 **BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC THĂNG LONG**Logo, company name

Description automatically generated

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG HỆ THỐNG KHUYẾN NGHỊ  
BÀI HÁT THEO SỞ THÍCH**

**Giảng viên hướng dẫn: TS. Nguyễn Thị Huyền Châu**

**Sinh viên thực hiện: A40714 – Phùng Tiến Thưởng**

**A28198 – Lương Gia Huy**

**Chuyên ngành: Trí tuệ nhân tạo**

**HÀ NỘI – 2024**

MINISTRY OF EDUCATION AND TRAINING

**THANG LONG UNIVERSITY**

---o0o---



**GRADUATION THESIS**

**DEVELOPING A SONG RECOMMENDATION SYSTEM**

**Intructor: Dr. Nguyen Thi Huyen Chau**

**Student: A40714 – Phung Tien Thuong**

**A28198 – Luong Gia Huy**

**HANOI - 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

Logo, company name

Description automatically generated**TRƯỜNG ĐẠI HỌC THĂNG LONG**

Logo, company name

Description automatically generated

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG HỆ THỐNG KHUYẾN NGHỊ  
BÀI HÁT THEO SỞ THÍCH**

**Giảng viên hướng dẫn: TS. Nguyễn Thị Huyền Châu**

**Sinh viên thực hiện: A40714 – Phùng Tiến Thưởng**

**A28198 – Lương Gia Huy**

**Chuyên ngành: Trí Tuệ nhân tạo**

**HÀ NỘI – 2024**

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin được gửi lời cảm ơn chân thành nhất tới toàn thể các thầy cô giáo trong khoa Công nghệ thông tin cũng như các thầy cô giảng dạy trong trường Đại học Thăng Long đã truyền đạt những kiến thức quý báu và bổ ích nhất cho chúng em trong những năm học vừa qua.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành và sâu sắc tới cô Nguyễn Thị Huyền Châu, giảng viên khoa Công nghệ thông tin, Đại học Thăng Long. Thời gian qua nhờ vào sự hướng dẫn tận tình và sự động viên, giúp đỡ ân cần từ cô Châu mà em mới có thành quả là xây dựng và hoàn thiện được Khoá luận tốt nghiệp như ngày hôm nay.

Bên cạnh đó, chúng em cũng xin chân thành gửi lời cảm ơn đến tất cả những người bạn cùng khoá, những người anh, người chị khoá trên đã nhiệt tình giúp đỡ và chia sẻ cho chúng em những kinh nghiệm và kiến thức đáng quý trong suốt quá trình chúng em xây dựng dự án này.

Vì kiến thức còn nhiều hạn chế, trong quá trình triển khai và thực hiện Khoá luận tốt nghiệp này chúng em không tránh khỏi có những sai sót không mong muốn, chính vì vậy chúng em rất mong nhận được những ý kiến đóng góp từ các thầy cô cũng như các bạn trong khoa để hoàn thiện dự án này cũng như bổ sung những kiến thức còn thiếu sót.

Cuối cùng chúng em xin kính gửi lời chúc tới toàn thể các thầy cô trong khoa Công nghệ thông tin cũng như các thầy cô đang giảng dạy trong trường Đại học Thăng Long luôn luôn mạnh khỏe và tràn đầy nhiệt huyết để truyền đạt kiến thức cho các thế hệ mai sau.

Hà Nội, ngày 20 tháng 06 năm 2024

Sinh viên thực hiện

Phùng Tiến Thưởng

Sinh viên thực hiện

Lương Gia Huy

**LỜI CAM ĐOAN**

Em xin cam đoan: Khóa luận tốt nghiệp với đề tài “Lập trình ứng dụng nhận diện cây trên nền tảng iOS” trong báo cáo này là của em. Những gì em viết ra không có sự sao chép từ các tài liệu, không sử dụng kết quả của người khác mà không trích dẫn cụ thể. Đây là công trình nghiên cứu của em phát triển, không sao chép mã nguồn của người khác. Nếu vi phạm những điều trên, em xin chấp nhận tất cả những truy cứu về trách nhiệm theo quy định của Trường Đại học Thăng Long.Hà Nội, ngày 31 tháng 07 năm 2022

Hà Nội, ngày 20 tháng 06 năm 2024

Sinh viên thực hiện

Phùng Tiến Thưởng

Sinh viên thực hiện

Lương Gia Huy

**MỤC LỤC**

[CHƯƠNG MỞ ĐẦU 3](#_Toc170667152)

[I. Mục tiêu 3](#_Toc170667153)

[II. Động lực bản thân 3](#_Toc170667154)

[III. Phạm vi và phương pháp nghiên cứu 4](#_Toc170667155)

[IV. Cấu trúc khóa luận 4](#_Toc170667156)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 5](#_Toc170667157)

[1.1 Tổng quan về trí tuệ nhân tạo [1] 5](#_Toc170667158)

[1.1.1. Khái niệm 5](#_Toc170667159)

[1.1.2. Các công nghệ phổ biến trong trí tuệ nhân tạo 5](#_Toc170667160)

[1.1.3. Các ứng dụng AI phổ biến 5](#_Toc170667161)

[1.2. Hệ thống khuyến nghị [2] 6](#_Toc170667162)

[1.3. Phương pháp lọc dựa trên nội dung (Content-based Filtering) [3] 7](#_Toc170667163)

[1.3.1. Nguyên lý hoạt động 7](#_Toc170667164)

[1.3.2. Các bước chính trong phương pháp Content-Based: 7](#_Toc170667165)

[1.3.3. Ứng dụng của tf-idf trong Content-Based 8](#_Toc170667166)

[1.3.4. Ưu điểm của phương pháp Content-Based: 8](#_Toc170667167)

[1.3.5. Hạn chế của phương pháp Content-Based: 9](#_Toc170667168)

[1.4. Phương pháp lọc cộng tác (Collaborative Filtering) [4] 9](#_Toc170667169)

[1.4.1. Nguyên lý hoạt động 9](#_Toc170667170)

[1.4.2. Ma trận người dùng - sản phẩm (User-Item Matrix): 9](#_Toc170667171)

[1.4.3. Các loại Collaborative Filtering 10](#_Toc170667172)

[1.4.4. User-Based Collaborative Filtering 10](#_Toc170667173)

[1.4.5. Item-Based Collaborative Filtering 10](#_Toc170667174)

[1.4.6. Ưu điểm của phương pháp Collaborative Filtering 11](#_Toc170667175)

[1.4.7. Hạn chế của phương pháp Collaborative Filtering 11](#_Toc170667176)

[1.5. Ngôn ngữ sử dụng 12](#_Toc170667177)

[1.5.1. JavaScript [5] 12](#_Toc170667178)

[1.5.2. Python [6] 13](#_Toc170667179)

[1.6. Các framework sử dụng 14](#_Toc170667180)

[1.6.1. FastAPI [7] 14](#_Toc170667181)

[1.6.2. VueJS [8] 15](#_Toc170667182)

[1.7. Tích hợp các nền tảng khác 16](#_Toc170667183)

[1.7.1. S3 storage [9] 16](#_Toc170667184)

[1.7.2. MongoDB [10] 17](#_Toc170667185)

[1.7.3. Microsoft SQL Server[11] 18](#_Toc170667186)

[CHƯƠNG 2.  ĐẶC TẢ VÀ THIẾT KẾ HỆ THÔNG 19](#_Toc170667187)

[2.1. Mô tả bài toán 19](#_Toc170667188)

[2.1.1. Gợi ý nhạc 19](#_Toc170667189)

[2.1.2. Tạo một danh sách 19](#_Toc170667190)

[2.1.3. Tìm kiếm bài hát 19](#_Toc170667191)

[2.2. Yêu cầu nghiệp vụ 19](#_Toc170667192)

[2.3. Sơ đồ Usecase các chức năng chính 20](#_Toc170667193)

[2.3.1. Sơ đồ Usecase 20](#_Toc170667194)

[2.3.2. Các Usecase 21](#_Toc170667195)

[2.3.3. Bảng ánh xạ yêu cầu nghiệp vụ với các Usecase 22](#_Toc170667196)

[2.4. Sơ đồ kiến trúc tổng thể 23](#_Toc170667197)

[2.5. Cấu trúc chung 24](#_Toc170667198)

[2.6. Thiết kế một số chức năng 24](#_Toc170667199)

[2.6.1. UC#01: Đăng ký 24](#_Toc170667200)

[2.6.2. UC#04: Tạo danh sách tự tạo 25](#_Toc170667201)

[2.6.3. UC#05: Thêm một bài hát vào danh sách tự tạo 25](#_Toc170667202)

[2.6.4. UC#09: Lấy danh sách các bài hát của danh sách tự tạo 26](#_Toc170667203)

[2.6.5. UC#11: Gợi ý bài hát theo đặc điểm từng người dùng 26](#_Toc170667204)

[2.6.6. UC#15: Phát bài hát tiếp theo trong hàng đợi 27](#_Toc170667205)

[2.6.7. UC#17: Thêm bài hát mới vào database 27](#_Toc170667206)

[2.7. Thiết kế CSDL 28](#_Toc170667207)

[2.7.1. Mô tả hệ thống 28](#_Toc170667208)

[2.7.2. Lựa chọn các danh từ làm lớp thực thể hoặc thuộc tính 28](#_Toc170667209)

[2.7.3. Mô tả tổng quan từng bảng 29](#_Toc170667210)

[2.7.4. Chi tiết các bảng 29](#_Toc170667211)

[2.8. Thiết kế API 31](#_Toc170667212)

[2.8.1. Cấu trúc tổng thể của API 31](#_Toc170667213)

[2.8.2. Api hệ thống sử dụng 31](#_Toc170667214)

[CHƯƠNG 3: THU THẬP DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH 37](#_Toc170667215)

[3.1. Thu thập dữ liệu 37](#_Toc170667216)

[3.2. Thống kê và tiền xử lý dữ liệu 37](#_Toc170667217)

[3.2.1. Bộ dữ liệu train\_triplets 37](#_Toc170667218)

[3.2.2. Bộ dữ liệu unique\_tracks 38](#_Toc170667219)

[3.2.3. Tiền xử lý 38](#_Toc170667220)

[3.3. Xây dựng mô hình 40](#_Toc170667221)

[3.3.1. Phân chia dữ liệu 40](#_Toc170667222)

[3.3.2. Xây dựng SparseTensor 41](#_Toc170667223)

[3.3.3. Khởi tạo embedding 41](#_Toc170667224)

[3.3.4. Tính toán lỗi 42](#_Toc170667225)

[3.3.5. Huấn luyện mô hình 43](#_Toc170667226)

[3.3.6. Đánh giá 45](#_Toc170667227)

[3.3.7. Xây dựng hàm gợi ý 45](#_Toc170667228)

[CHƯƠNG 4. CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG 47](#_Toc170667229)

[4.1. Môi trường cài đặt 47](#_Toc170667230)

[4.2. Trình diễn các chức năng của hệ thống 47](#_Toc170667231)

[4.2.1. Chức năng đăng nhập 47](#_Toc170667232)

[4.2.2. Chức năng thêm một bài hát vào danh sách yêu thích 48](#_Toc170667233)

[4.2.3. Lấy các danh sách bài hát của danh sách tự tạo 49](#_Toc170667234)

[4.2.4. Chức năng tìm kiếm bài hát 50](#_Toc170667235)

[4.2.5. Chức năng thêm mới dữ liệu 51](#_Toc170667236)

[KẾT LUẬN 54](#_Toc170667237)

[TÀI LIỆU THAM KHẢO 55](#_Toc170667238)

**DANH MỤC BẢNG BIỂU**

Bảng 1.1: Ví dụ về ma trận người dùng – sản phẩm (User-Item Matrix)………………………9

Bảng 2.1: Bảng ánh xạ yêu cầu nghiệp vụ với các Usecase…………………………………..22

Bảng 2.2: Danh sách các bảng trong CSDL…………………………………………………..29

Bảng 2.3: Bảng cơ sở dữ liệu Users…………………………………………………………...29

Bảng 2.4: Bảng cơ sở dữ liệu MuscicListCustom…………………………………………….30

Bảng 2.5: Bảng cơ sở dữ liệu SongData………………………………………………………30

Bảng 2.6: Bảng cơ sở dữ liệu UserData……………………………………………………….30

Bảng 2.7: API hệ thống sử dụng………………………………………………………………31

**DANH MỤC HÌNH ẢNH**

Hình 2.1: Sơ đồ Usecase hệ thống…………………………………………………………...20

Hình 2.2: Sơ đồ kiến trúc tổng thể hệ thống…………………………………………………23

Hình 2.3: Cấu trúc chung của chức năng……………………………………………………..24

Hình 2.4: Sơ đồ tuần tự chức năng Đăng ký…………………………………………………..24

Hình 2.5: Sơ đồ tuần tự chức năng Tạo danh sách tự tạo…………………………………….25

Hình 2.6: Sơ đồ tuần tự chức năng Thêm một bài hát vào danh sách tự tạo…………………25

Hình 2.7: Sơ đồ tuần tự chức năngLấy danh sách các bài hát của danh sách tự tạo………….26

Hình 2.8: Sơ đồ tuần tự chức năng Gợi ý bài hát theo từng người dùng ……………………..26

Hình 2.9: Sơ đồ tuần tự chức năng Phát bài hát tiếp theo trong hàng đợi……………………27

Hình 2.10: Sơ đồ tuần tự chức năng Thêm bài hát mới vào datatabase ………………………27

Hình 3.1: Bộ dữ liệu train\_triplets…………………………………………………………...37

Hình 3.2: Bộ dữ liệu unique\_tracks………………………………………………………….38

Hình 3.3: Bộ dữ liệu sau khi gộp train\_triplets và unique\_tracks……………………………39

Hình 3.4: Biểu đồ lượng lượt nghe trên toàn bộ tập dữ liệu…………………………………39

Hình 4.1: Màn hình đăng nhập………………………………………………………………47

Hình 4.2: Màn hình chính……………………………………………………………………47

Hình 4.3: Màn hình nhập nếu nhập thông tin sai…………………………………………….48

Hình 4.4: Màn hình chọn tính năng tại trang chủ……………………………………………..48

Hình 4.5: Màn hình danh sách các bài hát sau khi thêm……………………………………..49

Hình 4.6: Màn hình chọn tính năng tại trang chủ……………………………………………..49

Hình 4.7: Màn hình danh sách các bài hát tự tạo…………………………………………….50

Hình 4.8: Màn hình tìm kiếm………………………………………………………………...50

Hình 4.9:Màn hình hiển thị kết quả tìm kiếm………………………………………………..51

Hình 4.10: Màn hình chính có nút vào profile……………………………………………….51

Hình 4.11: Màn hình trang cá nhân…………………………………………………………..52

Hình 4.12: Hình ảnh cấu trúc của file excel………………………………………………….52

Hình 4.13: Màn hình thành công…………………………………………………………….52

Hình 4.14: Màn hình thông báo không có quyền…………………………………………….53

# CHƯƠNG MỞ ĐẦU

## Mục tiêu

Hiện nay, với sự phát triển mạnh mẽ của các nền tảng phát nhạc trực tuyến, nhu cầu nghe nhạc theo sở thích cá nhân của người dùng ngày càng gia tăng. Tuy nhiên, với số lượng bài hát khổng lồ, việc tìm kiếm và khám phá những bài hát mới phù hợp với sở thích cá nhân trở nên khá phức tạp và tốn thời gian. Đối với những người yêu thích âm nhạc, việc lựa chọn được những bài hát đúng gu âm nhạc của mình không phải lúc nào cũng dễ dàng.

