UNIVERSITY OF INFORMATION TECHNOLOGY

FACULTY OF COMPUTER NETWORK AND COMMUNICATION

Logo, company name

Description automatically generated

**Báo cáo**

Môn: Mật mã học

Cuối kì I (2024 – 2025)

**ENCRYPTION AUDIO DATA ON MULTIMEDIA**

**PRODUCT SERVICE PLATFORM**

Student: Nguyễn Huỳnh Nhân

Student ID Number: 23521080

Student: Võ Hoài Nam

Student ID Number: 23520990

Class: NT219.P22.ANTT

University of Information Technology

Lecturer: PhD. Nguyễn Ngọc Tự

**Hồ Chí Minh City, April 2025**

**Lời cảm ơn**

Lời nói đầu tiên, chúng em xin cảm ơn cố vấn của chúng em, Tiến sĩ Nguyễn Ngọc Tự, vì sự chỉ đạo và giám sát nhất quán cũng như cung cấp những thông tin quan trọng của dự án và giúp đỡ chúng em hoàn thành nghiên cứu.

Nguyễn Huỳnh Nhân – 23521080 – ANTT2023

Võ Hoài Nam – 23520990 – ANTT2023

Contents

[I. Tổng quan đề tài: 4](#_Toc155562174)

[1. Chủ đề: 4](#_Toc155562175)

[2. Ngữ cảnh vấn đề: 4](#_Toc155562176)

[2.1. Lí do chọn đề tài: 4](#_Toc155562177)

[2.2. Mục tiêu: 4](#_Toc155562178)

[3. Vai trò của các bên liên quan: 5](#_Toc155562179)

[4. Giải pháp: 5](#_Toc155562180)

[II. Bối cảnh: 5](#_Toc155562181)

[1. Double DNA encoding: 5](#_Toc155562182)

[2. Chaotic map: 6](#_Toc155562183)

[III. Hướng giải quyết và code: 7](#_Toc155562184)

[1. Solution Architecture: 7](#_Toc155562185)

[2. Encryption: 8](#_Toc155562186)

[3. Decryption: 13](#_Toc155562187)

[IV. Thực nghiệm và kết quả: 18](#_Toc155562188)

[V. Tổng kết và công việc tương lai: 18](#_Toc155562189)

[1. Tổng kết: 18](#_Toc155562190)

[2. Công việc tương lai: 18](#_Toc155562191)

# I. Tổng quan đề tài:

## 1. Chủ đề:

- Project topic: Mã hóa âm thanh

- Project title: Mã hóa dữ liệu âm thanh trên nền tảng dịch vụ sản phẩm đa phương tiện

## 2. Ngữ cảnh vấn đề:

- Lượng dữ liệu âm thanh đang ngày càng tăng lớn, bao gồm các nội dung như phim ảnh, video, podcast, âm nhạc trực tuyến... Điều quan trọng là cần có các giải pháp mã hóa hiệu quả để lưu trữ, truyền tải và bảo mật dữ liệu âm thanh.

- Nhiều dịch vụ số ngày nay cung cấp nội dung đa phương tiện kết hợp video, hình ảnh và âm thanh. Việc xử lý và mã hóa dữ liệu âm thanh trên các nền tảng này là rất cần thiết.

- Nghiên cứu về mã hóa dữ liệu âm thanh giúp cải thiện chất lượng trải nghiệm nghe nhạc, xem phim của người dùng trên các thiết bị di động và internet.

### 2.1. Lí do chọn đề tài:

- Âm thanh là một phần quan trọng của dịch vụ sản phẩm đa phương tiện. Việc mã hóa dữ liệu âm thanh đảm bảo rằng người dùng có thể truy cập và tận hưởng âm thanh một cách chất lượng và liền mạch trên nền tảng này.

- Mã hóa dữ liệu âm thanh có thể cung cấp cơ chế bảo mật để đảm bảo rằng dữ liệu âm thanh không bị truy cập trái phép hoặc biến đổi trong quá trình truyền tải.

### 2.2. Mục tiêu:

- Từ những điều đó, ta cần phải xây dựng một hệ thống mã hoá và giải mã để có thể bảo đảm về tính bảo mật, tính toàn vẹn và tính khả dụng của âm thanh khi được phát trực tuyến

### 3. Vai trò của các bên liên quan:

- Nhà cung cấp dịch vụ (Service Provider): Thực hiện các biện pháp bảo mật, cung cấp cơ chế mã hóa và đảm bảo tính khả dụng và chất lượng âm thanh cho người dùng.

- Tác giả (Author): Đăng tải các bản nhạc, file âm thanh.

- Người dùng (User): Những người đăng kí/ đăng nhập, mua bản quyền và sử dụng dịch vụ để có thể nghe các nội dung được phát.

