CN LAB 3

**CS21B2028**

**NITIN REDDY K**

1. Write a Server program with threads, where each thread can handle a single client.

Ensure server get connected to multiple clients. All the client messages to be displayed in server. Provide a mechanism to disconnect the client.

Implement with python socket programming.

# Demonstrate with local loop IP

CODE:

# SERVER:

import socket import threading

IP=socket.gethostbyname(socket.gethostname()) port=200

ADDR=(IP,port)

size=1024 format='utf-8'

disconnect\_msg="Disconn"

def handle\_client(conn,addr):

print(f"[NEW CONNECTION] {addr} connected.") flag = True

while flag:

msg = conn.recv(size).decode(format) if msg == disconnect\_msg:

flag = False

print(f"[{addr}] {msg}")

msg = f"Msg received: {msg}" conn.send(msg.encode(format))

conn.close()

def main():

print("[STARTING] Server is starting...")

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) s.bind(ADDR)

s.listen()

print(f"[LISTENING] Server is listening on {IP}:{port}")

while True:

conn, addr = s.accept()

thread = threading.Thread(target=handle\_client, args=(conn,addr)) thread.start()

print(f"[ACTIVE CONNECTIONS] {threading.activeCount() - 1}")

if \_\_name ==' main\_\_': main()

**CLIENT:**

import socket

IP=socket.gethostbyname(socket.gethostname()) port=200

ADDR=(IP,port)

size=1024 format='utf-8'

disconnect\_msg="Disconn"

def main():

c=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM) c.connect(ADDR)

print("[CONNECTED] client connect to the server at {IP}:{PORT}") flag=True

while flag:

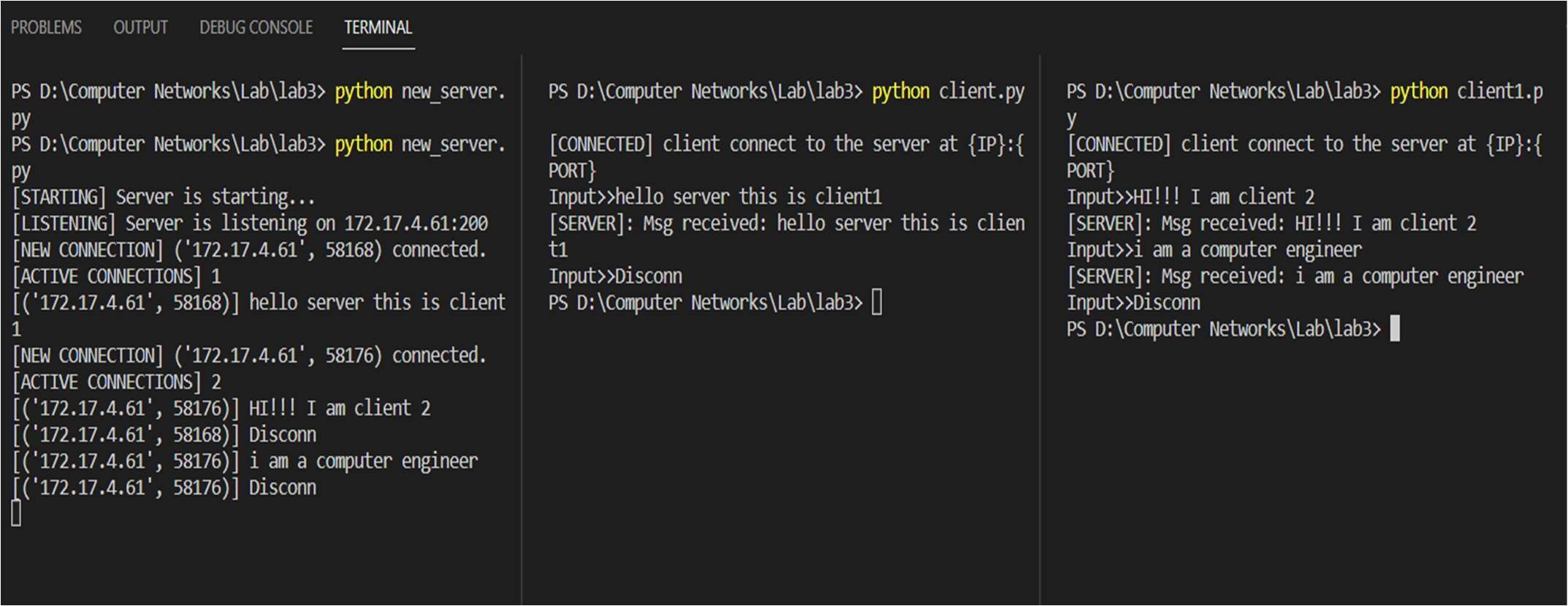
msg=input("Input>>") c.send(msg.encode(format)) if msg==disconnect\_msg:

flag=False else:

msg=c.recv(1024).decode(format) print(f"[SERVER]: {msg}")

if \_\_name ==' main\_\_': main()

