COMPUTER COMMUNICATION NETWORKS

PROJECT REPORT

PROJECT TITLE: **TCP CLIENT-SERVER APPLICATION**

GROUP MEMBERS:

ZAINAB RAZA (CS – 22054)

MOUNAZZA ANWAR (CS – 22063)

ASIYA FATIMA (CS – 22064)

AFFAF ARIF (CS – 22069)

BATCH: 2022

SUBMISSION DATE: NOV 25, 2024

SUBMITTED TO: MISS HAMNAH UMAR

INTRODUCTION

This project implements a multi-client TCP chatroom application in Python, designed to facilitate real-time communication. The application integrates administrator privileges, enabling effective management of the chatroom. The architecture ensures seamless interaction between clients and robust moderation features.

FEATURES

Real-Time Messaging:

- Multiple clients can send and receive messages simultaneously.
- Messages are broadcasted to all connected users in the chatroom.

Admin Controls:

The user with the nickname "admin" has exclusive privileges, including:

- **Kicking Users:** Removes users from the chatroom using the /kick <nickname> command.
- **Banning Users:** Permanently prevents users from rejoining with /ban <nickname>.

Persistent Bans

- Banned nicknames are recorded in a file named bans.txt.
- These bans persist across server restarts, ensuring long-term enforcement.

USAGE INSTRUCTION

Starting the Server

- Run server.py to initialize the chat server.
- Default IP: 127.0.0.1
- Default Port: 55556

Connecting Clients

- Run client.py for each client instance.
- Provide a unique nickname when prompted.
- **Administrator Login:** Use the nickname admin and the default password “adminpass” to access admin privileges.

Administrator Commands

- **/kick <nickname>:** Disconnects the specified user from the chatroom.
- **/ban <nickname>:** Bans the specified user and records their nickname in bans.txt.

FUNCTIONALITY OVERVIEW

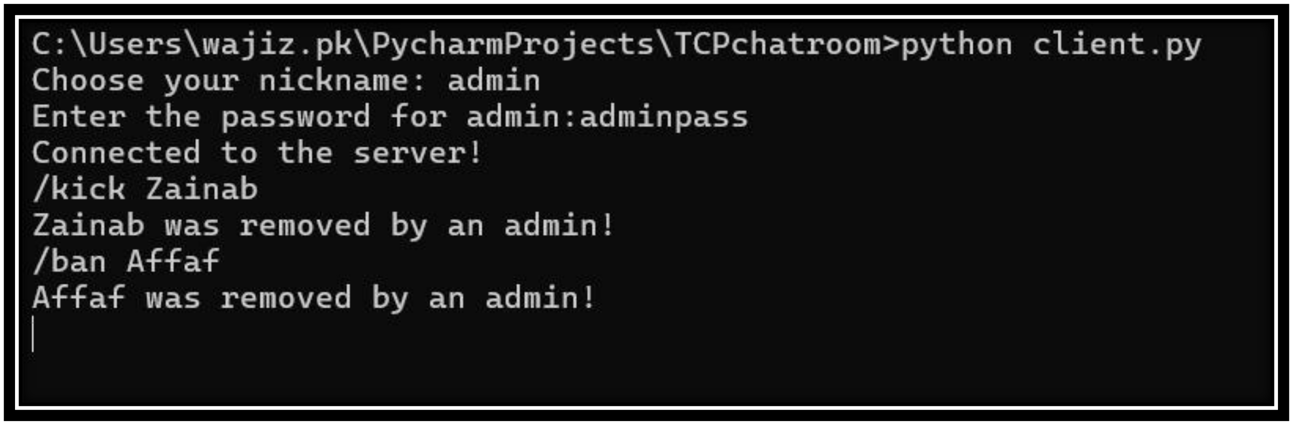
Server Behavior

- Listens for incoming client connections.
- Manages message broadcasting and moderation commands.
- Maintains a list of banned nicknames to block unauthorized access.

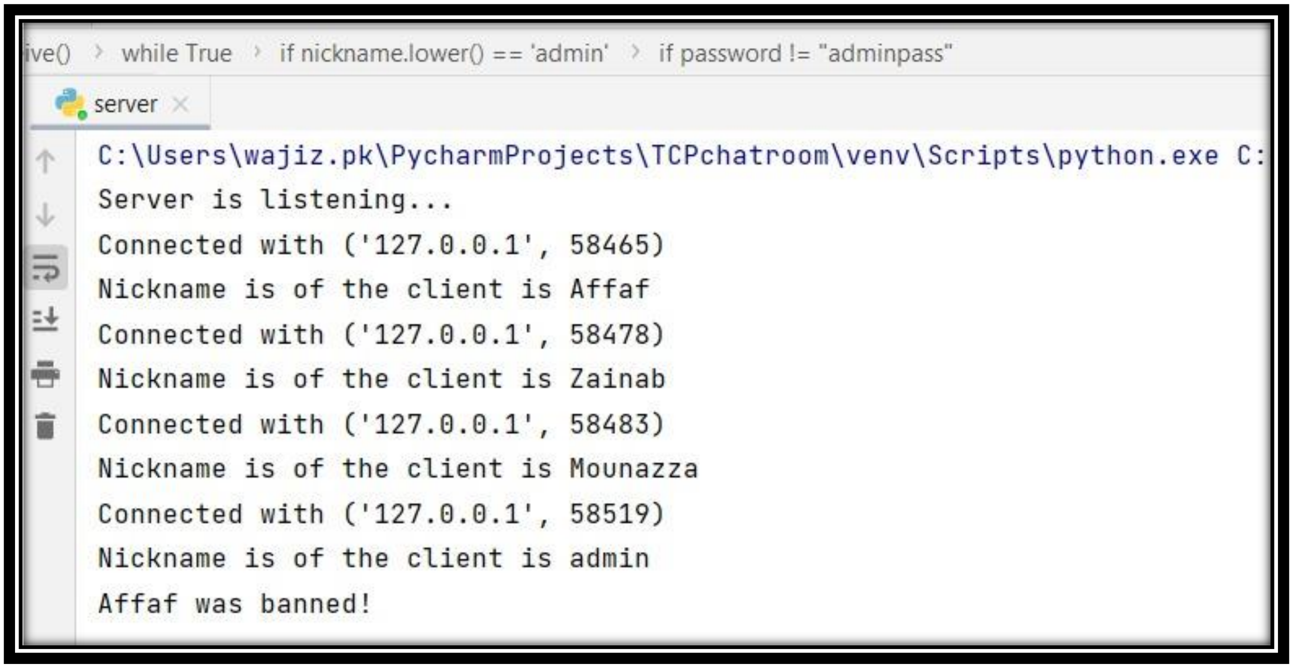
Client Behavior

- Connects to the server using client.py.
- Interacts with other users via real-time text exchange.
- Receives notifications for admin actions (e.g., kicks or bans).

