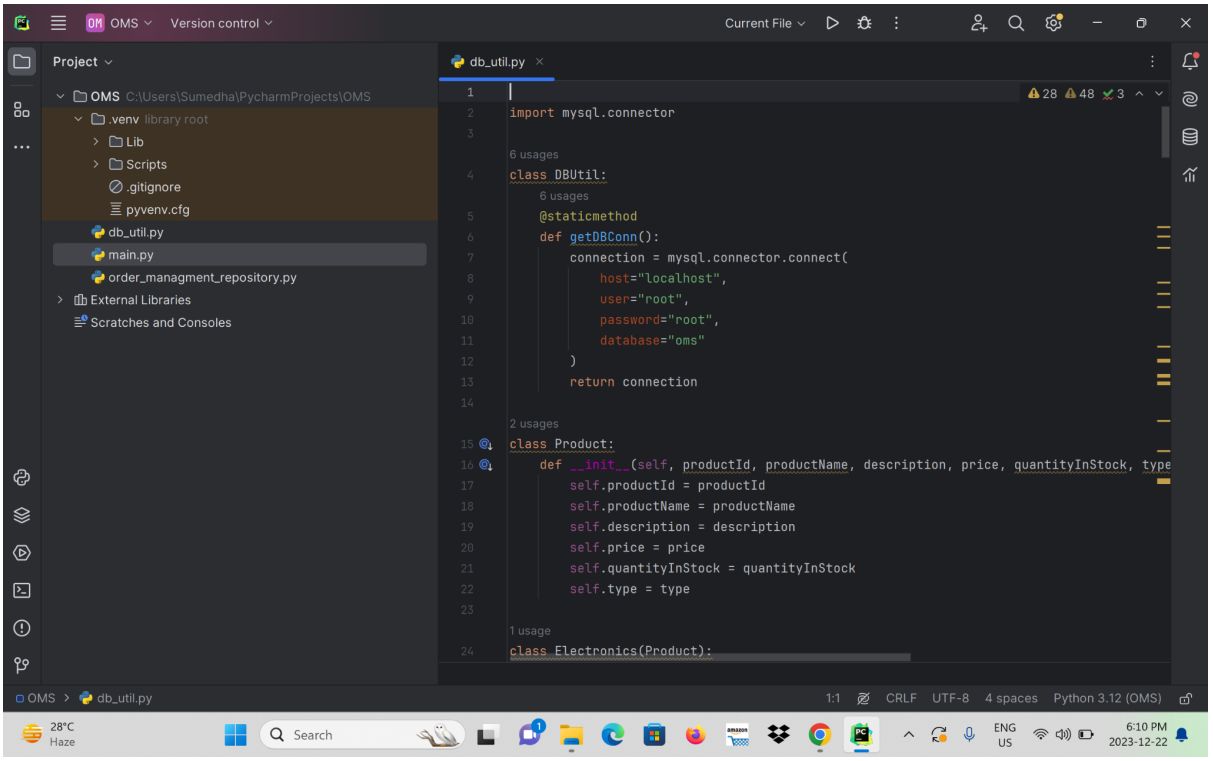
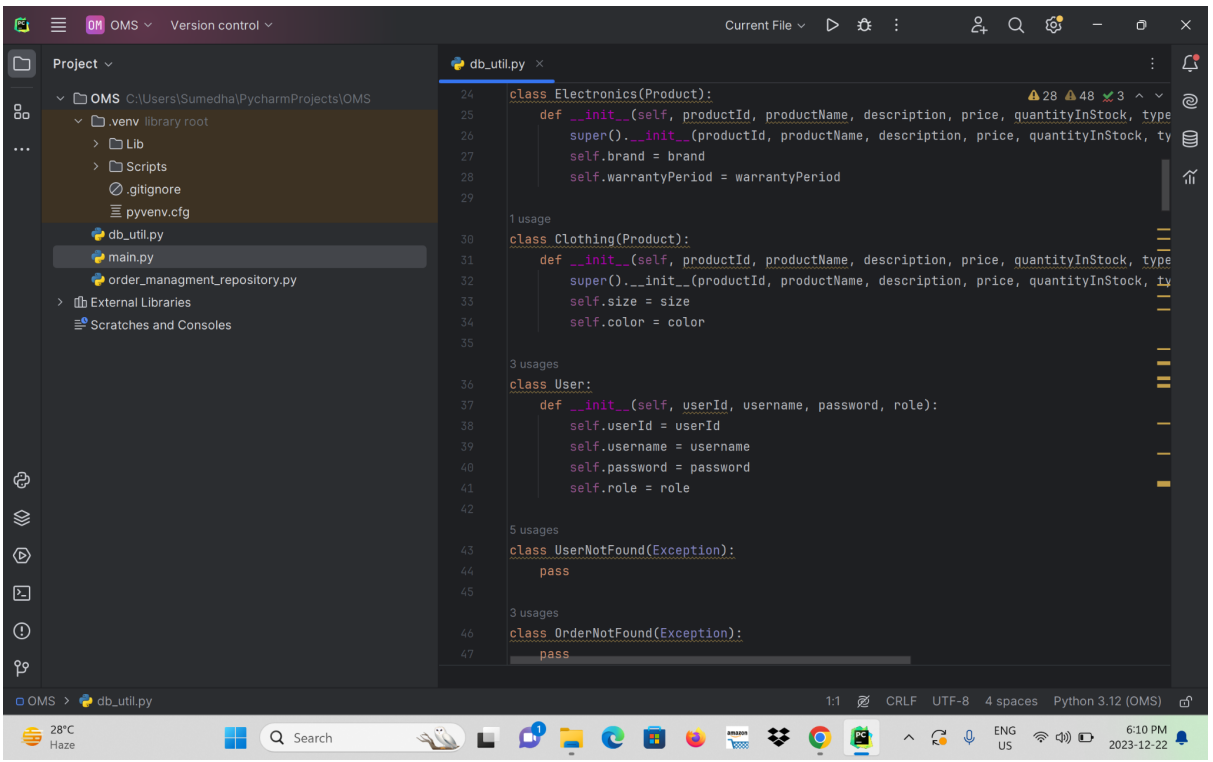
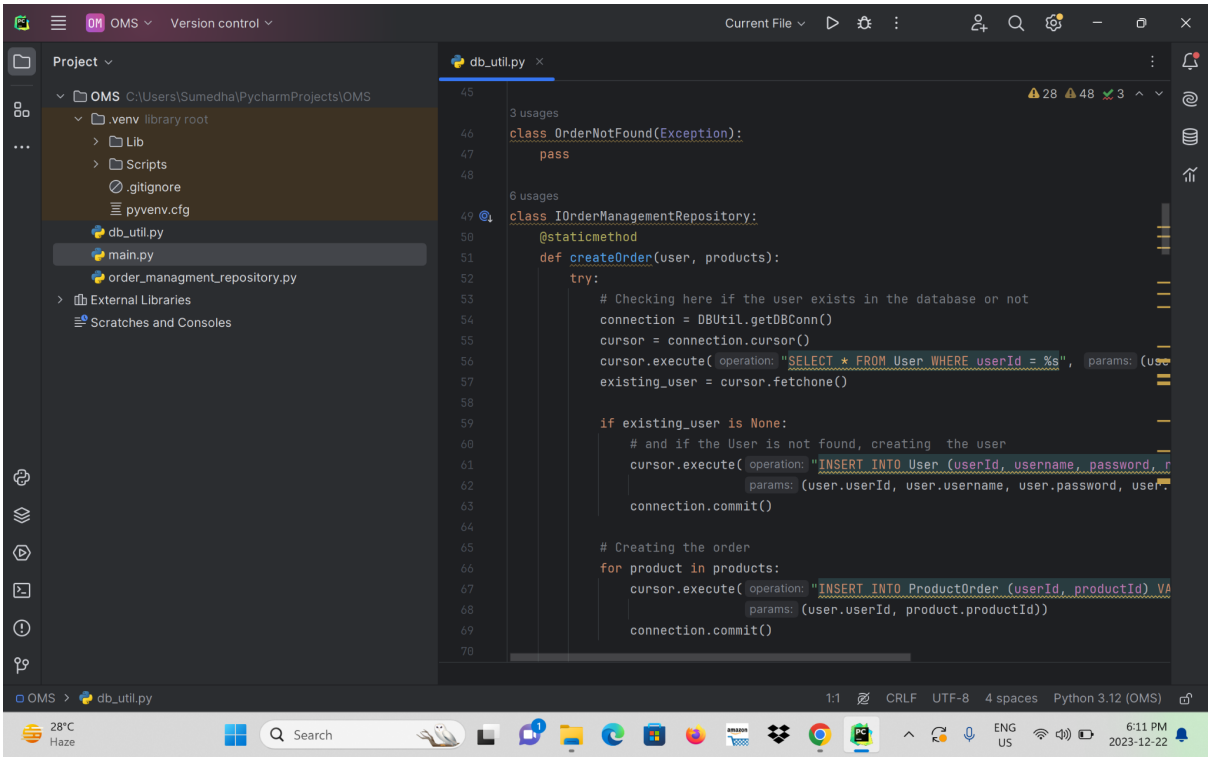
CODING CHALLENGE:

