Name: Prabir Tarafdar

Roll No.: 35000118051

Dept.: CSE 5th Sem

Subject: Computer Networks Lab

Sub Code: PCC-CS692

1. **A Client connected with the Server**

server.py

```
import socket               # Import socket module

s = socket.socket()         # Create a socket object
host = '192.168.0.7'  # Get local machine name
port = 6000              # Reserve a port for your service.
s.bind((host, port))        # Bind to the port

s.listen(5)                 # Now wait for client connection.
print(s)
print('server is ready')
while True:
    c, addr = s.accept()     # Establish connection with client.
    print('Got connection from', addr)
    c.send(b'Thank you for connecting')
    c.close()                # Close the connection
    break
s.close()
```

client.py

```
import socket               # Import socket module

s = socket.socket()         # Create a socket object
host = '192.168.0.7'  # Get local machine name
port = 6000                # Reserve a port for your service.

s.connect((host, port))
print(s.recv(1024))
s.close()                     # Close the socket when done
```

Input/ Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\PRABIR\All_my_works\CN lab\CN_Lab_Day_2 code\1_Day_2\Sample_01_Simple_client_server> python server.py
<socket.socket fd=300, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.0.7', 6000)>
server is ready
Got connection from ('192.168.0.7', 58686)
PS C:\Users\PRABIR\All_my_works\CN lab\CN_Lab_Day_2 code\1_Day_2\Sample_01_Simple_client_server>
```

```
PS C:\Users\PRABIR\All_my_works\CN lab\CN_Lab_Day_2 code\1_Day_2\Sample_01_Simple_client_server> python client.py
b'Thank you for connecting'
PS C:\Users\PRABIR\All_my_works\CN lab\CN_Lab_Day_2 code\1_Day_2\Sample_01_Simple_client_server>
```

2.Simple "Hello World" message received through server by the client:

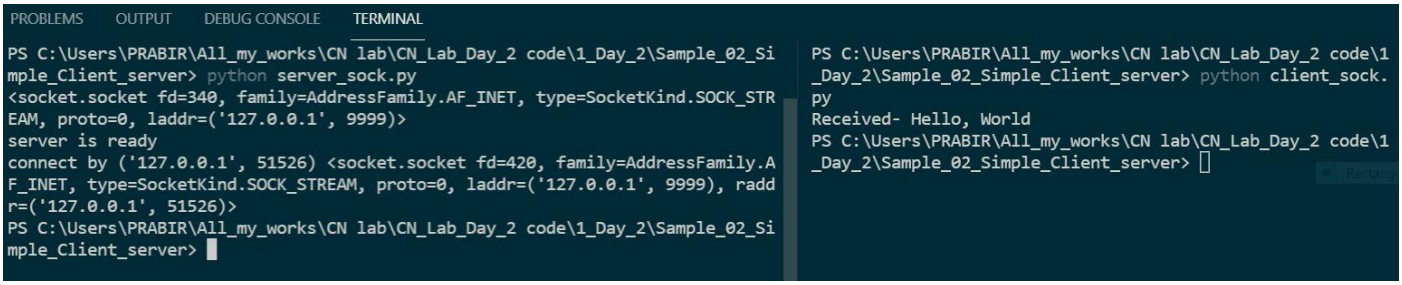
Server_sock.py

```
import socket

HOST = 'localhost'
PORT = 9999
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(1)
print(s)
print('server is ready')
conn, addr = s.accept()
print('connect by', addr, conn)
while 1:
    data = conn.recv(12)
    if not data:
        break
    conn.send(data)
conn.close()
```

client_sock.py

```
import socket

HOST = 'localhost'
PORT = 9999
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
s.send(b'Hello, World')
data = s.recv(12)
s.close()
# print 'Received', repr(data)
print('Received-', data.decode())
```

Input/Output:



3. A client received the server current running time, date, day:

sever.py

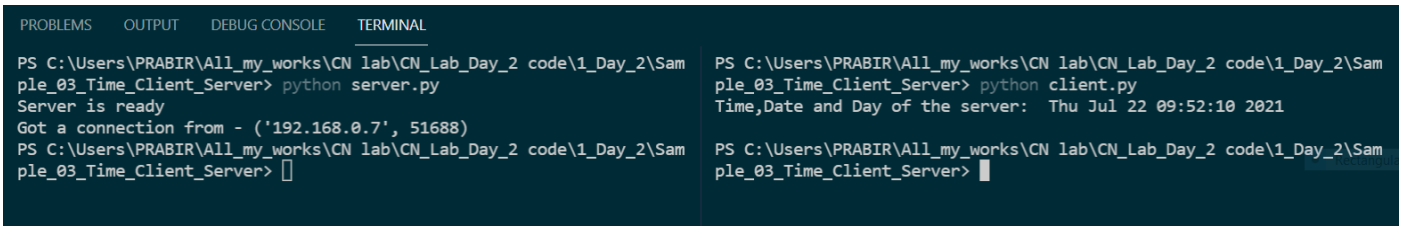
```
import socket
import time

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

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

port = 8000

# bind to the port
serversocket.bind((host, port))

# queue up to 5 requests
serversocket.listen(5)
print('Server is ready')
while True:
    # establish a connection
    clientsocket, addr = serversocket.accept()

    print("Got a connection from - %s" % str(addr))
    currentTime = time.ctime(time.time()) + "\r\n"
    clientsocket.send(currentTime.encode('ascii'))
    clientsocket.close()
    break
serversocket.close()
```

client.py

