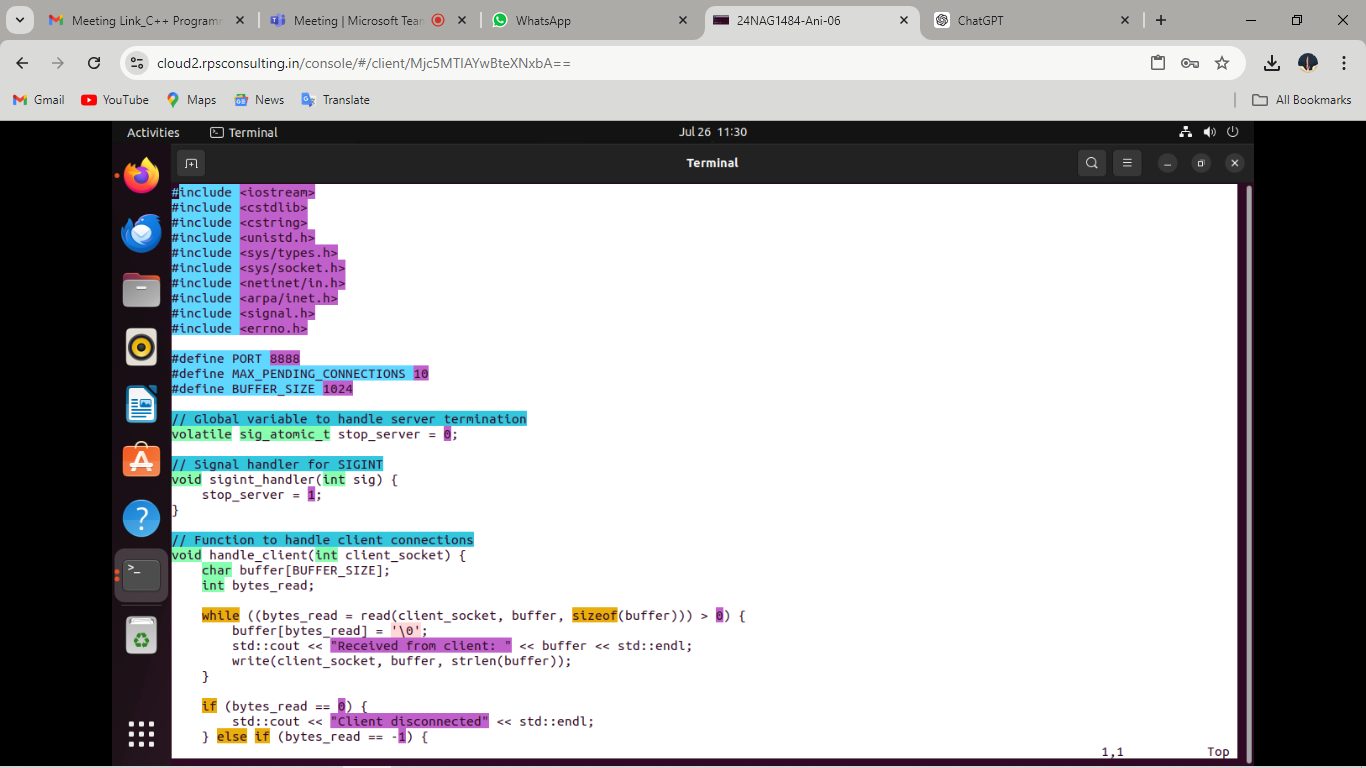
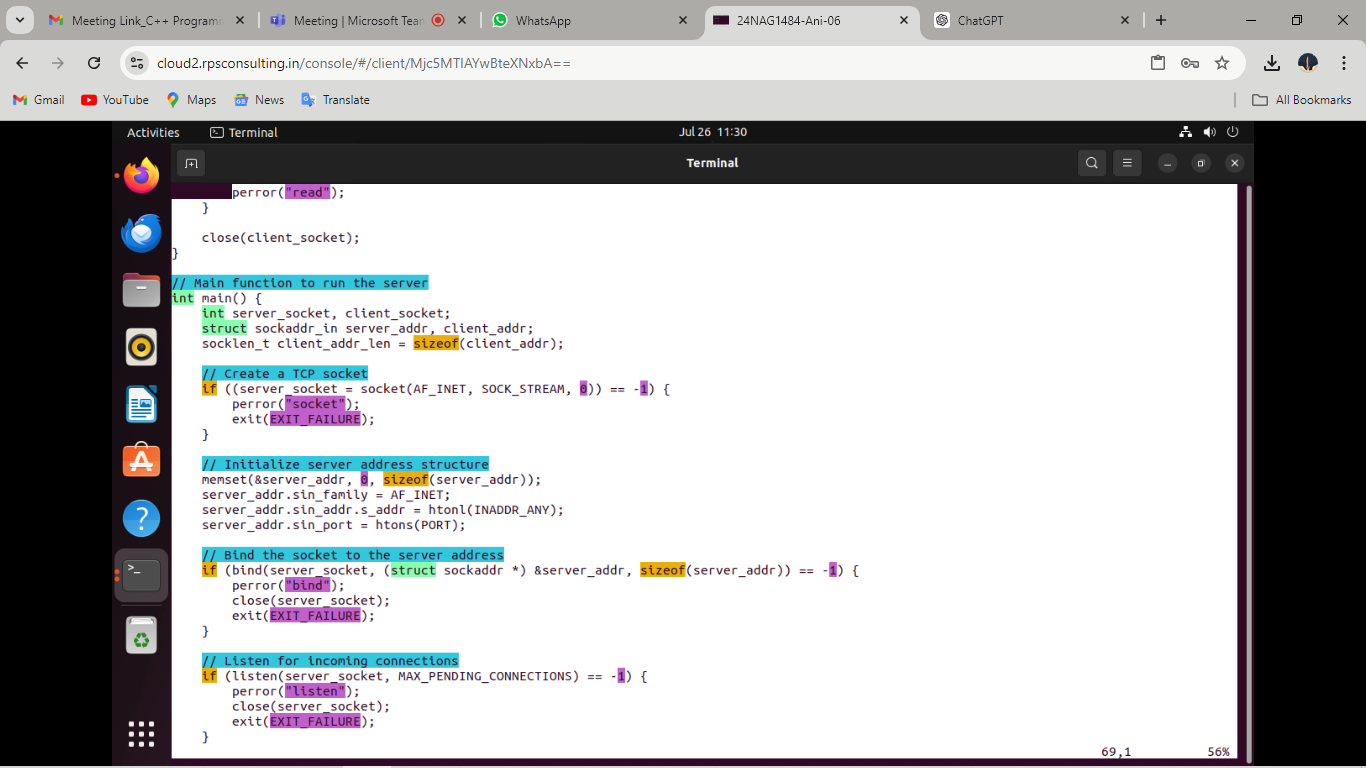
