**TRƯỜNG ĐẠI HỌC TRÀ VINH**

**TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ**

**KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN KẾT THÚC HỌC PHẦN

CÔNG NGHỆ PHẦN MỀM

(MSHP: 220055)

TÊN ĐỀ TÀI

XÂY DỰNG WEBSITE NGHE NHẠC MELODIFY

Sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| 110122126 | Trầm Khôi Nguyên | DA22TTB |
| 110122135 | Nguyễn Duy Phát | DA22TTB |
| 110122223 | Nguyễn Đình Nhật Huy | DA22TTB |

**Giáo viên hướng dẫn:** TS. Nguyễn Bảo Ân

Trà Vinh, tháng 06 năm 2025

**LỜI CẢM ƠN**

Lời nói đầu, chúng em xin gửi lời cảm ơn đặc biệt đến thầy Nguyễn Bảo Ân đã giảng dạy và hỗ trợ chúng em trong môn Công nghệ phần mềm trong thời gian qua, thầy đã luôn truyền đạt những kiến thức và kinh nghiệm quý báu giúp chúng em vượt qua những khó khăn và có thêm nhiều kiến thức để thực hiện đề tài một cách trọn vẹn.

Những góp ý và lời khuyên của thầy là nguồn động viên và kích lệ tinh thần vô cùng to lớn, tiếp thêm cho chúng em sự tự tin trong suốt quá trình thực hiện đề tài

Do chưa có nhiều kinh nghiệm làm để tài cũng như những hạn chế về kiến thức, chúng em hiểu rằng đề tài chắc chắn sẽ không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự nhận xét, ý kiến đóng góp, từ phía thầy, cô để đề tài của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU 7](#_Toc203819332)

[1.1 Giới thiệu về dự án và chủ đề 7](#_Toc203819333)

[1.2 Lý do chọn đề tài 7](#_Toc203819334)

[1.3 Mục tiêu đề tài 7](#_Toc203819335)

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU 9](#_Toc203819336)

[2.1 Các chức năng chỉnh của hệ thống 9](#_Toc203819337)

[2.1.1 Đăng nhập và đăng ký người dùng 9](#_Toc203819338)

[2.1.2 Tìm kiếm 9](#_Toc203819339)

[2.1.3 Quản lý Playlist 9](#_Toc203819340)

[2.1.4 Theo dõi nghệ sĩ 9](#_Toc203819341)

[2.1.5 Yêu thích bài hát 9](#_Toc203819342)

[2.1.6 Nghe nhạc trực tuyến 9](#_Toc203819343)

[2.2 Các yêu cầu phi chức năng của hệ thống 9](#_Toc203819344)

[2.2.1 Giao diện thân thiện 9](#_Toc203819345)

[2.2.2 Hiệu suất 9](#_Toc203819346)

[2.2.3 Bảo mật 9](#_Toc203819347)

[CHƯƠNG 3: THIẾT KẾ HỆ THỐNG 10](#_Toc203819348)

[3.1 Kiến trúc tổng thể 10](#_Toc203819349)

[3.1.1 Kiến trúc Client-Server 10](#_Toc203819350)

[3.1.2 Mô hình MVC 11](#_Toc203819351)

[3.2 Thiết kế cơ sở dữ liệu 12](#_Toc203819352)

[3.2.1 Sơ đồ quan hệ dữ liệu 12](#_Toc203819353)

[3.3 Thiết kế API 12](#_Toc203819354)

[3.3.1 Endpoint Auth 12](#_Toc203819355)

[3.3.2 Endpoint Artists 13](#_Toc203819356)

[3.3.3 Endpoint Favorite 15](#_Toc203819357)

[3.3.4 Endpoint Follow 17](#_Toc203819358)

[3.3.5 Endpoint Playlist 18](#_Toc203819359)

[3.3.6 Endpoint PlaylistSong 21](#_Toc203819360)

[3.3.7 Endpoint Songs 23](#_Toc203819361)

[3.3.8 Endpoint User 26](#_Toc203819362)

[3.4 Thiết kế giao diện UI/UX (Huy) 29](#_Toc203819363)

[CHƯƠNG 4: TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG 30](#_Toc203819364)

[4.1 Các công nghệ sử dụng 30](#_Toc203819365)

[4.1.1 Ngôn ngữ lập trình 30](#_Toc203819366)

[4.1.2 Framework và thư viện 30](#_Toc203819367)

[4.2 Quy trình CI/CD với Github Actions 30](#_Toc203819368)

[4.3 Cấu hình Docker và quy trình triển khai ứng dụng (Huy) 31](#_Toc203819369)

[CHƯƠNG 5: QUẢN LÝ DỰ ÁN 32](#_Toc203819370)

[5.1 Lập kế hoạch sản phẩm (Product Backlog) 32](#_Toc203819371)

[5.2 Lập kế hoạch sprint (Sprint Backlog) 34](#_Toc203819372)

[CHƯƠNG 6: KIỂM THỬ 40](#_Toc203819373)

[6.1 Chiến lược kiểm thử và công cụ sử dụng 40](#_Toc203819374)

[6.1.1 Kiểm thử API thủ công (Postman) 40](#_Toc203819375)

[6.1.2 Tự động hóa kiểm thử (CI/CD trên Github Actions) 40](#_Toc203819376)

[6.2 Kết quả kiểm thử API 40](#_Toc203819377)

[6.2.1 Kiểm thử xác thực 40](#_Toc203819378)

[6.2.2 Kiểm thử quản lý bài hát 42](#_Toc203819379)

[CHƯƠNG 7: ĐÁNH GIÁ VÀ KIẾT LUẬN (Phát) 45](#_Toc203819380)

[7.1 Kết quả đạt được 45](#_Toc203819381)

[7.2 Hạn chế 45](#_Toc203819382)

[7.3 Hướng phát triển 45](#_Toc203819383)

**DANH MỤC HÌNH ẢNH VÀ BẢNG**

[Hình 3.1 Kiến trúc hệ thống website nghe nhạc Melodify 10](#_Toc203819384)

[Hình 3.2 Các thành phần của mô hình MVC 11](#_Toc203819385)

[Hình 3.3 Database diagram 12](#_Toc203819386)

[Hình 3.4 Các endpoint xác thực (Auth) 13](#_Toc203819387)

[Hình 3.5 Các endpoint nghệ sĩ (Artist) 13](#_Toc203819388)

[Hình 3.6 Các endpoint yêu thích bài hát (Favorite) 15](#_Toc203819389)

[Hình 3.7 Các endpoint theo dõi nghệ sĩ (Follow) 17](#_Toc203819390)

[Hình 3.8 Các endpoint về danh sách phát (Playlist) 18](#_Toc203819391)

[Hình 3.9 Các endpoint về bài hát trong danh sách phát (Playlist Song) 21](#_Toc203819392)

[Hình 3.10 Các endpoint về bài hát (Song) 23](#_Toc203819393)

[Hình 3.11 Các endpoint về người dùng (User) 27](#_Toc203819394)

[Hình 5.1 Các task của Epic thiết kế cơ sở dữ liệu 32](#_Toc203819395)

[Hình 5.2 Các task của Epic triển khai cơ sở dữ liệu 32](#_Toc203819396)

[Hình 5.3 Các task của Epic đăng ký tài khoản 32](#_Toc203819397)

[Hình 5.4 Các task của Epic đăng nhập tài khoản 32](#_Toc203819398)

[Hình 5.5 Các task của Epic chức năng bài hát 33](#_Toc203819399)

[Hình 5.6 Các task của Epic chức năng tìm kiếm bài hát 33](#_Toc203819400)

[Hình 5.7 Các task của Epic chức năng nghệ sĩ 33](#_Toc203819401)

[Hình 5.8 Các task của Epic chức năng playlist 33](#_Toc203819402)

[Hình 5.9 Các task của Epic chức năng phát nhạc 34](#_Toc203819403)

[Hình 5.10 Các task của Epic chức năng cập nhật thông tin người dùng 34](#_Toc203819404)

[Hình 5.11 Các task của Epic triển khai 34](#_Toc203819405)

[Hình 5.12 Các task của Sprint 1 35](#_Toc203819406)

[Hình 5.13 Burndown chart của Sprint 1 35](#_Toc203819407)

[Hình 5.14 Các task của Sprint 2 36](#_Toc203819408)

[Hình 5.15 Burndown chart của Sprint 2 36](#_Toc203819409)

[Hình 5.16 Các task của Sprint 3 37](#_Toc203819410)

[Hình 5.17 Burndown chart của Sprint 3 37](#_Toc203819411)

[Hình 5.18 Các task của Sprint 4 38](#_Toc203819412)

[Hình 5.19 Burndown chart của Sprint 4 38](#_Toc203819413)

[Hình 5.20 Các task của Sprint 5 39](#_Toc203819414)

[Hình 5.21 Burndown chart của Sprint 5 39](#_Toc203819415)

[Hình 6.1 Kết quả kiểm thử API đăng nhập 41](#_Toc203819416)

[Hình 6.2 Kết quả kiểm thử API đăng ký 41](#_Toc203819417)

[Hình 6.3 Kết quả kiểm thử API lấy danh sách tất cả bài hát 42](#_Toc203819418)

[Hình 6.4 Kết quả kiểm thử API thêm bài hát 42](#_Toc203819419)

[Hình 6.5 Kết quả kiểm thử API xóa bài hát 43](#_Toc203819420)

[Hình 6.6 Kết quả kiểm thử API cập nhật bài hát 43](#_Toc203819421)

[Hình 6.7 Kết quả kiểm thử API phát bài hát 44](#_Toc203819422)

# CHƯƠNG 1: GIỚI THIỆU

## 1.1 Giới thiệu về dự án và chủ đề

Dự án “Xây dựng website nghe nhạc Melodify” hướng đến việc xây dựng một nền tảng nghe nhạc trực tuyến hiện đại, thân thiện với người dùng, hỗ trợ các tính năng như đăng nhập và đăng ký tài khoản, khám phá các playlist nhạc, tìm kiếm bài hát, tạo và quản lý playlist, theo dõi nghệ sĩ yêu thích, nghe nhạc online, và cá nhân hóa trải nghiệm người dùng. Dự án được phát triển bằng ReactJS cho giao diện người dùng (frontend), .NET Core cho dịch vụ phía máy chủ (backend) và sử dụng SQL Server làm hệ quản trị cơ sở dữ liệu. Hệ thống có khả năng mở rộng, dễ bảo trì, và tích hợp API để xử lý dữ liệu nhạc, người dùng và nội dung liên quan.

## 1.2 Lý do chọn đề tài

Âm nhạc là một phần không thể thiếu trong đời sống hiện đại, giúp con người thư giãn, giảm stress và thể hiện cảm xúc. Trong bối cảnh công nghệ phát triển, nhu cầu nghe nhạc trực tuyến ngày càng phổ biến, đặc biệt là trên các nền tảng web và thiết bị di động. Tuy nhiên, nhiều nền tảng hiện có thường bị giới hạn bởi quảng cáo, thiếu tính cá nhân hóa, hoặc không phù hợp với thị hiếu người dùng nội địa.

Do đó, nhóm lựa chọn đề tài “Xây dựng website nghe nhạc Melodify” nhằm:

- Ứng dụng kiến thức đã học vào một sản phẩm thực tế mang tính giải trí và tiện ích cao.

- Tạo cơ hội thực hành phát triển ứng dụng web theo mô hình client-server.

- Khám phá các công nghệ hiện đại như ReactJS, .NET Core, quản lý dữ liệu nhạc, xác thực người dùng và các kỹ thuật tối ưu hóa hiệu suất.

- Hướng đến việc phát triển một nền tảng có thể mở rộng trong tương lai, phù hợp tích hợp với API tạo nhạc hoặc AI gợi ý nhạc thông minh.

## 1.3 Mục tiêu đề tài

- Dự án Melodify hướng đến việc xây dựng một website nghe nhạc đáp ứng các mục tiêu sau:

+ Cung cấp nền tảng nghe nhạc đơn giản, dễ sử dụng, giao diện thân thiện với người dùng.

+ Cho phép người dùng đăng ký, đăng nhập và quản lý thông tin cá nhân.

+ Hỗ trợ tìm kiếm bài hát theo từ khóa nhanh chóng và chính xác.

+ Tính năng tạo, chỉnh sửa và xóa playlist cá nhân, giúp người dùng lưu trữ và sắp xếp bài hát yêu thích.

+ Cho phép người dùng theo dõi nghệ sĩ để cập nhật những bản phát hành mới.

+ Nghe nhạc trực tuyến từ thư viện nhạc nội bộ của hệ thống.

+ Quản lý hiệu quả các tệp âm thanh và thông tin liên quan đến bài hát, nghệ sĩ, playlist.

# CHƯƠNG 2: PHÂN TÍCH YÊU CẦU

## Các chức năng chỉnh của hệ thống

### Đăng nhập và đăng ký người dùng

Người dùng có thể đăng nhập vào tài khoản hiện có hoặc đăng ký tài khoản mới

thông qua email.

### Tìm kiếm

Người dùng có thể tìm kiếm bài hát qua từ khóa.

### Quản lý Playlist

Người dùng có thể tạo, chỉnh sửa và xóa các playlist cá nhân.

### Theo dõi nghệ sĩ

Người dùng có thể theo dõi nghệ sĩ yêu thích để cập nhật thông tin về các bài hát

mới của họ.

### Yêu thích bài hát

Người dùng có thể chọn yêu thích những bài hát để thêm những bài hát vào danh sách yêu thích của mình.

### Nghe nhạc trực tuyến

Người dùng có thể nghe nhạc trực tuyến từ thư viện âm nhạc đa dạng của ứng dụng.

## Các yêu cầu phi chức năng của hệ thống

### Giao diện thân thiện

Giao diện được thiết kế thân thiện với người dùng và dễ sử dụng giúp nâng cao trải nghiệm nghe nhạc.

### Hiệu suất

Hệ thống phải có thời gian phản hồi nhanh (tối đa 2 giây đối với các thao tác cơ bản như tìm kiếm, bấm nghe, đăng nhập, đăng ký,…).

### Bảo mật

Hệ thống phải có cơ chế phân quyền: chỉ người dùng đã đăng nhập mới được phép sử dụng các chức năng cá nhân như tạo playlist, theo dõi nghệ sĩ.

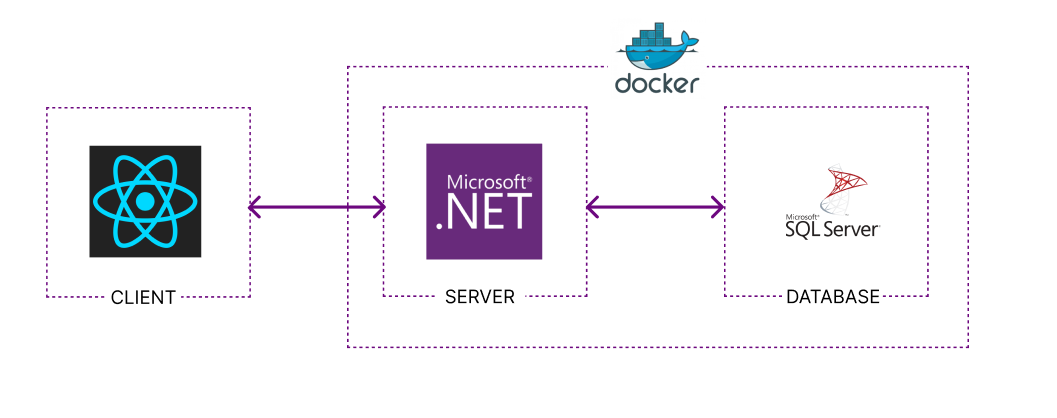
Thông tin người dùng như mật khẩu phải được bảo vệ bằng cách mã hóa, sử dụng các cơ chế xác thực an toàn JWT (JSON Web Token).

