COMPUTER NETWORKS LAB

ASSIGNMENT-1

SAI DINESH

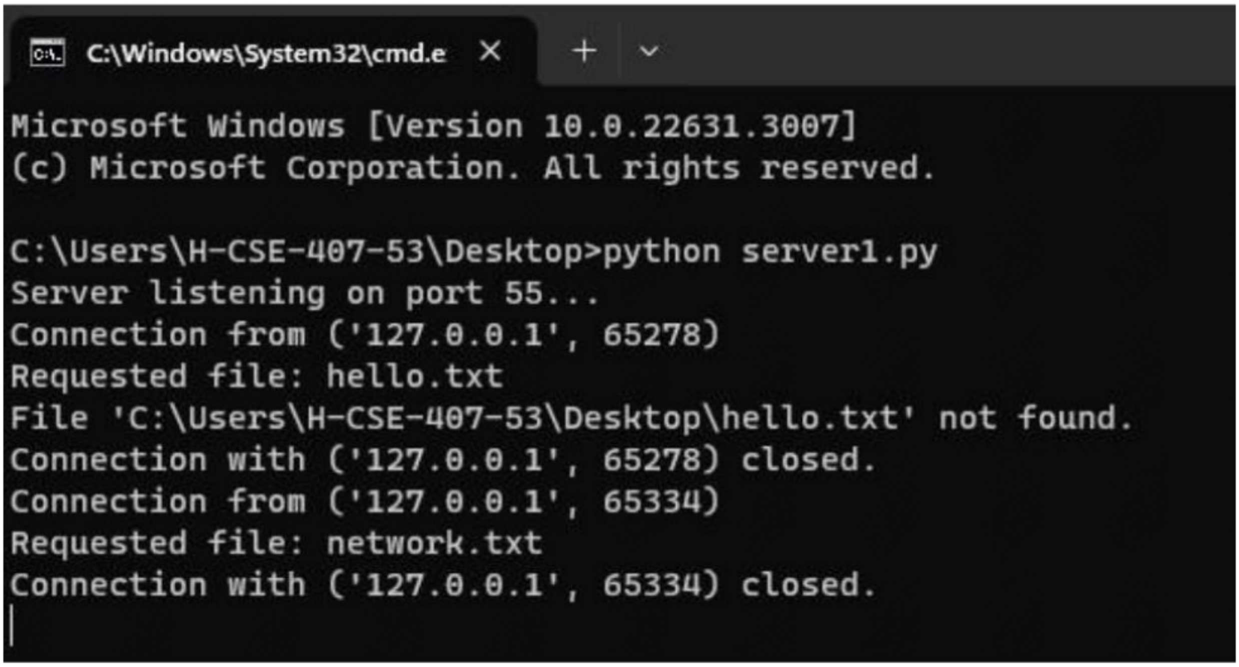
HU22CSEN0300193

Write a program to create a chat server that listens to port 54 using stream sockets. Write a simple client program to connect to the server. Send multiple text messages from the client to the server and vice versa. When either party types “Bye”, close the connection

SERVER.py:

```
import socket
import os


def send_file(client_socket, filename):
    try:
        with open(filename, 'rb') as file:
            file_data = file.read()
            client_socket.sendall(file_data)
    except FileNotFoundError:
        print(f"File '{filename}' not found.")
        client_socket.sendall(b"File not found")


def main():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(('localhost', 55))
    server_socket.listen(1)

    print("Server listening on port 55...")
    while True:
        client_socket, client_address = server_socket.accept()
        print(f"Connection from {client_address}")

        request = client_socket.recv(1024).decode('utf-8').strip()

        if request.startswith("GET "):
            filename = request.split(" ")[1]
            print(f"Client requested file: {filename}")
            send_file(client_socket, filename)

        client_socket.close()

if __name__ == "__main__":
    main()
```

OUTPUT:

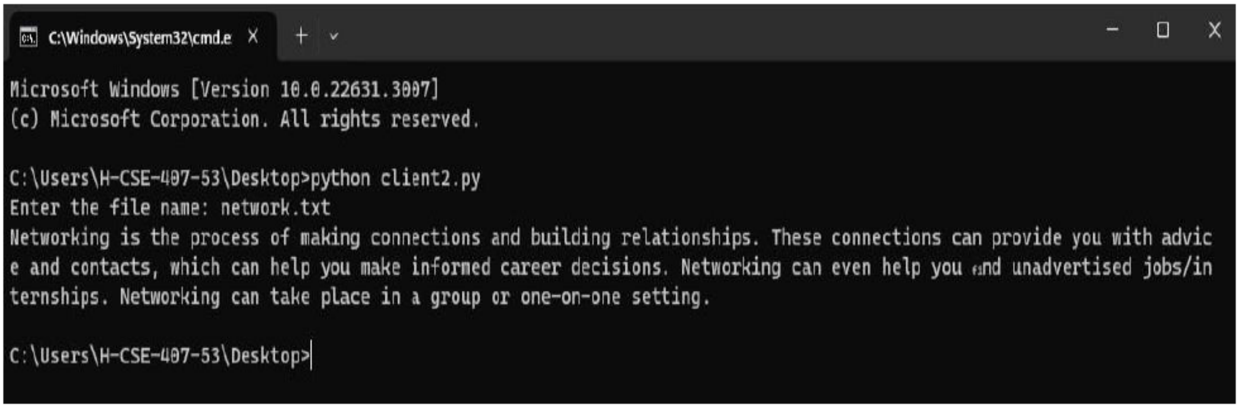


```
Microsoft Windows [Version 10.0.22631.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\H-CSE-407-53\Desktop>python server1.py
Server listening on port 55...
Connection from ('127.0.0.1', 65278)
Requested file: hello.txt
File 'C:\Users\H-CSE-407-53\Desktop\hello.txt' not found.
Connection with ('127.0.0.1', 65278) closed.
Connection from ('127.0.0.1', 65334)
Requested file: network.txt
Connection with ('127.0.0.1', 65334) closed.
```

