1. Explain my code.

There are 4 main parts to my code. The first part is making sure the users connect to the server. The second is the function to send messages. The third is the function for receiving messages. The fourth is the main logic that allows chatting between two people.

I will show each section separately.

Part 1: connect to server.

```
# connect to the server
server = im.IMServerProxy('https://web.cs.manchester.ac.uk/u38174ys/comp28112_ex1/IMserver.php'
# tell the user that he/she is connected to the server
print ('Connected to server')
```

This step is that making a server connected successfully. And print a message to tell the user, you already connected to the server.

Part 2: send message.

```
# send message to another user
def send():

    # input the message from user
    print ('Type your message: ')
    myMessage = input()
    # store the message in the server
    server['message'] = myMessage
    # tell the user that the message is sending
    print ('Sending...')
    # tell the user that he/she is waiting for reply
    print ('Waiting for message...')
```

Every user will use the function to send and receive messages, so I have split the function into two parts.

For send message, the system will ask one user to type the message he/she want to send to another user. After one user input the message, I use server['message'] to save the value to the server. And the system will tell this user to wait a second for another user reply.

Part 3: receive message.

```
# receive message from another user
def recieve():

    # print the message from another user
    print(server['message'])
    # tell the user that the message is recieved
    print ('Message recieved & please reply / no other message')
```

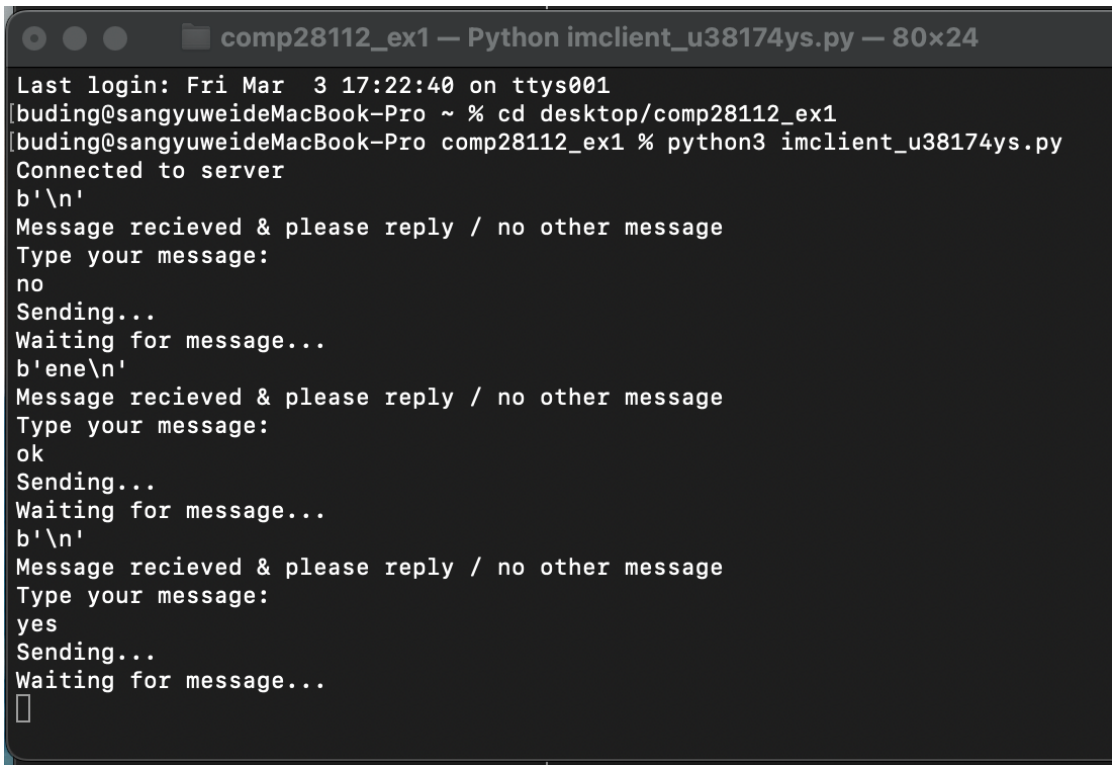
For receive message, the system will print message which saved by server (one user typed). And tell the user message received. If the message is empty, it means no one send message to you. You can send message to another person.

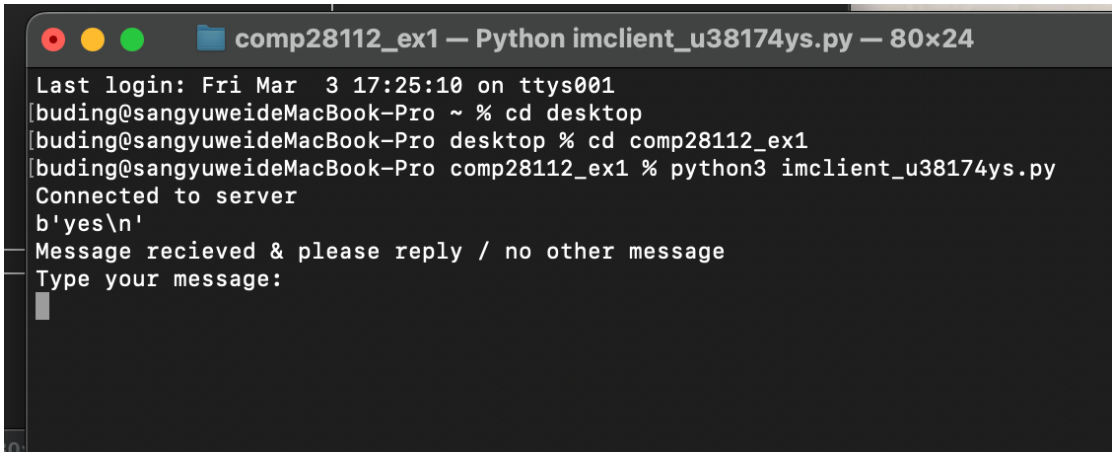
Part 4: main loop.

```
# main function
# check whether there are keys in the server
# if there are no keys in the server, first user sends message directly

if server.keys() == [b'']:
    server.clear()
    send()

# if there are keys in the server, do the following
else:

    while True:

        # if the message is empty, send message
        if server['message'] == b'':
            send()

        # if the message is not empty, receive message
        else:
            recieve()
            server['message'] = b''
            send()

        # stop a while, make sure user can read the message and reply
        time.sleep(15)
```

For main function, it describe in the picture.

2. How to run





For both users, the first step is to find the location of the file and run the .py file with python3. When the first user goes into the program, there is no message in front of it because it is the first person to use it, so it goes straight to the 'type message' screen. When the first user sends the message, the second user goes to the page and receives the message from the first user.

And the system will tell the second user to reply to what you receive.