Họ và tên: Nguyễn Thị Huyền Trang

Lớp: ANHTTT

BÁO CÁO BTL KĨ THUẬT LIÊN MẠNG

Truyền mã hóa file qua giao thức TCP/UDP

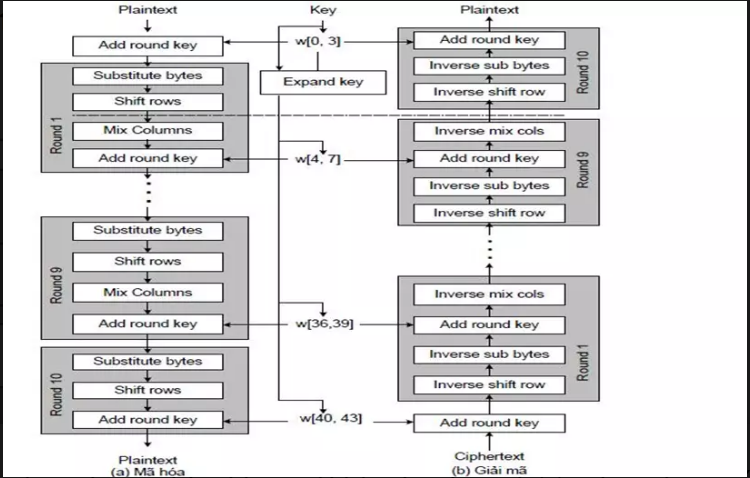
*(mã hóa chat realtime tin nhắn và file)*

1. Tổng quan về Socket TCP và AES:

1.1 AES:

* AES (viết tắt của từ tiếng anh: Advanced Encryption Standard, hay Tiêu chuẩn mã hóa nâng cao) là một thuật toán mã hóa khối được chính phủ Hoa Kỳ áp dụng làm tiêu chuẩn mã hóa.
* Thuật toán được xây dựng dựa trên Rijndael Cipher phát triển bởi 2 nhà mật mã học người Bỉ: Joan Daemen và Vincent Rijmen.
* AES làm việc với các khối dữ liệu 128bit và độ dài khóa 128bit, 192bit hoặc 256bit. Các khóa mở rộng sử dụng trong chu trình được tạo ra bởi thủ tục sinh khóa Rijndael.
* Hầu hết các phép toán trong thuật toán AES đều thực hiện trong một trường hữu hạn của các byte. Mỗi khối dữ liệu đầu vào 128bit được chia thành 16byte, có thể xếp thành 4 cột, mỗi cột 4 phần tử hay một ma trận 4x4 của các byte, nó gọi là ma trận trạng thái. Tùy thuộc vào độ dài của khóa khi sử dụng 128bit, 192bit hay 256bit mà thuật toán được thực hiện với số lần lặp khác nhau.

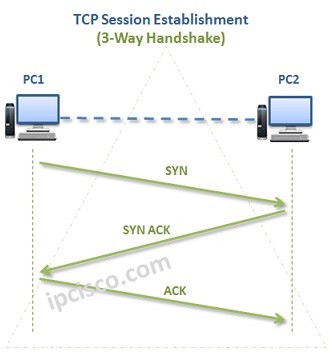
1.2 Sơ đồ mã hóa:



1.3 Socket TCP

Socket là giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều, hay còn gọi là two-way communication để kết nối 2 process trò chuyện với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket.

chức năng của socket là kết nối giữa client và server thông qua TCP/IP và UDP để truyền và nhận giữ liệu qua Internet. Giao diện lập trình ứng dụng mạng này chỉ có thể hoạt động khi đã có thông tin về thông số IP và số hiệu cổng của 2 ứng dụng cần trao đổi dữ liệu cho nhau.



