TRƯỜNG ĐẠI HỌC LÂM NGHIỆP

**KHOA CƠ ĐIỆN CÔNG TRÌNH**



**BÁO CÁO MÔN HỌC:**

**CÁC HỆ THỐNG ĐA PHƯƠNG TIỆN**

**ĐỀ TÀI:**

**CHƯƠNG TRÌNH XỬ LÝ AUDIO MP3 (HIỂN THỊ, ĐỌC GHI METADATA, CẮT, CHUYỂN ĐỔI ĐỊNH DẠNG)**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | Trần Xuân Hòa |
| **Sinh viên thực hiện:** | Nguyễn Khả Đăng |
| **Mã sinh viên:** | 2041070501 |
| **Lớp:** | K65-HTTT |

**Hà Nội, 2024**

**Mục Lục**

[CHƯƠNG 1 TỔNG QUAN CÁC VẤN ĐỀ NGHIÊN CỨU 1](#_Toc167350909)

[1.1 Đặt vấn đề 1](#_Toc167350910)

[1.2 Mục tiêu 2](#_Toc167350911)

[1.2.1 Mục tiêu tổng quát 2](#_Toc167350912)

[1.2.2 Mục tiêu cụ thể 2](#_Toc167350913)

[1.3 Đối tượng và phạm vi nghiên cứu 3](#_Toc167350914)

[1.3.1 Đối tượng nghiên cứu 3](#_Toc167350915)

[1.3.2 Phạm vi nghiên cứu 3](#_Toc167350916)

[1.4 Phương pháp nghiên cứu 4](#_Toc167350917)

[CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 5](#_Toc167350918)

[2.1 Mô tả bài toán 5](#_Toc167350919)

[2.2 Lựa chọn giải thuật 6](#_Toc167350920)

[2.3 Lựa chọn công nghệ sử dụng xây dựng hệ thống 7](#_Toc167350921)

[2.3.1 Front End 8](#_Toc167350922)

[2.3.2 Back End 10](#_Toc167350923)

[CHƯƠNG 3 PHÁT TRIỂN ỨNG DỤNG 12](#_Toc167350924)

[3.1 Phân tích hệ thống 12](#_Toc167350925)

[3.1.1 Xác định các tác nhân 12](#_Toc167350926)

[3.1.2 Xác định các use case chính 12](#_Toc167350927)

[3.1.3 Biểu đồ use case hệ thống 13](#_Toc167350928)

[3.2 Kịch bản use case 13](#_Toc167350929)

[3.2.1 Hiển thị biểu đồ sóng âm 13](#_Toc167350930)

[3.2.2 Cắt file mp3 14](#_Toc167350931)

[3.2.3 Chuyển đổi định dạng 14](#_Toc167350932)

[3.2.4 Metadata 15](#_Toc167350933)

[3.3 Xây dựng biểu đồ hoạt động 16](#_Toc167350934)

[3.3.1 Hiển thị biểu đồ sóng âm 16](#_Toc167350935)

[3.3.2 Cắt file mp3 16](#_Toc167350936)

[3.3.3 Chuyển đổi định dạng 17](#_Toc167350937)

[3.3.4 Metadata 17](#_Toc167350938)

[CHƯƠNG 4 LẬP TRÌNH PHÁT TRIỂN 18](#_Toc167350939)

[4.1 Kết quả - Các phương thức xử lý chức năng 18](#_Toc167350940)

[4.1.1 Hiển thị biểu đồ sóng âm 18](#_Toc167350941)

[4.1.2 Cắt file mp3 18](#_Toc167350942)

[4.1.3 Chuyển đổi định dạng 20](#_Toc167350943)

[4.1.4 Hiển thị, đọc, ghi metadata 20](#_Toc167350944)

[4.2 Kết quả - Giao diện lập trình 23](#_Toc167350945)

[CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 24](#_Toc167350946)

[5.1 Kết luận 24](#_Toc167350947)

[5.2 Hướng phát triển 24](#_Toc167350948)

**Danh Mục Ảnh**

[Hình 1 Sơ đồ use case tổng quát 13](#_Toc167350949)

[Hình 2 Biểu đồ hoạt động hiển thị biểu đồ sóng âm 16](#_Toc167350950)

[Hình 3 Biểu đồ hoạt động cắt file mp3 16](#_Toc167350951)

[Hình 4 Biểu đồ hoạt động chuyển đổi định dạng 17](#_Toc167350952)

[Hình 5 Biểu đồ hoạt động metadata 17](#_Toc167350953)

[Hình 6 Kết quả giao diện lập trình 23](#_Toc167350954)

**Danh Mục Bảng**

[Bảng 1 Bảng xác định tác nhân 12](#_Toc167350963)

[Bảng 2 Bảng xác định usecase chính 12](#_Toc167350964)

[Bảng 3 Kịch bản use case hiển thị biểu đồ sóng âm 13](#_Toc167350965)

[Bảng 4 Kịch bản use case cắt file mp3 14](#_Toc167350966)

[Bảng 5 Kịch bản use case chuyển đổi định dạng 14](#_Toc167350967)

[Bảng 6 Kịch bản use case metadata 15](#_Toc167350968)

# TỔNG QUAN CÁC VẤN ĐỀ NGHIÊN CỨU

## Đặt vấn đề

Việc phát triển một chương trình xử lý audio MP3 có tính năng hiển thị, đọc ghi metadata, cắt, và chuyển đổi định dạng là cần thiết vì các lý do sau đây:

1. Nhu cầu quản lý và chỉnh sửa nhạc cá nhân:

- Với sự phổ biến của các thiết bị lưu trữ nhạc kỹ thuật số, người dùng thường xuyên cần quản lý và chỉnh sửa các tập tin nhạc MP3 của họ. Khả năng đọc và ghi metadata giúp người dùng tổ chức và tìm kiếm các bản nhạc dễ dàng hơn.

- Việc cắt các đoạn nhạc hoặc chuyển đổi định dạng giúp người dùng tạo ra các đoạn nhạc phù hợp với nhu cầu sử dụng khác nhau, như tạo nhạc chuông điện thoại hoặc chuẩn bị file cho các trình phát khác nhau.

2. Hỗ trợ công việc chuyên môn trong lĩnh vực truyền thông và giải trí:

- Các chuyên gia âm thanh và biên tập viên cần các công cụ mạnh mẽ để chỉnh sửa và quản lý các tập tin âm thanh. Một chương trình đa chức năng sẽ giúp tăng hiệu suất làm việc và đảm bảo chất lượng sản phẩm đầu ra.