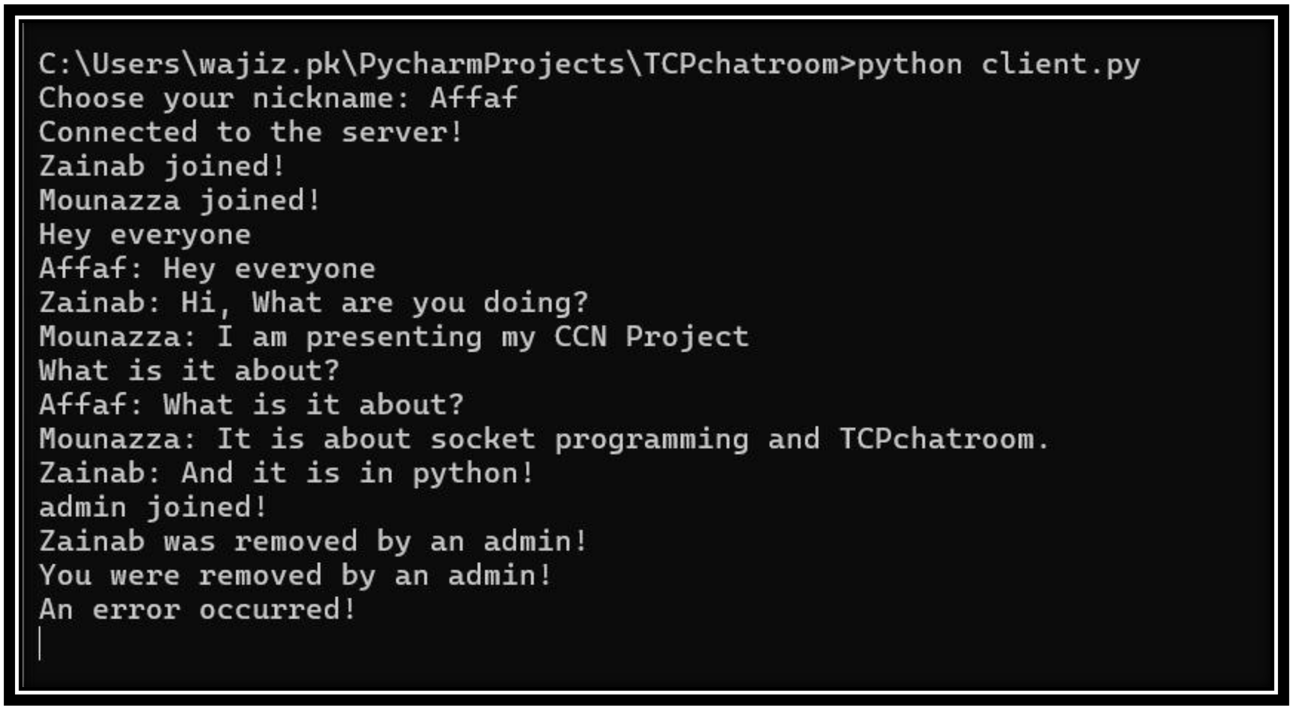
EVIDENCE OF FUNCTIONALITY



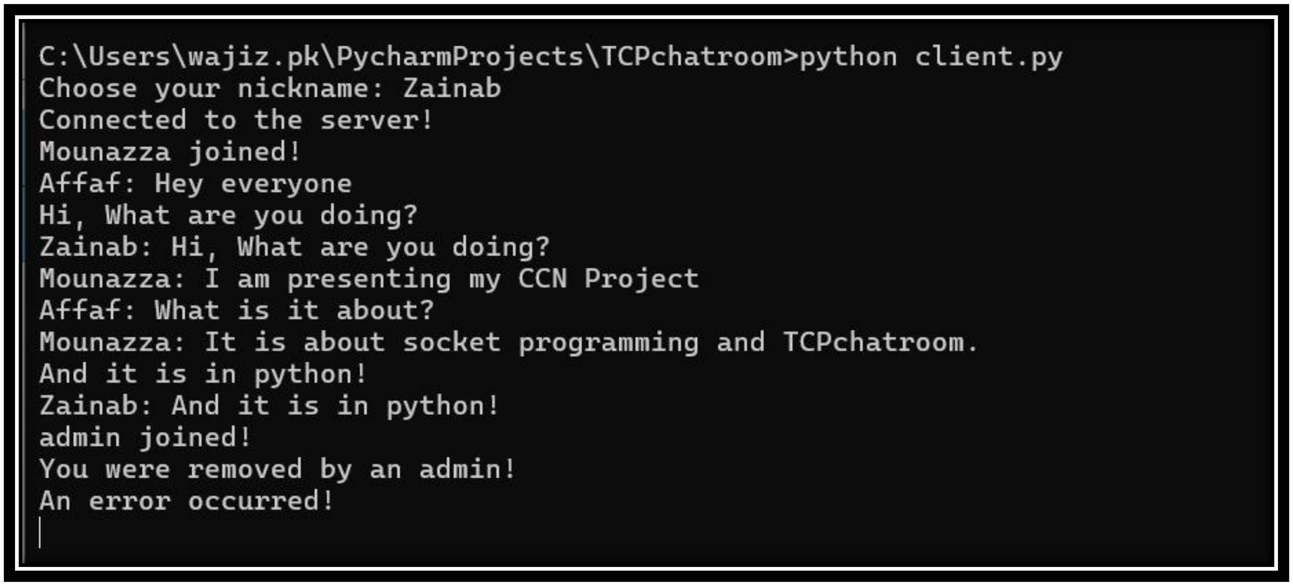
```
C:\Users\wajiz.pk\PycharmProjects\TCPchatroom>python client.py
Choose your nickname: admin
Enter the password for admin:adminpass
Connected to the server!
/kick Zainab
Zainab was removed by an admin!
/ban Affaf
Affaf was removed by an admin!
```