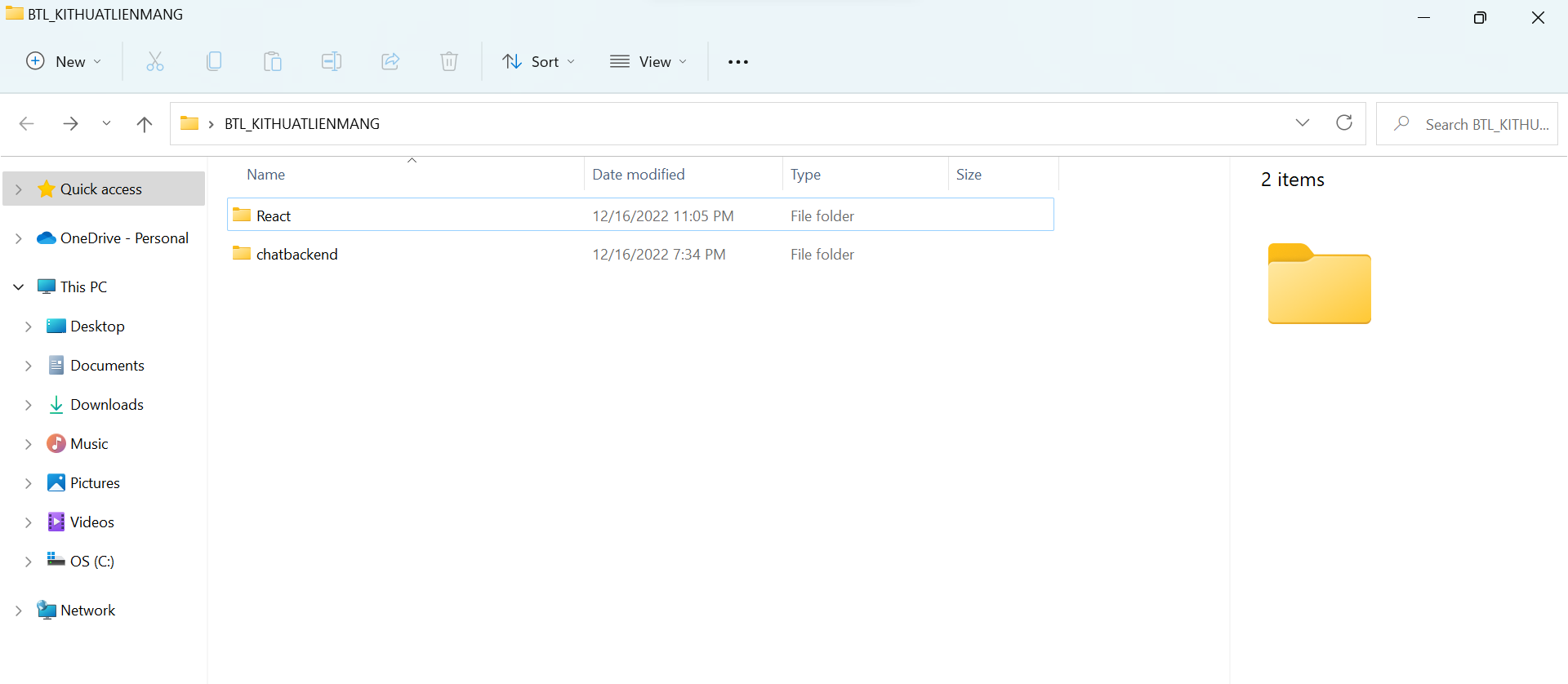
*Hình Quy trình bắt tay 3 bước của TCP*

1. Chương trình chat realtime mã hóa tin nhắn, file bằng AES

Môi trường : chạy trên visual code

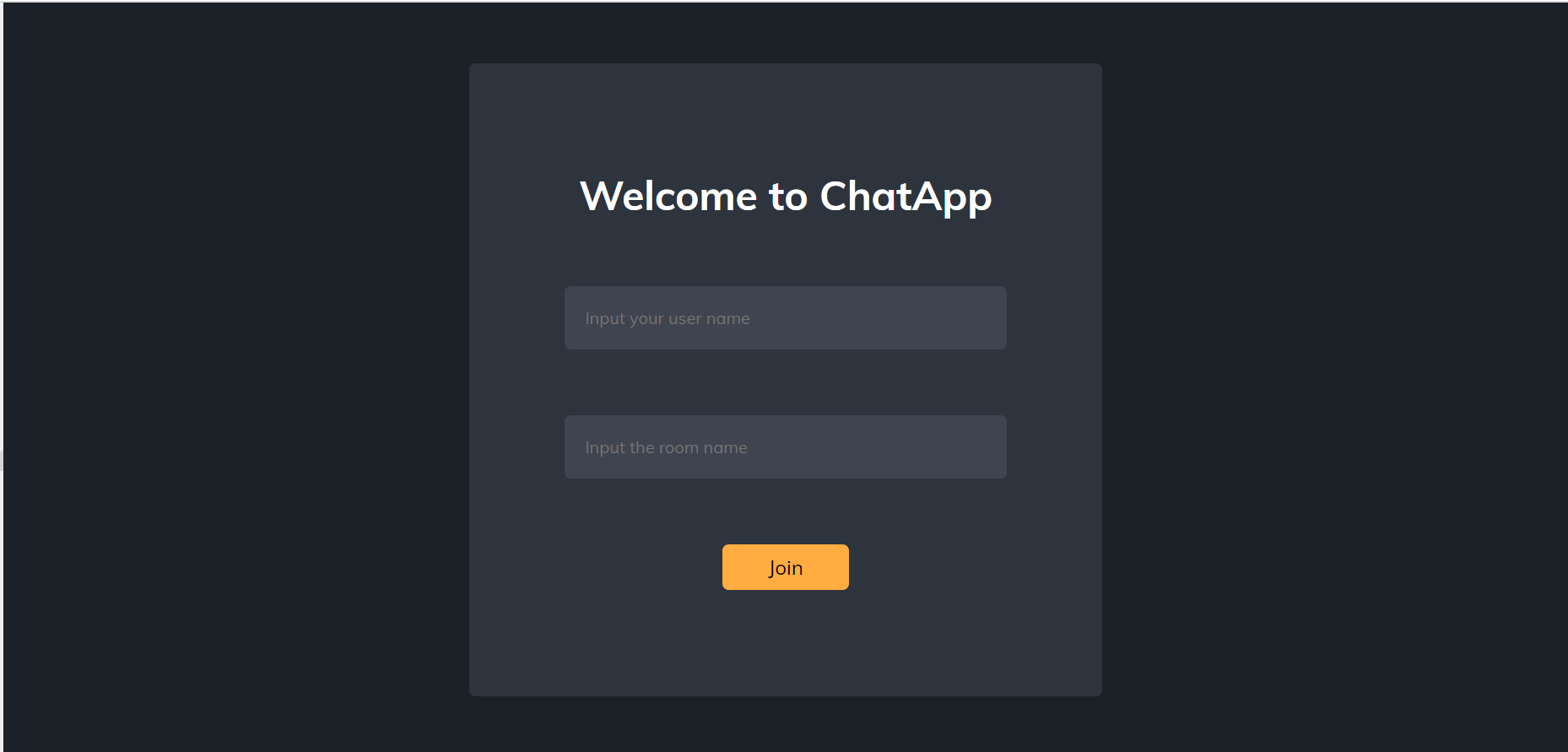
