**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO DỰ ÁN MÔN PHÁT TRIỂN**

**ỨNG DỤNG WEB 1**

**TÊN ĐỀ TÀI: MUSIC APP**

Giảng viên hướng dẫn: Thầy Bùi Chí Thành

Sinh viên thực hiện: Nguyễn Thông Thiên

Học Kỳ I - Năm 2024

Lớp: 64.CNTT-CLC1

**Mục lục:**

1. **Giới thiệu 3**
2. **Phân tích yêu cầu 3**
3. **Thiết kế 8**
4. **Cài đặt 15**
5. **Kiểm thử 19**
6. **Kết luận 20**
7. **Tài liệu tham khảo 21**
8. **Giới Thiệu**
9. **Mục tiêu của project**

* Hỗ trợ người dùng quản lý danh sách bài hát có đủ các chức năng cơ bản.
* Học tập và làm quen với ASP.NET MVC, Entity Framework, và SQL Server.

1. **Phạm vi và hạn chế**

* Phạm vi: Tìm kiếm, quản lý bài hát, playlist và nghe nhạc.
* Hạn chế: Không hỗ trợ tải trực tiếp tệp mp3 lên từ máy tính cá nhân mà yêu cầu sử dụng URL trực tuyến (Dropbox).

1. **Công nghệ sử dụng**

* ASP.NET MVC Framework: Xây dựng backend và kết nối dữ liệu.
* Entity Framework: Tương tác với cơ sở dữ liệu SQL Server.
* SQL Server: Lưu trữ thông tin người dùng, bài hát, và playlist.
* HTML, CSS, JavaScript, Bootstrap: Xây dựng giao diện người dùng.
* Dropbox: Lưu trữ các tệp mp3 và hình ảnh liên quan đến bài hát.

1. **Phân Tích Yêu Cầu**
2. **Yêu cầu chức năng**

* **Chức năng admin:**
* Xem/xóa/sửa/thêm danh sách bài hát mới.
* Xem/xóa/sửa/thêm danh sách và thông tin của user.
* Lưu ý: Không cho phép upload file trực tiếp mà sẽ lưu link direct trong SQL Server (link của file mp3/image đã có ở GG Drive).
* Ngoài những chức năng trên, Admin còn có thể sử dụng các chức năng của User
* **Chức năng user:**
* Xem thông tin cá nhân.
* Xem danh sách bài hát, playlist, album, nghệ sĩ.
* Tạo, sửa, xóa playlist.
* Thêm/ xóa bài hát từ playlist.
* Tìm kiếm bài hát qua thanh search.
* Nghe nhạc.
* Hiển thị thông tin bài hát, album, nghệ sĩ ở Rightsection.
* **Chức năng phát nhạc:**
* Play/pause pre/next bài hát.
* Điều khiển nhạc qua thanh slider.

1. **Yêu cầu phi chức năng**

* Giao diện mượt mà và dễ sử dụng.
* Tốc độ phản hồi nhanh khi truy vấn SQL Server và link direct (mp3/image)
* Link nhạc và hình ảnh được hiển thị từ DropBox (dùng direct link).

1. **Use case diagram và mô tả**

* **Actor trong hệ thống:**
* Admin**:** Quản lý bài hát, quản lý user.
* User**:** Sử dụng các chức năng chính như tạo/sửa/xóa playlist, nghe nhạc, tìm kiếm.
* **Các Use Case chính:**
* Admin**:**
  + Quản lý bài hát và người dùng.
* User**:**
  + Quản lý playlist: Tạo, sửa, xóa playlist.
  + Thêm/Xóa bài hát vào playlist.
  + Nghe nhạc.
  + Tìm kiếm bài hát.
  + Xem thông tin bài hát, album, nghệ sĩ.
* **Use Case "Nghe nhạc":**
* Actor: User
* Mô tả:
  + User thực hiện click vào icon Play trong danh sách để chọn bài hát muốn nghe.
  + Hệ thống thực hiện truy vấn đến Cơ sở dữ liệu (SQL Server) để lấy thông tin bài hát, bao gồm (Name Song, Artist, Image Song, File mp3 Path (Link nhạc direct) )
  + Hệ thống trả về thông tin bài hát và gửi link nhạc trực tiếp đến trình phát nhạc tích hợp trong giao diện web (HTML5 Audio Player).
  + Giao diện Right Section hiển thị đầy đủ thông tin bài hát, bao gồm: Image, thông tin nếu có khi song thuộc 1 album/ artist nào đó.
* **Use Case "Tìm kiếm bài hát":**
* Actor: User
* Mô tả:
  + User nhập từ khóa vào thanh tìm kiếm.
  + Hệ thống gửi truy vấn đến cơ sở dữ liệu để tìm các kết quả liên quan (song, album, artist) ứng với mỗi tab.
  + Hiển thị danh sách kết quả phù hợp.

1. **Sequence diagram cho các chức năng chính**

#### **Sequence Diagram: Tìm kiếm bài hát**

* Actors**:**
  + User/ Admin.
  + HomeController: nhận từ khóa tìm kiếm từ client (giao diện) và xử lý logic liên quan bằng phương thức Search. Gửi truy vấn SQL tới cơ sở dữ liệu và trả kết quả lại cho client.
  + Database: Lưu trữ thông tin bài hát, nghệ sĩ, album.
* Các bước**:**
  + Người dùng (User/Admin):
    - Nhập từ khóa vào thanh tìm kiếm.
    - Nhấn nút Search hoặc phím Enter.
  + JavaScript gửi yêu cầu AJAX với từ khóa tìm kiếm tới server:
    - Lấy từ khóa từ thanh tìm kiếm.
    - Kiểm tra xem từ khóa có hợp lệ (không trống).
    - Gửi yêu cầu AJAX tới phương thức Search trong HomeController, kèm theo từ khóa tìm kiếm.
  + HomeController: Nhận từ khóa và Gọi stored procedure trong SQL Server.
  + Database (SQL Server) trả về danh sách bài hát phù hợp (bao gồm thông tin như tên bài hát, nghệ sĩ, album, và đường dẫn file MP3).
  + HomeController: nhận kết quả từ stored procedurechuyển thành danh sách JSON và trả về cho client.
  + JavaScript hiển thị danh sách bài hát lên giao diện (ListView trong Center Section) bằng phương thức displaySearchResults.

#### **Sequence Diagram: Nghe nhạc**

* Actors**:**
  + User/ Admin
  + HomeController: Điều khiển logic xử lý và truy vấn thông tin bài hát từ cơ sở dữ liệu thông qua phương thức GetSongInfo.
  + Database: Lưu trữ thông tin bài hát, nghệ sĩ, album. Trả về chi tiết bài hát bao gồm thông tin cơ bản và liên kết đến file MP3.
* Các bước**:**
  + User nhấn vào biểu tượng **Play** bên cạnh một bài hát trong danh sách (.btn-play-song).
  + JavaScript thu thập thông tin bài hát từ các thuộc tính của phần tử HTML liên quan (như data-song-id, data-song-file).
  + Gửi yêu cầu đến server:
    - Nếu bài hát chưa phát, JavaScript gọi phương thức playSong(song).
    - Phương thức này gửi yêu cầu AJAX đến HomeController.GetSongInfo để lấy thông tin chi tiết của bài hát.
  + HomeController:
    - Phương thức GetSongInfo nhận songId từ yêu cầu.
    - Gọi truy vấn Entity Framework để lấy thông tin bài hát, bao gồm (SongName, ArtistName, Bio, AlbumName, FilePath, ImagePath).
    - Trả kết quả dưới dạng JSON.
  + Client nhận thông tin và phát nhạc
    - JavaScript nhận JSON chứa thông tin bài hát từ server.
    - Cập nhật thông tin vào giao diện: hình ảnh bài hát (#song-image),Tên bài hát và nghệ sĩ (#now-playing-song-name, #artist-name)
    - Cập nhật nguồn phát nhạc (#audio-player) và bắt đầu phát nhạc.
  + Giao diện cập nhật thông tin bài hát trong RightSection.

#### **Sequence Diagram: Thêm bài hát vào playlist**

#### Actors**:**

* + User/Admin:chọn bài hát từ danh sách và thêm nó vào một playlist hiện có.
  + HomeController:
    - Xử lý logic thêm bài hát vào playlist thông qua phương thức AddSongToPlaylist.
  + Database: Lưu thông tin liên kết giữa bài hát và playlist (bảng SongPlaylist).

#### Các bước **:**

* + User:
    - User nhấn nút Add to Playlist (biểu tượng +) bên cạnh bài hát.
    - Một danh sách các playlist hiện có sẽ xuất hiện (dưới dạng dropdown).
  + JavaScript xử lý sự kiện: Gửi yêu cầu AJAX tới /Home/AddSongToPlaylist với songId và playlistId
  + HomeController:
    - Nhận yêu cầu và xử lý logic: Truy vấn playlist và kiểm tra bài hát và thêm vào playlist..
    - Cập nhật dữ liệu trong Database.
  + Database: Quan hệ giữa bài hát và playlist được lưu vào bảng trung gian (nếu sử dụng quan hệ nhiều-nhiều).
  + JavaScript: Nhận phản hồi JSON và hiển thị kết quả trên giao diện.

1. **Thiết Kế**
2. **Kiến trúc tổng thể:**

#### **Mô hình tổng quan:**

* Hệ thống sử dụng mô hình MVC (Model-View-Controller) để phân chia trách nhiệm giữa các thành phần, giúp việc phát triển, bảo trì và mở rộng ứng dụng trở nên dễ dàng hơn.

#### **Thành phần chính trong kiến trúc MVC:**

##### **Model:**

* Chức năng:
  + Chịu trách nhiệm kết nối và quản lý dữ liệu từ cơ sở dữ liệu SQL Server.
  + Xử lý các truy vấn thêm, sửa, xóa, và lấy dữ liệu từ các bảng như:
    - Song: Lưu thông tin bài hát (tên, link nhạc, link ảnh, album...).
    - Playlist: Lưu danh sách playlist của user.
    - User: Quản lý thông tin người dùng (tên, email, mật khẩu).
    - Album và Artist: Quản lý thông tin album và nghệ sĩ.
* Công nghệ sử dụng:
  + Entity Framework (EF): Để quản lý dữ liệu và thực hiện truy vấn SQL một cách dễ dàng.
  + LINQ: Hỗ trợ truy vấn dữ liệu linh hoạt trong các lớp Model.

##### 

##### **View:**

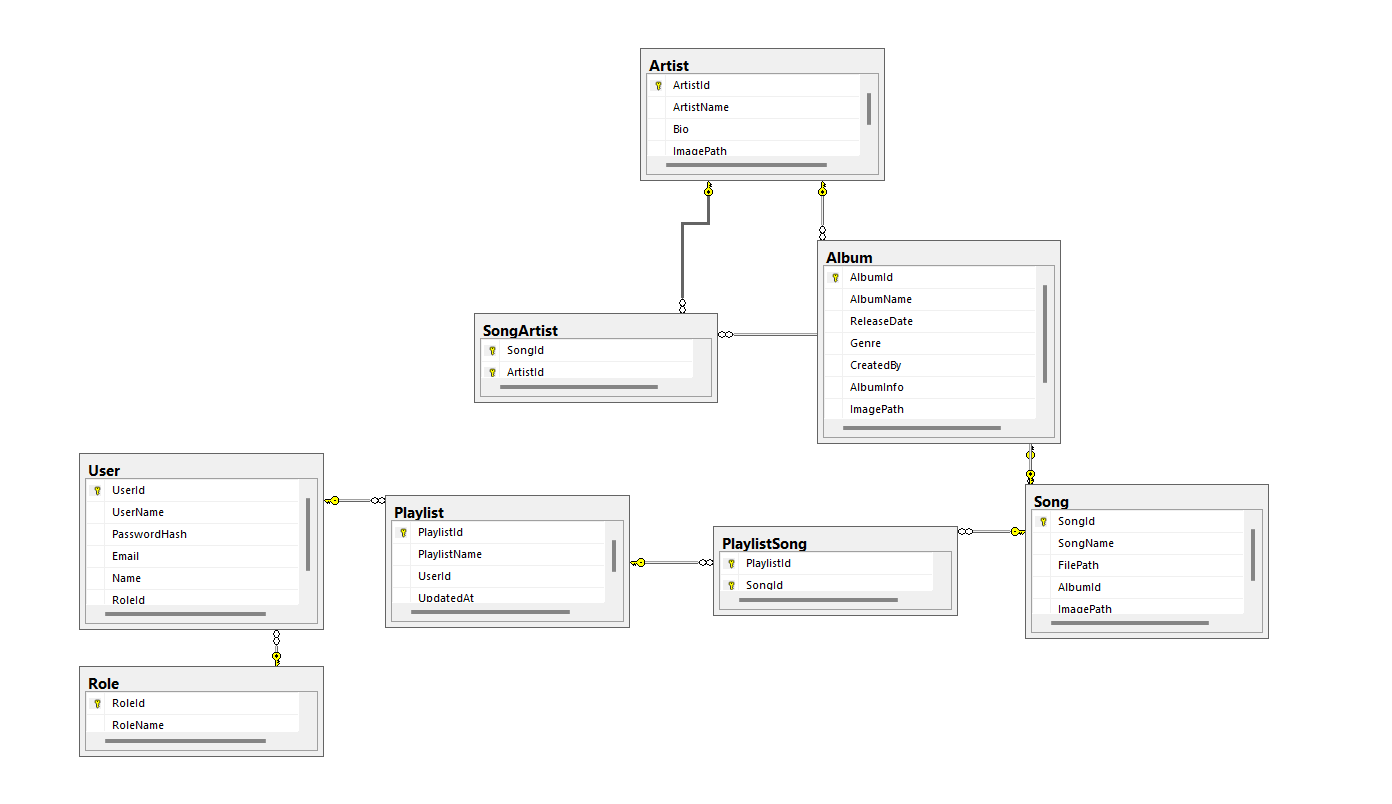
* Chức năng:
  + Hiển thị giao diện người dùng và nhận tương tác từ người dùng.
* Các thành phần giao diện chính:
  + Thanh tìm kiếm:
    - Thành phần: Input text và Button để tìm kiếm bài hát, playlist, album, hoặc nghệ sĩ.
  + Danh sách bài hát:
    - Hiển thị danh sách bài hát, kết quả tìm kiếm, hoặc nội dung trong playlist.
    - Thành phần chính:
      * Button: Hành động như Play, Add to Playlist.
      * ListView: Hiển thị danh sách bài hát.
      * TabPane:
        + Songs Tab : Hiển thị danh sách bài hát
        + Albums Tab: Hiển thị danh sách album.
        + Artists Tab: Hiển thị danh sách nghệ sĩ.
  + Right Section:
    - Hiển thị thông tin chi tiết về bài hát, album, nghệ sĩ.
    - Thành phần chính:
      * TabPane:
        + Image: Ảnh bài hát hoặc album.
        + Text: Thông tin tên bài hát, album, nghệ sĩ.
  + Thanh phát nhạc:
    - Điều khiển phát nhạc với các chức năng Play/Pause, Next/Previous, Seek, Volume
    - Thành phần chính:
      * Slider: Thanh trượt để tua bài hát.
      * Label: Hiển thị tên bài hát và thời gian phát.
  + Công nghệ sử dụng:
    - HTML/CSS: Thiết kế giao diện người dùng.
    - Bootstrap: Hỗ trợ responsive và các thành phần giao diện.
    - JavaScript (jQuery): Tương tác động trên giao diện.

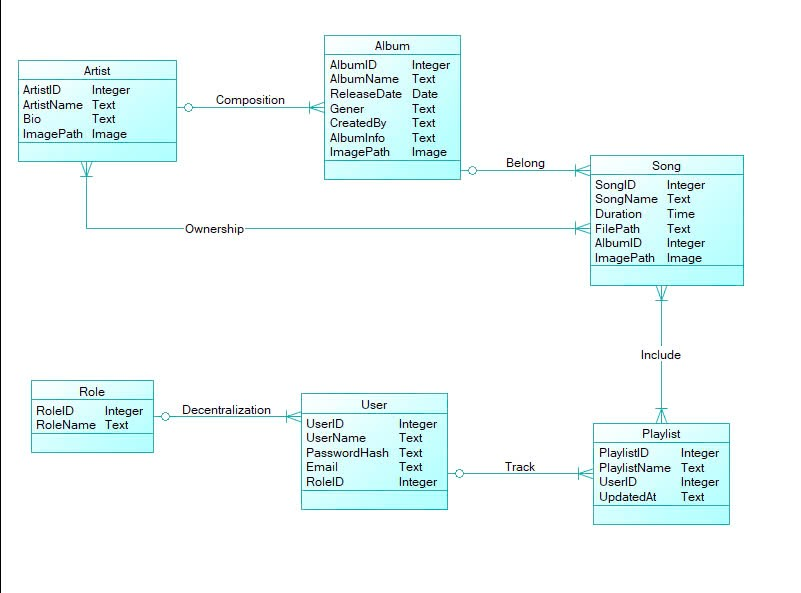
##### **Controller:**

* Chức năng:
  + Xử lý logic nghiệp vụ và giao tiếp giữa Model và View.
* Các công việc chính:
  + Nhận sự kiện từ giao diện (như click, double-click, nhập dữ liệu).
  + Gửi yêu cầu tới Model để xử lý dữ liệu hoặc truy vấn cơ sở dữ liệu.
  + Nhận dữ liệu từ Model và cập nhật lại View.
* Các controller chính:
  + HomeController: Quản lý logic chính của user (tìm kiếm, phát nhạc, quản lý playlist...).
  + AccountController: Xử lý login/logout.
  + Profile\_64132265Controller: Quản lý thông tin người dùng.
* Công nghệ sử dụng:
  + ASP.NET MVC Framework: Xây dựng các controller và xử lý routing.

#### **. Công nghệ sử dụng:**

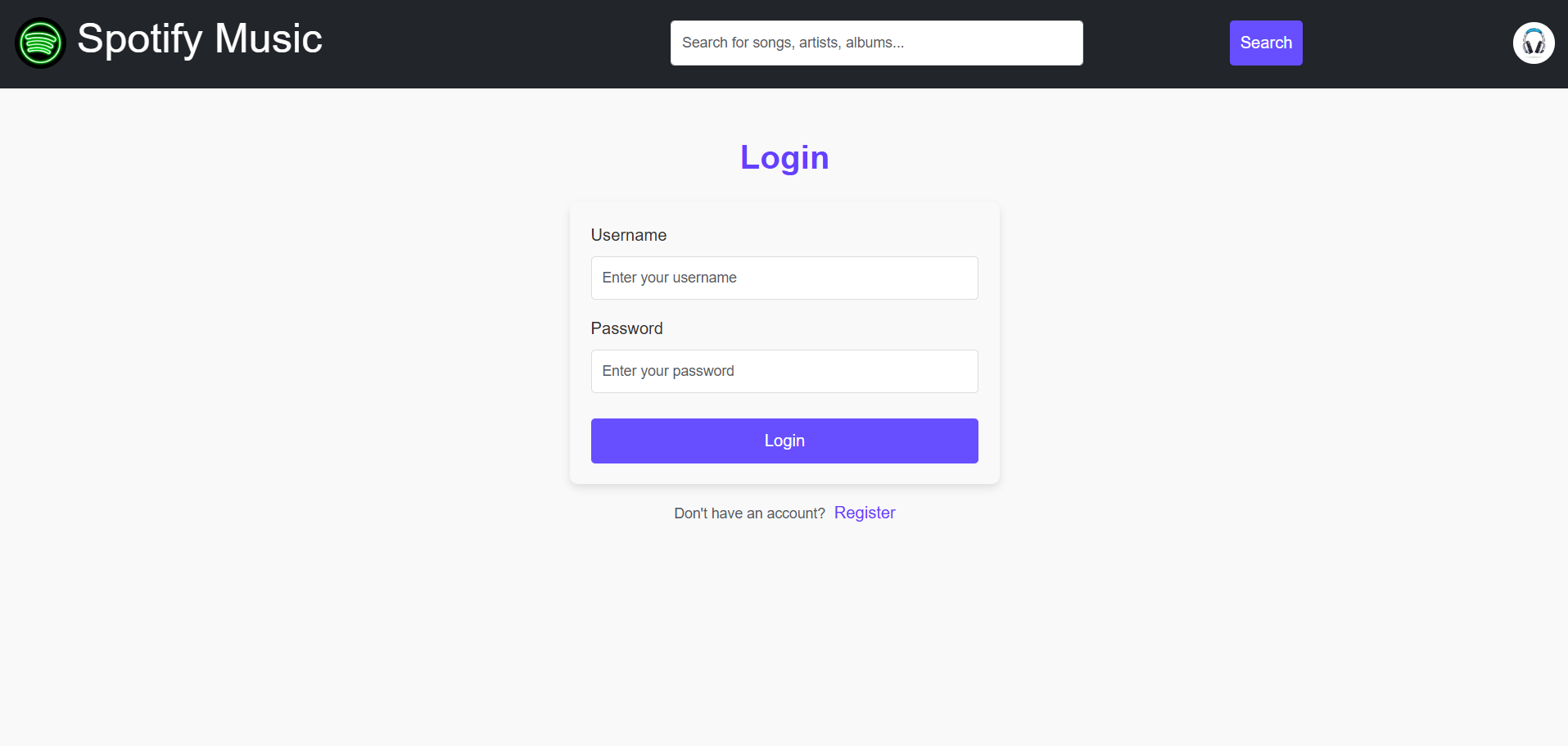
* Ngôn ngữ lập trình: C#..
* Framework: ASP.NET MVC.
* Cơ sở dữ liệu: SQL Server.
* Thư viện hỗ trợ:
  + ntity Framework: Tương tác với cơ sở dữ liệu.
  + jQuery: Xử lý tương tác người dùng..
  + Bootstrap: Thiết kế giao diện
* Công cụ phát triển:
  + Visual Studio: IDE chính để phát triển ứng dụng.
  + SQL Server Management Studio (SSMS): Để quản lý cơ sở dữ liệu.

1. **DatabaseSchema**
2. **Class Diagram**

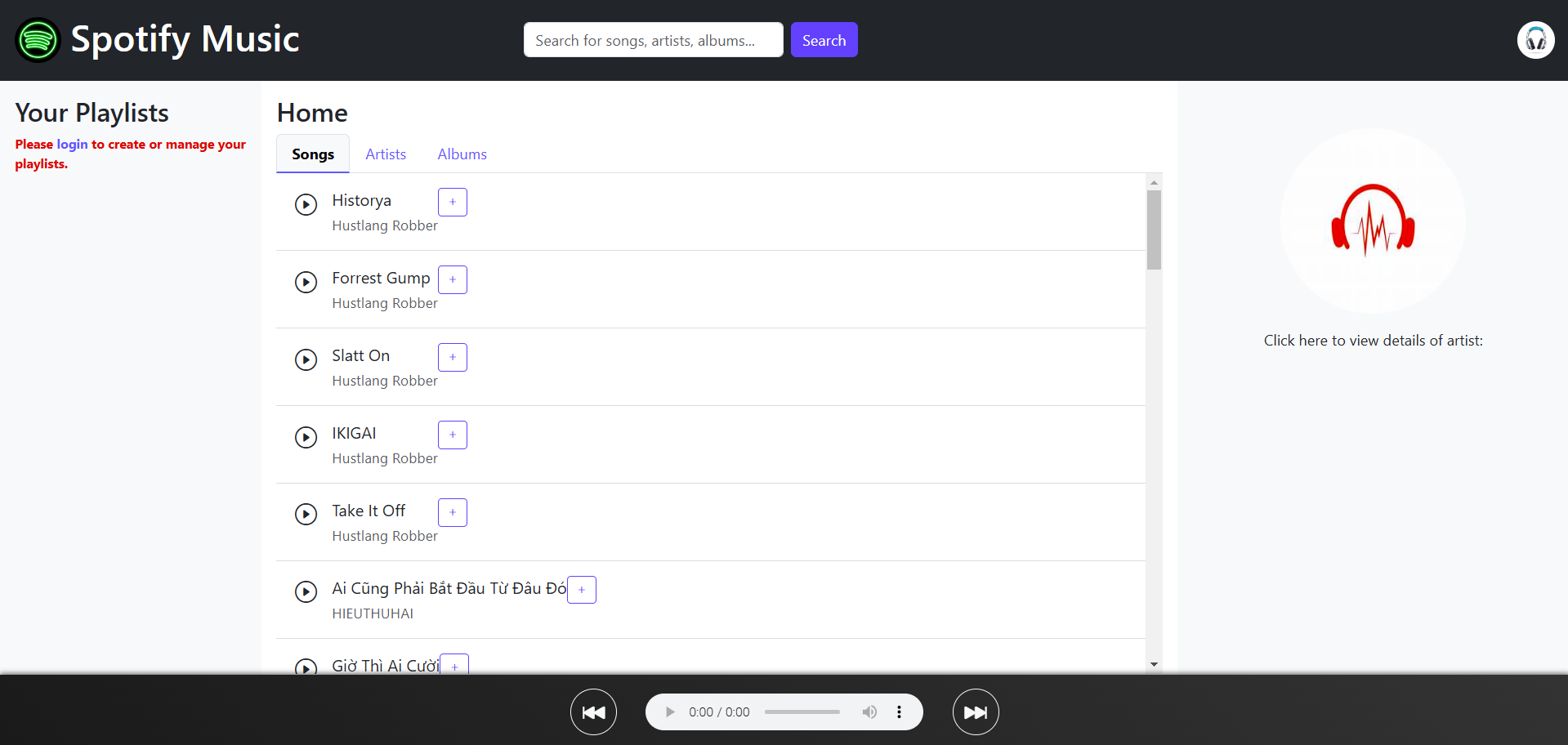


1. **Giao diện người dùng**

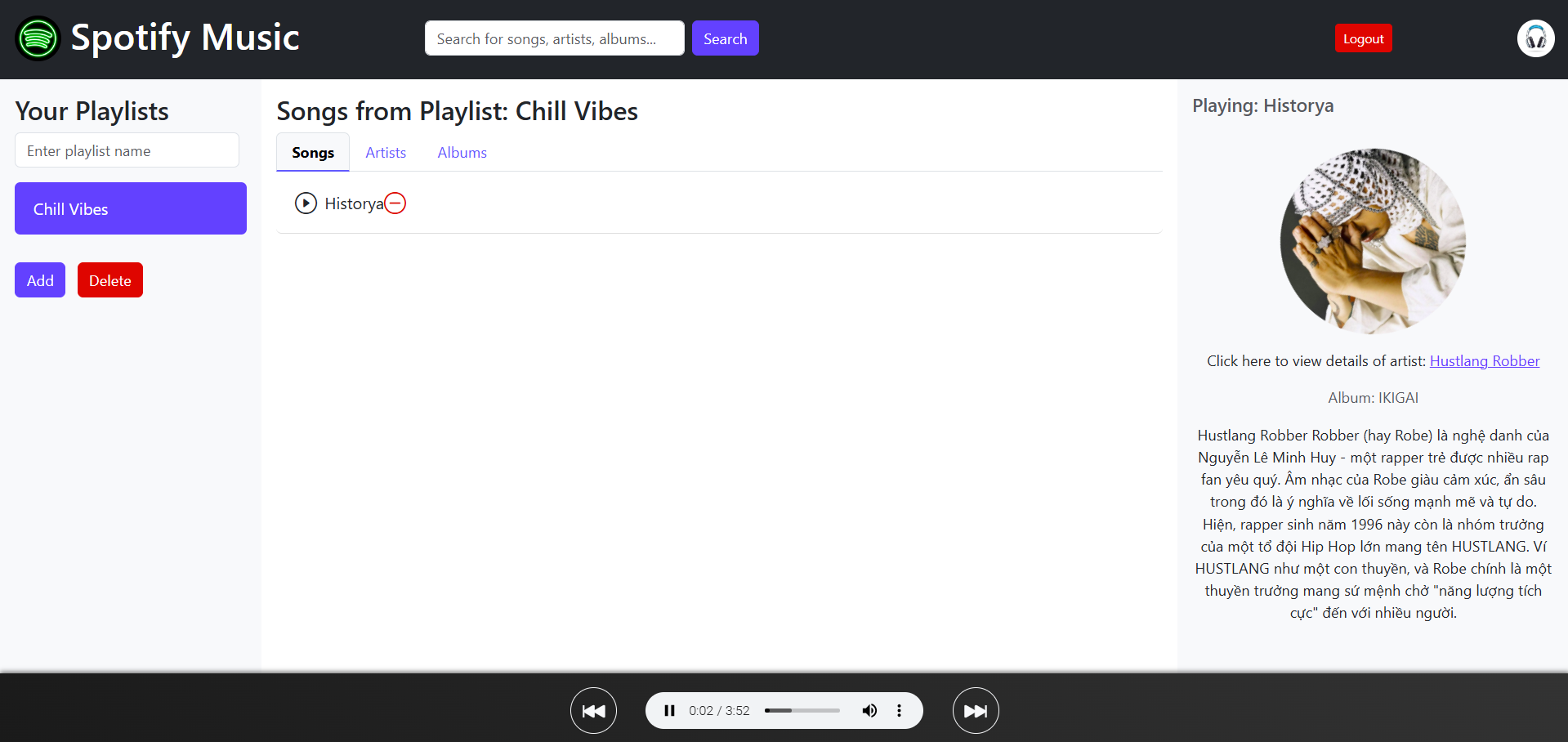
**Login Page**

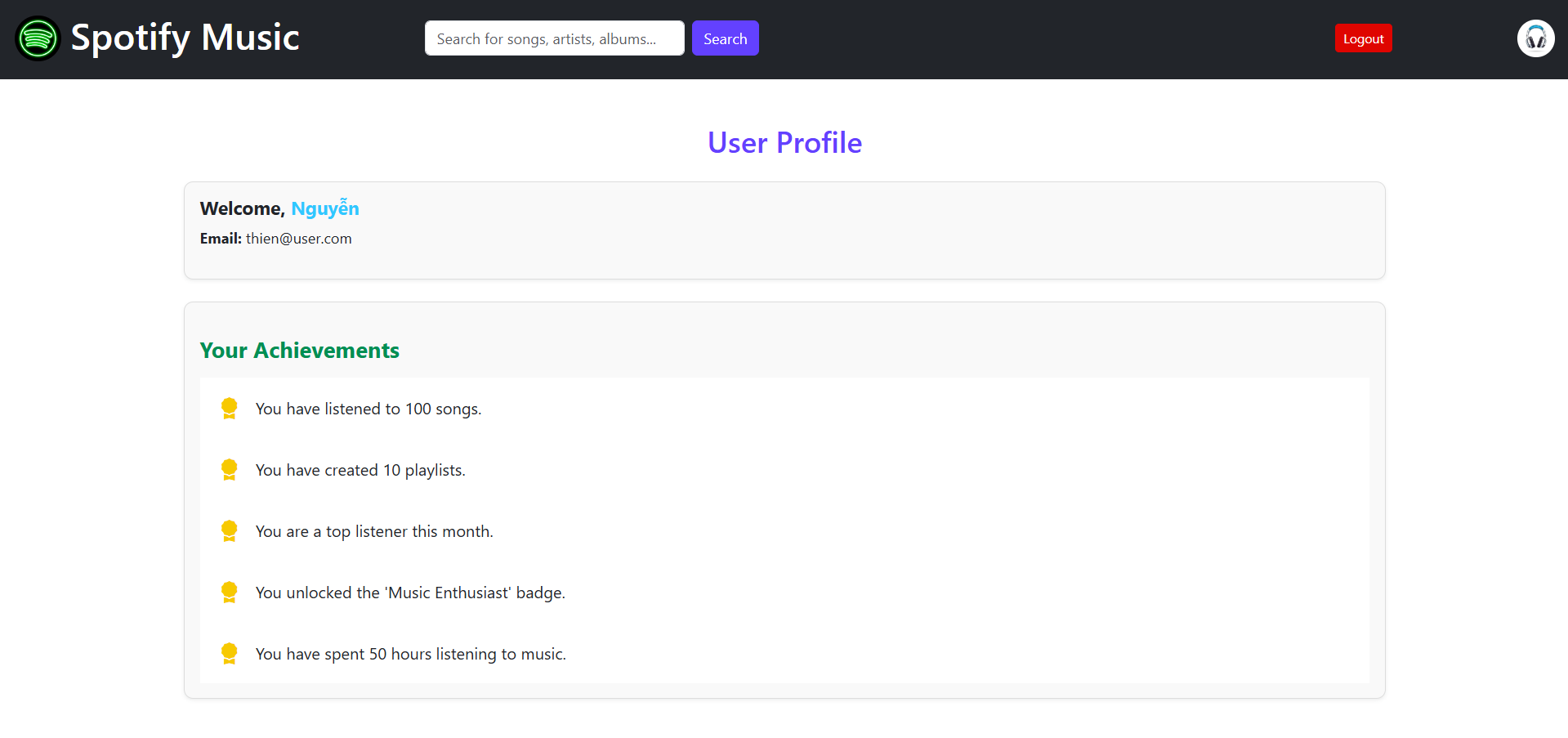
****

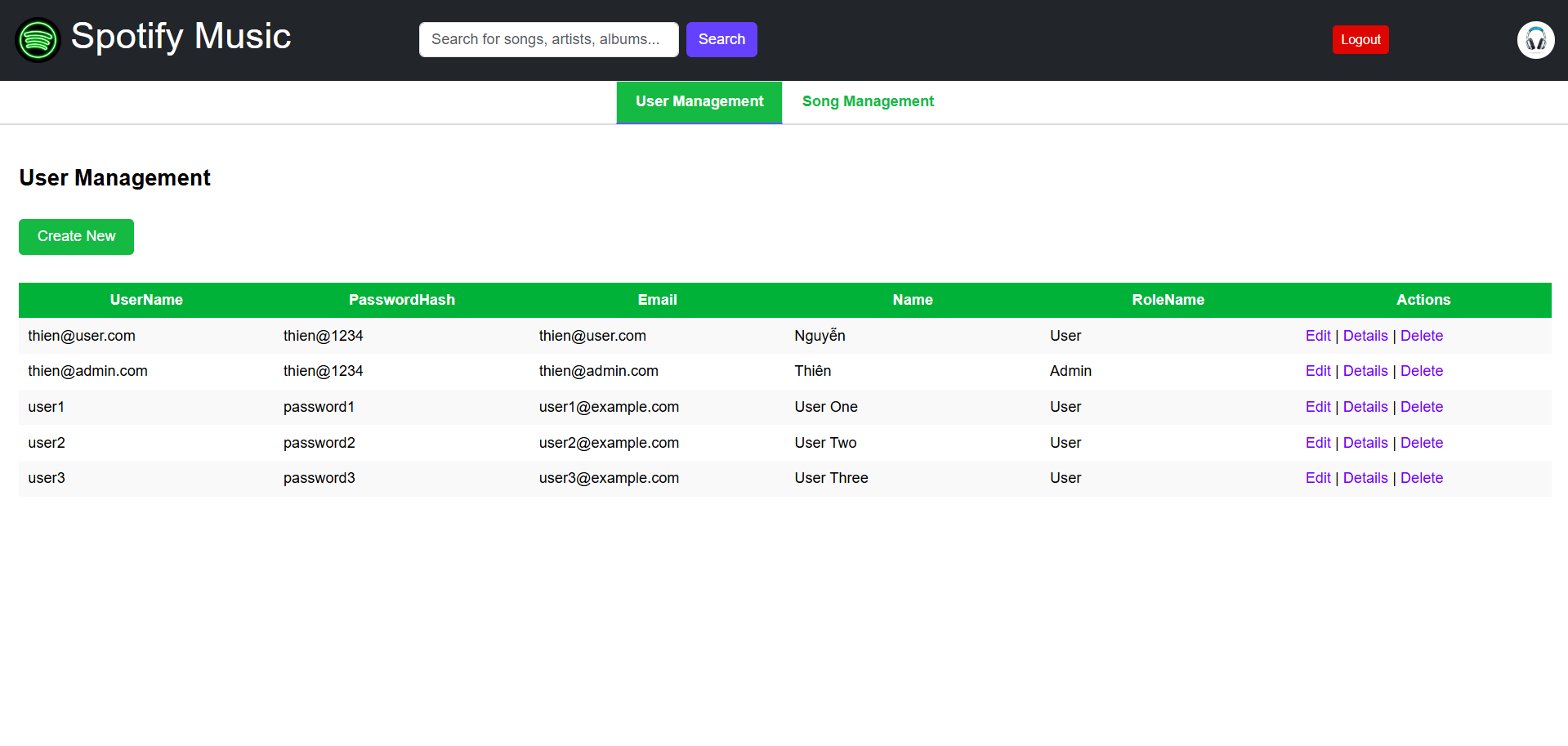
**Home**

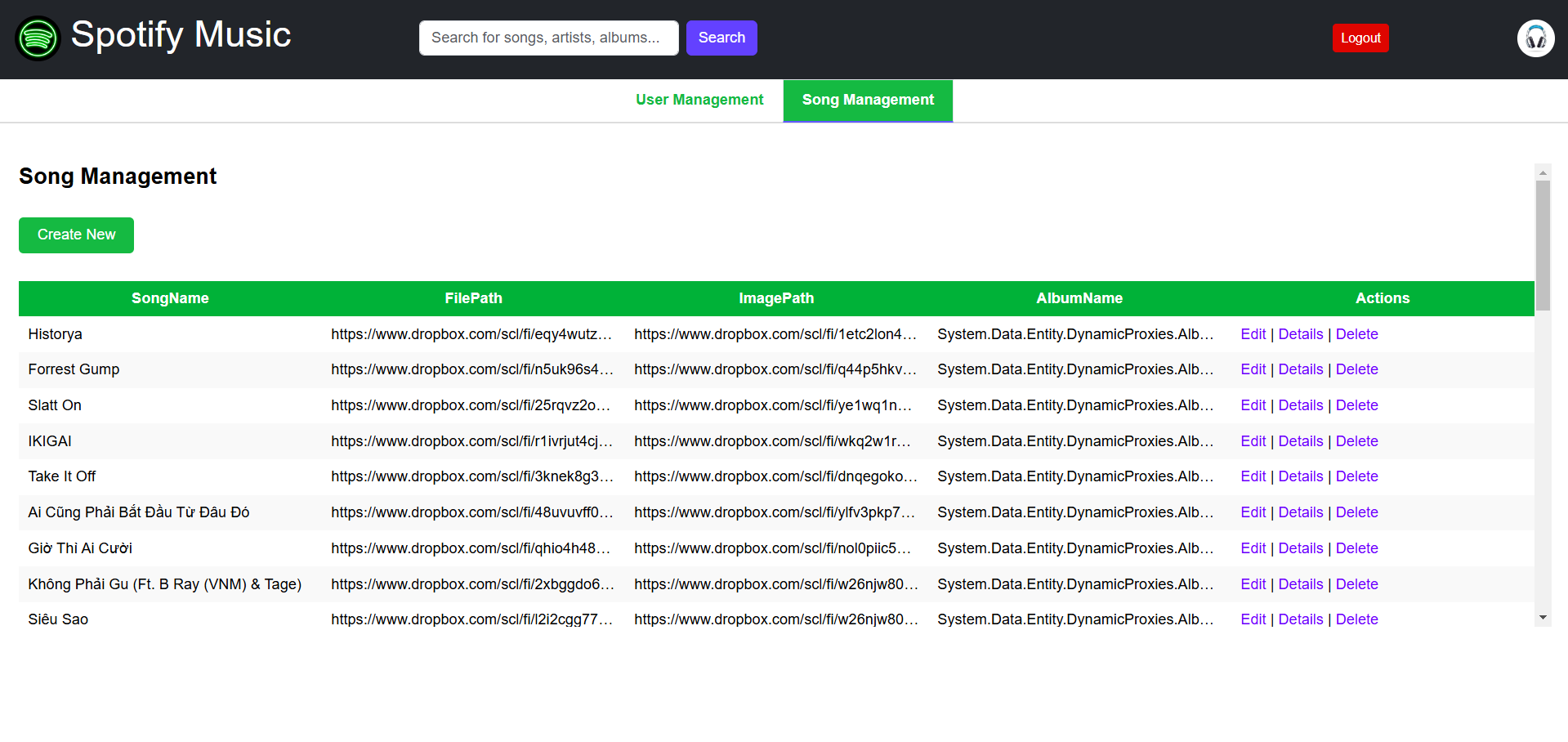
****

**Phát nhạc**

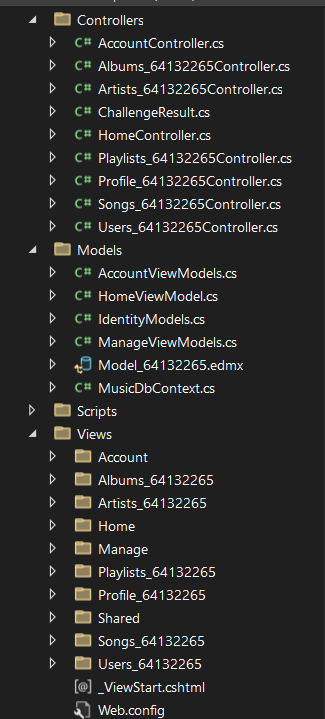
****

**UserPage**

**AdminPage**

****

1. **Cài Đặt**
2. **Cấu trúc source code**

****

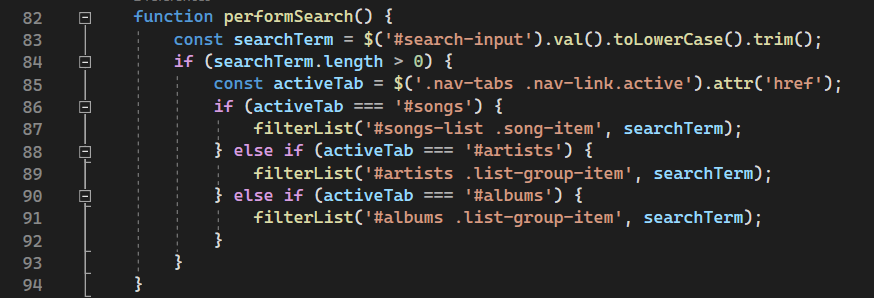
1. **Các thuật toán quan trọng**

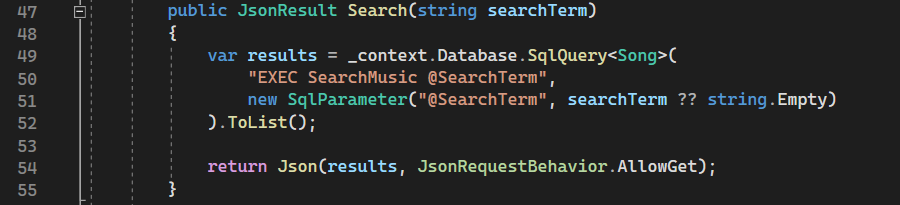
* **Thuật toán phát nhạc từ URL:** 
  + Xử lý sự kiện "Play":
    - Khi người dùng nhấn nút "Play", JavaScript thu thập thông tin bài hát từ DOM (ID, tên, file nhạc, hình ảnh).
  + Cập nhật giao diện:
    - Cập nhật thông tin bài hát đang phát (hình ảnh, tên bài hát, nghệ sĩ) trên giao diện.
  + Phát nhạc:
    - Sử dụng HTML5 Audio Player để phát nhạc từ link trực tiếp.
  + Cập nhật trạng thái trình phát:
    - Hiển thị thời gian phát nhạc (current time, duration).
    - Đồng bộ thanh trượt (slider) để tua bài hát.
* **Thuật toán tìm kiếm bài hát (Search Algorithm):**
  + Nhận từ khóa từ giao diện người dùng:
    - Người dùng nhập từ khóa vào thanh tìm kiếm.
    - JavaScript thu thập giá trị từ input và gửi yêu cầu AJAX tới server.
  + Xử lý trên server:
    - Server nhận từ khóa từ request và thực hiện truy vấn tìm kiếm trong cơ sở dữ liệu.
    - Truy vấn tìm kiếm các bản ghi (bài hát, album, nghệ sĩ) có tên chứa từ khóa.
  + Trả về kết quả:
    - Server trả về danh sách kết quả dưới dạng JSON, bao gồm tên bài hát, album, nghệ sĩ.
  + Hiển thị kết quả trên giao diện:
    - JavaScript nhận kết quả và cập nhật danh sách bài hát phù hợp trên giao diện.
* **Thuật toán thêm bài hát vào Playlist:**
  + Xử lý sự kiện "Add to Playlist":
    - Khi người dùng nhấn vào nút thêm bài hát, JavaScript gửi yêu cầu AJAX tới server, kèm theo ID bài hát và playlist.
  + Xử lý trên server:
    - Server kiểm tra:
      * Playlist có tồn tại không.
      * Bài hát đã có trong playlist chưa.
    - Nếu hợp lệ, thêm bài hát vào playlist trong cơ sở dữ liệu.
  + Trả kết quả:
    - Server trả về trạng thái thành công hoặc lỗi.
    - JavaScript hiển thị thông báo trên giao diện.

