Assignment 3: Client Server System

Name: Prathamesh Agawane

MIS: 1422o03001

Batch: T4 – COMP-Div2

Build a simple client-server system, where you use the client to chat with a server. The protocol between the client and server is as follows.

* The server is first started on a known port.
* The client program is started (server IP and port are provided on the command line).
* The client connects to the server and then asks the user for input. The user types his message on the terminal. The user's input is sent to the server via the connected socket.
* The server reads the user's input from the client socket.
* The client then reads the reply from the server.
* The server must reply to the client.
* The client asks the user for the next message to send to the server.
* **server.py**

import socket

# Define the server IP address and port

serverIP = "192.168.0.103"  # Server's IP address

serverPort = 12345  # Choose a port number

# Create a socket object

serverSocket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

# Bind the socket to the server IP and port

serverSocket.bind((serverIP, serverPort))

# Listen for incoming connections (maximum 1 client in this example)

serverSocket.listen(1)

print(f"Server is listening on {serverIP}:{serverPort}")

# Accept a client connection

clientSocket, clientAddress = serverSocket.accept()

print(f"Accepted connection from {clientAddress}")

while True:

    # Receive data from the client

    clientMSG = clientSocket.recv(1024).decode('utf-8')

    # Check if the client wants to exit

    if clientMSG.lower() == "exit":

        print("Client has requested to exit the chat.")

        break

    # Print the received message

    print(f"Client: {clientMSG}")

    # Get a message from the server user and send it to the client

    serverMSG = input("Server: ")

    clientSocket.send(serverMSG.encode('utf-8'))

# Close the sockets

clientSocket.close()

serverSocket.close()

* **client.py**

import socket

import sys

# Check for correct command-line arguments

if len(sys.argv) != 3:

    print("Usage: python client.py <server\_ip> <server\_port>")

    sys.exit(1)

# Parse command-line arguments

serverIP = sys.argv[1]

serverPort = int(sys.argv[2])

# Create a socket object

clientSocket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

# Connect to the server

clientSocket.connect((serverIP, serverPort))

print(f"Connected to server {serverIP}:{serverPort}")

while True:

    # Get a message from the user

    userMSG = input("Client: ")

    # Check if the user wants to exit

    if userMSG.lower() == "exit":

        clientSocket.send("exit".encode('utf-8'))

        print("Exiting the chat.")

        break

    # Send the user's message to the server

    clientSocket.send(userMSG.encode('utf-8'))

    # Receive and print the server's reply

    serverMSG = clientSocket.recv(1024).decode('utf-8')

    print(f"Server: {serverMSG}")

# Close the socket

clientSocket.close()

* **OUTPUT:-**

