**Problem Statement 1** :

**Build Command-Line Product Management Application using Hibernate**

**Structure of the given Problem:**

ProductManagementApplicationUsingHibernate/   
│── src/main/java/com/mphasis/app/   
│   ├── ProductManagementApp.java

│   
│── src/main/java/com/mphasis/dao/   
│   ├── ProductManagementDAO.java

│   
│── src/main/java/com/mphasis/dbutil/   
│   ├── HibernateUtil.java

│   ├── hibernate.cfg.xml

│   
│── src/main/java/com/mphasis/domain/   
│   ├── Product.java

│   ├── product.hbm.xml

│

│── pom.xml                 

**Source Code:**

**ProductManagementApp.java**

**package** com.mphasis.app;

**import** com.mphasis.dao.ProductManagementDAO;

**import** com.mphasis.domain.Product;

**import** java.util.List;

**import** java.util.Scanner;

**public** **class** ProductManagementApp {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

ProductManagementDAO dao = **new** ProductManagementDAO();

**while** (**true**) {

System.***out***.println("\nProduct Management Options:");

System.***out***.println("A. View Products");

System.***out***.println("B. Add Product");

System.***out***.println("C. Update Product");

System.***out***.println("D. Delete Product");

System.***out***.println("E. Search Product");

System.***out***.println("F. Exit");

System.***out***.println("======================================================================================================");

System.***out***.print("Enter an option \n");

System.***out***.println("======================================================================================================");

**char** choice = scanner.next().toUpperCase().charAt(0);

**switch** (choice) {

**case** 'A':

System.***out***.println("-------------------------------------------------------------------------------------------------");

List<Product> products = dao.viewProducts();

**for** (Product product : products) {

 System.***out***.println(product);

}

System.***out***.println("-------------------------------------------------------------------------------------------------");

**break**;

**case** 'B':

System.***out***.println("-------------------------------------------------------------------------------------------------");

System.***out***.print("Enter Product Id: \n");

System.***out***.println("-------------------------------------------------------------------------------------------------");

String id= scanner.next();

System.***out***.println("-------------------------------------------------------------------------------------------------");

System.***out***.print("Enter Product Name: \n");

System.***out***.println("-------------------------------------------------------------------------------------------------");

String name = scanner.next();

System.***out***.println("-------------------------------------------------------------------------------------------------");

System.***out***.print("Enter Product Price: \n");

System.***out***.println("-------------------------------------------------------------------------------------------------");

**double** price = scanner.nextDouble();

System.***out***.println("-------------------------------------------------------------------------------------------------");

dao.addProduct(**new** Product(id,name, price));

List<Product> products1 = dao.viewProducts();

**for** (Product product : products1) {

 System.***out***.println(product);

}

**break**;

**case** 'C':

System.***out***.println("-------------------------------------------------------------------------------------------------");

System.***out***.print("Enter Product ID: \n");

System.***out***.println("-------------------------------------------------------------------------------------------------");

String updateId = scanner.next();

System.***out***.println("-------------------------------------------------------------------------------------------------");

System.***out***.print("Enter New Product Name: \n");

System.***out***.println("-------------------------------------------------------------------------------------------------");

String newName = scanner.next();

System.***out***.println("-------------------------------------------------------------------------------------------------");

System.***out***.print("Enter New Product Price: \n");

System.***out***.println("-------------------------------------------------------------------------------------------------");

**double** newPrice = scanner.nextDouble();

System.***out***.println("-------------------------------------------------------------------------------------------------");

dao.updateProduct(updateId, newName, newPrice);

**break**;

**case** 'D':

System.***out***.println("-------------------------------------------------------------------------------------------------");

System.***out***.print("Enter Product ID: \n");

System.***out***.println("-------------------------------------------------------------------------------------------------");

String deleteId = scanner.next();

dao.deleteProduct(deleteId);

**break**;

**case** 'E':

System.***out***.println("-------------------------------------------------------------------------------------------------");

System.***out***.print("Enter Product ID: \n");

System.***out***.println("-------------------------------------------------------------------------------------------------");

String searchId = scanner.next();

System.***out***.println(dao.searchProduct(searchId));

**break**;

**case** 'F':

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Thank You\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

scanner.close();

**return**;

**default**:

System.***out***.println("Invalid choice. Try again.");