CLIENT:

```
import socket


def main():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_address = ('localhost', 55)


    try:
        client_socket.connect(server_address)


        filename = input("Enter the name of the file to request: ")
        request = f"GET {filename}"
        client_socket.sendall(request.encode('utf-8'))

        file_data = client_socket.recv(1024)

        if file_data == b"File not found":
            print("File not found on the server.")
        else:
            with open(filename, 'wb') as file:
                file.write(file_data)
                print(f"File '{filename}' received successfully.")

    finally:
        client_socket.close()

if __name__ == "__main__":
    main()
```

OUTPUT:



```
Microsoft Windows [Version 10.0.22631.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\H-CSE-407-53\Desktop>python client2.py
Enter the file name: network.txt
Networking is the process of making connections and building relationships. These connections can provide you with advic
e and contacts, which can help you make informed career decisions. Networking can even help you find unadvertised jobs/in
ternships. Networking can take place in a group or one-on-one setting.

C:\Users\H-CSE-407-53\Desktop>
```

COMPUTER NETWORKS ASSIGNMENT

M. Sai Dinesh
HU22CSEN0300193

CODE FOR SERVER :

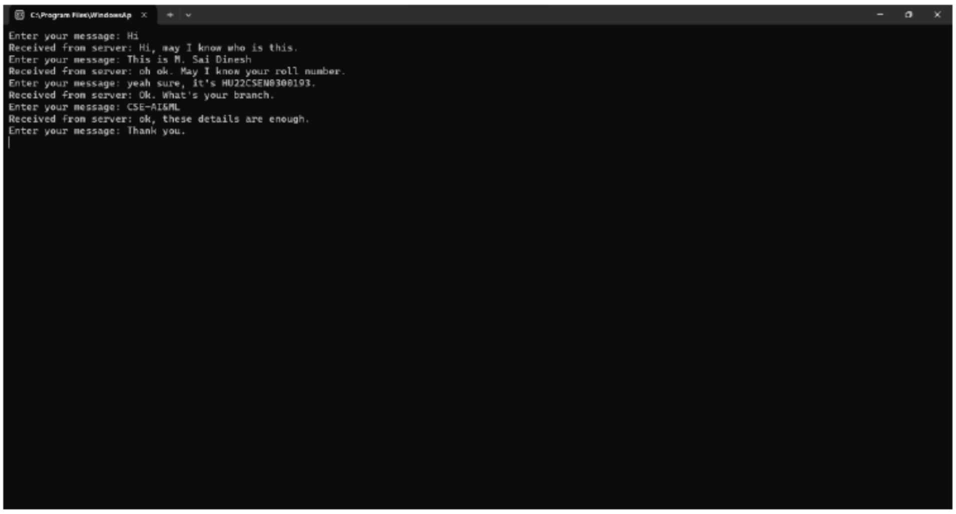
Write a program to create a server that listens to port 55 using stream sockets. Write a simple client program to connect to the server. The client should request for a text file and the server should return the file before terminating the connection

```
import socket
import threading

def handle_client(client_socket):
    while True:
        try:
            data = client_socket.recv(1024).decode('utf-8')
            if not data:
                break

            print(f"Received from client: {data}")

            if data.lower() == "bye":
                break

            response = input("Enter your response: ")
            client_socket.send(response.encode('utf-8'))

        except Exception as e:
            print(f"Error: {e}")
            break

    print("Connection closed.")
    client_socket.close()
def start_server():
    host = '127.0.0.1'
    port = 54
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind((host, port))
    server_socket.listen(5)

    print(f"Server listening on port {port}")

    while True:
        client_socket, addr = server_socket.accept()
        print(f"Accepted connection from {addr}")
        client_handler = threading.Thread(target=handle_client,
args=(client_socket,))
        client_handler.start()

if __name__ == "__main__":
    start_server()
```

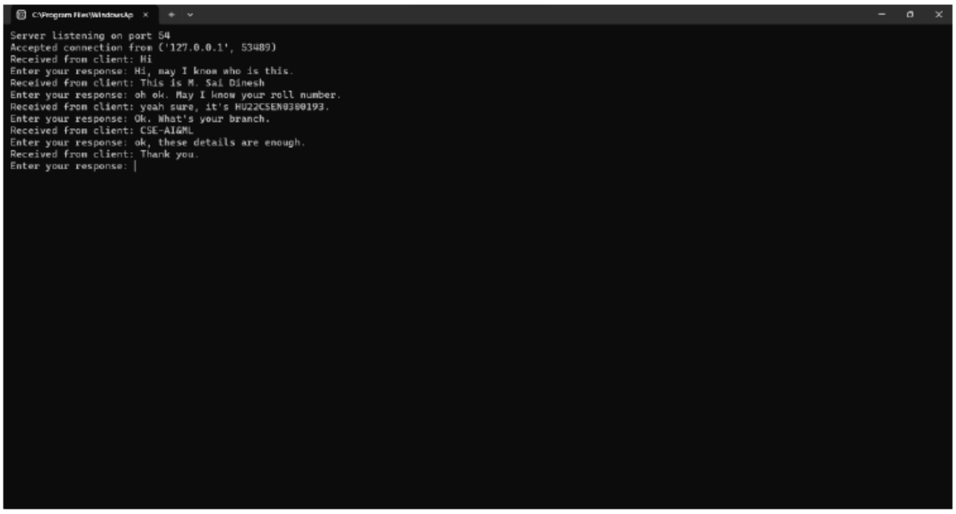
CODE FOR CLIENT :

```
import socket

def start_client():
    host = '127.0.0.1'
    port = 54

    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect((host, port))

    while True:
        message = input("Enter your message: ")
        client_socket.send(message.encode('utf-8'))

        if message.lower() == "bye":
            break

        response = client_socket.recv(1024).decode('utf-8')
        print(f"Received from server: {response}")

        if response.lower() == "bye":
            break

    print("Connection closed.")
    client_socket.close()

if __name__ == "__main__":
    start_client()
```

