

#### BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG KHOA CÔNG NGHỆ THÔNG TIN 1

BÁO CÁO THỰC TẬP CƠ SỞ

ĐỀ TÀI: WEB NGHE NHẠC CÁ NHÂN

**Họ và tên: Đỗ Thanh Trường**

**Mã sinh viên: B21DCCN738**

**Lớp: D21CQCN06-B**

**SĐT: 0392919475**

**Giảng viên: Nguyễn Duy Phương**

*Hà Nội*

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Duy Phương đã dạy bảo, truyền đạt những kiến thức quý báu cho em trong suốt thời gian học tập này. Trong thời gian tham gia lớp học thực tập cơ sở của thầy, em đã có thêm cho mình nhiều kiến thức bổ ích, đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để em có thể vững bước sau này.

Bộ môn thực tập cơ sở là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu còn nhiều bỡ ngỡ. Mặc dù em đã cố gắng hết sức nhưng chắc chắn bài báo cáo khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong thầy xem xét và góp ý để bài báo cáo của em được hoàn thiện hơn ạ.

Em xin chân thành cảm ơn thầy!

MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc167497675)

[CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI 4](#_Toc167497676)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 4](#_Toc167497677)

[1. Ngôn ngữ lập trình Web 4](#_Toc167497678)

[1.1. HTML (Hypertext Markup Language) 4](#_Toc167497679)

[1.2. CSS (Cascading Style Sheets) 5](#_Toc167497680)

[2. Thiết kế giao diện người dùng (UI/UX) cho Web nghe nhạc 6](#_Toc167497681)

[3. Xử lý sự kiện (Event Handling) 7](#_Toc167497682)

[4. Lưu trữ dữ liệu cho Web nghe nhạc 9](#_Toc167497683)

[a. SQL (Structured Query Language) 9](#_Toc167497684)

[b. NoSQL databases 9](#_Toc167497685)

[5. API và tích hợp dịch vụ trên Web nghe nhạc 10](#_Toc167497686)

[CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ WEB NGHE NHẠC CÁ NHÂN 10](#_Toc167497687)

[1. Mô tả bài toán 10](#_Toc167497688)

[2. Các quy trình trên Website 11](#_Toc167497689)

[3. Thiết kế giao diện Website 12](#_Toc167497690)

[a. Giao diện Website 12](#_Toc167497691)

[b. Xây dựng Website 13](#_Toc167497692)

[CHƯƠNG IV. KẾT LUẬN 19](#_Toc167497693)

[1. Ưu điểm 19](#_Toc167497694)

[2. Nhược điểm 20](#_Toc167497695)

[3. Kết luận chung 20](#_Toc167497696)

[CHƯƠNG V. TÀI LIỆU THAM KHẢO 20](#_Toc167497697)

# CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

Trong thời đại số hóa ngày nay, việc tạo ra một trang web để tận hưởng âm nhạc theo sở thích của mình không chỉ là một ý tưởng thú vị mà còn là một cách tuyệt vời để kết hợp giữa đam mê âm nhạc và kỹ năng lập trình.

Dự án này không chỉ là một cơ hội để khám phá sâu hơn về công nghệ web mà còn là một hành trình sáng tạo để xây dựng một trang web cá nhân phản ánh cá tính và sở thích âm nhạc của bản thân. Qua việc sử dụng các ngôn ngữ lập trình web như HTML, CSS và JavaScript, chúng ta sẽ học cách tạo ra giao diện người dùng tương tác, đồng thời kết hợp với các công nghệ cơ sở dữ liệu như SQL hoặc NoSQL để lưu trữ và quản lý các thông tin về bản nhạc, danh sách phát và thông tin người dùng.

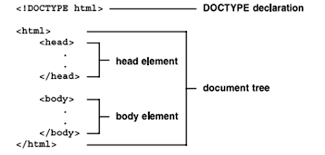
Trong dự án này, chúng ta sẽ khám phá cách tạo ra một trang web linh hoạt, có khả năng tương tác cao, từ việc tải và phát nhạc đến tạo và quản lý danh sách phát cá nhân. Đồng thời, chúng ta cũng sẽ nghiên cứu về các công nghệ mới như các API của dịch vụ nghe nhạc trực tuyến và phát triển kỹ năng tùy chỉnh và tích hợp chúng vào dự án của mình.

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT

1. Ngôn ngữ lập trình Web
   1. HTML (Hypertext Markup Language)

HTML (Hypertext Markup Language) là ngôn ngữ đánh dấu tiêu chuẩn được sử dụng để tạo ra cấu trúc và định dạng nội dung trên các trang web. Nó là ngôn ngữ cơ bản và cần thiết cho việc phát triển web, là nền tảng cho việc xây dựng các trang web đa dạng từ những trang web đơn giản đến những ứng dụng web phức tạp.

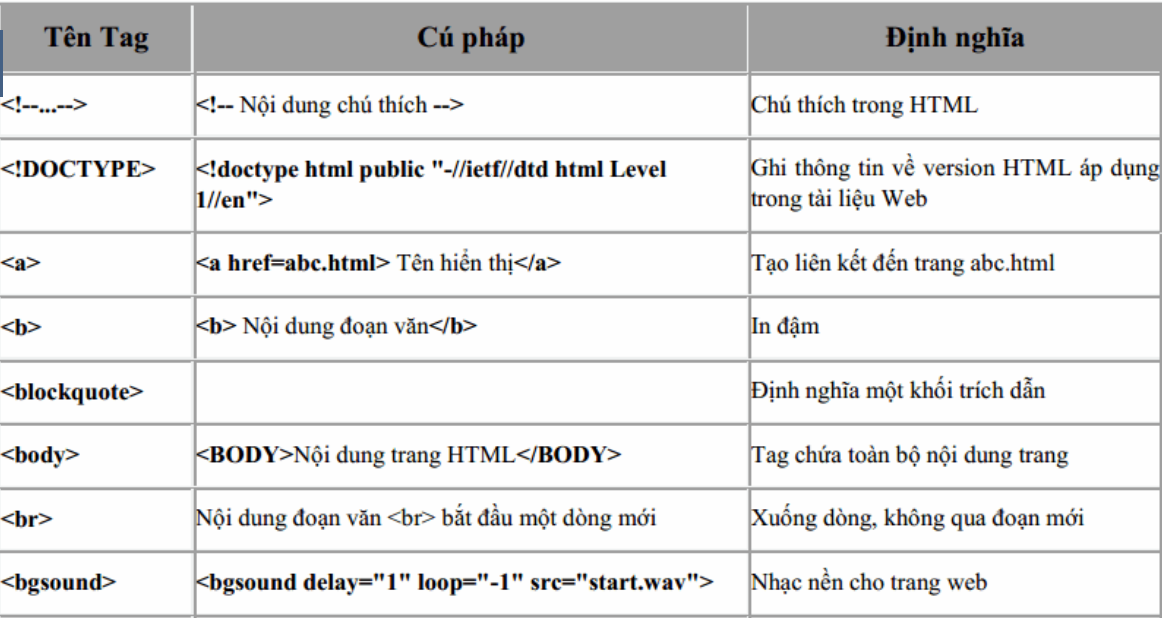
1. Cấu trúc một trang Web cơ bản



*Hình 1: Xác định cấu trúc trang Web của HTML*

HTML xác định cấu trúc của trang web bằng cách sử dụng các thẻ (tags) để đánh dấu các phần tử trên trang, chẳng hạn như tiêu đề, đoạn văn, hình ảnh, bảng, biểu mẫu, v.v.

1. Định dạng nội dung



*Hình 2: Một số thẻ định dạng thông dụng của HTML*

HTML cung cấp các thẻ để định dạng và hiển thị nội dung trên trang web, bao gồm văn bản, hình ảnh, video, âm thanh, và các phần tử tương tác khác.

1. Tính tương tác

Mặc dù HTML chủ yếu tập trung vào cấu trúc và định dạng, nhưng nó cũng cung cấp một số phương tiện để tạo ra các phần tử tương tác trên trang web thông qua các phần tử như nút, liên kết, biểu mẫu, v.v.