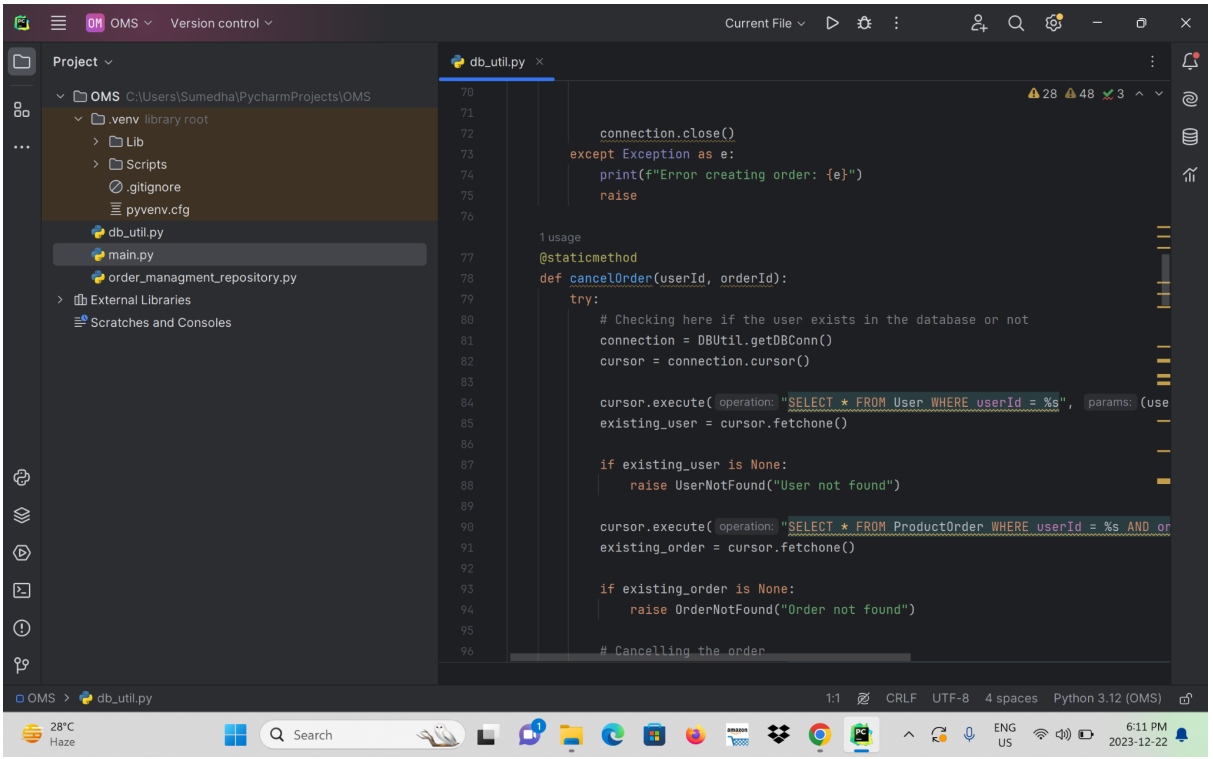


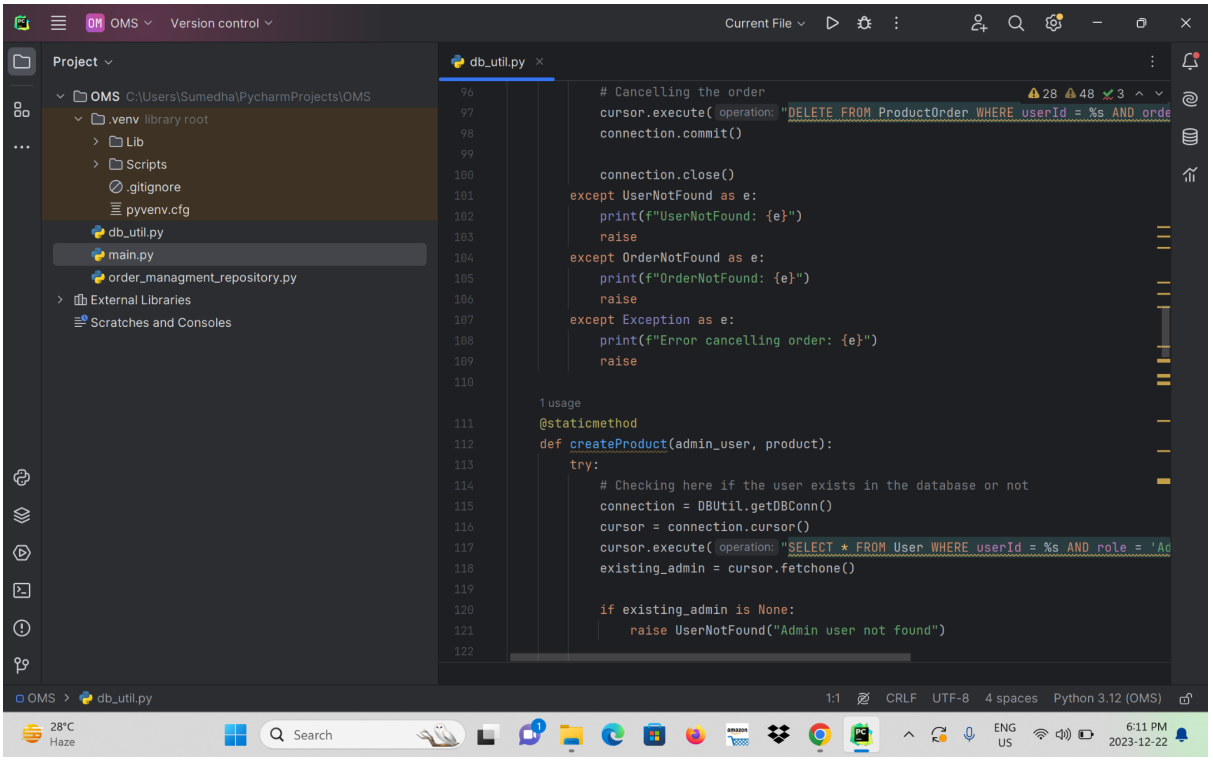
```
import mysql.connector

6 usages
class DBUtil:
    6 usages
    @staticmethod
    def getDBConn():
        connection = mysql.connector.connect(
            host="localhost",
            user="root",
            password="root",
            database="oms"
        )
        return connection

2 usages
class Product:
    def __init__(self, productId, productName, description, price, quantityInStock, type
        self.productId = productId
        self.productName = productName
        self.description = description
        self.price = price
        self.quantityInStock = quantityInStock
        self.type = type

1 usage
class Electronics(Product):
```

