

# TITLE:

NAME:SAURABH MUKHERJEE

CLASS:BCSE 3

ROLL NO:001910501006

GROUP:A1

ASSIGNMENT NUMBER:7

SUBJECT: COMPUTER NETWORKING LAB

## PROBLEM STATEMENT:

Implement any two protocols using TCP/UDP Socket as suitable.

1. BOOTP

2. FTP

3. DHCP

4. BGP

5. RIP

## PROTOCOLS:

### 1.FTP

FTP (File Transfer Protocol) is a network protocol for transmitting files between computers over Transmission Control Protocol/Internet Protocol connections. Within the TCP/IP suite, FTP is considered an application layer protocol. In an FTP transaction, the end user's computer is typically called the *local host*. The second computer involved in FTP is a *remote host*, which is usually a server. Both computers need to be connected via a network and configured properly to transfer files via FTP. Servers must be set up to run FTP services, and the client must have FTP software installed to access these services.

## 2.DHCP

Dynamic Host Configuration Protocol (DHCP) is a network management protocol used to automate the process of configuring devices on IP networks, thus allowing them to use network services such as DNS, NTP, and any communication protocol based on UDP or TCP. A DHCP server dynamically assigns an IP address and other network configuration parameters to each device on a network so they can communicate with other IP networks. DHCP is an enhancement of an older protocol called BOOTP.

### DESIGN:

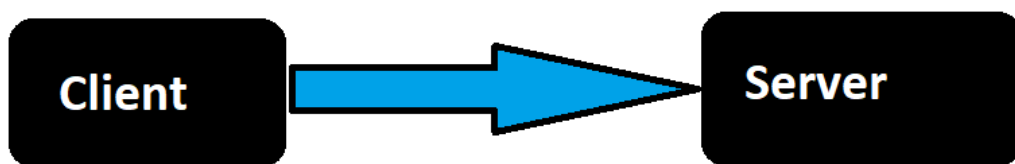
The system has two major components :

**1.Client**

**2.Server**

We have used the socket module programming in our case to create the client-server architecture and implemented the protocols FTP and DHCP.

