Assignment 1

Task 1: Classes and Their Attributes:

Task 2: Class Creation:

• Create the classes (Customers, Products, Orders, OrderDetails and Inventory) with the specified attributes.

• Implement the constructor for each class to initialize its attributes.

• Implement methods as specified.

Customers Class:

Attributes:

• CustomerID (int)

• FirstName (string)

• LastName (string)

• Email (string)

• Phone (string)

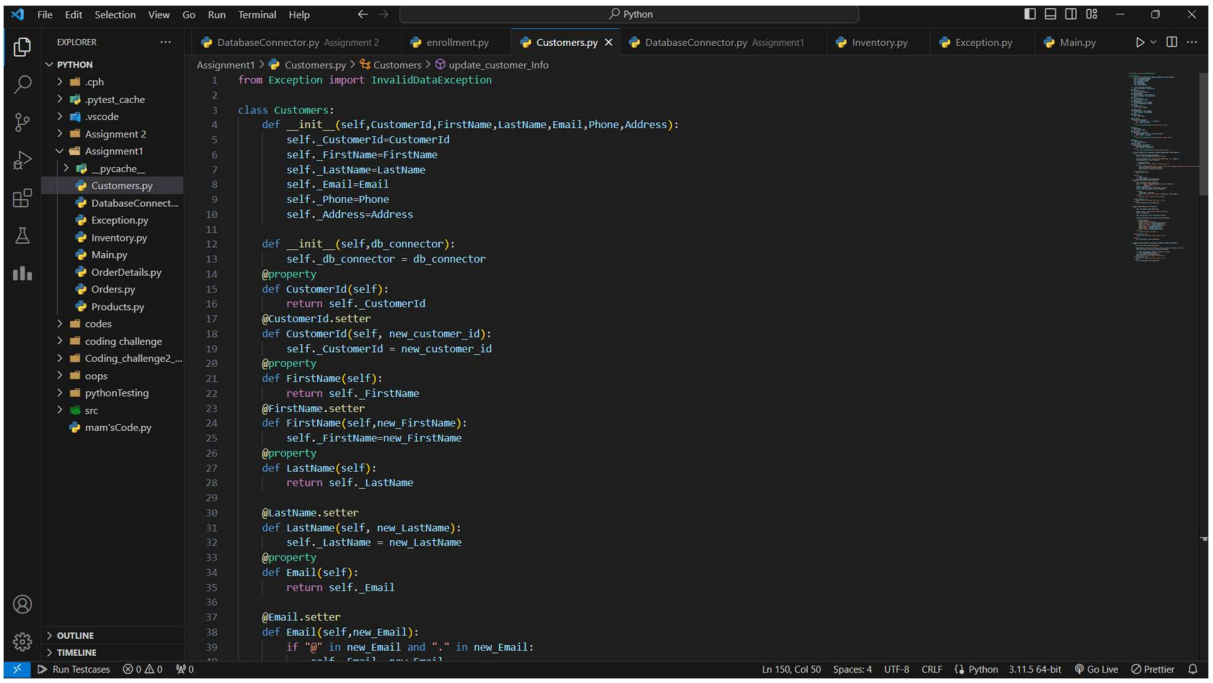
• Address (string)

Methods:

• CalculateTotalOrders(): Calculates the total number of orders placed by this customer.

• GetCustomerDetails(): Retrieves and displays detailed information about the customer.

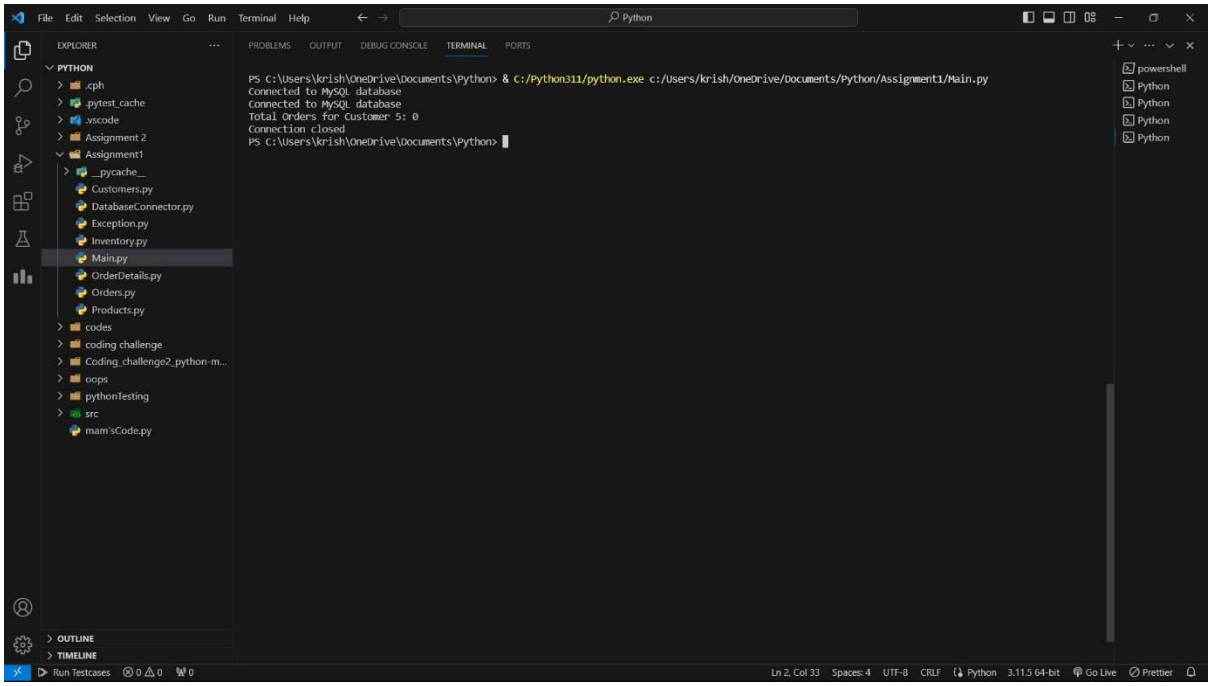
• UpdateCustomerInfo(): Allows the customer to update their information (e.g., email, phone, or address).

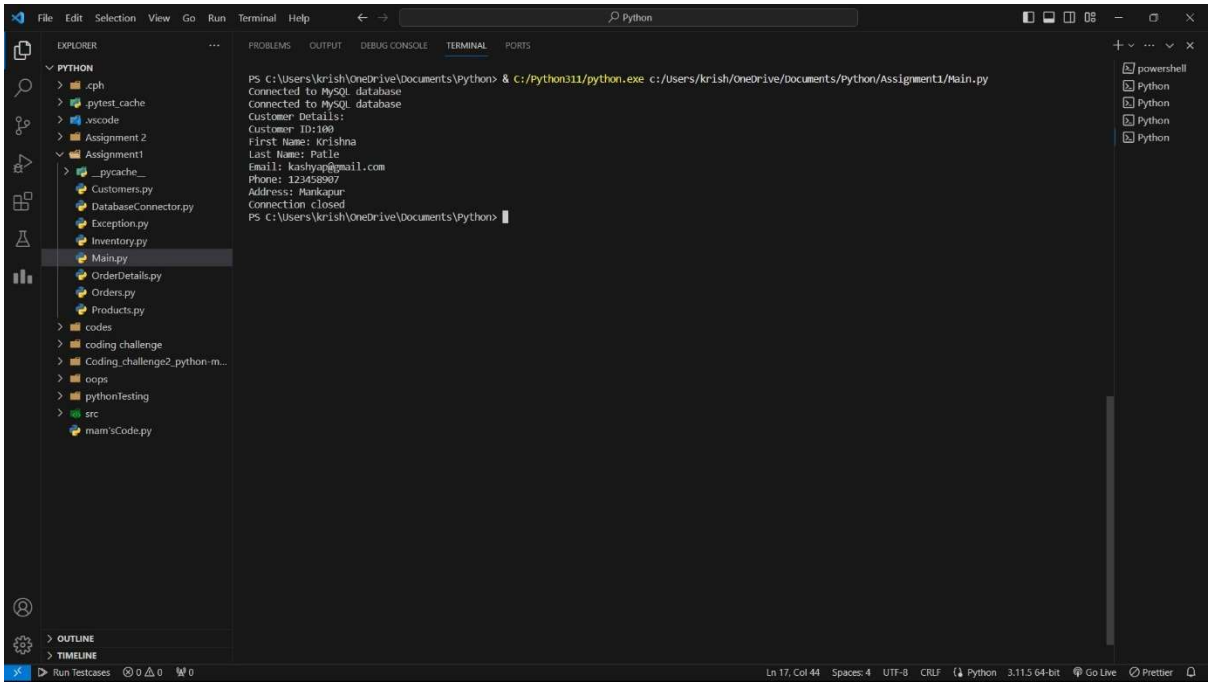


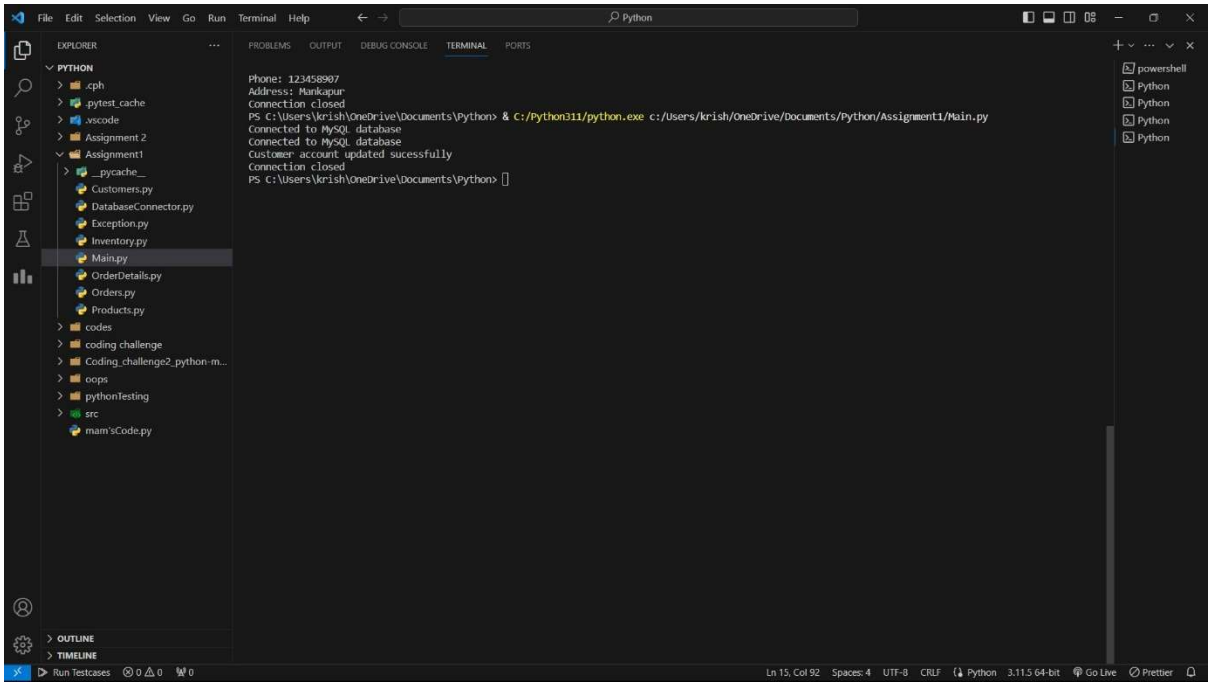

```python
    @Email.setter
    def Email(self,new_Email):
        if "@" in new_Email and "." in new_Email:
            self._Email =new_Email
        else:
            raise InvalidDataException("Invalid email format.")

    @property
    def Phone(self):
        return self._Phone
    @Phone.setter
    def Phone(self, new_Phone):
        if len(new_Phone) == 10 and new_Phone.isdigit():
            self._Phone = new_Phone
        else:
            raise InvalidDataException("Invalid phone number format.")

    @property
    def Address(self):
        return self._Address
    @Address.setter
    def Address(self, new_Address):
        if isinstance(new_Address, str):
            self._Address = new_Address
        else:
            raise InvalidDataException("Invalid address format")

    def create_customer(self,CustomerId,FirstName,LastName,Email, Phone,Address):
        try:
            self._db_connector.open_connection()
            cursor = self._db_connector.connection.cursor()

            cursor.execute("SELECT * FROM customers WHERE Email = %s", (Email,))
            existing_customer = cursor.fetchone()

            if existing_customer:
                print("Error: Email address is already in use.")
            else:
                cursor.execute("INSERT INTO customers (CustomerId,FirstName,LastName, Email, Phone,Address) VALUES (%s, %s, %s,%s,%s,%s)", (Custom
```

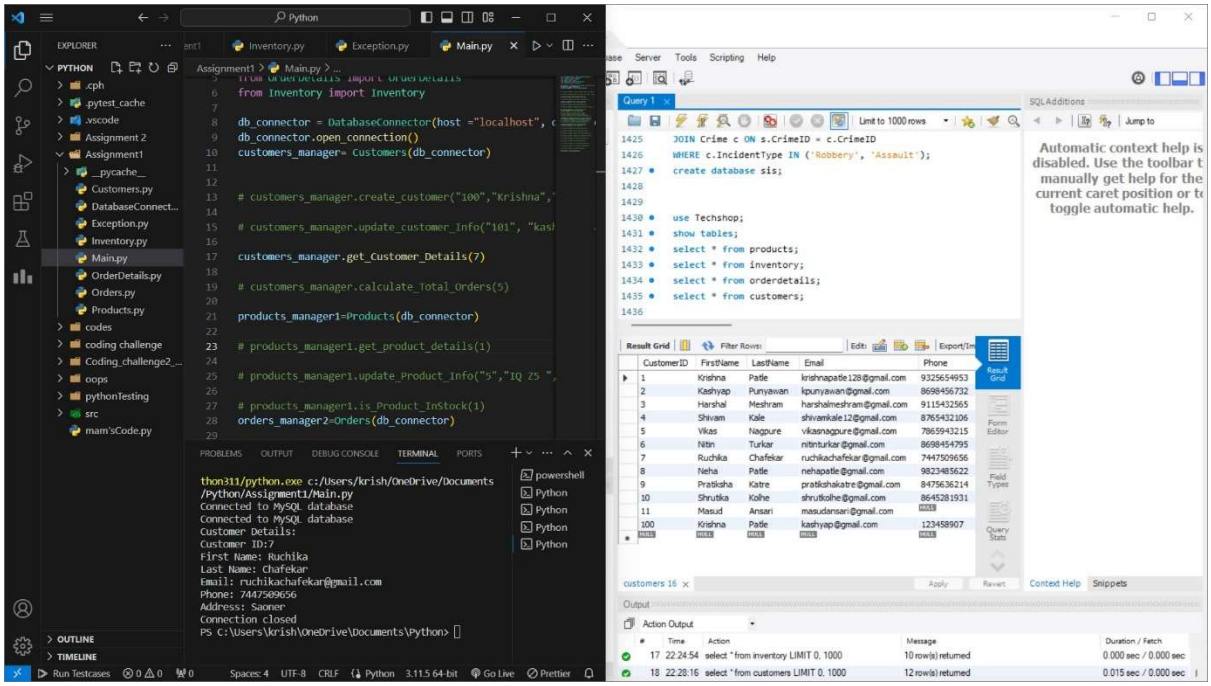
```
            print("Error: Email address is already in use.")
        else:
            cursor.execute("INSERT INTO customers (CustomerId,FirstName,LastName, Email, Phone,Address) VALUES (%s, %s, %s,%s,%s,%s)", (Custome
            self._db_connector.connection.commit()
            print("Customer created successfully")

    except Exception as e:
        print(f"Error: {e}")

    finally:
        if cursor:
            cursor.close()
        self._db_connector.close_connection()
def calculate_Total_Orders(self,Customerid):
    try:
        self._db_connector.open_connection()
        query = """ SELECT COUNT(OrderID) AS TotalOrders FROM Orders
                    WHERE Customerid = %s"""
        values = (Customerid,)
        self._db_connector.cursor.execute(query, values)
        result = self._db_connector.cursor.fetchone()

        if result:
            total_orders = result[0]
            print(f"Total Orders for Customer {Customerid}: {total_orders}")
        else:
            print("No orders found for this customer")

    except Exception as e:
        print(f"Error calculating total orders: {e}")
    finally:
        self._db_connector.close_connection()


def get_Customer_Details(self,Customerid):
    try:
        self._db_connector.open_connection()

        query ="SELECT * FROM customers WHERE CustomerId=%s"
```

```
            self._db_connector.cursor.execute(query,values)

            customer_details=self._db_connector.cursor.fetchone()

            if customer_details:
                print("Customer Details:")
                print(f"Customer ID:{customer_details[0]}")
                print(f"First Name: {customer_details[1]}")
                print(f"Last Name: {customer_details[2]}")
                print(f"Email: {customer_details[3]}")
                print(f"Phone: {customer_details[4]}")
                print(f"Address: {customer_details[5]}")
            else:
                print("Customer not found.")

        except Exception as e:
            print(f"Error getting customer details: {e}")

        finally:
            self._db_connector.close_connection()


    def update_customer_Info(self,Customerid,new_Email,new_Phone,new_Address):
        try:
            self._db_connector.open_connection()

            query="UPDATE customers SET Email=%s,Phone=%s, Address=%s WHERE Customerid=%s"
            values=(new_Email,new_Phone,new_Address,Customerid)

            with self._db_connector.connection.cursor() as cursor:
                cursor.execute(query,values)
            self._db_connector.connection.commit()
            print("Customer account updated sucessfully")
        except Exception as e:
            print(f"Error updating customer account :{e}")
        finally:
            self._db_connector.close_connection()
```


