**CASE STUDY : PRODUCT APP**

class Product:

def \_\_init\_\_(self, pid, name, category, price):

self.pid = pid

self.name = name

self.category = category

self.price = price

class ProductManager:

def \_\_init\_\_(self):

self.products = []

def add\_product(self):

pid = input("Enter Product ID: ")

name = input("Enter Product Name: ")

category = input("Enter Product Category: ")

price = float(input("Enter Product Price: "))

product = Product(pid, name, category, price)

self.products.append(product)

print("Product added successfully.")

def update\_product(self):

pid = input("Enter Product ID to update: ")

for product in self.products:

if product.pid == pid:

product.name = input("Enter new Product Name: ")

product.category = input("Enter new Product Category: ")

product.price = float(input("Enter new Product Price: "))

print("Product updated successfully.")

return

print("Product not found.")

def delete\_product(self):

pid = input("Enter Product ID to delete: ")

for product in self.products:

if product.pid == pid:

self.products.remove(product)

print("Product deleted successfully.")

return

print("Product not found.")

def get\_product\_by\_pid(self):

pid = input("Enter Product ID: ")

for product in self.products:

if product.pid == pid:

print(f"Product ID: {product.pid}, Name: {product.name}, Category: {product.category}, Price: {product.price}")

return

print("Product not found.")

def get\_all\_products(self):

if not self.products:

print("No products available.")

else:

for product in self.products:

print(f"Product ID: {product.pid}, Name: {product.name}, Category: {product.category}, Price: {product.price}")

def get\_products\_by\_category(self):

category = input("Enter Category: ")

filtered\_products = [product for product in self.products if product.category == category]

if filtered\_products:

for product in filtered\_products:

print(f"Product ID: {product.pid}, Name: {product.name}, Price: {product.price}")

else:

print(f"No products found in the '{category}' category.")

def get\_products\_between\_prices(self):

min\_price = float(input("Enter minimum price: "))

max\_price = float(input("Enter maximum price: "))

filtered\_products = [product for product in self.products if min\_price <= product.price <= max\_price]

if filtered\_products:

for product in filtered\_products:

print(f"Product ID: {product.pid}, Name: {product.name}, Category: {product.category}, Price: {product.price}")

else:

print(f"No products found between {min\_price} and {max\_price}.")

def main():

manager = ProductManager()

while True:

print("\n1. Add Product")

print("2. Update Product")

print("3. Delete Product")

print("4. Get Product By ID")

print("5. Get All Products")

print("6. Get Products By Category")

print("7. Get Products Between Prices")

print("8. Exit")

choice = input("Enter your choice: ")

if choice == '1':

manager.add\_product()

elif choice == '2':

manager.update\_product()

elif choice == '3':

manager.delete\_product()

elif choice == '4':

manager.get\_product\_by\_pid()

elif choice == '5':

manager.get\_all\_products()

elif choice == '6':

manager.get\_products\_by\_category()

elif choice == '7':

manager.get\_products\_between\_prices()

elif choice == '8':

print("Exiting...")

break

else:

print("Invalid choice, please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

main()