### FTP DIAGRAMMATIC FLOW SHOWING THE CLIENT TO SERVER COMMUNICATION



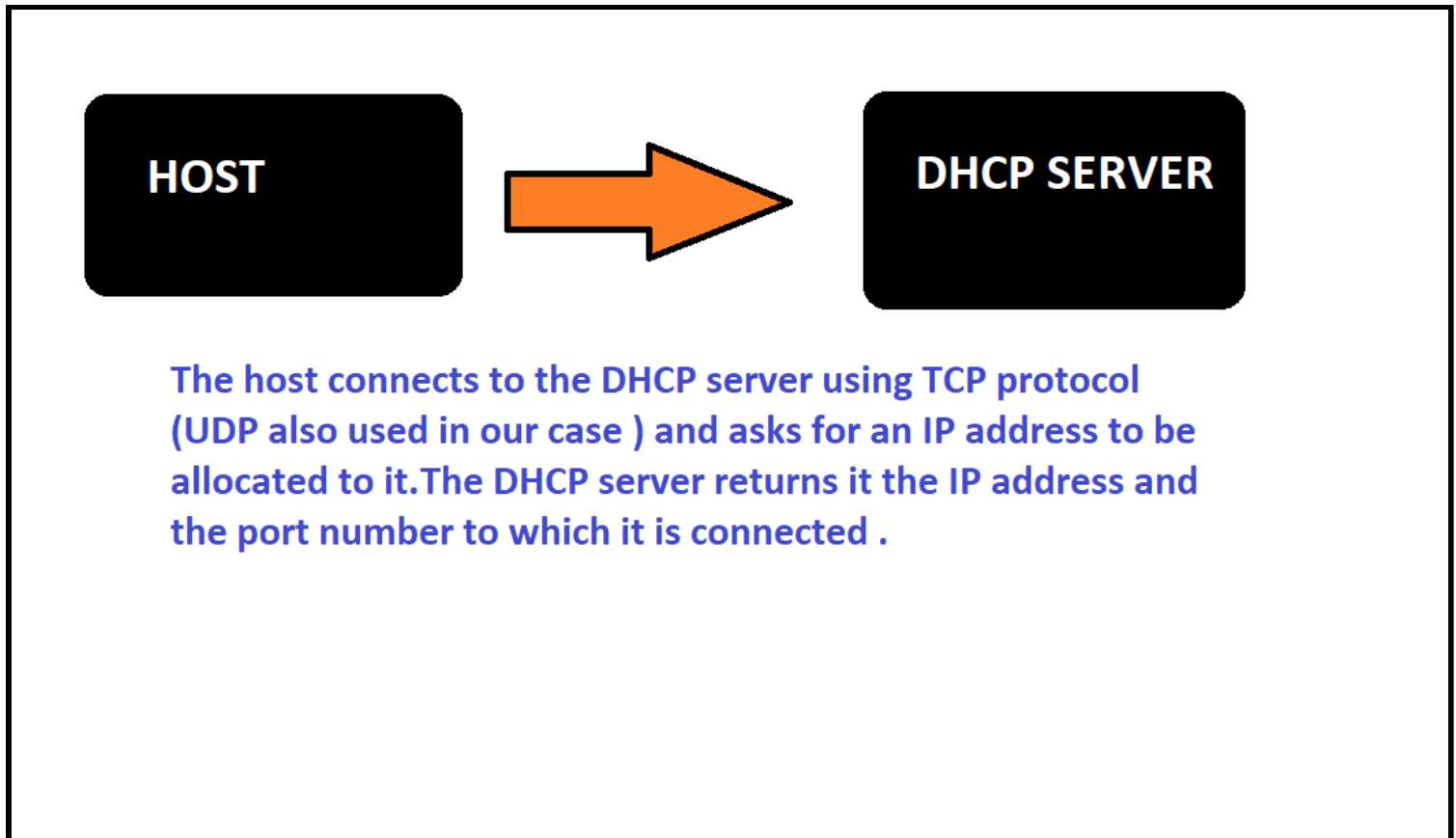
**FTP**

**1.Fetch a File**

**2.Rename a file on server**

**3.Upload a file on server database**

## FLOW DIAGRAM FOR DHCP SHOWING THE HOST TO SERVER COMMUNICATION .



## INPUT / OUTPUT FORMAT

### Input

#### 1.Client Side

As an input to the client side we have

FTP a.Name of the file to be fetched from the server .

DHCP a.Query requesting for the IP address from the server.

#### 2.Server Side

FTP a.The text file supplied as input to the server .

### Output

#### 1.Client Side

FTP a.As a client side output we print the data of the file received from the server .

DHCP b.The IP Address and the port number as a tuple of the client received from the server.

# IMPLEMENTATION

FOR FTP THE TRANSFER TAKES PLACE USING THE FILE STRUCTURE METHOD WHICH CONVERTS THE FILES AND ENCODES IT INTO BYTES AND THEN THE CLIENT MACHINE DECODES IT BEFORE READING IT .

## FTP IMPLEMENTATION :

TCP protocol has been used for this as the underlying protocol for implementing FTP .The server side has been made multi-threaded to allow multiple clients request simultaneously from a single server

### FTP CLIENT SIDE :

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

s.connect((socket.gethostname(), 12345))

filename = input("Enter filename to be searched: ")

s.send(bytes(filename, "utf-8"))

data = s.recv(1024).decode()

print("File data fetched successfully using FTP protocol")

print("File contains the following : ", data)

s.close()
```

### FTP SERVER SIDE

```
import socket

import random,pickle

from _thread import *

ThreadCount = 0
```

```
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
```

```
try:
```

```
    s.bind((socket.gethostname(),12345))
```

```
except socket.error as e:
```

```
    print(str(e))
```

```
print('Socket is listening..')
```

```
s.listen(1)
```

```
def multi_threaded_client(conn):
```

```
    filename = conn.recv(1024).decode()
```

```
    file = open(filename, 'r')
```

```
    data = file.read()
```

```
    file.close()
```

```
    conn.send(bytes(data, "utf-8"))
```

```
    conn.close()
```

```
while True:
```

```
    conn,address=s.accept()
```

```
    print('Connected to: ' + address[0] + ':' + str(address[1]))
```

```
    start_new_thread(multi_threaded_client, (conn, ))
```

```
    ThreadCount += 1
```

```
    print('Thread Number: ' + str(ThreadCount))
```

**DHCP CLIENT SIDE :**

**UDP SOCKET:**