File Edit Selection View Go Run Terminal Help
Python
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PYTHON
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnector.py
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_python-m...
oops
pythonTesting
src
mam'sCode.py
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Total Orders for Customer 5: 0
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
powershell
Python
Python
Python
Python
OUTLINE
TIMELINE
Run Testcases
Ln 2, Col 33 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier


File Edit Selection View Go Run Terminal Help
Python
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PYTHON
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnector.py
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_python-m...
oops
pythonTesting
src
mam'sCode.py
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Customer Details:
Customer ID:100
First Name: Krishna
Last Name: Patle
Email: kashyap@gmail.com
Phone: 123458907
Address: Mankapur
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
powershell
Python
Python
Python
Python
OUTLINE
TIMELINE
Run Testcases
Ln 17, Col 44 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier





Products Class:

Attributes:

- ProductID (int)
- ProductName (string)

• Description (string)

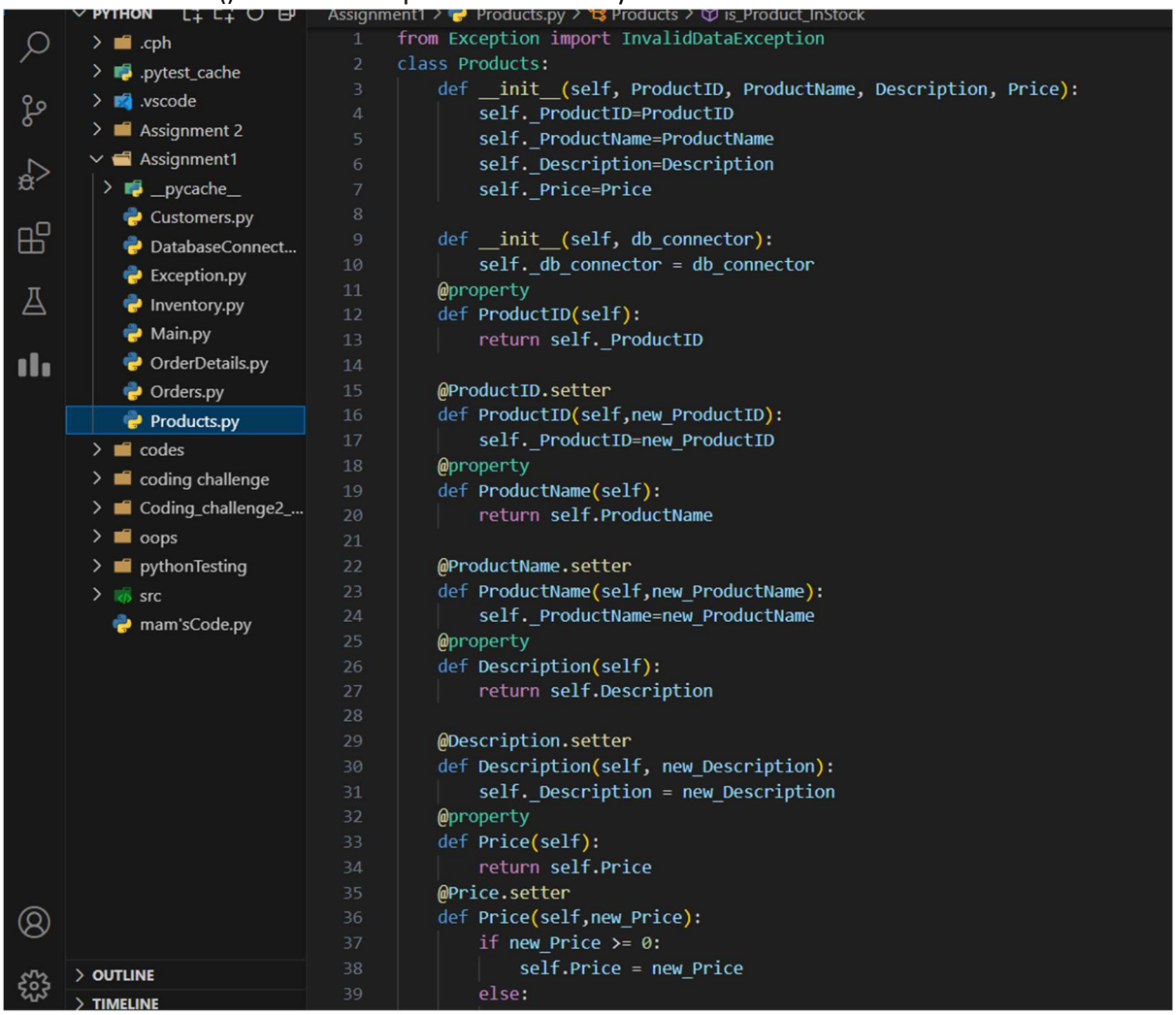
• Price (decimal)

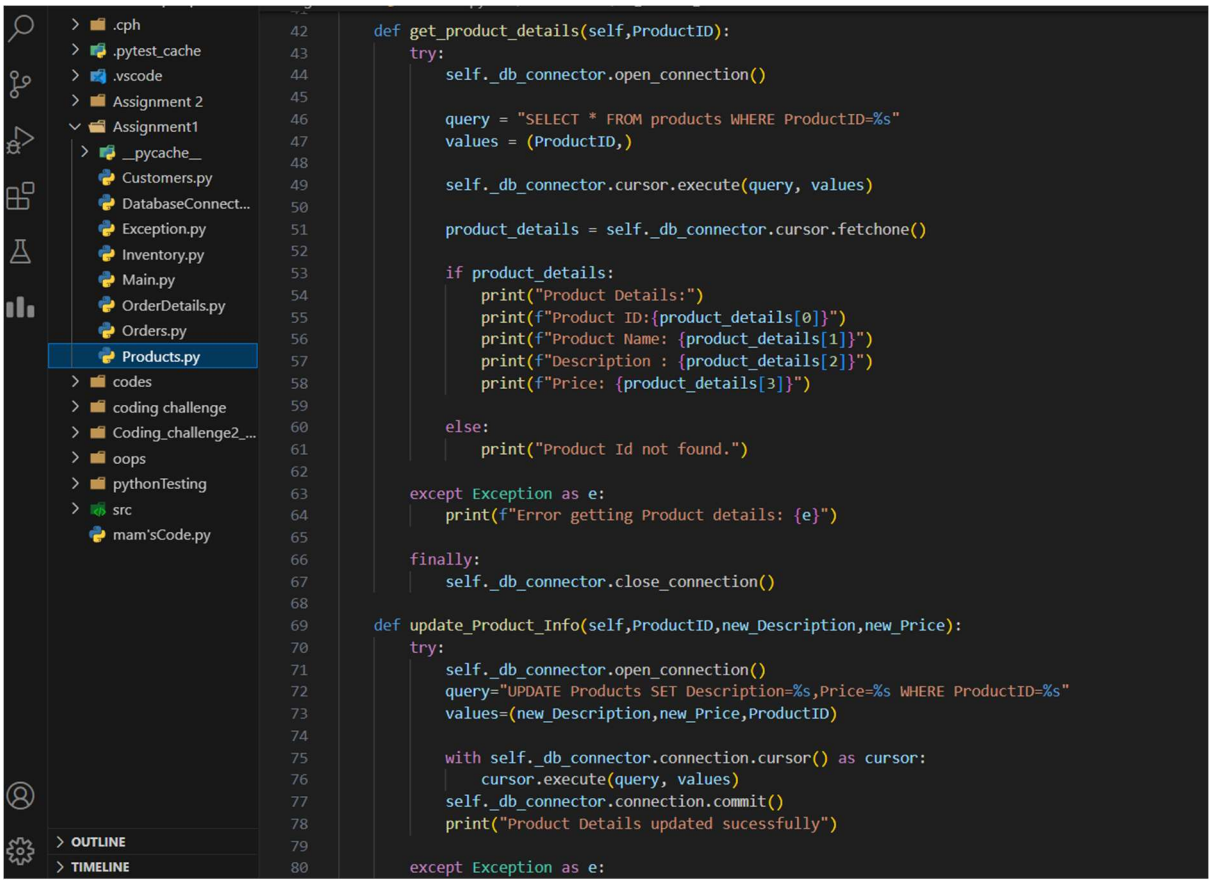
Methods:

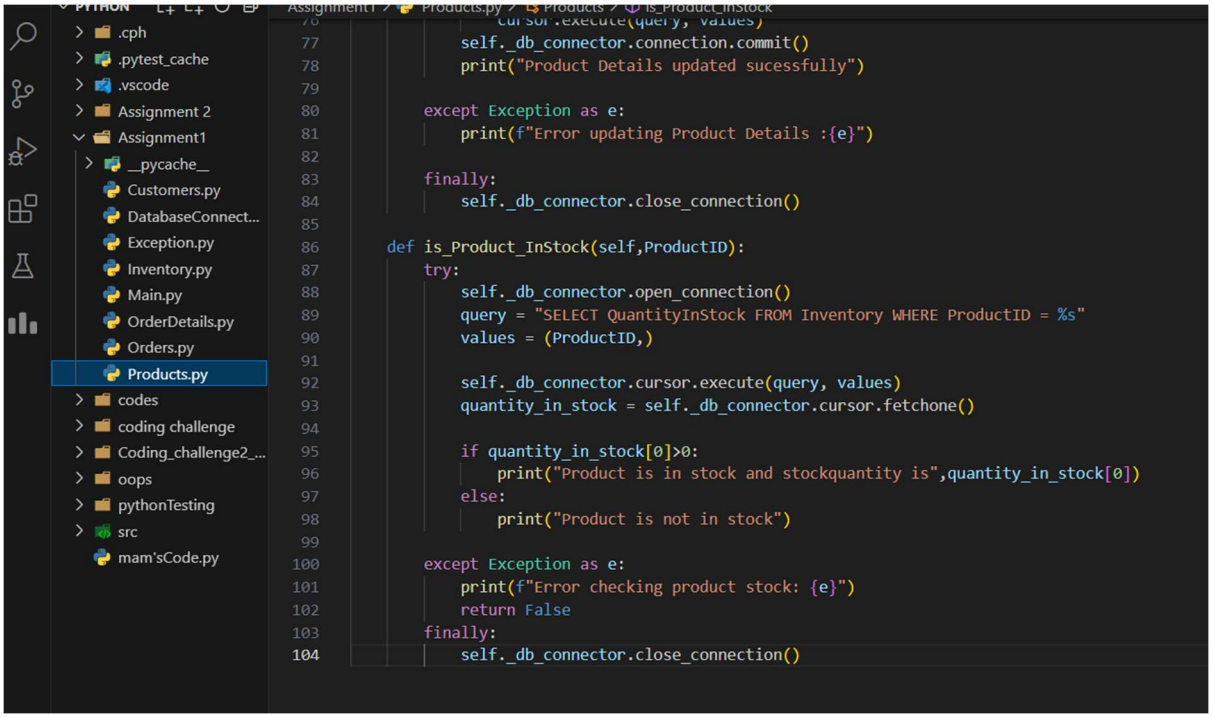
• GetProductDetails(): Retrieves and displays detailed information about the product.

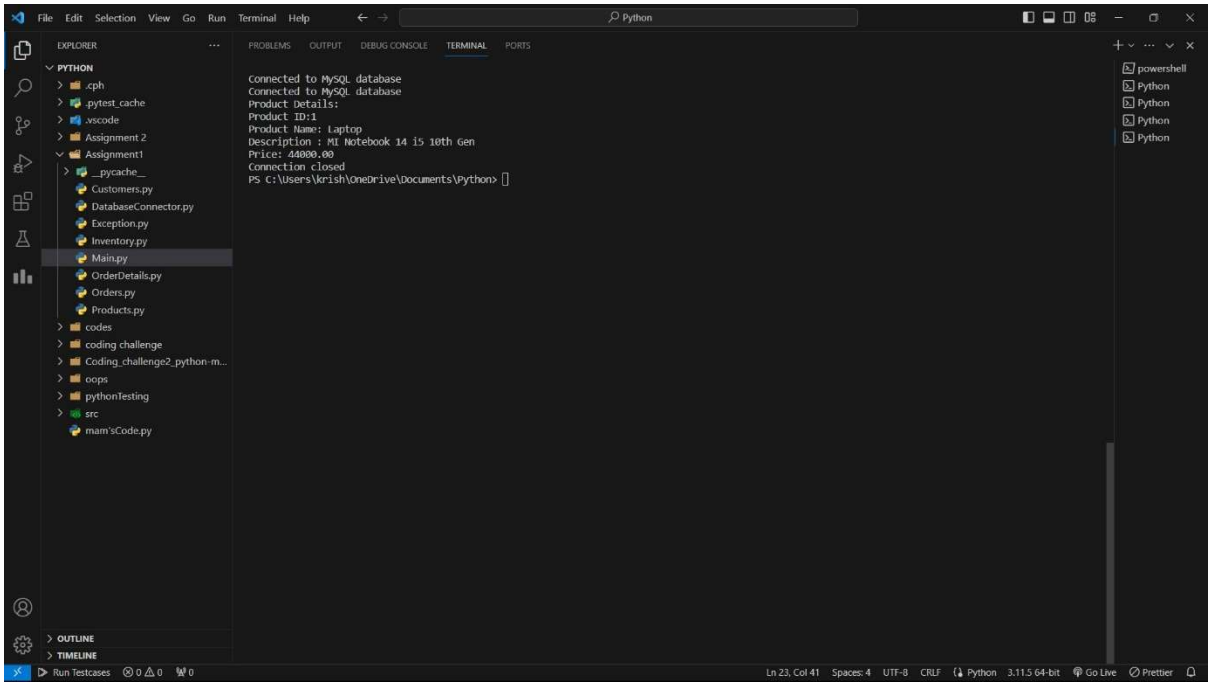
• UpdateProductInfo(): Allows updates to product details (e.g., price, description).

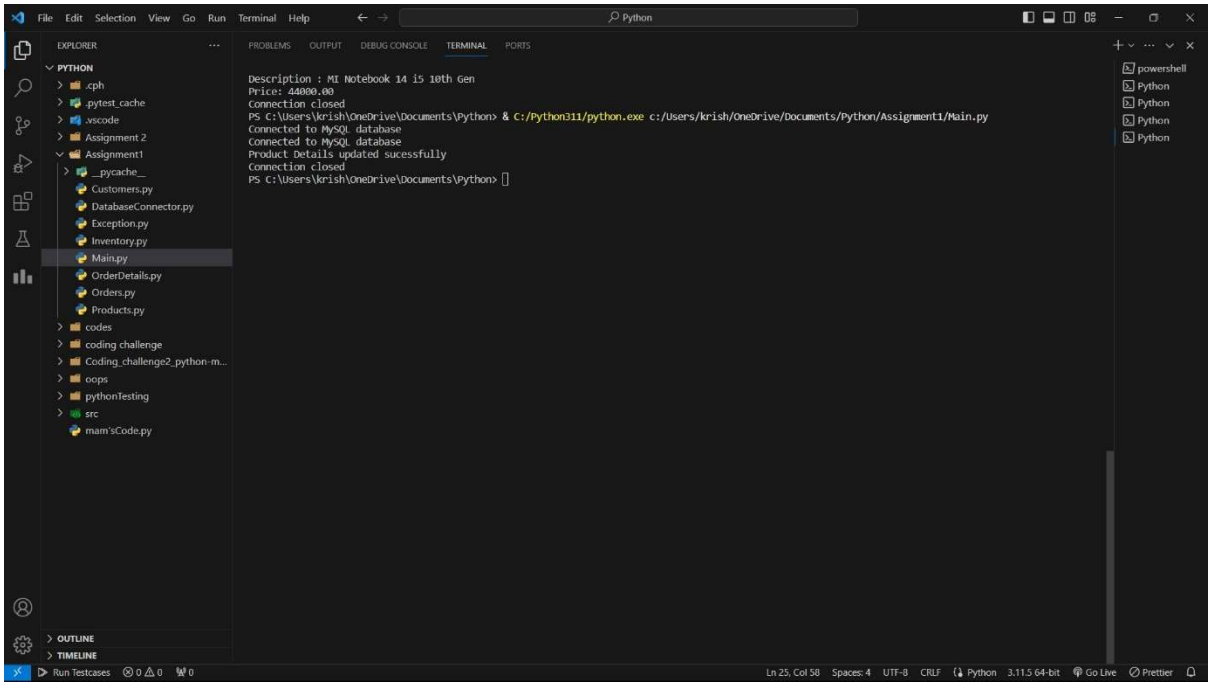
• IsProductInStock(): Checks if the product is currently in stock.