OUTPUT :



```
Enter your message: Hi
Received from server: Hi, may I know who is this.
Enter your message: This is M. Sai Dinesh
Received from server: oh ok. May I know your roll number.
Enter your message: yeah sure, it's HU22CSEN0300193.
Received from server: Ok. What's your branch.
Enter your message: CSE-AI&ML
Received from server: ok, these details are enough.
Enter your message: Thank you.
```



```
Server listening on port 54
Accepted connection from ('127.0.0.1', 53489)
Received from client: Hi
Enter your response: Hi, may I know who is this.
Received from client: This is M. Sai Dinesh
Enter your response: oh ok. May I know your roll number.
Received from client: yeah sure, it's HU22CSEN0300193.
Enter your response: Ok. What's your branch.
Received from client: CSE-AI&ML
Enter your response: ok, these details are enough.
Received from client: Thank you.
Enter your response:
```

COMPUTER NETWORKS ASSIGNMENT

M. Sai Dinesh
HU22CSEN0300193

CODE FOR SERVER :

Write a program to create a server that listens to port 53 using stream sockets. Write a simple client program to connect to the server. Send a simple text message “Hello” from the client to the server and the server to the client and close the connection.

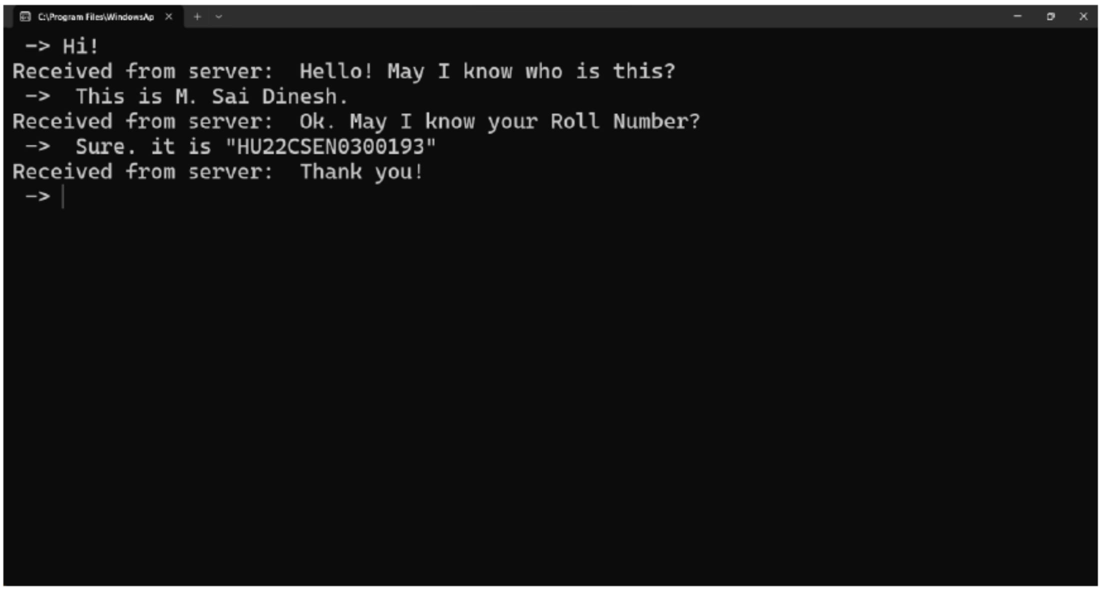
```
import socket

def server_program():
   host = socket.gethostname()
   port = 5000

   server_socket = socket.socket()
   server_socket.bind((host, port))

   server_socket.listen(2)
   conn, address = server_socket.accept()
   print("Connection from: " + str(address))
   while True:
      data = conn.recv(1024).decode()
      if not data:
         # if data is not received break
         break
      print("from connected user: " + str(data))
      data = input(' -> ')
      conn.send(data.encode())
   conn.close()

if __name__ == '__main__':
   server_program()
```

CODE FOR CLIENT :

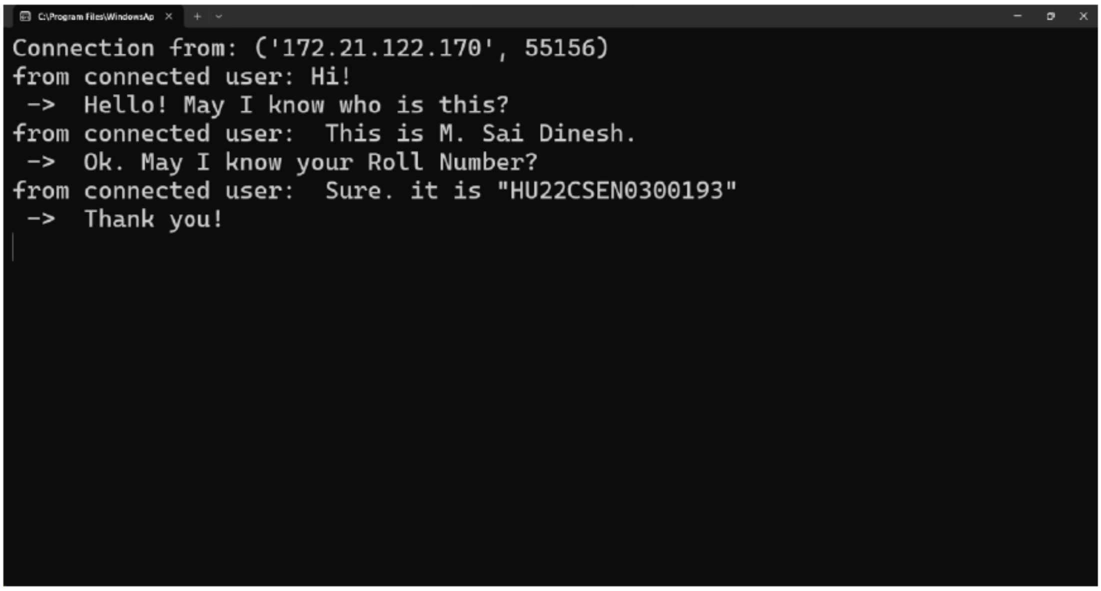
```
import socket

def client_program():
   host = socket.gethostname()
   port = 5000

   client_socket = socket.socket()
   client_socket.connect((host, port))
   message = input(" -> ")  # take input

   while message.lower().strip() != 'bye':
      client_socket.send(message.encode())
      data = client_socket.recv(1024).decode()

      print('Received from server: ' + data)
```

```
        message = input(" -> ")
    client_socket.close()


if __name__ == '__main__':
    client_program()
```