```
import socket

# create a socket object
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# get local machine name
host = socket.gethostname()
port = 8000

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

# Receive no more than 1024 bytes
tm = s.recv(1024)
```

```
print("Time,Date and Day of the server: ", tm.decode())
s.close()
```

Input/Output:



4. Addition of three numbers by server and send It to client:

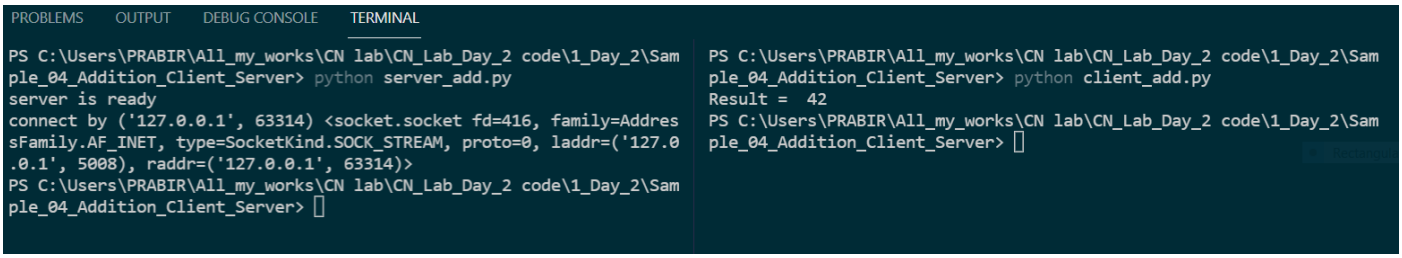
Server_add.py

```
import socket

HOST = 'localhost'
PORT = 5008
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(3)
# print s
print('server is ready')
conn, addr = s.accept()
print('connect by', addr, conn)
while 1:
    data = conn.recv(1024)

    a, b, c = str(bytes(data), 'utf-8').split('+', 3)

    ans = int(a) + int(b) + int(c)

    conn.send(bytes(str(ans), 'utf-8'))
    conn.close()
    break
s.close()
```

client_add.py

```
import socket

HOST = 'localhost'
PORT = 5008
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
s.send(bytes('20+12+10', 'utf-8'))
data = s.recv(1024)
print("Result = ", int(data))
```

Input/ Output:



5. Subtraction of three numbers by server and send it to client:

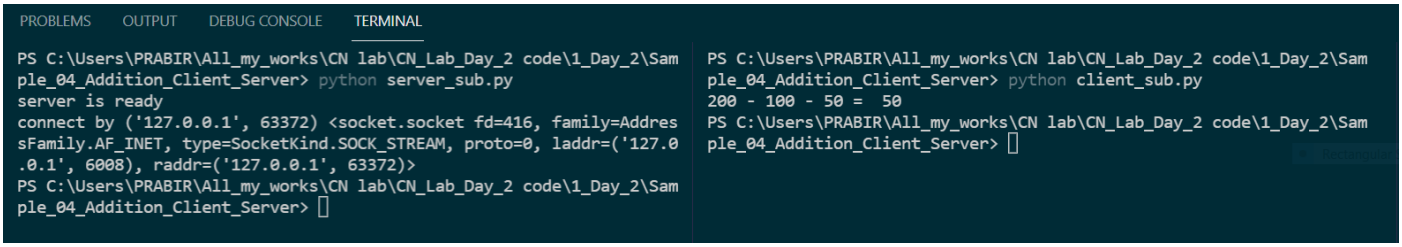
server_sub.py

```
import socket
HOST = 'localhost'
PORT = 6008
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(3)
print('server is ready')
conn, addr = s.accept()
print('connect by', addr, conn)
while 1:
    data = conn.recv(1024)
    a, b, c = str(bytes(data), 'utf-8').split('-', 3)
    ans = int(a) - int(b) - int(c)
    conn.send(bytes(str(ans), 'utf-8'))
    conn.close()
    break
s.close()
```

client_sub.py

```
import socket
HOST = 'localhost'
PORT = 6008
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
s.send(bytes('200-100-50', 'utf-8'))
data = s.recv(1024)
print("200 - 100 - 50 = ", int(data))
```

Input/Output:



6. Multiplication of three numbers by server and send it to client:

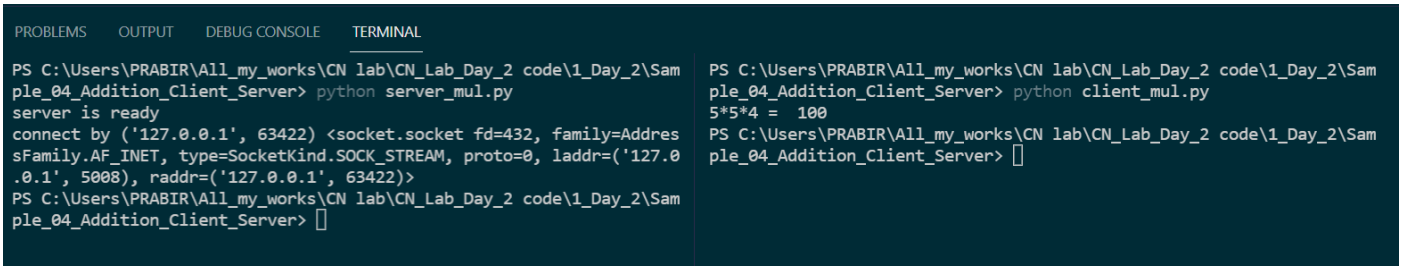
server_mul.py

