

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

[illegible]

Trà Vinh, ngày tháng năm

Giáo viên hướng dẫn

(Ký tên và ghi rõ họ tên)

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Trà Vinh, ngày tháng năm
Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Để hoàn thành khóa luận với đề tài: “Xây dựng website nghe nhạc trực tuyến”, em xin bày tỏ lòng biết ơn sâu sắc đến quý Thầy Cô ở trường, đặc biệt là các Thầy Cô thuộc Khoa Công Nghệ Thông Tin, những người đã tận tình giảng dạy và truyền đạt cho em những kiến thức chuyên môn quý báu trong suốt quá trình học tập tại trường.

Em đặc biệt gửi lời tri ân chân thành đến Thầy Đoàn Phước Miên, người đã tận tâm hướng dẫn, hỗ trợ và tạo điều kiện thuận lợi để em có thể hoàn thành tốt khóa luận tốt nghiệp này.

Trong quá trình thực hiện đề tài, em đã nỗ lực hết sức mình. Tuy nhiên, do hạn chế về kiến thức thực tiễn và kinh nghiệm, bài làm không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu từ quý Thầy Cô để bài khóa luận được hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

MỤC LỤC	iv
DANH MỤC HÌNH ẢNH.....	vi
DANH MỤC BẢNG	vii
PHẦN MỞ ĐẦU	8
1. Lý do chọn đề tài	8
2. Mục tiêu đề tài	8
3. Phương pháp nghiên cứu	2
4. Bố cục chuyên đề	2
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	3
1.1. Đối tượng mục tiêu nghiên cứu và phạm vi nghiên cứu.....	3
1.2. Tính khả thi	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	5
2.1. Tổng quan về ngôn ngữ ASP .NET	5
2.1.1. Giới thiệu ASP.NET Framework	5
2.1.2. Kiến trúc ASP.NET MVC	6
2.1.3. Lý do chọn Aps.NET MVC	6
2.2. Javascript.....	7
2.3. Bootstrap	10
2.4. Tổng quan về SQL	10
2.4.1. Hệ quản trị cơ sở dữ liệu SQL Server	10
2.4.2. Đặc điểm của SQL Server và đối tượng làm việc	10
2.4.3. Các đối tượng quan trọng trong SQL Server	11
CHƯƠNG 3: THỰC HIỆN HÓA NGHIÊN CỨU	13
3.1. Khảo sát hiện trạng	13
3.2. Phân tích hệ thống.....	14
3.2.1. Sơ đồ use case	14
3.3. Thiết kế hệ thống.....	16
3.3.1. Thiết kế cơ sở dữ liệu	16
3.3.2. Mô tả chi tiết các bảng	17

CHƯƠNG 4: KẾT QUẢ	20
4.1. cài đặt hệ thống	20
4.1.1. Cài đặt Visual Studio.....	20
4.1.2. Cài đặt SQL Server	20
4.2. Cài đặt giao diện	20
4.2.1. Giao diện trang chủ	20
4.2.2. Giao diện trang phát nhạc	21
4.2.3. Giao diện trang xếp hạng bài hát.....	21
4.2.4. Giao diện trang danh sách bài hát yêu thích.....	22
4.2.5. Giao diện trang quản lý bài hát	23
4.2.6. Giao diện quản lý ca sĩ	24
4.2.7. Giao diện trang quản lý tài khoản	24
4.3. Các bước chạy chương trình	25
4.4. Kết quả chạy thử	25
CHƯƠNG 5: KẾT LUẬN.....	27
5.1. Kết quả đạt được của đề tài.....	27
5.2. Hạn chế của đề tài	27
5.3. Kết luận	27
TÀI LIỆU THAM KHẢO	28

DANH MỤC HÌNH ẢNH

HÌNH 3.1: SƠ ĐỒ USE CASE TỔNG QUÁT	14
HÌNH 3.2: SƠ ĐỒ USE CASE ĐĂNG KÝ/ĐĂNG NHẬP	14
HÌNH 3.3: SƠ ĐỒ USE CASE XEM THÔNG TIN CÁ NHÂN.....	15
HÌNH 3.4: SƠ ĐỒ USE CASE XEM NGHE NHẠC	15
HÌNH 3.5: SƠ ĐỒ USE CASE TẠO PLAYLIST	15
HÌNH 3.6: SƠ ĐỒ USE CASE QUẢN LÝ BÀI HÁT	16
HÌNH 3.7: BIỂU ĐỒ DỮ LIỆU QUAN HỆ.....	16

DANH MỤC BẢNG

BẢNG 3.1. BẢNG ACCOUNT	17
BẢNG 3.2. BẢNG ALBUM	17
BẢNG 3.3. BẢNG MUSICGENRES	17
BẢNG 3.4. BẢNG SINGER	18
BẢNG 3.5. BẢNG MUSICIAN	18
BẢNG 3.6. BẢNG SONG.....	18
BẢNG 3.7. BẢNG FAVORITE.....	19

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài

Trong bối cảnh công nghệ số ngày càng phát triển mạnh, xu hướng nghe nhạc trực tuyến đang dần trở nên phổ biến và được người dùng ưa chuộng. Ngành công nghiệp âm nhạc – lĩnh vực gắn liền với nhu cầu giải trí và thưởng thức nghệ thuật – cũng không nằm ngoài xu thế đó, khi ngày càng phát triển mạnh mẽ nhờ vào sự hỗ trợ của các nền tảng số và ứng dụng trực tuyến.

Music Việt là một nền tảng âm nhạc uy tín, chuyên cung cấp các ca khúc, album và playlist đa dạng từ nhiều nghệ sĩ trong và ngoài nước. Tuy nhiên, hoạt động hiện tại của nền tảng vẫn chủ yếu dựa vào phương thức truyền thống, chưa tận dụng triệt để các lợi thế của môi trường trực tuyến để mở rộng phạm vi tiếp cận người nghe cũng như nâng cao chất lượng trải nghiệm người dùng.

Từ thực tế trên, em đã lựa chọn thực hiện đề tài “Xây dựng website nghe nhạc trực tuyến” với mong muốn ứng dụng công nghệ thông tin vào lĩnh vực âm nhạc, góp phần hỗ trợ nền tảng tiếp cận lượng người dùng lớn hơn, đồng thời nâng cao hiệu quả trong việc cung cấp dịch vụ nghe nhạc và quản lý nội dung số.

2. Mục tiêu đề tài

Đề tài hướng đến mục tiêu phát triển một website nghe nhạc trực tuyến, với việc tích hợp các chức năng cơ bản như sau:

- Quản lý danh mục bài hát, bao gồm các thông tin như tên bài hát, ca sĩ, thể loại, album, thời lượng, lượt nghe, v.v.
- Hỗ trợ người dùng trong việc tra cứu, tìm kiếm và nghe nhạc một cách nhanh chóng và tiện lợi.
- Quản lý thông tin người dùng, bao gồm lịch sử nghe nhạc, playlist cá nhân và các bài hát yêu thích.

- Cung cấp giao diện quản trị dành cho quản trị viên để dễ dàng theo dõi hoạt động hệ thống, thống kê lượt truy cập/nghe và cập nhật thông tin nội dung âm nhạc.

Website được phát triển trên nền tảng .NET Framework sử dụng mô hình MVC, kết hợp cùng các công nghệ web hiện đại nhằm đảm bảo hệ thống có khả năng mở rộng, vận hành ổn định và phù hợp với nhu cầu thực tế của người dùng.

3. Phương pháp nghiên cứu

- Tìm hiểu và thu thập thông tin về nhu cầu sử dụng dịch vụ nghe nhạc trực tuyến cũng như xu hướng phát triển hiện nay.
- Phân tích yêu cầu và thiết kế hệ thống thông tin phục vụ cho việc quản lý và cung cấp dịch vụ nghe nhạc trực tuyến.
- Vận dụng các công nghệ lập trình web để xây dựng hệ thống và kiểm thử tính năng nhằm đảm bảo hiệu quả sử dụng.

4. Bố cục chuyên đề

Chuyên đề được chia thành 3 chương chính:

Chương 1: Tổng quan đề tài

Chương 2: Cơ Sở lý thuyết

Chương 3: Nội dung thực hiện

Chương 4: Cài đặt và thử nghiệm

Chương 5: Kết luận

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Đối tượng mục tiêu nghiên cứu và phạm vi nghiên cứu

Đề tài tập trung nghiên cứu hai nhóm đối tượng chính. Nhóm đầu tiên là người dùng cuối – bao gồm tất cả cá nhân có nhu cầu nghe nhạc trực tuyến, không giới hạn về độ tuổi hay giới tính, có thể dễ dàng truy cập và sử dụng hệ thống để thưởng thức các ca khúc yêu thích, tạo playlist cá nhân và tương tác với nền tảng. Nhóm thứ hai là những người quản trị nội dung – cụ thể là quản trị viên hoặc nhân sự kỹ thuật của hệ thống, có nhiệm vụ cập nhật bài hát, quản lý thông tin nghệ sĩ, theo dõi hoạt động của người dùng và xử lý các nội dung vi phạm (nếu có).