OUTPUT :



```
 -> Hi!
Received from server:  Hello! May I know who is this?
 ->  This is M. Sai Dinesh.
Received from server:  Ok. May I know your Roll Number?
 ->  Sure. it is "HU22CSEN0300193"
Received from server:  Thank you!
 ->
```



```
Connection from: ('172.21.122.170', 55156)
from connected user: Hi!
 ->  Hello! May I know who is this?
from connected user:  This is M. Sai Dinesh.
 ->  Ok. May I know your Roll Number?
from connected user:  Sure. it is "HU22CSEN0300193"
 ->  Thank you!
```

COMPUTER NETWORK

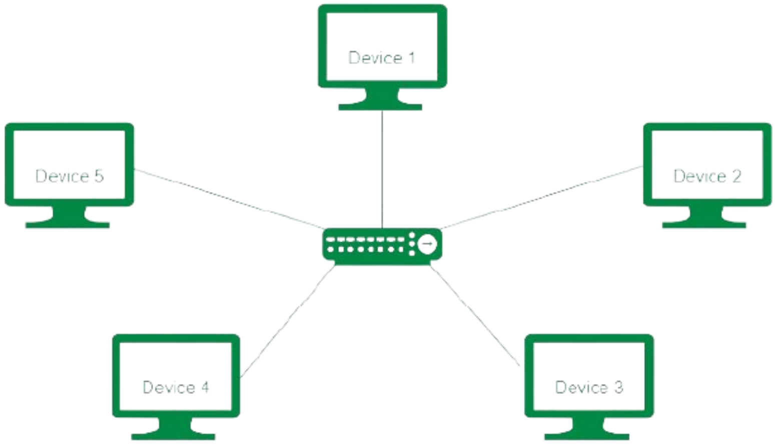
NETWORK TOPOLOGY

M. SAI DINESH - HU22CSEN0300193

What is Network Topology : A network topology is the physical and logical arrangement of nodes and connections in a network. Nodes usually include devices such as switches, routers and software with switch and router features.

Topology used in my Organization : STAR TOPOLOGY

What is Star Topology : Star topology is a network topology in which each network component is physically connected to a central node such as a router, hub or switch.



Star Topology

Type of connection devices used in my Organization :

- Modem
- Router
- NetworkSwitch
- Ethernet Hub
- Repeater

Speed of the wired and wireless LAN in my Organization :

I've Used these Commands To find the Speed of Wired & Wireless LAN

"wmic nic where netEnabled=true get name,speed"

