**TRƯỜNG ĐẠI HỌC SÀI GÒN**

****

BÁO CÁO ĐỒ ÁN

PHÁT TRIỂN PHẦN MỀM NGUỒN MỞ SPOTIFY CLONE

**Thành viên thực hiện:** 3122410240 – Nguyễn Nhật Minh 3122410245 – Triệu Hải Minh

3122410272 – Phan Thanh Tài Nguyên

**Giảng viên hướng dẫn:** Từ Lãng Phiêu

**Email liên hệ nhóm:** [nhatminh0202hc@gmail.com](mailto:nhatminh0202hc@gmail.com)

**Thành phố Hồ Chí Minh, 05/2025**

**Contents**

1. [Lời mở đầu](#_bookmark0) 3
2. [Cơ sở lý thuyết](#_bookmark1) 4
   1. [Công nghệ và thư viện sử dụng](#_bookmark2) 4
      1. [Ngôn ngữ lập trình](#_bookmark3) 4
      2. [Frontend](#_bookmark4) 4
      3. [Backend](#_bookmark5) 4
      4. [Cơ sở dữ liệu](#_bookmark6) 4
      5. [Lưu trữ và dịch vụ đám mây](#_bookmark7) 4
      6. [Công cụ hỗ trợ phát triển](#_bookmark8) 4
   2. [Cấu trúc mã nguồn](#_bookmark9) 5
      1. [Frontend (SolidJS)](#_bookmark10) 5
      2. [Backend (Django)](#_bookmark11) 5
   3. [Mô hình ứng dụng](#_bookmark12) 6
   4. [Các tính năng đã xây dựng](#_bookmark13) 6
      1. [Người dùng](#_bookmark14) 6
      2. [Quản lý âm nhạc](#_bookmark15) 6
      3. [Quản lý hệ thống](#_bookmark16) 6
   5. [Trực quan hóa quy trình bằng Flowchart](#_bookmark17) 7
      1. [Tính năng Thông](#_bookmark18) báo 7
   6. [Thiết kế cơ sở dữ liệu](#_bookmark19) 7
3. [Triển khai và đánh giá hệ thống](#_bookmark20) 9
   1. [Mô tả chức năng và kết quả đạt được](#_bookmark21) 9
      1. [Giao diện chức năng phía người dùng](#_bookmark22) 9
      2. [Giao diện chức năng phía quản trị viên](#_bookmark23) 20
   2. [Kết chương](#_bookmark24) 24
4. [Cài đặt và môi trường chạy](#_bookmark25) 25
   1. [Yêu cầu hệ thống](#_bookmark26) 25
   2. [Cài đặt backend](#_bookmark27) 25
   3. [Cài đặt frontend](#_bookmark28) 25
5. [Vai trò các thành viên](#_bookmark30) 26

# Lời mở đầu

Giữa kỷ nguyên số hóa, nhu cầu thưởng thức âm nhạc trực tuyến đã trở thành một phần không thể thiếu trong nhịp sống hiện đại. Nắm bắt xu hướng đó, dự án SpotifyClone ra đời với mục tiêu tái hiện những trải nghiệm âm nhạc cốt lõi mà nền tảng Spotify đã định hình. Ứng dụng web này được xây dựng không chỉ để mô phỏng các tính năng cơ bản, mà còn hướng đến việc tạo ra một không gian âm nhạc trực tuyến toàn diện.

SpotifyClone mang đến một hành trình âm nhạc liền mạch, từ việc đăng ký tài khoản cá nhân, khám phá thư viện nhạc phong phú thông qua chức năng tìm kiếm thông minh, tạo và quản lý những playlist mang đậm dấu ấn riêng, cho đến việc không ngừng mở rộng chân trời âm nhạc bằng tính năng khám phá nhạc mới. Dự án được phát triển dựa trên sự kết hợp mạnh mẽ của JavaScript cho giao diện người dùng trực quan, Python cùng framework Django mạnh mẽ cho phần backend, và MongoDB – một cơ sở dữ liệu linh hoạt được quản lý hiệu quả thông qua MongoDB Compass.