Tính tương tác của HTML thường được thực hiện thông qua việc sử dụng các phần tử HTML có khả năng tương tác và kết hợp với JavaScript để tạo ra các trang web động và chứa các tính năng tương tác. Một số cách mà HTML có thể tương tác với người dung như: Nút và Liên kết (Buttons and Links), Biểu mẫu (Forms), Sự kiện (Events), Multimedia,...

1. Tiêu chuẩn hóa

HTML tuân theo các tiêu chuẩn quốc tế được duy trì bởi tổ chức W3C (World Wide Web Consortium), giúp đảm bảo tính tương thích và tương lai của các trang web trên các trình duyệt khác nhau.

* 1. CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) là ngôn ngữ được sử dụng để điều chỉnh và định dạng giao diện người dùng của các trang web và các ứng dụng web. Nó cho phép chúng ta tạo ra các thiết kế đẹp mắt và chuyên nghiệp bằng cách điều chỉnh các phần tử HTML để chúng trở nên hấp dẫn và dễ đọc.

CSS cho phép chúng ta điều chỉnh các phần tử HTML bằng cách áp dụng các quy tắc về màu sắc, font chữ, kích thước, khoảng cách, và các thuộc tính khác để tạo ra giao diện trực quan và hấp dẫn.



*Hình 3: Điều chỉnh phần tử HTML bằng CSS*

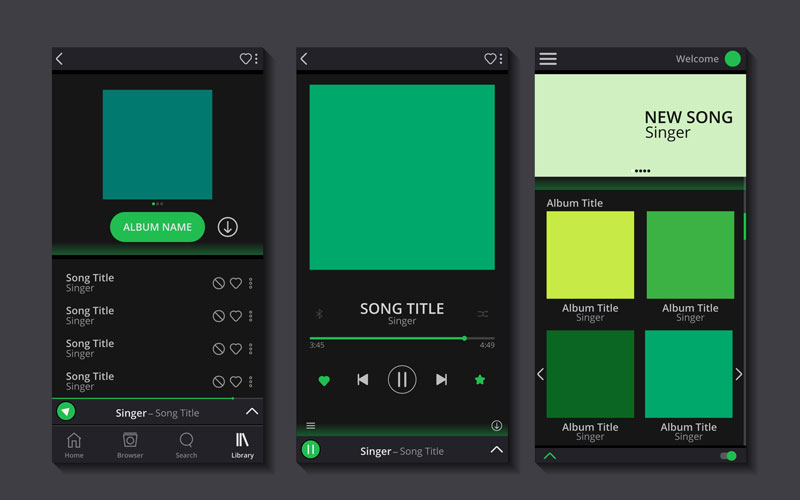
CSS giúp tách biệt hoàn toàn kiểu dáng và nội dung của trang web, cho phép chúng ta thay đổi giao diện mà không làm thay đổi nội dung HTML của trang. CSS cũng cho phép chúng ta xác định các kiểu dáng trong một tập tin riêng biệt, sau đó áp dụng chúng cho nhiều trang web khác nhau, giúp tăng tính tái sử dụng và dễ dàng bảo trì.

1. Thiết kế giao diện người dùng (UI/UX) cho Web nghe nhạc

Thiết kế website ảnh hưởng trực tiếp đến trải nghiệm của người dùng khi sử dụng nền tảng nghe nhạc. Một giao diện người dùng hấp dẫn, dễ sử dụng và thân thiện giúp người dùng dễ dàng tìm kiếm, phát nhạc và tương tác trên website. Ngoài ra, tốc độ tải trang nhanh cũng là yếu tố quan trọng để người dùng không gặp khó khăn trong việc truy cập và phát nhạc.

Với sự phổ biến của các thiết bị di động, thiết kế website nghe nhạc online cần phải tương thích với nhiều loại màn hình và các nền tảng khác nhau, bao gồm máy tính để bàn, điện thoại di động và máy tính bảng. Điều này đảm bảo rằng người dùng có thể truy cập và sử dụng dịch vụ nghe nhạc một cách thuận tiện từ bất kỳ thiết bị nào mà họ sở hữu.

Thiết kế website nghe nhạc online cung cấp khả năng thêm các tính năng và tùy chọn cho người dùng. Ví dụ, giao diện người dùng phù hợp có thể cung cấp khả năng tạo danh sách phát cá nhân, chia sẻ âm nhạc với chúng ta bè, tạo ra playlist theo sở thích và khám phá các bài hát mới. Các tính năng như tìm kiếm nâng cao, gợi ý nhạc và chế độ phát ngẫu nhiên cũng làm tăng tính hấp dẫn và tiện ích của nền tảng nghe nhạc.



*Hình 4: Ví dụ về giao diện nghe nhạc cho người dùng cá nhân*

Để thiết kế một Website nghe nhạc cơ bản, chúng ta cần có một số nội dung sau:

* Giới thiệu: Bao gồm các thông tin tổng quan về âm nhạc.
* Thông tin bài hát, album, MV: Hiển thị các thông tin như nội dung bài hát, nguồn bài hát, thư viện hình ảnh, bài hát liên quan, thông tin ca/nhạc sĩ,...kèm theo nhiều danh mục nhỏ khác như bài hát hot nhất, bảng xếp hạng bài hát, album mới nhất, bài hát mới nhất,...cho phép người dùng có thể nhận xét về nội dung bài hát.
* Đăng ký thành viên: Cho phép người dùng đăng ký, đăng nhập thành viên để tham gia nghe - tải nhạc, mua nhạc,...
* Dịch vụ: Tải nhạc, mua nhạc, tích hợp tính năng mua nhạc, thanh toán trực tiếp bằng thẻ cào, ATM,..
* Tìm kiếm: Với tính năng này, khi khách hàng truy cập có thể tìm kiếm thông tin mà họ cần theo các từ khóa như tên bài hát, ca sĩ, tên album hoặc bất kỳ nội dung nào khác.
* Tin tức: Cập nhật các tin tức về các bài hát, album mới nhất
* Thống kê: Tích hợp bộ đếm số người đang truy cập vào web, số người đang online, thống kê số hóa được tìm trên web, bài hát nghe nhiều nhất.

1. Xử lý sự kiện (Event Handling)

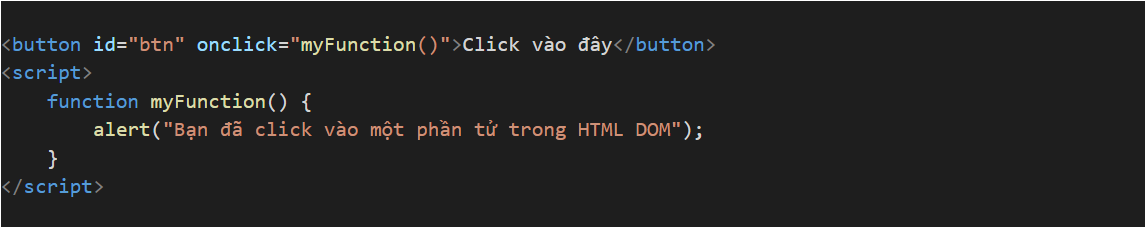
Một trang web với các thao tác như click chuột, hay di chuột,… Đây được xem là các sự kiện và JavaScript có thể bắt các sự kiện này thông qua HTML DOM.

Các sự kiện này được cụ thể bằng các thuộc tính HMTL như: onclick (click chuột), onmouseover (di chuột qua), ….

Và bằng cách lắng nghe các sự kiện và người dùng tương tác, ta gửi trả phản hồi lại tương ứng đó gọi là xử lý sự kiện.