* client : reactjs
* server: nodejs

Cấu trúc project: Gồm 2 foder ( foder React – chạy client, foder chatbackend – chạy server)



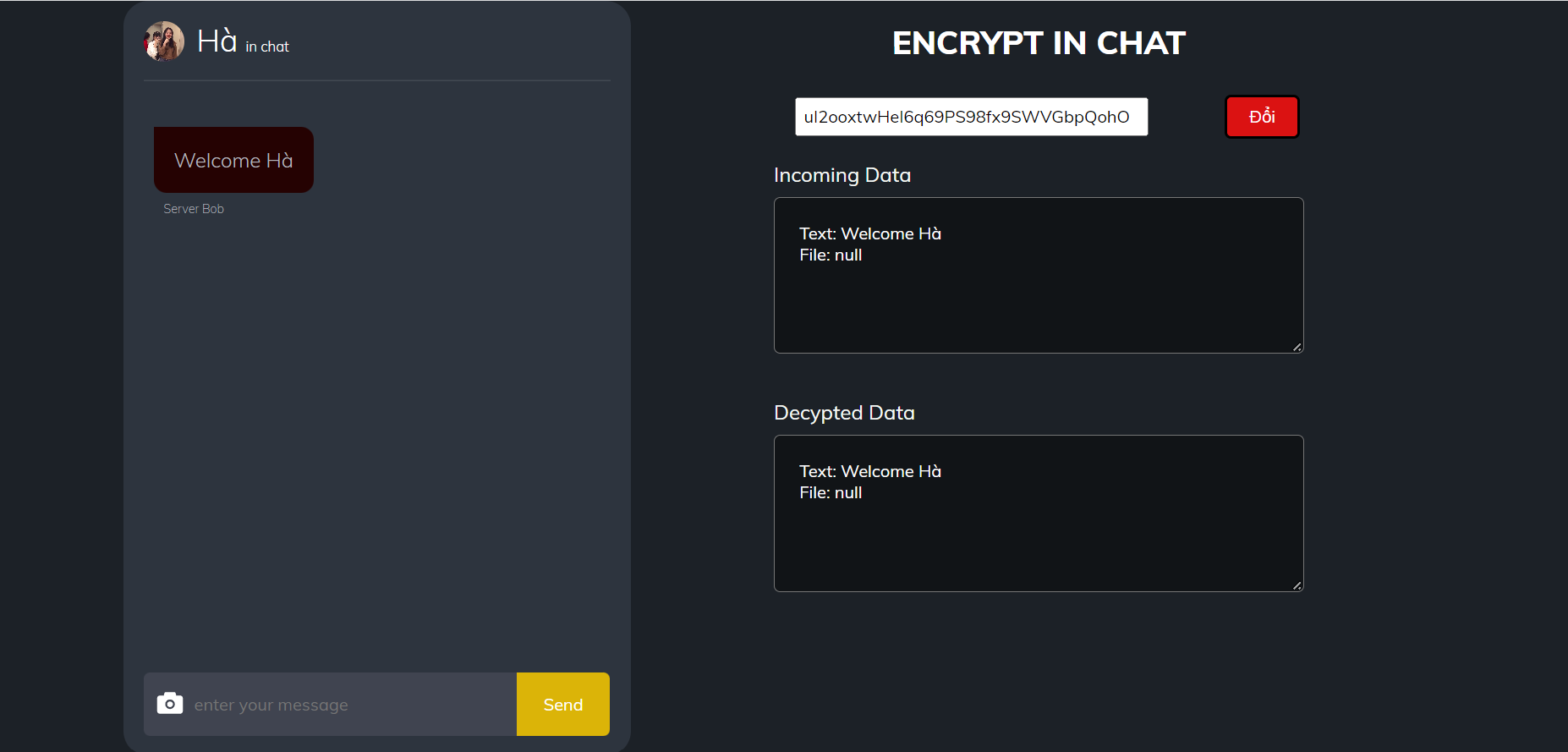
Bước 1: Chạy “node server” khởi tạo server trên cổng 8080