- Hạ tầng (Infrastructure): Cung cấp nền tích cho dịch vụ lưu trữ, truyền tải dữ liệu mã hóa âm thanh an toàn, ổn định.

- Kẻ tấn công (Attacker): Cố gắng xâm phạm hệ thống để truy cập trái phép vào nội dung bản quyền.

## 4. Giải pháp:

- Xây dựng nền tảng cung cấp dịch vụ nghe nhạc trực tuyến cho người dùng

- Mã hoá dữ liệu và lưu trữ

- Giải mã và cung cấp cho nền tảng giúp cung cấp đến người dùng

# II. Bối cảnh:

## 1. Stream cipher (mã hóa dòng ):

- Stream cipher là một loại thuật toán mã hóa khối đối xứng mà mã hóa dữ liệu bit-by-bit hoặc byte-by-byte. Nó hoạt động bằng cách tạo ra một chuỗi bit giả ngẫu nhiên (gọi là keystream - dòng khóa ),sau đó kết hợp chuõi này với bản rõ( plaintext) bằng phép toán XOR

- Nguyên lý hoạt động:  
 + Bộ sinh dòng khóa:đây là thành phần cốt lõi sử dụng một khóa bí mật (secret key) và thường là một giá trị khởi tạo (IV) để tạo ra một chuỗi bit dài,trông có vẻ ngẫu nhiên.

+ Phép toán XOR: mỗi bit của bản rõ được XOR với bit tương ứng của dòng khóa để tạo ra bản mã(ciphertext).Quá trình giải mã cũng tương tự: bản mã được XOR với cùng 1 dòng khóa để khôi phục plaintext

- Ưu điểm:

+ Tốc độ cao: thường rất nhanh trong việc mã hóa và giải mã, phù hợp cho các ứng dụng yêu cầu hiệu suất cao như truyền tải dữ liệu thời gian thực.

+ Không lan truyền lỗi: Lỗi trong bản mã thường chỉ ảnh hưởng đến bit tương ứng trong bản rõ, không lan sang các bit khác

+ Không yêu cầu padding: không cần thêm phần đệm vào cuối dữ liệu.

- Nhược điểm:

+ Dễ bị tấn công nếu dòng khóa bị tái sử dụng: nếu cùng một khóa được sử dụng để mã hóa hai bản rõ khác nhau, tính bảo mật sẽ bị phá vỡ nghiêm trịng.

+ yêu cầu đồng bộ hóa: client -server phải trao đổi khóa

## 2. AES(Advanced Encrypted

- AES là một thuật toán mã hóa khối đối xứng thuộc loại block cipher(mã hóa khối ). Nó mã hóa dữ liệu theo từng khối có kích thước cố định (128bit hoặc 256bit).AES là tiêu chuẩn mã hóa được sử dụng rộng rãi nhất trên thế giới, được chính phủ Hoa Kỳ và tổ nhiều tổ chức,ngành công nghiệp khác chấp nhận.

- Nguyên lý hoạt động:  
 + Khối cố định: mã hóa các khối dữ liệu 128 bit.Nếu plaintext không đủ sẽ padding thêm   
 + Số vòng lặp: quá trình mã hóa và giải mã bao gồm nhiều vòng lặp (10,12 hoặc 14 vòng tùy thuộc vào độ dài khóa:128,182 hoặc 256 bit)  
 + Các phép biến đổi : mỗi vòng lặp bao gồm các phép biến đổi như Subytes(thay thế bytes),shiftRows,mixcolumns và addroundkey

- Ưu điêm   
 + Tính bảo mật cao: đã được kiểm định rộng rãi bởi cộng đồng mật mã học toàn cầu và đuọc coi lầ cực kỳ an toàn nếu triển khai đúng cách và sử dụng các chế độ hoạt động phù hợp (GCM hay CTR)

+ Hiệu suất tốt: có thể được tối ưu bằng phần cứng, mang lại tốc độ cao trên nhiều nền tảng