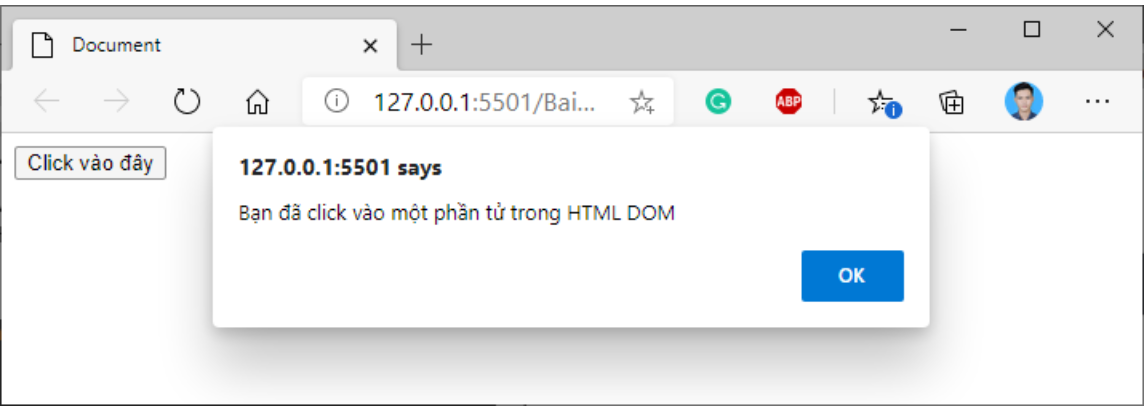
Chúng ta có thể thêm sự kiện bằng cách thêm thuộc tính sự kiện vào phần tử HTML. DOM trong JavaScript có thể thao tác với các phần tử của HTML và xử lý sự kiện.

Ví dụ một phản ứng với sự kiện click chuột của người dùng:



*Hình 5: Ví dụ về sự kiện (events) trong HTML*

Khi đó, kết quả trả lại trên trang Web ta sẽ được như sau:



*Hình 6: Kết quả trả lại một event*

***Xử lý sự kiện JavaScript DOM:***

1. Chèn vào thuộc tính sự kiện của HTML DOM:

Cú pháp tổng quát để thêm một sự kiện mới vào một phần tử HTML như sau:



*Hình 7: Cú pháp thêm một sự kiện mới vào HTML*

Trong đó:

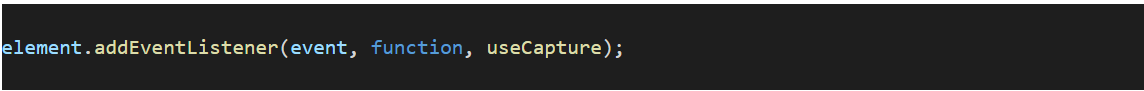
* method(): Là hàm xử lý DOM để chỉ định một phần tử HTML
* events: Là tên sự kiện sẽ được thực thi (tham khảo sự kiện ở bảng trên)
* function(){ ... }: Là code JavaScript mà khi sự kiện được kích hoạt thì hàm này sẽ thực thi.

1. Sử dụng phương thức addEvenListener():

Phương thức addEvenListener(): Phương thức này cũng cho phép ta gán sự kiện cho một phần tử HTML DOM, điểm khác các phương thức trên là có thể gán một hoặc nhiều events vào một phần tử HTML mà không xóa đi các sự kiện khác.

Khi ta sử dụng phương thức addEventsListenner(), JavaScript sẽ tách biệt nó với HTML Makup.

Cú pháp để thêm sự kiện như sau:



*Hình 8: Cú pháp thêm sự kiện bằng phương thức addEventListener()*

Trong đó:

* event: Là tên sự kiện (chúng ta có thể tham khảo ở bảng phía trên).
* function: Là hàm JavaScript mà chúng ta muốn thực thi khi sự kiện kích hoạt.
* useCapture: Là một giá trị boolean value cho hai lựa chọn: event bubbling hoặc event capturing.

1. Xóa một sự kiện với removeEventListener():

Phương thức removeEventListener() cho phép xóa sự kiện đã được gắn vào phần tử HTML trước đó bằng phương thức addEventListener()

Cú pháp:

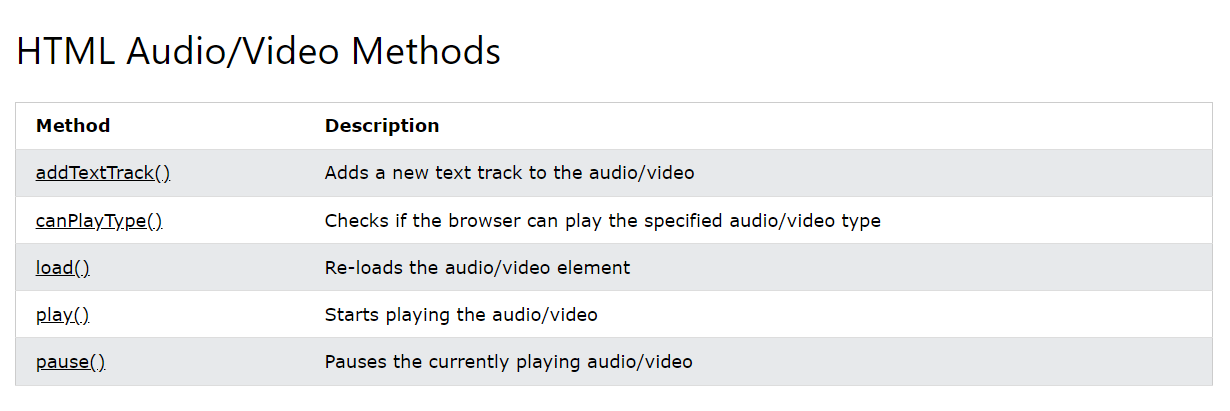


*Hình 9: Cú pháp xóa sự kiện bằng phương thức removeEventListener()*

Trong đó:

* event: Là sự kiện cần xóa
* function: Là hàm thực thi cùng với sự kiện.

***Một số events có thể thao tác với Web nghe nhạc:***



*Hình 10: Các methods tương tác với âm thanh và video*

1. Lưu trữ dữ liệu cho Web nghe nhạc

Để xây dựng một ứng dụng web nghe nhạc cá nhân đáng tin cậy và linh hoạt, việc lưu trữ dữ liệu là một phần không thể thiếu. Cơ sở dữ liệu đóng vai trò quan trọng trong việc quản lý thông tin về các bài hát, danh sách phát và thông tin người dùng. Chúng ta có thể lưu trữ dữ liệu qua SQL (Structured Query Language) hoặc NoSQL databases.

1. SQL (Structured Query Language)

SQL là một trong những lựa chọn phổ biến nhất cho các hệ thống quản lý cơ sở dữ liệu quan hệ. Với SQL, chúng ta có thể thiết kế và quản lý các bảng dữ liệu theo cấu trúc cố định và quan hệ giữa chúng. Điều này làm cho SQL trở thành một lựa chọn lý tưởng cho việc lưu trữ thông tin về các bài hát, danh sách phát và thông tin người dùng trong ứng dụng nghe nhạc cá nhân của chúng ta. Các hệ thống quản lý cơ sở dữ liệu quan hệ như MySQL, PostgreSQL và SQL Server thường được sử dụng cho các ứng dụng web phổ biến.

1. NoSQL databases

Ngoài SQL, các hệ thống NoSQL cung cấp các lựa chọn linh hoạt hơn cho việc lưu trữ dữ liệu. NoSQL databases không yêu cầu cấu trúc quan hệ cố định và có thể chứa dữ liệu dưới nhiều dạng khác nhau như tài liệu, cột, hoặc đồ thị. Điều này có thể phù hợp với các ứng dụng web nghe nhạc cá nhân khi cần xử lý lượng dữ liệu lớn và có tính linh hoạt cao trong việc thay đổi cấu trúc dữ liệu. Các hệ thống NoSQL phổ biến bao gồm MongoDB, Cassandra và Redis.

1. API và tích hợp dịch vụ trên Web nghe nhạc

API (Application Programming Interface) đóng vai trò quan trọng trong việc tương tác với các dịch vụ như Spotify, Apple Music hoặc YouTube để tải về thông tin về bài hát và phát nhạc từ các nguồn khác nhau. Việc sử dụng các API này có thể mở ra nhiều cơ hội mới và cung cấp cho người dùng trải nghiệm nghe nhạc đa dạng và phong phú.