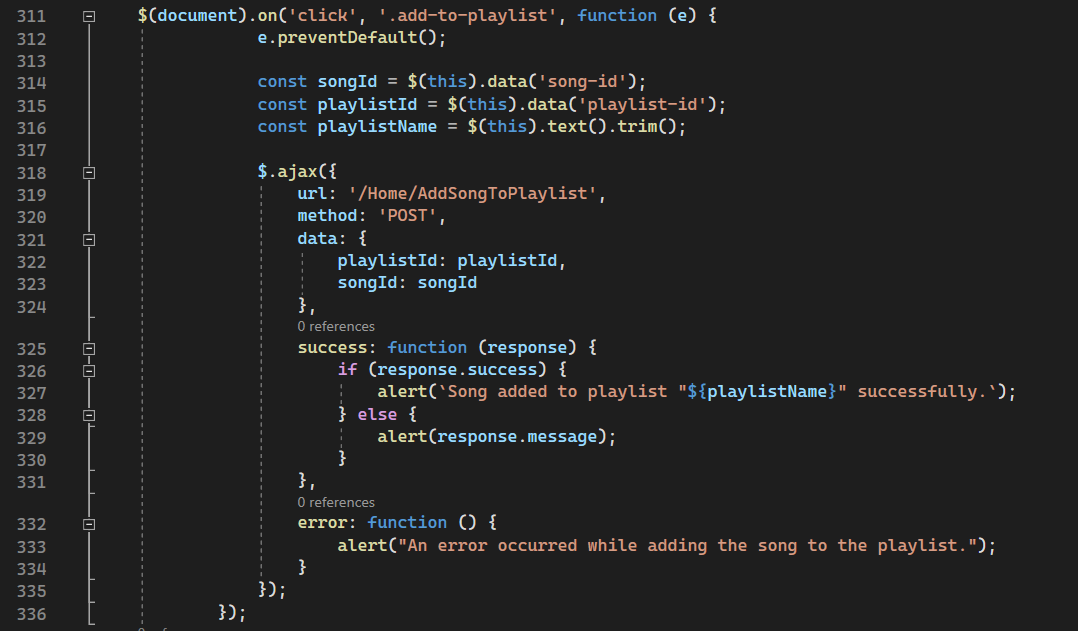
1. **Code mẫu cho các chức năng chính**

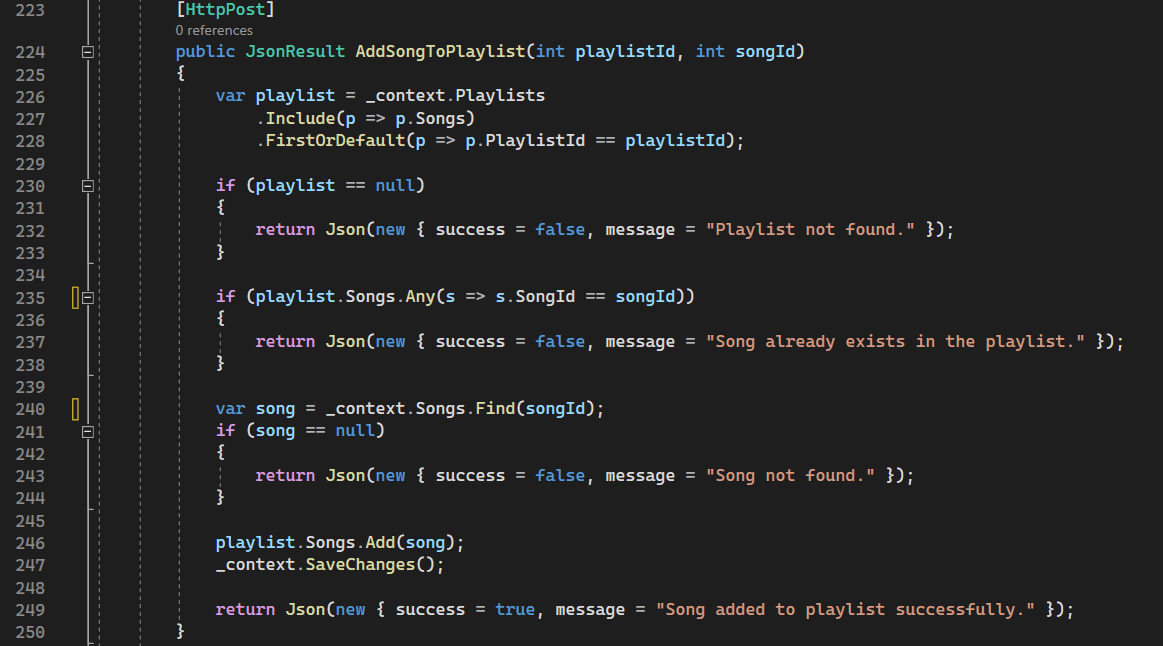
**Chức năng phát nhạc:**



**Chức năng tìm kiếm bài hát:** 



**Chức năng thêm bài hát vào Playlist:**



1. **Kiểm Thử**
2. **Kế hoạch kiểm thử:**

* **Mục tiêu kiểm thử:**
* Đảm bảo rằng các chức năng chính trong hệ thống hoạt động đúng yêu cầu.
* Xác định và sửa các lỗi trong quá trình phát triển.
* Đảm bảo giao diện người dùng hoạt động ổn định và tương thích.
* **Phạm vi kiểm thử:**
* Chức năng User:
  + Đăng nhập và tìm kiếm bài hát, album, nghệ sĩ.
  + Phát nhạc, điều khiển phát nhạc (play/pause, tua).
  + Tạo, xóa playlist.
  + Thêm và xóa bài hát trong playlist.
* Chức năng Admin:
  + Thêm, sửa, xóa bài hát.
  + Quản lý danh sách user.
* Hiệu suất:
  + Tốc độ phản hồi khi tìm kiếm và tải dữ liệu.
  + Tính ổn định khi xử lý dữ liệu lớn.
* **Loại kiểm thử:**
* Kiểm thử chức năng: Đảm bảo các chức năng hoạt động như mong muốn.
* Kiểm thử giao diện: Kiểm tra bố cục, hiển thị của giao diện.
* Kiểm thử hiệu suất: Đánh giá tốc độ tải và xử lý dữ liệu.
* **Công cụ kiểm thử:**
* Thủ công: Thực hiện kiểm tra trực tiếp các chức năng trên giao diện.

1. **Kết quả kiểm thử**

| **STT** | **Chức năng** | **Mô tả** | **KQ mong đợi**   |  | | --- | | **KQ thực tế** | **Trạng thái** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Đăng nhập | Đăng nhập bằng tài khoản hợp lệ | Đăng nhập thành công và chuyển trang | Đăng nhập thành công | Pass |
| 2 | Tìm kiếm bài hát | Tìm kiếm bài hát theo từ khóa | Danh sách bài hát hiển thị đúng | Hiển thị bài hát khớp với từ khóa | Pass |
| 3 | Thêm playlist | Người dùng thêm một playlist mới vào danh sách | Playlist được thêm thành công và hiển thị trong danh sách | Playlist hiển thị đúng, lưu vào database | Pass |
| 4 | Xóa playlist | Xóa một playlist | Playlist bị xóa khỏi cơ sở dữ liệu | Xóa thành công | Pass |
| 5 | Phát nhạc | Người dùng nhấn đúp vào bài hát để phát nhạc | Bài hát được phát qua URL, thông tin hiển thị ở giao diện | Nhạc phát đúng, ảnh hiển thị đúng | Pass |