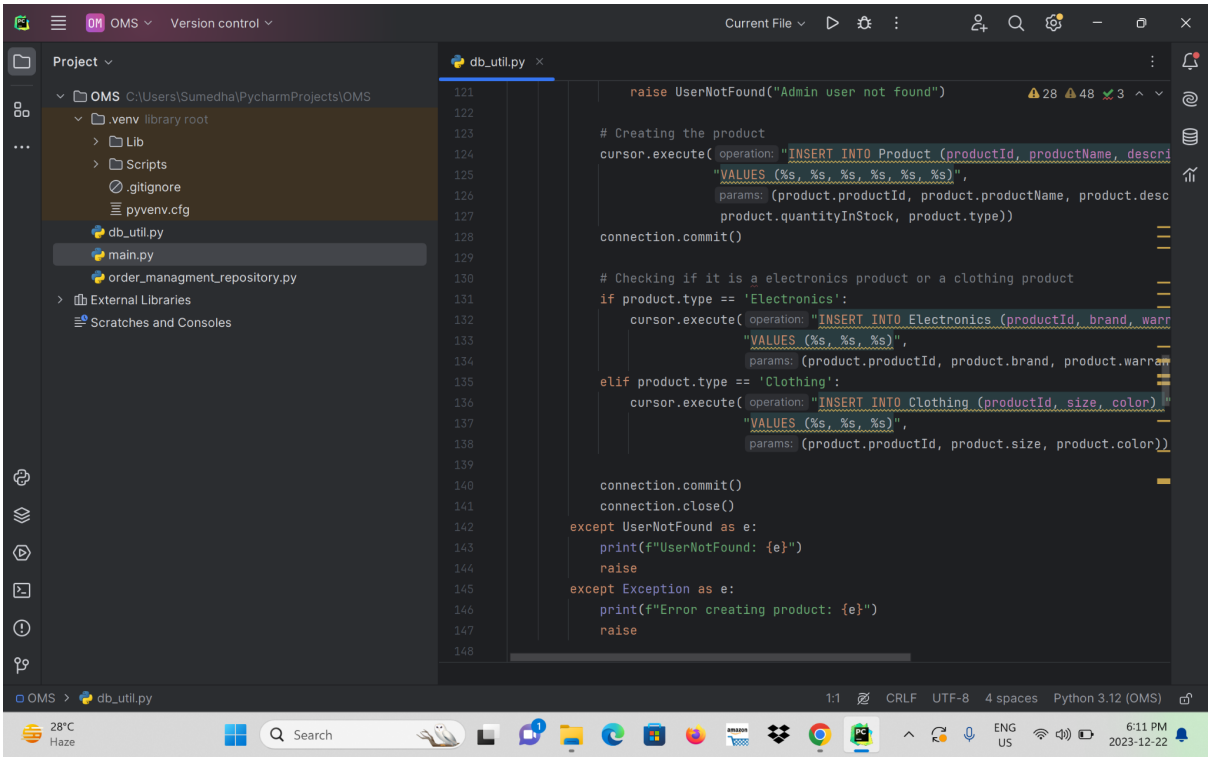


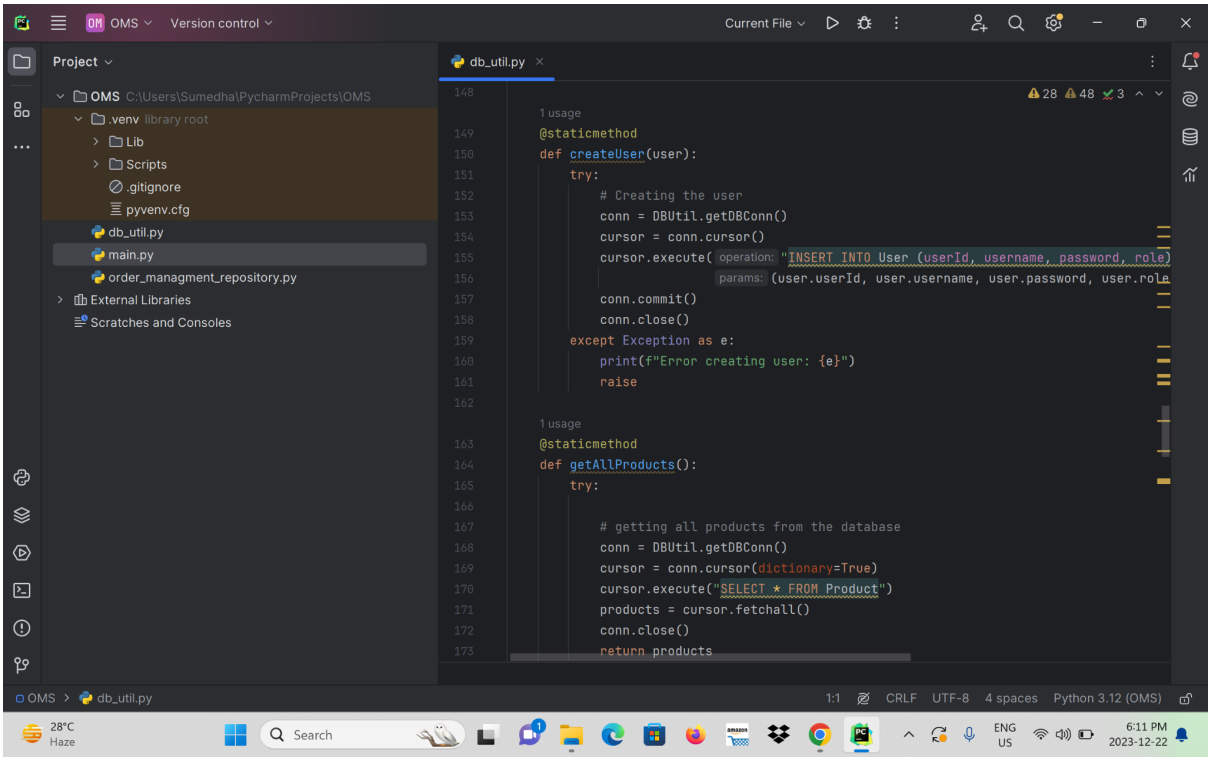
```
class Electronics(Product):
    def __init__(self, productId, productName, description, price, quantityInStock, type
        super().__init__(productId, productName, description, price, quantityInStock, ty
        self.brand = brand
        self.warrantyPeriod = warrantyPeriod

1 usage
class Clothing(Product):
    def __init__(self, productId, productName, description, price, quantityInStock, type
        super().__init__(productId, productName, description, price, quantityInStock, ty
        self.size = size
        self.color = color

3 usages
class User:
    def __init__(self, userId, username, password, role):
        self.userId = userId
        self.username = username
        self.password = password
        self.role = role

5 usages
class UserNotFound(Exception):
    pass

3 usages
class OrderNotFound(Exception):
    pass
```

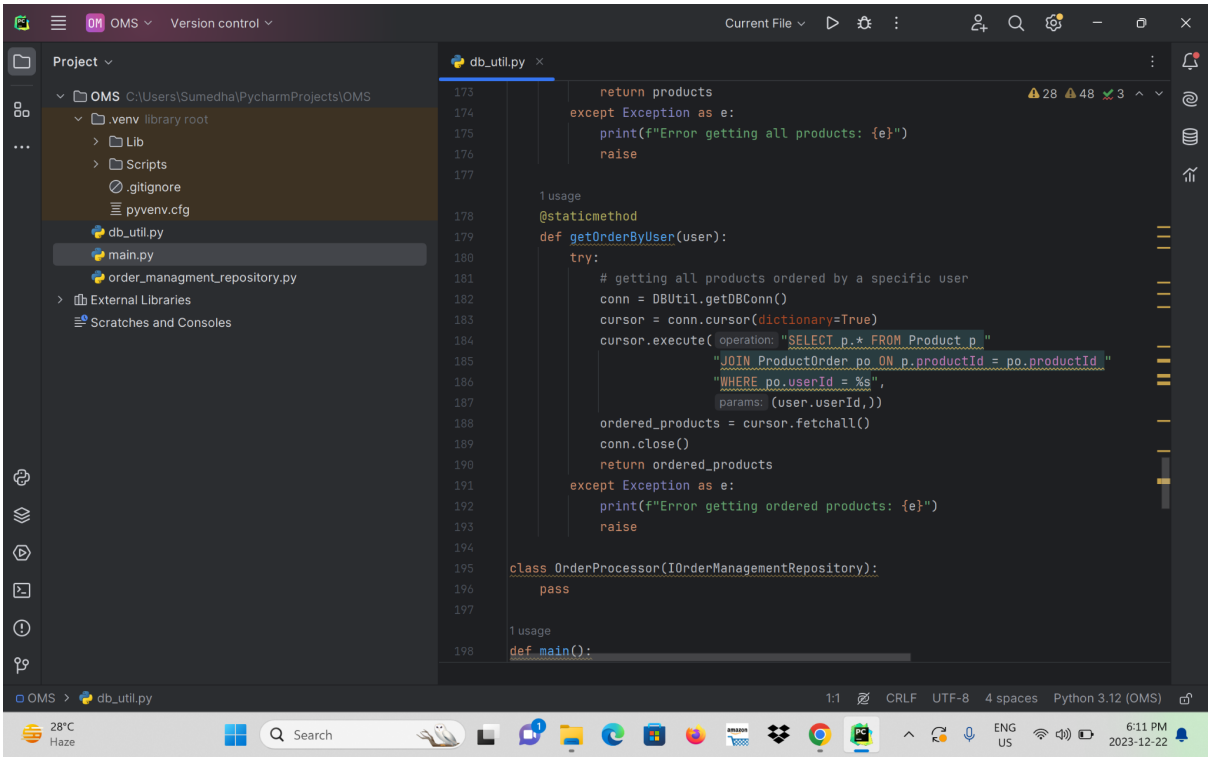

OMS
Version control
Current File
Project
db_util.py
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
db_util.py
main.py
order_managment_repository.py
External Libraries
Scratches and Consoles
3 usages
class OrderNotFound(Exception):
pass
6 usages
class IOrderManagementRepository:
@staticmethod
def createOrder(user, products):
try:
# Checking here if the user exists in the database or not
connection = DBUtil.getDBConn()
cursor = connection.cursor()
cursor.execute(operation: "SELECT * FROM User WHERE userId = %s", params: (use
existing_user = cursor.fetchone()
if existing_user is None:
# and if the User is not found, creating the user
cursor.execute(operation: "INSERT INTO User (userId, username, password, r
params: (user.userId, user.username, user.password, user
connection.commit()
# Creating the order
for product in products:
cursor.execute(operation: "INSERT INTO ProductOrder (userId, productId) VA
params: (user.userId, product.productId))
connection.commit()
28 48 3
OMS db_util.py
1:1 CRLF UTF-8 4 spaces Python 3.12 (OMS)
28°C Haze
Search
ENG US
6:11 PM 2023-12-22

