

ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



Họ và tên sinh viên : Dương Ngọc Nguyên

BÁO CÁO

THỰC TẬP TỐT NGHIỆP

**Công ty thực tập : Công ty TNHH Giải Pháp Phần
Mềm Tường Minh (TMA Solutions)**

Chuyên gia hướng dẫn : Huỳnh Trọng Tuyển

Giảng viên hướng dẫn : PGS.TS Nguyễn Tuấn Đăng

TP. Hồ Chí Minh, tháng 08 năm 2025

ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



Họ và tên sinh viên : Dương Ngọc Nguyên

BÁO CÁO

THỰC TẬP TỐT NGHIỆP

**Công ty thực tập : Công ty TNHH Giải Pháp Phần
Mềm Tường Minh (TMA Solutions)**

Chuyên gia hướng dẫn : Huỳnh Trọng Tuyển

Giảng viên hướng dẫn : PGS.TS Nguyễn Tuấn Đăng

TP. Hồ Chí Minh, tháng 08 năm 2025

MỤC LỤC

NHẬN XÉT CỦA CHUYÊN GIA DOANH NGHIỆP	I
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	II
LỜI MỞ ĐẦU	1
Chương 1: GIỚI THIỆU	3
1.1. Giới thiệu công ty thực tập	3
1.1.1. Tổng quan về Công ty TMA Solutions	3
1.1.2. Lĩnh vực hoạt động và phòng ban thực tập	3
1.1.3. Chuyên gia hướng dẫn tại doanh nghiệp	3
1.2. Nhiệm vụ thực tập	3
1.2.1. Tên đề tài	3
1.2.2. Mục tiêu đề tài	4
1.2.3. Phạm vi công việc được giao	4
1.3. Kết luận chương 1	5
Chương 2: NỘI DUNG THỰC TẬP	6
2.1. Giai đoạn chuẩn bị và nghiên cứu nền tảng	6
2.1.1. Nghiên cứu chuyên sâu về công nghệ và nền tảng	6
2.1.2. Thiết lập môi trường và cấu trúc dự án	8
2.2. Phân tích và thiết kế hệ thống	10
2.2.1. Phân tích yêu cầu	10
2.2.2. Thiết kế kiến trúc tổng quan	11
2.2.3. Thiết kế cơ sở dữ liệu (MySQL)	13
2.3. Triển khai chi tiết các chức năng	14
2.3.1. Xây dựng RESTful API (Backend)	14
2.3.2. Xây dựng UI (Frontend)	17
2.4. Hoàn thiện ứng dụng	19
2.5. Tổng quan	25
2.6. Kết luận chương 2	26
Chương 3: KẾT QUẢ THỰC TẬP	27
3.1. Tổng kết các kết quả cụ thể đã đạt được	27
3.1.1. Về mặt kiến thức và kỹ năng	27

3.1.2. Về mặt tài liệu	28
3.2. Kết luận chương 3	36
Chương 4: KẾT LUẬN VÀ KIẾN NGHỊ	37
4.1. Kết luận	37
4.2. Kiến nghị	37
4.2.1. Hướng phát triển của đề tài	37
4.2.2. Kiến nghị với doanh nghiệp (Công ty TMA Solutions)	38
4.2.3. Kiến nghị với Khoa và Nhà trường (Khoa CNTT - ĐH Sài Gòn)	38
TÀI LIỆU THAM KHẢO	40

MỤC LỤC HÌNH ẢNH

Hình 1 : Cấu trúc thư mục frontend	9
Hình 2 : Cấu trúc thư mục backend	9
Hình 3 : Cấu trúc cơ sở dữ liệu	14
Hình 4 : Danh sách API	19
Hình 5 : Giao diện trang chủ nền sáng	20
Hình 6 : Pop-up đăng nhập	20
Hình 7 : Giao diện đăng ký tài khoản	21
Hình 8 : Giao diện đặt lại mật khẩu	21
Hình 9 : Trang tìm kiếm bài hát	21
Hình 10 : Trang danh sách bài hát yêu thích	22
Hình 11 : Modal sách bài hát trong playlist	22
Hình 12 : Modal đổi mật khẩu	22
Hình 13 : Trang lịch sử thanh toán	23
Hình 14 : Giao diện quản lý của Admin	23
Hình 15 : Tab quản lý danh sách bài hát	23
Hình 16 : Modal thêm mới bài hát	24
Hình 17 : Model chỉnh sửa bài hát	24
Hình 18 : Tab thống kê của tất cả thanh toán	24
Hình 19 : Các chức năng chính của ứng dụng (key deliverables)	25

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

[illegible]

LỜI MỞ ĐẦU

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, công nghệ thông tin đã và đang khẳng định vai trò không thể thiếu trong mọi lĩnh vực của đời sống. Nhận thức được tầm quan trọng của việc kết hợp giữa lý thuyết được trang bị ở nhà trường và kinh nghiệm làm việc thực tiễn, kỳ thực tập tốt nghiệp là cơ hội quý báu để sinh viên chúng em được tiếp cận với môi trường làm việc chuyên nghiệp, áp dụng kiến thức đã học vào các dự án thực tế, từ đó củng cố chuyên môn và định hướng cho sự nghiệp tương lai.

Trước hết, em xin trân trọng gửi lời cảm ơn sâu sắc đến Ban Giám hiệu Trường Đại học Sài Gòn, Ban Chủ nhiệm Khoa Công nghệ thông tin đã tạo điều kiện và giới thiệu chúng em đến với các doanh nghiệp uy tín để thực hiện kỳ thực tập này.

Em cũng xin chân thành cảm ơn Ban Lãnh đạo Công ty TMA Solutions đã tiếp nhận và tạo cho em một môi trường làm việc năng động, chuyên nghiệp và đầy thử thách trong suốt thời gian qua. Đặc biệt, em xin gửi lời cảm ơn đến anh **Huỳnh Trọng Tuyên**, người đã trực tiếp hướng dẫn, chỉ bảo tận tình và chia sẻ những kinh nghiệm làm việc quý báu, giúp em vượt qua những khó khăn ban đầu và hoàn thành tốt nhiệm vụ được giao.

Em xin trân trọng cảm ơn thầy **Nguyễn Tuấn Đăng**, giảng viên hướng dẫn của em. Sự quan tâm, động viên và những góp ý chuyên môn sâu sắc của thầy đã giúp em có định hướng rõ ràng và hoàn thiện báo cáo một cách tốt nhất.

Trong kỳ thực tập từ ngày 12/05/2025 đến ngày 08/08/2025, em đã được giao đề tài “**Xây dựng ứng dụng Music Manager sử dụng NodeJS(ReactJS, ExpressJS)**”. Đây là một đề tài giúp em không chỉ vận dụng kiến thức về lập trình mà còn được tiếp cận với các công nghệ mới.

Nội dung của bài báo cáo được trình bày qua các chương:

- **Chương 1:** Giới thiệu chung về công ty thực tập và nhiệm vụ được giao.

- **Chương 2:** Trình bày nội dung thực tập(phân tích, thiết kế và triển khai).
- **Chương 3:** Mô tả và đánh giá các kết quả đạt được của ứng dụng.
- **Chương 4:** Tổng kết lại quá trình thực tập, đưa ra kết luận và các kiến nghị.

Mặc dù đã rất cố gắng, nhưng do thời gian và kinh nghiệm thực tế còn hạn chế, bài báo cáo chắc chắn không thể tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu từ quý Thầy Cô và các anh chị tại công ty để bài báo cáo của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 08 tháng 08 năm 2025

Sinh viên thực hiện

Dương Ngọc Nguyên

Chương 1: GIỚI THIỆU

1.1. Giới thiệu công ty thực tập

1.1.1. Tổng quan về Công ty TMA Solutions

- **Tên công ty:** Công ty TNHH Giải Pháp Phần Mềm Tường Minh (TMA Solutions)
- **Địa chỉ:** 10 Đặng Văn Ngữ, Phường Phú Nhuận, TP. Hồ Chí Minh.
- **Website:** www.tmasolutions.vn
- **Giới thiệu:** Được thành lập từ năm 1997, TMA Solutions là một trong những công ty công nghệ hàng đầu tại Việt Nam, chuyên cung cấp dịch vụ phát triển phần mềm và giải pháp công nghệ thông tin cho nhiều khách hàng trong và ngoài nước như Mỹ, Canada, Úc, Châu Âu, Nhật Bản và Singapore. Với mục tiêu mang đến các sản phẩm và dịch vụ chất lượng cao, TMA Solutions luôn không ngừng nghiên cứu, ứng dụng các công nghệ tiên tiến nhằm giúp đối tác nâng cao năng lực cạnh tranh trên thị trường toàn cầu.

1.1.2. Lĩnh vực hoạt động và phòng ban thực tập

TMA Solutions hoạt động chính trong các lĩnh vực:

- Phát triển ứng dụng di động (iOS, Android).
- Phát triển ứng dụng web và hệ thống quản lý (CMS).
- Tư vấn và cung cấp các giải pháp công nghệ thông tin cho doanh nghiệp.

Trong kỳ thực tập này, em được phân công vào vị trí **Intern NodeJS Developer** tại phòng DC22 - Program1. Đây là môi trường làm việc năng động, chuyên nghiệp, tạo điều kiện để em được tiếp xúc với các dự án thực tế, học hỏi quy trình làm việc theo mô hình Agile/Scrum và nâng cao kỹ năng chuyên môn.

1.1.3. Chuyên gia hướng dẫn tại doanh nghiệp

Trong suốt quá trình thực tập, em đã nhận được sự hướng dẫn tận tình từ anh **Huỳnh Trọng Tuyển**, hiện đang là **Team Leader** tại công ty.

1.2. Nhiệm vụ thực tập

1.2.1. Tên đề tài

Dưới sự hướng dẫn của chuyên gia tại công ty, em đã được giao nhiệm vụ thực hiện đề tài: **"Xây dựng ứng dụng Music Manager sử dụng NodeJS(ReactJS, ExpressJS)"**.

1.2.2. Mục tiêu đề tài

Đề tài được thực hiện nhằm đạt được các mục tiêu chính sau:

- **Về kiến thức:**
 - Nghiên cứu và làm chủ công nghệ ReactJS kết hợp với Material UI để xây dựng giao diện người dùng hiện đại và thân thiện.
 - Tìm hiểu và áp dụng mô hình quản lý trạng thái hiệu quả bằng Redux Toolkit nhằm tối ưu hiệu suất và khả năng mở rộng của ứng dụng.
 - Phát triển backend với NodeJS và ExpressJS, xây dựng hệ thống API RESTful để quản lý bài hát, playlist, wishlist, người dùng và phân quyền.
 - Tích hợp Redis (Upstash) để lưu trữ cache và xử lý các tác vụ bất đồng bộ, giúp nâng cao hiệu năng hệ thống.
 - Học cách làm việc với các API bên ngoài (MoMo API).
- **Về kỹ năng:**
 - Nâng cao kỹ năng lập trình JavaScript và phát triển ứng dụng với NodeJS.
 - Rèn luyện kỹ năng phân tích yêu cầu, thiết kế hệ thống và giải quyết vấn đề.
 - Làm quen với quy trình phát triển phần mềm chuyên nghiệp.
 - Tìm hiểu và tích hợp cổng thanh toán MoMo (sandbox) để hỗ trợ người dùng thực hiện các giao dịch thanh toán trực tuyến một cách an toàn và thuận tiện.
- **Về sản phẩm:**
 - Xây dựng một ứng dụng web quản lý và phát nhạc trực tuyến hoàn chỉnh, có thể triển khai và hoạt động ổn định trên môi trường thực tế với đầy đủ chức năng như xác thực người dùng, quản lý playlist, tích hợp thanh toán MoMo và tối ưu hiệu suất với Redis.

1.2.3. Phạm vi công việc được giao

Để hoàn thành đề tài, em đã thực hiện các công việc chính sau:

- Tìm hiểu, cài đặt và cấu hình môi trường phát triển NodeJS.