Một hệ thống khuyến nghị bài hát hiệu quả có thể giúp người dùng dễ dàng tiếp cận với những bài hát phù hợp với sở thích cá nhân của họ, tạo ra trải nghiệm nghe nhạc thú vị và tối ưu hóa thời gian tìm kiếm. Tuy nhiên, việc xây dựng một hệ thống khuyến nghị chính xác và hiệu quả đòi hỏi sự phân tích sâu rộng về dữ liệu âm nhạc cũng như hiểu rõ về sở thích và hành vi của người dùng.

Vì vậy, khóa luận này sẽ tập trung vào việc xây dựng một hệ thống khuyến nghị bài hát dựa trên sở thích cá nhân của người dùng. Hệ thống sẽ sử dụng các thuật toán học máy và khai phá dữ liệu để phân tích và nhận diện sở thích âm nhạc của người dùng, từ đó đưa ra các gợi ý bài hát phù hợp. Bằng cách kết hợp các kỹ thuật phân tích dữ liệu hiện đại và công nghệ học máy tiên tiến, hệ thống không chỉ giúp người dùng tìm kiếm được những bài hát yêu thích một cách nhanh chóng mà còn khám phá những bài hát mới mà họ có thể chưa từng nghe đến nhưng phù hợp với gu âm nhạc của họ.

## Động lực bản thân

Từ các nhu cầu thực tế hiện nay, việc áp dụng công nghệ vào lĩnh vực giải trí, đặc biệt là âm nhạc, có thể mang lại nhiều giá trị không chỉ trong việc thỏa mãn nhu cầu cá nhân mà còn trong việc nâng cao trải nghiệm người dùng. Bên cạnh đó, trí tuệ nhân tạo và học máy đang ngày càng phát triển mạnh mẽ, với mong muốn áp dụng các mô hình tiên tiến này vào lĩnh vực âm nhạc, tôi quyết định chọn đề tài "Xây dựng hệ thống khuyến nghị bài hát theo sở thích".

Việc thực hiện đề tài này là cơ hội để tôi tự đánh giá năng lực và phát triển bản thân trong lĩnh vực lập trình và trí tuệ nhân tạo. Làm việc với dữ liệu âm nhạc thực tế và áp dụng các thuật toán học máy sẽ giúp tôi học hỏi thêm nhiều kiến thức quý báu và phát triển kỹ năng phân tích dữ liệu và lập trình.

Tôi hy vọng hệ thống khuyến nghị bài hát không chỉ đơn thuần là một công cụ hữu ích trong việc cá nhân hóa trải nghiệm nghe nhạc mà còn giúp người dùng khám phá thêm nhiều thể loại âm nhạc mới mẻ, phong phú. Bên cạnh đó, việc kết nối người dùng thông qua âm nhạc cũng góp phần xây dựng một cộng đồng chia sẻ đam mê âm nhạc, tạo điều kiện cho việc trao đổi kiến thức và kinh nghiệm, thúc đẩy sự sáng tạo và nâng cao nhận thức về giá trị văn hóa âm nhạc trong cuộc sống.

## Phạm vi và phương pháp nghiên cứu

* **Thời gian nghiên cứu:** 6 tháng
* **Phương pháp nghiên cứu:** Khóa luận sử dụng ngôn ngữ Python để xử lý dữ liệu và xây dựng mô hình kết hợp với JavaScript để xây dựng trang web.

## Cấu trúc khóa luận

Khóa luận gồm các chương với nội dung như sau:

* **Lời mở đầu:**

Khóa luận sẽ giới thiệu về phần mục tiêu, động lực của bản thân, phạm vi đề tài và cấu trúc khóa luận.

* **Chương 1:** Cơ sở lý thuyết

Khoá luận giới thiệu về lý thuyết chung và những công nghệ đã sử dụng.

* **Chương 2:** Đặc tả và thiết kế hệ thống

Khoá luận đi sâu vào việc mô tả bài toán, về yêu cầu nghiệp vụ, kiến trúc tổng thể và mô tả chi tiết từng chức năng của hệ thống.

* **Chương 3:** Thu thập dữ liệu và xây dựng mô hình

Khóa luận mô tả các bước thu thập và xử lý dữ liệu. Thực hiện xây dựng mô hình và đánh giá kết quả.

* **Chương 4:** Cài đặt và triển khai hệ thống

Khoá luận mô tả môi trường cài đặt và biểu diễn các chức năng được phát triển trong hệ thống.

* **Kết luận**

Khoá luận tổng kết các kiến thức, kinh nghiệm thu được trong quá trình thực hiện đề tài và các hướng phát triển trong tương lai.

* **Tài liệu tham khảo:**

Khoá luận sẽ liệt kê ra những tài liệu đã sử dụng và đã có trích dẫn trong bài làm.

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## 1.1 Tổng quan về trí tuệ nhân tạo [1]

### 1.1.1. Khái niệm

Trí tuệ nhân tạo (AI) là một lĩnh vực của khoa học máy tính tập trung vào việc tạo ra các hệ thống và công nghệ có khả năng thực hiện những nhiệm vụ mà thông thường đòi hỏi trí tuệ của con người. Các nhiệm vụ này bao gồm học hỏi, suy luận, nhận thức, xử lý ngôn ngữ, và ra quyết định. Mục tiêu chính của AI là phát triển các thuật toán và mô hình để máy tính có thể tự động hóa và cải thiện hiệu suất thực hiện các nhiệm vụ phức tạp.

### 1.1.2. Các công nghệ phổ biến trong trí tuệ nhân tạo

Trí tuệ nhân tạo bao gồm nhiều công nghệ và phương pháp khác nhau. Dưới đây là một số công nghệ phổ biến trong AI:

* **Học máy (Machine Learning**): Đây là công nghệ cốt lõi của AI, tập trung vào việc phát triển các thuật toán cho phép máy tính học từ dữ liệu và cải thiện theo thời gian. Học máy bao gồm các thuật toán như hồi quy tuyến tính, cây quyết định, rừng ngẫu nhiên, và mạng nơ-ron nhân tạo.
* **Học sâu (Deep Learning):** Một phân nhánh của học máy, học sâu sử dụng các mạng nơ-ron sâu với nhiều lớp để xử lý và phân tích dữ liệu phức tạp. Học sâu được ứng dụng rộng rãi trong nhận diện hình ảnh, nhận dạng giọng nói, và xử lý ngôn ngữ tự nhiên.
* **Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP):** Công nghệ này giúp máy tính hiểu và tương tác với ngôn ngữ con người, cho phép thực hiện các nhiệm vụ như dịch ngôn ngữ, phân tích cảm xúc, và tạo văn bản tự động.
* **Thị giác máy tính (Computer Vision):** Tập trung vào việc giúp máy tính nhận diện và hiểu các thông tin hình ảnh và video. Công nghệ này được sử dụng trong nhận diện khuôn mặt, xe tự lái, và chẩn đoán y khoa.
* **Học tăng cường (Reinforcement Learning):** Một phương pháp học tập dựa trên phần thưởng và hình phạt, giúp máy tính học cách hành động trong môi trường để tối đa hóa một số khía cạnh của hiệu suất.

### 1.1.3. Các ứng dụng AI phổ biến

Trí tuệ nhân tạo đã được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Dưới đây là một số ứng dụng phổ biến của AI:

* **Hệ thống khuyến nghị (Recommendation Systems):** Sử dụng học máy để gợi ý các sản phẩm, dịch vụ hoặc nội dung cho người dùng dựa trên sở thích và hành vi của họ. Ví dụ như gợi ý bài hát trên các nền tảng âm nhạc (Spotify, Apple Music), gợi ý phim trên các dịch vụ streaming (Netflix), hay gợi ý sản phẩm trên các trang thương mại điện tử (Amazon).
* **Nhận diện hình ảnh và thị giác máy tính (Image Recognition and Computer Vision)**: Dùng để nhận diện khuôn mặt, đối tượng, hoặc cảnh trong hình ảnh và video. Ứng dụng trong an ninh, xe tự lái, và chẩn đoán y khoa.
* **Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP):** Giúp máy tính hiểu và tương tác với ngôn ngữ con người. Ứng dụng trong chatbot, dịch máy, và phân tích cảm xúc.
* **Phát hiện gian lận (Fraud Detection):** Sử dụng trong các hệ thống tài chính và thương mại điện tử để phát hiện và ngăn chặn các hành vi gian lận dựa trên phân tích dữ liệu giao dịch.
* **Robot tự hành (Autonomous Robots):** Sử dụng AI để phát triển các robot có khả năng tự hành và thực hiện các tác vụ mà không cần sự can thiệp của con người. Ứng dụng trong sản xuất, chăm sóc sức khỏe, và dịch vụ.

## Hệ thống khuyến nghị [2]

Hệ thống khuyến nghị (Recommendation Systems) là một lĩnh vực quan trọng trong trí tuệ nhân tạo và học máy, với mục tiêu chính là đề xuất các sản phẩm, dịch vụ hoặc nội dung phù hợp với sở thích và nhu cầu của người dùng. Các hệ thống này được ứng dụng rộng rãi trong nhiều lĩnh vực như: thương mại điện tử, xem phim, nghe nhạc, mạng xã hội và nhiều lĩnh vực khác.

**Nguyên lý hoạt động:** Hệ thống khuyến nghị hoạt động bằng cách phân tích dữ liệu lịch sử về hành vi người dùng, từ đó dự đoán và gợi ý những mục có thể gây hứng thú cho họ. Có ba phương pháp chính được sử dụng trong các hệ thống khuyến nghị:

* **Lọc theo nội dung (Content-based Filtering):** Dựa trên các đặc điểm của mục và sở thích trước đây của người dùng. Hệ thống sẽ đề xuất các mục có đặc điểm tương tự với những mục mà người dùng đã thích hoặc tương tác.
* **Lọc cộng tác (Collaborative Filtering):** 
  + Dựa trên người dùng (User-based): Đề xuất các mục dựa trên sự tương đồng giữa người dùng hiện tại và những người dùng khác có hành vi tương tự.
  + Dựa trên mục (Item-based): Đề xuất các mục dựa trên sự tương đồng giữa các mục mà người dùng đã thích hoặc tương tác.
* **Phương pháp lai (Hybrid Methods):** Kết hợp cả lọc cộng tác và lọc theo nội dung để tận dụng ưu điểm của cả hai phương pháp và cải thiện độ chính xác của khuyến nghị.

## Phương pháp lọc dựa trên nội dung (Content-based Filtering) [3]

Phương pháp Content-Based (dựa trên nội dung) là một trong những kỹ thuật phổ biến được sử dụng trong hệ thống khuyến nghị, đặc biệt trong lĩnh vực gợi ý bài hát. Phương pháp này dựa trên phân tích và so sánh các thuộc tính của đối tượng để đưa ra gợi ý phù hợp với sở thích của người dùng.

### Nguyên lý hoạt động

Phương pháp Content-Based sử dụng thông tin về các đặc điểm, thuộc tính của bài hát như thể loại, nhịp điệu, tempo, lời bài hát, nhạc cụ sử dụng, và các thuộc tính âm nhạc khác. Những thuộc tính này được trích xuất và lưu trữ trong một cơ sở dữ liệu để so sánh và tìm ra các bài hát có đặc điểm tương tự.

### Các bước chính trong phương pháp Content-Based:

1. ***Trích xuất thuộc tính***: Mỗi bài hát được phân tích để trích xuất các thuộc tính đặc trưng. Ví dụ, một bài hát có thể được phân tích để xác định thể loại (rock, pop, jazz), nhịp điệu (nhanh, chậm), cảm xúc (vui vẻ, buồn bã), và các yếu tố âm nhạc khác. Để trích xuất thuộc tính từ lời bài hát, các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) được áp dụng, trong đó tf-idf (term frequency-inverse document frequency) là một phương pháp phổ biến.
2. ***Xây dựng hồ sơ người dùng***: Dựa trên lịch sử nghe nhạc của người dùng, hệ thống sẽ xây dựng một hồ sơ người dùng chứa các thuộc tính của những bài hát mà người dùng đã thích hoặc thường xuyên nghe. Hồ sơ này thể hiện sở thích âm nhạc của người dùng.
3. ***So sánh và gợi ý****:* Hệ thống so sánh các thuộc tính của bài hát mới với hồ sơ người dùng để tìm ra các bài hát có đặc điểm tương tự. Các bài hát có thuộc tính gần giống với những bài hát mà người dùng đã thích sẽ được gợi ý.

### Ứng dụng của tf-idf trong Content-Based

Tf-idf (term frequency-inverse document frequency) là một kỹ thuật thống kê được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản so với toàn bộ tập hợp các văn bản. Trong ngữ cảnh hệ thống gợi ý bài hát, tf-idf có thể được áp dụng để trích xuất và đánh giá các từ khóa quan trọng trong lời bài hát.

1. ***Công thức tf-idf:***

* Tf (term frequency): Tần suất xuất hiện của một từ trong một văn bản. Tf được tính bằng số lần xuất hiện của từ đó chia cho tổng số từ trong văn bản.
* Idf (inverse document frequency): Tần suất nghịch đảo tài liệu, được tính bằng logarit của tổng số tài liệu chia cho số tài liệu có chứa từ đó.

Công thức idf là:

Trong đó, là tổng số tài liệu và ​ là số tài liệu chứa từ t.

