

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO
PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

SPOTIFY CLONE

Giảng viên hướng dẫn: Từ Lăng Phiêu

Nhóm sinh viên thực hiện:

3121410317 – Lành Kửng Mẫn

3119410290 – Nguyễn Hữu Nhơn

3121410051 – Ngô Trí Anh

Thành phố Hồ Chí Minh, tháng 5 năm 2025

LỜI CẢM ƠN

Trước hết, chúng em xin gửi lời cảm ơn chân thành đến các thầy cô trong khoa Công nghệ thông tin của trường Đại học Sài Gòn. Trong suốt thời gian qua, sự hướng dẫn và chia sẻ kiến thức từ các thầy cô đã giúp chúng em tích lũy được những kiến thức và kỹ năng cơ bản, tạo nền tảng vững chắc cho sự nghiệp tương lai của chúng em. Đặc biệt, chúng em muốn bày tỏ lòng biết ơn sâu sắc đến thầy Từ Lăng Phiêu, người đã dành thời gian và tâm huyết để hướng dẫn nhóm chúng em hoàn thành dự án ngành. Sự quan tâm và hỗ trợ của thầy đã giúp chúng em tiến bộ và tự tin hơn trong quá trình nghiên cứu và thực hiện dự án.

Chúng em cũng xin gửi lời cảm ơn sâu sắc đến gia đình và bạn bè, những người đã luôn ủng hộ và chia sẻ cùng chúng em trong suốt thời gian học tập và nghiên cứu. Đặc biệt, chúng em muốn cảm ơn các thành viên trong nhóm, những người đã cùng nhau làm việc chăm chỉ và đóng góp ý kiến quý báu để chúng em hoàn thành dự án này một cách thành công. Trong quá trình thực hiện dự án, chúng em đã gặp phải nhiều khó khăn và thách thức, nhưng nhờ sự học hỏi và nỗ lực, chúng em đã vượt qua được nhiều trở ngại. Chúng em xin trân trọng mọi ý kiến đóng góp và nhận xét từ các thầy cô để có thể học hỏi và phát triển kiến thức của mình, áp dụng chúng vào thực tế một cách hiệu quả trong tương lai.

Nhóm chúng em xin chân thành cảm ơn!

MỤC LỤC

LỜI CẢM ƠN.....	1
MỤC LỤC.....	2
CHƯƠNG 1: GIỚI THIỆU.....	3
1. Lý do chọn đề tài.....	3
2. Mục tiêu dự án.....	3
3. Phạm vi dự án.....	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	4
1. Spotify là gì?.....	4
2. Spotify Clone là gì?.....	5
3. Các công nghệ sử dụng trong dự án.....	5
CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG.....	6
1. Use Case Diagram.....	6
2. Mô tả các chức năng chính.....	7
3. Biểu đồ ERD (Entity Relationship Diagram).....	8
CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI.....	8
1. Môi trường cài đặt.....	8
2. Các bước triển khai.....	9
3. Một số code quan trọng:.....	10
CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC.....	10
1. Các chức năng đã hoàn thiện.....	10
2. Giao diện thực tế (Frontend).....	11
CHƯƠNG 6: ĐÁNH GIÁ – KẾT LUẬN – HƯỚNG PHÁT TRIỂN.....	12
1. Tóm tắt kết quả đạt được.....	12
2. Hạn chế và khó khăn.....	14
3. Hướng phát triển:.....	15

CHƯƠNG 1: GIỚI THIỆU

1. Lý do chọn đề tài

Âm nhạc đóng vai trò quan trọng trong đời sống tinh thần của con người. Với sự phát triển của công nghệ, việc nghe nhạc không còn giới hạn trong các thiết bị phát truyền thống, mà đã chuyển sang nền tảng trực tuyến (online) với khả năng truy cập dễ dàng, đa dạng nội dung và cá nhân hóa trải nghiệm người dùng. Trong số các nền tảng âm nhạc phổ biến hiện nay, Spotify là một trong những ứng dụng nổi bật với giao diện hiện đại, tính năng phong phú và thuật toán gợi ý thông minh.

Với mong muốn tìm hiểu sâu hơn về cách hoạt động của một nền tảng nghe nhạc hiện đại cũng như rèn luyện kỹ năng lập trình web, nhóm chúng em quyết định thực hiện đề tài Spotify Clone. Việc xây dựng một hệ thống mô phỏng chức năng cơ bản của Spotify sẽ giúp nhóm củng cố kiến thức về backend với Django, frontend với React, cũng như cơ sở dữ liệu và tư duy thiết kế hệ thống phần mềm.

Bên cạnh đó, đề tài này cũng là cơ hội để nhóm phát triển khả năng làm việc nhóm, quản lý dự án, và tìm hiểu sâu hơn về các kỹ thuật hiện đại như streaming nội dung, quản lý người dùng, phân quyền và giao tiếp giữa frontend và backend.

2. Mục tiêu dự án

Mục tiêu chính của dự án là xây dựng một hệ thống web mô phỏng nền tảng nghe nhạc trực tuyến **Spotify Clone**, với các chức năng cơ bản nhất nhằm đáp ứng nhu cầu trải nghiệm âm nhạc của người dùng. Cụ thể, dự án hướng đến các mục tiêu sau:

- **Xây dựng hệ thống nghe nhạc trực tuyến đơn giản:** Cho phép người dùng có thể đăng nhập, tìm kiếm, phát nhạc, xem danh sách bài hát yêu thích, tải bài hát yêu thích và tạo album cá nhân.
- **Phát triển giao diện người dùng hiện đại:** Sử dụng **React** để thiết kế giao diện web thân thiện, trực quan, dễ sử dụng và phản hồi nhanh.
- **Xây dựng hệ thống backend linh hoạt:** Sử dụng **Django** để xử lý nghiệp vụ, quản lý dữ liệu người dùng, bài hát, album, quyền truy cập và hệ thống quản trị.
- **Quản lý nội dung đa phương tiện:** Hỗ trợ phát nhạc (audio) và video âm nhạc từ server một cách mượt mà, đảm bảo hiệu năng khi có nhiều người dùng truy cập đồng thời.
- **Tăng cường khả năng quản trị hệ thống:** Thiết kế trang quản trị (admin) để dễ dàng kiểm soát nội dung bài hát, người dùng, phân quyền và dữ liệu hệ thống.
- **Thực hành tích hợp toàn bộ quy trình phát triển phần mềm web:** Từ thiết kế cơ sở dữ liệu, xây dựng API, kết nối frontend-backend, đến kiểm thử và triển khai.

Thông qua việc thực hiện đề tài này, nhóm mong muốn phát triển tư duy lập trình, rèn luyện kỹ năng teamwork, và làm quen với quy trình phát triển ứng dụng web hoàn chỉnh.

3. Phạm vi dự án

Dự án **Spotify Clone** được thực hiện trong phạm vi học thuật, với mục đích mô phỏng một phần chức năng của hệ thống nghe nhạc trực tuyến Spotify. Dựa trên thời gian, nguồn lực và năng lực kỹ thuật hiện tại, nhóm tập trung vào những nội dung sau:

Chức năng chính triển khai

- Đăng ký và đăng nhập người dùng.
- Phát nhạc.
- Tạo, quản lý album và danh sách bài hát yêu thích.
- Tìm kiếm bài hát theo tên hoặc nghệ sĩ.
- Giao diện người dùng thân thiện (bằng React).
- Hệ thống backend quản lý người dùng, bài hát và album (bằng Django).
- Trang Admin cho phép quản trị người dùng, bài hát và album.

Chức năng chưa triển khai

- Phát video âm nhạc.
- Tải video âm nhạc.
- Kết nối với API ngoài.

Hệ thống triển khai

- Giao diện người dùng: React.
- Backend: Django (Python).
- Cơ sở dữ liệu: SQLite3, MySQL.
- Chạy cục bộ (localhost).

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

1. Spotify là gì?

Spotify là một nền tảng dịch vụ phát nhạc trực tuyến (streaming) nổi tiếng trên toàn cầu, cho phép người dùng nghe hàng triệu bài hát, album, podcast và nội dung âm thanh từ nhiều nghệ sĩ, nhà sản xuất khác nhau. Spotify được thành lập vào năm 2006 tại Thụy Điển bởi Daniel Ek và Martin Lorentzon và chính thức ra mắt năm 2008. Tính đến nay, Spotify đã trở thành một trong những ứng dụng phát nhạc phổ biến nhất thế giới, với hàng trăm triệu người dùng hoạt động hàng tháng.

Các đặc điểm nổi bật của Spotify:

- **Phát nhạc theo yêu cầu:** Người dùng có thể tìm kiếm và phát bất kỳ bài hát, nghệ sĩ, hoặc album nào mình thích.
- **Gợi ý thông minh:** Spotify sử dụng thuật toán và trí tuệ nhân tạo để gợi ý nhạc phù hợp với sở thích cá nhân, như Daily Mix, Discover Weekly,...
- **Tạo playlist cá nhân và chia sẻ:** Người dùng có thể tạo danh sách phát (playlist), chia sẻ với bạn bè hoặc cộng đồng.
- **Hỗ trợ nhiều nền tảng:** Có thể sử dụng trên Windows, macOS, iOS, Android, và web.
- **Chế độ miễn phí & cao cấp:** Spotify cung cấp gói miễn phí (có quảng cáo) và gói Premium (không quảng cáo, phát offline, chất lượng cao).

Ứng dụng trong đề tài:

Việc nghiên cứu và phát triển một hệ thống **Spotify Clone** giúp nhóm hiểu rõ cách vận hành một nền tảng nghe nhạc hiện đại, cũng như áp dụng các kiến thức về lập trình web, cơ sở dữ liệu, giao diện người dùng, xử lý media và quản lý người dùng.

2. Spotify Clone là gì?

Spotify Clone là một phiên bản mô phỏng lại các tính năng và giao diện của một ứng dụng gốc nổi tiếng nhằm mục đích học tập, nghiên cứu hoặc phát triển ý tưởng dựa trên nền tảng sẵn có. Việc tạo một app clone không có nghĩa là sao chép hoàn toàn hay vi phạm bản quyền, mà là cách để sinh viên hoặc lập trình viên học hỏi cách hoạt động của những hệ thống lớn trong thực tế.

Mục đích của việc xây dựng clone app:

- **Học hỏi kiến trúc hệ thống thực tế:** Hiểu được cách tổ chức backend, frontend và cơ sở dữ liệu.
- **Luyện tập kỹ năng lập trình:** Áp dụng các công nghệ như Python, Django, React, xử lý media,...
- **Thực hành quy trình phát triển phần mềm:** Từ phân tích yêu cầu, thiết kế hệ thống, đến triển khai và kiểm thử.
- **Sáng tạo và mở rộng tính năng:** Nhóm có thể phát triển thêm các chức năng riêng hoặc cải tiến giao diện, trải nghiệm người dùng.

Clone Spotify trong đề tài:

Trong khuôn khổ đề tài, nhóm lựa chọn clone ứng dụng **Spotify** để xây dựng một nền tảng phát nhạc trực tuyến có các chức năng cơ bản như:

- Nghe nhạc
- Quản lý album.
- Tải bài hát.
- Quản lý danh sách bài hát yêu thích
- Đăng nhập cho người dùng và admin
- Quản lý người dùng, bài hát (admin)
- Giao diện web thân thiện, dễ sử dụng

3. Các công nghệ sử dụng trong dự án

Trong quá trình phát triển hệ thống Spotify Clone, nhóm đã sử dụng một số công nghệ phổ biến hiện nay để đảm bảo hiệu suất, tính mở rộng và dễ bảo trì. Dưới đây là các công nghệ chính:

1. Python

- Là ngôn ngữ lập trình chính trong phần backend.
- Dễ học, cú pháp rõ ràng, có nhiều thư viện hỗ trợ như Django, requests, pandas, v.v.

2. Django (Framework Python)

- Django là một web framework mạnh mẽ giúp phát triển nhanh các ứng dụng web theo mô hình MVC (Model-View-Controller).
- Hỗ trợ ORM, routing, quản lý user, hệ thống admin,... phù hợp cho ứng dụng có quản lý dữ liệu như Spotify Clone.

3. React (Frontend)

- React là thư viện JavaScript phổ biến được phát triển bởi Facebook.
- Giúp xây dựng giao diện người dùng động, tái sử dụng component và dễ quản lý trạng thái ứng dụng.

4. SQLite3 (Cơ sở dữ liệu)

- Là hệ quản trị cơ sở dữ liệu nhẹ, tích hợp sẵn với Django.
- Phù hợp cho các ứng dụng nhỏ hoặc mục đích học tập.

5. HTML, CSS, JavaScript

- Dùng để xây dựng và tùy chỉnh giao diện web.
- Kết hợp với React để tối ưu trải nghiệm người dùng.

6. Git & GitHub

- Git được sử dụng để quản lý phiên bản mã nguồn.
- GitHub hỗ trợ lưu trữ, chia sẻ và cộng tác giữa các thành viên trong nhóm.

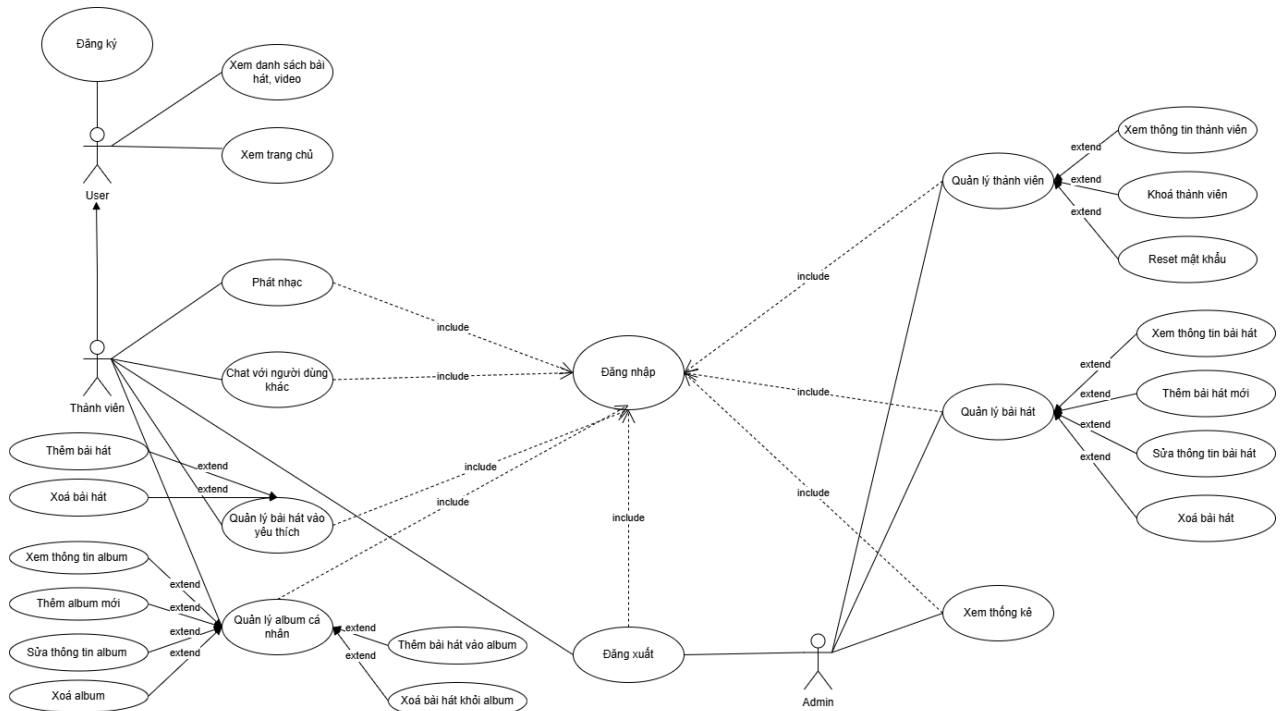
7. VS Code

- Là trình soạn thảo mã nguồn chính được nhóm sử dụng để viết code cho cả frontend và backend.
- Có hỗ trợ nhiều extension giúp lập trình hiệu quả hơn.

CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG

1. Use Case Diagram

Use Case Diagram là sơ đồ biểu diễn các chức năng chính mà hệ thống cung cấp cho người dùng (actor). Trong dự án Spotify Clone, hệ thống có hai nhóm người dùng chính là **User** (người dùng phổ thông) và **Admin** (người quản trị hệ thống). Mỗi actor sẽ tương tác với hệ thống qua các chức năng khác nhau.



2. Mô tả các chức năng chính

Dưới đây là các chức năng chính của hệ thống Spotify Clone, được chia theo từng nhóm người dùng:

2.1. Chức năng dành cho người dùng (User):

- **Đăng ký tài khoản:**
Người dùng có thể tạo tài khoản mới bằng email, tên người dùng và mật khẩu.
- **Đăng nhập:**
Cho phép người dùng đăng nhập vào hệ thống bằng email và mật khẩu đã đăng ký.
- **Tìm kiếm bài hát:**
Hỗ trợ người dùng tìm bài hát theo tên, ca sĩ hoặc thể loại.
- **Nghe nhạc:**
Phát nhạc trực tuyến (streaming), có thể play/pause, chuyển bài, lặp lại hoặc phát ngẫu nhiên.
- **Tải bài hát:**

Tải bài hát về máy dưới định dạng file mp3.

- **Tạo và quản lý album cá nhân:**

Người dùng có thể tự tạo album chứa các bài hát yêu thích để dễ dàng quản lý.

- **Thêm bài hát vào danh sách yêu thích:**

Chức năng giúp người dùng lưu lại những bài hát yêu thích để nghe lại sau.

- **Xem danh sách yêu thích:**

Truy cập danh sách bài hát đã lưu và có thể phát nhạc trực tiếp từ đây.

2.2 Chức năng dành cho quản trị viên (Admin):

- **Đăng nhập quản trị:**

Admin đăng nhập bằng tài khoản riêng để truy cập giao diện quản lý.

- **Quản lý người dùng:**

Admin có thể xem danh sách người dùng, khóa tài khoản, hoặc xóa tài khoản vi phạm.

- **Quản lý bài hát:**

Cho phép admin thêm mới bài hát, sửa thông tin hoặc xóa bài hát khỏi hệ thống.

- **Tải bài hát:**

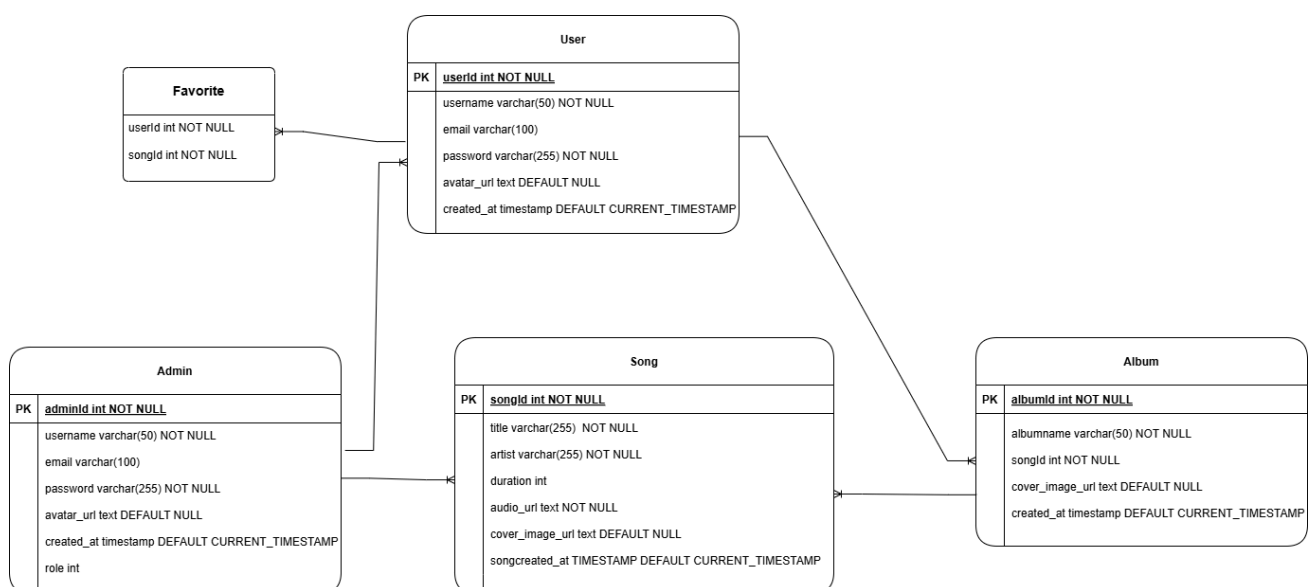
Tải bài hát về máy dưới định dạng file mp3.

- **Xem thống kê hệ thống:**

Hiển thị các số liệu như số lượng người dùng, số lượt nghe nhạc, các bài hát phổ biến,...

3. Biểu đồ ERD (Entity Relationship Diagram)

Biểu đồ ERD (Entity Relationship Diagram) thể hiện mối quan hệ giữa các thực thể trong hệ thống *Spotify Clone*. Đây là bước quan trọng để thiết kế cơ sở dữ liệu một cách trực quan, rõ ràng và logic.



CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI

1. Môi trường cài đặt

Để phát triển và triển khai hệ thống Spotify Clone, nhóm sử dụng các công cụ và môi trường sau:

Thành phần	Công cụ/Sản phẩm sử dụng	Ghi chú
Hệ điều hành	Linux, Windows 10, 11	Phát triển và kiểm thử trên cả hai nền tảng
Ngôn ngữ lập trình	Python 3.11, JavaScript (ES6+)	Backend (Python), Frontend (JS)
Backend Framework	Django 4.x + Django REST Framework	Xây dựng API
Frontend Framework	ReactJS (Tailwind Css)	Giao diện người dùng
Database	SQLite3 (dev), PostgreSQL (prod)	Lưu trữ thông tin người dùng, bài hát,...
Trình quản lý gói	pip (Python), npm/yarn (NodeJS)	Cài đặt thư viện
IDE & Editor	VS Code, PyCharm	Viết mã và debug
API Client	Postman	Kiểm thử API
GitHub	Git/GitHub	Quản lý mã nguồn
Trình duyệt	Google Chrome	Kiểm tra giao diện

Cấu hình hệ thống tối thiểu đề xuất:

- **RAM:** ≥ 4 GB
- **Ổ cứng:** ≥ 2 GB trống
- **Python:** 3.10 trở lên
- **NodeJS:** 16.x trở lên

2. Các bước triển khai

2.1. Cài đặt Backend (Django)

2.1.1 Tạo môi trường ảo:

```
python -m venv venv
source venv/bin/activate    # Trên Linux/Mac
venv\Scripts\activate      # Trên Windows
```

2.1.2 Cài đặt Django và các thư viện cần thiết:

```
pip install django djangorestframework
```

2.1.3 Khởi tạo project và app:

```
django-admin startproject spotify_clone
cd spotify_clone
python manage.py startapp music
```

2.1.4 Cấu hình settings.py:

- Thêm `rest_framework`, `music` vào `INSTALLED_APPS`
- Cấu hình `DATABASES`, `STATIC_URL`, v.v.

2.1.5 Chạy migrate và server:

```
python manage.py makemigrations
python manage.py migrate
python manage.py runserver
```

2.2 Cài đặt Frontend (React)

2.2.1 Tạo project React:

```
npx create-react-app spotify-client
cd spotify-client
```

2.2.2 Cài đặt các thư viện frontend:

```
npm install axios react-router-dom
```

2.2.3 Tạo các component giao diện:

- Trang login/register
- Trang phát nhạc
- Trang admin

2.2.4 Chạy ứng dụng frontend:

```
npm start
```

2.3 Kết nối Frontend & Backend

- Đảm bảo backend cho phép CORS (cài `django-cors-headers`)
- Cấu hình URL API trong React trở về `http://localhost:8000/api/`

3. Một số code quan trọng:

CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC

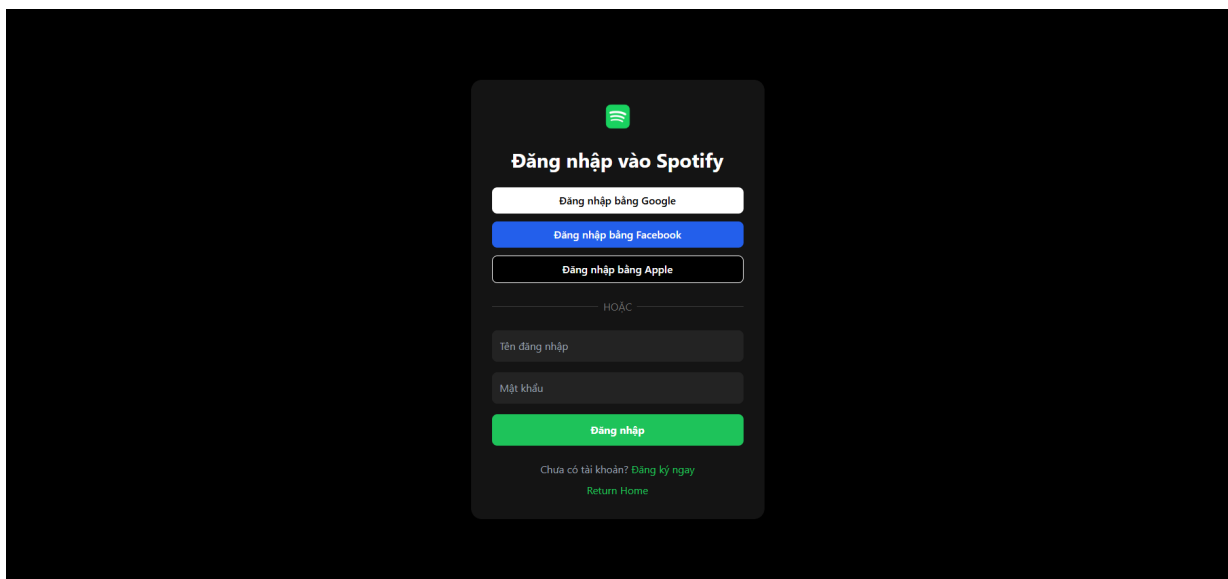
1. Các chức năng đã hoàn thiện

Dự án *Spotify Clone* đã hoàn thiện một số chức năng cốt lõi như sau:

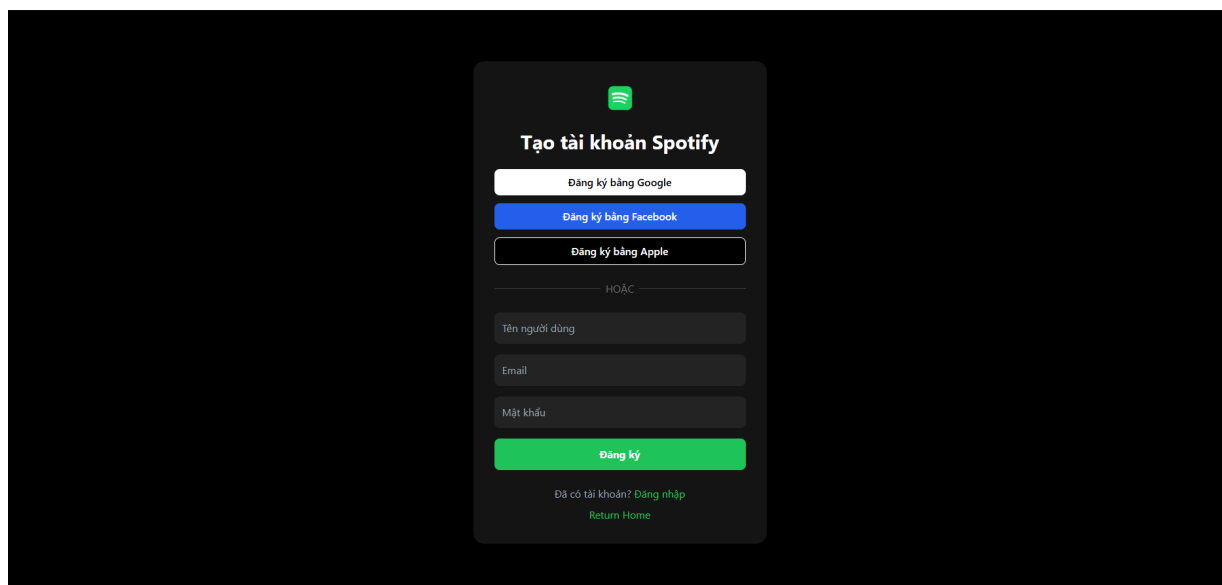
STT	Chức năng	Mô tả
1	Đăng ký, đăng nhập người dùng	Cho phép người dùng tạo tài khoản, đăng nhập để sử dụng hệ thống
2	Phát nhạc trực tuyến	Người dùng có thể chọn bài hát và phát trực tuyến qua trình phát nhạc
3	Danh sách bài hát	Hiển thị tất cả bài hát đang có trong hệ thống
4	Tạo album	Người dùng có thể thêm bài hát vào album cá nhân
6	Quản lý user (Admin)	Admin có thể xem thông tin người dùng, khoá người dùng và reset mật khẩu
7	Yêu thích bài hát	Người dùng có thể thêm bài hát vào danh sách yêu thích của họ
8	Giao diện người dùng	Thiết kế bằng React, đơn giản và thân thiện

