**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

*…………, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*……….., ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành và sự tri ân sâu sắc đến Quý Thầy, Cô khoa Công nghệ Phần mềm, Trường Đại học Công nghệ Thông tin. Những kiến thức quý báu và nền tảng chuyên môn vững chắc mà các Thầy, Cô đã tận tình truyền đạt trong suốt những năm tháng ngồi trên ghế nhà trường chính là hành trang vô giá giúp em thực hiện được đồ án này.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc nhất đến TS. Đoàn Phước Miền. Thầy không chỉ là người trực tiếp hướng dẫn, định hướng đề tài mà còn luôn dành thời gian quan tâm, tận tình chỉ bảo và đưa ra những góp ý xác đáng giúp em tháo gỡ những khó khăn trong quá trình nghiên cứu và hoàn thiện sản phẩm.

Em cũng xin gửi lời cảm ơn đến gia đình, bạn bè – những người đã luôn bên cạnh động viên, ủng hộ tinh thần và tạo mọi điều kiện thuận lợi nhất để em có thể toàn tâm toàn ý hoàn thành đồ án tốt nghiệp này.

Mặc dù đã nỗ lực hết mình với thái độ làm việc nghiêm túc, song do giới hạn về mặt thời gian cũng như kinh nghiệm thực tiễn của bản thân còn hạn chế, nên đồ án khó tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm cùng những ý kiến đóng góp, chỉ bảo quý báu của Quý Thầy, Cô để đồ án được hoàn thiện hơn cũng như để em có thêm bài học kinh nghiệm cho công việc sau này.

Em xin chân thành cảm ơn!

|  |  |
| --- | --- |
|  | **Sinh viên thực hiện**  ***(Ký và ghi rõ họ tên)***  **Nguyễn Hữu Nghĩa** |

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 1](#_Toc219064579)

[1.1. Tên đề tài 1](#_Toc219064580)

[1.2. Lý do chọn đề tài 1](#_Toc219064581)

[1.3. Mục tiêu nghiên cứu 1](#_Toc219064582)

[1.3.1. Mục tiêu chung 1](#_Toc219064583)

[1.3.2. Mục tiêu cụ thể 1](#_Toc219064584)

[1.4. Phương pháp nghiên cứu 2](#_Toc219064585)

[1.4.1. Phương pháp nghiên cứu lý thuyết 2](#_Toc219064586)

[1.4.2. Phương pháp nghiên cứu thực nghiệm 2](#_Toc219064587)

[1.5. Đối tượng và phạm vi nghiên cứu 2](#_Toc219064588)

[1.5.1. Đối tượng nghiên cứu 2](#_Toc219064589)

[1.5.2. Phạm vi nghiên cứu 3](#_Toc219064590)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 4](#_Toc219064591)

[2.1. Tổng quan về Internet và World Wide Web 4](#_Toc219064592)

[2.1.1. Internet (Mạng lưới toàn cầu) 4](#_Toc219064593)

[2.1.2. World Wide Web (WWW - Mạng lưới thông tin) 4](#_Toc219064594)

[2.1.3. Kiến trúc hoạt động của một Website 4](#_Toc219064595)

[2.1.4. Quy trình xử lý một truy cập Website (Request - Response Cycle) 5](#_Toc219064596)

[2.2. Ngôn ngữ đánh dấu siêu văn bản (HTML5) 6](#_Toc219064597)

[2.2.1. Tổng quan 6](#_Toc219064598)

[2.2.2. Cấu trúc và thẻ cơ bản 6](#_Toc219064599)

[2.2.3. Đánh giá ưu nhược điểm 6](#_Toc219064600)

[2.3. Ngôn ngữ định dạng (CSS3) 7](#_Toc219064601)

[2.3.1. Tổng quan 7](#_Toc219064602)

[2.3.2. Cấu trúc và kỹ thuật áp dụng 7](#_Toc219064603)

[2.3.3. Đánh giá ưu nhược điểm 7](#_Toc219064604)

[2.4. Ngôn ngữ lập trình phía Client (JavaScript) 8](#_Toc219064605)

[2.4.1. Tổng quan 8](#_Toc219064606)

[2.4.2. Cấu trúc và xử lý logic 8](#_Toc219064607)

[2.4.3. Đánh giá ưu nhược điểm 8](#_Toc219064608)

[2.5. Ngôn ngữ truy vấn dữ liệu (SQL) 9](#_Toc219064609)

[2.5.1. Tổng quan 9](#_Toc219064610)

[2.5.2. Các câu lệnh cơ bản áp dụng 9](#_Toc219064611)

[2.5.3. Đánh giá ưu nhược điểm (SQL Server) 9](#_Toc219064612)

[2.6. Công nghệ nền tảng Server (ASP.NET WebForms) 9](#_Toc219064613)

[2.6.1. Tổng quan 9](#_Toc219064614)

[2.6.2. Cơ chế hoạt động trong dự án 10](#_Toc219064615)

[2.6.3. Đánh giá ưu nhược điểm 10](#_Toc219064616)

[CHƯƠNG 3: THỰC HIỆN HÓA NGHIÊN CỨU 11](#_Toc219064617)

[3.1. Phân tích thiết kế hệ thống 11](#_Toc219064618)

[3.1.1. Mô tả bài toán 11](#_Toc219064619)

[3.1.2. Xác định yêu cầu 11](#_Toc219064620)

[3.1.3. Mô tả chức năng 12](#_Toc219064621)

[3.1.4. Đề xuất mô hình 13](#_Toc219064622)

[3.1.5. Sơ đồ luồng xử lý phát nhạc (AJAX Flow) 14](#_Toc219064623)

[3.2. Thiết kế cơ sở dữ liệu 14](#_Toc219064624)

[3.2.1. Xác định các bảng dữ liệu 14](#_Toc219064625)

[3.2.2. Thiết kế quan hệ giữa các bảng 15](#_Toc219064626)

[CHƯƠNG 4: KẾT QUẢ VÀ ĐÁNH GIÁ 17](#_Toc219064627)

[4.1. Một số hình ảnh chương trình (Mô tả kết quả từng chức năng) 17](#_Toc219064628)

[4.1.1. Giao diện trang chủ (Home Page) 17](#_Toc219064629)

[4.1.2. Chức năng Đăng ký / Đăng nhập 18](#_Toc219064630)

[4.1.3. Form Đăng nhập: 18](#_Toc219064631)

[4.1.4. Form Đăng ký: 19](#_Toc219064632)

[4.1.5. Trạng thái giao diện sau khi đăng nhập: 20](#_Toc219064633)

[4.1.6. Chức năng Tìm kiếm bài hát (Live Search) 21](#_Toc219064634)

[4.1.7. Chức năng Nghe nhạc và Điều khiển (Player Control) 22](#_Toc219064635)

[4.1.8. Chức năng Yêu thích (Favorites) 23](#_Toc219064636)

[4.1.9. Trang Quản trị hệ thống (Admin Dashboard) 24](#_Toc219064637)

[4.1.10. Quản lý Bài hát (Admin) 25](#_Toc219064638)

[4.1.11. Quản lý Ca sĩ (Admin) 26](#_Toc219064639)

[4.2. Cấu trúc dự án (Project Structure) 27](#_Toc219064640)

[4.3. Đánh giá hiệu năng hệ thống 28](#_Toc219064641)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 29](#_Toc219064642)

[5.1. Đánh giá tổng quan 29](#_Toc219064643)

[5.2. Kết quả đạt được 29](#_Toc219064644)

[5.3. So sánh công nghệ ASP.NET WebForms và MVC 30](#_Toc219064645)

[5.4. Hướng phát triển 30](#_Toc219064646)

[5.5. Kết luận chung 31](#_Toc219064647)

[DANH MỤC TÀI LIỆU THAM KHẢO 32](#_Toc219064648)

**DANH MỤC HÌNH ẢNH**

[Hình 1‑1. Quy trình phát triển website HN Melody 2](#_Toc219064649)

[Hình 2‑1. Sườn cấu trúc HTML trang HN Melody 6](#_Toc219064650)

[Hình 2‑2. Cấu trúc CSS hiệu ứng đĩa nhạc xoay trong trang HN Melody 7](#_Toc219064651)

[Hình 2‑3. Đoạn mã JavaScript khởi tạo đối tượng app và xử lý sự kiện phát/dừng nhạc 8](#_Toc219064652)

[Hình 3‑1. Sơ đồ Use Case tổng quát của hệ thống HN Melody 12](#_Toc219064653)

[Hình 3‑2. Sơ đồ luồng dữ liệu DFD Level 0 13](#_Toc219064654)

[Hình 3‑3. Sơ đồ quan hệ thực thể (ERD) 13](#_Toc219064655)

[Hình 3‑4. Kiến trúc tổng thể hệ thống 14](#_Toc219064656)

[Hình 4‑1. Giao diện trang chủ của website HN Melody 17](#_Toc219064657)

[Hình 4‑2. Giao diện Form Đăng nhập 18](#_Toc219064658)

[Hình 4‑3. Giao diện Form Đăng nhập và Đăng ký 19](#_Toc219064659)

[Hình 4‑4. Giao diện trang chủ sau khi đăng nhập thành công 20](#_Toc219064660)

[Hình 4‑5. Kết quả tìm kiếm bài hát theo từ khóa 21](#_Toc219064661)

[Hình 4‑6. Các nút điều khiển trình phát nhạc 22](#_Toc219064662)

[Hình 4‑7. Danh sách bài hát yêu thích 23](#_Toc219064663)

[Hình 4‑8. Giao diện tổng quan trang Admin 24](#_Toc219064664)

[Hình 4‑9. Giao diện Quản lý Bài hát 25](#_Toc219064665)

[Hình 4‑10. Giao diện Quản lý Ca sĩ 26](#_Toc219064666)

[Hình 4‑11. Cấu trúc thư mục dự án trong Visual Studio 27](#_Toc219064667)

**DANH MỤC BẢNG BIỂU**

[Bảng 3‑1. Cấu trúc dữ liệu bảng Users 14](#_Toc219064675)

[Bảng 3‑2. Cấu trúc dữ liệu bảng Artists 15](#_Toc219064676)

[Bảng 3‑3. Cấu trúc dữ liệu bảng Songs 15](#_Toc219064677)

[Bảng 3‑4. Cấu trúc dữ liệu bảng Favorites 15](#_Toc219064678)

[Bảng 4‑1. Bảng đánh giá hiệu năng hệ thống 28](#_Toc219064679)

[Bảng 5‑1. So sánh ASP.NET WebForms và ASP.NET MVC 30](#_Toc219064680)

# TỔNG QUAN ĐỀ TÀI

## Tên đề tài

"Nghiên cứu, xây dựng và phát triển hệ thống website nghe nhạc trực tuyến HN Melody sử dụng công nghệ ASP.NET WebForms."

## Lý do chọn đề tài

Trong kỷ nguyên số hóa hiện nay, nhu cầu giải trí, đặc biệt là nghe nhạc trực tuyến, đã trở thành một phần không thể thiếu trong đời sống tinh thần của con người. Thay vì phải tải nhạc về máy lưu trữ như trước đây, người dùng ngày càng ưa chuộng các nền tảng Streaming (phát trực tuyến) vì sự tiện lợi và kho tàng dữ liệu khổng lồ. Những lý do chính khiến tôi chọn đề tài này bao gồm:

* Nhu cầu thực tế: Các nền tảng như Spotify, ZingMP3 hay Apple Music đang thống trị thị trường. Việc xây dựng một hệ thống nghe nhạc riêng giúp tôi hiểu rõ cơ chế hoạt động của việc xử lý Media, Streaming và quản lý dữ liệu đa phương tiện trên môi trường Web.
* Củng cố kiến thức nền tảng: ASP.NET WebForms là một công nghệ nền tảng quan trọng trong hệ sinh thái .NET. Việc áp dụng công nghệ này kết hợp với C# và SQL Server giúp tôi nắm vững kiến thức về lập trình phía Server (Back-end) và cách giao tiếp với Client thông qua WebMethod (AJAX).
* Rèn luyện kỹ năng Front-end: Dự án yêu cầu xử lý logic nghe nhạc phức tạp (phát, tạm dừng, tua, lặp lại, ngẫu nhiên) bằng Javascript thuần (Vanilla JS) mà không phụ thuộc vào thư viện bên thứ 3, giúp nâng cao tư duy thuật toán và kỹ năng xử lý DOM.
* Tính ứng dụng: Sản phẩm sau khi hoàn thiện có thể triển khai thực tế, phục vụ nhu cầu nghe nhạc cá nhân hoặc phát triển thành một trang chia sẻ nhạc cộng đồng quy mô nhỏ.

