**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: TỔNG QUAN 10](#_Toc199749490)

[1.1 Lý do chọn đề tài 10](#_Toc199749491)

[1.2 Mục tiêu đề tài 10](#_Toc199749492)

[1.3 Bảng phân công công việc 11](#_Toc199749493)

[CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG VÀ XÁC ĐỊNH YÊU CẦU 12](#_Toc199749494)

[2.1 Mô tả bài toán 12](#_Toc199749495)

[2.2 Các yêu cầu chức năng của hệ thống 13](#_Toc199749496)

[2.2.1 Các yêu cầu về lưu trữ 13](#_Toc199749497)

[2.2.2 Các yêu cầu về tra cứu 13](#_Toc199749498)

[2.2.3 Các yêu cầu về kết xuất, thống kê 14](#_Toc199749499)

[2.2.4 Các yêu cầu về tính toán 14](#_Toc199749500)

[2.3 Các yêu cầu phi chức năng của hệ thống 14](#_Toc199749501)

[2.3.1 Các yêu cầu về bảo mật 14](#_Toc199749502)

[2.3.2 Các yêu cầu về hiệu suất 14](#_Toc199749503)

[2.3.3 Khả năng mở rộng 15](#_Toc199749504)

[2.3.4 Tính tương thích 15](#_Toc199749505)

[2.3.5 Tính duy trì 15](#_Toc199749506)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 16](#_Toc199749507)

[3.1 Phân tích và thiết kế thành phần dữ liệu 16](#_Toc199749508)

[3.1.1 Mô hình dữ liệu mức quan niệm 16](#_Toc199749509)

[3.1.2 Mô hình dữ liệu mức luận lý 16](#_Toc199749510)

[3.1.3 Mô hình dữ liệu mức vật lý 17](#_Toc199749511)

[3.1.4 Lược đồ cơ sở dữ liệu SQL Server 17](#_Toc199749512)

[3.2 Phân tích và thiết kế thành phần xử lý 18](#_Toc199749513)

[3.2.1 Mô hình phân cấp chức năng 18](#_Toc199749514)

[3.2.2 Mô hình UseCase 18](#_Toc199749515)

[3.2.3 Mô tả các UseCase 18](#_Toc199749516)

[CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM 21](#_Toc199749517)

[4.1 Tạo cơ sở dữ liệu 21](#_Toc199749518)

[4.2 Phân quyền 23](#_Toc199749519)

[4.3 Cài đặt Trigger, View, Store Procedure 25](#_Toc199749520)

[4.3.1 Trigger 25](#_Toc199749521)

[4.3.2 View 32](#_Toc199749522)

[4.3.3 Store Procedure 34](#_Toc199749523)

[4.4 Sao lưu, phục hồi dữ liệu 43](#_Toc199749524)

[4.4.1 Sao lưu 43](#_Toc199749525)

[4.4.2 Phục hồi 44](#_Toc199749526)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 45](#_Toc199749527)

[5.1 Kết quả đạt được 45](#_Toc199749528)

[5.2 Hạn chế 45](#_Toc199749529)

[5.3 Hướng phát triển 45](#_Toc199749530)

**DANH MỤC HÌNH ẢNH**

[Hình 3.1 Mô hình dữ liệu mức quan niệm 16](#_Toc199750408)

[Hình 3.2 Mô hình dữ liệu mức luận lý 16](#_Toc199750409)

[Hình 3.3 Mô hình dữ liệu mức vật lý 17](#_Toc199750410)

[Hình 3.4 Lược đồ cơ sở dữ liệu SQL Server 17](#_Toc199750411)

[Hình 3.5 Mô hình phân cấp chức năng 18](#_Toc199750412)

[Hình 3.6 Mô hình UseCase 18](#_Toc199750413)

[Hình 4.1 Bảng vai trò 21](#_Toc199750414)

[Hình 4.2 Bảng người dùng 21](#_Toc199750415)

[Hình 4.3 Bảng tour 21](#_Toc199750416)

[Hình 4.4 Bảng đặt tour 22](#_Toc199750417)

[Hình 4.5 Bảng thanh toán 22](#_Toc199750418)

[Hình 4.7 Bảng đánh giá 23](#_Toc199750419)

[Hình 4.8 Cài đặt phân quyền khách hàng 24](#_Toc199750420)

[Hình 4.9 Cài đặt phân quyền nhân viên 24](#_Toc199750421)

[Hình 4.10 Cài đặt phân quyền quản trị viên 25](#_Toc199750422)

[Hình 4.11 Cài đặt trigger kiểm tra số lượng người tham gia tour 26](#_Toc199750423)

[Hình 4.12 Kết quả trigger kiểm tra số lượng người tham gia tour 26](#_Toc199750424)

[Hình 4.13 Cài đặt trigger kiểm tra định dạng email 27](#_Toc199750425)

[Hình 4.14 Kết quả trigger kiểm tra định dạng email 27](#_Toc199750426)

[Hình 4.15 Cài đặt trigger kiểm tra đánh giá 27](#_Toc199750427)

[Hình 4.16 Kết quả trigger kiểm tra đánh giá 28](#_Toc199750428)

[Hình 4.17 Cài đặt trigger kiểm tra định dạng số điện thoại 28](#_Toc199750429)

[Hình 4.18 Kết quả trigger kiểm tra định dạng số điện thoại 28](#_Toc199750430)

[Hình 4.19 Cài đặt trigger kiểm tra tên tour 29](#_Toc199750431)

[Hình 4.20 Kết quả trigger kiểm tra tên tour 29](#_Toc199750432)

[Hình 4.21 Cài đặt trigger kiểm tra điểm khởi hành và điểm đến 29](#_Toc199750433)

[Hình 4.22 Kết quả trigger kiểm tra điểm khởi hành và điểm đến 30](#_Toc199750434)

[Hình 4.23 Cài đặt trigger kiểm tra ngày khởi hành 30](#_Toc199750435)

[Hình 4.24 Kết quả trigger kiểm tra ngày khởi hành 30](#_Toc199750436)

[Hình 4.25 Cài đặt trigger kiểm tra ngày kết thúc 31](#_Toc199750437)

[Hình 4.26 Kết quả trigger kiểm tra ngày kết thúc 31](#_Toc199750438)

[Hình 4.27 Cài đặt trigger kiểm tra số người tối đa 31](#_Toc199750439)

[Hình 4.28 Kết quả trigger kiểm tra số người tối đa 31](#_Toc199750440)

[Hình 4.29 Cài đặt trigger kiểm tra giá tour 32](#_Toc199750441)

[Hình 4.30 Kết quả trigger kiểm tra giá tour 32](#_Toc199750442)

[Hình 4.31 Cài đặt view thống kê doanh thu theo tour 32](#_Toc199750443)

[Hình 4.32 Kết quả view thống kê doanh thu theo tour 33](#_Toc199750444)

[Hình 4.33 Cài đặt view thống kê khách hàng thường xuyên 33](#_Toc199750445)

[Hình 4.34 Kết quả view thống kê khách hàng thường xuyên 33](#_Toc199750446)

[Hình 4.35 Cài đặt view lịch sử đặt tour của một khách hàng 33](#_Toc199750447)

[Hình 4.36 Kết quả view lịch sử đặt tour của một khách hàng 34](#_Toc199750448)

[Hình 4.37 Cài đặt view toàn bộ thông tin khách hàng 34](#_Toc199750449)

[Hình 4.38 Kết quả view toàn bộ thông tin khách hàng 34](#_Toc199750450)

[Hình 4.39 Cài đặt procedure tạo tour 35](#_Toc199750451)

[Hình 4.40 Kết quả sử dụng procedure tạo tour 35](#_Toc199750452)

[Hình 4.41 Cài đặt procedure sửa tour 36](#_Toc199750453)

[Hình 4.42 Kết quả sử dụng procedure sửa tour 36](#_Toc199750454)

[Hình 4.43 Cài đặt procedure xóa tour 37](#_Toc199750455)

[Hình 4.44 Kết quả sử dụng procedure xóa tour 37](#_Toc199750456)

[Hình 4.45 Cài đặt procedure tìm kiếm tour 37](#_Toc199750457)

[Hình 4.46 Kết quả sử dụng procedure tìm kiếm tour 38](#_Toc199750458)

[Hình 4.47 Cài đặt procedure đăng ký tài khoản 38](#_Toc199750459)

[Hình 4.48 Kết quả sử dụng procedure đăng ký tài khoản 38](#_Toc199750460)

[Hình 4.49 Cài đặt procedure sửa thông tin người dùng 39](#_Toc199750461)

[Hình 4.50 Kết quả sử dụng procedure sửa thông tin người dùng 39](#_Toc199750462)

[Hình 4.51 Cài đặt procedure xóa người dùng 39](#_Toc199750463)

[Hình 4.52 Kết quả sử dụng procedure xóa người dùng 40](#_Toc199750464)

[Hình 4.53 Cài đặt procedure đặt tour 40](#_Toc199750465)

[Hình 4.54 Kết quả sử dụng procedure đặt tour 40](#_Toc199750466)

[Hình 4.55 Cài đặt procedure cập nhật trạng thái thanh toán 41](#_Toc199750467)

[Hình 4.56 Kết quả sử dụng procedure cập nhật trạng thái thanh toán 41](#_Toc199750468)

[Hình 4.57 Cài đặt procedure thêm đánh giá tour 41](#_Toc199750469)

[Hình 4.58 Kết quả sử dụng procedure thêm đánh giá tour 42](#_Toc199750470)

[Hình 4.59 Cài đặt procedure hủy đặt tour 42](#_Toc199750471)

[Hình 4.60 Kết quả sử dụng procedure hủy đặt tour 43](#_Toc199750472)

[Hình 4.61 Cài đặt procedure thống kê doanh thu theo thời gian 43](#_Toc199750473)

[Hình 4.62 Kết quả sử dụng procedure thống kê doanh thu theo thời gian 43](#_Toc199750474)

[Hình 4.63 Cài đặt sao lưu cơ sở dữ liệu 44](#_Toc199750475)

[Hình 4.64 Kết quả sao lưu cơ sở dữ liệu 44](#_Toc199750476)

[Hình 4.65 Phục hồi cơ sở dữ liệu 44](#_Toc199750477)

**DANH MỤC BẢNG BIỂU**

[Bảng 1.1 Bảng phân công công việc 11](#_Toc199750479)