Tf-idf của một từ trong một văn bản được tính bằng cách nhân tf và idf của từ đó:

1. ***Áp dụng tf-idf trong Content-Based:***

* Trích xuất từ khóa: Lời bài hát được phân tích để trích xuất các từ khóa quan trọng dựa trên giá trị tf-idf. Những từ khóa có giá trị tf-idf cao được coi là quan trọng và đại diện cho nội dung của bài hát.
* Xây dựng vector thuộc tính: Mỗi bài hát được biểu diễn bằng một vector các từ khóa với giá trị tf-idf tương ứng. Các vector này được sử dụng để so sánh và tìm kiếm các bài hát tương tự.
* So sánh và gợi ý: Hệ thống tính toán độ tương đồng giữa vector thuộc tính của bài hát mới và các vector thuộc tính trong hồ sơ người dùng. Các bài hát có vector thuộc tính gần nhất với những bài hát trong hồ sơ người dùng sẽ được gợi ý.

### Ưu điểm của phương pháp Content-Based:

* ***Không phụ thuộc vào dữ liệu người dùng khác***: Phương pháp này không cần có dữ liệu từ những người dùng khác để hoạt động hiệu quả. Điều này rất hữu ích trong trường hợp người dùng mới không có lịch sử nghe nhạc phong phú.
* ***Gợi ý dựa trên đặc điểm rõ ràng***: Vì phương pháp dựa trên phân tích các thuộc tính cụ thể của bài hát, người dùng có thể nhận được những gợi ý sát với sở thích âm nhạc cá nhân.

### Hạn chế của phương pháp Content-Based:

* ***Khó khăn trong việc trích xuất thuộc tính phức tạp***: Một số thuộc tính âm nhạc phức tạp có thể khó được trích xuất và biểu diễn một cách chính xác.
* ***Khả năng khám phá hạn chế***: Hệ thống có thể chỉ gợi ý các bài hát tương tự với những gì người dùng đã nghe trước đó, hạn chế khả năng khám phá những thể loại nhạc hoặc bài hát mới lạ.

## Phương pháp lọc cộng tác (Collaborative Filtering) [4]

Phương pháp Collaborative Filtering (lọc cộng tác) là một trong những kỹ thuật phổ biến nhất được sử dụng trong hệ thống gợi ý. Khác với phương pháp Content-Based, Collaborative Filtering dựa vào phân tích hành vi và sở thích của nhiều người dùng để đưa ra gợi ý phù hợp. Phương pháp này giả định rằng những người có sở thích giống nhau trong quá khứ sẽ có những sở thích tương tự trong tương lai.

### Nguyên lý hoạt động

Collaborative Filtering hoạt động bằng cách xây dựng một ma trận người dùng - sản phẩm (user-item matrix) từ dữ liệu người dùng. Ma trận này chứa các đánh giá hoặc tương tác của người dùng với các sản phẩm (trong trường hợp này là các bài hát). Từ ma trận này, hệ thống có thể tìm ra các mẫu (pattern) trong hành vi của người dùng để dự đoán sở thích và đưa ra các gợi ý.

### Ma trận người dùng - sản phẩm (User-Item Matrix):

Ma trận người dùng - sản phẩm là một bảng dữ liệu trong đó hàng đại diện cho người dùng và cột đại diện cho sản phẩm (bài hát). Mỗi ô trong ma trận chứa giá trị thể hiện mức độ tương tác của người dùng với sản phẩm, chẳng hạn như số lần nghe, đánh giá hoặc điểm số.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Song\_1 | Song\_2 | Song\_3 | Song\_4 |
| User\_A | 5 | ? | 3 | 1 |
| User\_B | 4 | 2 | ? | 5 |
| User\_C | 3 | 1 | 2 | ? |
| User\_D | ? | 5 | 4 | 2 |

*Bảng 1.1: Ví dụ về ma trận người dùng – sản phẩm (User-Item Matrix)*

Trong User-Item Matrix ví dụ trong bảng 1.1, dấu "?" đại diện cho những giá trị chưa biết mà hệ thống cần dự đoán. Dựa trên các giá trị đã biết, hệ thống sẽ ước lượng các giá trị chưa biết và đưa ra gợi ý.

### Các loại Collaborative Filtering

Collaborative Filtering có thể được chia thành hai loại chính: User-Based Collaborative Filtering và Item-Based Collaborative Filtering. Mỗi loại có nguyên lý hoạt động và phương pháp tính toán độ tương đồng khác nhau.

### User-Based Collaborative Filtering

Phương pháp này tập trung vào sự tương đồng giữa người dùng. Hệ thống sẽ tìm những người dùng có lịch sử nghe nhạc giống với người dùng hiện tại và sử dụng thông tin này để gợi ý những bài hát mà họ đã thích nhưng người dùng hiện tại chưa nghe.

Các bước chính:

1. ***Tính toán độ tương đồng giữa các người dùng*:** Sử dụng ma trận người dùng - sản phẩm để tính toán độ tương đồng giữa người dùng hiện tại và những người dùng khác. Một số phương pháp phổ biến để tính độ tương đồng bao gồm Độ tương đồng cosine (Cosine Similarity), Hệ số tương quan Pearson (Pearson Correlation Coefficient), và Khoảng cách Euclidean (Euclidean Distance).
2. ***Xây dựng hồ sơ người dùng***: Dựa trên lịch sử nghe nhạc của người dùng, hệ thống sẽ xây dựng một hồ sơ người dùng chứa các thuộc tính của những bài hát mà người dùng đã thích hoặc thường xuyên nghe. Hồ sơ này thể hiện sở thích âm nhạc của người dùng.
3. ***So sánh và gợi ý****:* Hệ thống so sánh các thuộc tính của bài hát mới với hồ sơ người dùng để tìm ra các bài hát có đặc điểm tương tự. Các bài hát có thuộc tính gần giống với những bài hát mà người dùng đã thích sẽ được gợi ý.

### Item-Based Collaborative Filtering

Phương pháp này tập trung vào sự tương đồng giữa các sản phẩm. Hệ thống sẽ tìm các bài hát có lịch sử được người dùng đánh giá tương tự và sử dụng thông tin này để gợi ý bài hát cho người dùng.

Các bước chính:

1. ***Tính toán độ tương đồng giữa các bài hát:*** Sử dụng ma trận người dùng - sản phẩm để tính toán độ tương đồng giữa các bài hát. Một số phương pháp phổ biến để tính độ tương đồng giữa các bài hát bao gồm Độ tương đồng cosine (Cosine Similarity) và Hệ số tương quan Pearson (Pearson Correlation Coefficient).
2. ***Lựa chọn những bài hát có độ tương đồng cao nhất:*** Từ các độ tương đồng đã tính toán, chọn ra những bài hát có độ tương đồng cao nhất với những bài hát mà người dùng hiện tại đã nghe và thích.
3. ***Gợi ý những bài hát tương đồng này cho người dùng hiện tại:*** Sử dụng thông tin về các bài hát tương đồng để đưa ra gợi ý cho người dùng hiện tại.

### Ưu điểm của phương pháp Collaborative Filtering

* **Khả năng cá nhân hóa cao:** Collaborative Filtering dựa trên hành vi và sở thích thực tế của người dùng, do đó, hệ thống có khả năng đưa ra các gợi ý rất phù hợp và cá nhân hóa cao.
* **Không yêu cầu dữ liệu về nội dung sản phẩm:** Phương pháp này không yêu cầu thông tin chi tiết về sản phẩm mà chỉ cần dữ liệu về tương tác giữa người dùng và sản phẩm. Điều này rất hữu ích khi dữ liệu nội dung của sản phẩm khó thu thập hoặc không đầy đủ.
* **Khả năng mở rộng:** Collaborative Filtering có thể dễ dàng mở rộng để gợi ý cho các sản phẩm mới mà không cần phải cập nhật lại toàn bộ hệ thống. Chỉ cần có đủ dữ liệu tương tác mới giữa người dùng và sản phẩm, hệ thống có thể tiếp tục hoạt động hiệu quả.
* **Khả năng khám phá:** Phương pháp này giúp người dùng khám phá ra những sản phẩm mới mà họ có thể chưa từng nghĩ đến, dựa trên hành vi của những người dùng khác có sở thích tương tự.

### Hạn chế của phương pháp Collaborative Filtering

* **Vấn đề Cold Start:** Collaborative Filtering gặp khó khăn khi có ít dữ liệu hoặc dữ liệu không đầy đủ, đặc biệt là với người dùng mới (cold start problem). Khi không có đủ thông tin về sở thích hoặc hành vi của người dùng mới, hệ thống khó đưa ra gợi ý chính xác.
* **Tính toán phức tạp:** Tính toán độ tương đồng giữa người dùng hoặc sản phẩm đòi hỏi tài nguyên tính toán lớn, đặc biệt là khi số lượng người dùng và sản phẩm rất lớn. Điều này có thể dẫn đến vấn đề về hiệu suất và chi phí tính toán cao.
* **Vấn đề dữ liệu thưa thớt:** Ma trận người dùng - sản phẩm thường rất thưa thớt (sparse), tức là có rất nhiều ô trống. Điều này làm cho việc tính toán và dự đoán trở nên khó khăn hơn, dẫn đến độ chính xác của hệ thống có thể bị ảnh hưởng.
* **Thiên vị xu hướng chính:** Hệ thống có thể thiên vị những sản phẩm hoặc người dùng phổ biến, làm giảm tính đa dạng của các gợi ý. Những sản phẩm ít phổ biến có thể ít được gợi ý hơn mặc dù chúng có thể phù hợp với sở thích của một số người dùng.
* **Vấn đề bảo mật và riêng tư:** Việc thu thập và lưu trữ dữ liệu về hành vi và sở thích của người dùng có thể gây ra vấn đề về bảo mật và riêng tư. Người dùng có thể lo ngại về việc dữ liệu cá nhân của họ bị lạm dụng hoặc truy cập trái phép.

## Ngôn ngữ sử dụng

### JavaScript [5]

JavaScript là một ngôn ngữ lập trình đa năng được sử dụng rộng rãi trên toàn thế giới, đặc biệt là trong phát triển web. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web, không chỉ vậy JS còn có thể làm nhiều nhiệm vụ khác bên phía server. Nó được phát triển bởi Brendan Eich vào năm 1995 khi ông làm việc tại Netscape Communications Corporation. JavaScript thường được viết tắt là JS và không nên nhầm lẫn với Java, một ngôn ngữ lập trình khác.

**Ưu điểm:**

Đa nền tảng: JavaScript có thể chạy trên nhiều nền tảng khác nhau, bao gồm trình duyệt web, máy chủ (với Node.js), và các ứng dụng di động và máy tính để bàn. Điều này làm cho JavaScript trở thành một lựa chọn linh hoạt cho nhiều loại dự án.

Thực thi phía khách: JavaScript chủ yếu được thực thi phía khách (client-side), tức là mã JavaScript sẽ được tải về và có thể chạy ngay trực tiếp trên trình duyệt của người dùng. Điều này giúp giảm tải cho máy chủ và cải thiện tốc độ và hiệu suất của trang web và cải thiện tối đa trải nghiệm người dùng.

Tương tác và động: JavaScript đặc biệt mạnh ở khả năng tạo nên một giao diện web động cho phép tương tác, tạo ra các sự kiện và các hiệu ứng, cung cấp trải nghiệm người dùng tốt hơn bằng cách xử lý và tạo ra các hiệu ứng như nhấp chuột, nhập liệu từ bàn phím, và các thao tác kéo-thả.

Khả năng mở rộng: Với sự hỗ trợ của các công cụ như Node.js, JavaScript có thể được sử dụng để phát triển các ứng dụng web phía máy chủ (server-side) có khả năng mở rộng, xử lý nhiều yêu cầu đồng thời mà không gặp vấn đề về hiệu suất.

Tích hợp dễ dàng với các ngôn ngữ và công nghệ khác: JavaScript có khả năng tích hợp tốt với các ngôn ngữ và công nghệ khác như HTML, CSS, PHP, Python, và nhiều ngôn ngữ lập trình khác, giúp tạo ra các ứng dụng web mạnh mẽ và phức tạp. Nhờ vào những ưu điểm này, JavaScript đã trở thành một phần không thể thiếu và đứng đầu trong lĩnh trong phát triển web về cả mặt giao diện lẫn lẫn trên phía server hiện đại và ngày càng được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau.

**Nhược điểm:**

* JavaScript chạy trên trình duyệt của người dùng, điều này dễ dẫn tới các vấn đề liên quan đến bảo mật, an toàn thông tin nếu mã JavaScript không được viết cẩn thận. Các vấn đề như XSS (Cross-Site Scripting) và CSRF (Cross-Site Request Forgery) có thể xảy ra nếu không có biện pháp bảo vệ thích hợp.
* Thiếu hỗ trợ cho lập trình song song: JavaScript không hỗ trợ lập trình song song một cách tự nhiên mà thông qua cơ chế bất đồng bộ, điều này có thể làm hạn chế hiệu suất của các ứng dụng yêu cầu xử lý đồng thời nhiều tác vụ. Mặc dù Web Workers có thể được sử dụng để thực hiện một số tác vụ song song, nhưng chúng có giới hạn và phức tạp hơn trong việc sử dụng.

### Python [6]

Python là một ngôn ngữ lập trình động không định kiểu được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học. Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học không có định kiểu phức tạp cần ràng buộc trong thời gian biên dịch và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

Python hữu ích trong việc lập trình mã phía máy chủ bởi vì ngôn ngữ này cung cấp nhiều thư viện bao gồm mã viết sẵn cho các hàm backend phức tạp, giúp tốc độc ra sản phẩm nhanh chóng. Các nhà phát triển cũng sử dụng một loạt các khung Python cung cấp tất cả những công cụ cần thiết để xây dựng ứng dụng web một cách nhanh chóng, không cần phải làm lại từ đầu và dễ dàng trong quá trình đưa sản phẩm ra thực tế hơn.

**Ưu điểm:**

* Cú pháp cơ bản giống tiếng Anh.
* Cải thiện năng suất làm việc vì các nhà phát triển sử dụng ít dòng mã hơn để viết một chương trình Python.
* Thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ.
* Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
* Có rất nhiều tài nguyên hữu ích trên internet
* Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau như Windows, macOS, Linux và Unix.

