Name: JATIN SINGH
SECTION:631/A
UID:222BCS10887

Subject: Project Based Learning in Java
Subject Code: 22CSH-359

EXPERIMENT-9

Hibernate-based application to perform CRUD (Create, Read, Update, Delete) operations on a Student entity using Hibernate ORM with MySQL. Requirements: 1. Configure Hibernate using hibernate.cfg.xml. 2. Create an Entity class (Student.java) with attributes: id, name, and age. 3. Implement Hibernate SessionFactory to perform CRUD operations. 4. Test the CRUD functionality with sample data

```java
// File: Student.java
package com.example.entity;

import javax.persistence.*;

@Entity
@Table(name = "students")
public class Student {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "name", nullable = false)
    private String name;

    @Column(name = "age")
    private int age;

    // Default constructor required by Hibernate
    public Student() {
    }

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

    // Getters and Setters
    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;
    }
```

Name: JATIN SINGH
SECTION:631/A
UID:222BCS10887

Subject: Project Based Learning in Java
Subject Code: 22CSH-359

```
<!-- File: 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>
        <!-- Database connection settings -->
        <property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/hibernatedb?useSSL=false&amp;serverTimezone=UTC</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">password</property>

        <!-- SQL dialect -->
        <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>

        <!-- Echo all executed SQL to stdout -->
        <property name="hibernate.show_sql">true</property>
        <property name="hibernate.format_sql">true</property>

        <!-- Drop and re-create the database schema on startup -->
        <property name="hibernate.hbm2ddl.auto">update</property>

        <!-- Mapping files -->
        <mapping class="com.example.entity.Student"/>
    </session-factory>
</hibernate-configuration>

<!-- File: HibernateUtil.java -->
package com.example.util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
    private static final SessionFactory sessionFactory = buildSessionFactory();

    private static SessionFactory buildSessionFactory() {
        try {
            // Create the SessionFactory from hibernate.cfg.xml
            return new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }

    public static void shutdown() {
        // Close caches and connection pools
```

Name: JATIN SINGH
SECTION:631/A
UID:222BCS10887


```java
import java.util.List;
import org.hibernate.Session;
import org.hibernate.Transaction;
import com.example.entity.Student;
import com.example.util.HibernateUtil;

public class StudentDAO {

    // Create
    public void saveStudent(Student student) {
        Transaction transaction = null;
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            // Start the transaction
            transaction = session.beginTransaction();

            // Save the student object
            session.save(student);

            // Commit the transaction
            transaction.commit();
        } catch (Exception e) {
            if (transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }
    }

    // Read by ID
    public Student getStudentById(int id) {
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            return session.get(Student.class, id);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    // Read all
    public List<Student> getAllStudents() {
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            return session.createQuery("from Student", Student.class).list();
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    // Update
    public void updateStudent(Student student) {
        Transaction transaction = null;
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            // Start the transaction
```

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

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>


    <groupId>com.example</groupId>

    <artifactId>hibernate-crud-demo</artifactId>

    <version>1.0-SNAPSHOT</version>


    <properties>

        <maven.compiler.source>11</maven.compiler.source>

        <maven.compiler.target>11</maven.compiler.target>

        <hibernate.version>5.6.10.Final</hibernate.version>

        <mysql.version>8.0.29</mysql.version>

    </properties>


    <dependencies>

        <!-- Hibernate Core -->

        <dependency>

            <groupId>org.hibernate</groupId>

            <artifactId>hibernate-core</artifactId>

            <version>${hibernate.version}</version>

        </dependency>


        <!-- MySQL Connector -->

        <dependency>

            <groupId>mysql</groupId>
```

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

            <version>${mysql.version}</version>

        </dependency>

    </dependencies>

</project>
```