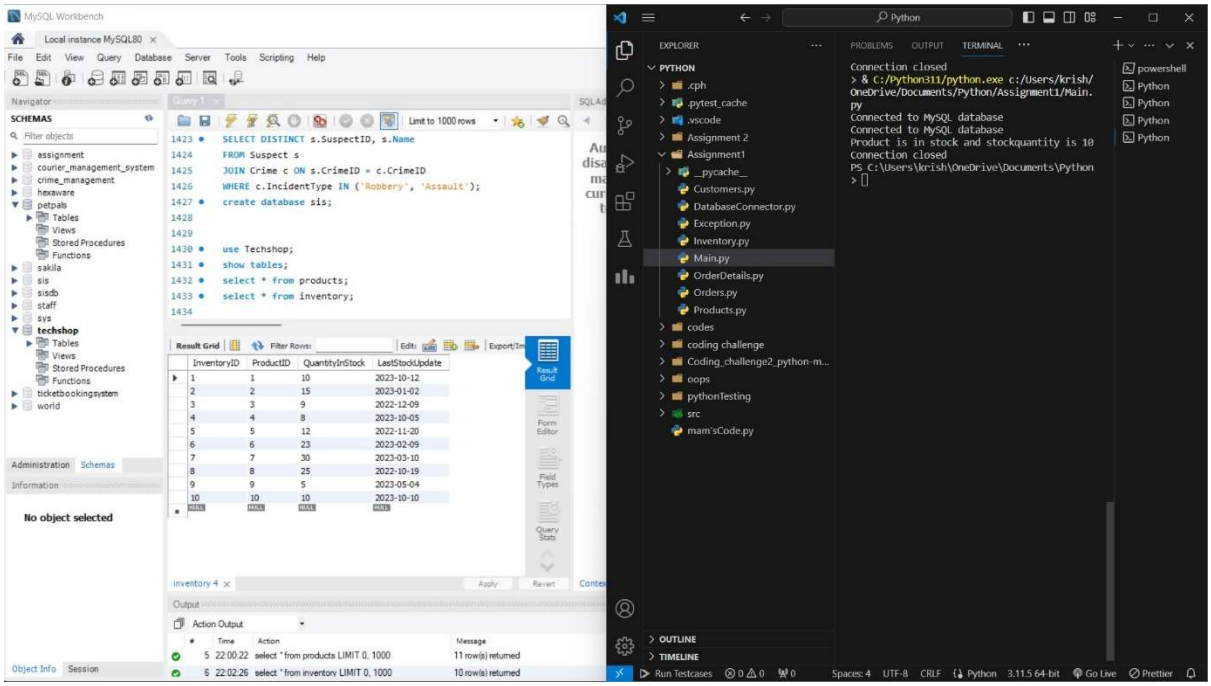


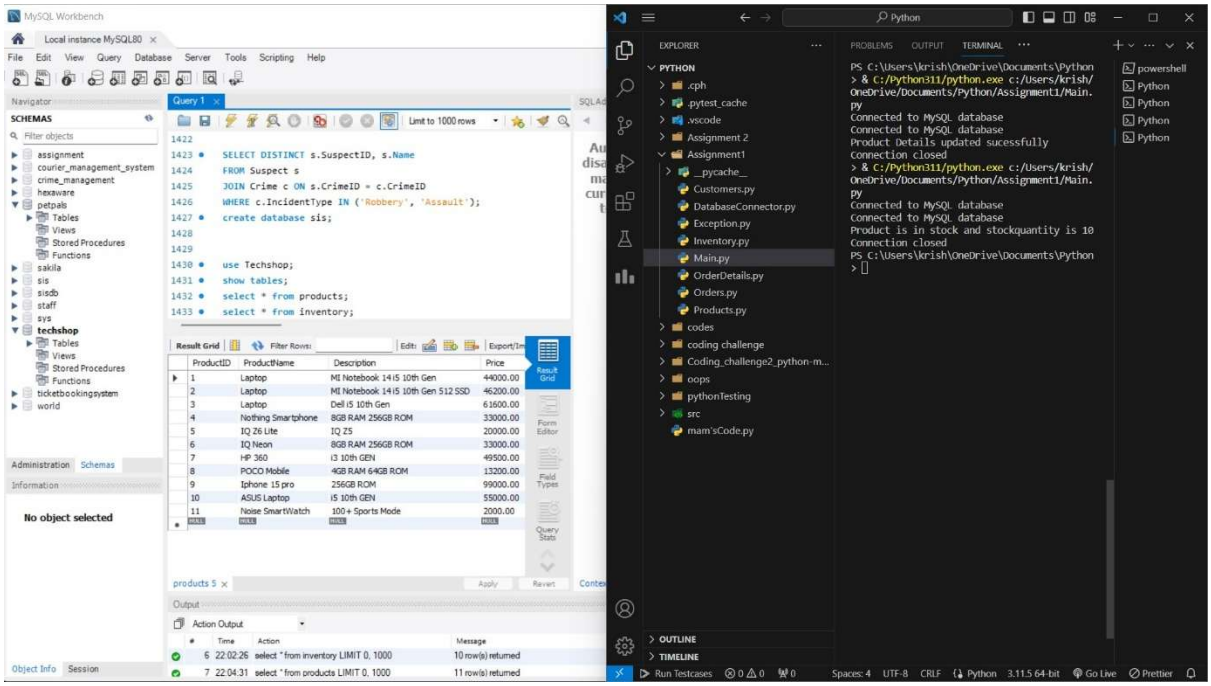

.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnect...
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
TIMELINE
def get_product_details(self,ProductID):
try:
self._db_connector.open_connection()
query = "SELECT * FROM products WHERE ProductID=%s"
values = (ProductID,)
self._db_connector.cursor.execute(query, values)
product_details = self._db_connector.cursor.fetchone()
if product_details:
print("Product Details:")
print(f"Product ID:{product_details[0]}")
print(f"Product Name: {product_details[1]}")
print(f"Description : {product_details[2]}")
print(f"Price: {product_details[3]}")
else:
print("Product Id not found.")
except Exception as e:
print(f"Error getting Product details: {e}")
finally:
self._db_connector.close_connection()
def update_Product_Info(self,ProductID,new_Description,new_Price):
try:
self._db_connector.open_connection()
query="UPDATE Products SET Description=%s,Price=%s WHERE ProductID=%s"
values=(new_Description,new_Price,ProductID)
with self._db_connector.connection.cursor() as cursor:
cursor.execute(query, values)
self._db_connector.connection.commit()
print("Product Details updated sucessfully")
except Exception as e:

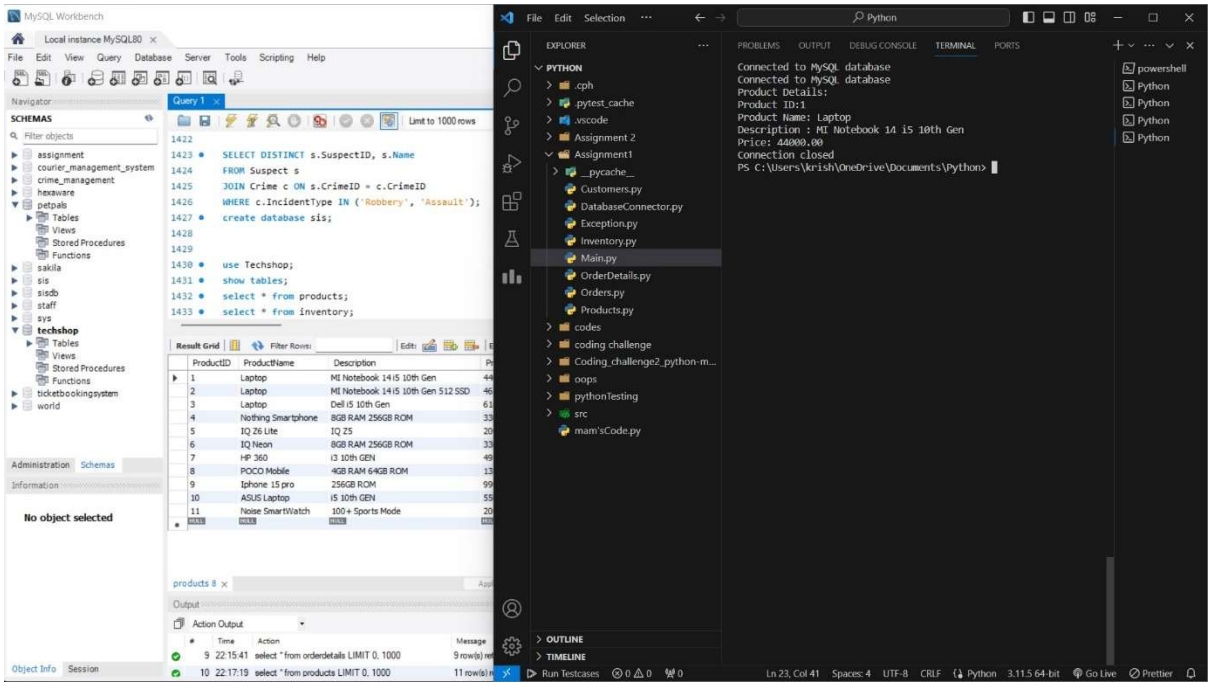

Assignment1 > Products.py > Products > is_Product_InStock
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnect...
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_...
oops
pythonTesting
src
mam'sCode.py
self._db_connector.connection.commit()
print("Product Details updated sucessfully")
except Exception as e:
print(f"Error updating Product Details :{e}")
finally:
self._db_connector.close_connection()
def is_Product_InStock(self,ProductID):
try:
self._db_connector.open_connection()
query = "SELECT QuantityInStock FROM Inventory WHERE ProductID = %s"
values = (ProductID,)
self._db_connector.cursor.execute(query, values)
quantity_in_stock = self._db_connector.cursor.fetchone()
if quantity_in_stock[0]>0:
print("Product is in stock and stockquantity is",quantity_in_stock[0])
else:
print("Product is not in stock")
except Exception as e:
print(f"Error checking product stock: {e}")
return False
finally:
self._db_connector.close_connection()


File Edit Selection View Go Run Terminal Help
Python
EXPLORER
PYTHON
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnector.py
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_python-m...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Connected to MySQL database
Connected to MySQL database
Product Details:
Product ID:1
Product Name: Laptop
Description : MI Notebook 14 i5 10th Gen
Price: 44000.00
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
powershell
Python
Python
Python
Python
Run Testcases
Ln 23, Col 41 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier


File Edit Selection View Go Run Terminal Help
Python
EXPLORER
PYTHON
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnector.py
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_python-m...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Description : MI Notebook 14 i5 10th Gen
Price: 44000.00
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Product Details updated sucessfully
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
powershell
Python
Python
Python
Python
Run Testcases
Ln 25, Col 58 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
assignment
courier_management_system
crime_management
hexaware
petpals
Tables
Views
Stored Procedures
Functions
sakila
sis
sisdb
staff
sys
techshop
ticketbookingsystem
world
1423 SELECT DISTINCT s.SuspectID, s.Name
1424 FROM Suspect s
1425 JOIN Crime c ON s.CrimeID = c.CrimeID
1426 WHERE c.IncidentType IN ('Robbery', 'Assault');
1427 create database sis;
1430 use Techshop;
1431 show tables;
1432 select * from products;
1433 select * from inventory;
Result Grid
InventoryID ProductID QuantityInStock LastStockUpdate
1 1 10 2023-10-12
2 2 15 2023-01-02
3 3 9 2022-12-09
4 4 8 2023-10-05
5 5 12 2022-11-20
6 6 23 2023-02-09
7 7 30 2023-03-10
8 8 25 2022-10-19
9 9 5 2023-05-04
10 10 10 2023-10-10
No object selected
Action Output
5 22:00:22 select * from products LIMIT 0, 1000 11 row(s) returned
6 22:02:26 select * from inventory LIMIT 0, 1000 10 row(s) returned
Python
EXPLORER
PROBLEMS OUTPUT TERMINAL
PYTHON
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnector.py
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_python-m...
oops
pythonTesting
src
mam'sCode.py
Connection closed
> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Product is in stock and stockquantity is 10
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python
powershell
Python
Python
Python
Python
OUTLINE
TIMELINE


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
1423 SELECT DISTINCT s.SuspectID, s.Name
1424 FROM Suspect s
1425 JOIN Crime c ON s.CrimeID = c.CrimeID
1426 WHERE c.IncidentType IN ('Robbery', 'Assault');
1427 create database sis;
1430 use Techshop;
1431 show tables;
1432 select * from products;
1433 select * from inventory;
Result Grid
ProductID ProductName Description Price
1 Laptop MI Notebook 14 i5 10th Gen 44000.00
2 Laptop MI Notebook 14 i5 10th Gen 512 SSD 46200.00
3 Laptop Dell i5 10th Gen 61600.00
4 Nothing Smartphone 8GB RAM 256GB ROM 33000.00
5 IQ Z6 Lite IQ Z5 20000.00
6 IQ Neon 8GB RAM 256GB ROM 33000.00
7 HP 360 i3 10th GEN 49500.00
8 POCO Mobile 4GB RAM 64GB ROM 13200.00
9 Iphone 15 pro 256GB ROM 99000.00
10 ASUS Laptop i5 10th GEN 55000.00
11 Noise SmartWatch 100+ Sports Mode 2000.00
No object selected
Action Output
6 22:02:26 select * from inventory LIMIT 0, 1000 10 row(s) returned
7 22:04:31 select * from products LIMIT 0, 1000 11 row(s) returned
Python
PS C:\Users\krish\OneDrive\Documents\Python
> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Product Details updated sucessfully
Connection closed
> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Product is in stock and stockquantity is 10
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python



Orders Class:

Attributes:

• OrderID (int)

• Customer (Customer) - Use composition to reference the Customer who placed the order.

• OrderDate (DateTime)

• TotalAmount (decimal)

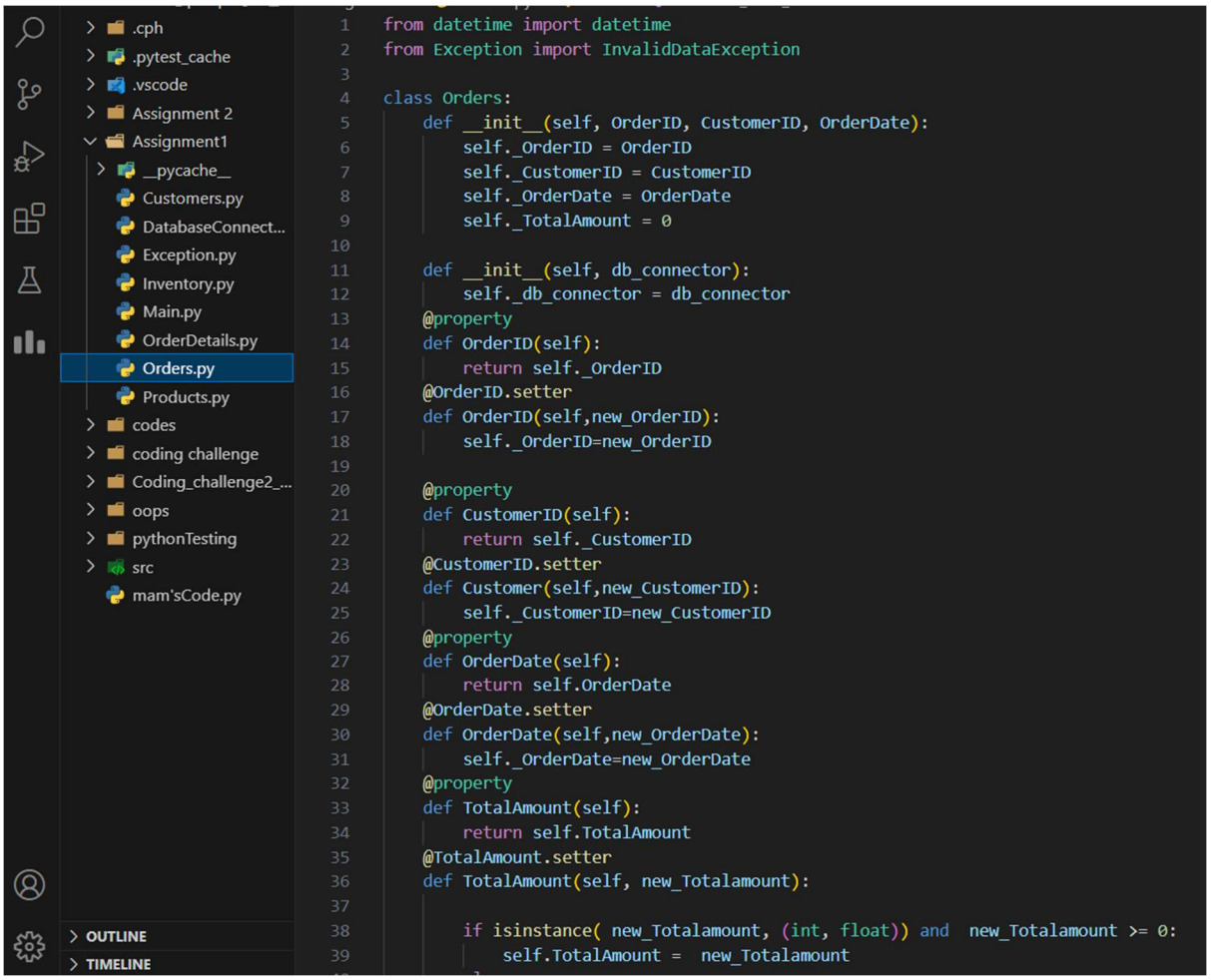
Methods:

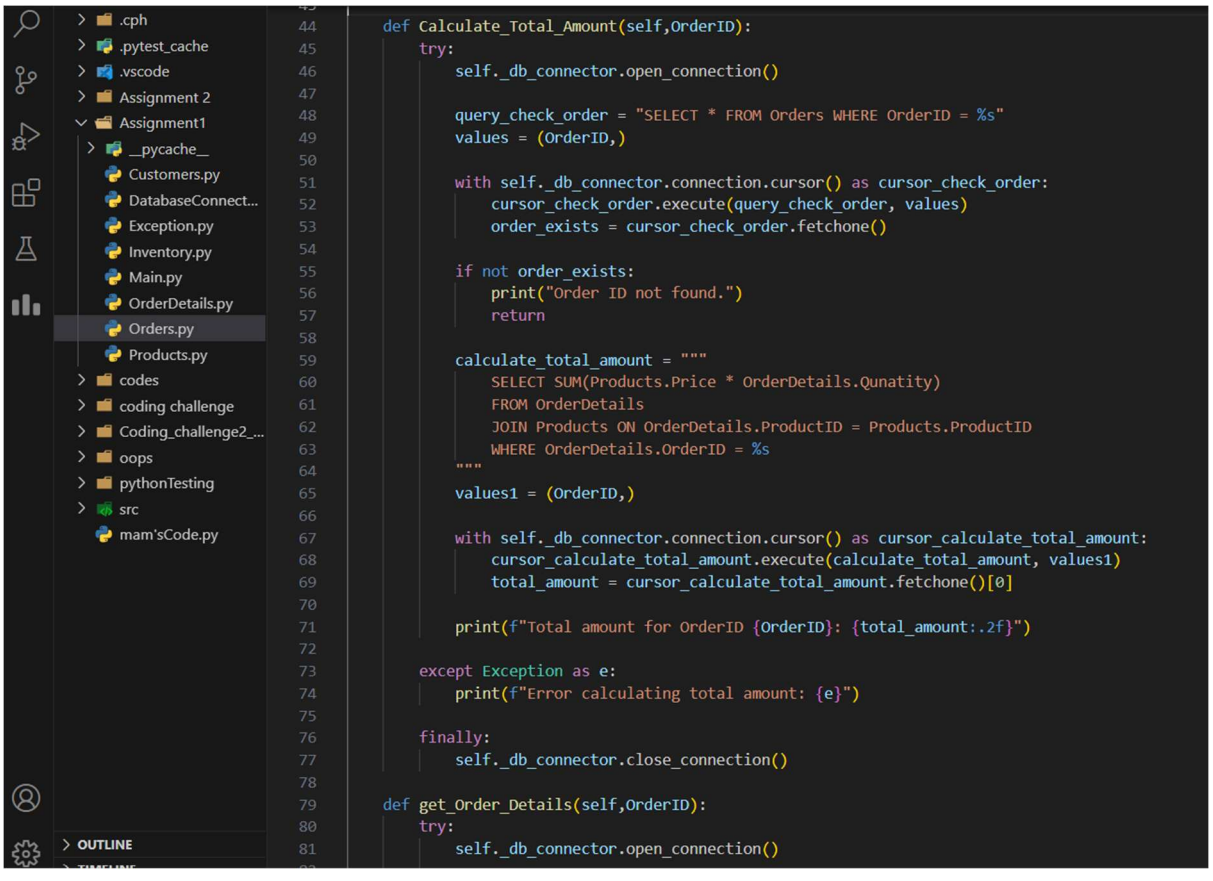
• CalculateTotalAmount() - Calculate the total amount of the order.