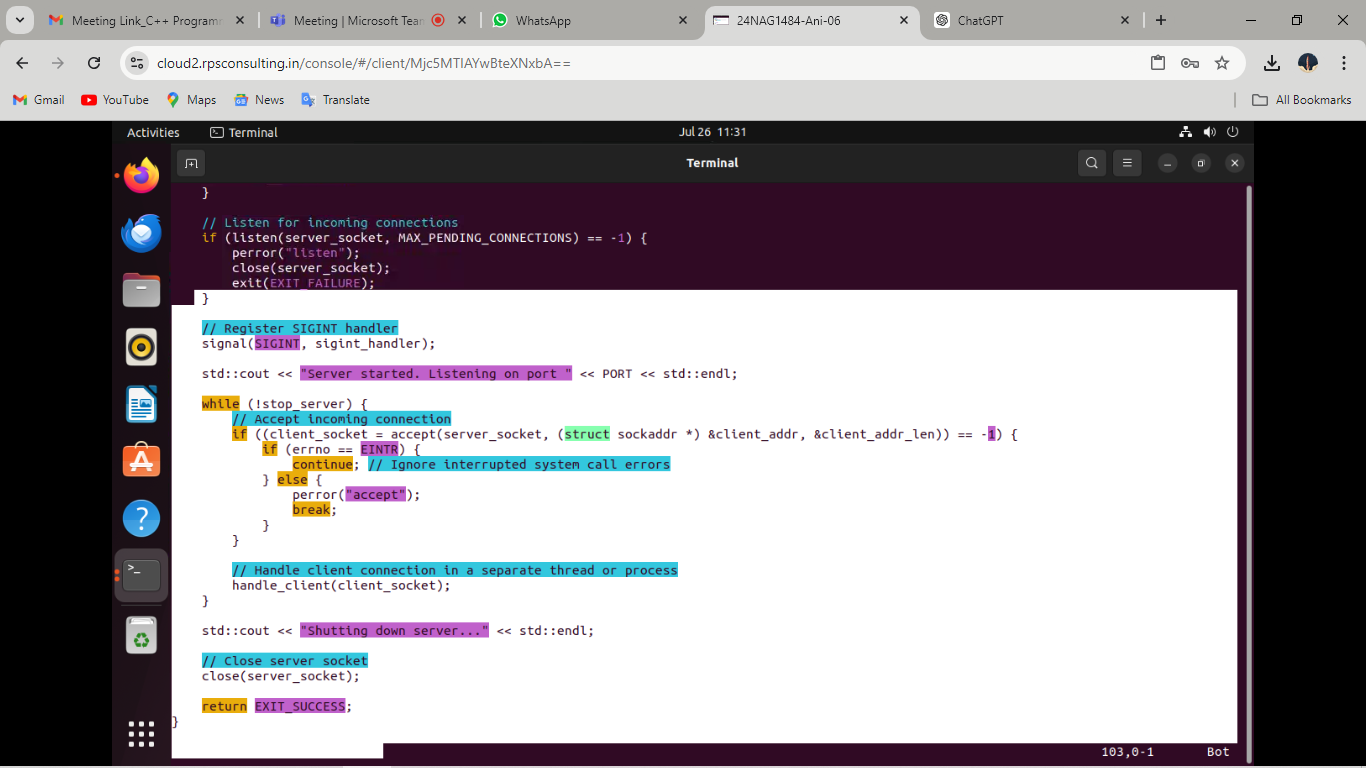
**Day – 7** LSP Assignment ( Task – 1)