Mục tiêu chiến lược của SpotifyClone không chỉ dừng lại ở việc xây dựng một hệ thống phát nhạc trực tuyến hoàn chỉnh với giao diện thân thiện. Chúng tôi còn tập trung vào việc kiến tạo một trải nghiệm nghe nhạc mượt mà, ổn định trên nền tảng web, đồng thời đảm bảo khả năng mở rộng và hiệu suất tối ưu cho hệ thống. Dự án này còn là cơ hội để ứng dụng sâu rộng kiến thức phát triển ứng dụng fullstack, xây dựng một hệ thống quản lý người dùng và nội dung âm nhạc hiệu quả, và đặt nền móng vững chắc cho việc phát triển các tính năng nâng cao trong tương lai.

# Cơ sở lý thuyết

## Công nghệ và thư viện sử dụng

### Ngôn ngữ lập trình

**Python:** Backend của dự án được xây dựng trên Python, một ngôn ngữ nổi tiếng với cú pháp đơn giản, dễ học và một cộng đồng hỗ trợ nhiệt tình. Điểm mạnh vượt trội của Python còn nằm ở kho thư viện đồ sộ, cung cấp giải pháp toàn diện cho mọi nhu cầu backend, từ xử lý dữ liệu chuyên sâu, phát triển API hiện đại, đến khả năng kết nối linh hoạt với các cơ sở dữ liệu.

**JavaScript:** Sử dụng cho phần frontend nhằm xây dựng giao diện người dùng hiện đại, linh hoạt và có khả năng phản hồi nhanh.

### Frontend

* HTML5/CSS3/JavaScript: Ngôn ngữ cơ bản để xây dựng giao diện người dùng
* Solid.js: Thư viện JavaScript để xây dựng giao diện người dùng tương tác
* HTML5 Audio API: là một tập hợp các tính năng của HTML5 cho phép bạn **phát, điều khiển và thao tác âm thanh trên trình duyệt** mà **không cần plugin** (như Flash). Nó được xây dựng dựa trên phần tử <audio> trong HTML5. Kết hợp với **SolidJS** để điều khiển trạng thái phát (play, pause, seek, volume, loop, v.v.).

### Backend

* + - * Python: Ngôn ngữ lập trình server-side
      * Django: Framework Python cho phát triển web

### Cơ sở dữ liệu

* + - * MongoDB: Hệ quản trị cơ sở dữ liệu quan hệ

### Lưu trữ và dịch vụ đám mây

Toàn bộ media (ảnh, nhạc, video) được lưu trữ trên hệ thống server nội bộ. Trong các bản nâng cấp, nhóm dự kiến tích hợp các dịch vụ cloud như Firebase Storage hoặc Amazon S3 để tối ưu hoá hiệu năng và tính mở rộng.

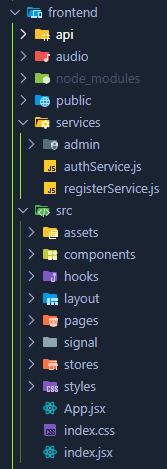
### Công cụ hỗ trợ phát triển

* + - * **Visual Studio Code**: trình soạn thảo mã nguồn chính.
      * **Git + GitHub**: quản lý phiên bản mã nguồn.
      * **MongoDB Compass**: thao tác cơ sở dữ liệu.

## Cấu trúc mã nguồn

### Frontend (SolidJS)

Mã nguồn Solid được chia thành các module:

* + - * **components/**: chứa các thành phần giao diện dùng chung.
      * **pages/**: các trang tương ứng với từng chức năng.
      * **store/**: quản lý trạng thái toàn cục bằng Redux.
      * **assets/**: chứa ảnh, icon, video,...
      * **api/:** gọi đến server

1

### Backend (Django)

Backend Django gồm các app như:

* **users**: xử lý người dùng, đăng ký, đăng nhập, thông tin cá nhân.
* **albums, songs, artists, artistFollow, ArtistOfSong, Playlists, Album, SongInAlbum, SongInPlaylist**: quản lý dữ liệu âm nhạc.
* **Notification:** Để lưu trữ thông báo
* A screenshot of a computer

  AI-generated content may be incorrect.**History:** Để lưu trữ lịch sử

## Mô hình ứng dụng

Hệ thống được xây dựng theo mô hình client-server, trong đó:

* **Frontend (SolidJS)** đóng vai trò là client, chịu trách nhiệm hiển thị giao diện người dùng và gửi yêu cầu đến máy chủ thông qua các API.
* **Backend (Python - Django)** đóng vai trò là server, xử lý các yêu cầu từ phía client, thực hiện các logic nghiệp vụ và truy xuất dữ liệu từ cơ sở dữ liệu.
* **Cơ sở dữ liệu (MongoDB)** lưu trữ thông tin người dùng, sản phẩm, giao dịch,... và được backend truy cập thông qua các truy vấn.