- Thiết kế giao diện người dùng (UI/UX) bằng ReactJS kết hợp với Material UI, đảm bảo trải nghiệm trực quan và thân thiện.
- Xây dựng chức năng xác thực người dùng (đăng ký, đăng nhập, đăng xuất) bằng ExpressJS, sử dụng JWT để bảo mật phiên đăng nhập.
- Thiết kế và triển khai cơ sở dữ liệu với MYSQL, lưu trữ thông tin người dùng, bài hát, playlist và wishlist.
- Tích hợp API MoMo (sandbox) để triển khai chức năng thanh toán khi người dùng mua gói dịch vụ.
- Áp dụng mô hình quản lý trạng thái với Redux Toolkit, giúp kiểm soát luồng dữ liệu xuyên suốt toàn bộ ứng dụng.
- Phát triển các tính năng nâng cao như tạo playlist cá nhân, thêm bài hát vào wishlist, tìm kiếm thông minh và đề xuất nhạc.
- Tối ưu hiệu suất với Redis (Upstash) để cache dữ liệu và cải thiện tốc độ hệ thống.

1.3. Kết luận chương 1

Chương 1 đã trình bày tổng quan về công ty thực tập TMA Solutions, đồng thời xác định rõ nhiệm vụ, mục tiêu và phạm vi của đề tài "Xây dựng hệ thống quản lý và phát nhạc trực tuyến sử dụng NodeJS(ReactJS, ExpressJS)". Các nội dung này là cơ sở quan trọng để triển khai các phần tiếp theo của báo cáo. Chương 2 sẽ đi sâu vào việc trình bày các cơ sở lý thuyết liên quan và phân tích chi tiết hệ thống.

Chương 2: NỘI DUNG THỰC TẬP

2.1. Giai đoạn chuẩn bị và nghiên cứu nền tảng

Trước khi bắt đầu phát triển các chức năng chính của dự án Music Manager, ba tuần đầu tiên của kỳ thực tập được dành riêng cho việc tìm hiểu công nghệ nền tảng một cách có hệ thống và thiết lập môi trường phát triển chuyên nghiệp. Đây là giai đoạn quan trọng nhằm đảm bảo hiểu rõ kiến trúc hệ thống, công nghệ sử dụng như ReactJS, NodeJS, ExpressJS, MySQL, Redis và MoMo API, từ đó tạo nền tảng vững chắc cho các giai đoạn phát triển tiếp theo.

2.1.1. Nghiên cứu chuyên sâu về công nghệ và nền tảng

Quá trình nghiên cứu không chỉ dừng lại ở việc đọc tài liệu lý thuyết mà tập trung vào việc hiểu sâu bản chất công nghệ và thực hành trực tiếp trên thực tế để chuyển hóa kiến thức thành kỹ năng lập trình vững chắc.

- **Ngôn ngữ lập trình JavaScript/TypeScript:**
 - **Nội dung nghiên cứu:** Tìm hiểu toàn diện về cú pháp và các khái niệm cốt lõi của JavaScript, bao gồm: kiểu dữ liệu, khai báo biến (let, const), toán tử, cấu trúc điều khiển (if, switch, for, while), và hàm. Bên cạnh đó, đặc biệt chú trọng đến các khái niệm nâng cao như:
 - **Functions:** Cách định nghĩa, tham số (bắt buộc, tùy chọn, theo tên), và hàm mũi tên (arrow function).
 - **Object-Oriented Programming (OOP):** Cách tạo và sử dụng class, constructor, inheritance, abstract class, và interface.
 - **Collections:** Cách làm việc hiệu quả với Array, Set, và Map, bao gồm các phương thức bậc cao như map, filter, reduce.
 - **Null Safety:** Hiểu rõ và áp dụng các khái niệm về an toàn null (null safety) để viết mã nguồn an toàn và giảm thiểu lỗi runtime.
 - **Asynchronous Programming:** Nắm vững cách xử lý các tác vụ bất đồng bộ, một phần không thể thiếu trong lập trình di động, thông qua Promise, async/await.

- **Phương pháp thực hành:** Với mỗi khái niệm đã học, luyện tập viết các đoạn mã ví dụ và chạy thử bằng NodeJS hoặc trực tiếp trên trình duyệt để rèn luyện cú pháp và tư duy lập trình. Ví dụ: khai báo biến, viết hàm kiểm tra số chẵn/lẻ, sử dụng map để xử lý mảng, viết điều kiện if lồng nhau, và thực hành xử lý bất đồng bộ với Promise, async/await.
- **Framework ExpressJS và ReactJS cơ bản:**
 - **Kiến trúc ReactJS component:** Nắm vững mô hình phát triển theo hướng component của ReactJS với triết lý "Everything is a component", đồng thời hiểu rõ sự khác biệt giữa component có trạng thái (stateful) và không có trạng thái (stateless). Nghiên cứu vòng đời của component thông qua các hook như useEffect,...
 - **Bố cục giao diện (Layout Components):** Tạo ra một giao diện demo để thực hành xây dựng bố cục trang bằng cách kết hợp các thành phần như Grid, Box, Stack trong Material UI để sắp xếp các phần tử theo chiều ngang, dọc và tự động điều chỉnh kích thước theo độ rộng màn hình. Em cũng áp dụng thuộc tính flex, gap, justifyContent, alignItems để điều chỉnh tỷ lệ không gian linh hoạt giữa các thành phần con.
 - **ExpressJS và RESTful API:** Tìm hiểu và thực hành xây dựng các API RESTful sử dụng ExpressJS, tổ chức mã nguồn rõ ràng thành các phần: routes, controllers, middlewares:
 - Xử lý các HTTP request (GET, POST, PUT, DELETE).
 - Quản lý lỗi và trả về mã trạng thái phù hợp (ví dụ: 200, 400, 401, 500).
 - Tạo middleware xác thực người dùng bằng JWT để bảo vệ các route yêu cầu đăng nhập và phân quyền, sử dụng express-validator để kiểm tra dữ liệu đầu vào của request.
 - **Xử lý Form và Validation:**
 - Thực hành xây dựng các biểu mẫu (form) như đăng ký và đăng nhập bằng cách sử dụng thư viện Formik để quản lý trạng thái biểu mẫu, kết hợp với Yup để xác thực dữ liệu đầu vào, định nghĩa các hàm kiểm tra điều kiện hợp lệ của các trường như: định dạng email, độ dài tối thiểu của mật khẩu, xác nhận mật khẩu trùng khớp. Toàn bộ quá trình xác thực đều được hiển

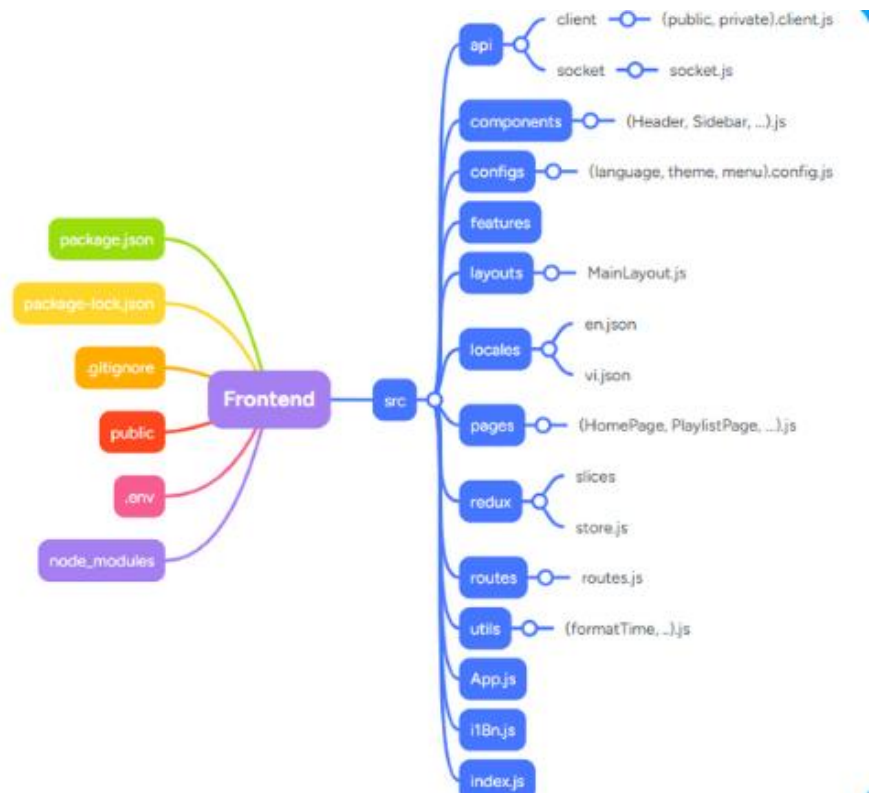
thị rõ ràng cho người dùng thông qua các thông báo lỗi dưới từng trường nhập liệu.

- Bên cạnh đó, cũng làm việc với tài nguyên giao diện như ảnh và biểu tượng: hiển thị ảnh từ thư mục public, ảnh từ URL bên ngoài, và sử dụng các biểu tượng từ thư viện Material UI Icons hoặc Lucide React để làm phong phú thêm giao diện người dùng.

2.1.2. Thiết lập môi trường và cấu trúc dự án

Việc thiết lập một nền tảng vững chắc và môi trường làm việc đồng nhất ngay từ đầu có vai trò cực kỳ quan trọng trong suốt quá trình phát triển phần mềm.

- **Quản lý phiên bản NodeJS với NVM (Node Version Manager):**
 - **Lý do:** Trong các dự án thực tế, việc sử dụng phiên bản NodeJS không đồng nhất giữa các thành viên hoặc khác biệt giữa môi trường phát triển và sản xuất thường là nguyên nhân gây ra nhiều lỗi khó xác định. Vì vậy, chủ động sử dụng NVM để quản lý phiên bản NodeJS sẽ giúp tránh các lỗi môi trường.
 - **Triển khai:** Cài đặt NVM trên máy cục bộ và quy định sử dụng phiên bản NodeJS cụ thể (ví dụ: 20.04).
- **Cấu trúc thư mục Frontend:**
 - Sau khi khởi tạo dự án React tái tổ chức lại thư mục theo mô hình "feature-based" (theo tính năng).



Hình 1: Cấu trúc thư mục frontend

- **Cấu trúc thư mục Backend:**

- Tương tự frontend, backend được chia thành các tầng rõ ràng:



Hình 2: Cấu trúc thư mục backend

2.2. Phân tích và thiết kế hệ thống

Giai đoạn này là xương sống của toàn bộ dự án, nơi các yêu cầu được chuyển hóa thành một bản thiết kế kỹ thuật chi tiết, làm kim chỉ nam cho quá trình triển khai. Quá trình này bao gồm việc phân tích yêu cầu, lựa chọn kiến trúc, thiết kế các thành phần cốt lõi như quản lý trạng thái, cơ sở dữ liệu và điều hướng.

2.2.1. Phân tích yêu cầu

Dựa trên nhiệm vụ được giao, em đã tiến hành phân tích và phân loại các yêu cầu của hệ thống thành hai nhóm chính:

- **Yêu cầu chức năng (Functional Requirements):** Mô tả những gì hệ thống phải làm.
 - **Xác thực (Authentication):** Hệ thống phải cho phép người dùng đăng ký bằng email, đăng nhập bằng email, đăng xuất, khôi phục mật khẩu đã quên và thay đổi mật khẩu khi đã đăng nhập.
 - **Người dùng (User):** Người dùng xem các thông tin cá nhân, chỉnh sửa và thay đổi mật khẩu.
 - **Quản lý bài hát (Song):** Hiển thị toàn bộ các bài hát, hỗ trợ phân trang đảm bảo mượt mà, tìm kiếm bài hát theo tên, nghệ sĩ, thêm bài hát mới, chỉnh sửa bài hát, xóa bài hát khỏi danh sách.
 - **Quản lý danh sách phát (Playlist):** Người dùng đã đăng nhập phải có thể tạo mới nhiều playlist, xóa playlist, thêm bài hát vào playlist, xóa bài hát khỏi playlist và xem danh sách nhạc của playlist.
 - **Quản lý danh sách yêu thích (Wishlist):** Người dùng đã đăng nhập có thể thêm bài hát, xóa bài hát khỏi wishlist và xem danh sách nhạc trong wishlist.
 - **Trình phát nhạc (Player):** Phát nhạc từ bài hát bất kì, bài hát trong playlist, wishlist, hỗ trợ tạm dừng – phát tiếp (pause/play), phát bài hát tiếp theo hoặc trước đó (play next/previous), điều chỉnh âm lượng.
 - **Chức năng khác:** Người dùng có thể chuyển đổi màu nền (light/dark mode), chuyển đổi ngôn ngữ sang (tiếng anh/việt).

