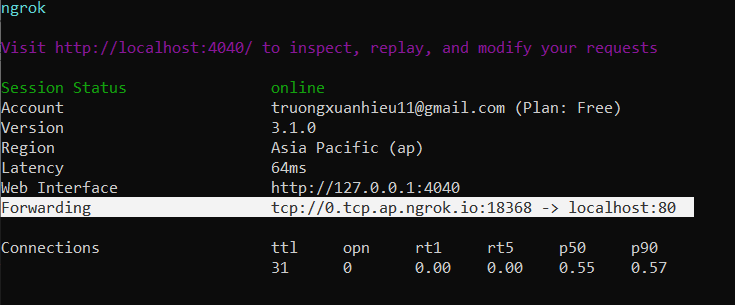
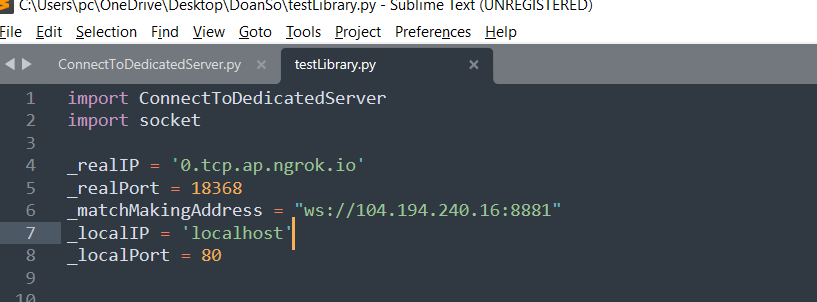
How to connect to Matchmaking server and Clients

1. **Run ngrok: Open cmd and type *ngrok tcp 80***

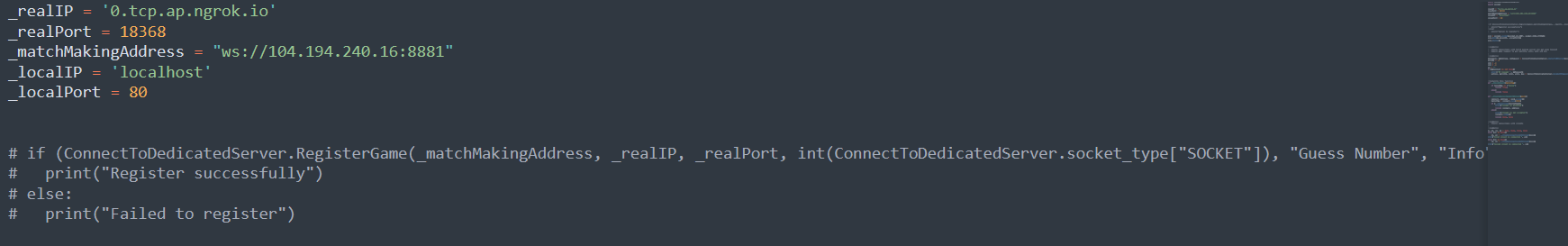


1. **Copy Forwarding line (the highlight line above) and paste its information into file testLibrary.py like this**



\_matchMakingAddress is static, so it doesn’t need to change

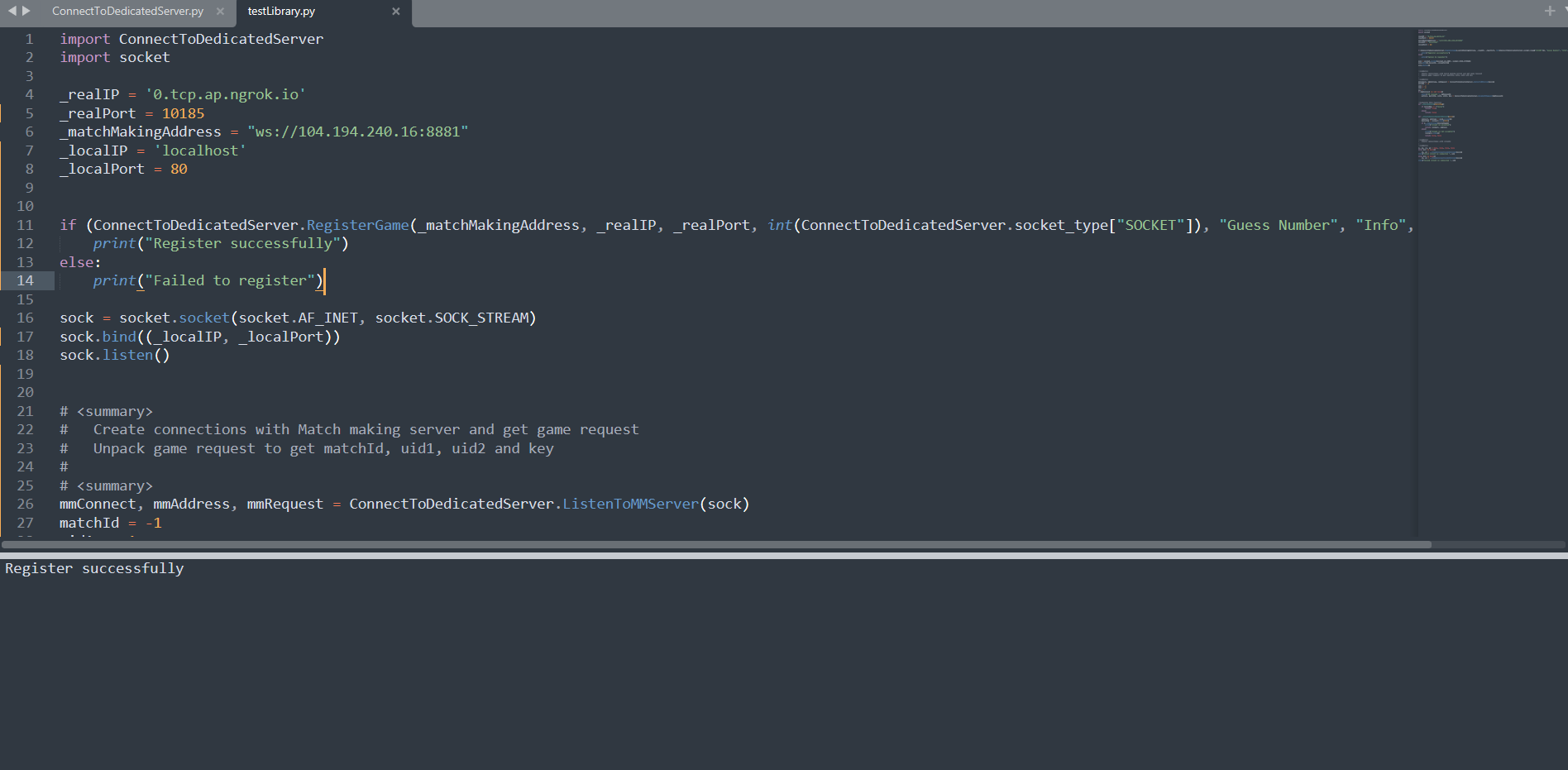
1. **Right behind these variables, there are 4 lines which are marked as comment**