```
import socket
```

```
msgFromClient = "Please provide me an IP address"
```

```
bytesToSend = str.encode(msgFromClient)
```

```

serverAddressPort    = ("127.0.0.1", 20001)

bufferSize           = 1024


# Create a UDP socket at client side

UDPClientSocket = socket.socket(family=socket.AF_INET,
                                type=socket.SOCK_DGRAM)

# Send to server using created UDP socket

UDPClientSocket.sendto(bytesToSend, serverAddressPort)

msgFromServer = UDPClientSocket.recvfrom(bufferSize)

msg = "The IP Address that has been assigned is :
{}".format(msgFromServer[0])

print(msg)

```

## TCP SOCKET:

```

import socket


# Initially take the address to be empty tuple and then after server assigns
it , display it .

Address = ("","")

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

s.connect((socket.gethostname(), 5000))

s.send(bytes("Requesting port number of the host" , "utf-8"))

Address = s.recv(1024).decode()

print("The host address: " ,Address)

s.close()

```

## DHCP SERVER SIDE :

## UDP SOCKET:

```

import socket


localIP      = "127.0.0.1"

localPort    = 20001


# Create a datagram socket

```

```

UDPServerSocket = socket.socket(family=socket.AF_INET,
type=socket.SOCK_DGRAM)

UDPServerSocket.bind((localIP, localPort))

print("DHCP server up and listening")

while(True):

    bytesAddressPair = UDPServerSocket.recvfrom(1024)
    message = bytesAddressPair[0]
    address = bytesAddressPair[1]

    clientMsg = "Message from Client:{}".format(message)
    clientIP  = "Client IP Address:{}".format(address)

    print(clientMsg)
    print(clientIP)

    # Sending a reply with ip address
    bytesToSend      = str.encode(clientIP)
    UDPServerSocket.sendto(bytesToSend, address)

```

## TCP SOCKET:

```

import socket

while True:

    s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    s.bind((socket.gethostname(),5000))

    print("Listening for a connection on its own port....")
    s.listen(1)
    conn,addr=s.accept()
    print("Connected by", addr)

    message = conn.recv(1024).decode() #receives requesting port number of
the host

    print(message, " received from ", addr ) #display message received
    conn.send(bytes(str(addr), "utf-8"))

```

```
conn.close()

s.close()
```

## RESULTS :

CLIENT 1:

INPUT FILE REQUESTED FROM SERVER :

FILENAME-SAURABH.TXT

DATA-Hello , How are you ?

CLIENT 2:

INPUT FILE REQUESTED FROM THE SERVER:

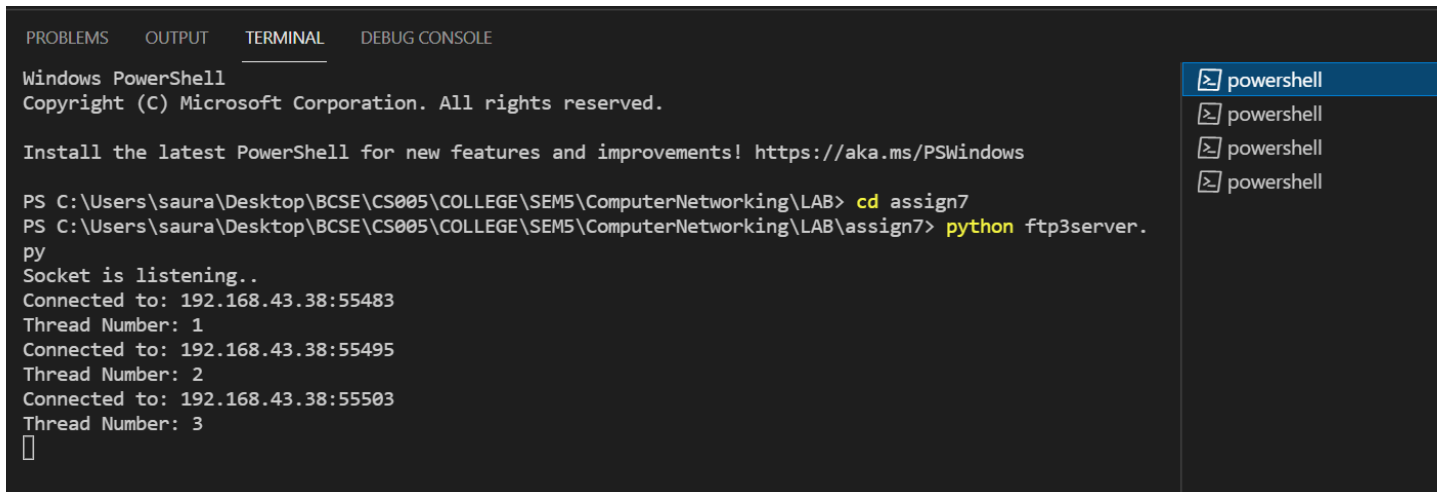
FILENAME-SAURABH2.TXT

DATA-This is another file which is requested by some other client !

INPUT FILE REQUESTED FROM THE SERVER :

FILENAME-SAURABH3.TXT

DATA-This is the third file requested by some another client !



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB> cd assign7
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7> python ftp3server.py
Socket is listening..
Connected to: 192.168.43.38:55483
Thread Number: 1
Connected to: 192.168.43.38:55495
Thread Number: 2
Connected to: 192.168.43.38:55503
Thread Number: 3
█
```



Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB> cd assign7
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7> python ftp2client.py
Enter filename to be searched: saurabh.txt
File data fetched successfully using FTP protocol
File contains the following : Hello , How are you ?
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7> 
```

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB> cd assign7
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7> python ftp2client.py
Enter filename to be searched: saurabh2.txt
File data fetched successfully using FTP protocol
File contains the following : This is another file which is requested by some other client !
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7> 
```

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB> cd assign7
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7> python ftp2client.py
Enter filename to be searched: saurabh3.txt
File data fetched successfully using FTP protocol
File contains the following : This is the third file requested by some another client !
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7> 
```

## RESULTS FOR DHCP :

In our simulation DHCP is using the address by which the client is connected to it and replying it that address to simulate the reply with IP ADDRESS and the port number .

## USING TCP :

```
1 import socket
2
3 # Initially take the address to be empty tuple and then after server assigns it , display it .
4 Address = ("","")
5 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6 s.connect((socket.gethostname(), 5000))
7 s.send(bytes("Requesting port number of the host" , "utf-8"))
8 Address = s.recv(1024).decode()
9 print("The host address: " ,Address)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB> & C:/Users/saura/AppData/Local/Programs/Python/Python39/python.exe c:/Users/saura/Desktop/BCSE/CS005/COLLEGE/SEM5/ComputerNetworking/LAB/ASSIGN7/dhcpServer.py  
Listening for a connection on its own port....  
Connected by ('192.168.43.38', 55673)  
Requesting port number of the host received from ('192.168.43.38', 55673)  
Listening for a connection on its own port....  
█

Python powershell

```
1 import socket
2
3 # Initially take the address to be empty tuple and then after server assigns it , display it .
4 Address = ("","")
5 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6 s.connect((socket.gethostname(), 5000))
7 s.send(bytes("Requesting port number of the host" , "utf-8"))
8 Address = s.recv(1024).decode()
9 print("The host address: " ,Address)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB> cd assign7  
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7> python dhcpclient.py  
The host address: ('192.168.43.38', 55673)  
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7> █

Python powershell

## USING UDP:

dhcp\_UDP\_server.py - LAB - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

- LAB
  - ASSIGN1
  - ASSIGN2
  - ASSIGN3
  - FINAL
  - IMAGES
  - TRIAL
  - ASSIGN4
  - CLIENT.py
  - SERVER.py
  - ASSIGN7

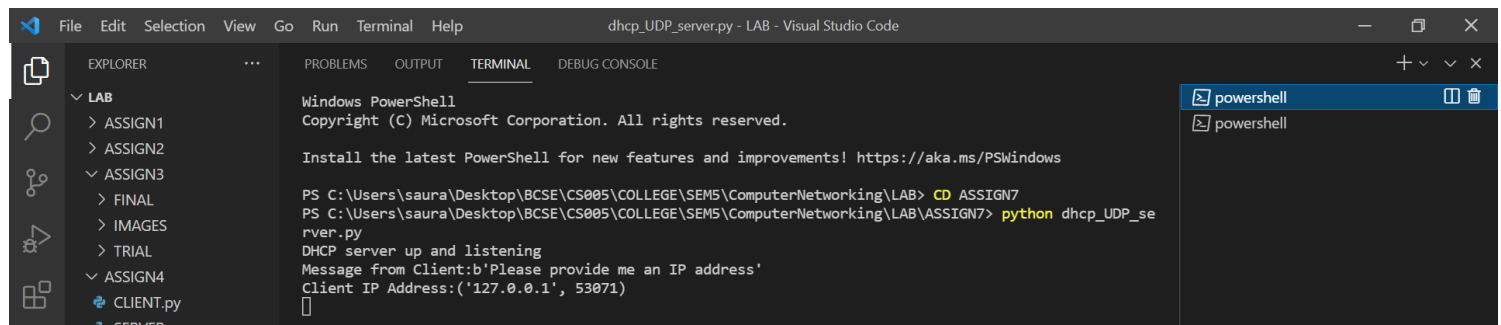
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB> cd ASSIGN7  
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\ASSIGN7> python dhcp\_UDP\_server.py  
DHCP server up and listening  
█

powershell