- **Yêu cầu Phi chức năng (Non-functional Requirements):** Mô tả các tiêu chuẩn về chất lượng của hệ thống.
 - **Hiệu năng (Performance):** Ứng dụng phải phản hồi nhanh, đảm bảo không có hiện tượng giật, lag trong quá trình sử dụng. Đặc biệt, màn hình trò chuyện cần tải lịch sử tin nhắn mượt mà, kể cả khi dữ liệu lớn — được xử lý bằng cơ chế tải dữ liệu theo từng phần (lazy loading hoặc phân trang).
 - **Tính khả dụng (Usability):** Giao diện người dùng cần được thiết kế trực quan, dễ hiểu và thân thiện, giúp người dùng dễ dàng thao tác mà không cần tài liệu hướng dẫn. Các thành phần giao diện phải tuân theo các nguyên tắc UX/UI cơ bản.
 - **Tính bảo trì (Maintainability):** Mã nguồn phải được tổ chức hợp lý, dễ đọc và dễ hiểu. Việc chỉnh sửa, cập nhật hoặc bổ sung tính năng mới phải được thực hiện thuận tiện và nhanh chóng. Đạt được điều này thông qua việc áp dụng kiến trúc quản lý trạng thái rõ ràng (như Redux/Zustand với ReactJS) và cấu trúc thư mục phân tách theo tính năng (feature-based structure).
 - **Tính nhất quán (Consistency):** Trải nghiệm người dùng cần được đảm bảo đồng bộ trên toàn bộ hệ thống: từ cách bố trí giao diện, màu sắc, font chữ đến hành vi của các nút và thành phần giao tiếp người dùng. Giao diện phải nhất quán giữa các màn hình và thiết bị khác nhau.

2.2.2. Thiết kế kiến trúc tổng quan

Để đáp ứng các yêu cầu về hiệu năng, tính bảo trì và khả năng mở rộng của hệ thống, em đã lựa chọn kiến trúc Monolithic kết hợp mô hình phân lớp (Layered Architecture) cho cả frontend và backend, được kết nối thông qua giao tiếp API nội bộ gồm:

- **Frontend (chịu trách nhiệm hiển thị giao diện người dùng và xử lý các tương tác trên trình duyệt)**
 - **features/:** Chứa các module chức năng độc lập như auth, songs, playlist, payment, mỗi module có sub-folder riêng cho components, api.
 - **pages/:** Chứa các trang giao diện chính (HomePage, PlaylistPage,...).
 - **store/:** Sử dụng Redux Toolkit để quản lý trạng thái tập trung.

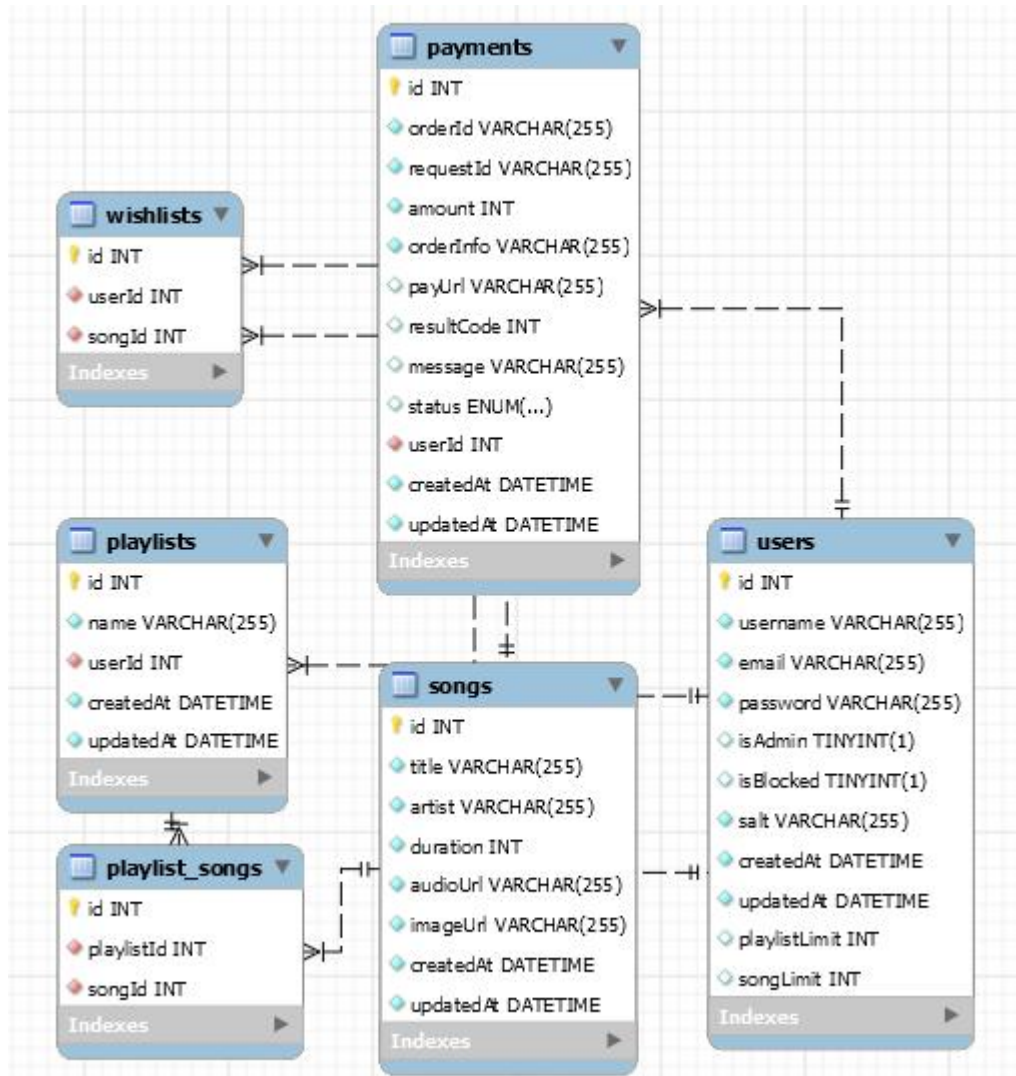
- **api/**: Tổ chức các hàm gọi API REST đến backend, dùng Axios instance để cấu hình chung.
- **router/**: Quản lý toàn bộ hệ thống định tuyến bằng react-router-dom.
- **configs/**: Chứa các file cấu hình cơ bản (themeMode, language,...),
- **layouts/**: Chứa file bố cục giao diện chính (MainLayout).
- **utils/**: Chứa các hàm tiện ích tái sử dụng.
- **Backend (chịu trách nhiệm xử lý logic nghiệp vụ, xác thực người dùng, lưu trữ dữ liệu)**
 - **Lớp định tuyến (Routing layer)**: Là nơi tiếp nhận yêu cầu từ frontend thông qua các endpoint RESTful. Mỗi route tương ứng với một module cụ thể như auth.routes.js, song.routes.js, payment.routes.js. Lớp này không xử lý logic nghiệp vụ mà chỉ định tuyến và gọi controller tương ứng.
 - **Lớp xử lý nghiệp vụ (Controller layer)**: Bao gồm các file controller nơi triển khai các logic cốt lõi như xác thực người dùng, tạo playlist, xử lý thanh toán MoMo... Controller tiếp nhận request từ router, gọi các hàm để xử lý logic, và trả về kết quả cho frontend.
 - **Lớp tương tác với dữ liệu (Model layer)**: Giao tiếp trực tiếp với cơ sở dữ liệu (MySQL thông qua Sequelize ORM). Lớp này che giấu toàn bộ logic thao tác dữ liệu, giúp phần còn lại của hệ thống không cần quan tâm tới chi tiết truy vấn SQL hay cấu trúc bảng.

2.2.3. Thiết kế cơ sở dữ liệu (MySQL)

Để lưu trữ dữ liệu một cách có tổ chức, đảm bảo tính toàn vẹn và dễ dàng truy xuất, hệ thống sử dụng cơ sở dữ liệu quan hệ MySQL. Kiến trúc quan hệ này giúp xác định rõ các thực thể (entity) trong hệ thống và mối quan hệ giữa chúng thông qua các bảng (table) được thiết kế chặt chẽ gồm các bảng sau:

- **Table users:** Lưu trữ thông tin định danh người dùng bao gồm email, mật khẩu đã mã hóa (hash), thông tin hồ sơ cá nhân, loại tài khoản (email), và thời gian tạo tài khoản. Mỗi người dùng có thể tạo nhiều playlist và có nhiều lượt thanh toán.
- **Table songs:** Chứa thông tin các bài hát được hệ thống quản lý như tên bài hát, nghệ sĩ, đường dẫn ảnh, đường dẫn file nhạc, thời lượng... Đây là nguồn dữ liệu chính được liên kết tới playlist và wishlist.
- **Table playlists:** Đại diện cho danh sách phát mà người dùng tự tạo. Mỗi playlist được gắn với một user_id, có tên, mô tả và thời gian tạo.
- **Table wishlists:** Cho phép người dùng đánh dấu các bài hát yêu thích. Một người dùng có thể có nhiều bài hát yêu thích, và một bài hát có thể được nhiều người thêm vào danh sách yêu thích.
- **Table playlist_songs:** Bảng trung gian thể hiện mối quan hệ nhiều-nhiều giữa bảng playlist và bảng songs. Ngoài ra còn có thể lưu thông tin thứ tự sắp xếp các bài hát trong playlist.

- **Table payments:** Ghi lại các giao dịch thanh toán, sử dụng sandbox API của MoMo. Thông tin bao gồm: số tiền thanh toán, trạng thái giao dịch, mã giao dịch, hình thức thanh toán và thời gian thực hiện. Mỗi giao dịch liên kết với một người dùng cụ thể.



Hình 3: Cấu trúc cơ sở dữ liệu

2.3. Triển khai chi tiết các chức năng

Phần này sẽ đi sâu vào việc triển khai từng chức năng, phân tích cách sử dụng và quan trọng hơn là bóc tách chi tiết luồng logic hoạt động bên trong, từ tương tác của người dùng trên giao diện đến cách xử lý và cập nhật trạng thái.

2.3.1. Xây dựng RESTful API (Backend)

Đây là phần xử lý cốt lõi của hệ thống, chịu trách nhiệm tiếp nhận yêu cầu từ phía frontend, xử lý logic nghiệp vụ, truy vấn cơ sở dữ liệu và trả về kết quả. Các chức

năng backend được triển khai theo kiến trúc phân lớp rõ ràng (route, controller, service, model), đảm bảo dễ mở rộng và bảo trì.

- **Authentication**

- **API:**

1. POST /auth/signup: Đăng ký, tạo user mới, lưu OTP vào Redis, gửi OTP qua email.
2. POST /auth/verify-otp: Kiểm tra OTP hợp lệ từ Redis và kích hoạt tài khoản.
3. POST /auth/resent-otp: Tạo mới OTP và gửi lại email.
4. POST /auth/signin: Xác thực bằng email/password, trả access token và refresh token.
5. POST /auth/reset-password: Đặt lại mật khẩu nếu OTP hợp lệ.
6. POST /auth/renew-accesstoken: Tạo mới access token với JWT.

- **User**

- **API:**

1. GET /users/info: Lấy thông tin user dựa trên access token.
2. PUT /users/change-password: Kiểm tra mật khẩu cũ, cập nhật mật khẩu mới.

- **User**

- **API:**

1. GET /songs/all-songs: Trả danh sách bài hát.
2. GET /songs/search?keyword="": Tìm kiếm bài hát theo từ khoá.
3. GET /songs/trending-songs: Trả danh sách bài hát trending.

- **Playlist**

- **API:**

1. GET /playlists/all-playlists: Lấy tất cả playlist của người dùng.
2. GET /playlists/all-songs/:playlistId: Lấy bài hát trong một playlist.
3. POST /playlists/create: Tạo playlist mới.
4. POST /playlists/add-song/:playlistId/:songId: Thêm bài vào playlist.