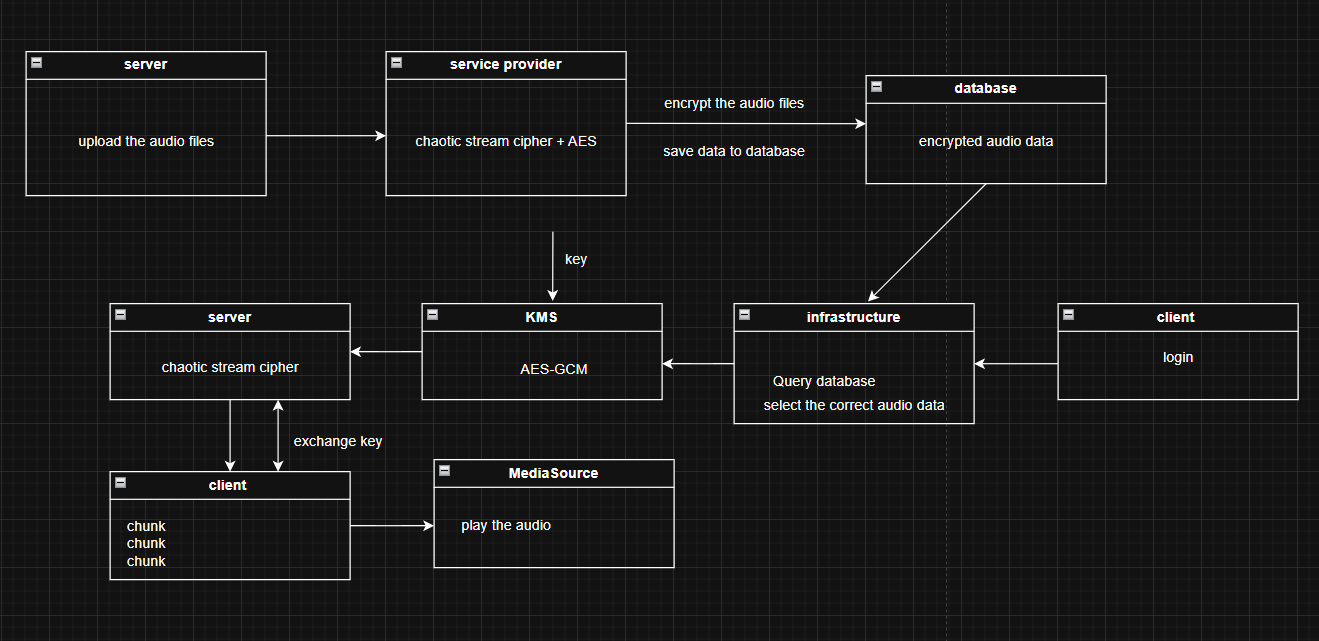
+ Tính phổ biến và tương thích: được hỗ trợ rộng rãi trong hầu hết các ngôn ngữ lập trình,hệ điều hành và thiết bị

- Nhược điểm:  
 + Yêu cầu padding : cần có cơ chế thêm xóa padding để xử lý dữ liệu có độ dài khoảng phỉa là bội số của kích thước khối

+ Lan truyền lỗi: Một lỗi trong bản mã có thể ảnh hưởng đến toàn bộ khối giải mã ( tùy thuộc vào chế độ hoạt động .

# III. Hướng giải quyết và code:

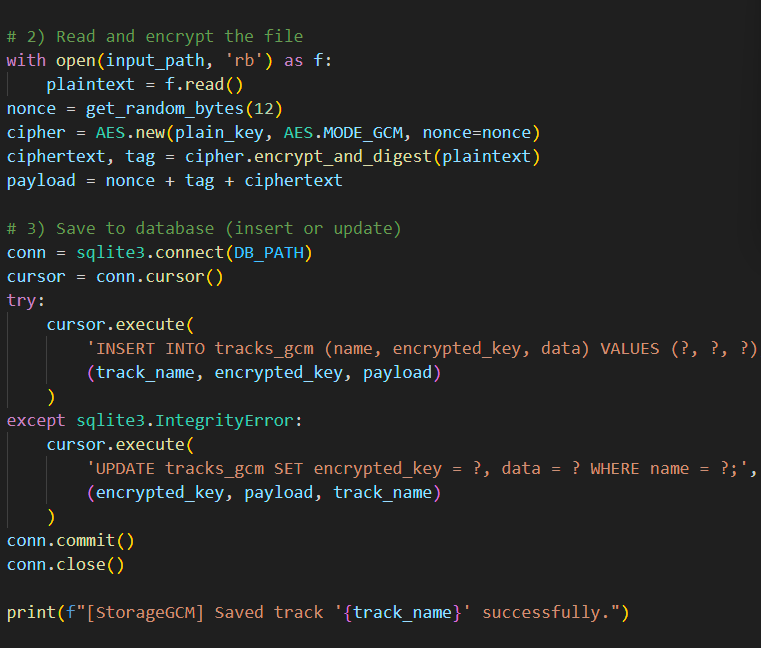
## 1. Solution Architecture:



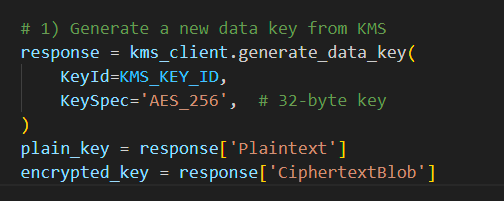
1. Tác giả tải file bài nhạc lên hệ thống dưới dạng bản rõ (vs .mp3 )
2. Server sẽ mã hóa file âm thanh đó và lưu vào database  
   2.1 sinh khóa mã hóa dữ liệu bằng KMS: hệ thống gửi yêu cầu tới AWS KMS để sinh ra một data encrypt key(DEK) mới.KMS sẽ trả về hai phiên bản DEK  
   + Plaintext DEK: khóa nguyên bản dùng để mã hóa dữ liệu   
   + Encrypted DEK: Khóa DEK đã được mã hóa bởi KMS   
   2.2 **Mã hóa Dữ liệu Âm thanh:** Hệ thống sử dụng **Plaintext DEK** (vừa nhận được từ KMS) để mã hóa dữ liệu âm thanh của bài hát bằng thuật toán **AES-GCM**.  
   2.3 Lưu vào database   
   + **Encrypted DEK:** Khóa DEK đã được KMS mã hóa.  
   + **Nonce và Tag của AES-GCM:** Các giá trị cần thiết cho quá trình giải mã AES-GCM sau này.
3. Khi người dùng muốn nghe nhạc thì phải login
4. Client yêu cầu bài hát
5. Hệ thống sẽ truy vấn và chọn dữ liệu bài hát đã được mã hóa đúng với yêu cầu của người dùng từ cơ sở dữ liệu   
   + lấy ra DEK , Nonce,tag
6. **File bài hát này sẽ kết hợp key được quản lý bởi KMS để được giải mã theo từng chunk âm thanh.  
   **6.1**. Giải mã DEK bằng KMS:** Hệ thống (trên server) gửi **Encrypted DEK** (vừa lấy từ database) đến AWS KMS. KMS sẽ giải mã và trả về **Plaintext DEK**.  
   ****6.2.** Giải mã AES-GCM:** Hệ thống sử dụng **Plaintext DEK** (vừa nhận được từ KMS), cùng với **Nonce** và **Tag** (lấy từ database), để giải mã lớp AES-GCM của dữ liệu âm thanh.   
   ****6.3**. Chia chunk và truyền tải:** Dữ liệu âm thanh sau khi giải mã AES-GCM được chia thành các chunk nhỏ và mã hóa bằng chaotic. Server truyền các chunk này (vẫn đang ở dạng Chaotic) về phía trình duyệt của người dùng. Kèm theo mỗi chunk hoặc qua một cơ chế khác, chaotic\_seed cũng được truyền tải an toàn đến client.
7. **Giao diện MediaSource sẽ được hiển thị làm giao diện chính để phát nhạc cho người dùng theo từng chunk âm thanh được tải lên.**   
   ****7.1.** Giải mã Chaotic (trên Client):** Trình duyệt của người dùng (sử dụng JavaScript) nhận các chunk dữ liệu đã được Chaotic mã hóa từ server. Với chaotic\_seed nhận được, client sẽ sử dụng thuật toán giải mã Chaotic để khôi phục các chunk dữ liệu âm thanh về dạng bản rõ.  
   ****7.2.** Phát nhạc:** Các chunk dữ liệu âm thanh bản rõ này được đưa vào MediaSource API của trình duyệt, cho phép phát nhạc liên tục và mượt mà.

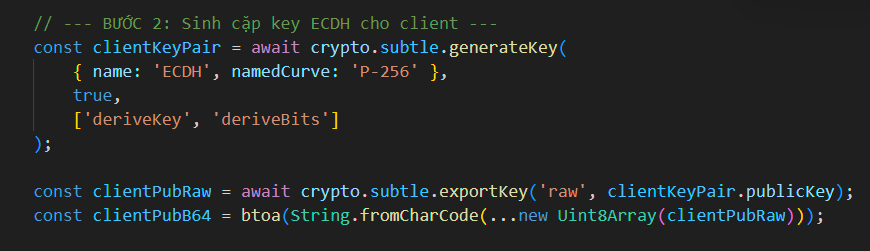
## 2. Encryption:

**- Nhập và ghi dữ liệu âm thanh vào database**

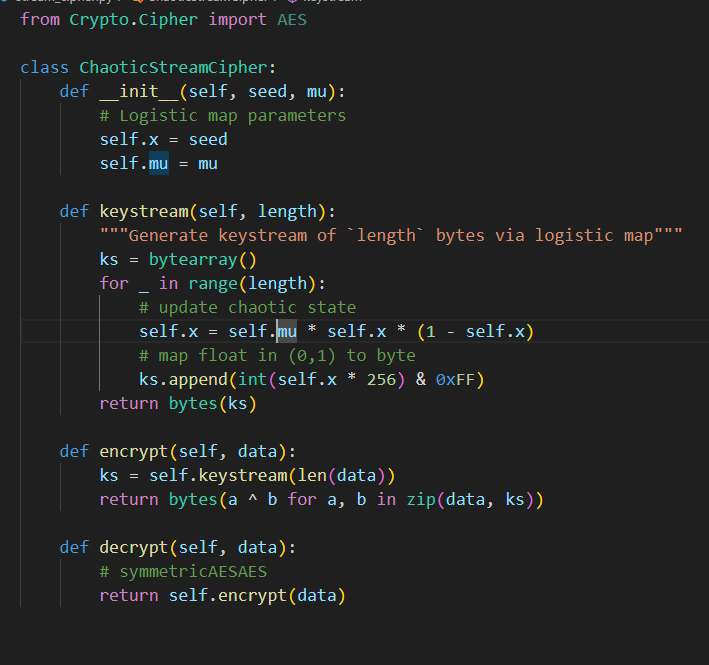


**- Khởi tạo khóa**





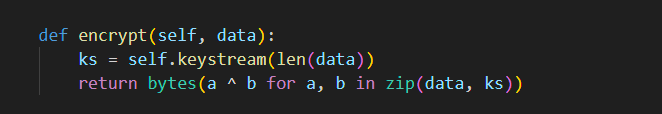
**- Tạo ra chuỗi ngẫu nhiên bằng chaotic map:**