Về phạm vi nghiên cứu, đề tài tập trung vào việc tìm hiểu và ứng dụng các kiến thức lý thuyết về phân tích và thiết kế hệ thống thông tin cho nền tảng âm nhạc trực tuyến. Song song đó, quá trình xây dựng hệ thống sử dụng cơ sở dữ liệu SQL Server và ngôn ngữ lập trình C# làm nền tảng chính. Môi trường phát triển Visual Studio được lựa chọn để xây dựng logic xử lý và thiết kế giao diện người dùng, đảm bảo hiệu suất cũng như tính trực quan của hệ thống.

1.2. Tính khả thi

Thị trường nghe nhạc trực tuyến đang có sự phát triển vượt bậc và thu hút sự quan tâm lớn từ phía người dùng. Việc sở hữu một nền tảng web nghe nhạc sẽ giúp tận dụng tối đa tiềm năng rộng lớn của lĩnh vực này, đáp ứng xu hướng tiêu dùng giải trí hiện đại, khi ngày càng nhiều người lựa chọn thưởng thức âm nhạc qua các thiết bị kết nối internet.

Người dùng khi nghe nhạc trực tuyến được hưởng lợi về sự thuận tiện và linh hoạt cao. Họ có thể dễ dàng tìm kiếm, khám phá các ca khúc, playlist phù hợp với sở thích, đồng thời nghe nhạc mọi lúc mọi nơi mà không cần cài đặt phức tạp hay phải

mua đĩa CD như trước. Điều này cũng góp phần đơn giản hóa quá trình quản lý và vận hành một nền tảng âm nhạc số.

Công nghệ và phần mềm hỗ trợ phát triển các hệ thống phát nhạc trực tuyến ngày càng hoàn thiện, cung cấp các công cụ mạnh mẽ giúp việc xây dựng, cập nhật và bảo trì website trở nên dễ dàng hơn. Các nền tảng mã nguồn mở và dịch vụ điện toán đám mây hiện đại giúp giảm chi phí triển khai, đồng thời tăng khả năng mở rộng và bảo mật cho hệ thống.

Về mặt kinh doanh, quản lý một nền tảng nghe nhạc trực tuyến không chỉ tạo nguồn thu từ quảng cáo hoặc gói người dùng trả phí, mà còn có thể mở rộng thông qua hợp tác bản quyền với nghệ sĩ, phân phối nội dung độc quyền, triển khai các chương trình gợi ý thông minh hoặc bán sản phẩm kỹ thuật số liên quan đến âm nhạc. Tất cả những yếu tố này tạo nên tiềm năng lợi nhuận cao và cơ hội phát triển bền vững trong ngành công nghiệp giải trí.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về ngôn ngữ ASP .NET

ASP.NET là nền tảng mạnh mẽ do Microsoft phát triển nhằm xây dựng các ứng dụng web có tính động, bảo mật và hiệu suất cao. Nó hỗ trợ nhiều mô hình lập trình đa dạng như Web Forms, MVC (Model-View-Controller), Web API và Blazor, giúp các lập trình viên dễ dàng lựa chọn công nghệ phù hợp với yêu cầu dự án.

Ưu điểm nổi bật của ASP.NET là khả năng tích hợp sâu với nền tảng .NET, tận dụng các tính năng như quản lý bộ nhớ, bảo mật, kiểm soát trạng thái, xác thực người dùng, bộ nhớ đệm (caching) và quản lý phiên làm việc. Nhờ đó, việc phát triển ứng dụng web trở nên thuận tiện, hiệu quả và an toàn hơn.

Kiến trúc của ASP.NET hỗ trợ việc phân tách các lớp chức năng rõ ràng, đặc biệt là qua mô hình MVC – phân chia giữa dữ liệu (Model), giao diện (View) và xử lý logic (Controller). Cách tổ chức này giúp ứng dụng dễ dàng mở rộng, bảo trì và thuận lợi cho việc thực hiện kiểm thử đơn vị (unit testing).

Ngoài ra, ASP.NET còn nổi bật với khả năng mở rộng tốt, hiệu năng cao cùng với một cộng đồng phát triển rộng lớn. Nó là lựa chọn phổ biến trong nhiều dự án như hệ thống quản lý doanh nghiệp, cổng thông tin, nền tảng thương mại điện tử và các dịch vụ web.

Tóm lại, ASP.NET là một nền tảng hiện đại, toàn diện, đáp ứng đầy đủ nhu cầu phát triển từ các dự án đơn giản đến phức tạp, giúp triển khai và vận hành hệ thống một cách hiệu quả.

2.1.1. Giới thiệu ASP.NET Framework

ASP.NET là một framework ứng dụng web do Microsoft phát triển và cung cấp, giúp lập trình viên tạo ra các trang web động, ứng dụng web và dịch vụ web. Trên nền tảng ASP.NET, ASP.NET MVC hỗ trợ các nhà phát triển phần mềm xây dựng

ứng dụng web theo kiến trúc MVC. Mô hình MVC là một design pattern tiêu chuẩn, dùng để phân chia ứng dụng thành ba thành phần chính gồm model, view và controller.

2.1.2. Kiến trúc ASP.NET MVC

Controller đóng vai trò là trung gian kết nối giữa người dùng và hệ thống. Nhiệm vụ của nó là nhận các yêu cầu từ phía người dùng, xử lý các logic liên quan, sau đó tạo ra và cấu hình Model phù hợp. Sau khi hoàn tất xử lý, Controller sẽ gửi Model đó đến View để dữ liệu được trình bày một cách trực quan cho người dùng.

View chính là phần giao diện của ứng dụng, chịu trách nhiệm hiển thị thông tin tới người dùng. Nó nhận Model từ Controller và dùng dữ liệu trong Model để hiển thị nội dung tương ứng. View không thực hiện xử lý logic nghiệp vụ mà chỉ tập trung vào việc thể hiện dữ liệu.

Model đại diện cho dữ liệu cũng như các logic nghiệp vụ của ứng dụng. Đây là nơi lưu trữ và quản lý thông tin, đồng thời đảm bảo tính chính xác và toàn vẹn của dữ liệu. Model kết nối Controller và View, giúp trao đổi dữ liệu hiệu quả và duy trì sự phân tách rõ ràng giữa giao diện và phần xử lý logic.

2.1.3. Lý do chọn Aps.NET MVC

Việc sử dụng ASP.NET MVC làm nền tảng để phát triển ứng dụng web xuất phát từ nhiều điểm mạnh nổi bật, đặc biệt là khả năng phân tách rõ ràng các nhiệm vụ trong ứng dụng như xử lý logic nhập liệu, nghiệp vụ và giao diện người dùng. Mô hình phân tầng MVC không chỉ giúp tổ chức mã nguồn một cách khoa học và dễ dàng bảo trì, mà còn hỗ trợ đắc lực cho việc kiểm thử, đặc biệt là kiểm thử đơn vị. ASP.NET MVC được thiết kế với định hướng phát triển theo phương pháp TDD (Test Driven Development), cho phép lập trình viên viết mã kiểm thử trước khi triển khai logic chính thức.

Các tính năng chủ đạo của MVC được xây dựng dựa trên các interface, tạo điều kiện thuận lợi để sử dụng các đối tượng giả lập (mock objects) trong quá trình kiểm thử. Các mock này mô phỏng hành vi của các thành phần thực tế trong ứng dụng, giúp kiểm thử diễn ra độc lập mà không cần chạy toàn bộ hệ thống ASP.NET thật sự. Điều

này giúp tăng tốc độ phát triển và tạo sự linh hoạt khi dùng các framework kiểm thử như MSTest, xUnit hoặc NUnit — tất cả đều tương thích với nền tảng .NET.

ASP.NET MVC còn được biết đến với khả năng mở rộng (extensibility) và tích hợp linh hoạt (pluggability). Tất cả thành phần trong framework — từ view engine, routing, binding tham số đến action filters — đều có thể được tùy biến hoặc thay thế để phù hợp với nhu cầu riêng của dự án. Bên cạnh đó, ASP.NET MVC hỗ trợ mạnh mẽ các kỹ thuật hiện đại như Dependency Injection (DI) và Inversion of Control (IoC). Nhờ DI, các lớp không cần tự tạo đối tượng phụ thuộc mà sẽ nhận chúng từ bên ngoài, giúp giảm sự phụ thuộc giữa các thành phần và nâng cao khả năng kiểm thử. IoC giúp hệ thống được kết nối một cách lỏng lẻo, dễ dàng thay đổi, quản lý và tuân thủ các nguyên tắc thiết kế hướng đối tượng hiện đại.

Ngoài ra, ASP.NET MVC vẫn tận dụng được nhiều tính năng mạnh mẽ sẵn có của nền tảng ASP.NET như xác thực người dùng, phân quyền truy cập, quản lý phiên làm việc, caching, cấu hình hệ thống và các tính năng bảo mật cùng tối ưu hiệu năng. Điều này giúp rút ngắn đáng kể thời gian xây dựng các chức năng nền tảng, để lập trình viên có thể tập trung phát triển nghiệp vụ chính.