```python
import socket
HOST = 'localhost'
PORT = 5008
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(3)
# print s
print('server is ready')
conn, addr = s.accept()
print('connect by', addr, conn)
while 1:
    data = conn.recv(1024)

    a, b, c = str(bytes(data), 'utf-8').split('*', 3)

    ans = int(a) * int(b) * int(c)

    conn.send(bytes(str(ans), 'utf-8'))
    conn.close()
    break
s.close()
```

client_mul.py

```python
import socket

HOST = 'localhost'
PORT = 5008
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
s.send(bytes('5*5*4', 'utf-8'))
data = s.recv(1024)
print("5*5*4 = ", int(data))
```

Input/Output



7. Division of two numbers by server and send it to client:

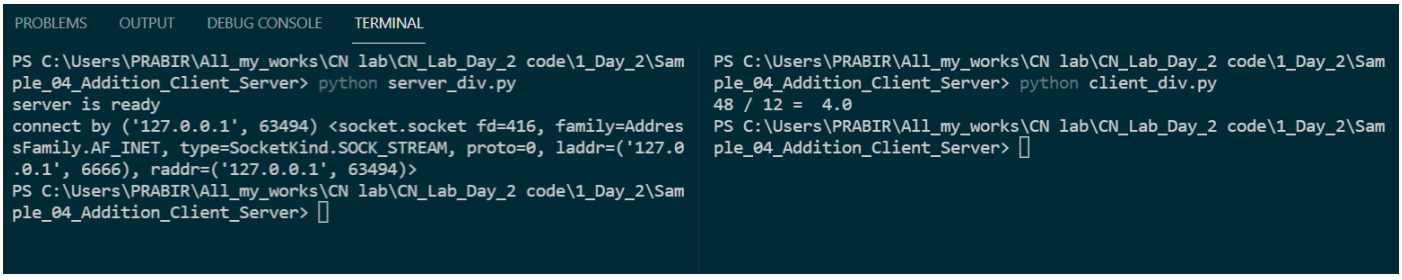
server_div.py

```python
import socket
HOST = 'localhost'
PORT = 6666
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(3)
# print s
print('server is ready')
conn, addr = s.accept()
print('connect by', addr, conn)
while 1:
    data = conn.recv(1024)

    a, b = str(bytes(data), 'utf-8').split('/', 3)

    ans = int(a) / int(b)

    conn.send(bytes(str(ans), 'utf-8'))
    conn.close()
    break
s.close()
```

client_div.py

```python
import socket
HOST = 'localhost'
PORT = 6666
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
s.send('48/12'.encode())
data = s.recv(1024)
print("48 / 12 = ", data.decode())
```

Input/Output



8. client connect with the server and enter his/her name and get a connecting wish from server.

server.py

```python
import socket
host = '192.168.0.7'
port = 9999
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
s.bind((host, port))
s.listen(5)
# print(s)
print('server is ready')

while True:
    c, addr = s.accept()
    name = c.recv(1024).decode()
    print("Connected With", addr, name)
    c.send(bytes(f'Thank you for connecting {name}', 'utf-8'))
    c.close()
s.close()
```

client.py

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
host = '192.168.0.7'
port = 9999
s.connect((host, port))
name = input("Enter your name: ")
s.send(bytes(name, 'utf-8'))
print(s.recv(1024).decode())
s.close()
```

Input/Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\PRABIR\All_my_works\CN lab\CN_Lab_Day_2 code\Lab_D3_mywo
rk> python server.py
server is ready
Connected With ('192.168.0.7', 50990) Prabir
Connected With ('192.168.0.7', 50991) Rahul
Connected With ('192.168.0.7', 50992) Pranto
```

```
PS C:\Users\PRABIR\All_my_works\CN lab\CN_Lab_Day_2 code\Lab_D3_mywo
rk> python client.py
Enter your name: Prabir
Thank you for connecting Prabir
PS C:\Users\PRABIR\All_my_works\CN lab\CN_Lab_Day_2 code\Lab_D3_mywo
rk> python client.py
Enter your name: Rahul
Thank you for connecting Rahul
PS C:\Users\PRABIR\All_my_works\CN lab\CN_Lab_Day_2 code\Lab_D3_mywo
rk> python client.py
Enter your name: Pranto
Thank you for connecting Pranto
PS C:\Users\PRABIR\All_my_works\CN lab\CN_Lab_Day_2 code\Lab_D3_mywo
rk>
```

9. client server calculator:

server_calculator.py

