Project 4 Write-up

Introduction:

This project implements a client/server chat program that allows two people to converse over a network, via sockets. The client will initiate a conversation with a server by connecting to the host address and port number of the server. After the connection is established, the client will start the conversation by typing some message to the client. After the client's message is received, the server will respond back with its own message. The communication will flow back-and-forth in this manner until either the client or server enters in '/q', which is the kill signal for the program.

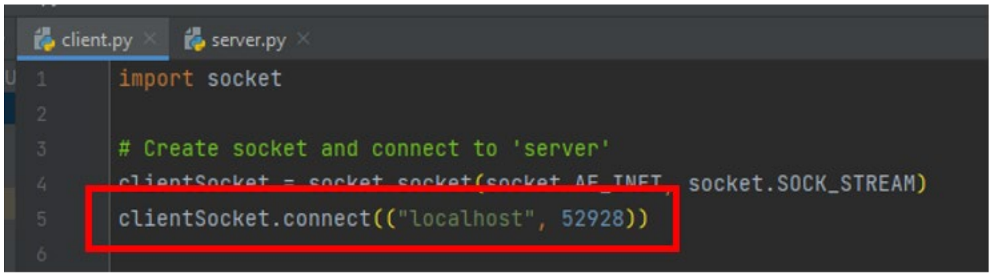
Instructions:

Platform – Python V-3.9.6

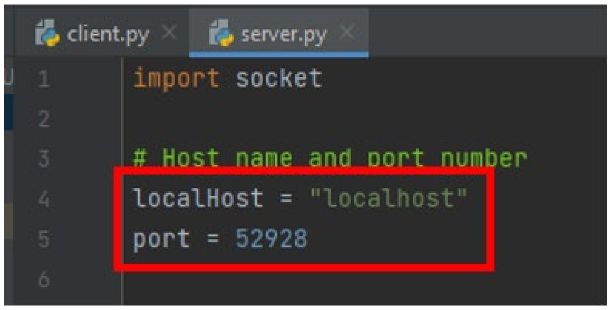
File Name –client.py, server.py

Operating Instructions –

1. Update hostname and port numbers as necessary. The hostname and port numbers need to match in both programs.



```
import socket

# Create socket and connect to 'server'
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientSocket.connect(("localhost", 52928))
```



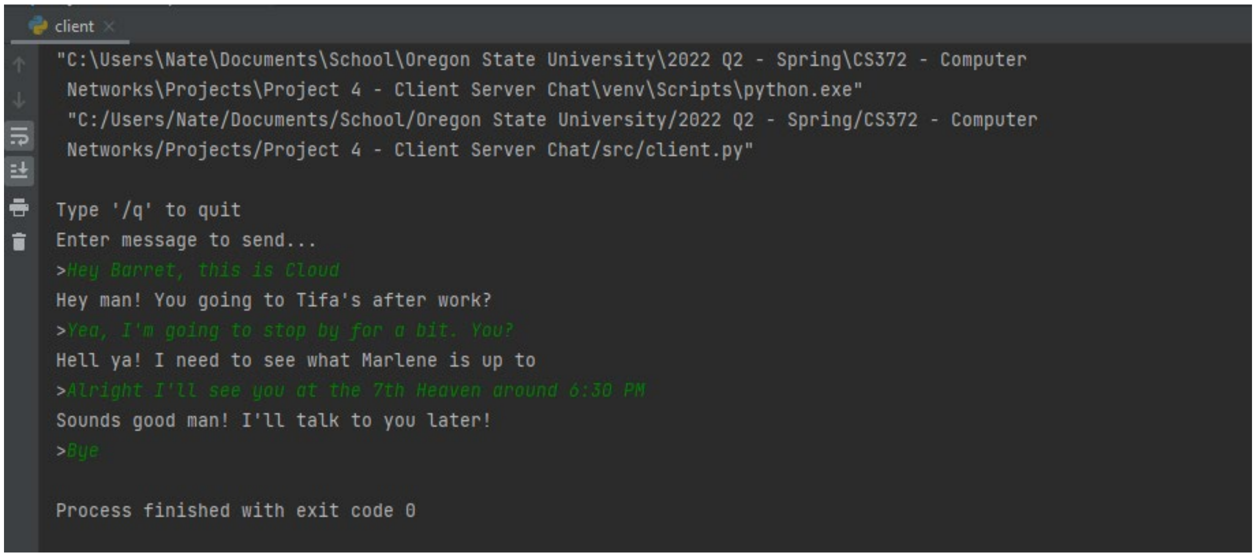
```
import socket

# Host name and port number
localHost = "localhost"
port = 52928
```

