Title: Secure Chat Application with End-to-End Encryption

---

1. Introduction: In today's digital world, communication privacy is critical. With rising threats of data breaches and surveillance, secure messaging systems are in demand. This project aims to create a real-time chat application using Python, Flask, and SocketIO that ensures end-to-end encryption for private and secure communication.

---

2. Abstract: The Secure Chat App is a terminal-based messaging system where users can send and receive encrypted messages in real-time. Using Flask for the server-side framework, Flask-SocketIO for real-time communication, and the `cryptography` library for AES-based symmetric encryption, the project ensures that messages are encrypted during transmission. This system is designed for local use but sets the foundation for secure communication platforms.

---

3. Tools & Technologies Used: - **Programming Language:** Python 3 - **IDE:** Visual Studio Code (VS Code) - **Framework:** Flask - **Real-time Communication:** Flask-SocketIO - **Encryption:** cryptography (Fernet - AES symmetric encryption) - **Terminal/Command Line:** For running server and clients

---

4. Steps Involved in Building the Project: 1. **Environment Setup:** Installed Python and necessary libraries (flask, flask-socketio, cryptography, requests, websocket-client). 2. **Project Structure Created:** - `app.py` - Flask server handling message transfer - `client.py` - Terminal-based client for chatting - `encryption_utils.py` - Encryption/decryption functions using a fixed shared Fernet key 3. **Encryption Logic:** Implemented symmetric encryption (AES via Fernet) to secure message contents. 4. **Server Logic:** Handles message receiving, decrypts it, re-encrypts and broadcasts to all clients. 5. **Client Logic:** Accepts user input, encrypts the message, sends it to server and decrypts received messages. 6. **Testing:** Multiple client terminals simulated chat sessions successfully with proper encryption & decryption.

---

5. Conclusion: The Secure Chat App successfully demonstrates a real-time messaging system with end-to-end encryption using simple and efficient Python tools. It provides a strong foundation for building more complex secure messaging platforms. This project not only enhances understanding of Flask and encryption but also promotes privacy-first development practices.

(End of Report)