**Nhược điểm:**

* Tốc độ thực thi chậm, hiệu suất thấp so với ngôn ngữ biên dịch.
* Tiêu thụ bộ nhớ lớn
* Không an toàn cao và có nguy cơ bảo mật, có một số hạn chế khi sử dụng Python để truy cập cơ sở dữ liệu

## Các framework sử dụng

### FastAPI [7]

* FastAPI là một framework hiện đại, nhanh và hiệu quả cho việc xây dựng các ứng dụng API bằng Python 3.7+ dựa trên các tiêu chuẩn Python type hints và OpenAPI. Thiết kế của nó được xây dựa tập trung trên một thư viện tiêu chuẩn là Pydantic chuyên dùng để gợi ý kiểu để xác minh, tuần tự hoá và giải tuần tự hoá. Nó được thiết kế để mang lại hiệu suất cao, tăng tốc độ phát triển sản phẩm và dễ dàng sử dụng, cả trong việc phát triển và triển khai các API web.

Một số điểm nổi bật và đặc điểm chính của FastAPI:

* Hiệu suất cao: FastAPI được xây dựng trên nền của Starlette cho web framework core và thư viện tiêu chuẩn Pydantic để xử lý dữ liệu. Kết hợp này cho phép nó hiệu suất tương đương với NodeJS và Go, đồng thời vượt trội hơn các framework ra đời trước đó cùng thuộc ngôn ngữ python là Flask và Django nhờ vào tính năng bất đồng bộ.
* Hỗ trợ sử dụng “tiêm phụ thuộc” (Dependency Injection): FastAPI hỗ trợ design pattern này cho lập trình viên một cách dễ dàng. Tăng tính mô đun cho mã nguồn, đạt được khả năng IOC ( inversion of control).
* Tự động tạo tài liệu: FastAPI tự động tạo ra tài liệu mà người dùng không phải cài đặt thêm gì cả. Tài liệu này tương thích với OpenAPI (trước đây là Swagger) và JSON Schema. Điều này cho phép người dùng thao tác với các API thông qua giao diện đồ hoạ là Swagger UI mà ko cần thiết phải cài các phần mềm tương tác như postman.
* Type hints: FastAPI cho phép sử dụng các gợi ý kiểu tiêu chuẩn (type hint) của Python để xác định và kiểm tra các đầu vào và đầu ra của API. Điều này giúp mã nguồn dễ đọc hơn và giảm thiểu lỗi.
* Tự động xử lý lỗi: FastAPI tự động kiểm tra và xử lý các lỗi đầu vào dựa trên gợi ý kiểu và các ràng buộc đã định nghĩa trong mà nguồn để giúp phát hiện lỗi tự sớm và trả về thông báo lỗi một cách chi tiết cho người dùng.
* Hỗ trợ Asynchronous: FastAPI hỗ trợ các hàm đồng bộ và không đồng bộ, giúp tận dụng tối đa hiệu suất và khả năng mở rộng. Đặc biệt là trong các tác vụ liên quan đến I/O bound trong xử lý các công việc, tác vụ tốn nhiều thời gian.
* Dễ dàng tích hợp: FastAPI dễ dàng tích hợp với các thư viện và công cụ phổ biến khác như SQLAlchemy, MongoDB, Redis và đặc biệt tích hợp sẵn các hệ thống xác thực hàng đầu hiện nay như OAuth2 và JWT. Trong tài liệu chính thức của FastAPI cũng giới thiệu, hỗ trợ viết mã một cách chi tiết và giải thích đầy đủ cách tích hợp với các phương thức xác thực và uỷ quyền (OAuth2, JWT) cái mà các công ty công nghệ hàng đầu đang sử dụng.
* Cộng đồng và tài liệu: FastAPI có một cộng đồng phát triển mạnh mẽ và tài liệu chính thức chi tiết của nhà phát triển giúp dễ dàng tìm kiếm hỗ trợ và tài nguyên làm việc

### VueJS [8]

Vue.js là một framework JavaScript mạnh mẽ và linh hoạt để xây dựng giao diện người dùng (UI). Nó được tạo ra bởi Evan You và phát hành lần đầu vào năm 2014. Vue.js đã nhanh chóng trở thành một trong những framework phổ biến nhất trong cộng đồng phát triển web nhờ vào sự đơn giản và hiệu quả của nó. Nó thiết kế theo hướng cho phép và khuyến khích phát triển ứng dụng theo từng bước. VueJs đáp ứng được dễ dàng nhu cầu xây dựng những ứng dụng đơn trang và có độ phức tạp cao.

* Dễ học và dễ sử dụng: Vue.js có cú pháp đơn giản và dễ hiểu, giúp các nhà phát triển mới bắt đầu dễ dàng tiếp cận và sử dụng. Đồng thời tài liệu dễ tiếp cận, chi tiết giúp người học giải quyết vấn đề nhanh chóng.
* Reactive Data Binding: Vue.js sử dụng hệ thống binding dữ liệu reactivity, cho phép tự động cập nhật giao diện người dùng khi dữ liệu thay đổi mà không cần phải can thiệp trực tiếp.
* Component-Based Architecture: Vue.js cho phép xây dựng ứng dụng bằng cách chia nhỏ thành các component độc lập, dễ quản lý và tái sử dụng. Mỗi một component có thể chứa HTML, CSS và JavaScript giúp phát triển và gỡ lỗi mã nguồn trở nên dễ dàng hơn.
* Virtual DOM: Vue.js sử dụng Virtual DOM để tối ưu hiệu suất. Điều này giúp giảm thiểu số lần thao tác trực tiếp với DOM thật, tăng tốc độ render và cập nhật giao diện.
* Directives: Vue cung cấp các chỉ thị mạnh mẽ, đơn giản và dễ hiểu như v-if, v-for, v-bind, và v-model để tăng khả năng linh hoạt của UI, đồng thời cải thiện trải nghiệm người dùng.
* Tích hợp dễ dàng: VueJs có thể tích hợp dễ dàng vào các dự án hiện có hoặc sử dụng cùng với các thư viện khác mà không gặp nhiều vấn đề.

## Tích hợp các nền tảng khác

### S3 storage [9]

Amazon Simple Storage Service (Amazon S3) là một dịch vụ lưu trữ kiểu đối tượng (object storage) cung cấp bởi Amazon Web Services (AWS). Nó được thiết kế để lưu trữ và truy xuất bất kỳ lượng dữ liệu từ bất kỳ đâu trên web. S3 là một trong những dịch vụ đầu tiên và phổ biến nhất của AWS, và nó cung cấp một hệ thống lưu trữ đáng tin cậy, bảo mật và có khả năng mở rộng cao.

S3 storage có rất nhiều điểm mạnh khi để lưu trữ nhiều kiểu dữ liệu khác nhau sau đây là một số điểm chính:

* Khả năng mở rộng không giới hạn: S3 được thiết kế để có thể xử lý bất kỳ lượng dữ liệu nào, từ vài GB đến hàng petabyte. Điều này có nghĩa là bạn không phải lo lắng về hệ thống lưu trữ và có tính mở rộng nhanh bất cứ lúc nào khi nhu cầu tăng lên.
* Độ bền và sẵn sàng cao: S3 cam kết cung cấp độ bền dữ liệu lên đến 99.999999999% (11 số chín) cho các đối tượng được lưu trữ. Điều này đạt được thông qua việc sao chép dữ liệu trên nhiều khu vực (tính HA) và cơ sở dữ liệu trong hệ thống của AWS. S3 cũng cung cấp độ sẵn sàng dịch vụ 99.99%, đảm bảo dữ liệu luôn có thể truy cập được khi cần.
* Quản lý dữ liệu: S3 cung cấp nhiều tính năng quản lý dữ liệu như versioning (quản lý phiên bản), lifecycle policies (chính sách vòng đời) để tự động di chuyển hoặc xóa dữ liệu không cần thiết, và cross-region replication (sao chép dữ liệu qua nhiều khu vực) để tăng cường độ bền và khả năng phục hồi dữ liệu..
* Tính nhất quán: Amazon S3 cung cấp tính nhất quán cho các hoạt động ghi mới và đọc ngay sau ghi cho các hoạt động ghi chồng (PUTS of new objects), giúp đảm bảo rằng dữ liệu luôn nhất quán và cập nhật.
* Khả năng tích hợp mạnh mẽ: Amazon S3 dễ dàng tích hợp với nhiều dịch vụ khác của AWS như EC2, Lambda, CloudFront, và Athena, cũng như các ứng dụng và công cụ bên thứ ba. Điều này làm cho S3 trở thành một thành phần trung tâm trong hệ sinh thái AWS. Nhờ khả năng tích hợp mạnh mẽ giúp cho S3 có thể làm kho chứa dữ liệu cho nhiều ứng dụng và có thể chuyển đổi dữ liệu sang các hệ thống khác dễ dàng khi có sự cố hay nhu cầu khác.
* Hiệu suất cao: S3 được tối ưu hóa để cung cấp hiệu suất cao cho các yêu cầu đọc và ghi, giúp đáp ứng các ứng dụng đòi hỏi tốc độ truy xuất dữ liệu nhanh chóng.

### MongoDB [10]

MongoDB là một cơ sở dữ liệu NoSQL mã nguồn mở được thiết kế để lưu trữ, quản lý và truy vấn dữ liệu theo cách linh hoạt, hiệu quả không yêu cầu sự chặt chẽ. Nó sử dụng định dạng tài liệu (document) lưu trữ dữ liệu dưới dạng JSON (hoặc BSON – một phiên bản nhị phân của JSON), cho phép lưu trữ dữ liệu có cấu trúc phức tạp mà không cần tuân theo một cấu trúc bảng cứng nhắc cần sự ràng buộc quan hệ của khoá chính với khoá phụ trong các bảng như các cơ sở dữ liệu quan hệ (SQL).

* Tính linh hoạt cao: Dữ liệu trong MongoDB được lưu trữ dưới dạng tài liệu cho phép lưu trữ dữ liệu có cấu trúc phức tạp, mà không cần phải tuân thủ một cấu trúc cứng nhắc, tuân theo ràng buộc về quan hệ giữa các bảng nhất định như trong các cơ sở dữ liệu quan hệ. Điều này rất hữu ích cho các ứng dụng có yêu cầu thay đổi cấu trúc dữ liệu thường xuyên và liên tục, phù hợp khi cấu trúc dữ liệu hay thay đổi, làm mới.
* Khả Năng Mở Rộng Tốt : MongoDB được thiết kế để mở rộng dễ dàng, cả theo chiều ngang (sharding) và chiều dọc nhờ tính không ràng buộc trong quan hệ giữa các bảng hay có thể hiểu là collection. Tính năng sharding cho phép chia nhỏ cơ sở dữ liệu trên nhiều máy chủ cho phép tính toán song song, giúp tăng khả năng xử lý nhanh chóng và lưu trữ dữ đặc biệt là ứng dụng trong dữ liệu lớn.
* Hiệu suất cao: MongoDB sử dụng các chỉ mục (indexes) để cải thiện, tăng tốc hiệu suất truy vấn. Đồng thời nó cũng hỗ trợ các truy vấn phức tạp, tìm kiếm toàn văn, và các thao tác tổng hợp (aggregation) mạnh mẽ.
* Khả Năng Tương Tác Với Dữ Liệu Lớn: Nhờ khả năng mở rộng dễ dàng theo chiều ngang, chiều dọc và tính linh hoạt của nó nên mongoDB có thể thao tác với dữ liệu lớn, loại dữ liệu phức tạp với nhiều chiều mà cơ sở dữ liệu truyền thống với sự liên kết về quan hệ không thể giải quyết dược.
* Khả năng lưu trữ dữ liệu đa dạng: MongoDB hỗ trợ nhiều loại dữ liệu khác nhau như văn bản số, mảng và các đối tượng nhúng, (embedded documents). Điều này cho phép bạn lưu trữ dữ liệu theo cách tự nhiên, linh hoạt mà không cần phải thêm bảng, tạo các bảng trung gian, chuyển đổi và phân tách dữ liệu hay đối tượng.
* Tính Tương Tác Cao: MongoDB hỗ trợ nhiều thư viện và các công cụ hỗ trợ nhiều ngôn ngữ lập trình phổ biến như Python, Node.js, Java.. để có thể dễ dàng tích hợp vào ứng dụng.
* Khả Năng Hỗ Trợ ACID: Từ phiên bản MongoDB 4.0 trở lên hỗ trỡ các giao dịch ACID (Atomicity, Consistency, Isolation, Durability) dựa trên nhiều tài liệu, giúp đảm bảo tính toàn vẹn dữ liệu trong các ứng dụng yêu cầu tính nhất quán cao. Tính chât này giúp môngDB càng ổn định hơn để thay thế các cơ sở dữ liệu thông thường, không chỉ là một cơ sở dữ liệu cho dữ liệu lớn.
* Cộng Đồng và Hệ Sinh Thái Phát Triển: MongoDB có một cộng đồng lớn và hoạt động mạnh mẽ, cung cấp nhiều tài liệu hướng dẫn, diễn đàn hỗ trợ và các công cụ tiện ích MongoDB Atlas, dịch vụ cơ sở dữ liệu đám mây của MongoDB, cũng giúp quản lý và triển khai MongoDB dễ dàng trên các nền tảng đám mây như AWS, Google Cloud, và Azure.

### Microsoft SQL Server[11]

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft. Nó hỗ trợ nhiều tính năng quản lý dữ liệu, từ lưu trữ và truy xuất dữ liệu đến phân tích dữ liệu phức tạp. SQL Server được sử dụng rộng rãi trong các tổ chức lớn và nhỏ để hỗ trợ các ứng dụng doanh nghiệp, trang web và các giải pháp phân tích dữ liệu.

* Hiệu năng cao: SQL Server có khả năng xử lý lượng dữ liệu lớn với tốc độ nhanh chóng nhờ vào các tối ưu hóa hiệu suất như tính năng chỉ mục (indexing), tối ưu hóa truy vấn và cách sử dụng bộ nhớ đệm (caching) để tăng tốc độ truy vấn.
* Khả năng bảo mật mạnh mẽ: SQL Server cung cấp các tính năng bảo mật cao cấp, bao gồm mã hóa dữ liệu, quản lý quyền truy cập dựa trên vai trò (role-based access control), và khả năng kiểm toán (auditing) để theo dõi các hoạt động trong cơ sở dữ liệu.
* Khả năng khả dụng cao: Các tính năng như Always On Availability Groups, Mirroring và Replication giúp đảm bảo tính khả dụng và khả năng phục hồi của cơ sở dữ liệu trong trường hợp xảy ra sự cố.
* Hỗ trợ ACID: SQL Server đảm bảo tính nhất quán và toàn vẹn của dữ liệu thông qua việc hỗ trợ đầy đủ các thuộc tính giao dịch ACID (Atomicity, Consistency, Isolation, Durability).
* Công Cụ Quản Lý và Phân Tích Dữ Liệu: SQL Server cung cấp nhiều công cụ mạnh mẽ như SQL Server Management Studio (SSMS), SQL Server Data Tools (SSDT), và SQL Server Reporting Services (SSRS) để quản lý, phát triển và phân tích dữ

# CHƯƠNG 2.  ĐẶC TẢ VÀ THIẾT KẾ HỆ THÔNG

## 2.1. Mô tả bài toán

Ứng dụng nghe nhạc có tính năng gợi ý nhạc theo sở thích của người dùng , có thể tạo một danh sách nhạc của bản thân mình theo một sở thích tuỳ ý hoặc tìm kiếm một bài hát. Tính năng gợi ý sẽ tăng trải nghiệm người dùng giúp cho khách hàng khám phá ra nhiều bài hát mới.