**DANH MỤC CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Từ đầy đủ** |
| **1** | sp | Stored Procedure |
| **2** | vw | View |

# 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.

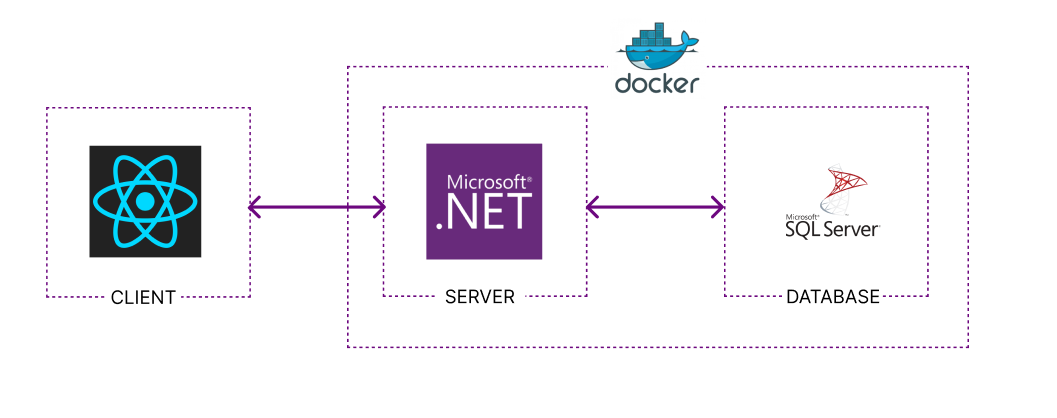
Thông tin người dùng như mật khẩu phải được bảo vệ bằng cách mã hóa, sử dụng các cơ chế xác thực an toàn JWT (JSON Web Token).

# 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.

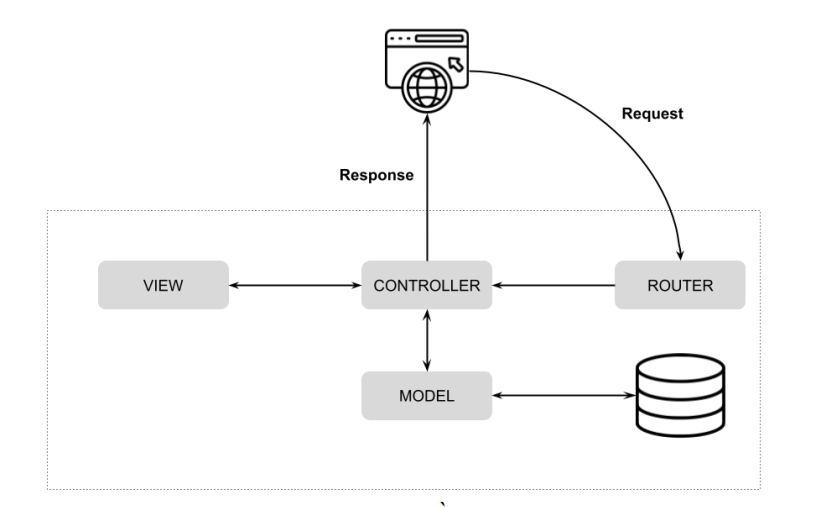
- Tầng Server: Phát triển bằng .NET, xử lý tất cả logic nghiệp vụ của ứng dụng. Đảm nhiệm việc xác thực người dùng, xử lý yêu cầu tìm kiếm, tạo playlist, cập nhật thông tin cá nhân. Giao tiếp với cơ sở dữ liệu để lưu trữ và truy xuất dữ liệu liên quan đến bài hát, nghệ sĩ, tài khoản, playlist, cung cấp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

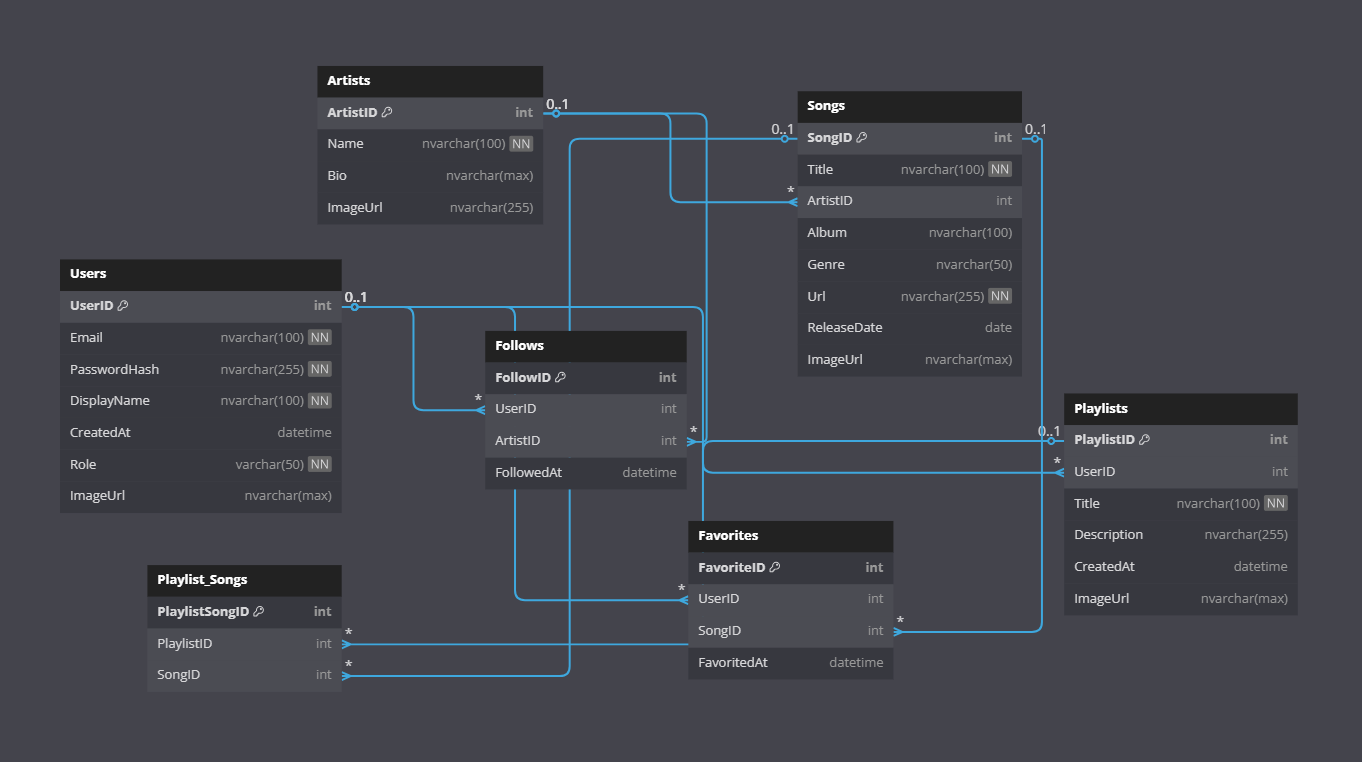
- Playlists: Quản lý danh sách phát nhạc cá nhân của người dùng.

- 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.



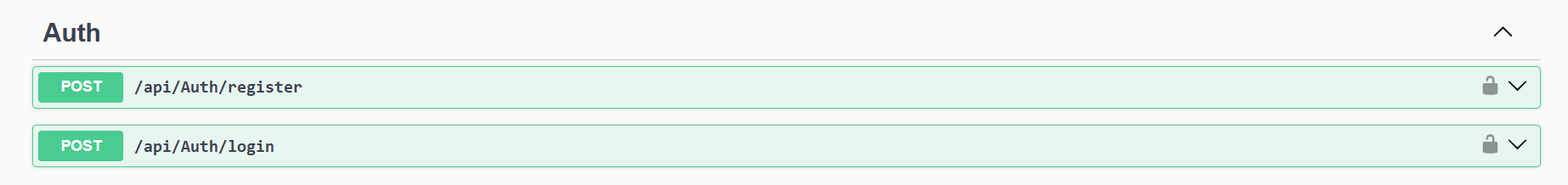
Hình 3.3 Database diagram

## 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:

### Enpoint Auth

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



**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:

{

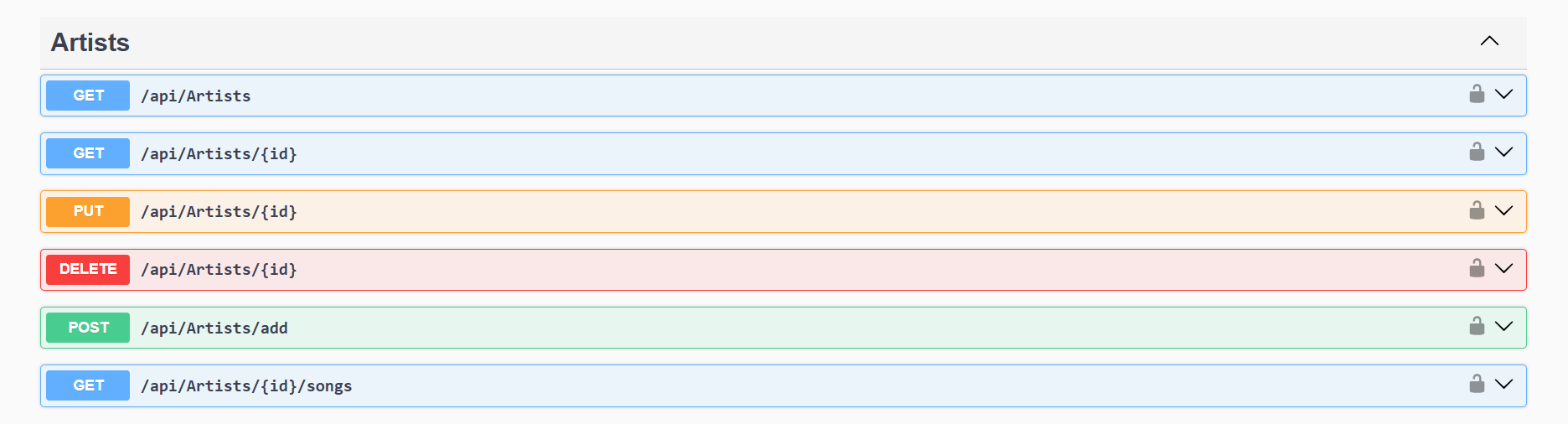
"message": "Đăng nhập thành công!",

"token": "eyJhbGciOiJIUzI1NiIsIn..."