Phiên bản ASP.NET MVC 5 được trang bị Razor View Engine — công cụ giúp xây dựng giao diện nhanh gọn và hiệu quả hơn nhiều so với Web Forms View Engine. Razor có cú pháp đơn giản, dễ đọc, dễ viết và tích hợp chặt chẽ giữa C# và HTML, giúp tiết kiệm thời gian và công sức cho lập trình viên khi thiết kế giao diện người dùng.

Tổng thể, tất cả những đặc điểm này cho thấy ASP.NET MVC là lựa chọn phù hợp cho các ứng dụng web hiện đại, yêu cầu cao về khả năng kiểm thử, mở rộng và bảo trì lâu dài.

2.2. Javascript

JavaScript được tạo ra bởi Brendan Eich khi ông làm việc tại Netscape, và lần đầu tiên được tích hợp vào trình duyệt Navigator 2.0 vào năm 1995. Kể từ năm 1996, cả Netscape lẫn Microsoft đều bắt đầu tích hợp JavaScript vào trình duyệt của họ.

Hiện nay, hầu hết các trình duyệt phổ biến đều tích hợp sẵn JavaScript. Chính JavaScript cũng là một trong những yếu tố quan trọng góp phần khơi mào cho cuộc cạnh tranh giữa các trình duyệt hiện đại. Mặc dù đã tồn tại từ rất lâu, JavaScript vẫn

giữ vị trí then chốt trong việc phát triển web. Do đó, việc học và nắm vững JavaScript là điều không thể thiếu đối với bất kỳ ai theo đuổi lập trình web.

JavaScript là gì?

Về cách thực thi mã lệnh:

Trong phát triển web hiện nay, mã chương trình thường được xử lý theo hai cách: phía máy chủ (Server-Side) và phía trình duyệt (Client-Side).

- **Xử lý phía trình duyệt (Client-Side):** Mã sẽ được thực thi trực tiếp trên máy tính của người truy cập thông qua trình duyệt, và chỉ kết quả mới được gửi lên máy chủ.
- **Xử lý phía máy chủ (Server-Side):** Mã sẽ được gửi đến máy chủ để xử lý, sau đó kết quả được trả về trình duyệt của người dùng.

JavaScript là một ngôn ngữ kịch bản phía trình duyệt, nghĩa là các đoạn mã JavaScript được chạy trên thiết bị của người dùng thay vì trên máy chủ. Do đó, JavaScript không thể truy cập trực tiếp vào cơ sở dữ liệu nằm trên server.

Về phương diện phong cách lập trình

Ban đầu, JavaScript được xem là một ngôn ngữ dựa trên đối tượng (Object-Based). Tuy nhiên, từ khi Ajax phát triển, JavaScript đã dần tiếp cận phong cách lập trình hướng đối tượng (OOP). Ngày nay, có thể coi JavaScript là một ngôn ngữ hướng đối tượng.

Cú pháp của JavaScript có nhiều điểm tương đồng với các ngôn ngữ như C, Perl, và Java – đặc biệt là ở các cấu trúc điều khiển như if, while, hay for.

Tóm lại, JavaScript là một ngôn ngữ lập trình:

- Chạy ở phía trình duyệt.
- Có khả năng lập trình theo hướng đối tượng.

JavaScript được sử dụng để làm gì?

- Tăng cường tính tương tác cho các trang HTML.
- Xử lý các hành động như sự kiện người dùng hoặc kiểm tra form đầu vào.
- Xác thực dữ liệu trước khi gửi lên máy chủ.
- Làm cho trang web phản hồi linh hoạt hơn với người dùng.
- Tùy chỉnh hành vi của trình duyệt.

Ưu điểm của JavaScript

- Miễn phí hoàn toàn và dễ tiếp cận đối với người mới học.

- Hoạt động trên hầu hết các hệ điều hành nếu có trình duyệt hỗ trợ.
- Giúp giảm tải công việc cho server bằng cách xử lý ngay trên trình duyệt.
- Là nền tảng kiến thức vững chắc để học các công nghệ hiện đại như Ajax, Atlas,...

Nhược điểm của JavaScript

- Không có trình biên dịch riêng biệt; mã được trình duyệt dịch trực tiếp khi chạy. Nếu trình duyệt không hỗ trợ JavaScript hoặc bị tắt, chương trình sẽ không chạy được.
- Có thể gây nặng nề cho website nếu dùng không hợp lý.
- Mức độ bảo mật thấp, vì mã nguồn dễ bị người khác xem và chỉnh sửa.

Nguyên tắc lập trình trong JavaScript

- Phân biệt chữ hoa và chữ thường trong tên biến, hàm,...
- Kết thúc mỗi dòng lệnh bằng dấu ;.
- Điều kiện phải nằm trong ngoặc tròn ().
- Khi có nhiều câu lệnh bên trong các cấu trúc điều khiển như if, while, hoặc for, cần sử dụng ngoặc nhọn { } để nhóm lệnh.
- Dấu . được dùng để truy cập thuộc tính hoặc gọi phương thức của một đối tượng.

Ajax là gì?

Ajax (viết tắt của Asynchronous JavaScript and XML) là một kỹ thuật lập trình web cho phép trang web giao tiếp với máy chủ và cập nhật nội dung một phần của trang mà không cần phải tải lại toàn bộ trang web. Điều này mang lại trải nghiệm mượt mà và nhanh chóng hơn cho người dùng.

Trong lĩnh vực phát triển web hiện đại, Ajax đóng một vai trò thiết yếu trong việc xây dựng các ứng dụng web có tính tương tác cao. Vì dựa trên nền tảng JavaScript, Ajax hoạt động trực tiếp trên trình duyệt của người dùng (client-side). Do đó, mỗi người truy cập đều xử lý riêng biệt và không gây ảnh hưởng lẫn nhau.

Ngày nay, có rất nhiều thư viện và framework JavaScript như jQuery, Angular,... được phát triển để hỗ trợ việc sử dụng Ajax một cách thuận tiện và hiệu quả hơn, giúp lập trình viên thao tác nhanh chóng với ít mã lệnh hơn.

2.3. Bootstrap

Bootstrap là một framework giao diện người dùng phía trước (Front-end), được xây dựng bằng SASS và sau đó biên dịch ra CSS. Đây là một thư viện tích hợp sẵn nhiều đoạn mã CSS tiện ích, giúp lập trình viên viết HTML và CSS dễ dàng, nhanh chóng hơn. Trong khi việc viết CSS thuần túy thường khiến người lập trình phải đối mặt với việc kiểm tra độ tương thích trên nhiều trình duyệt và thiết bị, thì Bootstrap đã giải quyết phần lớn vấn đề này. Những người phát triển Bootstrap đã tối ưu để đảm bảo sự tương thích cao, giảm đáng kể công sức cho lập trình viên.

Bootstrap sử dụng hệ thống lưới (grid system) để chia bố cục trang web thành 12 cột dựa trên chiều rộng 960px, giúp dễ dàng xây dựng các thiết kế phản hồi (responsive). Framework này còn đi kèm với nhiều thành phần phổ biến như: menu, tab, tooltip, popup,... để hỗ trợ xây dựng giao diện nhanh hơn. Ngoài CSS, Bootstrap còn tích hợp các hiệu ứng động thông qua JavaScript, cụ thể là sử dụng jQuery. Vì thế, để tận dụng toàn bộ tính năng của Bootstrap, cần phải thêm thư viện jQuery vào dự án.

Một số ưu điểm khi dùng Bootstrap:

- Giúp tạo giao diện nhanh và chuẩn.
- Học và sử dụng dễ dàng.

Hỗ trợ nhiều hiệu ứng JavaScript nhờ jQuery

2.4. Tổng quan về SQL

2.4.1. Hệ quản trị cơ sở dữ liệu SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh, sử dụng ngôn ngữ Transact-SQL (T-SQL) để trao đổi dữ liệu giữa client và server. Một hệ thống RDBMS bao gồm các cơ sở dữ liệu (Databases), bộ xử lý cơ sở dữ liệu (Database engine) cùng nhiều thành phần quản lý và ứng dụng liên quan.

SQL Server được thiết kế tối ưu để hoạt động hiệu quả trong các môi trường dữ liệu quy mô rất lớn (Very Large Database Environment), có thể lên tới hàng Terabyte và hỗ trợ đồng thời hàng nghìn kết nối người dùng. Phiên bản SQL Server 2000 đặc biệt tích hợp tốt với nhiều dịch vụ khác của Microsoft như Internet Information Server (IIS), E-Commerce Server, Proxy Server, và các dịch vụ máy chủ khác.

2.4.2. Đặc điểm của SQL Server và đối tượng làm việc

SQL là một ngôn ngữ tương tự tiếng Anh, giúp người dùng dễ dàng trong việc đọc, hiểu và soạn thảo các câu lệnh truy vấn dữ liệu.