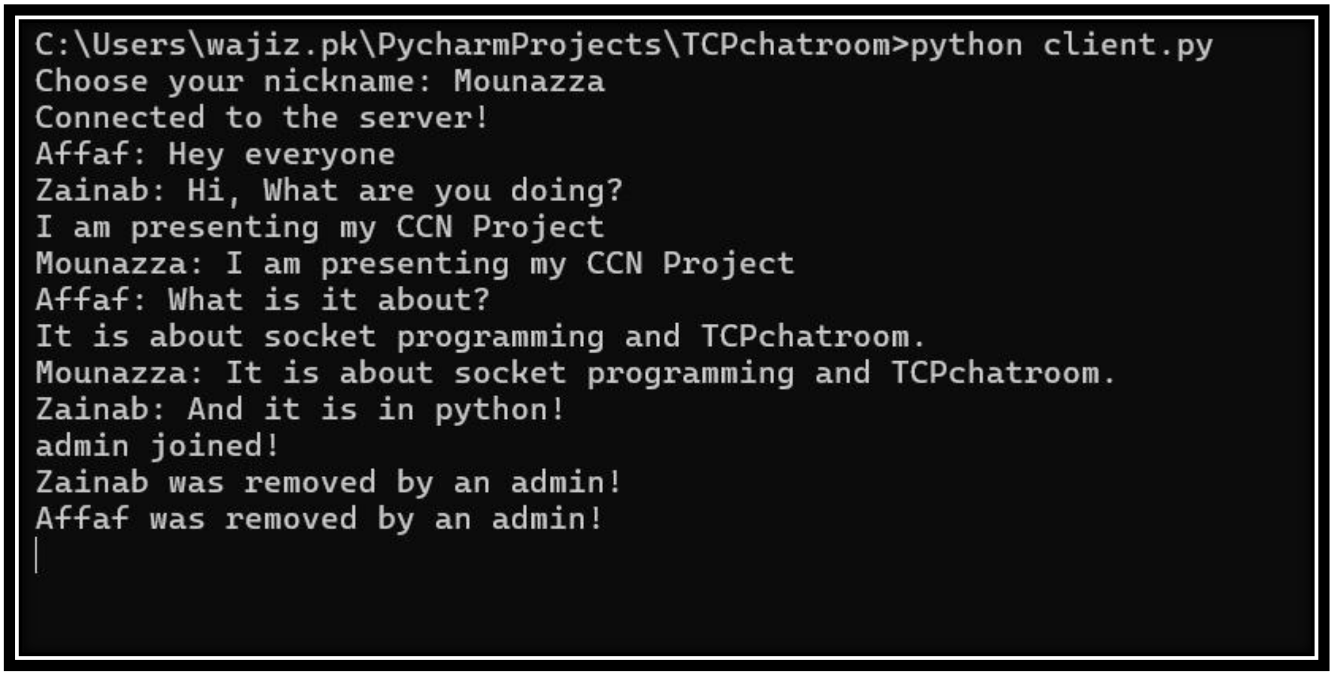
```
C:\Users\wajiz.pk\PycharmProjects\TCPchatroom\venv\Scripts\python.exe C:
Server is listening...
Connected with ('127.0.0.1', 58465)
Nickname is of the client is Affaf
Connected with ('127.0.0.1', 58478)
Nickname is of the client is Zainab
Connected with ('127.0.0.1', 58483)
Nickname is of the client is Mounazza
Connected with ('127.0.0.1', 58519)
Nickname is of the client is admin
Affaf was banned!
```



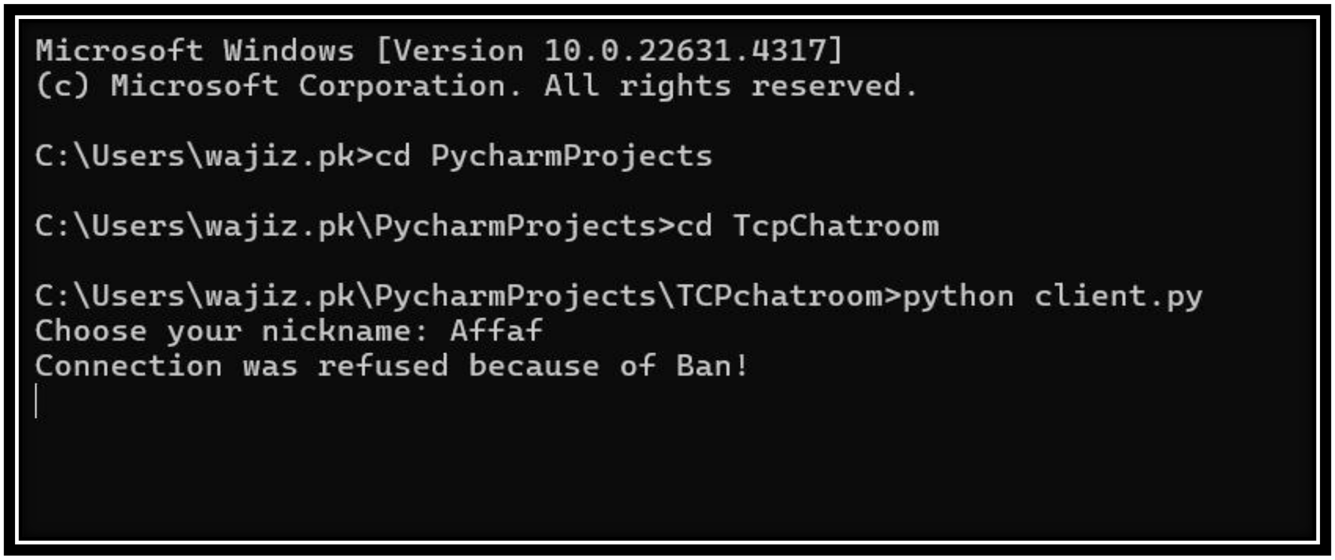
```
C:\Users\wajiz.pk\PycharmProjects\TCPchatroom>python client.py
Choose your nickname: Affaf
Connected to the server!
Zainab joined!
Mounazza joined!
Hey everyone
Affaf: Hey everyone
Zainab: Hi, What are you doing?
Mounazza: I am presenting my CCN Project
What is it about?
Affaf: What is it about?
Mounazza: It is about socket programming and TCPchatroom.
Zainab: And it is in python!
admin joined!
Zainab was removed by an admin!
You were removed by an admin!
An error occurred!
```



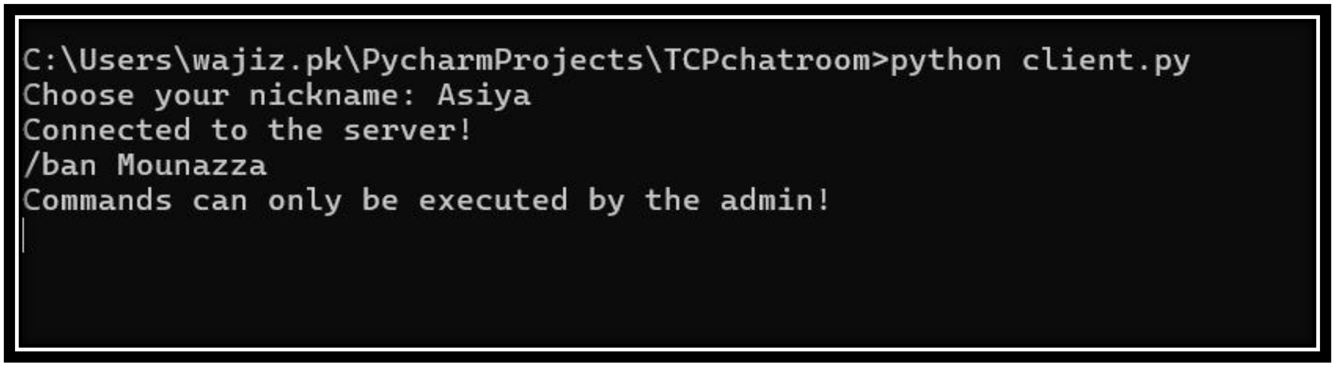
```
C:\Users\wajiz.pk\PycharmProjects\TCPchatroom>python client.py
Choose your nickname: Zainab
Connected to the server!
Mounazza joined!
Affaf: Hey everyone
Hi, What are you doing?
Zainab: Hi, What are you doing?
Mounazza: I am presenting my CCN Project
Affaf: What is it about?
Mounazza: It is about socket programming and TCPchatroom.
And it is in python!
Zainab: And it is in python!
admin joined!
You were removed by an admin!
An error occurred!
```



