**A white rectangular frame with black border

AI-generated content may be incorrect.**

TRƯỜNG ĐẠI HỌC SÀI GÒN

A blue circle with white text

AI-generated content may be incorrect.

A blue circle with white text

AI-generated content may be incorrect.

BÁO CÁO ĐỒ ÁN MÔN HỌC

PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

WEB NGHE NHẠC SPOTIFY

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn: | | Ths.Từ Lãng Phiêu |
| Nhóm thực hiện: |  | Nhóm 19 |
| Thành viên: |  |  |
| Ngô Thanh Tân | 3121410442 |  |
| Trần Hải Dương | 3121410131 |  |
| Lê Nguyễn Nhất Tâm | 3122410369 |  |

# 

# LỜI CẢM ƠN

Lời cảm ơn đầu tiên chúng em xin gửi đến Thầy Phiêu cùng các quý thầy cô khoa Công nghệ Thông tin trường Đại học Sài Gòn đã truyền dạy kiến thức cho chúng em trong thời gian qua để chúng em có thể hoàn thành quá trình nghiên cứu và thực hiện đề tài đồ án.

Dù đã nỗ lực hết mình, nhưng với điều kiện thời gian hạn chế và kinh nghiệm còn thiếu sót, bài báo cáo của chúng em không tránh khỏi những thiếu sót. Chúng em hi vọng nhận được sự đóng góp và chia sẻ từ quý Thầy Cô cùng các bạn để chúng em có thể hoàn thiện hơn, nâng cao kiến thức và kỹ năng cho hành trang sự nghiệp và học tập trong tương lai.

Cuối cùng chúng em kính chúc quý thầy cô dồi dào sức khỏe, niềm tin để tiếp tục sự nghiệp cao quý của mình là truyền đạt kiến thức cho thế hệ mai sau.

Chúng em xin chân thành cảm ơn. Trân trọng.

TP. Hồ Chí Minh, ngày 05/05/2025 Nhóm sinh viên thực hiện

Toàn thể thành viên nhóm

# Mục lục

[LỜI CẢM ƠN 1](#_Toc197379337)

[Mục lục 2](#_Toc197379338)

[1.Giới thiệu dự án 4](#_Toc197379339)

[1.1.Bối cảnh thực tiễn 4](#_Toc197379340)

[1.2.Lý do chọn đề tài 4](#_Toc197379341)

[1.3.Mục tiêu của dự án 5](#_Toc197379342)

[1.4.Đặc điểm kỹ thuật 5](#_Toc197379343)

[1.5.Đối tượng sử dụng 5](#_Toc197379344)

[2.Cơ sở lý thuyết 6](#_Toc197379345)

[2.1.Cấu trúc phần mềm 6](#_Toc197379346)

[2.1.1.Frontend (Client) 6](#_Toc197379347)

[2.1.2.Backend (Server) 6](#_Toc197379348)

[2.1.3.Các mẫu kiến trúc áp dụng 6](#_Toc197379349)

[2.2.Luồng dữ liệu hệ thống 7](#_Toc197379350)

[2.2.1.Luồng dữ liệu RESTful API 7](#_Toc197379351)

[2.Frontend gửi yêu cầu đến Backend: 7](#_Toc197379352)

[3.Backend xử lý yêu cầu: 7](#_Toc197379353)

[4.Frontend nhận và xử lý phản hồi: 7](#_Toc197379354)

[2.3.Công nghệ, thư viện và framework được sử dụng 8](#_Toc197379355)

[2.3.1.Frontend 8](#_Toc197379356)

[2.3.2.Backend 8](#_Toc197379357)

[2.4.Mô tả cơ sở dữ liệu 9](#_Toc197379358)

[2.5.Cấu trúc mã nguồn 10](#_Toc197379359)

[2.5.1.Backend 11](#_Toc197379360)

[2.5.2.Frontend 12](#_Toc197379361)

[3.Giao diện ứng dụng 13](#_Toc197379362)