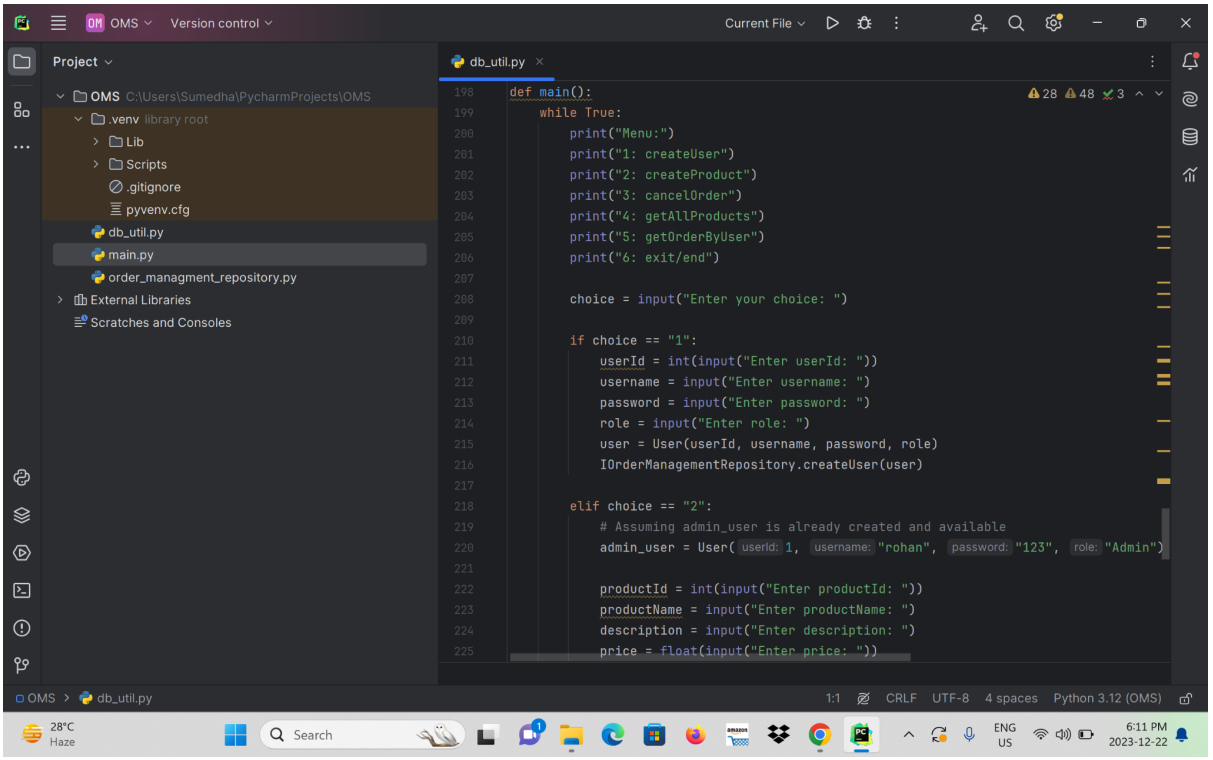

OMS
Version control
Current File
Project
db_util.py
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
db_util.py
main.py
order_managment_repository.py
External Libraries
Scratches and Consoles
connection.close()
except Exception as e:
print(f"Error creating order: {e}")
raise
1 usage
@staticmethod
def cancelOrder(userId, orderId):
try:
# Checking here if the user exists in the database or not
connection = DBUtil.getDBConn()
cursor = connection.cursor()
cursor.execute(operation: "SELECT * FROM User WHERE userId = %s", params: (use
existing_user = cursor.fetchone()
if existing_user is None:
raise UserNotFound("User not found")
cursor.execute(operation: "SELECT * FROM ProductOrder WHERE userId = %s AND or
existing_order = cursor.fetchone()
if existing_order is None:
raise OrderNotFound("Order not found")
Cancelling the order
28 48 3
OMS db_util.py
1:1 CRLF UTF-8 4 spaces Python 3.12 (OMS)
28°C Haze
Search
ENG US
6:11 PM 2023-12-22