Luồng xử lý cơ bản của ứng dụng như sau:

1. Người dùng tương tác với giao diện frontend.
2. Frontend gửi request đến backend thông qua các API.
3. Backend xử lý yêu cầu, truy vấn hoặc cập nhật dữ liệu trong MongoDB.
4. Backend trả kết quả về cho frontend.
5. Frontend hiển thị dữ liệu hoặc phản hồi tương ứng cho người dùng.

Mô hình này giúp tách biệt giữa giao diện và xử lý logic, tạo điều kiện thuận lợi cho việc bảo trì, mở rộng và phát triển độc lập giữa frontend và backend.

## Các tính năng đã xây dựng

### Người dùng

* + - * Đăng ký, đăng nhập, chỉnh sửa thông tin cá nhân.
      * Quản lý bài hát yêu thích, danh sách phát.
      * Có thể đăng kí thành nghệ sĩ và upload được các bài nhạc
      * Theo dõi ca sĩ yêu thích
      * Nếu lên nghệ sĩ có thể tự tạo thêm album với các bài hát cá nhân đã được đăng tải

### Quản lý âm nhạc

* + - * Nghe nhạc, tìm kiếm bài hát, xem lịch sử đã nghe
      * Hiển thị thông tin bài hát,album.

### Quản lý hệ thống

* + - * Tạo, sửa, xoá bài hát, ca sĩ, người dùng.

**2.4.4 Chức năng thông báo 2 chiều**

Người dùng đã theo dõi ca sĩ có thể nhận thông tin về các bài nhạc mới nhất.

Nghệ sĩ khi có người theo dõi mới sẽ được thông báo

## Trực quan hóa quy trình bằng Flowchart

### Tính năng Thông báo

Flowchart biểu diễn quy trình gửi – nhận tin nhắn giữa hai người dùng:

* + - * Người dùng theo dõi → frontend gửi yêu cầu POST đến backend.
      * Backend lưu thông báo vào database →cập nhật real-time.
      * Thông báo sẽ hiển thị ở giao diện người nhận.

## A screenshot of a computer AI-generated content may be incorrect.Thiết kế cơ sở dữ liệu

Sơ đồ thiết kế giao diện:

Figure 1: Sơ đồ Entity-Relationship (ER)

**Bảng User**

* id: Khóa chính, định danh người dùng
* full\_name: Tên người dùng
* email: Email người dùng (duy nhất)
* password: Mật khẩu đã hash
* avatar\_url: Thời gian tạo tài khoản
* status: Trạng thái tài khoản

**Bảng Song**

* id: Khóa chính, định danh bài hát
* song\_name: Tên bài hát
* duration: Thời lượng bài hát (giây)
* song\_url: Đường dẫn đến file âm thanh
* picture\_url: Khóa ngoại liên kết với Albums
* release\_date: Ngày đăng
* listen\_count: Lượt nghe
* artist: Tham chiếu tới đối tượng trong bảng Artist

**Bảng Album**

* id: Khóa chính, định danh album
* album\_name: Tên album
* artist: Tham chiếu tới đối tượng trong bảng Artist
* release\_date: Ngày phát hành
* album\_picture: Đường dẫn đến ảnh bìa album

**Bảng Artist**

* id: Khóa chính, định danh nghệ sĩ
* artist\_name: Tên nghệ sĩ
* description: Tiểu sử nghệ sĩ
* user: Tham chiếu tới đối tượng người dùng trong bảng User
* country: Thành phố
* active\_years: Năm tài khoản được kích hoạt

**Bảng Playlist**

* id: Khóa chính, định danh playlist
* playlist\_name: Tên playlist
* user: Tham chiếu tới đối tượng trong bảng Artist
* playlist\_picture: Ảnh bìa playlist

**Bảng ArtistFollow**

* id: Khóa chính
* artist: Tham chiếu tới 1 đối tượng trong bảng Artist
* user: Tham chiếu tới 1 đối tượng trong bảng User

**Bảng ArtistOfSong**

* id: Khóa chính
* artist: Tham chiếu tới 1 đối tượng trong bảng Artist
* song: Tham chiếu tới 1 đối tượng trong bảng Song