- Khả năng chuyển đổi định dạng là thiết yếu trong việc chuẩn bị các tệp âm thanh cho các nền tảng khác nhau, từ phát thanh truyền hình đến các dịch vụ phát trực tuyến.

3. Tối ưu hóa và bảo trì hệ thống dữ liệu âm thanh:

- Các thư viện nhạc lớn cần được quản lý hiệu quả để tránh lãng phí dung lượng lưu trữ và đảm bảo rằng dữ liệu được sắp xếp hợp lý. Một chương trình có thể giúp tự động hóa quá trình này, giảm thiểu công việc thủ công và lỗi con người.

- Chuyển đổi định dạng tệp giúp tối ưu hóa không gian lưu trữ và cải thiện hiệu suất truy cập dữ liệu âm thanh trên các thiết bị có tài nguyên hạn chế.

4. Phát triển kỹ năng lập trình và kiến thức kỹ thuật:

- Việc thực hiện dự án này không chỉ cung cấp cho sinh viên những kỹ năng thực tiễn trong lập trình và xử lý âm thanh, mà còn giúp họ hiểu sâu hơn về các chuẩn định dạng âm thanh và các công nghệ liên quan.

- Đây cũng là cơ hội để sinh viên khám phá và áp dụng các thư viện và công cụ lập trình hiện đại, như ID3v2 cho metadata, và các thư viện xử lý âm thanh như SOX và LAME.

5. Tăng cường khả năng tích hợp và mở rộng ứng dụng:

- Một chương trình xử lý audio MP3 có thể dễ dàng tích hợp với các hệ thống lớn hơn như phần mềm chỉnh sửa video, hệ thống quản lý nội dung (CMS), hoặc các ứng dụng di động, mở rộng khả năng sử dụng của nó.

- Khả năng mở rộng và tùy biến là yếu tố quan trọng trong việc phát triển các ứng dụng phần mềm hiện đại, đảm bảo rằng chương trình có thể đáp ứng nhu cầu ngày càng tăng và đa dạng của người dùng.

Từ những lý do trên, có thể thấy rằng việc phát triển một chương trình xử lý audio MP3 không chỉ đáp ứng nhu cầu cá nhân và chuyên nghiệp mà còn đóng góp vào sự phát triển công nghệ và cải thiện hiệu quả công việc trong nhiều lĩnh vực khác nhau.

## Mục tiêu

### Mục tiêu tổng quát

Phát triển một chương trình xử lý audio MP3 đa chức năng với các tính năng hiển thị, đọc ghi metadata, cắt và chuyển đổi định dạng nhằm đáp ứng nhu cầu quản lý, chỉnh sửa và tối ưu hóa tệp âm thanh của người dùng cá nhân và chuyên nghiệp. Chương trình sẽ giúp nâng cao hiệu suất làm việc, đảm bảo chất lượng âm thanh và tích hợp dễ dàng vào các hệ thống lớn hơn.

### Mục tiêu cụ thể

1. Hiển thị thông tin tệp MP3:

- Xây dựng giao diện người dùng trực quan để hiển thị các thông tin chi tiết của tệp MP3 như tên bài hát, nghệ sĩ, album, thể loại, và thời lượng.

- Hiển thị biểu đồ sóng âm và các thuộc tính kỹ thuật như bitrate, sample rate, và kênh âm thanh.

2. Đọc và ghi metadata:

- Triển khai chức năng đọc metadata từ các tệp MP3, hỗ trợ các phiên bản ID3 (ID3v1, ID3v2).

- Cho phép người dùng chỉnh sửa và cập nhật metadata, bao gồm thông tin bài hát, nghệ sĩ, album, và ảnh bìa.

- Tích hợp khả năng lưu các thay đổi vào tệp MP3 mà không làm ảnh hưởng đến chất lượng âm thanh.

3. Cắt tệp âm thanh:

- Phát triển tính năng cho phép người dùng chọn và cắt các đoạn nhạc từ tệp MP3.

- Cung cấp công cụ xem trước đoạn nhạc được cắt để đảm bảo tính chính xác và đáp ứng nhu cầu sử dụng của người dùng.

4. Chuyển đổi định dạng tệp âm thanh:

- Hỗ trợ chuyển đổi tệp MP3 sang các định dạng âm thanh phổ biến khác như WAV, AAC, FLAC, OGG.

- Đảm bảo quá trình chuyển đổi giữ nguyên chất lượng âm thanh gốc hoặc cho phép người dùng tùy chọn các thiết lập để tối ưu hóa kích thước và chất lượng tệp đầu ra.

5. Tối ưu hóa và quản lý tệp âm thanh:

- Cung cấp tính năng tối ưu hóa dung lượng và chất lượng tệp MP3, bao gồm nén tệp và điều chỉnh bitrate.

- Hỗ trợ người dùng quản lý thư viện nhạc một cách hiệu quả, bao gồm các công cụ tìm kiếm, sắp xếp và phân loại tệp âm thanh dựa trên metadata.

6. Tích hợp với các hệ thống và ứng dụng khác:

- Thiết kế chương trình với khả năng tích hợp API để kết nối với các ứng dụng quản lý nội dung, phần mềm chỉnh sửa video, hoặc các nền tảng phát trực tuyến.

- Hỗ trợ các plugin và tiện ích mở rộng để chương trình có thể mở rộng chức năng và tích hợp dễ dàng vào môi trường làm việc của người dùng.

7. Phát triển và thử nghiệm:

- Tiến hành thử nghiệm chương trình trên nhiều hệ điều hành và thiết bị khác nhau để đảm bảo tính ổn định và hiệu suất cao.

- Thu thập phản hồi từ người dùng để cải tiến và cập nhật chương trình, đáp ứng tốt hơn nhu cầu thực tế.

## Đối tượng và phạm vi nghiên cứu

### Đối tượng nghiên cứu

Tệp MP3 và các định dạng âm thanh khác: Tìm hiểu về cấu trúc tệp MP3, các loại metadata như ID3v1 và ID3v2, các thuật toán nén âm thanh, và các định dạng âm thanh khác như WAV, AAC, FLAC, OGG.

Các công cụ và thư viện xử lý âm thanh: Khảo sát và sử dụng các công cụ và thư viện phần mềm phổ biến để xử lý âm thanh như FFMPEG, Librosa, Pydub, và Mutagen. Tìm hiểu cách thức chúng hỗ trợ trong việc hiển thị, chỉnh sửa, cắt ghép, và chuyển đổi định dạng tệp âm thanh.