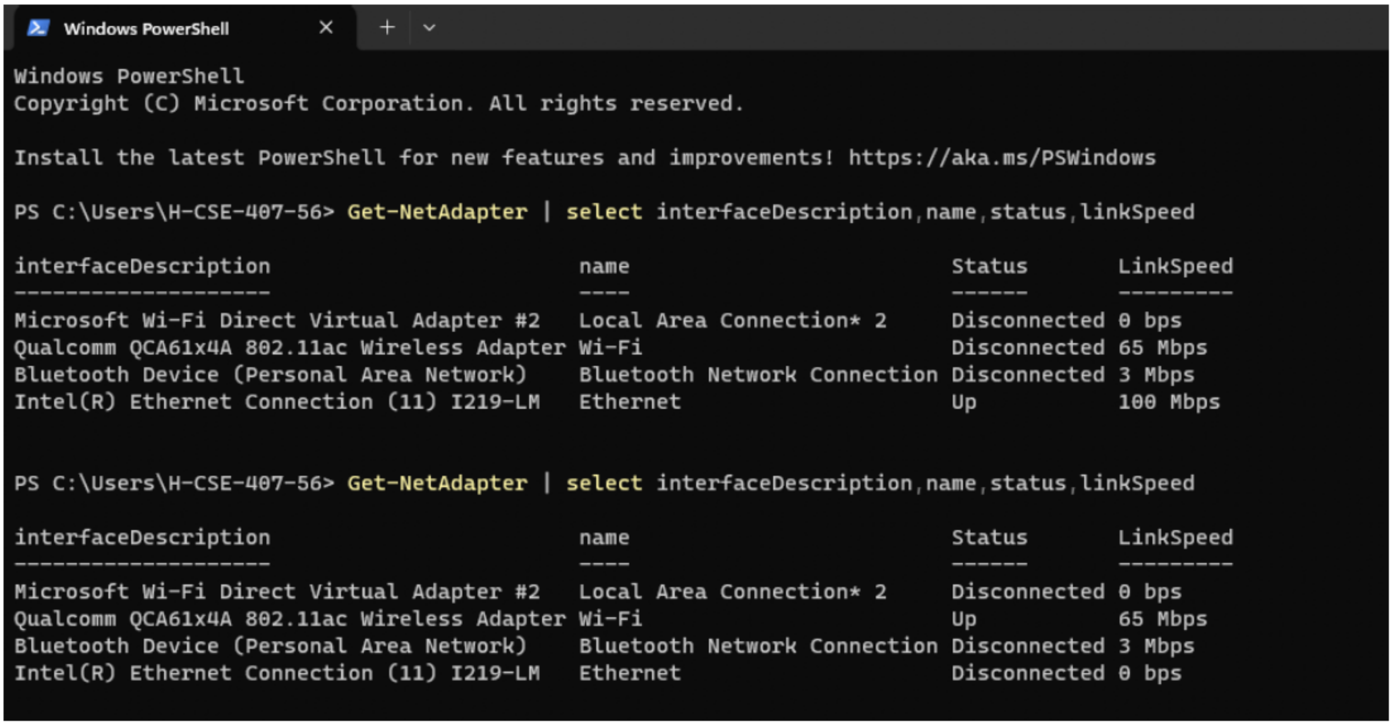
&

"Get-NetAdapter | select interfaceDescription,name,status,linkSpeed"

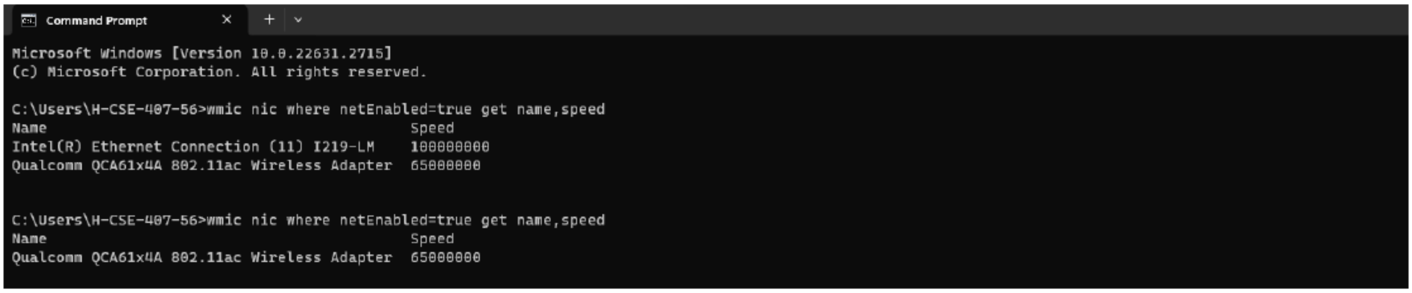
On Windows Powershell & Command Prompt Respectively.

Speed With Wired LAN : 100MBPS

Speed with Wireless LAN : 65 MBPS



```
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\H-CSE-407-56> Get-NetAdapter | select interfaceDescription,name,status,linkSpeed

interfaceDescription                        name                         Status       LinkSpeed
--------------------                        ----                         ------       ---------
Microsoft Wi-Fi Direct Virtual Adapter #2   Local Area Connection* 2     Disconnected 0 bps
Qualcomm QCA61x4A 802.11ac Wireless Adapter Wi-Fi                        Disconnected 65 Mbps
Bluetooth Device (Personal Area Network)    Bluetooth Network Connection Disconnected 3 Mbps
Intel(R) Ethernet Connection (11) I219-LM   Ethernet                     Up           100 Mbps


PS C:\Users\H-CSE-407-56> Get-NetAdapter | select interfaceDescription,name,status,linkSpeed

interfaceDescription                        name                         Status       LinkSpeed
--------------------                        ----                         ------       ---------
Microsoft Wi-Fi Direct Virtual Adapter #2   Local Area Connection* 2     Disconnected 0 bps
Qualcomm QCA61x4A 802.11ac Wireless Adapter Wi-Fi                        Up           65 Mbps
Bluetooth Device (Personal Area Network)    Bluetooth Network Connection Disconnected 3 Mbps
Intel(R) Ethernet Connection (11) I219-LM   Ethernet                     Disconnected 0 bps
```



```
Microsoft Windows [Version 10.0.22631.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\H-CSE-407-56>wmic nic where netEnabled=true get name,speed
Name                                         Speed
Intel(R) Ethernet Connection (11) I219-LM    100000000
Qualcomm QCA61x4A 802.11ac Wireless Adapter  65000000


C:\Users\H-CSE-407-56>wmic nic where netEnabled=true get name,speed
Name                                         Speed
Qualcomm QCA61x4A 802.11ac Wireless Adapter  65000000
```