Bước 2: Chạy “npm start” chạy giao diện client

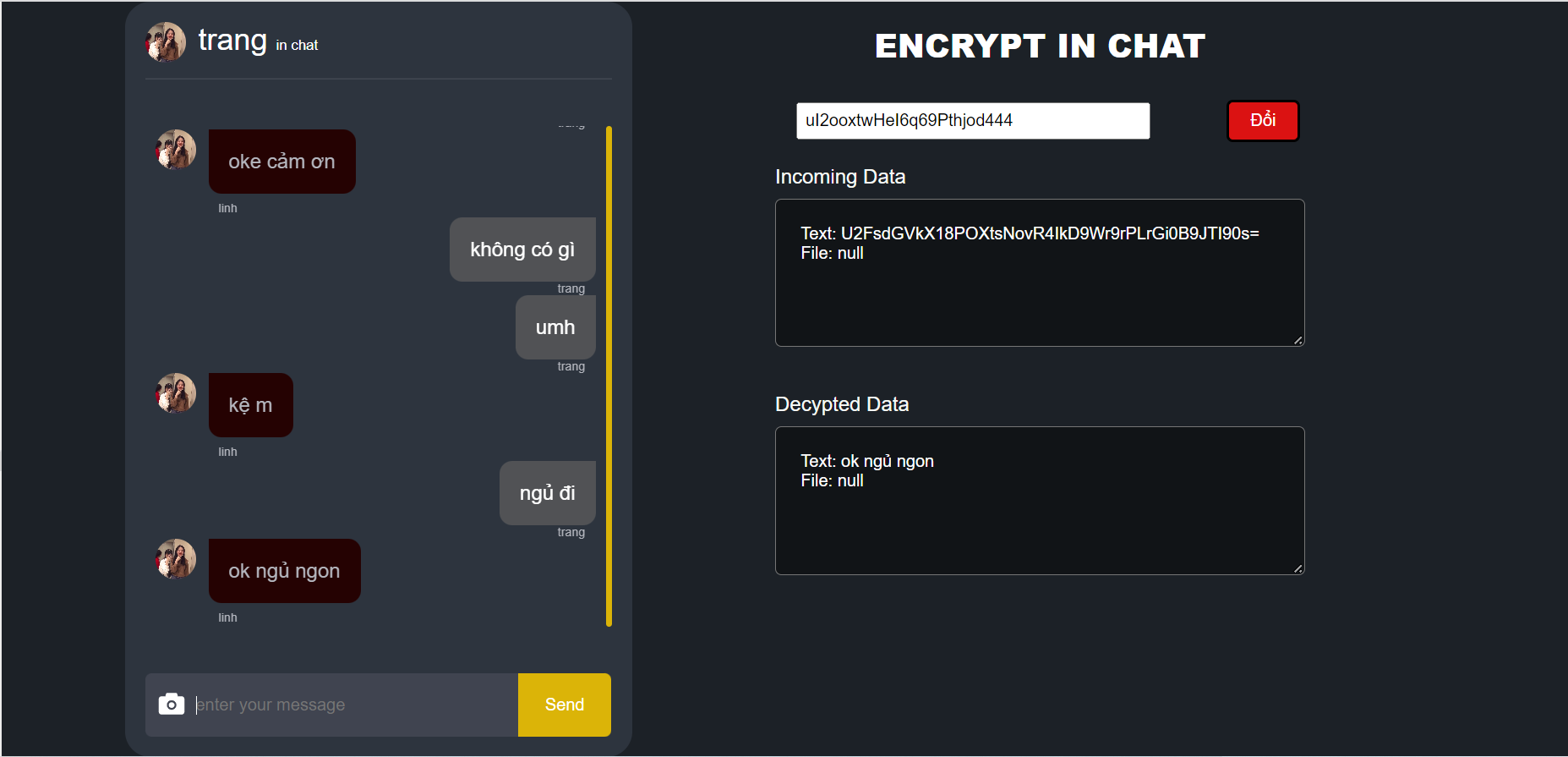


*Hình 1.1 : Giao diện đăng nhập vào phòng chat*

Bước 3: Nhập thông điệp muốn gửi qua input, dữ liệu chưa gửi chưa mã hóa, hiển thị trong hộp thoại Decypted Data, dữ liệu đã gửi và mã hóa hiển thị trong hôp thoại Incoming Data.