Tích hợp API cho một trang web nghe nhạc có thể bao gồm nhiều bước, tùy thuộc vào dịch vụ nhạc cụ thể mà chúng ta muốn tích hợp và chức năng cụ thể mà chúng ta muốn cung cấp cho người dùng. Dưới đây là một hướng dẫn tổng quan về cách tích hợp API cho web nghe nhạc:

* Chọn dịch vụ nhạc và tìm hiểu về API:

Chọn một dịch vụ nhạc phổ biến như Spotify, Apple Music, YouTube Music, hoặc SoundCloud.

Tìm hiểu về API mà dịch vụ nhạc đó cung cấp và các chức năng chúng ta có thể sử dụng để tạo ra trang web nghe nhạc.

* Đăng ký và xác thực với API:

Đăng ký tài khoản phát triển và lấy các thông tin xác thực như Client ID, Client Secret, hoặc API Key từ dịch vụ nhạc.

Sử dụng thông tin xác thực này để gửi các yêu cầu API và truy cập dữ liệu từ dịch vụ nhạc.

* Xây dựng chức năng tìm kiếm và hiển thị danh sách bài hát:

Sử dụng API để tìm kiếm bài hát dựa trên tiêu chí như tiêu đề, nghệ sĩ, hoặc thể loại.

Hiển thị kết quả tìm kiếm dưới dạng danh sách bài hát trên trang web của chúng ta.

* Phát nhạc và quản lý danh sách phát:

Sử dụng API để phát nhạc trực tiếp từ dịch vụ nhạc hoặc tạo danh sách phát cá nhân cho người dùng.

Cho phép người dùng thêm bài hát vào danh sách phát, xóa bài hát, hoặc thay đổi thứ tự của các bài hát trong danh sách phát.

* Tích hợp tính năng khác:

Tùy thuộc vào API và dịch vụ nhạc chúng ta sử dụng, chúng ta có thể tích hợp các tính năng khác như xem thông tin về nghệ sĩ và album, tạo danh sách phát ngẫu nhiên, hoặc chia sẻ bài hát trên mạng xã hội.

* Bảo mật và quản lý xác thực:

Đảm bảo rằng chúng ta bảo mật thông tin xác thực của ứng dụng của mình như Client ID, Client Secret, hoặc API Key.

Quản lý xác thực và token đăng nhập để đảm bảo tính bảo mật và tuân thủ các quy định của dịch vụ nhạc.

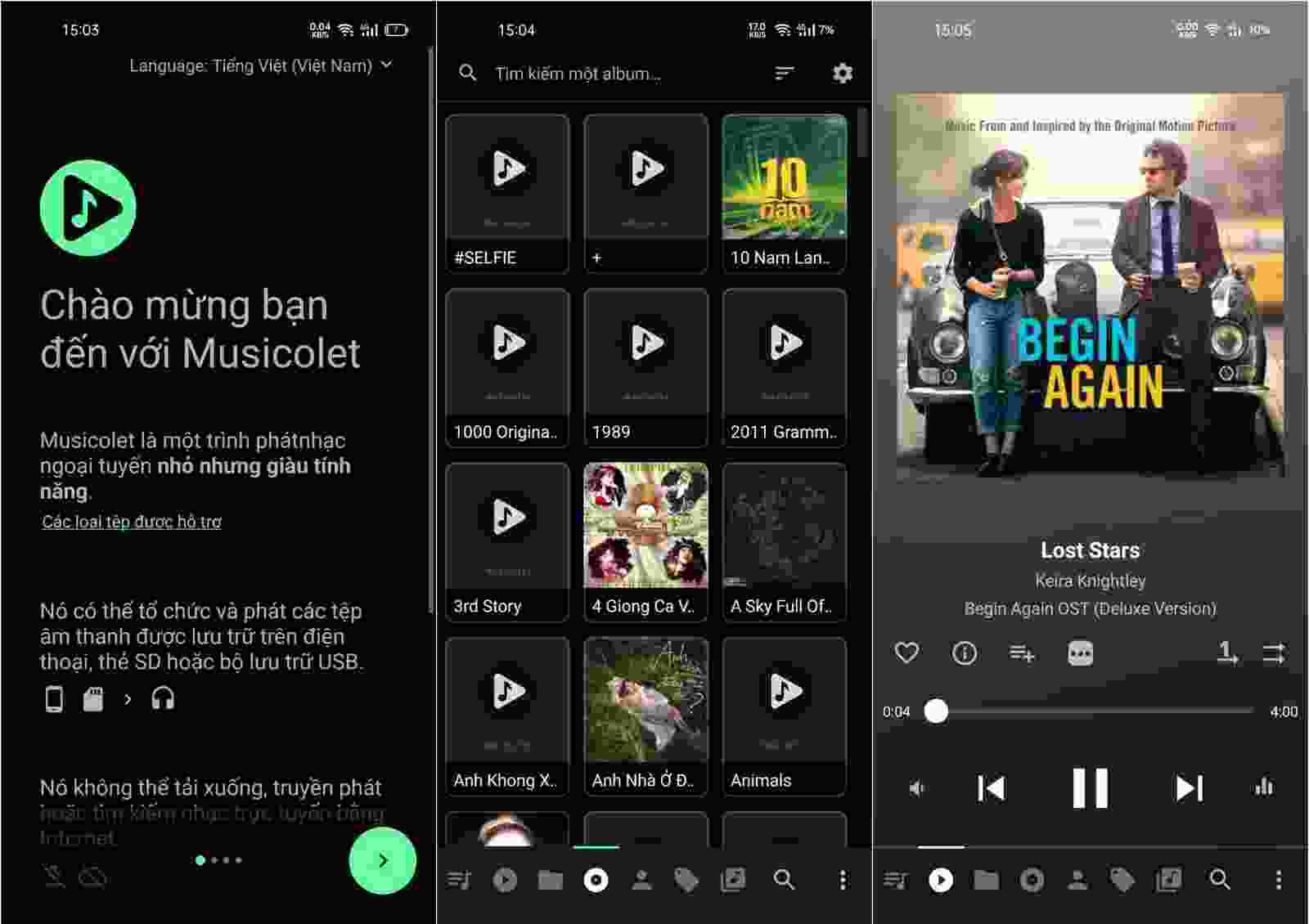
* Kiểm tra tích hợp:

Kiểm tra tích hợp của chúng ta bằng cách thử nghiệm các chức năng trên trang web của chúng ta và đảm bảo rằng mọi thứ hoạt động đúng cách.

# CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ WEB NGHE NHẠC CÁ NHÂN

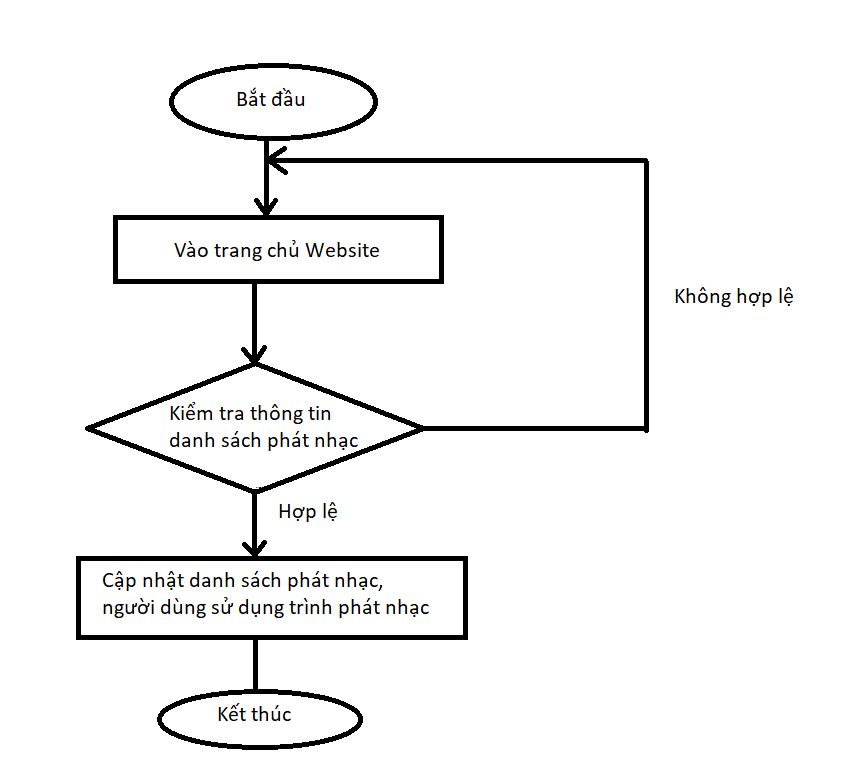
1. Mô tả bài toán

Khi truy cập vào trang web Music Player, người dùng sẽ được chào đón bằng một danh sách các bài hát và thông tin về tác giả, ca sĩ thể hiện bài hát đó. Trải nghiệm của họ sẽ được tối ưu hóa để thực hiện các tác vụ cơ bản của một trang web nghe nhạc, bao gồm chọn bài hát, phát nhạc (Play), tạm dừng nhạc (Pause), và chuyển tiếp giữa các bài hát một cách dễ dàng. Đồng thời, họ cũng có thể thưởng thức âm nhạc theo thời gian thực của từng bài hát, tạo ra một trải nghiệm nghe nhạc trực tuyến mượt mà và thú vị.



*Hình 11 : Ví dụ minh họa khi người dùng truy cập vào Web nghe nhạc*

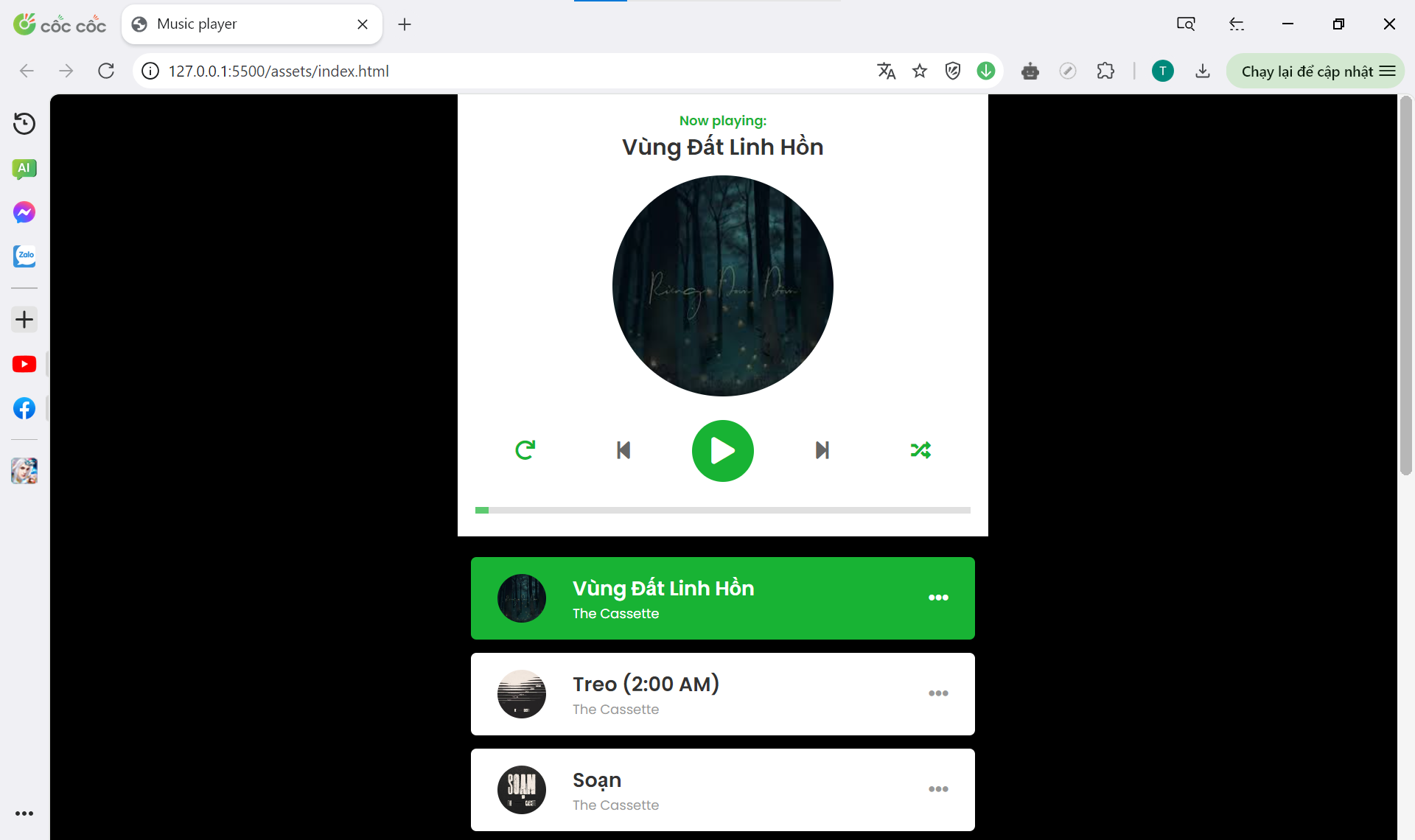
1. Các quy trình trên Website



*Hình 12. Quy trình của Website nghe nhạc*

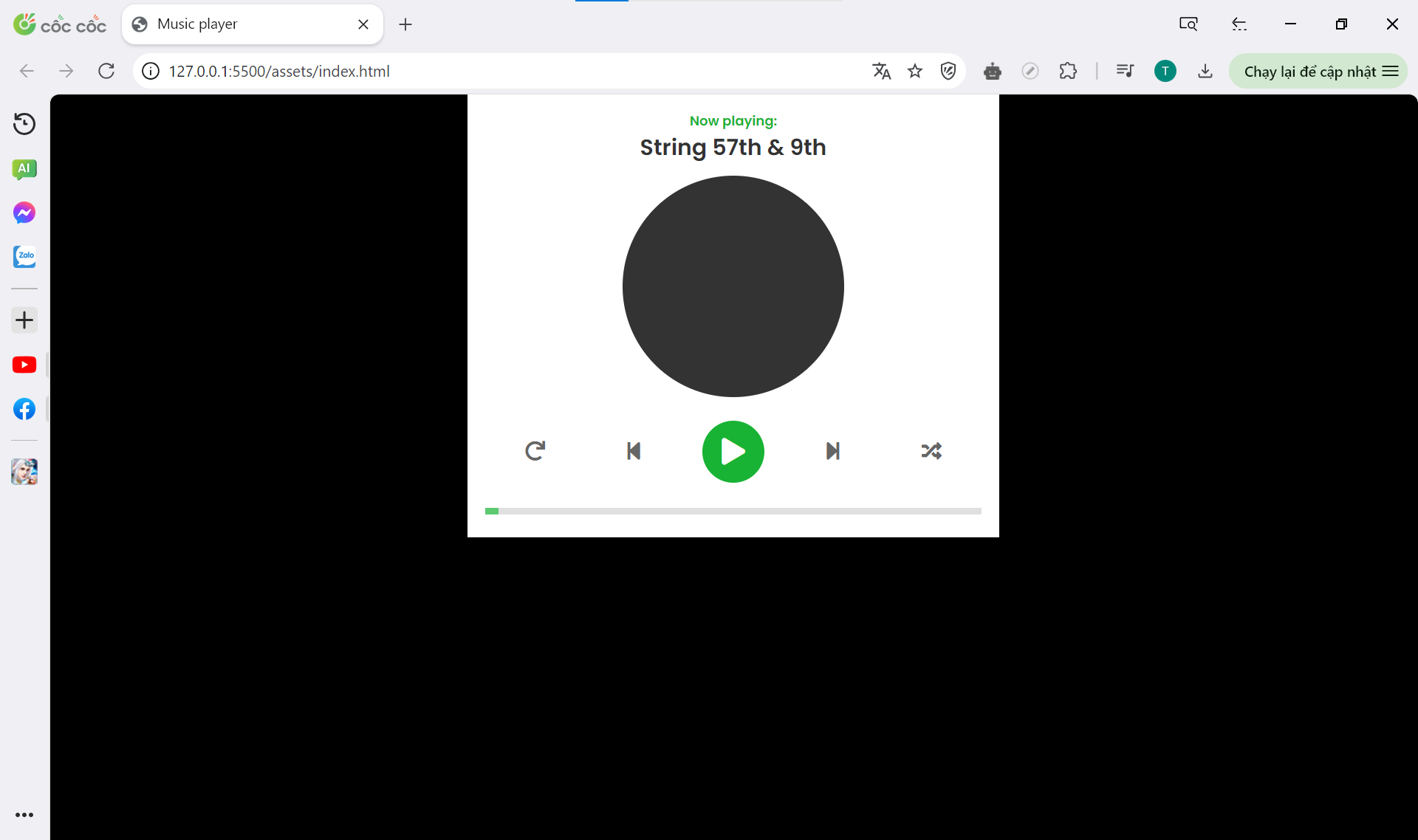
* Mô tả:
* Người sử dụng vào Website để nghe nhạc, Website sẽ phản hồi lại giao diện nghe nhạc nếu như dữ liệu danh sách bài hát hợp lệ và người dung có thể sử dụng các tác vụ cơ bản trên website
* Nếu dữ liệu danh sách các bài hát không hợp lệ thì website sẽ không hiện lên danh sách bài hát và người dung cũng không thể thao tác trên website cho đến khi dữ liệu danh sách bài hát được update hợp lệ