### Phạm vi nghiên cứu

1. Tính năng chương trình:

- Hiển thị thông tin và metadata: Chương trình sẽ hiển thị thông tin cơ bản và chi tiết của tệp MP3 bao gồm các thông tin metadata và các thuộc tính kỹ thuật.

- Chỉnh sửa metadata: Tích hợp các chức năng cho phép đọc, sửa đổi và lưu trữ lại metadata trong các tệp MP3.

- Cắt ghép tệp âm thanh: Phát triển các công cụ để cắt và ghép các đoạn âm thanh một cách chính xác và dễ dàng.

- Chuyển đổi định dạng: Hỗ trợ chuyển đổi tệp MP3 sang các định dạng âm thanh phổ biến khác.

2. Phạm vi kỹ thuật:

- Thư viện và công cụ: Áp dụng các thư viện và công cụ mã nguồn mở như FFMPEG, Librosa, Pydub, và Mutagen để phát triển các tính năng của chương trình.

- Giao diện người dùng: Phát triển giao diện người dùng trên nền tảng phổ biến nhất cụ thể là website.

3. Giới hạn nghiên cứu:

- Chất lượng âm thanh: Đảm bảo chương trình duy trì chất lượng âm thanh gốc sau khi xử lý, cắt ghép và chuyển đổi định dạng.

- Hiệu suất: Đảm bảo chương trình hoạt động ổn định và hiệu quả trên các cấu hình phần cứng khác nhau.

## Phương pháp nghiên cứu

Kết hợp nghiên cứu lý thuyết và thực hành:

* Phương pháp lý thuyết: Đọc và tìm hiểu tài liệu, kiến thức liên quan đến các đọc metadata và chuyển đổi định dạng audio.
* Phương pháp thực hành: Thực hiện xây dựng hệ thống và đánh giá tính hiệu quả trong thực tế.

Các phương pháp nghiên cứu khác:

* Phương pháp chuyên gia: Tham khảo ý kiến của giảng viên hướng dẫn Trần Xuân Hòa và các giảng viên trong ngành Hệ thống thông tin.

# CƠ SỞ LÝ THUYẾT

## Mô tả bài toán

Bài toán đặt ra là phát triển một chương trình xử lý audio MP3 với các chức năng hiển thị thông tin, đọc ghi metadata, cắt ghép, và chuyển đổi định dạng. Mục tiêu là tạo ra một công cụ hữu ích cho việc chỉnh sửa tệp âm thanh MP3, đáp ứng nhu cầu của các người dùng cá nhân:

1. Hiển thị thông tin tệp MP3:

- Vấn đề: Người dùng thường cần biết các thông tin chi tiết về tệp MP3 như tên bài hát, nghệ sĩ, album, thể loại, thời lượng, bitrate, và các thông tin kỹ thuật khác. Hiện tại, các công cụ có sẵn trên thị trường có thể không hiển thị đầy đủ hoặc dễ sử dụng.

- Giải pháp: Phát triển giao diện người dùng trực quan để hiển thị tất cả các thông tin cần thiết một cách rõ ràng và dễ hiểu.

2. Đọc và ghi metadata:

- Vấn đề: Metadata giúp người dùng tổ chức và tìm kiếm các tệp MP3. Tuy nhiên, việc chỉnh sửa metadata có thể phức tạp và đòi hỏi công cụ đặc biệt.

- Giải pháp: Triển khai chức năng đọc và ghi metadata, hỗ trợ các phiên bản ID3v1 và ID3v2. Cho phép người dùng chỉnh sửa và cập nhật metadata một cách dễ dàng và lưu lại các thay đổi mà không ảnh hưởng đến chất lượng tệp âm thanh.

3. Cắt ghép tệp âm thanh:

- Vấn đề: Người dùng cần cắt các đoạn nhạc để tạo nhạc chuông, trích đoạn, hoặc chỉnh sửa các phần không mong muốn. Việc này đòi hỏi một công cụ cắt ghép chính xác và dễ sử dụng.

- Giải pháp: Phát triển tính năng cắt ghép tệp âm thanh, cho phép người dùng chọn đoạn cần cắt, xem trước và lưu lại đoạn cắt một cách chính xác.

4. Chuyển đổi định dạng tệp âm thanh:

- Vấn đề: Người dùng có thể cần chuyển đổi tệp MP3 sang các định dạng khác để tương thích với các thiết bị hoặc phần mềm khác nhau. Chuyển đổi định dạng cần đảm bảo chất lượng âm thanh và hiệu suất xử lý.

- Giải pháp: Hỗ trợ chuyển đổi tệp MP3 sang các định dạng âm thanh phổ biến như WAV, AAC, FLAC, OGG. Cung cấp tùy chọn cho người dùng để tùy chỉnh các thông số chuyển đổi như bitrate và sample rate.

5. Tích hợp với các hệ thống và ứng dụng khác:

- Vấn đề: Chương trình cần có khả năng tích hợp với các hệ thống lớn hơn như phần mềm chỉnh sửa video, hệ thống quản lý nội dung (CMS), hoặc các ứng dụng di động.

- Giải pháp: Thiết kế chương trình với khả năng tích hợp API để kết nối với các ứng dụng khác. Hỗ trợ các plugin và tiện ích mở rộng để mở rộng chức năng và đáp ứng nhu cầu sử dụng đa dạng.

## Lựa chọn giải thuật

1. Đọc và hiển thị thông tin metadata:

- Các thư viện hỗ trợ: MusicBrainz Picard, jsmediatags.js, TagLib

Đánh giá:

- jsmediatags.js: Dễ sử dụng, hỗ trợ tốt cho việc đọc metadata từ các tệp MP3 trên trình duyệt. Khả năng tương thích cao và có thể xử lý nhiều định dạng metadata khác nhau như ID3v1, ID3v2.

- MusicBrainz Picard: Một công cụ mạnh mẽ với cơ sở dữ liệu lớn, tuy nhiên phức tạp hơn và có thể không cần thiết cho các ứng dụng đơn giản.

- TagLib: Thư viện C++ mạnh mẽ, hỗ trợ nhiều định dạng tệp âm thanh và metadata, nhưng đòi hỏi kiến thức lập trình C++.

Lựa chọn: jsmediatags.js là lựa chọn phù hợp cho việc đọc và hiển thị metadata trong ứng dụng web, dễ tích hợp và đủ mạnh mẽ cho nhu cầu của bạn.

2. Hiển thị MP3 dưới dạng tần số:

- Các thư viện hỗ trợ: Web Audio API, P5.js (sound library), wavesurfer.js

Đánh giá:

- wavesurfer.js: Dễ sử dụng, chuyên dụng cho việc hiển thị sóng âm thanh, hỗ trợ nhiều tính năng hữu ích như zoom, cắt đoạn.