5. POST /playlists/delete-multiple-song: Xoá nhiều bài khỏi playlist.
 6. DELETE /playlists/delete-song/:playlistId/:songId: Xoá một bài hát.
 7. DELETE /playlists/delete/:playlistId: Xoá playlist.
- **Wishlist**
 - **API:**
 1. GET /wishlists/all-songs: Lấy danh sách bài hát yêu thích.
 2. POST /wishlists/add-songs/:songId: Danh sách nhạc yêu thích.
 3. POST /wishlists/delete-multiple-song: Xoá nhiều bài (truyền danh sách songId).
 4. DELETE /wishlists/delete-song/:songId: Xoá một bài hát.
 - **Admin**
 - **API:**
 1. GET /admins/user-stats: Thống kê người dùng.
 2. GET /admins/song-stats: Thống kê bài hát, trending.
 3. GET /admins/playlist-stats: Thống kê playlist.
 4. GET /admins/artist-stats: Cập nhật nghệ sĩ.
 5. GET /admins/payment-stats: Cập nhật thanh toán.
 6. PUT /admins/block-user/:userId: Khoá người dùng.
 7. PUT /admins/unblock-user/:userId: Mở khoá người dùng.
 8. PUT /admins/update-song/:songId: Cập nhật bài hát.
 9. POST /admins/create-song: Thêm bài hát mới.
 10. DELETE /admins/delete-song/:songId: Xoá bài hát.
 - **Payment**
 - **API:**
 1. GET /payments/all-payments: Lấy danh sách thanh toán.
 2. POST /payments/create-payment: Tạo đơn hàng, trả URL redirect đến MoMo.
 3. POST /payments/ipn: Nhận thông báo từ MoMo server (IPN), xác thực và cập nhật DB.

2.3.2 Xây dựng UI (Fontend)

Đây là phần giao diện người dùng của hệ thống, chịu trách nhiệm hiển thị dữ liệu, tiếp nhận thao tác từ người dùng và gửi yêu cầu tới backend thông qua các API.

- Authentication

- o Các modal:

1. Đăng ký: Form nhập email, mật khẩu, xác nhận mật khẩu.
2. Xác minh OTP: Form nhập mã OTP đã gửi qua email.
3. Đăng nhập: Cho phép đăng nhập bằng email/password hoặc Google.
4. Quên mật khẩu / Đặt lại mật khẩu: Nhập email nhận OTP và đặt lại mật khẩu.
5. Đổi mật khẩu: Khi đã đăng nhập.

- o Tích hợp:

1. Validation bằng Formik + Yup.
2. Hiển thị toast thông báo trạng thái (thành công, lỗi).
3. Gọi các API auth để thực hiện các bước tương ứng.
4. Lưu token vào sessionStorage, set header mặc định cho Axios.
5. Tự động refresh token khi hết hạn (interceptor + retry request).
6. Kiểm tra authState để chuyển hướng người dùng.

- User

- o Các component:

1. Trang thông tin người dùng: hiển thị tên, email,...
2. Form đổi mật khẩu: yêu cầu nhập mật khẩu cũ và mới.

- o Tích hợp:

1. Toast feedback.
2. Spinner loading trong lúc xử lý.
3. Giao diện đồng bộ với theme sáng/tối.

- Song

- o Các component:

1. Trang hiển thị các bài hát:
2. Dùng pagination (page, limit).
3. Hiển thị thông tin: ảnh bài hát, tiêu đề, nghệ sĩ.

- 4.Thanh tìm kiếm bài hát realtime.
- 5.GridLayout để hiển thị các bài trending.
- 6.Các nút chức năng phát nhạc.
- o Tích hợp:
 - 1.Skeleton loading.
 - 2.Hiệu ứng animation khi dữ liệu load xong (Framer Motion).
- Wishlist
 - o Các component:
 - 1.Trang Wishlist hiển thị dạng bảng hoặc danh sách bài hát.
 - 2.Nút "Xoá" một bài hát hoặc tick chọn nhiều bài và xoá hàng loạt.
 - 3.Phân trang đầy đủ cho danh sách.
 - 4.Các nút chức năng thao tác bài hát.
 - o Tích hợp:
 - 1.Giao diện xử lý empty state (khi chưa có bài yêu thích).
 - 2.Xác nhận trước khi xoá (sweetAlert).
 - 3.Ẩn thị thông báo sau mỗi thao tác.
- Playlist
 - o Các component:
 - 1.Trang Playlist hiển thị toàn bộ playlist.
 - 2.Modal tạo playlist mới.
 - 3.Trang chi tiết Playlist hiển thị bài hát trong playlist.
 - 4.Nút thêm, xoá bài hát, xoá playlist.
 - o Tích hợp:
 - 1.Component Table hỗ trợ select, paginate.
 - 2.Modal chọn bài hát để thêm vào playlist.
 - 3.Giao diện responsive cho cả mobile.
- Admin
 - o Tab quản lý bài hát:
 - 1.Ẩn thị danh sách bài hát và thống kê.
 - 2.CRUD bài hát: hiển thị form tạo/sửa, xoá.
 - o Tab quản lý người dùng:

1. Hiện thị ảnh sách người dùng, trạng thái (real time).
 2. Nút khoá, mở khoá tài khoản.
- o Tab quản lý playlist:
 1. Hiện thị danh sách playlist và thống kê.
 - o Tab quản lý nghệ sĩ:
 1. Hiện thị danh sách nghệ sĩ và thống kê.
 - o Tab quản lý thanh toán:
 1. Hiện thị danh sách thanh toán và thống kê.

2.4. Hoàn thiện ứng dụng

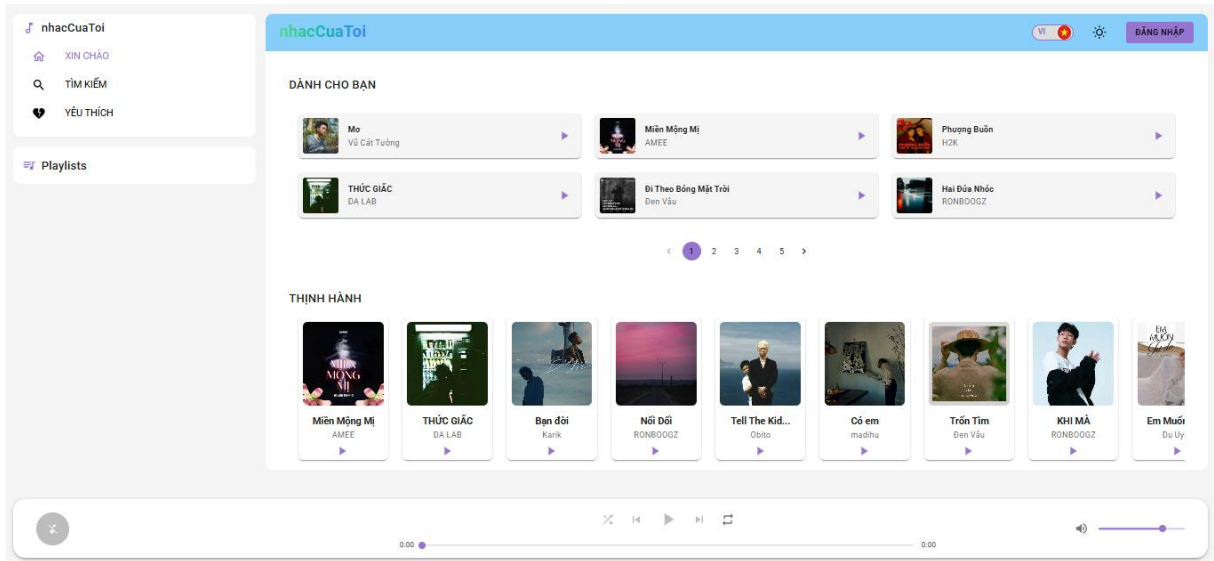
Hoàn thành nhiều chức năng quan trọng của hệ thống Music Manager, bao gồm cả backend (xử lý logic và dữ liệu) lẫn frontend (giao diện người dùng). Dưới đây là phần trình bày kết quả đã thực hiện, kèm theo hình ảnh minh họa.

- **RESTful API (Backend)**

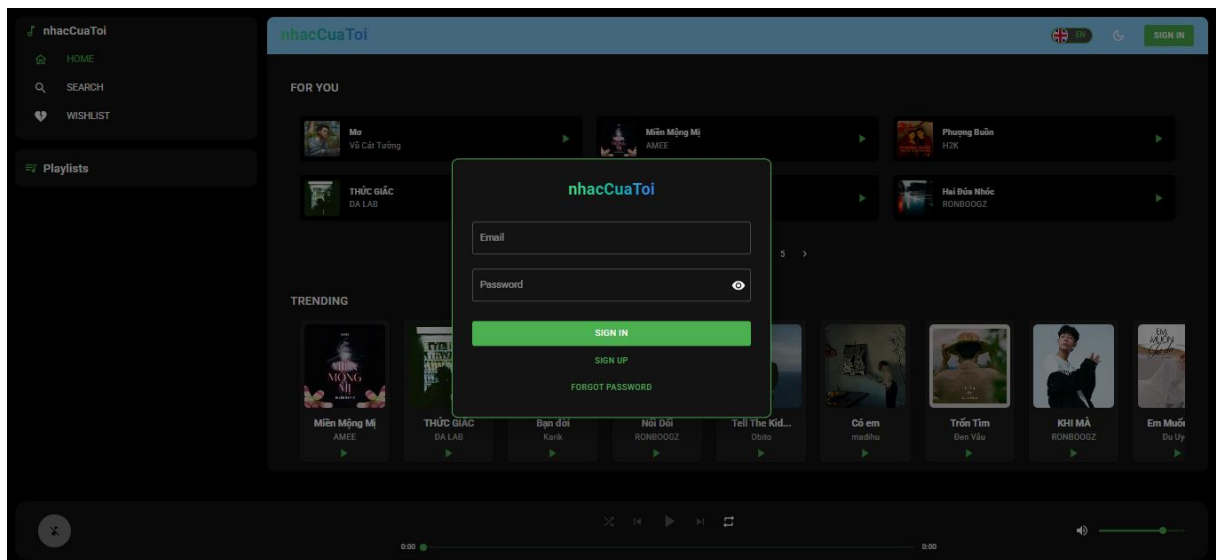


Hình 4: Danh sách API

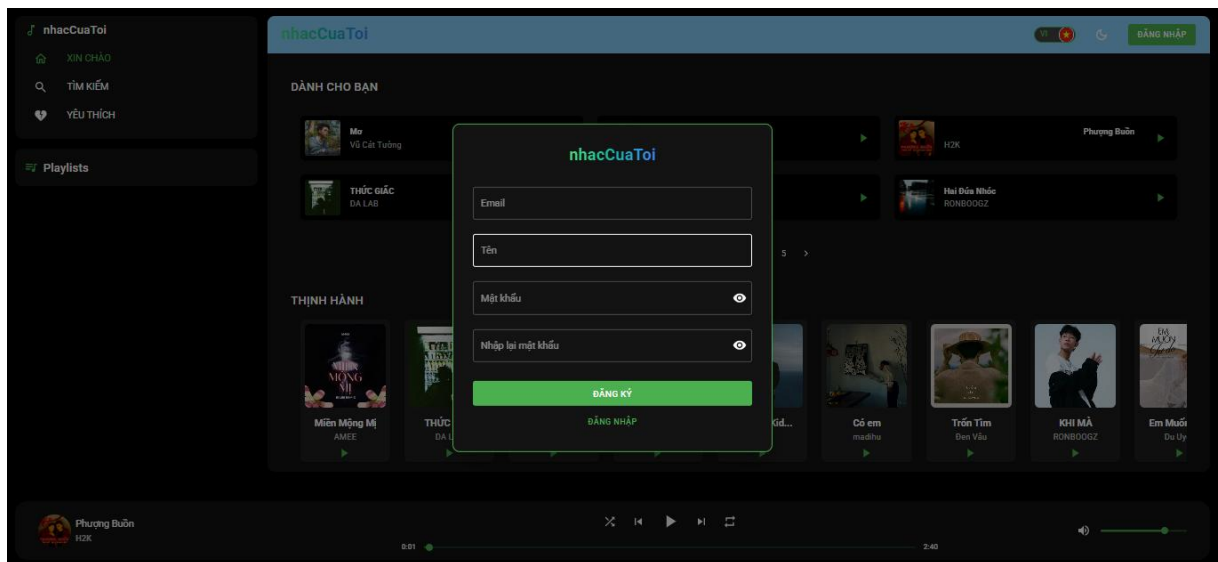
- UI (Frontend)