## Mục tiêu nghiên cứu

### Mục tiêu chung

Xây dựng thành công website nghe nhạc trực tuyến HN Melody với giao diện hiện đại (Dark Mode), tốc độ tải trang nhanh, ổn định và đầy đủ các chức năng quản lý, nghe nhạc cơ bản.

### Mục tiêu cụ thể

* Phân tích hệ thống: Thiết kế cơ sở dữ liệu quan hệ (ERD) chuẩn hóa để lưu trữ bài hát, ca sĩ, người dùng và playlist yêu thích.
* Xây dựng giao diện: Thiết kế giao diện người dùng (UI) hiện đại, tối ưu trải nghiệm (UX) với tông màu tối, tương thích tốt trên các trình duyệt phổ biến.
* Phát triển chức năng nghe nhạc: Xây dựng trình phát nhạc (Player) với các tính năng: Play/Pause, Next/Prev, Random (ngẫu nhiên), Repeat (lặp lại), Seek (tua thời gian).
* Xây dựng trang quản trị (Admin): Cung cấp công cụ cho Admin quản lý kho nhạc (Thêm/Sửa/Xóa bài hát, ca sĩ) một cách trực quan, bảo mật.
* Tối ưu hiệu năng: Sử dụng kỹ thuật WebMethod (AJAX) để tải dữ liệu bài hát và thao tác cơ sở dữ liệu mà không cần tải lại toàn bộ trang (No-reload).

## Phương pháp nghiên cứu

### Phương pháp nghiên cứu lý thuyết

* Nghiên cứu công nghệ: Tìm hiểu sâu về vòng đời của trang ASP.NET WebForms, cơ chế hoạt động của System.Web.Services.WebMethod, và các thẻ HTML5 Audio/Video.
* Tham khảo mô hình: Phân tích giao diện và luồng hoạt động (User Flow) của các website nghe nhạc lớn (như Soundcloud, ZingMP3) để học hỏi cách bố trí bố cục và tối ưu trải nghiệm người dùng.

### Phương pháp nghiên cứu thực nghiệm

* Phát triển phần mềm: Áp dụng quy trình phát triển

Hình 1‑1. Quy trình phát triển website HN Melody

* Clean Code: Viết mã nguồn Javascript theo hướng đối tượng (Object-based) để code gọn gàng, dễ bảo trì và nâng cấp.
* Kiểm thử hộp đen: Thực hiện kiểm thử các chức năng người dùng và quản trị viên trong nhiều kịch bản để phát hiện và sửa lỗi logic.

## Đối tượng và phạm vi nghiên cứu

### Đối tượng nghiên cứu

Đối tượng chính: Hệ thống website nghe nhạc HN Melody.

Người dùng hệ thống:

* Khách (Guest): Có thể nghe nhạc, xem danh sách bài hát, tìm kiếm.
* Người dùng (User): Đăng nhập, hệ thống ghi nhớ trạng thái nghe nhạc.
* Quản trị viên (Admin): Có quyền truy cập trang quản trị (Dashboard) để thao tác dữ liệu hệ thống.

### Phạm vi nghiên cứu

Về mặt chức năng:

* Hệ thống Đăng ký, Đăng nhập và phân quyền Admin/User.
* Trình phát nhạc đầy đủ tính năng (xử lý sự kiện Audio DOM).
* Chức năng tìm kiếm bài hát theo tên (Live Search).
* Trang quản trị (Admin Dashboard) quản lý Bài hát và Nghệ sĩ.

Về mặt công nghệ:

* Backend: ASP.NET WebForms (.NET Framework), C#.
* Database: Microsoft SQL Server.
* Frontend: HTML5, CSS3 (Custom Style - Không dùng Bootstrap), Javascript (Vanilla JS - Không dùng jQuery/React/Vue).
* Giao thức: AJAX (thông qua Fetch API và WebMethod).

# NGHIÊN CỨU LÝ THUYẾT

## Tổng quan về Internet và World Wide Web

Trước khi đi sâu vào các công nghệ lập trình cụ thể, cần nắm vững nền tảng hạ tầng và môi trường vận hành của hệ thống, đó là Internet và World Wide Web.

### Internet (Mạng lưới toàn cầu)

* Định nghĩa: Internet là một hệ thống thông tin toàn cầu bao gồm các mạng máy tính được liên kết với nhau, sử dụng bộ giao thức chuẩn TCP/IP (Transmission Control Protocol/Internet Protocol) để phục vụ hàng tỷ người dùng trên toàn thế giới.
* Bản chất: Internet là phần hạ tầng vật lý (phần cứng), bao gồm các cáp quang biển, vệ tinh, router (bộ định tuyến), server (máy chủ) và các thiết bị đầu cuối kết nối với nhau.
* Giao thức TCP/IP: Là "ngôn ngữ" chung giúp các thiết bị khác nhau (PC, Laptop, Smartphone) có thể hiểu và trao đổi dữ liệu.
* IP (Internet Protocol): Định danh địa chỉ của máy tính trên mạng (Ví dụ: 192.168.1.1).
* TCP (Transmission Control Protocol): Đảm bảo dữ liệu được chia nhỏ thành các gói tin (packets), truyền đi và lắp ghép lại chính xác tại đích đến.

### World Wide Web (WWW - Mạng lưới thông tin)

* Định nghĩa: World Wide Web (gọi tắt là Web) là một không gian thông tin toàn cầu, nơi các tài liệu và tài nguyên web được xác định bằng các URL (Uniform Resource Locators), được liên kết với nhau bởi các siêu liên kết (Hyperlinks) và truy cập thông qua Internet.
* Bản chất: Web là một dịch vụ chạy trên nền tảng hạ tầng của Internet. Nếu ví Internet là hệ thống đường xá, thì Web chính là các cửa hàng, dịch vụ vận chuyển hàng hóa trên con đường đó.
* Cha đẻ: Được phát minh bởi Tim Berners-Lee vào năm 1989 tại CERN.

### Kiến trúc hoạt động của một Website

Website hoạt động dựa trên mô hình Client-Server (Khách - Chủ), bao gồm các thành phần cốt lõi sau:

* A. Web Client (Phía khách)
* Là các thiết bị của người dùng (User) có cài đặt trình duyệt web (Web Browser) như Chrome, Edge, Firefox.
* Vai trò: Gửi yêu cầu (Request) đến máy chủ và hiển thị kết quả (Response) cho người dùng dưới dạng giao diện đồ họa (GUI).
* Rendering Engine: Là bộ phận cốt lõi của trình duyệt (ví dụ V8 Engine của Chrome) giúp dịch mã HTML/CSS/JS thành hình ảnh, văn bản mà người dùng nhìn thấy.
* B. Web Server (Phía máy chủ)
* Là các máy tính cấu hình cao được kết nối Internet 24/7, nơi lưu trữ mã nguồn website (Source Code) và Cơ sở dữ liệu (Database).
* Phần mềm Web Server: Trong dự án HN Melody sử dụng công nghệ .NET, nên Web Server thường dùng là IIS (Internet Information Services) của Microsoft. Nó chịu trách nhiệm lắng nghe các yêu cầu từ Client và trả về dữ liệu tương ứng.
* C. Hệ thống tên miền (DNS - Domain Name System)
* Máy tính chỉ hiểu các dãy số địa chỉ IP (ví dụ: 142.250.1.1), nhưng con người lại dễ nhớ tên miền (ví dụ: google.com).
* Chức năng: DNS đóng vai trò như một cuốn "danh bạ điện thoại", tự động dịch tên miền mà người dùng nhập vào thành địa chỉ IP của máy chủ chứa website đó.
* D. Giao thức HTTP/HTTPS
* HTTP (Hypertext Transfer Protocol): Giao thức truyền tải siêu văn bản. Đây là bộ quy tắc quy định cách Client và Server "nói chuyện" với nhau.
* HTTPS (Secure): Phiên bản bảo mật của HTTP, sử dụng chứng chỉ SSL/TLS để mã hóa dữ liệu truyền tải, ngăn chặn hacker đánh cắp thông tin (đặc biệt quan trọng với chức năng Đăng nhập/Đăng ký của dự án).

### Quy trình xử lý một truy cập Website (Request - Response Cycle)

Khi người dùng truy cập vào trang chủ HN Melody, quy trình sau sẽ diễn ra trong tíc tắc:

1. Phân giải tên miền (DNS Lookup): Trình duyệt hỏi DNS Server: "IP của trang web https://www.google.com/search?q=hnmelody.com là gì?". DNS trả về địa chỉ IP của máy chủ.
2. Thiết lập kết nối (TCP Handshake): Trình duyệt và Máy chủ thực hiện bắt tay 3 bước để tạo kết nối tin cậy.
3. Gửi yêu cầu (HTTP Request): Trình duyệt gửi một gói tin yêu cầu: "Cho tôi xem trang chủ (Default.aspx)".
4. Xử lý tại Server (Server Processing):

* IIS nhận yêu cầu.
* ASP.NET biên dịch mã C# (Code Behind).
* Nếu cần danh sách bài hát, Server sẽ truy vấn SQL Server để lấy dữ liệu.

1. Trả về kết quả (HTTP Response): Server đóng gói mã HTML, CSS, JS và dữ liệu bài hát, gửi ngược lại cho trình duyệt.
2. Hiển thị (Rendering): Trình duyệt nhận mã nguồn, xây dựng cây DOM (Document Object Model) và vẽ lên màn hình giao diện máy nghe nhạc cho người dùng.

## Ngôn ngữ đánh dấu siêu văn bản (HTML5)

### Tổng quan

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu tiêu chuẩn để tạo ra các trang web. Phiên bản HTML5 là phiên bản mới nhất, hỗ trợ mạnh mẽ các thẻ đa phương tiện (Multimedia) mà không cần đến các plugin bên thứ ba như Flash trước đây. Trong dự án HN Melody, HTML5 đóng vai trò là "bộ khung xương" định hình cấu trúc giao diện.

### Cấu trúc và thẻ cơ bản

Một trang HTML được cấu thành từ các phần tử (Elements), được bao bọc bởi các thẻ (Tags). Cấu trúc mã cơ bản trong dự án:



Hình 2‑1. Sườn cấu trúc HTML trang HN Melody

Trong dự án này, thẻ <audio> là thành phần quan trọng nhất. Nó cung cấp các API để trình duyệt có thể đọc, giải mã và phát các file âm thanh (.mp3).

### Đánh giá ưu nhược điểm

Ưu điểm:

* Là chuẩn quốc tế, hiển thị tốt trên mọi trình duyệt (Chrome, Edge, Firefox).
* Cấu trúc ngữ nghĩa (Semantic) tốt, hỗ trợ SEO.
* HTML5 hỗ trợ sẵn Media (Audio/Video), phù hợp cho web nghe nhạc.

Nhược điểm:

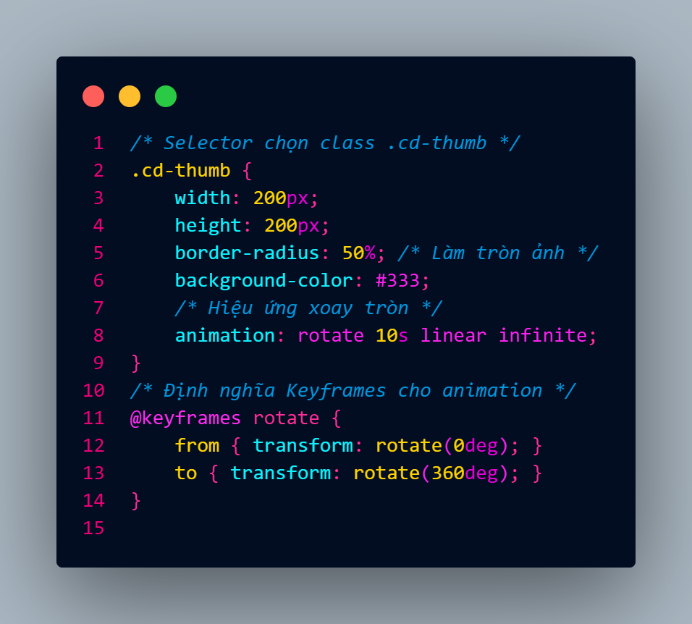
* HTML chỉ là ngôn ngữ tĩnh, không thể xử lý logic (cộng trừ nhân chia, truy vấn dữ liệu). Cần phải kết hợp với ngôn ngữ lập trình khác.