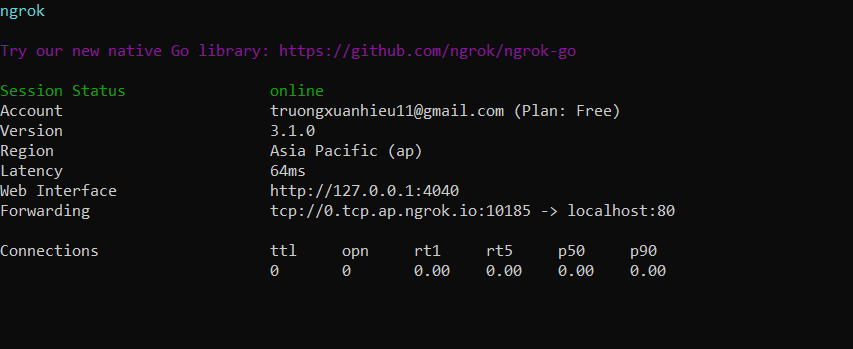


These lines are used to register this game to the Matchmaking server, so in the first run after you start or restart ngrok, you should enable them before running. If not, Matchmaking server can’t detect your new ngrok address.

After that, in the next runs, disable these 4 lines. If not, they will continue registering your game to Matchmaking server and create a lot of duplicate game instances