# OUTPUT:



Here a single server is able to receive messages from different clients using threads which are running in a infinite loop to until a disconnect msg is sent from client

# Connect with multiple client with different Ips CODE:

**SERVER:**

import socket import threading

IP=socket.gethostbyname(socket.gethostname()) port=200

ADDR=(IP,port)

size=1024 format='utf-8'

disconnect\_msg="Disconn"

def handle\_client(conn,addr):

print(f"[NEW CONNECTION] {addr} connected.") flag = True

while flag:

msg = conn.recv(size).decode(format) if msg == disconnect\_msg:

flag = False

print(f"[{addr}] {msg}")

msg = f"Msg received: {msg}"

conn.send(msg.encode(format)) conn.close()

def main():

print("[STARTING] Server is starting...")

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) s.bind(ADDR)

s.listen()

print(f"[LISTENING] Server is listening on {IP}:{port}")

while True:

conn, addr = s.accept()

thread = threading.Thread(target=handle\_client, args=(conn,addr)) thread.start()

print(f"[ACTIVE CONNECTIONS] {threading.activeCount() - 1}")

if \_\_name ==' main\_\_': main()

# CLIENT:

#client Program

import socket

IP= "192.168.137.119"

(socket.gethostname())

#socket.gethostbyname

PORT= 200 #5566

ADDR = (IP, PORT) SIZE= 1024 FORMAT= "utf-8"

DISCONNECT\_MSG = "Disconn"

def main():

client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) client.connect (ADDR) #connect to the ip address and port print (f

[CONNECTED] Client connected to server at (IP): (PORT)") print(f"[connected] client connect to server at {IP}:{PORT}")

connected = True while connected:#

msg = input("> ") #get the input message from client with ">" prompt client.send(msg.encode(FORMAT)) #sending the message to client

if msg == DISCONNECT\_MSG: #if !Disconnect (use same format) message is sent come out of the while loop

connected = False

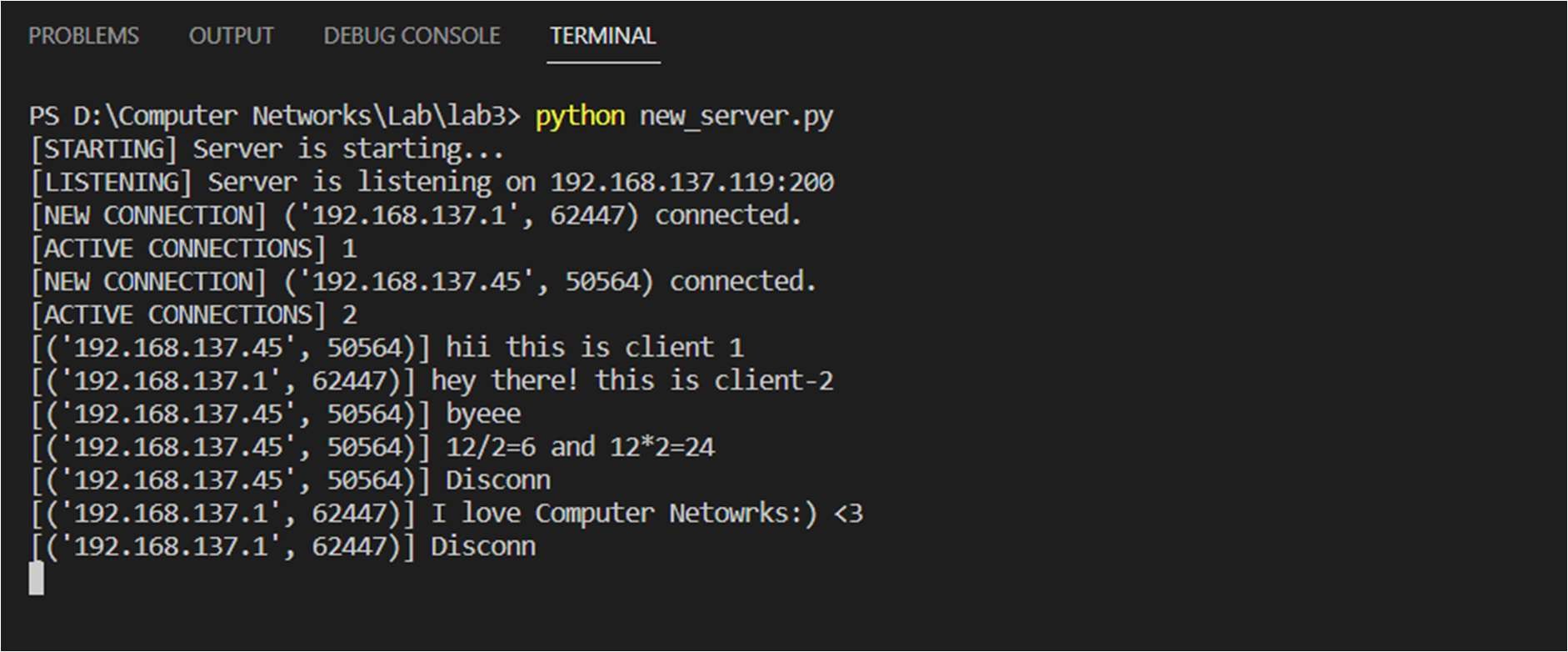
else:

msg = client.recv(SIZE).decode(FORMAT)

print (f" [SERVER] (msg)")#server is sending message

if name ==' main ': main()

**OUTPUT**



Here I provided my ip address to my fellow friends which they kept in the ip variable in client program with same port of mine .Which enables my server to clients and receive messages

# Modify the program where server can send messages to specific client.

**SERVER:**

import socket import threading

IP=socket.gethostbyname(socket.gethostname()) port=200

ADDR=(IP,port)

size=1024 format='utf-8'

disconnect\_msg="Disconn" dic={}

def print\_dict(dic):

for i in dic.items(): print(f"User IP:{i[0]}")

def handle\_client(conn,addr):

print(f"[NEW CONNECTION] {addr} connected.") dic[addr[0]]=conn

flag = True while flag:

flag=True def main():

print("[STARTING] Server is starting...")

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) s.bind(ADDR)

s.listen()

print(f"[LISTENING] Server is listening on {IP}:{port}")

while True:

conn, addr = s.accept()

thread = threading.Thread(target=handle\_client, args=(conn,addr)) thread.start()

print(f"[ACTIVE CONNECTIONS] {threading.activeCount() - 1}") if(threading.active\_count()-1):

while(1):

print("User In active \n") print\_dict(dic) print("\n")

k=int(input("Enter \n 1.Send Msg \n 2.Exit \n")) if(k==1):

s\_ip=input("Enter IP address: ") conn=dic.get(s\_ip,0)

if(conn==0):

print("User does not exists\n") else:

data=input(f"Message {s\_ip}>> ") conn.send(data.encode(format)) if(data==disconnect\_msg):

conn.close()

print(f"{s\_ip} disconnected") dic.pop(s\_ip)

elif k==2:

break else:

print("Invalid Choice")

if \_\_name ==' main\_\_': main()

# CLIENT:

import socket

IP= '192.168.137.119' #socket.gethostbyname (socket.gethostname()) PORT= 200

ADDR = (IP, PORT) SIZE= 1024 FORMAT= "utf-8"

DISCONNECT\_MSG = "Disconn"

def main():

client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) client.connect (ADDR)

print(f"[connected] client connect to server at {IP}:{PORT}")

connected = True while connected:

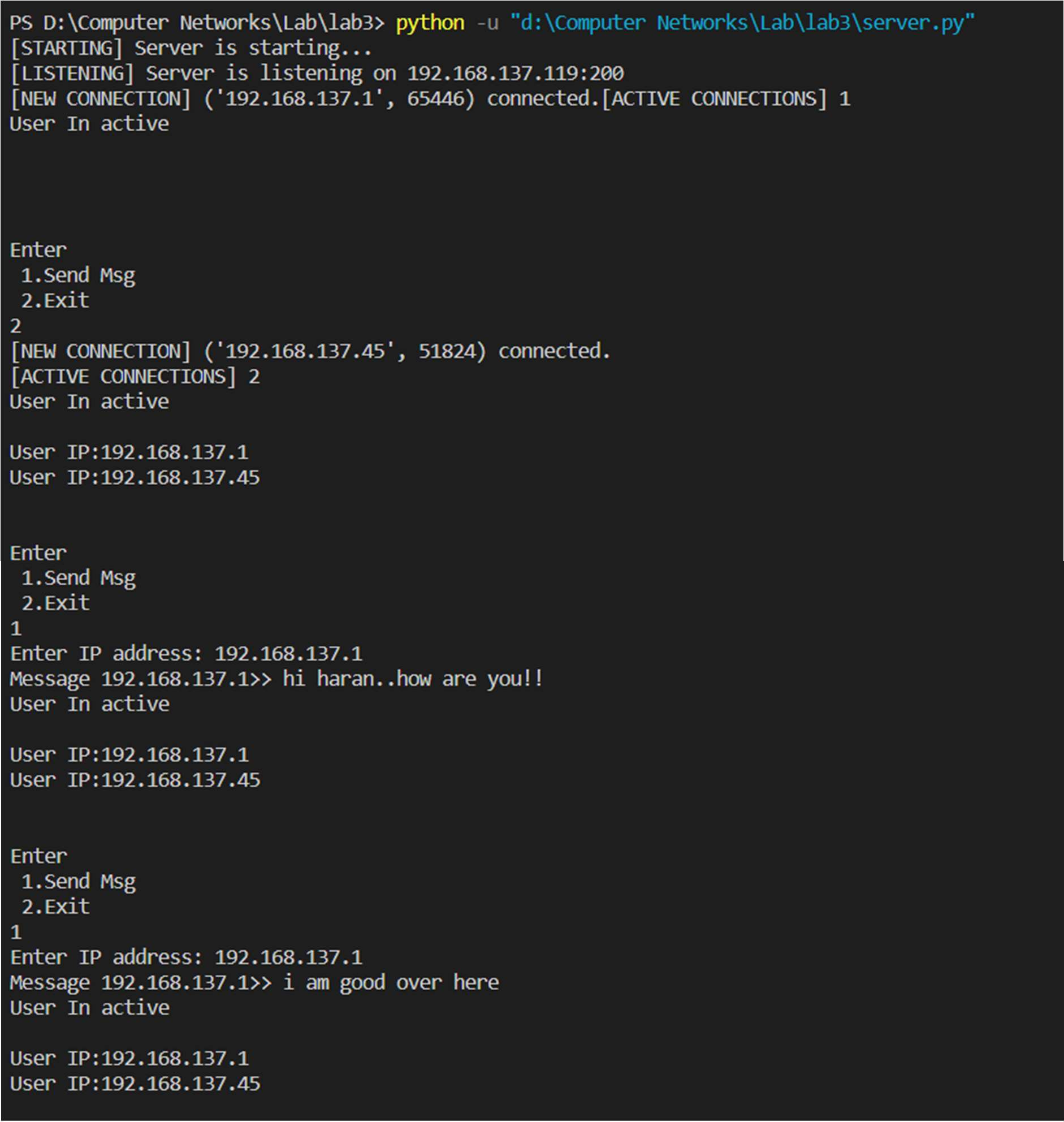
msg = client.recv(SIZE).decode(FORMAT) if(msg==DISCONNECT\_MSG):

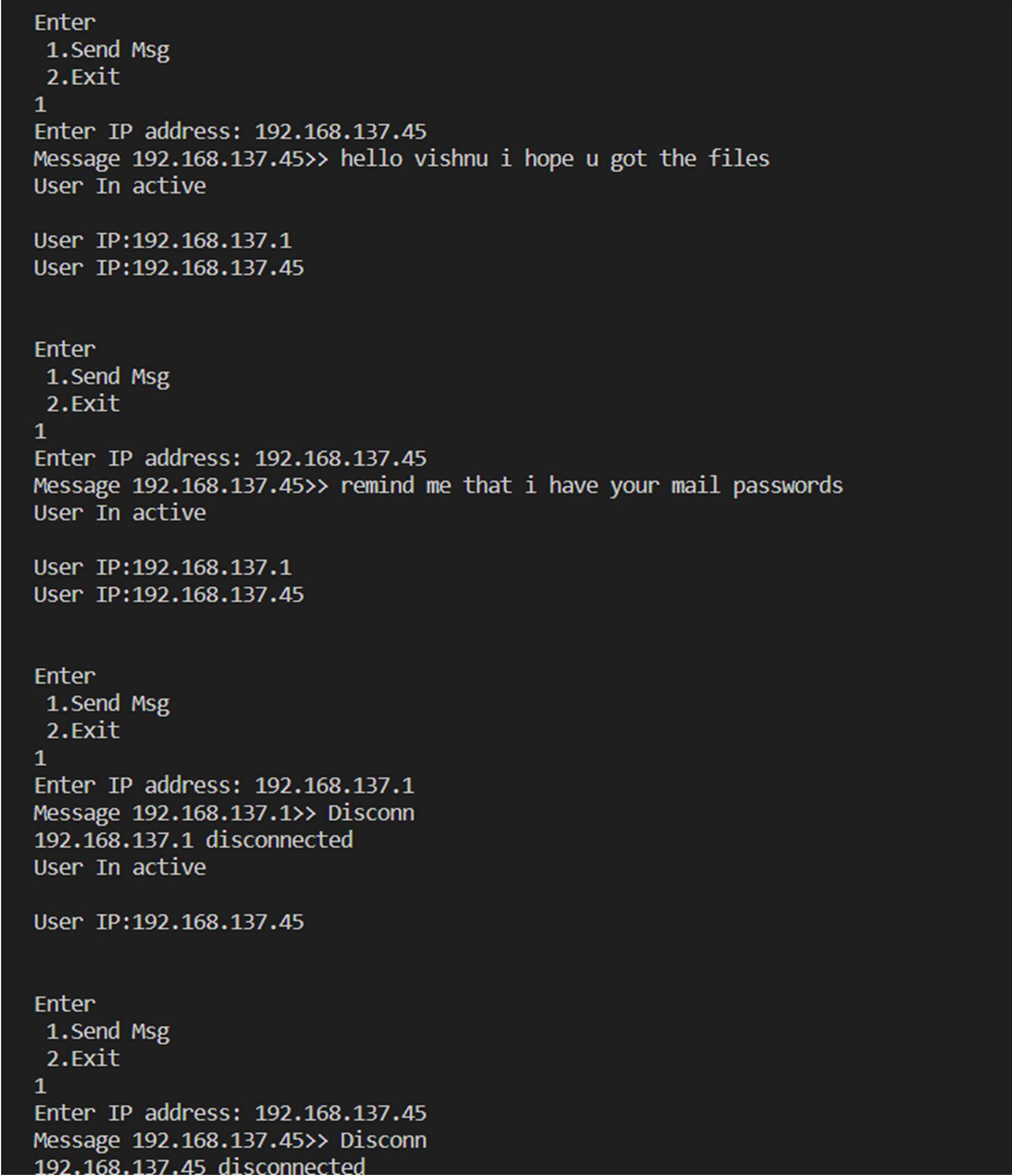
connected=False print("Disconnected")