## Ngôn ngữ định dạng (CSS3)

### Tổng quan

CSS (Cascading Style Sheets) là ngôn ngữ dùng để mô tả cách hiển thị của các phần tử HTML. CSS3 mang lại khả năng tùy biến giao diện mạnh mẽ với các hiệu ứng chuyển động (Animation), bố cục linh hoạt (Flexbox/Grid). Trong dự án, CSS3 chịu trách nhiệm "trang điểm" cho website, tạo nên giao diện Dark Mode hiện đại.

### Cấu trúc và kỹ thuật áp dụng

* Cấu trúc cơ bản của một luật CSS bao gồm: Selector (Bộ chọn) + Property (Thuộc tính) + Value (Giá trị).
* Ví dụ đoạn code xử lý hiệu ứng đĩa than xoay trong dự án:

Hình 2‑2. Cấu trúc CSS hiệu ứng đĩa nhạc xoay trong trang HN Melody

* Các kỹ thuật nâng cao được sử dụng:
* Flexbox: Để căn giữa các nút điều khiển (Play, Next, Prev) và danh sách bài hát.
* CSS Variables (Biến): Sử dụng var(--primary-color) để quản lý màu sắc đồng bộ toàn trang.
* Responsive: Sử dụng đơn vị rem và % để giao diện co giãn tốt trên màn hình máy tính.

### Đánh giá ưu nhược điểm

Ưu điểm:

* Tách biệt nội dung (HTML) và hình thức (CSS), giúp mã nguồn gọn gàng.
* Tiết kiệm băng thông do file CSS được trình duyệt cache lại.
* Khả năng tạo hiệu ứng mượt mà (60fps) tốt hơn dùng Javascript.

Nhược điểm:

* Khác biệt hiển thị giữa các trình duyệt cũ (IE) và mới (nhưng hiện nay không còn là vấn đề lớn).

## Ngôn ngữ lập trình phía Client (JavaScript)

### Tổng quan

JavaScript (JS) là ngôn ngữ lập trình kịch bản chạy trên trình duyệt người dùng. Đây là "bộ não" của giao diện, xử lý các sự kiện click chuột, phát nhạc, tính toán thời gian bài hát. Dự án sử dụng Vanilla JS (JS thuần) thay vì các thư viện nặng nề để tối ưu tốc độ.

### Cấu trúc và xử lý logic

* JavaScript trong dự án được tổ chức theo mô hình Đối tượng (Object Literal) để gom nhóm các chức năng, giúp code sạch (Clean Code).
* Cấu trúc code cơ bản:

Hình 2‑3. Đoạn mã JavaScript khởi tạo đối tượng app và xử lý sự kiện phát/dừng nhạc

* Kỹ thuật AJAX (Fetch API) cũng được sử dụng để gọi dữ liệu từ Server về mà không cần tải lại trang (No-reload), giúp trải nghiệm nghe nhạc không bị gián đoạn.

### Đánh giá ưu nhược điểm

Ưu điểm:

* Tốc độ xử lý cực nhanh do chạy trực tiếp trên máy người dùng.
* Giảm tải cho Server.
* Tạo ra trải nghiệm người dùng mượt mà, tương tác tức thời.

Nhược điểm:

* Bảo mật kém hơn xử lý phía Server (code JS hiển thị công khai).
* Nếu người dùng tắt JavaScript trên trình duyệt, website sẽ mất chức năng.

## Ngôn ngữ truy vấn dữ liệu (SQL)

### Tổng quan

SQL (Structured Query Language) là ngôn ngữ chuẩn hóa để thao tác với Cơ sở dữ liệu quan hệ. Dự án sử dụng hệ quản trị Microsoft SQL Server để lưu trữ bền vững thông tin bài hát, tài khoản và nghệ sĩ.

### Các câu lệnh cơ bản áp dụng

Dự án sử dụng đầy đủ 4 thao tác CRUD (Create, Read, Update, Delete):

* SELECT (Lấy dữ liệu): Dùng để hiển thị danh sách bài hát ra trang chủ.

SELECT SongID, Title, Image, Url FROM Songs WHERE ViewCount > 1000

* INSERT (Thêm mới): Dùng trong trang Admin khi thêm bài hát.

INSERT INTO Songs (Title, ArtistID, Url) VALUES (N'Em của ngày hôm qua', 1, '/music/song1.mp3')

* UPDATE (Cập nhật): Dùng để sửa thông tin bài hát hoặc tăng lượt xem.

UPDATE Songs SET ViewCount = ViewCount + 1 WHERE SongID = @id

* DELETE (Xóa): Xóa bài hát khỏi hệ thống.

DELETE FROM Songs WHERE SongID = @id

### Đánh giá ưu nhược điểm (SQL Server)

Ưu điểm:

* Tính bảo mật cao, quản lý quyền truy cập chặt chẽ.
* Đảm bảo tính toàn vẹn dữ liệu (Data Integrity) thông qua các ràng buộc khóa chính, khóa ngoại.
* Khả năng xử lý lượng dữ liệu lớn, truy vấn phức tạp nhanh chóng.

Nhược điểm:

* Chi phí bản quyền cao (với bản Enterprise, tuy nhiên bản Express miễn phí đủ dùng cho học tập).
* Yêu cầu tài nguyên máy chủ (RAM/CPU) cao hơn so với lưu trữ file thường.

## Công nghệ nền tảng Server (ASP.NET WebForms)

### Tổng quan

ASP.NET WebForms là một mô hình phát triển web thuộc hệ sinh thái .NET Framework của Microsoft. Nó cung cấp mô hình lập trình hướng sự kiện (Event-driven) tương tự như lập trình ứng dụng Windows, giúp việc xây dựng các ứng dụng quản lý dữ liệu trở nên nhanh chóng.

### Cơ chế hoạt động trong dự án

* Khác với mô hình MVC hiện đại, WebForms sử dụng cặp file .aspx (Giao diện) và .aspx.cs (Code Behind - Logic C#).
* Điểm đặc biệt trong dự án HN Melody là việc ứng dụng WebMethod (Page Methods):
* Thay vì để WebForms tự động Postback (tải lại toàn bộ trang) mỗi khi bấm nút, dự án sử dụng các hàm public static với thuộc tính [WebMethod].
* JavaScript sẽ gọi các hàm này để lấy dữ liệu JSON.
* Lợi ích: Kết hợp được sức mạnh quản lý của C# phía Server và sự mượt mà của AJAX phía Client.

### Đánh giá ưu nhược điểm

Ưu điểm:

* Tốc độ phát triển nhanh nhờ bộ công cụ Visual Studio và các Server Controls có sẵn.
* Dễ dàng kết nối với SQL Server thông qua thư viện ADO.NET (System.Data.SqlClient).
* Cấu trúc Code-Behind tách biệt logic C# khỏi giao diện HTML.

Nhược điểm:

* Sinh ra mã HTML đôi khi dư thừa (ViewState) làm tăng dung lượng trang (Tuy nhiên dự án đã khắc phục bằng cách dùng WebMethod).
* Khó kiểm soát ID của các phần tử HTML hơn so với MVC.

# THỰC HIỆN HÓA NGHIÊN CỨU

## Phân tích thiết kế hệ thống

### Mô tả bài toán

* Website nghe nhạc trực tuyến HN Melody là hệ thống giải trí đa phương tiện kết nối người nghe với kho nhạc số. Hệ thống được xây dựng để giải quyết các vấn đề sau:
* Vấn đề của người nghe nhạc:
* Phải tải nhạc về máy gây tốn dung lượng lưu trữ.
* Các trang web hiện tại chứa quá nhiều quảng cáo hoặc giao diện rối rắm.
* Khó khăn trong việc tìm kiếm và lưu trữ các bài hát yêu thích cá nhân.
* Vấn đề của người quản trị (Admin):
* Cần một công cụ quản lý kho nhạc tập trung (thêm, sửa, xóa bài hát/nghệ sĩ).
* Thống kê được lượt nghe để nắm bắt xu hướng.
* Giải pháp đề xuất: Xây dựng website HN Melody với các tính năng trọng tâm:
* Trình phát nhạc (Player) hiện đại, hỗ trợ đầy đủ các tính năng điều khiển.
* Hệ thống tài khoản để lưu trữ bài hát yêu thích.
* Công cụ tìm kiếm bài hát theo thời gian thực (Live Search).
* Trang quản trị (Dashboard) trực quan dành cho Admin.

### Xác định yêu cầu

#### Yêu cầu chức năng

* Đối với Khách (Guest - Chưa đăng nhập):
* Nghe nhạc: Phát nhạc, tạm dừng, chuyển bài, tua nhạc.
* Xem danh sách: Xem danh sách bài hát mới, top bài hát.
* Tìm kiếm: Tìm kiếm bài hát theo tên hoặc tên ca sĩ.
* Đối với Người dùng (User - Đã đăng nhập):
* Đăng nhập/Đăng xuất: Truy cập hệ thống bằng tài khoản cá nhân.
* Lưu trạng thái: Hệ thống tự động ghi nhớ bài hát đang nghe dở và các cài đặt (lặp lại/ngẫu nhiên).
* Yêu thích: Thêm bài hát vào danh sách yêu thích (Favorites).
* Đối với Quản trị viên (Admin):
* Quản lý Bài hát: Thêm mới bài hát (nhập link nhạc, ảnh, tên), sửa thông tin, xóa bài hát.
* Quản lý Nghệ sĩ: Thêm, sửa, xóa thông tin ca sĩ.
* Thống kê: Xem danh sách bài hát, sắp xếp theo lượt xem để đánh giá độ phổ biến.

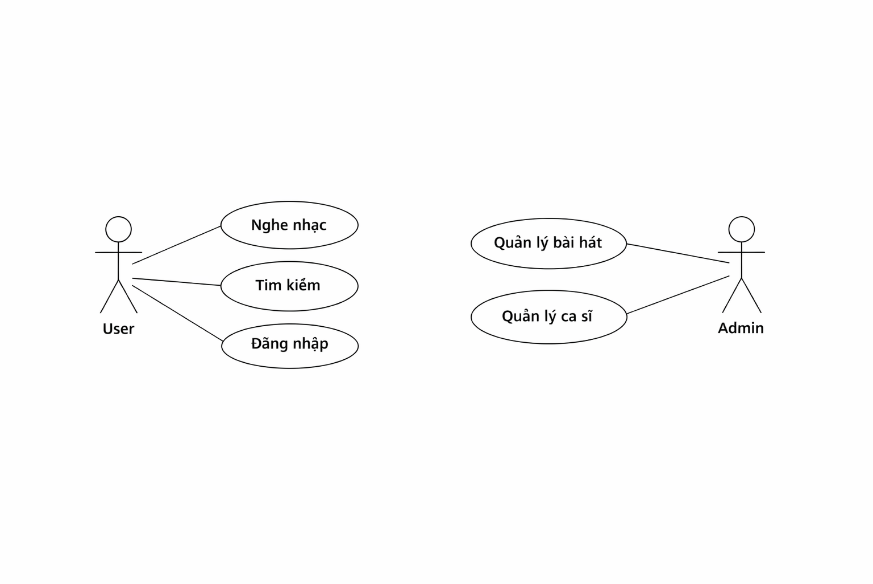
#### Yêu cầu phi chức năng

* Hiệu suất: Trình phát nhạc phải phản hồi tức thì (dưới 0.5s), không bị gián đoạn khi chuyển trang (nhờ AJAX).
* Giao diện: Thiết kế Dark Mode (chế độ tối) giúp giảm mỏi mắt, Responsive tốt trên cả máy tính và điện thoại.
* Bảo mật: Mật khẩu người dùng phải được mã hóa, trang Admin được bảo vệ bằng cơ chế phân quyền (Authentication/Authorization).
* Khả năng mở rộng: Cấu trúc Code theo hướng đối tượng (Object-based) dễ dàng thêm tính năng Playlist hoặc Comment sau này.

### Mô tả chức năng

#### Sơ đồ Use Case

Use Case tổng quát: Mô tả tương tác giữa Người dùng và Admin với hệ thống.

