LAN Chat Application

CSE 328 – Computer Networks Project — Spring 2024

Mohamed Mahmoud Gomaa Abdullah — **120210310** — CSE Section **4**

Introduction

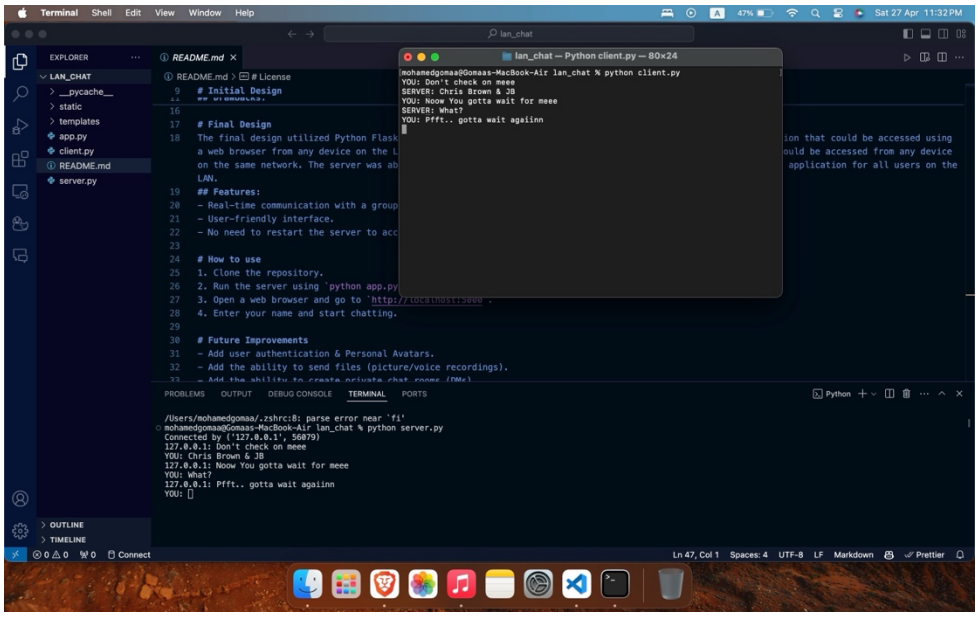
This is a simple LAN chat application that allows users to chat with each other on the same network. To provide real-time communication, the application was built using socket programming in Python. At first prototype, the application had a server and client side, their code can be found in client.py & server.py. This approach had drawbacks, which were solved in the final design using Python Flask and SocketIO.

Technologies

Python, Flask, SocketIO, HTML5, CSS, JavaScript

Initial Design

The initial design was based on a server-client model. Each was a python script that used the socket module to communicate with each other.



Drawbacks:

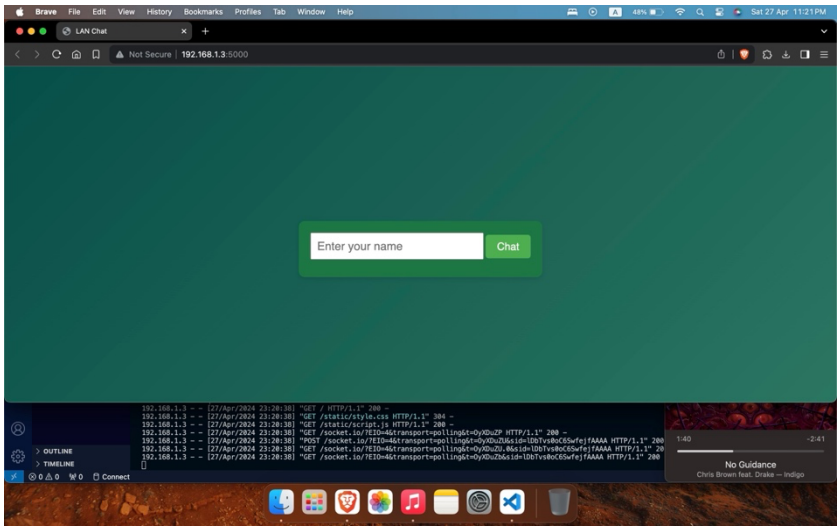
1. The server & client had to be running on a separate terminal window with no GUI.
2. Each user had to send only one message and wait for a response before sending another message.
3. The server could only handle one client at a time (maximum of 2-person communication).
4. The server had to be restarted to accept a new client.

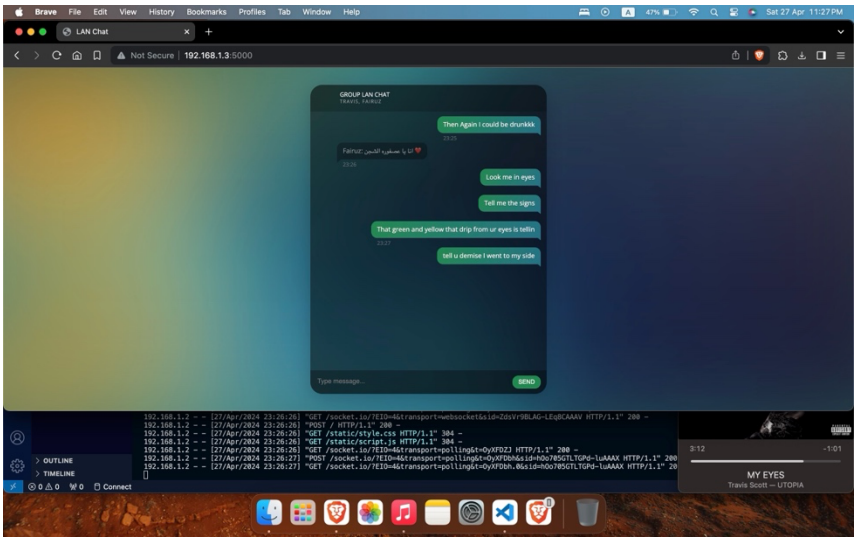
Final Design

The final design utilized Python Flask for server and SocketIO for the client. The client became a web application that could be accessed using a web browser from any device on the LAN. The server was able to handle multiple clients at the same time and could be accessed from any device on the same network. The server was able to broadcast messages to all connected clients, making it a group chat application for all users on the LAN.

Features:

1. Real-time communication with a group of users.
2. User-friendly interface.
3. No need to restart the server to accept new clients.





How to use

1. Clone the repository.
2. Run the server using python app.py.
3. Open a web browser and go to http://localhost:5000
4. Enter your name and start chatting.

Future Improvements

5. Add user authentication & Personal Avatars.
6. Add the ability to send files (picture/voice recordings).
7. Add the ability to create private chat rooms (DMs).
8. Allow Push Notifications for new messages.

Credits

Much appreciation goes to the 3 articles listed below in referances, Chris Brown, Drake, Travis Scott, Fairuz, Islam Kabonga for their contribution to the working playlist[1], and the generous Fabio Ottaviani who designed the UI and made it open source.

References

1. Python Sockets: An Introduction
2. Building Apps using Flask, SocketIO, and JavaScript Socket.IO - Part 1
3. How to Build a Simple Real-Time Application using Flask, React, and Socket.IO
4. UI Design

License

[1] For the playlist, please send an email to mohamed.abdullah@ejust.edu.eg