### 2.1.1. Gợi ý nhạc

Đây là tính năng chính của ứng dụng giúp cho người dùng có thể có được những bài hát theo sở thích của người dùng. Công nghệ dựa trên tương tác của người dùng với nhau sau đó sẽ tính toán sự giống nhau trong sở thích, từ đó đưa ra các kết quả gợi ý nhạc cho người dùng.

### 2.1.2. Tạo một danh sách

Tính năng này cho phép người dùng tạo một danh sách nhạc tự tạo. Từ đó người dùng có thể thêm bất kỳ bài hát nào vào danh sách tự tạo đó. Người dùng cũng có thể đổi tên danh sách, xoá danh sách cũng như có thể xoá bài hát đã thêm vào đó.

### 2.1.3. Tìm kiếm bài hát

Tính năng này cho phép người dùng có thể tìm kiếm một bài hát nào đó trong cơ sở dữ liệu, đương nhiên là có thể phát bài hát đó đồng thời thêm bài hát đó vào danh sách của cá nhân nếu muốn.

## 2.2. Yêu cầu nghiệp vụ

* **BR1. Đăng nhập:** Ứng dụng phải đảm bảo khả năng đăng nhập, đăng xuất, chỉnh sửa thông tin tài khoản của người dùng.
* **BR2. Đăng ký tài khoản:** Ứng dụng phải đảm bảo khả năng đăng ký tài khoản của người dùng. Sau khi đăng ký, người dùng có thể đăng nhập lại vào tài khoản đó và lưu trữ được dữ liệu cũ
* **BR3. Gợi ý bài hát:** Ứng dụng phải đảm bảo khả năng gợi ý bài hát cho người dùng dựa khi biết một số yêu cầu đầu vào
* **BR4. Tìm kiếm bài hát:** Ứng dụng phải đảm bảo khả năng tìm kiếm bài hát theo tên người dùng nhập
* **BR5. Tạo một danh sách cá nhân:** Ứng dụng phải đảm bảo khả năng người dùng có thể tạo một danh sách của cá nhân họ
* **BR6. Xoá một danh sách cá nhân:** Ứng dụng phải đảm bảo khả năng người dùng có thể xoá một danh sách của cá nhân họ
* **BR7. Thêm bài hát mới vào danh sách:** Ứng dụng phải đảm bảo khả năng người dùng có thể thêm nhạc vào danh sách cá nhân của mình
* **BR8. Xem danh sách bài hát trong danh sách:** Ứng dụng phải đảm bảo khả năng xem danh sách bài hát trong một danh sách cá nhân
* **BR9. Xoá bài hát khỏi danh sách:** Ứng dụng phải đảm bảo khả năng xoá một bài hát khỏi danh sách cá nhân
* **BR10. Đổi tên danh sách:** Ứng dụng phải đảm bảo khả năng thay đổi tên một danh sách cá nhân
* **BR11. Phát nhạc:** Ứng dụng phải đảm bảo khả năng phát được cái bài hát hiển thị trên ứng dụng
* **BR12. Hàng đợi nhạc:** Ứng dụng phải đảm bảo khả năng có một hàng đợi để phát nhạc
* **BR13. Thêm nhạc vào cơ sở dữ liệu:** Ứng dụng phải đảm bảo khả năng có thể thêm nhạc vào cơ sở dự liệu và kho nhạc
* **BR14. Xoá nhạc khỏi cơ sở dữ liệu:** Ứng dụng phải đảm bảo khả năng có thể xoá nhạc khỏi cơ sở dự liệu và kho nhạc

## Sơ đồ Usecase các chức năng chính

### Sơ đồ Usecase

*Hình 2.1: Sơ đồ Usecase hệ thống*

### Các Usecase

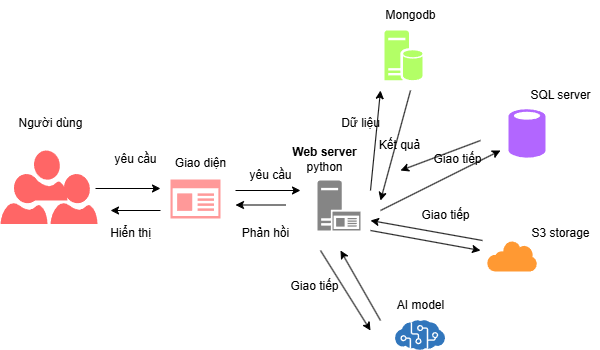
* + **UC#01 - Đăng ký tài khoản**: Hệ thống yêu cầu tất cả các tác nhân tham gia HT phải đăng nhập để có thể thực hiện các chức năng do hệ thống cung cấp. Do vậy, việc đầu tiên cần làm là tạo 1 tài khoản để sử dụng.
  + **UC#02 - Đăng nhập**: Hệ thống yêu cầu tất cả các tác nhân tham gia HT phải đăng nhập để có thể thực hiện các chức năng do hệ thống cung cấp.
  + **UC#03 - Đăng xuất**: Hệ thống cho phép người dùng có thể đăng xuất khỏi hệ thống.
  + **UC#04 – Tạo một danh sách tự tạo**: Hệ thống cho phép người dùng có thể tạo một danh sách cá nhân
  + **UC#05 – Đưa bài hát vào danh sách**: Hệ thống cho phép người dùng có thể đưa bài hát vào danh sách tự tạo
  + **UC#06 – Xoá bài hát vào danh sách**: Hệ thống cho phép người dùng có thể xoá bài hát khỏi danh sách tự tạo
  + **UC#07 – Xoá danh sách tự tạo**: Hệ thống cho phép người dùng có thể xoá một danh sách tự tạo
  + **UC#08 – Đổi tên danh sách tự tạo**: Hệ thống cho phép người dùng có thể đổi tên một danh sách tự tạo
  + **UC#09 - Xem danh sách bài hát trong một danh sách**: Hệ thống cho phép người dùng có thể xem danh sách các bài hát đã thêm vào một danh sách cá nhân
  + **UC#10 – Xem được danh sách các danh sách tự tạo**: Hệ thống cho phép người dùng xem được danh sách các danh sách tự tạo
  + **UC#11 – Gợi ý bài hát**: Hệ thống cho phép người dùng xem được các bài hát được hệ thống gợi ý cho người dùng
  + **UC#12 – Tìm kiếm một bài hát**: Hệ thống cho phép người dùng có thể tìm kiếm một bài hát nằm trong cơ sở dữ liệu
  + **UC#13 – Phát một bài hát**: Hệ thống cho phép người dùng có thể phát một bài hát người dùng chọn
  + **UC#13 – Lấy danh sách bài hát khi nhập id**: Hệ thống cho lấy thông tin về từng bài hát khi nhận vào id
  + **UC#15 – Đi tới bài hát tiếp theo**: Hệ thống cho phép người dùng có thể phát một bài hát tiếp theo trong hàng đợi
  + **UC#16 – Phát một một bài hát trước đó**: Hệ thống cho phép người dùng có thể phát một bài hát trước đó trong hàng đợi
  + **UC#17 – Thêm bài hát vào cơ sở dữ liệu**: Hệ thống cho phép người dùng có quyền admin có thể thêm bài hát vào cơ sở dữ liệu
  + **UC#18 – Xoá bài hát khỏi cơ sở dữ liệu**: Hệ thống cho phép người dùng có quyền admin có thể xoá bài hát khỏi cơ sở dữ liệu

### Bảng ánh xạ yêu cầu nghiệp vụ với các Usecase

|  |  |  |
| --- | --- | --- |
| **BR** | **Mô tả** | **UC** |
| BR1 | Đăng nhập | UC01, UC03 |
| BR2 | Đăng ký tài khoản | UC02 |
| BR3 | Gợi ý bài hát | UC11 |
| BR4 | Tìm kiếm bài hát | UC12 |
| BR5 | Tạo một danh sách cá nhân | UC04 |
| BR6 | Xoá một danh sách cá nhân | UC07 |
| BR7 | Thêm bài hát mới vào danh sách | UC05 |
| BR8 | Xem danh sách bài hát trong danh sách | UC09, UC14 |
| BR9 | Xoá bài hát khỏi danh sách | UC06 |
| BR10 | Đổi tên danh sách | UC08 |
| BR11 | Phát nhạc | UC13 |
| BR12 | Hàng đợi nhạc | UC15, UC16 |
| BR13 | Thêm nhạc vào cơ sở dữ liệu | UC12, UC13 |
| BR14 | Xoá nhạc khỏi cơ sở dữ liệu | UC12 |

*Bảng 2.1: Bảng ánh xạ yêu cầu nghiệp vụ với các Usecase*

## Sơ đồ kiến trúc tổng thể



*Hình 2.2: Sơ đồ kiến trúc tổng thể hệ thống*

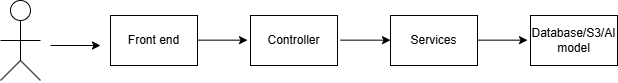
Hệ thống được thiết kế với 2 phần chính, bao gồm: client-side và server-side.

* Client-side: Ứng dụng web (native – viết bằng VueJs)
  + Người dùng tương tác với ứng dụng.
  + Ứng dụng trao đổi dữ liệu với server thông qua phương thức HTTP để phục vụ việc

phát bài hát và nhận gợi ý bài hát

* Server-side: Sử dụng python (FastAPI)
  + Giao tiếp với mongodb để lấy thông tin về nhạc và dữ liệu người dùng
  + Giao tiếp với SQL server để lấy thông tin về đăng nhập người dùng
  + Giao tiếp với S3 storage để tải nhạc về
  + Giao tiếp với AI model để trả về kết quả gợi ý nhạc cho người dùng

## Cấu trúc chung

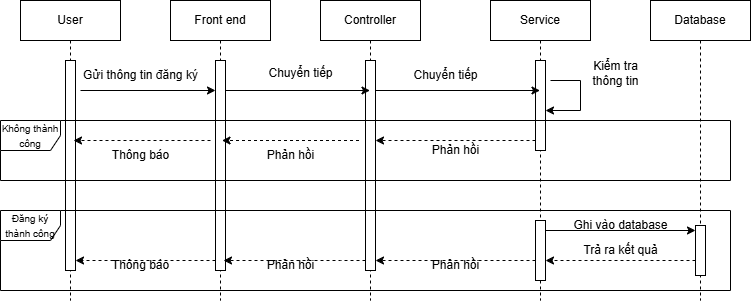


*Hình 2.3: Cấu trúc chung của chức năng*

* **Actor**: Tác nhân tham gia hệ thống
* **Front end**: UI để người dùng tương tác với hệ thống
* **Controller**: Lớp nhận vào các đầu vào
* **Services**: Lớp thực thể để xử lý logic của hệ thống
* **Database/ S3/ AI model**: Cơ sở dữ liệu, kho chứa đối tượng hoặc là mô hình AI

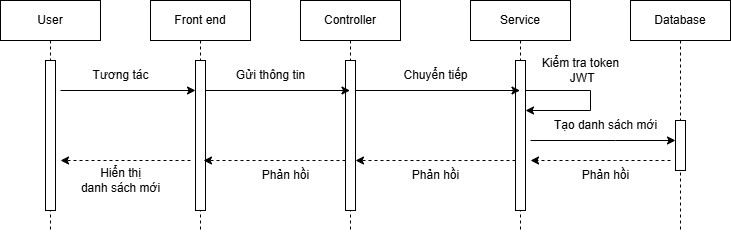
## 2.6. Thiết kế một số chức năng

### 2.6.1. UC#01: Đăng ký



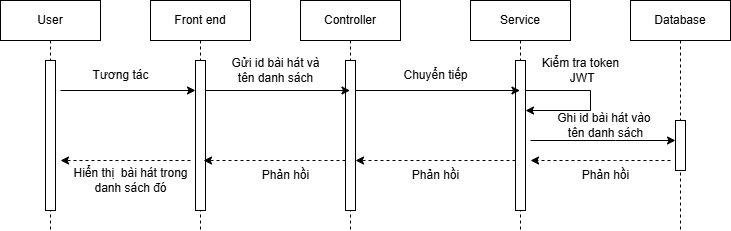
*Hình 2.4: Sơ đồ tuần tự chức năng Đăng ký*

### 2.6.2. UC#04: Tạo danh sách tự tạo



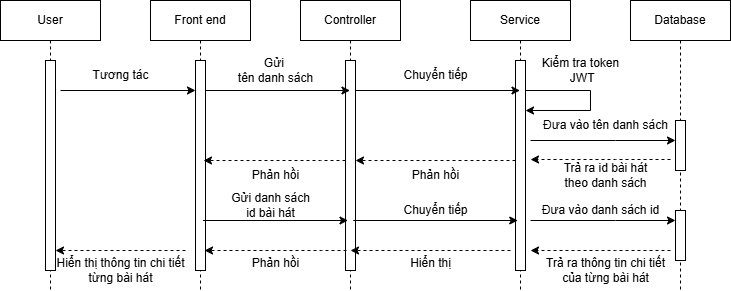
*Hình 2.5: Sơ đồ tuần tự chức năng Tạo danh sách tự tạo*

### 2.6.3. UC#05: Thêm một bài hát vào danh sách tự tạo



*Hình 2.6: Sơ đồ tuần tự chức năng* *Thêm một bài hát vào danh sách tự tạo*

### 2.6.4. UC#09: Lấy danh sách các bài hát của danh sách tự tạo

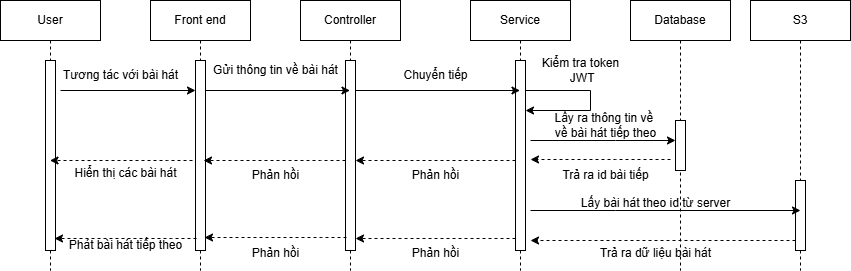


*Hình 2.7: Sơ đồ tuần tự chức năng lấy danh sách các bài hát của danh sách tự tạo*