Hình 3‑1. Sơ đồ Use Case tổng quát của hệ thống HN Melody

#### Mô tả chi tiết các Use Case chính

UC1: Nghe nhạc (Play Music)

* Actor: Người dùng, Khách.
* Mô tả: Người dùng chọn và nghe một bài hát từ danh sách.
* Luồng chính:
* Người dùng truy cập trang chủ hoặc thanh tìm kiếm.
* Người dùng nhấn vào một bài hát hoặc nút Play.
* Hệ thống tải dữ liệu bài hát (MP3, Ảnh, Tên) qua AJAX.
* Trình phát nhạc hiển thị thông tin và bắt đầu phát âm thanh.
* Thanh tiến trình (Progress bar) chạy theo thời gian thực.

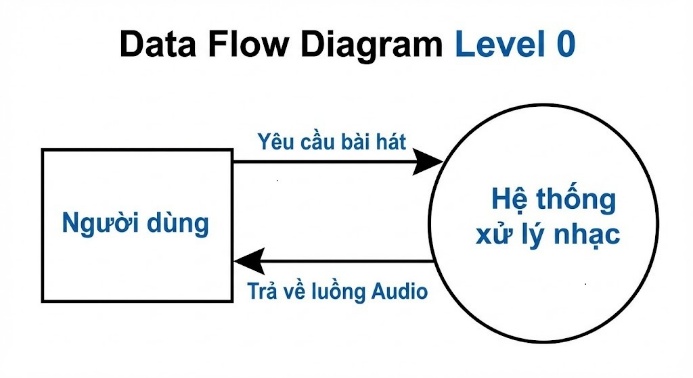
UC2: Quản lý Bài hát (Manage Songs)

* Actor: Admin.
* Điều kiện: Đã đăng nhập tài khoản Admin.
* Luồng chính:
* Admin truy cập trang Quản trị (Admin.aspx).
* Hệ thống hiển thị danh sách bài hát dưới dạng bảng.
* Admin chọn chức năng Thêm mới, Sửa hoặc Xóa.
* Nhập liệu (nếu thêm/sửa) và xác nhận.
* Hệ thống cập nhật vào CSDL SQL Server và thông báo kết quả.

### Đề xuất mô hình

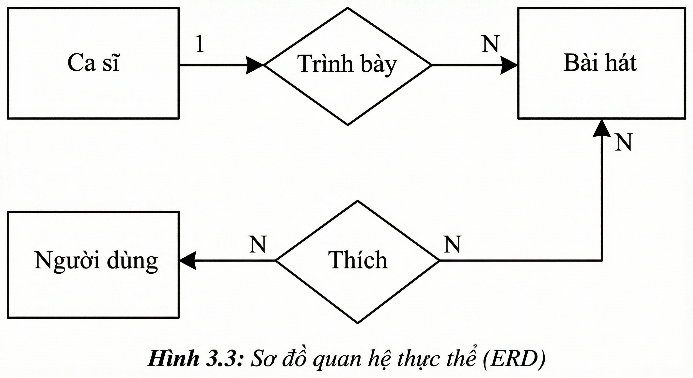
#### Sơ đồ luồng dữ liệu (DFD)

Mô tả cách dữ liệu bài hát di chuyển từ Server xuống Trình duyệt người dùng.

Hình 3‑2. Sơ đồ luồng dữ liệu DFD Level 0

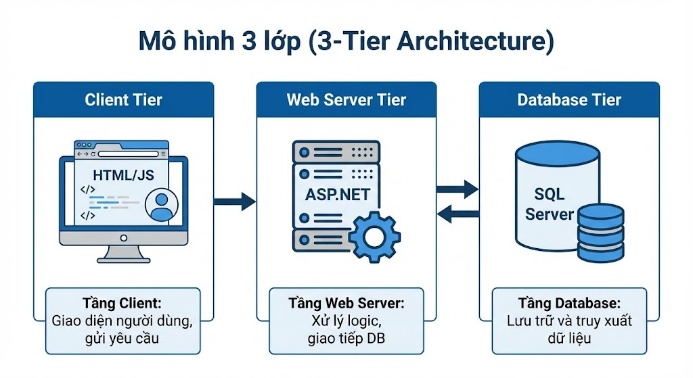
#### Sơ đồ quan hệ thực thể (ERD)

Mô tả mối quan hệ giữa Bài hát, Ca sĩ và Người dùng.

Hình 3‑3. Sơ đồ quan hệ thực thể (ERD)

#### Sơ đồ kiến trúc hệ thống

Mô tả mô hình 3 lớp: Client (HTML/JS) - Web Server (ASP.NET) - Database (SQL Server).

Hình 3‑4. Kiến trúc tổng thể hệ thống

### Sơ đồ luồng xử lý phát nhạc (AJAX Flow)

Khác với các ứng dụng ASP.NET WebForms truyền thống thường phải tải lại toàn bộ trang (Postback) cho mỗi thao tác xử lý sự kiện, hệ thống **HN Melody** áp dụng cơ chế xử lý bất đồng bộ (Asynchronous) để tối ưu hóa trải nghiệm nghe nhạc. Quy trình xử lý kỹ thuật khi người dùng chọn một bài hát được mô tả chi tiết như sau:

Quy trình xử lý chi tiết:

1. **Sự kiện (Event Trigger):** Người dùng nhấn nút "Play" trên giao diện danh sách bài hát.
2. **Xử lý tại Client (JavaScript):** Hệ thống bắt sự kiện onclick, chặn hành vi tải lại trang mặc định của trình duyệt (preventDefault).
3. **Gửi yêu cầu (Request):** JavaScript sử dụng Fetch API để gửi một yêu cầu HTTP POST bất đồng bộ đến WebMethod Default.aspx/GetSongs (hoặc phương thức xử lý tương ứng) kèm theo ID bài hát.
4. **Xử lý tại Server (C#):** WebMethod nhận yêu cầu, thực hiện kết nối tới SQL Server để truy vấn thông tin chi tiết của bài hát (đường dẫn file âm thanh, ảnh bìa, tên nghệ sĩ).
5. **Phản hồi (Response):** Server trả về dữ liệu kết quả dưới định dạng JSON nhẹ.
6. **Cập nhật giao diện (Render):** JavaScript nhận chuỗi JSON, phân tích cú pháp (Parse) và cập nhật thuộc tính src cho thẻ HTML5 <audio>, đồng thời thay đổi giao diện Player mà không làm gián đoạn trải nghiệm người dùng.

## Thiết kế cơ sở dữ liệu

Dựa trên phân tích, hệ thống sử dụng Microsoft SQL Server với các bảng dữ liệu sau:

### Xác định các bảng dữ liệu

#### Bảng Users (Người dùng)

Mục đích: Lưu trữ thông tin tài khoản và phân quyền.

Bảng 3‑1. Cấu trúc dữ liệu bảng Users

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| UserID | INT | PK, Identity | Mã người dùng (Tự tăng) |
| Username | VARCHAR(50) | NOT NULL, UNIQUE | Tên đăng nhập |
| Password | VARCHAR(100) | NOT NULL | Mật khẩu (đã mã hóa) |
| Fullname | NVARCHAR(100) | NULL | Tên hiển thị |
| Avatar | VARCHAR(255) | NULL | Đường dẫn ảnh đại diện |
| IsAdmin | BIT | DEFAULT 0 | 1: Admin, 0: User thường |

#### Bảng Artists (Nghệ sĩ)

Mục đích: Quản lý thông tin ca sĩ, nhạc sĩ.

Bảng 3‑2. Cấu trúc dữ liệu bảng Artists

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| ArtistID | INT | PK, Identity | Mã nghệ sĩ |
| Name | NVARCHAR(100) | NOT NULL | Tên nghệ sĩ |
| Image | VARCHAR(255) | NULL | Ảnh đại diện nghệ sĩ |

#### Bảng Songs (Bài hát)

Mục đích: Bảng quan trọng nhất, lưu trữ thông tin bài hát.

Bảng 3‑3. Cấu trúc dữ liệu bảng Songs

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| SongID | INT | PK, Identity | Mã bài hát |
| Title | NVARCHAR(200) | NOT NULL | Tên bài hát |
| ArtistID | INT | FK, NOT NULL | Mã nghệ sĩ (Liên kết bảng Artists) |
| Url | VARCHAR(500) | NOT NULL | Đường dẫn file nhạc (.mp3) |
| Image | VARCHAR(500) | NULL | Đường dẫn ảnh bìa (.jpg) |
| Duration | INT | DEFAULT 0 | Thời lượng (tính bằng giây) |

#### Bảng Favorites (Yêu thích)

Mục đích: Lưu trữ danh sách bài hát mà người dùng đã thả tim.

Bảng 3‑4. Cấu trúc dữ liệu bảng Favorites

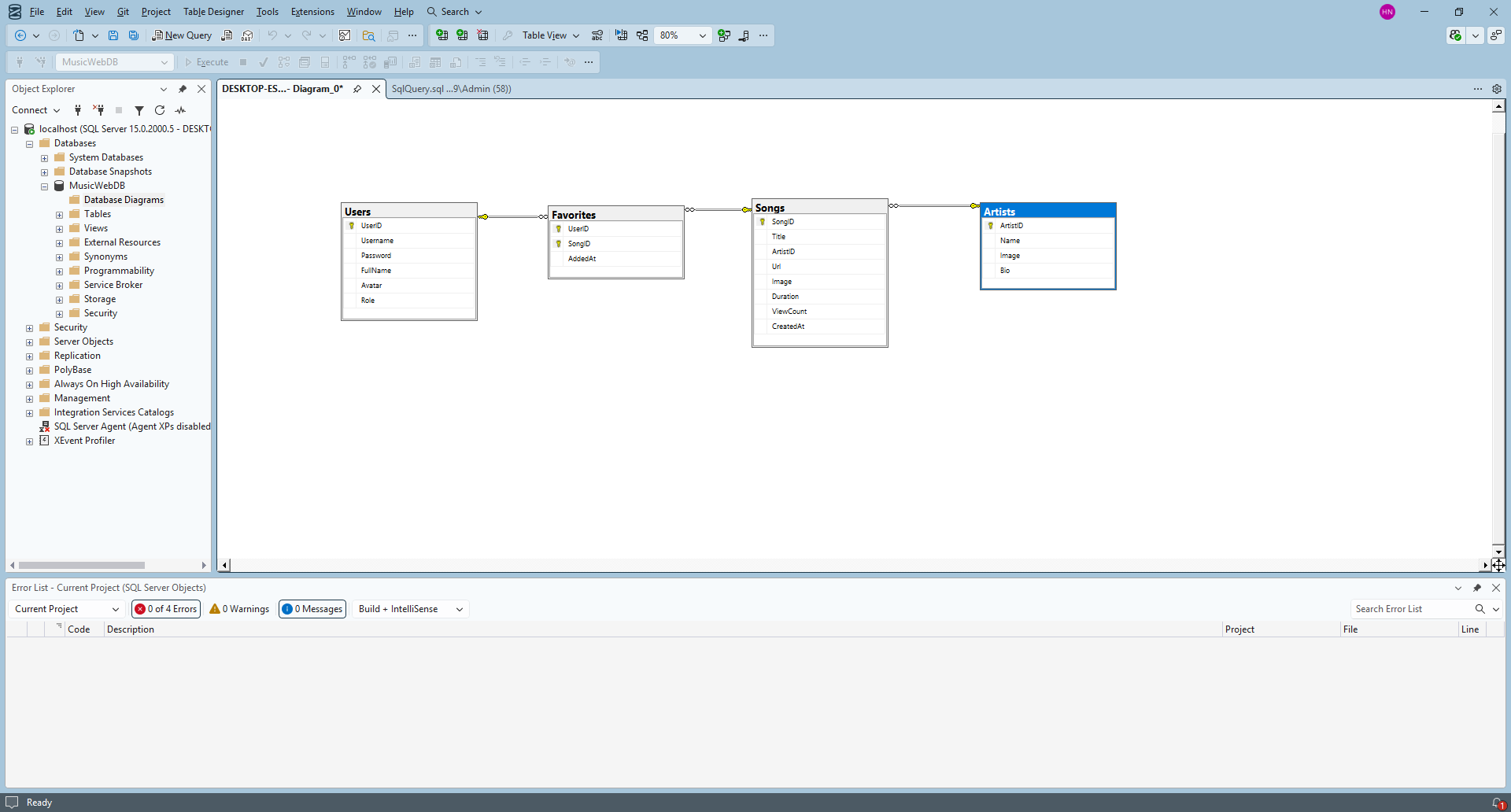
|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| FavID | INT | PK, Identity | Mã yêu thích |
| UserID | INT | FK, NOT NULL | Mã người dùng |
| SongID | INT | FK, NOT NULL | Mã bài hát |
| DateAdded | DATETIME | DEFAULT GETDATE() | Ngày thêm vào yêu thích |

### Thiết kế quan hệ giữa các bảng

#### Các mối quan hệ chính

* Artists - Songs (1:N):
* Một Nghệ sĩ có thể trình bày nhiều Bài hát.
* Khóa ngoại: Songs.ArtistID tham chiếu tới Artists.ArtistID.
* Users - Favorites (1:N):
* Một Người dùng có thể yêu thích nhiều bài hát.
* Khóa ngoại: Favorites.UserID tham chiếu tới Users.UserID.
* Songs - Favorites (1:N):
* Một Bài hát có thể nằm trong danh sách yêu thích của nhiều người (thông qua bảng trung gian Favorites).
* Khóa ngoại: Favorites.SongID tham chiếu tới Songs.SongID.