}

### Enpoint Artists

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



**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:

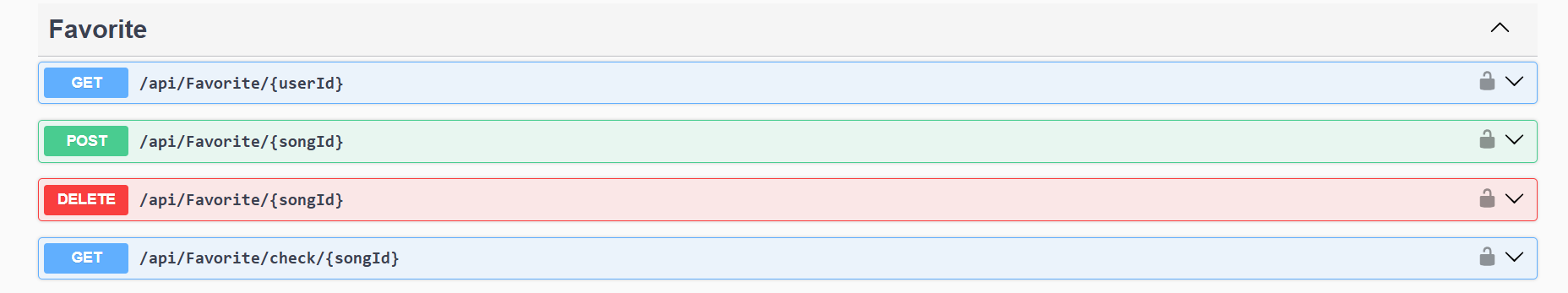
{

"message": "Thêm nghệ sĩ thành công!"

}

### Enpoint Favorite

Enpoint 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.



**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:

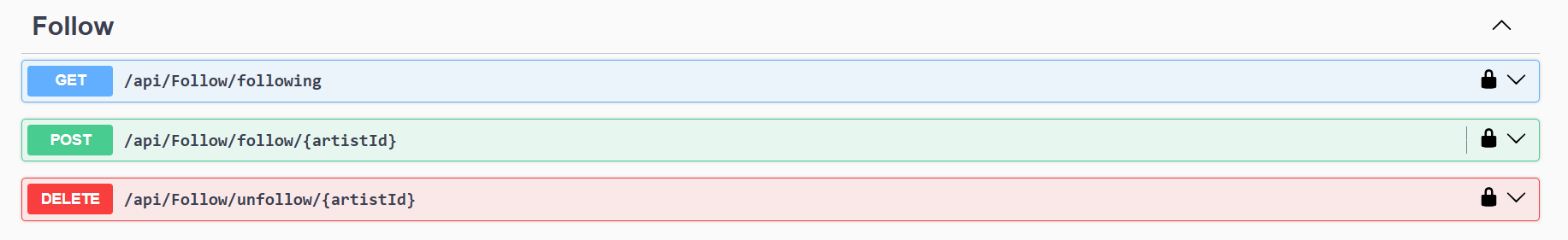
- songId (integer)

Respone body:

true

### Enpoint Follow

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



**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:

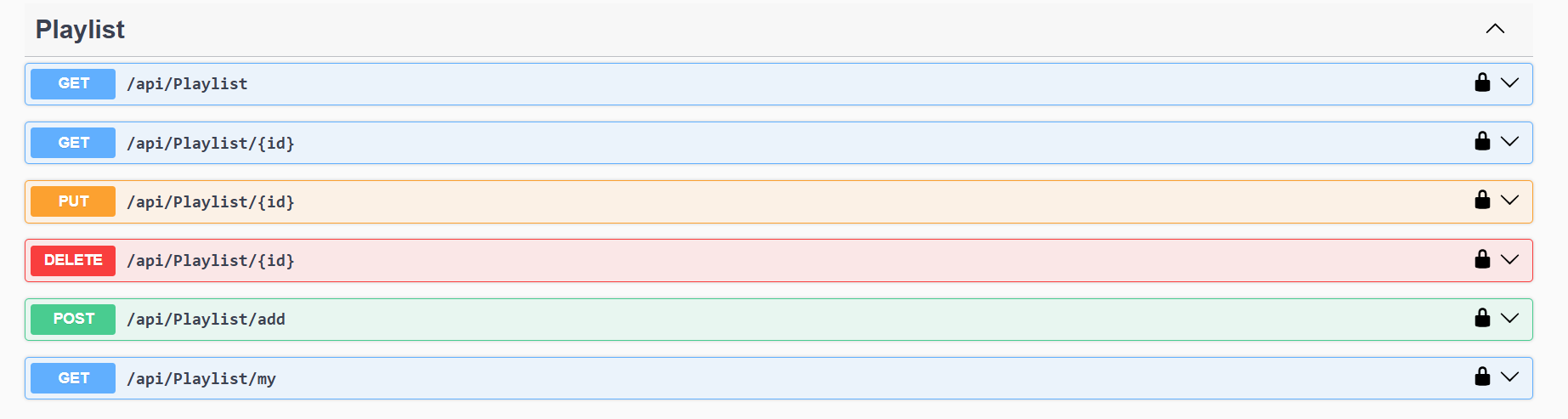
- artistId (integer)

Respone body:

Đã bỏ theo dõi Nghệ sĩ thành công!

### Enpoint Playlist

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

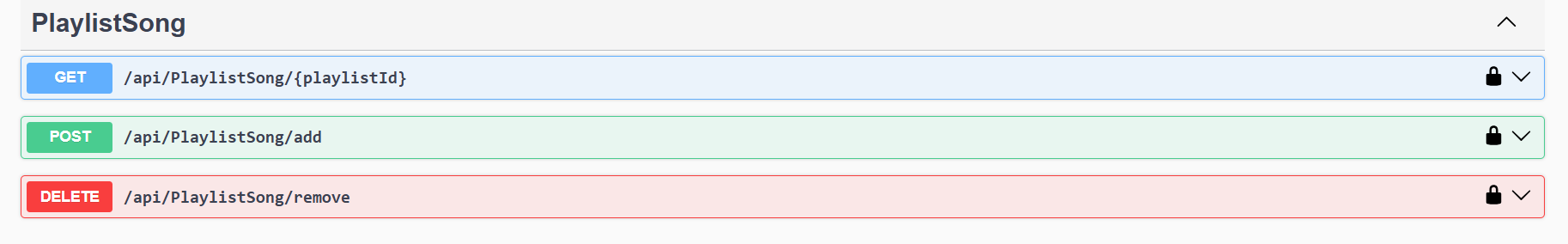
"imageUrl": "/app-asset/img/default-playlist.jpg"

}

]

### Enpoint PlaylistSong

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



**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)

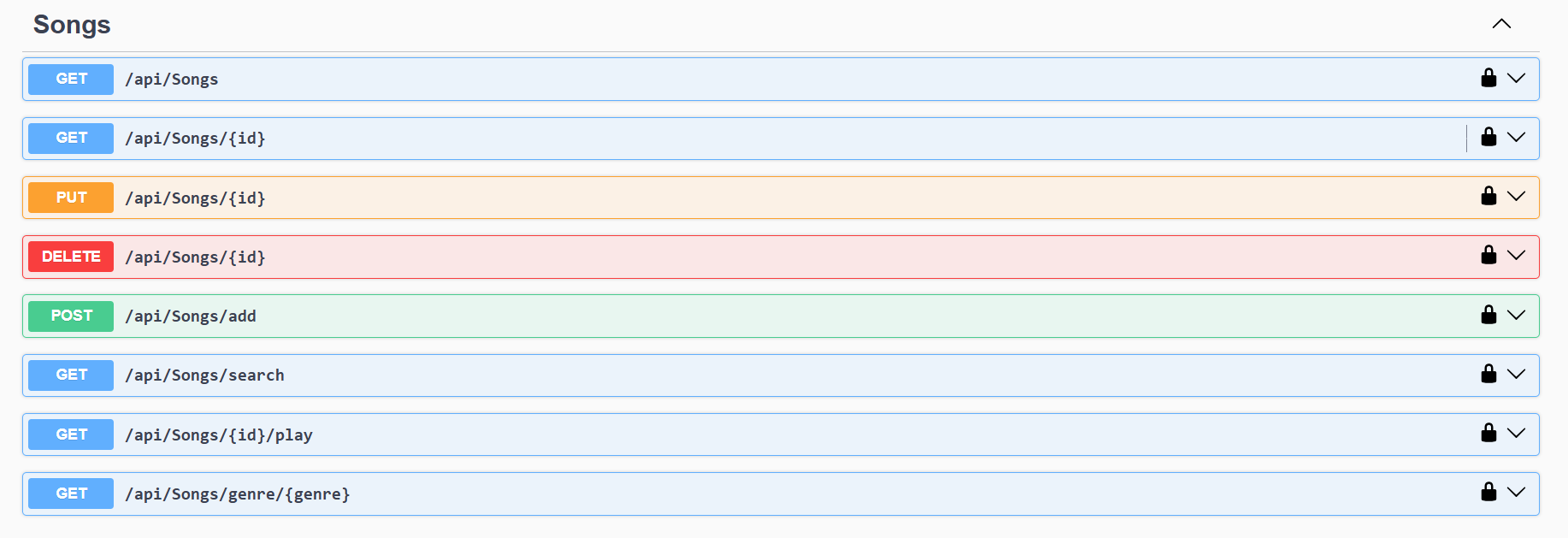
- songId (interger)

Respone body:

Đã xóa bài hát khỏi Playlist thành công!

