Networking

Practical NO:- 8 ~~Database Connectivity~~

Q1.] Write a python program to implement client server architecture. in which client will request for current time from server.

```
#Server side script
import socket
import time

#create a socket object
serversocket = socket.socket( socket.AFINET, socket. SOCK_STREAM)

#get local machine name
host = socket.gethostname()
port = 9999
serversocket. bind((host,port))

#queue upto 5 request
serversocket.listen(5)

While True:
    #establish connection
    clientsocket ,addr = serversocket. accept()
    print("Connected to %s", %str(addr))
    currenttime = time.ctime (time.time()) + "\r\n"
    clientsocket. send (currenttime.encode("ascii"))
    clientsocket. close()

# Client Side Script
import socket
#create a socket object
s = socket.socket( socket.AF_INET , socket. SOCK_STREAM)
```

```
#get local machine name
host = socket.gethostname()
port = 9999

# connection to hostname on the port
s.connect((host, port))

# recieve no more than 1024 bytes.
tm = s.recv(1024)
s.close()
print("The time is : %s" % tm.decode("ascii"))
```

Output :-

```
# Server
Connected to ('192.168.5.1', 52523)

#client
The time is : Mon Mar 17 13:40:31 2025.
```

2) Demonstrate UDP.

Code:

```
# UDP server program
import socket
# Create a UDP socket
sock = socket.socket(socket._AF_INET, socket.SOCK_DGRAM)
# Bind the socket to the port
server_address = ('localhost', 12345)
print('Starting up on {} port {}'.format(*server_address))
sock.bind(server_address)

while True:
    print("\n Waiting to recieve message")
    data, address = sock.recvfrom(4096)
    print("Recieved {} bytes from {}".format(len(data), address))
    print(data.encode())

    if data:
        sent = sock.sendto(data, address)
        print('Sent {} bytes back to {}'.format(sent, address))

# UDP Client program
import socket
# Create a UDP Client
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
server_address = ('localhost', 12345)
try:
    # Send data
    message = b'This is the message. It will be echoed'.
    sent = sock.sendto(message, server_address)
```

```
# Recieve response
  print('Waiting to recieve')
  data ,server = sock.recvfrom(4096)
  print('Recieved {!r}'.format(data))
finally:
    print('closing socket')
    socket.close()
```

Output:-

```
# Server
Starting up on localhost port 12345
Waiting to recieve message

# Client
Sending b'This is the message. It will be echoed'
Waiting to recieve.
closing socket.
```

24/03