Exp 12.

DATE:-

AIM:-

To implement echo client server using TCP/UDP sockets.

ALGORITHM:-

Server.py:-

→ Create a UDPSocket
→ Bind the socket to specific IP address (127.0.0.1) & port (12345)
→ Continuous listen for incoming message
→ When message recieved - decode it
→ display message along with sender address
→ Repeat infinitely

Client.py

→ Create UDP Socket
→ Set a timeout for socket to avoid waiting
→ Send a predefined message hello to server IP address & port #12345.
→ If no response recieved in timeout period, print timeout message.
→ close socket.

CODE

Server.py

```
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()}')
if __name__ == "__main__":
    startserver()
```

Client.py

```
def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.
                                    SOCK_DGRAM) as s:
        s.timeout(5)
        try:
            s.sendto(b'Hello', (host, port))
            print("message done to the server")
        except socket.timeout:
            print('Request timed out')
if __name__ == "__main__"
    pingserver()
```

OUTPUT

Server.py

>python server.py

>>
UDP Server running on 127.0.0.1 : 12345

Client.py

>python client.py

>>
message sent to server

Server terminal:

Received message from ('127.0.0.1', 56003): Hello

RESULT:-

Thus the program of echo client server using UDP socket has been implemented & executed successfully.