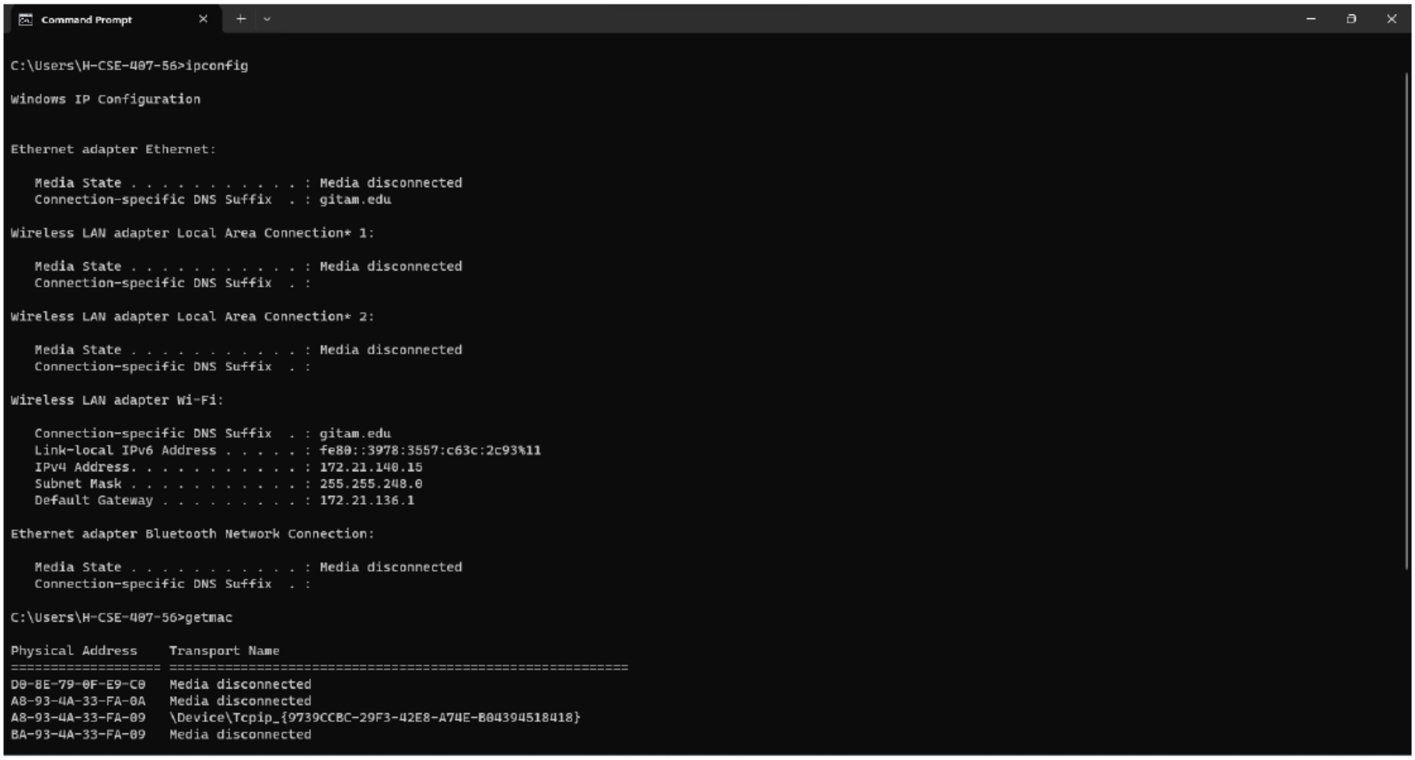
MAC and IP addresses and the subnet mask of my computer :

I've Used the commands “ipconfig” & “getmac” in Command Prompt to find the IP Address, Subnet Mask & mac address of my Computer.

IPv4 Address : 172.21.140.15

Subnet Mask : 255.255.248.0

MAC Address : A8-93-4A-33-FA-09



```
C:\Users\H-CSE-407-56>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : gitam.edu

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : gitam.edu
   Link-local IPv6 Address . . . . . : fe80::3978:3557:c63c:2c93%11
   IPv4 Address. . . . . . . . . . . : 172.21.140.15
   Subnet Mask . . . . . . . . . . . : 255.255.248.0
   Default Gateway . . . . . . . . . : 172.21.136.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\H-CSE-407-56>getmac

Physical Address    Transport Name
=================== =======================================================
D0-8E-79-0F-E9-C0   Media disconnected
A8-93-4A-33-FA-0A   Media disconnected
A8-93-4A-33-FA-09   \Device\Tcpip_{9739CCBC-29F3-42E8-A74E-B04394518418}
BA-93-4A-33-FA-09   Media disconnected
```