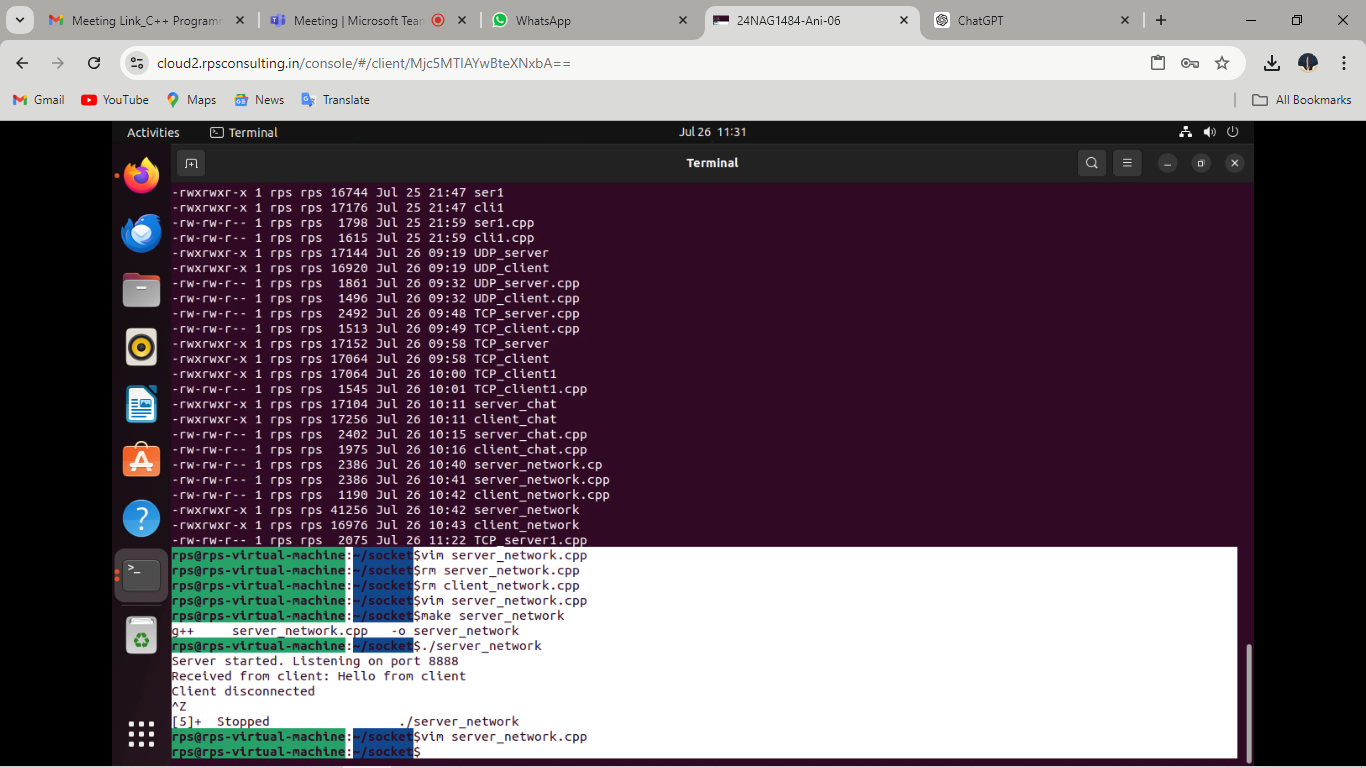
1. **Design and implement a network service that reliably handles concurrent client connections while ensuring graceful termination in response to external signals (e.g., SIGTERM, SIGINT). The service must maintain data consistency and avoid resource leaks throughout its lifecycle.**
2. **server – side**