- Web Audio API: Mạnh mẽ và linh hoạt, nhưng phức tạp hơn để triển khai và yêu cầu nhiều mã tùy chỉnh.

- P5.js (sound library): Hỗ trợ tốt cho việc hiển thị và xử lý âm thanh, nhưng không chuyên dụng cho việc hiển thị sóng âm thanh như wavesurfer.js.

Lựa chọn: wavesurfer.js là lựa chọn tốt nhất cho việc hiển thị sóng âm thanh, dễ sử dụng và tích hợp, cung cấp các tính năng hữu ích.

3. Ghi lại thông tin metadata:

- Các thư viện hỗ trợ: getID3, eyed3 (Python), TagLib

Đánh giá:

- getID3: Hỗ trợ nhiều định dạng âm thanh và metadata, dễ sử dụng với PHP.

- eyed3: Một thư viện mạnh mẽ dành cho Python, tốt cho việc xử lý metadata nhưng yêu cầu kiến thức về Python.

- TagLib: Đã được đề cập ở trên, rất mạnh mẽ nhưng phức tạp hơn để sử dụng.

Lựa chọn: getID3 là lựa chọn hợp lý, đặc biệt khi sử dụng PHP. Nó hỗ trợ tốt và dễ tích hợp.

4. Chuyển đổi định dạng:

- Các thư viện hỗ trợ: FFMPEG, Audacity, lame encoding mp3

Đánh giá:

- lame encoding mp3: Rất tốt cho việc mã hóa MP3, được tối ưu hóa cho chất lượng âm thanh cao.

- FFMPEG: Công cụ mạnh mẽ cho mọi loại chuyển đổi âm thanh và video, nhưng có thể phức tạp hơn để cấu hình.

- Audacity libraries: Mạnh mẽ và thân thiện với người dùng, nhưng chủ yếu là một ứng dụng độc lập, khó tích hợp vào các dự án tùy chỉnh.

Lựa chọn: lame encoding mp3 là lựa chọn tối ưu cho việc mã hóa MP3, đặc biệt khi yêu cầu về chất lượng âm thanh cao.

5. Cắt tệp âm thanh:

- Các thư viện hỗ trợ: FFMPEG, Pydub, Sound eXchange (SoX)

Đánh giá:

- SoX: Rất mạnh mẽ và linh hoạt, hỗ trợ nhiều chức năng xử lý âm thanh.

- FFMPEG: Cũng rất mạnh mẽ, hỗ trợ nhiều loại tệp và chức năng, nhưng có thể phức tạp để cấu hình và sử dụng.

- Pydub: Thư viện Python đơn giản, dễ sử dụng nhưng không mạnh mẽ bằng SoX.

Lựa chọn: SoX là lựa chọn tốt nhất cho việc cắt tệp âm thanh, mạnh mẽ và linh hoạt.

## Lựa chọn công nghệ sử dụng xây dựng hệ thống

Với mục tiêu ưu tiên sự tiện lợi cho người dùng và tiết kiệm chi phí duy trì và phát triển, hệ thống sẽ được xây dựng dưới dạng Website, cụ thể:

* Khả năng truy cập trên nhiều nền tảng: Website có thể truy cập từ mọi thiết bị có kết nối internet, bao gồm cả máy tính, điện thoại di động và máy tính bảng. Điều này làm tăng sự tiện lợi cho người dùng, không cần phải cài đặt ứng dụng riêng biệt trên từng thiết bị.
* Dễ dàng tiếp cận và chia sẻ: Website có thể truy cập ngay lập tức thông qua trình duyệt web, không cần phải tải xuống và cài đặt từ cửa hàng ứng dụng. Điều này làm giảm rào cản cho người dùng mới và tạo thuận lợi cho việc chia sẻ thông tin với người khác.
* Chi phí phát triển và duy trì thấp hơn: Xây dựng và duy trì một website thường ít tốn kém hơn so với phát triển và duy trì một ứng dụng. Không cần phải lo lắng về các vấn đề như tương thích với nhiều hệ điều hành và cập nhật định kỳ cho ứng dụng trên nhiều nền tảng.
* Thu thập dữ liệu dễ dàng hơn: Website có thể tích hợp các biểu đồ, bản đồ và công cụ tương tác để hiển thị dữ liệu một cách trực quan. Việc thu thập và phân tích dữ liệu từ người dùng cũng có thể được thực hiện một cách dễ dàng hơn thông qua giao diện web.
* Cập nhật nhanh chóng: Website có thể được cập nhật ngay lập tức mà không cần phải chờ đợi quá trình duyệt ứng dụng từ cửa hàng ứng dụng. Điều này cho phép bạn nhanh chóng cập nhật thông tin về cháy rừng và các tình hình khẩn cấp.

### Front End

#### HTML5

HTML là viết tắt của HyperText Markup Language, là một ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo ra các trang web. HTML được sử dụng để cấu trúc nội dung của một trang web, bao gồm văn bản, hình ảnh, âm thanh, video và các yếu tố khác.

HTML 5 là phiên bản mới nhất của HTML, được phát hành vào năm 2014. HTML 5 bổ sung thêm nhiều tính năng mới cho HTML, bao gồm:

* Hỗ trợ các định dạng phương tiện mới, chẳng hạn như video HTML5 và âm thanh HTML5.
* Hỗ trợ các canvas và SVG, cho phép tạo các hình ảnh và đồ họa động.
* Hỗ trợ các API mới, chẳng hạn như API vị trí và API bảo mật.

#### CSS3

CSS là viết tắt của Cascading Style Sheets, là một ngôn ngữ định dạng văn bản được sử dụng để định dạng nội dung của các trang web. CSS được sử dụng để kiểm soát màu sắc, kích thước, phông chữ, bố cục và các thuộc tính khác của nội dung HTML.

CSS3 bổ sung thêm nhiều tính năng mới cho CSS, bao gồm:

* Hỗ trợ các hiệu ứng hình ảnh và chuyển động mới, chẳng hạn như box-shadow, border-radius và animation.
* Hỗ trợ các định dạng văn bản mới, chẳng hạn như text-shadow và font-smoothing.
* Hỗ trợ các bố cục mới, chẳng hạn như flexbox và grid.

#### Bootstrap

Bootstrap là một framework mã nguồn mở được sử dụng để xây dựng các trang web và ứng dụng web. Bootstrap cung cấp các thành phần và lớp CSS được định sẵn để giúp việc thiết kế và phát triển các trang web trở nên nhanh chóng và dễ dàng hơn.