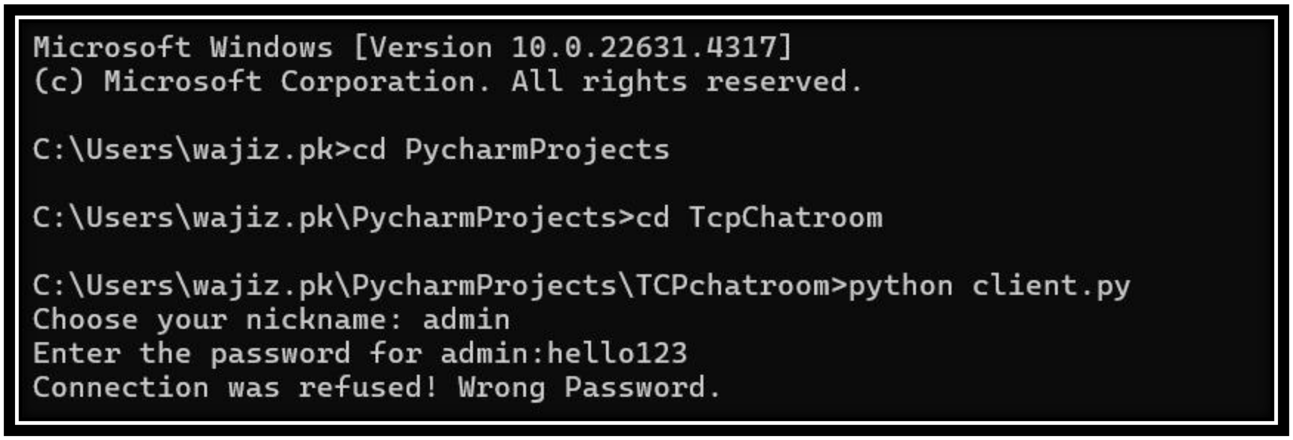
```
C:\Users\wajiz.pk\PycharmProjects\TCPchatroom>python client.py
Choose your nickname: Mounazza
Connected to the server!
Affaf: Hey everyone
Zainab: Hi, What are you doing?
I am presenting my CCN Project
Mounazza: I am presenting my CCN Project
Affaf: What is it about?
It is about socket programming and TCPchatroom.
Mounazza: It is about socket programming and TCPchatroom.
Zainab: And it is in python!
admin joined!
Zainab was removed by an admin!
Affaf was removed by an admin!
```



```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\wajiz.pk>cd PycharmProjects

C:\Users\wajiz.pk\PycharmProjects>cd TcpChatroom

C:\Users\wajiz.pk\PycharmProjects\TCPchatroom>python client.py
Choose your nickname: Affaf
Connection was refused because of Ban!
```



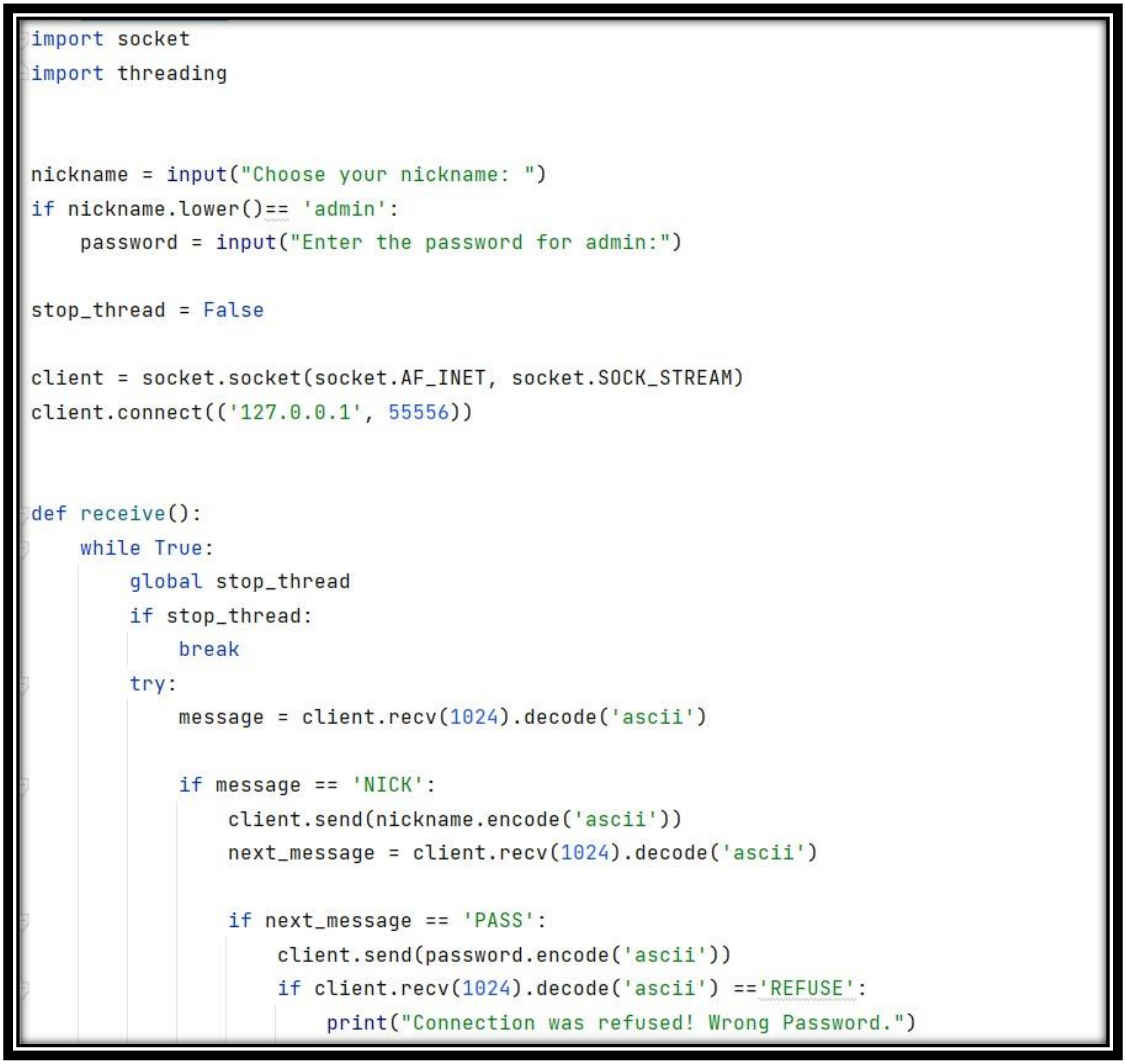
```
C:\Users\wajiz.pk\PycharmProjects\TCPchatroom>python client.py
Choose your nickname: Asiya
Connected to the server!
/ban Mounazza
Commands can only be executed by the admin!
```



