AUBoutique Project Report:

Team members:

1. Georges Khater, 202400576, (55%) , gak22@mail.aub.edu.lb
2. Alexandre Karam, 202401561 , (45%) , abk23@mail.aub.edu

Project GitHub repository:

https://github.com/gkhater/Project-Phase1

Project Demo Link:

https://youtu.be/961QzTC6220?si=tFN8mFuQ_Ghsbual

System Architecture and Communication protocol:

1. System Architecture:

   Client: The client-side application handles user interactions. Users can sign up, log in, browse, buy, and sell products, as well as send messages to other users.

   - Server: The server manages user accounts, products listing, and transactions. It maintains a database of users and products, along with active user sessions, facilitating communication and data exchange between clients.

2. Communication Protocol

   The communication between client and server is based on a TCP connection:

   - Initialization: Clients establish a TCP connection to the server, providing options for registration or login.
   - Commands and Response: Clients send commands and receives structured responses from the server.
   - Session Management: Server keeps track of active sessions and user status.

   Commands supported in communication include:

   - User commands: SIGN UP, LOG IN, LOGOUT
   - Messaging Commands: TEXT [username] [messages]
   - Products Commands: BUY [Product id], ADD [name] [price] [description], PRODUCTS, SOLD
   - Status commands: CHECK [username], ONLINE

Project Features:

User registration	New users can sign up with name, email, username, and password.
User login	Existing users log in with username and password
Messaging between users	Users can send messages to other online users
Product listing	Users can view all products available for purchase
Product addition	Users can add new products for sale
Product purchase	Users can buy listed products
View sold products	Users can view items they have sold
Online user status check	Users can check if another user is online
Help Command	Lists all available commands
Logout	Ends user session and logs them out

Task Breakdown Table:

TASK	TEAM MEMBER RESPONSIBLE	PERCENTAGE CONTRIBUTION
System Architecture Design	Alexandre	5%
Protocol Design	Georges	15%
Client-Side Development	Georges - Alexandre	10% - 10% (T: 20%)
Server-Side Development	Georges - Alexandre	20% - 15% (T:35%)
Testing and Debugging	Georges - Alexandre	5% - 5% (T:10%)
Documentation	Georges - Alexandre	5% - 10% (T: 15%)

Overall Tasks	Georges – Alexandre	55% - 45% (T: 100%)

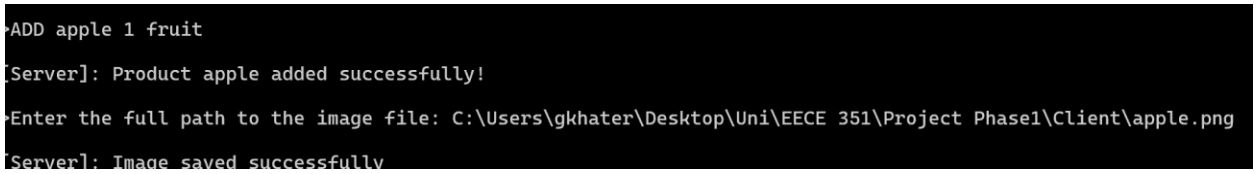
Description of implementation:

- Registration: The server verifies that the username consists of valid characters. The system also checks if the username already exists in the database. If everything is fine, the user is added to the SQLite database.
- Login: The server checks the given credentials in the database.
- Adding product: The user provides if all fields are provided the product is given a unique ID and added to the database. The user is also asked to provide an image path for the product.
- Viewing products: Retrieves a list of items with a count greater than 0. Formats and sends the list back to the user with ID, name, description, price, and seller name.
- Buying products: If the product is available the count is set to 0, and updates the buyer field with the buyers username.
- Viewing sold products: fetches from the table all the products sold with the seller's username.
- Messaging: The server checks if the recipient is online. If he is online, the message is forwarded with the sender's information. If he is offline, the sender is informed that the message can't be delivered.
- User status: Users can check if users are online by referencing online_users.
- LOGOUT: Removes the user from online_users, and closes the connection with them.

