Chat Application using Socket Programming

Overview

Design a LAN-based real-time chat application that enables multiple users to communicate over TCP sockets. The system supports basic messaging and can be extended with private messaging and optional file sharing.

Tools Required

Programming Language:

- Python

Modules & Libraries:

- `socket` (for networking)
- `threading` (to handle multiple clients)
- `tkinter` (optional, for GUI)

Development Steps

1. Build TCP Server

- Initialize a TCP socket server
- Accept incoming connections
- Use threading to handle multiple clients concurrently
- Broadcast messages to all connected clients

2. Build Client Program

- Create a client socket to connect to the server
- Enable send and receive functionality
- Display incoming messages in real-time

3. Enhance Functionality

- Implement client identification for private messaging
- Add support for sending small files or attachments (optional)
- Log messages with timestamps (optional feature)

4. Build GUI (Optional)

- Create a chat window using `tkinter`
    - Input box for typing messages
    - Send button
    - Scrollable chat history area

Expected Output

- A working real-time chat system over LAN
- Support for multiple clients using threads
- Text-based CLI or GUI for sending and receiving messages
- Demonstration of socket and threading concepts in practice