Sumit Patel - 40

Experiment - 07

Server.py

```
import socket
import threading

# Server Configuration
PORT = 5050
SERVER = socket.gethostbyname(socket.gethostname())  ADDR = (SERVER, PORT)
FORMAT = 'utf-8'
HEADER = 64
DISCONNECT_MESSAGE = '!DISCONNECT'

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind(ADDR)

# Client Handling Function
def clientHandling(connection, address):
    print(f'[NEW CONNECTION] {address} connected')
    isConnected = True
    while isConnected:
        messageLength = connection.recv(HEADER).decode(FORMAT)
        if messageLength:
            messageLength = int(messageLength)
            message = connection.recv(messageLength).decode(FORMAT)
            if message == DISCONNECT_MESSAGE:
                isConnected = False
            print(f'[{address}] {message}')
            connection.send('Message Received'.encode(FORMAT))
    connection.close()

# Start Server
def start():
    server.listen()
    print(f'[LISTENING] Server is listening on {SERVER}')
    while True:
        connection, address = server.accept()
        thread = threading.Thread(target=clientHandling, args=(connection,
address))
```

```
        thread.start()
        print(f'[ACTIVE CONNECTIONS] {threading.active_count() - 1}')

print('[SERVER STARTING] Server has been started')
start()
```

client.py

```
import socket

# Client Configuration
PORT = 5050
SERVER = socket.gethostbyname(socket.gethostname())
ADDR = (SERVER, PORT)
FORMAT = 'utf-8'
HEADER = 64
DISCONNECT_MESSAGE = '!DISCONNECT'

# Create Client Socket
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect(ADDR)

# Function to Send Messages
def send(messages):
    message = messages.encode(FORMAT)
    messageLength = str(len(message)).encode(FORMAT)
    messageLength = messageLength.ljust(HEADER)
    client.send(messageLength)
    client.send(message)
    print(client.recv(HEADER).decode(FORMAT))

# Run Client Interaction
if __name__ == '__main__':
    while True:
        msg = input("Enter message (or type '!DISCONNECT' to exit): ")
        send(msg)
        if msg == DISCONNECT_MESSAGE:
            break
```

Output

On Server Side

```
[SERVER STARTING] Server has been started
[LISTENING] Server is listening on 192.168.0.104
[NEW CONNECTION] ('192.168.0.104', 53725) connected
[ACTIVE CONNECTIONS] 1
[('192.168.0.104', 53725)] Hello Python
[('192.168.0.104', 53725)] This is Experiment 07
[('192.168.0.104', 53725)] !Disconnet
[('192.168.0.104', 53725)] !DISCONNECT
```

On Client Side

```
Enter message (or type '!DISCONNECT' to exit): Hello Python
Message Received
Enter message (or type '!DISCONNECT' to exit): This is Experiment 07
Message Received
Enter message (or type '!DISCONNECT' to exit): !Disconnet
Message Received
Enter message (or type '!DISCONNECT' to exit): !DISCONNECT
Message Received
PS D:\ENGINEERING\Sem 4 - learning Hub\1.  Python Programs\EXP>
```