### Enpoint Songs

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



**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:

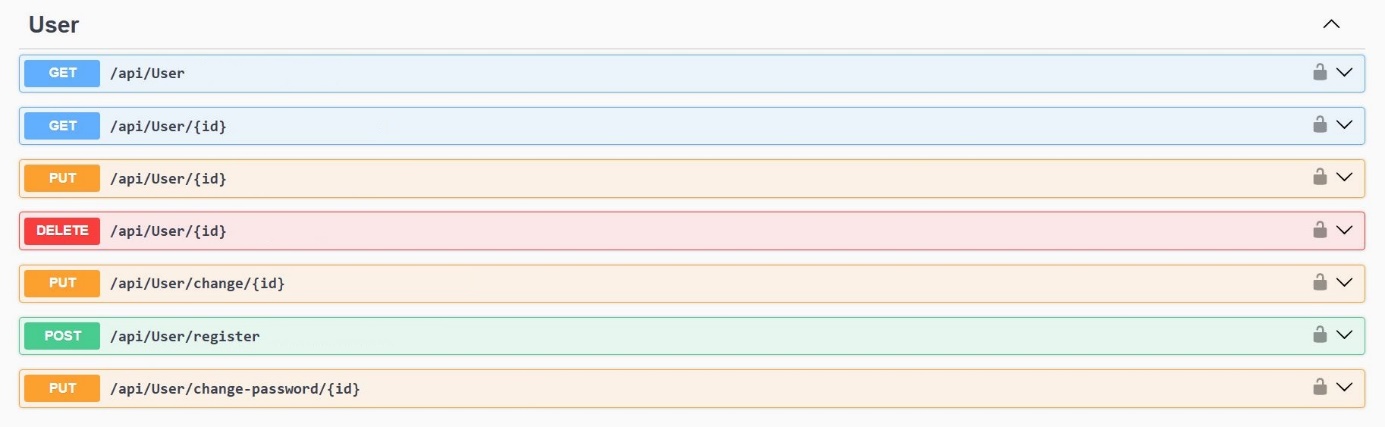
{

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

}

### Enpoint User

Enpoint 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.



**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 (Nguyên)

### 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

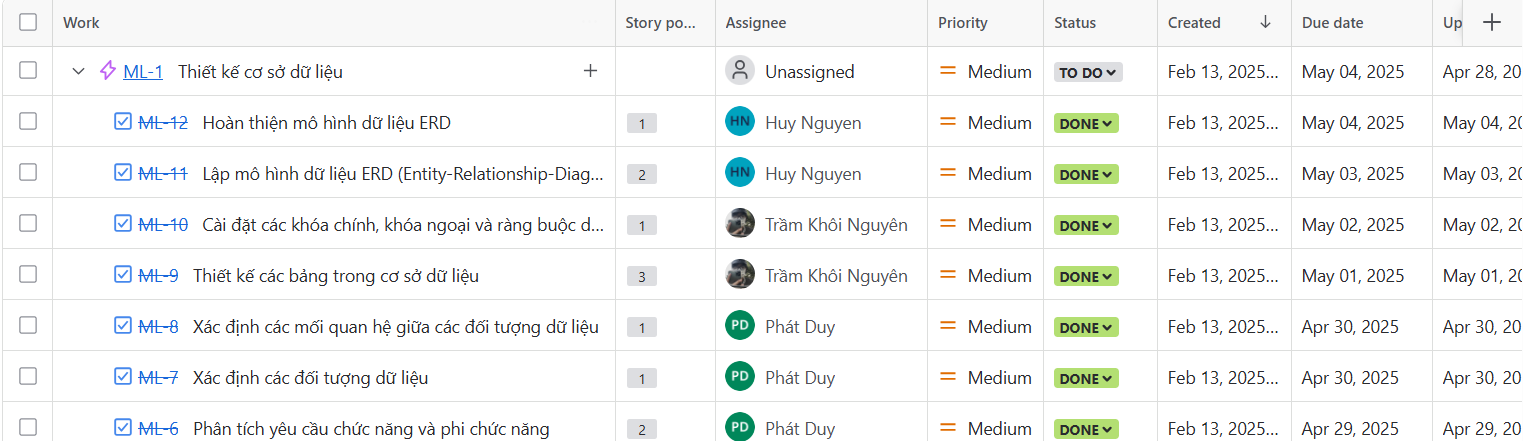
## 4.3 Cấu hình Docker và quy trình triển khai ứng dụng (Huy)

# CHƯƠNG 5: QUẢN LÝ DỰ ÁN (Nguyên)

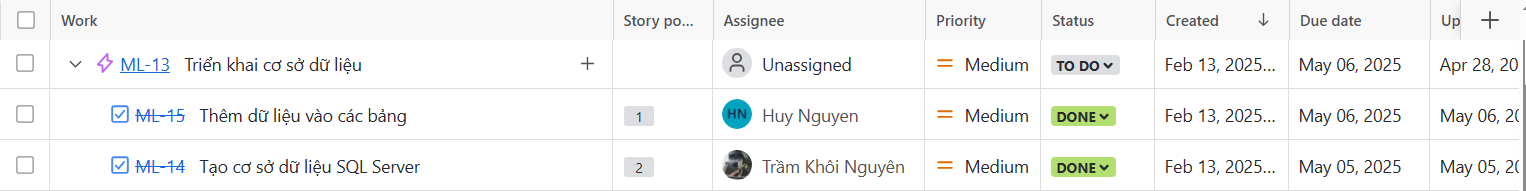
## 5.1 Lập kế hoạch sản phẩm (Product Backlog)

Gồm có 11 epic với các task tương ứng:

