# Client and Admin Server Communication Module:

Steps and Procedures:

1. Create the server.py file to write the configuration for the Admin server
2. Create the client.py file to write the configuration for the client server
3. Run each file in a separate terminal.

Note: If you want to view both terminal at the same time to make it vertically parallel to each other you can either split the terminal or move each terminal to an editor view area in a different tab each.

1. Run the server.py file first and it should give this upon running successfully,

**“Server is listening on localhost:12345”.**

1. Then run the client.py file in a separate terminal and it should give this upon running successfully,

**“Connected to localhost:12345”.**

1. Communication between both the Admin and Client server can now commence.
2. If both parties are done with communication either party can simply type “Exit” to terminate the connection between both servers.

**Server.py Code File:**

import socket

import sys

*# Server configuration*

HOST = 'localhost'  *# Uses your server's IP for remote access*

PORT = 12345

def main():

*# Create a socket*

    server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

    server\_socket.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

*# Bind the socket to the host and port*

    server\_socket.bind((HOST, PORT))

*# Listen for incoming connections (max 1 client)*

    server\_socket.listen(1)

    print(f"Server is listening on {HOST}:{PORT}")

*# Accept a client connection*

    client\_socket, client\_address = server\_socket.accept()

    print(f"Connected by {client\_address}")

    try:

        while True:

*# Receive data from the client*

            data = client\_socket.recv(1024).decode('utf-8')

            if not data:

                print("Client disconnected")

                break

            print(f"\nClient: {data}")

*# Check for exit command*

            if data.strip().lower() == 'exit':

                print("Client requested to exit")

                break

*# Admin's response*

            response = input("Admin: ").strip()

            client\_socket.send(response.encode('utf-8'))

*# Check if admin wants to exit*

            if response.strip().lower() == 'exit':

                print("Admin exiting...")

                break

    finally:

*# Close the connections*

        client\_socket.close()

        server\_socket.close()

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Client.py Code File:**

import socket

*# Client configuration*

HOST = 'localhost'  *# Server's IP (same as server's HOST)*

PORT = 12345

def main():

*# Create a socket*

    client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

    try:

*# Connect to the server*

        client\_socket.connect((HOST, PORT))

        print(f"Connected to {HOST}:{PORT}")

        while True:

*# Send message to server*

            message = input("\nYou: ").strip()

            client\_socket.send(message.encode('utf-8'))

*# Check for exit command*

            if message.strip().lower() == 'exit':

                print("Exiting...")

                break

*# Receive response from server*

            response = client\_socket.recv(1024).decode('utf-8')

            print(f"\nAdmin: {response}")

*# Check if server exited*

            if response.strip().lower() == 'exit':

                print("Server disconnected")

                break

    except ConnectionRefusedError:

        print("Error: Connection refused. Is the server running?")

    finally:

        client\_socket.close()

if \_\_name\_\_ == "\_\_main\_\_":

    main()