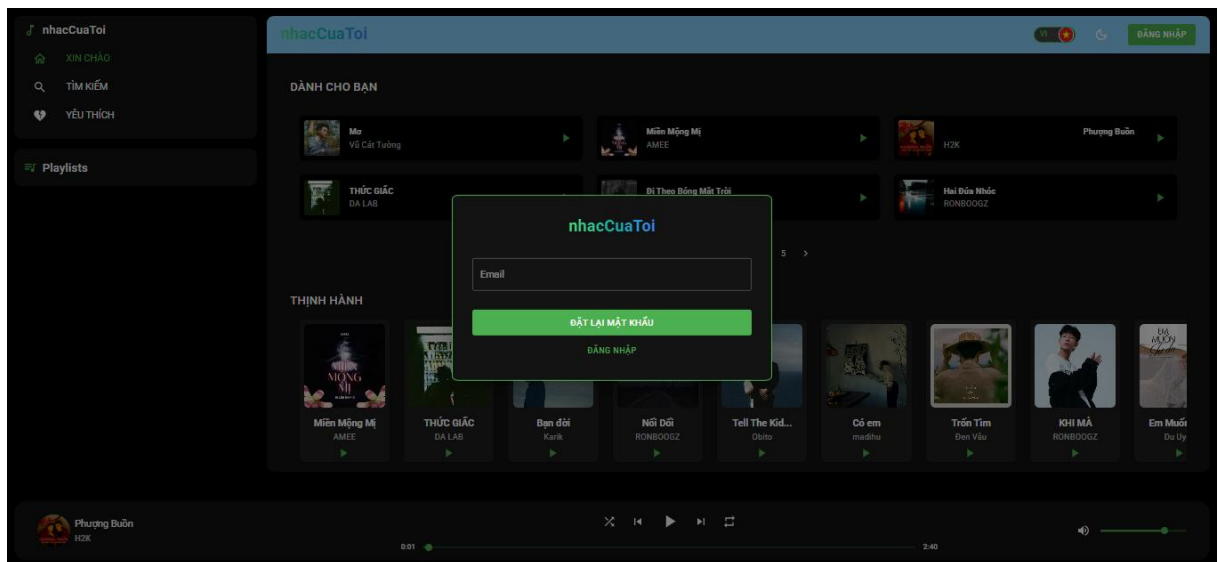
Hình 5: Giao diện trang chủ nền sáng



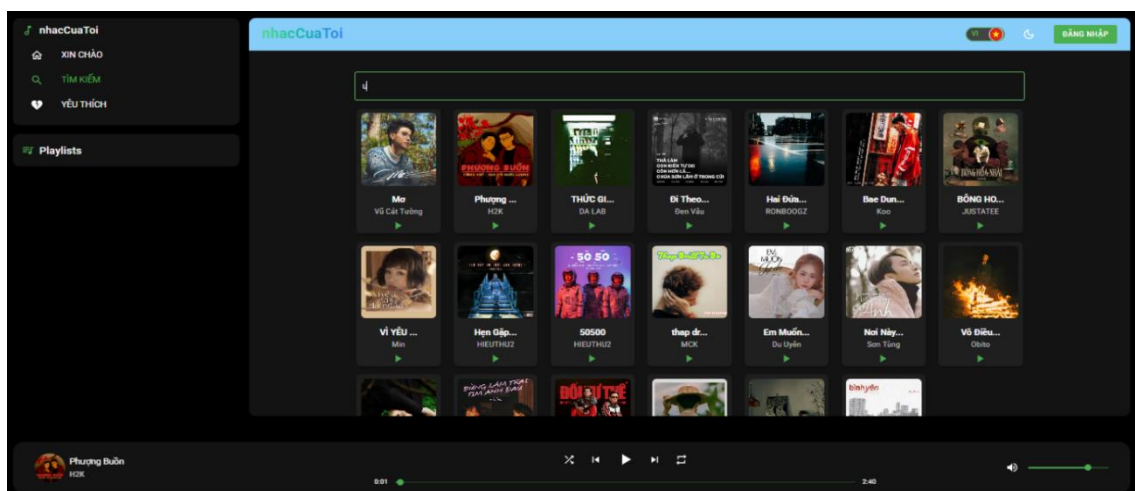
Hình 6: Pop-up đăng nhập



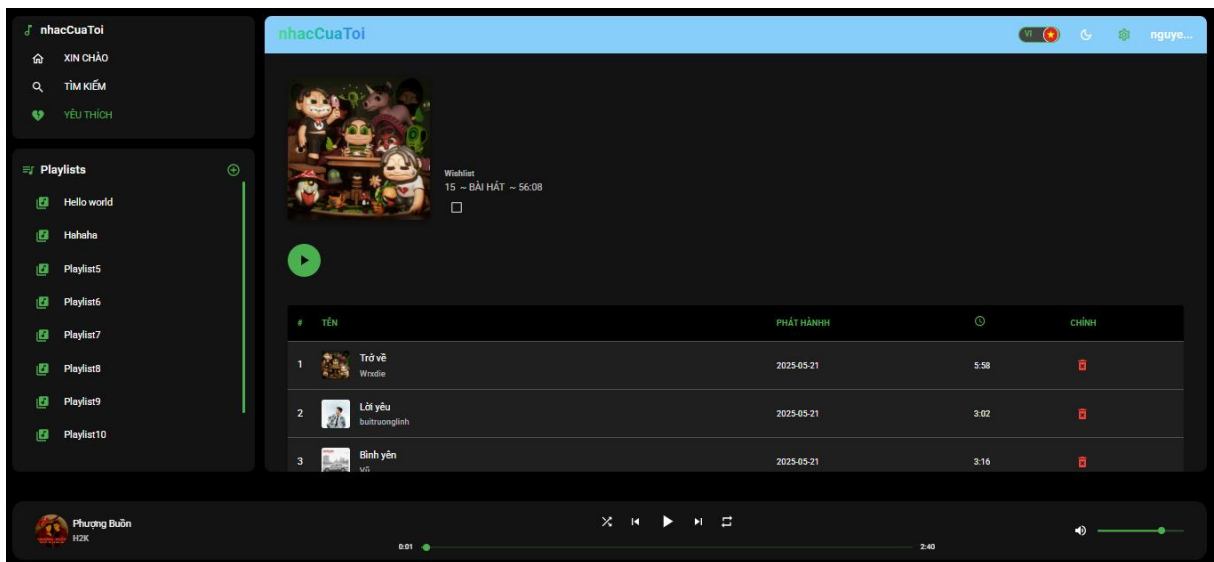
Hình 7: Giao diện đăng ký tài khoản



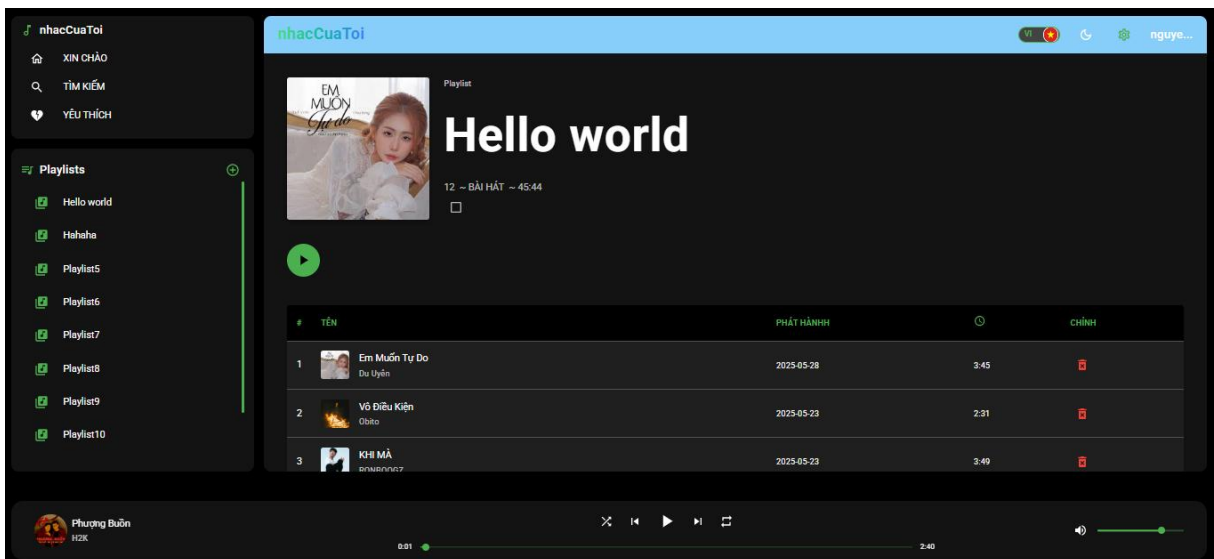
Hình 8: Giao diện đặt lại mật khẩu



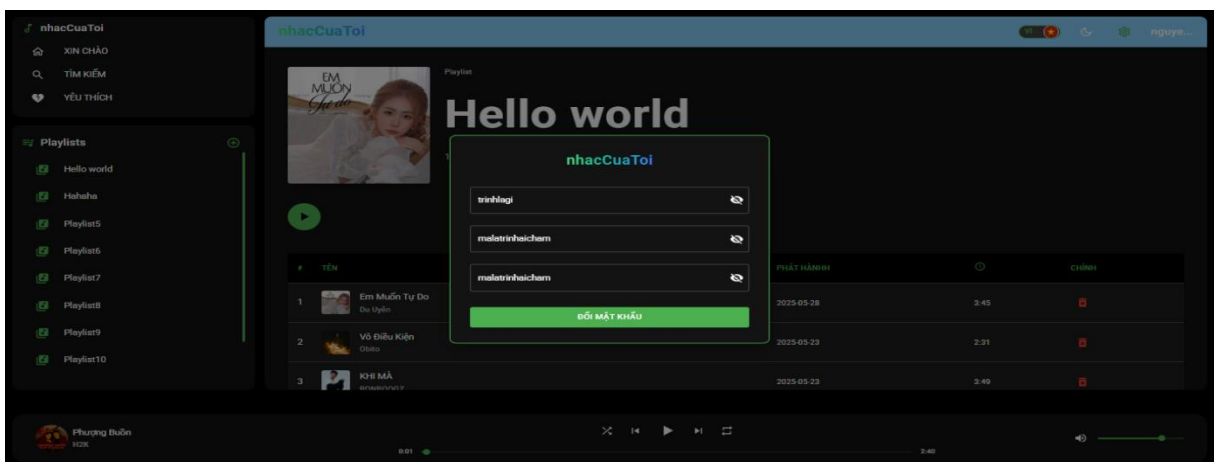
Hình 9: Trang tìm kiếm bài hát



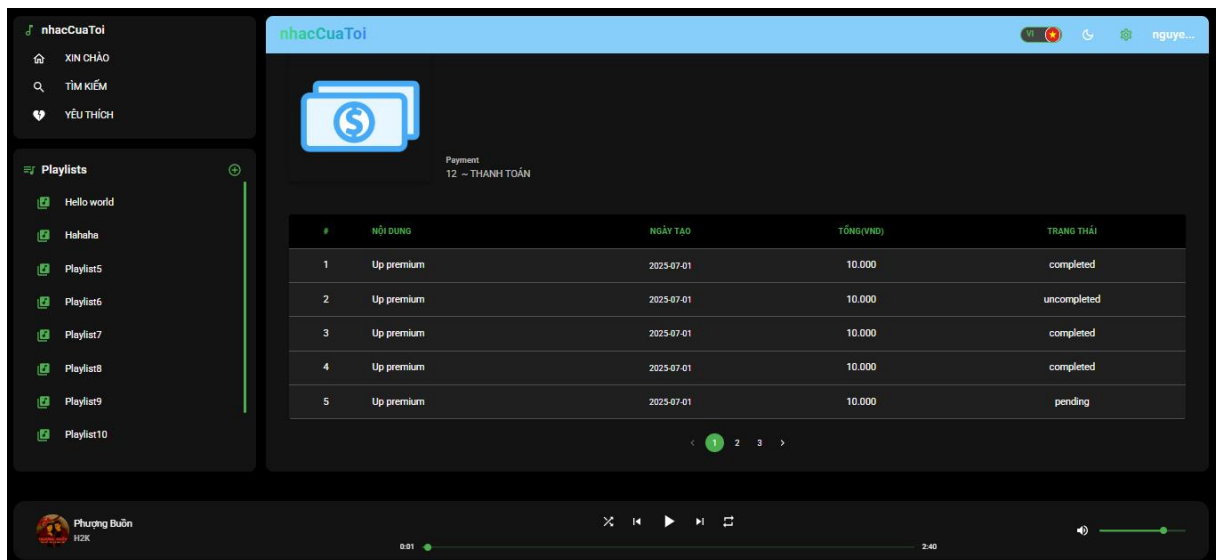
Hình 10: Trang danh sách bài hát yêu thích