1. **Run testLibrary.py. The first log will be like this**

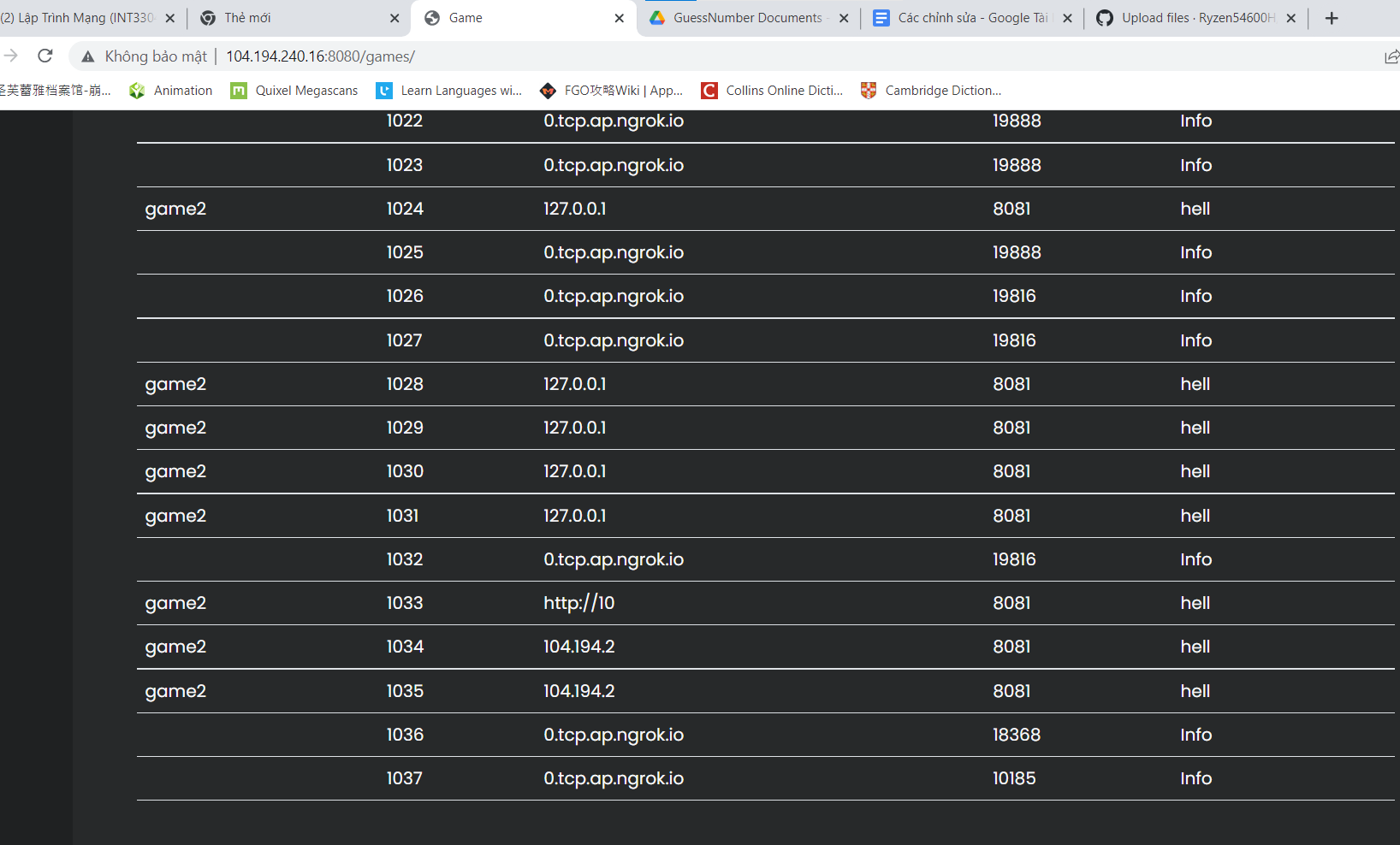




It says that Matchmaking server has recorded your game in port 10185

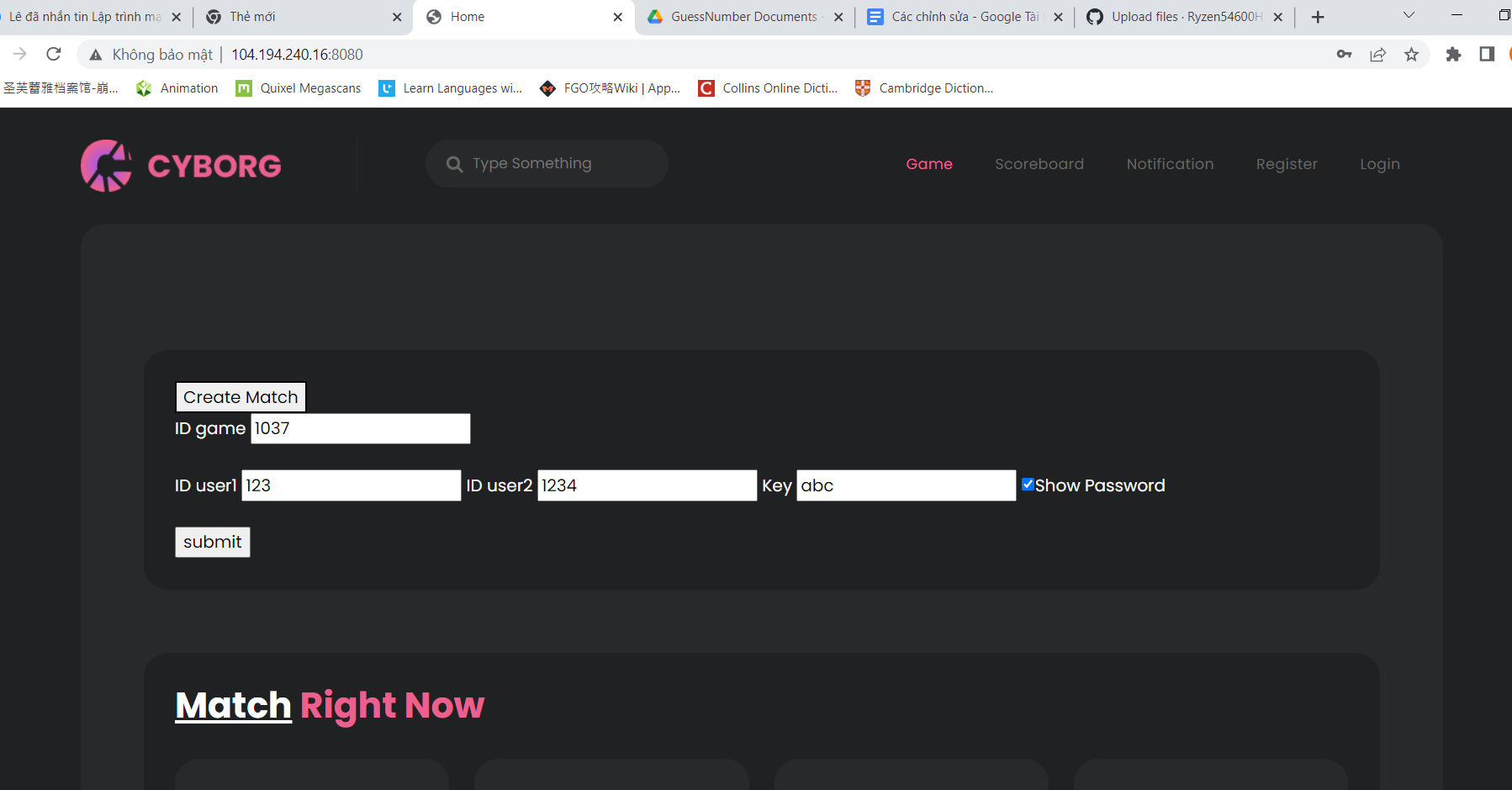
If log says that Failed to register, check your ip and port variables with ngrok ip and port. May be they are different. Change them and register your game again