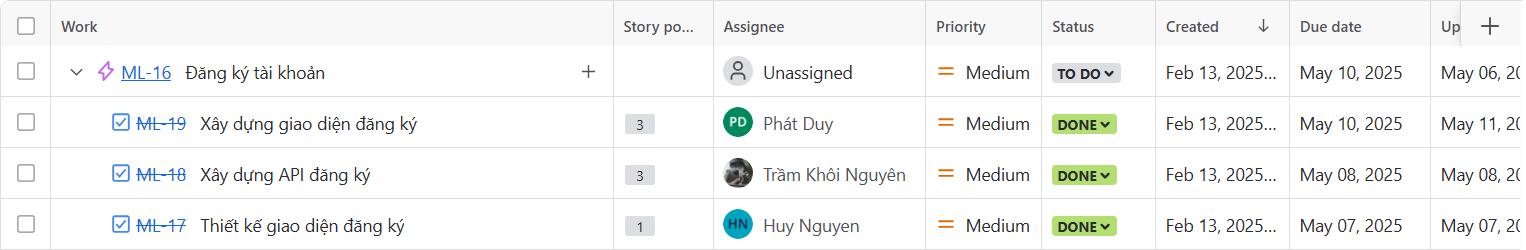
**Epic: Thiết kế cơ sở dữ liệu**

****

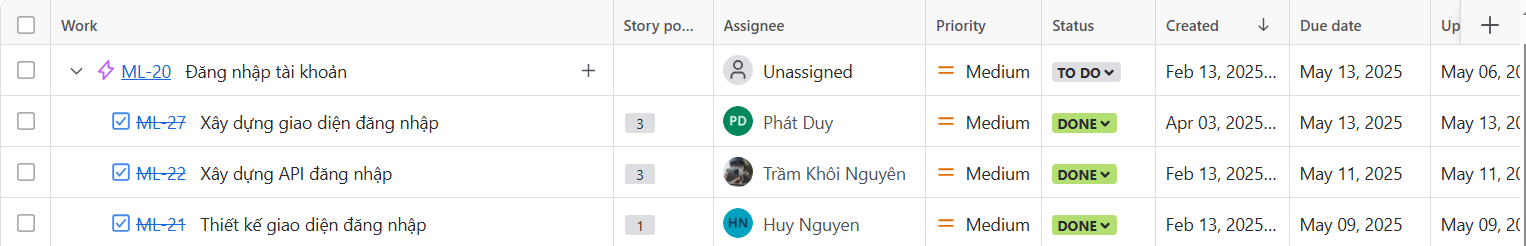
**Epic: Triển khai cơ sở dữ liệu**

****

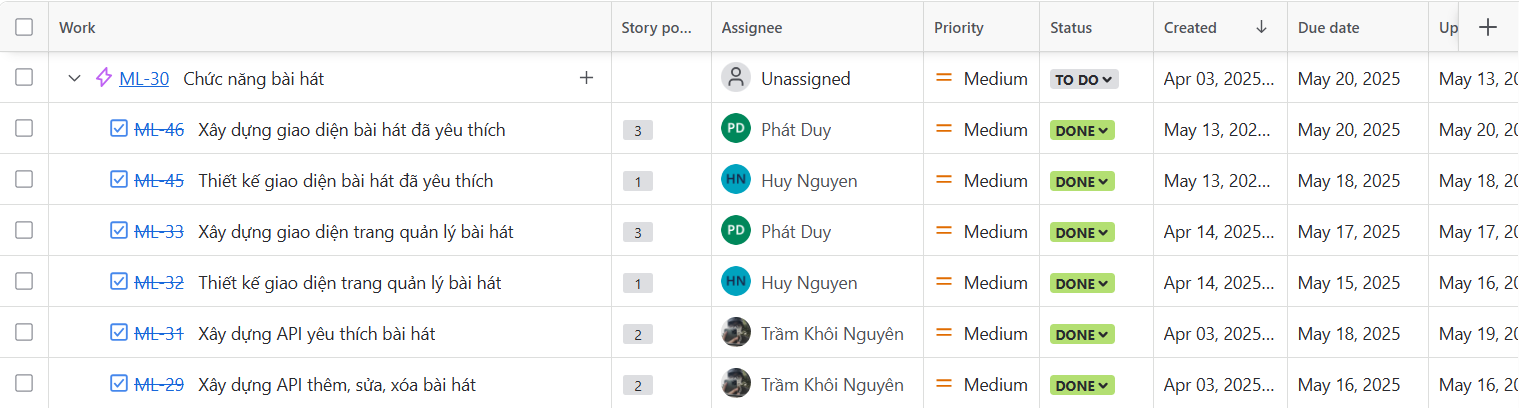
**Epic: Đăng ký tài khoản**

****

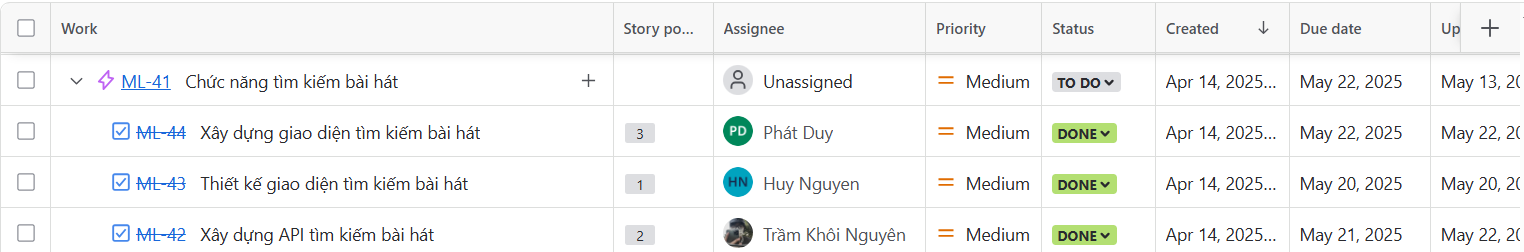
**Epic: Đăng nhập tài khoản**

****

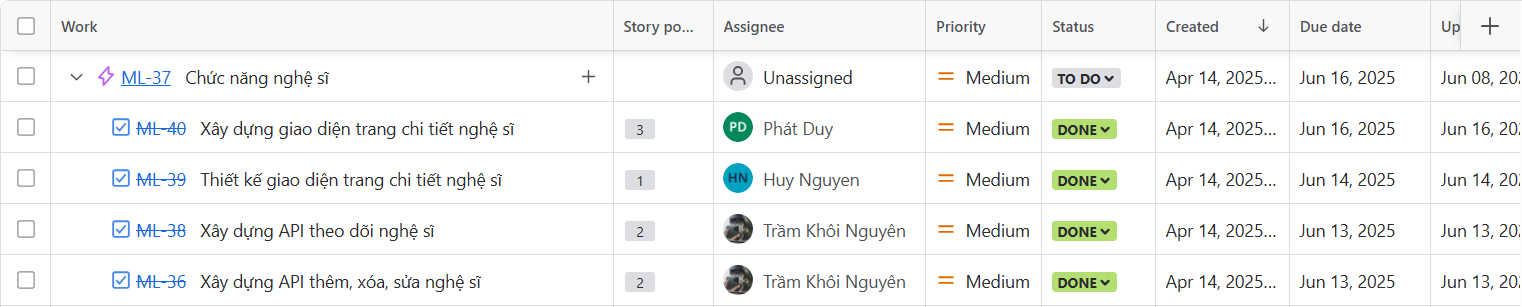
**Epic: Chức năng bài hát**

****

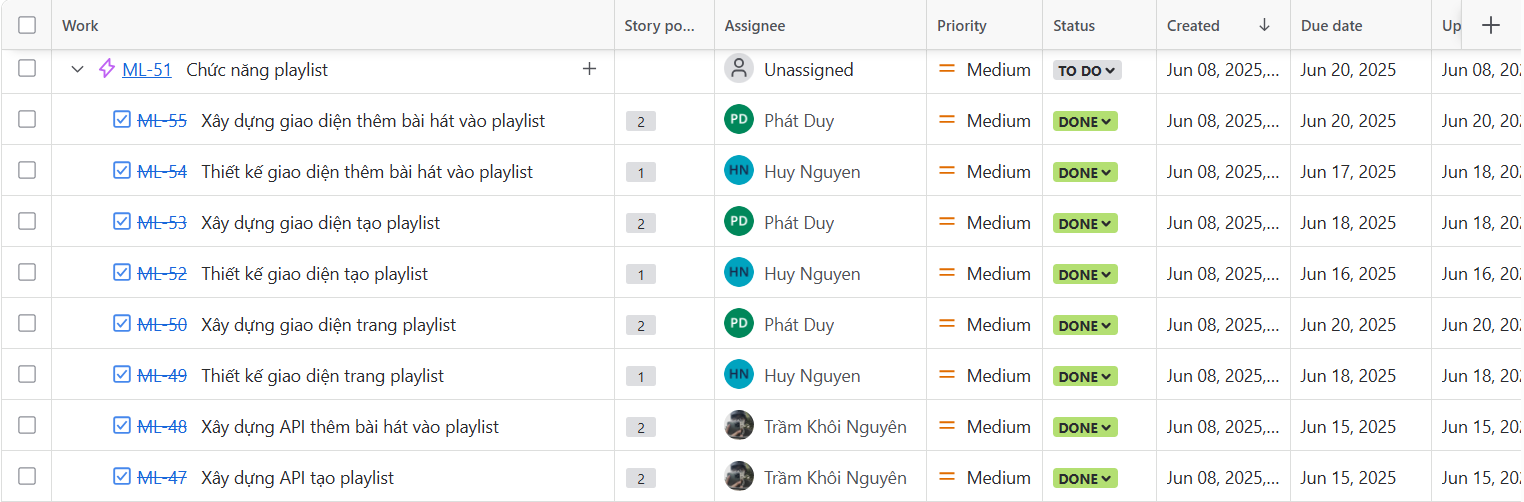
**Epic: Chức năng tìm kiếm bài hát**

****

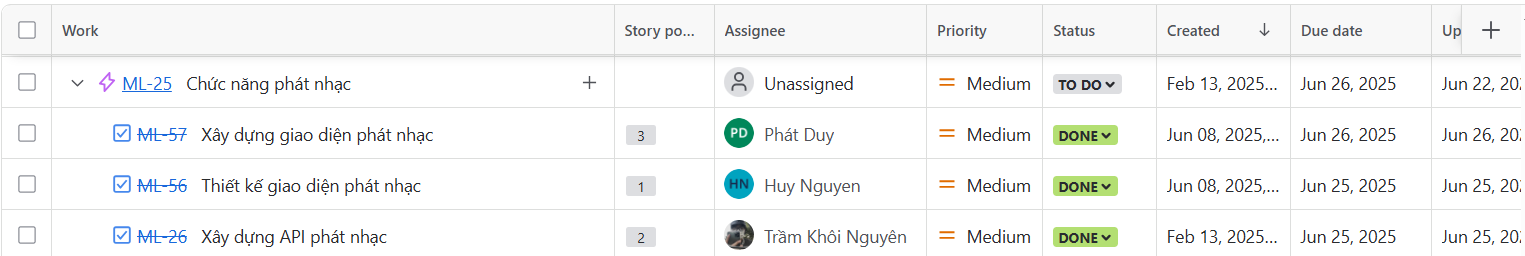
**Epic: Chức năng nghệ sĩ**

****

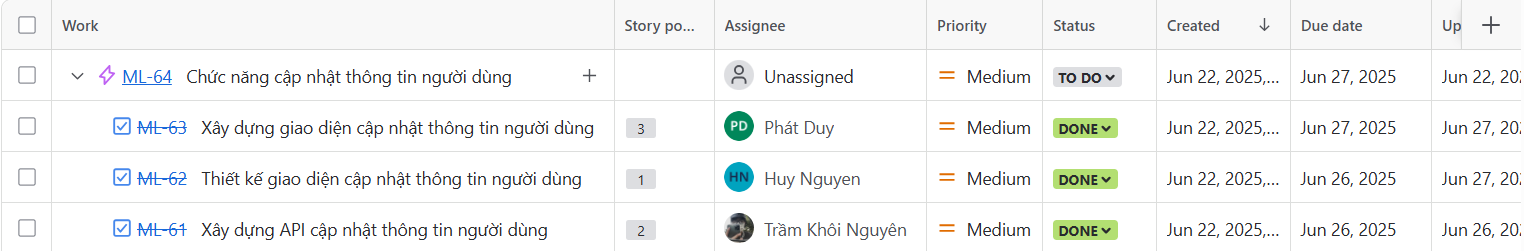
**Epic: Chức năng playlist**

****

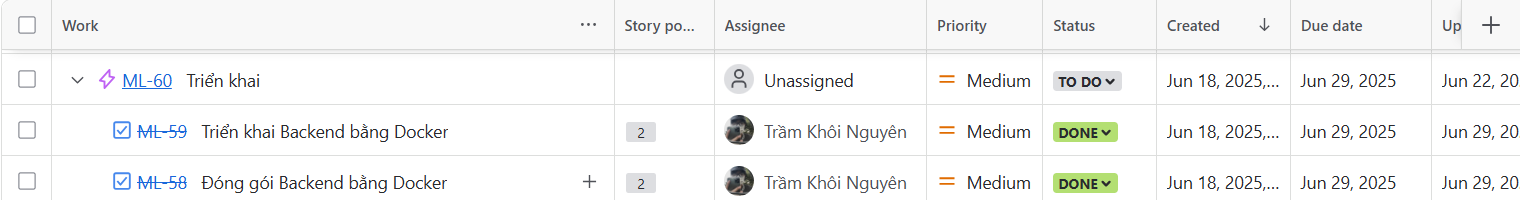
**Epic: Chức năng phát nhạc**

****

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

******

**Epic: Triển khai**

******

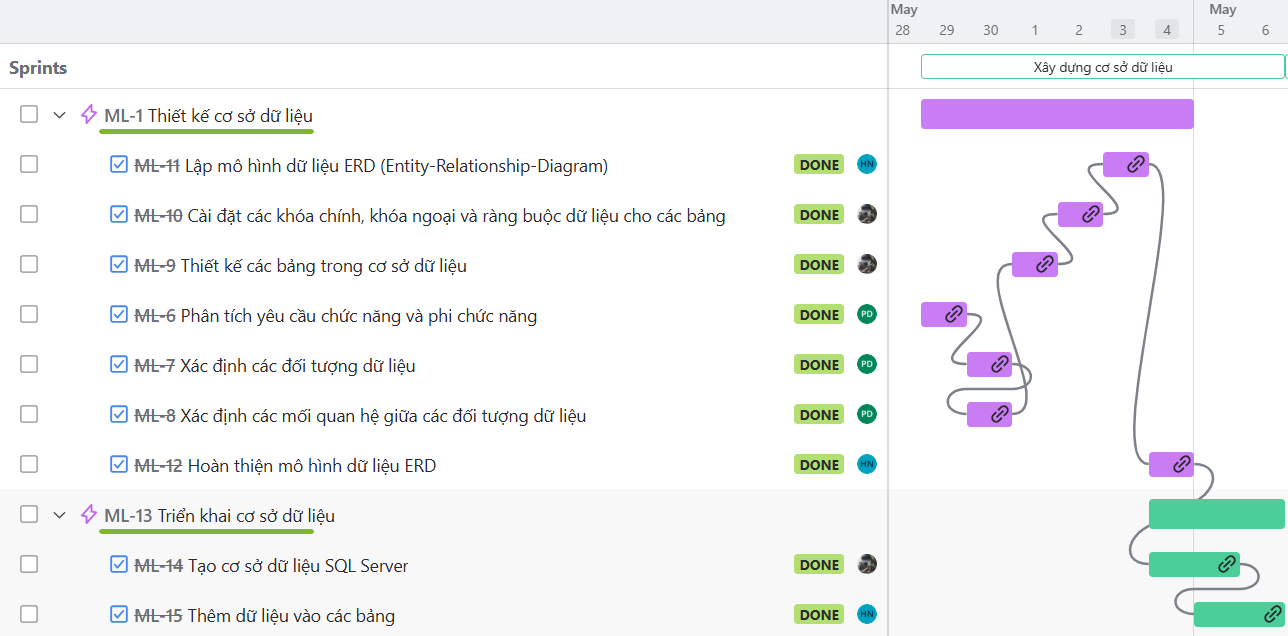
## 5.2 Lập kế hoạch sprint (Sprint Backlog)