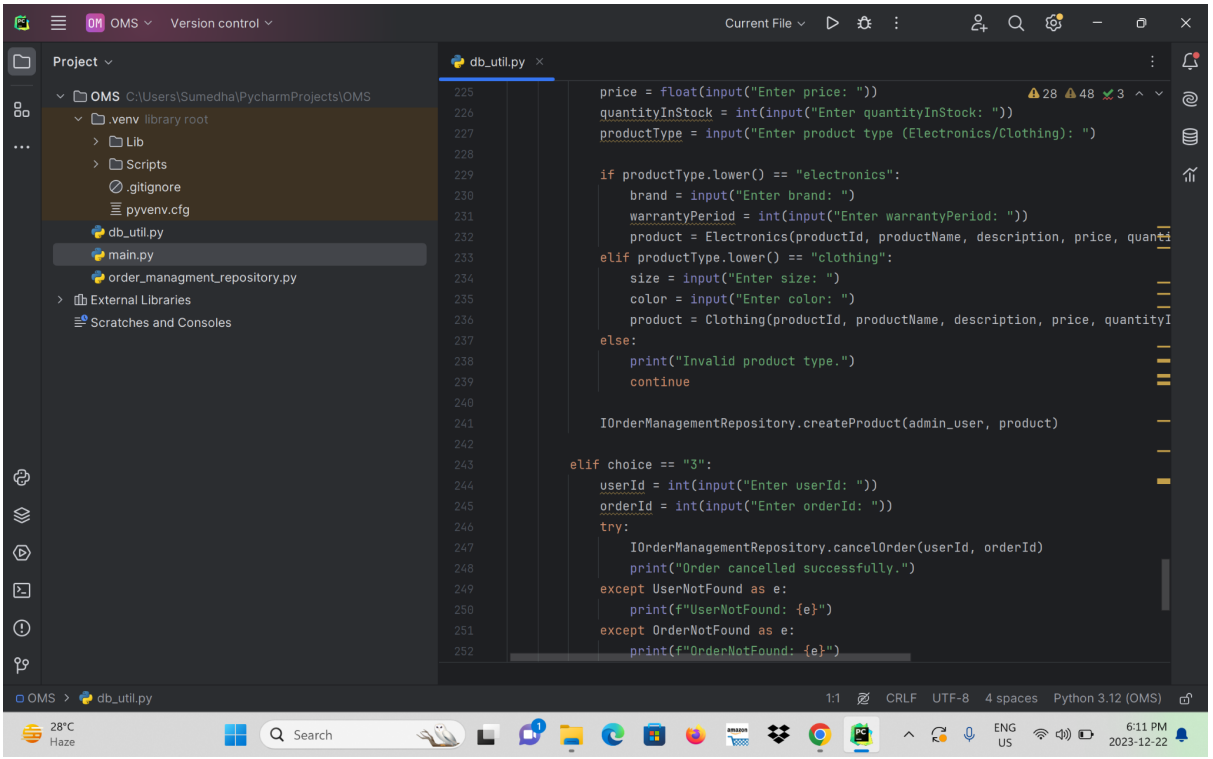

OMS
Version control
Current File
Project
db_util.py
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
db_util.py
main.py
order_managment_repository.py
External Libraries
Scratches and Consoles
Cancelling the order
cursor.execute(operation: "DELETE FROM ProductOrder WHERE userId = %s AND orde
connection.commit()
connection.close()
except UserNotFound as e:
print(f"UserNotFound: {e}")
raise
except OrderNotFound as e:
print(f"OrderNotFound: {e}")
raise
except Exception as e:
print(f"Error cancelling order: {e}")
raise
1 usage
@staticmethod
def createProduct(admin_user, product):
try:
# Checking here if the user exists in the database or not
connection = DBUtil.getDBConn()
cursor = connection.cursor()
cursor.execute(operation: "SELECT * FROM User WHERE userId = %s AND role = 'Ad
existing_admin = cursor.fetchone()
if existing_admin is None:
raise UserNotFound("Admin user not found")
OMS > db_util.py
1:1 CRLF UTF-8 4 spaces Python 3.12 (OMS)
28°C Haze
Search
ENG US
6:11 PM 2023-12-22

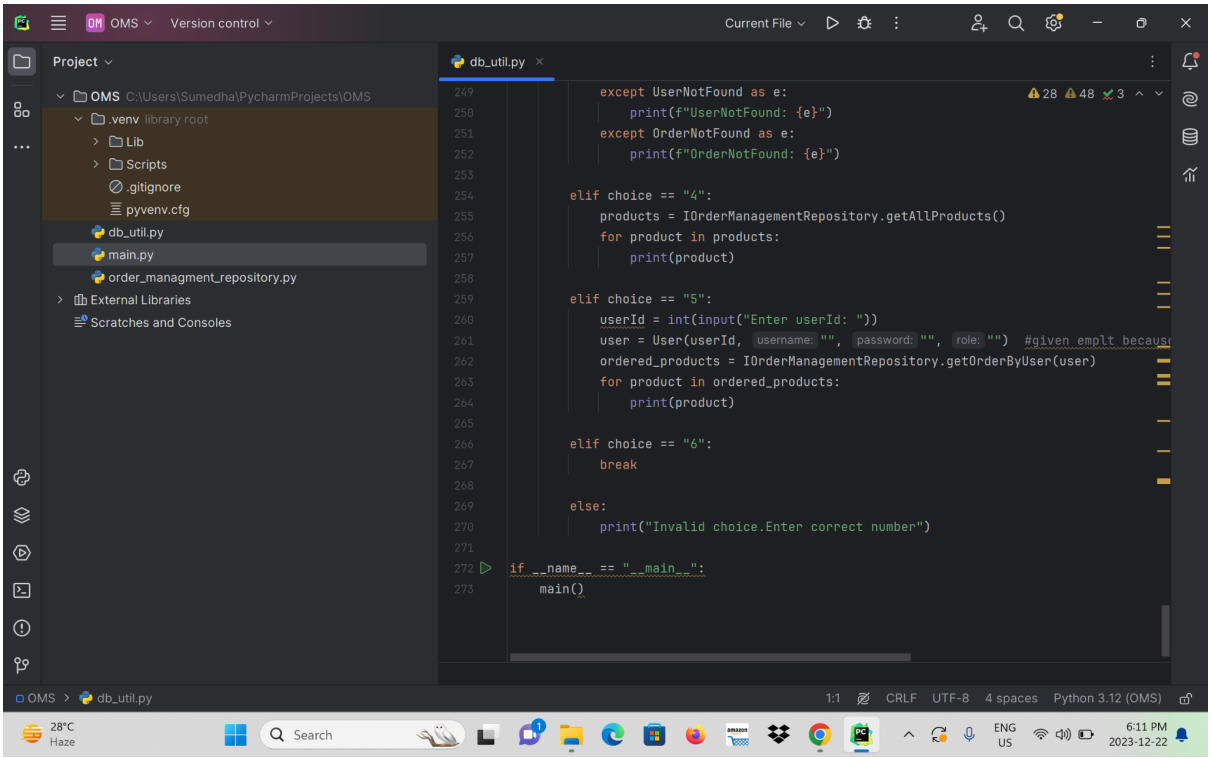

OMS
Version control
Current File
Project
db_util.py
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
db_util.py
main.py
order_managment_repository.py
External Libraries
Scratches and Consoles
raise UserNotFound("Admin user not found")
Creating the product
cursor.execute(operation: "INSERT INTO Product (productId, productName, descri
"VALUES (%s, %s, %s, %s, %s, %s)",
params: (product.productId, product.productName, product.desc
product.quantityInStock, product.type))
connection.commit()
# Checking if it is a electronics product or a clothing product
if product.type == 'Electronics':
cursor.execute(operation: "INSERT INTO Electronics (productId, brand, warr
"VALUES (%s, %s, %s)",
params: (product.productId, product.brand, product.warra
elif product.type == 'Clothing':
cursor.execute(operation: "INSERT INTO Clothing (productId, size, color) "
"VALUES (%s, %s, %s)",
params: (product.productId, product.size, product.color))
connection.commit()
connection.close()
except UserNotFound as e:
print(f"UserNotFound: {e}")
raise
except Exception as e:
print(f"Error creating product: {e}")
raise
OMS > db_util.py
1:1 CRLF UTF-8 4 spaces Python 3.12 (OMS)
28°C Haze
Search
ENG US
6:11 PM 2023-12-22