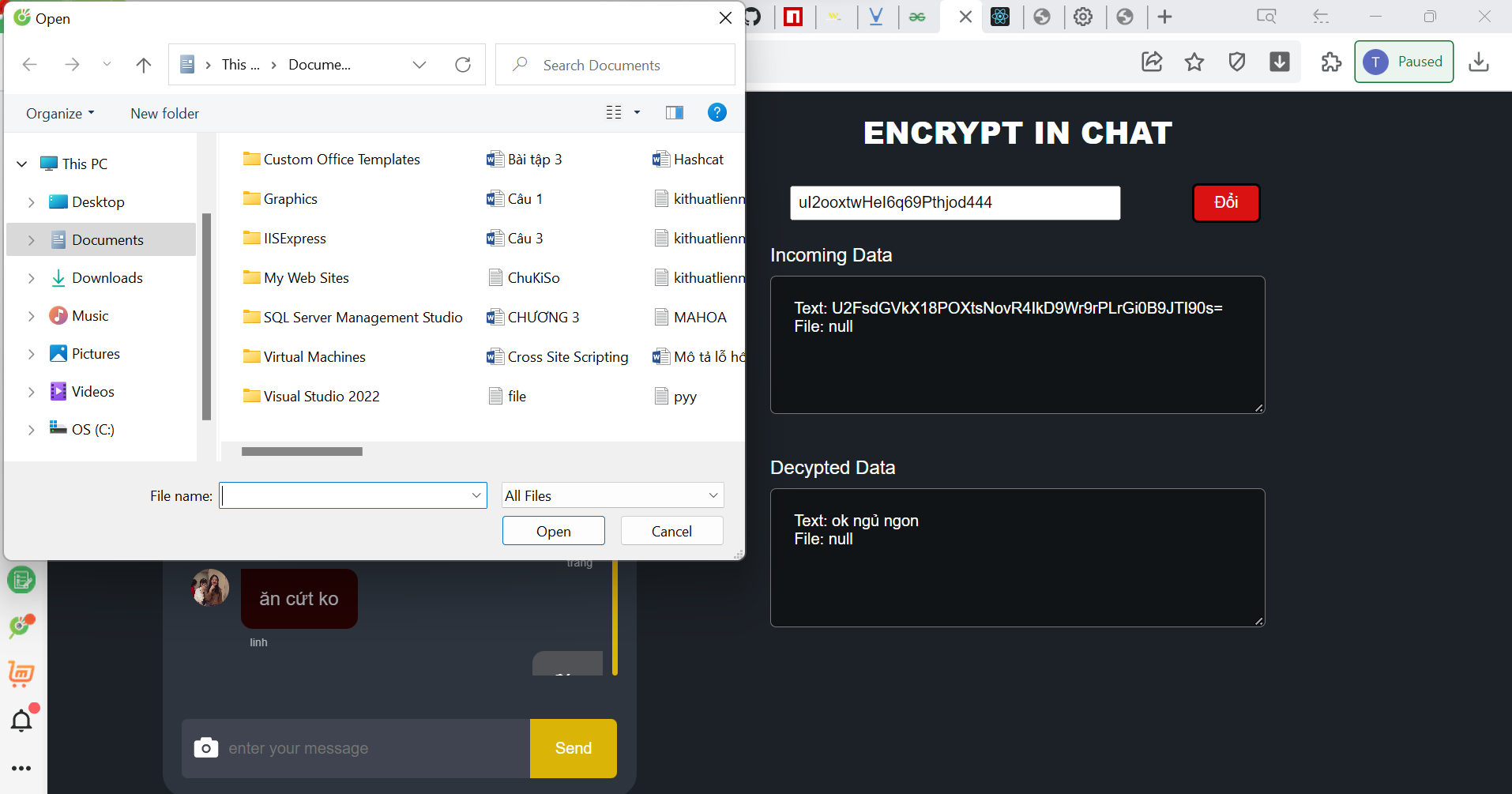


*Hình 1.2 Giao diện khi người dùng đăng nhập vào phòng chat thành công*

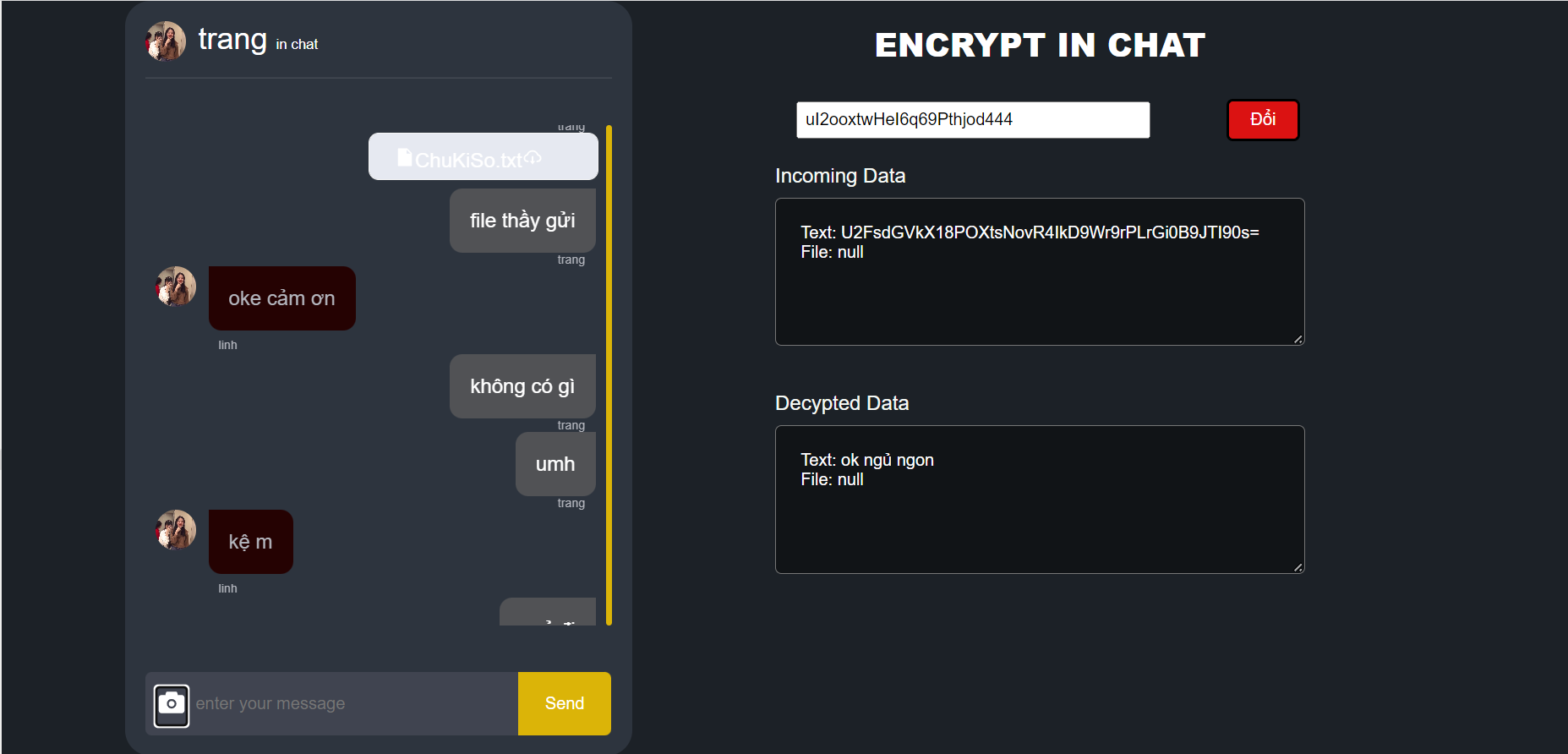