# THIẾT KẾ HỆ THỐNG

## 3.1 Kiến trúc tổng thể

### Kiến trúc Client-Server

Website nghe nhạc Melodify được tổ chức thành ba tầng chính, mỗi tầng có vai trò và chức năng cụ thể:



Hình 3.1 Kiến trúc hệ thống website nghe nhạc Melodify

- Tầng Client: Được xây dựng bằng ReactJS, là nơi người dùng tương tác với hệ thống. Giao diện hiển thị danh sách bài hát, tìm kiếm nhạc, tạo playlist, nghe nhạc trực tuyến. Gửi các yêu cầu HTTP (API request) đến tầng Server để lấy dữ liệu nhạc, thông tin người dùng, v.v.

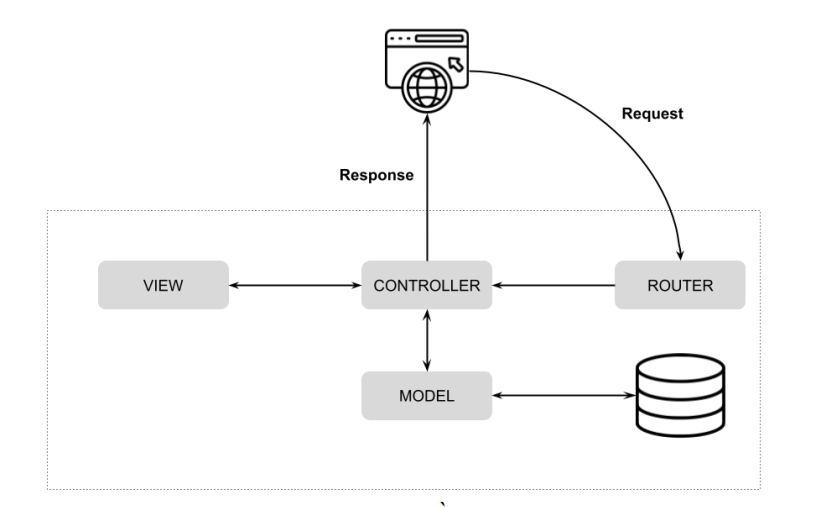
- Tầng Server: Phát triển bằng .NET, xử lý tất cả logic nghiệp vụ của ứng dụng. Đảm nhiệm việc xác thực người dùng, xử lý yêu cầu tìm kiếm, tạo playlist, cập nhật thông tin cá nhân. Giao tiếp với cơ sở dữ liệu để lưu trữ và truy xuất dữ liệu liên quan đến bài hát, nghệ sĩ, tài khoản, playlist, cung cấp các API RESTful cho tầng client sử dụng.

- Tầng Database: Sử dụng SQL Server để quản lý dữ liệu có cấu trúc như người dùng, bài hát, nghệ sĩ, playlist,… Đảm bảo dữ liệu luôn được lưu trữ an toàn, truy xuất nhanh chóng và dễ dàng mở rộng khi hệ thống phát triển.

Cấu trúc phân tầng này giúp phân chia rõ ràng các chức năng và trách nhiệm trong hệ thống phần mềm, từ việc tương tác với người dùng, xử lý logic kinh doanh đến quản lý và lưu trữ dữ liệu. Điều này tạo ra một hệ thống linh hoạt, dễ bảo trì và mở rộng

### Mô hình MVC

Ứng dụng Melodify sử dụng mô hình kiến trúc MVC (Model – View – Controller) trong phần backend (server) được phát triển bằng .NET. Đây là một mô hình thiết kế phổ biến trong phát triển phần mềm, đặc biệt phù hợp với các ứng dụng web có giao diện người dùng và logic xử lý rõ ràng.



Hình 3.2 Các thành phần của mô hình MVC

Model: Là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu (mysql, mssql… ). Nó sẽ bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu...

View: Là nơi chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh… Nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.

Controller: Là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng. Nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.

## Thiết kế cơ sở dữ liệu

### Sơ đồ quan hệ dữ liệu

Sơ đồ quan hệ dữ liệu của hệ thống Tourly gồm các bảng:

- Users: Lưu trữ thông tin người dùng như Email, mật khẩu, tên hiển thị, ảnh đại diện, quyền người dùng.

- Artists: Quản lý thông tin nghệ sĩ, bao gồm tên, tiểu sử, hình ảnh.

- Songs: Lưu trữ thông tin các bài hát, bao gồm tiêu đề, album, thể loại, ngày phát hành, đường dẫn nhạc và nghệ sĩ trình bày.

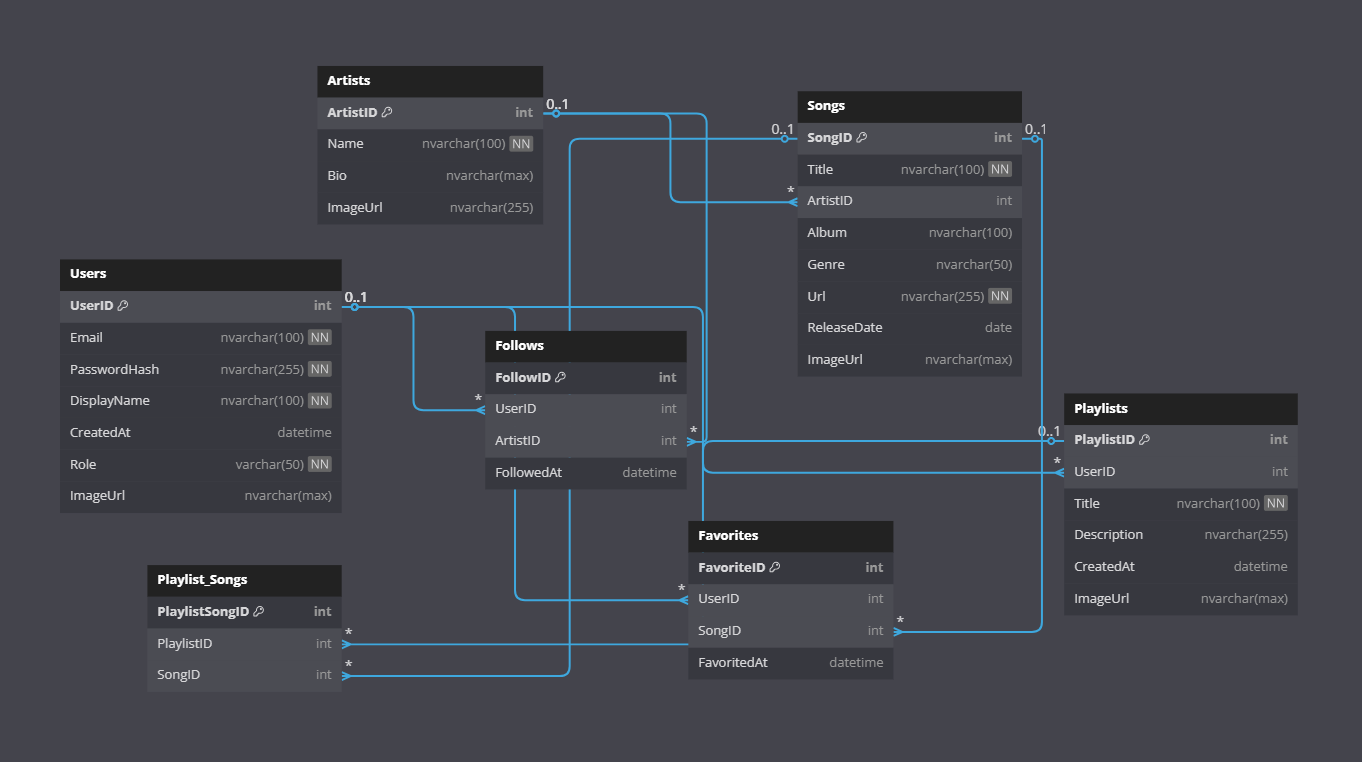
- Playlists: Quản lý danh sách phát nhạc cá nhân của người dùng.

- Favorites: Ghi nhận bài hát được người dùng yêu thích.

- Follows: Ghi nhận thông tin người dùng theo dõi nghệ sĩ.

- Playlist\_Songs: Bảng trung gian thể hiện mối quan hệ nhiều–nhiều giữa playlist và bài hát.

Cơ sở dữ liệu được thiết kế và triển khai trên SQL Server.



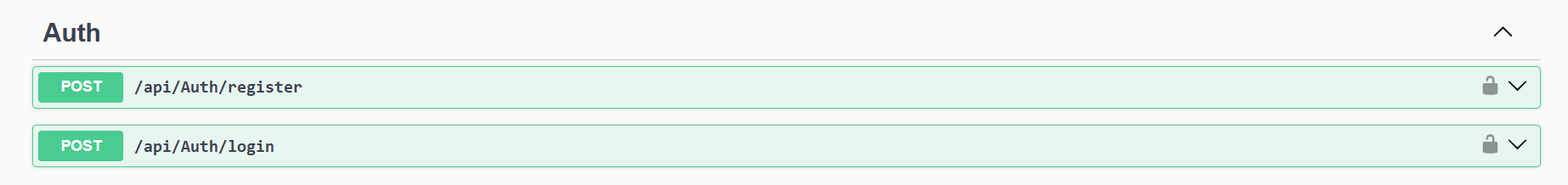
Hình 3.3 Database diagram

## 3.3 Thiết kế API

Ứng dụng Melodify được thiết kế theo kiến trúc RESTful API nhằm tách biệt frontend và backend, đồng thời đảm bảo khả năng mở rộng và tái sử dụng. Các API được xây dựng bằng .NET Core Web API và tài liệu hóa thông qua Swagger (OpenAPI 3.0), gồm các endpoint chính sau:

### Endpoint Auth

Endpoint này dùng để đăng ký và đăng nhập tài khoản người dùng.



Hình 3.4 Các endpoint xác thực (Auth)

**POST/api/Auth/register**

Dùng để đăng ký tài khoản.

Request body:

{

"displayName": "string",

"email": "string",

"password": "string"

}

Respone body:

{

"message": "Đăng ký thành công!"

}

**POST/api/Auth/login**

Dùng để đăng nhập tài khoản.

Request body:

{

"email": "string",

"password": "string"

}

Respone body:

{

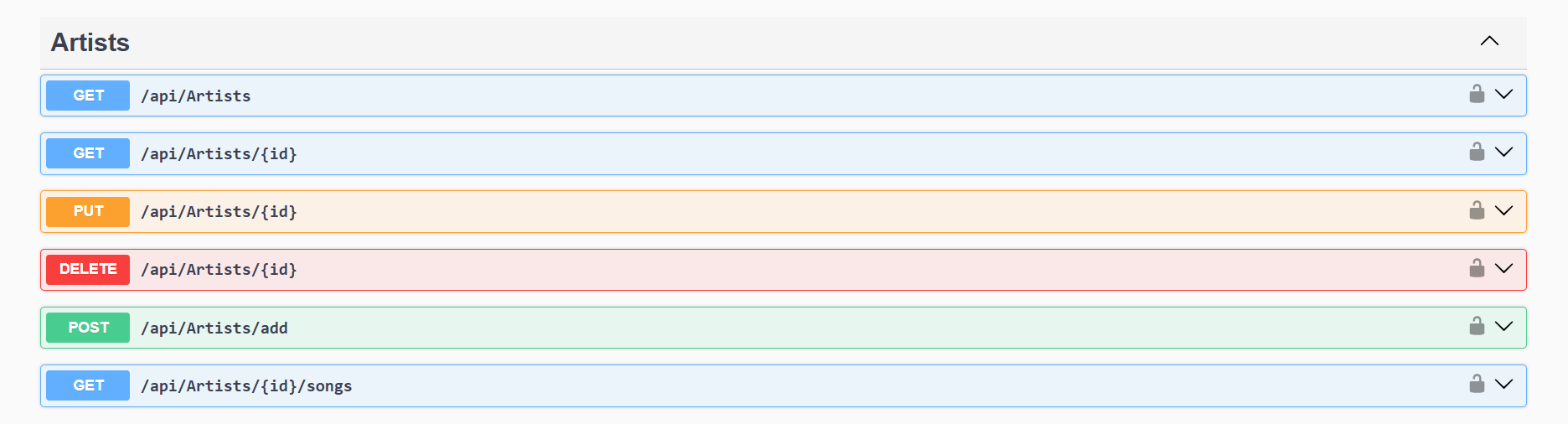
"message": "Đăng nhập thành công!",

"token": "eyJhbGciOiJIUzI1NiIsIn..."

}

### Endpoint Artists

Endpoint này dùng để quản lý thông tin nghệ sĩ, bao gồm thêm, sửa, xoá và truy xuất danh sách nghệ sĩ



Hình 3.5 Các endpoint nghệ sĩ (Artist)

**GET/api/Artists**

Dùng để truy xuất toàn bộ thông tin nghệ sĩ

Request body:

No parameters

Respone body:

[

{

"artistID": 3,

"name": "Sơn Tùng MTP",

"bio": "Nguyễn Thanh Tùng, thường được biết đến với nghệ danh Sơn Tùng M-TP, là một nam ca sĩ kiêm sáng tác nhạc, nhà sản xuất thu âm, rapper và diễn viên người Việt Nam. Nổi tiếng vì tầm ảnh hưởng sâu rộng đối với âm nhạc Việt Nam, anh được mệnh danh là Hoàng tử V-pop bởi Giải thưởng Âm nhạc Thế giới và BroadwayWorld.",

"imageUrl": "/data/artists/6a3444fd-6a96-42d5-8873-da845456597f.jpg"

}

]

**GET/api/Artists/{id}**

Dùng để truy xuất thông tin của một nghệ sĩ theo id.

Parameters:

- id (integer): 3

Respone body:

[

{

"artistID": 3,

"name": "Sơn Tùng MTP",

"bio": "Nguyễn Thanh Tùng, thường được biết đến với nghệ danh Sơn Tùng M-TP, là một nam ca sĩ kiêm sáng tác nhạc, nhà sản xuất thu âm, rapper và diễn viên người Việt Nam. Nổi tiếng vì tầm ảnh hưởng sâu rộng đối với âm nhạc Việt Nam, anh được mệnh danh là Hoàng tử V-pop bởi Giải thưởng Âm nhạc Thế giới và BroadwayWorld.",

"imageUrl": "/data/artists/6a3444fd-6a96-42d5-8873-da845456597f.jpg"

}

]

**PUT/api/Artists/{id}**

Dùng để sửa thông tin nghệ sĩ theo id.

Parameters:

- id (interger)

Request body:

- name (string)

- bio (string)

- imageFile (string)

Respone body:

{

"message": "Cập nhật nghệ sĩ thành công!"

}

**DELETE/api/Artists/{id}**

Dùng để xóa thông tin của một nghệ sĩ theo id.

Parameters:

- id (integer)

Respone body:

{

"message": "Đã xóa nghệ sĩ thành công!"

}

**POST/api/Artists/add**

Dùng để thêm nghệ sĩ mới.