Các tính năng chính của Bootstrap

* Bố cục: Bootstrap cung cấp các lớp lưới và các thành phần để tạo bố cục trang web đáp ứng.
* Thiết kế: Bootstrap cung cấp các thành phần và lớp CSS để tạo các trang web có thiết kế đẹp và hấp dẫn.
* Tương tác: Bootstrap cung cấp các thành phần và lớp CSS để tạo các trang web tương tác và đáp ứng.

Lợi ích của việc sử dụng Bootstrap

* Tiết kiệm thời gian và công sức: Bootstrap cung cấp các thành phần và lớp CSS được định sẵn, giúp việc thiết kế và phát triển các trang web trở nên nhanh chóng và dễ dàng hơn.
* Tăng khả năng tái sử dụng: Các thành phần và lớp CSS của Bootstrap có thể được sử dụng lại trong nhiều trang web khác nhau, giúp giảm thời gian và công sức phát triển.
* Tăng khả năng tương thích: Bootstrap được thiết kế để tương thích với nhiều trình duyệt và thiết bị, giúp đảm bảo rằng các trang web của bạn sẽ trông đẹp và hoạt động tốt trên tất cả các thiết bị.

#### JavaScript

JavaScript là một ngôn ngữ lập trình kịch bản hướng đối tượng được sử dụng để tạo các trang web và ứng dụng web tương tác. JavaScript được sử dụng để tạo các hiệu ứng động, tương tác với người dùng và thêm chức năng cho các trang web.

JavaScript được tạo ra bởi Brendan Eich vào năm 1995. JavaScript được ban đầu phát triển như một phần của Netscape Navigator, nhưng sau đó đã được chấp nhận bởi các trình duyệt web khác. JavaScript được phát hành lần đầu tiên với tên LiveScript, nhưng sau đó được đổi tên thành JavaScript vào năm 1996.

Lợi ích của việc sử dụng JavaScript

* Tạo các trang web và ứng dụng web tương tác: JavaScript có thể được sử dụng để tạo các trang web và ứng dụng web tương tác, cung cấp cho người dùng trải nghiệm tốt hơn.
* Tiết kiệm thời gian và công sức: JavaScript có thể được sử dụng để tạo các trang web và ứng dụng web nhanh chóng và dễ dàng hơn.
* Tăng khả năng tương thích: JavaScript tương thích với hầu hết các trình duyệt web, giúp đảm bảo rằng các trang web và ứng dụng web của bạn sẽ trông đẹp và hoạt động tốt trên tất cả các thiết bị.

#### Jquery

jQuery là một thư viện JavaScript mã nguồn mở được sử dụng để đơn giản hóa việc sử dụng JavaScript trên các trang web. jQuery cung cấp các hàm và phương thức sẵn có để thao tác với DOM (Document Object Model), xử lý sự kiện, tạo hiệu ứng hoạt hình và giao tiếp AJAX.

jQuery được tạo ra bởi John Resig vào năm 2006. jQuery được phát hành lần đầu tiên dưới dạng một dự án mã nguồn mở trên GitHub. jQuery đã nhanh chóng trở nên phổ biến và hiện là thư viện JavaScript được sử dụng rộng rãi nhất trên thế giới.

Lợi ích của việc sử dụng jQuery

* Đơn giản hóa việc sử dụng JavaScript: jQuery cung cấp các hàm và phương thức sẵn có để thao tác với DOM, xử lý sự kiện, tạo hiệu ứng hoạt hình và giao tiếp AJAX. Điều này làm cho việc sử dụng JavaScript trở nên dễ dàng hơn và nhanh chóng hơn.
* Tăng khả năng tái sử dụng: Các hàm và phương thức của jQuery có thể được sử dụng lại trong nhiều dự án khác nhau. Điều này giúp giảm thời gian và công sức phát triển.
* Tăng khả năng tương thích: jQuery tương thích với hầu hết các trình duyệt web, giúp đảm bảo rằng các trang web và ứng dụng web của bạn sẽ trông đẹp và hoạt động tốt trên tất cả các thiết bị.

### Back End

#### Laravel

Laravel là một framework phát triển ứng dụng web mã nguồn mở được xây dựng bằng ngôn ngữ lập trình PHP. Được phát triển và duy trì bởi Taylor Otwell, Laravel đã nhanh chóng trở thành một trong những framework phổ biến nhất cho phát triển web PHP hiện đại.

Laravel được ra đời lần đầu tiên vào năm 2011 và từ đó, nó đã trải qua nhiều phiên bản cải tiến, mỗi phiên bản mang lại những tính năng và cải tiến mới. Sự quan tâm và đóng góp của cộng đồng lập trình viên đã giúp Laravel phát triển mạnh mẽ, trở thành một trong những lựa chọn hàng đầu cho việc xây dựng các ứng dụng web hiện đại.

Lợi ích của Laravel:

* Cú pháp Rõ Ràng và Dễ Đọc: Laravel sử dụng cú pháp rõ ràng và dễ đọc, giúp những người phát triển tập trung vào logic kinh doanh thay vì lo lắng về các vấn đề kỹ thuật phức tạp.
* Eloquent ORM: Laravel cung cấp Eloquent ORM, một công cụ mạnh mẽ cho việc tương tác với cơ sở dữ liệu. Nó giúp đơn giản hóa quá trình truy vấn và thao tác dữ liệu.
* Hệ Thống Routing Mạnh Mẽ: Hệ thống routing của Laravel linh hoạt và mạnh mẽ, giúp dễ dàng xác định các tuyến đường (routes) cho ứng dụng.
* Blade Template Engine: Blade là một template engine mạnh mẽ trong Laravel, giúp tạo ra các giao diện người dùng một cách dễ dàng và linh hoạt.
* Middleware và Filter: Laravel cung cấp middleware và filter giúp kiểm soát quyền truy cập, xử lý các yêu cầu trước khi chúng đến đích.
* Artisan Console: Laravel đi kèm với Artisan, một công cụ dòng lệnh giúp tạo ra các thành phần của ứng dụng một cách nhanh chóng và dễ dàng.
* Đa Dạng Cộng Đồng và Tài Liệu:
* Cộng đồng sử dụng Laravel rộng lớn, mang lại sự hỗ trợ và đóng góp tích cực.
* Laravel có tài liệu chi tiết và chất lượng, giúp người phát triển nhanh chóng học và triển khai ứng dụng.

