**DAY-12 PRACTICE EXERCISE**

**BASICS OF JAVA**

**Product Class**

**package** com.mphasis.domain;

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

**private** String id;

**private** String name;

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

**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;

}

}

**DBUtil Class**

package com.mphasis.dbutil;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBUtil {

private static final String Driver = "com.mysql.cj.jdbc.Driver";

private static final String Url = "jdbc:mysql://localhost:3306/productdb";

private static final String User = "root";

private static final String Password = "J2EE@java";

public static String getDriver() {

try {

return Driver;

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

public static Connection getConnection() {

try {

return DriverManager.getConnection(Url, User, Password);

} catch (SQLException e) {

e.printStackTrace();

return null;

}

}

}

**ProductManagementDAO Class**

package com.mphasis.dao;

import com.mphasis.dbutil.DBUtil;

import com.mphasis.domain.Product;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

import javax.swing.text.AbstractDocument.BranchElement;

public class ProductManagementDAO {

Connection con = null;

PreparedStatement pstmt = null;

ResultSet rs = null;

public boolean addProduct(Product product) {

String query = "INSERT INTO products (product\_id,product\_name, product\_price) VALUES (?, ?, ?)";

Product p = searchProductById(product.getId());

if (p == null) {

try {

Class.forName(DBUtil.getDriver());

con = DBUtil.getConnection();

pstmt = con.prepareStatement(query);

pstmt.setString(1, product.getId());

pstmt.setString(2, product.getName());

pstmt.setDouble(3, product.getPrice());

pstmt.executeUpdate();

return true;

} catch (SQLException | ClassNotFoundException e) {

e.printStackTrace();

}

}

return false;

}

public void updateProduct(Product product) {

String query = "UPDATE products SET product\_name = ?, product\_price = ? WHERE product\_id = ?";

try {

Class.forName(DBUtil.getDriver());

con = DBUtil.getConnection();

pstmt = con.prepareStatement(query);

pstmt.setString(1, product.getName());

pstmt.setDouble(2, product.getPrice());

pstmt.setString(3, product.getId());

pstmt.executeUpdate();

} catch (SQLException | ClassNotFoundException e) {

e.printStackTrace();

}

}

public List<Product> getAllProducts() {

List<Product> productList = new ArrayList<>();

String query = "SELECT \* FROM products";

try {

Class.forName(DBUtil.getDriver());

con = DBUtil.getConnection();

pstmt = con.prepareStatement(query);

rs = pstmt.executeQuery(query);

while (rs.next()) {

productList.add(new Product(rs.getString("product\_id"), rs.getString("product\_name"),

rs.getDouble("product\_price")));

}

} catch (SQLException | ClassNotFoundException e) {

e.printStackTrace();

}

return productList;

}

public void deleteProduct(String id) {

String query = "DELETE FROM products WHERE product\_id = ?";

try {

Class.forName(DBUtil.getDriver());

con = DBUtil.getConnection();

pstmt = con.prepareStatement(query);

pstmt.setString(1, id);

pstmt.executeUpdate();

} catch (SQLException | ClassNotFoundException e) {

e.printStackTrace();

}

}

public Product searchProductById(String id) {

String query = "SELECT \* FROM products WHERE product\_id = ?";

Product product = null;

try {

Class.forName(DBUtil.getDriver());

con = DBUtil.getConnection();

pstmt = con.prepareStatement(query);

pstmt.setString(1, id);

rs = pstmt.executeQuery();

if (rs.next()) {

product = new Product(rs.getString("product\_id"), rs.getString("product\_name"),

rs.getDouble("product\_price"));

}

} catch (SQLException | ClassNotFoundException e) {

e.printStackTrace();

}

return product;

}

}

**ProductManagementApp Class**

package com.mphasis.app;

import com.mphasis.dao.ProductManagementDAO;

import com.mphasis.domain.Product;

import java.sql.SQLIntegrityConstraintViolationException;

import java.util.List;

import java.util.Scanner;

public class ProductManagementApp {

static ProductManagementDAO productDAO = new ProductManagementDAO();

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

public static void main(String[] args) {

while (true) {

System.out.println("Product Management System");

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.println("Enter an option : ");

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

String option = scanner.nextLine().toUpperCase();

switch (option) {

case "A":

viewProducts();

break;

case "B":

addProduct();

break;

case "C":

updateProduct();

break;

case "D":

deleteProduct();

break;

case "E":

searchProduct();

break;

case "F":

System.err.println("\*\*\*\*\*\*\*\*\*THANK YOU\*\*\*\*\*\*\*\*\*");

System.exit(0);

default:

System.err.println("Invalid option. Please try again.");

}

}

}

private static void viewProducts() {

List<Product> products = productDAO.getAllProducts();

if (products.isEmpty()) {

System.out.println("No products available.");

} else {

System.out.println("All Product List");

for (Product product : products) {

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

System.out.println("Product ID : " + product.getId());

System.out.println("Product Name : " + product.getName());

System.out.println("Product Price : " + product.getPrice());

}

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

}

}

private static void addProduct() {

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

System.out.println("Enter product Id: ");

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

String id = scanner.nextLine();

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

System.out.println("Enter product name: ");

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

String name = scanner.nextLine();

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

System.out.println("Enter product price: ");

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

double price = Double.parseDouble(scanner.nextLine());

Product product = new Product(id, name, price);

boolean b=productDAO.addProduct(product);

if(b) {

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

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

viewProducts();

}

else {

System.out.println("Duplicate Entey Id Is already Present");

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

}

}

private static void updateProduct() {

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

System.out.println("Enter product ID to update: ");

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

String id = scanner.nextLine();

Product isProduct = productDAO.searchProductById(id);

if (isProduct == null) {

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

} else {

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

System.out.println("Enter new product name: ");

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

String name = scanner.nextLine();

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

System.out.println("Enter new product price: ");

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

double price = Double.parseDouble(scanner.nextLine());

isProduct.setName(name);

isProduct.setPrice(price);

productDAO.updateProduct(isProduct);

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

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

viewProducts();

}

}

private static void deleteProduct() {

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

System.out.println("Enter product ID to delete: ");

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

String id = scanner.nextLine();

productDAO.deleteProduct(id);

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

}

private static void searchProduct() {

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

System.out.println("Enter product ID to search: ");

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

String id = scanner.nextLine();

Product product = productDAO.searchProductById(id);

if (product == null) {

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

} else {

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

System.out.println("Product ID : " + product.getId());

System.out.println("Product Name : " + product.getName());

System.out.println("Product Price : " + product.getPrice());

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

}

}

}