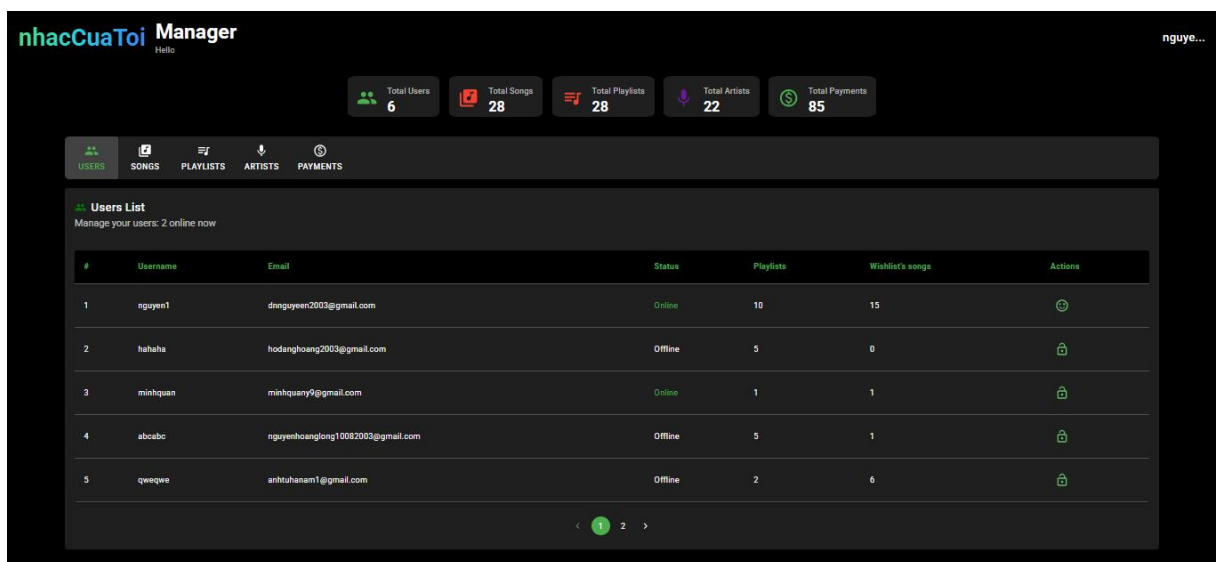
Hình 11: Modal sách bài hát trong playlist



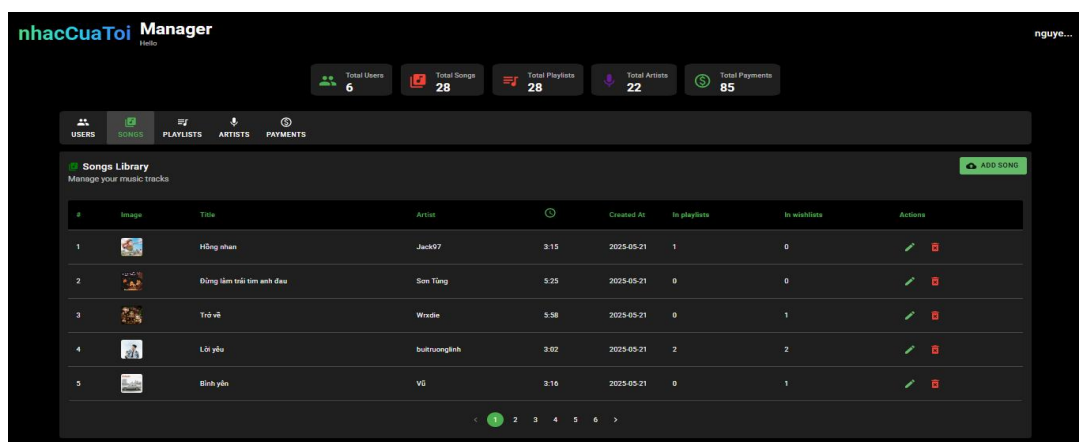
Hình 12: Modal đổi mật khẩu



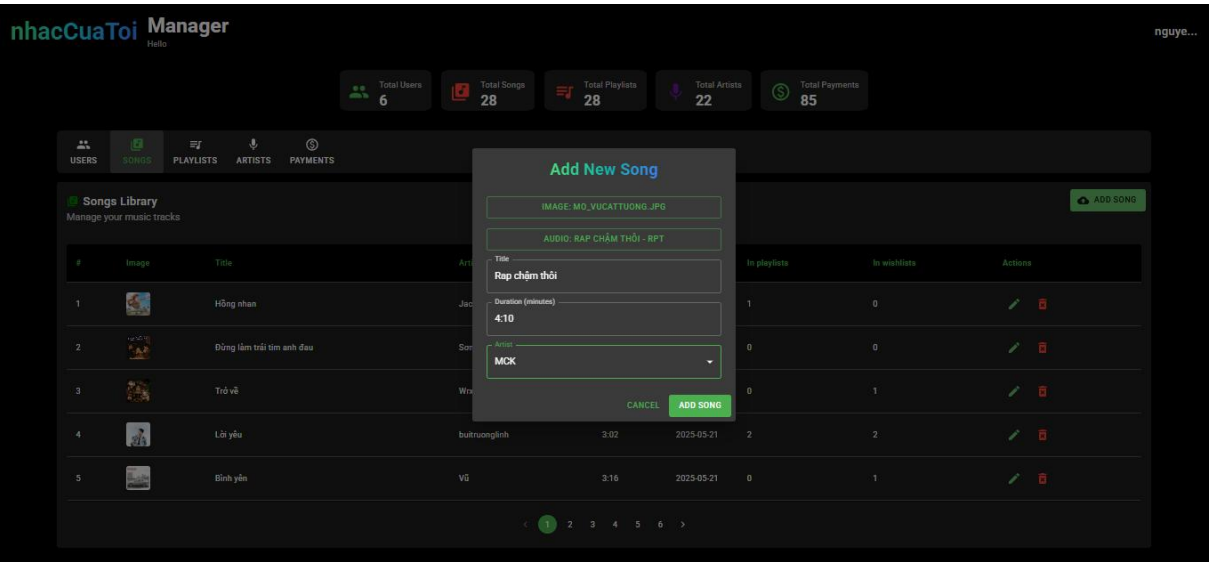
Hình 13: Trang lịch sử thanh toán



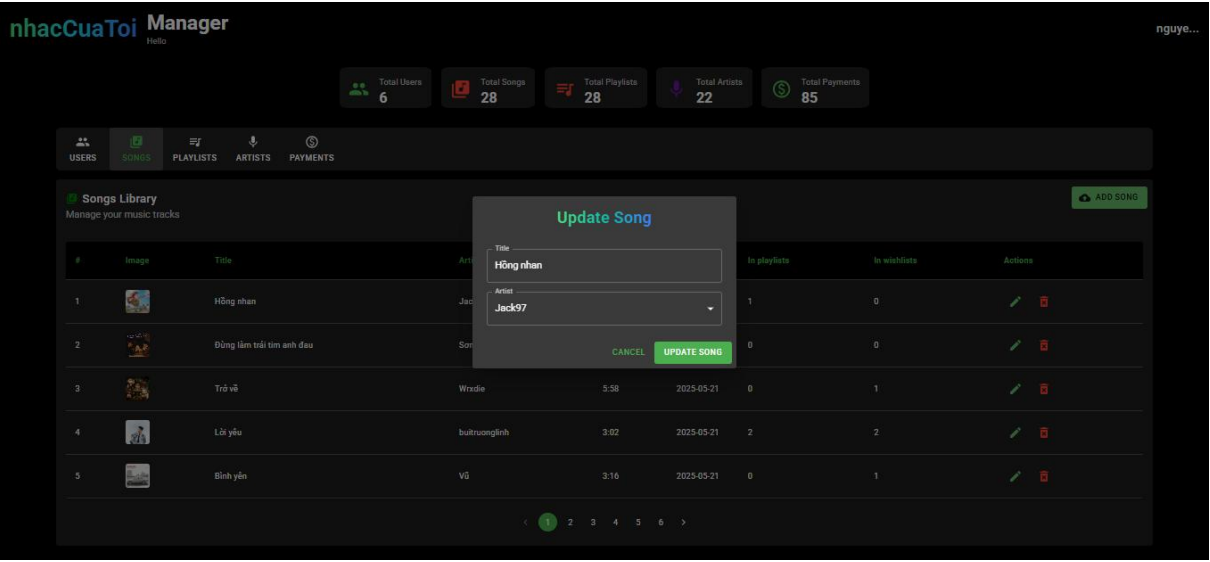
Hình 14: Giao diện quản lý của Admin



Hình 15: Tab quản lý danh sách bài hát



Hình 16: Modal thêm mới bài hát



Hình 17: Modal chỉnh sửa bài hát

nhacCuaToi Manager
Hello

nguye...

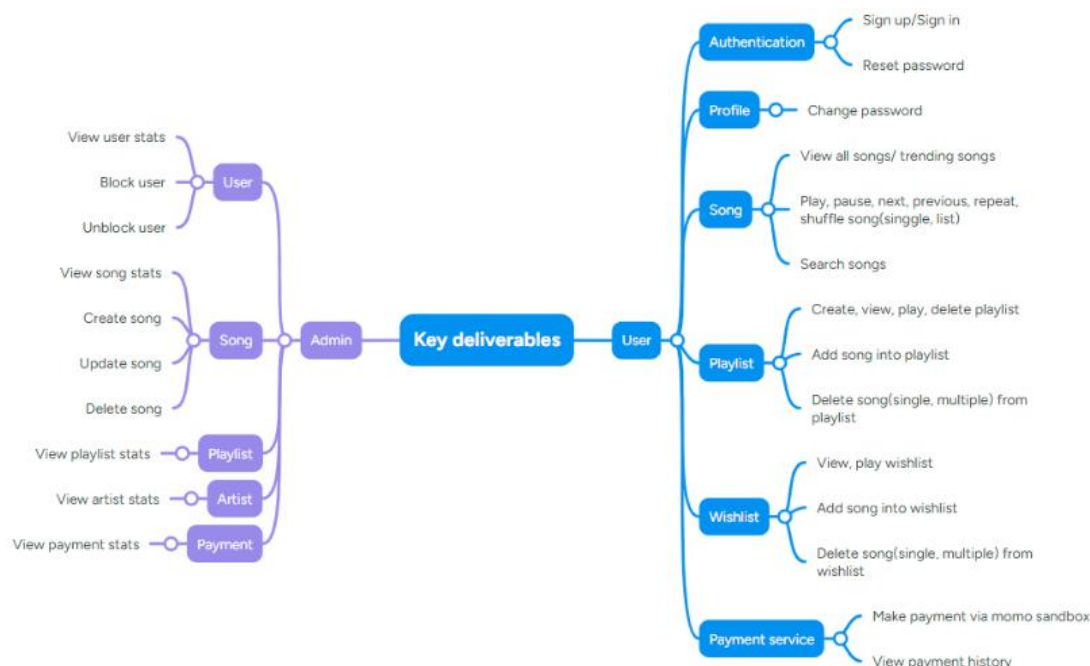
Total Users: 6, Total Songs: 28, Total Playlists: 28, Total Artists: 22, Total Payments: 85

USERS SONGS PLAYLISTS ARTISTS PAYMENTS

Payments
Total profit(VND): 360.000

#	OrderId	RequestId	Created At	Total	Content	Status	UserId
1	2ef3ee3e-e037-48a2-9eb3-a334333d8cfe	9222e2bf-5cd4-4c60-83cd-2ee0d0e91ff3	2025-07-01	10.000	Up premium	pending	1
2	478da905-1ab4-46da-afef-7f90ae750ba6	20fa1911-7ada-482c-ae66-a2ba9ed0e419	2025-07-01	10.000	Up premium	pending	1
3	6b4e0990-b814-4301-8666-a4e6311d38c	9e29d3a4-05da-44ad-a5c6-8f88912dc367	2025-07-01	10.000	Up premium	pending	1
4	0d0acda4-9316-44da-8a36-10b730791a32	9d51e0b8-c375-4516-b4ab-8a44e011abed	2025-07-01	10.000	Up premium	pending	1
5	5d9723af-cd71-43ad-b966-8d95765e7a66	db1c3380-98f4-4e42-8a80-e2462fcd80391	2025-07-01	10.000	Up premium	pending	1
6	c8d89c3c-6bb6-4a57-85a3-1a828eeef40	4dc40ee-bca3-4105-b809-023a20008e20	2025-07-01	10.000	Up premium	pending	1
7	0766f67e-d81e-432c-8196-3150e033b92a	2d3da8a6-757e-41c7-e901-715273e1da87	2025-07-01	10.000	Up premium	pending	1