# PHÁT TRIỂN ỨNG DỤNG

## Phân tích hệ thống

### Xác định các tác nhân

Bảng 1 Bảng xác định tác nhân

| **STT** | **Tác nhân** | **Mô tả** | **Chức năng** |
| --- | --- | --- | --- |
| 1 | User | Là người người trực tiếp sử dụng hệ thống. | Upload file mp3, chỉnh sửa metadata, chọn định dạng chuyển đổi, chọn khoảng thời gian thực hiện cắt |

### Xác định các use case chính

Bảng 2 Bảng xác định usecase chính

| **STT** | **Use case** | **Mô tả** |
| --- | --- | --- |
| 1 | Hiển thị biểu đồ sóng âm | Hệ thống sẽ thực hiện đọc file được tải lên và hiển thị lên biểu đồ sóng âm của file. |
| 2 | Cắt file | Hệ thống sẽ thực hiện cắt file được tải lên theo khoảng thời gian được người dùng chọn. |
| 3 | Chuyển đổi định dạng | Hệ thống sẽ thực hiện chuyển đổi định dạng file được tải lên thành định dạng được người dùng chọn. |
| 4 | Đọc – ghi – hiển thị metadata | Hệ thống sẽ thực hiện đọc metadata file được tải lên sau đó hiển thị lên cho người dùng và hỗ trợ ghi lại thông tin metadata cho file được tải lên theo thông tin được người dùng chỉnh sửa. |

### Biểu đồ use case hệ thống



Hình 1 Sơ đồ use case tổng quát

## Kịch bản use case

### Hiển thị biểu đồ sóng âm

Bảng 3 Kịch bản use case hiển thị biểu đồ sóng âm

| **Use case** | **Hiển thị biểu đồ sóng âm** |
| --- | --- |
| Tác nhân | Hệ thống |
| Mô tả use case | Hệ thống đọc file mp3 được upload và hiển thị biểu đồ sóng âm của file. |
| Tiền điều kiện | Chương trình lấy dữ liệu được kích hoạt sử dụng và được upload file mp3 |
| Luồng sự kiện chính | 1. Hệ thống thực hiện đọc file và hiển thị biểu đồ |
| Hậu điều kiện chính |  |
| Luồng sự kiện phụ | 1.a Hệ thống gặp lỗi khi thực hiện đọc file |

### Cắt file mp3

Bảng 4 Kịch bản use case cắt file mp3

| **Use case** | **Cắt file mp3** |
| --- | --- |
| Tác nhân | Hệ thống |
| Mô tả use case | Hệ thống thực hiện cắt file được upload theo khoảng thời gian được người dùng chọn |
| Tiền điều kiện | Chương trình lấy dữ liệu được kích hoạt sử dụng và được upload file mp3 |
| Luồng sự kiện chính | 1. Hệ thống thực hiện cắt file được upload theo khoảng thời gian được người dùng chọn |
| Hậu điều kiện chính |  |
| Luồng sự kiện phụ | 1.a Hệ thống gặp lỗi khi thực hiện đọc file  1.b Hệ thống gặp lỗi khi thực hiện cắt file |

### Chuyển đổi định dạng

Bảng 5 Kịch bản use case chuyển đổi định dạng

| **Use case** | **Chuyển đổi định dạng** |
| --- | --- |
| Tác nhân | Hệ thống |
| Mô tả use case | Hệ thống thực hiện chuyển đổi định dạng file được upload theo định đạng được người dùng chọn |
| Tiền điều kiện | Chương trình lấy dữ liệu được kích hoạt sử dụng và được upload file mp3 |
| Luồng sự kiện chính | 1. Hệ thống thực hiện chuyển đổi định dạng file được upload theo định đạng được người dùng chọn |
| Hậu điều kiện chính |  |
| Luồng sự kiện phụ | 1.a Hệ thống gặp lỗi khi thực hiện đọc file  1.b Hệ thống gặp lỗi khi thực hiện chuyển đổi định dạng |

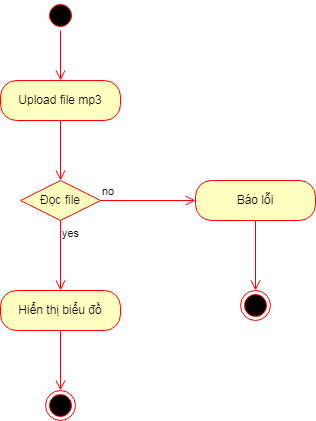
### Metadata

Bảng 6 Kịch bản use case metadata

| **Use case** | **Metadata** |
| --- | --- |
| Tác nhân | Hệ thống |
| Mô tả use case | Hệ thống đọc rồi hiển thị thông tin metadata và ghi lại thông tin mới nếu người dùng có sửa đổi |
| Tiền điều kiện | Chương trình lấy dữ liệu được kích hoạt sử dụng |
| Luồng sự kiện chính | 1. Hệ thống đọc rồi hiển thị thông tin metadata và ghi lại thông tin mới nếu người dùng có sửa đổi |
| Hậu điều kiện chính |  |
| Luồng sự kiện phụ | 1.a Hệ thống gặp lỗi khi thực hiện đọc file  1.b Hệ thống gặp lỗi khi thực hiện hiển thị  1.c Hệ thống gặp lỗi khi thực hiện lưu |

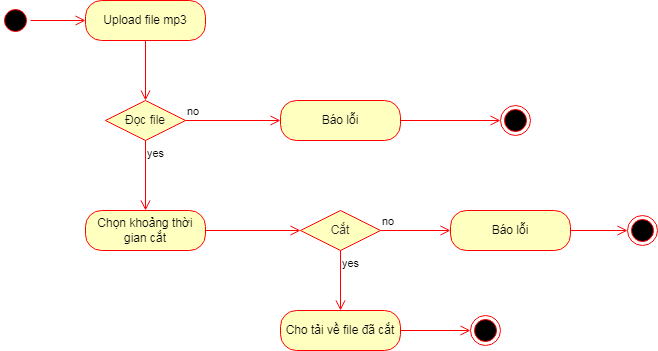
## Xây dựng biểu đồ hoạt động

### Hiển thị biểu đồ sóng âm



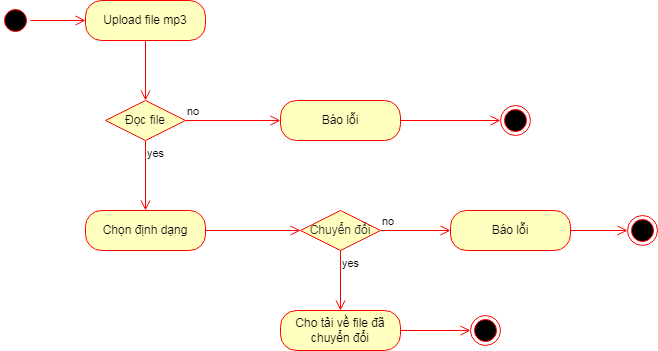
Hình 2 Biểu đồ hoạt động hiển thị biểu đồ sóng âm