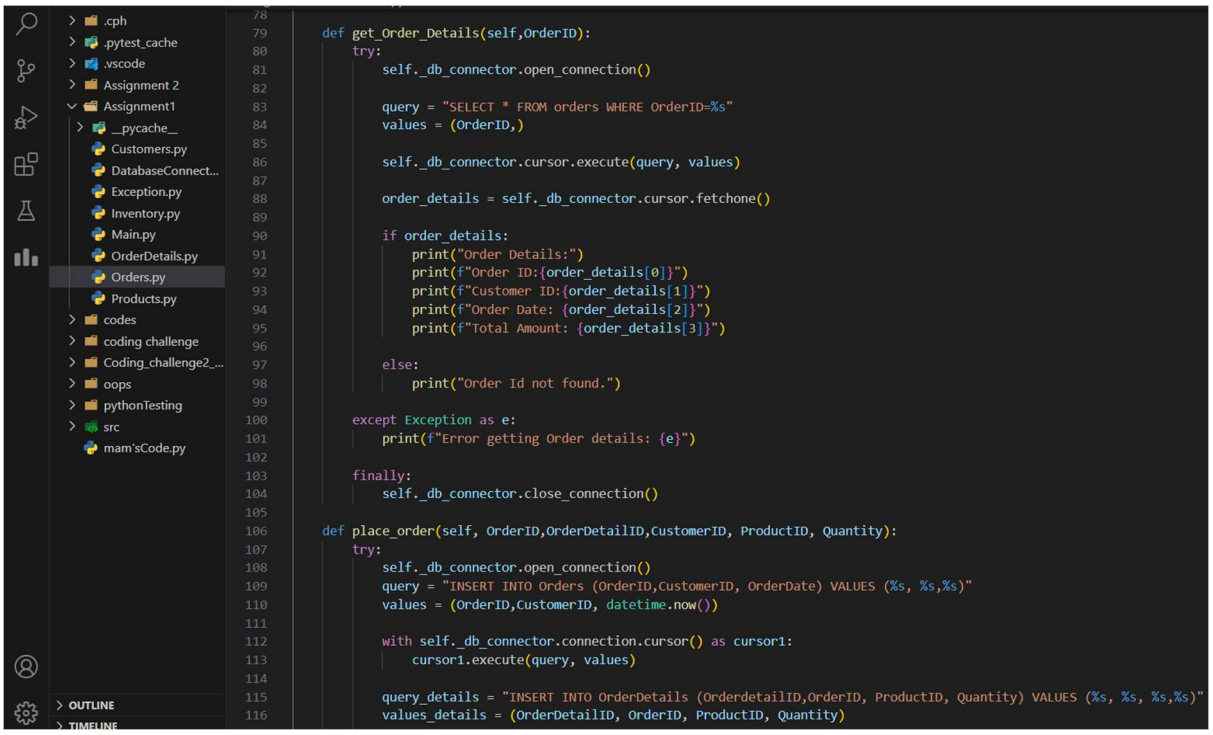
• GetOrderDetails(): Retrieves and displays the details of the order (e.g., product list and quantities).

• UpdateOrderStatus(): Allows updating the status of the order (e.g., processing, shipped).

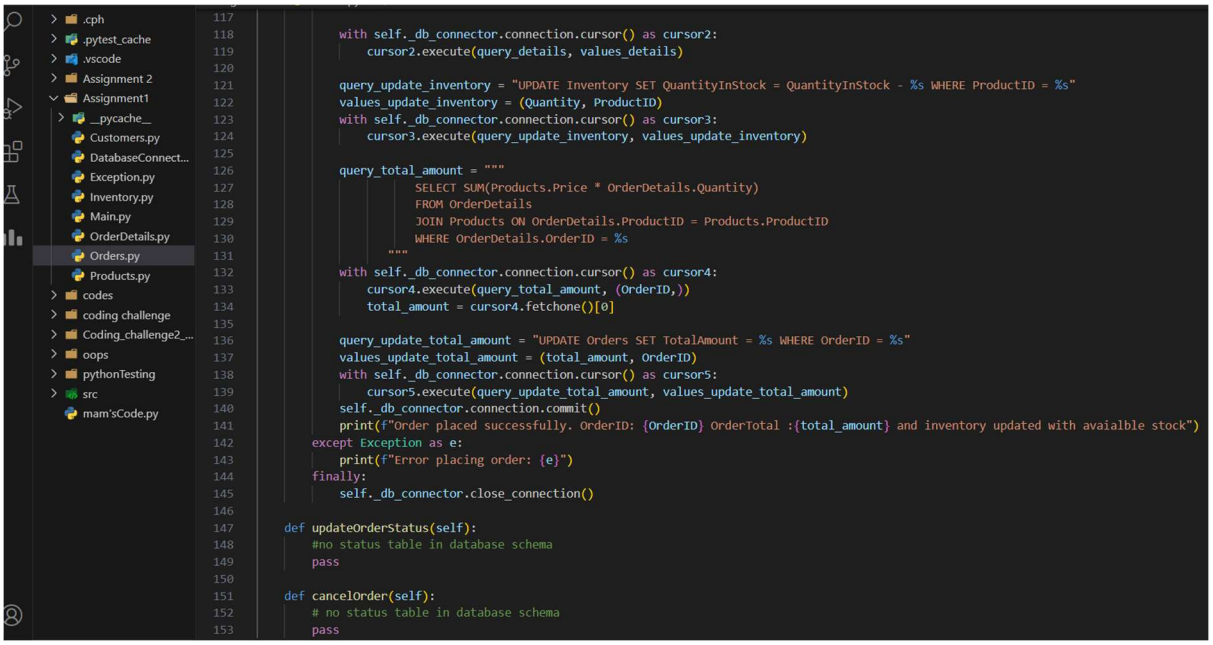
• CancelOrder(): Cancels the order and adjusts stock levels for products.


.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnect...
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
TIMELINE
from datetime import datetime
from Exception import InvalidDataException
class Orders:
def __init__(self, OrderID, CustomerID, OrderDate):
self._OrderID = OrderID
self._CustomerID = CustomerID
self._OrderDate = OrderDate
self._TotalAmount = 0
def __init__(self, db_connector):
self._db_connector = db_connector
@property
def OrderID(self):
return self._OrderID
@OrderID.setter
def OrderID(self,new_OrderID):
self._OrderID=new_OrderID
@property
def CustomerID(self):
return self._CustomerID
@CustomerID.setter
def Customer(self,new_CustomerID):
self._CustomerID=new_CustomerID
@property
def OrderDate(self):
return self.OrderDate
@OrderDate.setter
def OrderDate(self,new_OrderDate):
self._OrderDate=new_OrderDate
@property
def TotalAmount(self):
return self.TotalAmount
@TotalAmount.setter
def TotalAmount(self, new_Totalamount):
if isinstance(new_Totalamount, (int, float)) and new_Totalamount >= 0:
self.TotalAmount = new_Totalamount

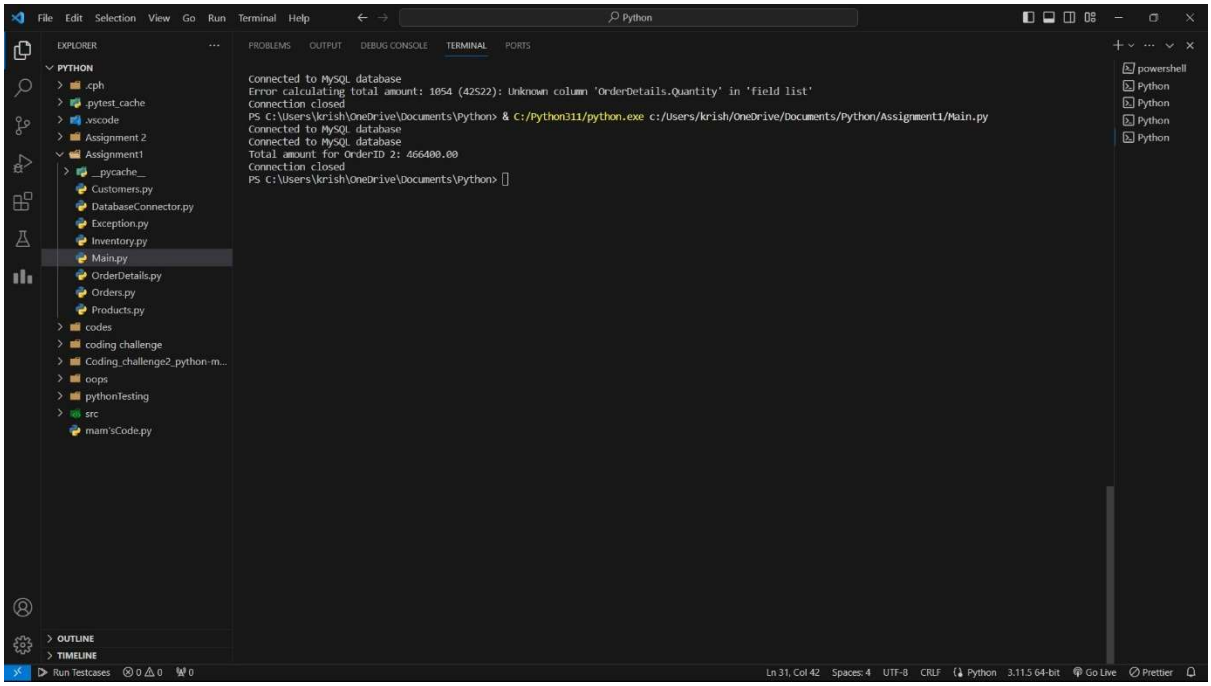

.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnect...
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
def Calculate_Total_Amount(self,OrderID):
try:
self._db_connector.open_connection()
query_check_order = "SELECT * FROM Orders WHERE OrderID = %s"
values = (OrderID,)
with self._db_connector.connection.cursor() as cursor_check_order:
cursor_check_order.execute(query_check_order, values)
order_exists = cursor_check_order.fetchone()
if not order_exists:
print("Order ID not found.")
return
calculate_total_amount = """
SELECT SUM(Products.Price * OrderDetails.Qunatity)
FROM OrderDetails
JOIN Products ON OrderDetails.ProductID = Products.ProductID
WHERE OrderDetails.OrderID = %s
"""
values1 = (OrderID,)
with self._db_connector.connection.cursor() as cursor_calculate_total_amount:
cursor_calculate_total_amount.execute(calculate_total_amount, values1)
total_amount = cursor_calculate_total_amount.fetchone()[0]
print(f"Total amount for OrderID {OrderID}: {total_amount:.2f}")
except Exception as e:
print(f"Error calculating total amount: {e}")
finally:
self._db_connector.close_connection()
def get_Order_Details(self,OrderID):
try:
self._db_connector.open_connection()

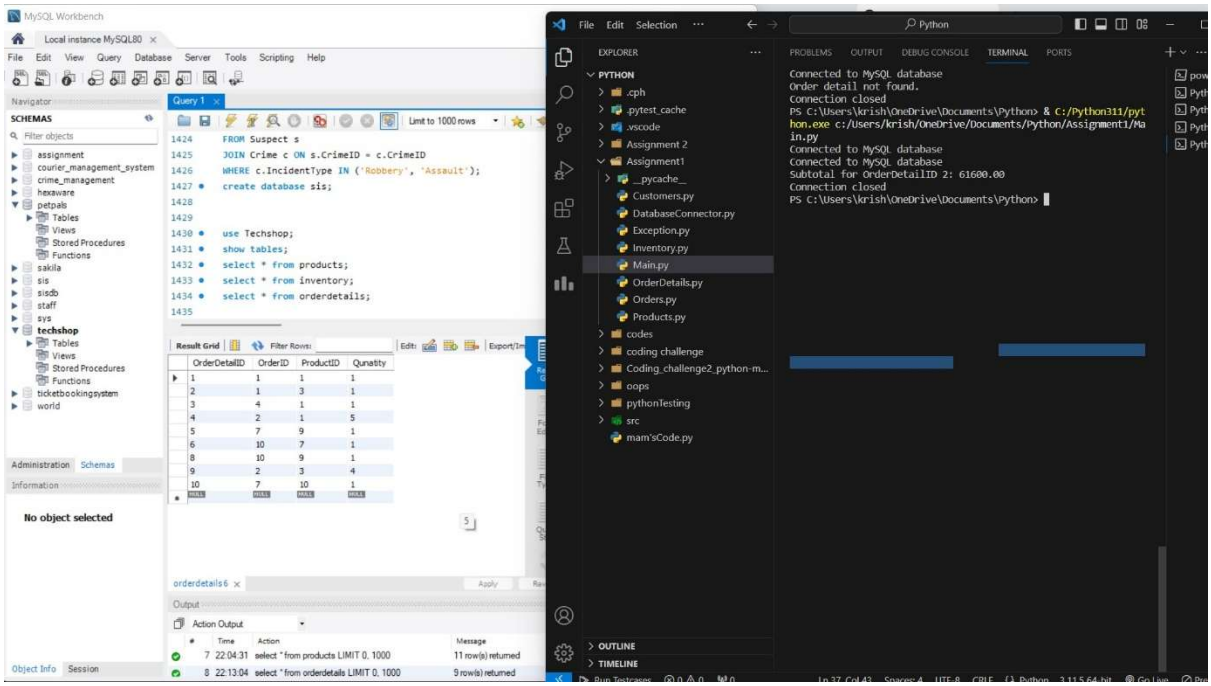



```python
    def get_Order_Details(self,OrderID):
        try:
            self._db_connector.open_connection()

            query = "SELECT * FROM orders WHERE OrderID=%s"
            values = (OrderID,)

            self._db_connector.cursor.execute(query, values)

            order_details = self._db_connector.cursor.fetchone()

            if order_details:
                print("Order Details:")
                print(f"Order ID:{order_details[0]}")
                print(f"Customer ID:{order_details[1]}")
                print(f"Order Date: {order_details[2]}")
                print(f"Total Amount: {order_details[3]}")

            else:
                print("Order Id not found.")

        except Exception as e:
            print(f"Error getting Order details: {e}")

        finally:
            self._db_connector.close_connection()

    def place_order(self, OrderID,OrderDetailID,CustomerID, ProductID, Quantity):
        try:
            self._db_connector.open_connection()
            query = "INSERT INTO Orders (OrderID,CustomerID, OrderDate) VALUES (%s, %s,%s)"
            values = (OrderID,CustomerID, datetime.now())

            with self._db_connector.connection.cursor() as cursor1:
                cursor1.execute(query, values)

            query_details = "INSERT INTO OrderDetails (OrderdetailID,OrderID, ProductID, Quantity) VALUES (%s, %s, %s,%s)"
            values_details = (OrderDetailID, OrderID, ProductID, Quantity)
```




```python
            with self._db_connector.connection.cursor() as cursor2:
                cursor2.execute(query_details, values_details)

            query_update_inventory = "UPDATE Inventory SET QuantityInStock = QuantityInStock - %s WHERE ProductID = %s"
            values_update_inventory = (Quantity, ProductID)
            with self._db_connector.connection.cursor() as cursor3:
                cursor3.execute(query_update_inventory, values_update_inventory)

            query_total_amount = """
                        SELECT SUM(Products.Price * OrderDetails.Quantity)
                        FROM OrderDetails
                        JOIN Products ON OrderDetails.ProductID = Products.ProductID
                        WHERE OrderDetails.OrderID = %s
                    """
            with self._db_connector.connection.cursor() as cursor4:
                cursor4.execute(query_total_amount, (OrderID,))
                total_amount = cursor4.fetchone()[0]

            query_update_total_amount = "UPDATE Orders SET TotalAmount = %s WHERE OrderID = %s"
            values_update_total_amount = (total_amount, OrderID)
            with self._db_connector.connection.cursor() as cursor5:
                cursor5.execute(query_update_total_amount, values_update_total_amount)
            self._db_connector.connection.commit()
            print(f"Order placed successfully. OrderID: {OrderID} OrderTotal :{total_amount} and inventory updated with avaialble stock")
        except Exception as e:
            print(f"Error placing order: {e}")
        finally:
            self._db_connector.close_connection()

    def updateOrderStatus(self):
        #no status table in database schema
        pass

    def cancelOrder(self):
        # no status table in database schema
        pass
```