Parameters:

- name (string)

- bio (string)

- imageFile (string)

Respone body:

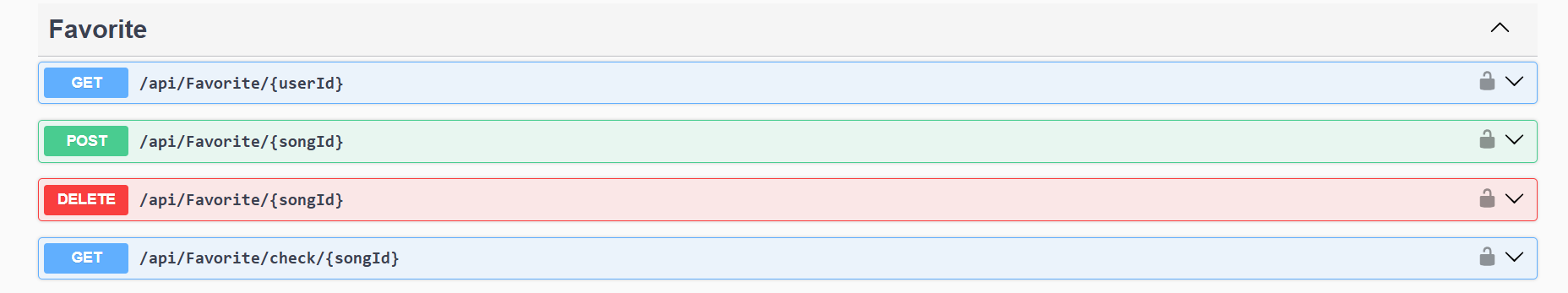
{

"message": "Thêm nghệ sĩ thành công!"

}

### Endpoint Favorite

Endpoint này dùng để quản lý bài hát yêu thích bao gồm lấy danh sách bài hát yêu thích, thêm bài hát vào danh sách yêu thích, xóa bài hát khỏi danh sách yêu thích, kiểm tra xem bài hát có nằm trong danh sách yêu thích của người dùng.



Hình 3.6 Các endpoint yêu thích bài hát (Favorite)

**GET/api/Favorite/{userId}**

Dùng để lấy danh sách bài hát yêu thích của người dùng theo userId.

Parameters:

- userId (integer)

Respone body:

[

{

"songID": 7,

"title": "Ai Cũng Phải Bắt Đầu Từ Đâu Đó",

"artistName": "HIEUTHUHAI",

"imageUrl": "/data/img/c8dc6d92-1465-4d9e-a825-8b8fe737e245.jfif",

"url": "/data/audio/02314dd1-3d45-43eb-b774-08aa17170817.mp3",

"favoritedAt": "2025-05-08T03:40:58.41"

},

{

"songID": 18,

"title": "Anh Cũng Muốn Là Đứa Trẻ",

"artistName": "Khiem",

"imageUrl": "/data/img/7073d405-8df0-4f4e-9fcd-27da164092f9.jpg",

"url": "/data/audio/82b1ab0b-bed3-4555-b5c3-46ad0c4e951b.mp3",

"favoritedAt": "2025-05-08T11:14:16.67"

},

{

"songID": 8,

"title": "The History Of Future",

"artistName": "Dan Dan",

"imageUrl": "/data/img/d1d21052-2c19-43a1-b430-3823f83ddd49.JPG",

"url": "/data/audio/65fac12c-6766-4492-9e99-089285fe0441.mp3",

"favoritedAt": "2025-06-01T05:15:26.333"

}

]]

**POST/api/Favorite/{songId}**

Dùng để thêm bài hát vào danh sách yêu thích của người dùng.

Parameters:

- songId (integer)

Respone body:

{

"message": "Đã thêm vào danh sách yêu thích.",

"data": {

"userId": 1,

"songId": 6,

"favoritedAt": "2025-07-02T03:30:56.0948163Z"

}

}

**DELETE/api/Favorite/{songId}**

Dùng để xóa bài hát khỏi danh sách yêu thích của người dùng.

Parameters:

- songId (integer)

Respone body:

{

"message": "Đã xóa khỏi danh sách yêu thích.",

"data": {

"userId": 1,

"songId": 6,

"favoritedAt": "2025-07-02T03:30:56.097"

}

}

**GET/api/Favorite/check/{songId}**

Dùng để kiểm tra xem bài hát có nằm trong danh sách yêu thích của người dùng hiện tại hay không.

Parameters:

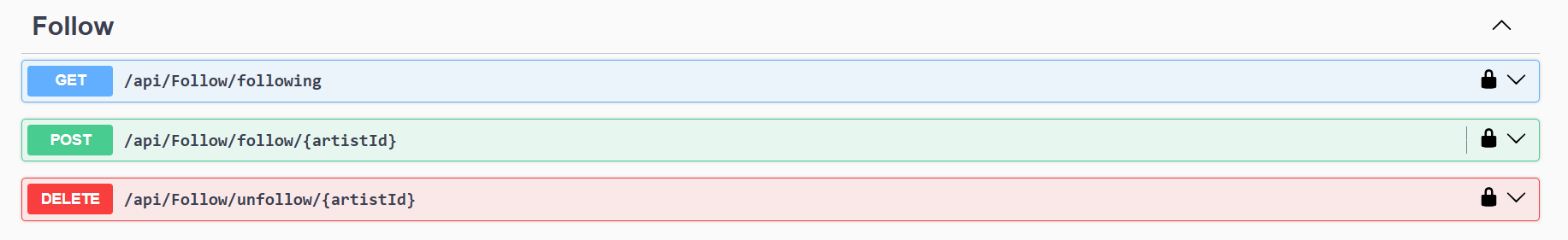
- songId (integer)

Respone body:

true

### Endpoint Follow

Endpoint này dùng cho chức năng theo dõi nghệ sĩ của người dùng.



Hình 3.7 Các endpoint theo dõi nghệ sĩ (Follow)

**GET/api/Follow/following**

Dùng để lấy danh sách nghệ sĩ đã theo dõi của người dùng.

Parameters:

No parameters

Respone body:

[

{

"artistID": 13,

"name": "MCK",

"bio": "Nghiêm Vũ Hoàng Long, thường được biết đến với nghệ danh MCK, là một nam rapper và ca sĩ kiêm sáng tác nhạc người Việt Nam. Năm 2020, anh trở nên nổi tiếng khi tham dự và đi tới vòng chung kết ở mùa đầu tiên của cuộc thi truyền hình Rap Việt.",

"imageUrl": "/data/artists/308ce054-58ca-4120-a9e9-95937f72aba3.jpg"

},

{

"artistID": 4,

"name": "Đen Vâu",

"bio": "Nguyễn Đức Cường, thường được biết đến với nghệ danh Đen hay Đen Vâu, là một nam rapper, nhạc sĩ và người dẫn chương trình người Việt Nam. Đen Vâu là một trong số ít nghệ sĩ thành công từ làn sóng underground và âm nhạc indie của Việt Nam.",

"imageUrl": "/data/artists/c9050500-2b2d-4989-96d7-73cdee6eb10c.jpg"

},

{

"artistID": 7,

"name": "Dan Dan",

"bio": "No information",

"imageUrl": "/data/artists/e4bdd052-d8bb-459b-94a5-f9e4fa960449.JPG"

}

]

**POST/api/Follow/follow/{artistId}**

Dùng để theo dõi nghệ sĩ có artistId.

Parameters:

- artistId (integer)

Respone body:

Đã theo dõi Nghệ sĩ thành công!

**DELETE/api/Follow/unfollow/{artistId}**

Dùng để bỏ theo dõi nghệ sĩ có artistId.

Parameters:

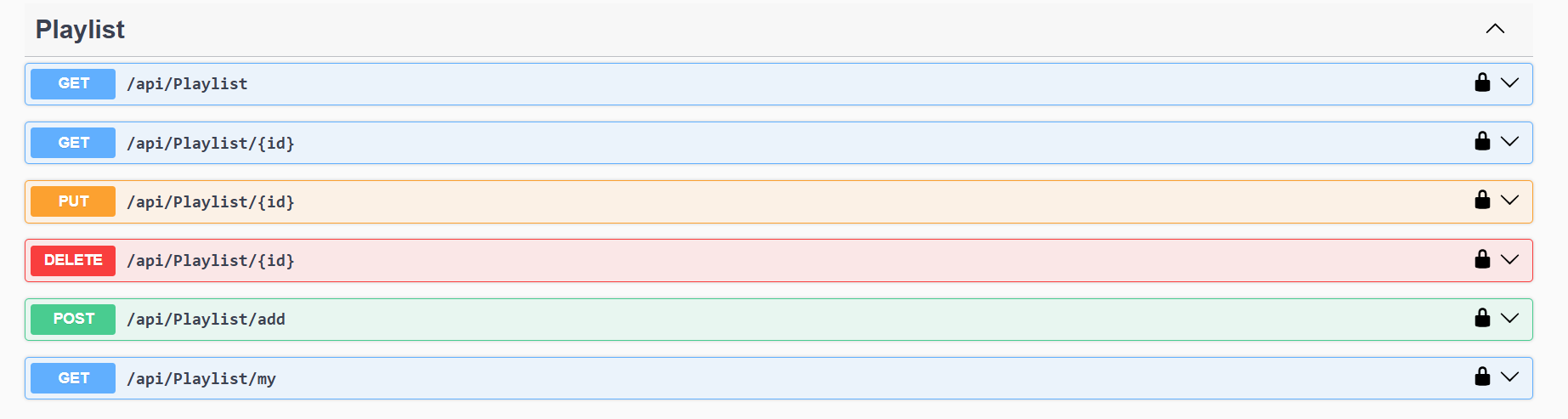
- artistId (integer)

Respone body:

Đã bỏ theo dõi Nghệ sĩ thành công!

### Endpoint Playlist

Endpoint này dùng để quản lý thông tin playlist, bao gồm thêm, sửa, xoá và truy xuất danh sách playlist.



Hình 3.8 Các endpoint về danh sách phát (Playlist)

**GET/api/Playlist**

Dùng để lấy danh sách tất cả các playlist.

Parameters:

No parameters

Respone body:

[

{

"playlistID": 1,

"userID": 1,

"title": "My playlist 1",

"description": "hihihihihi",

"createdAt": "2025-05-22T13:55:22.703",

"imageUrl": "/data/playlist/3eaa286b-b01c-4845-b380-3e61053b2e6c.png"

},

{

"playlistID": 4,

"userID": 15,

"title": "An Playlist",

"description": "hihi",

"createdAt": "2025-06-18T11:10:19.527",

"imageUrl": "/data/playlist/c3d8f450-45ba-4dc1-a56d-91d3a8a28adf.png"

},

{

"playlistID": 12,

"userID": 17,

"title": "Huy Playlist's",

"description": "sdsd",

"createdAt": "2025-07-01T15:04:31.423",

"imageUrl": "/uploads/playlists/0c247745-b65a-471e-8205-2e2b9da6ab61\_capybara-tvu.jpg"

}

]

**GET/api/Playlist/{id}**

Dùng để lấy thông tin chi tiết của một playlist theo id

Parameters:

- id (integer)

Respone body:

{

"playlistID": 1,

"userID": 1,

"title": "My playlist 1",

"description": "hihihihihi",

"createdAt": "2025-05-22T13:55:22.703",

"imageUrl": "/data/playlist/3eaa286b-b01c-4845-b380-3e61053b2e6c.png"

}

**PUT/api/Playlist/{id}**

Dùng để sửa thông tin chi tiết của một playlist theo id

Parameters:

- id (integer)

- title (string)

- description (string)

- imageFile (string)

Respone body:

{

"message": “Cập nhật Playlist thành công!”

}

**DELETE/api/Playlist/{id}**

Dùng để xóa một playlist theo id

Parameters:

- id (integer)

Respone body:

{

"message": “Đã xóa Playlist thành công!”

}

**POST/api/Playlist/add**

Dùng để thêm mới một playlist

Parameters:

- userId (interger)

- title (string)

- description (string)

- imageUrl (string)

Respone body:

{

"message": "Playlist đã được thêm!",

"playlist": {

"playlistID": 13,

"title": "string",

"description": "string",

"imageUrl": "/app-asset/img/default-playlist.jpg",

"createdAt": "2025-07-02T11:32:02.1531764+07:00",

"userID": 1

}

}

**GET/api/Playlist/my**

Dùng để lấy danh sách playlist của người dùng hiện tại (từ token xác thực)

Parameters:

No parameters

Respone body:

[

{

"playlistID": 1,

"userID": 1,

"title": "My playlist 1",

"description": "hihihihihi",

"createdAt": "2025-05-22T13:55:22.703",

"imageUrl": "/data/playlist/3eaa286b-b01c-4845-b380-3e61053b2e6c.png"

},

{

"playlistID": 13,

"userID": 1,

"title": "string",

"description": "string",

"createdAt": "2025-07-02T11:32:02.153",

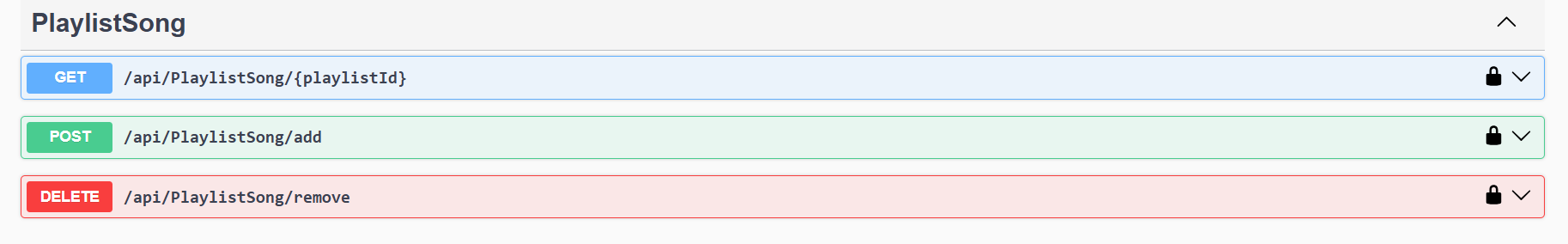
"imageUrl": "/app-asset/img/default-playlist.jpg"

}

]

### Endpoint PlaylistSong

Endpoint này dùng để thao tác với bài hát trong playlist, bao gồm thêm, xoá và truy xuất bài hát trong playlist.



Hình 3.9 Các endpoint về bài hát trong danh sách phát (Playlist Song)

**GET/api/Playlist/{playlistId}**

Dùng để lấy danh sách bài hát trong một playlist theo playlist id

Parameters:

- playlistId (integer)

Respone body:

[

{

"songID": 7,

"title": "Ai Cũng Phải Bắt Đầu Từ Đâu Đó",

"artistID": 5,

"artistName": "HIEUTHUHAI",

"album": "Ai Cũng Phải Bắt Đầu Từ Đâu Đó",

"genre": "Rap",

"url": "/data/audio/02314dd1-3d45-43eb-b774-08aa17170817.mp3",

"releaseDate": "2025-03-15T00:00:00",

"imageUrl": "/data/img/c8dc6d92-1465-4d9e-a825-8b8fe737e245.jfif"

},

{

"songID": 6,

"title": "test hihi",

"artistID": 3,

"artistName": "Sơn Tùng MTP",

"album": "MTP",

"genre": "Pop",

"url": "/data/audio/f60d6385-cc7c-4ded-9468-81d78cd9a3fb.mp3",

"releaseDate": "2025-03-13T00:00:00",

"imageUrl": "/data/img/4e1a390a-2e85-4cdb-bd51-53fee30cfc36.jpg"

},

{

"songID": 17,

"title": "Nhạc test",

"artistID": 3,

"artistName": "Sơn Tùng MTP",

"album": "sdasd",

"genre": "V-Pop",

"url": "/data/audio/5bbf9eb6-0d96-4505-9601-9c1c9737616e.mp3",

"releaseDate": "2025-03-24T00:00:00",

"imageUrl": "/data/img/e22456b2-dbd9-44a2-8b43-6fee18f37ff4.JPG"

},

{

"songID": 18,

"title": "Anh Cũng Muốn Là Đứa Trẻ",

"artistID": 12,

"artistName": "Khiem",

"album": "Anh Cũng Muốn Là Đứa Trẻ",

"genre": "Indie",

"url": "/data/audio/82b1ab0b-bed3-4555-b5c3-46ad0c4e951b.mp3",

"releaseDate": "2025-04-17T00:00:00",

"imageUrl": "/data/img/7073d405-8df0-4f4e-9fcd-27da164092f9.jpg"

}

]

**POST/api/PlaylistSong/add**

Dùng để thêm bài hát vào playlist

Parameters:

- playlistId (interger)

- songId (interger)

Respone body:

Đã thêm bài hát vào Playlist thành công!

