**Simple TCP Chat Application Documentation**

**Insights:**

During the development of this chat application, I acquired…

* The basics of socket programming in Python.
* How to handle multiple clients using threading.
* How to create a simple client-server architecture for a chat application.
* Learned about using TCP sockets for network communication.
* Explored how to create a basic server-client chat application.
* Gained knowledge about socket programming in Python.

Some challenges I encountered and how I overcame them...

* Ensuring proper error handling to avoid crashes when clients disconnect unexpectedly.
* Managing the threading aspect to ensure smooth communication between clients and the server.

**Implementation Details:**

* The program consists of two classes: Server and Client.
* The Server class listens for incoming connections from clients and handles each client connection in a separate thread.
* The Client class connects to the server and starts a separate thread to receive messages from the server.
* Both server and client use TCP sockets for communication.
* Additional modules or libraries used: threading
* Brief explanation of the program's logic and flow: The server listens for incoming connections and broadcasts messages received from one client to all other connected clients. The client connects to the server, sends messages to the server, and displays messages received from other clients via the server.
* Implemented a simple TCP chat application using Python.
* The application consists of a server component and a client component.
* The server listens for incoming connections from clients and handles message exchange.
* Clients connect to the server and send/receive messages.
* The communication is text-based and operates over a local network using TCP sockets.

**Future Improvements:**

* Implement error handling for network disconnections and exceptions.
* Add support for multiple clients connecting to the server simultaneously.
* Enhance the user interface with a GUI for a better user experience.