Final Report

110550162 洪慎廷

Topic: Design and Implement Simple Chat Server

Program Description:

My program consists of three python files “chat_server.py”, “datastructures.py”, “static_messages.py”. The descriptions of these files are below.

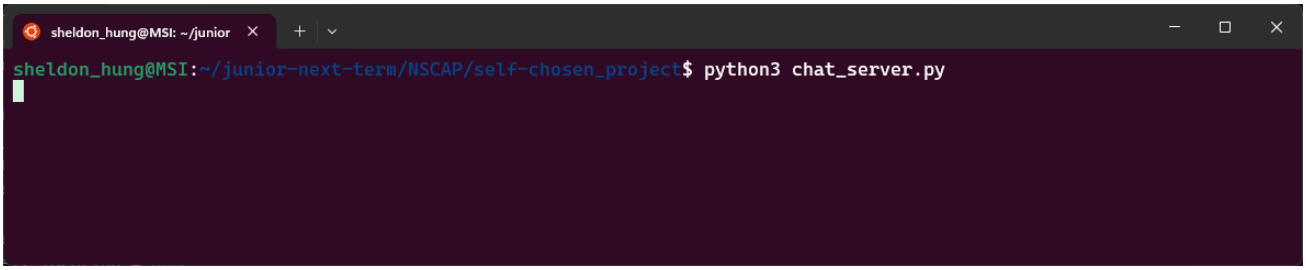
- chat_server.py
  - The main program of the server. It is responsible for handling multiple clients’ TCP connection, doing the authentication of each client, and sending and receiving commands or messages to and from the clients.
  - For handling multiple clients. I use a monitor “select” to handle multiple clients.
  - The server interacts with the clients by receiving user command from the clients and send the corresponding message back to the clients. The program will parse the received command and do some actions. The detail of the commands and actions will be described at the next section.
- datastructures.py
  - All the objects that will be used by the main program are defined in here. There are two objects that I defined, “UserInfo” and “ChatRoom”.
  - “UserInfo”: It is an object that stores a client’s information, including authentication data (username, password), user profile (age, birthday, job, phone_number, self_desription), and some control data of the client.
  - “ChatRoom”: It is an object that stores the information of a chatroom, for example, the name of the chatroom, the owner of the chatroom, is the chatroom private or public, who can or cannot enter the chatroom. Also, it has some functions for handling user commands that may modify the chatroom’s data.

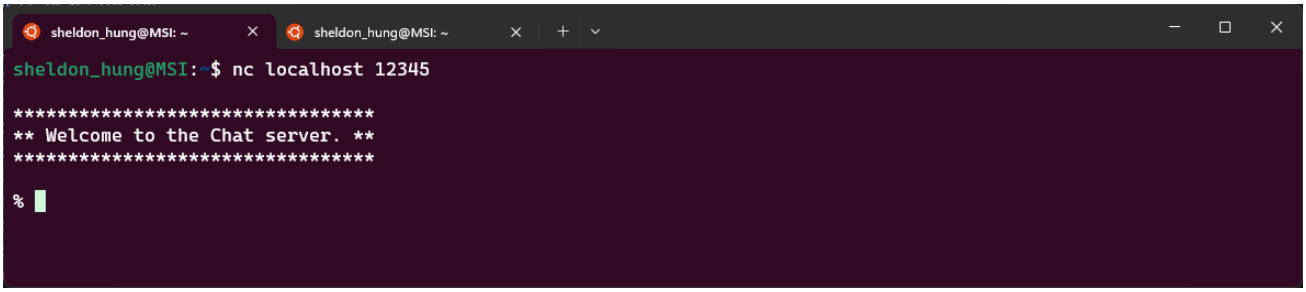
- static_messages.py
  - Just simply stores some static string that may be sent to the users.

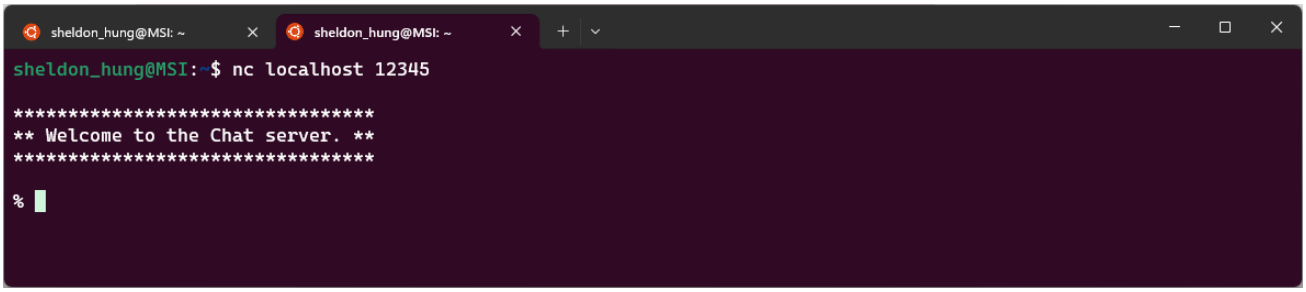
Final Results:

Below, I will display the final results by running the server program and open multiple bashes to play as the clients and interact with the server. Additionally, I will describe the commands and messages the clients send and the respond from the server.