**DELETE/api/PlaylistSong/remove**

Dùng để xóa bài hát khỏi playlist

Parameters:

- playlistId (interger)

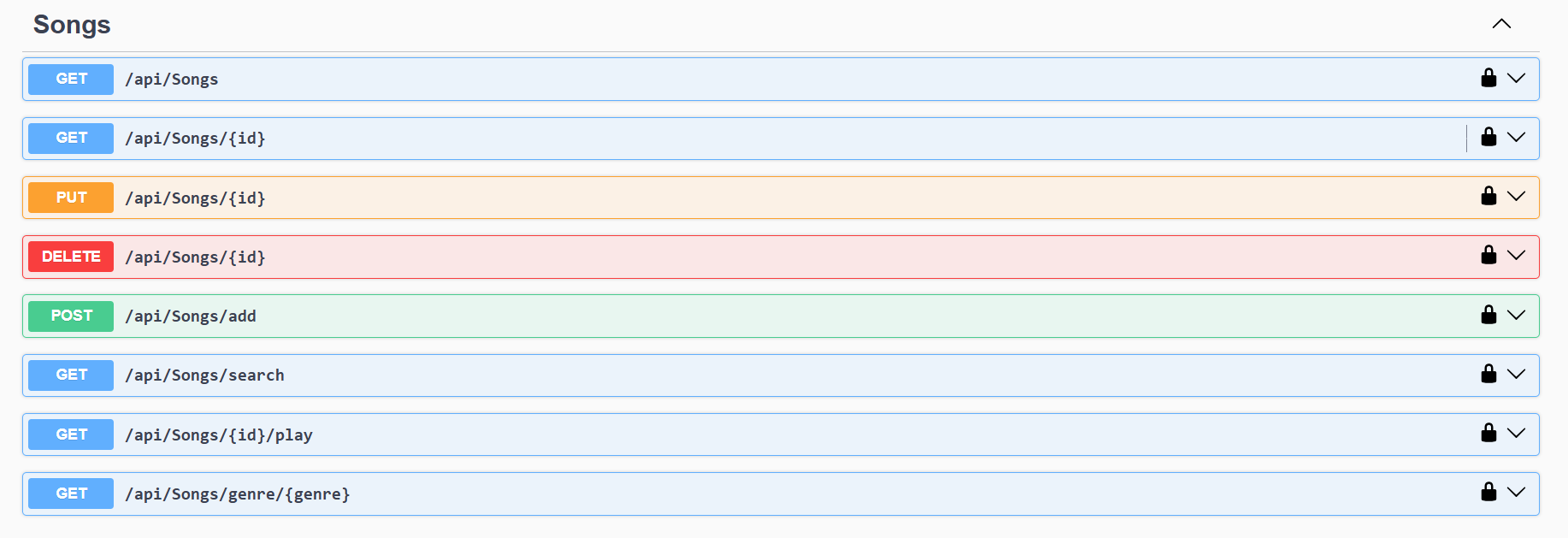
- songId (interger)

Respone body:

Đã xóa bài hát khỏi Playlist thành công!

### Endpoint Songs

Endpoint này dùng để quản lý và phát bài hát, bao gồm thêm, xoá, sửa, tìm kiếm và truy xuất bài hát.



Hình 3.10 Các endpoint về bài hát (Song)

**GET/api/Songs**

Dùng để lấy danh sách tất cả các bài hát.

Parameters:

No parameters

Respone body:

[

{

"songID": 6,

"title": "test hihi",

"artistID": 3,

"artistName": "Sơn Tùng MTP",

"album": "MTP",

"genre": "Pop",

"url": "/data/audio/f60d6385-cc7c-4ded-9468-81d78cd9a3fb.mp3",

"releaseDate": "2025-03-13T00:00:00",

"imageUrl": "/data/img/4e1a390a-2e85-4cdb-bd51-53fee30cfc36.jpg"

},

{

"songID": 7,

"title": "Ai Cũng Phải Bắt Đầu Từ Đâu Đó",

"artistID": 5,

"artistName": "HIEUTHUHAI",

"album": "Ai Cũng Phải Bắt Đầu Từ Đâu Đó",

"genre": "Rap",

"url": "/data/audio/02314dd1-3d45-43eb-b774-08aa17170817.mp3",

"releaseDate": "2025-03-15T00:00:00",

"imageUrl": "/data/img/c8dc6d92-1465-4d9e-a825-8b8fe737e245.jfif"

},

{

"songID": 8,

"title": "The History Of Future",

"artistID": 7,

"artistName": "Dan Dan",

"album": "The History Of Future",

"genre": "EDM",

"url": "/data/audio/65fac12c-6766-4492-9e99-089285fe0441.mp3",

"releaseDate": "2025-03-16T00:00:00",

"imageUrl": "/data/img/d1d21052-2c19-43a1-b430-3823f83ddd49.JPG"

},

{

"songID": 17,

"title": "Nhạc test",

"artistID": 3,

"artistName": "Sơn Tùng MTP",

"album": "sdasd",

"genre": "V-Pop",

"url": "/data/audio/5bbf9eb6-0d96-4505-9601-9c1c9737616e.mp3",

"releaseDate": "2025-03-24T00:00:00",

"imageUrl": "/data/img/e22456b2-dbd9-44a2-8b43-6fee18f37ff4.JPG"

},

{

"songID": 18,

"title": "Anh Cũng Muốn Là Đứa Trẻ",

"artistID": 12,

"artistName": "Khiem",

"album": "Anh Cũng Muốn Là Đứa Trẻ",

"genre": "Indie",

"url": "/data/audio/82b1ab0b-bed3-4555-b5c3-46ad0c4e951b.mp3",

"releaseDate": "2025-04-17T00:00:00",

"imageUrl": "/data/img/7073d405-8df0-4f4e-9fcd-27da164092f9.jpg"

},

{

"songID": 19,

"title": "Test postman 1",

"artistID": 3,

"artistName": "Sơn Tùng MTP",

"album": "Test postman",

"genre": "Pop",

"url": "/data/audio/fc3fc73e-ba65-433f-9de5-c6d1f8865fa2.mp3",

"releaseDate": "2025-05-01T00:00:00",

"imageUrl": "/data/img/b2ead15d-d6e0-4e1b-ae11-fa11f5f8fa5d.jpg"

}

]

**GET/api/Songs/{id}**

Dùng để lấy thông tin chi tiết của một bài hát theo id

Parameters:

- id (integer)

Respone body:

{

"songID": 6,

"title": "test hihi",

"artistID": 3,

"artistName": "Sơn Tùng MTP",

"album": "MTP",

"genre": "Pop",

"url": "/data/audio/f60d6385-cc7c-4ded-9468-81d78cd9a3fb.mp3",

"releaseDate": "2025-03-13T00:00:00",

"imageUrl": "/data/img/4e1a390a-2e85-4cdb-bd51-53fee30cfc36.jpg"

}

**PUT/api/Songs/{id}**

Dùng để sửa thông tin chi tiết của một bài hát theo id

Parameters:

- id (integer)

- title (string)

- artistId (interger)

- album (string)

- genre (string)

- releaseDate (datetime)

- imageFile (string)

- audioFile

Respone body:

{

"message": “Cập nhật bài hát thành công!”

}

**DELETE/api/Songs/{id}**

Dùng để xóa một bài hát theo id

Parameters:

- id (integer)

Respone body:

{

"message": “Đã xóa bài hát thành công!”

}

**POST/api/Songs/add**

Dùng để thêm mới bài hát

Parameters:

- title (string)

- artistId (interger)

- album (string)

- genre (string)

- releaseDate (datetime)

- imageFile (string)

- audioFile

Respone body:

{

"message": “Thêm bài hát thành công!”

}

**GET/api/Songs/search**

Dùng để tìm kiếm bài hát theo từ khóa.

Parameters:

- keyword

Respone body:

[

{

"songID": 18,

"title": "Anh Cũng Muốn Là Đứa Trẻ",

"artistID": 12,

"artistName": "Khiem",

"album": "Anh Cũng Muốn Là Đứa Trẻ",

"genre": "Indie",

"url": "/data/audio/82b1ab0b-bed3-4555-b5c3-46ad0c4e951b.mp3",

"releaseDate": "0001-01-01T00:00:00",

"imageUrl": "/data/img/7073d405-8df0-4f4e-9fcd-27da164092f9.jpg"

}

]

**GET/api/Songs/{id}/play**

Dùng để phát bài hát theo id.

Parameters:

- id (integer)

Respone body:

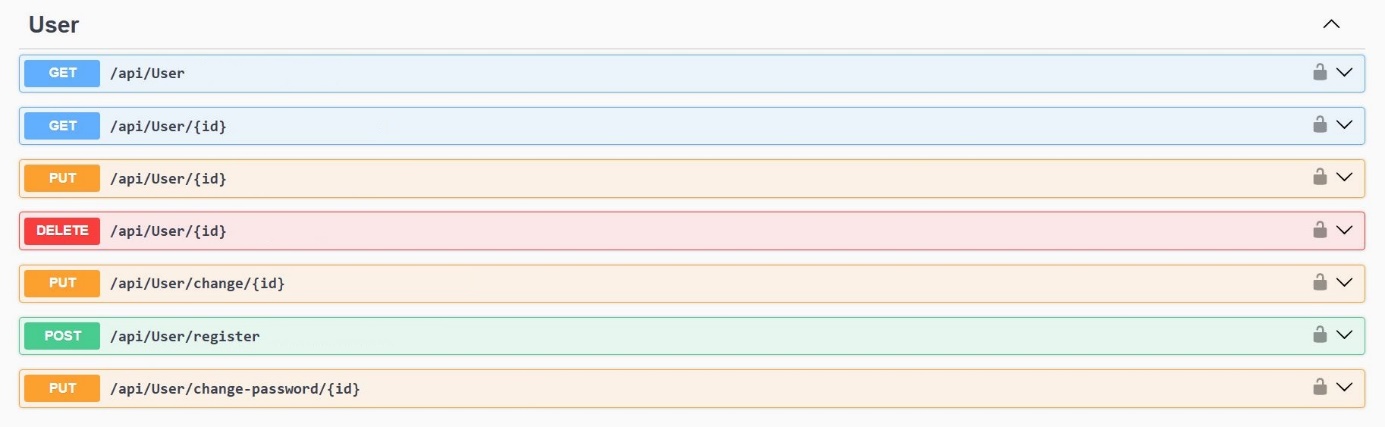
{

"url": "/data/audio/f60d6385-cc7c-4ded-9468-81d78cd9a3fb.mp3"

}

### Endpoint User

Endpoint này dùng để quản lý người dùng bao gồm thêm, xoá, sửa, lấy danh sách người dùng và xem thông tin của một người dùng cụ thể.



Hình 3.11 Các endpoint về người dùng (User)

**GET/api/User**

Dùng để lấy danh sách tất người dùng và yêu cầu đăng nhập với quyền admin.

Parameters:

No parameters

Respone body:

[

{

"userID": 1,

"email": "tramkhoinguyen27122@gmail.com",

"displayName": "Trầm Khôi Nguyên",

"createdAt": "2025-02-14T10:03:35.147",

"role": "Admin",

"imageUrl": "/data/users/95599e50-33cc-4774-9763-4c3944dddb9e.jpg"

},

{

"userID": 3,

"email": "test@gmail.com",

"displayName": "Test 2",

"createdAt": "2025-03-15T07:24:27.637",

"role": "User",

"imageUrl": "/data/users/a3ea86e2-8ee5-47c9-8f4b-e8cf363d1d6c.jpg"

},

{

"userID": 4,

"email": "nguyenkhoi@gmail.com",

"displayName": "Nguyên Khôi",

"createdAt": "2025-03-15T09:10:41.49",

"role": "User",

"imageUrl": "/data/users/864a0d33-f9cb-4961-aecb-24eda1a99d97.png"

}

]

**GET/api/User/{id}**

Dùng để lấy thông tin của một người dùng cụ thể theo id.

Parameters:

- id (interger)

Respone body:

{

"userID": 3,

"email": "test@gmail.com",

"displayName": "Test 2",

"createdAt": "2025-03-15T07:24:27.637",

"role": "User",

"imageUrl": "/data/users/a3ea86e2-8ee5-47c9-8f4b-e8cf363d1d6c.jpg"

}

**PUT/api/User/{id}**

Dùng để cập nhật thông tin của người dùng dành cho người dùng và admin yêu cầu đã đăng nhập

Parameters:

- id (integer)

Request body:

{

"userID": 0,

"email": "string",

"displayName": "string",

"createdAt": "2025-07-13T04:39:24.916Z",

"role": "string",

"imageUrl": "string"

}

Respone body:

{

"message": “Cập nhật người dùng thành công!”

}

**DELETE/api/User/{id}**

Dùng để xóa một người dùng cụ thể theo id dành cho admin.

Parameters:

- id (interger)

Respone body:

{

"message": "Xóa người dùng thành công!"

}

**POST/api/User/register**

Dùng thêm mới một người dùng dành cho admin

Request body:

- Email (string)

- Password (string)

- DisplayName (string)

- Role (string)

- ImageFile (string)

Respone body:

{

"userID": 23,

"email": "user@example.com",

"displayName": "string",

"createdAt": "2025-07-13T04:53:44.0118683Z",

"role": "string",

"imageUrl": "/data/users/9b7a01f0-9e62-4ca3-b514-d40b69c7dfef.jpg"

}

**PUT/api/User/change-password/{id}**

Dùng để đổi mật của người dùng .

Parameters:

- id (integer)

Request body:

{

"currentPassword": "string",

"newPassword": "string"

}

Respone body:

{

"message": "Đổi mật khẩu thành công!"

}

## 3.4 Thiết kế giao diện UI/UX (Huy)

# TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG

## 4.1 Các công nghệ sử dụng

### Ngôn ngữ lập trình

- C#: Ngôn ngữ chính để xây dựng API backend với ASP.NET Core.

- JavaScript (ES6+): Dùng cho frontend React, tương tác trình duyệt, xử lý logic người dùng.

- HTML5, CSS3: Thiết kế giao diện người dùng cơ bản.

- JSON: Định dạng dữ liệu chính để trao đổi giữa client và server.

### Framework và thư viện

- ASP.NET Core 8: Framework để xây dựng RESTful API cho backend.

- ReactJS: Thư viện xây dựng giao diện người dùng phía frontend.

- Entity Framework Core: ORM dùng để thao tác cơ sở dữ liệu SQL Server.

- React Router DOM: Quản lý định tuyến trong ứng dụng React.

- Swagger / Swashbuckle: Tự động sinh tài liệu API cho backend.

- JWT (JSON Web Token): Sử dụng để xác thực và phân quyền người dùng.

- Docker: Dùng để container hóa backend, frontend và cơ sở dữ liệu.

- Docker Compose: Quản lý các dịch vụ như API, database, frontend một cách tập trung.

## 4.2 Quy trình CI/CD với Github Actions

Để tự động hóa quá trình kiểm thử và triển khai backend, dự án Melodify đã áp dụng CI/CD (Continuous Integration / Continuous Deployment) sử dụng GitHub Actions. Việc này giúp giảm thiểu lỗi thủ công, đảm bảo tính ổn định và tăng tốc độ triển khai ứng dụng.