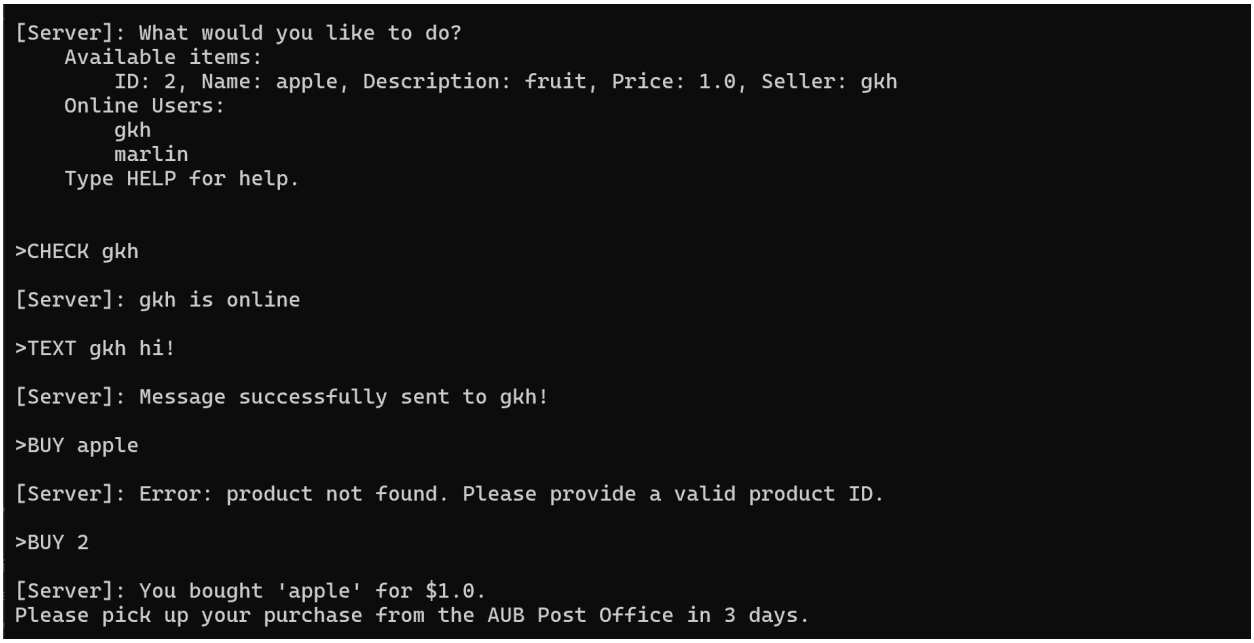
NOTE: We are storing the picture of the product after Adding, but since we are still using the command line in this phase , we have no way of showing it while listing. We do not need to store the file to the image in the database since it is a fixed pattern ".\Images\ID_{ID}.png" (ID is stored in the DB).

Snapshots of the application depicting the main features:

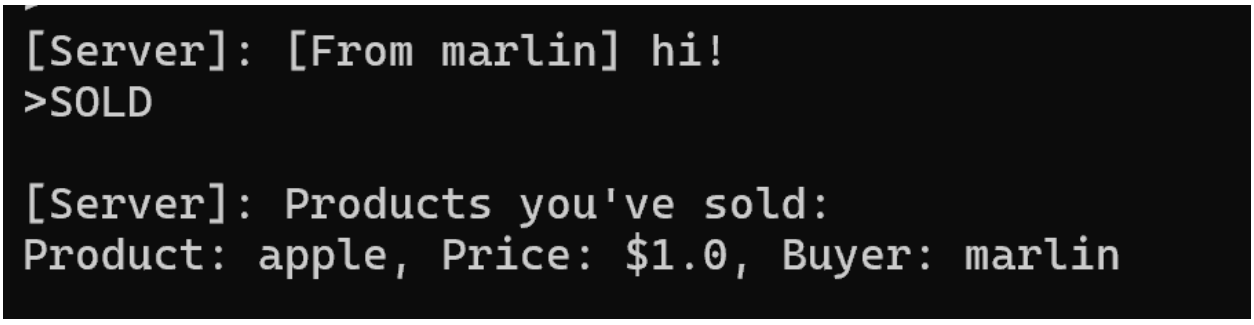
User 1:



User 2:



User 1:



Server:

```
C:\Users\gkhater\Desktop\Uni\EECE 351\Project Phase1\Server>python main.py
Enter your port number: 5001
Listening on port 5001
Server listening on port 5001...
Connection established with ('127.0.0.1', 58273)
Connection established with ('127.0.0.1', 58281)
User khater has disconnected.
Connection established with ('127.0.0.1', 58293)
User khater has disconnected.
User marlin has disconnected.
```