

**TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN**



PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Giảng viên hướng dẫn: Từ Lăng Phiêu

ĐỀ TÀI

Phát triển ứng dụng nghe nhạc Spotify

GVHD: Từ Lăng Phiêu

Nhóm: 08

SV: Phạm Bảo Nghiêm - 3121410346
nghiempham0404@gmail.com
Nguyễn Trung Anh - 3121410056
anhzero816@gmail.com
Nguyễn Minh Kha - 3121410248
kha.nguyenminh167@gmail.com

Năm học: 2024 - 2025
TP.HCM, tháng 5 năm 2025

1. Lời mở đầu

Trong thời đại công nghệ hiện đại, phát triển phần mềm mã nguồn mở không chỉ là một lĩnh vực hứa hẹn mà còn là một cơ hội để cộng đồng phát triển và chia sẻ kiến thức. Bộ môn **Phát triển Phần mềm Mã nguồn Mở** không chỉ giúp chúng ta hiểu rõ hơn về quy trình phát triển phần mềm mà còn khuyến khích tinh thần cộng đồng và sáng tạo.

Trong báo cáo này, chúng em sẽ trình bày về quá trình phát triển phần mềm **Spotify**, một ứng dụng web dựa trên **Django** — một framework phát triển web mã nguồn mở phổ biến. Chúng em sẽ xem xét cách mà việc sử dụng Django đã hỗ trợ quá trình phát triển, cũng như lợi ích và thách thức mà chúng em đã gặp phải trong quá trình này.

Báo cáo sẽ tập trung vào việc trình bày kiến thức và kinh nghiệm mà chúng em thu được từ việc tham gia vào dự án này, cũng như những bài học quý giá mà chúng em hy vọng sẽ chia sẻ với mọi người. Chúng em hy vọng rằng thông tin trong báo cáo này sẽ mang lại giá trị và cung cấp cái nhìn sâu hơn về quá trình phát triển phần mềm mã nguồn mở.

Cuối cùng, chúng em xin cảm ơn thầy **Từ Lăng Phiêu**, thầy đã hỗ trợ chúng em trong khâu kiến thức, cũng như quá trình phát triển dự án. Sự hỗ trợ tận tình của thầy là yếu tố quan trọng giúp dự án được hoàn thành. Một lần nữa xin cảm ơn thầy rất nhiều.

2. Giới thiệu phần mềm

Spotify là một ứng dụng web để phát những bài nhạc bằng Django và ReactJS. Ứng dụng sử dụng các ngôn ngữ và công nghệ sau:

- **Ngôn ngữ:**

- Backend: Python
- Frontend: HTML/CSS/JavaScript

- **Thư viện/Framework:**

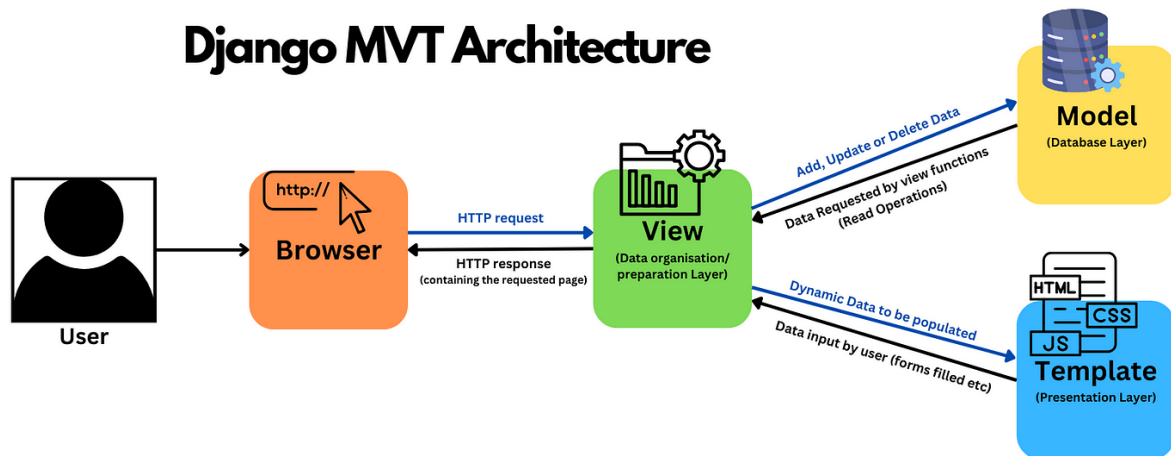
- Backend: Django Rest Framework
- Frontend: ReactJS

Các chức năng chính của ứng dụng:

- Quản lý các album, bài nhạc, nghệ sĩ, tài khoản người dùng.
- Chức năng phát nhạc.
- Chức năng phát video âm nhạc.
- Chức năng tải nhạc.
- Chức năng tạo album và danh sách bài hát.

3. Giới thiệu về Django Framework

Django là một web framework miễn phí, mã nguồn mở dựa trên ngôn ngữ Python và hỗ trợ mô hình kiến trúc **Model-Template-View (MTV)**. Framework này giúp việc phát triển ứng dụng web trở nên nhanh chóng và hiệu quả, đồng thời giải quyết nhiều vấn đề phức tạp trong phát triển web như kết nối cơ sở dữ liệu, quản trị viên với giao diện trực quan, tự động cài đặt và dễ dàng tùy chỉnh.



Hình 1: Kiến trúc của Django Framework

Ưu điểm nổi bật của Django Framework:

- **Nhanh**: Django được thiết kế để giúp các nhà phát triển tạo ứng dụng web từ ý tưởng đến triển khai một cách nhanh chóng.
- **Bảo mật**: Django tích hợp sẵn nhiều middleware giúp xử lý các lỗi bảo mật phổ biến như SQL Injection, Cross-site Scripting (XSS) và Cross-Site Request Forgery (CSRF).
- **Khả năng mở rộng**: Django được sử dụng trong nhiều trang web có lưu lượng truy cập cao như YouTube, Instagram. Framework có khả năng mở rộng quy mô linh hoạt.
- **“Batteries included”**: Django đi kèm với nhiều thư viện hỗ trợ các tác vụ thông thường như xác thực người dùng, quản lý nội dung, RSS feed, v.v.
- **Đa năng**: Có thể ứng dụng Django cho nhiều loại website, từ blog, mạng xã hội đến hệ thống quản lý nội dung, và các hệ thống lớn.
- **Cộng đồng và tài liệu phong phú**: Django có một cộng đồng tích cực và nhiều tài liệu, hướng dẫn, thư viện mở rộng giúp lập trình viên học hỏi và giải quyết vấn đề nhanh chóng.

Nhược điểm của Django Framework:

- **Khả năng tùy chỉnh cao đòi hỏi kiến thức chuyên sâu**: Để điều chỉnh ứng dụng theo nhu cầu riêng, người phát triển cần có kiến thức sâu rộng về Django.

- **Không phù hợp với ứng dụng nhỏ:** Django có thể quá nặng nề cho những ứng dụng đơn giản, nhỏ gọn.
- **Khó cho người mới bắt đầu:** Django yêu cầu kiến thức nền tảng về Python và phát triển web, có thể gây khó khăn cho người mới học lập trình.

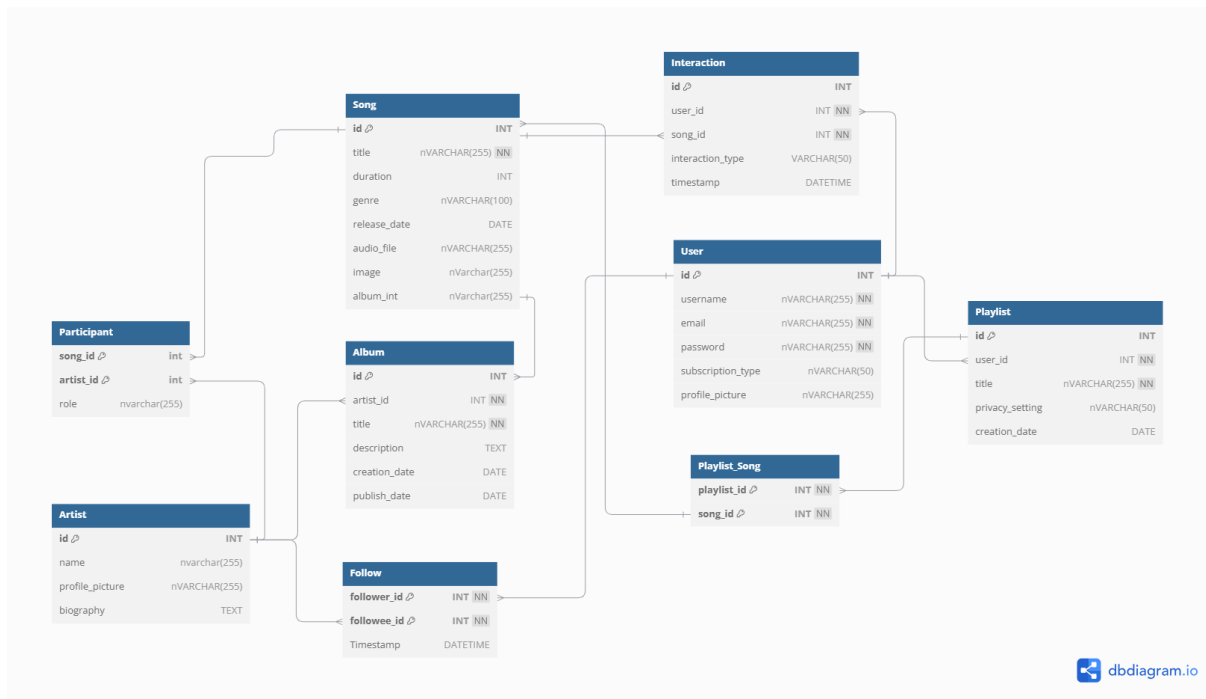
Một số ứng dụng tiêu biểu của Django:

- **Website Doanh nghiệp:** Dùng để xây dựng trang giới thiệu sản phẩm, dịch vụ, và thông tin liên hệ.
- **Blog và hệ thống tin tức:** Hỗ trợ quản lý bài viết, bình luận và phân loại nội dung dễ dàng.
- **Trang web thương mại điện tử:** Có thể phát triển các trang bán hàng trực tuyến với giỏ hàng, quản lý sản phẩm và thanh toán an toàn.

4. Spotify (Clone)

4.1. Thiết kế ứng dụng

4.1.1. Cơ sở dữ liệu



Hình 2: Cơ sở dữ liệu dự án

```

CREATE TABLE 'Song' (
  'id' INT PRIMARY KEY AUTO_INCREMENT,
  'title' nVARCHAR(255) NOT NULL,
  'duration' INT,
  'genre' nVARCHAR(100),
  'release_date' DATE,
  'audio_file' nVARCHAR(255),
  'image' nVarchar(255),
  'album_int' nVarchar(255)
);

CREATE TABLE 'User' (
  'id' INT PRIMARY KEY AUTO_INCREMENT,
  'username' nVARCHAR(255) NOT NULL,
  'email' nVARCHAR(255) NOT NULL,
  'password' nVARCHAR(255) NOT NULL,
  'subscription_type' nVARCHAR(50),
  'profile_picture' nVARCHAR(255)
);

CREATE TABLE 'Playlist' (
  'id' INT PRIMARY KEY AUTO_INCREMENT,
  'user_id' INT NOT NULL,
  'title' nVARCHAR(255) NOT NULL,
  'privacy_setting' nVARCHAR(50),
  'creation_date' DATE
);

CREATE TABLE 'Album' (
  'id' INT PRIMARY KEY AUTO_INCREMENT,
  'artist_id' INT NOT NULL,
  'title' nVARCHAR(255) NOT NULL,
  'description' TEXT,
  'creation_date' DATE,
  'publish_date' DATE
);

CREATE TABLE 'Artist' (
  'id' INT PRIMARY KEY AUTO_INCREMENT,
  'name' nvarchar(255),
  'profile_picture' nVARCHAR(255),
  'biography' TEXT
);

CREATE TABLE 'Participant' (
  'song_id' int,
  'artist_id' int,
  'role' nvarchar(255),
  PRIMARY KEY ('song_id', 'artist_id')
);

CREATE TABLE 'Interaction' (
  'id' INT PRIMARY KEY AUTO_INCREMENT,
  'user_id' INT NOT NULL,
  'song_id' INT NOT NULL,
  'interaction_type' VARCHAR(50),
  'timestamp' DATETIME

```

```

);

CREATE TABLE 'Follow' (
    'follower_id' INT NOT NULL,
    'followee_id' INT NOT NULL,
    'Timestamp' DATETIME,
    PRIMARY KEY ('follower_id', 'followee_id')
);

CREATE TABLE 'Playlist_Song' (
    'playlist_id' INT NOT NULL,
    'song_id' INT NOT NULL,
    PRIMARY KEY ('playlist_id', 'song_id')
);

ALTER TABLE 'Playlist' ADD FOREIGN KEY ('user_id') REFERENCES 'User'
    ('id');

ALTER TABLE 'Interaction' ADD FOREIGN KEY ('song_id') REFERENCES
    'Song' ('id');

ALTER TABLE 'Playlist_Song' ADD FOREIGN KEY ('playlist_id') REFERENCES
    'Playlist' ('id');

ALTER TABLE 'Album' ADD FOREIGN KEY ('artist_id') REFERENCES 'Artist'
    ('id');

ALTER TABLE 'Participant' ADD FOREIGN KEY ('artist_id') REFERENCES
    'Artist' ('id');

ALTER TABLE 'Participant' ADD FOREIGN KEY ('song_id') REFERENCES
    'Song' ('id');

ALTER TABLE 'Follow' ADD FOREIGN KEY ('followee_id') REFERENCES
    'Artist' ('id');

ALTER TABLE 'Follow' ADD FOREIGN KEY ('follower_id') REFERENCES 'User'
    ('id');

ALTER TABLE 'Interaction' ADD FOREIGN KEY ('user_id') REFERENCES
    'User' ('id');

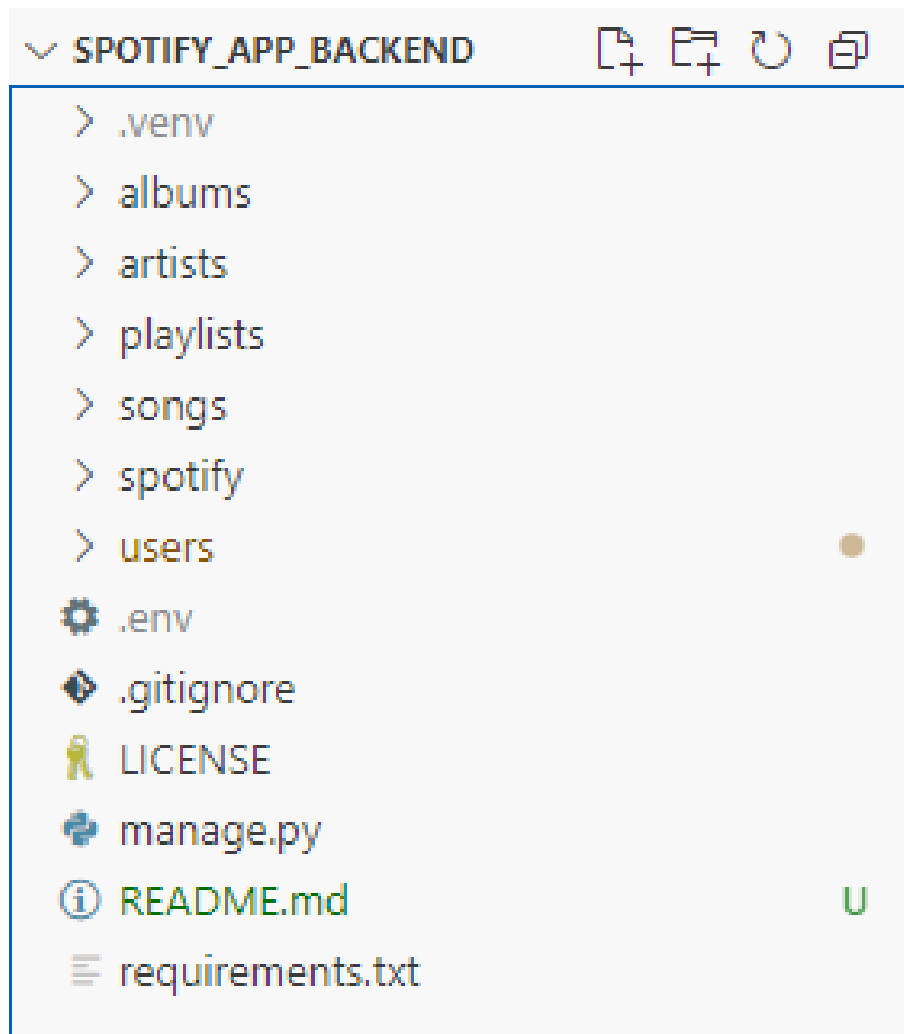
ALTER TABLE 'Album' ADD FOREIGN KEY ('id') REFERENCES 'Song'
    ('album_int');

ALTER TABLE 'Song' ADD FOREIGN KEY ('id') REFERENCES 'Playlist_Song'
    ('song_id');

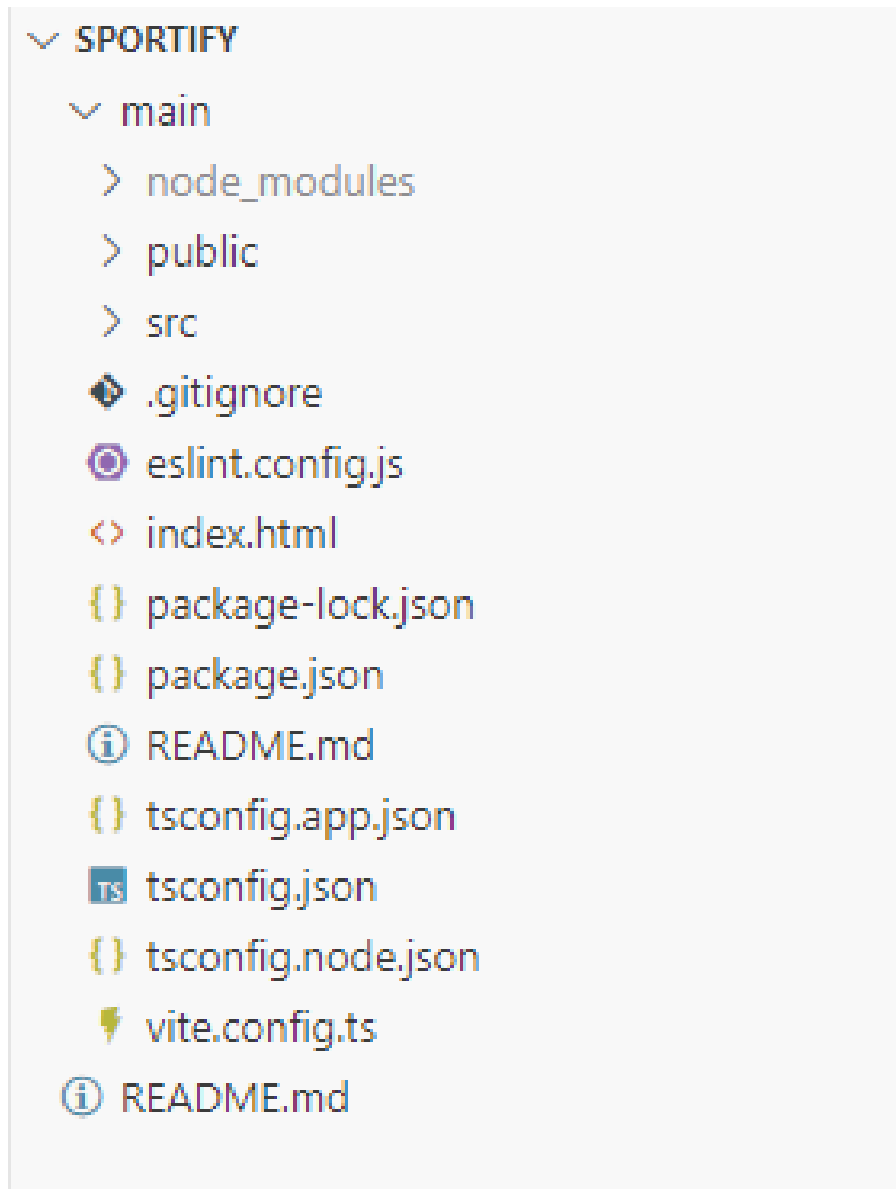
```

Listing 1: Câu lệnh SQL tạo CSDL

4.1.2. Cấu trúc mã nguồn



Hình 3: Cây thư mục mã nguồn của ứng dụng ở backend



Hình 4: Cây thư mục mã nguồn của ứng dụng ở frontend

4.2. Hướng dẫn sử dụng ứng dụng

4.2.1. Cài đặt ứng dụng

Yêu cầu tối thiểu:

- Python phiên bản 3.11.2 trở lên.
- Đã cài đặt WSL.
- MySQL.

Các bước cài đặt:

1. Đảm bảo thỏa mãn các yêu cầu tối thiểu.
2. Clone dự án về máy:

```
git clone https://github.com/nghiempham0404/spotify_app_backend.git
cd spotify_app_backend.git
```

3. Tải về các thư viện cần thiết:

```
pip install -r requirements.txt
```

4. Chuyển đổi dữ liệu:

```
python manage.py migrate
```

5. Khởi tạo superuser:

```
python manage.py createsuperuser
```

6. Chạy dự án:

```
python manage.py runserver
```

7. Truy cập trình duyệt tại <http://127.0.0.1:8000>

4.2.2 Sử dụng ứng dụng





4.2.2.1 Chức năng quản lý của Admin

Trang quản trị được thiết kế nhằm cung cấp toàn bộ quyền kiểm soát hệ thống cho người quản trị (Admin). Giao diện quản trị trực quan và thân thiện, cho phép thực hiện các thao tác quản lý nhanh chóng và chính xác. Cụ thể, các chức năng chính mà Admin có thể thực hiện bao gồm:

- **Quản lý người dùng (User):**

- Xem danh sách tất cả người dùng đã đăng ký trong hệ thống.
- Tìm kiếm, lọc người dùng theo tên, email, vai trò,...
- Thêm mới tài khoản người dùng (nếu cần).
- Chỉnh sửa thông tin cá nhân như họ tên, email, mật khẩu, ảnh đại diện.
- Phân quyền hoặc thu hồi quyền truy cập (ví dụ: người dùng thường, ca sĩ, hoặc admin phụ).
- Khóa/mở khóa tài khoản trong trường hợp vi phạm hoặc cần quản lý bảo mật.


ADMIN

Song	20	KhaNguyen@gmail.com	Kha	Minh		0	—	Edit Delete
User	21	KhanhLinh@gmail.com	—	Linh		0	—	Edit Delete
Participant	22	Ly@gmail.com	Ly	Ly		0	—	Edit Delete
Artist	23	QuangHung@gmail.com	Quang	Hung		0	—	Edit Delete
Album								
Log out								

Hình 5: Trang quản lý người dùng

- Quản lý bài hát:

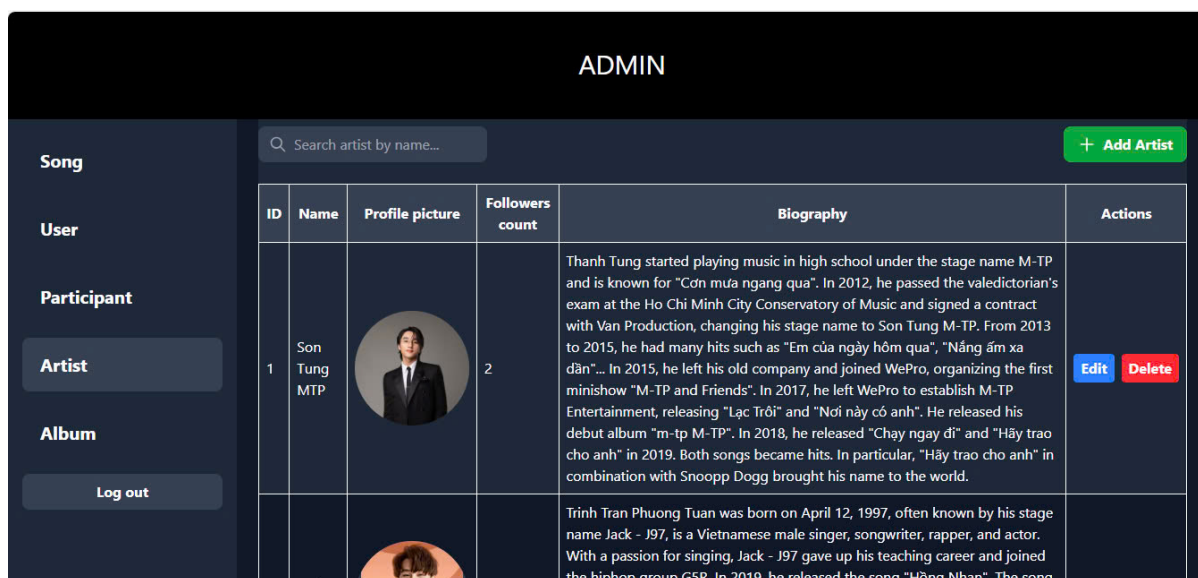
- Thêm bài hát mới vào hệ thống bao gồm: tên bài hát, ảnh đại diện, lời bài hát, tệp âm thanh,...
- Cập nhật thông tin bài hát: sửa đổi tiêu đề, thay đổi file nhạc, cập nhật lời nhạc,...
- Gán bài hát với ca sĩ, thể loại, album cụ thể.
- Xóa bài hát khi không còn phù hợp hoặc có yêu cầu từ tác giả.
- Hiển thị thống kê lượt nghe, lượt yêu thích, và bình luận của từng bài hát.

ADMIN								
Song User Participant Artist Album Log out	<input type="text" value="Name song..."/> + Add Song							
	Add New Song							
	<input type="text" value="Title"/>			<input type="text" value="Chọn album"/>				
	<input type="text" value="Duration"/>			<input type="text" value="Gener"/>				
	<input type="text" value="Image Path"/>			<input type="text" value="Audio File Path"/>				
Cancel Save								
	ID	Image	Title	Duration	Gener	Participant	Album	Actions
	2		Bạc Phận	4:00		—	1	Edit Delete

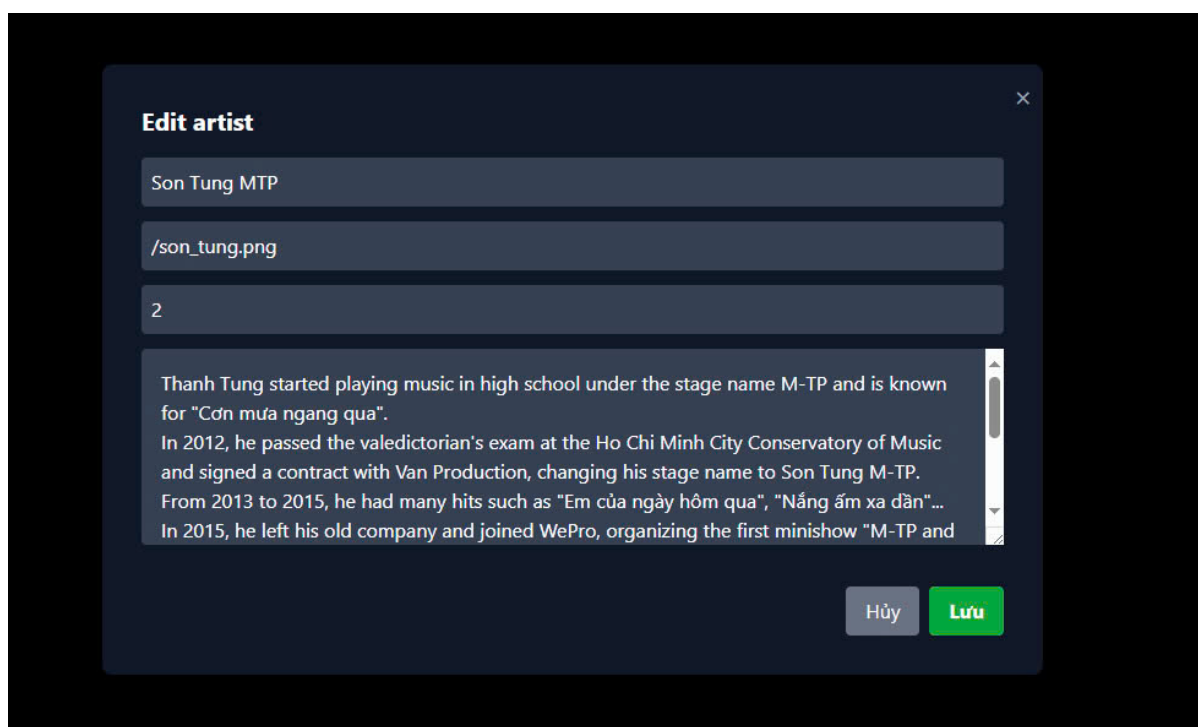
Hình 6: Trang quản lý bài hát

- Quản lý ca sĩ:

- Thêm mới ca sĩ: bao gồm tên nghệ danh, ảnh đại diện, tiểu sử,...
- Chỉnh sửa thông tin ca sĩ: cập nhật ảnh, thông tin tiểu sử,...
- Liên kết ca sĩ với các bài hát hoặc album họ tham gia.
- Xóa ca sĩ khỏi hệ thống trong trường hợp không còn hoạt động hoặc theo yêu cầu.



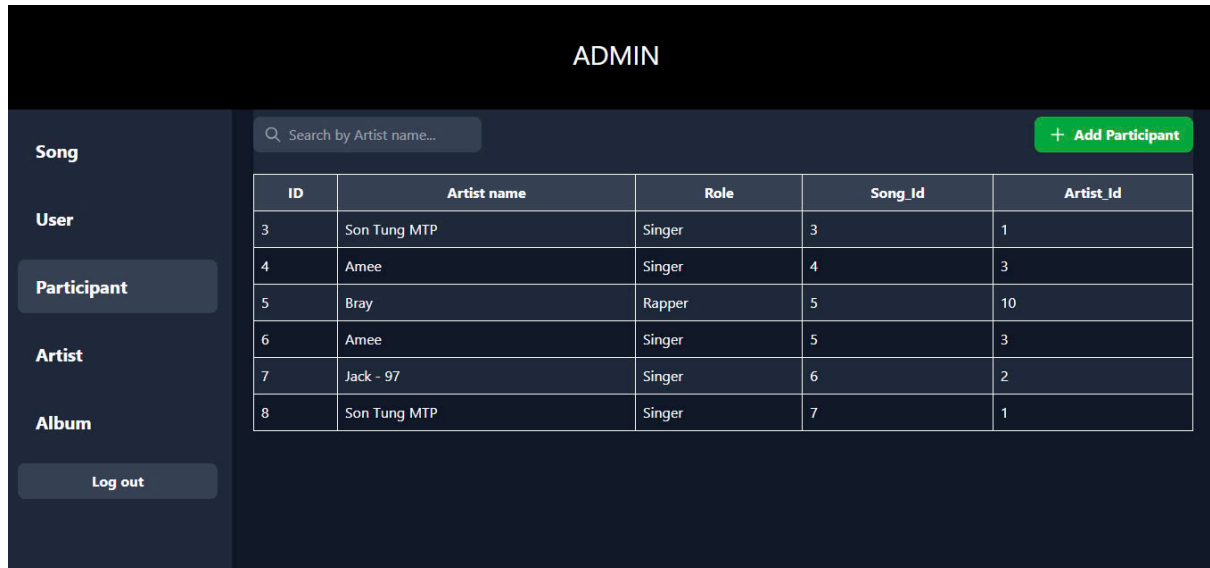
Hình 7: Trang quản lý ca sĩ



Hình 8: Trang chi tiết ca sĩ

- Quản lý album:

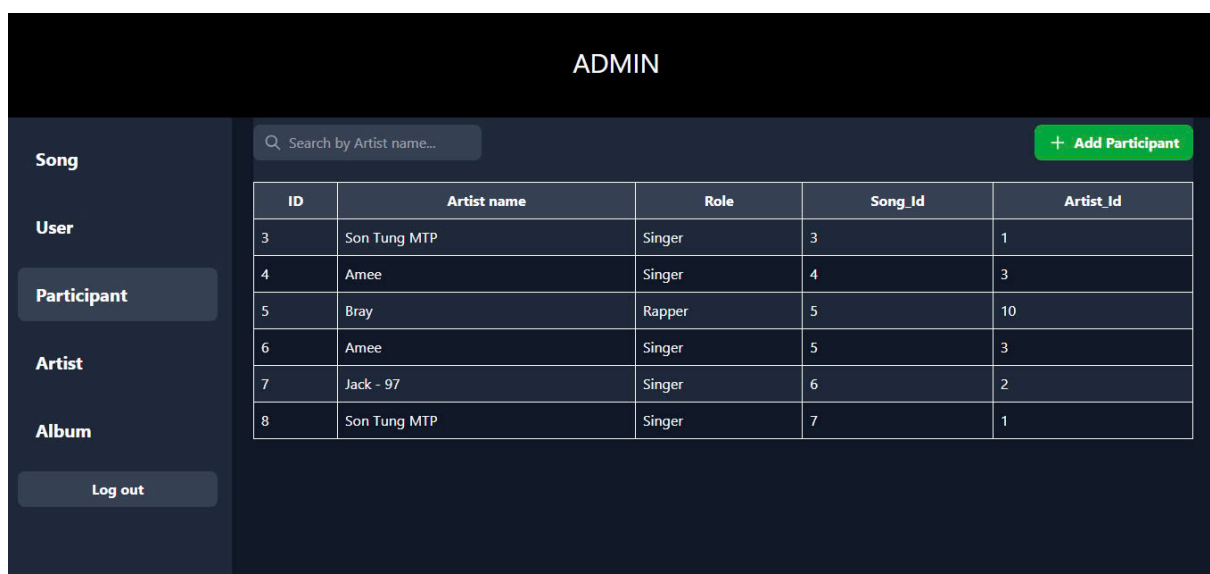
- Tạo album mới: đặt tên album, chọn ảnh bìa, thêm mô tả, ngày phát hành,...
- Thêm bài hát vào album cụ thể.
- Sắp xếp thứ tự các bài hát trong album.
- Chỉnh sửa thông tin album hoặc xóa album khi cần thiết.



Hình 9: Trang quản album

• Quản lý chi tiết tham gia bài hát:

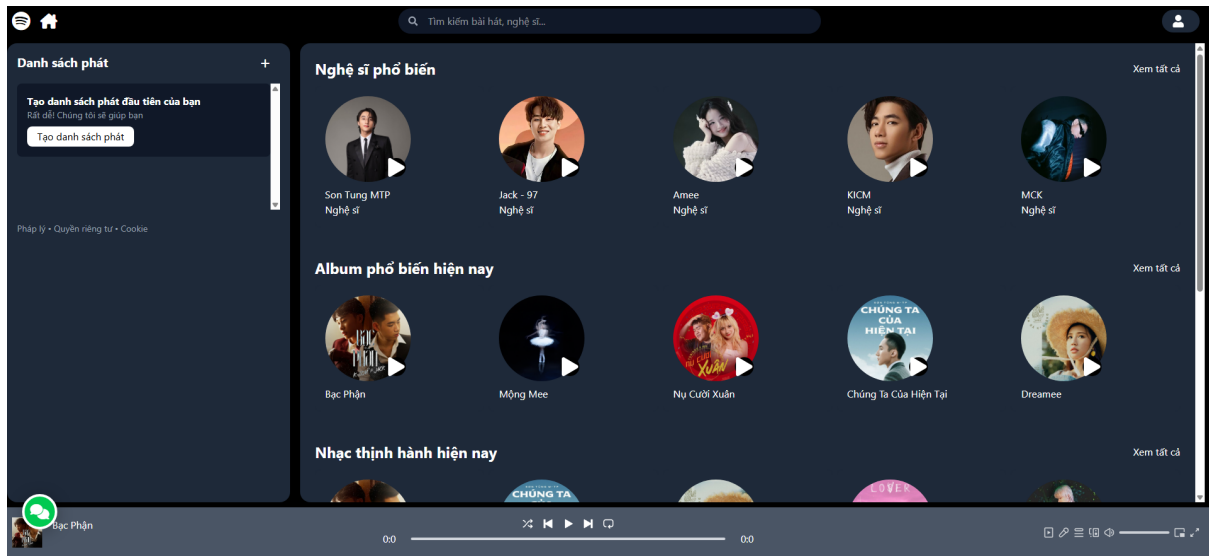
- Thiết lập mối quan hệ giữa bài hát và các ca sĩ tham gia.
- Xác định vai trò của từng ca sĩ trong bài hát: hát chính, hát phụ, sáng tác, phối khí,...
- Cho phép chỉnh sửa hoặc xóa chi tiết tham gia nếu có thay đổi về thông tin nghệ sĩ.



Hình 10: Trang quản lý chi tiết tham gia bài hát

Thông qua các chức năng trên, admin có thể đảm bảo việc duy trì nội dung của hệ thống luôn cập nhật, chính xác và đáng tin cậy. Đồng thời, hệ thống còn hỗ trợ công cụ lọc, tìm kiếm, phân trang để giúp quá trình quản lý được thuận tiện và hiệu quả hơn.

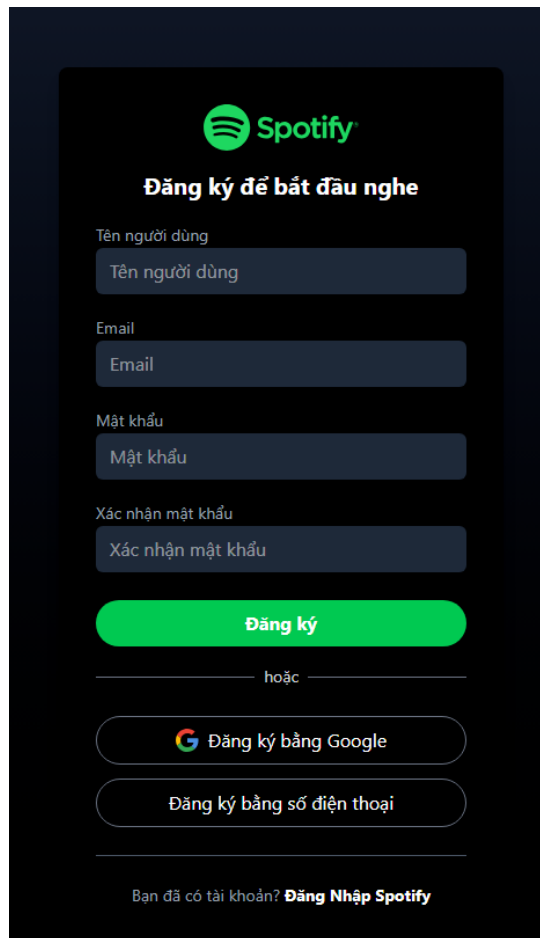
4.2.2.2 Chức năng quản lý của User



Hình 11: Trang chủ mặc định của user

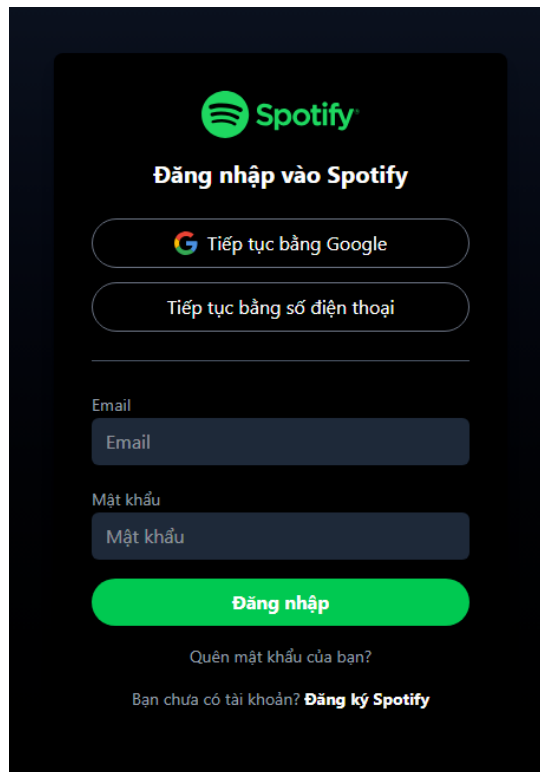
Hệ thống cung cấp các chức năng cơ bản và nâng cao nhằm hỗ trợ người dùng có thể trải nghiệm âm nhạc một cách thuận tiện và cá nhân hóa. Cụ thể, các chức năng chính dành cho người dùng bao gồm:

- Đăng ký, đăng nhập:

The image shows the Spotify registration page. At the top is the Spotify logo. Below it is the heading "Đăng ký để bắt đầu nghe". The form contains four input fields: "Tên người dùng" (Username), "Email", "Mật khẩu" (Password), and "Xác nhận mật khẩu" (Confirm password). Each field has a placeholder text matching the label. Below the fields is a large green "Đăng ký" (Sign up) button. Underneath the button is a horizontal line with the word "hoặc" (or) in the center. Below this line are two buttons: "Đăng ký bằng Google" (Sign up with Google) and "Đăng ký bằng số điện thoại" (Sign up with phone number). At the bottom of the form is a link that says "Bạn đã có tài khoản? Đăng Nhập Spotify" (Do you have an account? Log in Spotify).

Hình 12: Trang đăng ký user

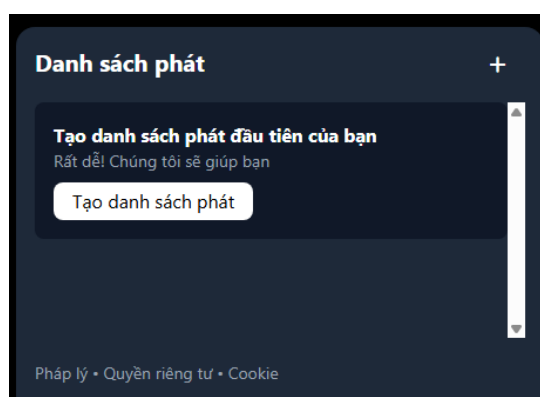
- Cho phép người dùng đăng ký tài khoản mới bằng email, tên người dùng và mật khẩu.
- Thực hiện đăng nhập vào hệ thống để truy cập các chức năng cá nhân.
- Hệ thống hỗ trợ xác thực với JWT và bảo vệ mật khẩu bằng cơ chế mã hóa (hashpassword).
- Thông báo lỗi hoặc yêu cầu nhập lại thông tin nếu đăng nhập thất bại.

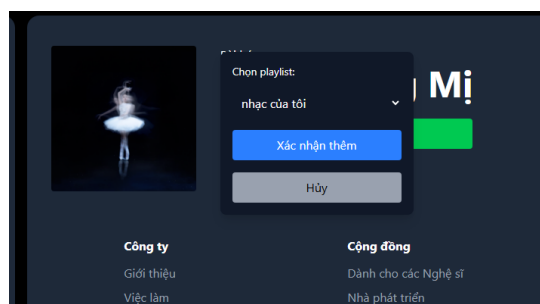
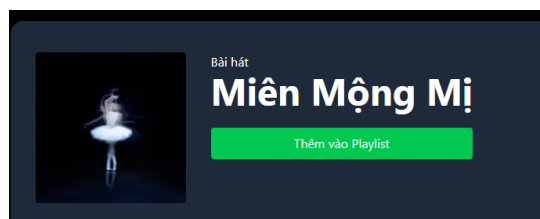
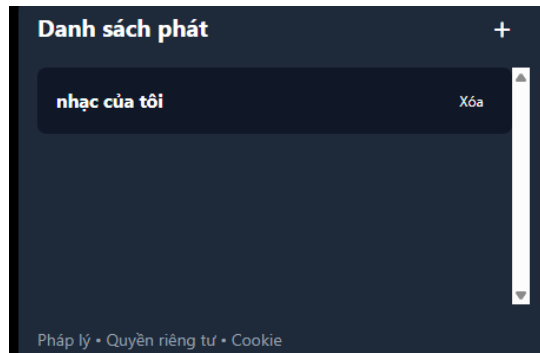
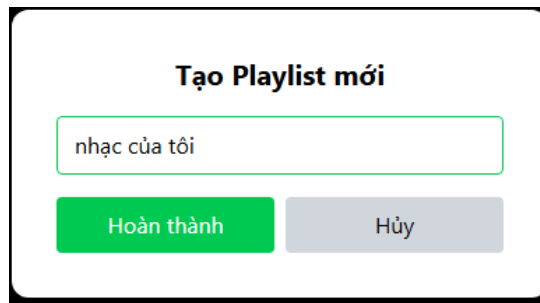


Hình 13: Trang đăng nhập user

- **Playlist cá nhân:**

- Người dùng có thể tạo nhiều playlist theo sở thích cá nhân.
- Thêm hoặc xóa bài hát trong playlist tùy ý.
- Xóa playlist khi không cần thiết.
- Nghe lại các playlist đã tạo bất kỳ lúc nào.

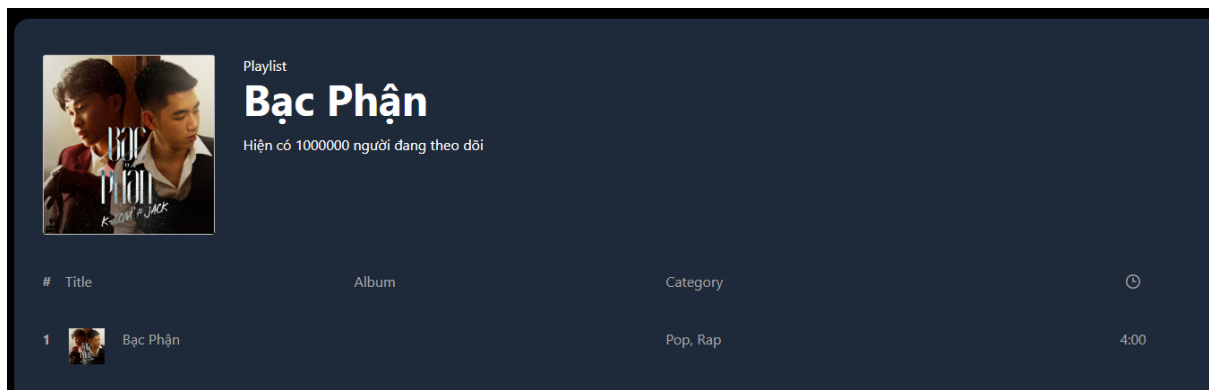
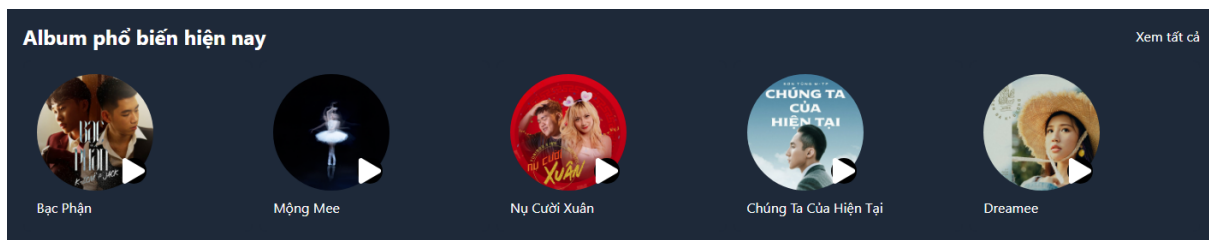




Hình 14: Giao diện người dùng thao tác với playlist

• Xem album:

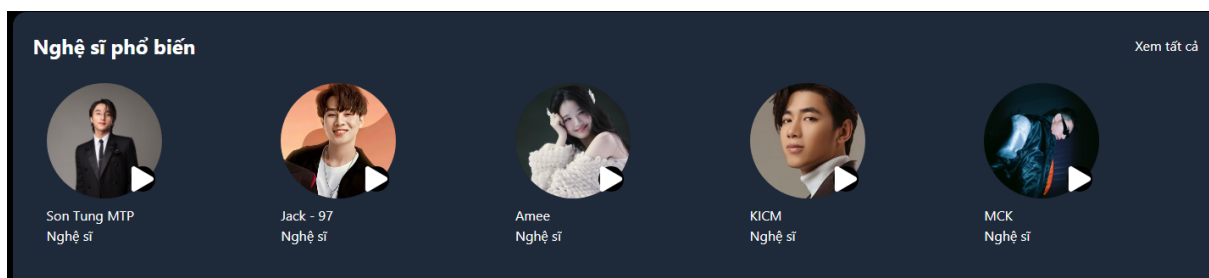
- Xem danh sách các album âm nhạc đang có trên hệ thống.
- Truy cập thông tin chi tiết album: tên album, ca sĩ trình bày, ngày phát hành, danh sách bài hát.
- Nghe toàn bộ album hoặc chọn bài hát bất kỳ trong album.

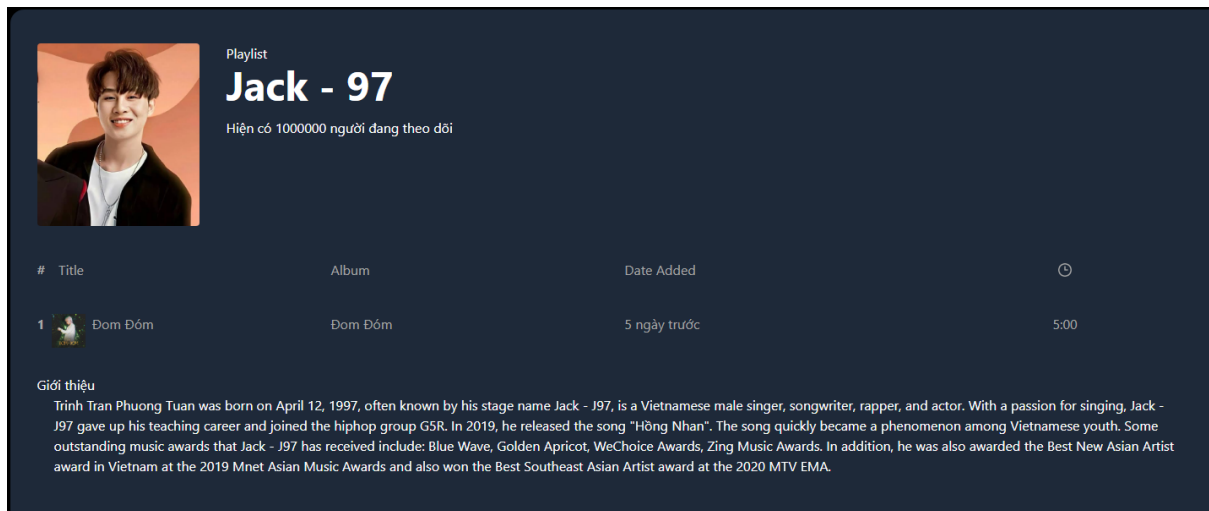


Hình 15: Giao diện người dùng thao tác với album

- **Xem thông tin nghệ sĩ:**

- Truy cập trang cá nhân của nghệ sĩ để xem tiểu sử, ảnh đại diện, và các bài hát liên quan.
- Theo dõi các nghệ sĩ yêu thích (nếu có hỗ trợ tính năng).
- Xem toàn bộ album hoặc bài hát của nghệ sĩ đó.





Hình 16: Giao diện người dùng thao tác với ca sĩ/nghệ sĩ

- **Trình phát nhạc:**

- Phát bài nhạc với giao diện thân thiện, trực quan.
- Hỗ trợ các chức năng như: phát, tạm dừng, tua, phát lại, phát ngẫu nhiên.



Hình 17: Giao diện người dùng thao tác phát nhạc

5.Đóng góp thành viên

Phạm Bảo Nghiêm

- Thiết kế cơ sở dữ liệu.
- Quyết định các chức năng.
- Thiết kế và xây dựng các api endpoint cho backend.
- Tiến hành triển khai dự án trên máy chủ AWS.
- Viết toàn bộ báo cáo dự án.

Nguyễn Trung Anh

- Xây dựng front end phía user.
- Xây dựng các tính năng xác thực đăng ký, đăng nhập.
- Xây dựng tính năng xem bài hát, xem album, xem nghệ sĩ.
- Xây dựng tính năng phát nhạc, tạo playlist, phát playlist.

Nguyễn Minh Kha

- Xây dựng front end phía admin.
- Xây dựng toàn bộ các chức năng quản lý bài hát, chi tiết tham gia bài hát,album, ca sĩ, user
- Tìm kiếm các file mp3, hình ảnh album, ca sĩ.

6. Tài liệu tham khảo

- 1 W3School Python Tutorial, <https://www.w3schools.com/python/>, lần truy cập cuối: 5/3/2025.
- 2 Python Documentation, <https://www.python.org/doc/>, lần truy cập cuối: 4/4/2025.
- 3 Django Documentation, <https://docs.djangoproject.com/en/5.0/>, lần truy cập cuối: 5/5/2025.
- 4 Django Documentation, <https://www.w3schools.com/react/default.asp>, lần truy cập cuối: 5/5/2025.

7. Lời cảm ơn

Trong quá trình thực hiện báo cáo, chúng em đã có cơ hội tìm hiểu sâu sắc hơn về quy trình phát triển phần mềm mã nguồn mở – từ những bước khởi đầu cơ bản cho đến các giai đoạn phức tạp, đòi hỏi nhiều nỗ lực và kỹ năng.

Qua đó, chúng em nhận thấy rằng phần mềm mã nguồn mở mang lại nhiều lợi ích thiết thực cho cả cá nhân lẫn doanh nghiệp. Không chỉ giúp tiết kiệm chi phí, mã nguồn mở còn có tính linh hoạt cao, cho phép người dùng dễ dàng tùy chỉnh và điều chỉnh theo nhu cầu riêng. Bên cạnh đó, nhờ vào cộng đồng phát triển rộng lớn, các phần mềm mã nguồn mở thường được phát hiện lỗi và cải tiến nhanh chóng, từ đó nâng cao chất lượng và độ tin cậy của sản phẩm.

Chúng em tin rằng những kiến thức và kinh nghiệm tích lũy được trong quá trình thực hiện báo cáo sẽ là nền tảng vững chắc để chúng em tiếp tục học hỏi và phát triển trong lĩnh vực công nghệ thông tin sau này.

Cuối cùng, chúng em xin bày tỏ lòng biết ơn chân thành đến thầy Từ Lăng Phiêu vì sự hướng dẫn tận tình, sự hỗ trợ quý báu và những điều kiện thuận lợi mà thầy đã tạo ra để chúng em có thể hoàn thành tốt báo cáo này.