```
# @ Name: Prabir Tarafdar
# @ Date: 25/05/2021
import socket
HOST = 'localhost'
PORT = 9999
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(5)
# print s
print('server is ready')
conn, addr = s.accept()
```

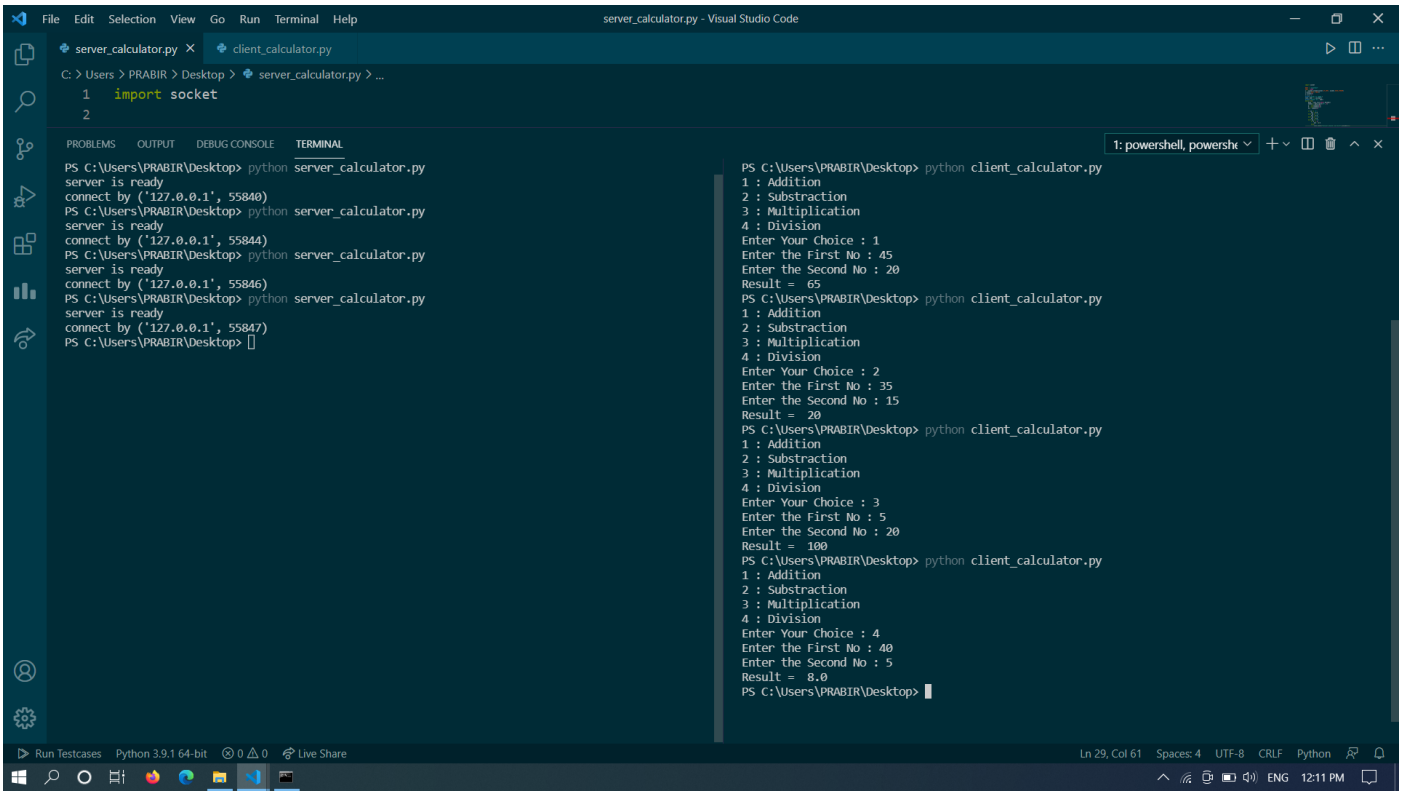
```
print('connect by', addr)
while True:
    data = conn.recv(1024).decode()
    tokens = data.split('#', 3)
    a = int(tokens[0])
    b = int(tokens[1])
    op = tokens[2]

    if op == '1':
        ans = a+b
    elif op == '2':
        ans = a-b
    elif op == '3':
        ans = a*b
    elif op == '4':
        if b != 0:
            ans = a/b
        else:
            conn.send(("Divisor should  not be 0").encode())
    conn.send(str(ans).encode())
    conn.close()
    break
s.close()
```

client_calculator.py

```
# @ Name: Prabir Tarafdar
# @ Date: 25/05/2021
def menuBar():
    Dict = {
        1: 'Addition',
        2: 'Substraction',
        3: 'Multiplication',
        4: 'Division'
    }
    for i in Dict:
        print(f"{i} : {Dict.get(i)}")
    k = input("Enter Your Choice : ")
    return k


if __name__ == "__main__":

    import socket

    HOST = 'localhost'
    PORT = 9999
```

```
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((HOST, PORT))

    op = menuBar()
    a = input('Enter the First No : ')
    b = input('Enter the Second No : ')

    c = a+'#'+b+'#'+op
    s.send(c.encode('ascii',))
    data = s.recv(1024)
    print('Result = ', data.decode())
    s.close()
```

Input/Output:



10. Download and Upload - client server program:

down_up_server.py

```
import socket
import os
HOST = 'localhost'
PORT = 5008
s = socket.socket()
s.bind(('', PORT))
s.listen(2)
print('FTP Server is ready....!!!')
conn, addr = s.accept()
print('connect by', addr)
itemList = os.listdir()
```

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

if x == "1":
    conn.send(str(itemList).encode())  # send present file list

    keyfile = conn.recv(1024).decode()  # receve the expected file name
    if(keyfile in itemList):
        while (1):
            f = open(keyfile, "rb")
            l = f.read(1024)
            while(l):
                conn.send(l)
                l = f.read(1024)
                if len(l) == 0:
                    break
            print(f'{keyfile} - sent successfully..')
            f.close()
            break
        conn.close()
        s.close()
    else:
        conn.send("Not".encode())
        conn.close()
        s.close()
if x == '2':
    recvFileName = conn.recv(1024)

    f = open(recvFileName, "wb")
    l = conn.recv(1024)
    size = 0
    while(l):
        f.write(l)
        size += len(l)
        l = conn.recv(1024)
        if l == '':
            break
    print(f'file size={size/1024} KB  \n Receved successfully...')

    f.close()
    conn.close()
    s.close()

else:
    conn.close()
    s.close()
```