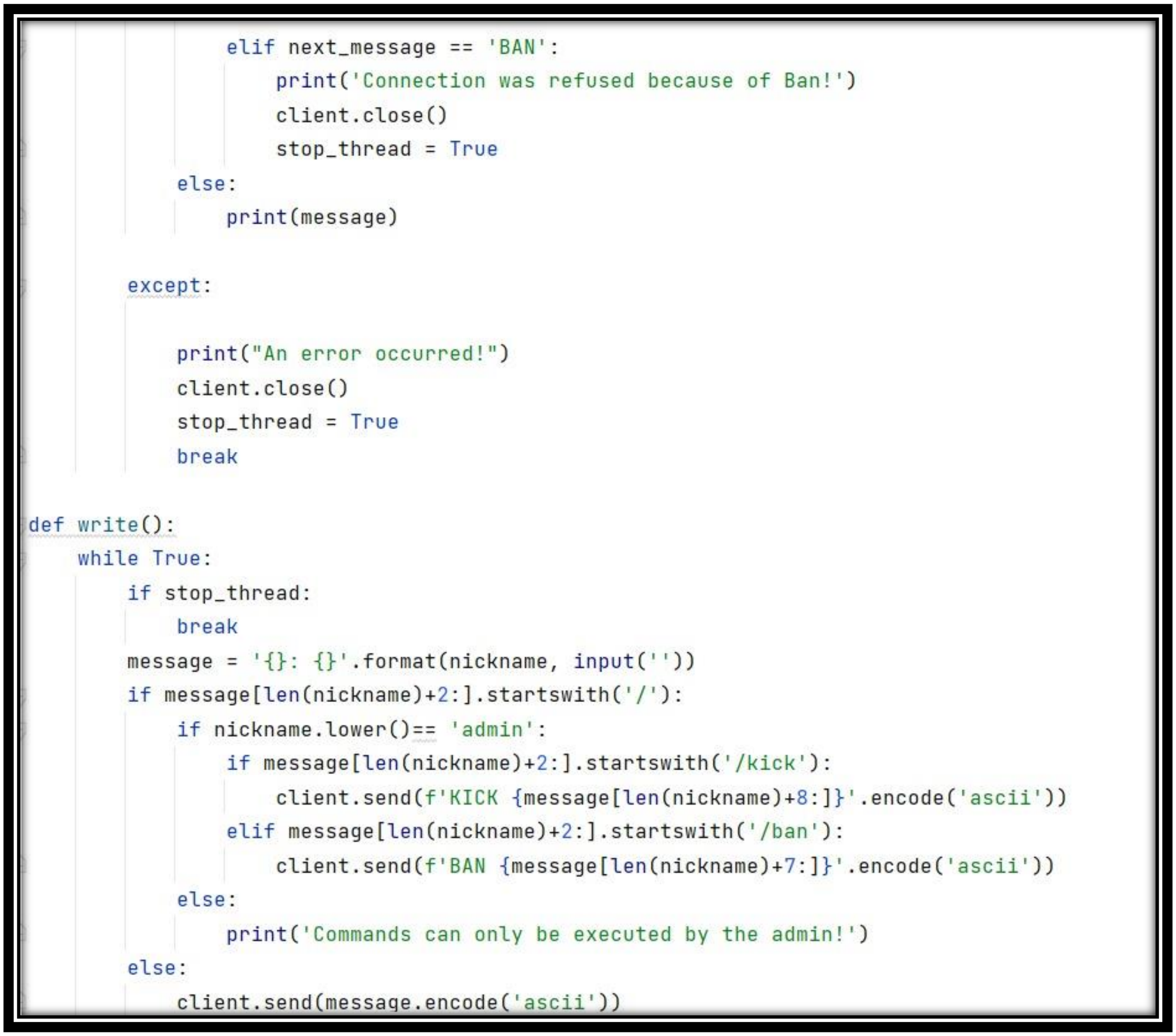
```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\wajiz.pk>cd PycharmProjects

C:\Users\wajiz.pk\PycharmProjects>cd TcpChatroom

C:\Users\wajiz.pk\PycharmProjects\TCPchatroom>python client.py
Choose your nickname: admin
Enter the password for admin:hello123
Connection was refused! Wrong Password.
```

IMPLEMENTATION

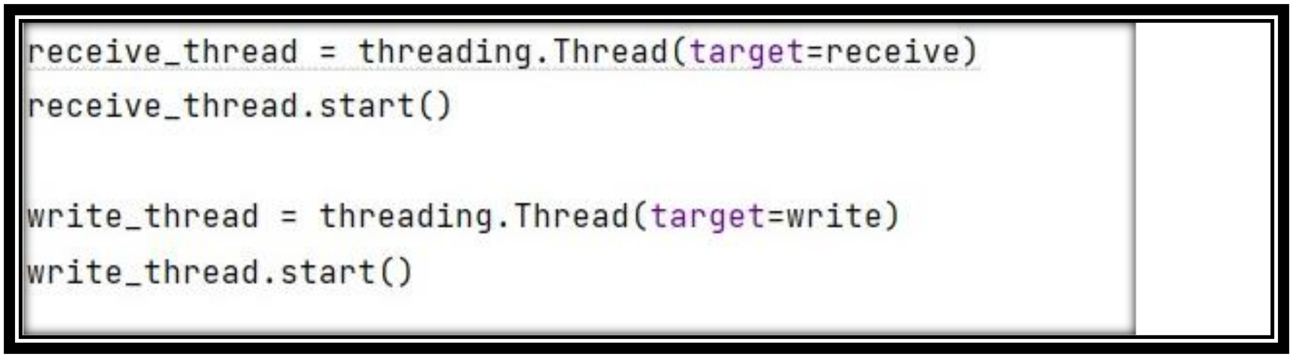
Client.txt:



```
import socket
import threading


nickname = input("Choose your nickname: ")
if nickname.lower()== 'admin':
    password = input("Enter the password for admin:")

stop_thread = False

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect(('127.0.0.1', 55556))


def receive():
    while True:
        global stop_thread
        if stop_thread:
            break
        try:
            message = client.recv(1024).decode('ascii')

            if message == 'NICK':
                client.send(nickname.encode('ascii'))
                next_message = client.recv(1024).decode('ascii')

                if next_message == 'PASS':
                    client.send(password.encode('ascii'))
                    if client.recv(1024).decode('ascii') =='REFUSE':
                        print("Connection was refused! Wrong Password.")
```