[4.Cách thức cài đặt ứng dụng 16](#_Toc197379363)

[4.1.Yêu cầu hệ thống 16](#_Toc197379364)

[4.2.Cài đặt Backend 16](#_Toc197379365)

[1.Clone repository: 16](#_Toc197379366)

[2.Tạo và kích hoạt môi trường ảo: 17](#_Toc197379367)

[3.Cài đặt dependencies: 17](#_Toc197379368)

[4.Tạo file .env: 17](#_Toc197379369)

[5.Thực hiện các migration: 17](#_Toc197379370)

[6.Tạo superuser (tùy chọn): 17](#_Toc197379371)

[7.Chạy server development 17](#_Toc197379372)

[4.3.Cài đặt Frontend 18](#_Toc197379373)

[1.Di chuyển đến thư mục frontend: 18](#_Toc197379374)

[2.Cài đặt dependencies: 18](#_Toc197379375)

[4.Chạy server development: 18](#_Toc197379376)

[5.Vai trò của từng thành viên 19](#_Toc197379377)

[1.Ngô Thanh Tân (3121410442) 19](#_Toc197379378)

[2.Trần Hải Dương (3121410131) 19](#_Toc197379379)

[3.Lê Nguyễn Nhất Tâm (3122410369) 19](#_Toc197379380)

# Giới thiệu dự án

## Bối cảnh thực tiễn

Trong thời đại kỹ thuật số hiện nay, nhu cầu giải trí trực tuyến ngày càng gia tăng mạnh mẽ. Các cách thức nghe nhạc cổ điển như mua DVD hay các băng cát xét dần trở nên lỗi thời mà thay vào đó là những nền tảng nghe nhạc trực tuyến như Spotify, Apple Music, và Zing MP3,... Nó không chỉ mang đến trải nghiệm âm nhạc phong phú mà còn giúp kết nối cộng đồng yêu nhạc trên khắp thế giới.

## Lý do chọn đề tài

Hiện nay, người dùng thường gặp khó khăn trong việc tìm kiếm các bài hát mới, chất lượng âm thanh trên các nền tảng trực tuyến không ổn định và việc quản lý thư viện nhạc cá nhân còn nhiều hạn chế. Điều này dẫn đến trải nghiệm người dùng không tốt và làm giảm sự hài lòng của khách hàng.

Nhận thấy nhu cầu rất lớn của mọi người trong việc nghe nhạc một cách dễ dàng và tiện lợi hơn, nhóm chúng tôi quyết định xây dựng một web nghe nhạc, ở đó người dùng có thể nghe

## Mục tiêu của dự án

Dự án này là một ứng dụng nghe nhạc hoàn chỉnh cho phép:

* Phát nhạc và video nhạc
* Tìm kiếm bài hát, album, nghệ sĩ
* Đăng ký và đăng nhập người dùng
* Tạo và quản lý danh sách phát cá nhân
* Theo dõi nghệ sĩ yêu thích
* Hiển thị các bài hát phổ biến, mới ra mắt
* Hiển thị lời bài hát
* Quản lý hồ sơ người dùng
* Đề xuất nội dung âm nhạc phù hợp với sở thích người dùng
* Trò chuyện với người dùng khác

## Đặc điểm kỹ thuật

Hệ thống sử dụng mô hình Client-Server, với các công nghệ sau:

* **Frontend**: ReactJS, Redux, Vite, Axios, Bootstrap
* **Backend**: Django, Django REST Framework
* **Database**: SQL (SQLite trong môi trường phát triển, PostgreSQL cho sản phẩm)
* **Authentication**: JWT (JSON Web Tokens)
* **Media Processing**: FFmpeg
* **Deployment**: Docker, Nginx

## Đối tượng sử dụng

Dự án hướng đến:

* Người yêu âm nhạc muốn thưởng thức nhạc trực tuyến
* Sinh viên hoặc cộng đồng có cùng đam mê âm nhạc với nhau
* Người dùng có nhu cầu tìm kiếm, phát hiện và chia sẻ nội dung âm nhạc mới
* Người muốn tạo và quản lý danh sách phát cá nhân

# Cơ sở lý thuyết

## Kiến trúc phần mềm

Dự án được xây dựng dựa trên kiến trúc Client-Server, một mô hình phổ biến trong phát triển ứng dụng web hiện đại. Kiến trúc này phân tách rõ ràng giữa giao diện người dùng (Client) và phần xử lý logic nghiệp vụ, quản lý dữ liệu (Server).