#### Ràng buộc toàn vẹn

* Ràng buộc tham chiếu: Khi xóa một Artist, hệ thống sẽ cảnh báo nếu Nghệ sĩ đó vẫn còn Songs liên quan (để tránh lỗi dữ liệu mồ côi). Tương tự khi xóa Songs sẽ ảnh hưởng đến Favorites.
* Ràng buộc duy nhất: Username trong bảng Users không được trùng nhau.
* Ràng buộc mặc định: ViewCount mặc định là 0 khi thêm mới bài hát.

Hình 3.5. Mô hình liên kết các bảng dữ liệu trong hệ thống HN Melody

# KẾT QUẢ VÀ ĐÁNH GIÁ

## Một số hình ảnh chương trình (Mô tả kết quả từng chức năng)

### Giao diện trang chủ (Home Page)

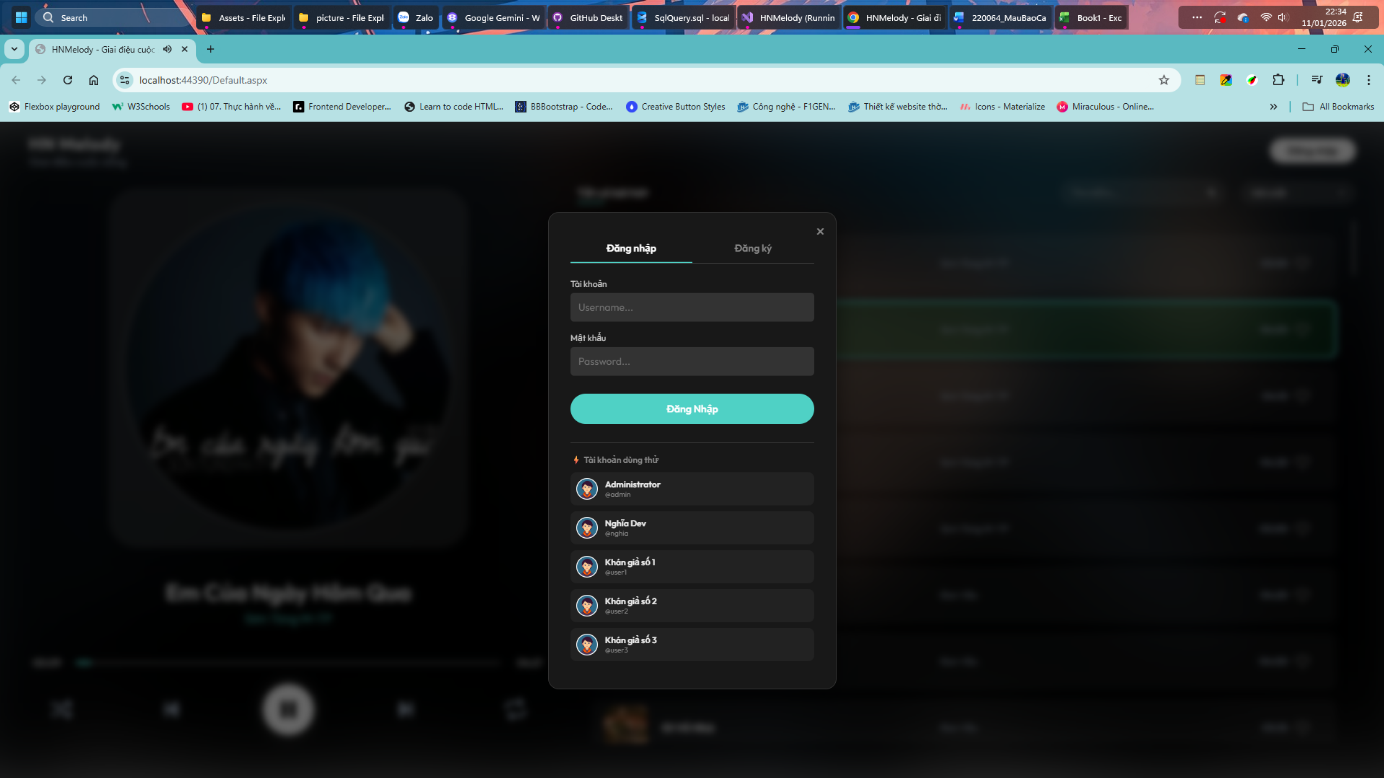
* Giao diện chính là nơi người dùng tương tác nhiều nhất, được thiết kế theo phong cách hiện đại (Dark Mode) với tông màu chủ đạo là xanh đen và xanh ngọc (Teal).
* Các thành phần chính:
* **Thanh điều hướng (Navbar):** Chứa logo "HN Melody", thanh tìm kiếm (Search Bar) và khu vực tài khoản người dùng.
* **Banner giới thiệu:** Hiển thị bài hát nổi bật nhất hoặc lời chào mừng.
* **Trình phát nhạc (Player):** Được ghim cố định ở bên trái, bao gồm ảnh đĩa than xoay tròn khi phát nhạc, tên bài hát, tên ca sĩ và các nút điều khiển (Play/Pause, Next, Prev, Random, Repeat).
* **Danh sách bài hát (Playlist):** Hiển thị danh sách các bài hát bên phải, bao gồm ảnh thumbnail, tên bài, ca sĩ và thời lượng. Bài hát đang phát sẽ được làm nổi bật (Active) với hiệu ứng sóng nhạc.

Hình 4‑1. Giao diện trang chủ của website HN Melody

### Chức năng Đăng ký / Đăng nhập

Hệ thống sử dụng cơ chế xác thực đơn giản nhưng hiệu quả, lưu trạng thái đăng nhập vào LocalStorage.

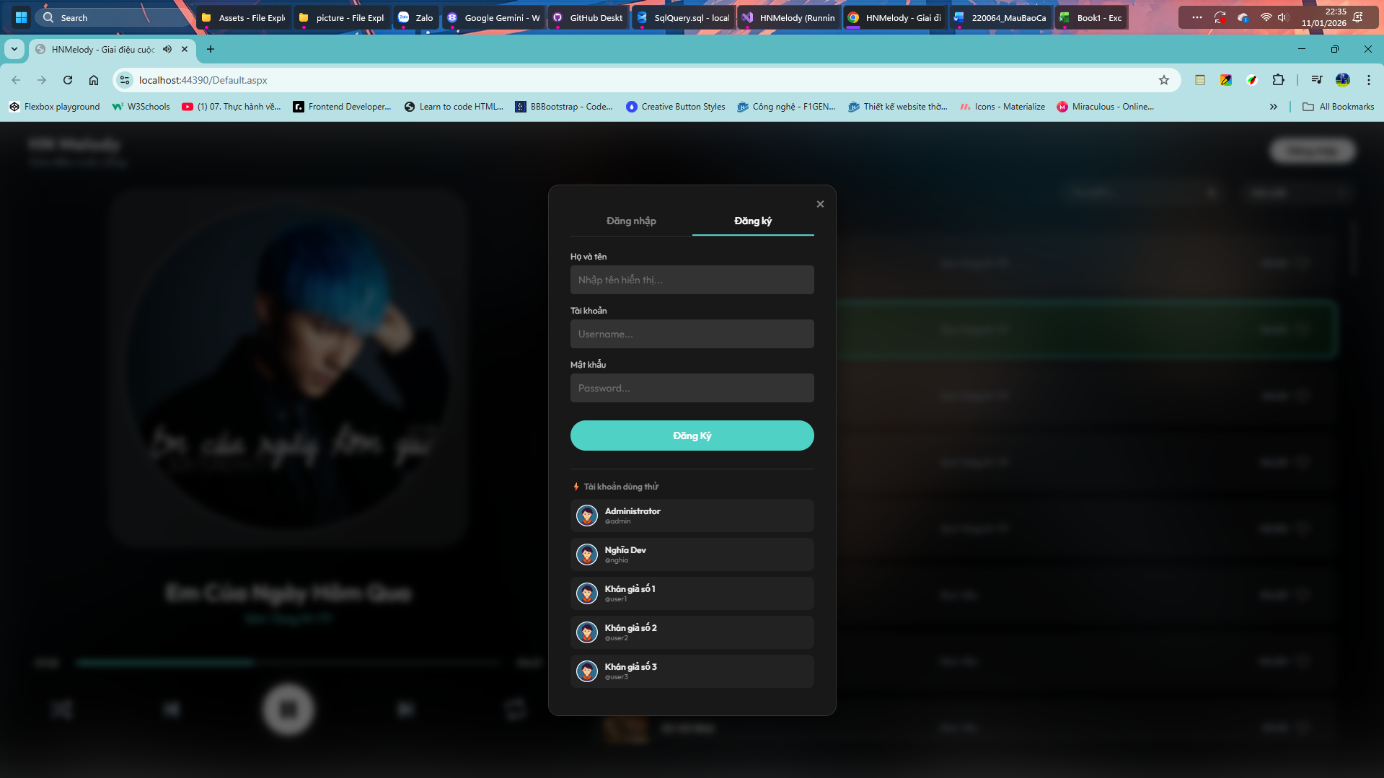
### Form Đăng nhập:

* Gồm 2 trường: Tên đăng nhập (Username) và Mật khẩu (Password).
* Nút "Đăng nhập" và liên kết chuyển sang trang Đăng ký.
* Hệ thống kiểm tra thông tin từ cơ sở dữ liệu SQL Server thông qua WebMethod.

Hình 4‑2. Giao diện Form Đăng nhập

### Form Đăng ký:

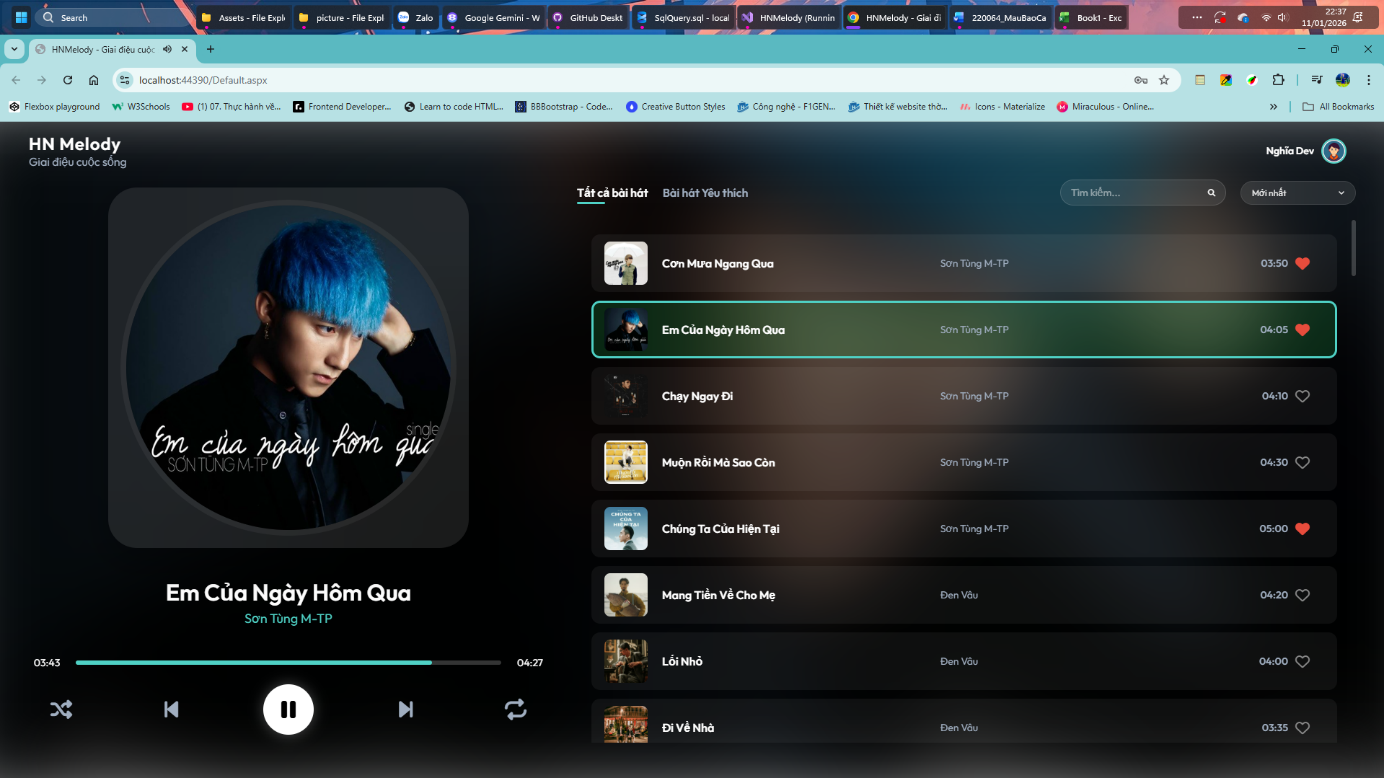
* Yêu cầu nhập: Họ tên, Tên đăng nhập, Mật khẩu.
* Kiểm tra trùng lặp tên đăng nhập trước khi tạo tài khoản mới.