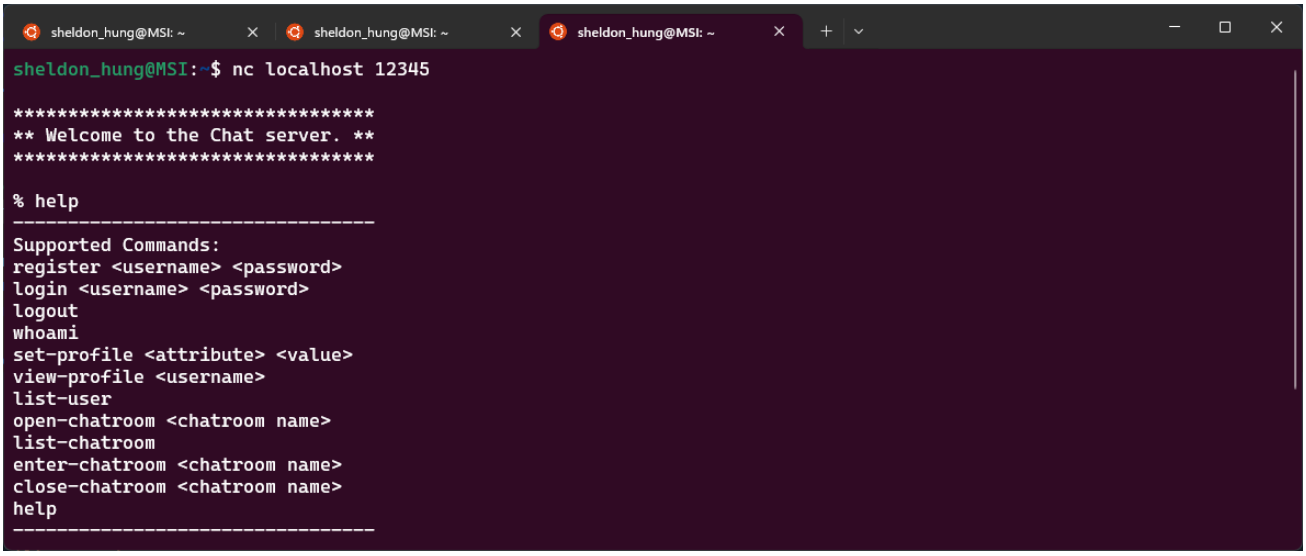
- Run the server and open two bashes playing as two clients (Tom and Bob), connecting to the server. The server will show a welcome message once a client is connected.



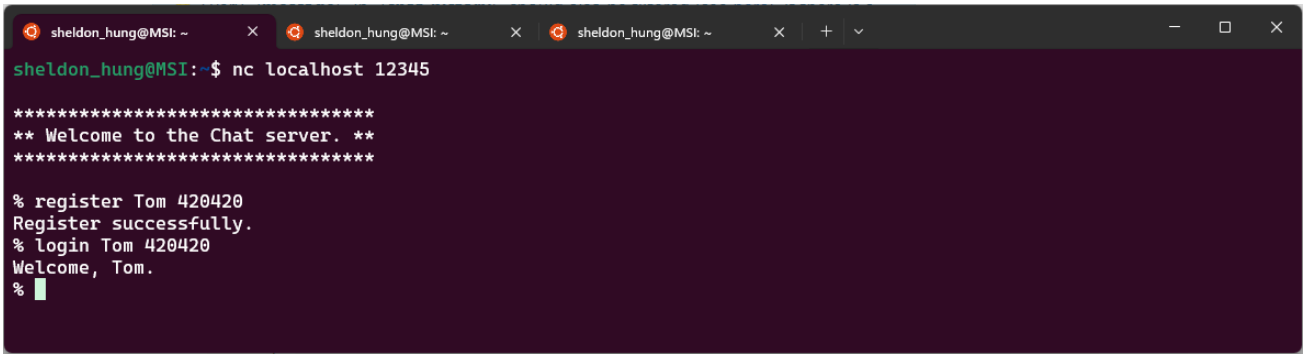


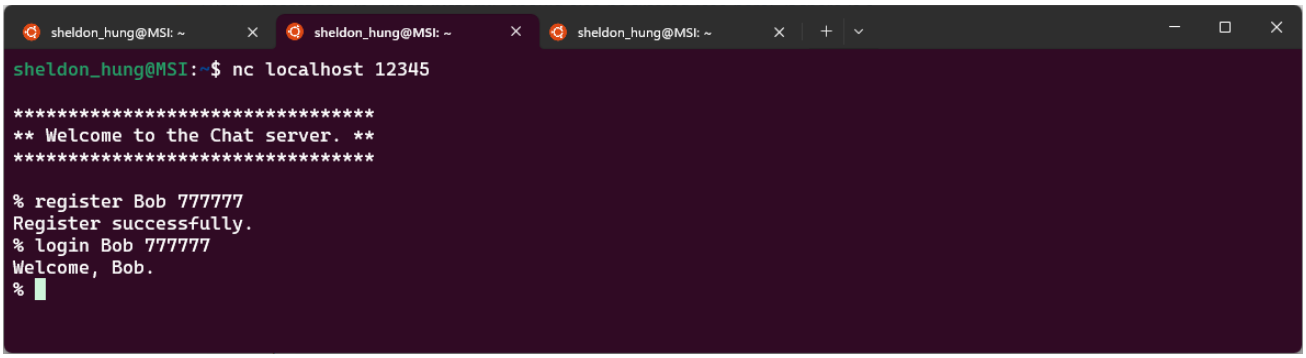


- Supplement: Type the command “help” to see all the commands the server supports. For input any invalid or authorized command, the server will send the correspond error messages.