1. **Then go to Matchmaking website, in tab Game, scroll down and you can see there is a game listens in port 10185**



That is the game instance we have just created

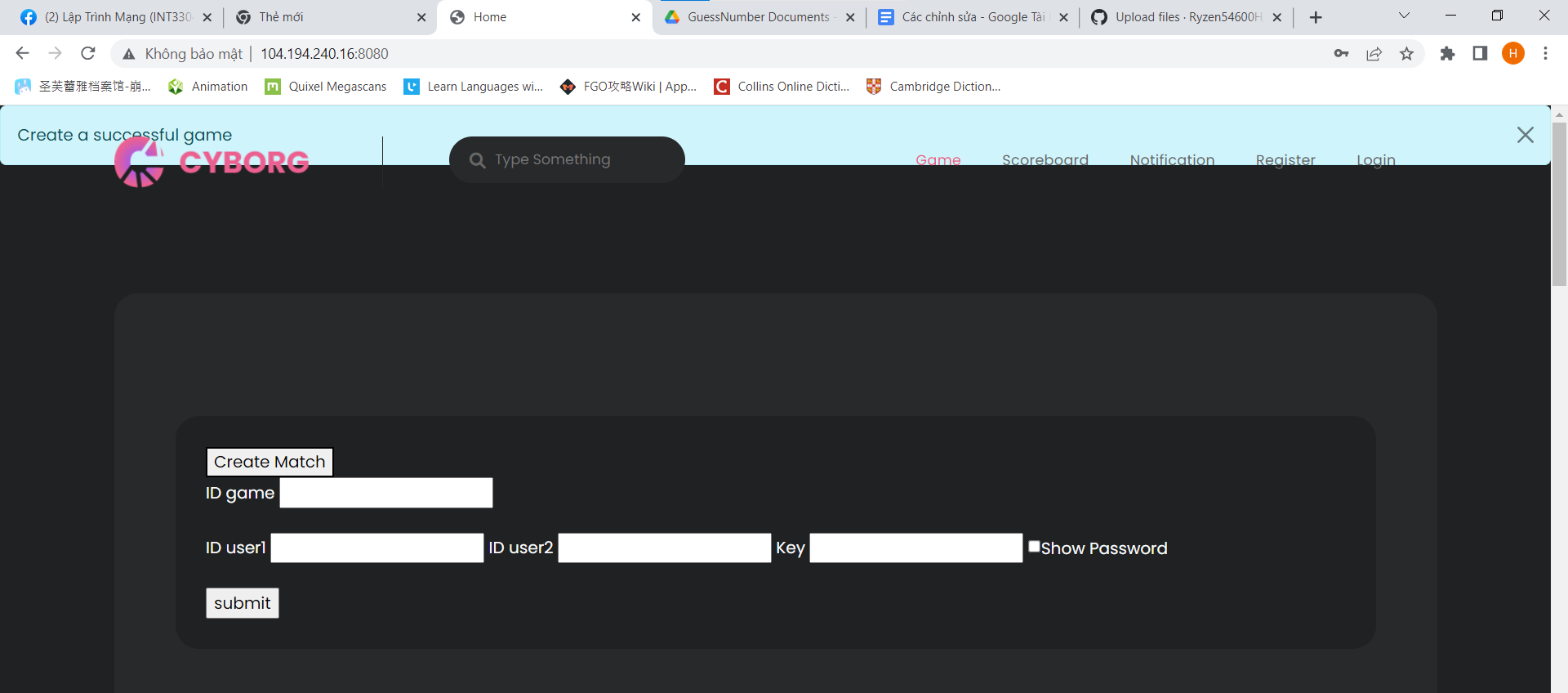
1. **Back to Create Match page, enter match information to create a game**



