**1. Set Up a Maven Project:**

<!-- Spring dependencies -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.10.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.3.10.RELEASE</version>

</dependency>

<!-- Hibernate dependencies -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.3.Final</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>5.3.10.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>5.3.10.RELEASE</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.23</version>

</dependency>

### 2. Create Hibernate Configuration:

<!-- src/main/resources/hibernate.cfg.xml -->

<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- JDBC Database connection settings -->

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

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/your\_database</property>

<property name="hibernate.connection.username">your\_username</property>

<property name="hibernate.connection.password">your\_password</property>

<!-- JDBC connection pool settings -->

<property name="hibernate.c3p0.min\_size">5</property>

<property name="hibernate.c3p0.max\_size">20</property>

<property name="hibernate.c3p0.timeout">300</property>

<property name="hibernate.c3p0.max\_statements">50</property>

<property name="hibernate.c3p0.idle\_test\_period">3000</property>

<!-- Specify dialect -->

<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<!-- Echo all executed SQL to stdout -->

<property name="hibernate.show\_sql">true</property>

<!-- Drop and re-create the database schema on startup -->

<property name="hibernate.hbm2ddl.auto">update</property>

<!-- Mention annotated class or specify packages to scan -->

<!-- <mapping class="com.example.YourEntityClass"/> -->

</session-factory>

</hibernate-configuration>

### 3. Create Entity Class:

// src/main/java/com/example/User.java

package com.example;

import javax.persistence.\*;

@Entity

@Table(name = "users")

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@Column(name = "username")

private String username;

@Column(name = "email")

private String email;

// getters and setters

}

### 4. Create Spring Configuration:

<!-- src/main/resources/applicationContext.xml -->

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<!-- Enable annotation-based configuration -->

<context:annotation-config/>

<!-- Scan for Spring components in the specified package -->

<context:component-scan base-package="com.example"/>

<!-- Hibernate SessionFactory bean -->

<bean id="sessionFactory" class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">

<property name="configLocation" value="classpath:hibernate.cfg.xml"/>

<property name="packagesToScan" value="com.example"/>

</bean>

<!-- Enable Hibernate transaction management -->

<bean id="transactionManager" class="org.springframework.orm.hibernate5.HibernateTransactionManager">

<property name="sessionFactory" ref="sessionFactory"/>

</bean>

</beans>

### 5. Create a DAO (Data Access Object):

// src/main/java/com/example/UserDAO.java

package com.example;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

@Repository

public class UserDAO {

@Autowired

private SessionFactory sessionFactory;

@Transactional

public void saveUser(User user) {

Session session = sessionFactory.getCurrentSession();

session.save(user);

}

@Transactional

public User getUserById(Long id) {

Session session = sessionFactory.getCurrentSession();

return session.get(User.class, id);

}

}

### 6. Create a Spring Service:

// src/main/java/com/example/UserService.java

package com.example;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

@Service

public class UserService {

@Autowired

private UserDAO userDAO;

@Transactional

public void saveUser(User user) {

userDAO.saveUser(user);

}

@Transactional

public User getUserById(Long id) {

return userDAO.getUserById(id);

}

}

### 7. Create a Main Application:

// src/main/java/com/example/MainApp.java

package com.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("classpath:applicationContext.xml");

UserService userService = context.getBean(UserService.class);

// Create and save a user

User user = new User();

user.setUsername("john\_doe");

user.setEmail("john.d