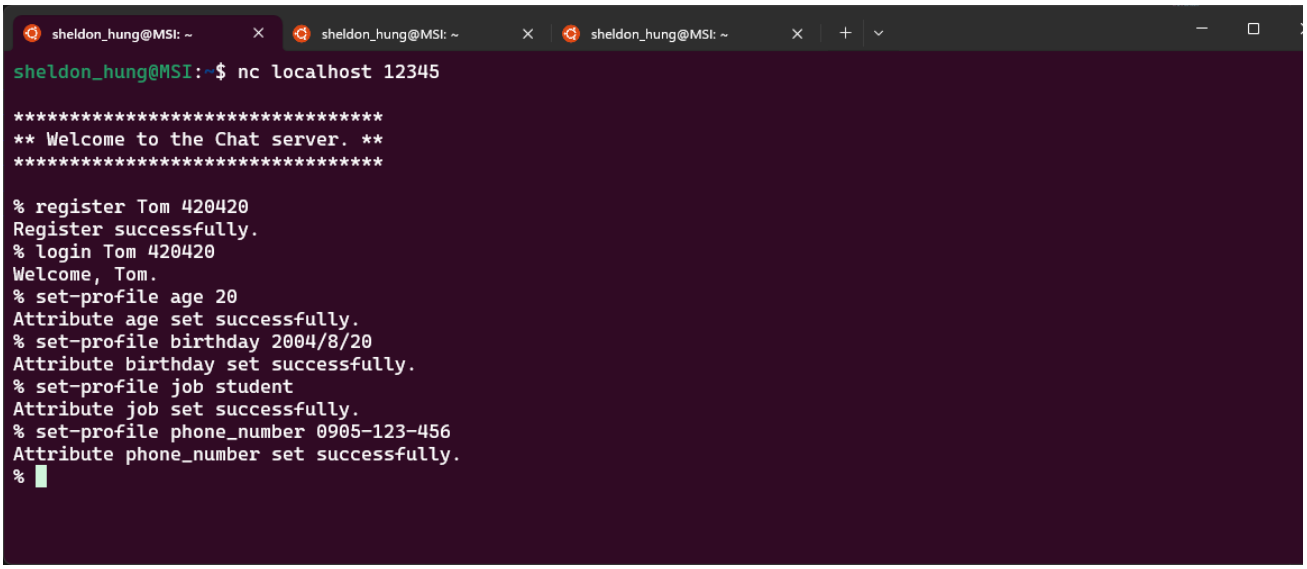


- The two clients (Tom and Bob) register an account with password and login the account, using the command “register {username} {password}” and “login {username} {password}”.

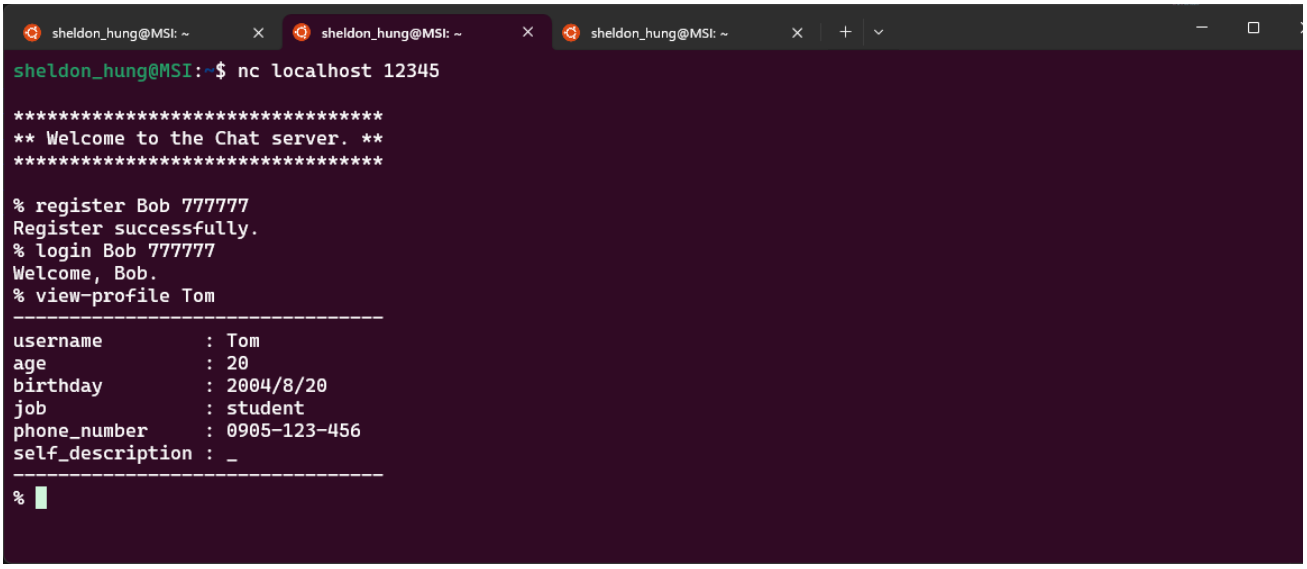




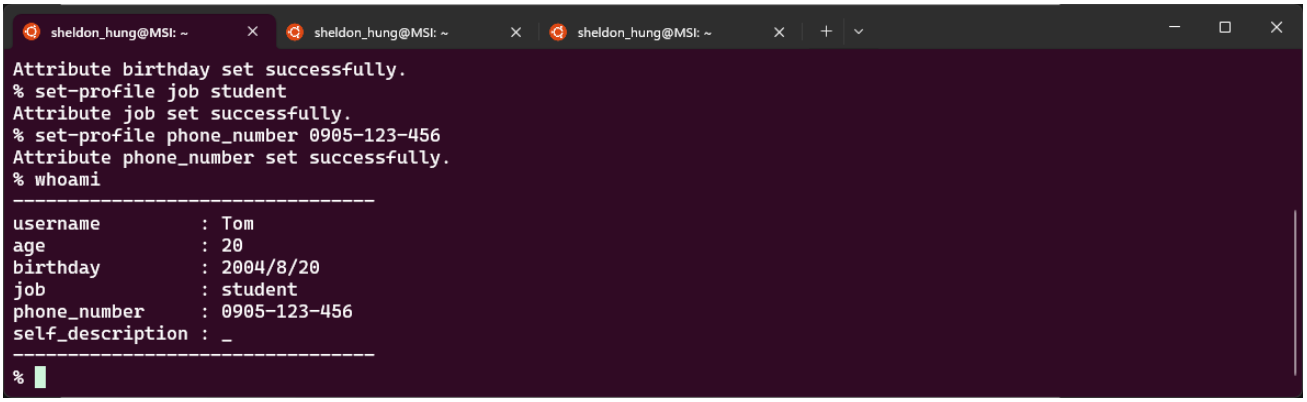
- Tom uses the command “set-profile {attribute} {value}” to set his own profile.