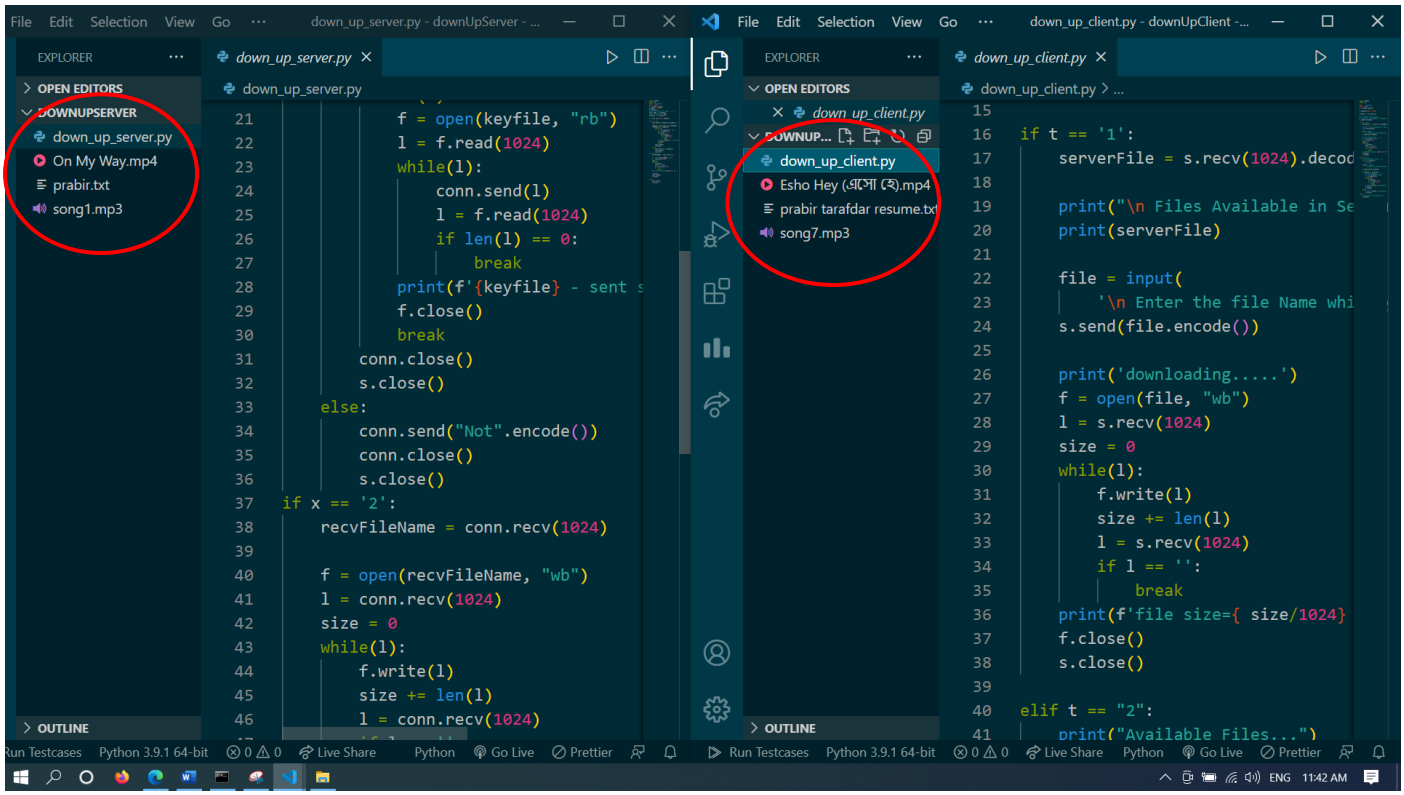
down_up_client.py

```
import socket
import os

HOST = '192.168.0.7'
PORT = 5008
s = socket.socket()

s.connect((HOST, PORT))

fileList = os.listdir()

print("Enter your choice")
t = input("1. Downloade\n2. Uploade\n")
s.send(t.encode())

if t == '1':
    serverFile = s.recv(1024).decode()

    print("\n Files Available in Server:\n")
    print(serverFile)

    file = input(
        '\n Enter the file Name whice you want to downloade (ex: abc.mp3)\n')
    s.send(file.encode())

    print('downloading.....')
    f = open(file, "wb")
    l = s.recv(1024)
    size = 0
    while(l):
        f.write(l)
        size += len(l)
        l = s.recv(1024)
        if l == '':
            break
    print(f'file size={ size/1024} KB  \n downloaded successfully...')
    f.close()
    s.close()

elif t == "2":
    print("Available Files...")
    for i in range(len(fileList)):
        if i > 0:
            print(fileList[i])
```