SQL hoạt động theo kiểu phi thủ tục, nghĩa là người dùng chỉ cần xác định dữ liệu hoặc thao tác cần thực hiện mà không phải lo lắng về cách thức thực hiện chi tiết. Hệ thống như SQL Server sẽ tự động tối ưu quá trình truy xuất và xử lý dữ liệu.

Ngôn ngữ SQL cung cấp một bộ lệnh đa dạng, bao gồm: chèn dữ liệu (INSERT), cập nhật (UPDATE), xóa bỏ (DELETE) các bản ghi trong bảng; tạo mới (CREATE), thay đổi (ALTER), xóa bỏ (DROP) các đối tượng trong cơ sở dữ liệu; cũng như kiểm soát quyền truy cập (GRANT, REVOKE) để đảm bảo an toàn cho dữ liệu.

Ngoài ra, SQL còn hỗ trợ các cơ chế đảm bảo tính nhất quán của dữ liệu thông qua các ràng buộc (Constraints) và quan hệ giữa các bảng (Relationships). Để sử dụng hiệu quả SQL, người dùng cần hiểu rõ cấu trúc của cơ sở dữ liệu mà họ thao tác.

2.4.3. Các đối tượng quan trọng trong SQL Server

Bên cạnh các bảng (Tables) và mối quan hệ (Relationships) đã đề cập trước đó, SQL Server còn hỗ trợ nhiều đối tượng mạnh mẽ khác nhằm quản lý và thao tác dữ liệu một cách hiệu quả hơn:

a. Store Procedure (Thủ tục lưu trữ)

Là tập hợp các câu lệnh SQL được biên dịch và lưu trữ trực tiếp trên máy chủ SQL Server. Store Procedure có thể được gọi và thực thi nhiều lần bởi các ứng dụng hoặc người dùng khác nhau.

❖ Ưu điểm:

- Tái sử dụng mã nguồn, tránh viết lại các đoạn SQL phức tạp nhiều lần.
- Do được biên dịch sẵn, Store Procedure thường có hiệu suất cao hơn so với các câu lệnh SQL riêng lẻ.
- Giảm lưu lượng truyền tải mạng bằng cách chỉ gửi lệnh gọi thay vì nhiều câu lệnh SQL rời.
- Nâng cao bảo mật khi chỉ cần cấp quyền chạy thủ tục mà không cho phép truy cập trực tiếp vào bảng.
- Đảm bảo tính nhất quán cho các quy trình nghiệp vụ phức tạp.

b. Function (Hàm)

Trong SQL Server, Function tương tự như hàm trong các ngôn ngữ lập trình, bao gồm các khối mã trả về một giá trị đơn (Scalar Function) hoặc một bảng dữ liệu (Table-Valued Function).

❖ *Lợi ích:*

- Thực hiện các phép tính phức tạp và trả về một giá trị duy nhất.
- Tạo ra các bảng ảo dựa trên logic nghiệp vụ, có thể dùng trong các truy vấn SELECT như bảng thật.
- Giúp đóng gói các logic nghiệp vụ để tái sử dụng ở nhiều nơi trong cơ sở dữ liệu.
- Nâng cao tính rõ ràng và dễ bảo trì bằng cách chia nhỏ các truy vấn phức tạp.

c. Indexing (Chỉ mục)

Index trong SQL Server tương tự như mục lục trong sách, là cấu trúc dữ liệu đặc biệt được tạo trên một hoặc nhiều cột của bảng nhằm tăng tốc độ tìm kiếm và truy xuất dữ liệu.

❖ *Lợi ích:*

- Giúp truy vấn dữ liệu nhanh hơn bằng cách định vị các bản ghi phù hợp mà không cần quét toàn bộ bảng.
- Nâng cao hiệu suất cho các bảng lớn và các truy vấn phức tạp.
- Có thể đảm bảo tính duy nhất cho các giá trị trong cột hoặc tổ hợp cột thông qua Unique Index.

❖ *Các loại Index phổ biến:*

- Clustered Index: Quy định cách dữ liệu được lưu trữ vật lý trên bảng, mỗi bảng chỉ có một Clustered Index, thường là trên cột khóa chính.
- Non-clustered Index: Là cấu trúc riêng biệt chứa các con trỏ đến dữ liệu, cho phép một bảng có nhiều Non-clustered Index.

❖ *Lưu ý khi sử dụng Index:*

- Việc tạo và duy trì chỉ mục đòi hỏi thêm không gian lưu trữ và có thể làm chậm thao tác INSERT, UPDATE, DELETE do phải cập nhật chỉ mục.
- Cần chọn lựa cẩn thận các cột dùng nhiều trong điều kiện WHERE, JOIN hoặc ORDER BY để tạo chỉ mục hiệu quả.
- Tránh tạo quá nhiều chỉ mục trên cùng một bảng vì có thể làm giảm hiệu năng tổng thể.

CHƯƠNG 3: THỰC HIỆN HÓA NGHIÊN CỨU

3.1. Khảo sát hiện trạng

Hiện nay, các nền tảng nghe nhạc trực tuyến như Zing MP3, Spotify, Nhaccuatui,... đã và đang phát triển mạnh mẽ, cung cấp trải nghiệm âm nhạc đa dạng và tiện lợi cho người dùng. Tuy nhiên, đối với các cá nhân, nhóm phát triển nhỏ hoặc sinh viên, việc xây dựng được một nền tảng nghe nhạc đơn giản, thân thiện, dễ sử dụng và có tính năng cơ bản là một nhu cầu thiết yếu để:

Lưu trữ, quản lý và phát nhạc cá nhân.

Trải nghiệm quy trình phát triển một hệ thống web hoàn chỉnh.

Rèn luyện kỹ năng lập trình thực tế từ frontend, backend, đến xử lý dữ liệu đa phương tiện.

Khảo sát cho thấy:

- Nhiều sinh viên hoặc cá nhân yêu nhạc hiện vẫn sử dụng Google Drive, Dropbox để chia sẻ nhạc, thiếu tính năng tìm kiếm, playlist, phân loại bài hát...
- Chưa có hệ thống cá nhân hóa trải nghiệm nghe nhạc (playlist, bài hát yêu thích, gợi ý theo lượt nghe).
- Không có khu vực quản trị để kiểm soát nội dung nhạc, người dùng hay thống kê lượt truy cập/nghe bài.

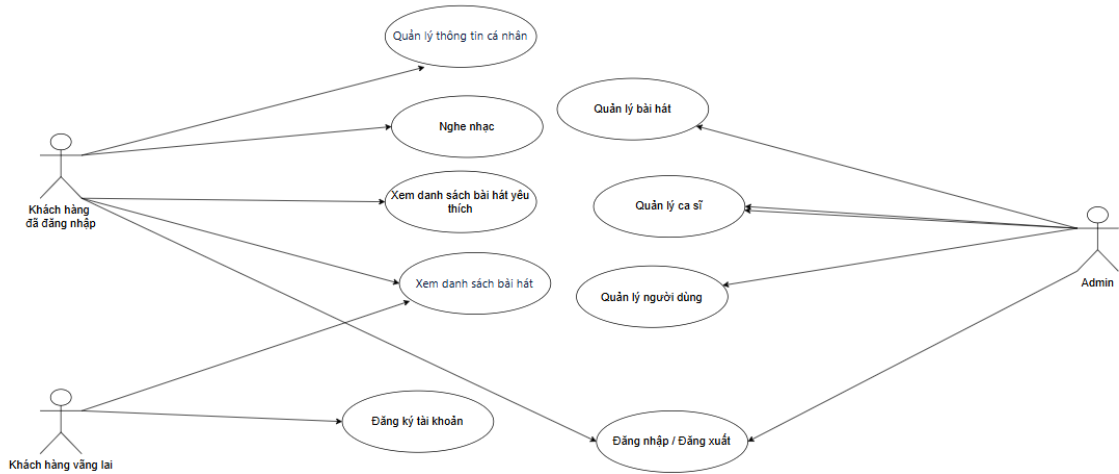
Từ thực trạng này, đề tài xây dựng một website nghe nhạc trực tuyến là cần thiết, nhằm:

- Cung cấp nền tảng lưu trữ và nghe nhạc trực tuyến cá nhân hoặc nhóm sử dụng.
- Quản lý bài hát, thể loại, ca sĩ, playlist một cách tập trung và hiệu quả.
- Mang lại trải nghiệm thân thiện và tiện ích cho người dùng yêu âm nhạc.