*Hình 1.3 Giao diện hiển thị quá trình chat của người dùng*

Bước 4: Chọn file cần gửi, file gửi mã hóa bằng AES

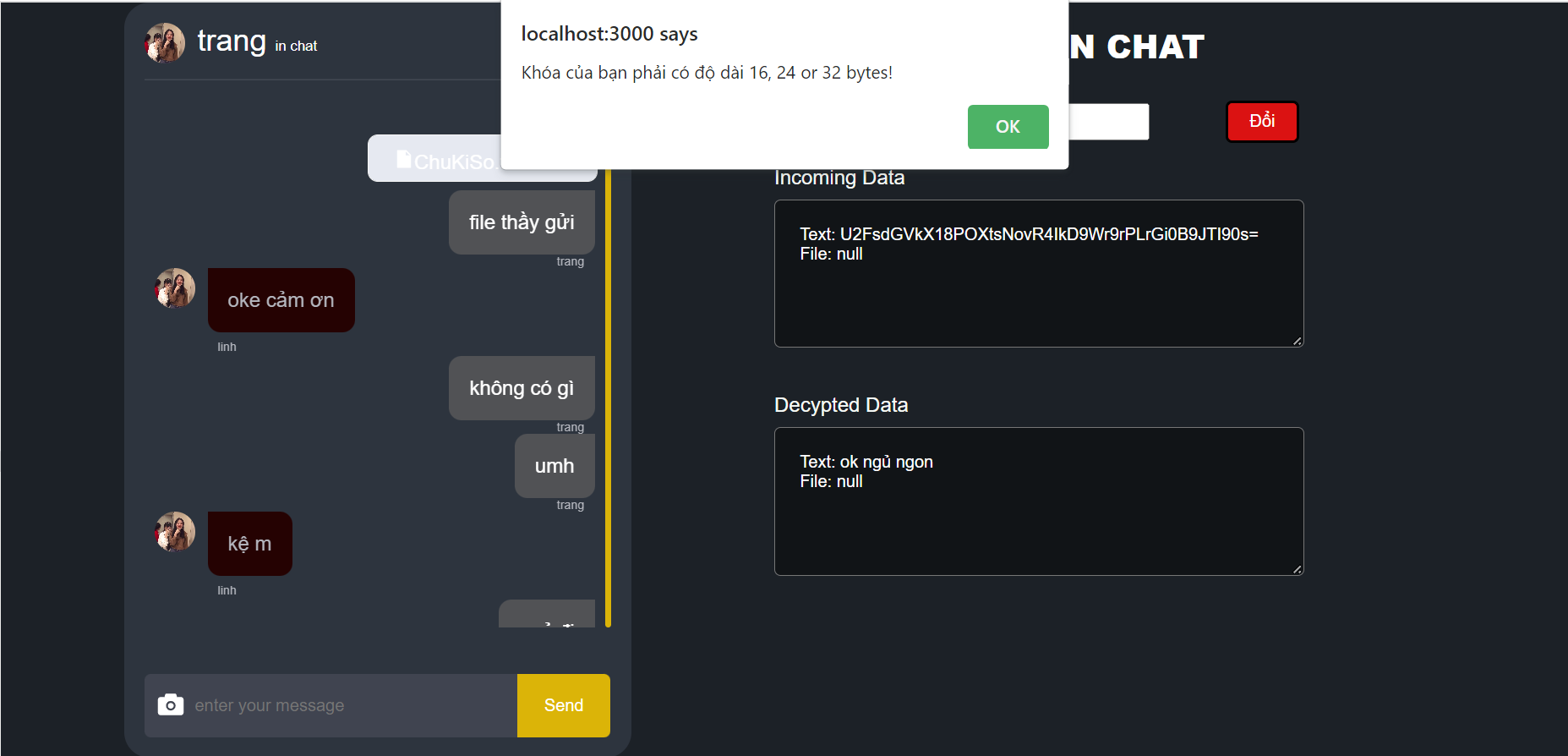


*Hình 1.4: Người dùng chọn file cần gửi*