**Bảng ArtistRequest**

* id: Khóa chính
* user: Tham chiếu tới 1 đối tượng trong bảng Artist
* status: Trạng thái của đơn đợi duyệt lên ca sĩ
* request\_date: Ngày đăng
* link\_confirm: Minh chứng xác nhận

**Bảng AuthToken**

* id: Khóa chính
* token: Token lưu trữ phiên đăng nhập cho người dùng
* user: Người được phát token

**Bảng FavoriteSong**

* id: Khóa chính
* user: Tham chiếu tới người dùng đã thích bài này
* song Tham chiếu tới bài hát được yêu thích

**Bảng History**

* id: Khóa chính
* user: Tham chiếu tới người dùng đã nghe bài này
* song Tham chiếu tới bài hát được nghe
* time: Thời gian nghe

**Bảng Notification**

* id: Khóa chính
* status: Trạng thái đã đọc hay chưa
* description: nội dung tin nhắn được gửi
* time: Thời gian gửi
* img: Hình ảnh bài hát đươc gửi
* user: Tham hiếu tới đối tượng user đượ gửi ới
* artist: Tham chiếu tới nghệ sĩ của bài hát

**Bảng SongInAlbum**

* id: Khóa chính
* album: Tham chiếu tới đối tượng album trong bảng Album
* song Tham chiếu tới bài hát được thêm vào album

**Bảng SongInPlaylist**

* id: Khóa chính
* playlist: Tham chiếu tới đối tượng playlist trong bảng Playlist
* song Tham chiếu tới bài hát được thêm vào album

**Bảng SongRequset**

* id: Khóa chính
* status: Trạng thái đã đọc hay chưa
* release\_date: Ngày phát hành
* duration: Thời gian của bài nhạc được tính theo s
* picture\_url: Hình ảnh của bài hát
* song\_url: đường dẫn của nhạc
* artist: Tham hiếu tới đối tượng artist là người gửi bài hát
* cover\_artists: Mảng chứa các đội tượng artist khác có tham gia bài hát
* title: Tên bài hát

# Triển khai và đánh giá hệ thống

## Mô tả chức năng và kết quả đạt được

### Giao diện chức năng phía người dùng

**1.1 Trang chủ** Trang chủ hiển thị các mục nổi bật: bài hát thịnh hành, danh sách phát đề xuất, album mới ra mắt. Dữ liệu được lấy từ backend và hiển thị động bằng React.

A screenshot of a music album

AI-generated content may be incorrect.

Figure 2: Giao diện trang chủ Spotify Clone khi chưa đăng nhập

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3: Giao diện trang chủ Spotify Clone khi chưa đăng nhập

A screenshot of a music player

AI-generated content may be incorrect.

Figure 4: Giao diện trang chủ Spotify Clone khi đăng nhập

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5: Giao diện trang chủ Spotify Clone khi đăng nhập

**1.3 Chi tiết album** Giao diện chi tiết hiển thị thông tin album. Người dùng có thể phát nhạc, thêm vào danh sách yêu thích hoặc danh sách phát.

A screenshot of a video player

AI-generated content may be incorrect.

Figure 6: Chi tiết album

**1.4 Chức năng nghe nhạc** Hỗ trợ phát nhạc trực tuyến với các chức năng: phát / tạm dừng, chuyển tiếp, Chuyển sang bài hát tiếp theo trong album, thời gian chạy, và thích hoặc thêm vào playlist



Figure 7: Nghe nhạc

* 1. **Bài hát yêu thích** Người dùng có thể thêm / xoá bài hát khỏi danh sách yêu thích. Danh sách này được lưu trong cơ sở dữ liệu.