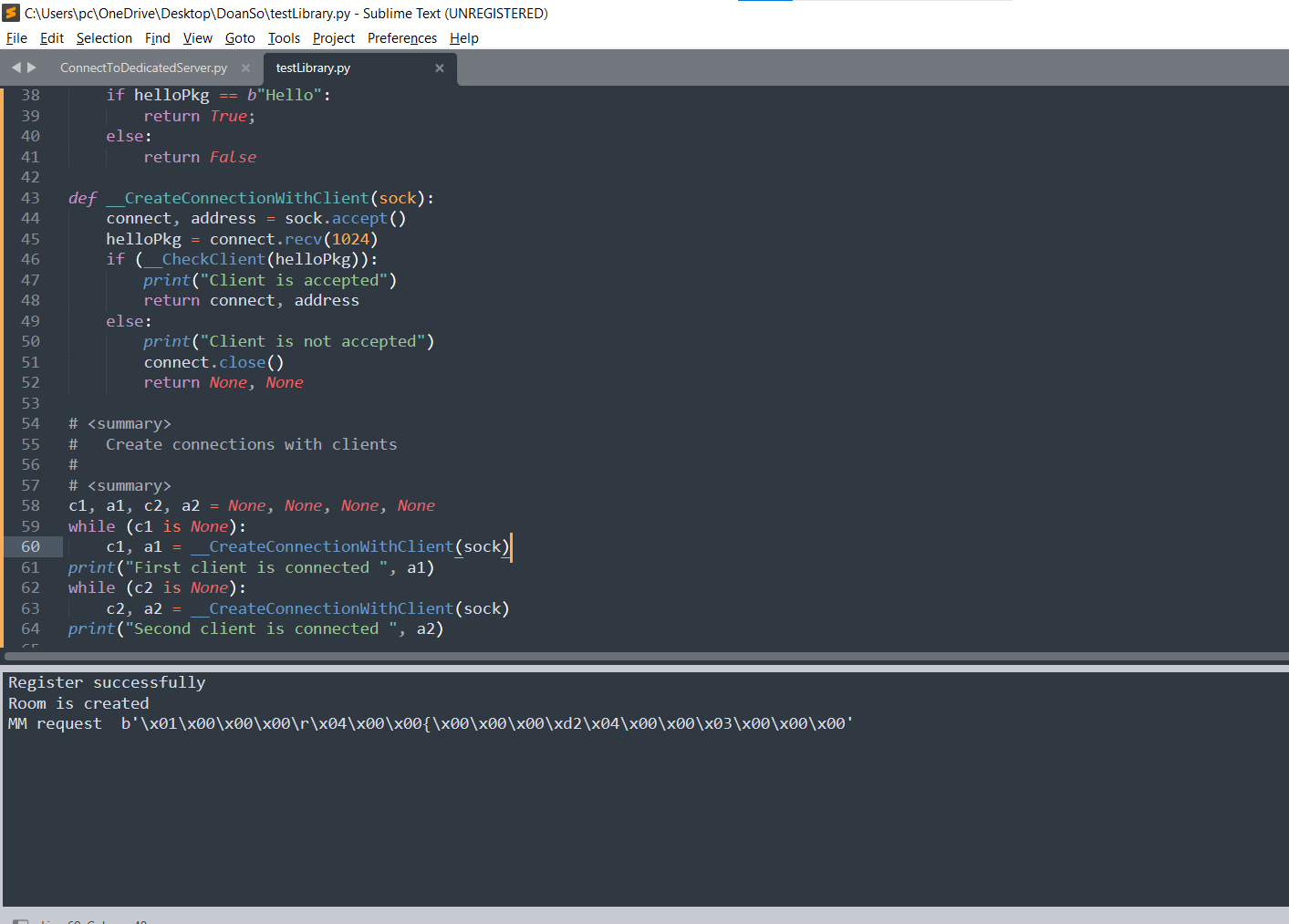
ID game is in the second column in tab game. As we showed in the image above, ID game is 1037. Copy it and paste to ID game input field

ID user1, ID user2 and Key now can be randomed

1. **Press submit. The website will show this notification to tell you that your game is requested**