*Hình 1.5: File gửi thành công hiển thị trên màn hình chat*

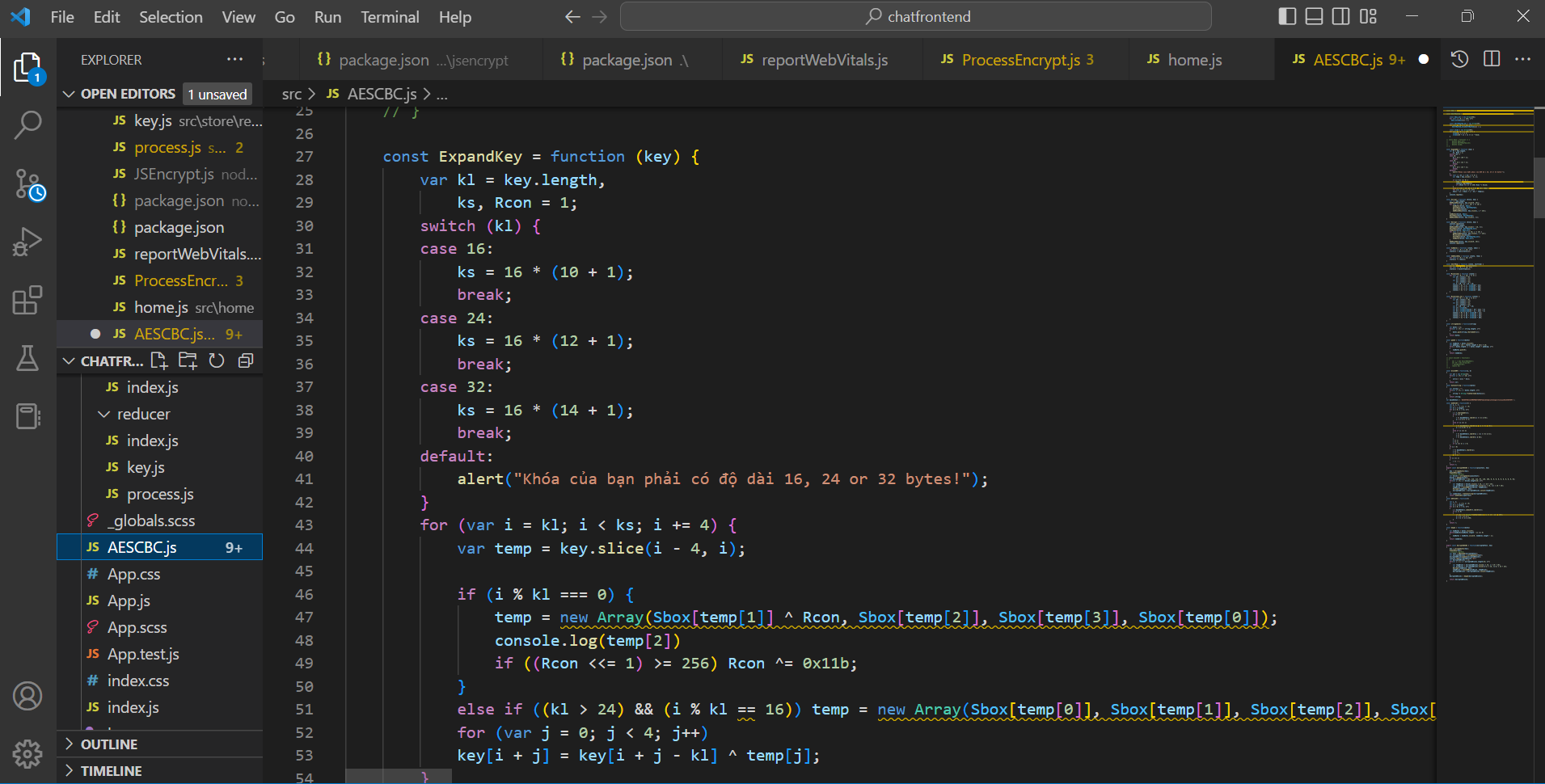
Bước 5: Thay đổi khóa bí mật với 3 chế độ (16, 24 hoặc 32 bytes)