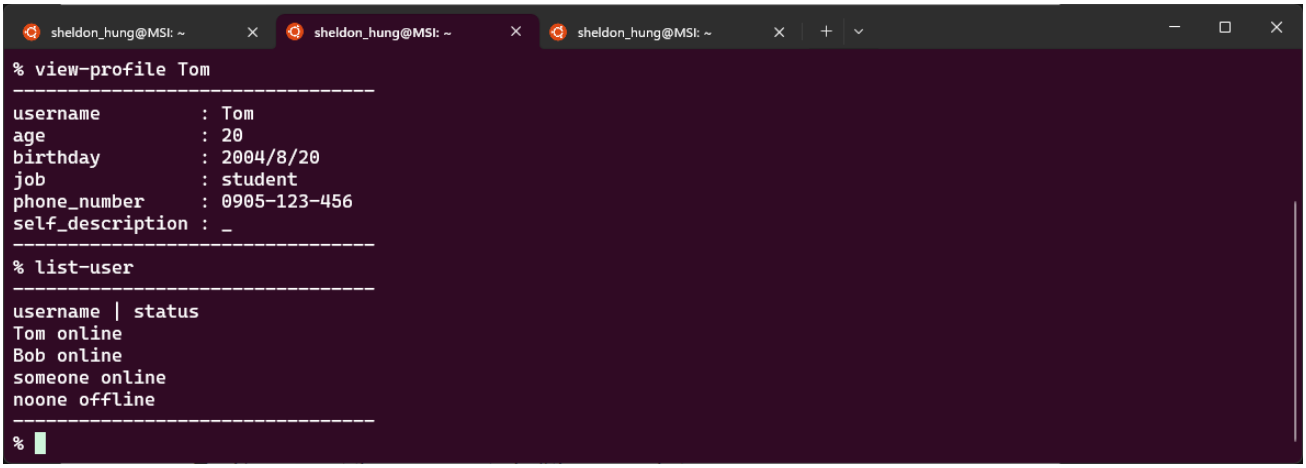
- Bob uses the command "view-profile {username}” to view Tom’s profile.



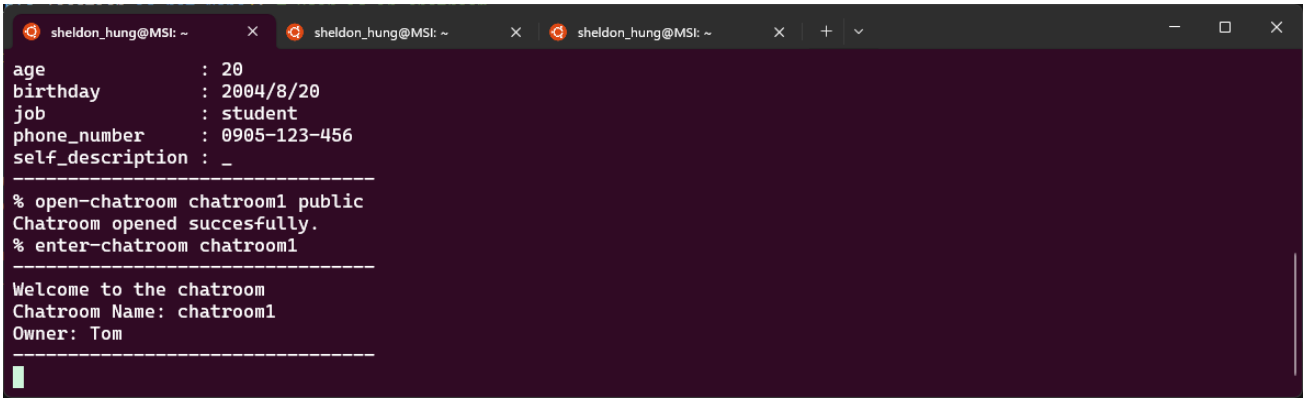
- Tom can also view his own profile by using the command “whoami”.



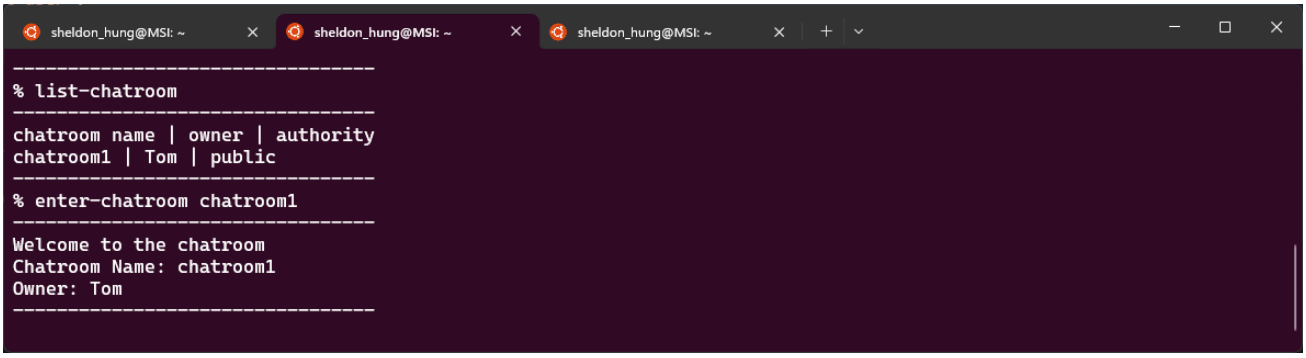
- Bob uses the command “list-user” to view all the users in the server. The server will print all the username and if they are online or offline.