### UC#11: Gợi ý bài hát theo đặc điểm từng người dùng

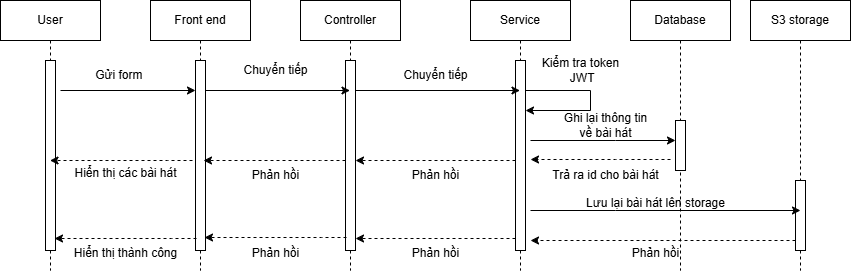
*Hình 2.8: Sơ đồ tuần tự chức năng* *Gợi ý bài hát theo từng người dùng*

### UC#15: Phát bài hát tiếp theo trong hàng đợi



*Hình 2.9: Sơ đồ tuần tự chức năng Phát bài hát tiếp theo trong hàng đợi*

### UC#17: Thêm bài hát mới vào database



*Hình 2.10: Sơ đồ tuần tự chức năng Thêm bài hát mới vào database*

## Thiết kế CSDL

### Mô tả hệ thống

Hệ thống được thiết kế nhằm mục đích lưu thông tin đăng nhập người dùng, thông tin về bài hát và các thông tin liên quan đến bài hát của từng người dùng. Trong đó, thông tin người dùng gồm có tên đăng nhập, email, tên đầy đủ, vài trò của người dùng, mật khẩu đã được mã hoá. Người dùng có thể tạo, sửa, xoá, đổi tên danh sách bài hát cá nhân và xem thông tin chi tiết về những bài hát trong danh sách cá nhân đó. Sau khi bài hát đã được thêm vào danh sách cá nhân, người dùng có thể xoá đi các bài hát đã thêm. Người dùng có thể tìm kiếm một bài hát trên cơ sở dữ liệu. Khi người dùng phát một bài hát đồng thời bài hát đó được thêm vào hàng đợi và người dùng cho thể lùi về bài hát trước đó hoặc đi tới bài hát tiếp theo.

### Lựa chọn các danh từ làm lớp thực thể hoặc thuộc tính

* Microsoft SQL server
  + Thông tin về người dùng -> Lớp users: tên đăng nhập, email, tên đầy đủ, vai trò người dùng, mật khẩu đã được mã hoá
* MongoDB
  + Thông tin âm nhạc người dùng -> Lớp UserData: danh sách nhạc cá nhân (**musicListCustom**), vị trí hiện tại của danh sách phát (**current\_index**), danh sách nhạc hiện tại (**current\_list**)
  + Thông tin về danh sách tự tạo -> Lớp MusicListCustom: id của người dùng, danh sách nhạc mà người dùng tự tạo (**có thể thêm bất ký trường nào để mở rộng**)
  + Thông tin về bài hát-> Lớp SongData: id của bài hát, tên bài hát, thời gian bài hát, nghệ sĩ trình bày

### Mô tả tổng quan từng bảng

|  |  |  |
| --- | --- | --- |
| **Microsoft SQL Server** | | |
| **STT** | **Tên bảng** | **Mô tả** |
| 1 | users | Lưu trữ thông tin đăng nhập và cá nhân người dùng |

|  |  |  |
| --- | --- | --- |
| **MongoDB** | | |
| **STT** | **Tên bảng** | **Mô tả** |
| 1 | MusicListCustom | Lưu trữ thông tin liên quan đến danh sách nhạc cá nhân |
| 2 | SongData | Lưu trữ thông tin chi tiết của nhạc |
| 3 | UserData | Lưu trữ thông tin liên quan đến âm nhạc của cá nhân |

*Bảng 2.2: Danh sách các bảng trong CSDL*

### Chi tiết các bảng

|  |  |  |  |
| --- | --- | --- | --- |
| **Microsoft SQL Server** | | | |
| **Bảng** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| users | username | STRING | Primary key |
| email | STRING | Email người dùng |
| fullname | STRING | Tên đầy đủ người dùng |
| rolename | STRING | Vai trò người dùng |
|  | hashed\_password | STRING | Mật khẩu đã được mã hoá |

*Bảng 2.3: Bảng cơ sở dữ liệu Users*

|  |  |  |  |
| --- | --- | --- | --- |
| **MongoDB** | | | |
| **Bảng** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| MusicListCustom | \_id | STRING | Primary key |
| {any\_value} | ARRAY[STRING] | Tên trường của danh sách nhạc cá nhân |
|  | {any\_value} | ARRAY[STRING] | Tên trường của danh sách nhạc cá nhân |
|  | {……} | ARRAY[STRING] | Tên trường của danh sách nhạc cá nhân |

*Bảng 2.4: Bảng cơ sở dữ liệu MuscicListCustom*

|  |  |  |  |
| --- | --- | --- | --- |
| **MongoDB** | | | |
| **Bảng** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| SongData | \_id | STRING | Primary key |
| name | INT | Tên bài hát |
| duration | INT | Thời gian một bài hát |
|  | artist | STRING | Tên nghệ sĩ trình bày |

*Bảng 2.5: Bảng cơ sở dữ liệu SongData*

|  |  |  |  |
| --- | --- | --- | --- |
| **MongoDB** | | | |
| **Bảng** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| UserData | \_id | STRING | Primary key |
| musicListCustom | ARRAY[STRING] | Danh sách các danh sách cá nhân |
| current\_index | STRING | Vị trí hiện tại của danh sách phát |
| current\_list | ARRAY[STRING] | Danh sách phát nhạc hiện tại |

*Bảng 2.6: Bảng cơ sở dữ liệu UserData*

## Thiết kế API

### Cấu trúc tổng thể của API

API sử dụng cấu trúc với 3 thành phần chính là Controller, Service và Database. Phần View được lược bỏ. Lúc này, phần Controller chỉ trả về các dữ liệu Json.

Phần Controller được chia thành các Action khác nhau. Mỗi Action thực hiện môn nghiệp vụ duy nhất trong nhóm nghiệp vụ được bao bọc bởi Controller. Và controller sẽ gọi tới Service là phần xử lý logic nghiệp vụ.

Phần Service là nơi xử lý nghiệp vụ của ứng dụng, nó sẽ chịu trách nhiệm xử lý mọi logic của ứng dụng và nếu cần bất kỳ dữ liệu nó sẽ gọi tới Database.

Phần Database là nơi chịu trách nhiệm kết nối với cơ sở dữ liệu và thực hiện các truy vấn theo yêu cầu của Service.

Về kiểu HTTP Request, API có hỗ trợ Restful Api. Qua đó các router của API đa số sẽ sử dụng danh từ và tác thao tác với API sẽ phụ thuộc vào 4 kiểu Request là: GET; POST; PUT, DELETE và tham số đường dẫn truyền vào.

### Api hệ thống sử dụng

Người dùng có thể phát bài hát, nhận được gợi ý từ AI model, tạo một danh sách nhạc cá nhân và tương tác với nó. Ngoài ra thêm vài trò admin có thể thêm mới nhạc hoặc xoá nhạc đi.

Tham số user\_id được tự động gọi trong controller và nó sẽ tự động kiểm tra quyền của người dùng trong token, ở đây có hai quyền là admin và user. Nếu hợp lệ nó sẽ trả ra id người dùng và thực hiện, nếu không nó sẽ báo bạn không có quyền

|  |  |
| --- | --- |
| **URL** | **Chi tiết** |
| **/audio/get\_audio** | Mục đích: Lấy nhạc từ S3 storage Phương thức: Get  Yêu cầu đặc biệt: Không  Tham số:   * id (param) - kiểu string, bắt buộc: id bài hát * token (cookies) – tự động kiểm tra |
| **/auth/login\_for\_token** | Mục đích: Đăng nhập Phương thức: POST  Yêu cầu đặc biệt: Không  Tham số:   * form\_data (body) – json object, bắt buộc: chứa thông tin đăng nhập của người dùng |
| **/auth/register** | Mục đích: Đăng ký tài khoản Phương thức: POST  Yêu cầu đặc biệt: Không  Tham số:   * user\_info (body) – json object, bắt buộc: chứa thông tin đăng ký tài khoản của người dùng |
| **/auth/valid\_token** | Mục đích: Kiểm tra xem token có hợp lệ không  Phương thức: GET Yêu cầu đặc biệt: Không  Tham số:   * token (cookies): tự động kiểm tra hợp lệ và quyền |
| **/user\_router/get\_list** | Mục đích: Lấy ra danh sách nhạc cá nhân  Phương thức: GET Yêu cầu đặc biệt: Không Tham số:   * user\_id - string: tự động được trích từ token |
| **/user\_router/adding\_list** | Mục đích: Thêm mới danh sách cá nhân  Phương thức: POST Yêu cầu đặc biệt: Không Tham số:   * custom\_list\_input (body) – json object: chứa thông tin về một danh sách để thêm mới vào * user\_id - string: tự động được trích từ token |
| **/user\_router/rename\_list** | Mục đích: Xoá danh sách cá nhân  Phương thức: DELETE Yêu cầu đặc biệt: Không Tham số:   * new\_update (body) – json object, bắt buộc * user\_id - kiểu string: tự động được trích từ token |
| **/user\_router/delete\_list** | Mục đích: Xoá danh sách cá nhân  Phương thức: DELETE Yêu cầu đặc biệt: Không Tham số:   * index (param) – int, bắt buộc: vị trí của danh sách cá nhân * user\_id - kiểu string: tự động được trích từ token |
| **/user\_router/get\_info\_song\_from\_list** | Mục đích: Lấy thông tin chi tiết về bài hát  Phương thức: POST Yêu cầu đặc biệt: Không Tham số:   * list\_song\_id (body) – json, bắt buộc: danh sách id của bài hát * user\_id - kiểu string: tự động được trích từ token |
| **/user\_router/get\_detail\_custom\_list** | Mục đích: Lấy danh sách id bài hát của một danh sách cá nhân  Phương thức: GET Yêu cầu đặc biệt: Không Tham số:   * name\_custom\_list (param) – string, bắt buộc: tên của danh sách cá nhân * user\_id - kiểu string: tự động được trích từ token |
| **/user\_router/add\_song\_to\_custom\_list** | Mục đích: Thêm một bài hát vào danh sách cá nhân  Phương thức: POST  Yêu cầu đặc biệt: Không Tham số:   * info\_song (body) – json, bắt buộc: thông tin về bài hát và danh sách * user\_id - kiểu string: tự động được trích từ token |
| **/user\_router/remove\_song\_from\_custom\_list** | Mục đích: Xoá một bài hát khỏi danh sách cá nhân  Phương thức: DELETE  Yêu cầu đặc biệt: Không Tham số:   * custom\_list\_name (param) – string, bắt buộc: tên của danh sách cá nhân * index (param) – string, bắt buộc: chỉ số của của bài hát * user\_id - kiểu string: tự động được trích từ token |
| **/user\_router/next\_song\_current** | Mục đích: Trả ra id của bài hát tiếp theo trong danh sách phát  Phương thức: PUT  Yêu cầu đặc biệt: Không Tham số:   * user\_id - kiểu string: tự động được trích từ token |
| **/user\_router/head\_back\_song\_current** | Mục đích: Trả ra id của bài hát trước đó trong danh sách phát  Phương thức: DELETE  Yêu cầu đặc biệt: Không Tham số:   * user\_id - kiểu string: tự động đượctrích từ token |
| **/user\_router/get\_current\_song** | Mục đích: Trả ra id của bài hát hiện tại trong danh sách phát  Phương thức: GET  Yêu cầu đặc biệt: Không Tham số:   * user\_id - kiểu string: tự động đượctrích từ token |
| **/user\_router/search\_for\_song** | Mục đích: Tìm kiếm bài hát theo từ khoá  Phương thức: GET  Yêu cầu đặc biệt: Không Tham số:   * key\_words – kiểu string: từ khoá tìm kiếm * user\_id - kiểu string: tự động được trích từ token |
| **/user\_router/get\_recommend\_song** | Mục đích: Lấy những bài hát gợi ý cho cá nhân  Phương thức: GET  Yêu cầu đặc biệt: Không Tham số:   * user\_id - kiểu string: tự động được trích từ token |
| **/admin\_router/adding\_new\_music** | Mục đích: Thêm bài hát mới vào database và s3 storage  Phương thức: POST  Yêu cầu đặc biệt: vai trò admin Tham số:   * file – kiểu file excel: file chứa thông tin về nhạc * user\_id - kiểu string: tự động được trích từ token |
| **/admin\_router/detelte\_music** | Mục đích: Thêm bài hát mới vào database và s3 storage  Phương thức: DELETE  Yêu cầu đặc biệt: vai trò admin Tham số:   * id (param): id của bài hát cần xoá * user\_id - kiểu string: tự động được trích từ token |

*Bảng 2.7: API hệ thống sử dụng*

# CHƯƠNG 3: THU THẬP DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH

## 3.1. Thu thập dữ liệu

* **Nguồn thu thập dữ liệu:** Dữ liệu được dùng cho nghiên cứu gồm 2 tập train\_triplets và unique\_tracks nằm trong bộ dữ liệu “Million song dataset” được thu thập từ trang mạng: millionsongdataset.com
* **Phương pháp thu thập dữ liệu:** Bộ dữ liệu được tải trực tiếp từ trang mạng millionsongdataset.com
* **Dạng dữ liệu:** Cả 2 tập train\_triplets và unique\_track đều là file dữ liệu txt
* **Kích thước:** 
  + train\_triplets.txt: 2.9 GB
  + unique\_track: 82 MB
* **Nhãn dữ liệu:** Dữ liệu không có nhãn

## 3.2. Thống kê và tiền xử lý dữ liệu

### 3.2.1. Bộ dữ liệu train\_triplets

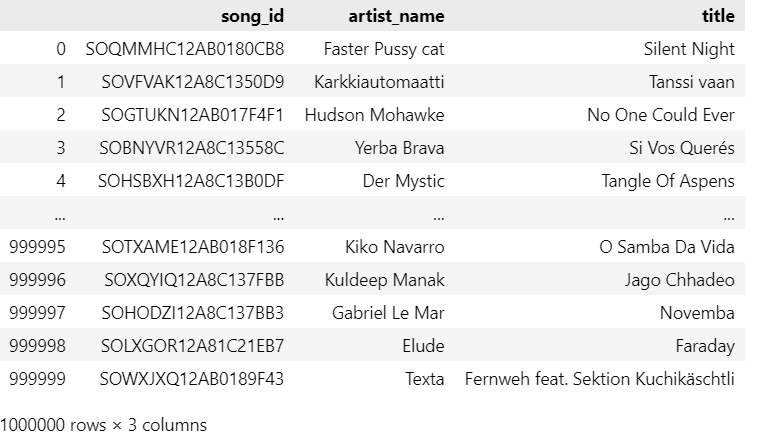
Bộ dữ liệu train\_triplets chứa thông tin về người dùng có 48373586 bản ghi và 3 trường dữ liệu tương ứng là user\_id (id người dùng), song\_id (id bài hát) và listen\_count (số lượt nghe tương ứng).