### Cắt file mp3



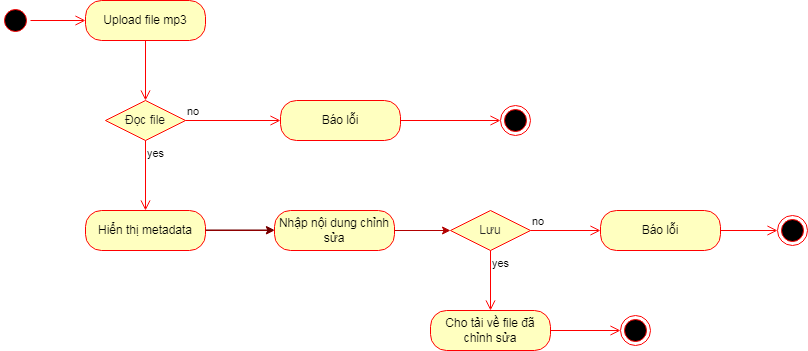
Hình 3 Biểu đồ hoạt động cắt file mp3

### Chuyển đổi định dạng



Hình 4 Biểu đồ hoạt động chuyển đổi định dạng

### Metadata



Hình 5 Biểu đồ hoạt động metadata

# LẬP TRÌNH PHÁT TRIỂN

## Kết quả - Các phương thức xử lý chức năng

### Hiển thị biểu đồ sóng âm

var wavesurfer = null;

$('#fileInput').on('change', (e) => {

    if (wavesurfer !== null) {

        wavesurfer.destroy();

    }

    $('#audio').html('');

    const file = e.target.files[0];

    if (file) {

        // khởi tạo wave

        wavesurfer = WaveSurfer.create({

            container: '#waveform',

            waveColor: 'violet',

            progressColor: 'purple',

            plugins: [

                WaveSurfer.timeline.create({

                    container: '#wave-timeline'

                })

            ]

        });

        // đọc file và hiển thị wave

        const reader = new FileReader();

        reader.onload = (event) => {

            wavesurfer.load(event.target.result);

        };

        reader.readAsDataURL(file);

    }

}

### Cắt file mp3

// trích đoạn audio

$('#extract').on('click', () => {

    var start\_time = $('#start\_time').val();

    var end\_time = $('#end\_time').val();

    if(!start\_time || !end\_time){

        alert('Vui lòng nhập khoảng thời gian để thực hiện trích xuất');

    }else{

        let total\_start = (parseInt((start\_time.split(':'))[0], 10) \* 60) + parseInt((start\_time.split(':'))[1], 10);

        let total\_end = (parseInt((end\_time.split(':'))[0], 10) \* 60) + parseInt((end\_time.split(':'))[1], 10);

        var max\_time = wavesurfer.getDuration();

        if(total\_end > max\_time){

            alert('Không hợp lệ - Tổng thời gian trích xuất đã lớn hơn thời lượng file!');

        }else if(total\_start > total\_end){

            alert('Không hợp lệ - Thời gian bắt đầu trích xuất không thể lớn hơn thời gian kết thúc trích xuất!');

        }else{

            $('#post\_form').attr('action', `/extract/${total\_start}/${(total\_end-total\_start)}`).submit();

        }

    }

});

    public function postExtract($start, $end, Request $request){

        File::deleteDirectory(public\_path('uploads'));

        $name = 'Extract\_' . $request->file('fileInput')->getClientOriginalName();

        $request->file('fileInput')->move('lame', 'convert.mp3');

        $cmd = 'cd ' . public\_path('lame') . ' & lame.exe -V2 convert.mp3 --decode';

        exec($cmd, $output, $return\_var);

        if ($return\_var === 0) {

            if (!is\_dir(public\_path('uploads'))) {

                mkdir(public\_path('uploads'), 0777, true);

            }

            rename(public\_path('lame/convert.wav'), public\_path('sox/convert.wav'));

            $cmd = 'cd ' . public\_path('sox') . " & sox.exe -t wav convert.wav extracted.wav trim $start $end";

            exec($cmd, $output, $return\_var);

            if ($return\_var === 0){

                rename(public\_path('sox/extracted.wav'), public\_path('lame/extracted.wav'));

                $cmd = 'cd ' . public\_path('lame') . ' & lame.exe -V2 extracted.wav converted.mp3';

                exec($cmd, $output, $return\_var);

                if ($return\_var === 0){

                    rename(public\_path('lame/converted.mp3'), public\_path('uploads/'.$name));

                    unlink(public\_path('lame/convert.mp3'));

                    unlink(public\_path('lame/extracted.wav'));

                    unlink(public\_path('sox/convert.wav'));

                    return response()->download(public\_path('uploads/'.$name));

                }else{

                    dd('Lỗi trích xuất 2');

                }

            } else {

                dd('Lỗi trích xuất 1');

            }

        } else {

            dd('Lỗi chuyển đổi định dạng');

        }

    }

### Chuyển đổi định dạng

// chuyển đổi định dạng

$('#convert').on('click', () => {

    $('#post\_form').attr('action', '/convert').submit();

});

    public function postConvert(Request $request)

    {

        // Xóa folder tạo ra từ lần trc

        File::deleteDirectory(public\_path('uploads'));

        $name = 'Convert\_' . pathinfo($request->file('fileInput')->getClientOriginalName(), PATHINFO\_FILENAME).'.'.$request->type\_format;

        $request->file('fileInput')->move('lame', 'convert.mp3');

        $cmd = 'cd ' . public\_path('lame') . ' & lame.exe -V2 convert.mp3 converted.' . $request->type\_format;

        exec($cmd, $output, $return\_var);

        // Kiểm tra kết quả trả về từ lệnh

        if ($return\_var === 0) {

            if (!is\_dir(public\_path('uploads'))) {

                mkdir(public\_path('uploads'), 0777, true);

            }

            rename(public\_path('lame/converted.' . $request->type\_format), public\_path('uploads/'.$name));

            unlink(public\_path('lame/convert.mp3'));

            return response()->download(public\_path('uploads/'.$name));

        } else {

            // Lệnh không thực thi thành công

            dd('Lỗi chuyển đổi định dạng');

        }

    }

### Hiển thị, đọc, ghi metadata

// Đọc metadata từ file MP3

jsmediatags.read(file, {

    onSuccess: (tag) => {

        displayMetadata(tag);

    },

    onError: (error) => {

        console.log(':(', error.type, error.info);