OrderDetails Class:

Attributes:

- OrderDetailID (int)
- Order (Order) - Use composition to reference the Order to which this detail belongs.
- Product (Product) - Use composition to reference the Product included in the order detail.
- Quantity (int)

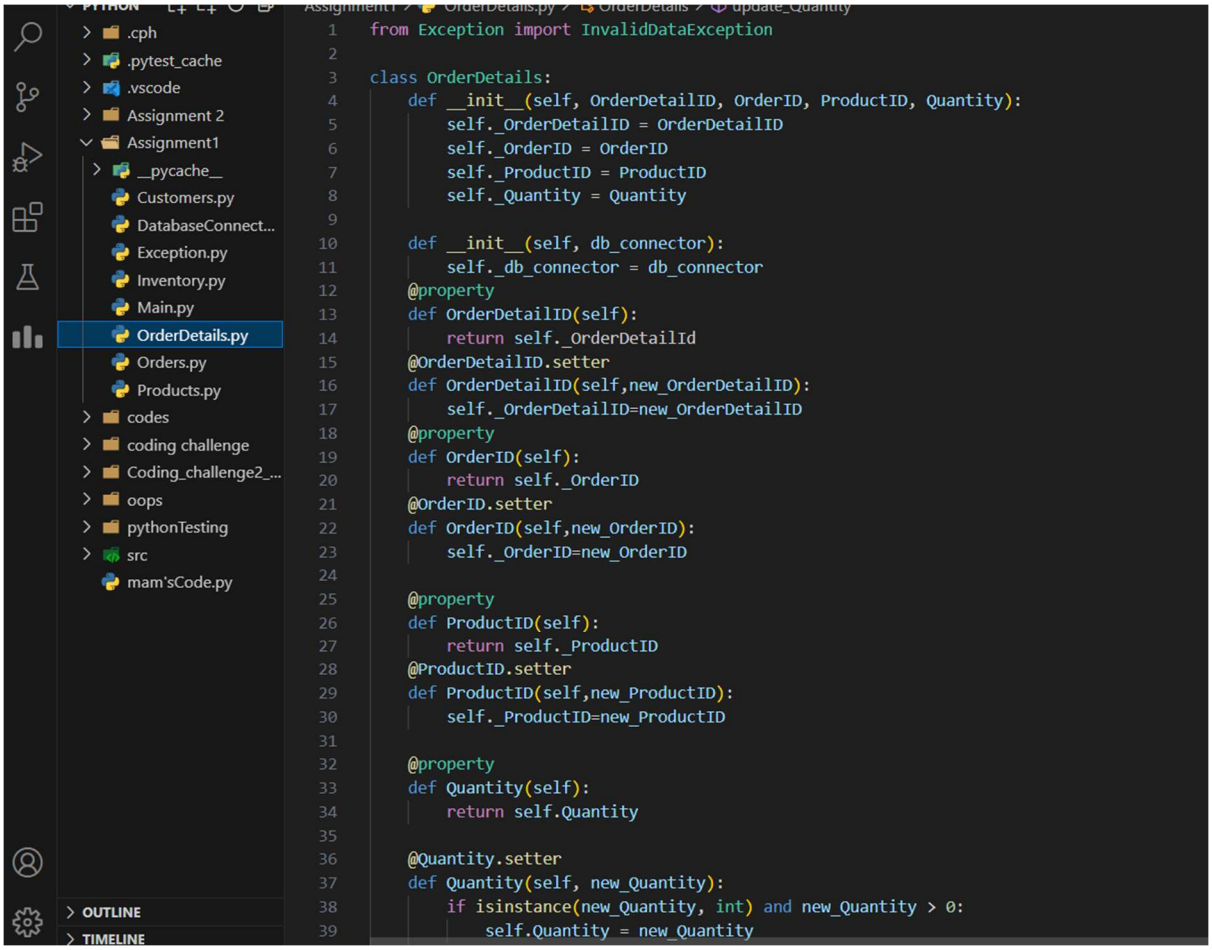
Methods:

• CalculateSubtotal() - Calculate the subtotal for this order detail.

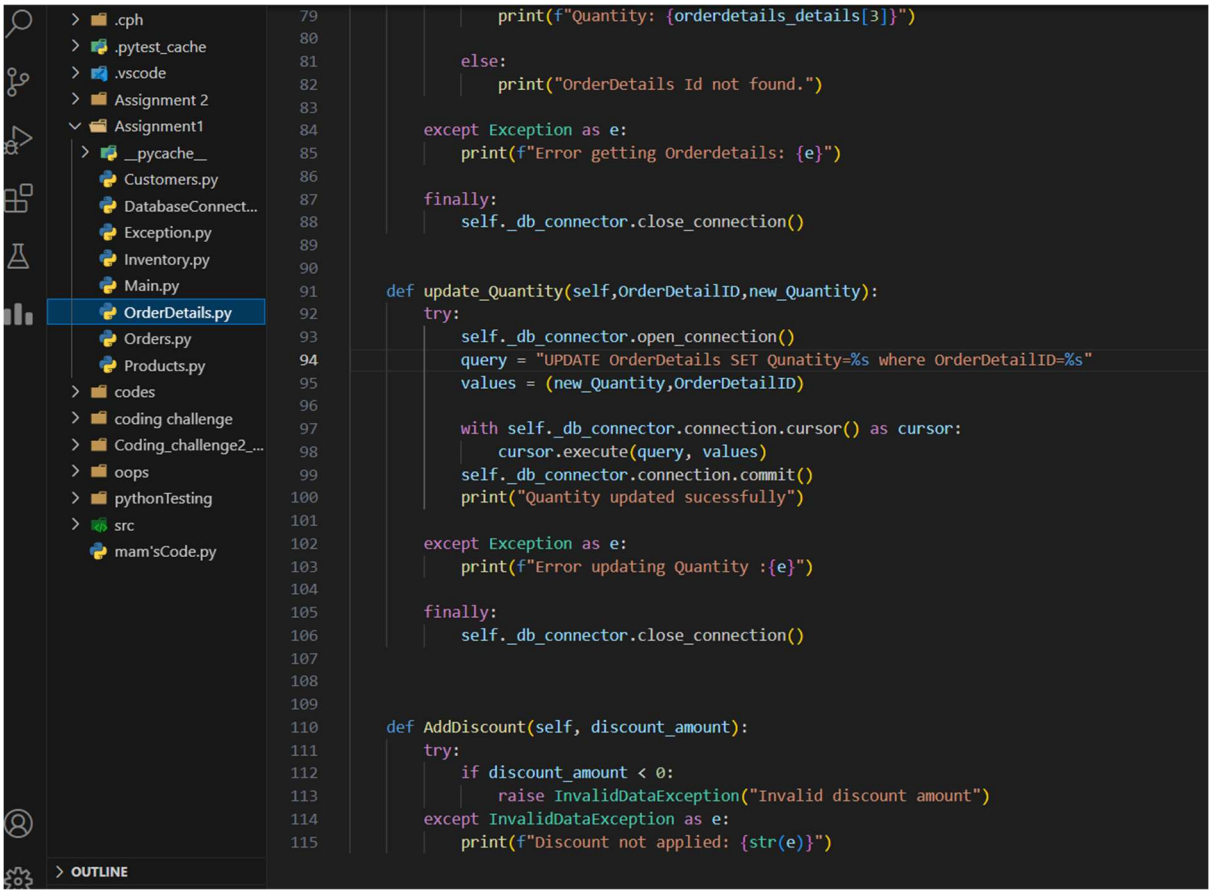
• GetOrderDetailInfo(): Retrieves and displays information about this order detail.

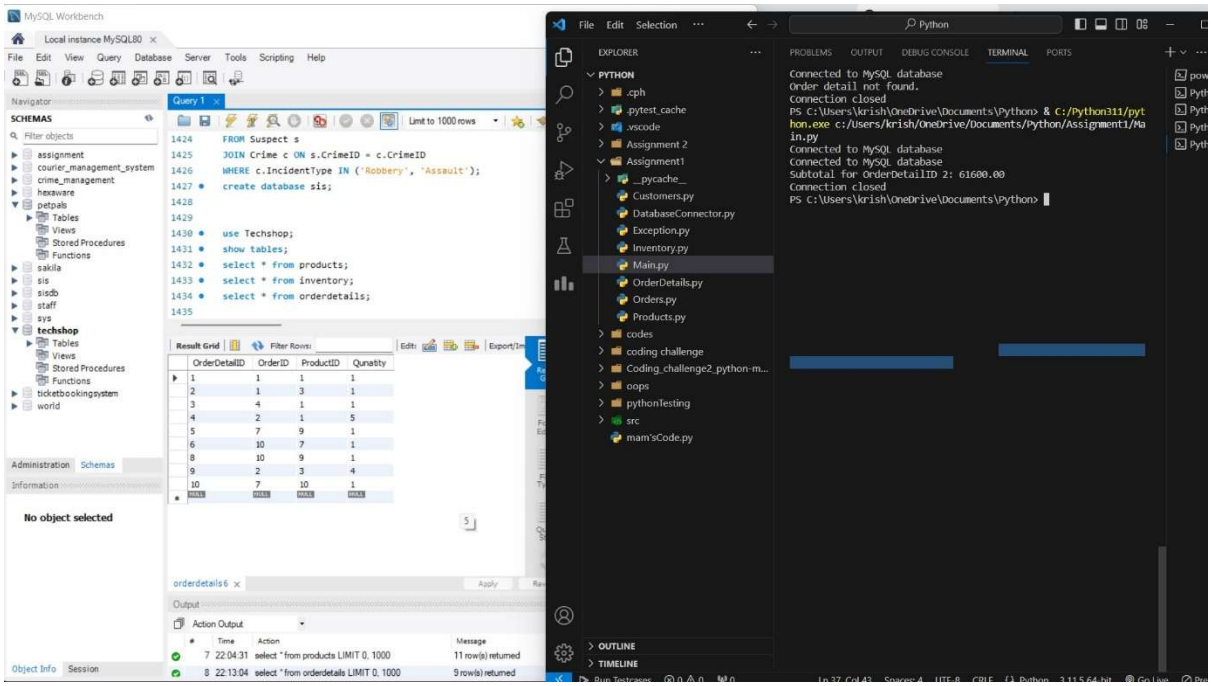
• UpdateQuantity(): Allows updating the quantity of the product in this order detail.

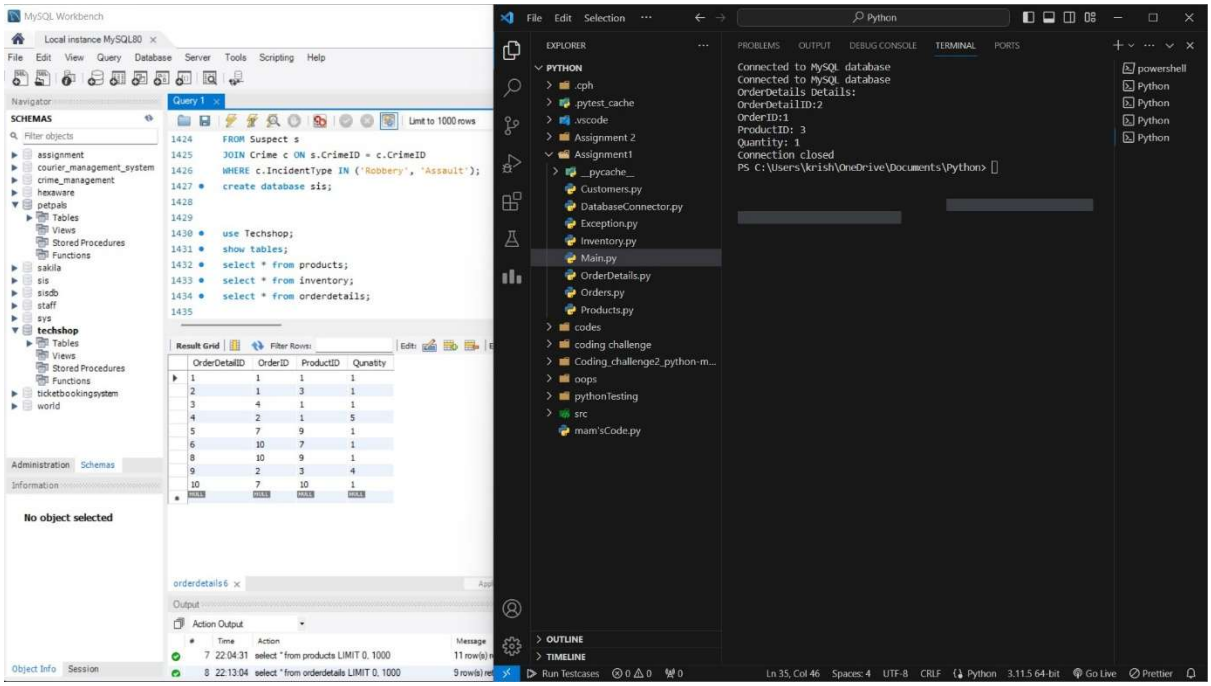
• AddDiscount(): Applies a discount to this order detail.

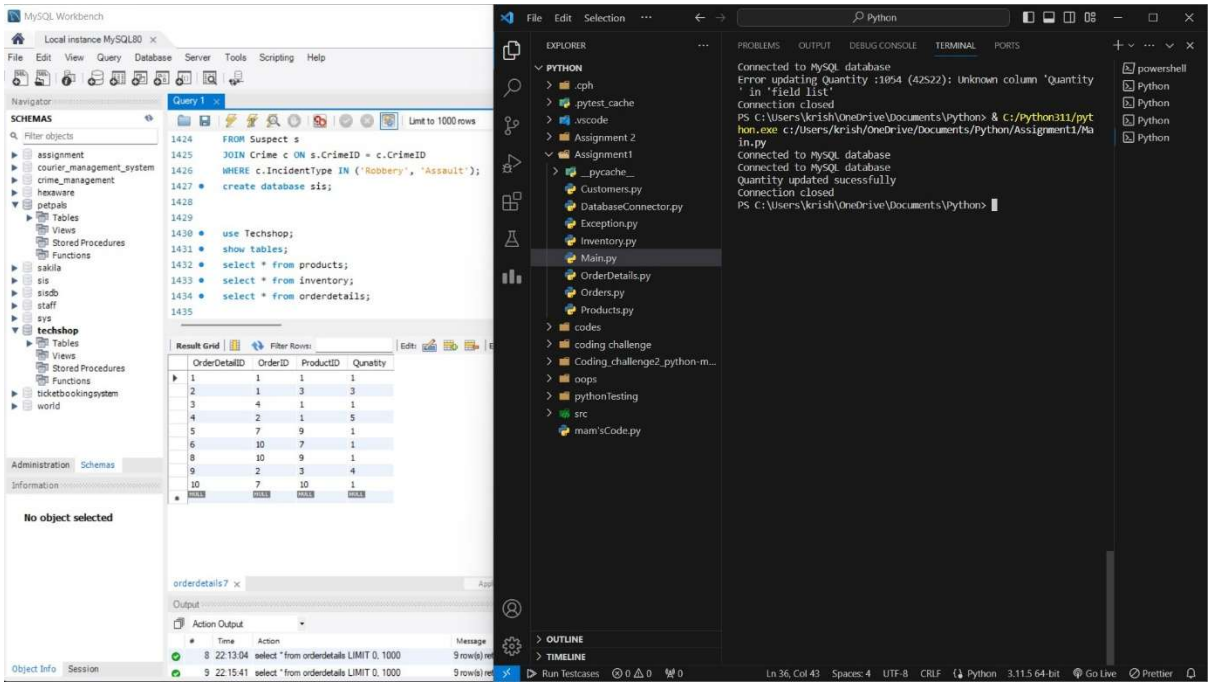



.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnect...
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
def Calculate_Subtotal(self, OrderDetailID):
try:
self._db_connector.open_connection()
query = "SELECT OD.Qunatity, P.Price FROM Orderdetails OD INNER JOIN Products P ON OD.ProductID = P.ProductID WHE
values = (OrderDetailID,)
with self._db_connector.cursor as cursor:
cursor.execute(query, values)
order_data = cursor.fetchone()
if order_data:
quantity, price = order_data
subtotal = quantity * price
print(f"Subtotal for OrderDetailID {OrderDetailID}: {subtotal}")
else:
print("Order detail not found.")
except Exception as e:
print(f"Error calculating subtotal: {e}")
finally:
self._db_connector.close_connection()
def get_OrderDetail_Info(self, OrderDetailID):
try:
self._db_connector.open_connection()
query = "SELECT * FROM orderdetails WHERE OrderDetailID=%s"
values = (OrderDetailID,)
self._db_connector.cursor.execute(query, values)
orderdetails_details = self._db_connector.cursor.fetchone()
if orderdetails_details:
print("OrderDetails Details:")
print(f"OrderDetailID:{orderdetails_details[0]}")
print(f"OrderID:{orderdetails_details[1]}")
print(f"ProductID: {orderdetails_details[2]}")
print(f"Quantity: {orderdetails_details[3]}")