****

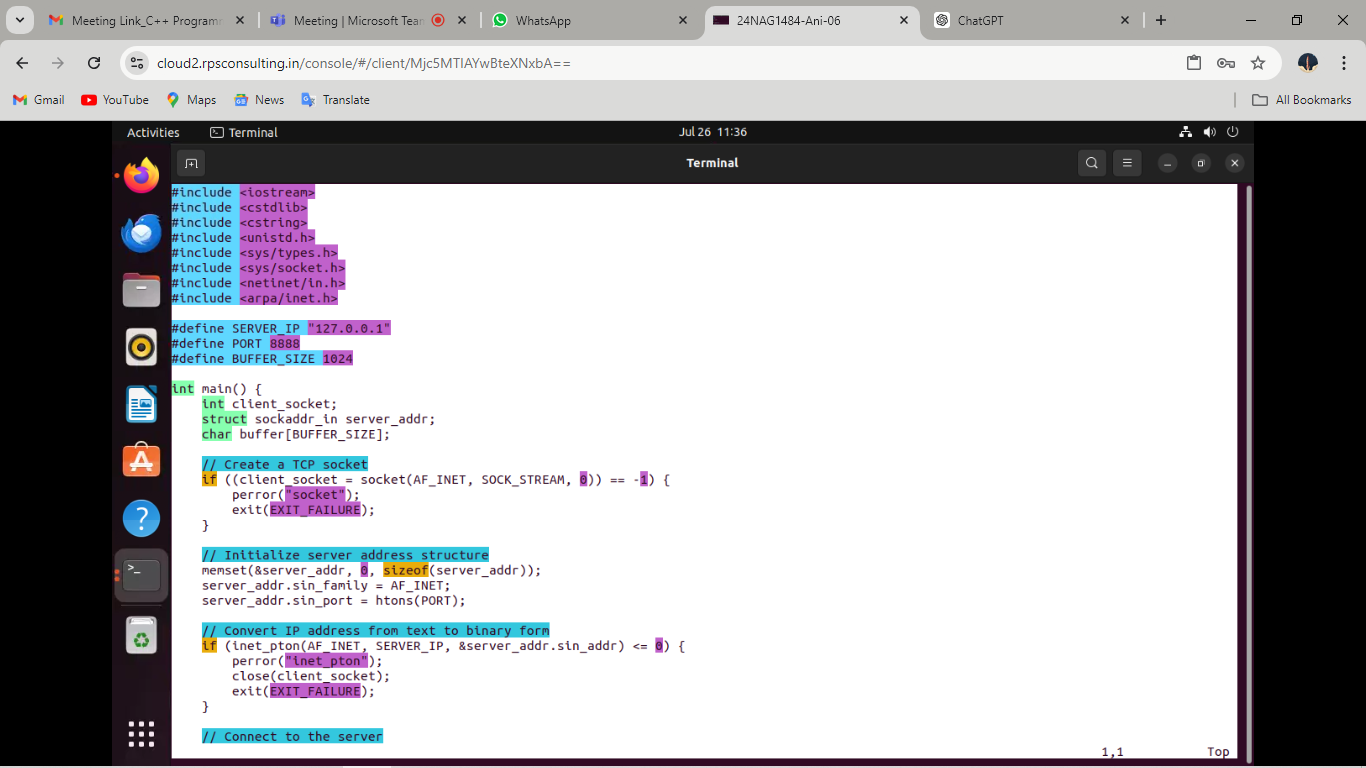
****

****

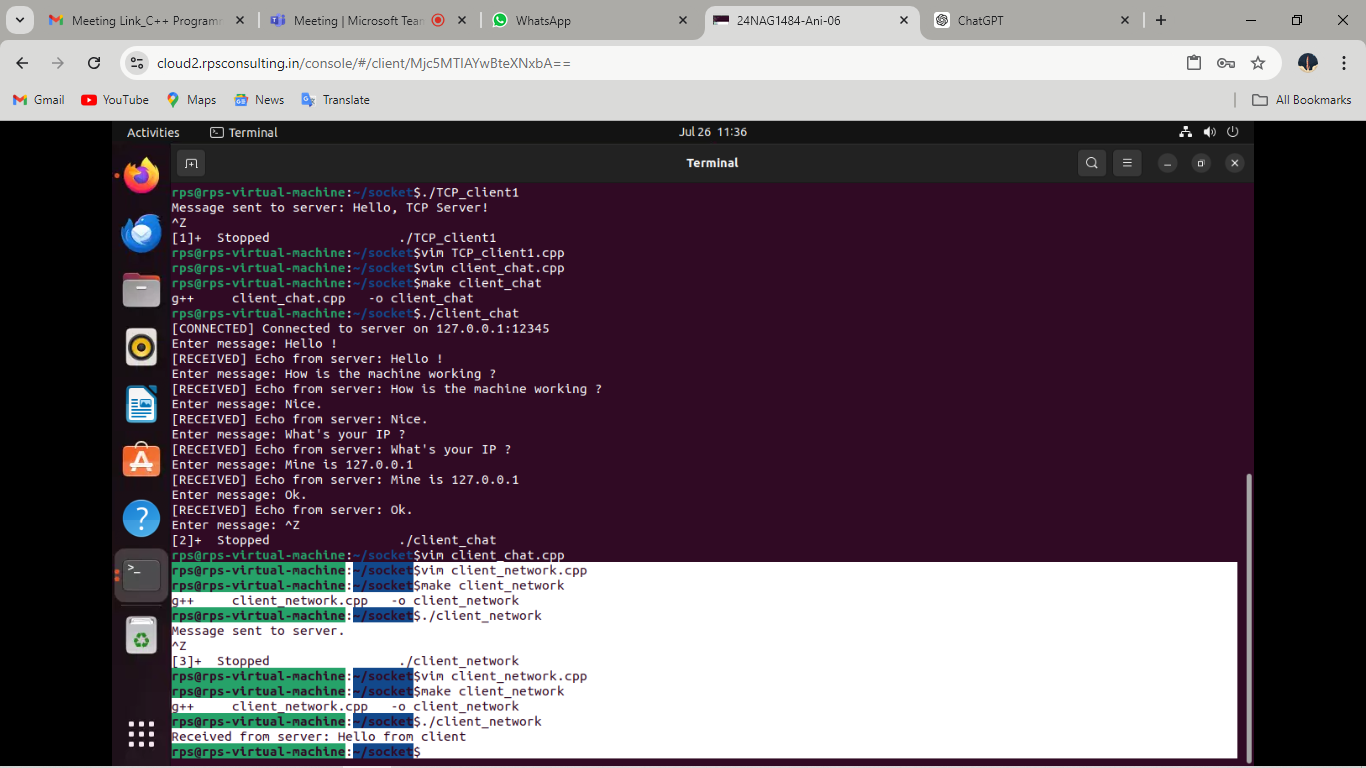
**Execution : server code**

****

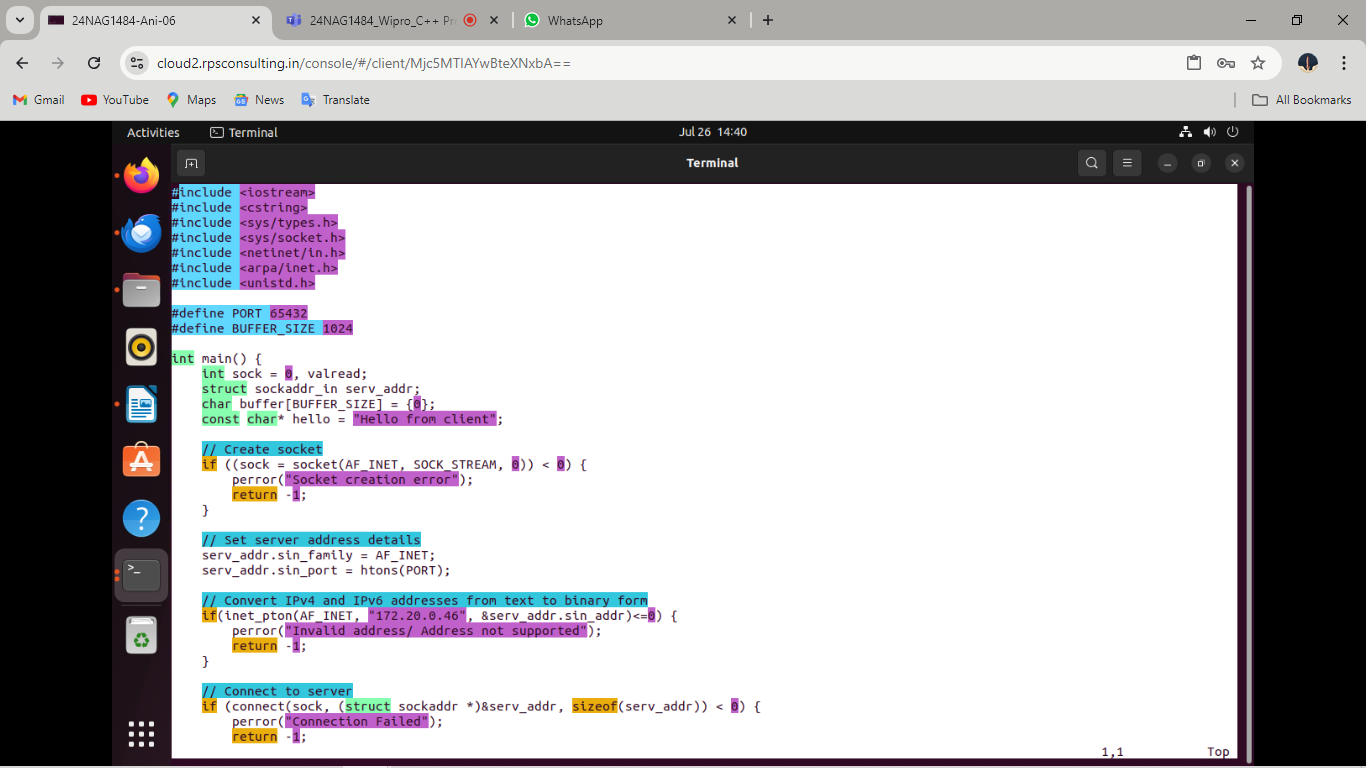