Hình 18: Tab thống kê của tất cả thanh toán



Hình 19: Các chức năng chính của ứng dụng (key deliverables)

2.5. Tổng quan

Qua quá trình thực tập và phát triển ứng dụng em đã có cơ hội tiếp cận thực tế quy trình xây dựng phần mềm và rút ra nhiều bài học quý báu

- Về kỹ thuật:** Em đã áp dụng thành thạo kiến trúc phân lớp (controller – service – model) để tổ chức mã nguồn backend một cách rõ ràng, dễ bảo trì. Bên cạnh đó, em còn tích hợp thành công Redis vào các luồng xác thực và giới hạn truy cập, triển khai xác thực JWT, xử lý thanh toán với MoMo, xây dựng các API RESTful và tổ chức cơ sở dữ liệu quan hệ với Sequelize ORM. Quá trình xử lý logic API cũng giúp em nâng cao kỹ năng làm việc với dữ liệu và nghiệp vụ.
- Về quy trình:** Em học được cách phân tích yêu cầu, chia nhỏ các chức năng, thiết kế luồng API rõ ràng trước khi lập trình. Việc sử dụng Git để quản lý phiên bản, phân nhánh cho từng chức năng riêng biệt, commit rõ ràng và kiểm thử giúp quá trình phát triển diễn ra ổn định và có thể kiểm soát tốt ứng dụng, dễ dàng cho việc bảo trì và mở rộng.
- Về sản phẩm:** Ứng dụng đã đáp ứng đầy đủ các chức năng như đăng ký, xác thực OTP, đăng nhập, tạo playlist, thêm bài hát vào wishlist, thanh toán, và trang quản trị dành cho admin. Giao diện đơn giản, dễ sử dụng, có hỗ trợ dark mode và đa

ngôn ngữ. Dù vậy, ứng dụng vẫn còn tiềm năng để mở rộng như tích hợp gợi ý bài hát, nghe offline, chia sẻ playlist với người khác...

- **Về khó khăn gặp phải:** Việc tích hợp Redis ban đầu chưa quen nên gặp lỗi xử lý bất đồng bộ. Triển khai thanh toán MoMo sandbox tốn thời gian vì tài liệu hạn chế và thiếu ví dụ mẫu. Xử lý nhiều mối quan hệ trong database đòi hỏi kỹ thuật và tư duy chính xác để tránh lỗi truy vấn hoặc dữ liệu dư thừa. Việc đồng bộ giữa frontend – backend cũng là thử thách, đặc biệt trong việc xử lý lỗi, token hết hạn, và đảm bảo định dạng dữ liệu phản hồi nhất quán.
- **Về kiến thức học được:** Sử dụng thành thạo các công cụ: Postman, Git, Sequelize, Redis, Socket.IO, MoMo API. Hiểu và triển khai được luồng xác thực an toàn. Hiểu được cách xây dựng RESTful API theo chuẩn, sử dụng mã trạng thái HTTP hợp lý và thiết kế phản hồi đồng nhất. Nắm chắc kiến trúc backend hiện đại, tách biệt rõ các lớp, đảm bảo dễ mở rộng về sau. Làm việc nhóm hiệu quả hơn thông qua việc tổ chức mã nguồn rõ ràng, đặt tên nhất quán và phân công chức năng hợp lý.

2.6. Kết luận chương 2

Chương 2 đã trình bày một cách toàn diện quá trình xây dựng ứng dụng quản lý âm nhạc, từ việc hình thành ý tưởng, lựa chọn công nghệ phù hợp, thiết kế kiến trúc hệ thống cho đến triển khai chi tiết các chức năng. Thành quả đạt được không chỉ là một sản phẩm phần mềm hoàn chỉnh và có tính ứng dụng thực tiễn, mà còn là hành trang quý giá về kiến thức và kinh nghiệm mà em tích lũy được trong suốt quá trình thực tập – một nền tảng vững chắc cho con đường phát triển nghề nghiệp trong lĩnh vực phần mềm sau này.

Chương 3: KẾT QUẢ THỰC TẬP

Chương này tổng hợp các kết quả cụ thể đã đạt được trong suốt 12 tuần thực tập tại Công ty TMA Solutions. Nội dung chính bao gồm việc tổng kết các thành tựu về mặt sản phẩm và kỹ năng, nhật ký công việc chi tiết hàng tuần, và các biểu mẫu ghi nhận, đánh giá quá trình thực tập từ phía chuyên gia doanh nghiệp và giảng viên hướng dẫn.

3.1. Tổng kết các kết quả cụ thể đã đạt được

Kết thúc kỳ thực tập, với sự hướng dẫn của chuyên gia tại công ty và giảng viên nhà trường, em đã hoàn thành tốt các nhiệm vụ được giao và đạt được những kết quả cụ thể, rõ ràng trên nhiều phương diện:

3.1.1. Về mặt kiến thức và kỹ năng

Bên cạnh sản phẩm cụ thể, kỳ thực tập đã giúp em củng cố và tích lũy được nhiều kiến thức, kỹ năng quan trọng

- **Về kiến thức công nghệ:**

- **React và Redux:** Nắm vững cách xây dựng giao diện người dùng hiện đại với React, quản lý trạng thái hiệu quả bằng Redux Toolkit, và tối ưu hiệu năng thông qua kỹ thuật phân trang và caching.
- **Thiết kế kiến trúc rõ ràng:** Biết cách tổ chức dự án theo hướng modular, tách biệt từng tính năng, sử dụng React Router, Axios instance, và cấu trúc thư mục dễ mở rộng.
- **Xây dựng API với ExpressJS:** Có khả năng thiết kế RESTful API chuẩn, tách biệt route, controller, service và model, đồng thời áp dụng các nguyên tắc phân lớp rõ ràng để dễ bảo trì.
- **Sử dụng MySQL và Sequelize:** Hiểu và thao tác thành thạo với cơ sở dữ liệu quan hệ, sử dụng Sequelize ORM để tối ưu truy vấn và quan hệ giữa các bảng dữ liệu.
- **Tích hợp Redis và MoMo API:** Biết cách lưu cache và OTP qua Redis, hiểu và tích hợp quy trình thanh toán MoMo (sandbox) với bảo mật và xác nhận từ IPN (Instant Payment Notification).

- **Về kỹ năng mềm:**

- **Kỹ năng giải quyết vấn đề:** Rèn luyện khả năng phân tích một vấn đề kỹ thuật, tìm kiếm các giải pháp khả thi và lựa chọn phương án tối ưu để triển khai.
- **Kỹ năng làm việc theo quy trình:** Làm quen với quy trình phát triển phần mềm cơ bản, từ khâu phân tích yêu cầu, thiết kế, triển khai cho đến kiểm thử.
- **Kỹ năng tự học và nghiên cứu:** Chủ động tìm hiểu tài liệu kỹ thuật, đọc mã nguồn của các thư viện và học hỏi các công nghệ mới để áp dụng vào dự án.

3.1.2. Về mặt tài liệu

- Hoàn thành quyển báo cáo thực tập tốt nghiệp chi tiết, trình bày toàn bộ quá trình thực hiện dự án từ phân tích, thiết kế, triển khai cho đến kết quả và đánh giá.
- Tạo ra bộ mã nguồn hoàn chỉnh, có cấu trúc tốt, sẵn sàng cho việc bảo trì và phát triển trong tương lai.

Chi tiết quá trình thực tập tại công ty qua các tuần, em xin được trình bày trong các bảng sau:

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

Tp. Hồ Chí Minh, ngày 08 tháng 08 năm 2025

BẢNG GHI NHẬN KẾT QUẢ THỰC TẬP HÀNG TUẦN

Một số thông tin liên hệ

Họ và tên: Dương Ngọc Nguyên

Ngày sinh: 03/05/2003

Mã số sinh viên: 3121411148

Lớp: DCT121C2

Ngành học: Công nghệ thông tin

Email: dnnguyen2003@gmail.com

Điện thoại: 0522584519

Chuyên gia doanh nghiệp: Huỳnh Trọng Tuyển

Email: httuyen@tma.com.vn

Điện thoại: 0389842420

Giảng viên hướng dẫn: Nguyễn Tuấn Đăng

Email: dangnt@sgu.edu.vn

Điện thoại: 0913655977

Tuần	Nội dung thực tập (do chuyên gia của doanh nghiệp giao)	Kết quả thực tập (do chuyên gia của doanh nghiệp đánh giá)
------	--	--

1 Từ ngày 12 /05/ 2025 đến ngày 16/ 05/ 2025	<ul style="list-style-type: none"> - HTML, CSS, JavaScript cơ bản (3 ngày JavaScript, 1 ngày HTML/CSS, 3 ngày TypeScript). - Luyện ES6+, classes, async/await, module import/export. 	
2 Từ ngày 19/ 05/ 2025 đến ngày 23/ 05/ 2025	<ul style="list-style-type: none"> - Git/SVN, Linux cơ bản, SASS, JSON Server. - Làm quen môi trường. 	
3 Từ ngày 26/ 05/ 2025 đến ngày 30/ 05/ 2025	<ul style="list-style-type: none"> - ReactJS + Redux: Component, Props, State, Event, Hooks, Routing, Store. 	
4 Từ ngày 02/ 06/ 2025 đến ngày 06/ 06/ 2025	Giao diện & Chức năng cơ bản <ul style="list-style-type: none"> - Login, Register, Forgot Password, Change Password Modal - HomePage: hiển thị danh sách bài hát, Search - PlaylistPage, WishlistPage: Xây dựng UI, SongTable 	
5 Từ ngày 09/ 06/ 2025 đến ngày 13/ 06/ 2025	Admin & Layout <ul style="list-style-type: none"> - Hoàn thiện PlaylistPage, WishlistPage - Update MainLayout (SidebarMenu, HomePage) - AdminPage: Header, StatsDashboard, SongTable, UserTable 	
6 Từ ngày 16/ 06/ 2025	Chức năng nâng cao <ul style="list-style-type: none"> - Admin: Block/Unblock User, Edit/Delete/Modify Song 	

<p>đến ngày 20/ 06/2025</p>	<ul style="list-style-type: none"> - Thêm/Xóa bài vào Playlist/Wishlist - Xây dựng AudioPlayer: play/pause, volume, repeat, random 	
<p>7</p> <p>Từ ngày 23/ 06/ 2025</p> <p>đến ngày 27/ 06/ 2025</p>	<p>UI & Experience</p> <ul style="list-style-type: none"> - Responsive HomePage, PlaylistPage, WishlistPage - SearchPage (search bài hát) - Đổi ngôn ngữ (i18next), Light/Dark mode (Material UI) - PageNotFound, BottomNavigate, PlaybackController responsive 	
<p>8</p> <p>Từ ngày 30/ 06/ 2025</p> <p>đến ngày 04/ 07/ 2025</p>	<p>Tối ưu backend & tính năng thanh toán</p> <ul style="list-style-type: none"> - Swagger API docs - Redis cache backend controller - Xây dựng PaymentPage, PaymentTable - Cập nhật UserMenu, Admin Controller 	
<p>9</p> <p>Từ ngày 07/ 07/ 2025</p> <p>đến ngày 11/ 07/ 2025</p>	<p>Refactor</p> <ul style="list-style-type: none"> - Refactor code, sửa UI PlaylistList - Sửa send email, tối ưu folder structure - Pagination cho table 	
<p>10</p> <p>Từ ngày 14/ 07/ 2025</p> <p>đến ngày 18/ 07/ 2025</p>	<p>Refactor</p> <ul style="list-style-type: none"> - Fix UI bug, refactor PlaybackController - Tích hợp Momo Payment, - Update Payment Controller 	
<p>11</p> <p>Từ ngày 21/ 07/ 2025</p> <p>đến ngày 25/ 07/ 2025</p>	<p>Chức năng realtime quản lý người dùng</p> <ul style="list-style-type: none"> - Track user status (online/offline) qua Socket.IO - Block user realtime từ Admin Dashboard - Cập nhật socket event handler, Admin Controller 	

	- Hiện thị danh sách user online, thao tác block trực tiếp	
12 Từ ngày 28/07/2025 đến ngày 01/08/2025	Refactor & kiểm thử lại các chức năng - Refactor code - Chuẩn hóa cấu trúc thư mục dự án - Kiểm thử toàn bộ flow người dùng và admin để đảm bảo tính ổn định.	
13 Từ ngày 04/08/2025 đến ngày 08/08/2025	Hoàn thiện các báo cáo - Tổng hợp phát triển theo tuần - Viết báo cáo tiến độ, tài liệu. - Chuẩn bị slide thuyết trình và demo sản phẩm - Rà soát và chỉnh sửa nội dung báo cáo theo góp ý	

Chuyên gia doanh nghiệp hướng dẫn thực tập
(Ký tên và ghi họ tên)

Ghi chú:

-**Chuyên gia doanh nghiệp** ghi nhận kết quả thực tập của **sinh viên** theo tuần và gửi qua email cho **giảng viên hướng dẫn** khi kết thúc các tuần 3,6 của đợt thực tập. BẢNG GHI NHẬN KẾT QUẢ THỰC TẬP TỐT NGHIỆP HÀNG TUẦN này là một trong những hồ sơ kèm theo quyền báo cáo thực tập tốt nghiệp.