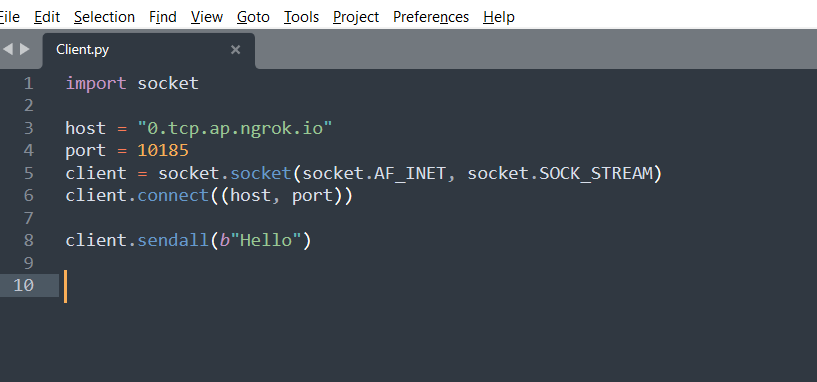


And back to your server, there are 2 new lines in log area



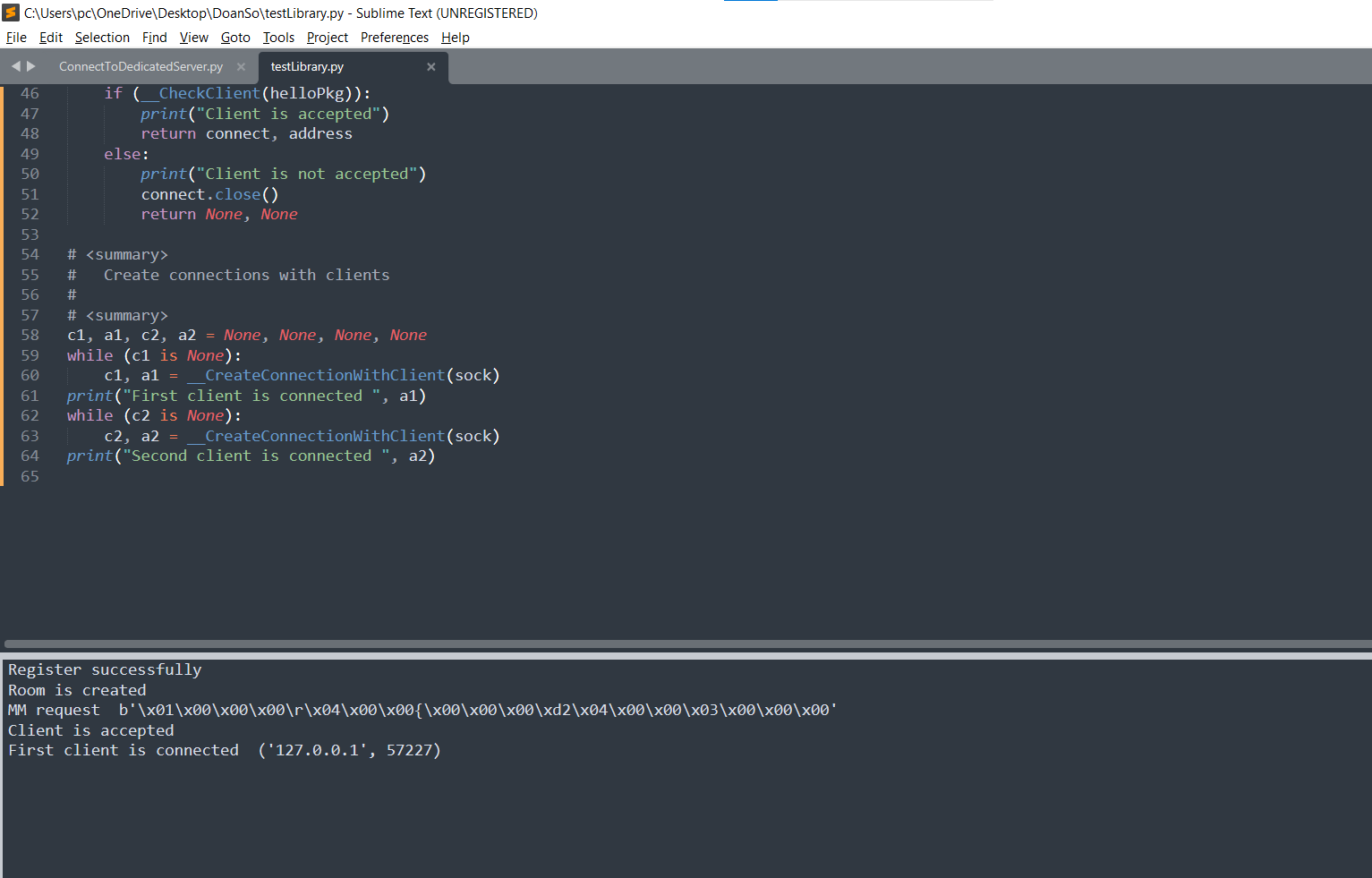
When you reach this point, creating connection with Matchmaking server is finished

1. **Open file Client.py and run it as first Client**



Before running, you have to change host and port to fit with your ngrok Forwarding

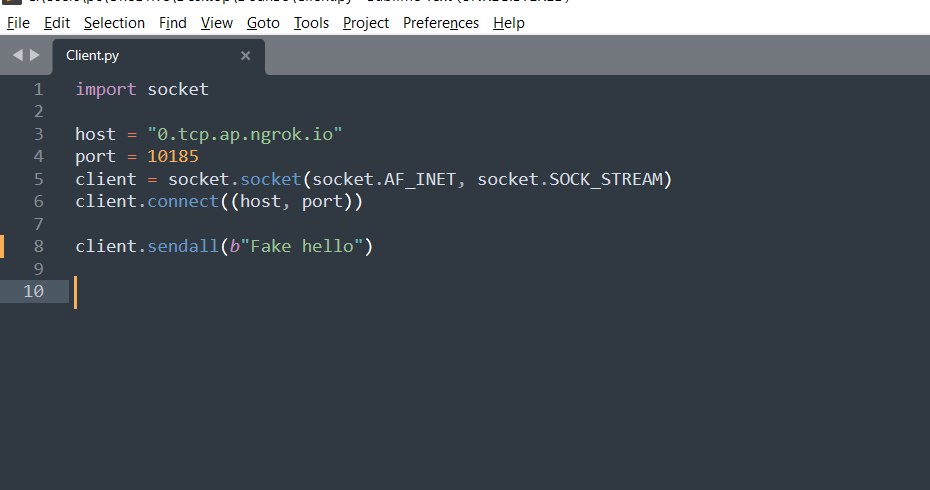
After running Client.py, you should see new 2 log lines in testLibrary.py like this