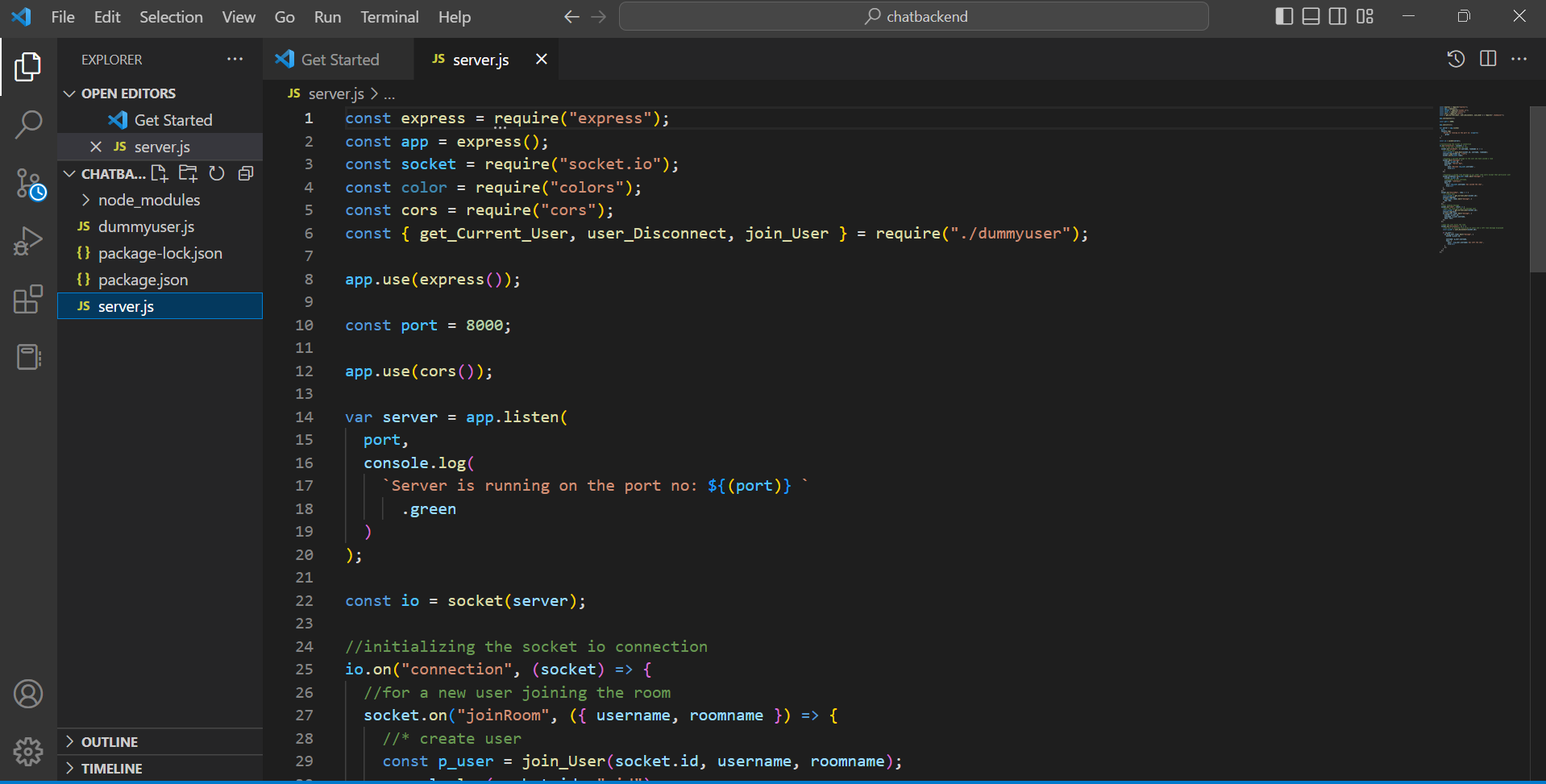
*Hình 1.6: Giao diện yêu cầu người dùng thay đổi khóa bí mật theo 3 chế độ*

3. Một số đoạn code chính của chương trình chat realtime mã hóa tin nhắn, file bằng AES

3.1 Code viết thuật toán AESCBC cải tiến từ thuật toán AES với 3 độ dài khóa (16, 24 hoặc 32 bytes)



3.2 Code tạo server sử dụng socket TCP nhận truyền tin qua socket.on và socket.emit



3.3 Code mã hóa thông điệp và file trong giao diện client trước khi gửi dữ liệu lên server