```
File Edit Selection View Go Run Terminal Help dhcp_UDP_server.py - LAB - Visual Studio Code

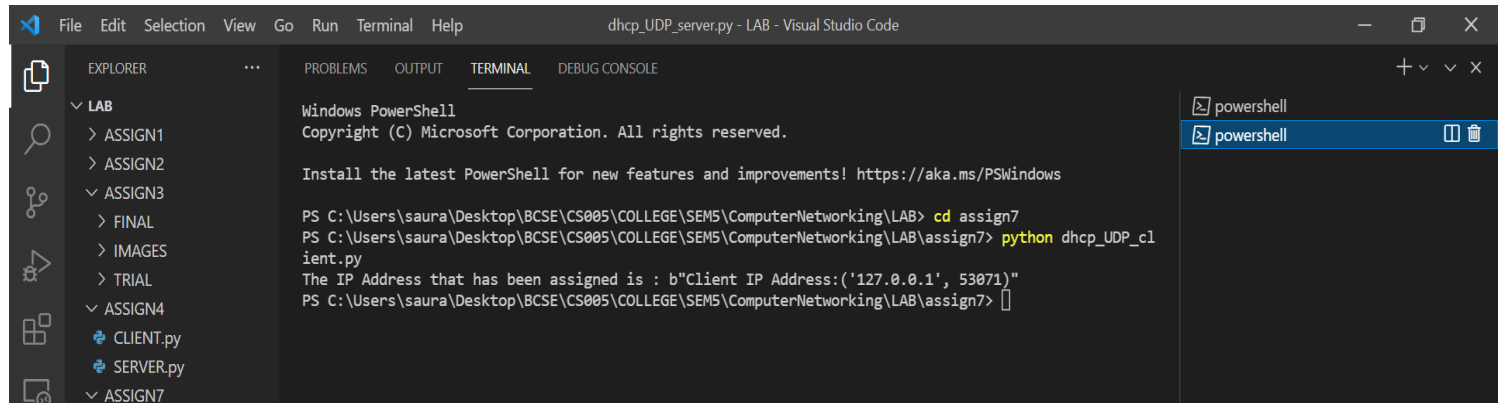
EXPLORER
LAB
  ASSIGN1
  ASSIGN2
  ASSIGN3
  FINAL
  IMAGES
  TRIAL
  ASSIGN4
  CLIENT.py
  SERVER.py

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB> cd ASSIGN7
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\ASSIGN7> python dhcp_UDP_server.py
DHCP server up and listening
Message from Client:b'Please provide me an IP address'
Client IP Address:('127.0.0.1', 53071)
```



```
File Edit Selection View Go Run Terminal Help dhcp_UDP_server.py - LAB - Visual Studio Code

EXPLORER
LAB
  ASSIGN1
  ASSIGN2
  ASSIGN3
  FINAL
  IMAGES
  TRIAL
  ASSIGN4
  CLIENT.py
  SERVER.py
  ASSIGN7

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB> cd assign7
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7> python dhcp_UDP_client.py
The IP Address that has been assigned is : b"Client IP Address:('127.0.0.1', 53071)"
PS C:\Users\saura\Desktop\BCSE\CS005\COLLEGE\SEM5\ComputerNetworking\LAB\assign7>
```

## ANALYSIS:

This project helped me clear the fundamental concepts of the TCP AND UDP protocols and also help to understand how the protocol works in real scenario.Coding was really a nice experience for me and I learned a lot doing this .