2. Start up the server application.
3. Start up the client application.

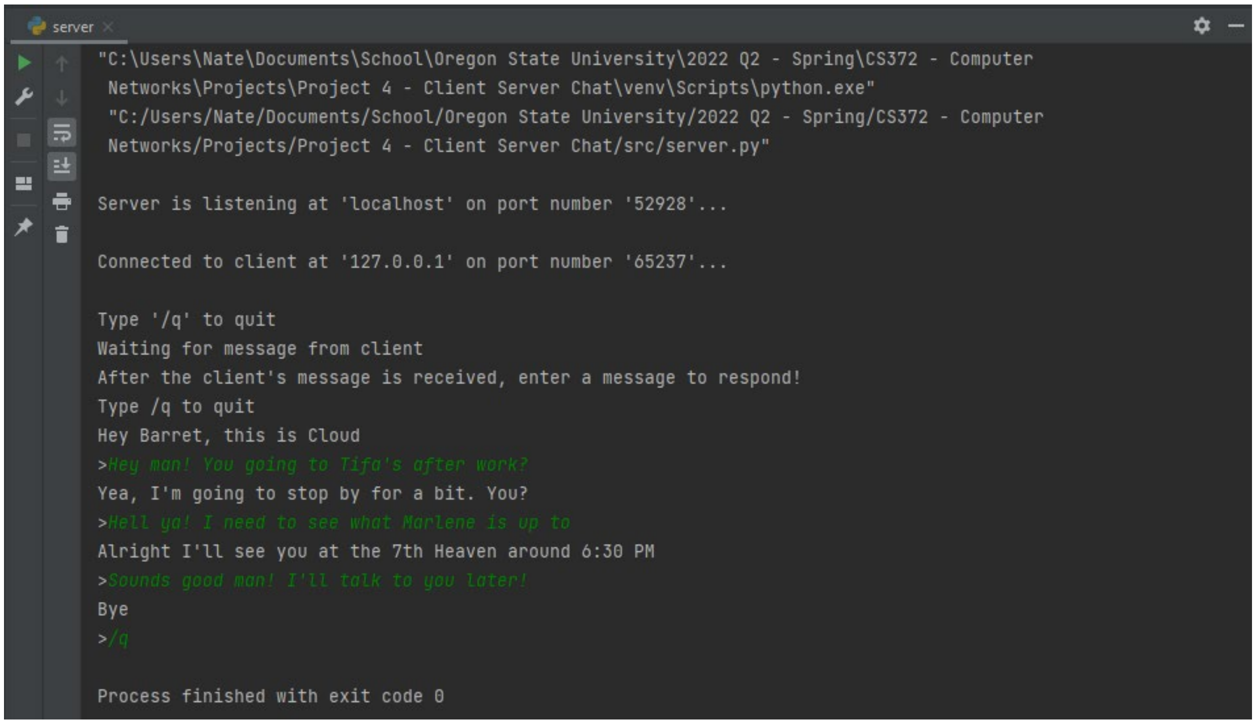
**Note: It is important to have the server application running before the client application is started up, since the client application automatically connects with the server. If the client application is started up first, it will crash.*

Screenshots:

A short conversation between Cloud and Barret from Final Fantasy 7!



```
"C:\Users\Nate\Documents\School\Oregon State University\2022 Q2 - Spring\CS372 - Computer
 Networks\Projects\Project 4 - Client Server Chat\venv\Scripts\python.exe"
 "C:/Users/Nate/Documents/School/Oregon State University/2022 Q2 - Spring/CS372 - Computer
 Networks/Projects/Project 4 - Client Server Chat/src/client.py"

Type '/q' to quit
Enter message to send...
>Hey Barret, this is Cloud
Hey man! You going to Tifa's after work?
>Yea, I'm going to stop by for a bit. You?
Hell ya! I need to see what Marlene is up to
>Alright I'll see you at the 7th Heaven around 6:30 PM
Sounds good man! I'll talk to you later!
>Bye

Process finished with exit code 0
```



```
"C:\Users\Nate\Documents\School\Oregon State University\2022 Q2 - Spring\CS372 - Computer
 Networks\Projects\Project 4 - Client Server Chat\venv\Scripts\python.exe"
 "C:/Users/Nate/Documents/School/Oregon State University/2022 Q2 - Spring/CS372 - Computer
 Networks/Projects/Project 4 - Client Server Chat/src/server.py"

Server is listening at 'localhost' on port number '52928'...

Connected to client at '127.0.0.1' on port number '65237'...

Type '/q' to quit
Waiting for message from client
After the client's message is received, enter a message to respond!
Type /q to quit
Hey Barret, this is Cloud
>Hey man! You going to Tifa's after work?
Yea, I'm going to stop by for a bit. You?
>Hell ya! I need to see what Marlene is up to
Alright I'll see you at the 7th Heaven around 6:30 PM
>Sounds good man! I'll talk to you later!
Bye
>/q

Process finished with exit code 0
```

Comments/Questions:

I got the program to work based on the specification, but I want to take the application a bit further. I couldn't quite figure out how to make it work, but I was trying to get the program to act as two completely independent programs, in which you wouldn't have to wait on the other person's reply before sending your next message. Instead of a back-and-forth conversation, you could just type as much as you want to the other person and get their responses back as they type them. More like a traditional text messaging or instant messaging application. The more I think about it I think this would mean setting up two sockets for each communication. The client would have one socket dedicated to sending messages and another socket dedicated to receiving messages. The server would have a similar set-up. I think something similar to this is how you could enable to the communication true "two-way" communication.

Citations:

Laboratories, M. (n.d.). Making HTTP requests with sockets in Python. Internal Pointers. Retrieved April 15, 2022, from https://www.internalpointers.com/post/making-http-requests-sockets-python

Bodnar, J. (n.d.). Python socket. Python Socket - Python network programming with sockets. Retrieved April 15, 2022, from https://zetcode.com/python/socket/

Real Python. (2022, February 21). Socket programming in python (guide). Real Python. Retrieved April 15, 2022, from https://realpython.com/python-sockets/