.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnect...
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
print(f"Quantity: {orderdetails_details[3]}")
else:
print("OrderDetails Id not found.")
except Exception as e:
print(f"Error getting Orderdetails: {e}")
finally:
self._db_connector.close_connection()
def update_Quantity(self,OrderDetailID,new_Quantity):
try:
self._db_connector.open_connection()
query = "UPDATE OrderDetails SET Qunatity=%s where OrderDetailID=%s"
values = (new_Quantity,OrderDetailID)
with self._db_connector.connection.cursor() as cursor:
cursor.execute(query, values)
self._db_connector.connection.commit()
print("Quantity updated sucessfully")
except Exception as e:
print(f"Error updating Quantity :{e}")
finally:
self._db_connector.close_connection()
def AddDiscount(self, discount_amount):
try:
if discount_amount < 0:
raise InvalidDataException("Invalid discount amount")
except InvalidDataException as e:
print(f"Discount not applied: {str(e)}")


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
Query 1
1424 FROM Suspect s
1425 JOIN Crime c ON s.CrimeID = c.CrimeID
1426 WHERE c.IncidentType IN ('Robbery', 'Assault');
1427 create database sis;
1428
1429
1430 use Techshop;
1431 show tables;
1432 select * from products;
1433 select * from inventory;
1434 select * from orderdetails;
1435
Result Grid
OrderDetailID OrderID ProductID Qunatity
1 1 1 1
2 1 3 1
3 4 1 1
4 2 1 5
5 7 9 1
6 10 7 1
8 10 9 1
9 2 3 4
10 7 10 1
No object selected
Output
Action Output
7 22:04:31 select * from products LIMIT 0, 1000 11 row(s) returned
8 22:13:04 select * from orderdetails LIMIT 0, 1000 9 row(s) returned
File Edit Selection
Python
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PYTHON
Connected to MySQL database
Order detail not found.
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Subtotal for OrderDetailID 2: 61600.00
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
Assignment 2
Assignment1
Customers.py
DatabaseConnector.py
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
oops
pythonTesting
src
mam'sCode.py
OUTLINE
TIMELINE


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
Query 1
1424 FROM Suspect s
1425 JOIN Crime c ON s.CrimeID = c.CrimeID
1426 WHERE c.IncidentType IN ('Robbery', 'Assault');
1427 create database sis;
1428
1429
1430 use Techshop;
1431 show tables;
1432 select * from products;
1433 select * from inventory;
1434 select * from orderdetails;
1435
Result Grid
OrderDetailID OrderID ProductID Qunatity
No object selected
Output
Action Output
7 22:04:31 select * from products LIMIT 0, 1000 11 row(s) returned
8 22:13:04 select * from orderdetails LIMIT 0, 1000 9 row(s) returned
File Edit Selection
Python
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PYTHON
Connected to MySQL database
Connected to MySQL database
OrderDetails Details:
OrderDetailID:2
OrderID:1
ProductID: 3
Quantity: 1
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
powershell
Python
Python
Python
Python
Assignment 2
Assignment1
Customers.py
DatabaseConnector.py
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
oops
pythonTesting
src
mam'sCode.py
OUTLINE
TIMELINE
Run Testcases
Ln 35, Col 46 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier



Inventory class:

Attributes:

• InventoryID(int)

• Product (Composition): The product associated with the inventory item.

• QuantityInStock: The quantity of the product currently in stock.

• LastStockUpdate

Methods:

• GetProduct(): A method to retrieve the product associated with this inventory item.

• GetQuantityInStock(): A method to get the current quantity of the product in stock.

• AddToInventory(int quantity): A method to add a specified quantity of the product to the inventory.

• RemoveFromInventory(int quantity): A method to remove a specified quantity of the product from the inventory.

• UpdateStockQuantity(int newQuantity): A method to update the stock quantity to a new value.

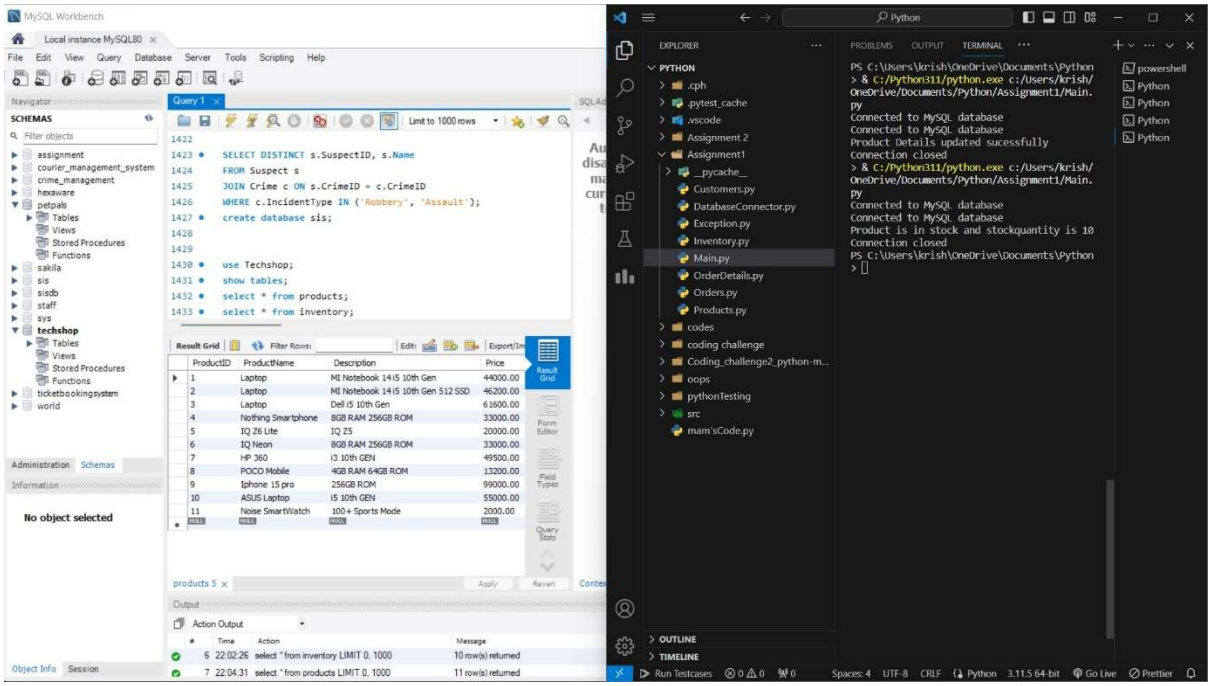
• IsProductAvailable(int quantityToCheck): A method to check if a specified quantity of the product is available in the inventory.

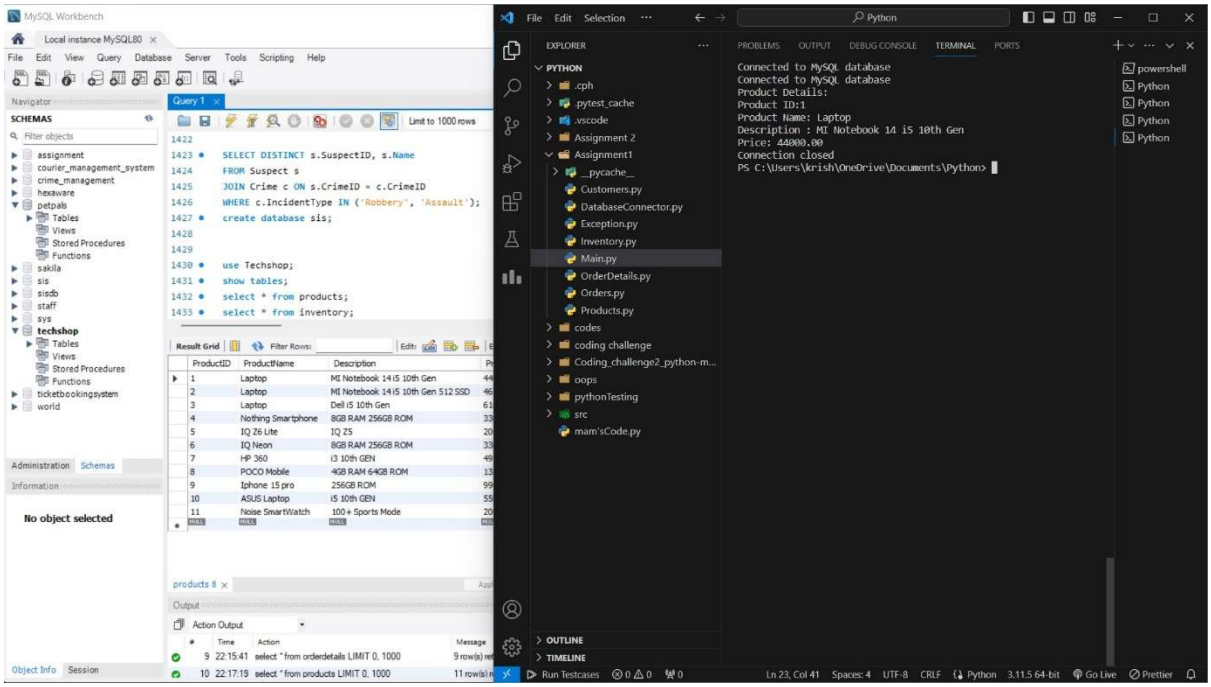
• GetInventoryValue(): A method to calculate the total value of the products in the inventory based on their prices and quantities.

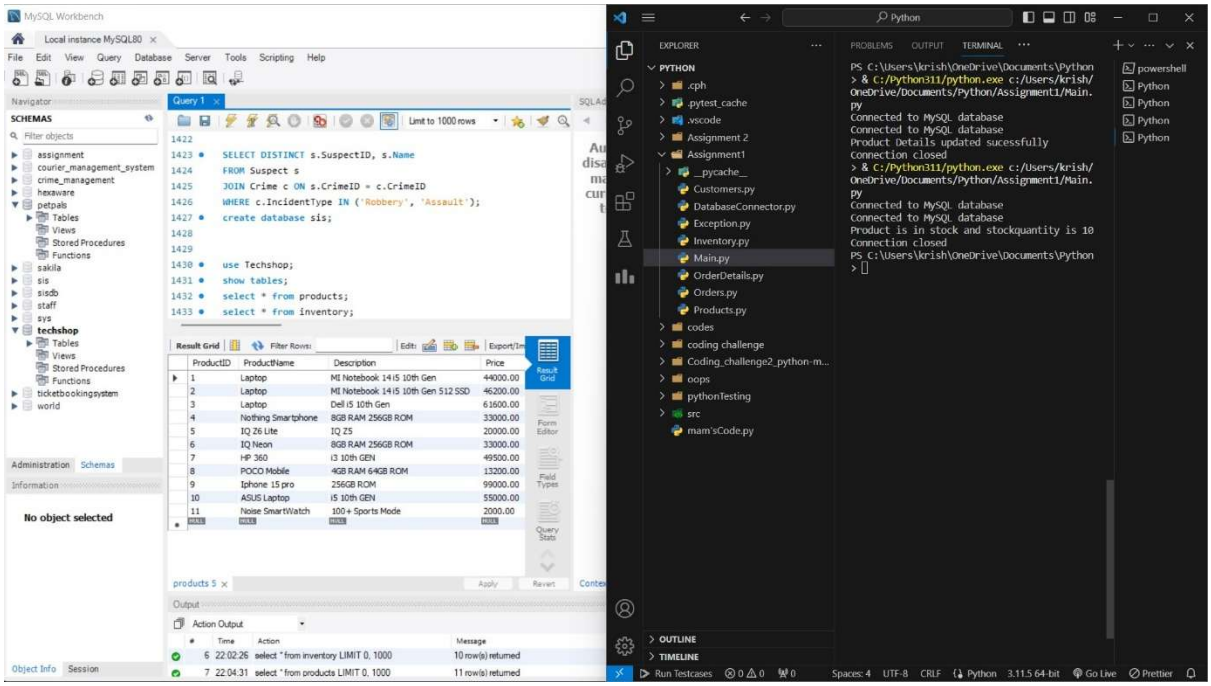
• ListLowStockProducts(int threshold): A method to list products with quantities below a specified threshold, indicating low stock.

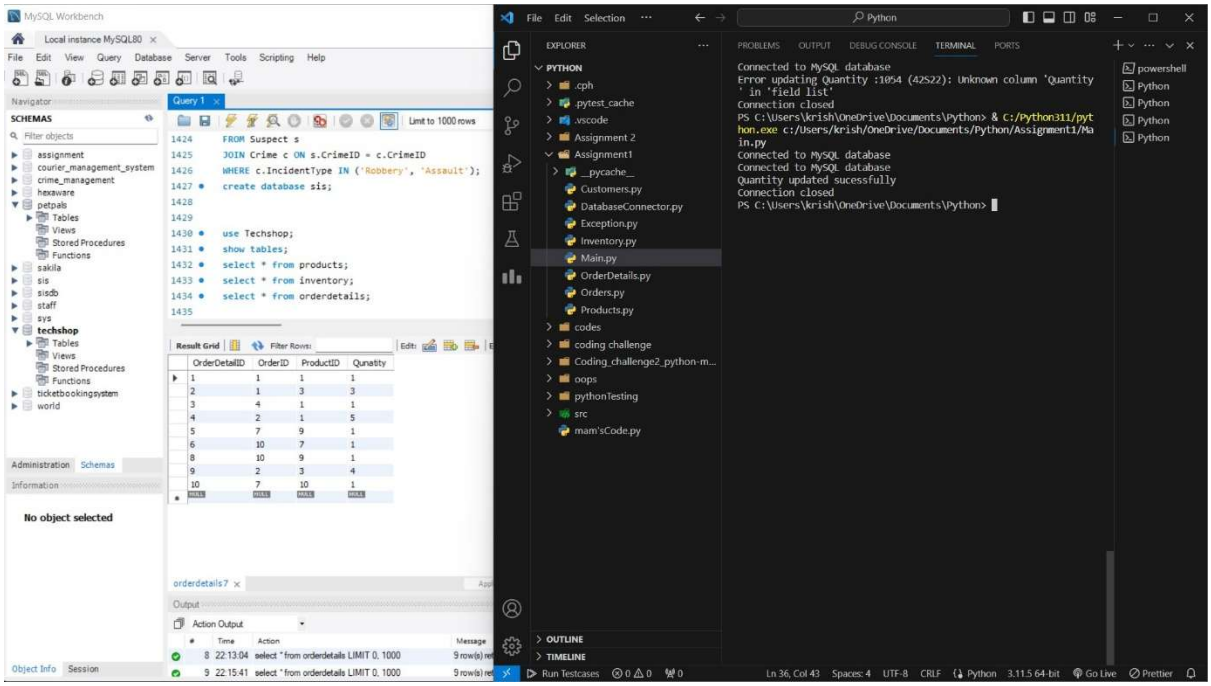
• ListOutOfStockProducts(): A method to list products that are out of stock

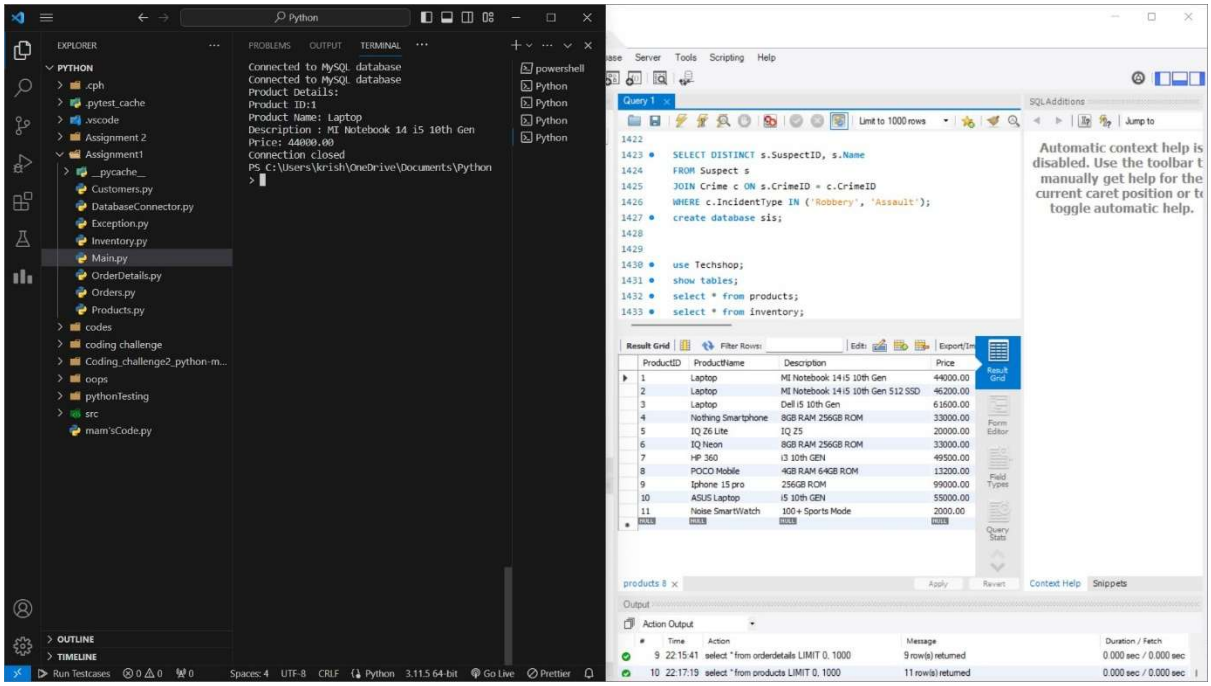
• ListAllProducts(): A method to list all products in the inventory, along with their quantities.

