Server

```
import socket
import threading

# Function to handle communication with each client
def handle_client(client_socket, client_address):
    print(f"New connection: {client_address}")
    while True:
        try:
            message = client_socket.recv(1024)
            if not message:
                break  # No more data from the client
            print(f"Message from {client_address}: {message.decode('utf-8')}")
            broadcast(message, client_socket)  # Send message to all clients
        except:
            break

    client_socket.close()
    print(f"Connection closed: {client_address}")

# Function to broadcast messages to all connected clients
def broadcast(message, sender_socket):
    for client in clients:
        if client != sender_socket:
            try:
                client.send(message)
            except:
                clients.remove(client)

# Set up the server socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(("0.0.0.0", 5555))  # Bind to all interfaces on port 5555
server_socket.listen(5)  # Listen for up to 5 incoming connections

clients = []

print("Server started. Waiting for clients...")

# Accept client connections and handle them in separate threads
while True:
    client_socket, client_address = server_socket.accept()
    clients.append(client_socket)
    threading.Thread(target=handle_client, args=(client_socket, client_address)).start()
```

#client

```
# -*- coding: utf-8 -*-
"""
Created on Wed Feb  5 11:32:47 2025

@author: STUDENT
"""

import socket
import threading

# Function to receive messages from the server
def receive_messages(client_socket):
    while True:
        try:
            message = client_socket.recv(1024)
            print(f"\nNew message: {message.decode('utf-8')}")
        except:
            print("Connection lost.")
            break

# Set up the client socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect(("127.0.0.1", 5555))  # Connect to the server

# Start the thread for receiving messages
threading.Thread(target=receive_messages, args=(client_socket,)).start()

# Send messages to the server
while True:
    message = input()
    if message:
        client_socket.send(message.encode('utf-8'))
```

Server output

```
In [1]: runfile('C:/Users/STUDENT.LAB03-03/.spyder-py3/server -gc.py', wdir='C:/Users/
STUDENT.LAB03-03/.spyder-py3')
Server started. Waiting for clients...
New connection: ('127.0.0.1', 51621)
Message from ('127.0.0.1', 51621): hello
Message from ('127.0.0.1', 51621): hiii
Message from ('127.0.0.1', 51621): 123
Message from ('127.0.0.1', 51621): 345
Message from ('127.0.0.1', 51621): hello manali
```

#client output

```
In [1]: runfile('C:/Users/STUDENT.LAB03-03/.spyder-py3/client-gc.py', wdir='C:/Users/
STUDENT.LAB03-03/.spyder-py3')
hello
hiii
123
345
hello manali
```