Quy trình thực hiện:

1. Kiểm tra và tải mã nguồn

2. Thiết lập môi trường build Docker

3. Build container cho backend và cơ sở dữ liệu

4. Chạy ứng dụng và database trong môi trường test

5. Kiểm tra API có hoạt động không

6. Hiển thị container đang chạy (debug)

7. Tắt toàn bộ container sau kiểm thử

8. Triển khai lên VPS qua SSH

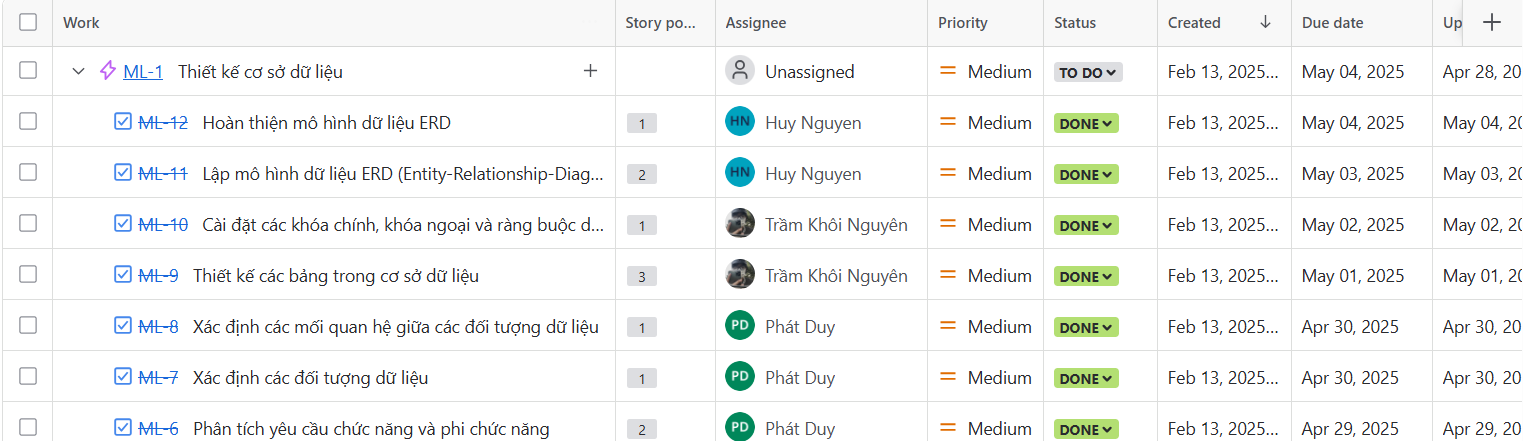
## 4.3 Cấu hình Docker và quy trình triển khai ứng dụng (Huy)

# QUẢN LÝ DỰ ÁN

## 5.1 Lập kế hoạch sản phẩm (Product Backlog)

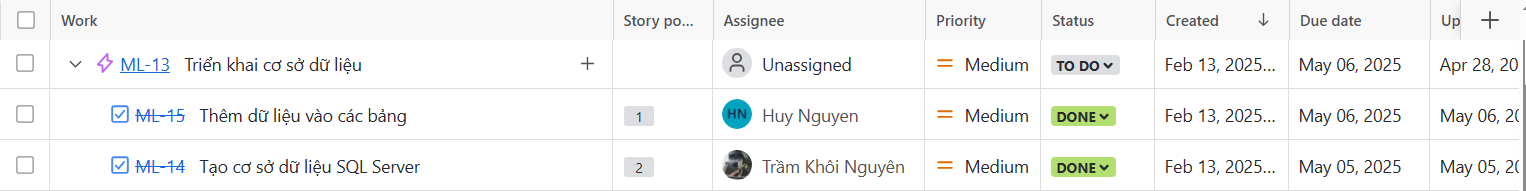
Gồm có 11 epic với các task tương ứng:

**Epic: Thiết kế cơ sở dữ liệu**

****

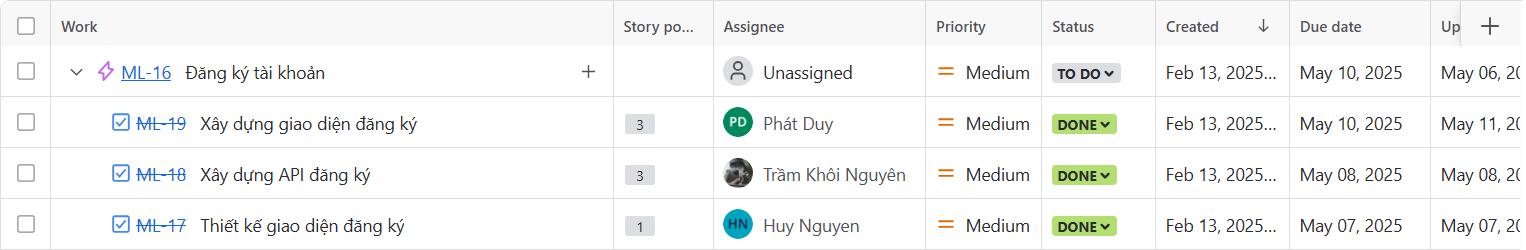
Hình 5.1 Các task của Epic thiết kế cơ sở dữ liệu

**Epic: Triển khai cơ sở dữ liệu**

****

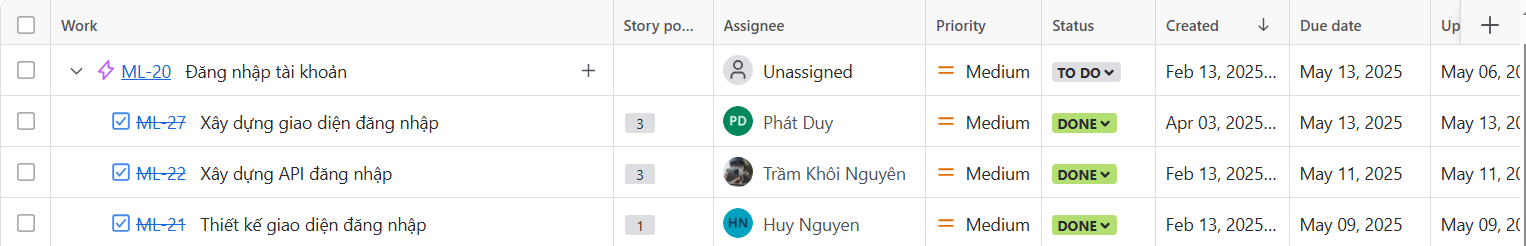
Hình 5.2 Các task của Epic triển khai cơ sở dữ liệu

**Epic: Đăng ký tài khoản**

****

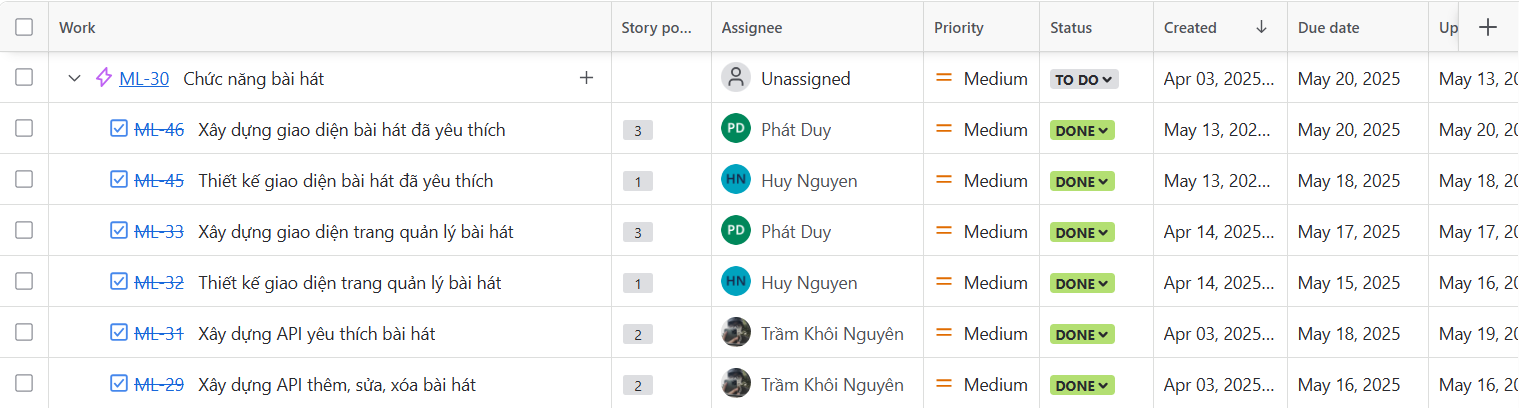
Hình 5.3 Các task của Epic đăng ký tài khoản

**Epic: Đăng nhập tài khoản**

****

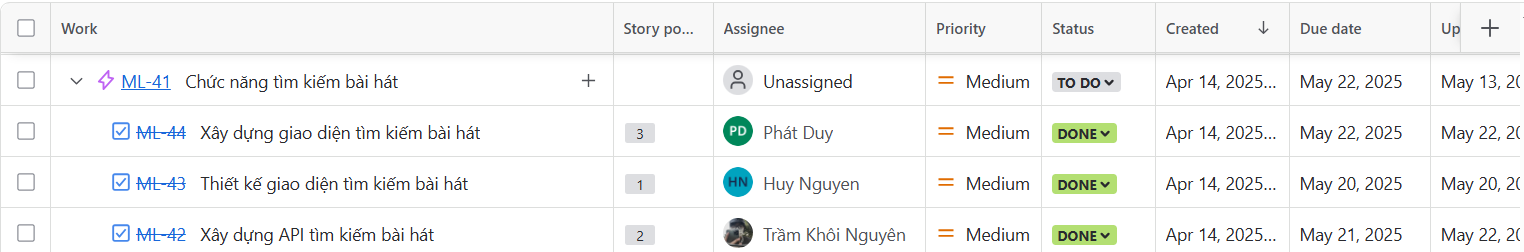
Hình 5.4 Các task của Epic đăng nhập tài khoản

**Epic: Chức năng bài hát**

****

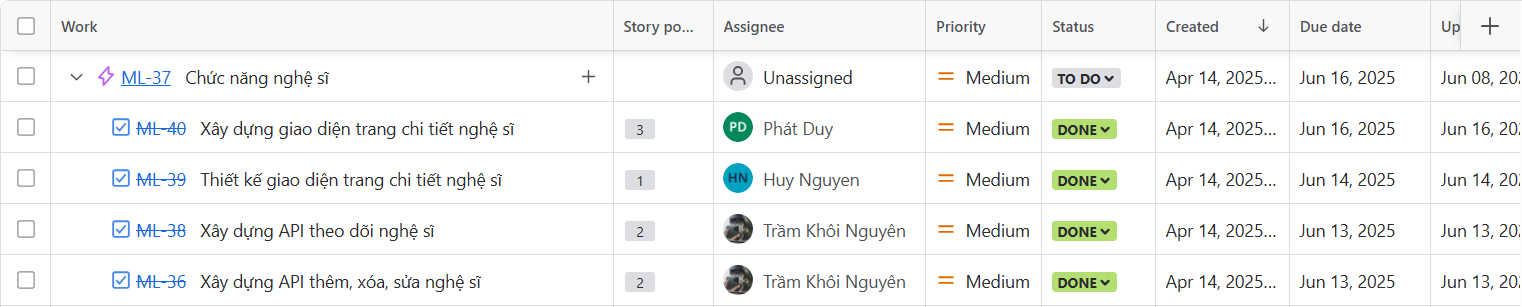
Hình 5.5 Các task của Epic chức năng bài hát

**Epic: Chức năng tìm kiếm bài hát**

****

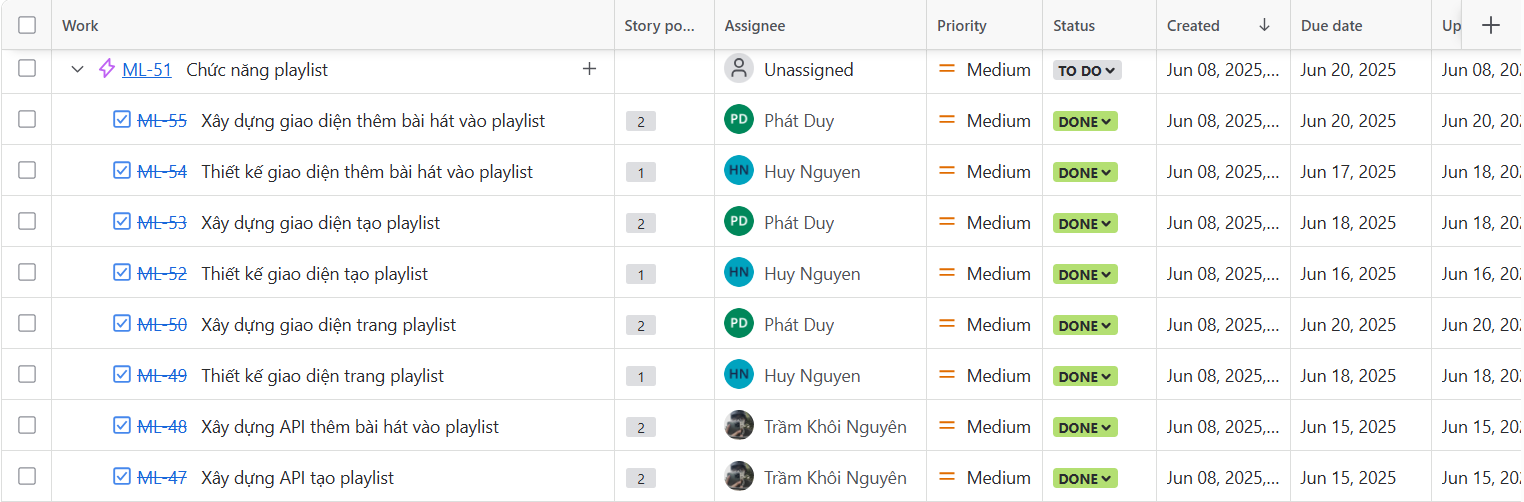
Hình 5.6 Các task của Epic chức năng tìm kiếm bài hát

**Epic: Chức năng nghệ sĩ**

****

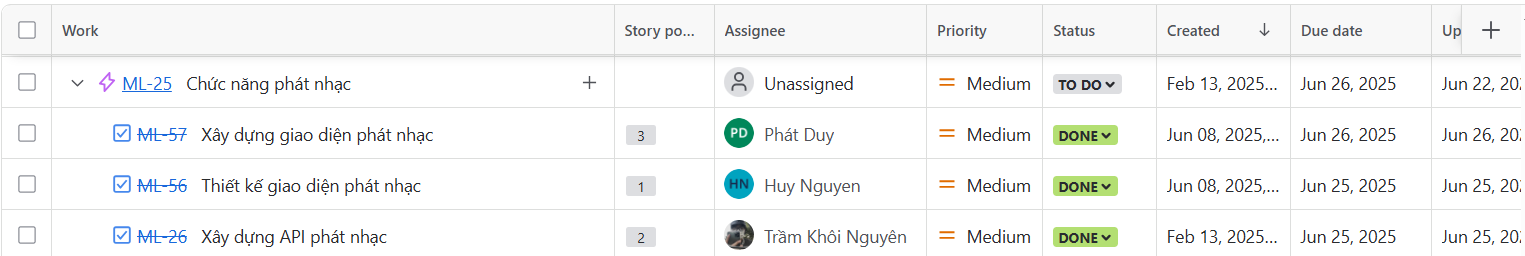
Hình 5.7 Các task của Epic chức năng nghệ sĩ