print (f" [SERVER] {msg}") if \_\_name == " main ":

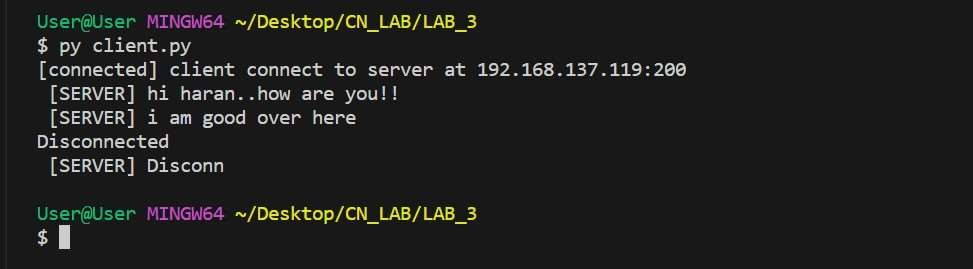
main()

**SERVER OUTPUT:**

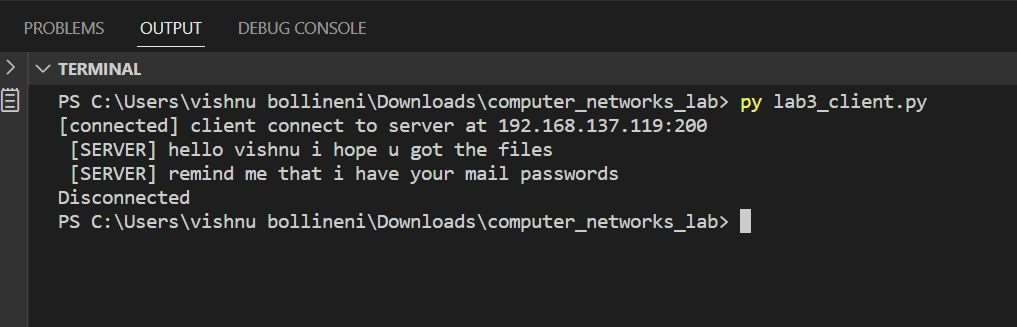




# CLIENT 1 OUPUT



**CLIENT 2 OUTPUT:**



# EXPLANATION:

I am running threads to connect with my clients using threading lib

When ever a new user comes to my server I am appending that client IP address and connection to my client\_dictionary .An infinite loop will run in the main function to provide choice for server program to send the messages to specific client by taking the IP address to whom he want to send the message using conn.send() data required to be sent will be sent

Client will receive the message in the terminal if the received message is Disconn it will get checked in the server program and will disconnect that particular client. “Diconnected from the server “ will be printed in the clients console