### Frontend (Client)

Frontend được xây dựng bằng ReactJS, một thư viện JavaScript hiệu suất cao để xây dựng giao diện người dùng. Chúng tôi sử dụng Vite làm công cụ build hiện đại giúp tăng tốc quá trình phát triển.

Các thành phần chính của Frontend:

* + - * **ReactJS**: Thư viện JavaScript để xây dựng UI dựa trên component
      * **Rx**: Quản lý trạng thái ứng dụng hiệu quả
      * **Axios**: Thư viện HTTP client để gửi các yêu cầu đến API của Backend
      * **Bootstrap**: Framework CSS cho thiết kế giao diện
      * **React Router**: Quản lý điều hướng trong ứng dụng Frontend giao tiếp với Backend thông qua RESTful API.

### Backend (Server)

Backend được phát triển bằng Django, một framework Python mạnh mẽ, an toàn và linh hoạt. Django REST Framework được sử dụng để xây dựng API.

Các thành phần chính của Backend:

* + - * **Django**: Framework web Python
      * **Django REST framework**: Xây dựng RESTful API
      * **JWT Authentication**: Xác thực người dùng bằng JSON Web Tokens
      * **SQLite/PostgreSQL**: Hệ thống quản lý cơ sở dữ liệu
      * **FFmpeg**: Xử lý và biến đổi các tệp âm thanh và video

### Các mẫu kiến trúc áp dụng

* + - * **Model-View-Controller (MVC)**: Mẫu kiến trúc cơ bản được Django áp dụng (mặc dù Django gọi là MVT - Model-View-Template)
      * **RESTful API**: Thiết kế API tuân thủ các nguyên tắc REST
      * **Redux Pattern**: Quản lý trạng thái ứng dụng theo luồng dữ liệu một chiều ở phía client
      * **Component-Based Architecture**: Phát triển UI dựa trên các thành phần độc lập và tái sử dụng
      * **Repository Pattern**: Trừu tượng hóa việc truy cập dữ liệu

## Luồng dữ liệu hệ thống

Hệ thống hoạt động dựa trên mô hình Client-Server, với luồng dữ liệu chủ yếu thông qua HTTP/RESTful API.

### Luồng dữ liệu RESTful API

1. **Người dùng tương tác với Frontend**:
   * Người dùng thực hiện hành động trên giao diện (nhấn nút Play, tìm kiếm bài hát, đăng nhập, v.v.)
   * Component React xử lý sự kiện và cập nhật state hoặc gọi action Redux

### Frontend gửi yêu cầu đến Backend:

* + Action Redux gọi API thông qua Axios
  + Yêu cầu HTTP (GET, POST, PUT, DELETE) được gửi đến endpoint API tương ứng
  + JWT token được đính kèm trong header Authorization nếu cần xác thực

### Backend xử lý yêu cầu:

* + Middleware xác thực JWT token (nếu cần)
  + URL dispatcher chuyển yêu cầu đến view function hoặc class-based view tương ứng
  + Views xử lý logic nghiệp vụ, tương tác với model để truy xuất hoặc cập nhật dữ liệu
  + Serializers chuyển đổi dữ liệu model thành JSON để gửi về client

### Frontend nhận và xử lý phản hồi:

* + Reducer Redux cập nhật state dựa trên dữ liệu từ API
  + UI được render lại để hiển thị dữ liệu mới