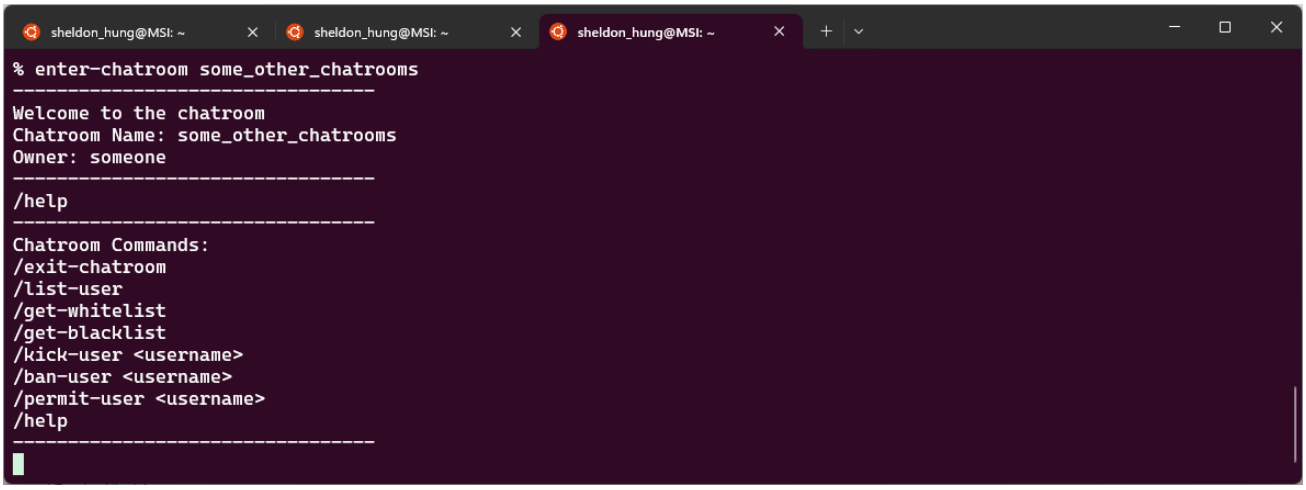
- Tom opens a public chatroom by using the command “open-chatroom {chatroom name} {public or private}. A public chatroom means that anyone can enter in the chatroom if they aren’t in the blacklist. Then Tom enters the chatroom by using the command “enter-chatroom {chatroom name}”.



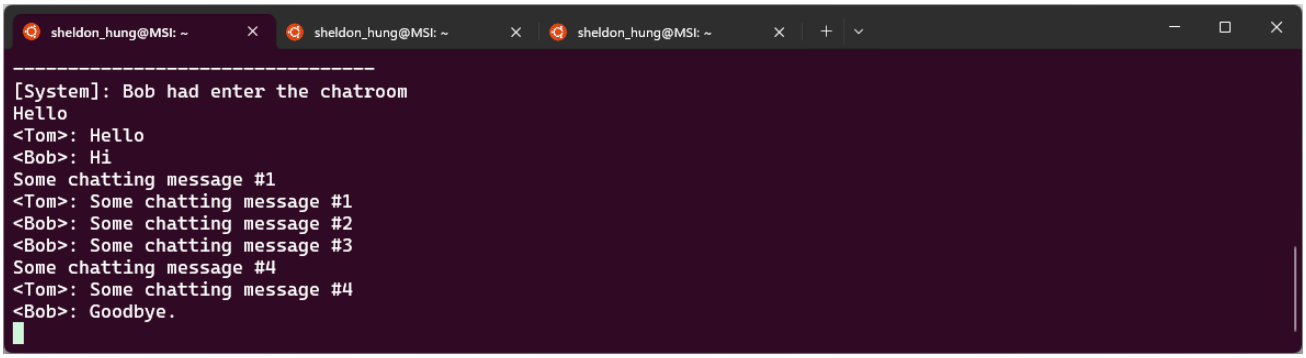
- Bob uses the command “list-chatroom” to view all the chatrooms in the server. The server will print out the chatroom name, chatroom owner, and if the chatroom is public or private. Then use the command “enter-chatroom {chatroom name}” to enter Tom’s chatroom.

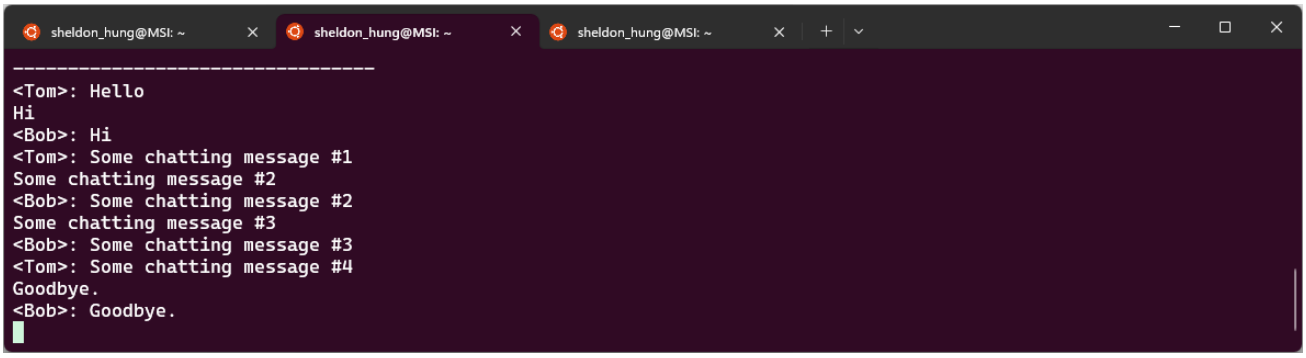


- Supplement: Type the command "/help" to see all the chatroom commands the server supports. In the chatroom, commands must start with a prefix "/" to let the server know the client is sending a command, or the sever will view it as a chatting message.