    }

});

    public function postInput(Request $request)

    {

        // Xóa folder tạo ra từ lần trc

        File::deleteDirectory(public\_path('uploads'));

        // lưu file mp3

        if (!is\_dir(public\_path('uploads'))) {

            mkdir(public\_path('uploads'), 0777, true);

        }

        $file\_path = $request->file('fileInput')->move('uploads', 'Edit\_' . $request->file('fileInput')->getClientOriginalName());

        $full\_file\_path = public\_path($file\_path);

        // khởi tạo trình đọc ghi metadata

        $TextEncoding = 'UTF-8';

        $getID3 = new getID3();

        $getID3->setOption(['encoding' => $TextEncoding]);

        $tagwriter = new getid3\_writetags();

        $tagwriter->filename = $full\_file\_path;

        $tagwriter->tagformats = ['id3v2.3'];

        $tagwriter->overwrite\_tags = true;

        $tagwriter->remove\_other\_tags = false;

        $tagwriter->tag\_encoding = $TextEncoding;

        // gán thông tin từ request

        $TagData = [

            'album' => [$request->album],

            'artist' => [$request->artist],

            'genre' => [$request->genre],

            'title' => [$request->title],

        ];

        // riêng ảnh ktra có truyền trong request thì thay ảnh

        if ($request->file('image')) {

            $img\_path = $request->file('image')->move('uploads', 'Edit\_' . $request->file('image')->getClientOriginalName());

            $full\_img\_path = public\_path($img\_path);

            if ($APICdata = file\_get\_contents($full\_img\_path)) {

                if ($exif\_imagetype = exif\_imagetype($full\_img\_path)) {

                    $TagData['attached\_picture'][0]['data']          = $APICdata;

                    $TagData['attached\_picture'][0]['picturetypeid'] = 0x03;

                    $TagData['attached\_picture'][0]['description']   = 'Cover Art';

                    $TagData['attached\_picture'][0]['mime']          = image\_type\_to\_mime\_type($exif\_imagetype);

                } else {

                    dd('invalid image format (only JPG, JPEG, PNG)');

                }

            } else {

                dd('cannot open ' . htmlentities($\_FILES['userfile']['tmp\_name']));

            }

        }

        // lưu

        $tagwriter->tag\_data = $TagData;

        if ($tagwriter->WriteTags()) {

            return response()->download($full\_file\_path);

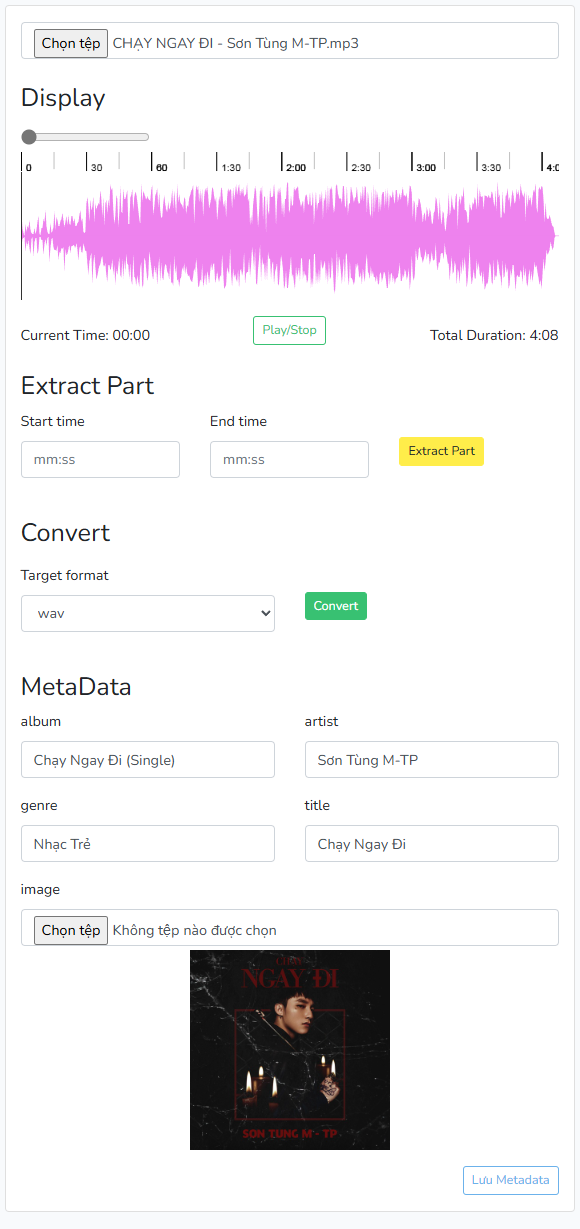
        } else {

            dd('Failed to write tags!');

        }

    }

## Kết quả - Giao diện lập trình



Hình 6 Kết quả giao diện lập trình

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

**Kết quả đạt được**

Chương trình có thể hiển thị và chỉnh sửa metadata của tệp MP3: Người dùng có khả năng xem và chỉnh sửa các thông tin như tên bài hát, nghệ sĩ, album, thể loại và năm phát hành.

Hiển thị sóng âm thanh dưới dạng tần số: Chương trình cung cấp một giao diện trực quan cho người dùng để xem sóng âm thanh và điều chỉnh các phần của tệp âm thanh.

Cắt ghép tệp âm thanh một cách chính xác: Chương trình cho phép người dùng cắt ghép các đoạn âm thanh một cách chính xác, giúp họ tạo ra các bản nhạc chuông, trích đoạn hay chỉnh sửa các phần không mong muốn của tệp âm thanh.

Chuyển đổi định dạng tệp âm thanh: Người dùng có khả năng chuyển đổi các tệp MP3 sang các định dạng khác như WAV, AAC, FLAC hoặc OGG để tương thích với nhiều thiết bị và ứng dụng khác nhau.

**Hạn chế**

Giới hạn về tính năng: Mặc dù chương trình cung cấp một loạt các tính năng quan trọng, nhưng có thể vẫn còn thiếu một số tính năng phức tạp và cao cấp, như hiệu chỉnh âm thanh đa kênh hoặc xử lý tệp âm thanh lớn.

## Hướng phát triển

Mở rộng tính năng: Phát triển và thêm các tính năng mới như hiệu chỉnh âm thanh, xử lý tệp âm thanh lớn, hoặc tích hợp với các dịch vụ và nền tảng âm nhạc trực tuyến.

Giao diện người dùng: Cải thiện giao diện người dùng để làm cho chương trình trở nên dễ sử dụng và thân thiện hơn với người dùng.