3.2. Phân tích hệ thống

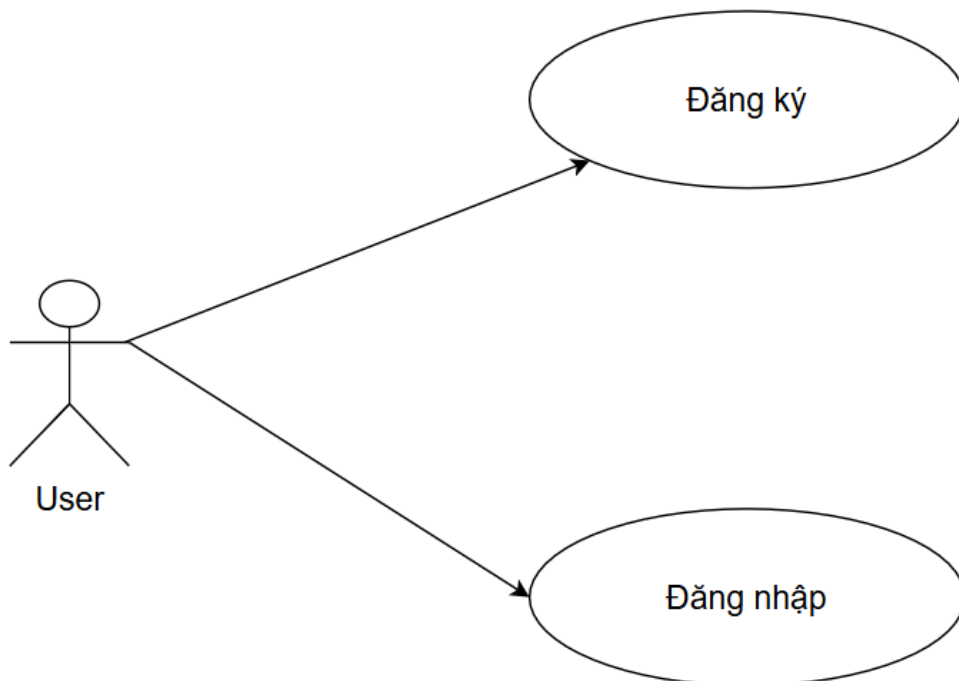
3.2.1. Sơ đồ use case

3.2.1.1. Sơ đồ use case tổng quát



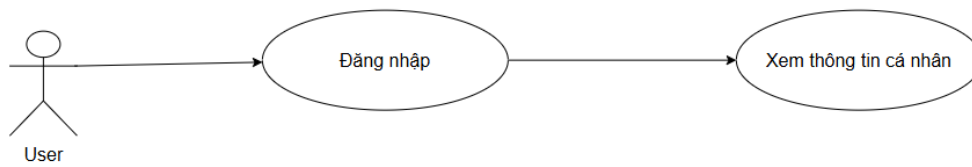
Hình 3.1: Sơ đồ use case tổng quát

3.2.1.2. Sơ đồ use case đăng ký/dăng nhập



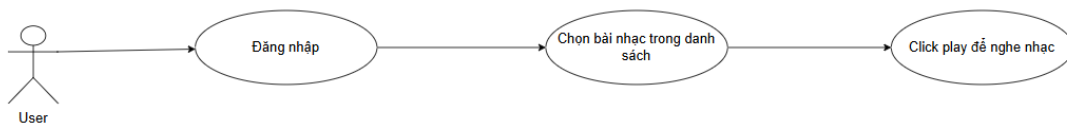
Hình 3.2: Sơ đồ use case đăng ký/dăng nhập

3.2.1.3. Sơ đồ use case xem thông tin cá nhân



Hình 3.3: Sơ đồ use case xem thông tin cá nhân

3.2.1.4. Sơ đồ use case nghe nhạc



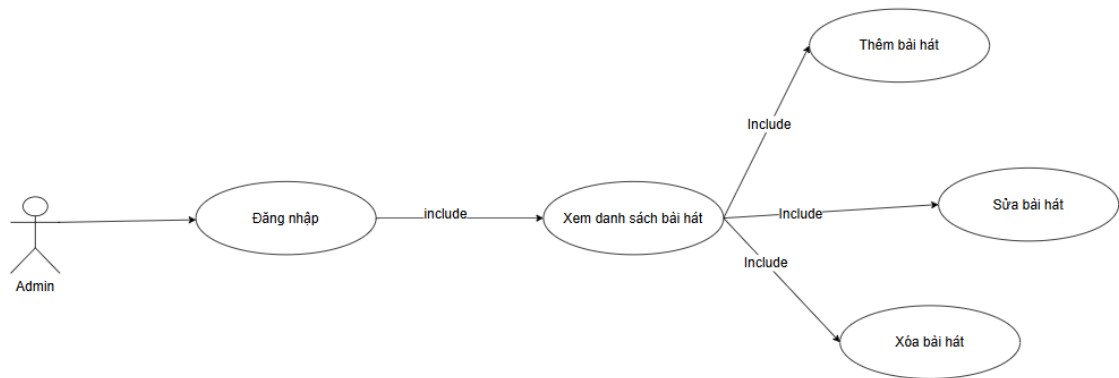
Hình 3.4: Sơ đồ use case xem nghe nhạc

3.2.1.5. Sơ đồ use case tạo playlist



Hình 3.5: Sơ đồ use case tạo playlist

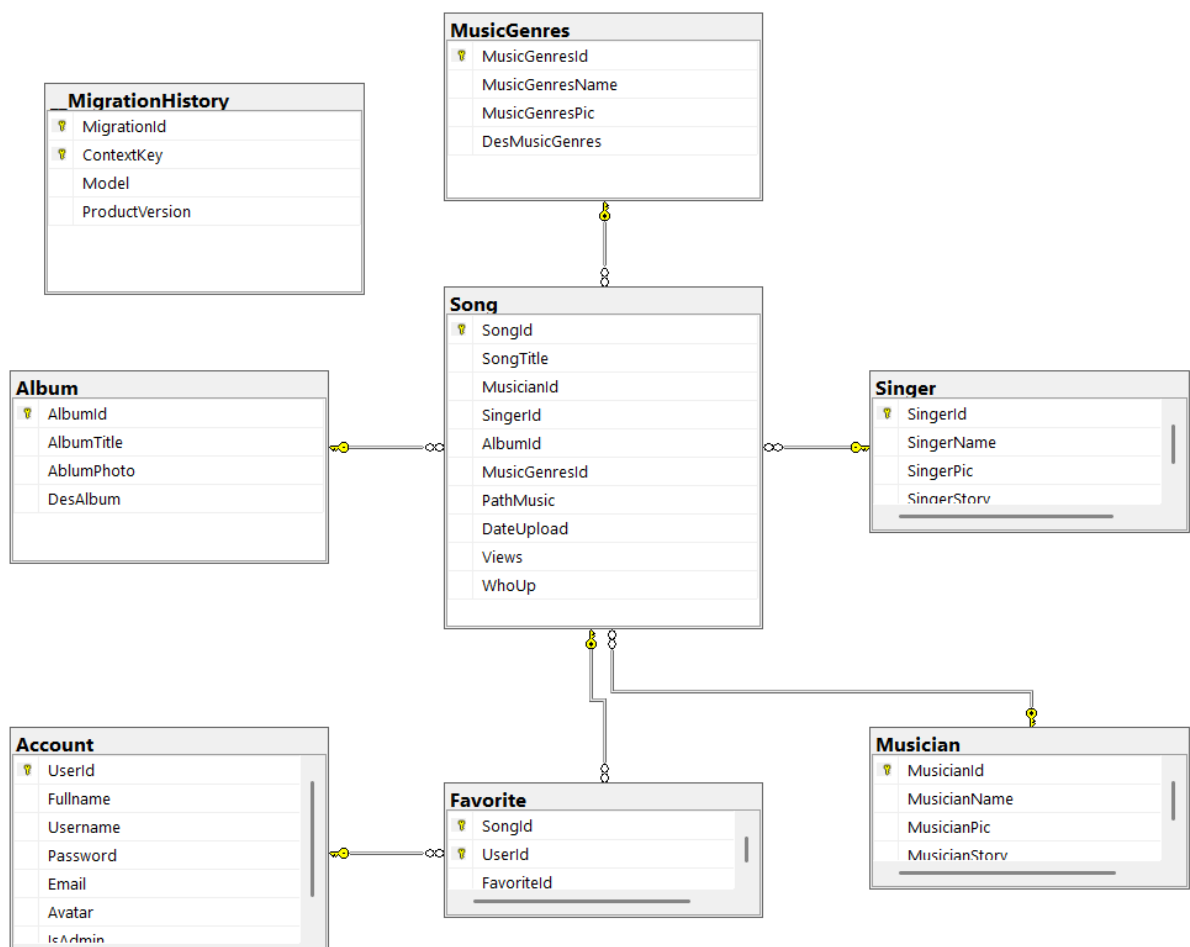
3.2.1.6. Sơ đồ use case quản lý bài hát



Hình 3.6: Sơ đồ use case quản lý bài hát

3.3. Thiết kế hệ thống

3.3.1. Thiết kế cơ sở dữ liệu



Hình 3.77: Biểu đồ dữ liệu quan hệ

3.3.2. Mô tả chi tiết các bảng

Bảng Account

Mô tả: Lưu trữ thông tin người dùng của hệ thống. Bao gồm thông tin đăng nhập, email, ảnh đại diện và quyền.