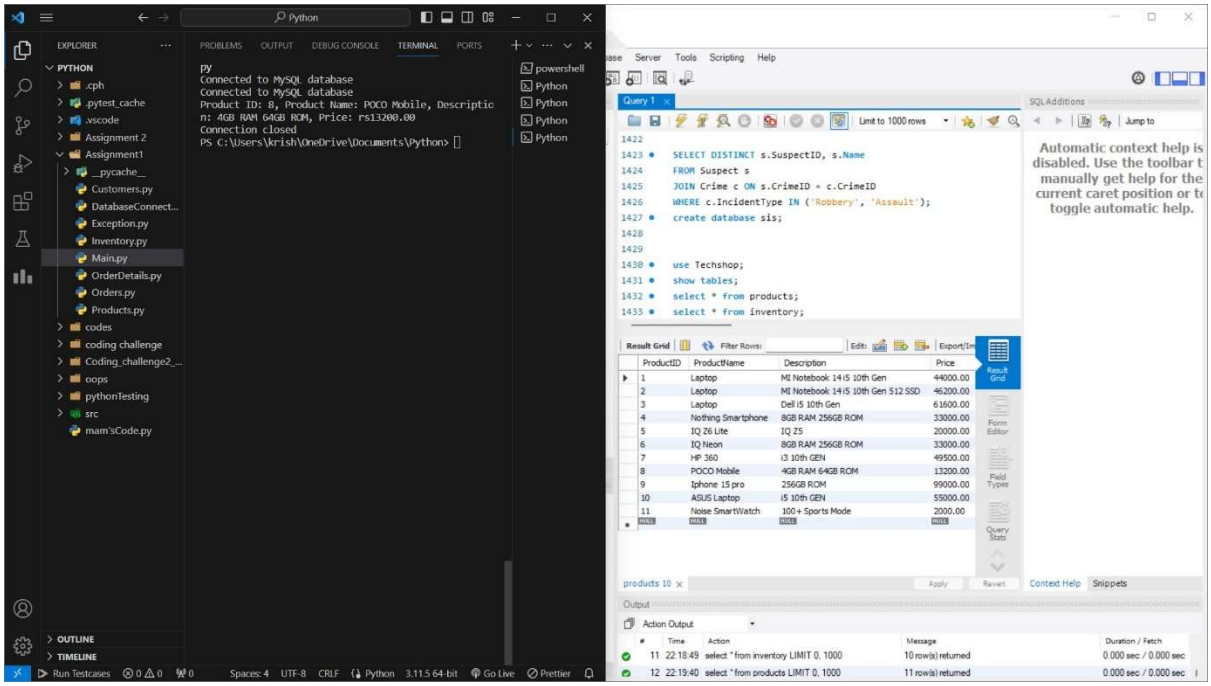


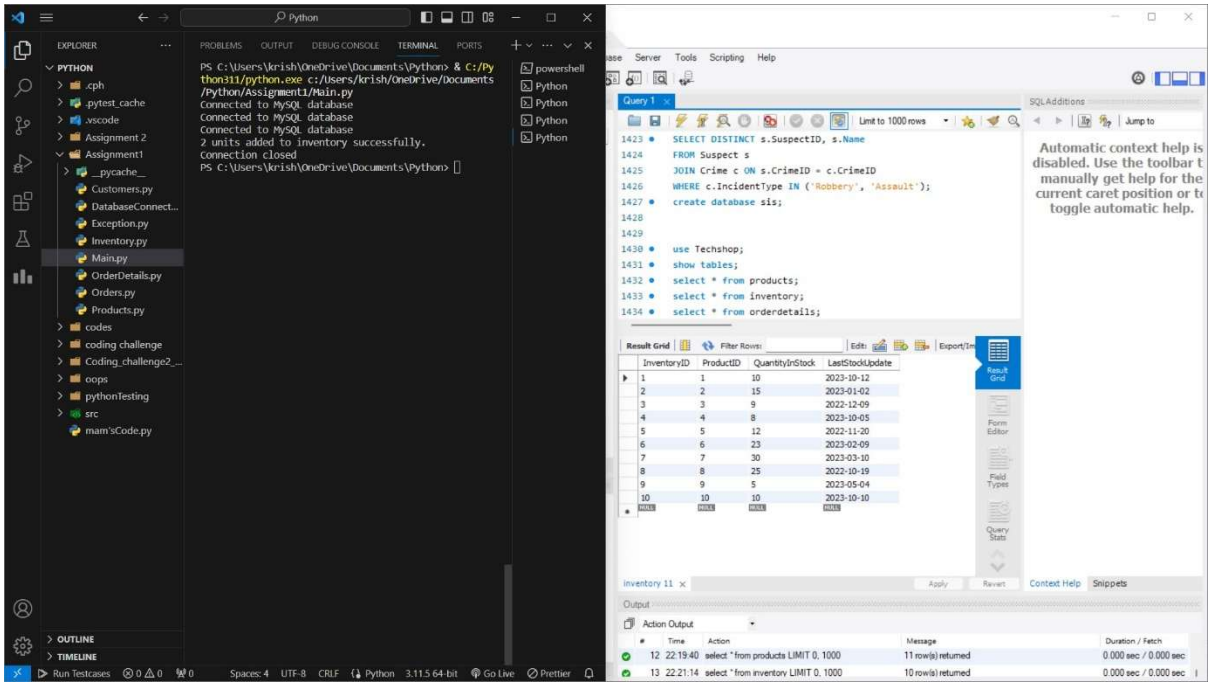


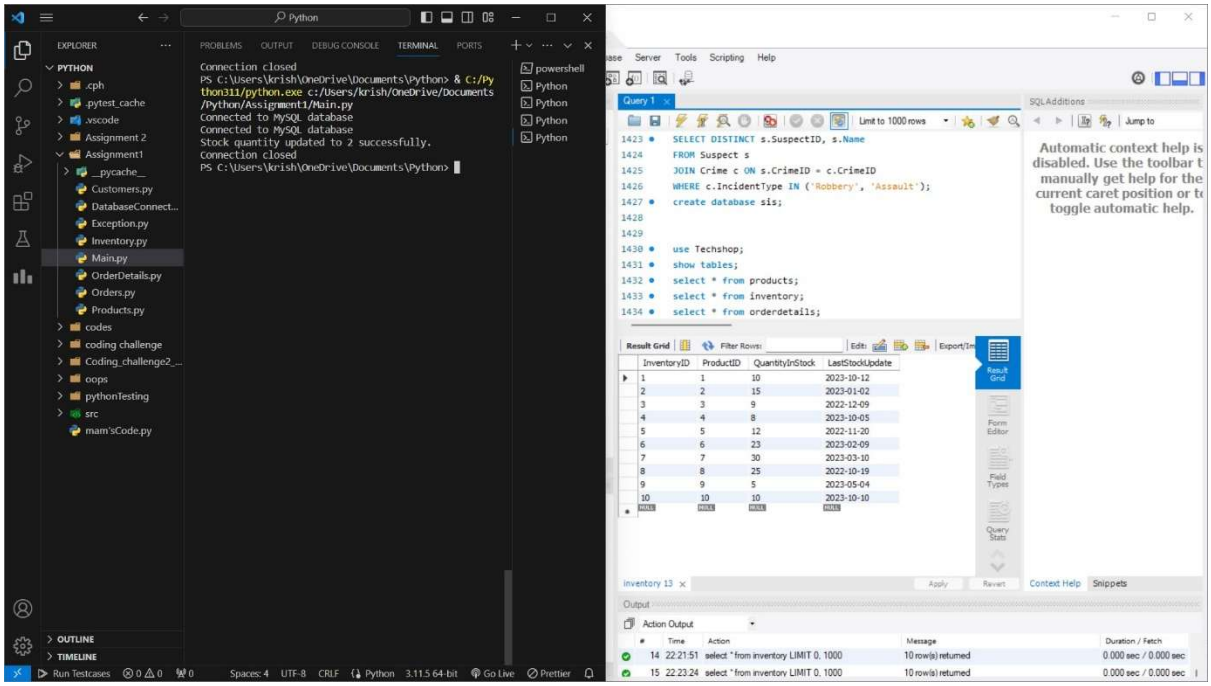

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
assignment
courier_management_system
crime_management
hexaware
petpals
Tables
Views
Stored Procedures
Functions
sakila
sis
sisdb
staff
sys
techshop
ticketbookingsystem
world
Query 1
SELECT DISTINCT s.SuspectID, s.Name
FROM Suspect s
JOIN Crime c ON s.CrimeID = c.CrimeID
WHERE c.IncidentType IN ('Robbery', 'Assault');
create database sis;
use Techshop;
show tables;
select * from products;
select * from inventory;
Result Grid
ProductID ProductName Description Price
1 Laptop MI Notebook 14 i5 10th Gen 44000.00
2 Laptop MI Notebook 14 i5 10th Gen 512 SSD 46200.00
3 Laptop Dell i5 10th Gen 61600.00
4 Nothing Smartphone 8GB RAM 256GB ROM 33000.00
5 IQ Z6 Lite IQ Z5 20000.00
6 IQ Neon 8GB RAM 256GB ROM 33000.00
7 HP 360 i3 10th GEN 49500.00
8 POCO Mobile 4GB RAM 64GB ROM 13200.00
9 Iphone 15 pro 256GB ROM 99000.00
10 ASUS Laptop i5 10th GEN 55000.00
11 Noise SmartWatch 100+ Sports Mode 2000.00
No object selected
Output
Action Output
6 22:02:26 select * from inventory LIMIT 0, 1000 10 row(s) returned
7 22:04:31 select * from products LIMIT 0, 1000 11 row(s) returned
Python
EXPLORER
PYTHON
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnector.py
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_python-m...
oops
pythonTesting
src
mam'sCode.py
PROBLEMS OUTPUT TERMINAL
PS C:\Users\krish\OneDrive\Documents\Python
> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Product Details updated sucessfully
Connection closed
> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Product is in stock and stockquantity is 10
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python
powershell
Python
Python
Python
Python
OUTLINE
TIMELINE
Run Testcases
Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier

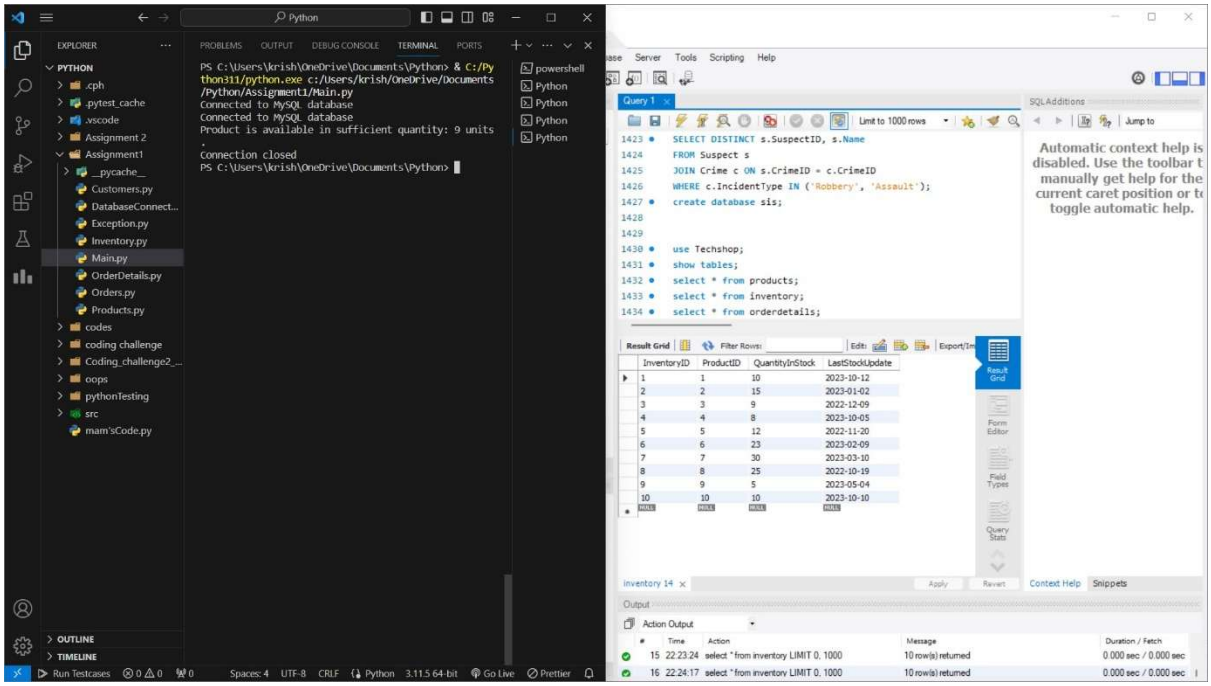

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
Query 1
FROM Suspect s
JOIN Crime c ON s.CrimeID = c.CrimeID
WHERE c.IncidentType IN ('Robbery', 'Assault');
create database sis;
use Techshop;
show tables;
select * from products;
select * from inventory;
select * from orderdetails;
Result Grid
OrderDetailID OrderID ProductID Qunatity
1 1 1 1
2 1 3 3
3 4 1 1
4 2 1 5
5 7 9 1
6 10 7 1
8 10 9 1
9 2 3 4
10 7 10 1
No object selected
Output
Action Output
8 22:13:04 select * from orderdetails LIMIT 0, 1000 9 row(s) ret
9 22:15:41 select * from orderdetails LIMIT 0, 1000 9 row(s) ret
File Edit Selection
Python
EXPLORER
PYTHON
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Connected to MySQL database
Error updating Quantity :1054 (42S22): Unknown column 'Quantity ' in 'field list'
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Quantity updated sucessfully
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
powershell
Python
Python
Python
Python
OUTLINE
TIMELINE
Run Testcases
Ln 36, Col 43 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier

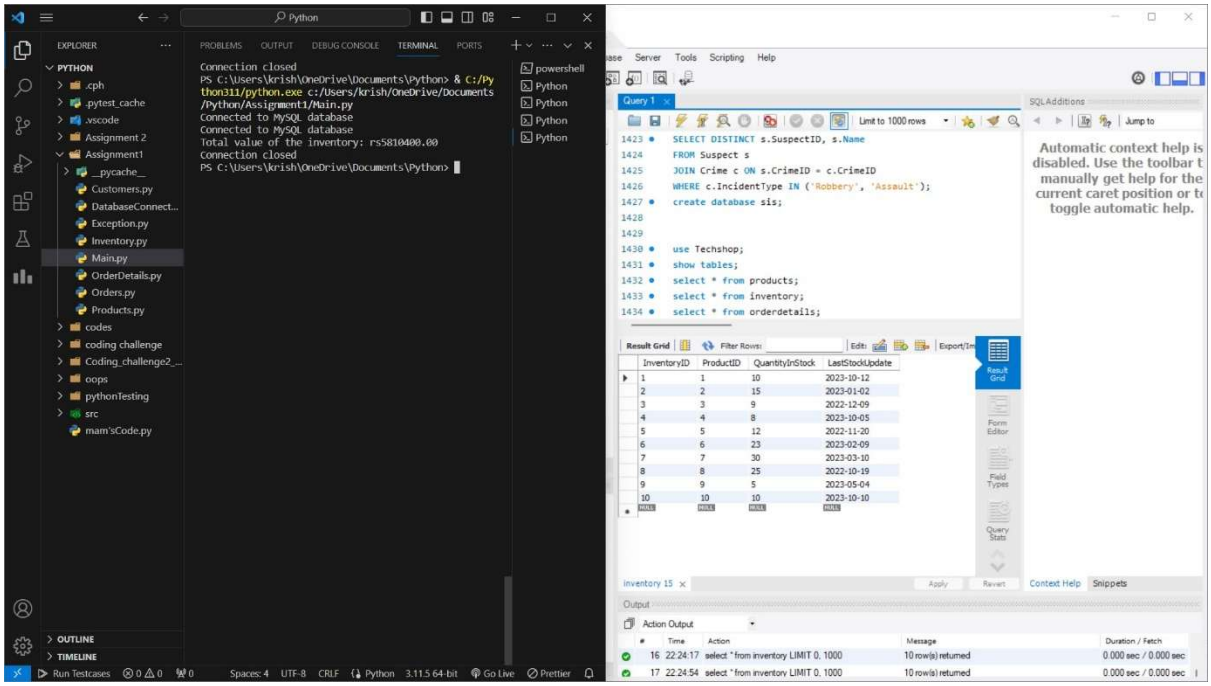

Python
EXPLORER
PROBLEMS OUTPUT TERMINAL
Connected to MySQL database
Connected to MySQL database
Product Details:
Product ID:1
Product Name: Laptop
Description : MI Notebook 14 i5 10th Gen
Price: 44000.00
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python
Server Tools Scripting Help
Query 1
SELECT DISTINCT s.SuspectID, s.Name
FROM Suspect s
JOIN Crime c ON s.CrimeID = c.CrimeID
WHERE c.IncidentType IN ('Robbery', 'Assault');
create database sis;
use Techshop;
show tables;
select * from products;
select * from inventory;
Result Grid
ProductID ProductName Description Price
1 Laptop MI Notebook 14 i5 10th Gen 44000.00
2 Laptop MI Notebook 14 i5 10th Gen 512 SSD 46200.00
3 Laptop Dell i5 10th Gen 61600.00
4 Nothing Smartphone 8GB RAM 256GB ROM 33000.00
5 IQ Z6 Lite IQ Z5 20000.00
6 IQ Neon 8GB RAM 256GB ROM 33000.00
7 HP 360 i3 10th GEN 49500.00
8 POCO Mobile 4GB RAM 64GB ROM 13200.00
9 Iphone 15 pro 256GB ROM 99000.00
10 ASUS Laptop i5 10th GEN 55000.00
11 Noise SmartWatch 100+ Sports Mode 2000.00
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Output
Action Output
9 22:15:41 select * from orderdetails LIMIT 0, 1000 9 row(s) returned 0.000 sec / 0.000 sec
10 22:17:19 select * from products LIMIT 0, 1000 11 row(s) returned 0.000 sec / 0.000 sec