## Công nghệ, thư viện và framework được sử dụng

### Frontend

* + - * **ReactJS**: Thư viện JavaScript để xây dựng giao diện người dùng
      * **Redux**: Quản lý state tập trung cho ứng dụng
      * **Vite**: Công cụ build nhanh cho ứng dụng web hiện đại
      * **Axios**: Thư viện HTTP client dựa trên Promise
      * **Bootstrap**: Framework CSS
      * **React Router**: Quản lý điều hướng trong ứng dụng SPA
      * **HTML5 Audio/Video API**: Phát nội dung đa phương tiện
      * **Web Storage API**: Lưu trữ dữ liệu cục bộ (localStorage/sessionStorage)

### Backend

* + - * **Python**: Ngôn ngữ lập trình cho backend
      * **Django**: Web framework Python
      * **Django REST Framework**: Xây dựng RESTful API
      * **JWT Authentication**: Xác thực người dùng
      * **SQLite/PostgreSQL**: Hệ quản trị cơ sở dữ liệu
      * **FFmpeg**: Xử lý audio và video
      * **Pillow**: Xử lý hình ảnh
      * **django-cors-headers**: Xử lý CORS (Cross-Origin Resource Sharing)
      * **python-decouple**: Quản lý biến môi trường
      * **Mutagen**: Xử lý metadata của file âm thanh

## Mô tả cơ sở dữ liệu

Cơ sở dữ liệu của hệ thống bao gồm các bảng chính sau:

1. **CustomUser**: Lưu thông tin người dùng
   * id: integer (primary key)
   * username: varchar(150)
   * email: varchar(254)
   * password: varchar(128)
   * first\_name: varchar(150)
   * last\_name: varchar(150)
   * is\_active: boolean
   * date\_joined: datetime
   * is\_premium: boolean
   * image\_url: varchar(100)
2. **Artist**: Lưu thông tin nghệ sĩ
   * id: integer (primary key)
   * name: varchar(255)
   * image: varchar(100)
3. **Track**: Lưu thông tin bài hát
   * id: integer (primary key)
   * title: varchar(255)
   * artists\_id: foreign key to Artists
   * album\_id: foreign key to Album
   * duration: float
   * file: varchar(100)
   * image\_url: varchar(100)
   * release\_date: date
   * lyrics: text
   * point: integer
   * is\_Premium: boolean
   * category: varchar (audio/video)
4. **Album**: Lưu thông tin album
   * id: integer (primary key)
   * title: varchar(255)
   * artist\_id: integer (foreign key to Artist)
   * release\_date: date
   * image\_url: varchar (100)
   * point: integer
   * description: text
5. **Playlist**: Lưu thông tin danh sách phát
   * id: integer (primary key)
   * name: varchar(255)
   * user\_id: integer (foreign key to User)
   * image\_url: varchar (100)
   * song: ManyToMany field to Track

## Cấu trúc mã nguồn

Dự án được chia thành hai phần chính: Frontend (React) và Backend (Django).

(trang sau)

### Backend A screenshot of a computer program AI-generated content may be incorrect.

### Frontend A screenshot of a computer program AI-generated content may be incorrect.

# Giao diện ứng dụng

Giao diện của ứng dụng được thiết kế đơn giản, trực quan và thân thiện với người dùng. Bao gồm các màn hình chính sau:

1. **Trang chủ**: Hiển thị các bài hát mới, playlists đề xuất, nghệ sĩ nổi bật  
     
   A screenshot of a video game

   AI-generated content may be incorrect.
2. **Trang đăng nhập/đăng ký**: Cho phép người dùng tạo tài khoản hoặc đăng nhập  
     
   A screenshot of a login screen

   AI-generated content may be incorrect.  
     