1. **client – side**

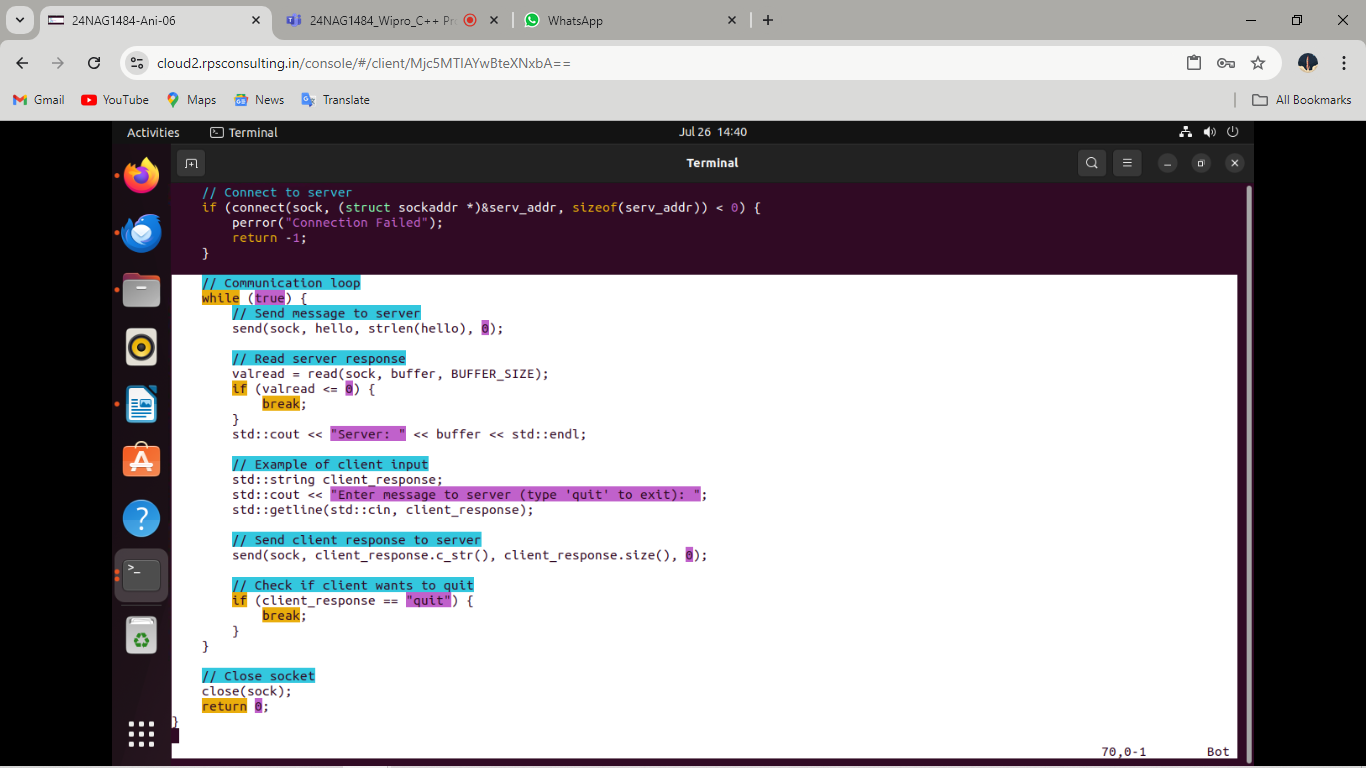
****

****

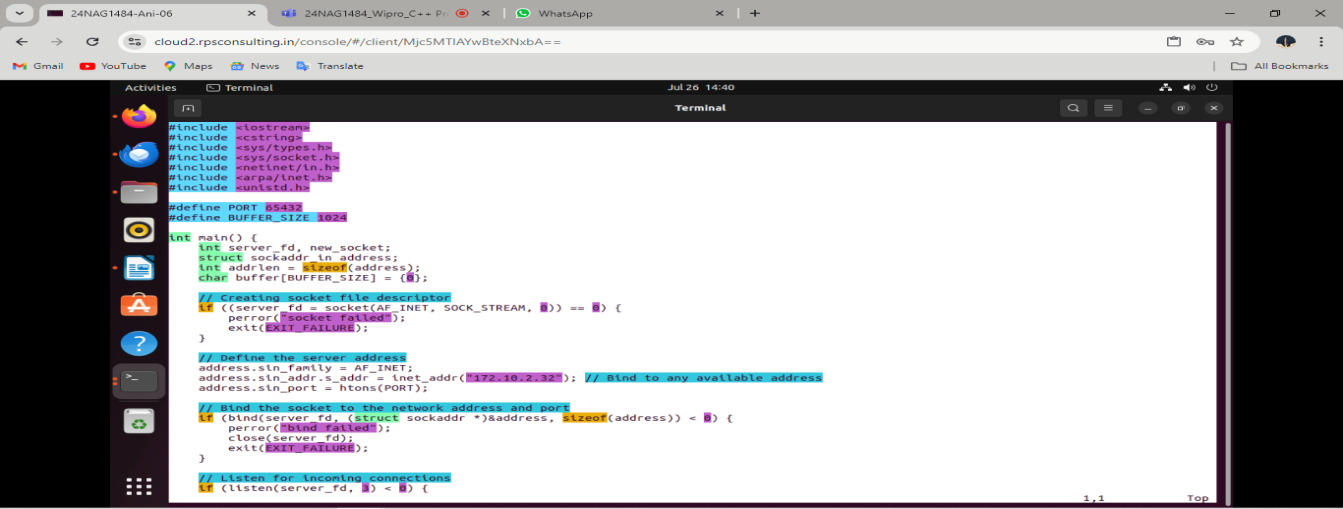
