Hibernate CRUD Application

Name: Lokesh Yadav

UID: 22BCS16604

Section: 631 Group A

1. Student Entity (Student.java)

```
package com.example;

import jakarta.persistence.*;

@Entity
@Table(name = "students")
public class Student {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    private String name;
    private int age;

    public Student() {}

    public Student(String name, int age) {
        this.name = name;
        this.age = age;
    }

    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 int getAge() { return age; }
    public void setAge(int age) { this.age = age; }
}
```

2. StudentDao for CRUD Operations (StudentDao.java)

```
package com.example;

import org.hibernate.*;
import org.hibernate.cfg.Configuration;
import java.util.List;

public class StudentDao {
    private static SessionFactory factory = new Configuration().configure().buildSessionFactory();
    public void addStudent(Student s) {
        Session session = factory.openSession();
        Transaction tx = session.beginTransaction();
        session.save(s);
        tx.commit();
        session.close();
    }

    public Student getStudent(int id) {
        Session session = factory.openSession();
        Student s = session.get(Student.class, id);
        session.close();
        return s;
    }

    public void updateStudent(Student s) {
        Session session = factory.openSession();
        Transaction tx = session.beginTransaction();
        session.update(s);
        tx.commit();
        session.close();
    }

    public void deleteStudent(int id) {
        Session session = factory.openSession();
        Transaction tx = session.beginTransaction();
        Student s = session.get(Student.class, id);
        if (s != null) {
            session.delete(s);
            tx.commit();
        }
        session.close();
    }

    public List<Student> getAllStudents() {
        Session session = factory.openSession();
        List<Student> list = session.createQuery("from Student", Student.class).list();
        session.close();
        return list;
    }
}
```

3. Main Test Class (Main.java)

```
package com.example;

public class Main {
    public static void main(String[] args) {
        StudentDao dao = new StudentDao();

        // Create
        Student s1 = new Student("Lokesh", 21);
        dao.addStudent(s1);

        // Read
        Student s = dao.getStudent(s1.getId());
        if (s != null) {
            System.out.println("Read: " + s.getName());
        }

        // Update
        s.setName("Lokesh Yadav");
        dao.updateStudent(s);

        // Delete
        dao.deleteStudent(s.getId());

        // Display All
        dao.getAllStudents().forEach(stud ->
                System.out.println(stud.getId() + ": " + stud.getName() + " - " +
stud.getAge())
        );
    }
}
```

4. Hibernate Configuration (hibernate.cfg.xml)

```
<?xml version='1.0' encoding='utf-8'?>
<!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/studentdb</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">password</property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="hibernate.hbm2ddl.auto">update</property>
        <property name="show_sql">true</property>
        <mapping class="com.example.Student"/>
    </session-factory>
</hibernate-configuration>
```

5. Sample Output

```
Hibernate: insert into students (age, name) values (?, ?)
Read: Lokesh
Hibernate: update students set age=?, name=? where id=?
Hibernate: delete from students where id=?
Hibernate: select student0_.id as id1_0_, student0_.age as age2_0_, student0_.name as
name3_0_ from students student0_
```