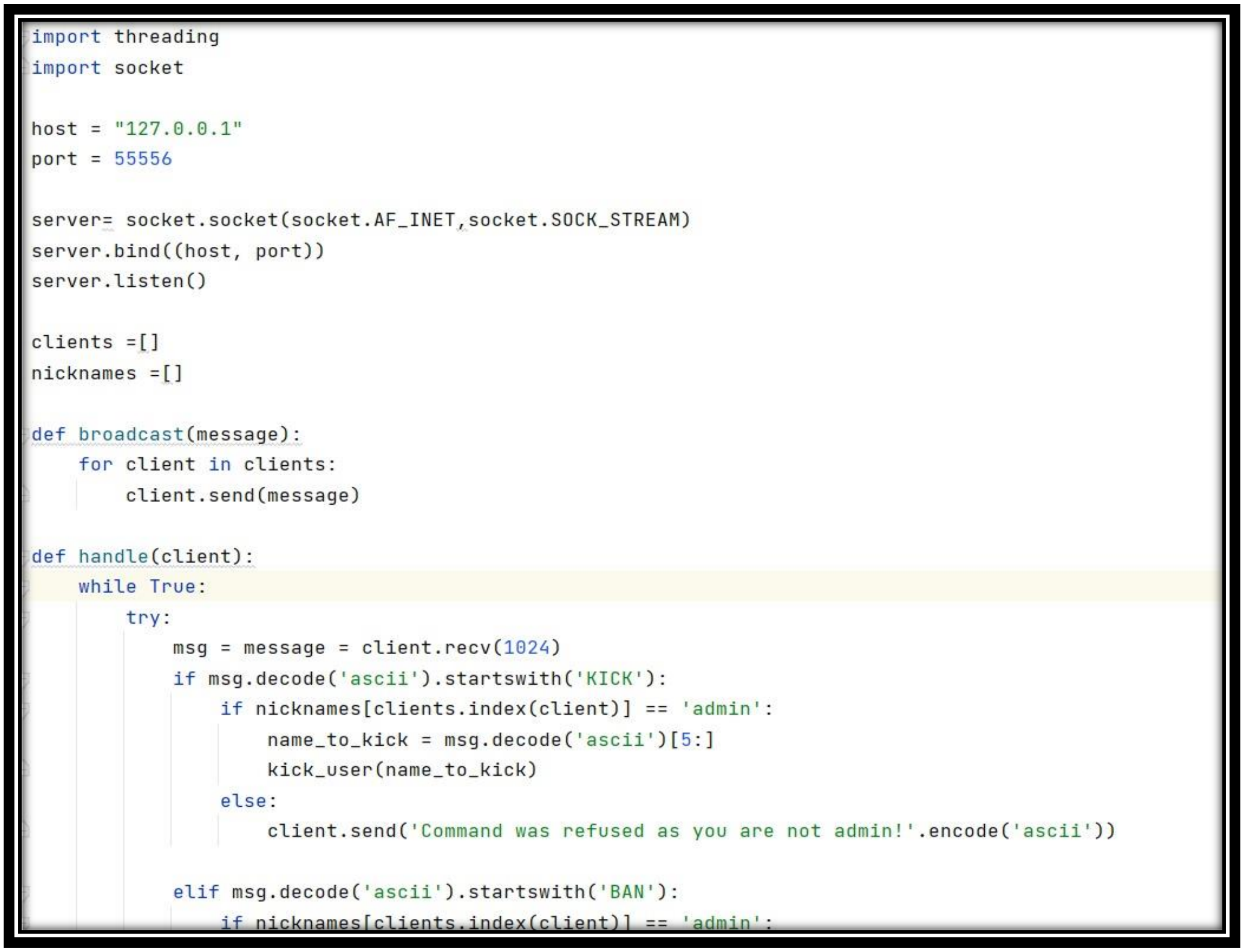


```
                elif next_message == 'BAN':
                    print('Connection was refused because of Ban!')
                    client.close()
                    stop_thread = True
            else:
                print(message)

        except:

            print("An error occurred!")
            client.close()
            stop_thread = True
            break

def write():
    while True:
        if stop_thread:
            break
        message = '{}: {}'.format(nickname, input(''))
        if message[len(nickname)+2:].startswith('/'):
            if nickname.lower()== 'admin':
                if message[len(nickname)+2:].startswith('/kick'):
                    client.send(f'KICK {message[len(nickname)+8:]}'.encode('ascii'))
                elif message[len(nickname)+2:].startswith('/ban'):
                    client.send(f'BAN {message[len(nickname)+7:]}'.encode('ascii'))
            else:
                print('Commands can only be executed by the admin!')
        else:
            client.send(message.encode('ascii'))
```