****

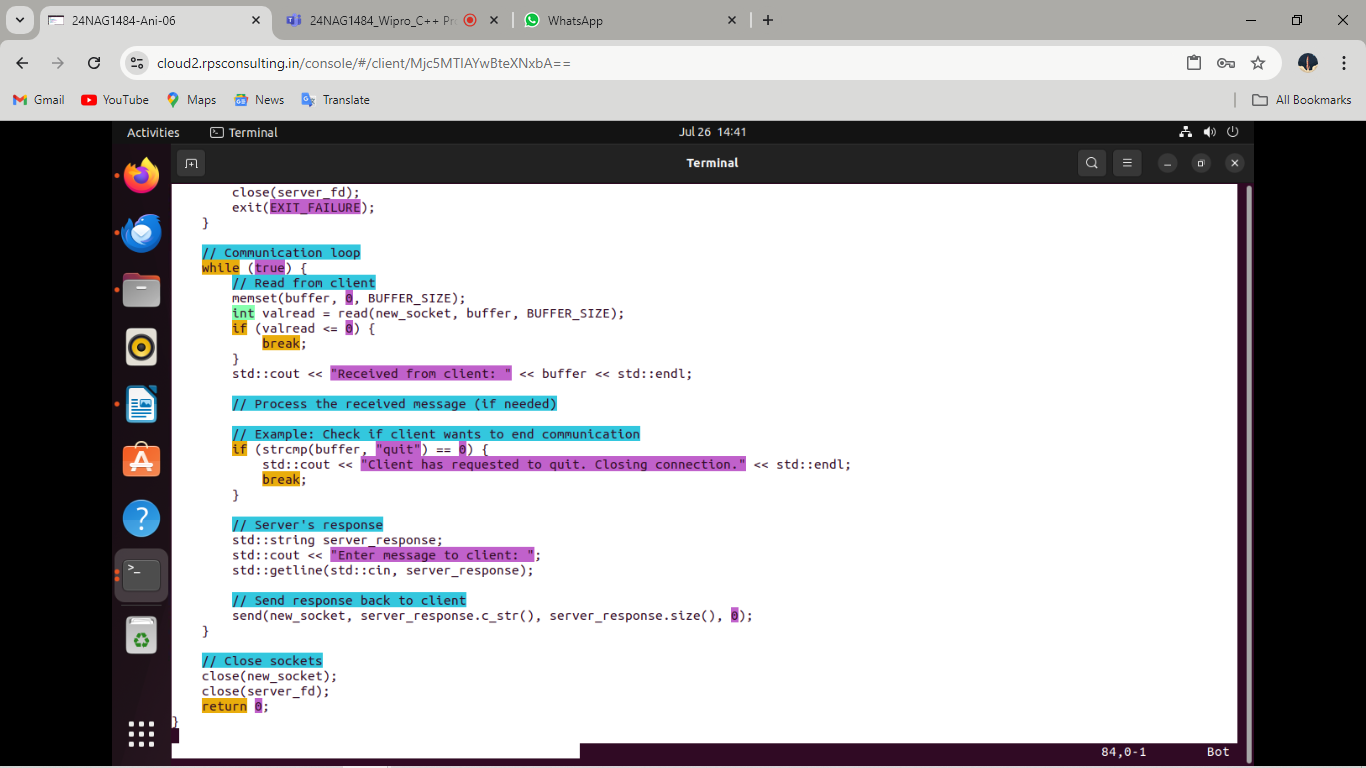
1. **Task -2 : Implement code for real-time chat in two different sytem’s IP’s using client server socket.**
2. **client - side**