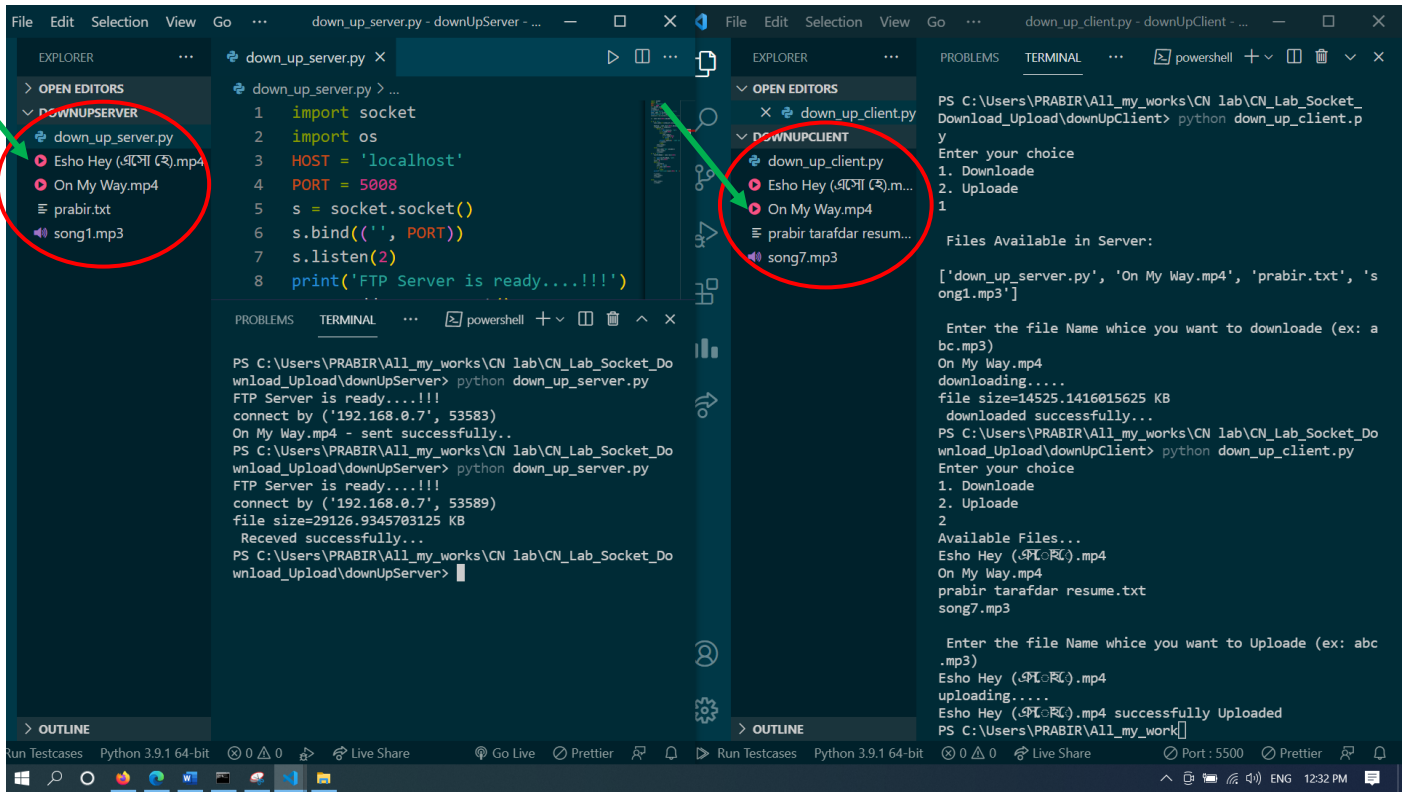
```
    infile = input(
        '\n Enter the file Name whice you want to Uploade (ex: abc.mp3)\n')

    s.send(infile.encode())

    if(infile in fileList):
        print('uploading.....')
        f = open(infile, "rb")
        l = f.read(1024)
        while(l):
            s.send(l)
            l = f.read(1024)
            if len(l) == 0:
                break
        print(f'{infile} successfully Uploaded')
        f.close()
        s.close()
    else:
        print(f"{infile} not present in your device!!")
        s.close()
else:
    print("Enter valid Option!!!")
    s.close()
```

Input/Output:

The file status **Before** doing anything.



The file status **After** download and upload.



11. Download using threade - client server program:

Multiple clients can download same item at a time.

Server.py

```
import socket
from threading import Thread
#from SocketServer import ThreadingMixIn

TCP_IP = '192.168.0.7'
TCP_PORT = 8888
BUFFER_SIZE = 1024


class ClientThread(Thread):

    def __init__(self, ip, port, sock):
        Thread.__init__(self)
        self.ip = ip
        self.port = port
        self.sock = sock
        print(" New thread started for "+ip+":"+str(port))

    def run(self):
        filename = 'Nill Digonte.mp3'
        f = open(filename, 'rb')
        while True:
```