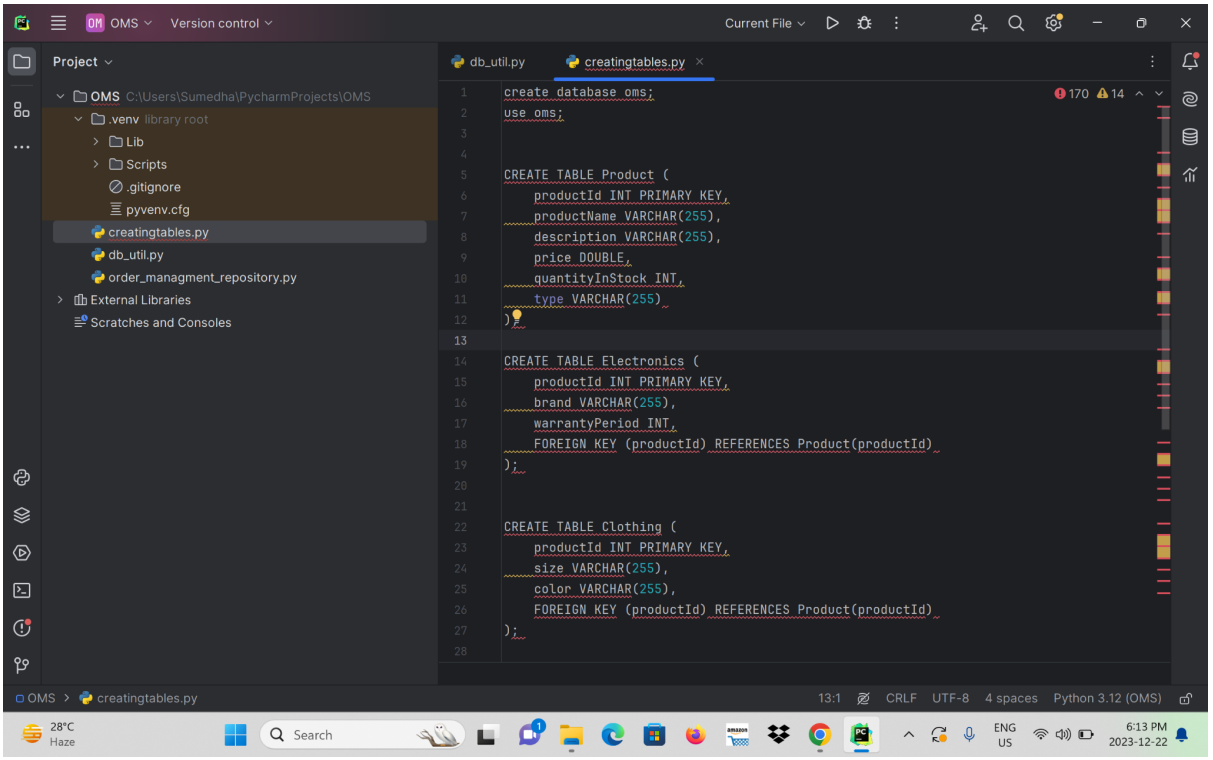

OMS
Version control
Current File
Project
db_util.py
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
db_util.py
main.py
order_managment_repository.py
External Libraries
Scratches and Consoles
28 48 3
1 usage
@staticmethod
def createUser(user):
try:
# Creating the user
conn = DBUtil.getDBConn()
cursor = conn.cursor()
cursor.execute(operation: "INSERT INTO User (userId, username, password, role)
params: (user.userId, user.username, user.password, user.role
conn.commit()
conn.close()
except Exception as e:
print(f"Error creating user: {e}")
raise
1 usage
@staticmethod
def getAllProducts():
try:
# getting all products from the database
conn = DBUtil.getDBConn()
cursor = conn.cursor(dictionary=True)
cursor.execute("SELECT * FROM Product")
products = cursor.fetchall()
conn.close()
return products
OMS > db_util.py
1:1 CRLF UTF-8 4 spaces Python 3.12 (OMS)
28°C Haze
Search
ENG US
6:11 PM 2023-12-22