Tên cột	Kiểu dữ liệu	Mô tả
Tên cột	Kiểu dữ liệu	Mô tả
UserId	int	Khóa chính, mã người dùng
Fullname	nvarchar(255)	Họ tên đầy đủ
Username	varchar(64)	Tên đăng nhập
Password	varchar(8000)	Mật khẩu
Email	nvarchar(100)	Địa chỉ email
Avatar	varchar(255)	Ảnh đại diện
IsAdmin	tinyint	Có phải quản trị viên

Bảng 3.1. Bảng Account

Bảng Album

Mô tả: Chứa thông tin về các album nhạc như tiêu đề, ảnh đại diện, mô tả nội dung..

Tên cột	Kiểu dữ liệu	Mô tả
AlbumId	int	Khóa chính, mã album
AlbumTitle	nvarchar(100)	Tên album
AlbumPhoto	varchar(255)	Ảnh đại diện album
DesAlbum	nvarchar(500)	Mô tả album

Bảng 3.2. Bảng Album

Bảng MusicGenres

Mô tả: Lưu thông tin các thể loại nhạc như tên thể loại, hình ảnh và mô tả.

Tên cột	Kiểu dữ liệu	Mô tả
MusicGenreId	int	Khóa chính, mã thể loại
MusicGenreName	nvarchar(100)	Tên thể loại nhạc
MusicGenrePic	varchar(255)	Hình ảnh thể loại
DesMusicGenres	nvarchar(500)	Mô tả thể loại

Bảng 3.3. Bảng MusicGenres

Bảng Singer

Mô tả: Lưu trữ thông tin ca sĩ thể hiện bài hát, gồm tên, ảnh đại diện và tiểu sử.

Tên cột	Kiểu dữ liệu	Mô tả
SingerId	int	Khóa chính, mã ca sĩ
SingerName	nvarchar(255)	Tên ca sĩ
SingerPic	varchar(255)	Ảnh ca sĩ
SingerStory	nvarchar(1000)	Tiểu sử ca sĩ

Bảng 3.4. Bảng Singer

Bảng Musician

Mô tả: Chứa thông tin về người sáng tác (nhạc sĩ) như tên, hình ảnh và tiểu sử.

Tên cột	Kiểu dữ liệu	Mô tả
MusicianId	int	Khóa chính, mã nhạc sĩ
MusicianName	nvarchar(255)	Tên nhạc sĩ
MusicianPic	varchar(255)	Ảnh nhạc sĩ
MusicianStory	nvarchar(1000)	Tiểu sử nhạc sĩ

Bảng 3.5. Bảng Musician

Bảng Song

Mô tả: Bảng trung tâm chứa thông tin về bài hát bao gồm tên, thể loại, nhạc sĩ, ca sĩ, album, đường dẫn file nhạc, ngày tải lên, lượt xem và người đăng tải.

Tên cột	Kiểu dữ liệu	Mô tả
SongId	int	Khóa chính, mã bài hát
SongTitle	nvarchar(100)	Tên bài hát
MusicianId	int	Khóa ngoại đến bảng Musician
SingerId	int	Khóa ngoại đến bảng Singer
AlbumId	int	Khóa ngoại đến bảng Album
MusicGenreId	int	Khóa ngoại đến bảng MusicGenres
PathMusic	varchar(255)	Đường dẫn đến file nhạc
DateUpload	date	Ngày tải lên
Views	int	Lượt xem
WhoUp	nvarchar(100)	Người đăng tải

Bảng 3.6. Bảng Song

Bảng Favorite

Mô tả: Lưu các bài hát mà người dùng đã đánh dấu yêu thích. Mỗi bản ghi là một cặp bài hát và người dùng.

Tên cột	Kiểu dữ liệu	Mô tả
SongId	int	Khóa ngoại đến bảng Song
UserId	int	Khóa ngoại đến bảng Account
FavoriteId	int	Khóa chính hoặc mã yêu thích (tùy)

Bảng 3.7. Bảng Favorite

CHƯƠNG 4: KẾT QUẢ

4.1. Cài đặt hệ thống

4.1.1. Cài đặt Visual Studio

- Tải visual studio:
 - Truy cập: <https://visualstudio.microsoft.com>
 - Chọn bản: visual studio 2022 community (miễn phí cho cá nhân, sinh viên)
- Trong quá trình cài đặt, chọn các workload sau:
 - ASP .NET and web development
 - Data storage and processing (để làm việc với sql server)

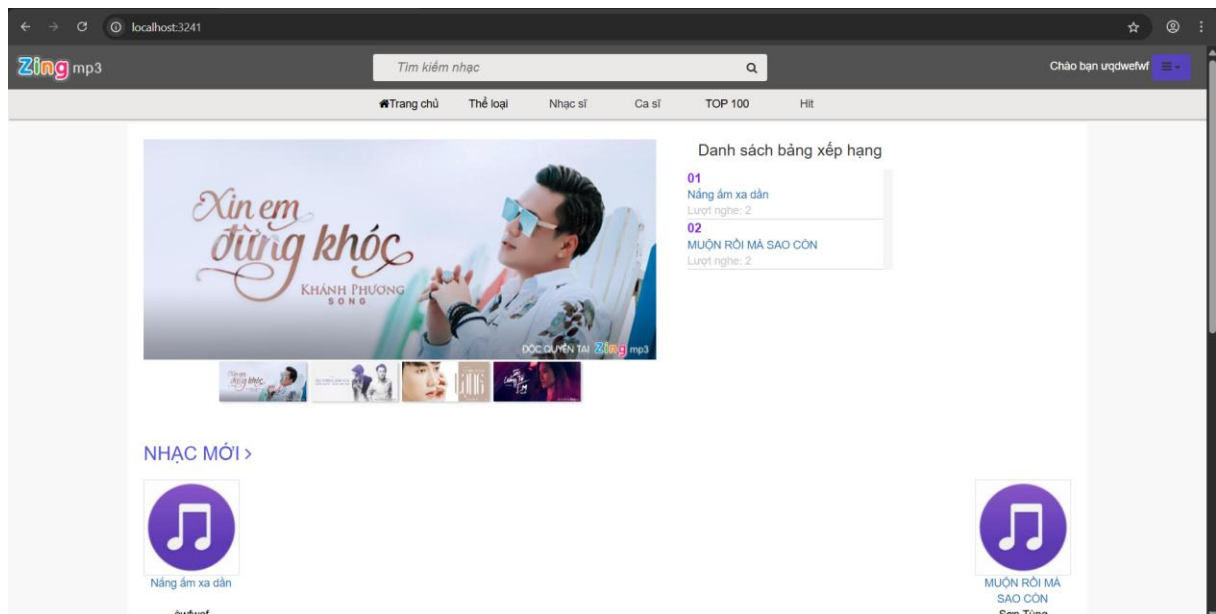
4.1.2. Cài đặt SQL Server

- Tải sql server:
 - Truy cập: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>
 - Chọn bản: SQL Server 2022 Express (miễn phí)
- Cài đặt SQL Server Management Studio (SSMS):
 - Truy cập: <https://aka.ms/ssms>
 - SSMS là công cụ để quản lý cơ sở dữ liệu sql server bằng giao diện trực quan.

4.2. Cài đặt giao diện

4.2.1. Giao diện trang chủ

- Giao diện hiển thị danh mục bài hát, mục Đăng ký và Đăng nhập.
- Từ trang chủ, người dùng có thể tiếp cận toàn bộ chức năng của hệ thống.



Hình 4.1: Hình ảnh giao diện trang chủ

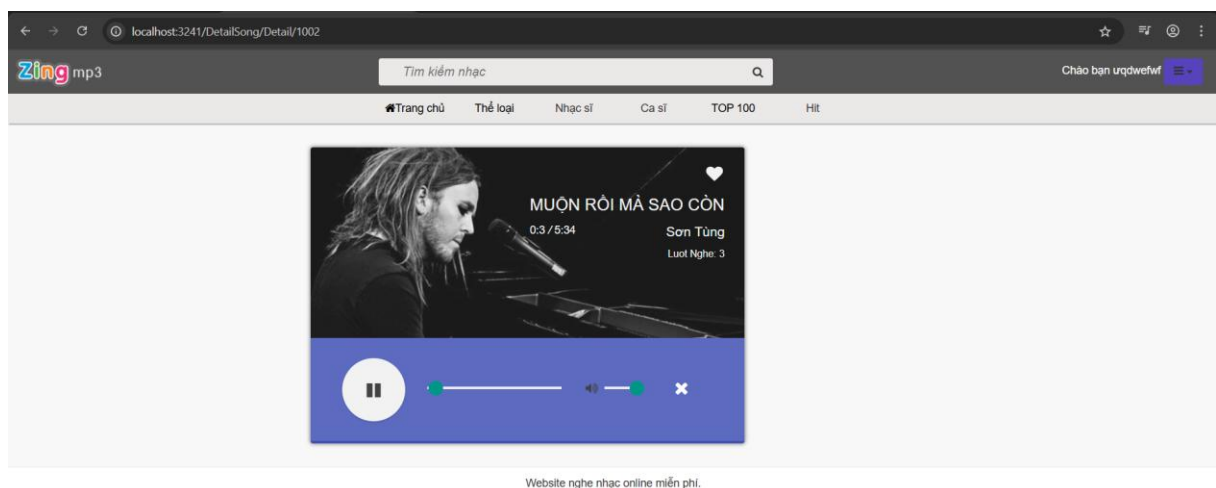
4.2.2. Giao diện trang phát nhạc

Chức năng:

- Hiện thị thông tin bài hát: tên, ca sĩ, ảnh bìa.