*Hình 3.1: Bộ dữ liệu train\_triplets*

### 3.2.2. Bộ dữ liệu unique\_tracks

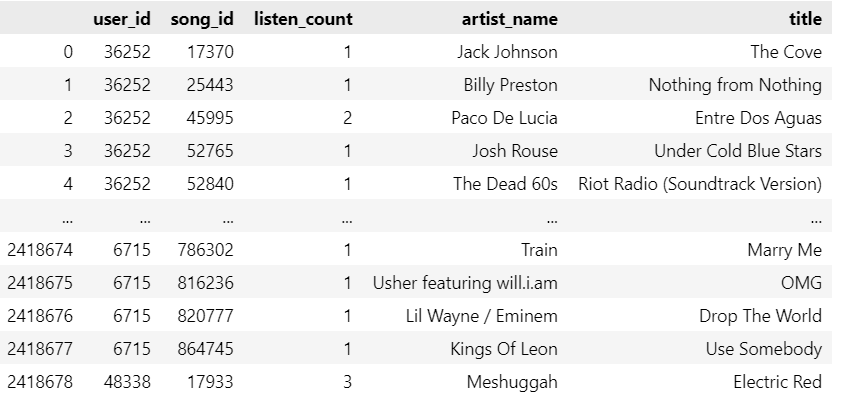
Bộ dữ liệu unique\_tracks chứa thông tin về 1000000 bài hát tương ứng với 1000000 bản ghi và gồm có 3 trường dữ liệu là: song\_id (id bài hát), artist\_name (tên nghệ sĩ), title (tên bài hát).



*Hình 3.2: Bộ dữ liệu unique\_tracks*

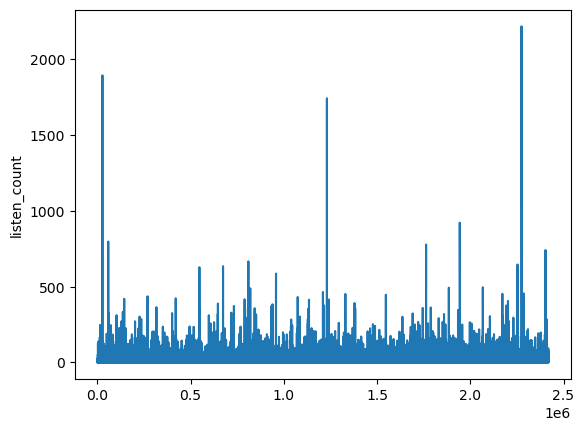
### 3.2.3. Tiền xử lý

Do thiết bị và khả năng xử lý còn hạn chế nên KLTN chỉ sử dụng 5% dữ liệu từ bộ dữ liệu train\_triplets.txt, tương ứng với 2418673 bản ghi đầu tiên trong tập dữ liệu gốc.

1. **LabelEncoder:** Chuyển các trường dữ liệu user\_id và song\_id ở cả tập train\_triplet và tập unique\_tracks từ dạng chuỗi về dạng số.
2. **Gộp dữ liệu:** Gộp chung 2 tập dữ liệu train\_triplets.txt và unique\_tracks.txt thành một tập thông qua trường dữ liệu user\_id chung của cả 2 tập 

*Hình 3.3: Bộ dữ liệu sau khi gộp train\_triplets và unique\_tracks*

1. **Chuẩn hóa dữ liệu:**

****

*Hình 3.4: Biểu đồ lượng lượt nghe trên toàn bộ tập dữ liệu*

Số 'list\_count' cao nhất là 2213, trong khi số thấp nhất là 1. Phạm vi lớn này sẽ tạo ra vấn đề cho mô hình và yêu cầu chuẩn hóa.

* **Chuẩn hóa Z-scale:**
  + Công thức:
  + Trong đó:
    - X là giá trị ban đầu của dữ liệu
    - là giá trị trung bình của dữ liệu
    - là độ lệch chuẩn của dữ liệu
    - Z là giá trị sau khi chuẩn hóa
* **Chuẩn hóa Min-Max:** 
  + Công thức:
  + Trong đó:
    - X là giá trị ban đầu của dữ liệu
    - là giá trị nhỏ nhất của dữ liệu
    - là giá trị lớn nhất của dữ liệu
    - là giá trị sau khi chuẩn hóa

## 3.3. Xây dựng mô hình

### 3.3.1. Phân chia dữ liệu

* Tập huấn luyện: Chứa 90% dữ liệu và được sử dụng để huấn luyện mô hình. Trong quá trình này, mô hình học cách dự đoán lượt nghe dựa trên dữ liệu lịch sử.
* Tập kiểm tra: Chứa 10% dữ liệu và được sử dụng để đánh giá hiệu suất của mô hình. Điều này giúp đảm bảo rằng mô hình không bị overfitting (khớp quá mức) với dữ liệu huấn luyện.

Tập dữ liệu gồm 2418679 mẫu được chia thành hai tập huấn luyện và kiểm tra với tỷ lệ 9:1:

* Tập huấn luyện: 2176811 mẫu
* Tập kiểm tra: 241868 mẫu

### 3.3.2. Xây dựng SparseTensor

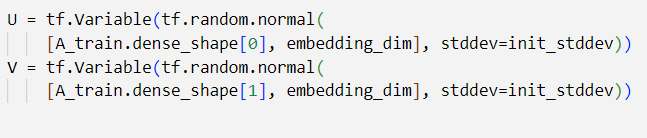
Sau khi chia dữ liệu, KLTN thực hiện chuyển đổi dữ liệu thành dạng ma trận thưa (SparseTensor). Điều này giúp xử lý hiệu quả hơn khi làm việc với các thuật toán machine learning, đặc biệt là khi dữ liệu rất lớn và thưa thớt.

**SparseTensor**: Là cấu trúc dữ liệu hiệu quả để lưu trữ và xử lý các ma trận thưa. Thay vì lưu trữ tất cả các giá trị (bao gồm cả giá trị không), SparseTensor chỉ lưu trữ các phần tử không phải là số 0 cùng với chỉ số của chúng.

### 3.3.3. Khởi tạo embedding

**Embedding:** là các đại diện số học của người dùng và bài hát trong một không gian vector đa chiều. Bước này khởi tạo các embedding ngẫu nhiên cho người dùng và bài hát. Quá trình huấn luyện sẽ tinh chỉnh các giá trị này để tối ưu hóa dự đoán của mô hình.

* **Embedding của người dùng (U):** Vector đại diện cho người dùng.
* **Embedding của bài hát (V):** Vector đại diện cho bài hát.
* Khởi tạo với giá trị ngẫu nhiên để bắt đầu quá trình học.

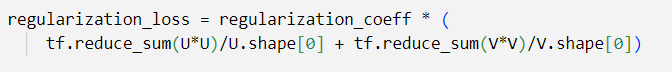


* **tf.Variable**: Tạo các biến TensorFlow cho các embedding.
* **tf.random.normal**: Khởi tạo các giá trị ngẫu nhiên từ phân phối chuẩn với độ lệch chuẩn xác định.

### 3.3.4. Tính toán lỗi

Lỗi (loss) được tính toán dựa trên chênh lệch giữa dự đoán của mô hình và giá trị thực tế. Tổng lỗi bao gồm ba thành phần: lỗi huấn luyện, lỗi regularization, và gravity loss.

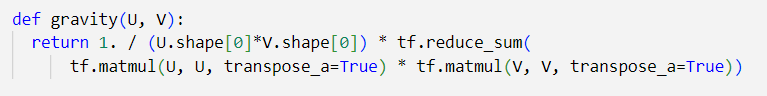
* **Lỗi huấn luyện (training error):** Chênh lệch giữa dự đoán và giá trị thực tế trong tập huấn luyện. KLTN sử dụng phương pháp mean square error (MSE) để đo lường lỗi.
  + Công thức MSE:
  + Trong đó:
    - n: số lượng mẫu
    - : giá trị thực tế của mẫu thứ i
    - : giá trị dự đoán của mẫu thứ i
    - : Bình phương sai số, giúp loại bỏ các sai số âm và làm nổi bật các sai số lớn
  + MSE càng nhỏ, mô hình dự đoán càng chính xác. MSE bằng 0 nghĩa là mô hình dự đoán hoàn toàn chính xác giá trị thực tế
* **Regularization loss:** Được thêm vào để tránh overfitting bằng cách phạt các embedding có giá trị lớn. Điều này giúp các embedding không quá khớp với dữ liệu huấn luyện và có khả năng tổng quát tốt hơn với dữ liệu mới.



* + regularization\_coeff: Hệ số điều chỉnh mức độ phạt.
  + tf.reduce\_sum(U\*U)/U.shape[0]: Tính tổng bình phương các phần tử trong embedding của người dùng và chia cho số lượng người dùng để lấy giá trị trung bình.
  + tf.reduce\_sum(V\*V)/V.shape[0]: Tương tự cho embedding của bài hát.
* **Gravity loss:** Giúp giảm tương tác giữa các embedding, ngăn chặn việc các embedding của người dùng và bài hát trở nên quá gần nhau, giúp tránh overfitting.



* + gravity\_coeff: Hệ số điều chỉnh mức độ ảnh hưởng của gravity loss.
  + gravity(U, V): Hàm tính toán gravity loss giữa embedding của người dùng và bài hát.



* + tf.matmul(U, U, transpose\_a=True): Tính tích vô hướng giữa các embedding của người dùng.
  + tf.matmul(V, V, transpose\_a=True): Tương tự cho embedding của bài hát.
  + tf.reduce\_sum: Tính tổng của các tích vô hướng này.
  + 1. / (U.shape[0]\*V.shape[0]): Bình quân hóa gravity loss theo số lượng người dùng và bài hát
* **Tổng hợp các thành phần lỗi:** Tổng lỗi sẽ được tính bằng cách cộng các thành phần lỗi huấn luyện, regularization loss và gravity loss.



* **Kết luận:** Phần tính toán lỗi rất quan trọng vì nó ảnh hưởng trực tiếp đến cách mô hình học từ dữ liệu và tối ưu hóa các tham số embedding. Tổng lỗi bao gồm ba thành phần: lỗi huấn luyện, regularization loss, và gravity loss, giúp đảm bảo rằng mô hình không chỉ học tốt từ dữ liệu huấn luyện mà còn có khả năng tổng quát tốt đối với dữ liệu mới.

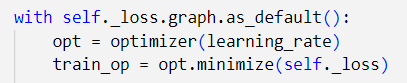
### 3.3.5. Huấn luyện mô hình

Mục tiêu của quá trình huấn luyện mô hình là tối ưu hóa các embedding của người dùng và bài hát sao cho mô hình có thể dự đoán chính xác số lượt nghe bài hát (listen\_count) của người dùng. Chúng ta sử dụng thuật toán Gradient Descent để tối ưu hóa hàm lỗi (total\_loss) thông qua nhiều vòng lặp huấn luyện.



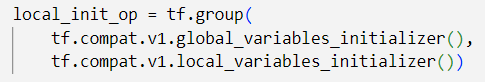
* Gradient Descent: Thuật toán này tìm các giá trị tối ưu cho các biến bằng cách di chuyển theo hướng ngược lại với gradient của hàm lỗi.
* Số vòng lặp (num\_iterations): Số lần mô hình cập nhật các tham số trong quá trình huấn luyện.
* learning\_rate: Tốc độ học của thuật toán, ảnh hưởng đến mức độ điều chỉnh các tham số trong mỗi bước.

1. **Thiết lập biểu đồ tính toán và tối ưu hóa**

****

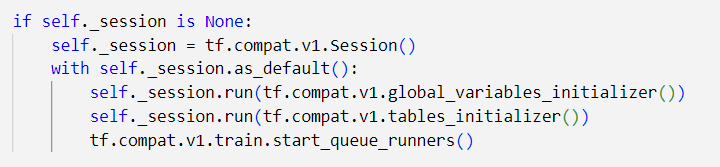
* self.\_loss.graph.as\_default(): Đặt biểu đồ tính toán hiện tại làm mặc định.
* optimizer(learning\_rate): Tạo một optimizer với learning rate xác định.
* opt.minimize(self.\_loss): Định nghĩa toán tử huấn luyện để giảm thiểu hàm lỗi.

1. **Khởi tạo các biến trong biểu đồ tính toán**

****

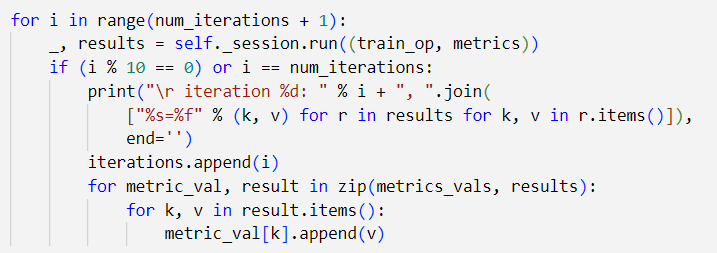
* tf.group(): Tạo một toán tử nhóm để khởi tạo các biến toàn cục và cục bộ.
* tf.compat.v1.global\_variables\_initializer(): Khởi tạo tất cả các biến toàn cục.
* tf.compat.v1.local\_variables\_initializer(): Khởi tạo tất cả các biến cục bộ.