**Epic: Chức năng playlist**

****

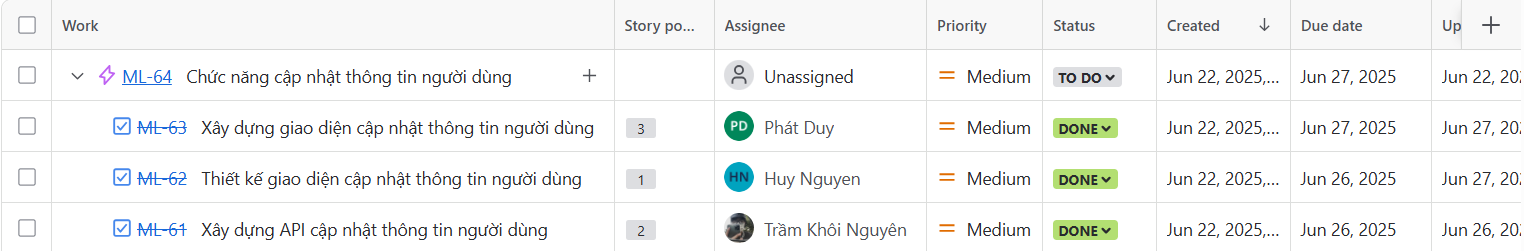
Hình 5.8 Các task của Epic chức năng playlist

**Epic: Chức năng phát nhạc**

****

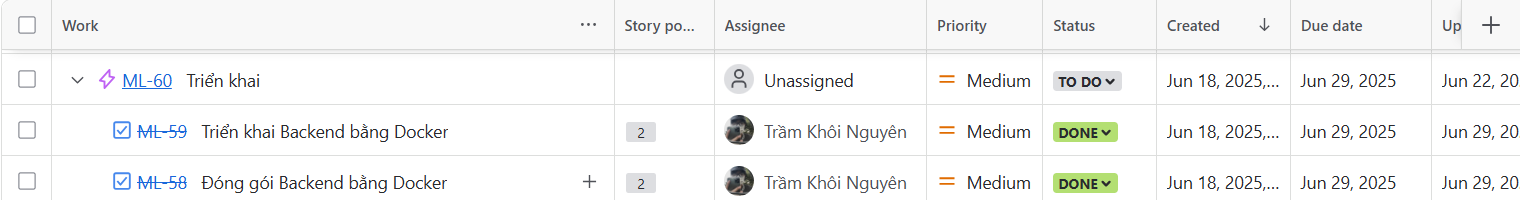
Hình 5.9 Các task của Epic chức năng phát nhạc

**Epic: Chức năng cập nhật thông tin người dùng**

******

Hình 5.10 Các task của Epic chức năng cập nhật thông tin người dùng

**Epic: Triển khai**

******

Hình 5.11 Các task của Epic triển khai

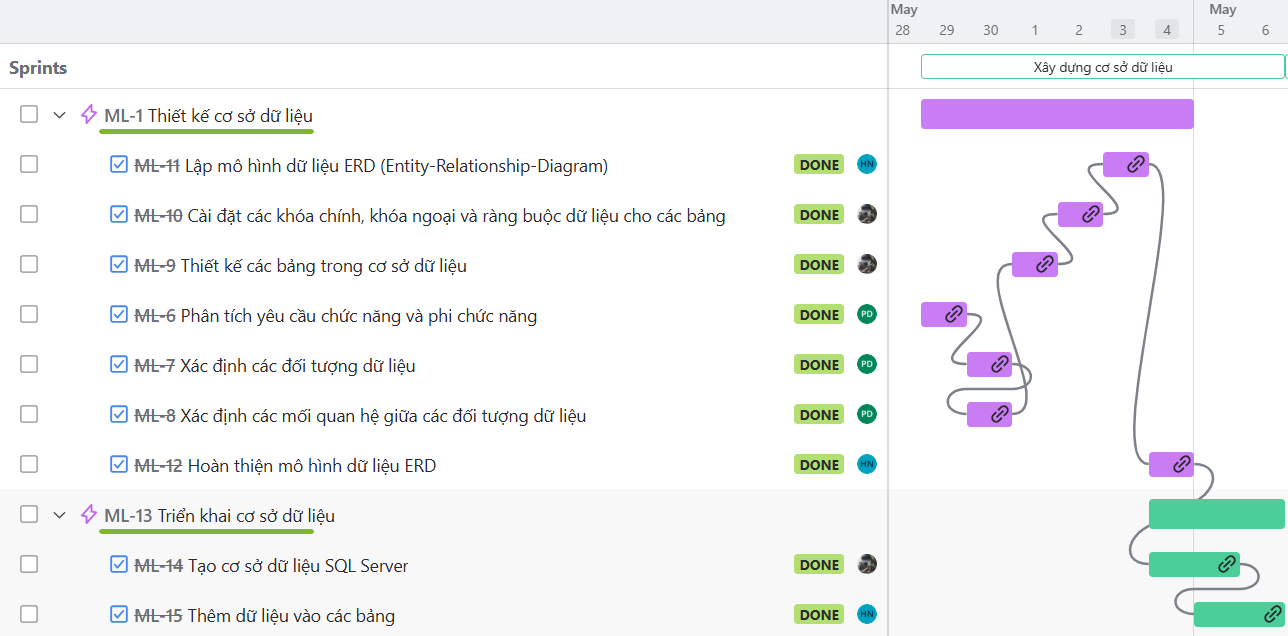
## 5.2 Lập kế hoạch sprint (Sprint Backlog)

**Sprint 1: Xây dựng cơ sở dữ liệu**

Ngày bắt đầu: 29/04/2025

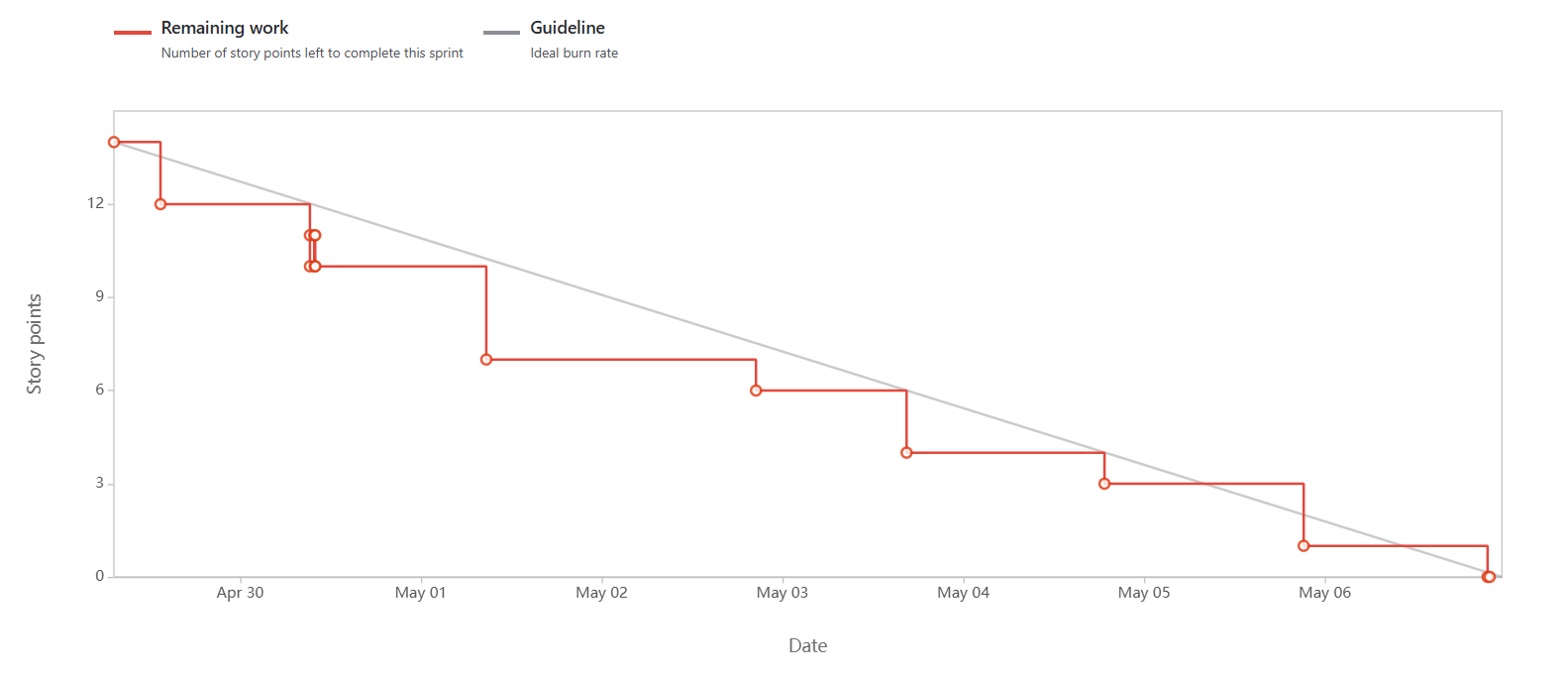
Ngày kết thúc: 06/05/2025

Mục tiêu: Tạo cơ sở dữ liệu cơ bản cho dự án để đảm bảo sự chuẩn bị cho việc phát triển các tính năng chính trong tương lai.