**- Tạo Shared key**

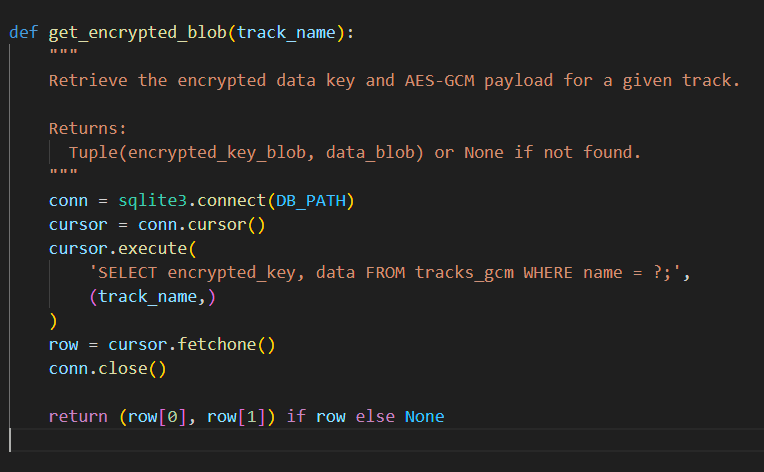


**- Encryption:**

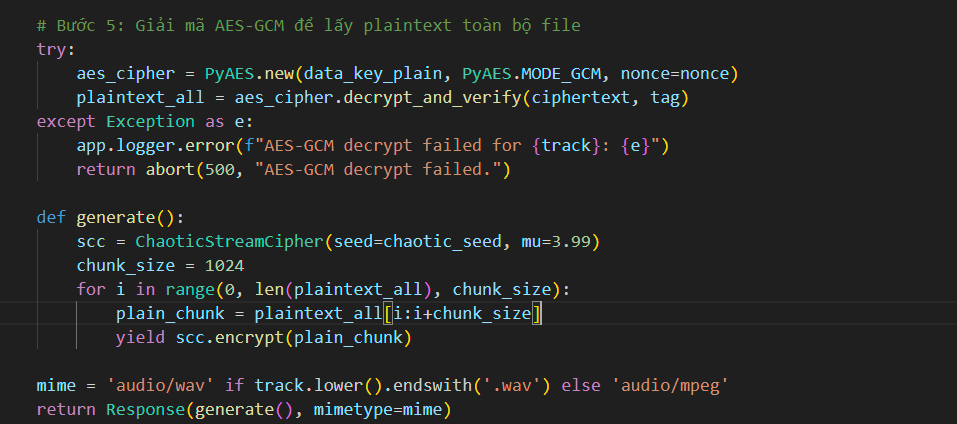


## 3. Decryption:

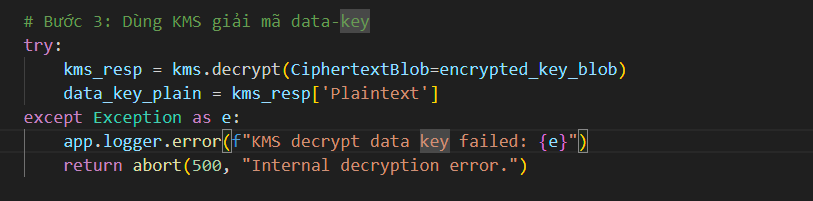
**- Đọc dữ liệu đầu vào:**



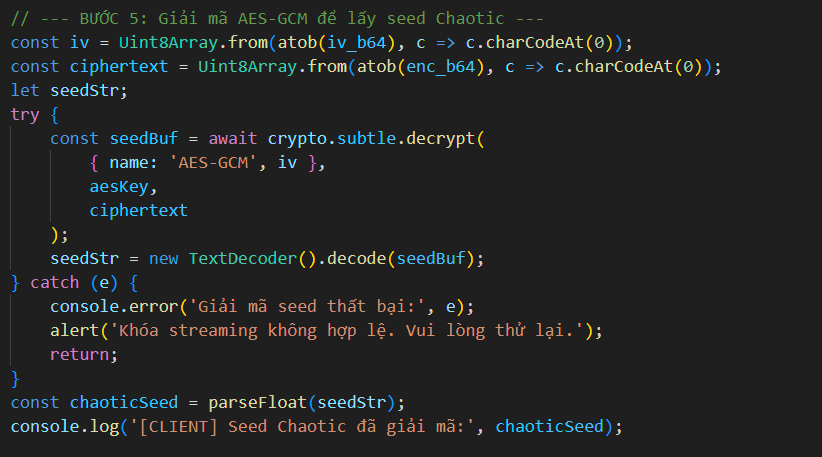
- Giải mã khóa AES



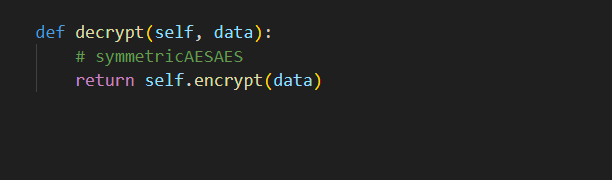
**- Giải mã DEK**



- **Giải mã Khóa Chaotic (Seed):**



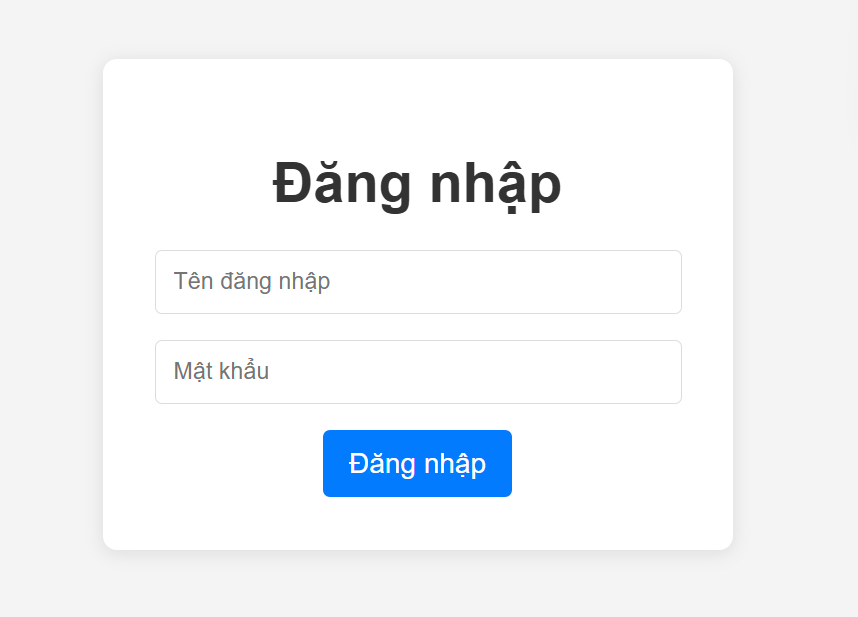