   A screenshot of a computer login form

   AI-generated content may be incorrect.
3. **Trang tìm kiếm**: Cho phép tìm kiếm bài hát, album, nghệ sĩ  
     
   A screenshot of a video game

   AI-generated content may be incorrect.
4. **Trang chi tiết bài hát**: Hiển thị thông tin bài hát, lời bài hát, và các chức năng phát nhạc

1. **Trang playlist**: Hiển thị và quản lý danh sách phát  
     
   A screenshot of a music player

   AI-generated content may be incorrect.
2. **Trang hồ sơ người dùng**: Hiển thị thông tin người dùng, bài hát đã thích, playlist đã tạo  
     
   A screenshot of a phone

   AI-generated content may be incorrect.
3. **Trình phát nhạc**: Hiển thị ở dưới cùng của giao diện, cho phép điều khiển việc phát nhạc  
     
   A screenshot of a video

   AI-generated content may be incorrect.

# Cách thức cài đặt ứng dụng

## Yêu cầu hệ thống

Để cài đặt và chạy ứng dụng trên môi trường phát triển cục bộ, bạn cần:

* **Python**: Phiên bản 3.8 trở lên
* **Node.js**: Phiên bản 14.x trở lên
* **npm** hoặc **yarn**: Quản lý package cho JavaScript
* **Git**: Quản lý phiên bản
* **SQLite** hoặc **PostgreSQL**: Cơ sở dữ liệu
* **FFmpeg**: Xử lý audio và video (tùy chọn)

## Cài đặt Backend

### Clone repository:

bash

ggiitt cclloonnee hhttttppss::////ggiitthhuubb..ccoomm//ttoonnggddoonnggcchhaanngg//CCooddeeMMNNMM..ggiitt

ccdd CCooddeeMMNNMM//bbaacckkeenndd

### Tạo và kích hoạt môi trường ảo: A close-up of a white background AI-generated content may be incorrect.

### Cài đặt dependencies: A close-up of a text AI-generated content may be incorrect.

### Tạo file .env: A screen shot of a computer code AI-generated content may be incorrect.

### Thực hiện các migration: A close-up of a website AI-generated content may be incorrect.

### Tạo superuser (tùy chọn): A close-up of a website AI-generated content may be incorrect.

### Chạy server development A close-up of a website AI-generated content may be incorrect.

## Cài đặt Frontend

### Di chuyển đến thư mục frontend:

bash

ccdd //ffrroonntteenndd

### Cài đặt dependencies:

bash

nnppmm iinnssttaallll

*## hhooặặcc*

yyaarrnn iinnssttaallll

1. **Tạo file .env**: Tạo file

.env

trong thư mục

và thêm:

VVIITTEEAAPPIIUURRLL==hhttttpp::////llooccaallhhoosstt::88000000//aappii

frontend

### Chạy server development:

bash

nnppmm

rruunn

ddeevv

*## hhooặặcc*

yyaarrnn ddeevv

1. **Truy cập ứng dụng**: Mở trình duyệt và truy cập:

[http://localhost:5173](http://localhost:5173/)

# Vai trò của từng thành viên

## Ngô Thanh Tân (3121410442)

* + Thiết kế và phát triển cả phần Frontend và Backend
  + Xây dựng RESTful API cho các chức năng chính
  + Thiết kế cơ sở dữ liệu và quan hệ giữa các entity
  + Triển khai hệ thống xác thực JWT
  + Phát triển giao diện người dùng với ReactJS và Redux
  + Phát triển trình phát nhạc và video
  + Triển khai hệ thống thanh toán paypal, phân cấp premium
  + Khối lượng công việc: 50%

## Trần Hải Dương (3121410131)

* + Hỗ trợ phát triển phần Backend
  + Tích hợp API từ Backend
  + Hỗ trợ phát triển trang admin
  + Kiểm thử và sửa lỗi
  + Khối lượng công việc: 20%

## Lê Nguyễn Nhất Tâm (3122410369)

* + Hỗ trợ phát triển phần Frontend
  + Hỗ trợ xây dựng các trang và component UI trong ứng dụng
  + Kiểm thử và sửa lỗi giao diện
  + Phát triển trang thông tin cá nhân người dùng
  + Viết báo cáo đồ án
  + Khối lượng công việc: 30%