Eric Guttmann

Files:
db_main.py
modules.py
assigment4.sql

This assignment uses the previous data from assignment 2 to build a CLI Ecommerce database. The order of design was this: SQL database and tables, the main layout, the functions, and at the end, the stored procedures. While the design was made this way, when it came to actually coding, I started with the stored procedures and worked backwards. After that, I tried to implement the stored procedures into the functions in the module.py file. After that, I first tried to establish a connection to the database and then make the layout of the menu. The project was made with incremental design in mind.

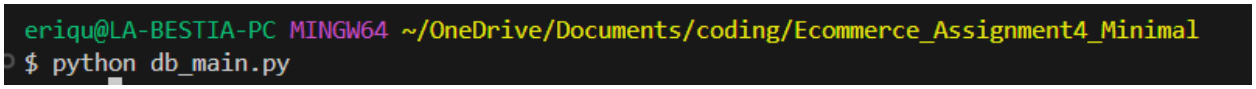
Using Incremental design, I was able to debug much more efficiently. The first error I came across was the database connection. This step involves making the connection from pgAdmin and than in Python establishing a connection. This part took more research, since I had to make sure my server was running properly and the login credentials matched. Even when the database was connected, I had to make sure my Python file accurately retrieved the data. After the connection was established, I started working on the menu options, just making the layout so I could plug the functions later.

Along with the database connection, some of the functions also proved challenging to implement. The hardest function in this entire program was the placeorder function. Even with the placeorder function in SQL, there was still more logic that had to be implemented, such as adding items and calculating the total. Along with that, this project was trickier since debugging involved two languages, and thus 2 layers of complexity. Some debugging required trial and error, and back-and-forth fixing between python and SQL.

The final step in my design process was error handling. This was achieved with try-exception statements to already existing functions. It was easier to test first with the proper inputs then try to make the code error-proof from the start.

Here are some screen shots showing the place orders function demo:

1) Type main filename



2) Enter login credentials

```
eriqu@LA-BESTIA-PC MINGW64 ~/OneDrive/Documents/coding/Ecommerce_Assignment4_Minimal
$ python db_main.py
Host: localhost
Port: 5433
Database name: Ecommerce
Username: postgres
Password:
```

3) Choose a management option

```
Connected successfully.


=== E-COMMERCE CLI (Individual) ===
1. Product Management
2. Customer Management
3. Order Processing
4. Review Management
5. Exit

Choose:
```

```
Choose: 3

=== ORDER PROCESSING ===
1. Place order
2. View all orders
3. View order details
4. Back

Choose:
```

4) Place an order

```
Choose: 1
Customer ID: 1
Shipping Address: 123 top view
Payment Method: PayPal

Order created with ID: 9

Add product to cart? (y/n): y
Product ID: 101
Quantity: 1
Item added.
Add product to cart? (y/n): n

Final Order Total: $999.99

Order complete.
```

5) Verify order was placed by choosing option 3

```
Choose: 3
Order ID: 9
(9, 1, datetime.datetime(2025, 12, 2, 20, 43, 23, 568198), Decimal('999.99'), 'Pending', '123 top view', 'PayPal')
  Item: (101, 'iPhone 15 Pro', 1, Decimal('999.99'), Decimal('999.99'))
```

6) Exit the program

```
=== E-COMMERCE CLI (Individual) ===
1. Product Management
2. Customer Management
3. Order Processing
4. Review Management
5. Exit

Choose: 5
Goodbye!
```