Hành động người dùng:

- Nghe nhạc
- Thêm vào playlist hoặc thích bài hát



Hình 4.2: Hình ảnh giao diện trang phát nhạc

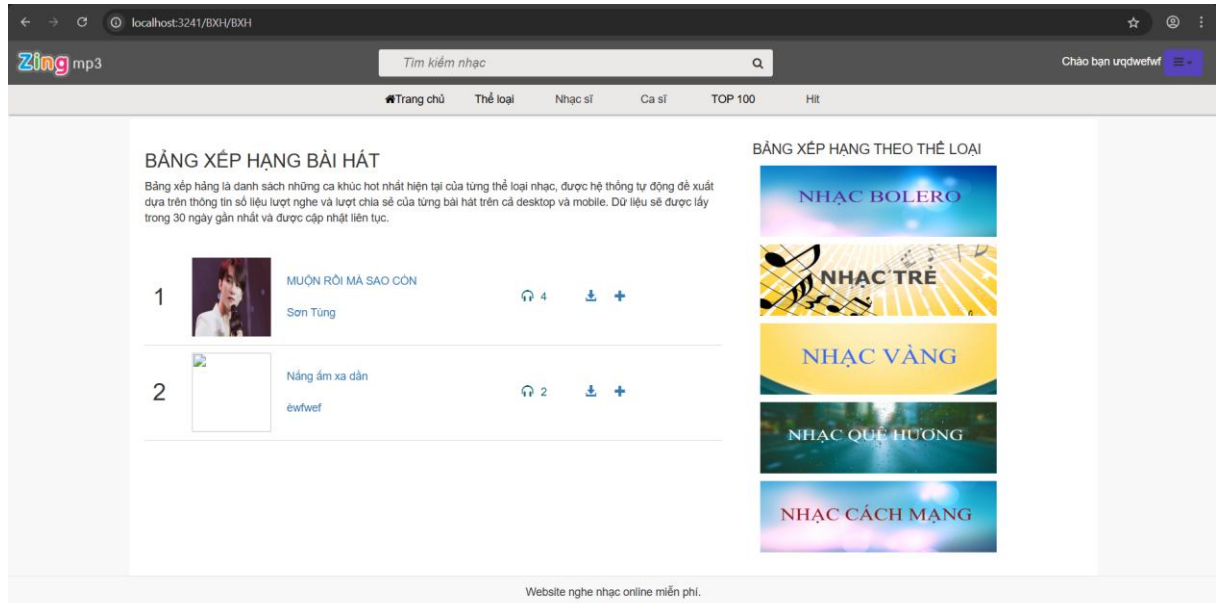
4.2.3. Giao diện trang xếp hạng bài hát

Chức năng:

- Hiện thị danh sách bài hát nghe nhiều nhất

Hành động người dùng:

- Nghe nhạc
- Thêm vào playlist phát



Hình 4.3: Hình ảnh giao diện trang xếp hạng bài hát

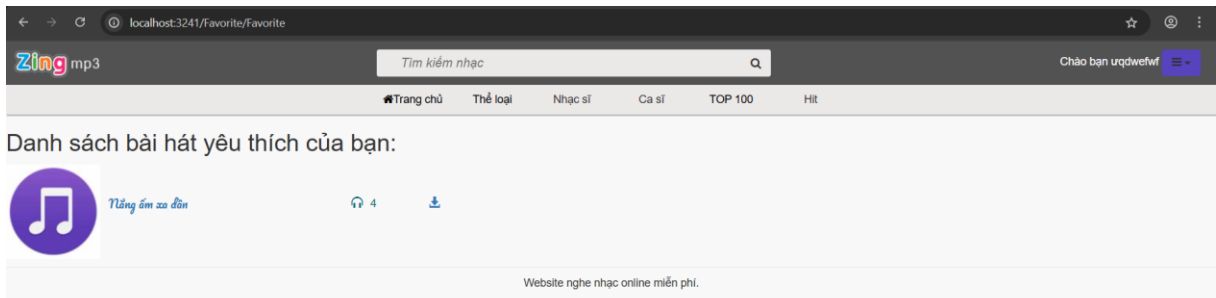
4.2.4. Giao diện trang danh sách bài hát yêu thích

Chức năng:

- Cho phép người dùng xem danh sách bài hát yêu thích.

Hành động người dùng:

- Nghe nhạc
- Thêm vào playlist phát
- Download nhạc về máy



Hình 4.4: Hình ảnh giao diện trang danh sách bài yêu thích

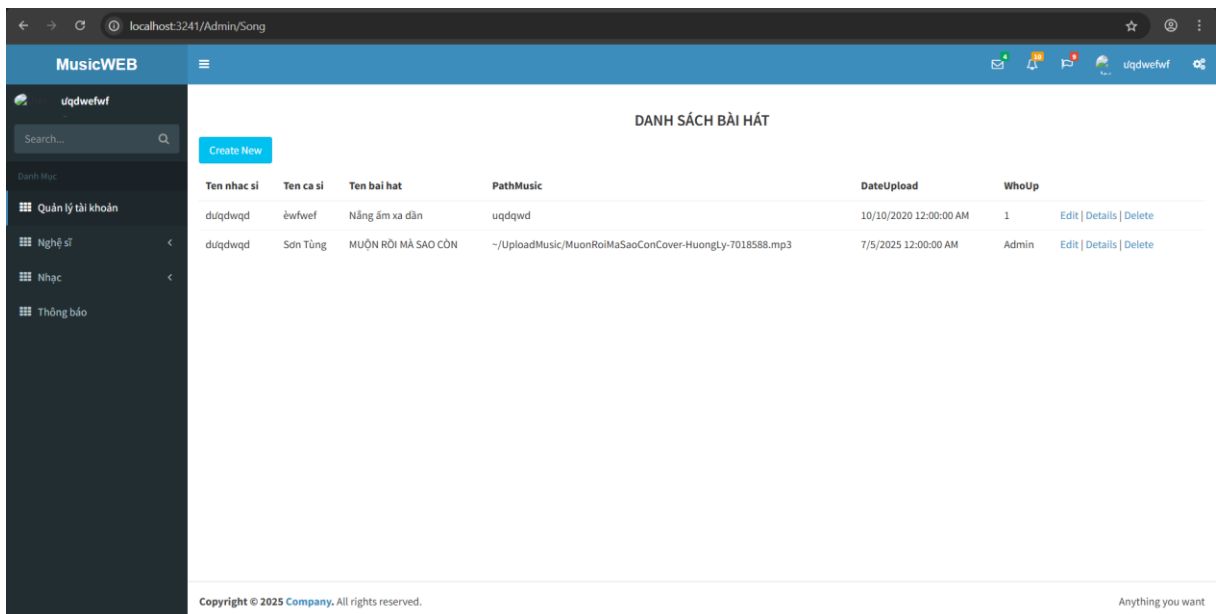
4.2.5. Giao diện trang quản lý bài hát

Chức năng:

- Cho phép quản trị viên quản lý danh sách bài hát.

Hành động người dùng:

- Thêm bài hát mới(file mp3, thông tin metadata)
- Sửa bài hát
- Xóa bài hát



Hình 4.5: Hình ảnh giao diện trang quản lý bài hát

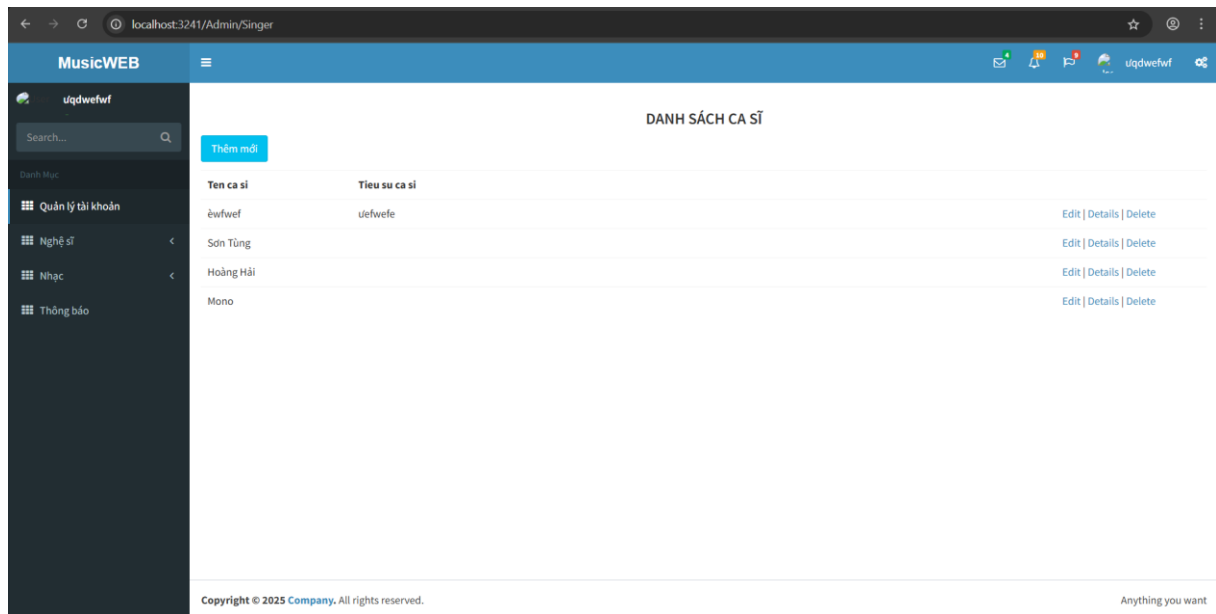
4.2.6. Giao diện quản lý ca sĩ

Chức năng:

- Cho phép quản trị viên quản lý danh sách ca sĩ.