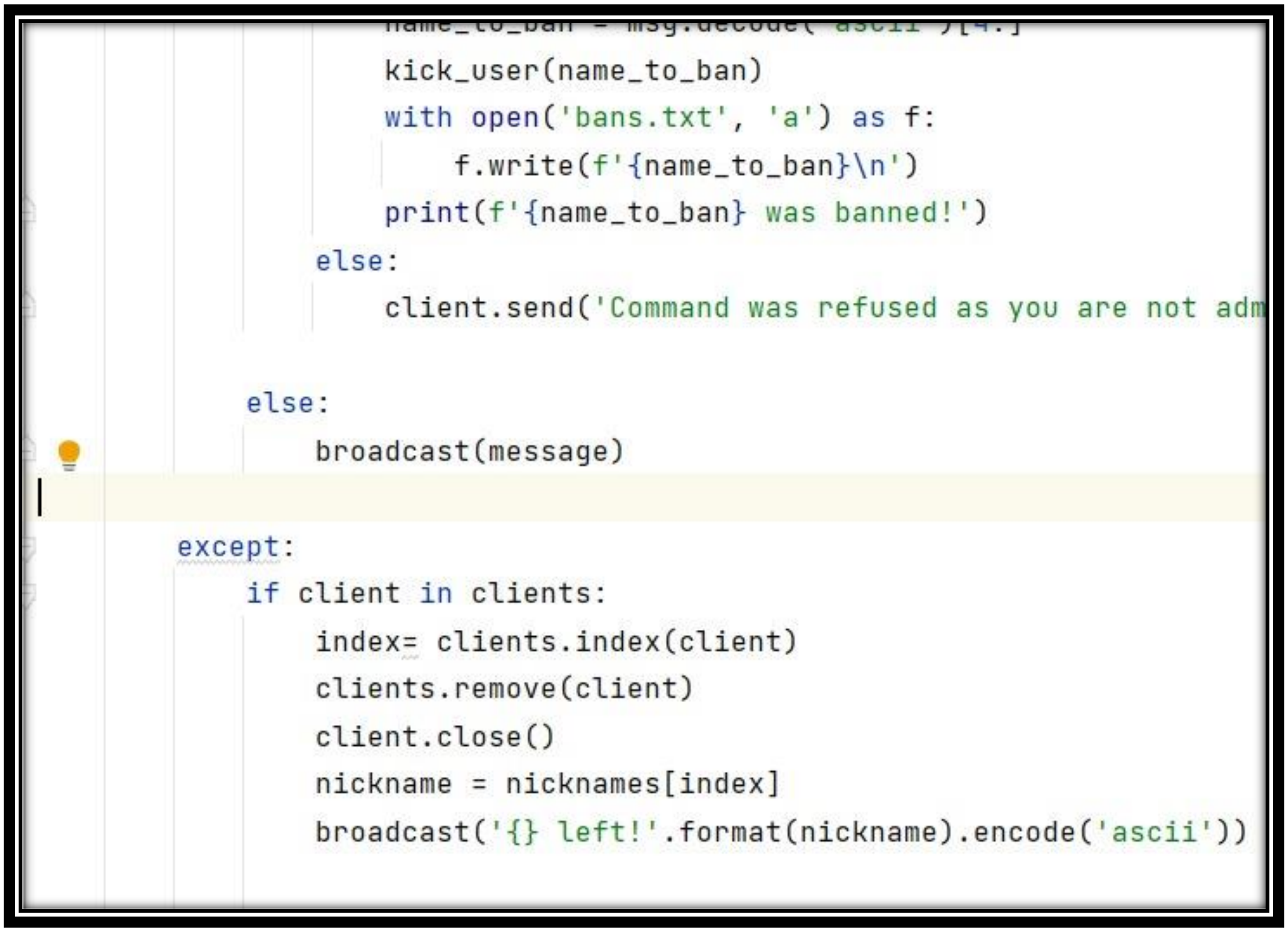


```
receive_thread = threading.Thread(target=receive)
receive_thread.start()

write_thread = threading.Thread(target=write)
write_thread.start()
```