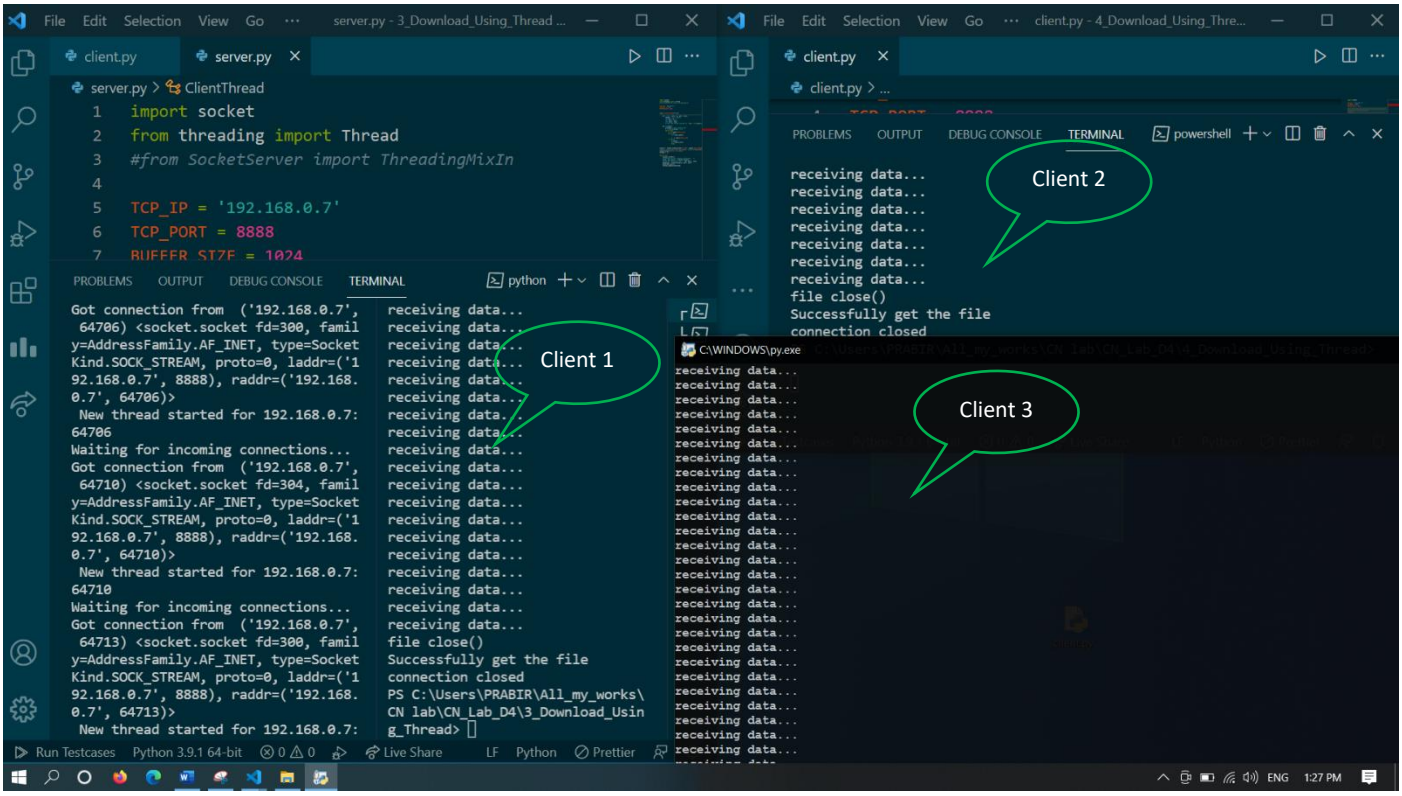
```
            l = f.read(BUFFER_SIZE)
            while (l):
                self.sock.send(l)

                l = f.read(BUFFER_SIZE)
            if not l:
                f.close()
                self.sock.close()
                break


tcpsock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
#tcpsock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
tcpsock.bind((TCP_IP, TCP_PORT))
threads = []

while True:
    tcpsock.listen(5)
    print("Waiting for incoming connections...")
    (conn, (ip, port)) = tcpsock.accept()
    print('Got connection from ', (ip, port), conn)
    newthread = ClientThread(ip, port, conn)
    newthread.start()
    threads.append(newthread)
```

client.py

```
import socket

TCP_IP = '192.168.0.7'
TCP_PORT = 8888
BUFFER_SIZE = 1024

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((TCP_IP, TCP_PORT))
size = 0
with open('Nill Digonte(download).mp3', 'wb') as f:
    print('file opened')
    while True:
        print('receiving data...')
        data = s.recv(BUFFER_SIZE)
        #print('data=%s', (data))

        if not data:
            f.close()
            print('file close()')
            break
```

```
        # write data to a file
        f.write(data)

print('Successfully get the file')
s.close()
print('connection closed')
```

output:



11. Simple chatroom between two clients through server.

Server.py

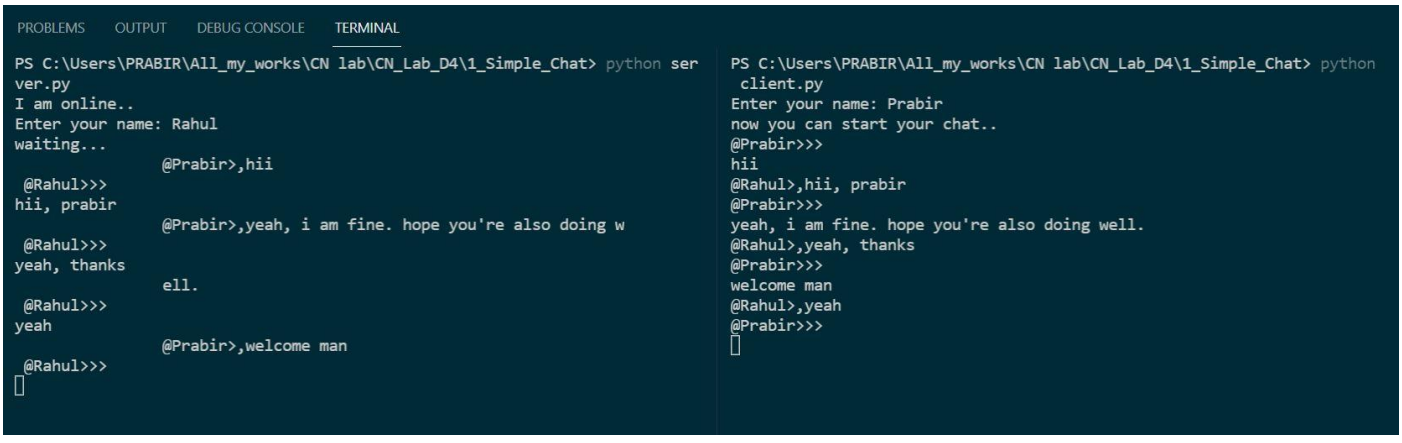
```
import socket
HOST = '192.168.0.7'
PORT = 5555
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(20)
print("I am online..")

name = input("Enter your name: ")
name = "@"+name+">"
print("waiting...")
conn, addr = s.accept()

while 1:
    data = conn.recv(50)
    print("\t\t"+data.decode())
    print(" "+name+">>")
    string = input()
```

```
    msg = name+","+string
    conn.send(msg.encode())
    if data == "quit":
        break
    if string == "quit":
        break

s.close()

conn.close()
```