1. Thiết kế giao diện Website
2. Giao diện Website



*Hình 13. Giao diện trang chủ Music Player*

1. Xây dựng Website
2. Tạo giao diện ban đầu cho Website Music Player bằng HTML và CSS:



*Hình 14. Giao diện cơ bản của Music Player*

Đoạn mã HTML sẽ định nghĩa giao diện của trình phát nhạc web trên. Nó bao gồm các phần chính sau:

* Một phần "Dashboard" hiển thị thông tin về bài hát đang phát, có nút điều khiển như phát, tạm dừng, chuyển bài và thanh trượt tiến độ.
* Một phần "Playlist" để hiển thị danh sách các bài hát trong danh sách phát của người dùng.
* Liên kết với các tệp CSS và các thư viện biểu tượng (icon) để tạo ra giao diện trực quan và thú vị.

1. Tạo các thao tác trên Website Music Player:
   * + Render bài hát

Code:

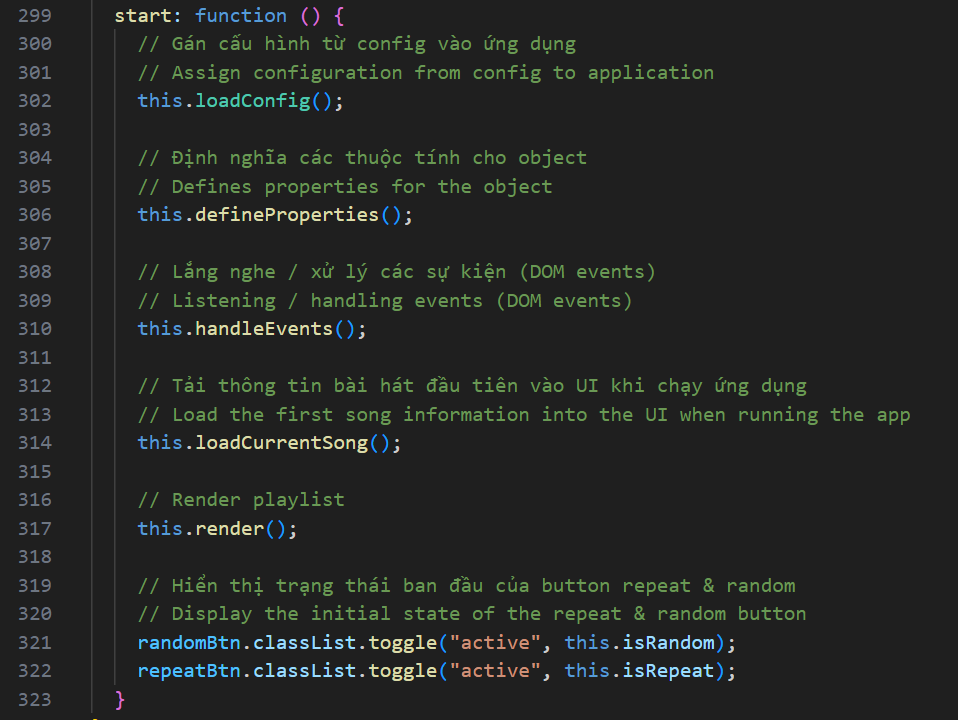


Trong đó, songs là một object được tạo ra để lưu thông tin các bài hát bào gồm, tên bài hát, tác giả, file ảnh CD, file nhạc mp3.



* + - Hàm khởi chạy Website:

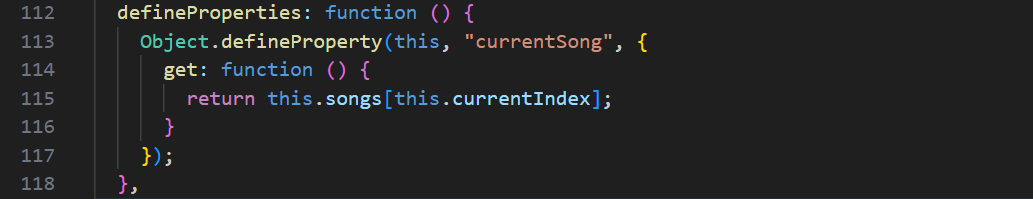
Code:



Trong đó, hàm khởi chạy sẽ gọi các hàm, thủ tục như this.loadConfig(), this.defineProperties(), this.handleEvents(), this.loadCurrentSong() và this.render()

1. Xây dựng các hàm xử lý sự kiện và xử lý bài hát
   * + Hàm định nghĩa thuộc tính

Code:



Phương thức này được sử dụng để định nghĩa các thuộc tính cho một đối tượng. Trong trường hợp này, nó đang định nghĩa một thuộc tính mới có tên là currentSong cho đối tượng hiện tại.

Cụ thể, phương thức Object.defineProperty() được sử dụng để định nghĩa thuộc tính currentSong. Thuộc tính này có một getter, được xác định thông qua một hàm, mục đích của getter là trả về giá trị của phần tử hiện tại trong mảng songs, mà là một thuộc tính khác của đối tượng (this). Phần tử hiện tại được xác định bởi chỉ số currentIndex.

