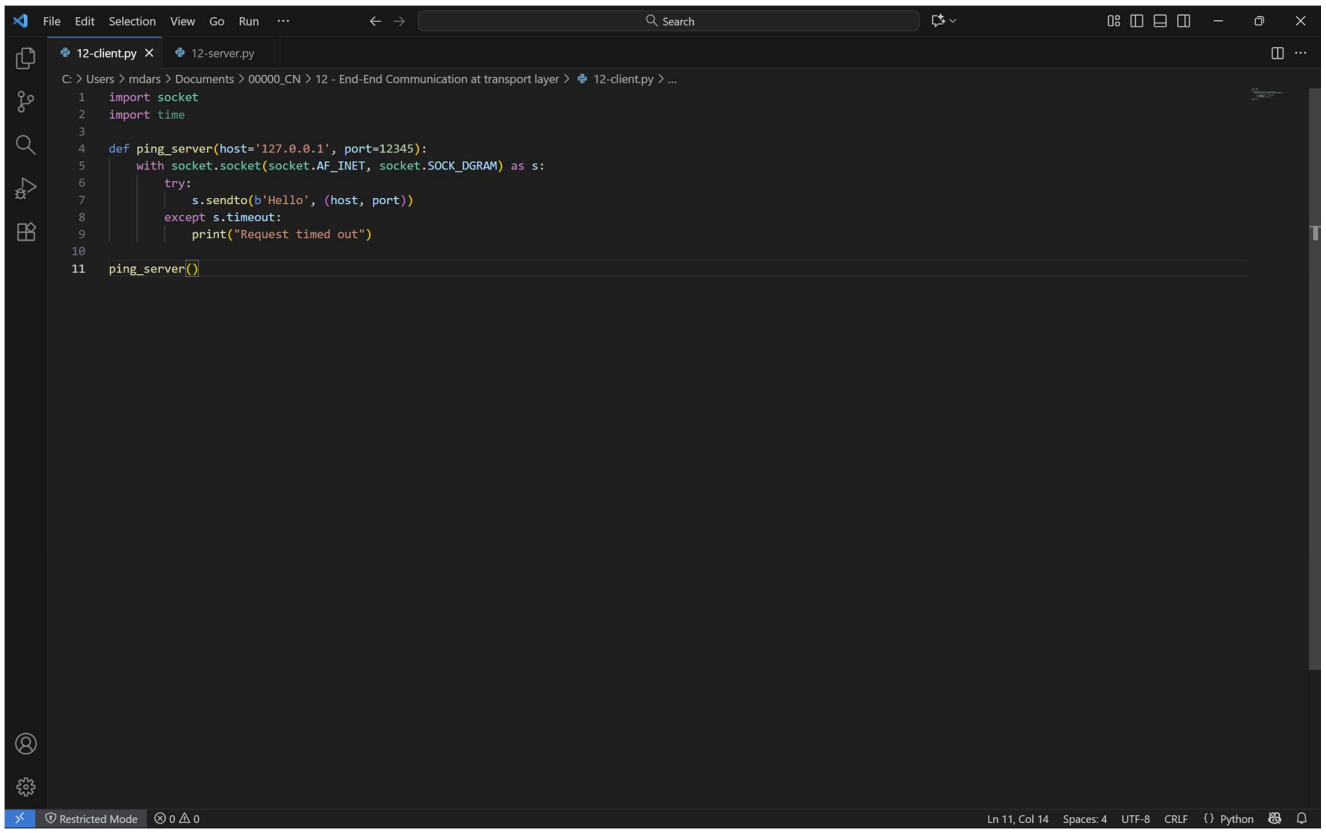
EXPERIMENT – 12

AIM: - a) Implement echo client server using TCP/UDP sockets.