Hình 5.12 Các task của Sprint 1

**Burndown chart của Sprint 1:**



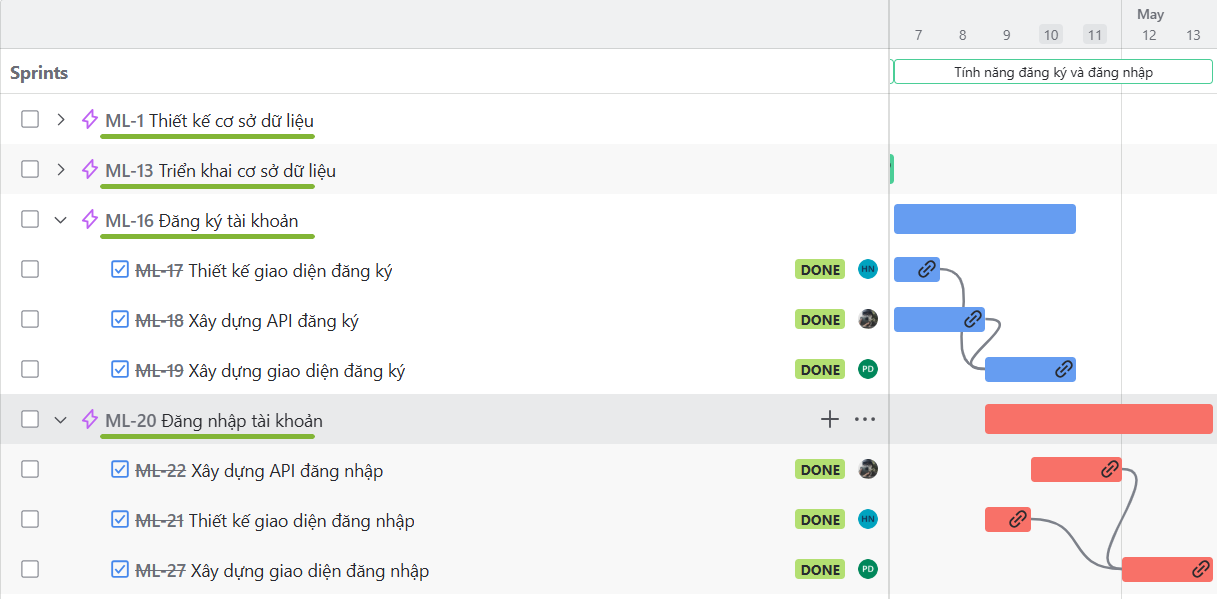
Hình 5.13 Burndown chart của Sprint 1

**Sprint 2: Tính năng đăng ký và đăng nhập**

Ngày bắt đầu: 07/05/2025

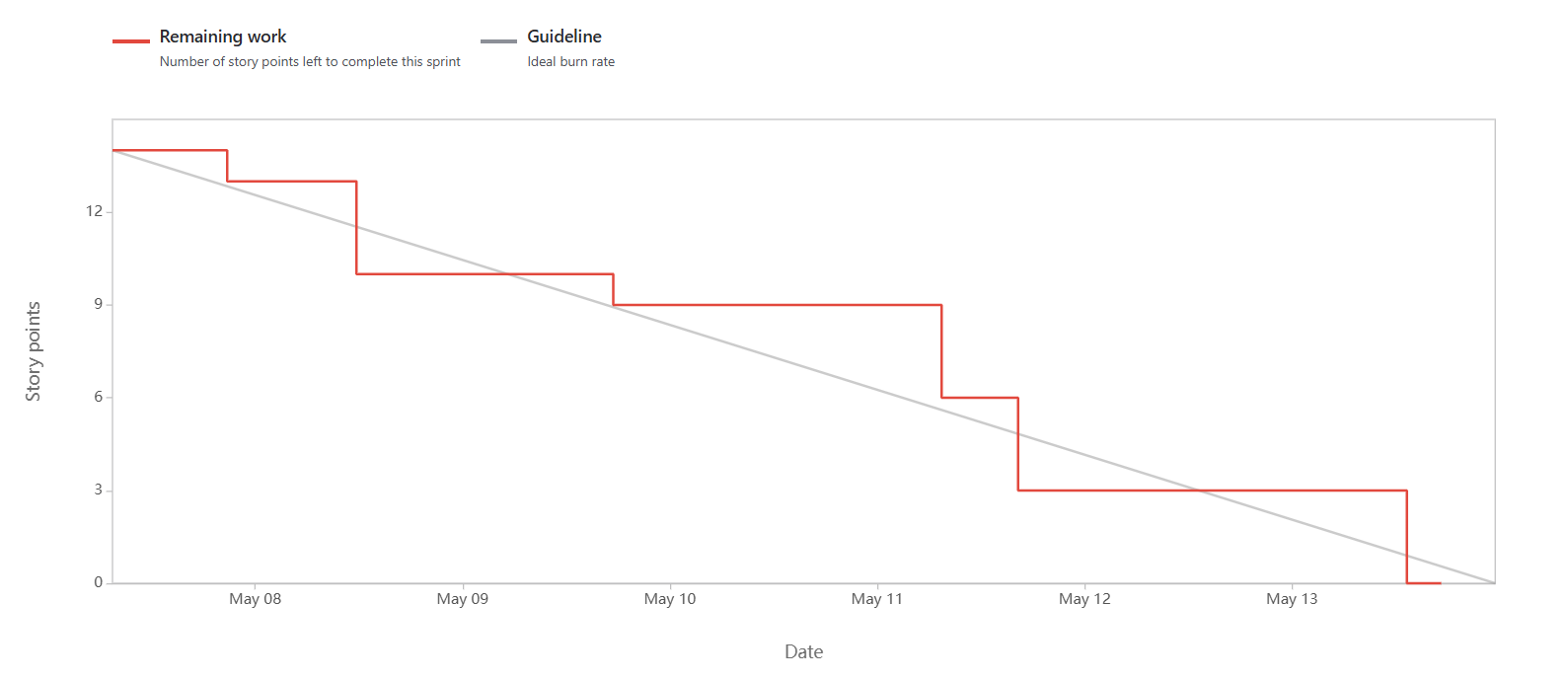
Ngày kết thúc: 13/05/2025

Mục tiêu: Hoàn thành tính năng đăng ký và đăng nhập tài khoản cho người dùng



Hình 5.14 Các task của Sprint 2

**Burndown chart của Sprint 2:**



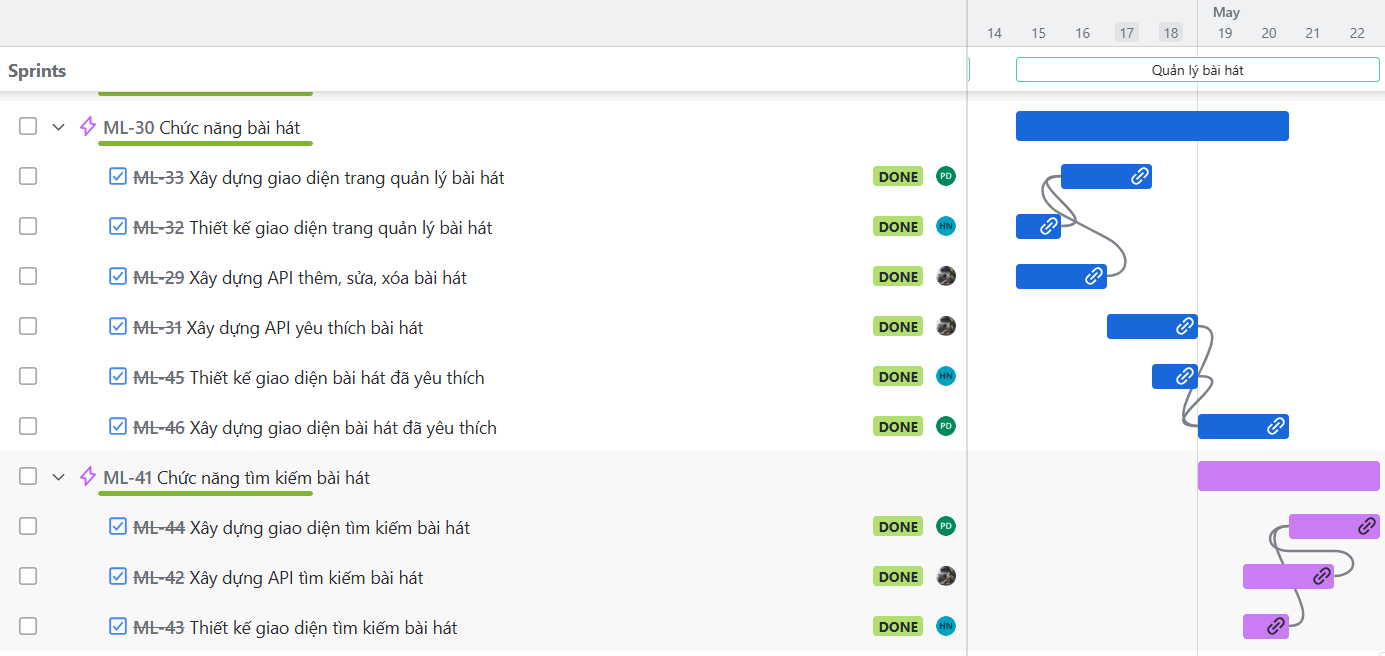
Hình 5.15 Burndown chart của Sprint 2

**Sprint 3: Quản lý bài hát**

Ngày bắt đầu: 15/05/2025

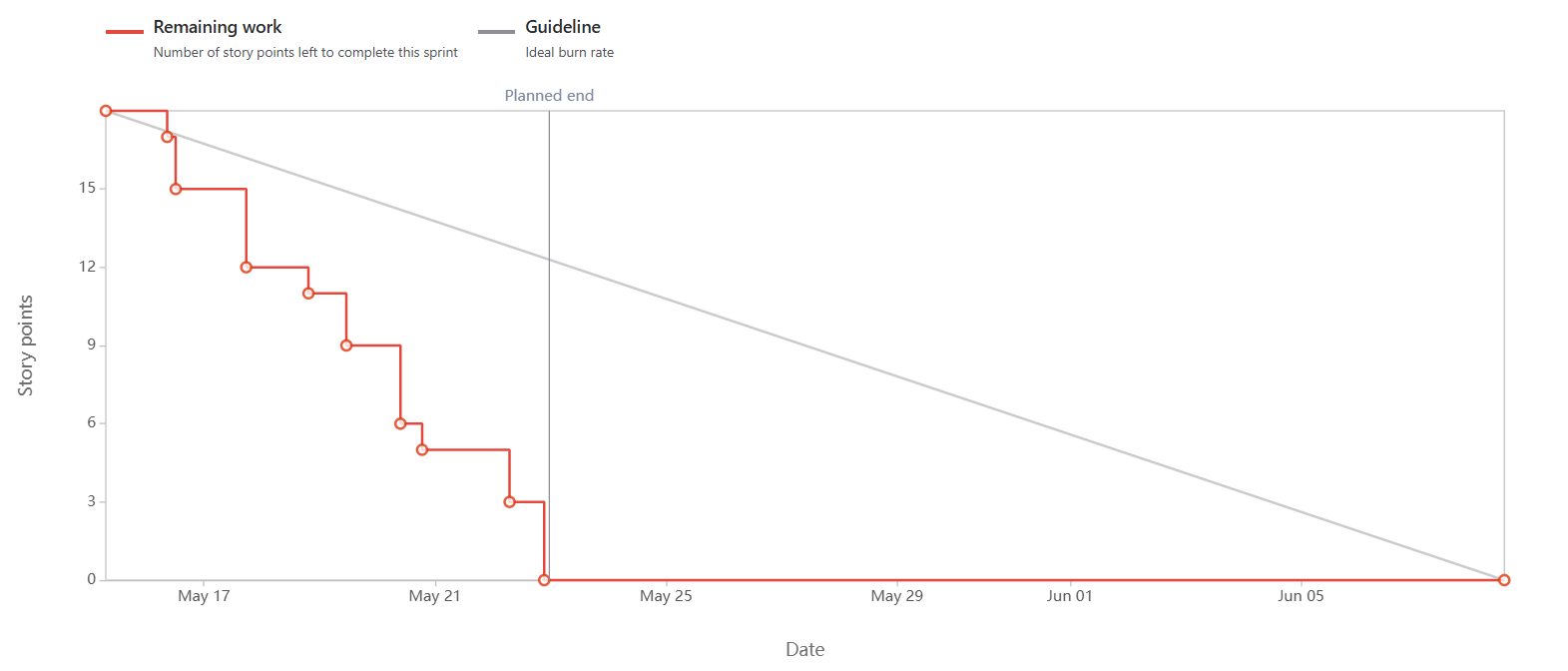
Ngày kết thúc: 22/05/2025

Mục tiêu: Hoàn thiện các chức năng cơ bản liên quan đến việc quản lý và tìm kiếm bài hát trong ứng dụng.



Hình 5.16 Các task của Sprint 3

**Burndown chart của Sprint 3:**



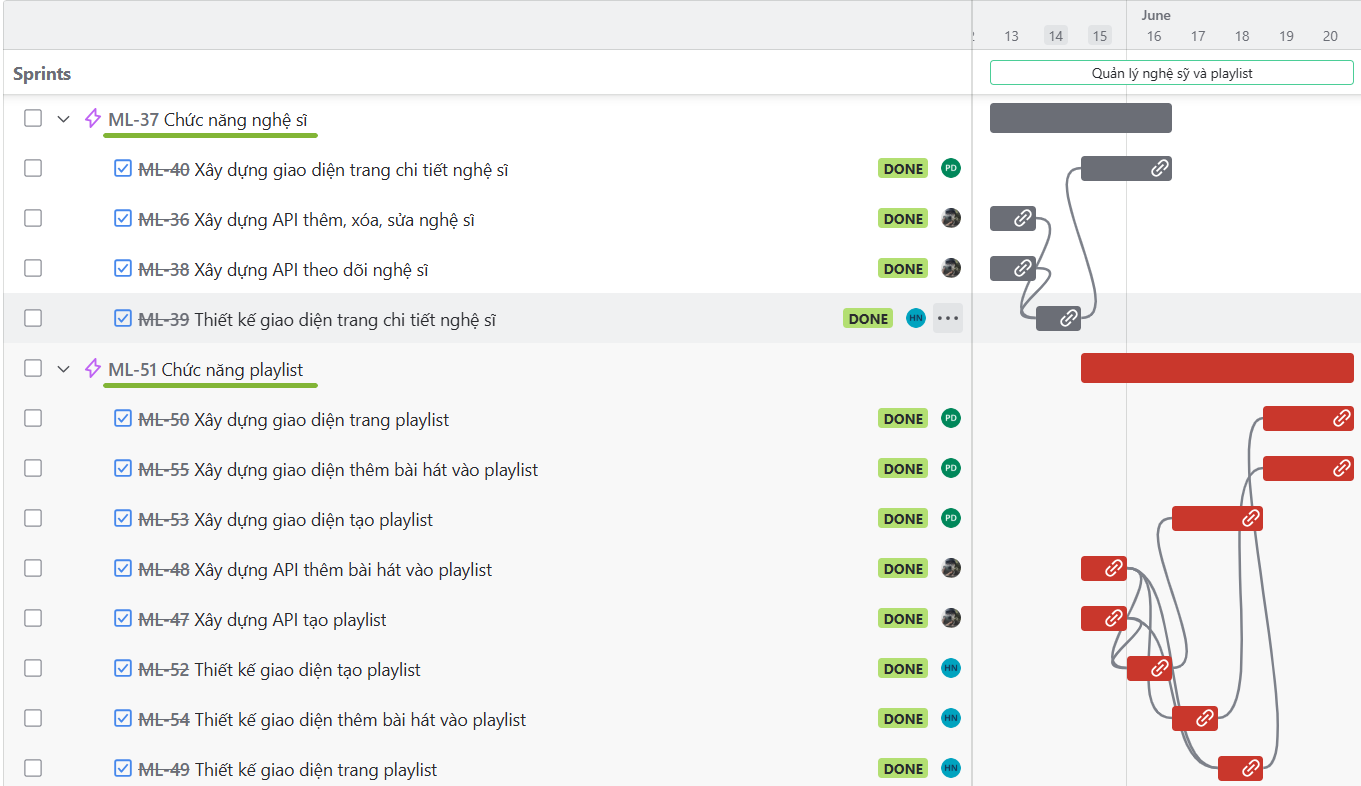
Hình 5.17 Burndown chart của Sprint 3

**Sprint 4: Quản lý nghệ sĩ và playlist**

Ngày bắt đầu: 13/06/2025

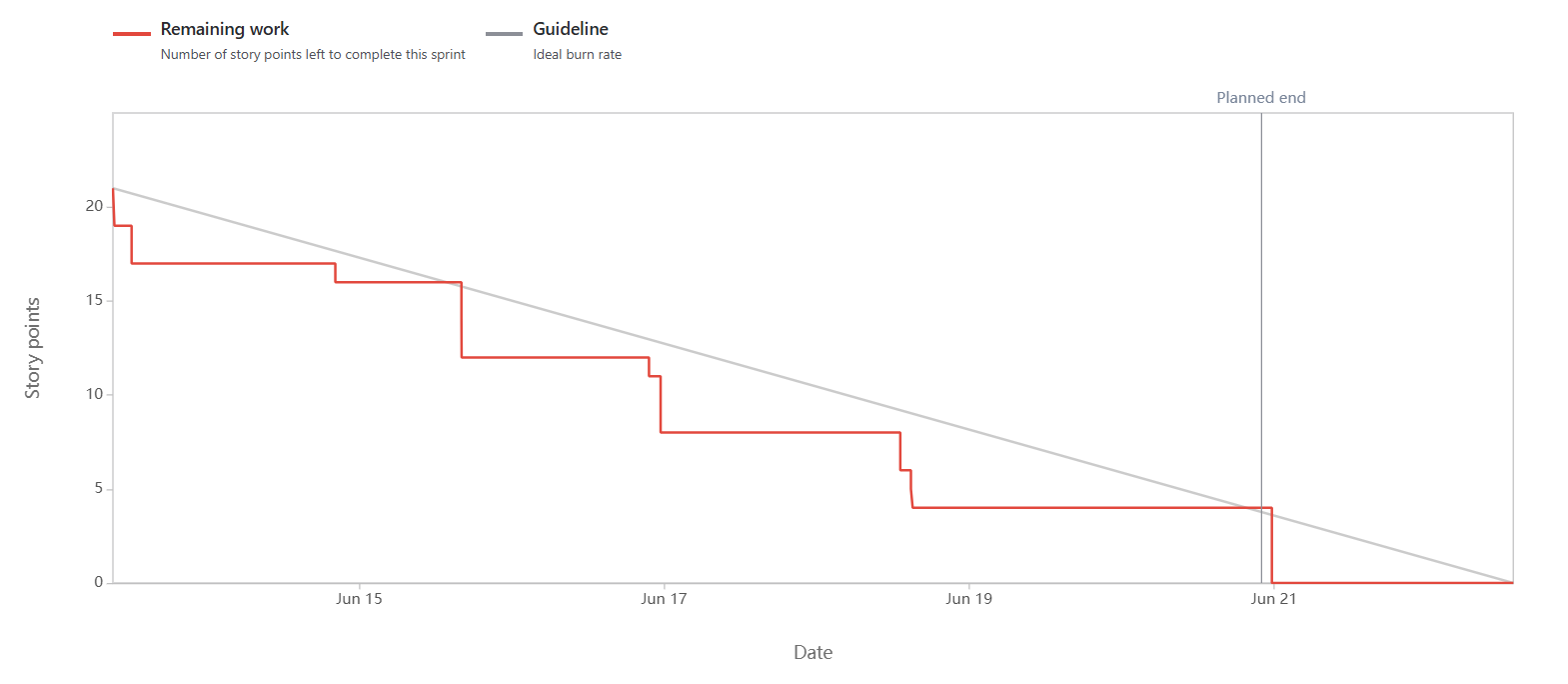
Ngày kết thúc: 20/06/2025

Mục tiêu: Hoàn thiện các chức năng cơ bản liên quan đến việc quản lý nghệ sĩ và playlist trong ứng dụng.