-Cột **Kết quả thực tập**, chuyên gia doanh nghiệp có thể ghi *Hoàn thành tốt, Hoàn thành, Không đạt* hoặc có thể ghi nhận chi tiết hơn.

BẢNG ĐÁNH GIÁ QUÁ TRÌNH THỰC TẬP TỐT NGHIỆP

Mẫu 07 TTTN k.CNTT

ỦY BAN NHÂN DÂN TP. HỒ CHÍ
MINH TRƯỜNG ĐẠI HỌC SÀI GÒN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

KHOA CNTT

BẢNG ĐÁNH GIÁ QUÁ TRÌNH THỰC TẬP TỐT NGHIỆP

(do chuyên gia doanh nghiệp đánh giá).

Họ và tên sinh viên: Dương Ngọc Nguyên

Ngày sinh: 03/05/2003

Mã số sinh viên: 3121411148

Lớp: DCT121C2

Thời gian thực tập: 12/5/2025 – 08/08/2025

Doanh nghiệp thực tập: Công ty TNHH Giải Pháp Phần Mềm Tường Minh (TMA Solutions)

Địa chỉ doanh nghiệp: 10 Đặng Văn Ngữ, Phường Phú Nhuận, TP. Hồ Chí Minh

Chuyên gia doanh nghiệp hướng dẫn: Huỳnh Trọng Tuyển

I. ĐÁNH GIÁ VỀ QUÁ TRÌNH THỰC TẬP

STT	Nội dung đánh giá	ĐIỂM		
		0	0.5	1
1	Khả năng thực hành			
2	Khả năng làm việc nhóm			
3	Tính thân thiện			
4	Tính năng động			
5	Tính thần sáng tạo			
6	Chấp hành nội quy cơ quan			

7	Giờ giấc làm việc			
8	Phương pháp làm việc			
9	Khối lượng công việc			
10	Báo cáo thực tập tốt nghiệp			

(theo thang điểm 10).

II. CÁC ĐÁNH GIÁ KHÁC:

.....

.....

.....

.....

III. KẾT QUẢ TỔNG HỢP:

Điểm tổng cộng :

XÁC NHẬN CỦA DOANH NGHIỆP

(đóng mộc tròn của doanh nghiệp, họ tên, ký tên)

Chuyên gia hướng dẫn

(Ký và ghi họ tên)

PHIẾU ĐÁNH GIÁ KẾT QUẢ THỰC TẬP TỐT NGHIỆP

Mẫu 08 TTTN k.CNTT

ỦY BAN NHÂN DÂN TP. HỒ CHÍ
MINH TRƯỜNG ĐẠI HỌC SÀI GÒN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

KHOA CNTT

PHIẾU ĐÁNH GIÁ KẾT QUẢ THỰC TẬP TỐT NGHIỆP

(do giảng viên hướng dẫn đánh giá)

Họ và tên sinh viên: Dương Ngọc Nguyên

Ngày sinh: 03/05/2003

Mã số sinh viên: 3121411148

Lớp: DCT121C2

Thời gian thực tập: 12/05/2025 – 08/08/2025

Doanh nghiệp thực tập: Công ty TNHH Giải Pháp Phần Mềm Tường Minh (TMA Solutions)

Địa chỉ doanh nghiệp: 10 Đặng Văn Ngữ, Phường Phú Nhuận, TP. Hồ Chí Minh

Chuyên gia doanh nghiệp hướng dẫn: Huỳnh Trọng Tuyên

I. ĐIỂM CỦA CHUYÊN GIA DOANH NGHIỆP :

(thang điểm 10)

II. ĐIỂM CỦA GIẢNG VIÊN HƯỚNG DẪN:

(thang điểm 10)

III. ĐIỂM TỔNG KẾT:

(trung bình cộng 2 cột điểm trên, thang điểm 10)

Xếp loại :

TP Hồ Chí Minh ngày 08 tháng 08 năm 2025

Giảng viên hướng dẫn

3.2. Kết luận chương 3

Chương 3 đã tổng hợp một cách chi tiết các công việc và kết quả đạt được trong suốt kỳ thực tập. Nhật ký công việc hàng tuần cho thấy sự tiến bộ rõ rệt trong việc áp dụng kiến thức và kỹ năng để hoàn thành một dự án phần mềm hoàn chỉnh. Các kết quả cụ thể về sản phẩm, kiến thức, kỹ năng cùng các biểu mẫu đánh giá đính kèm là minh chứng khách quan cho sự nỗ lực và thành quả mà em đã đạt được dưới sự hướng dẫn của công ty và nhà trường.

Chương 4: KẾT LUẬN VÀ KIẾN NGHỊ

4.1. Kết luận

Kỳ thực tập tốt nghiệp tại Công ty TMA Solutions là một giai đoạn học hỏi và trải nghiệm vô cùng quý báu, là cầu nối quan trọng giúp em liên kết giữa những kiến thức lý thuyết đã được trang bị tại nhà trường với môi trường làm việc thực tế đầy năng động và chuyên nghiệp.

Trong suốt 12 tuần thực tập, dưới sự hướng dẫn tận tình của chuyên gia tại công ty và giảng viên hướng dẫn, em đã hoàn thành tốt nhiệm vụ được giao là xây dựng ứng dụng "Music Manager". Qua đó, em không chỉ đạt được mục tiêu về mặt sản phẩm là một ứng dụng web hoàn chỉnh, mà quan trọng hơn là đã tích lũy được những kiến thức chuyên sâu và kỹ năng thực tiễn giá trị.

Em đã có cơ hội áp dụng và làm chủ các công nghệ hiện đại như ReactJS và NodeJS để xây dựng ứng dụng, áp dụng thành thạo Redux để quản lý trạng thái phức tạp, và tích hợp RESTful API được hoàn thiện với ExpressJS. Đặc biệt, việc làm việc với các công nghệ mới đã cho em một cái nhìn sâu sắc hơn về lĩnh vực công nghệ và cách tích hợp vào các sản phẩm phần mềm.

Bên cạnh kiến thức chuyên môn, em còn rèn luyện được các kỹ năng mềm quan trọng như kỹ năng phân tích vấn đề, tự nghiên cứu tài liệu, làm việc theo quy trình và quản lý thời gian. Kỳ thực tập đã thực sự giúp em trưởng thành hơn, tự tin hơn và có một định hướng rõ ràng hơn cho con đường sự nghiệp sau khi tốt nghiệp.

4.2. Kiến nghị

Để quá trình thực tập và phát triển dự án được tốt hơn trong tương lai, em xin đưa ra một số kiến nghị như sau:

4.2.1. Hướng phát triển của đề tài

Sản phẩm ứng dụng Music Manager đã hoàn thành các chức năng cơ bản, tuy nhiên vẫn còn nhiều tiềm năng để cải tiến và phát triển thêm các tính năng nâng cao, bao gồm:

- **Hỗ trợ offline:** Xây dựng cơ chế lưu trữ dữ liệu tạm thời trên thiết bị, cho phép người dùng nghe lại các bài hát ngay cả khi không có kết nối Internet.
- **Tương tác bằng giọng nói:** Tích hợp các công nghệ nhận dạng giọng nói (Speech-to-Text) và tổng hợp giọng nói (Text-to-Speech) để người dùng có thể thực hiện tìm kiếm mà không cần gõ phím.
- **Tính năng nghe nhạc chung:** Xây dựng chức năng cho phép nhiều người dùng có thể vào chung phòng nghe nhạc chung một cách đồng bộ
- **Tích hợp AI:** Tận dụng thế mạnh của AI để triển khai các tính năng gợi ý bài hát, playlist dựa trên cá nhân hóa.

4.2.2. Kiến nghị với doanh nghiệp (Công ty TMA Solutions)

Trước hết, em xin một lần nữa cảm ơn công ty đã tạo điều kiện tốt nhất cho em trong suốt kỳ thực tập. Để công tác hướng dẫn sinh viên thực tập trong các đợt tới được hiệu quả hơn nữa, em xin có một vài đề xuất nhỏ:

- Xem xét xây dựng một chương trình "onboarding" hoặc một dự án khởi động nhỏ dành cho thực tập sinh mới, giúp các bạn nhanh chóng làm quen với quy trình làm việc và bộ công cụ của công ty.
- Tổ chức thêm các buổi chia sẻ kỹ thuật (technical sharing) định kỳ, nơi các thực tập sinh có thể trình bày về những khó khăn, giải pháp của mình và học hỏi kinh nghiệm từ các lập trình viên có kinh nghiệm.

4.2.3. Kiến nghị với Khoa và Nhà trường (Khoa CNTT - ĐH Sài Gòn)

Em xin chân thành cảm ơn Khoa Công nghệ Thông tin và Nhà trường đã tổ chức và tạo điều kiện cho sinh viên được tham gia kỳ thực tập tốt nghiệp đầy ý nghĩa và thiết thực này. Thông qua kỳ thực tập, em đã có cơ hội tiếp cận với môi trường làm việc thực tế, nâng cao kỹ năng chuyên môn, kỹ năng mềm và hiểu rõ hơn về yêu cầu cũng như cơ hội nghề nghiệp trong lĩnh vực công nghệ thông tin.

Để chương trình thực tập ngày càng hoàn thiện và mang lại giá trị cao hơn cho sinh viên, em xin mạnh dạn đưa ra một số kiến nghị như sau:

Tăng cường tổ chức các buổi workshop, seminar, tọa đàm chuyên đề về những công nghệ và công cụ hiện đại đang được ứng dụng rộng rãi trong thực tế như ReactJS, NodeJS, Spring Boot, Docker, Kubernetes, các nền tảng điện toán đám mây (AWS, Azure, GCP), cũng như các thư viện quản lý trạng thái phổ biến như Redux, Zustand,... Những buổi chia sẻ này sẽ giúp sinh viên cập nhật kiến thức mới, định hướng rõ ràng hơn về công nghệ phù hợp với nhu cầu tuyển dụng.

Bổ sung các môn học chuyên sâu mang tính thực tiễn cao vào chương trình đào tạo chính khóa như: phát triển ứng dụng web và di động, kỹ thuật triển khai hệ thống phần mềm, kiểm thử phần mềm, bảo mật thông tin, AI ứng dụng trong sản phẩm thực tế,... để sinh viên có sự chuẩn bị tốt hơn trước khi bước vào môi trường làm việc.

Đẩy mạnh hơn nữa mối quan hệ hợp tác giữa Khoa và các doanh nghiệp phần mềm, thông qua các hoạt động như thực tập, tham quan doanh nghiệp, tài trợ học bổng, hoặc các chương trình đào tạo theo nhu cầu doanh nghiệp (on-demand training). Điều này không chỉ tạo ra nhiều cơ hội thực tập chất lượng hơn mà còn giúp Khoa và Nhà trường nắm bắt kịp thời xu hướng và yêu cầu thực tế của ngành, từ đó cập nhật và điều chỉnh nội dung giảng dạy một cách linh hoạt, phù hợp với thị trường lao động.

Khuyến khích sự kết nối giữa cựu sinh viên và sinh viên hiện tại thông qua các buổi giao lưu, chia sẻ kinh nghiệm nghề nghiệp để giúp sinh viên định hướng rõ ràng hơn trong quá trình học tập và phát triển bản thân.

Em rất mong những kiến nghị trên sẽ góp phần nhỏ bé vào quá trình nâng cao chất lượng đào tạo của Khoa và Nhà trường, giúp sinh viên Công nghệ Thông tin Đại học Sài Gòn ngày càng tự tin, bản lĩnh và sẵn sàng hội nhập vào môi trường làm việc chuyên nghiệp.

TÀI LIỆU THAM KHẢO

- [1]. Learning React: Modern Patterns for Developing React Apps
- [2]. Web Development with Node and Express: Leveraging the JavaScript Stack