**- Decryption:**



# IV. Thực nghiệm và kết quả:

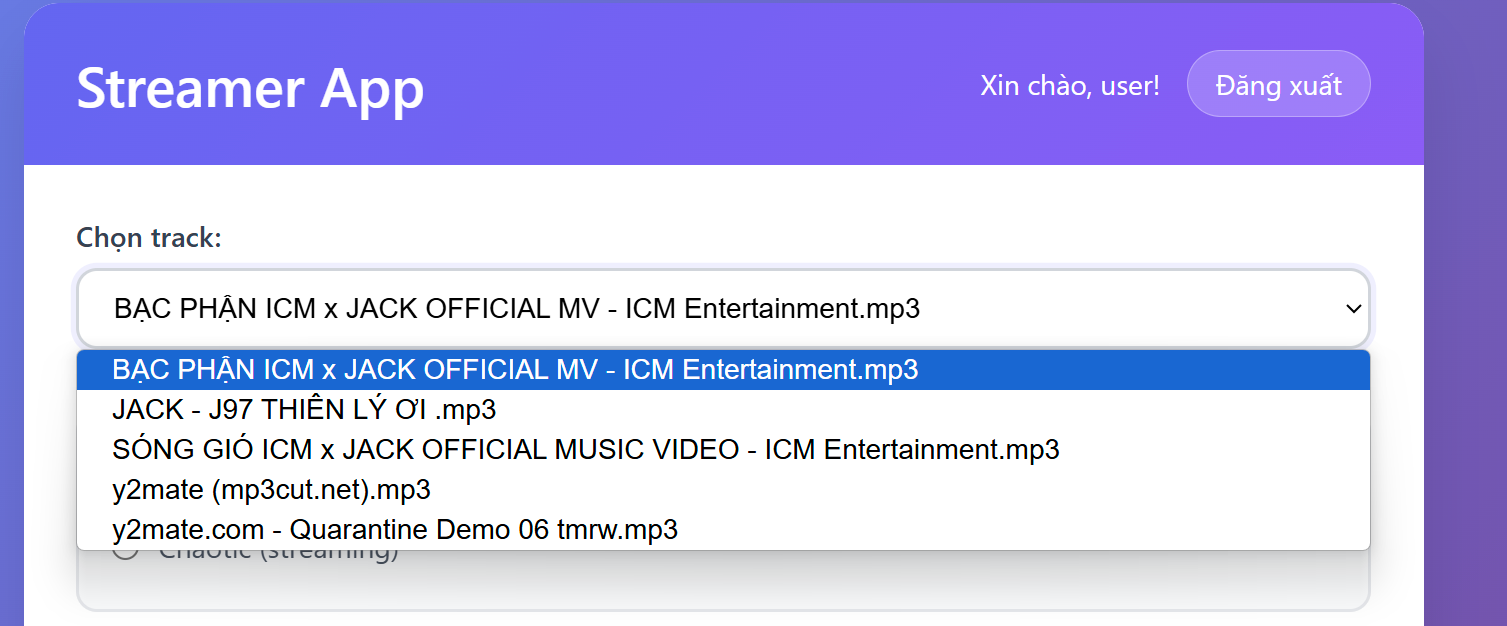
Thực hiện thực nghiệm tại file python server.py

Khi chạy server các file âm thanh gốc sẽ được mã hóa và đưa vào database

  
Yêu cầu người dung đăng nhập để sử dụng:

username=user và password:password123

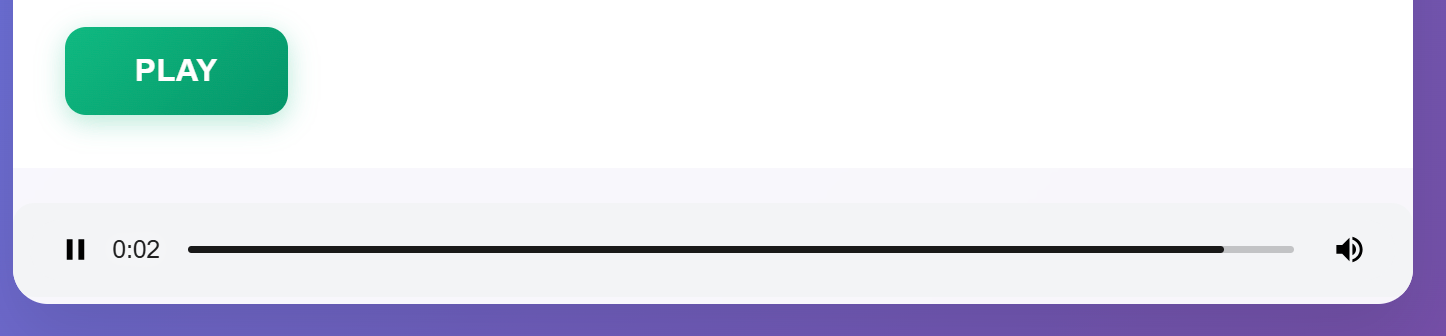
Nếu đăng nhập Thành công người dùng sẽ có thể chọn bài hát mình mong muốn