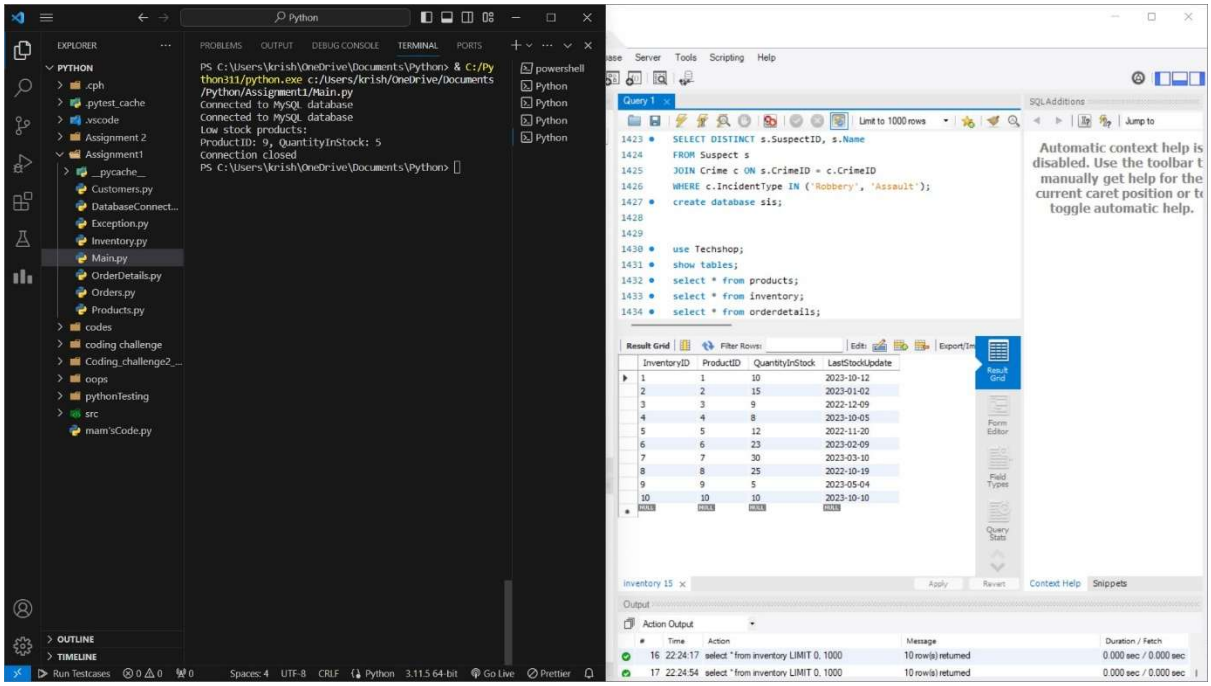

Python
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
py
Connected to MySQL database
Connected to MySQL database
Product ID: 8, Product Name: POCO Mobile, Descriptio n: 4GB RAM 64GB ROM, Price: rs13200.00
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
Server Tools Scripting Help
Query 1
SELECT DISTINCT s.SuspectID, s.Name
FROM Suspect s
JOIN Crime c ON s.CrimeID = c.CrimeID
WHERE c.IncidentType IN ('Robbery', 'Assault');
create database sis;
use Techshop;
show tables;
select * from products;
select * from inventory;
Result Grid
ProductID ProductName Description Price
1 Laptop MI Notebook 14 i5 10th Gen 44000.00
2 Laptop MI Notebook 14 i5 10th Gen 512 SSD 46200.00
3 Laptop Dell i5 10th Gen 61600.00
4 Nothing Smartphone 8GB RAM 256GB ROM 33000.00
5 IQ Z6 Lite IQ Z5 20000.00
6 IQ Neon 8GB RAM 256GB ROM 33000.00
7 HP 360 i3 10th GEN 49500.00
8 POCO Mobile 4GB RAM 64GB ROM 13200.00
9 Iphone 15 pro 256GB ROM 99000.00
10 ASUS Laptop i5 10th GEN 55000.00
11 Noise SmartWatch 100+ Sports Mode 2000.00
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Output
Action Output
11 22:18:49 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
12 22:19:40 select * from products LIMIT 0, 1000 11 row(s) returned 0.000 sec / 0.000 sec

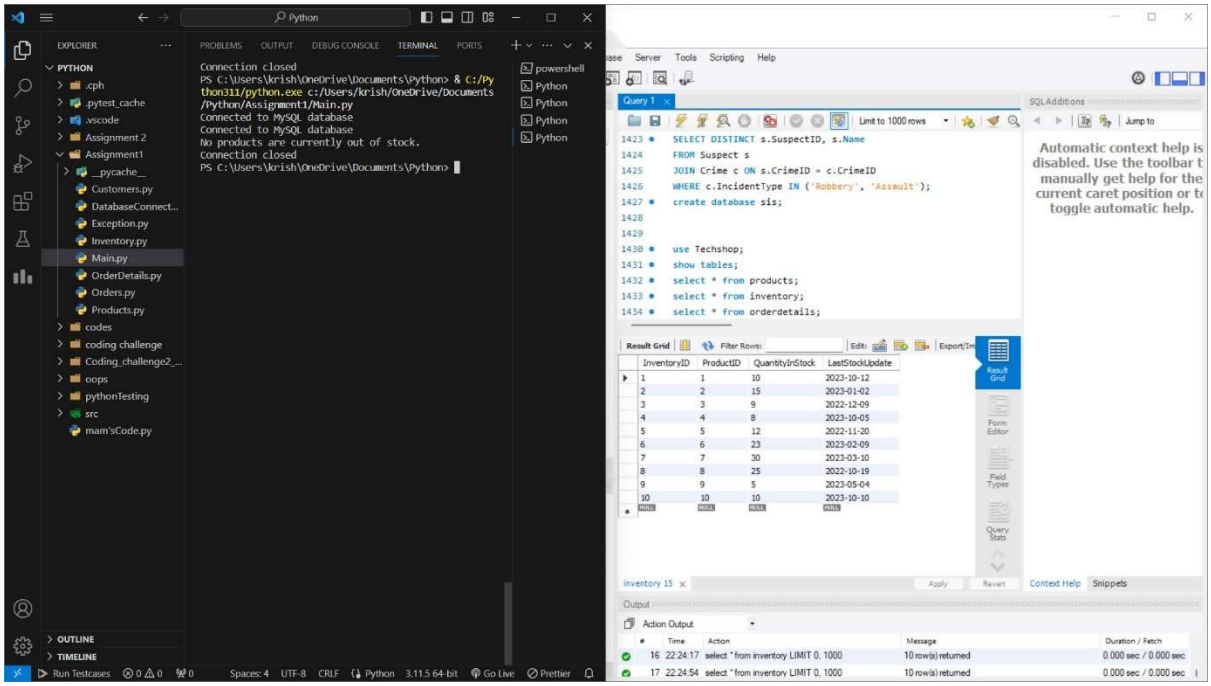

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Connected to MySQL database
2 units added to inventory successfully.
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
SELECT DISTINCT s.SuspectID, s.Name
FROM Suspect s
JOIN Crime c ON s.CrimeID = c.CrimeID
WHERE c.IncidentType IN ('Robbery', 'Assault');
create database sis;
use Techshop;
show tables;
select * from products;
select * from inventory;
select * from orderdetails;
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

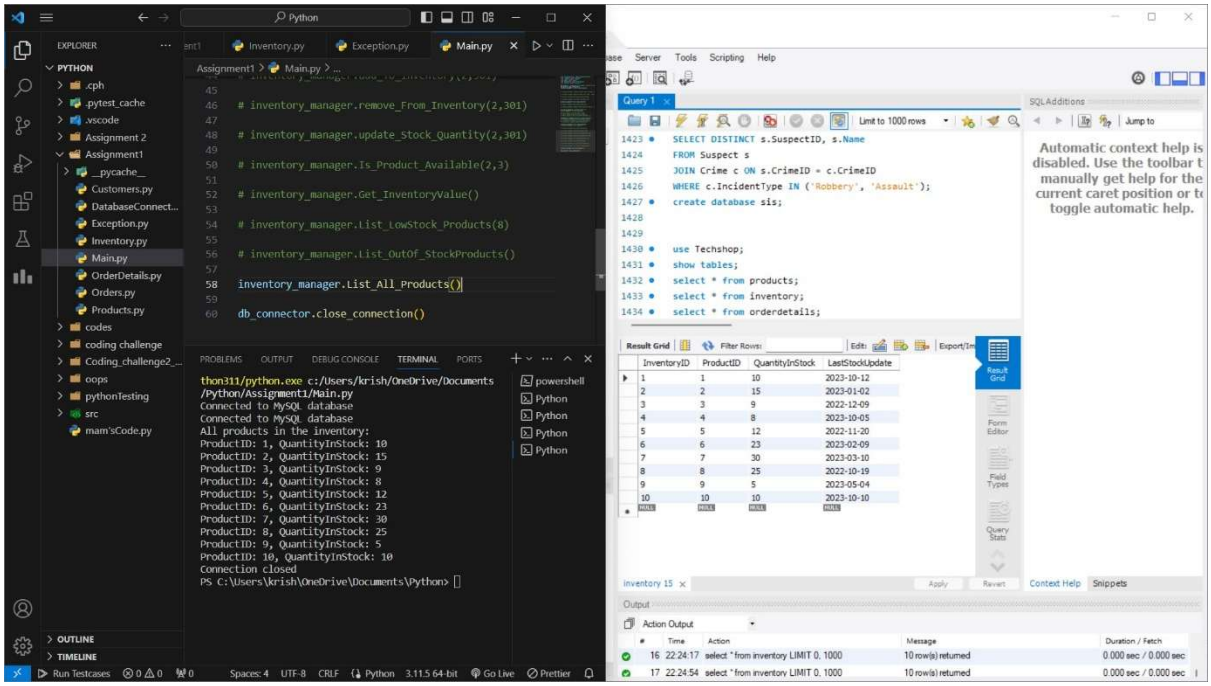

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Connected to MySQL database
Stock quantity updated to 2 successfully.
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
SELECT DISTINCT s.SuspectID, s.Name
FROM Suspect s
JOIN Crime c ON s.CrimeID = c.CrimeID
WHERE c.IncidentType IN ('Robbery', 'Assault');
create database sis;
use Techshop;
show tables;
select * from products;
select * from inventory;
select * from orderdetails;
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.


Python
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Product is available in sufficient quantity: 9 units
.
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
powershell
Python
Python
Python
Python
PYTHON
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnect...
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
TIMELINE
Run Testcases
Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier
Server Tools Scripting Help
Query 1
Limit to 1000 rows
1423 SELECT DISTINCT s.SuspectID, s.Name
1424 FROM Suspect s
1425 JOIN Crime c ON s.CrimeID = c.CrimeID
1426 WHERE c.IncidentType IN ('Robbery', 'Assault');
1427 create database sis;
1428
1429
1430 use Techshop;
1431 show tables;
1432 select * from products;
1433 select * from inventory;
1434 select * from orderdetails;
SQL Additions
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Result Grid
InventoryID ProductID QuantityInStock LastStockUpdate
1 1 10 2023-10-12
2 2 15 2023-01-02
3 3 9 2022-12-09
4 4 8 2023-10-05
5 5 12 2022-11-20
6 6 23 2023-02-09
7 7 30 2023-03-10
8 8 25 2022-10-19
9 9 5 2023-05-04
10 10 10 2023-10-10
inventory 14
Apply
Revert
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
15 22:23:24 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
16 22:24:17 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec


Python
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Total value of the inventory: rs5810400.00
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
powershell
Python
Python
Python
Python
PYTHON
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnect...
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
TIMELINE
Run Testcases
Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier
Server Tools Scripting Help
Query 1
Limit to 1000 rows
1423 SELECT DISTINCT s.SuspectID, s.Name
1424 FROM Suspect s
1425 JOIN Crime c ON s.CrimeID = c.CrimeID
1426 WHERE c.IncidentType IN ('Robbery', 'Assault');
1427 create database sis;
1428
1429
1430 use Techshop;
1431 show tables;
1432 select * from products;
1433 select * from inventory;
1434 select * from orderdetails;
SQL Additions
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Result Grid
InventoryID ProductID QuantityInStock LastStockUpdate
1 1 10 2023-10-12
2 2 15 2023-01-02
3 3 9 2022-12-09
4 4 8 2023-10-05
5 5 12 2022-11-20
6 6 23 2023-02-09
7 7 30 2023-03-10
8 8 25 2022-10-19
9 9 5 2023-05-04
10 10 10 2023-10-10
inventory 15
Apply
Revert
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
16 22:24:17 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
17 22:24:54 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec


Python
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
Low stock products:
ProductID: 9, QuantityInStock: 5
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
PYTHON
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnect...
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
TIMELINE
Run Testcases
Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier
Server Tools Scripting Help
Query 1
SELECT DISTINCT s.SuspectID, s.Name
FROM Suspect s
JOIN Crime c ON s.CrimeID = c.CrimeID
WHERE c.IncidentType IN ('Robbery', 'Assault');
create database sis;
use Techshop;
show tables;
select * from products;
select * from inventory;
select * from orderdetails;
Result Grid
InventoryID ProductID QuantityInStock LastStockUpdate
1 1 10 2023-10-12
2 2 15 2023-01-02
3 3 9 2022-12-09
4 4 8 2023-10-05
5 5 12 2022-11-20
6 6 23 2023-02-09
7 7 30 2023-03-10
8 8 25 2022-10-19
9 9 5 2023-05-04
10 10 10 2023-10-10
Output
Action Output
16 22:24:17 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
17 22:24:54 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
Context Help
Snippets


Python
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python> & C:/Python311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
No products are currently out of stock.
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
PYTHON
.cph
.pytest_cache
.vscode
Assignment 2
Assignment1
__pycache__
Customers.py
DatabaseConnect...
Exception.py
Inventory.py
Main.py
OrderDetails.py
Orders.py
Products.py
codes
coding challenge
Coding_challenge2_...
oops
pythonTesting
src
mam'sCode.py
OUTLINE
TIMELINE
Run Testcases
Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live Prettier
Server Tools Scripting Help
Query 1
SELECT DISTINCT s.SuspectID, s.Name
FROM Suspect s
JOIN Crime c ON s.CrimeID = c.CrimeID
WHERE c.IncidentType IN ('Robbery', 'Assault');
create database sis;
use Techshop;
show tables;
select * from products;
select * from inventory;
select * from orderdetails;
Result Grid
InventoryID ProductID QuantityInStock LastStockUpdate
1 1 10 2023-10-12
2 2 15 2023-01-02
3 3 9 2022-12-09
4 4 8 2023-10-05
5 5 12 2022-11-20
6 6 23 2023-02-09
7 7 30 2023-03-10
8 8 25 2022-10-19
9 9 5 2023-05-04
10 10 10 2023-10-10
Output
Action Output
16 22:24:17 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
17 22:24:54 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
Context Help
Snippets


Python
EXPLORER
PYTHON
Inventory.py
Exception.py
Main.py
Assignment1 > Main.py > ...
# inventory_manager.remove_From_Inventory(2,301)
# inventory_manager.update_Stock_Quantity(2,301)
# inventory_manager.Is_Product_Available(2,3)
# inventory_manager.Get_InventoryValue()
# inventory_manager.List_LowStock_Products(8)
# inventory_manager.List_OutOf_StockProducts()
inventory_manager.List_All_Products()
db_connector.close_connection()
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
thon311/python.exe c:/Users/krish/OneDrive/Documents/Python/Assignment1/Main.py
Connected to MySQL database
Connected to MySQL database
All products in the inventory:
ProductID: 1, QuantityInStock: 10
ProductID: 2, QuantityInStock: 15
ProductID: 3, QuantityInStock: 9
ProductID: 4, QuantityInStock: 8
ProductID: 5, QuantityInStock: 12
ProductID: 6, QuantityInStock: 23
ProductID: 7, QuantityInStock: 30
ProductID: 8, QuantityInStock: 25
ProductID: 9, QuantityInStock: 5
ProductID: 10, QuantityInStock: 10
Connection closed
PS C:\Users\krish\OneDrive\Documents\Python>
powershell
Python
Python
Python
Python
OUTLINE
TIMELINE
Run Testcases
Spaces: 4
UTF-8
CRLF
Python 3.11.5 64-bit
Go Live
Prettier
Server
Tools
Scripting
Help
Query 1
Limit to 1000 rows
SELECT DISTINCT s.SuspectID, s.Name
FROM Suspect s
JOIN Crime c ON s.CrimeID = c.CrimeID
WHERE c.IncidentType IN ('Robbery', 'Assault');
create database sis;
use Techshop;
show tables;
select * from products;
select * from inventory;
select * from orderdetails;
SQL Additions
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Result Grid
Filter Rows:
Edit:
Export/Im
InventoryID
ProductID
QuantityInStock
LastStockUpdate
1 1 10 2023-10-12
2 2 15 2023-01-02
3 3 9 2022-12-09
4 4 8 2023-10-05
5 5 12 2022-11-20
6 6 23 2023-02-09
7 7 30 2023-03-10
8 8 25 2022-10-19
9 9 5 2023-05-04
10 10 10 2023-10-10
Result Grid
Form Editor
Field Types
Query Stats
inventory 15
Apply
Revert
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
16 22:24:17 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
17 22:24:54 select * from inventory LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec