

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



Phát triển phần mềm mã nguồn mở

Phát triển phần mềm Spotify Clone

GVHD: Từ Lãng Phiêu
SV: Huỳnh Xuân Bách - 3121560013
Trần Quốc Sĩ - 3121410420
Email: hxb3011@outlook.com

TP. HỒ CHÍ MINH, THÁNG 5/2025

Mục lục

1	Phần giới thiệu	2
1.1	Mục tiêu	2
1.2	Yêu cầu chức năng	2
1.3	Các thư viện sử dụng	2
2	Thiết kế	4
2.1	Cơ sở dữ liệu	4
2.2	Cấu trúc dự án	5
2.2.1	Backend	5
2.2.2	Frontend	6
2.3	Giao diện	9
2.3.1	Trang chủ	10
2.3.2	Thể loại	10
2.3.3	Bản nhạc yêu thích	11
2.3.4	Danh sách phát thể loại	11
2.3.5	Danh sách phát	12
2.3.6	Tìm kiếm	12
2.3.7	Nghệ sĩ	13
2.3.8	Hồ sơ	13
2.3.9	Tải lên ảnh hồ sơ	13
2.3.10	Tạo danh sách phát	14
2.3.11	Xoá danh sách phát	14
2.3.12	Dăng nhập	15
2.3.13	Dăng ký	15
2.3.14	Trang admin - Hồ sơ	16
2.3.15	Trang admin - Chi tiết hồ sơ	16
2.3.16	Trang admin - Nghệ sĩ	17
2.3.17	Trang admin - Chi tiết nghệ sĩ	17
2.3.18	Trang admin - Thể loại	18
2.3.19	Trang admin - Chi tiết thể loại	18
2.3.20	Trang admin - Danh sách phát	19
2.3.21	Trang admin - Chi tiết danh sách phát	19
2.3.22	Trang admin - Bản nhạc	20
2.3.23	Trang admin - Chi tiết bản nhạc	20
3	Hướng dẫn cài đặt	21
4	Phân công	21



1 Phân giới thiệu

1.1 Mục tiêu

Thiết kế giao diện web phần mềm Spotify Clone và lập trình bằng ngôn ngữ Python.

1. **Frontend:** React hoặc Angular
2. **Backend:** Django
3. **Database:** Postgres

1.2 Yêu cầu chức năng

1. Chức năng phát nhạc:
2. Chức năng phát video âm nhạc:
3. Chức năng tải video âm nhạc:
4. Chức năng User tạo album, bài hát yêu thích
5. Trang Admin.
6. Optional: Có thêm tính năng chat tích hợp trong giao diện web spotify clone.

1.3 Các thư viện sử dụng

Django là một framework web mạnh mẽ và phổ biến được viết bằng Python, giúp các lập trình viên phát triển ứng dụng nhanh chóng, bảo mật và dễ mở rộng. Với mô hình MTV (Model-Template-View), Django giúp tổ chức mã nguồn rõ ràng và tối ưu hóa quy trình phát triển. Được trang bị sẵn các công cụ như hệ thống quản lý người dùng, ORM (Object-Relational Mapping) mạnh mẽ để làm việc với cơ sở dữ liệu, cùng khả năng mở rộng dễ dàng thông qua các ứng dụng tùy chỉnh. Django thường được sử dụng để xây dựng các hệ thống quản lý nội dung, nền tảng thương mại điện tử, API backend, ứng dụng mạng xã hội, và cả các hệ thống phân tích dữ liệu. Framework này trở thành lựa chọn lý tưởng cho những ai muốn tạo ứng dụng web nhanh chóng mà vẫn đảm bảo hiệu suất và bảo mật.

Django Rest Framework là một thư viện mạnh mẽ giúp xây dựng các API RESTful dễ dàng trên nền Django. Nó cung cấp các công cụ hỗ trợ xử lý dữ liệu, xác thực người dùng, phân quyền, và serialize dữ liệu theo nhiều định dạng như JSON. Với DRF, bạn có thể nhanh chóng thiết kế API REST với các lớp APIView, ViewSets, và sử dụng Serializers để chuyển đổi dữ liệu giữa Django models và định dạng API. Ngoài ra, DRF đi kèm với giao diện duyệt API tự động, giúp việc kiểm tra và tương tác với API trực quan hơn. Được sử dụng rộng rãi trong các ứng dụng web hiện đại, Django Rest Framework giúp tăng tốc phát triển API đồng thời giữ mã nguồn rõ ràng, dễ bảo trì.

Docker là một nền tảng mạnh mẽ giúp đóng gói, triển khai và chạy ứng dụng trong các container một cách dễ dàng. Với Docker, các lập trình viên có thể tạo môi trường nhất quán trên nhiều hệ thống khác nhau, từ máy cá nhân đến cloud, mà không gặp vấn đề về xung đột dependencies. Docker giúp tăng tốc quá trình phát triển bằng cách cung cấp một hệ thống nhẹ, linh hoạt, có thể mở rộng và dễ quản lý.

Các ứng dụng phổ biến của Docker bao gồm: Triển khai microservices (Chạy nhiều dịch vụ độc lập trong container, giúp phát triển ứng dụng linh hoạt hơn). CI/CD (Continuous Integration/Continuous Deployment - Tích hợp Docker vào pipeline để kiểm thử và triển khai nhanh



chóng). Phát triển ứng dụng đa nền tảng (Đảm bảo ứng dụng chạy ổn định trên nhiều môi trường khác nhau mà không cần điều chỉnh). Quản lý môi trường máy chủ (Giúp giảm tải việc cài đặt và cấu hình phần mềm bằng cách dùng container với môi trường sẵn sàng sử dụng).

React là một thư viện JavaScript phổ biến được phát triển bởi Facebook, giúp xây dựng giao diện người dùng (UI) một cách nhanh chóng và hiệu quả. Được thiết kế theo mô hình component-based, React cho phép lập trình viên tái sử dụng các thành phần UI để tạo ra những ứng dụng linh hoạt, tương tác cao. Một trong những điểm mạnh của React là Virtual DOM, giúp cập nhật giao diện nhanh chóng mà không cần render lại toàn bộ trang, cải thiện hiệu suất đáng kể.

Các ứng dụng phổ biến của React bao gồm: Single Page Applications (SPA - Tạo các ứng dụng web tương tác như Gmail, Facebook, Twitter). Ứng dụng di động (Kết hợp với React Native, React có thể phát triển ứng dụng chạy trên cả iOS và Android). Hệ thống Dashboard & Quản lý dữ liệu (Xây dựng UI linh hoạt cho các trang admin, phân tích dữ liệu). E-commerce & Web App (Tạo giao diện mượt mà cho các trang thương mại điện tử, hệ thống đặt hàng trực tuyến).

Redux là một thư viện quản lý trạng thái mạnh mẽ dành cho JavaScript, đặc biệt hữu ích trong các ứng dụng React. Nó giúp quản lý trạng thái toàn cục của ứng dụng một cách rõ ràng và nhất quán bằng cách sử dụng store tập trung, thay vì lưu trữ dữ liệu phân tán trong nhiều component. Redux hoạt động theo mô hình unidirectional data flow, đảm bảo trạng thái ứng dụng thay đổi một cách kiểm soát thông qua actions và reducers, giúp dễ dàng debug và mở rộng ứng dụng.

Các ứng dụng phổ biến của Redux bao gồm: Quản lý trạng thái phức tạp trong ứng dụng React (Được sử dụng rộng rãi trong ứng dụng web lớn, nơi nhiều component cần chia sẻ dữ liệu). Lưu trữ dữ liệu người dùng (Giúp duy trì phiên đăng nhập, giỏ hàng trong các trang thương mại điện tử). Đồng bộ hóa dữ liệu từ API (Kết hợp với Redux Thunk hoặc Redux Saga để xử lý các request bất đồng bộ). Tối ưu hóa hiệu suất (Giảm việc re-render không cần thiết bằng cách cập nhật trạng thái một cách hiệu quả).

Material UI (MUI) là một thư viện giao diện người dùng mạnh mẽ dành cho React, giúp các nhà phát triển tạo ra các ứng dụng web với thiết kế hiện đại theo phong cách Material Design của Google. MUI cung cấp một tập hợp các component có sẵn, từ button, modal, đến table và form input, giúp xây dựng giao diện một cách nhanh chóng mà vẫn đảm bảo tính thẩm mỹ và nhất quán.

Ứng dụng của Material UI: Tạo giao diện web chuyên nghiệp (Sử dụng các component được thiết kế sẵn giúp giảm thời gian xây dựng UI). Tối ưu hóa trải nghiệm người dùng (Material UI tuân theo các tiêu chuẩn thiết kế hiện đại, giúp ứng dụng thân thiện và dễ sử dụng). Tích hợp dễ dàng với React (Hỗ trợ hệ thống theme tùy chỉnh, giúp cá nhân hóa giao diện theo nhu cầu). Phát triển ứng dụng dashboard, admin panel (Nhờ vào các component mạnh mẽ như Table, Drawer, và Grid, việc xây dựng trang quản trị dễ dàng hơn).

TypeScript là một ngôn ngữ lập trình dựa trên JavaScript, được Microsoft phát triển nhằm cải thiện tính an toàn và khả năng mở rộng của mã nguồn. Với TypeScript, các lập trình viên có thể sử dụng kiểu dữ liệu tĩnh (static typing), giúp phát hiện lỗi ngay từ giai đoạn biên dịch thay vì lúc chạy chương trình. Điều này làm tăng hiệu suất phát triển, giảm lỗi và giúp quản lý các dự án lớn dễ dàng hơn.

Ứng dụng của TypeScript: Phát triển ứng dụng web lớn (Được sử dụng trong các framework như Angular, Next.js để đảm bảo tính ổn định của mã nguồn). Xây dựng backend với Node.js (TypeScript giúp tổ chức logic backend chặt chẽ hơn, đặc biệt khi làm việc với Express.js). Tối ưu hóa ứng dụng React (TypeScript giúp định nghĩa props rõ ràng trong các component, giúp tăng độ tin cậy và dễ bảo trì). Phát triển phần mềm quy mô lớn (Nhiều dự án lớn như VS Code,



Slack, và Airbnb sử dụng TypeScript để quản lý mã nguồn hiệu quả).

Tailwind CSS là một framework CSS phổ biến giúp phát triển giao diện web nhanh chóng bằng cách sử dụng các utility-first classes. Thay vì viết CSS tùy chỉnh, Tailwind cho phép lập trình viên áp dụng các class trực tiếp vào HTML để tạo kiểu ngay lập tức mà không cần viết file CSS riêng. Điều này giúp tăng tốc quá trình thiết kế, giảm sự phụ thuộc vào các stylesheet phức tạp và đảm bảo giao diện dễ bảo trì.

Ứng dụng của Tailwind CSS: Phát triển giao diện web hiện đại (Cung cấp các class tiện ích giúp thiết kế layout, typography, màu sắc và hiệu ứng nhanh chóng). Xây dựng ứng dụng mobile-friendly (Tailwind hỗ trợ responsive design trực tiếp trong cú pháp class mà không cần media queries phức tạp). Tạo giao diện tùy chỉnh dễ dàng (Không bị giới hạn bởi các component mặc định như Bootstrap, giúp thiết kế UI linh hoạt theo ý muốn). Tăng tốc phát triển UI trong các dự án lớn (Được sử dụng trong các hệ thống dashboard, trang thương mại điện tử, blog, và ứng dụng SaaS).

Vite là một công cụ build hiện đại dành cho các ứng dụng web, giúp tăng tốc quá trình phát triển bằng cách cải thiện hiệu suất biên dịch và hot module replacement (HMR). Không giống như Webpack, Vite sử dụng ES module trong trình duyệt và chỉ biên dịch mã khi cần thiết, giúp thời gian khởi động nhanh hơn đáng kể.

Ứng dụng của Vite: Phát triển frontend siêu nhanh (Tăng tốc quá trình khởi động và cập nhật mã với HMR hiệu quả). Tích hợp với React, Vue, Svelte (Hỗ trợ các framework phổ biến mà không cần cấu hình phức tạp). Xây dựng ứng dụng SPA và MPA (Cung cấp các tính năng tối ưu hóa cho cả ứng dụng đơn trang và đa trang). Tối ưu hóa cho production (Sử dụng Rollup để tạo bundle hiệu suất cao khi xuất bản ứng dụng).

2 Thiết kế

2.1 Cơ sở dữ liệu



dbdiagram.io

2.2 Cấu trúc dự án

2.2.1 Backend



```
✓ backend
  ✓ backend
    ✎ asgi.py
    ✎ settings.py
    ✎ urls.py
    ✎ wsgi.py
  > media
  > static
  ✓ track
    > migrations
    ✎ admin.py
    ✎ apps.py
    ✎ models.py
    ✎ serializers.py
    ✎ tests.py
    ✎ urls.py
    ✎ views.py
  ✓ user
    > migrations
    ✎ admin.py
    ✎ apps.py
    ✎ models.py
    ✎ permissions.py
    ✎ serializers.py
    ✎ tests.py
    ✎ urls.py
    ✎ views.py
  📁 .dockerignore
  $ .env.example
  ✎ manage.py
  ⌂ requirements.txt
```

2.2.2 Frontend



```
✓ frontend
  > public
  ✓ src
    ✓ api
      TS authApi.ts
      TS axios.ts
      TS trackApi.ts
    > assets/images
    ✓ components
      ✓ ActionbarItems
        ⚒ Player.tsx
        ⚒ TrackDownloadModal.tsx
        ⚒ TrackInfo.tsx
        ⚒ VolumeControl.tsx
      ✓ Artist
        ⚒ VerticalArtist.tsx
      ✓ Navbar
        ⚒ ProfileDropdown.tsx
      ✓ Playlist
        ⚒ HorizontalPlaylist.tsx
        ⚒ HorizontalPlaylistSceleton.tsx
        ⚒ PlaylistCreateModal.tsx
        ⚒ PlaylistDeleteModal.tsx
        ⚒ VerticalPlaylist.tsx
        ⚒ VerticalPlaylistSceleton.tsx
      ✓ Profile
        ⚒ ImageUploadModal.tsx
        ⚒ ProfilePhoto.tsx
      ✓ Search
        ⚒ SearchBar.tsx
      ✓ Track
        ⚒ TrackItem.tsx
        ⚒ TrackList.tsx
      ✓ UI
        ⚒ LikeButton.tsx
        ⚒ PlayButton.tsx
    ✓ containers
      ✓ Actionbar
        ⚒ Actionbar.tsx
```



```
✓ frontend
  ✓ src
    ✓ containers
      ✓ Navbar
        ☈ Navbar.tsx
      ✓ Playlist
        ☈ Playlist.tsx
      ✓ Profile
        ☈ Profile.tsx
      > Search
      ✓ Sidebar
        ☈ Sidebar.tsx
    ✓ pages
      ✓ ArtistPage
        ☈ ArtistPage.tsx
      ✓ GenrePage
        ☈ GenrePage.tsx
      ✓ Home
        ☈ Home.tsx
      ✓ LikedTracksPage
        ☈ LikedTracksPage.tsx
      ✓ Login
        ☈ Login.tsx
      ✓ PlaylistPage
        ☈ PlaylistPage.tsx
      ✓ Register
        ☈ Register.tsx
      ✓ Search
        ☈ Search.tsx
    ✓ shared/utils
      TS playlist-utils.ts
  ✓ store
    ✓ slices
      TS authSlice.ts
      TS notificationSlice.ts
      TS playerSlice.ts
      TS playlistSlice.ts
      TS reduxhooks.ts
      TS store.ts
```

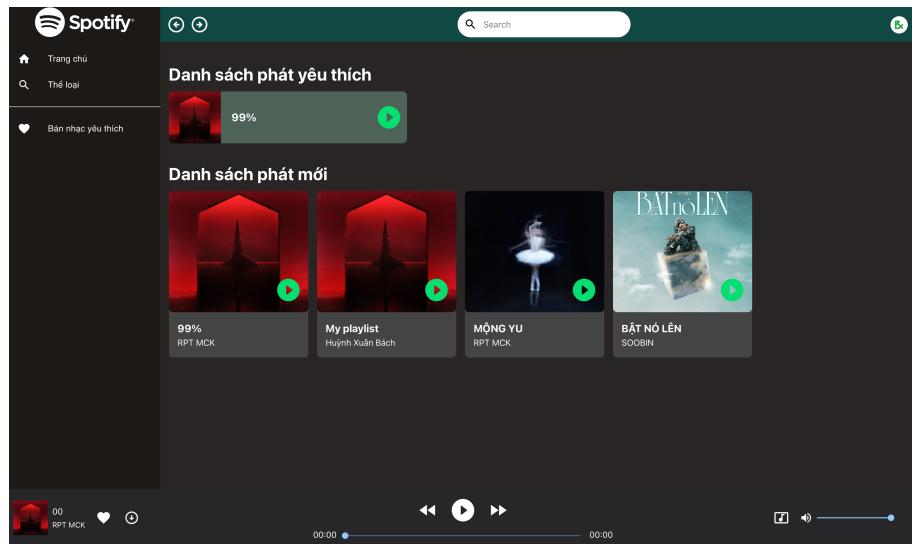


```
✓ frontend
  ✓ src
    ✓ store
      ✓ slices
        TS authSlice.ts
        TS notificationSlice.ts
        TS playerSlice.ts
        TS playlistSlice.ts
        TS reduxhooks.ts
        TS store.ts
      # App.css
      ☀ App.tsx
      # index.css
      🎨 logo.svg
      ☀ main.tsx
      TS types.ts
      TS vite-env.d.ts
    🚗 .dockerignore
    ⚡ .gitignore
    🎯 eslint.config.js
    <> index.html
    { package-lock.json
    { package.json
    ⓘ README.md
    JS tailwind.config.js
    { tsconfig.app.json
    TS tsconfig.json
    { tsconfig.node.json
    ⚡ vite.config.ts
```

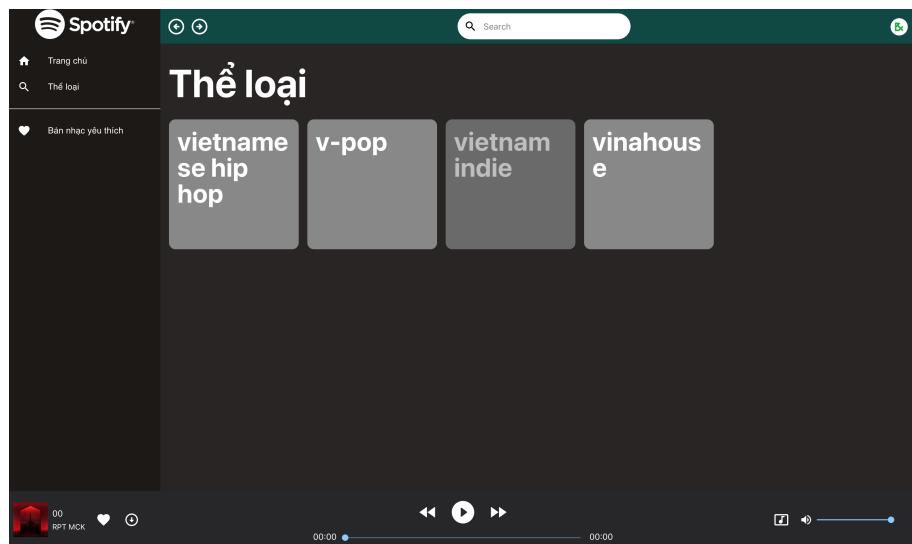
2.3 Giao diện



2.3.1 Trang chủ

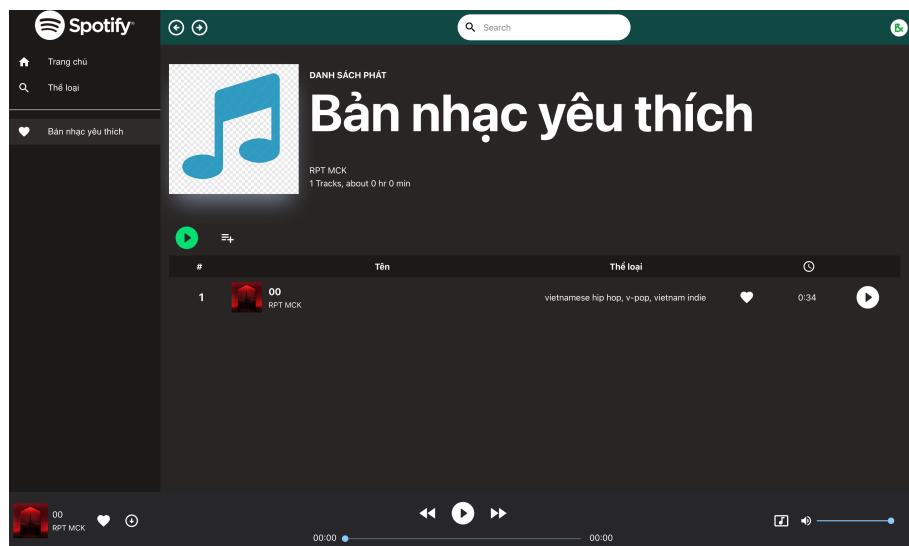


2.3.2 Thể loại

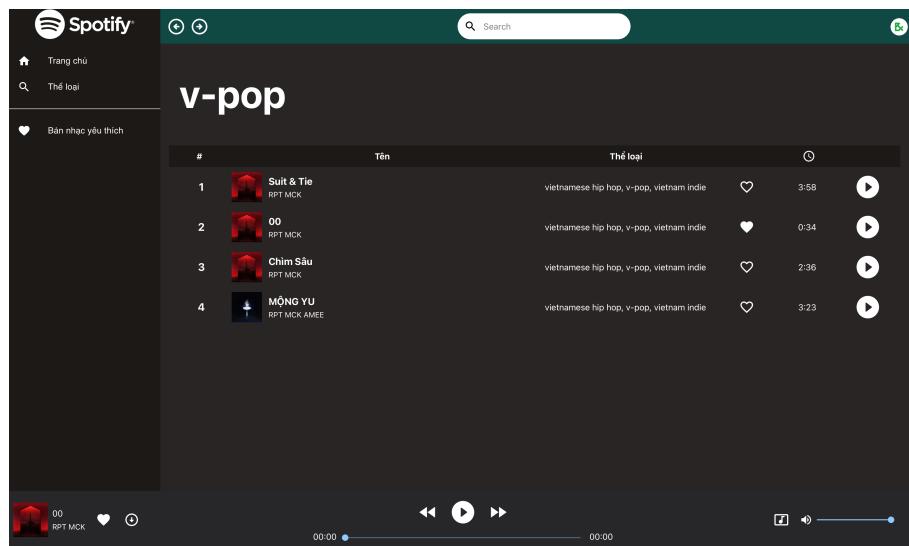




2.3.3 Bản nhạc yêu thích

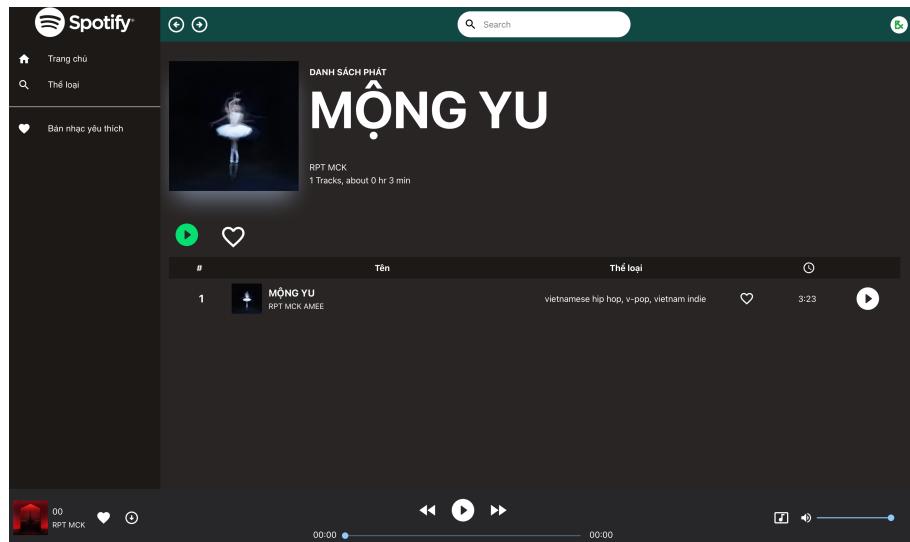


2.3.4 Danh sách phát thể loại

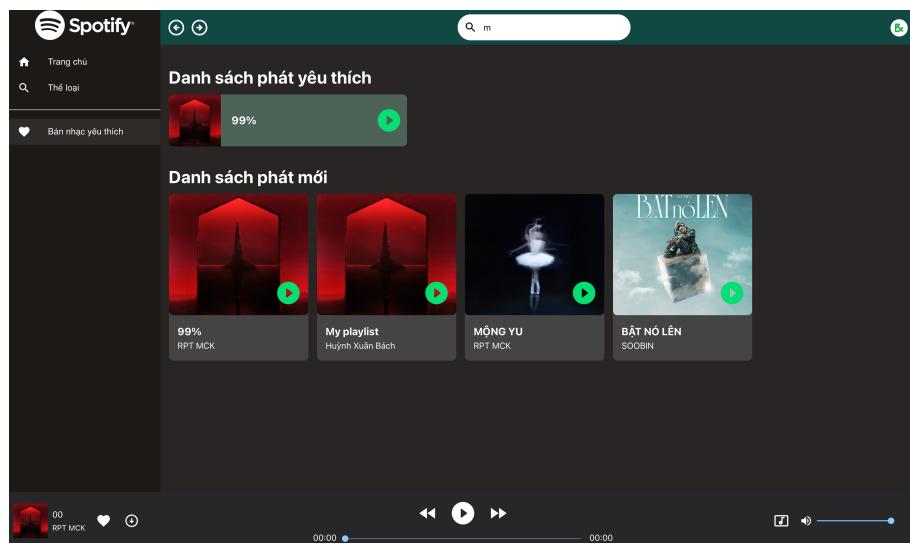




2.3.5 Danh sách phát

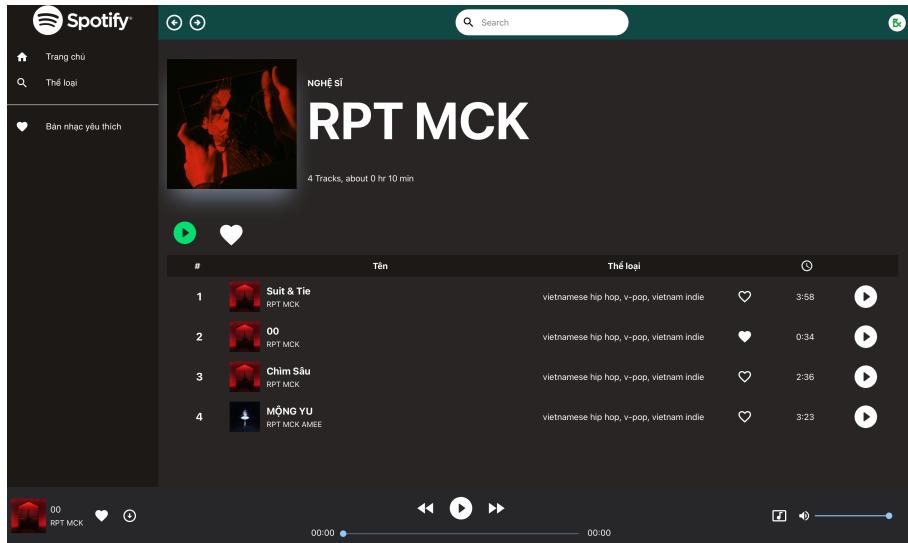


2.3.6 Tìm kiếm



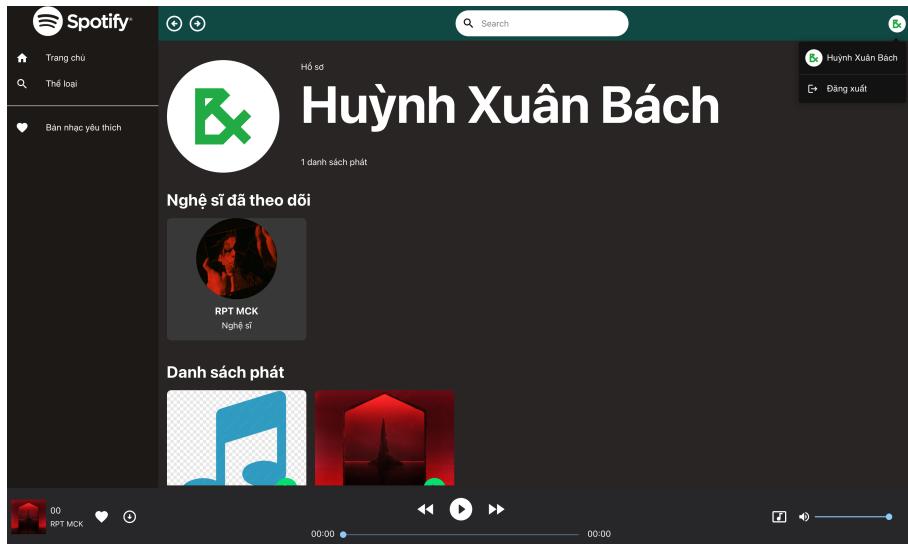


2.3.7 Nghệ sĩ



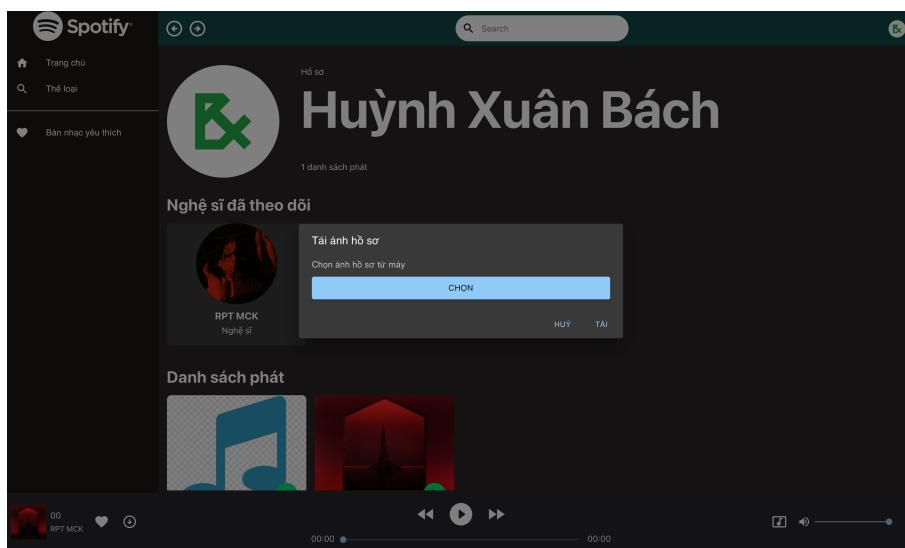
The screenshot shows the Spotify interface for the artist "RPT MCK". At the top, there's a search bar with the text "Search". Below it, the artist's name "RPT MCK" is displayed with a small profile picture. The page indicates there are "4 Tracks, about 0 hr 10 min". A track list follows, showing four songs: "Suit & Tie", "OO", "Chim Sâu", and "MỘNG YU". Each song has a play button, a heart icon for saving, and a timestamp (3:58, 0:34, 2:36, 3:23). The Spotify navigation bar is visible at the bottom.

2.3.8 Hồ sơ

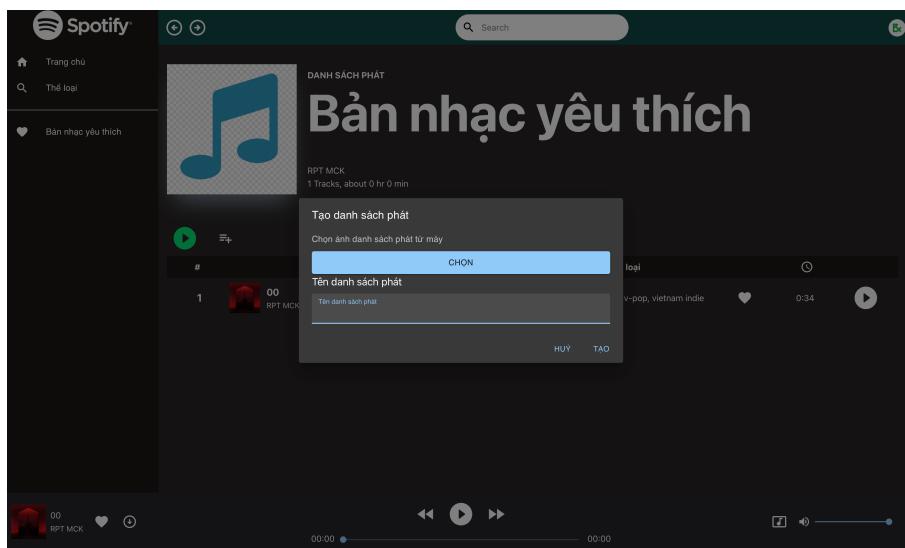


The screenshot shows the Spotify interface for the user "Huỳnh Xuân Bách". The profile picture is a green logo with a white 'X'. The user's name "Huỳnh Xuân Bách" is prominently displayed. It shows "1 danh sách phát" (1 playlist). Below that, it says "Nghệ sĩ đã theo dõi" (Artists followed) and lists "RPT MCK" with the subtitle "Nghệ sĩ". There's also a "Danh sách phát" (Playlists) section with two entries. The Spotify navigation bar is visible at the bottom.

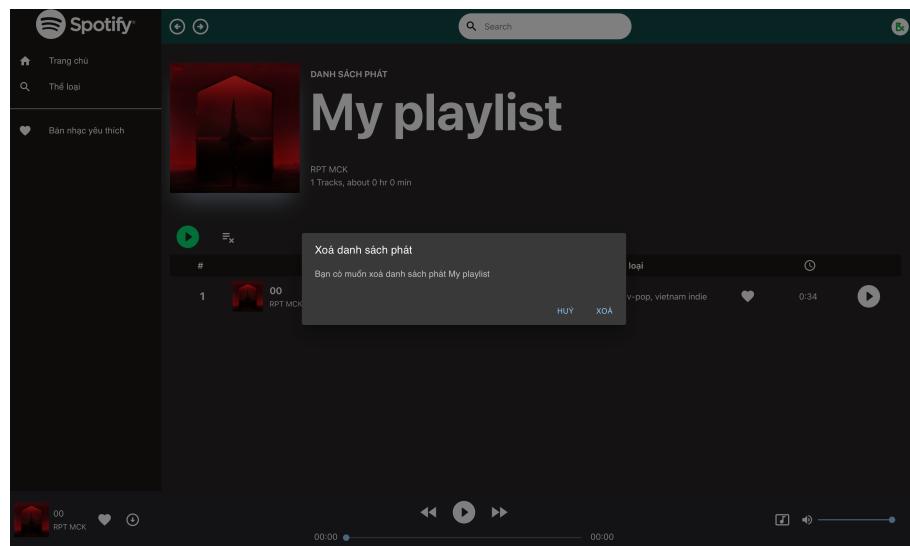
2.3.9 Tải lên ảnh hồ sơ



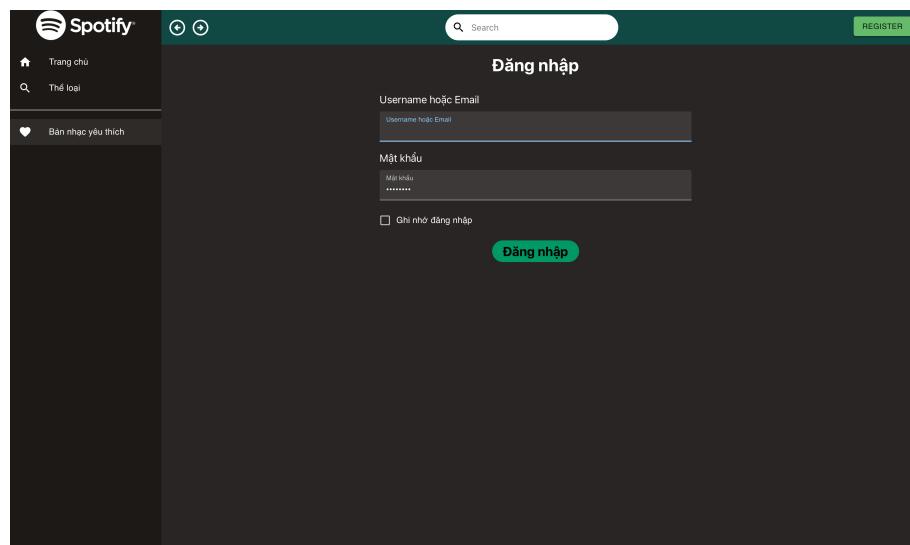
2.3.10 Tạo danh sách phát



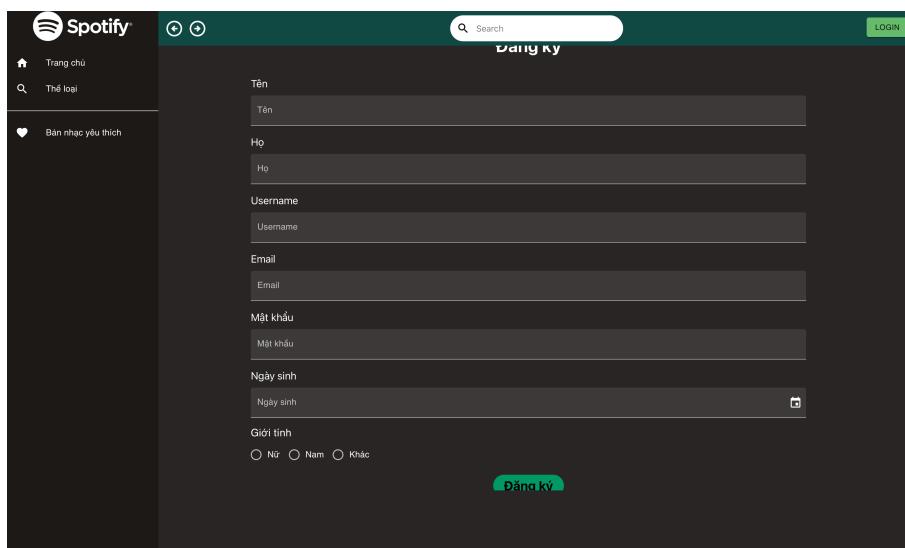
2.3.11 Xoá danh sách phát



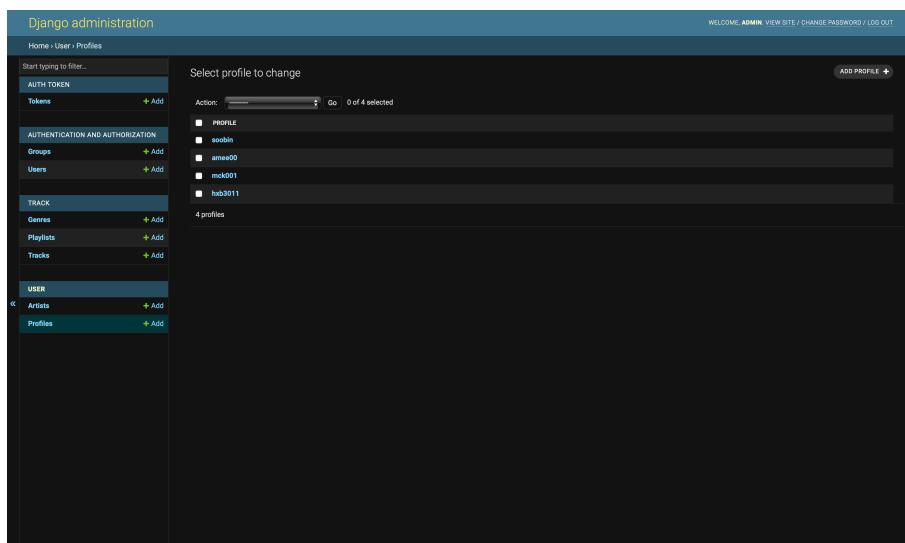
2.3.12 Đăng nhập



2.3.13 Đăng ký



2.3.14 Trang admin - Hồ sơ



2.3.15 Trang admin - Chi tiết hồ sơ



2.3.16 Trang admin - Nghệ sĩ

2.3.17 Trang admin - Chi tiết nghệ sĩ



Django administration

Home > User > Artists > RPT MCK

Start typing to filter...

AUTH TOKEN

Tokens [+ Add](#)

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

TRACK

Genres [+ Add](#)

Playlists [+ Add](#)

Tracks [+ Add](#)

USER

Artists [+ Add](#)

Profiles [+ Add](#)

Change artist

RPT MCK

User: [nsk001](#)

Image: Currently: artistimageab6781610000eb087791c136d7354ad7792356.jpg [Clear](#) [Change](#) Choose File No file selected

Cover: Currently: artistcoverab6781610000eb087791c136d7354ad7792355.TGpmvH.jpg [Clear](#) [Change](#) Choose File No file selected

Verified:

Delete [Save and add another](#) [Save and continue editing](#) [SAVE](#)

2.3.18 Trang admin - Thẻ loại

Django administration

Home > Track > Genres

Start typing to filter...

AUTH TOKEN

Tokens [+ Add](#)

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

TRACK

Genres [+ Add](#)

Playlists [+ Add](#)

Tracks [+ Add](#)

USER

Artists [+ Add](#)

Profiles [+ Add](#)

Select genre to change

Action: —— Go 0 of 4 selected

GENRE

vinehouse

vietnam indie

v-pop

vietnamese hip hop

4 genres

[ADD GENRE](#)

2.3.19 Trang admin - Chi tiết thẻ loại



Django administration

Home > Track > Genres > vinahouse

WELCOME ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT

Change genre

vinahouse

Title: vinahouse

Color: #888888

Delete

Save and add another

Save and continue editing

SAVE

2.3.20 Trang admin - Danh sách phát

Django administration

Home > Track > Playlists

WELCOME ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT

Select playlist to change

Action: —— Go | 0 of selected

ADD PLAYLIST +

PLAYLIST

BẮT NÓ LÊN

MỌNG YÙ

My playlist

99%

4 playlists

2.3.21 Trang admin - Chi tiết danh sách phát



Django administration

Home > Track > Playlists > 99%

Start typing to filter...

AUTH TOKEN

Tokens [Add](#)

AUTHENTICATION AND AUTHORIZATION

Groups [Add](#)

Users [Add](#)

TRACK

Genres [Add](#)

Playlists [Add](#)

Tracks [Add](#)

USER

Artists [Add](#)

Profiles [Add](#)

Change playlist

99%

Title: 99%

Author: mck001

Image: Currently: playlets/image/a64761d0000b273b315e0b7efbe57e9a2b0f01.jpg
Change: Choose File (no file selected)

Color: #888888

Featured

Hide

Delete Save and add another Save and continue editing **SAVE**

2.3.22 Trang admin - Bản nhạc

Django administration

Home > Track > Tracks

Start typing to filter...

AUTH TOKEN

Tokens [Add](#)

AUTHENTICATION AND AUTHORIZATION

Groups [Add](#)

Users [Add](#)

TRACK

Genres [Add](#)

Playlists [Add](#)

Tracks [Add](#)

USER

Artists [Add](#)

Profiles [Add](#)

Select track to change

Action: TRACK Go 0 of 4 selected

TRACK

MỘNG YU

Sult & Tie

Chim Sâu

00

4 tracks

ADD TRACK **+ Go**

2.3.23 Trang admin - Chi tiết bản nhạc



3 Hướng dẫn cài đặt

1. Cài đặt Docker
2. Mở Terminal tại thư mục gốc mã nguồn và chạy lệnh

```
1 docker compose -f assets/docker-compose.yml up -d
```

3. Truy cập trang web phần mềm tại địa chỉ <http://localhost:49084/>
4. Truy cập trang web quản trị (admin) tại địa chỉ <http://localhost:49088/admin/> với user-name và password đều là admin

4 Phân công



STT	Thành viên	Công việc	Dánh giá
1	Huỳnh Xuân Bách	Thể loại, Danh sách phát, Bài nhạc	50%
2	Trần Quốc Sĩ	Người dùng, Hồ sơ người dùng, Nghệ sĩ	50%

Tài liệu

- [1] Django REST framework “<https://www.django-rest-framework.org/>”, .
- [2] Docker Documentation “<https://docs.docker.com/>”, .
- [3] React Documentation “<https://devdocs.io/react/>”, .
- [4] Redux Toolkit “<https://redux-toolkit.js.org/>”, .
- [5] Material UI “<https://mui.com/material-ui/>”, .
- [6] TypeScript “<https://www.typescriptlang.org/docs/>”, .
- [7] Tailwind CSS Documentation “<https://v2.tailwindcss.com/docs/>”, .
- [8] Vite Documentation “<https://vite.dev/guide/>”, .