Hình 4‑3. Giao diện Form Đăng nhập và Đăng ký

### Trạng thái giao diện sau khi đăng nhập:

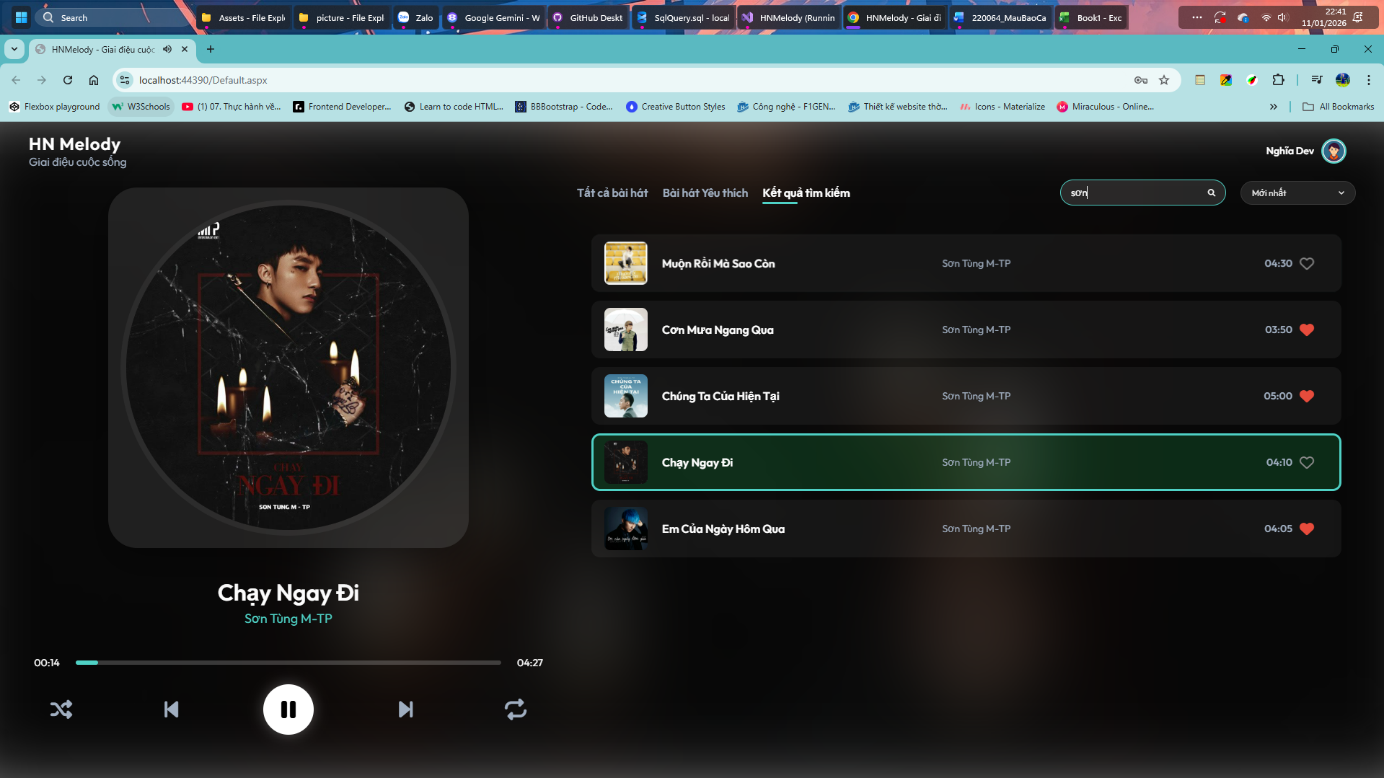
Sau khi người dùng xác thực thành công, hệ thống tự động cập nhật trạng thái giao diện (UI State) thông qua JavaScript mà không cần tải lại trang. Các thay đổi cụ thể bao gồm:

* Nút "Đăng nhập" được ẩn đi, thay thế bằng hiển thị Avatar và Tên người dùng.
* Tab "Yêu thích" được kích hoạt và hiển thị trên thanh điều hướng, cho phép người dùng truy cập danh sách bài hát cá nhân.
* Đối với tài khoản có quyền Quản trị (Admin), menu người dùng sẽ xuất hiện thêm liên kết truy cập vào trang Dashboard quản trị.

Hình 4‑4. Giao diện trang chủ sau khi đăng nhập thành công

### Chức năng Tìm kiếm bài hát (Live Search)

Chức năng tìm kiếm được xây dựng với kỹ thuật Debounce, giúp tối ưu hiệu năng.

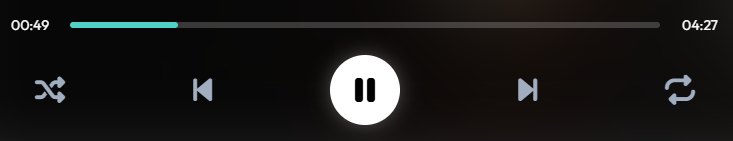
* Người dùng nhập từ khóa vào ô tìm kiếm.
* Hệ thống tự động lọc danh sách bài hát theo tên bài hoặc tên ca sĩ sau 0.5s ngừng gõ.
* Kết quả hiển thị ngay lập tức mà không cần tải lại trang.

Hình 4‑5. Kết quả tìm kiếm bài hát theo từ khóa

### Chức năng Nghe nhạc và Điều khiển (Player Control)

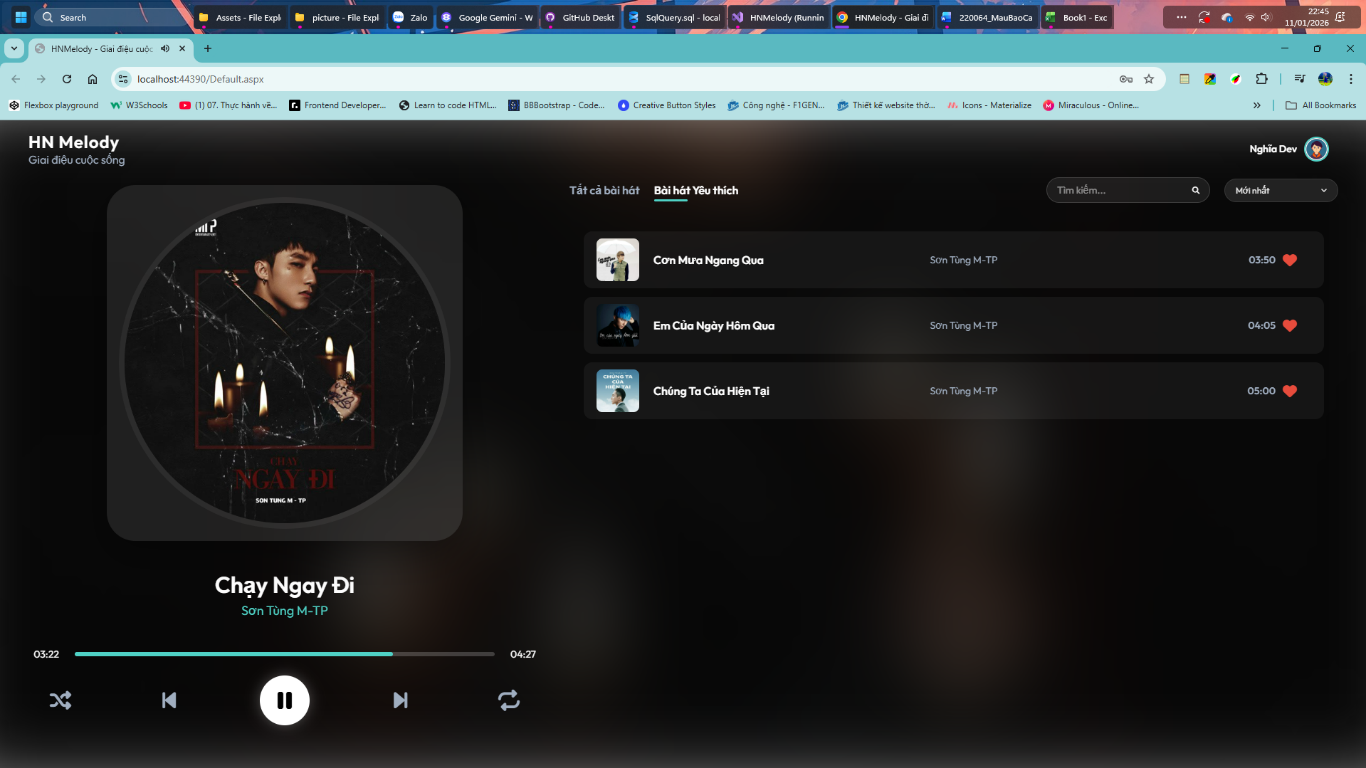
Đây là chức năng cốt lõi của hệ thống, xử lý các sự kiện Audio DOM của HTML5.

* Phát/Tạm dừng: Điều khiển trạng thái bài hát. Ảnh đĩa than sẽ quay khi nhạc phát và dừng khi nhạc tắt.
* Tua nhạc (Seek): Thanh tiến trình (Progress Bar) chạy theo thời gian thực. Người dùng có thể kéo thả để tua đến đoạn nhạc mong muốn.
* Chuyển bài: Tự động chuyển sang bài tiếp theo khi hết nhạc.
* Chế độ lặp lại/Ngẫu nhiên: Nút Repeat (lặp lại 1 bài) và Random (phát ngẫu nhiên) hoạt động đúng logic.