Computer Networks Lab

M.Sai Dinesh

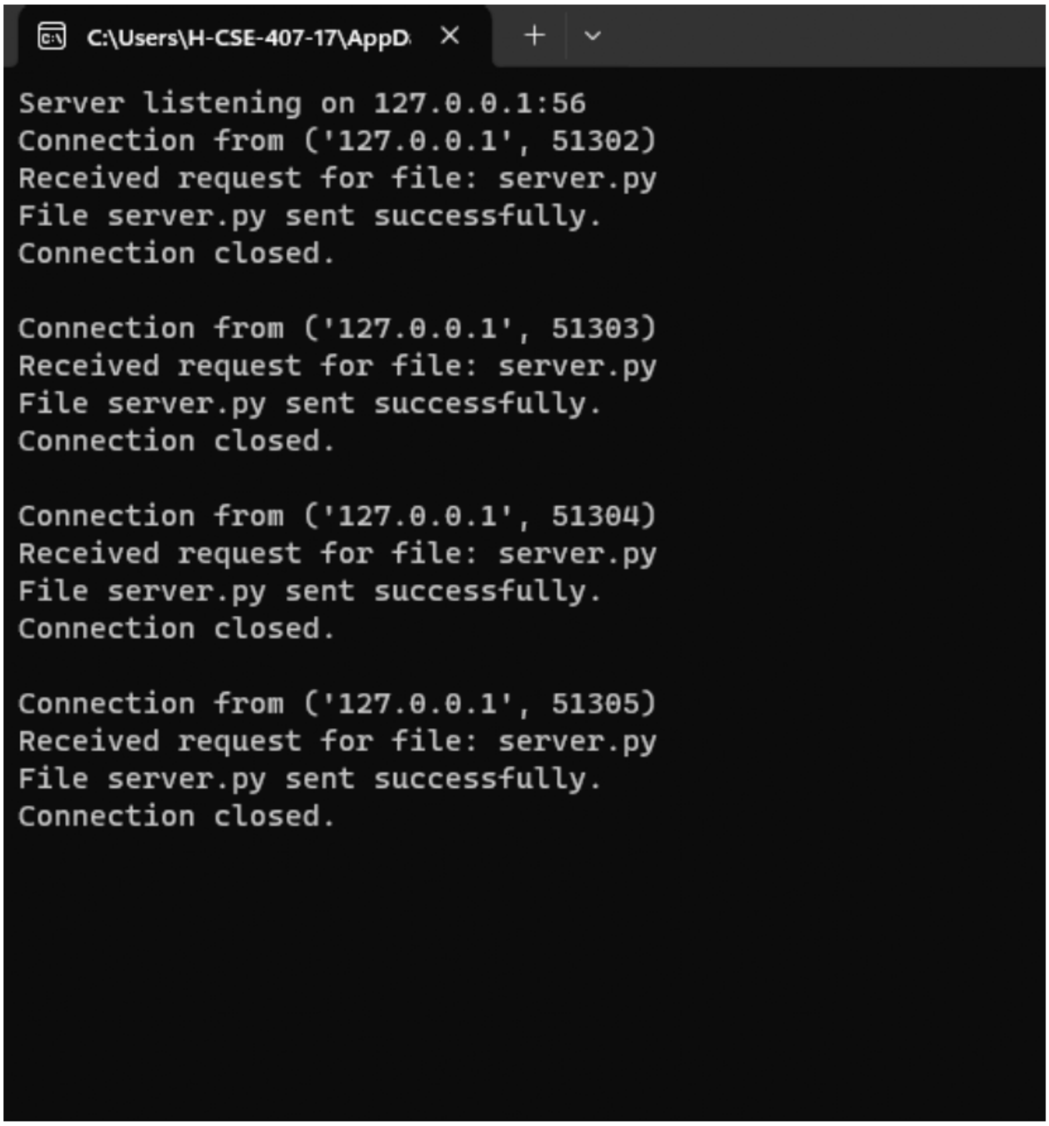
HU22CSEN0300193

Server program –

```
import socket

import os


def send_file(conn, filename):

    with open(filename, 'rb') as file:

        data = file.read(1024)

        while data:

            conn.send(data)

            data = file.read(1024)


def server():

    host = '127.0.0.1'

    port = 56


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

```
server_socket.bind((host, port))
server_socket.listen(5)

print(f"Server listening on {host}:{port}")

while True:
    conn, addr = server_socket.accept()
    print(f"Connection from {addr}")

    try:
        file_request = conn.recv(1024).decode('utf-8')
        print(f"Received request for file: {file_request}")

        if os.path.exists(file_request) and os.path.isfile(file_request):
            conn.send("OK".encode('utf-8'))
            send_file(conn, file_request)
            print(f"File {file_request} sent successfully.")
        else:
            conn.send("File not found.".encode('utf-8'))
            print("File not found.")

    except Exception as e:
        print(f"Error: {e}")
        conn.close()
        print("Connection closed.\n")

if __name__ == "__main__":
    server()
```

Server output –

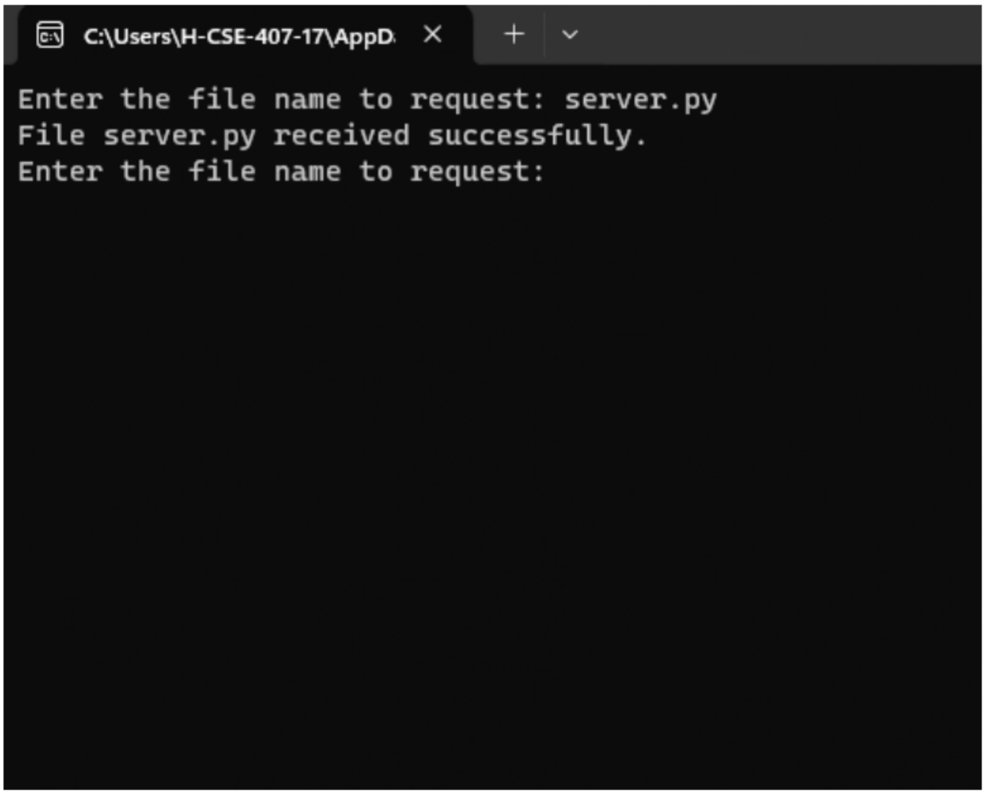


```
Server listening on 127.0.0.1:56
Connection from ('127.0.0.1', 51302)
Received request for file: server.py
File server.py sent successfully.
Connection closed.

Connection from ('127.0.0.1', 51303)
Received request for file: server.py
File server.py sent successfully.
Connection closed.

Connection from ('127.0.0.1', 51304)
Received request for file: server.py
File server.py sent successfully.
Connection closed.

