.createQuery("from Country", Country.class);  
 for (Country co : query.list()) {  
 System.*out*.println(co);  
 }  
  
 session.delete(c);  
  
 tx.commit();  
 } catch (Exception e) {  
 if (tx != null) tx.rollback();  
 e.printStackTrace();  
 } finally {  
 session.close();  
 }  
  
 HibernateUtil.*shutdown*();  
 }  
  
}

**OUTPUT:**