Hình 5.18 Các task của Sprint 4

**Burndown chart của Sprint 4:**



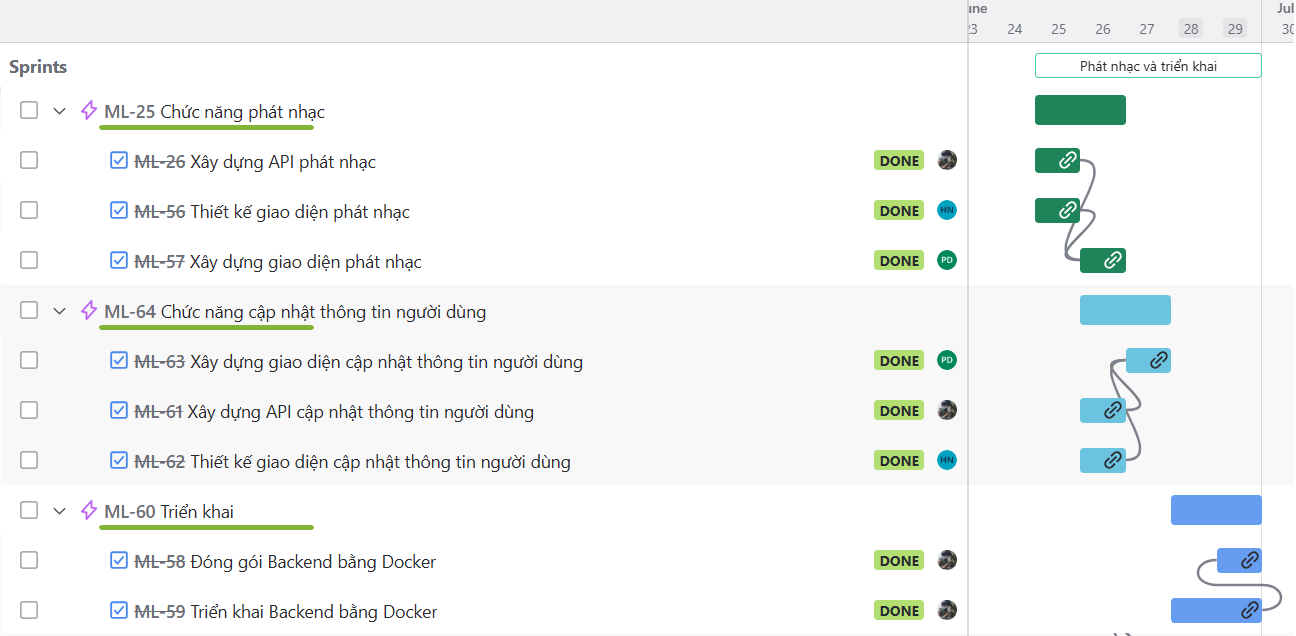
Hình 5.19 Burndown chart của Sprint 4

**Sprint 5: Phát nhạc và triển khai**

Ngày bắt đầu: 25/06/2025

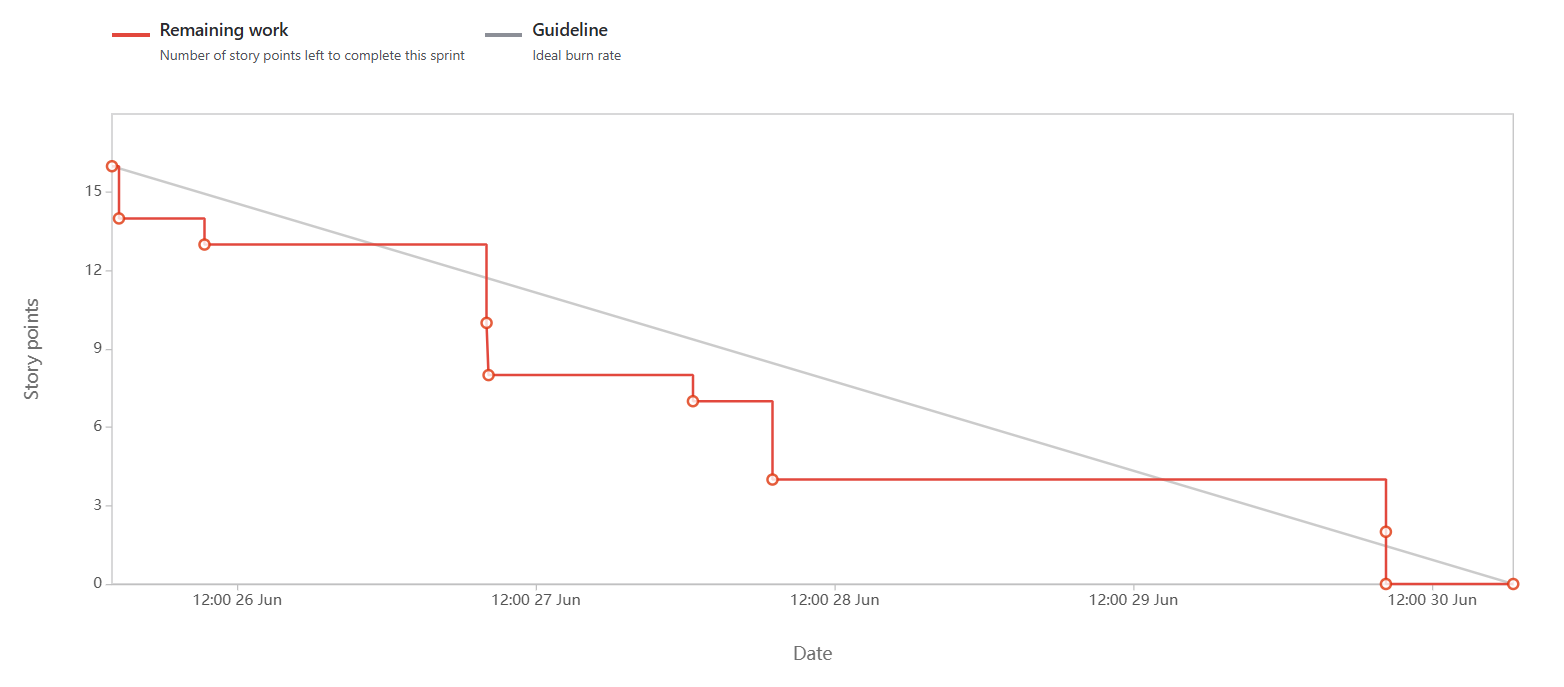
Ngày kết thúc: 29/06/2025

Mục tiêu: Hoàn thiện các chức năng liên quan đến phát nhạc và cập nhật thông tin người dùng. Triển khai Backend bằng Docker lên VPS.

****

Hình 5.20 Các task của Sprint 5

**Burndown chart của Sprint 5:**



Hình 5.21 Burndown chart của Sprint 5

# KIỂM THỬ

## 6.1 Chiến lược kiểm thử và công cụ sử dụng

### Kiểm thử API thủ công (Postman)

Để đảm bảo các API của hệ thống hoạt động đúng chức năng và trả về dữ liệu chính xác, nhóm đã sử dụng Postman – một công cụ kiểm thử API phổ biến – để thực hiện kiểm thử thủ công. Các bước tiến hành gồm:

- Gửi yêu cầu HTTP (GET, POST, PUT, DELETE) tới các endpoint trong API như /api/auth/login, /api/users, /api/playlists,...

- Kiểm tra mã phản hồi HTTP (status code) để đảm bảo API xử lý đúng (200 OK, 201 Created, 401 Unauthorized, 400 Bad Request, v.v.).

- Kiểm tra nội dung phản hồi (response body) để xác nhận dữ liệu trả về đúng định dạng và giá trị mong muốn.

- Kiểm tra các trường hợp đặc biệt (edge cases) như đăng nhập sai thông tin, thiếu token, dữ liệu không hợp lệ.

### Tự động hóa kiểm thử (CI/CD trên Github Actions)

Song song với kiểm thử thủ công, dự án cũng áp dụng kiểm thử tự động trong quy trình CI/CD (Continuous Integration / Continuous Deployment) sử dụng GitHub Actions. Mỗi khi có code mới được đẩy lên nhánh main, quy trình tự động sẽ được kích hoạt:

Bước 1: Checkout mã nguồn từ GitHub.

Bước 2: Thiết lập môi trường Docker và build các container cho backend và cơ sở dữ liệu.

Bước 3: Chạy các container và chờ API khởi động hoàn tất.

Bước 4: Gửi yêu cầu kiểm tra đến endpoint /swagger/index.html để xác nhận API đang hoạt động.

Bước 5: Nếu API phản hồi đúng, quy trình CI sẽ tiếp tục triển khai lên VPS qua SSH: pull code mới, build lại và khởi chạy lại ứng dụng.

## 6.2 Kết quả kiểm thử API

### Kiểm thử xác thực

**Đăng nhập:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 6.1 Kết quả kiểm thử API đăng nhập

**Đăng ký:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 6.2 Kết quả kiểm thử API đăng ký

### Kiểm thử quản lý bài hát

**Lấy danh sách tất cả bài hát:**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 6.3 Kết quả kiểm thử API lấy danh sách tất cả bài hát

**Thêm bài hát:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 6.4 Kết quả kiểm thử API thêm bài hát

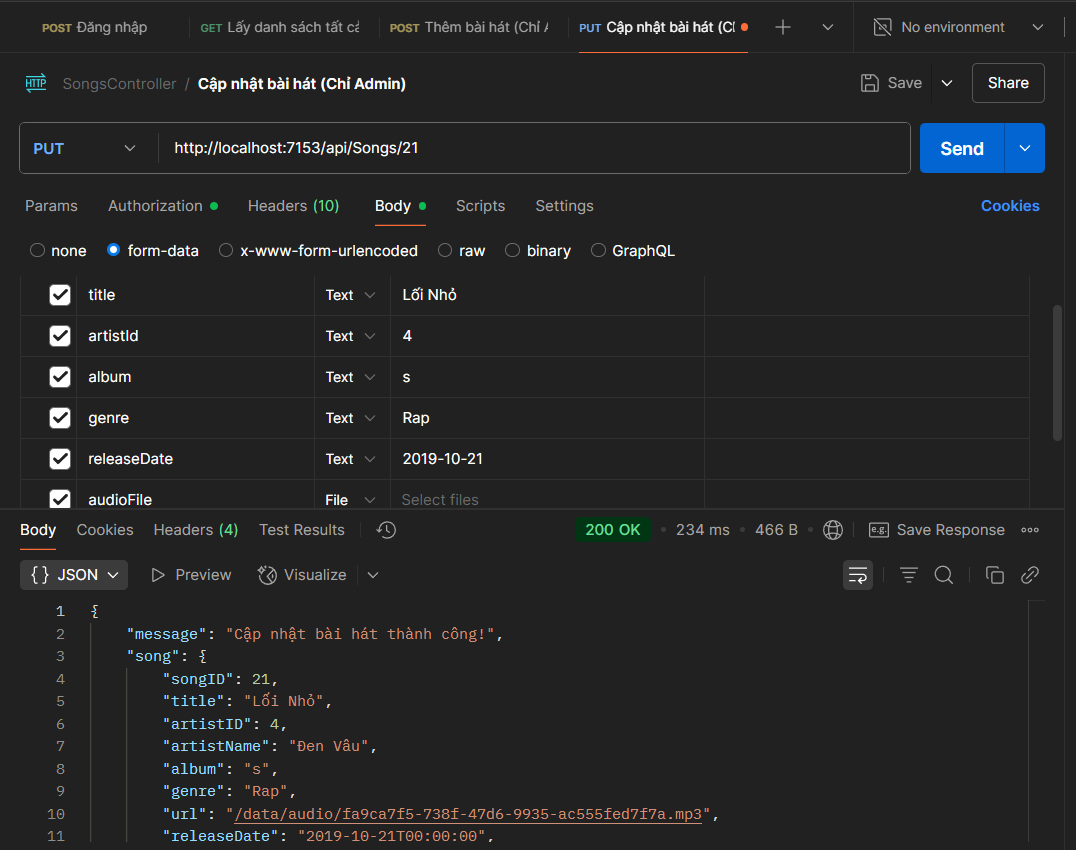
**Xóa bài hát:**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 6.5 Kết quả kiểm thử API xóa bài hát

**Cập nhật bài hát:**



Hình 6.6 Kết quả kiểm thử API cập nhật bài hát

**Phát bài hát:**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 6.7 Kết quả kiểm thử API phát bài hát

# CHƯƠNG 7: ĐÁNH GIÁ VÀ KIẾT LUẬN

## 7.1 Kết quả đạt được

Qua quá trình thực hiện đồ án, nhóm đã đạt được nhiều kết quả tích cực về cả mặt kỹ thuật lẫn quy trình phát triển phần mềm. Về chức năng, hệ thống Melodify đã xây dựng thành công một website nghe nhạc trực tuyến với đầy đủ các tính năng cơ bản như nghe nhạc, tìm kiếm bài hát, quản lý danh sách phát (playlist), thêm bài hát vào mục yêu thích và đăng ký/đăng nhập người dùng với phân quyền rõ ràng giữa người dùng thường và quản trị viên. Giao diện người dùng được thiết kế bằng ReactJS, đảm bảo tính thẩm mỹ và dễ sử dụng. Phần backend được xây dựng bằng .NET Core Web API, xử lý các yêu cầu từ frontend một cách hiệu quả và an toàn. Dữ liệu được lưu trữ và truy vấn thông qua hệ quản trị cơ sở dữ liệu SQL Server, đảm bảo tính toàn vẹn và hiệu suất truy cập. Ngoài ra, nhóm cũng đã triển khai CI/CD tự động thông qua GitHub Actions, giúp kiểm tra và đóng gói hệ thống mỗi khi có thay đổi mới, đồng thời sử dụng Docker để đóng gói và triển khai toàn bộ ứng dụng một cách dễ dàng và nhất quán giữa các môi trường. Đây là một bước tiến quan trọng trong việc áp dụng các công cụ và quy trình DevOps hiện đại vào dự án phần mềm thực tế.

## 7.2 Hạn chế

Mặc dù đạt được nhiều thành tựu đáng khích lệ, dự án Melodify vẫn tồn tại một số hạn chế nhất định cần được cải thiện trong tương lai. Trước hết, hệ thống hiện tại mới chỉ được triển khai chạy cục bộ bằng Docker Compose và đã triển khai lên VPS thông qua Github Actions, chưa được đưa lên môi trường sản xuất thực tế như các nền tảng đám mây (Azure, AWS), do đó chưa thể kiểm chứng độ ổn định và hiệu suất khi có lượng người dùng lớn truy cập đồng thời. Thứ hai, tính năng upload bài hát hiện chưa mở rộng cho người dùng cuối mà chỉ giới hạn trong vai trò quản trị viên với thao tác thêm bài hát trực tiếp vào cơ sở dữ liệu. Ngoài ra, các biện pháp bảo mật mới chỉ dừng ở mức cơ bản như phân quyền và kiểm tra đầu vào. Hơn nữa, hệ thống phát nhạc hiện vẫn bị ngắt khi người dùng chuyển trang, do chưa áp dụng kỹ thuật giữ nguyên player tại layout gốc. Cuối cùng, giao diện còn thiếu sự tương thích trên thiết bị di động, gây bất tiện khi sử dụng trên điện thoại hoặc máy tính bảng.

## 7.3 Hướng phát triển

Nhằm hoàn thiện và phát triển hệ thống Melodify một cách toàn diện hơn trong tương lai, nhóm đề xuất một số định hướng cải tiến cả về mặt chức năng lẫn kỹ thuật. Trước hết, cần triển khai hệ thống lên môi trường thực tế, chẳng hạn như sử dụng dịch vụ cloud (Azure, AWS, Heroku), đồng thời gắn domain riêng để phục vụ người dùng rộng rãi. Tiếp theo, hệ thống sẽ được tích hợp tính năng **phát nhạc liên tục khi chuyển trang**, bằng cách áp dụng kỹ thuật lưu player ở cấp layout hoặc sử dụng Redux/Context để giữ trạng thái phát nhạc toàn cục. Về mặt bảo mật, nhóm dự định áp dụng xác thực hai lớp (2FA) để tăng cường an toàn thông tin. Ngoài ra, giao diện sẽ được tối ưu theo hướng **mobile-first và responesive mobile** nhằm nâng cao trải nghiệm trên thiết bị di động. Một hướng phát triển quan trọng khác là cho phép người dùng tự upload nhạc cá nhân, chia sẻ playlist, tương tác bằng cách bình luận, đánh giá bài hát, từ đó tăng tính cộng đồng cho hệ thống. Cuối cùng, nhóm mong muốn tích hợp thêm hệ thống phân tích dữ liệu nghe nhạc như thống kê lượt nghe, top bài hát theo thời gian, giúp cải thiện trải nghiệm người dùng và hỗ trợ định hướng nội dung phát triển phù hợp.

**DANH MỤC TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | W3Schools, "W3Schools," 2025. [Online]. Available: https://www.w3schools.com/. [Accessed 20 6 2025]. |
| [2] | React, "React," 2025. [Online]. Available: https://react.dev/. [Accessed 18 6 2025]. |
| [3] | Microsoft, "Welcome to .NET," Microsoft, 2025. [Online]. Available: https://learn.microsoft.com/vi-vn/dotnet/welcome. [Accessed 12 6 2025]. |
| [4] | Scrum.org, "Scrum.org," 2025. [Online]. Available: https://www.scrum.org/. [Accessed 11 6 2025]. |