**Sprint 1: Xây dựng cơ sở dữ liệu**

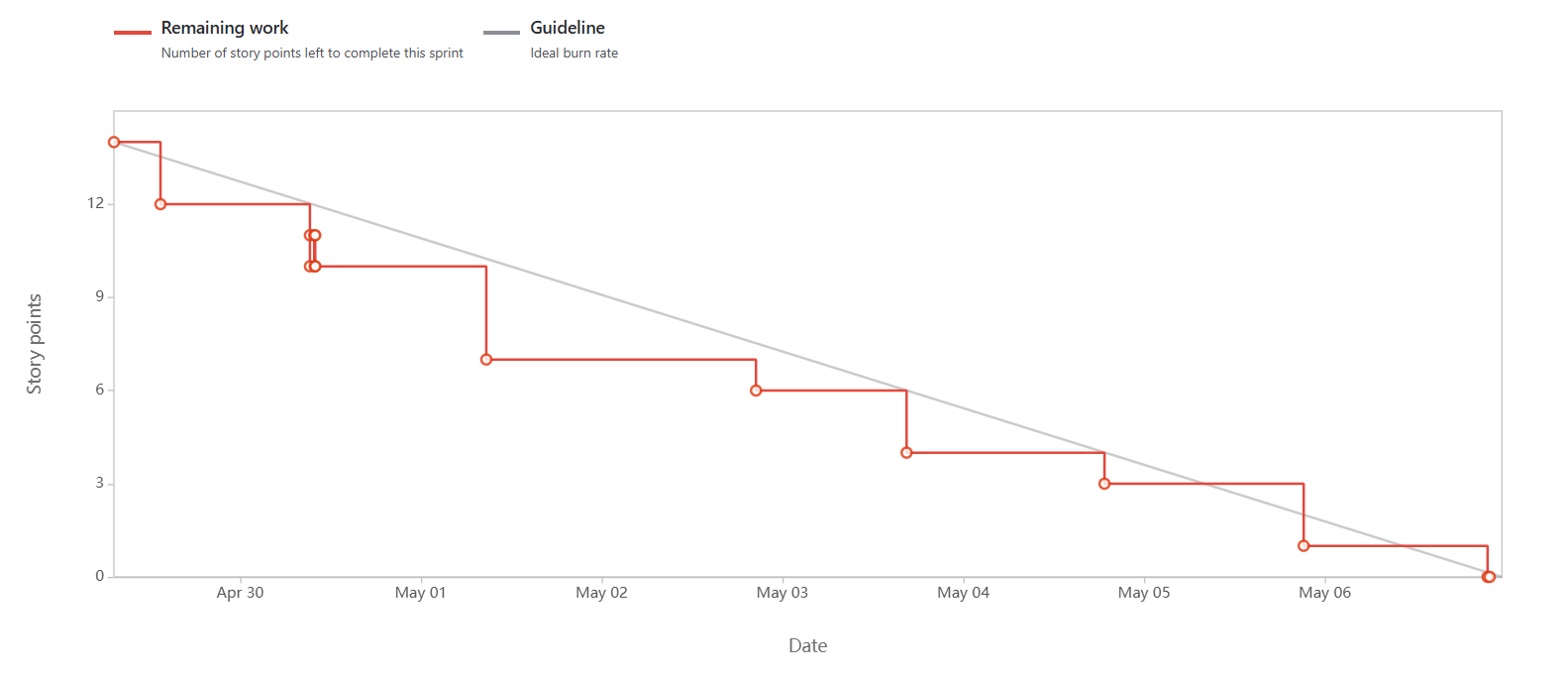
Ngày bắt đầu: 29/04/2025

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.



