# Python Chat App (LAN)

## INTRODUCTION

In today's digital era, communication plays a vital role in both personal and professional environments. With the advancement of networking technologies, real-time communication over local and wide area networks has become increasingly important. This project focuses on building a **LAN-based chat application using Python**, which allows multiple users connected to the same network to communicate with each other through text messages.

## ABSTRACT

This project is a simple LAN-based chat application developed using Python. It enables multiple users on the same Local Area Network (LAN) to communicate through real-time text messaging. The system follows a client-server architecture, where the server manages client connections and message broadcasting. Python's built-in socket module is used for network communication, and threading is used to handle multiple clients simultaneously. The application runs in the command-line interface, making it lightweight and easy to use. It is ideal for local messaging in offices, schools, or home networks. This project demonstrates the basics of socket programming and real-time data exchange in Python

## TOOLS & TECHNOLOGY

- Python
- Socket module
- Threading module
- **Tkinter** (for GUI interface)
- TCP/IP Protocol
- Local Area Network (LAN)

## STEPS

### Set Up the Environment

- Choose a code editor (e.g., VS Code, PyCharm, or Notepad++).
- Ensure all devices are connected to the same LAN.

### Import Required Modules

- Import socket for networking.
- Import threading to handle multiple users.
- (Optional) Import tkinter for GUI if you're building a graphical version.

**Create the Server Script**

- Initialize a socket to listen for incoming connections.
- Accept multiple clients using threads.
- Broadcast messages from one client to all others.

**Create the Client Script**

- Connect to the server using IP address and port.
- Send messages to the server.
- Receive and display messages from the server.

**Implement Multi-threading**

- Use threads to allow sending and receiving messages simultaneously without freezing the UI or terminal.

**Build a GUI using Tkinter**

- Create input boxes, chat display area, and send buttons.
- Connect GUI actions to client-side socket functions.

**Test the Application**

- Run the server on one machine.
- Run client scripts on one or more other machines on the same LAN.
- Test message sending and receiving.
- Add features like user names, or message logs if needed.

# CONCLUSION

The Python LAN Chat Application successfully demonstrates how real-time communication can be implemented over a Local Area Network using basic socket programming concepts. Through the client-server model and multi-threading, multiple users can exchange messages simultaneously without delays. This project provides a practical understanding of networking principles, Python's socket and threading modules, and the challenges of handling multiple connections. Although the app currently has a simple text-based interface, it lays the foundation for future enhancements like graphical interfaces, file sharing, and encryption. Overall, this project is a valuable step toward building more complex networked communication systems.