Project Report

Python LAN Chat Application

1. Objective

The objective of this project was to build a real-time local area network (LAN) chat application using Python. The application enables multiple users to communicate through a common server while maintaining a user-friendly interface.

2. Tools & Technologies

- Python 3
- socket – for network communication
- threading – to handle multiple clients concurrently
- Tkinter – for graphical user interface (GUI)
- File I/O – to save chat logs

3. System Design

Server Side

- Runs continuously, listening for client connections.
- Uses socket and threading to handle multiple clients.
- Broadcasts received messages to all connected clients.
- Logs all chat activity (join, leave, messages) into chat_logs.txt.

Client Side

- Connects to server using IP and port.
- Provides GUI with a chat display and input box.
- Sends messages to server and displays received messages.

- Supports commands:
  • /exit → disconnect from server
  • /mute → stop displaying messages locally
  • /unmute → resume displaying messages

4. Implementation Steps

1. Developed the socket server to accept connections and broadcast messages.
2. Added multi-client support using threading.
3. Created a Tkinter GUI for chat input/output.
4. Implemented join/leave event handling.
5. Integrated logging mechanism to save messages in a file.
6. Added command support (/exit, /mute, /unmute).

5. Testing

- A 2-user test was conducted on localhost (127.0.0.1).
- Extended testing was done over LAN by replacing localhost with server's IP.
- Verified message broadcasting, join/leave events, log creation, and command handling.

6. Results

- Successfully created a real-time chat system with multiple clients.
- Achieved stable message broadcasting and log saving.
- Provided a simple yet functional GUI for end users.

7. Future Enhancements

- Add encryption for secure communication.
- Provide user authentication (login system).
- Implement chat history viewer in the client GUI.
- Support for file sharing between users.

Conclusion

This project demonstrates how Python’s socket, threading, and Tkinter libraries can be combined to create a functional LAN-based chat application. It helped in understanding network programming, concurrency, and GUI development.