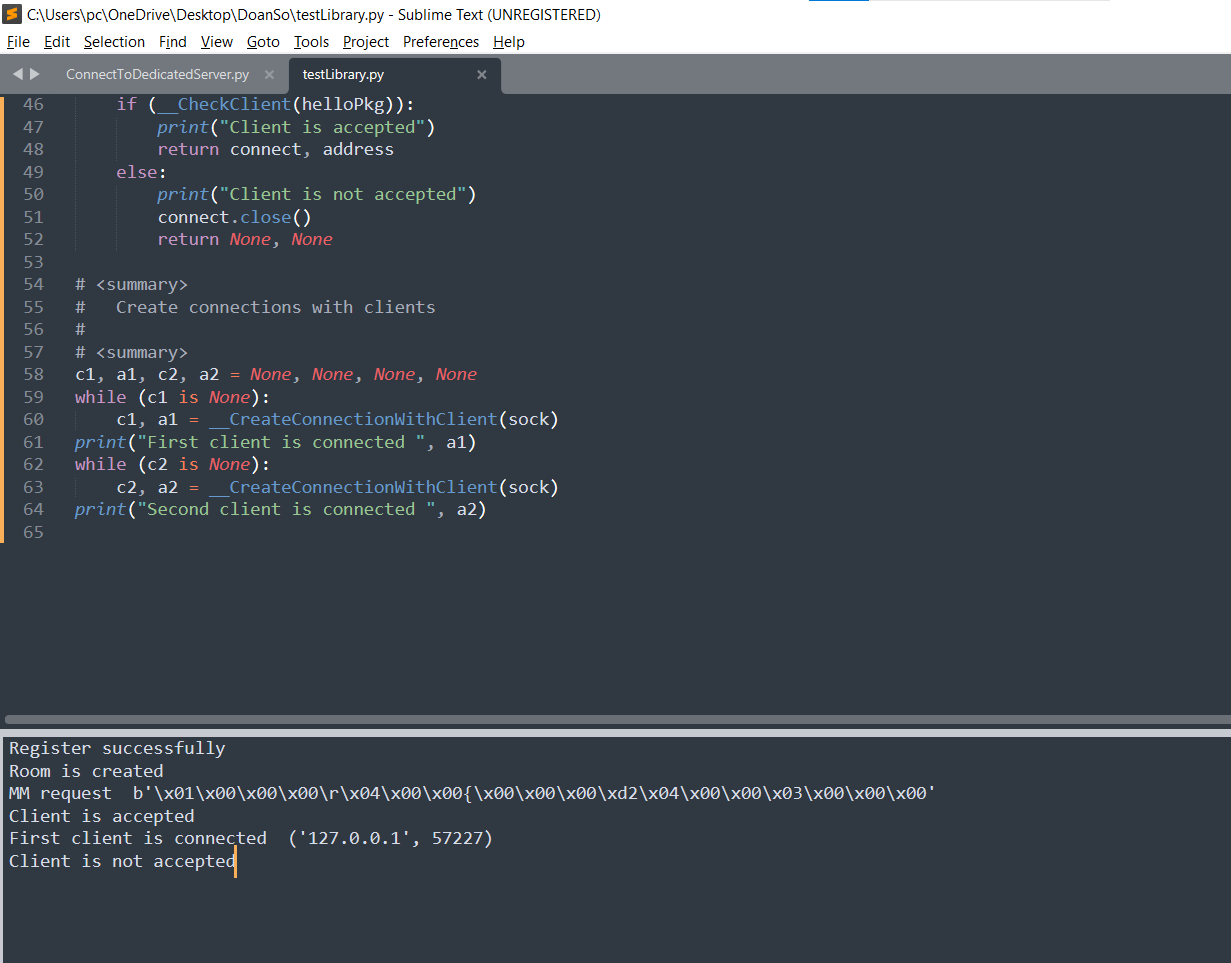
It shows that your Client’s connection is accepted

1. **Run the second Client**

Open Client.py in new tab to run it as second Client. But before running, change the messenger from “Hello” to “Fake hello” like this