****

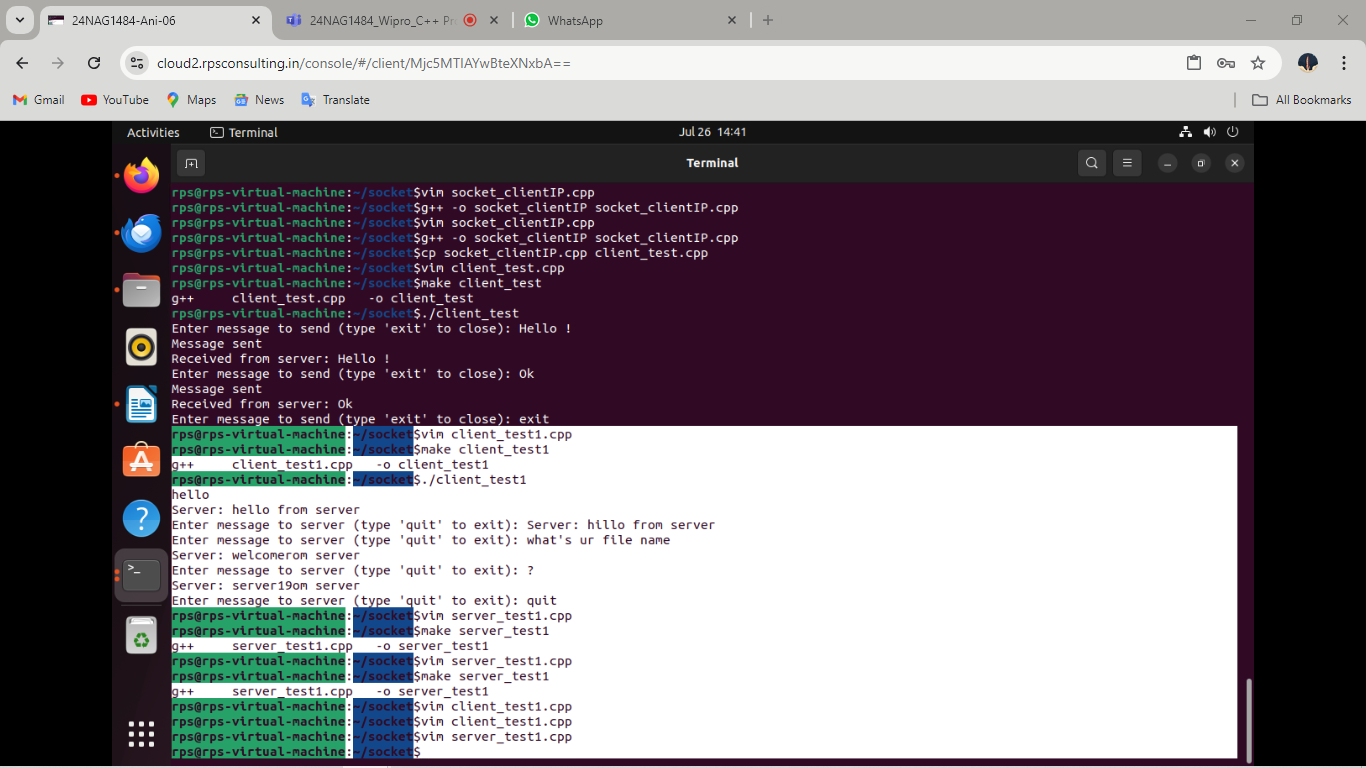
****

1. **server – side**

****

****

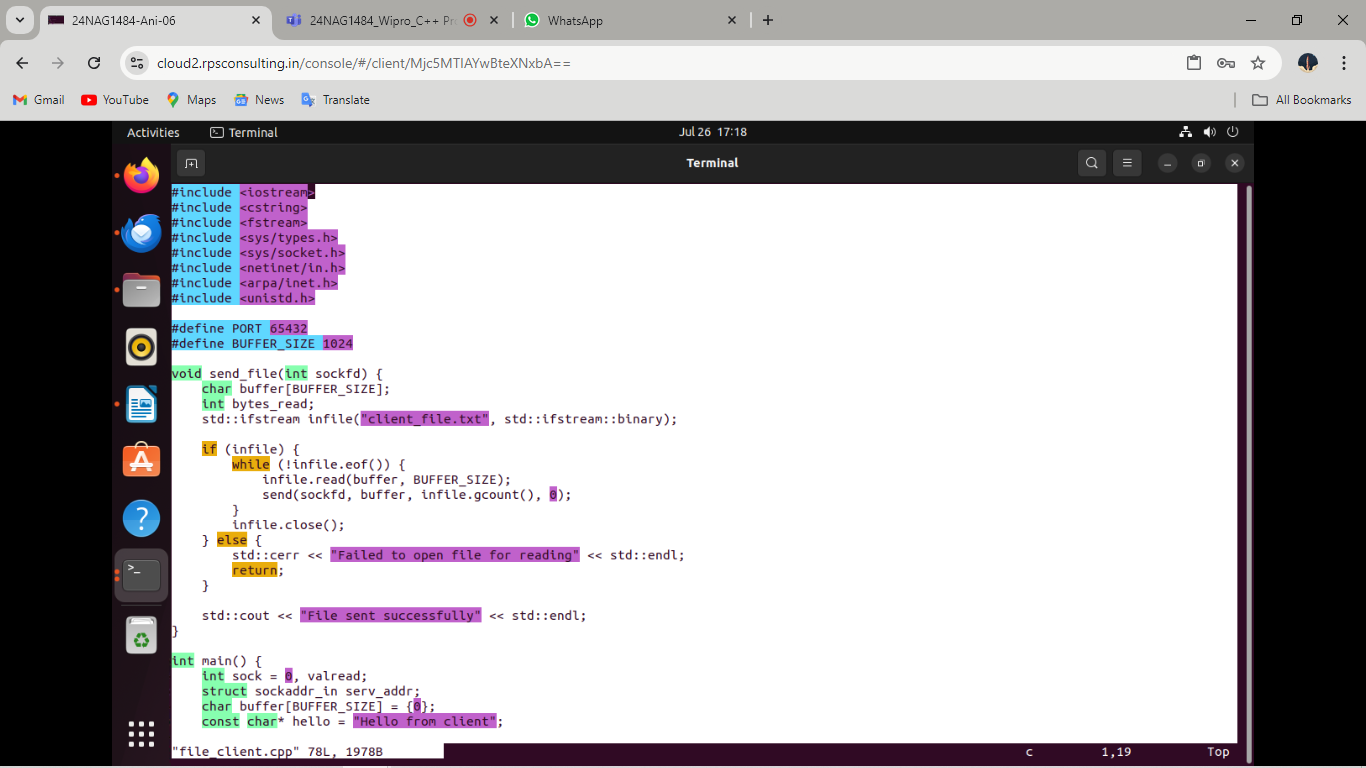
**Execution :**

****

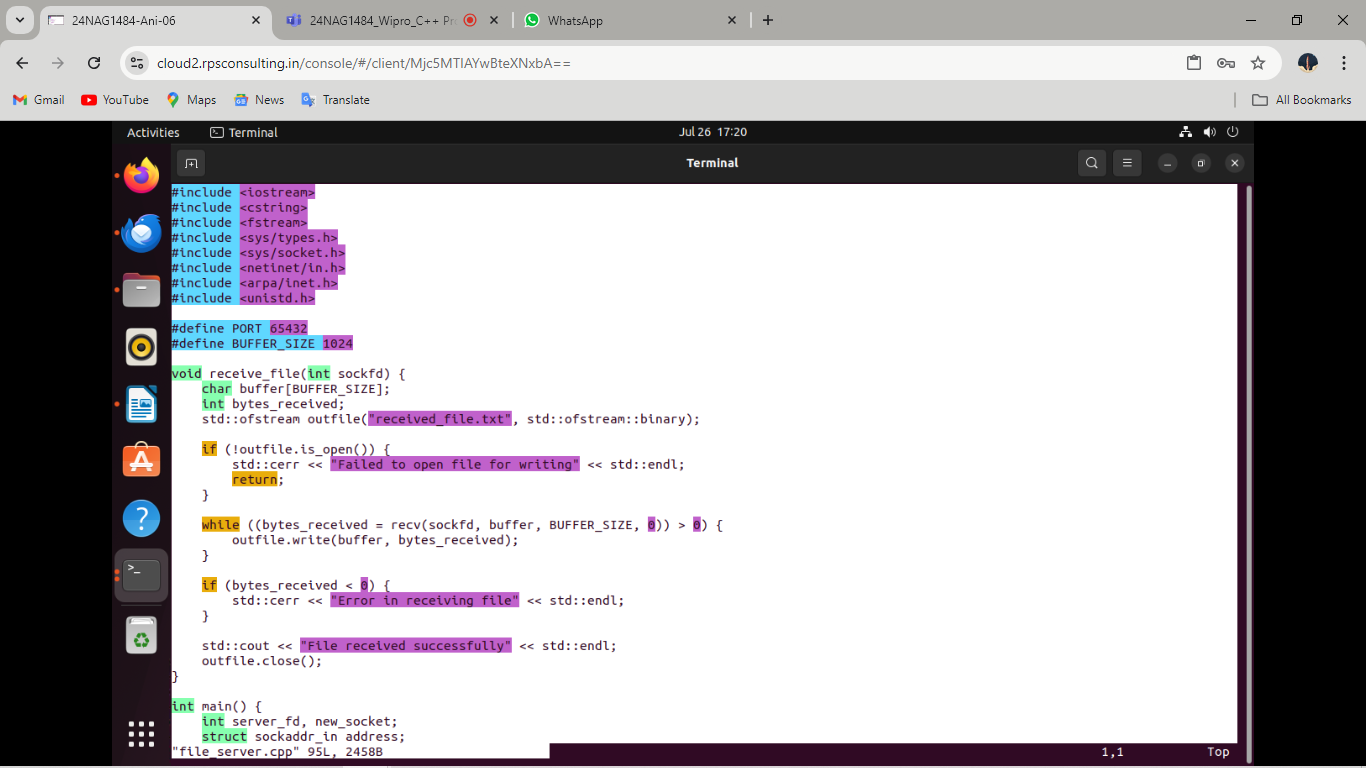
1. **Task : Create a text file and then from server to client and vice versa using two different system IPs.**

This program sets up a client that connects to a server **(running on 172.20.0.46:65432**), sends a **file (client\_file.txt)** to the server, receives a file from the server, and writes it to **received\_file.txt.** Error handling is included for file operations, socket creation, address conversion, connection, and file writing. **This example demonstrates basic file transfer using socket programming in C++.**

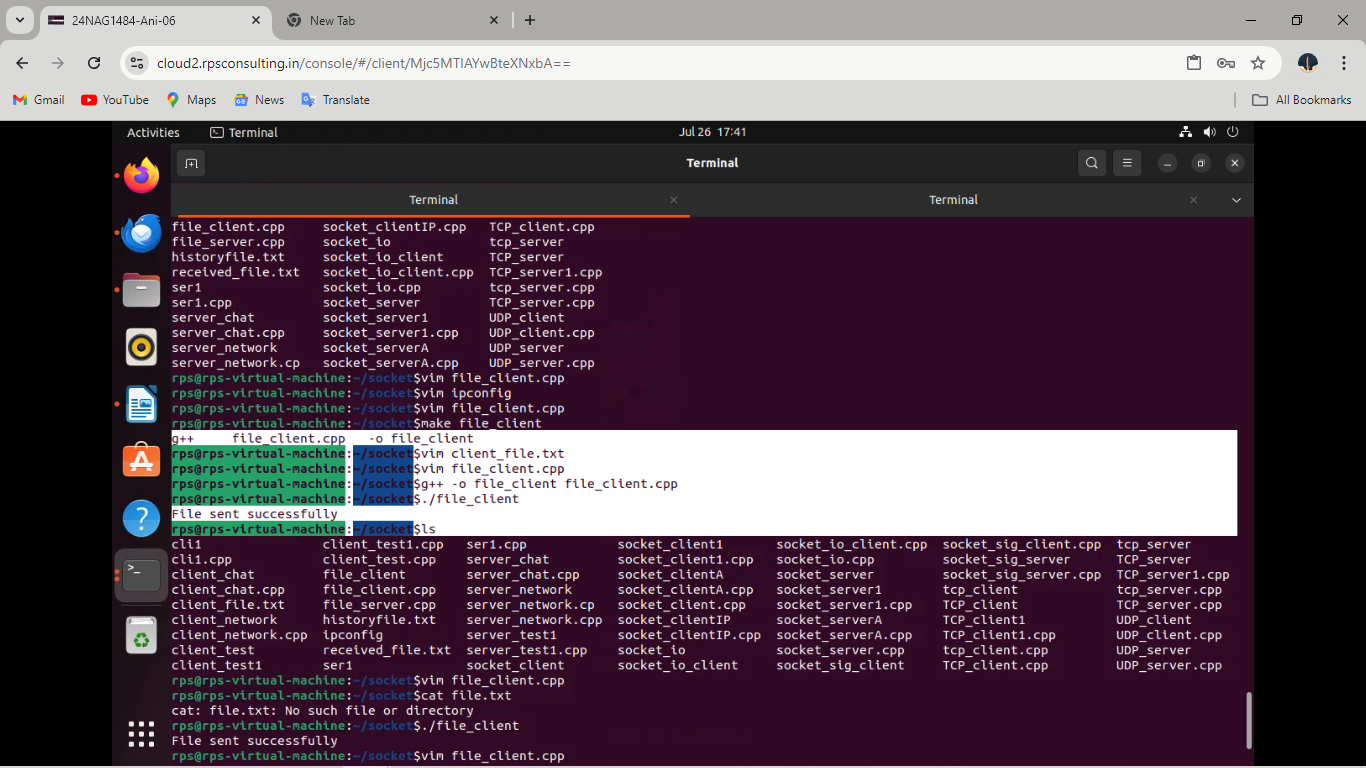
1. **Client side**

****

1. **Server-side**

****

**Implementation :**

****