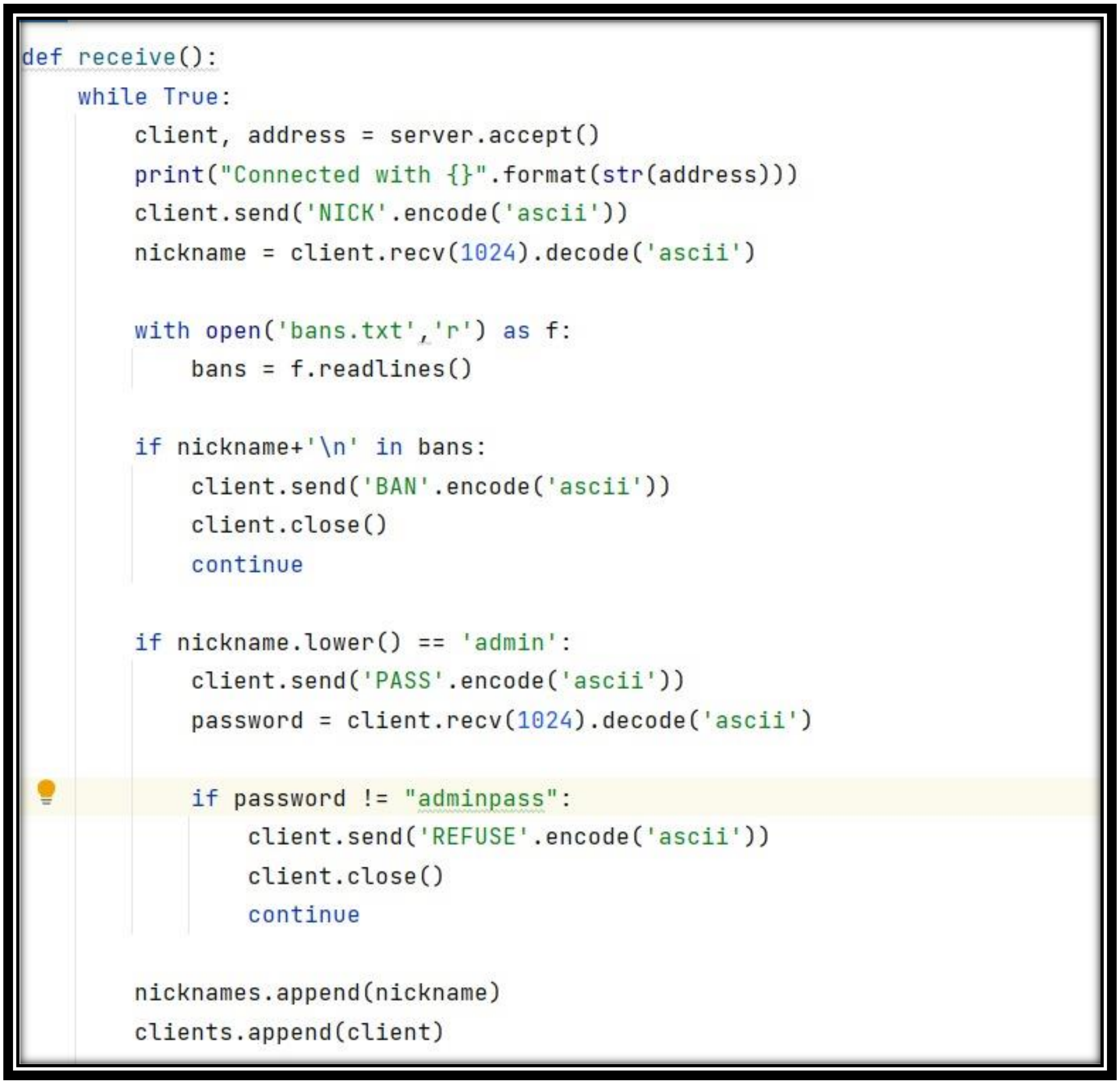
Server.txt



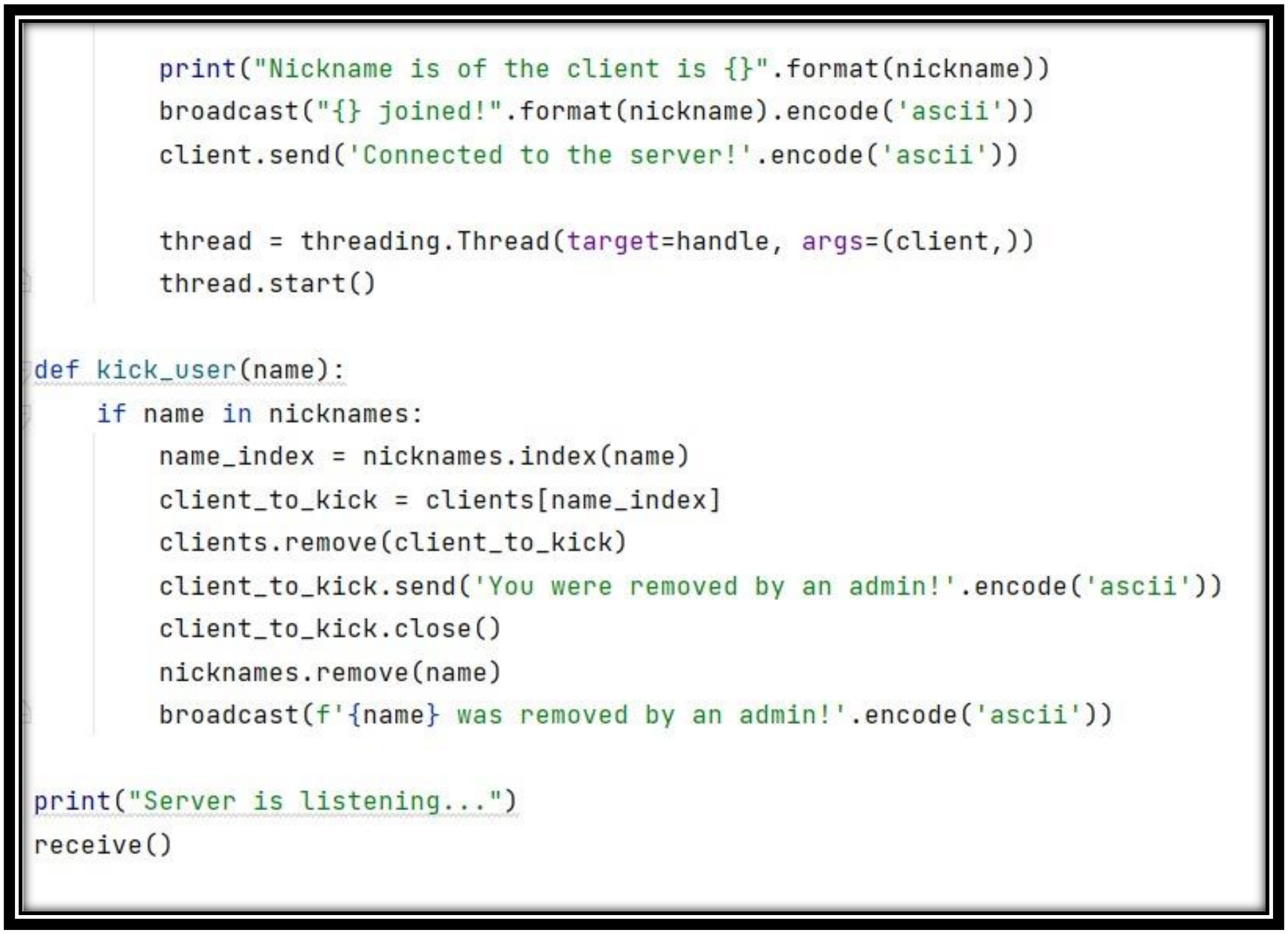
```
import threading
import socket

host = "127.0.0.1"
port = 55556

server= socket.socket(socket.AF_INET,socket.SOCK_STREAM)
server.bind((host, port))
server.listen()

clients =[]
nicknames =[]

def broadcast(message):
    for client in clients:
        client.send(message)

def handle(client):
    while True:
        try:
            msg = message = client.recv(1024)
            if msg.decode('ascii').startswith('KICK'):
                if nicknames[clients.index(client)] == 'admin':
                    name_to_kick = msg.decode('ascii')[5:]
                    kick_user(name_to_kick)
                else:
                    client.send('Command was refused as you are not admin!'.encode('ascii'))

            elif msg.decode('ascii').startswith('BAN'):
                if nicknames[clients.index(client)] == 'admin':
```



```
            name_to_ban = msg.decode('ascii')[4:]
            kick_user(name_to_ban)
            with open('bans.txt', 'a') as f:
                f.write(f'{name_to_ban}\n')
            print(f'{name_to_ban} was banned!')
        else:
            client.send('Command was refused as you are not adm

    else:
        broadcast(message)

except:
    if client in clients:
        index= clients.index(client)
        clients.remove(client)
        client.close()
        nickname = nicknames[index]
        broadcast('{} left!'.format(nickname).encode('ascii'))
```



```python
def receive():
    while True:
        client, address = server.accept()
        print("Connected with {}".format(str(address)))
        client.send('NICK'.encode('ascii'))
        nickname = client.recv(1024).decode('ascii')

        with open('bans.txt','r') as f:
            bans = f.readlines()

        if nickname+'\n' in bans:
            client.send('BAN'.encode('ascii'))
            client.close()
            continue

        if nickname.lower() == 'admin':
            client.send('PASS'.encode('ascii'))
            password = client.recv(1024).decode('ascii')

            if password != "adminpass":
                client.send('REFUSE'.encode('ascii'))
                client.close()
                continue

        nicknames.append(nickname)
        clients.append(client)
```



```
        print("Nickname is of the client is {}".format(nickname))
        broadcast("{} joined!".format(nickname).encode('ascii'))
        client.send('Connected to the server!'.encode('ascii'))

        thread = threading.Thread(target=handle, args=(client,))
        thread.start()

def kick_user(name):
    if name in nicknames:
        name_index = nicknames.index(name)
        client_to_kick = clients[name_index]
        clients.remove(client_to_kick)
        client_to_kick.send('You were removed by an admin!'.encode('ascii'))
        client_to_kick.close()
        nicknames.remove(name)
        broadcast(f'{name} was removed by an admin!'.encode('ascii'))

print("Server is listening...")
receive()
```

CONCLUSION

The TCP Chatroom Application demonstrates a robust implementation of a real-time communication system with added administrative control for effective chatroom management. Its multi-client architecture and persistent ban feature ensure a secure and interactive user experience. This project serves as a solid foundation for more advanced chat application development.