2. Giao diện thực tế (Frontend)

Trang đăng nhập / đăng ký

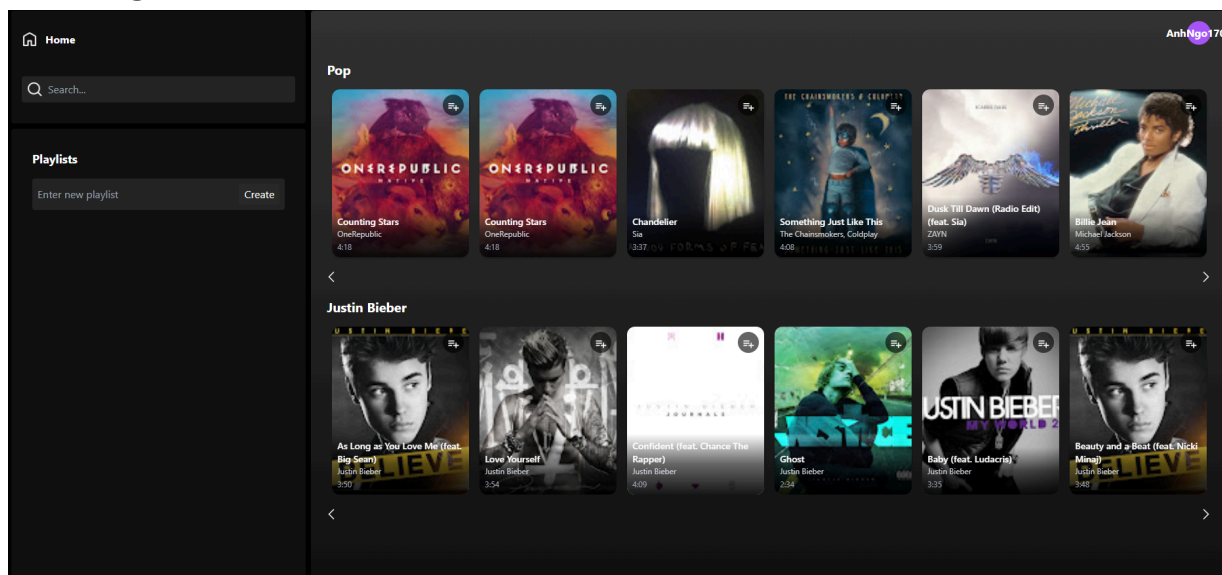


*Trang đăng nhập

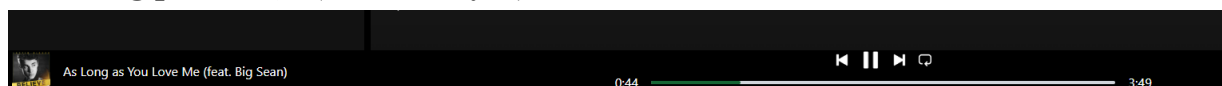


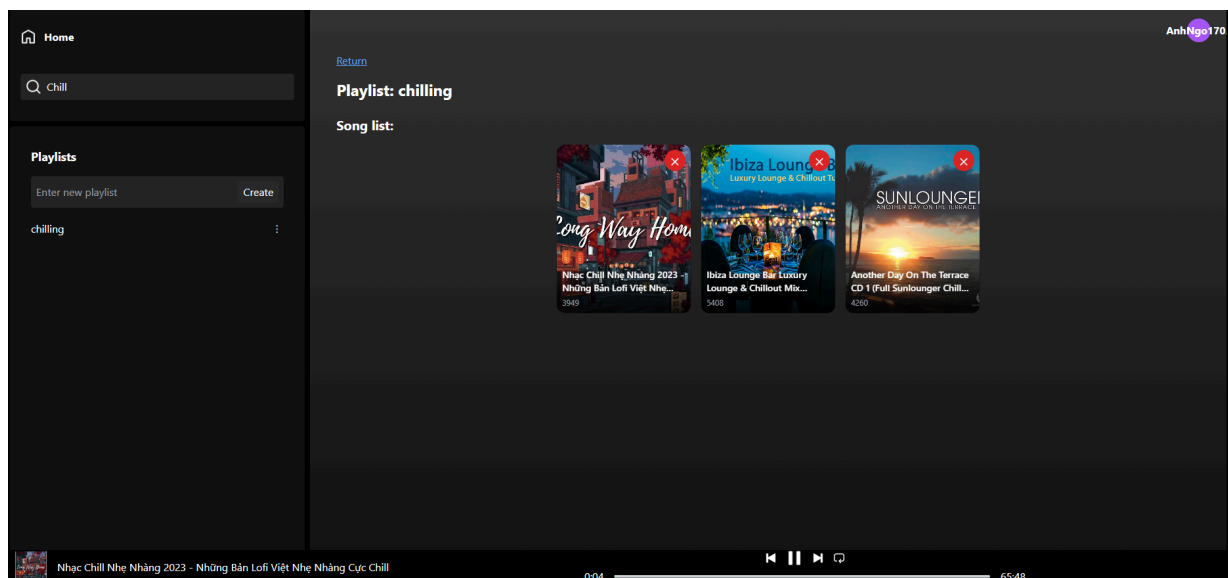
*Trang đăng kí

Trang danh sách bài hát



Trang phát nhạc (Music Player)





Giao diện quản lý cho Admin

CHƯƠNG 6: ĐÁNH GIÁ – KẾT LUẬN – HƯỚNG PHÁT TRIỂN

1. Tóm tắt kết quả đạt được

Sau thời gian làm việc nhóm và triển khai dự án, nhóm đã đạt được những kết quả sau:

1.1 Chức năng cốt lõi

- ★ **Xây dựng thành công hệ thống Spotify Clone cơ bản**, bao gồm các tính năng chính như:
 - Đăng nhập, đăng ký người dùng.
 - Giao diện người dùng
 - Phát nhạc trực tuyến
 - Tải bài hát yêu thích
 - Giao diện danh sách bài hát.
 - Tạo album cá nhân
 - Tạo danh sách bài hát yêu thích..
- ★ **Hoàn thiện giao diện người dùng bằng React**, thân thiện và dễ sử dụng.
- ★ **Xây dựng hệ thống backend bằng Django REST framework**, đảm bảo xử lý yêu cầu nhanh và rõ ràng theo mô hình RESTful.
- ★ **Lưu trữ dữ liệu hiệu quả bằng SQLite3** và tổ chức dữ liệu theo mô hình quan hệ (có sơ đồ ERD).
- ★ **Phân chia rõ ràng vai trò người dùng và Admin**, đảm bảo phân quyền và bảo mật cơ bản.

1.2 Kinh nghiệm làm việc nhóm:

Ngoài các kết quả về mặt chức năng kỹ thuật của hệ thống, trong quá trình thực hiện dự án Spotify Clone, nhóm cũng đã thu nhận được nhiều **kinh nghiệm thực tế quý báu** và **nâng cao năng lực cá nhân và làm việc nhóm**, cụ thể như sau:

- Nhóm gồm 3 thành viên, mỗi bạn có nền tảng kiến thức khác nhau nên việc phân chia công việc hợp lý là một trong những yếu tố then chốt. Trong quá trình làm, nhóm đã học cách:
 - Phân công vai trò cụ thể (Frontend, Backend, viết báo cáo, thiết kế use case diagram, ERD...).
 - Sử dụng các công cụ cộng tác hiệu quả như GitHub, Google Docs, Trello để quản lý tiến độ.

- Tổ chức họp nhóm định kỳ để cập nhật tiến độ, chia sẻ khó khăn và hỗ trợ nhau.
- Nhóm cũng trải qua các tình huống mâu thuẫn nhỏ về cách tổ chức code hoặc chọn công nghệ, tuy nhiên đã học được cách giao tiếp rõ ràng, lắng nghe ý kiến nhau và đưa ra quyết định tập thể, từ đó cải thiện kỹ năng làm việc nhóm rất nhiều.

1.3 Kiến thức và kỹ năng lập trình học được:

Thông qua việc tự tay thực hiện dự án từ đầu đến cuối, các thành viên trong nhóm đã có cơ hội củng cố và học thêm nhiều kỹ năng:

- **Kỹ năng lập trình Frontend với React:**
 - Hiểu được cấu trúc và vòng đời component.
 - Áp dụng React Hooks như useState, useEffect.
 - Giao tiếp với backend qua fetch hoặc axios.
 - Sử dụng thư viện UI hỗ trợ như Tailwind CSS hoặc Material UI.
- **Kỹ năng lập trình Backend với Django:**
 - Tạo mô hình dữ liệu (models), views và serializers.
 - Xây dựng API theo chuẩn RESTful.
 - Xử lý phân quyền (user và admin).
 - Kết nối và thao tác dữ liệu qua SQLite3.
- **Kỹ năng triển khai và kiểm thử:**
 - Cài đặt môi trường làm việc trên Linux.
 - Kiểm thử tính năng, fix bug, tối ưu hiệu năng cơ bản.
 - Biết cách kiểm soát lỗi thông qua debug và logging.
- **Kỹ năng bảo mật và phân quyền:**
 - Áp dụng xác thực token hoặc session.
 - Đảm bảo rằng admin có thể thực hiện nhiều chức năng hơn user thông thường.