Connection from ('127.0.0.1', 51305)
Received request for file: server.py
File server.py sent successfully.
Connection closed.
```

Client program –

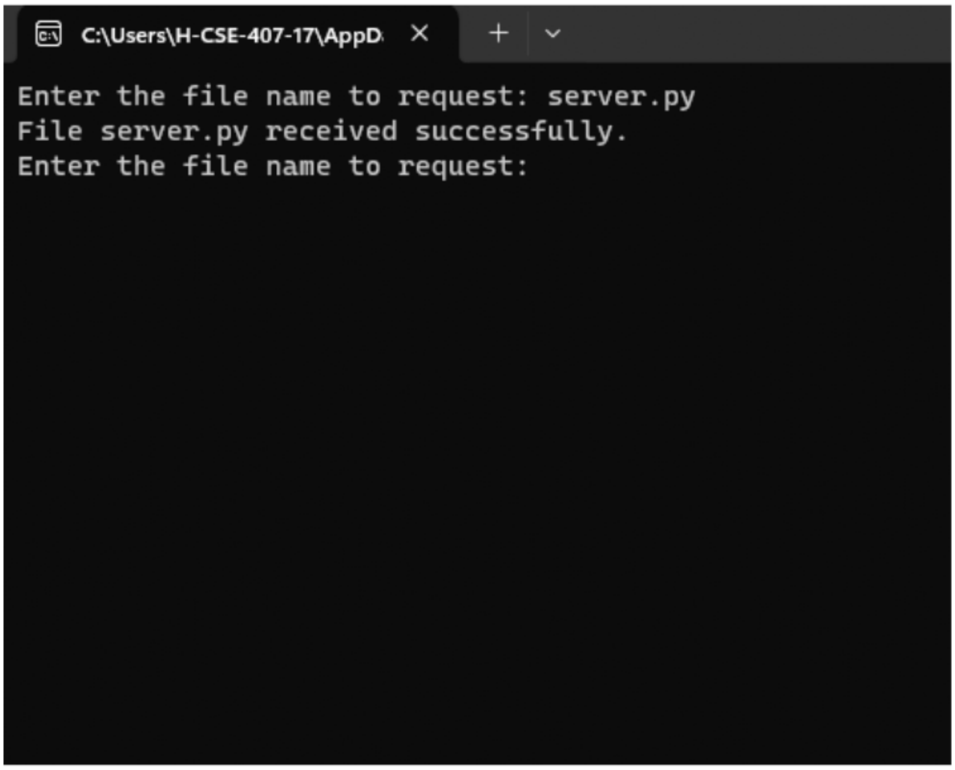
```
import socket


def request_file(filename):
    host = '127.0.0.1'
    port = 56


    client_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
    client_socket.connect((host, port))


    try:
        client_socket.send(filename.encode('utf-8'))
        response = client_socket.recv(1024).decode('utf-8')


        if response == "OK":
            with open(f"received_{filename}", 'wb') as file:
                data = client_socket.recv(1024)
                while data:
                    file.write(data)
                    data = client_socket.recv(1024)
            print(f"File {filename} received successfully.")
        else:
            print(response)
    except Exception as e:
        print(f"Error: {e}")

    finally:
        client_socket.close()

if __name__ == "__main__":
    for _ in range(3):  # Run three clients as an example
        file_to_request = input("Enter the file name to request: ")
        request_file(file_to_request)
```

Client 1 output –



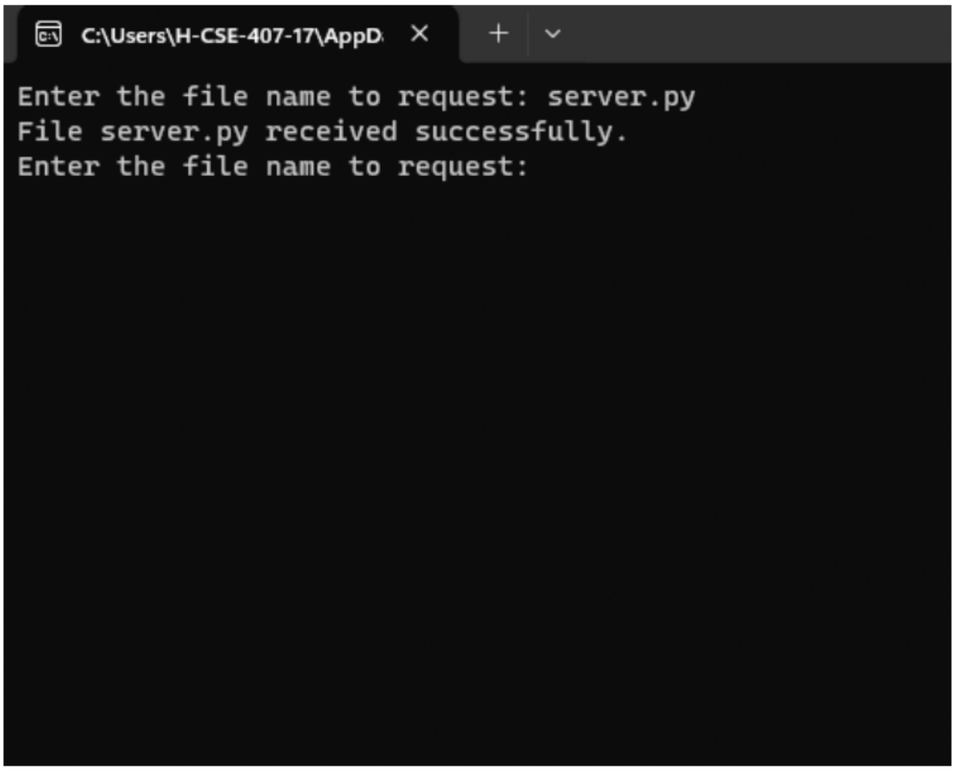
```
Enter the file name to request: server.py
File server.py received successfully.
Enter the file name to request:
```

Client 2 output –



```
Enter the file name to request: server.py
File server.py received successfully.
Enter the file name to request:
```

Client 3 output –



```
Enter the file name to request: server.py
File server.py received successfully.
Enter the file name to request:
```