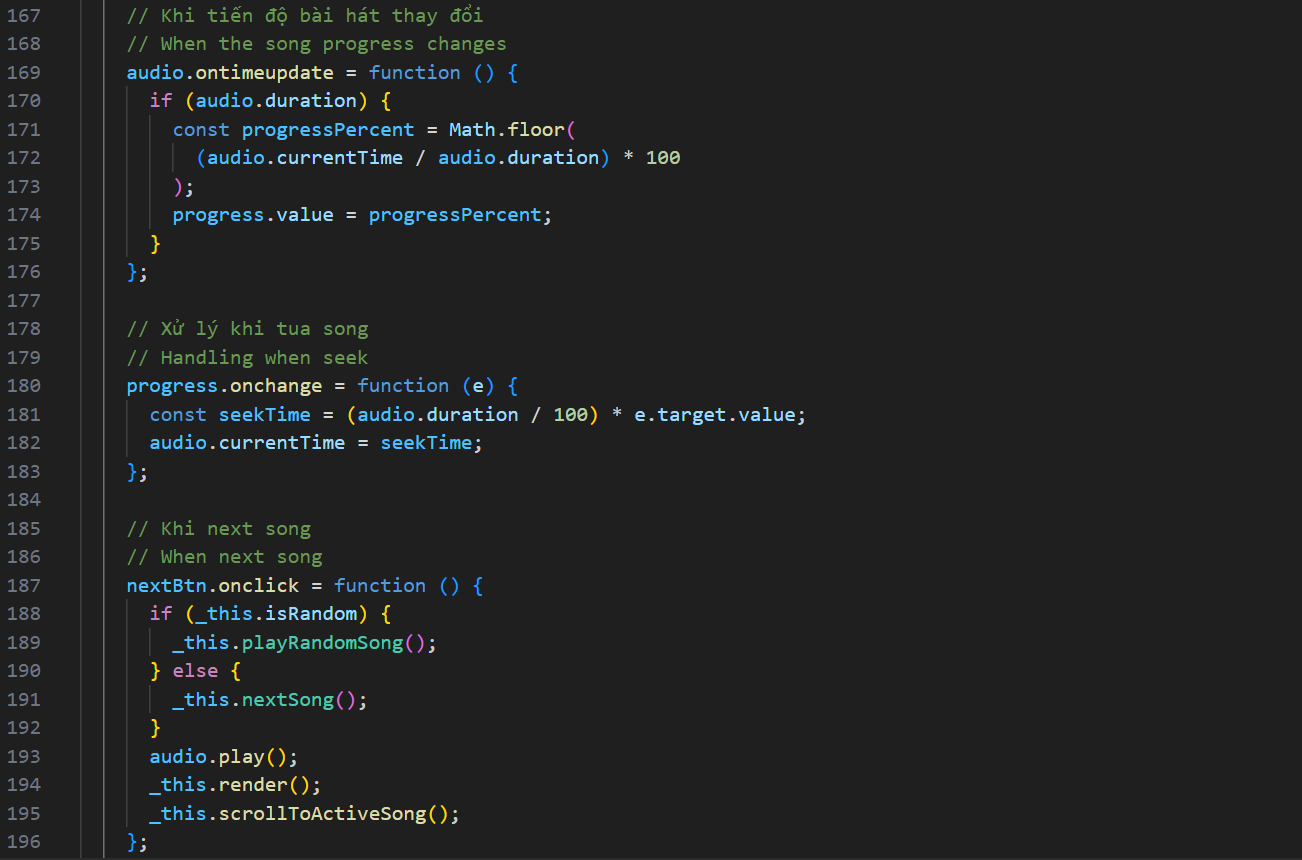
Điều này có nghĩa là khi chúng ta truy cập vào thuộc tính currentSong của đối tượng này, nó sẽ trả về bài hát hiện tại đang được phát, dựa trên chỉ số hiện tại được lưu trữ trong currentIndex.

* + - Hàm lắng nghe, xử lý các sự kiện (DOM events)

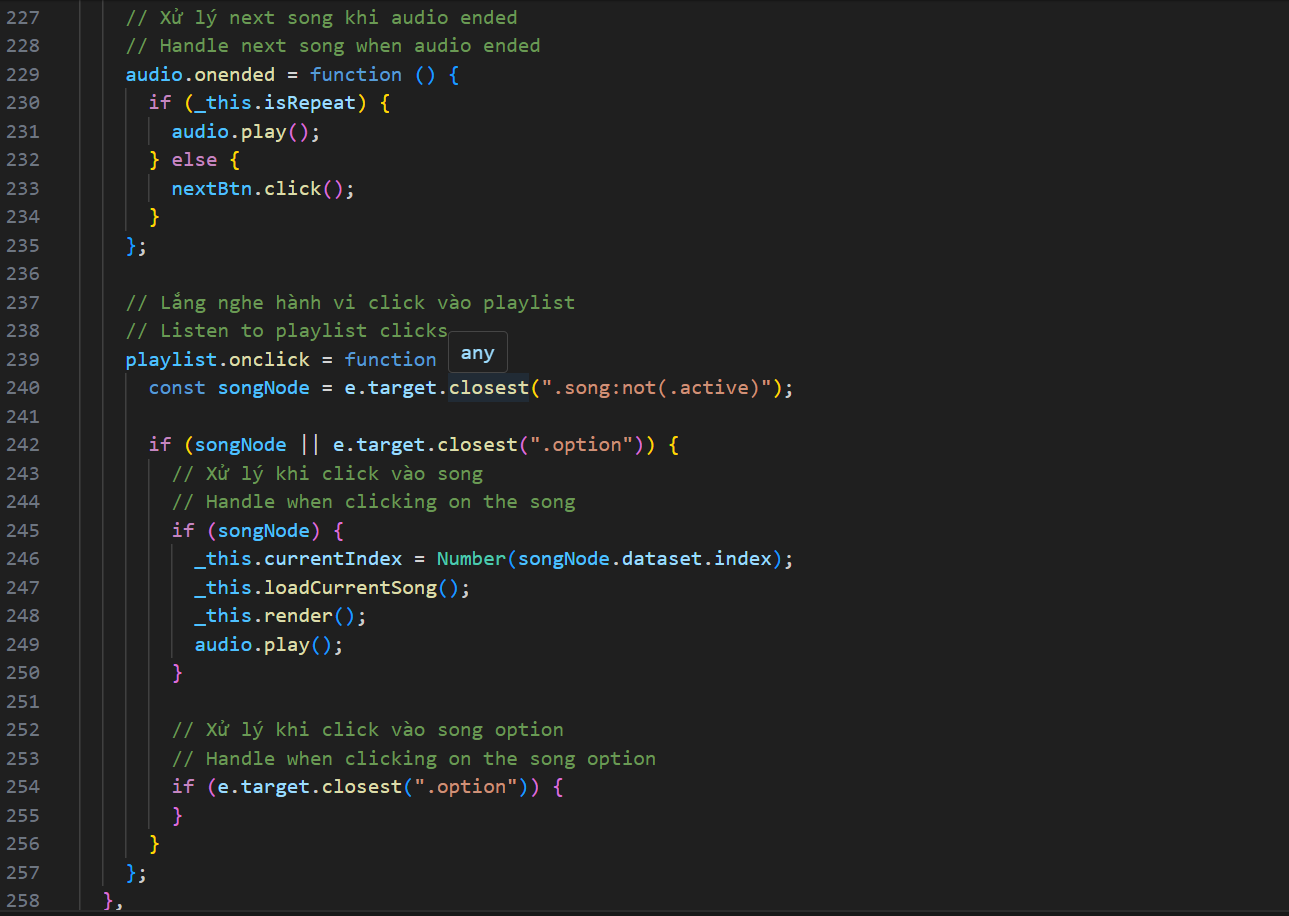
Code:





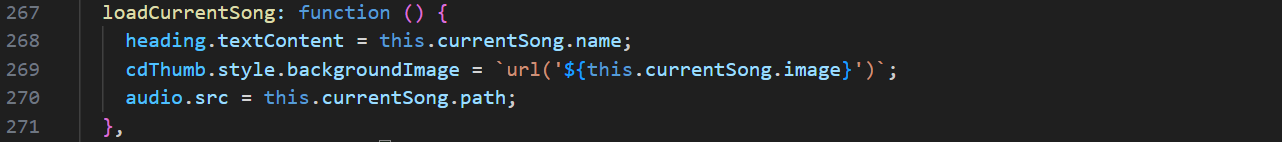






* Xử lý CD quay / dừng: Điều chỉnh animation để quay hoặc dừng quay của CD khi bắt đầu hoặc dừng phát nhạc.
* Xử lý phóng to / thu nhỏ CD: Điều chỉnh kích thước và độ trong suốt của CD dựa trên việc cuộn trang web.
* Xử lý khi click play: Xử lý sự kiện khi nút play được nhấn để bắt đầu hoặc tạm dừng phát nhạc.
* Khi song được play hoặc pause: Xử lý các sự kiện khi bài hát bắt đầu hoặc tạm dừng.
* Khi tiến độ bài hát thay đổi: Cập nhật thanh tiến độ của bài hát khi nó đang phát.
* Xử lý khi tua bài hát: Xử lý sự kiện khi người dùng tua bài hát bằng cách di chuyển thanh tua.
* Khi next hoặc prev song: Xử lý khi người dùng chuyển bài hát kế tiếp hoặc trước đó.
* Xử lý bật / tắt random song và lặp lại một song: Xử lý các sự kiện khi người dùng bật hoặc tắt chế độ ngẫu nhiên hoặc lặp lại.
* Xử lý next song khi audio ended: Xử lý khi bài hát kết thúc để chuyển sang bài hát tiếp theo hoặc lặp lại bài hát hiện tại.
* Lắng nghe hành vi click vào playlist: Xử lý khi người dùng click vào một bài hát trong danh sách phát để chuyển sang bài hát đó và phát.
  + - Hàm tải thông tin bài hát vào UI khi chạy ứng dụng