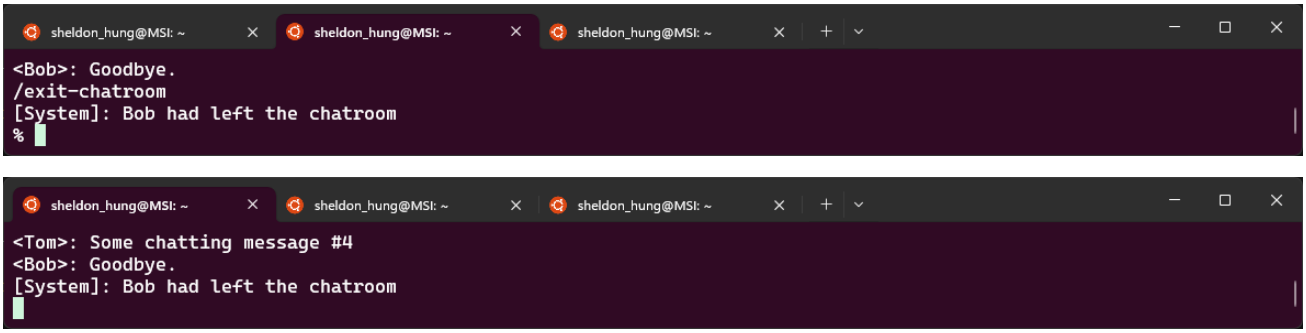


- When Bob enters Tom's chatroom, the server will inform all the other users (Tom) in the chatroom that Bob enters the chatroom. Then Tom and Bob can start chatting by just typing the chatting message.

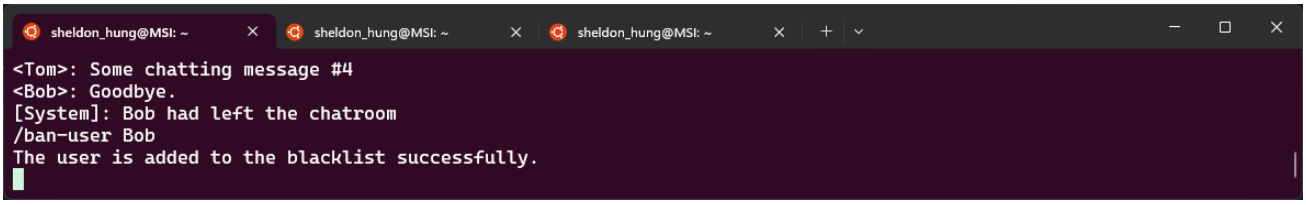


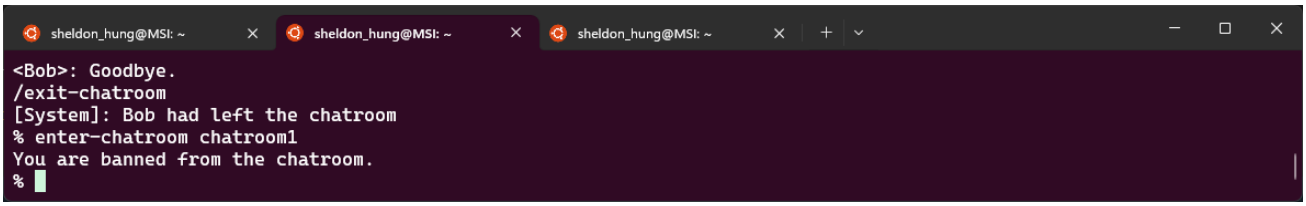


- Bob uses the chatroom command "/exit-chatroom" to leave Tom's chatroom. The server will inform all the other users (Tom) in the chatroom that Bob leaves the chatroom.

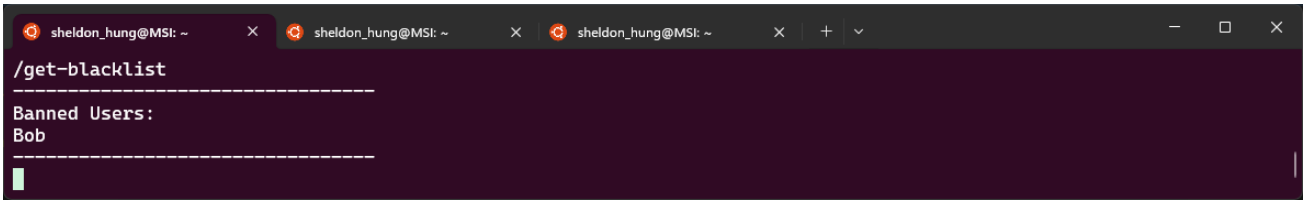


- Tom adds Bob into the chatroom's blacklist by using the command "/ban-user {username}". Then Bob cannot enter Tom's chatroom.