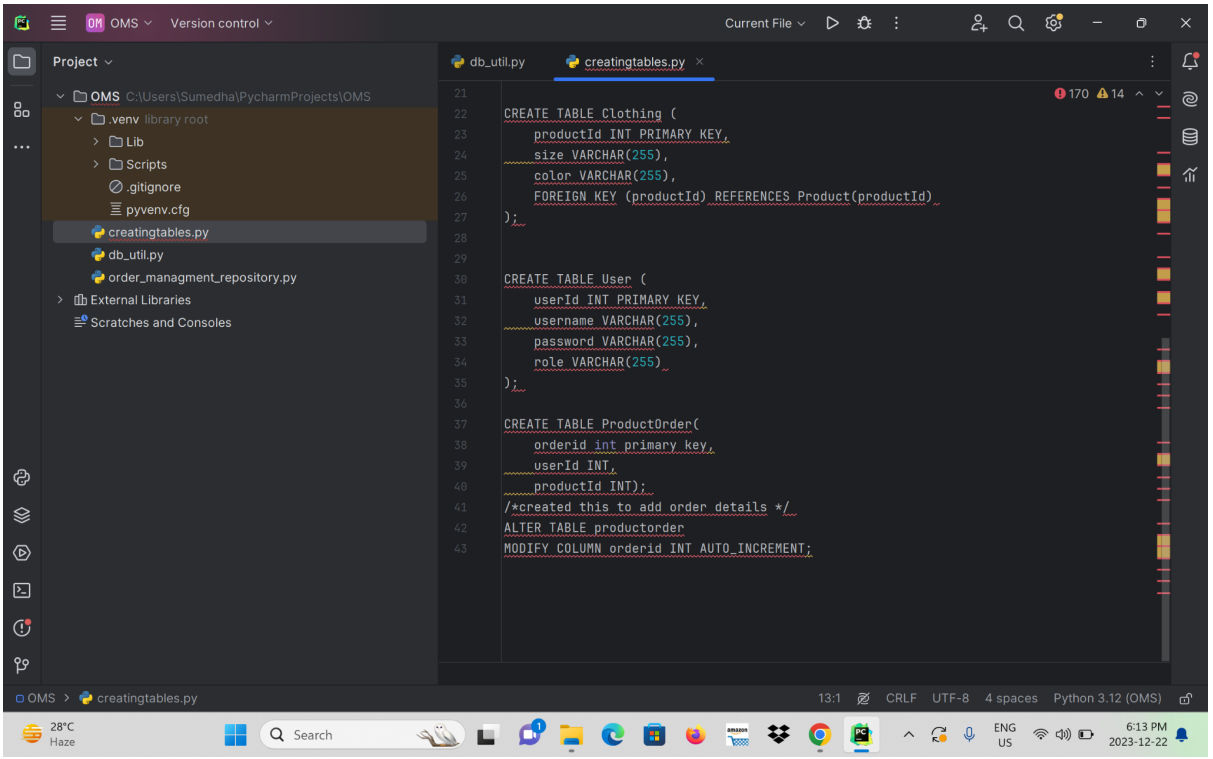

OMS
Version control
Current File
Project
db_util.py
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
db_util.py
main.py
order_managment_repository.py
External Libraries
Scratches and Consoles
return products
28 48 3
except Exception as e:
print(f"Error getting all products: {e}")
raise
1 usage
@staticmethod
def getOrderByUser(user):
try:
# getting all products ordered by a specific user
conn = DBUtil.getDBConn()
cursor = conn.cursor(dictionary=True)
cursor.execute(operation: "SELECT p.* FROM Product p "
"JOIN ProductOrder po ON p.productId = po.productId "
"WHERE po.userId = %s",
params: (user.userId,))
ordered_products = cursor.fetchall()
conn.close()
return ordered_products
except Exception as e:
print(f"Error getting ordered products: {e}")
raise
class OrderProcessor(IOrderManagementRepository):
pass
1 usage
def main():
OMS > db_util.py
1:1 CRLF UTF-8 4 spaces Python 3.12 (OMS)
28°C Haze
Search
ENG US
6:11 PM 2023-12-22


OMS
Version control
Current File
Project
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
db_util.py
main.py
order_managment_repository.py
External Libraries
Scratches and Consoles
db_util.py
def main():
while True:
print("Menu:")
print("1: createUser")
print("2: createProduct")
print("3: cancelOrder")
print("4: getAllProducts")
print("5: getOrderByUser")
print("6: exit/end")
choice = input("Enter your choice: ")
if choice == "1":
userId = int(input("Enter userId: "))
username = input("Enter username: ")
password = input("Enter password: ")
role = input("Enter role: ")
user = User(userId, username, password, role)
IOrderManagementRepository.createUser(user)
elif choice == "2":
# Assuming admin_user is already created and available
admin_user = User(userId: 1, username: "rohan", password: "123", role: "Admin")
productId = int(input("Enter productId: "))
productName = input("Enter productName: ")
description = input("Enter description: ")
price = float(input("Enter price: "))
28 48 3
OMS > db_util.py
1:1 CRLF UTF-8 4 spaces Python 3.12 (OMS)
28°C Haze
Search
ENG US
6:11 PM 2023-12-22


OMS
Version control
Current File
Project
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
db_util.py
main.py
order_managment_repository.py
External Libraries
Scratches and Consoles
db_util.py
price = float(input("Enter price: "))
quantityInStock = int(input("Enter quantityInStock: "))
productType = input("Enter product type (Electronics/Clothing): ")
if productType.lower() == "electronics":
brand = input("Enter brand: ")
warrantyPeriod = int(input("Enter warrantyPeriod: "))
product = Electronics(productId, productName, description, price, quanti
elif productType.lower() == "clothing":
size = input("Enter size: ")
color = input("Enter color: ")
product = Clothing(productId, productName, description, price, quantityI
else:
print("Invalid product type.")
continue
IOrderManagementRepository.createProduct(admin_user, product)
elif choice == "3":
userId = int(input("Enter userId: "))
orderId = int(input("Enter orderId: "))
try:
IOrderManagementRepository.cancelOrder(userId, orderId)
print("Order cancelled successfully.")
except UserNotFound as e:
print(f"UserNotFound: {e}")
except OrderNotFound as e:
print(f"OrderNotFound: {e}")
28 48 3
OMS > db_util.py
1:1 CRLF UTF-8 4 spaces Python 3.12 (OMS)
28°C Haze
Search
ENG US
6:11 PM 2023-12-22


OMS
Version control
Current File
Project
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
db_util.py
main.py
order_managment_repository.py
External Libraries
Scratches and Consoles
db_util.py
except UserNotFound as e:
print(f"UserNotFound: {e}")
except OrderNotFound as e:
print(f"OrderNotFound: {e}")
elif choice == "4":
products = IOrderManagementRepository.getAllProducts()
for product in products:
print(product)
elif choice == "5":
userId = int(input("Enter userId: "))
user = User(userId, username: "", password: "", role: "") #given emplt becaus
ordered_products = IOrderManagementRepository.getOrderByUser(user)
for product in ordered_products:
print(product)
elif choice == "6":
break
else:
print("Invalid choice.Enter correct number")
if __name__ == "__main__":
main()
OMS
db_util.py
1:1
CRLF
UTF-8
4 spaces
Python 3.12 (OMS)
28°C
Haze
Search
ENG
US
6:11 PM
2023-12-22


OMS
Version control
Current File
Project
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
creatingtables.py
db_util.py
order_managment_repository.py
External Libraries
Scratches and Consoles
db_util.py
creatingtables.py
create database oms;
use oms;
CREATE TABLE Product (
productId INT PRIMARY KEY,
productName VARCHAR(255),
description VARCHAR(255),
price DOUBLE,
quantityInStock INT,
type VARCHAR(255)
);
CREATE TABLE Electronics (
productId INT PRIMARY KEY,
brand VARCHAR(255),
warrantyPeriod INT,
FOREIGN KEY (productId) REFERENCES Product(productId)
);
CREATE TABLE Clothing (
productId INT PRIMARY KEY,
size VARCHAR(255),
color VARCHAR(255),
FOREIGN KEY (productId) REFERENCES Product(productId)
);
170 14
OMS
creatingtables.py
13:1
CRLF
UTF-8
4 spaces
Python 3.12 (OMS)
28°C
Haze
Search
ENG
US
6:13 PM
2023-12-22


OMS
Version control
Current File
Project
OMS C:\Users\Sumedha\PycharmProjects\OMS
.venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
creatingtables.py
db_util.py
order_managment_repository.py
External Libraries
Scratches and Consoles
db_util.py
creatingtables.py
CREATE TABLE Clothing (
productId INT PRIMARY KEY,
size VARCHAR(255),
color VARCHAR(255),
FOREIGN KEY (productId) REFERENCES Product(productId)
);
CREATE TABLE User (
userId INT PRIMARY KEY,
username VARCHAR(255),
password VARCHAR(255),
role VARCHAR(255)
);
CREATE TABLE ProductOrder(
orderid int primary key,
userId INT,
productId INT);
/*created this to add order details */
ALTER TABLE productorder
MODIFY COLUMN orderid INT AUTO_INCREMENT;
170 14
OMS
creatingtables.py
13:1
CRLF
UTF-8
4 spaces
Python 3.12 (OMS)
28°C
Haze
Search
ENG
US
6:13 PM
2023-12-22