CODE: -

```
import socket

def start_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        s.bind((host, port))
        print(f"UDP Server running on {host}:{port}")

        while True:
            data, addr = s.recvfrom(1024)
            print(f"Received message from {addr}: {data.decode()}")

import socket
import time

def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        try:
            s.sendto(b'Hello', (host, port))
        except s.timeout:
            print("Request timed out")

ping_server()
```

OUTPUT: -



RESULT: -

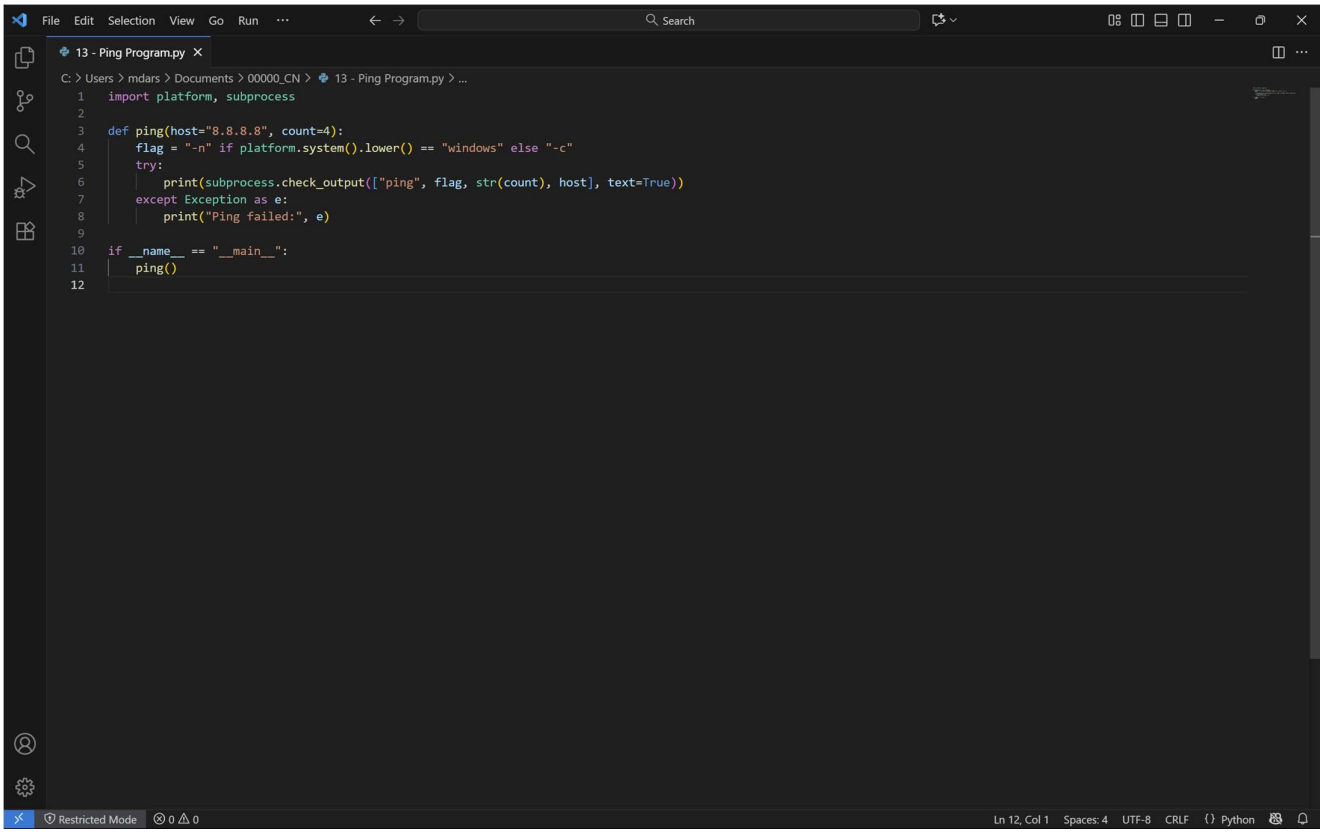
Implement echo client server using TCP/UDP sockets.

AIM: - b) Implement chat client server using TCP/UDP sockets.

CODE: -

```
import socket

def start_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        s.bind((host, port))
        print(f"UDP Server running on {host}:{port}")

        while True:
            data, addr = s.recvfrom(1024)
            print(f"Received message from {addr}: {data.decode()}")

import socket
import time

def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        try:
            s.sendto(b'Hello', (host, port))
        except s.timeout:
            print("Request timed out")

ping_server()
```

OUTPUT: -



RESULT: -

Implement chat client server using TCP/UDP sockets.