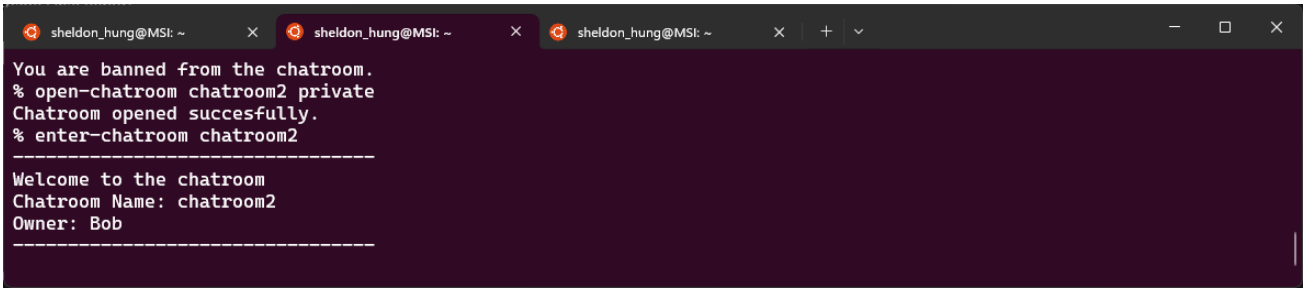




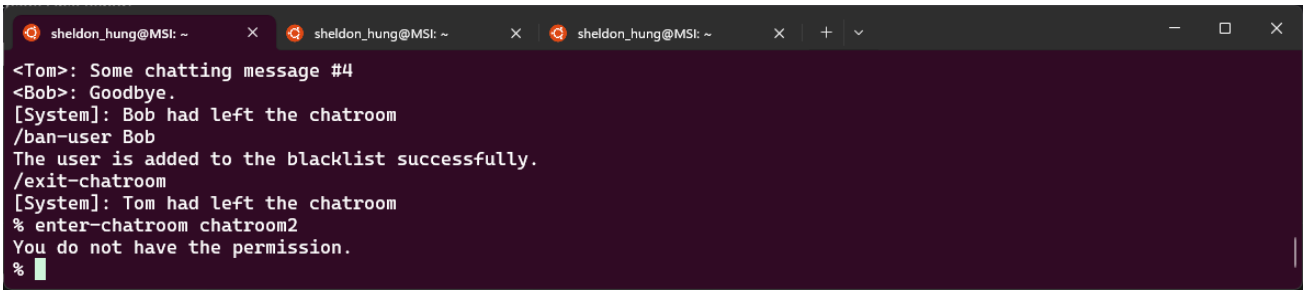
- Tom can view the chatroom's blacklist by using the command "get-blacklist".



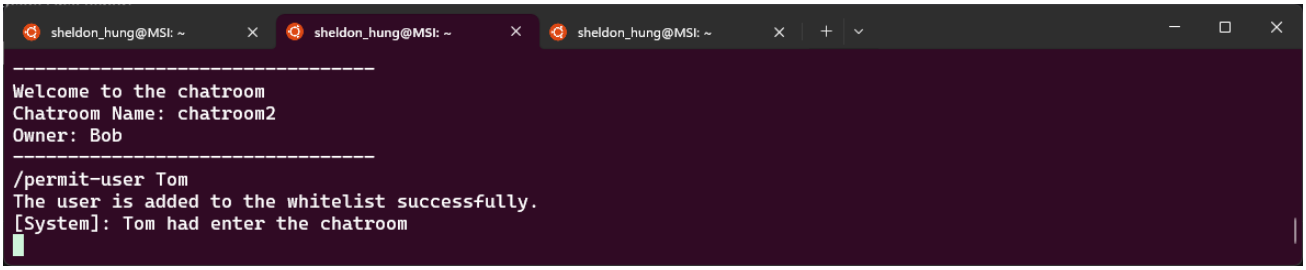
- Bob opens a private chatroom by using the command “open-chatroom {chatroom name} {public or private}. A private chatroom means that only the chatroom owner or user in the whitelist can enter the chatroom. Then Bob enters the chatroom by using the command “enter-chatroom {chatroom name}”.

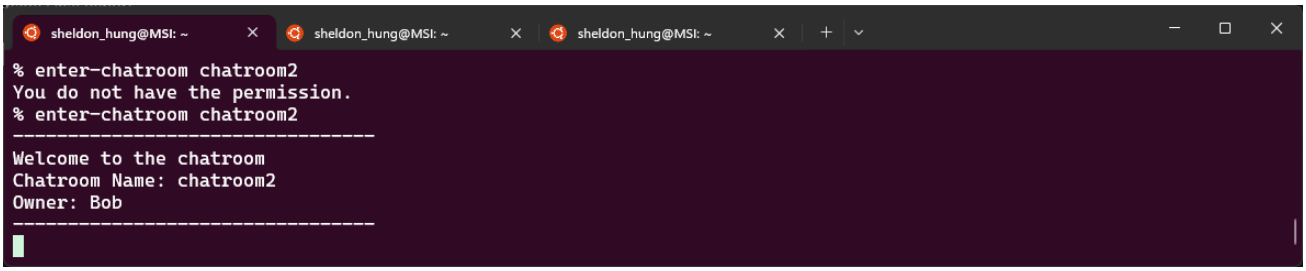


- Tom leaves his own chatroom and intend to enter Bob’s private chatroom. However, he isn’t in the chatroom’s whitelist. Thus, he can’t enter Bob’s chatroom.

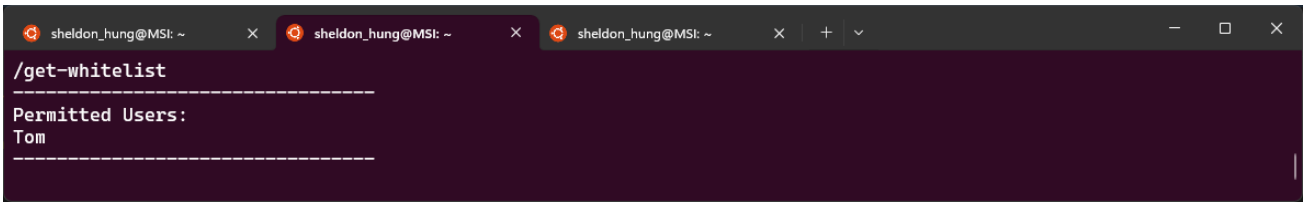


- Bob adds Tom to the chatroom’s whitelist by using the command “/permit-user {username}”. Then Tom can enter the chatroom.