| 6 | Xem chi tiết bài hát | Người dùng nhấn vào bài hát muốn phát | Hiển thị thông tin nghệ sĩ (tên, bài hát liên quan, ảnh) | Thông tin hiển thị đúng, đầy đủ | Pass |
| 7 | Thêm/sửa/xóa bài hát mới | Admin thêm/sửa/xóa một bài hát mới vào cơ sở dữ liệu | Bài hát được thêm/sửa/xóa thành công | Bài hát hiển thị đúng, lưu vào database | Pass |
| 8 | Hiển thị slider | Slider hiển thị khi bài hát bắt đầu phát | Thanh slider xuất hiện, hiển thị tiến trình bài hát | Thanh slider hiển thị đúng | Pass |
| 9 | Cập nhật slider | Thanh slider tự động cập nhật tiến trình bài hát | Slider cập nhật đúng với thời gian phát nhạc | Slider chưa cập nhật khớp với thời gian bài hát ( khi đến nửa bài thì thanh slider đã chạy xong) | Pass |
| 10 | Logout | Người dùng nhấn nút Logout để thoát khỏi hệ thống | Chuyển về màn hình đăng nhập | Chuyển đúng về màn hình đăng nhập | Pass |

1. **Kết Luận**
2. **Kết quả đạt được**

* Hoàn thành hệ thống quản lý bài hát với mô hình MVC, đảm bảo phân chia rõ ràng các thành phần Model, View, và Controller.
* Hệ thống kết nối ổn định với SQL Server, đảm bảo các thao tác thêm, sửa, xóa, và truy vấn dữ liệu hoạt động chính xác.
* Học tập và làm quen với ASP.NET MVC, Entity Framework, và SQL Server..

1. **Hạn chế và hướng phát triển**

##### **Hạn chế:**

* Chưa tích hợp tính năng upload file nhạc và ảnh trực tiếp từ người dùng.
* Chưa kiểm thử hiệu suất với dữ liệu lớn hoặc tải nặng.

##### **Hướng phát triển:**

* Thêm tính năng upload file và phát triển ứng dụng trên các nền tảng khác như di động.
* Cải thiện giao diện và tối ưu hóa hiệu năng xử lý dữ liệu.

1. **Bài học kinh nghiệm**

* Hiểu và áp dụng mô hình MVC giúp quản lý dự án dễ dàng hơn.
* Làm quen với ASP.NET MVC, Entity Framework, và SQL Server..
* Tăng cường kỹ năng lập kế hoạch, kiểm thử, và xử lý ngoại lệ trong quá trình phát triển.

1. **Tài Liệu Tham Khảo**
2. **ASP.NET MVC Documentation**
3. **Entity Framework Documentation**
4. [**SQL Server Documentation**](https://learn.microsoft.com/en-us/sql/)
5. **Chatgpt.com**