}

}

}

}

**ProductManagementDAO.java**

**package** com.mphasis.dao;

**import** com.mphasis.dbutil.HibernateUtil;

**import** com.mphasis.domain.Product;

**import** org.hibernate.Session;

**import** org.hibernate.Transaction;

**import** java.util.List;

**public** **class** ProductManagementDAO {

**public** **void** addProduct(Product product) {

Session session = HibernateUtil.*getSessionFactory*().openSession();

Transaction tx = session.beginTransaction();

session.save(product);

tx.commit();

session.close();

System.***out***.println("Product added successfully.");

}

**public** List<Product> viewProducts() {

Session session = HibernateUtil.*getSessionFactory*().openSession();

List<Product> products = session.createQuery("FROM Product", Product.**class**).list();

session.close();

**return** products;

}

**public** **void** updateProduct(String productId, String newName, **double** newPrice) {

Session session = HibernateUtil.*getSessionFactory*().openSession();

Transaction tx = session.beginTransaction();

Product product = session.get(Product.**class**, productId);

**if** (product != **null**) {

product.setName(newName);

product.setPrice(newPrice);

session.update(product);

tx.commit();

System.***out***.println("Product updated successfully.");

} **else** {

System.***out***.println("Product not found.");

}

session.close();

}

**public** **void** deleteProduct(String productId) {

Session session = HibernateUtil.*getSessionFactory*().openSession();

Transaction tx = session.beginTransaction();

Product product = session.get(Product.**class**, productId);

**if** (product != **null**) {

session.delete(product);

tx.commit();

System.***out***.println("Product deleted successfully.");

} **else** {

System.***out***.println("Product not found.");

}

session.close();

}

**public** Product searchProduct(String productId) {

Session session = HibernateUtil.*getSessionFactory*().openSession();

Product product = session.get(Product.**class**, productId);

session.close();

**return** product;

}

}

**HibernateUtil.java**

**package** com.mphasis.dbutil;

**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("com/mphasis/dbutil/hibernate.cfg.xml").buildSessionFactory();

} **catch** (Throwable ex) {

System.***err***.println("SessionFactory creation failed: " + ex);

**throw** **new** ExceptionInInitializerError(ex);

}

}

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

**return** ***sessionFactory***;

}

**public** **static** **void** shutdown() {

*getSessionFactory*().close();

}

}

**Product.java**

**package** com.mphasis.domain;

**public** **class** Product {

**private** String id;

**private** String name;

**private** **double** price;

**public** Product() {}

**public** Product(String id, String name, **double** price) {

**this**.id = id;

**this**.name = name;

**this**.price = price;

}

**public** String getId() {

**return** id;

}

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

**this**.id = id;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

@Override

**public** String toString() {

**return** "Product [ID=" + id + ", Name=" + name + ", Price=" + price + "]";

}

}

**Hibernate.cfg.xml**

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

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 5.3//EN"

"<http://hibernate.org/dtd/hibernate-configuration-3.0.dtd>">

<hibernate-configuration>

<session-factory>

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

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

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

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

<property name=*"connection.password"*>Password@12</property>

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

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

<mapping resource=*"com/mphasis/domain/product.hbm.xml"*/>

<mapping class=*"com.mphasis.domain.Product"*></mapping>

</session-factory>

</hibernate-configuration>

**Product.hbm.xml**

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

<!DOCTYPE hibernate-mapping PUBLIC

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

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

<hibernate-mapping>

<class name=*"com.mphasis.domain.Product"* table=*"product"*>

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

<generator class=*"assigned"*/>

</id>

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

<property name=*"price"* column=*"price"* type=*"double"*/>

</class>

</hibernate-mapping>

**Pom.xml**

<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 <https://maven.apache.org/xsd/maven-4.0.0.xsd>">

<modelVersion>4.0.0</modelVersion>

<groupId>com.mphasis</groupId>

<artifactId>productmanagementapplicationusinghibernate</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>productmanagementapplicationusinghibernate</name>

<description>Create a Product Management application which provides 6 operations to the user</description>

<dependencies>

<!-- <https://mvnrepository.com/artifact/com.mysql/mysql-connector-j> -->

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<version>9.3.0</version>

</dependency>

<dependency>

<groupId>org.hibernate.orm</groupId>

<artifactId>hibernate-core</artifactId>

<version>6.6.16.Final</version>

</dependency>

</dependencies>

</project>

**Create Database:**

   create database productdb;