Echo client

Ex. 12a
Date : 15/10/24

Aim : To implement echo client server using TCP/UDP sockets

Algorithms

Server.py:

→ create a UDP socket

→ bind the socket to specific IP address & port

→ when message received 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 12347

→ close socket after sending message

CODE

server.py

```
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 --":
Start server()
```

client.py

```
def ping -
with sock
        sock
s.timeout
try:
s.send
print (
print (
if --nam
ping -
```

output:

server.

Termin

```
> pytho
>> UD
```

client.

Termin

```
> pytho
>>
```

Serve

Client.py

```
def ping_server(host = '127.0.0.1', port=1234):
with socket.socket(socket.AF_INET,
        socket.sock_DGRAM) as s:
s.timeout(5)
try:
s.sendto(b'Hello", (host, port))
print("message sent to server")
print("Request timed out")
if __name__ == "__main__"
ping_server()
```

output:

server.py

Terminal

```
>python server.py
>> UDP server running on 127.0.0.1:1235
```

Client.py

Terminal

```
>python client.py
>> message sent to server
```

Server terminal:

```
received message from (127.0.0.1', 56003): Hello
```

Result:

Thus program of echo is executed successfully.

[illegible]