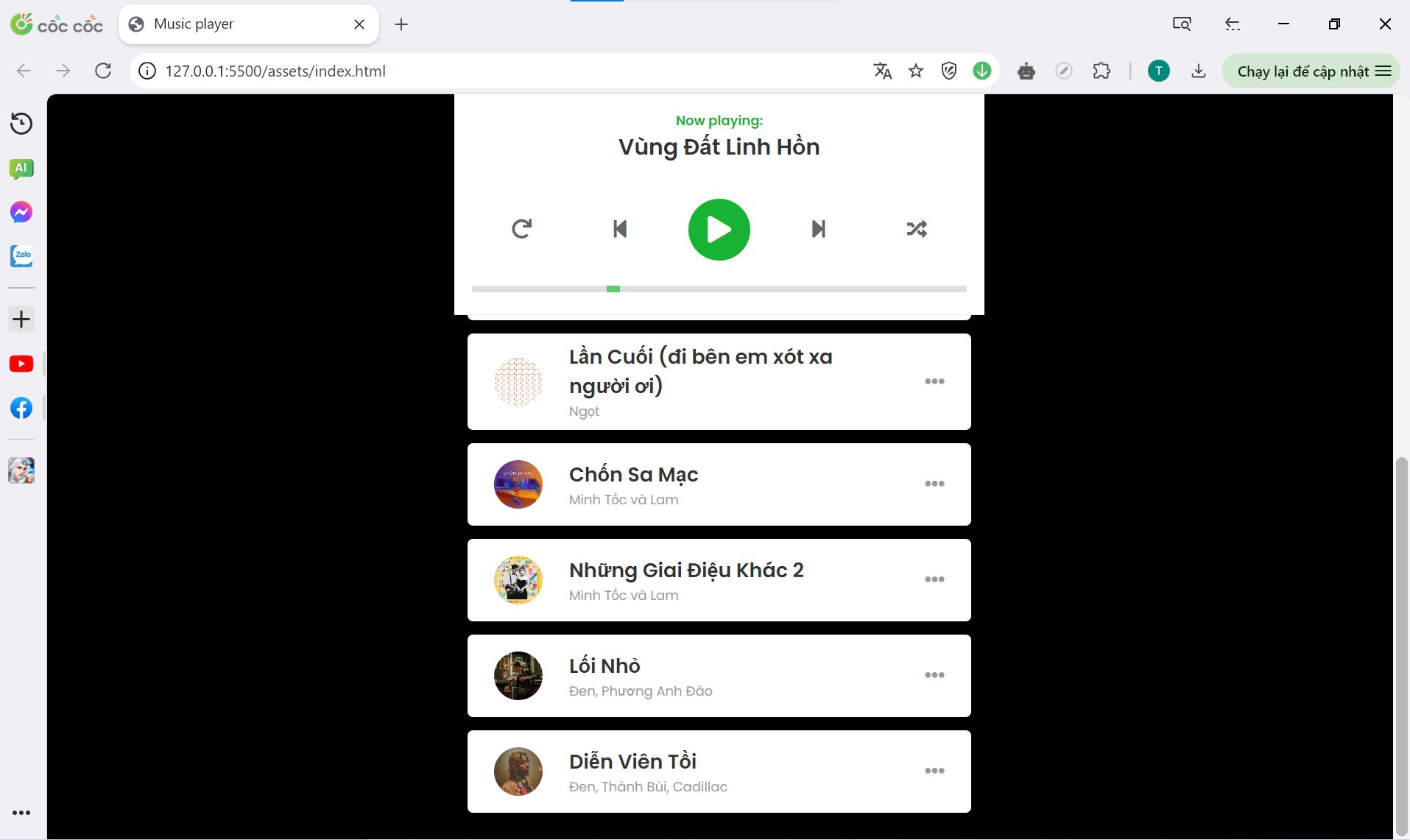
Code:



Hàm này được sử dụng để tải bài hát hiện tại lên trình phát nhạc. Cụ thể:

* Cập nhật nội dung của phần tử có id là "heading" để hiển thị tên của bài hát hiện tại.
* Cập nhật hình ảnh của CD để hiển thị hình ảnh của bài hát hiện tại.
* Thiết lập đường dẫn của audio để phát bài hát hiện tại bằng cách gán giá trị của thuộc tính "path" của đối tượng bài hát hiện tại vào thuộc tính "src" của phần tử audio.

1. Chương trình demo



*Hình 15. Giao diện Website Music Player hoàn chỉnh*

Link Website: <http://127.0.0.1:5500/assets/index.html>

Link Source Code web Music Player:

# CHƯƠNG IV. KẾT LUẬN

1. Ưu điểm

* Giao diện người dùng thân thiện và dễ sử dụng, giúp người dùng dễ dàng tìm kiếm và chọn bài hát.
* Giao diện thân thiện với người dùng trên cả máy tính và thiết bị di động.
* Hiển thị danh sách phát rõ ràng và dễ dàng quản lý.
* Cung cấp khả năng phát nhạc liên tục mà không bị gián đoạn.
* Cho phép người dùng tùy chỉnh cài đặt như lặp lại bài hát, chế độ ngẫu nhiên, hoặc điều chỉnh âm lượng.
* Hiển thị thông tin đầy đủ về bài hát và nghệ sĩ, bao gồm tên bài hát, nghệ sĩ, hình ảnh và thông tin mô tả.

1. Nhược điểm
   * Website còn đơn giản, chưa hoàn thiện nhiều chức năng, cần được mở rộng và nâng cấp nhiều lần

* Chưa có tính năng tìm kiếm linh hoạt và hiệu quả để người dùng tìm kiếm bài hát, nghệ sĩ hoặc album mong muốn.
* Chưa cung cấp liên kết hoặc thông tin về bài hát trên các nền tảng khác như Spotify hoặc Apple Music.
* Chưa có chức năng người dùng tương tác như thích, chia sẻ, hoặc thêm bài hát vào danh sách yêu thích.
* Các thông báo cho người dùng về các sự kiện như bài hát mới, album mới của nghệ sĩ yêu thích chưa được thêm vào chức năng.
* Chưa liên kết được cơ sở dữ liệu để cập nhật các chức năng liên quan đến thông tin người dung và danh sách bài hát
* Web chưa có cơ chế bảo mật để bảo vệ thông tin cá nhân của người dùng.
* Web chưa đáp ứng được nhu cầu người dùng quản lý tài khoản, bao gồm đăng nhập, đăng ký, và quản lý thông tin cá nhân.

1. Kết luận chung

Sau khoảng thời gian học tập và tìm hiểu, em đã cơ bản thực hành tạo lập được một trang Web nghe nhạc cá nhân với các tác vụ cơ bản nhất. Tuy nhiên, do còn thiếu sót ở nhiều lĩnh vực khác nên trang Web em tạo ra còn rất hạn chế về các tính năng khác. Mong thầy có nhận xét giúp đỡ để em có thể rút kinh nghiệm và làm tốt hơn cho các dự án sau này, em xin cảm ơn.

# CHƯƠNG V. TÀI LIỆU THAM KHẢO

Lập trình Web nghe nhạc từ F8: <https://fullstack.edu.vn/>

Youtube hướng dẫn: https://www.youtube.com/watch?v=vAecGPWxzFE