```java
package com.main;

import com.utility.MenuOption;

public class ProductApp {

public static void main(String[] args) {

MenuOption.productMenuOption();
}
}
```

```java
package com.utility;

import java.util.List;
import java.util.Scanner;

import com.entity.AddProduct;
import com.service.AddProductService;

public class MenuOption {

public static void productMenuOption() {

Scanner sc=new Scanner(System.in);
int choice;
String con="";
int id;
String name;
float price;
String result;
AddProductService pp = new AddProductService();

do {
System.out.println("Product Menu");
System.out.println("1:Add 2: Retrive All Products");
System.out.println("Enter your choice");
choice = sc.nextInt();
switch (choice) {
case 1:System.out.println("Enter product id");
 id = sc.nextInt();
 System.out.println("Enter product name");
 name= sc.next();
 System.out.println("Enter product price");
 price = sc.nextFloat();

 AddProduct p1 = new AddProduct(id, name, price);
 result = pp.storeProduct(p1);
 System.out.println(result);
break;
case 2:List<AddProduct> listOfProduct = pp.findAllProducts();
            for(AddProduct p:listOfProduct) {
                  System.out.println(p); // all student details
            }
```

```
            break;
default:
break;
}
System.out.println("Do you want to continue?(y/n)");
con = sc.next();
}while(con.equals("y"));

}
} package com.service;

import java.util.List;

import com.dao.AddProductDao;
import com.entity.AddProduct;

public class AddProductService {

AddProductDao pd = new AddProductDao();

public String storeProduct(AddProduct addproduct) {
if(pd.storeProduct(addproduct)>0) {
return "Product record stored successfully";
}else {
return "Record didn't store";
}
}


public List<AddProduct> findAllProducts() {
return pd.findAllProducts();
}
}
package com.dao;

import java.util.List;

import javax.persistence.TypedQuery;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import com.entity.AddProduct;
import com.resource.HibernateResource;

public class AddProductDao {

public int storeProduct(AddProduct addproduct) {
try {
SessionFactory sf = HibernateResource.getSessionFactory();
Session session = sf.openSession();
Transaction tran = session.getTransaction();
tran.begin();
session.save(addproduct);
```

```
tran.commit();
return 1;
} catch (Exception e) {
System.err.println(e);
return 0;
}
}

public List<AddProduct> findAllProducts() {
SessionFactory sf = HibernateResource.getSessionFactory();
Session session = sf.openSession();
TypedQuery tq = session.createQuery("from AddProduct");
List<AddProduct> listOfProducts= tq.getResultList();
return listOfProducts;
}
}
```

```
package com.entity;

import javax.persistence.Entity;
import javax.persistence.Id;

@Entity
public class AddProduct {

@Id
private int id;
private String name;
private float price;

public AddProduct() {
super();
// TODO Auto-generated constructor stub
}

public AddProduct(int id, String name, float price) {
super();
this.id = id;
this.name = name;
this.price = price;
}

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 float getPrice() {
return price;
}

public void setPrice(float price) {
this.price = price;
}

@Override
public String toString() {
return "Product [id=" + id + ", name=" + name + ", price=" + price + "]";
}

}
```

```
package com.resource;


import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;


public class HibernateResource {


static SessionFactory sf;

// This code execute only once.

static {

Configuration con = new Configuration();

con.configure("hibernate.cfg.xml");

sf = con.buildSessionFactory();

}


public static SessionFactory getSessionFactory() {

try {

return sf;

}catch(Exception e) {

System.err.println("Session Factory error "+e);

return null;

}
```

```
}

}
```

```
<?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>
<property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property
>
<property name="hibernate.connection.password">Un$ammu#2982</property>
<property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/mydb_phase2</pr
operty>
<property name="hibernate.connection.username">root</property>
<property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
<mapping class="com.entity.AddProduct"/>
</session-factory>
</hibernate-configuration>
```