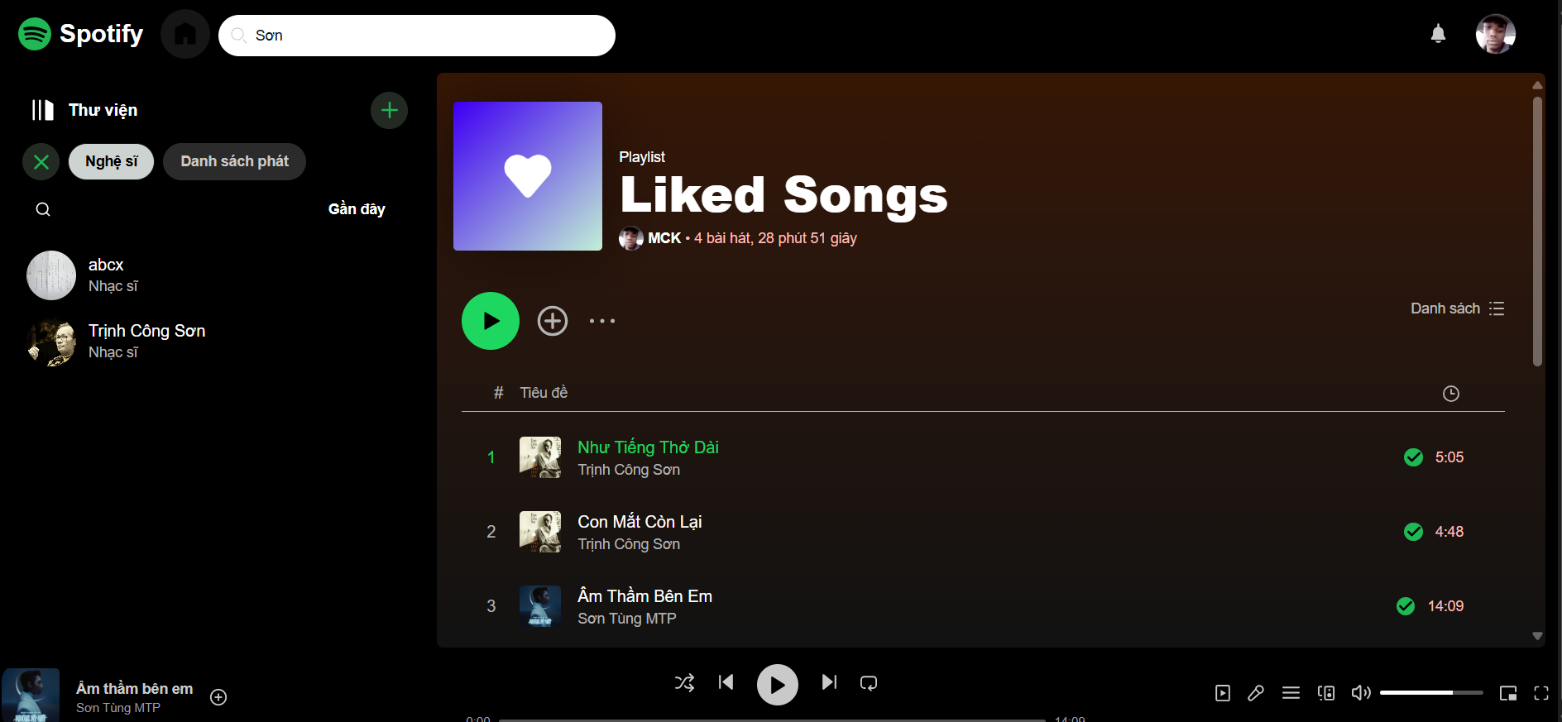


Figure 9: Bài hát yêu thích

A screenshot of a phone

AI-generated content may be incorrect.**1.5 Giao diện like nhạc hoặc thêm nhạc vào playlist** Cho phép quản lú các bài hát nằm trong playlist nào hoặc unlike.

* 1. **Danh sách phát nhạc** Cho phép tạo nhiều playlist, thêm hoặc xoá bài hát trong từng danh sách

A screenshot of a chat

AI-generated content may be incorrect.A screenshot of a video player

AI-generated content may be incorrect.

Figure 10: tạo playlist

A screenshot of a music player

AI-generated content may be incorrect.

Figure 10: Playlist khi mới được tạo

A screenshot of a phone

AI-generated content may be incorrect.

Figure 11: Quản lý các playlist

* 1. **Bảng xếp hạng** Hiển thị bảng xếp hạng các bài hát được phát nhiều nhất.

A collage of a person playing a guitar

AI-generated content may be incorrect.

* 1. **Bảng xếp hạng** Hiển thị các ca sĩ có lượt follow cao nhất

A person in a suit and tie

AI-generated content may be incorrect.

* 1. **Chức năng tìm kiếm** Tìm kiếm theo tên bài hát, ca sĩ. Gợi ý kết quả theo từ khoá nhập vào.

A screenshot of a phone

AI-generated content may be incorrect.

Figure 12: Giao diện tìm kiếm

* 1. A screenshot of a login box

     AI-generated content may be incorrect.**Chức năng đăng ký và đăng nhập** Giao diện đăng nhập có kiểm tra hợp lệ và thông báo lỗi rõ ràng.