Hình 4‑6. Các nút điều khiển trình phát nhạc

### Chức năng Yêu thích (Favorites)

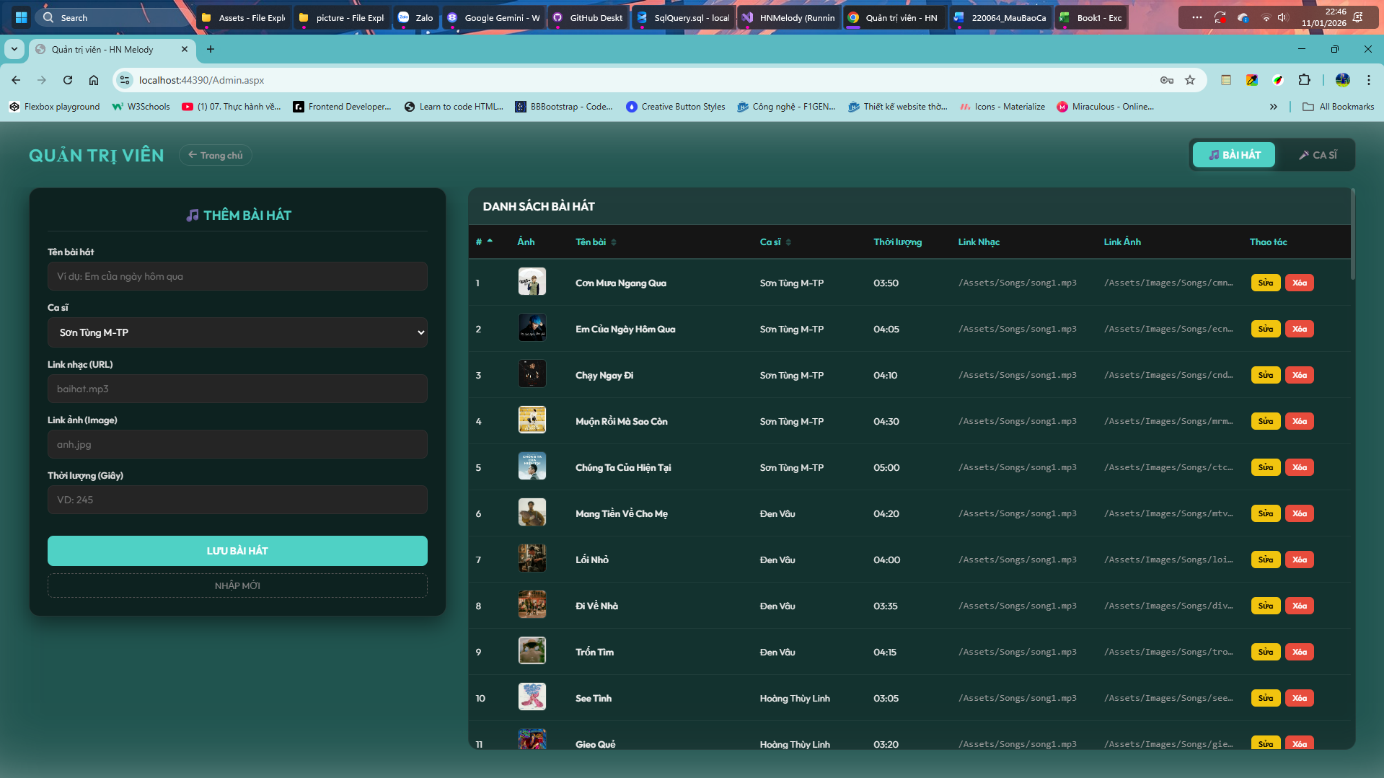
Người dùng sau khi đăng nhập có thể thả tim để lưu bài hát vào danh sách yêu thích.

* Nhấn biểu tượng trái tim để thêm/xóa bài hát khỏi danh sách yêu thích.
* Tab "Yêu thích" sẽ lọc và chỉ hiển thị các bài hát đã được thả tim.
* Dữ liệu được lưu trữ bền vững trong bảng Favorites của SQL Server.

Hình 4‑7. Danh sách bài hát yêu thích

### Trang Quản trị hệ thống (Admin Dashboard)

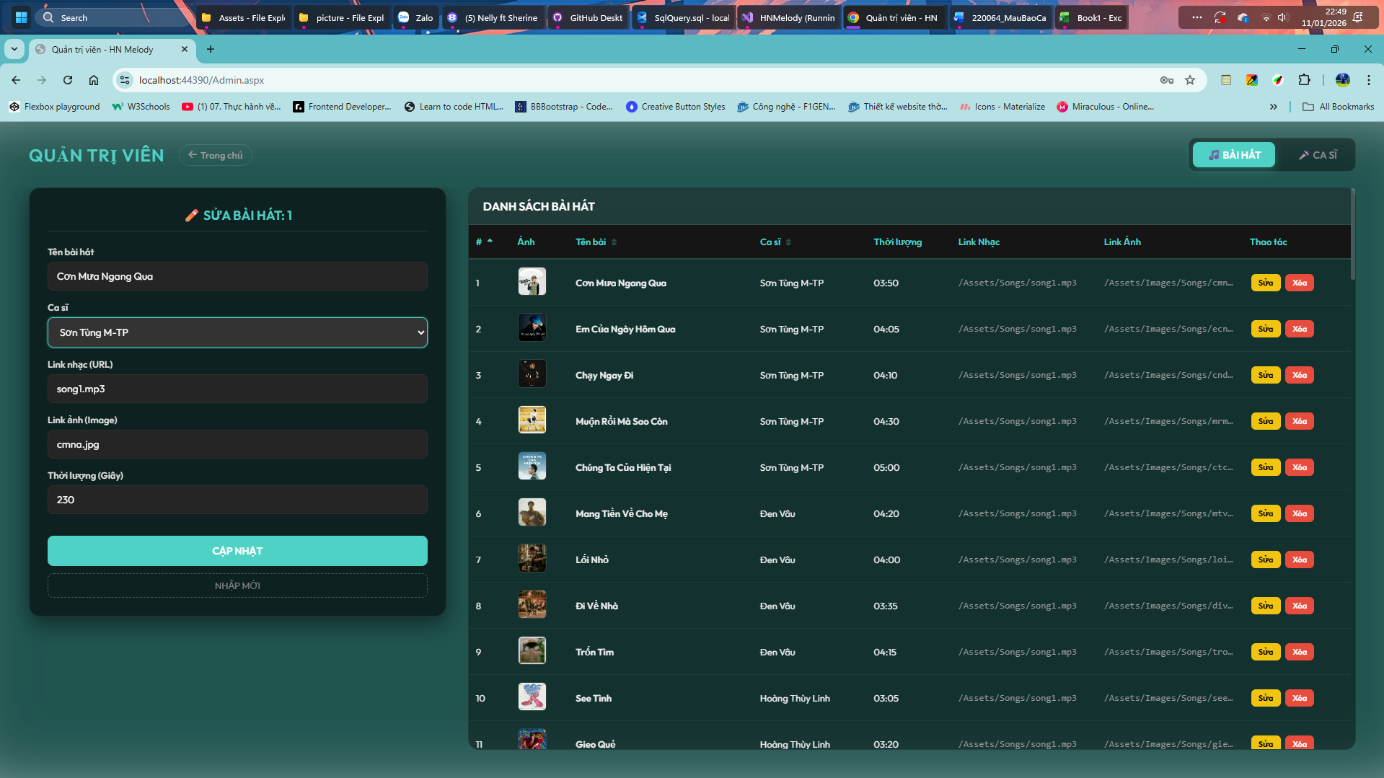
Trang dành riêng cho Quản trị viên, được bảo vệ bằng cơ chế phân quyền (chỉ tài khoản có Role='Admin' mới truy cập được).

* Menu bên trái hoặc Tab phía trên để chuyển đổi giữa quản lý "Bài hát" và "Ca sĩ".
* Nút "Trang chủ" để quay về giao diện nghe nhạc.

Hình 4‑8. Giao diện tổng quan trang Admin

### Quản lý Bài hát (Admin)

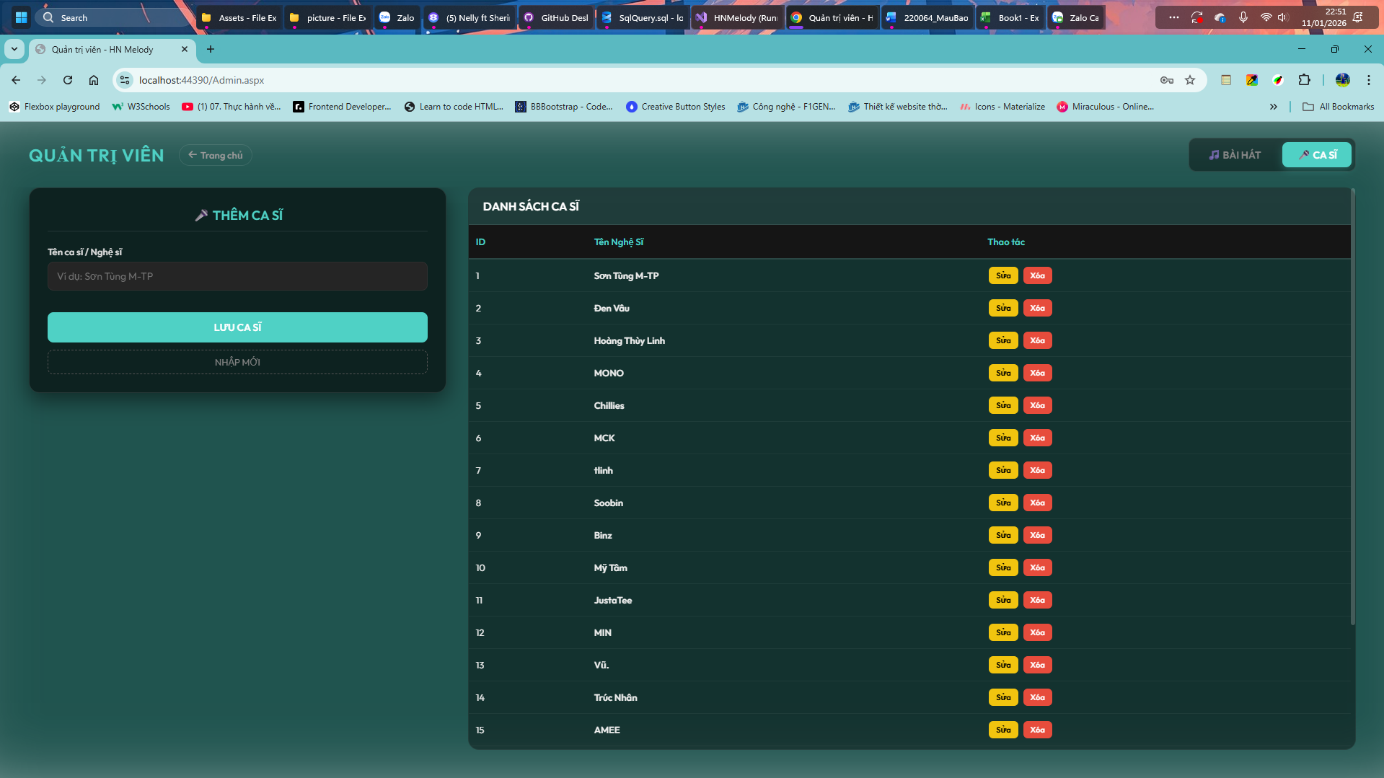
Cho phép Admin thực hiện đầy đủ các thao tác CRUD (Thêm, Xem, Sửa, Xóa) bài hát.

* Danh sách bài hát: Hiển thị dưới dạng bảng với các cột: ID, Ảnh, Tên bài, Ca sĩ, Thời lượng, Link nhạc.
* Form nhập liệu:
* Tự động gợi ý đường dẫn file (VD: nhập baihat.mp3 -> tự thành /Assets/Songs/baihat.mp3).
* Dropdown chọn Ca sĩ lấy dữ liệu động từ Database.
* Chức năng: Thêm mới, Cập nhật thông tin và Xóa bài hát.

Hình 4‑9. Giao diện Quản lý Bài hát

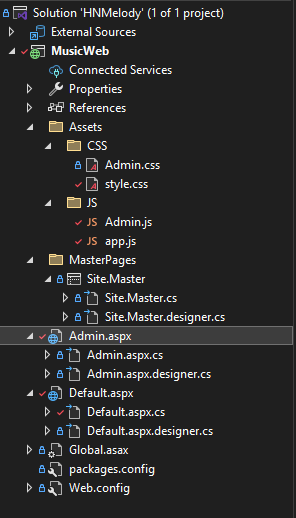
### Quản lý Ca sĩ (Admin)

Tương tự như quản lý bài hát, Admin có thể quản lý thông tin nghệ sĩ.

* Danh sách ca sĩ: Hiển thị ID, Tên và Ảnh đại diện.
* Ràng buộc toàn vẹn: Khi xóa ca sĩ, hệ thống sẽ kiểm tra và cảnh báo nếu ca sĩ đó đang có bài hát trong hệ thống.

Hình 4‑10. Giao diện Quản lý Ca sĩ

## Cấu trúc dự án (Project Structure)

Dự án được tổ chức theo cấu trúc chuẩn của ASP.NET WebForms, phân chia rõ ràng giữa Backend và Frontend.