Hành động người dùng:

- Thêm ca sĩ mới
- Sửa ca sĩ
- Xóa ca sĩ



Hình 4.6: Hình ảnh giao diện trang quản lý ca sĩ

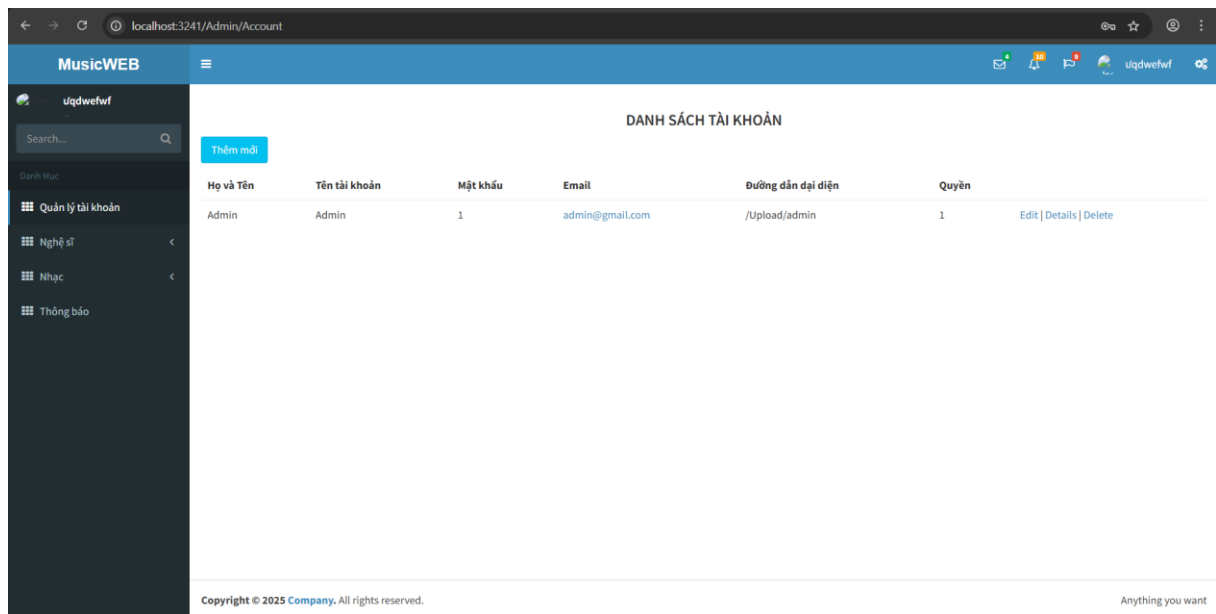
4.2.7. Giao diện trang quản lý tài khoản

Chức năng:

Cung cấp giao diện cho quản trị viên đăng nhập vào hệ thống để truy cập và thực hiện các chức năng quản lý tài khoản.

Hành động người dùng:

- Xem danh sách người dùng
- Thêm người dùng mới
- Chỉnh sửa thông tin người dùng
- Xóa hoặc khóa tài khoản
- Phân quyền người dùng (quản trị viên, người dùng thường, v.v.)



Hình 4.7: Hình ảnh giao diện trang quản lý tài khoản

4.3. Các bước chạy chương trình

- Bước 1: Mở Visual Studio và load dự án website nghe nhạc.
- Bước 2: Kiểm tra và đảm bảo SQL Server đang chạy, đồng thời cấu hình kết nối cơ sở dữ liệu trong file appsettings.json hoặc nơi cấu hình phù hợp.
- Bước 3: Thực hiện build dự án để kiểm tra không có lỗi (Build Solution).
- Bước 4: Chạy chương trình bằng cách nhấn Ctrl + F5 hoặc chọn Start Without Debugging.
- Bước 5: Trình duyệt web tự động mở ra trang chủ của website, người dùng có thể bắt đầu thao tác nghe nhạc hoặc đăng nhập.
- Bước 6: Đăng nhập với tài khoản admin để truy cập vào trang quản trị, thực hiện các chức năng quản lý như thêm nhạc, quản lý nhạc, phân quyền người dùng.
- Bước 7: Thực hiện các thao tác trên giao diện và kiểm tra kết quả hoạt động, đảm bảo dữ liệu được lưu trữ và cập nhật đúng trong cơ sở dữ liệu.

4.4. Kết quả chạy thử

- Giao diện hiển thị đúng, dễ sử dụng, phù hợp trên trình duyệt phổ biến.
- Chức năng admin hoạt động ổn định:
 - Nghe nhạc trực tuyến

- Tìm kiếm bài hát
 - Đăng ký/ Đăng nhập tài khoản
 - Giao diện admin quản lý dữ liệu
- Tất cả dữ liệu được lưu và truy xuất thành công từ cơ sở dữ liệu.
 - Giao diện thân thiện, hiển thị đúng các nội dung và trạng thái theo từng thao tác.

CHƯƠNG 5: KẾT LUẬN

5.1. Kết quả đạt được của đề tài

- Đã Hoàn Thiện Và Triển Khai Thành Công Website Nghe Nhạc Trực Tuyến.
- Giao Diện Website Được Thiết Kế Hiện Đại, Trực Quan, Dễ Sử Dụng Và Thân Thiện Với Người Dùng.
- Hệ Thống Cho Phép Người Dùng Đăng Ký, Đăng Nhập Tài Khoản Và Nghe Nhạc Miễn Phí Với Các Tính Năng Như Tìm Kiếm, Phân Loại Bài Hát Theo Thể Loại, Nghệ Sĩ, Album,...
- Website Hỗ Trợ Phát Nhạc Online Ổn Định, Có Thể Tạo Danh Sách Phát, Lưu Bài Hát Yêu Thích, Góp Phần Nâng Cao Trải Nghiệm Người Dùng.

5.2. Hạn chế của đề tài

- Một số tính năng nâng cao như gợi ý thông minh, đồng bộ hóa tài khoản đa thiết bị, chưa được triển khai đầy đủ.
- Tốc độ tải trang và phát nhạc đôi khi chưa được tối ưu với mạng yếu.
- Chưa tích hợp tính năng quản lý bản quyền âm nhạc, nội dung vẫn còn phụ thuộc vào dữ liệu thử nghiệm.
- Hệ thống bảo mật vẫn còn đơn giản, chưa áp dụng các cơ chế mã hóa nâng cao cho dữ liệu người dùng.

5.3. Kết luận

Website nghe nhạc đã góp phần tạo nên một nền tảng giải trí tiện ích, giúp người dùng dễ dàng tiếp cận và thưởng thức âm nhạc mọi lúc, mọi nơi.

Đề tài cũng hỗ trợ việc quản lý kho nhạc, người dùng, lượt phát,... một cách dễ dàng, góp phần hướng tới việc xây dựng một hệ thống âm nhạc số chuyên nghiệp hơn trong tương lai.

Trong quá trình thực hiện đề tài, do hạn chế về thời gian và kinh nghiệm thực tế, chắc chắn vẫn còn tồn tại những thiếu sót. Em rất mong nhận được sự góp ý từ quý thầy cô để hoàn thiện sản phẩm tốt hơn ở các phiên bản sau.

TÀI LIỆU THAM KHẢO

1. Microsoft Docs. *ASP.NET Core MVC Overview*.
<https://learn.microsoft.com/en-us/aspnet/core/mvc>
2. Microsoft Docs. *Entity Framework Core Documentation*.
<https://learn.microsoft.com/en-us/ef/core>
3. W3Schools. *Bootstrap 4 Tutorial*. <https://www.w3schools.com/bootstrap4>
4. Microsoft Docs. *SQL Server Documentation*.
<https://learn.microsoft.com/en-us/sql/sql-server>
5. Barryvdh. *Laravel-Dompdf Package Documentation*.
<https://github.com/barryvdh/laravel-dompdf>
6. Stack Overflow. *Cách tối ưu truy vấn SQL trong ASP.NET Core*.
<https://stackoverflow.com>