Figure 13: Giao diện đăng nhập

A screenshot of a login box

AI-generated content may be incorrect.

Figure 14: Giao diện đăng ký

* 1. **Chức năng xem và chỉnh sửa thông tin cá nhân** Người dùng có thể thay đổi tên hiển thị, email, mật khẩu, ảnh đại diện. Thông tin được đồng bộ với cơ sở dữ liệu.

A screenshot of a video chat

AI-generated content may be incorrect.

Figure 15: Giao diện thông tin cá nhân

A screenshot of a social media post

AI-generated content may be incorrect.

Figure 16: Giao diện đăng kí nghệ sĩ

A screenshot of a video chat

AI-generated content may be incorrect.

Figure 16: Giao diện chỉnh sửa thông tin cá nhân

A screenshot of a video chat

AI-generated content may be incorrect.

Figure 17: Giao diện thông tin cá nhân đối với nghệ sĩ

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a music album

AI-generated content may be incorrect.

Figure 17: Giao diện thêm bài hát mới và album mới

A screenshot of a phone

AI-generated content may be incorrect.

Figure 17: Giao diện chỉnh sửa thông tin đối với nghệ sĩ

**1.12 Chức năng thông báo trực tuyến** Cho phép người dùng thông báo trực tiếp trong thời gian thực khi có người dùng theo dõi hoặc nghệ sĩ đẩy bài hát mới.

A screenshot of a phone

AI-generated content may be incorrect.

Figure 17: Giao diện thông báo

### Giao diện chức năng phía quản trị viên

* 1. **Tổng quan** Duyệt bài hát. Sẽ có chức năng duyệt các hồ sơ ứng tuyển ca sĩ

A screenshot of a computer

AI-generated content may be incorrect.

Figure 20: Giao diện tổng quan

A screenshot of a phone

AI-generated content may be incorrect.

Figure 21: Giao diện khi duyệt

* 1. A screenshot of a computer

     AI-generated content may be incorrect.**Duyệt ca sĩ:** Duyệt các ca sĩ có đơn ứng tuyển

Figure 22: Giao diện quản lý tài khoản

A screenshot of a computer

AI-generated content may be incorrect.**2.7 Chức năng quản lý người dùng** Quản lý người dùng với chức năng xóa người dùng và block

**2.7 Chức năng quản lý ca sĩ:** Quản lý người dùng với chức năng xóa người dùng và block

A screenshot of a computer

AI-generated content may be incorrect.

Figure 23: Giao diện nghệ sĩ

**2.10 Chức năng quản lý Bài hát** Quản lý toàn bộ bài hát trong hệ thống: xoá

A screenshot of a computer

AI-generated content may be incorrect.

Figure 24: Giao diện quản lý bài hát

## Kết chương

Chương này đã trình bày chi tiết các chức năng chính đã được xây dựng trong hệ thống, từ phía người dùng đến quản trị viên. Qua quá trình triển khai, nhóm đã hoàn thiện nhiều chức năng thiết yếu, đảm bảo trải nghiệm người dùng tốt, đồng thời cung cấp giao diện quản trị đầy đủ để điều hành và mở rộng hệ thống trong tương lai.

# Cài đặt và môi trường chạy

## Yêu cầu hệ thống

* Hệ điều hành: Windows hoặc Ubuntu
* Python 3.9 trở lên
* NodeJS 18 trở lên
* MongoDB
* Trình duyệt hỗ trợ HTML5

## Cài đặt backend

1 git clone https://github.com/ngnhminh/NhacCuaTao.git

2 cd backend

3 virtualenv myenv hoặc python -m virtualenv myenv

4 myenv/ Scripts/ activate

5 pip install -r requirements.txt hoặc python -m pip install -r requirements. txt

7 daphne backend.asgi:application

8 Backend trang <http://127.0.0.1:8000/>

## Cài đặt frontend

1 git clone https://github.com/ngnhminh/NhacCuaTao.git

2 cd frontend

3 npm install

4 npm run dev

# Vai trò các thành viên

|  |  |  |
| --- | --- | --- |
| **Thành viên** | **Giao diện người dùng** | **Giao diện Admin** |
| Nguyễn Nhật Minh | Backend | Backend |
| Triệu Hải Minh | Frontend | Frontend |
| Phan Thanh Tài Nguyên | Frontend | Frontend |