client.py

```
import socket

HOST = '192.168.0.7'
PORT = 5555
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))

name = input("Enter your name: ")
name = "@"+name+">"
print("now you can start your chat..")
while 1:
    print(name+">>")
    string = input()
    msg = name+","+string
    s.send(msg.encode())
    data = s.recv(50)
    print(data.decode())
    if data == "quit":
        break
    if string == "quit":
        break

s.close()
```

Input/ Output:



12. A ChatRoom like WhatsApp group messaging (my project):

Multiple clients can connect with a server and chatting with each other.

Broadcasting Mechanism is used here.

Server.py

```
import socket
import threading

#host = socket.gethostbyname(socket.gethostname())
host = '192.168.0.7'
port = 5000

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((host, port))
server.listen()

clients = []
fullNames = []


def broadcast(message):
    for client in clients:
        client.send(message)


def handle(client):
    while True:
        try:
            message = client.recv(1024)
            broadcast(message)
        except:
            index = clients.index(client)
            clients.remove(client)
            client.close()
            name = fullNames[index]
            broadcast(f'{name} left the chat!'.encode('ascii'))
            fullNames.remove(name)
            break


def receive():
    while True:
        client, addr = server.accept()
        print(f"connected with--{str(addr)}")

        client.send('NICK'.encode('ascii'))
```

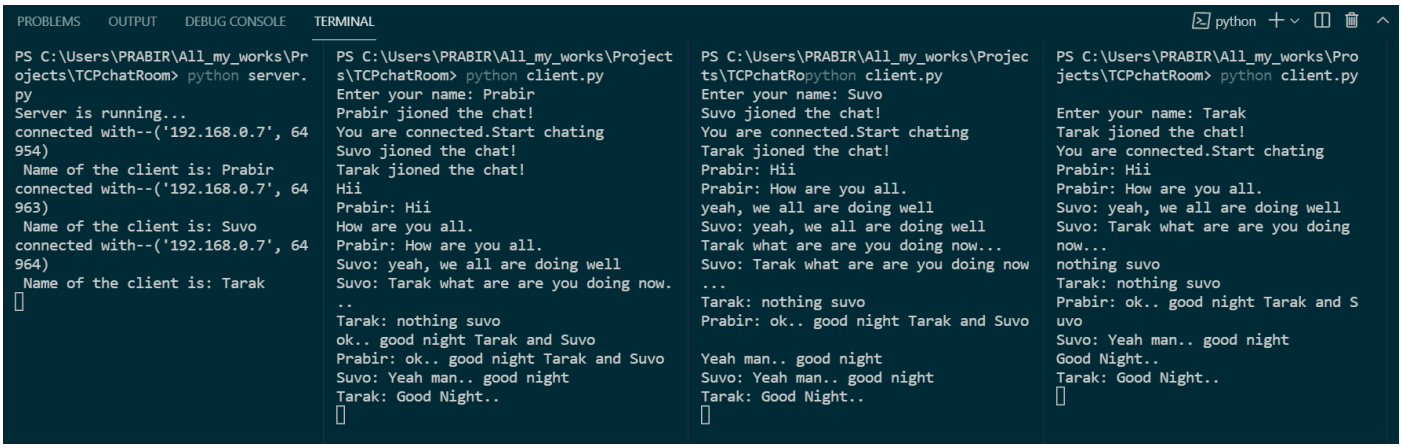
```
        name = client.recv(1024).decode('ascii')
        fullNames.append(name)
        clients.append(client)

        print(f" Name of the client is: {name}")
        broadcast(f"{name} jioned the chat!".encode('ascii'))
        client.send('You are connected.Start chating '.encode('ascii'))

        thread = threading.Thread(target=handle, args=(client,))
        thread.start()


print("Server is running...")
receive()
```

client.py

```
import socket
import threading

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect(('192.168.0.7', 5000))

name = input("Enter your name: ")


def receive():
    while True:
        try:
            message = client.recv(1024).decode('ascii')
            if message == 'NICK':
                client.send(name.encode('ascii'))
            else:
                print(message)
        except:
            print("[Error] Our team is working to fix it")
            client.close()
            break


def write():
    while True:
        message = f"{name}: {input()}"
        client.send(message.encode('ascii'))


receive_thread = threading.Thread(target=receive)
receive_thread.start()
```

```
write_thread = threading.Thread(target=write)
write_thread.start()
```

Output:



Thank You