Hình 4‑11. Cấu trúc thư mục dự án trong Visual Studio

Mô tả chi tiết các thư mục:

* App\_Code/ (hoặc Models/): Chứa các lớp xử lý kết nối CSDL (Database.cs).
* Assets/: Chứa tài nguyên tĩnh của trang web.
* CSS/: Chứa file style.css (Giao diện chính).
* JS/: Chứa app.js (Logic nghe nhạc) và admin.js (Logic quản trị).
* Images/: Chứa ảnh đại diện ca sĩ, ảnh bìa bài hát.
* Songs/: Chứa các file nhạc .mp3.
* Admin.aspx & Admin.aspx.cs: Trang giao diện và code xử lý cho Admin.
* Default.aspx & Default.aspx.cs: Trang chủ và code xử lý cho Người dùng.
* Web.config: Chứa chuỗi kết nối CSDL (Connection String).

## Đánh giá hiệu năng hệ thống

Hiệu năng là yếu tố then chốt quyết định trải nghiệm người dùng (UX) trên các ứng dụng giải trí đa phương tiện. Hệ thống **HN Melody** đã được đo lường và đánh giá dựa trên các tiêu chí về tốc độ phản hồi và mức độ tiêu thụ tài nguyên mạng. Kết quả thực nghiệm được tổng hợp trong bảng dưới đây:

Bảng 4‑1. Bảng đánh giá hiệu năng hệ thống

|  |  |  |
| --- | --- | --- |
| **Tiêu chí đánh giá** | **Kết quả thực nghiệm (Ước tính)** | **Đánh giá** |
| **Thời gian phản hồi Play/Pause** | < 300 ms | Rất tốt, phản hồi gần như tức thì. |
| **Thời gian tìm kiếm (Live Search)** | 400 – 600 ms | Tốt, độ trễ ở mức chấp nhận được, không gây khó chịu. |
| **Số request khi chuyển bài** | 1 Request (AJAX JSON) | Tối ưu băng thông mạng so với việc tải lại cả trang HTML. |
| **Tải lại trang (Full Postback)** | Không (0 lần) | Trải nghiệm mượt mà, âm nhạc không bị ngắt quãng. |
| **Mức sử dụng tài nguyên Client** | Thấp – Trung bình | Tương thích tốt với các thiết bị cấu hình thấp. |

Nhận xét: Việc áp dụng kỹ thuật WebMethod kết hợp với AJAX thay vì cơ chế Postback truyền thống của ASP.NET WebForms đã giúp giảm tải đáng kể cho Web Server. Hệ thống chỉ truyền tải dữ liệu cần thiết (JSON text) thay vì toàn bộ mã HTML của trang, giúp thao tác nghe nhạc diễn ra liên tục, liền mạch và tiết kiệm băng thông cho người dùng.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Đánh giá tổng quan

Đồ án "Xây dựng website nghe nhạc trực tuyến HN Melody sử dụng công nghệ ASP.NET WebForms" đã hoàn thành tốt các mục tiêu đề ra ban đầu. Hệ thống hoạt động ổn định, giao diện hiện đại (Dark Mode), tốc độ phản hồi nhanh nhờ áp dụng kỹ thuật tải trang bất đồng bộ (AJAX), đáp ứng tốt nhu cầu giải trí và quản lý nhạc cơ bản.

Qua quá trình thực hiện đồ án, tôi đã gặt hái được những kết quả quan trọng về mặt kiến thức và kỹ năng:

* Về công nghệ: Nắm vững quy trình xây dựng ứng dụng web với ASP.NET WebForms, hiểu sâu về cơ chế giao tiếp giữa Client và Server thông qua WebMethod để tối ưu hóa hiệu năng.
* Về kỹ năng lập trình: Nâng cao tư duy thuật toán thông qua việc xử lý logic phức tạp bằng JavaScript thuần (Vanilla JS) (xử lý phát nhạc, random, repeat, seek...) mà không phụ thuộc vào thư viện có sẵn.
* Về cơ sở dữ liệu: Có kinh nghiệm thiết kế và quản trị cơ sở dữ liệu SQL Server, biết cách xây dựng các ràng buộc toàn vẹn và tối ưu hóa truy vấn dữ liệu.
* Về tư duy sản phẩm: Rèn luyện được khả năng thiết kế giao diện (UI/UX) tập trung vào trải nghiệm người dùng, giải quyết các bài toán thực tế như xử lý điều hướng trang và lưu trữ trạng thái người dùng (LocalStorage).

## Kết quả đạt được

Sau quá trình nghiên cứu và thực hiện, đồ án đã đạt được những kết quả cụ thể sau:

* Về mặt sản phẩm:
* Xây dựng hoàn thiện website nghe nhạc với đầy đủ các chức năng cốt lõi: Nghe nhạc, điều khiển trình phát (Play, Pause, Next, Prev, Seek), chế độ lặp lại/ngẫu nhiên và tìm kiếm bài hát theo thời gian thực (Live Search).
* Thiết kế thành công giao diện người dùng hiện đại, thân thiện và tương thích tốt trên các trình duyệt phổ biến.
* Xây dựng hệ thống quản trị (Admin Dashboard) cho phép quản lý chặt chẽ dữ liệu Bài hát, Nghệ sĩ và phân quyền người dùng.
* Hệ thống cơ sở dữ liệu SQL Server được thiết kế chuẩn hóa, đảm bảo tính toàn vẹn dữ liệu và hiệu năng truy vấn ổn định.
* Về mặt kỹ thuật và công nghệ:
* Áp dụng thành công kỹ thuật AJAX thông qua WebMethod, giúp website hoạt động mượt mà, không bị tải lại trang khi thao tác, nâng cao trải nghiệm người dùng.
* Tổ chức mã nguồn JavaScript theo mô hình đối tượng (Object-based), giúp code gọn gàng, dễ quản lý và bảo trì.
* Giải quyết triệt để các bài toán logic phức tạp như: Xử lý vòng lặp chuyển trang (Redirect Loop) giữa Admin/User và logic tự động phát nhạc (Autoplay) thông minh.

## So sánh công nghệ ASP.NET WebForms và MVC

Để có cái nhìn khách quan về giải pháp công nghệ được lựa chọn, bảng dưới đây so sánh sự khác biệt giữa ASP.NET WebForms (sử dụng trong đồ án) và mô hình ASP.NET MVC hiện đại.

Bảng 5‑1. So sánh ASP.NET WebForms và ASP.NET MVC

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **ASP.NET WebForms (HN Melody)** | **ASP.NET MVC / Core** |
| Mô hình kiến trúc | Hướng sự kiện (Event-driven) & Page Controller | Model – View – Controller (Tách biệt rõ ràng) |
| Kiểm soát HTML | Hạn chế (Do Server Controls tự sinh mã) | Rất tốt (Kiểm soát 100% mã HTML sinh ra) |
| Cơ chế hoạt động | Dựa trên ViewState và Postback | Dựa trên các Action và Routing |
| Độ phức tạp | Dễ tiếp cận cho người mới bắt đầu | Cần thời gian tìm hiểu kiến trúc |
| Khả năng tùy biến | Trung bình | Cao, dễ dàng mở rộng và kiểm thử (Unit Test) |

Lý do lựa chọn: Mặc dù ASP.NET MVC là xu hướng hiện đại, việc lựa chọn ASP.NET WebForms cho đồ án này là phù hợp với mục tiêu học tập: Nắm vững tư duy lập trình hướng sự kiện, hiểu rõ vòng đời trang web (Page Lifecycle) và cách xử lý dữ liệu phía Server trước khi tiếp cận các kiến trúc phức tạp hơn.

## Hướng phát triển

Để hệ thống HN Melody hoàn thiện hơn và có khả năng ứng dụng thực tế cao hơn, tôi đề xuất các hướng nghiên cứu và phát triển tiếp theo:

* Nâng cấp chức năng quản lý:
* Tích hợp tính năng Upload file trực tiếp từ giao diện Admin (thay vì nhập đường dẫn thủ công), tự động lưu trữ file vào thư mục Server.
* Bổ sung chức năng thống kê báo cáo chi tiết bằng biểu đồ (Chart) về lượt truy cập và xu hướng nghe nhạc.
* Mở rộng tính năng người dùng:
* Phát triển tính năng Playlist cá nhân: Cho phép người dùng tự tạo nhiều danh sách phát riêng biệt ngoài danh sách "Yêu thích".
* Tích hợp tính năng Bình luận và Đánh giá bài hát để tăng tính tương tác cộng đồng.
* Xây dựng thuật toán Gợi ý bài hát (Recommendation System) dựa trên lịch sử nghe nhạc của người dùng.
* Tối ưu hóa và Bảo mật:
* Triển khai cơ chế xác thực bảo mật cao hơn (OAuth, xác thực qua Email).
* Nghiên cứu chuyển đổi lưu trữ file nhạc sang các dịch vụ đám mây (Cloud Storage) như AWS S3 hoặc Google Cloud để giảm tải cho Web Server.
* Phát triển phiên bản ứng dụng di động (Mobile App) sử dụng chung cơ sở dữ liệu và API với website.

## Kết luận chung

Thông qua đồ án này, sinh viên không chỉ nắm vững kỹ thuật lập trình website nghe nhạc, mà còn hiểu rõ bản chất của mô hình Client–Server, xử lý bất đồng bộ (AJAX) và tổ chức hệ thống phần mềm theo hướng dễ bảo trì. Đồ án không chỉ dừng lại ở một sản phẩm học tập mà còn là nền tảng vững chắc để tôi tiếp tục nghiên cứu các kiến trúc hiện đại hơn như ASP.NET Core API, Microservices hoặc các Framework Front-end (React/VueJS) trong tương lai, hướng tới việc trở thành một lập trình viên Full-stack chuyên nghiệp.

# DANH MỤC TÀI LIỆU THAM KHẢO

**Tiếng Việt:**

1. Phương Lan. (2012). *Giáo trình Thiết kế Web căn bản với HTML, CSS và JavaScript*. Nhà xuất bản Giáo dục Việt Nam.
2. Phạm Hữu Khang. (2010). *Xây dựng ứng dụng Web với ASP.NET 4.0 & C#*. Nhà xuất bản Lao động Xã hội.
3. Nguyễn Văn Ba. (2009). *Giáo trình Hệ quản trị cơ sở dữ liệu*. Nhà xuất bản Đại học Quốc gia Hà Nội.
4. Trung tâm Tin học Đại học Khoa học Tự nhiên. (2015). *Lập trình ứng dụng Web với ASP.NET WebForms*. Tài liệu lưu hành nội bộ.
5. Dương Quang Thiện. (2008). *Phân tích và thiết kế hệ thống thông tin*. Nhà xuất bản Tổng hợp TP.HCM.

**Tiếng Anh (Quốc tế):**

1. Microsoft Corporation. (2024). *ASP.NET Web Forms Documentation*. Retrieved from <https://learn.microsoft.com/en-us/aspnet/web-forms/>
2. Microsoft Corporation. (2024). *Exposing Web Services to Client Script (WebMethod)*. Retrieved from [https://learn.microsoft.com/en-us/previous-versions/aspnet/byxd99hx(v=vs.100](https://www.google.com/search?q=https://learn.microsoft.com/en-us/previous-versions/aspnet/byxd99hx(v%3Dvs.100))
3. Evjen, B., Hanselman, S., & Rader, D. (2010). *Professional ASP.NET 4 in C# and VB*. Wrox Press.
4. Mozilla Developer Network (MDN). (2024). *HTML5 Audio and Video API*. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio>
5. W3Schools. (2024). *SQL Server Tutorial*. Retrieved from <https://www.w3schools.com/sql/>
6. Spaanjaars, I. (2014). *Beginning ASP.NET 4.5.1: in C# and VB*. Wrox Press.
7. Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development*. Wiley.
8. Crockford, D. (2008). *JavaScript: The Good Parts*. O'Reilly Media. (Cuốn sách kinh điển về JS thuần).

**Website & Tài nguyên trực tuyến:**

1. Stack Overflow. (2024). *ASP.NET Web Forms Questions*. Retrieved from <https://stackoverflow.com/questions/tagged/webforms>
2. CSS-Tricks. (2024). *A Complete Guide to Flexbox*. Retrieved from <https://css-tricks.com/snippets/css/a-guide-to-flexbox/> (Tài liệu tham khảo cho giao diện)
3. Font Awesome. (2024). *Icon Library Documentation*. Retrieved from <https://fontawesome.com/docs>

Github: https://github.com/huunghia160699/ASPNET-DT23TTC13-NguyenHuuNghia-MusicWebsite.git