**Burndown chart của Sprint 1:**

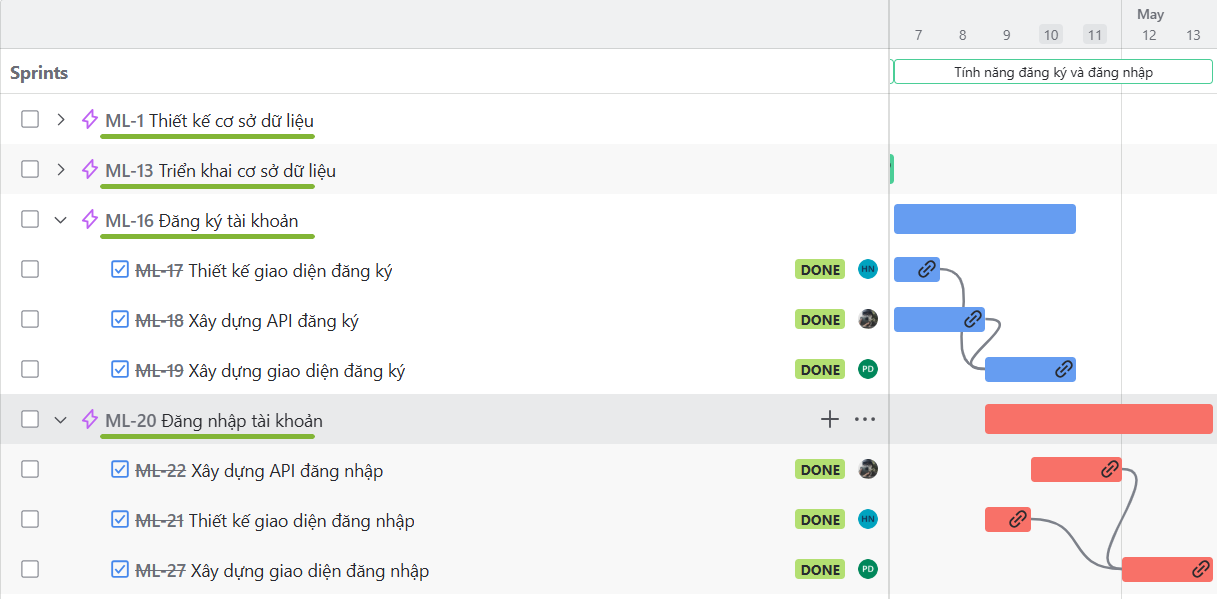


**Sprint 2: Tính năng đăng ký và đăng nhập**

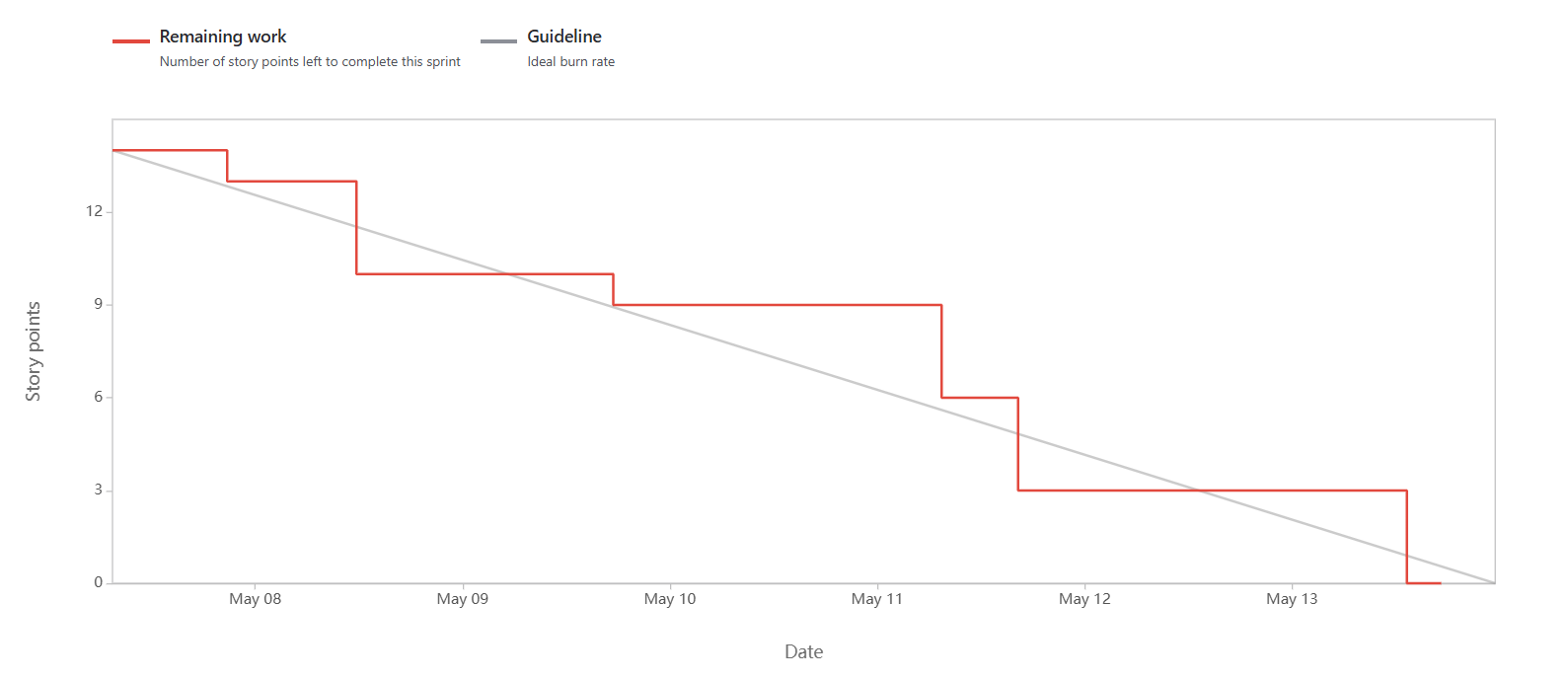
Ngày bắt đầu: 07/05/2025

Ngày kết thúc: 13/05/2025

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



**Burndown chart của Sprint 2:**

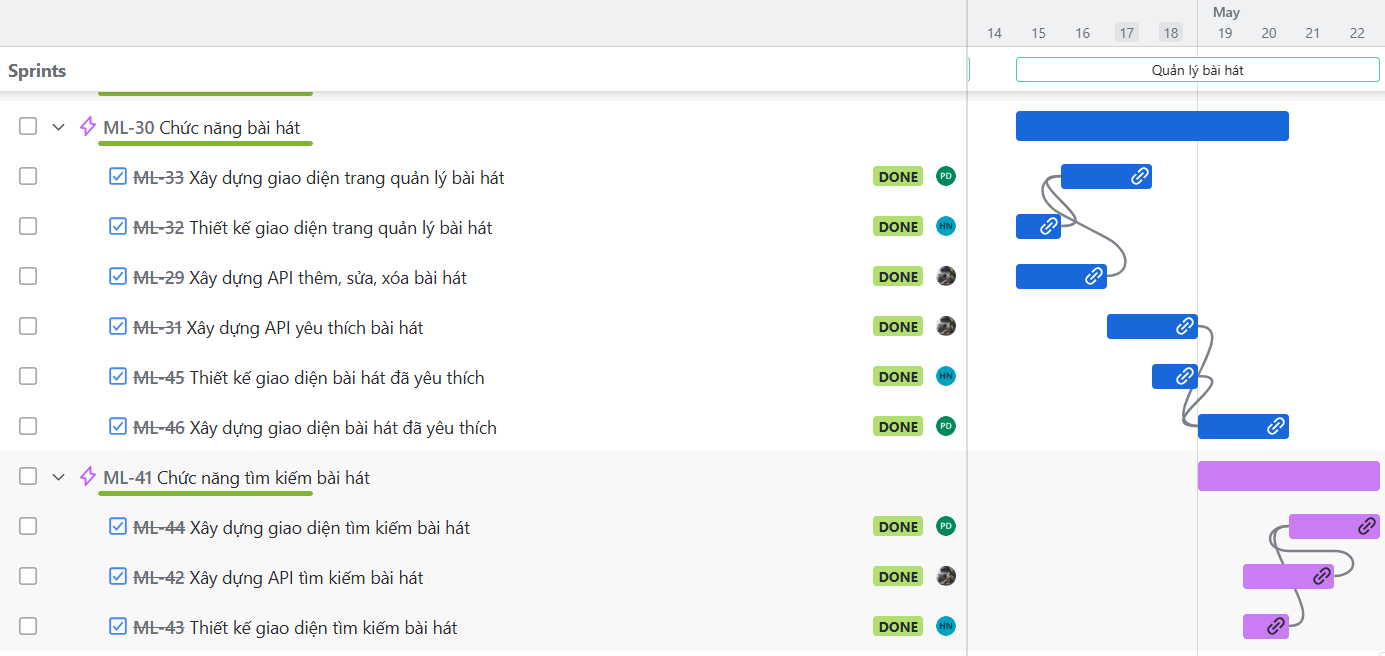


**Sprint 3: Quản lý bài hát**

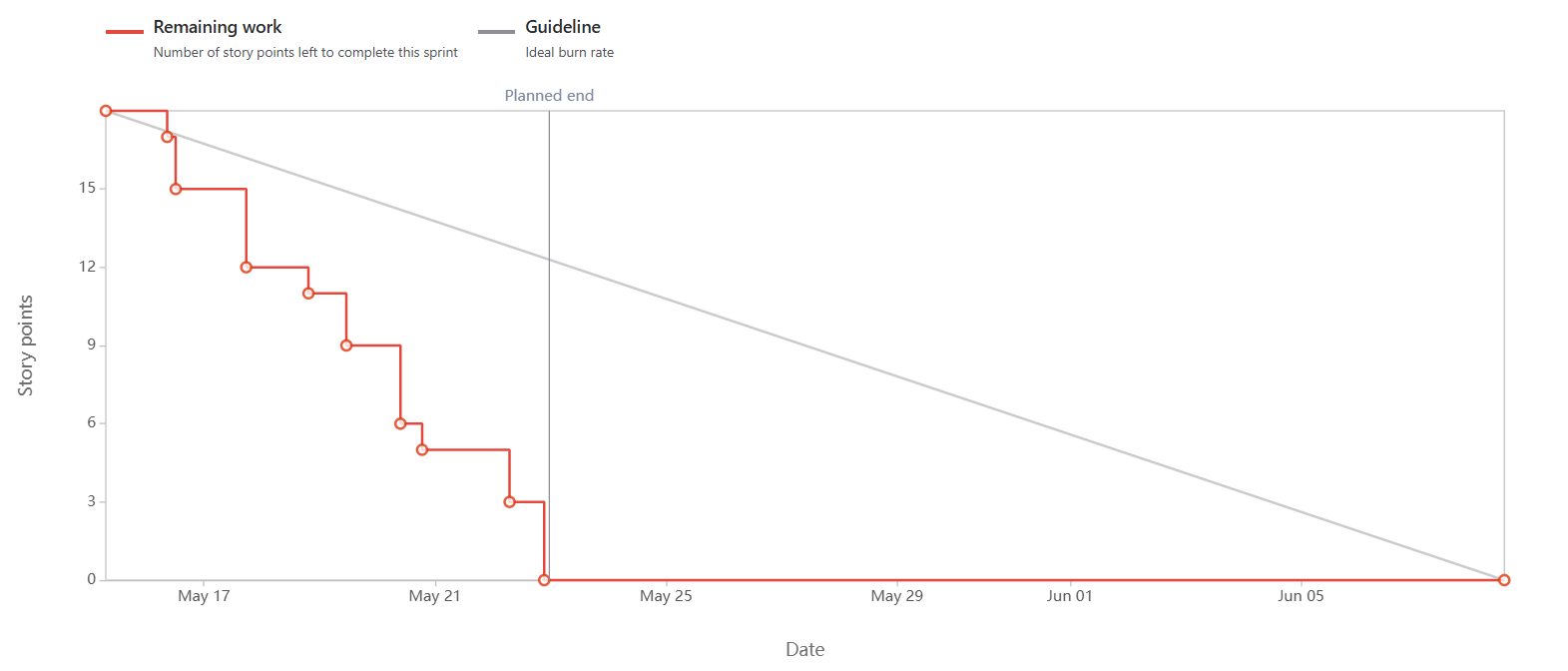
Ngày bắt đầu: 15/05/2025

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.