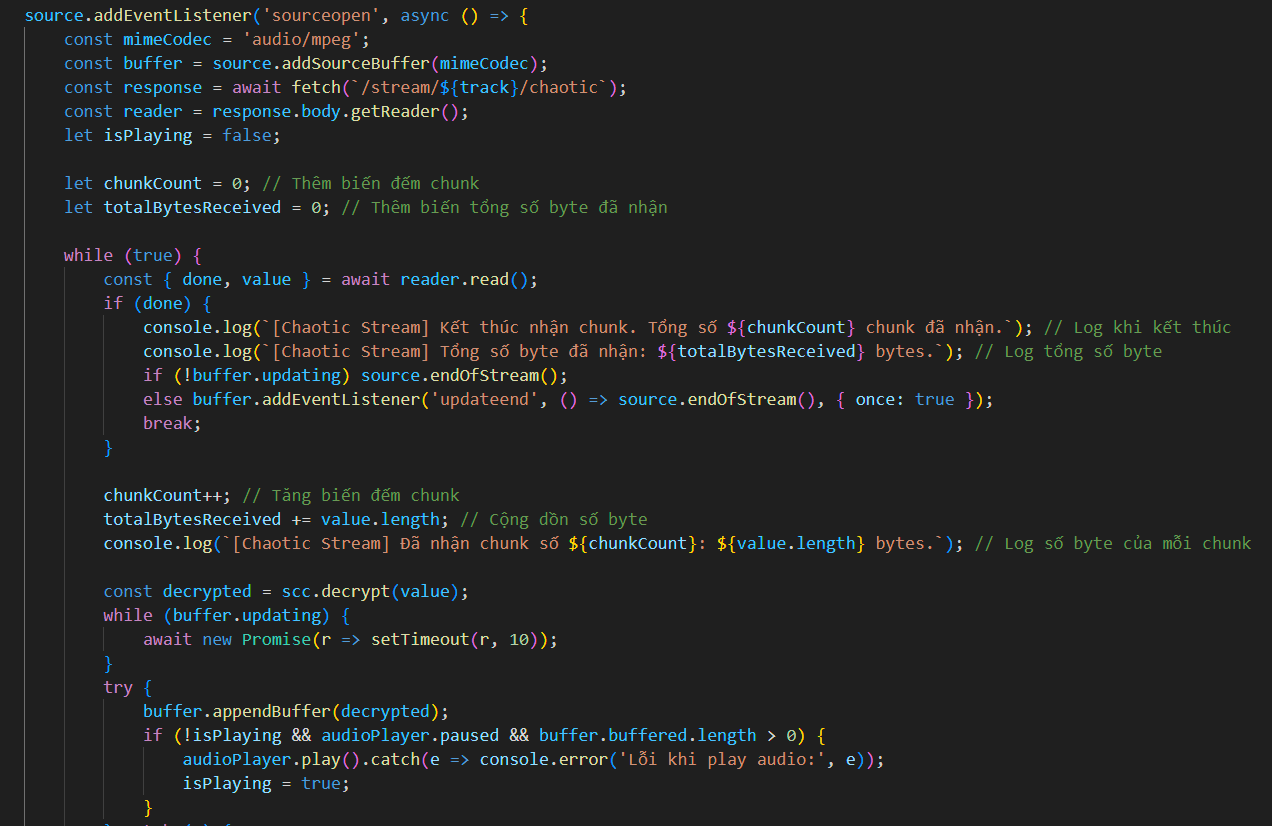
Khi chọn bài hát mong muốn, quá trình chuẩn bị sẽ bắt đầu, quá trình này sẽ mất một khoản thời gian và khi hoàn thành sẽ thông báo cho người dùng:

Khi hoàn tất, một key sẽ được cung cấp cho người dùng thông qua một bảng trong cơ sở dữ liệu (key chỉ được dùng 1 lần)

Ngay sau đó quá trình giải mã sẽ thực hiện để cung cấp bài hát người dùng đến thông qua key.



Sau khi mã hoá quá trình phát nhạc sẽ được gọi, từng chunk sẽ được truyền đến người dùng thông qua Mediasource



# V. Tổng kết và công việc tương lai:

## 1. Tổng kết:

- Tạo ra nền tảng đơn giản cung cấp dịch vụ nghe nhạc cho người dùng

- Mã hóa và giải mã thành công, thu về biểu đồ như mong đợi

- Đảm bảo dữ liệu được đưa đến người dùng như mong muốn, không thể bị đánh cắp bởi attacker

- Toàn bộ dữ liệu trước khi được đưa vào lưu trữ đều được mã hoá

## 2. Công việc tương lai:

- Nghiên cứu và ứng dụng các thuật toán mã hóa mới như mã hóa không mất dữ liệu, mã hóa trích xuất đặc trưng và nhận diện giọng nói để nâng cao chất lượng và độ an toàn.

- Phát triển các giải pháp mã hóa dữ liệu âm thanh phù hợp với các định dạng file khác nhau như mp3, wav, m4a.

- Tích hợp mã hóa âm thanh vào các nền tảng phổ biến như YouTube, Spotify, SoundCloud để tối ưu hóa trải nghiệm người dùng.

- Nghiên cứu các giải pháp mã hóa âm thanh trực tuyến và trên thiết bị di động nhằm bảo vệ quyền riêng tư và bản quyền.

- Xây dự hệ thống mang tính trực tuyến

- Cung cấp thêm các dịch vụ cần thiết cho các bên liên quan

- Nâng cấp thuật toán để tối ưu thời gian thực thi