Timothy Butler
March 19, 2023
CS 372 – 400
Intro to Networks
Project #4

Client/Server Chat

Introduction:

This is a simple client and server chat program. A server will listen on localhost on a port number that is provided already in the code. The client side will then try to connect and if successful will begin to chat with the server. Follow the instructions below and start chatting (with yourself of course).

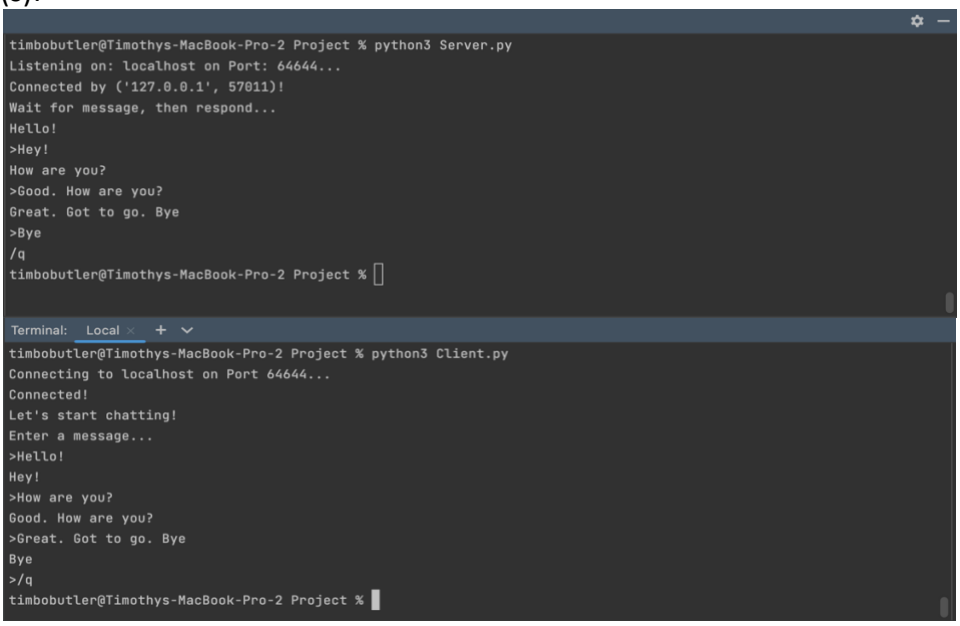
To run the program:

1. Start your local terminal and find the folder where files Client.py and Server.py
2. Run the statement in your terminal:
   python3 Server.py
3. Open another terminal window (set terminals side-by-side)
4. Run the statement in your new terminal:
   python3 Client.py
5. Begin the chat on the Client-side and respond on the Server-side. Go back and forth from the two terminals to simulate a live chat room!

Comments:

Be sure to run the Server.py program first!

Screenshot(s):



Client.py | Server.py

```python
# Author: Timothy Butler
# Date: 03/19/2023
# Course: CS372 - 400
# Sources:
# https://realpython.com/python-sockets/#tcp-sockets

import socket
import time

# Connect to the server
HOST = "localhost"
PORT = 64644
print(f"Connecting to {HOST} on Port {PORT}...")
time.sleep(0.5)
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    print("Connected!\nLet's start chatting!")
    time.sleep(0.5)
    print("Enter a message...")
    while True:
        message = input('>')
        s.send(message.encode())
        # if /q, then quit
        if message == '/q':
            break
        data = s.recv(4096).decode()
        # if /q, then quit
        if data == '/q':
            break
        print(data)

```

Client.py | Server.py

```python
# Author: Timothy Butler
# Date: 03/19/2023
# Course: CS372 - 400
# Sources:
# https://realpython.com/python-sockets/#tcp-sockets

import socket

HOST = "localhost"
PORT = 64644
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    print(f"Listening on: {HOST} on Port: {PORT}...")
    conn, addr = s.accept()
    print(f"Connected by {addr}!\nWait for message, then respond...")
    while True:
        data = conn.recv(4096).decode()
        print(data)
        # if /q, then quit
        if data == '/q':
            break
        message = input('>')
        conn.send(message.encode())
        # if /q, then quit
        if message == '/q':
            break
    # Close the connection
    conn.close()
```