Experiment 6

Aim: - To create a chat server between client and server in TCP/IP protocol.

Steps for server: -

1. We import socket module which helps in establishing the client/server communication.
2. Bind the server and client using bind function.
3. Use broadcast function to broadcast the message you get from some client.
4. Use handle function to handle all the incoming and outgoing of messages in the chat server.
5. Receive function is used to receive all the messages you get from client or multiple clients and using threading function we can implement multiple functions or multiple chat clients.
6. Close the server.

Steps for Client: -

1. We import socket module which helps in establishing the client/server communication.
2. Make the receive function to receive messages from other clients and the server.
3. Create a write function so that the clients can write the chats which ever they want. Threading is used to have multiple clients.
4. Use send functions and write functions to send and receive the messages from other clients.
5. Close the clients.

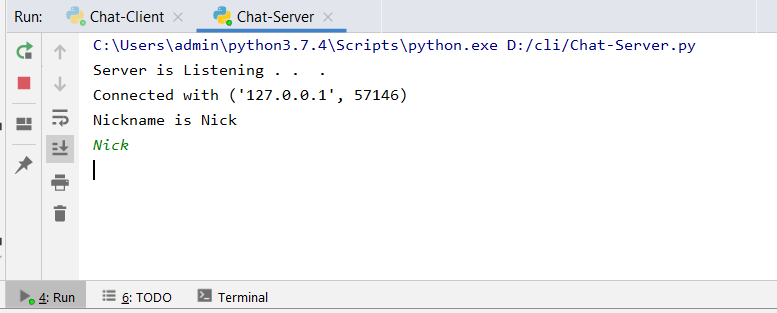
Code for Server :-

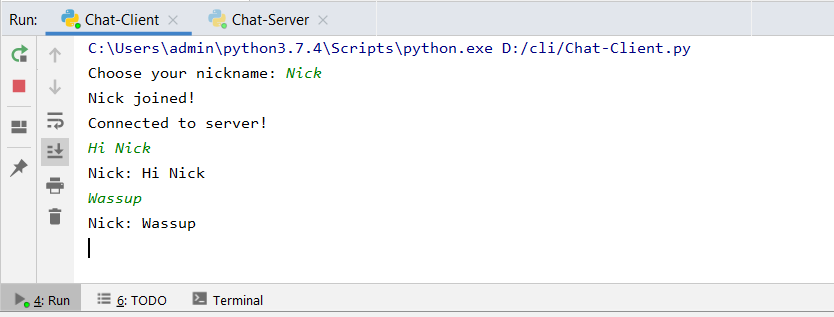
**import** socket  
**import** threading  
host = **'127.0.0.1'**port = 55555  
  
*# Starting Server*server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
server.bind((host, port))  
server.listen()  
  
*# Lists For Clients and Their Nicknames*clients = []  
nicknames = []  
  
**def** broadcast(message):  
 **for** client **in** clients:  
 client.send(message)  
  
**def** handle(client):  
 **while True**:  
 **try**:  
 *# Broadcasting Messages* message = client.recv(1024)  
 broadcast(message)  
 **except**:  
 *# Removing And Closing Clients* index = clients.index(client)  
 clients.remove(client)  
 client.close()  
 nickname = nicknames[index]  
 broadcast(**'{} left!'**.format(nickname).encode(**'ascii'**))  
 nicknames.remove(nickname)  
 **break  
  
def** receive():  
 **while True**:  
 *# Accept Connection* client, address = server.accept()  
 print(**"Connected with {}"**.format(str(address)))  
  
 *# Request And Store Nickname* client.send(**'NICK'**.encode(**'ascii'**))  
 nickname = client.recv(1024).decode(**'ascii'**)  
 nicknames.append(nickname)  
 clients.append(client)  
  
 *# Print And Broadcast Nickname* print(**"Nickname is {}"**.format(nickname))  
 broadcast(**"{} joined!"**.format(nickname).encode(**'ascii'**))  
 client.send(**'Connected to server!'**.encode(**'ascii'**))  
  
 *# Start Handling Thread For Client* thread = threading.Thread(target=handle, args=(client,))  
 thread.start()  
  
print(**"Server is Listening . . ."**)  
receive()

Code for Client :-

**import** socket  
**import** threading  
  
nickname = input(**"Choose your nickname: "**)  
  
*# Connecting To Server*client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
client.connect((**'127.0.0.1'**, 55555))  
  
**def** receive():  
 **while True**:  
 **try**:  
 *# Receive Message From Server  
 # If 'NICK' Send Nickname* message = client.recv(1024).decode(**'ascii'**)  
 **if** message == **'NICK'**:  
 client.send(nickname.encode(**'ascii'**))  
 **else**:  
 print(message)  
 **except**:  
 *# Close Connection When Error* print(**"An error occured!"**)  
 client.close()  
 **break  
def** write():  
 **while True**:  
 message = **'{}: {}'**.format(nickname, input(**''**))  
 client.send(message.encode(**'ascii'**))  
  
receive\_thread = threading.Thread(target=receive)  
receive\_thread.start()  
  
write\_thread = threading.Thread(target=write)  
write\_thread.start()

Console Screenshots : -

1. Server Console : -

1. Client Console : -

Result : - The chat program between client and server was successfully made.