1.4 Phát triển tư duy hệ thống và tư duy thiết kế phần mềm:

Nhóm cũng học được cách **thiết kế hệ thống phần mềm theo hướng mô-đun hóa** và hiểu hơn về:

- Cách tổ chức code hợp lý, dễ mở rộng.
- Việc **tách riêng frontend và backend** giúp tăng tính linh hoạt, tái sử dụng, dễ bảo trì hệ thống.

Tóm lại, dự án không chỉ là một bài tập kỹ thuật mà còn là một trải nghiệm thực tế giúp các thành viên trưởng thành hơn về cả chuyên môn lẫn kỹ năng làm việc chuyên nghiệp.

2. Hạn chế và khó khăn

2.1 Kiến thức và kinh nghiệm còn hạn chế

Đây là lần đầu nhóm thực hiện một dự án full-stack kết hợp frontend (React) và backend (Django), nên không tránh khỏi lúng túng ban đầu trong việc tổ chức dự án và chia tách các thành phần.

2.2 Hạn chế về giao diện và tính năng nâng cao

- Dự án chủ yếu tập trung vào chức năng cơ bản như phát nhạc, quản lý user và admin, tìm kiếm bài hát. Các tính năng nâng cao như phát video, tải video, hoặc tích hợp chat chưa thể triển khai do thời gian và nhân lực có hạn.
- Giao diện UI chưa thực sự đẹp mắt hoặc thân thiện như các ứng dụng thương mại do thiếu kỹ năng thiết kế UI/UX chuyên sâu.

2.3 Quản lý thời gian và tiến độ

- Do thời gian thực hiện bị chia nhỏ bởi lịch học và các môn học khác, nên đôi khi việc họp nhóm không đồng bộ, dễ bị trì trệ.
- Nhóm phải liên tục điều chỉnh kế hoạch, chia lại công việc để đảm bảo tiến độ dự án.
- Một số vấn đề và khó khăn dẫn đến sự bất cập và cho ra thành phẩm thật sự không như mong đợi và còn quá nhiều thiếu sót.

Dù còn một số thiếu sót, nhóm vẫn coi đây là một trải nghiệm quý giá, từ đó rút ra được bài học thực tế và xác định rõ hơn định hướng phát triển kỹ năng trong tương lai.

3. Hướng phát triển:

Sau khi hoàn thành phiên bản đầu tiên của dự án Spotify Clone với các chức năng cơ bản, nhóm đã rút ra nhiều kinh nghiệm quý báu và đồng thời cũng nhận thấy còn nhiều tiềm năng để mở rộng, cải tiến hệ thống trong tương lai. Một số định hướng phát triển như sau:

3.1. Hoàn thiện tính năng nâng cao

- **Tạo playlist cá nhân:** Cho phép người dùng tạo, chỉnh sửa, và chia sẻ các danh sách phát yêu thích.
- **Phát video âm nhạc:** Mở rộng hệ thống để phát nhạc kèm theo video, tương tự như YouTube Music.
- **Tải video âm nhạc**

- **Gợi ý bài hát:** Tích hợp các thuật toán Machine Learning hoặc hệ thống recommendation để gợi ý bài hát phù hợp với thói quen nghe nhạc của người dùng.
- **Tích hợp chức năng bình luận và đánh giá:** Tạo môi trường tương tác, nơi người dùng có thể để lại nhận xét hoặc chia sẻ cảm nghĩ về bài hát.

3.2. Cải thiện giao diện người dùng (UI/UX)

- **Thiết kế lại giao diện:** Áp dụng các nguyên tắc UI/UX hiện đại nhằm tối ưu trải nghiệm người dùng trên cả máy tính và thiết bị di động.
- **Đa ngôn ngữ:** Hỗ trợ nhiều ngôn ngữ, không chỉ tiếng Việt và tiếng Anh, nhằm mở rộng phạm vi người dùng.

3.3. Bảo mật và xác thực người dùng

- **Xác thực OAuth (Google, Facebook):** Cho phép người dùng đăng nhập nhanh bằng tài khoản Google hoặc Facebook.
- **Mã hóa mật khẩu:** Áp dụng các phương pháp mã hóa an toàn hơn (bcrypt, hash + salt).
- **Xác minh email:** Gửi email xác nhận khi đăng ký để tăng độ tin cậy và bảo mật.

3.4. Triển khai thực tế trên nền tảng cloud

- **CI/CD:** Thiết lập quy trình triển khai liên tục để cập nhật nhanh chóng tính năng mới và vá lỗi.
- **Sử dụng Docker:** Đóng gói ứng dụng frontend, backend và database thành container để dễ dàng triển khai.
- **Sử dụng Cloud Storage (AWS S3, Firebase):** Lưu trữ và truyền tải file nhạc/video hiệu quả hơn.

3.5. Phân tích hành vi người dùng

- **Tích hợp công cụ phân tích:** Dùng Google Analytics hoặc hệ thống tùy chỉnh để thu thập dữ liệu người dùng như: thời lượng nghe nhạc, lượt truy cập, bài hát yêu thích,...
- **Dashboard quản trị:** Cung cấp giao diện thống kê dữ liệu, giúp admin theo dõi hiệu suất hệ thống và xu hướng người dùng.

3.6. Phát triển ứng dụng di động

- **React Native hoặc Flutter:** Tái sử dụng codebase để phát triển ứng dụng Spotify Clone cho Android và iOS, mở rộng khả năng tiếp cận người dùng.

Những định hướng trên không chỉ giúp hệ thống trở nên hoàn thiện hơn mà còn tạo điều kiện để nhóm phát triển kỹ năng, nghiên cứu công nghệ mới và hướng đến những sản phẩm có giá trị thực tiễn cao.