**Burndown chart của Sprint 3:**

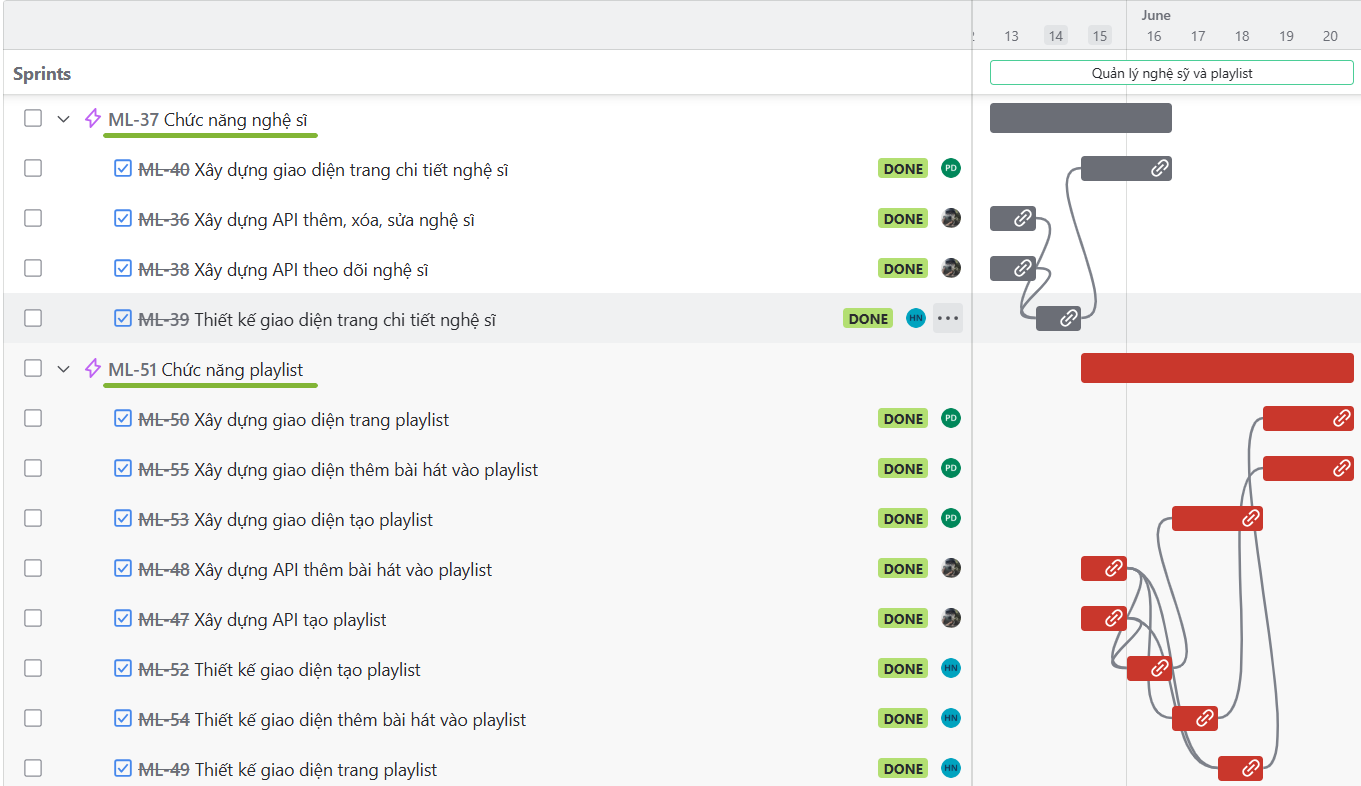


**Sprint 4: Quản lý nghệ sĩ và playlist**

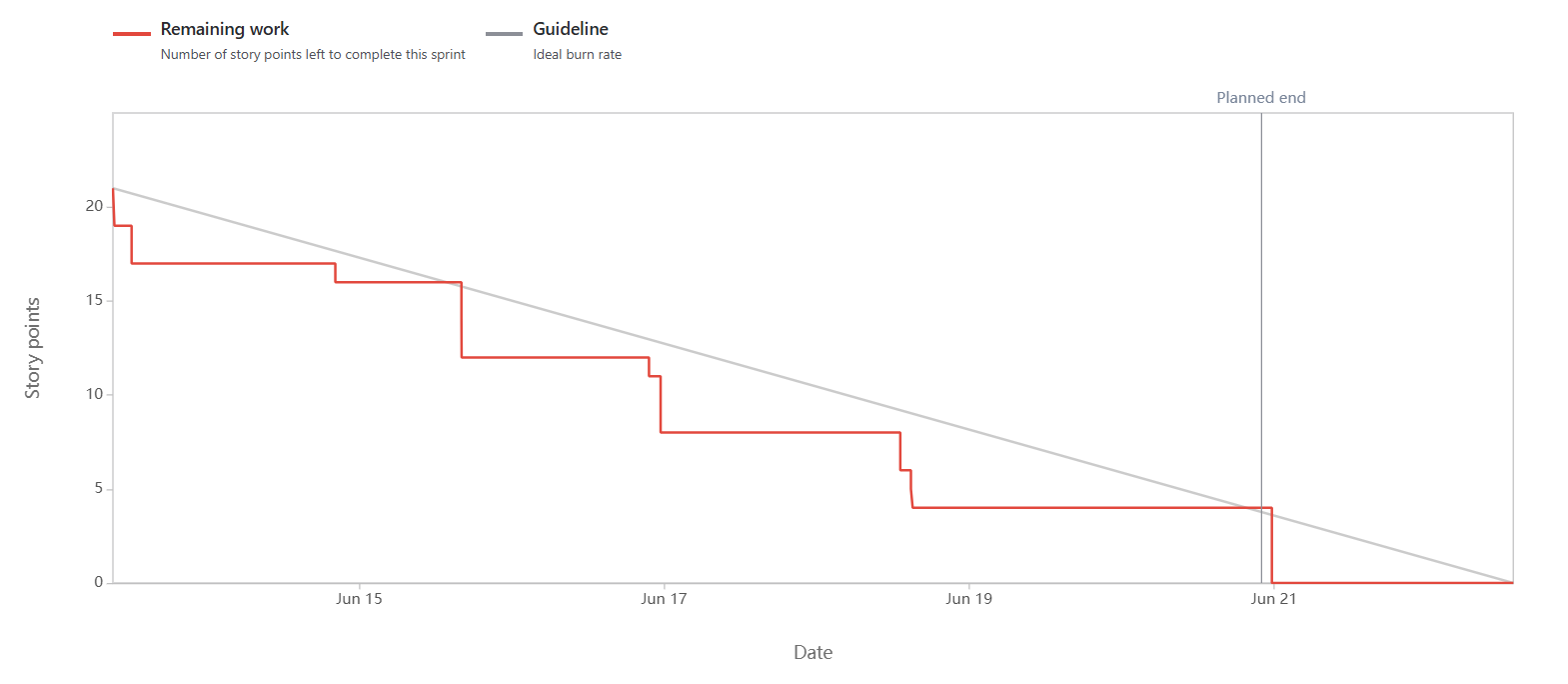
Ngày bắt đầu: 13/06/2025

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.



**Burndown chart của Sprint 4:**

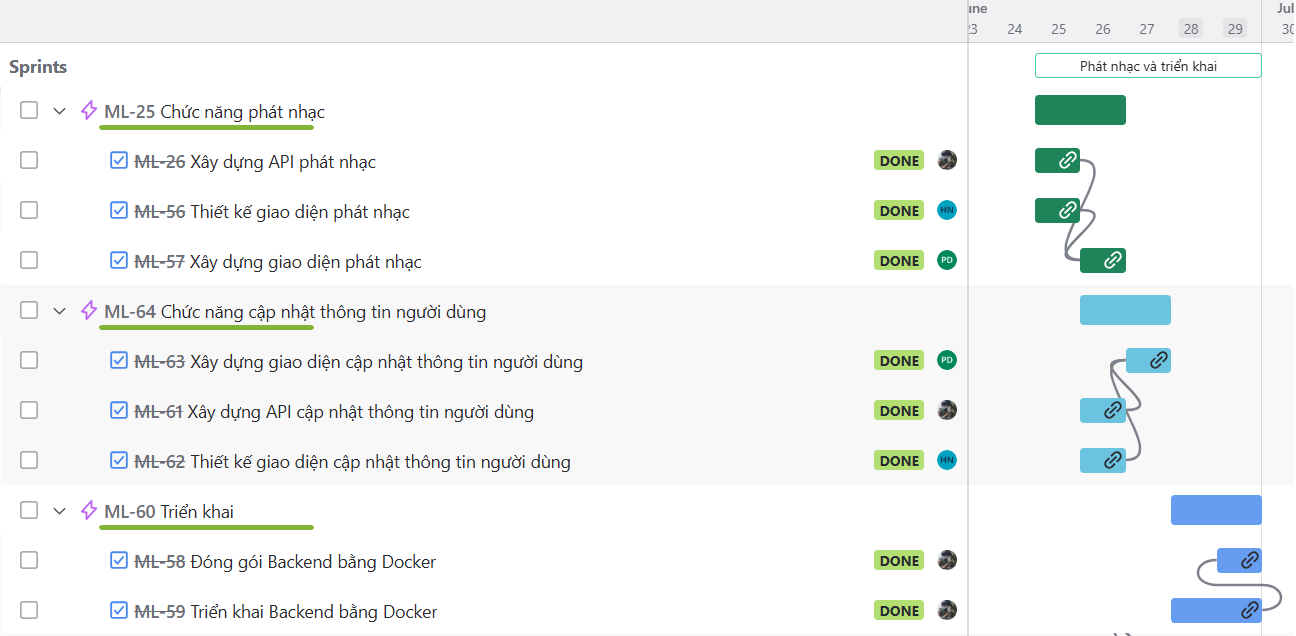


**Sprint 5: Phát nhạc và triển khai**

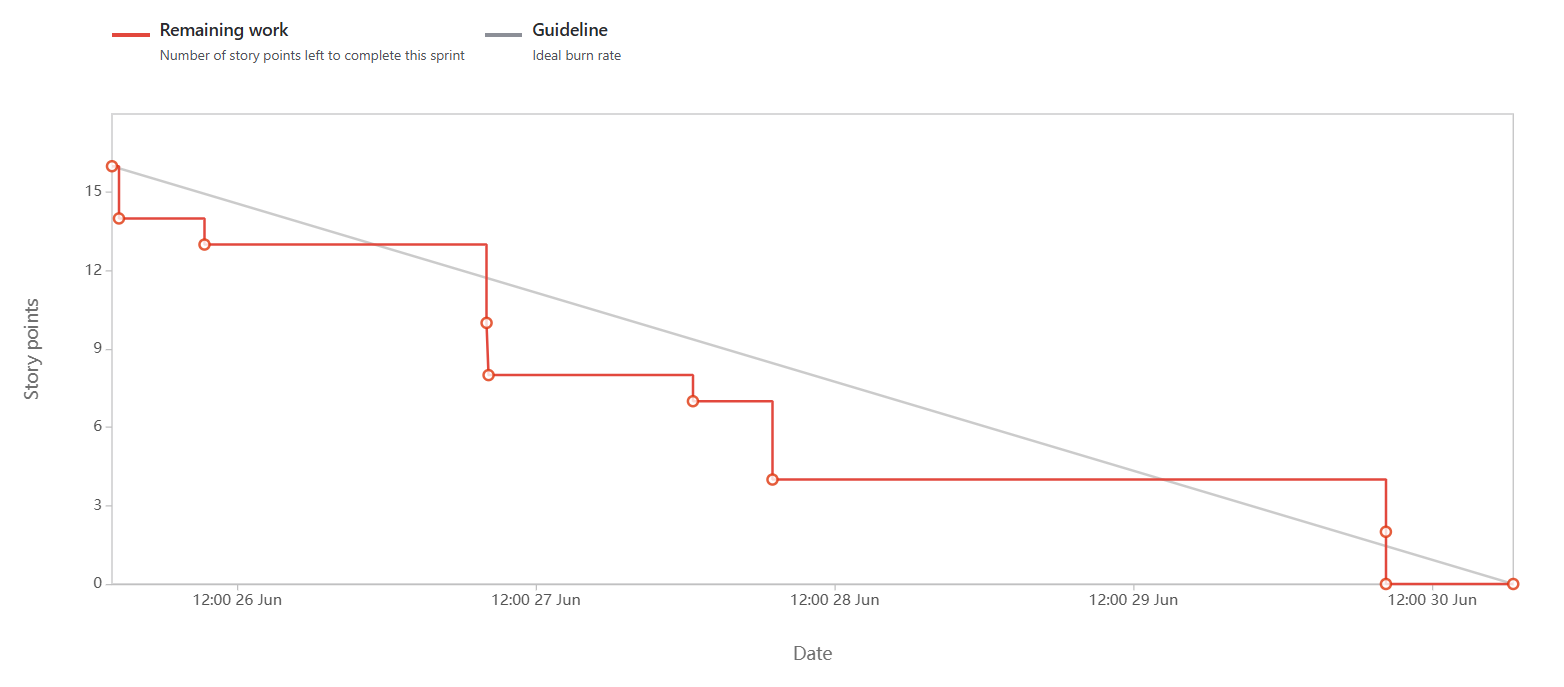
Ngày bắt đầu: 25/06/2025

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.

****

**Burndown chart của Sprint 5:**



# CHƯƠNG 6: KIỂM THỬ (Nguyên)

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

## 6.2 Kết quả kiểm thử API

# CHƯƠNG 7: ĐÁNH GIÁ VÀ KIẾT LUẬN (Phát)

## 7.1 Kết quả đạt được

## 7.2 Hạn chế

## 7.3 Hướng phát triển

**DANH MỤC TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | P. T. P. Nam, Tài liệu giảng dạy Môn Hệ quản trị cơ sở dữ liệu, Trà Vinh: Trường Đại học Trà Vinh, 2015. |
| [2] | Coronel and Carlos, Database systems, Australia: Cengage Learning, 2019. |