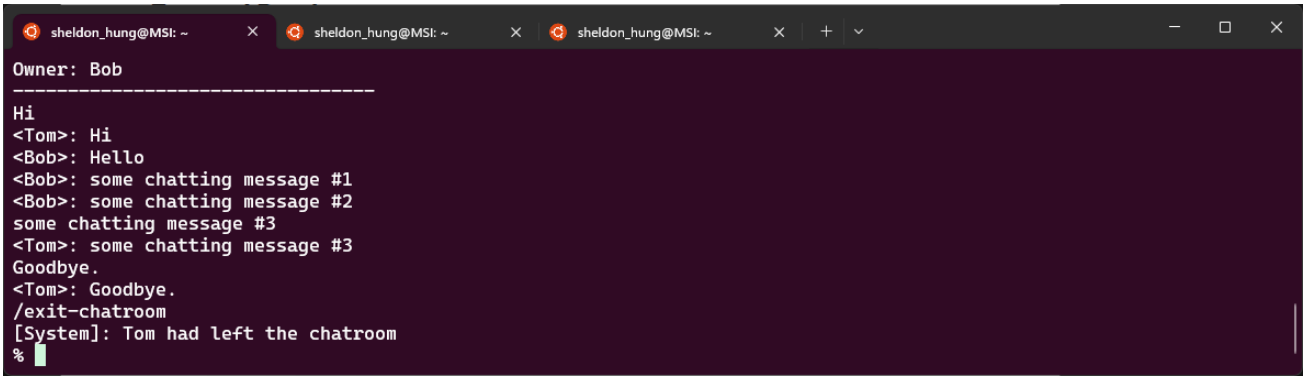


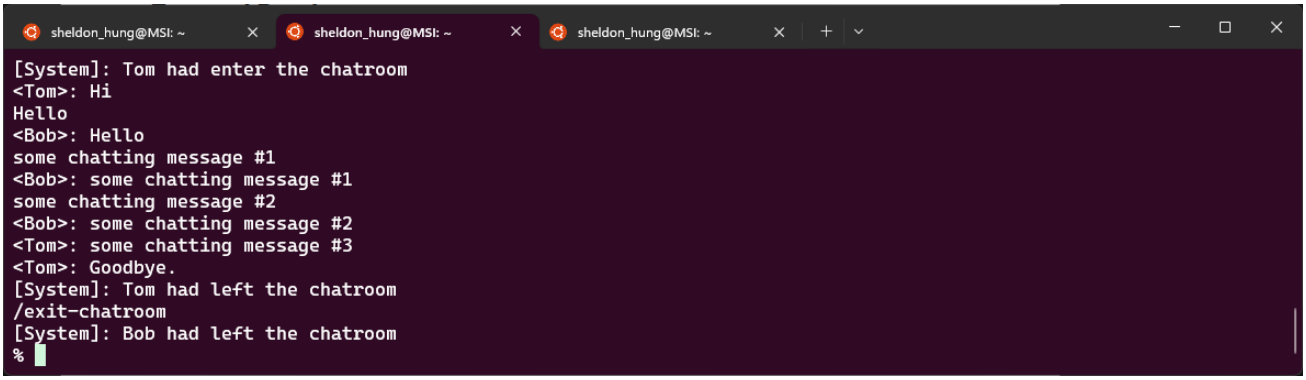


- Bob can view the chatroom's whitelist by using the command "get-whitelist".

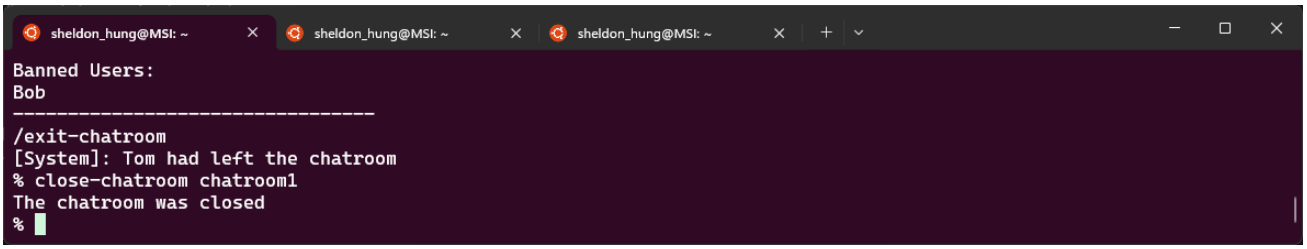


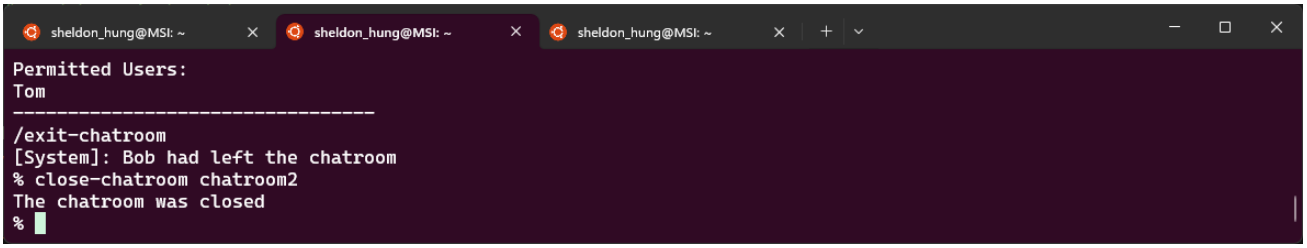
- Bob and Tom chat in the chatroom and then exit the chatroom.



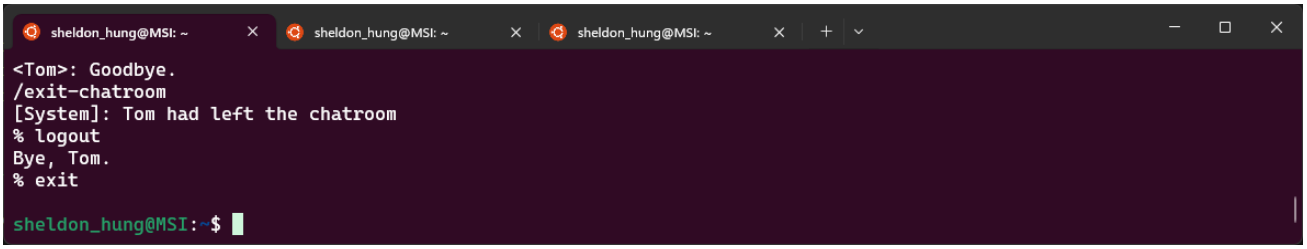


- Tom and Bob close their own chatroom by using the command “close-chatroom”.





- Tom logout his account by using the command “logout”. Then close the connection by using the command “exit”.



- Bob closes the connection by using the command “exit” without logging out the server. The server will still logout Bob implicitly.