**SecureChat - End-to-End Encrypted Messaging**

* **Introduction:**
* SecureChat is a web-based messaging application designed to provide secure communication between users through end-to-end encryption. By utilizing RSA and AES encryption algorithms, SecureChat ensures that messages remain private and are only accessible to the intended recipients. This project aims to demonstrate the implementation of secure messaging protocols in a user-friendly interface.
* **Abstract:**
* The SecureChat application allows users to register, log in, and communicate securely. Upon registration, users are assigned a unique RSA key pair for secure key exchange. Messages are encrypted using AES, ensuring that even if intercepted, the content remains unreadable without the appropriate keys. The application leverages Flask for the backend and Socket.IO for real-time communication, providing a seamless user experience.
* **Tools Used:**

1. Flask: A lightweight web framework for Python, used to build the server-side application.
2. Flask-SocketIO: A library that enables real-time communication between clients and the server.
3. Cryptography: A Python library for implementing secure encryption and decryption algorithms.
4. Tailwind CSS: A utility-first CSS framework for styling the frontend.
5. CryptoJS: A JavaScript library for cryptographic operations on the client side.
6. Steps Involved in Building the Project

* **Setting Up the Environment:**
* Install Python and create a virtual environment.
* Install necessary Python packages using pip.
* **Building the Backend:**
* Created a Flask application to handle user registration and login.
* Implemented RSA key pair generation for each user.
* Set up Socket.IO for real-time messaging capabilities.
* **Implementing Encryption:**
* Used the Cryptography library to implement RSA and AES encryption.
* Created functions for encrypting and decrypting messages.
* **Developing the Frontend:**
* Designed the user interface using HTML and Tailwind CSS.
* Implemented JavaScript to handle user interactions and communicate with the backend via Socket.IO.
* **Testing the Application:**
* Conducted thorough testing to ensure that user registration, login, and messaging functionalities work as intended.
* Tested the encryption and decryption processes to verify the security of the messages.
* **Conclusion:**
* SecureChat demonstrates the feasibility of building a secure messaging application using modern web technologies. By implementing end-to-end encryption, the application ensures that user communications remain private and secure. This project not only highlights the importance of security in digital communication but also serves as a foundation for further enhancements, such as adding features like group messaging and file sharing.