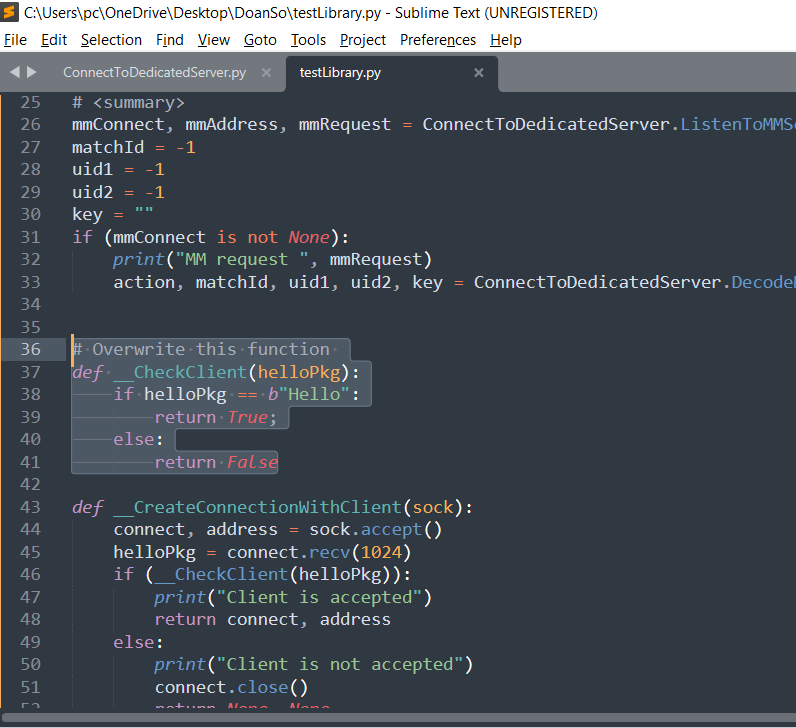


Then you will see testLibrary.py refuses this connection



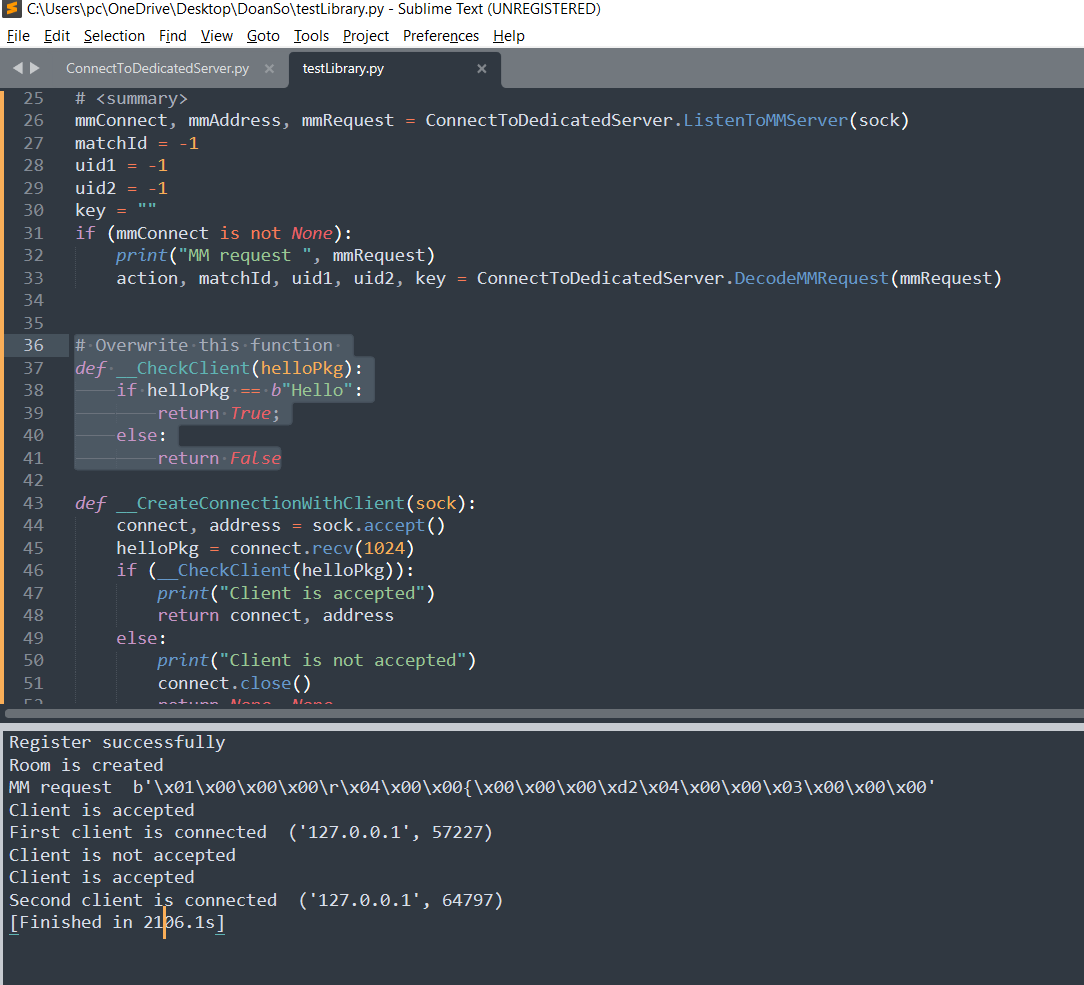
It is because testLibrary.py use function called \_\_CheckClient to check the messengers received from Clients. For now, Client have to send “Hello” in order to be accepted.

In future, you should overwrite this function so it can check the key clients received from Matchmaking server



1. **Get Client.py back as it was then run it as second Client**

Change the messenger in Client.py back to “Hello” and run it. You will see new 3 lines in testLibrary.py say that it accepted the second Client and then end the program



Results

1. Register game to Matchmaking server
2. Connect to Matchmaking server and hold the connection with it by *mmConnect* and *mmAddress*
3. Create connections with 2 clients and hold the connections with them by *c1*, *a1*, *c2* and *a2*. Besides, it guarantee that all the clients’ connections have to meet some conditions (will be defined later) to be accepted