1. **Khởi tạo phiên TensorFlow**

****

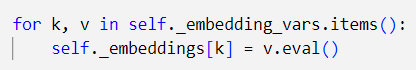
* self.\_session = tf.compat.v1.Session(): Tạo một phiên TensorFlow mới nếu chưa có.
* self.\_session.run(): Chạy các toán tử khởi tạo trong phiên TensorFlow.

1. **Vòng lặp huấn luyện**



* Lặp lại quá trình huấn luyện cho đến khi đạt đến số vòng lặp tối đa.
* self.\_session.run((train\_op, metrics)): Chạy toán tử huấn luyện và tính toán các chỉ số hiệu suất.

1. **Lưu lại embedding đã học**

****

* **v.eval():** Lấy giá trị hiện tại của biến TensorFlow và lưu vào từ điển self.\_embeddings.

### 3.3.6. Đánh giá

Sau khi thực hiện huấn luyện mô hình với các tham số:

* num\_iterations = 1000
* regularization\_coeff = 0.1
* gravity\_coeff = 1.0
* learning\_rate = 20

**Kết quả:**

MSE trên tập huấn luyện: 0.06514032

MSE trên tập kiểm tra: 0.06520172

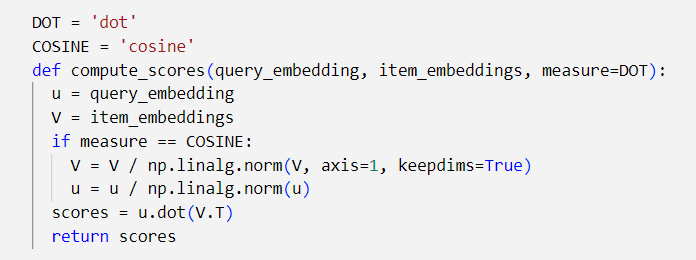
regularization\_loss: 0.017398924

gravity\_loss: 0.00021622922

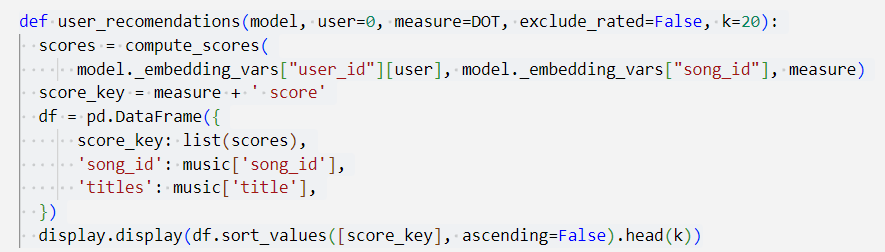
### 3.3.7. Xây dựng hàm gợi ý

Sau khi đã huấn luyện mô hình và có được các embedding của người dùng và bài hát, bước cuối cùng là sử dụng những thông tin này để gợi ý các bài hát mà người dùng có thể thích. Điều này được thực hiện thông qua hàm user\_recommendations(). Hàm này tính toán điểm số cho các bài hát dựa trên embedding của người dùng và bài hát, sau đó sắp xếp và hiển thị danh sách các bài hát gợi ý.

Hàm **compute\_scores** tính toán điểm số dựa trên embedding của người dùng và bài hát.



Hàm compute\_scores có thể sử dụng 2 phương pháp đo là: dot (Tích vô hướng) và cosine (cosine similarity).



Hàm gợi ý cho người dùng: **user\_recomendations**

* Đầu vào:
  + model: mô hình đã huấn luyện.
  + user: Id của người dùng cần gợi ý bài hát.
  + measure: phương pháp đo lường (dot hoặc cosine).
  + exclude\_rated: tùy chọn loại bỏ các bài hát đã được người dùng đánh giá.
  + k: số lượng bài hát gợi ý.
* Đầu ra: Danh sách bài hát được sắp xếp theo điểm số (dot hoặc cosine) giảm dần.

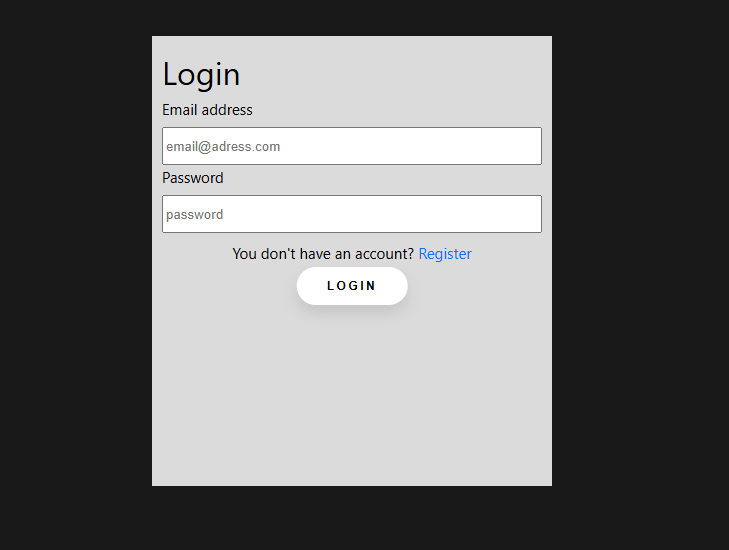
# CHƯƠNG 4. CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

## Môi trường cài đặt

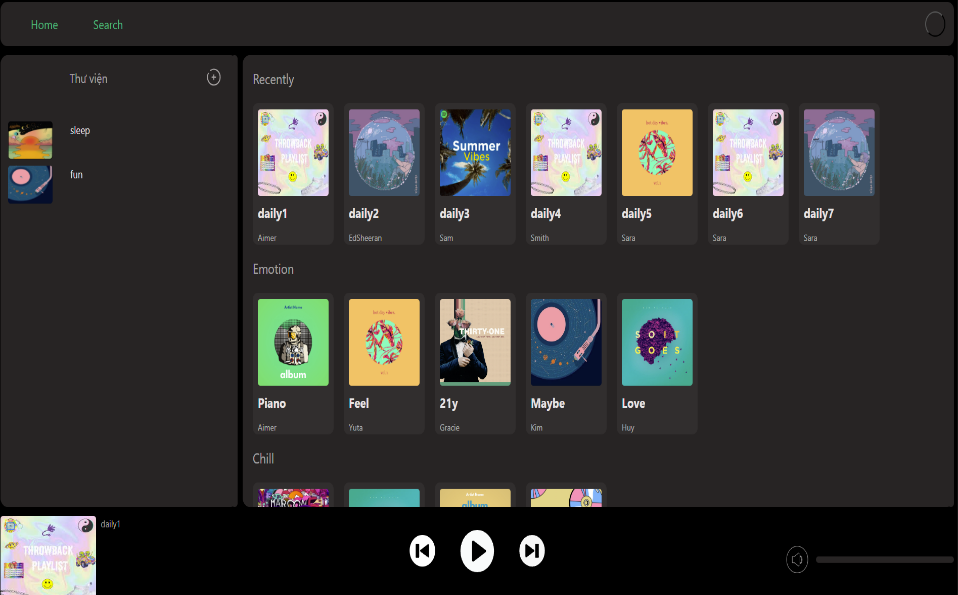
* Hệ điều hành bất kỳ
* Cần cài các ngôn ngữ và framework cần thiết
* Kết nối internet

## Trình diễn các chức năng của hệ thống

### Chức năng đăng nhập

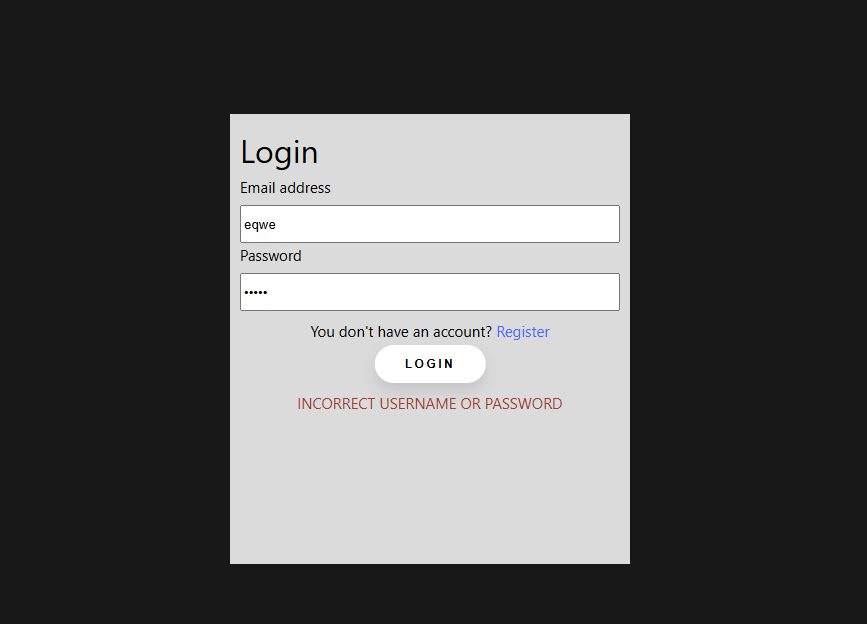
Tại màn hình, nhập thông tin để đăng nhập

*Hình 4.1: Màn hình đăng nhập*

Nếu đăng nhập thành công hệ thống sẽ hiện ra màn hình chính

*Hình 4.2: Màn hình chính*

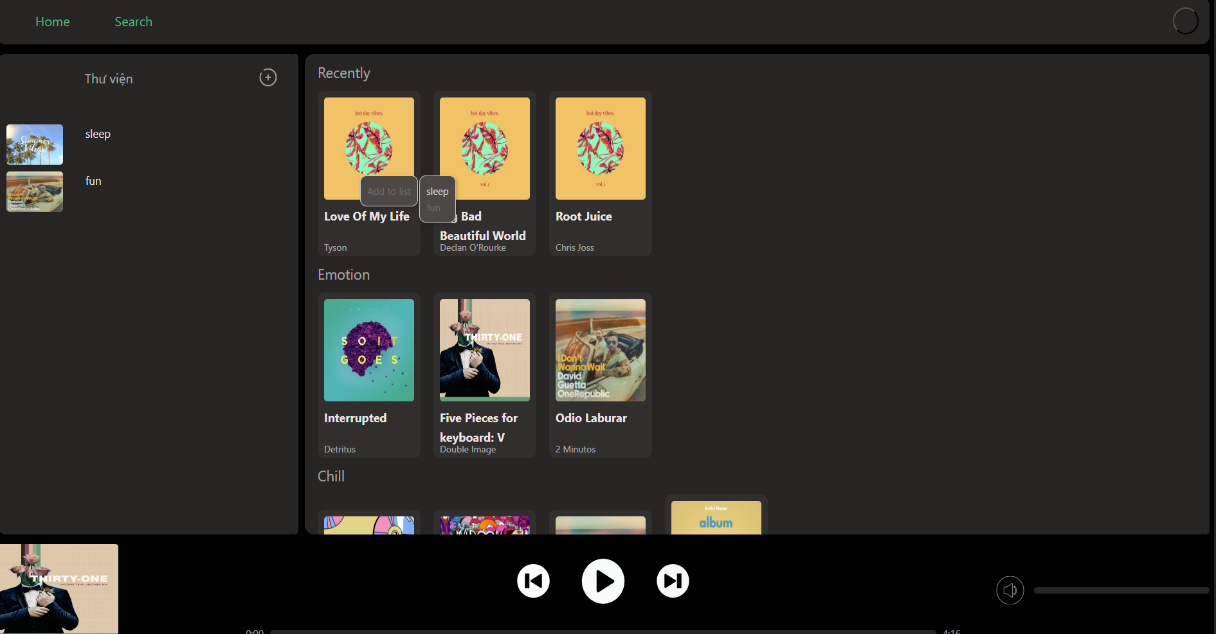
Nếu tài khoản người dùng chọn chưa có trên hệ thống hoặt sai mật khẩu sẽ có thông báo hiện ra.



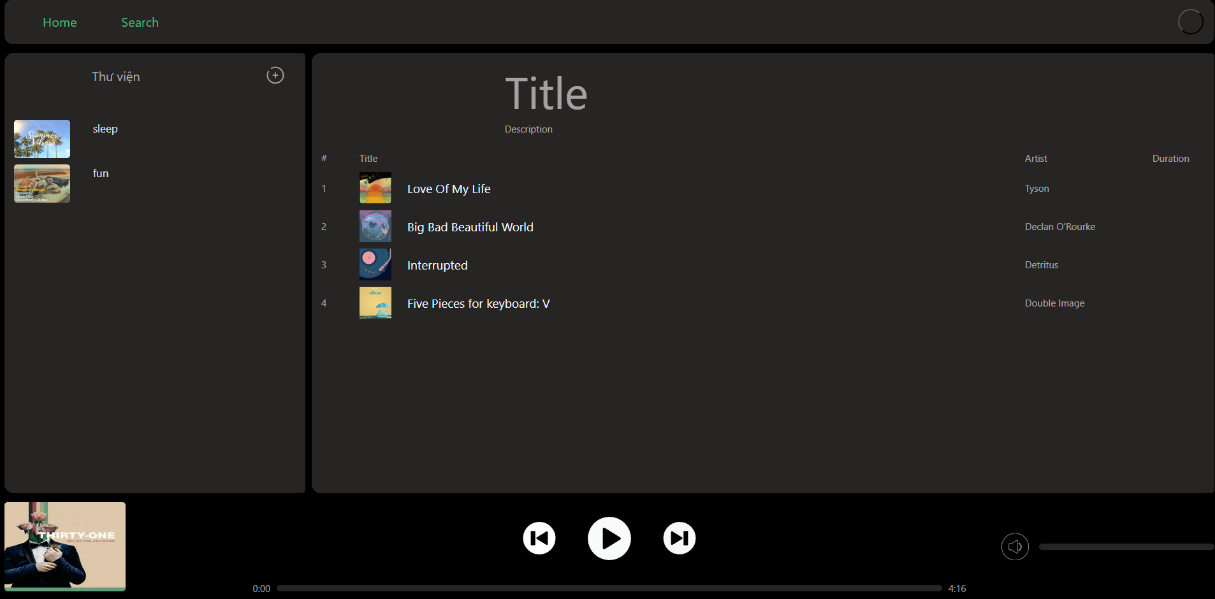
*Hình 4.3: Màn hình nhập nếu nhập thông tin sai*

### Chức năng thêm một bài hát vào danh sách yêu thích

Tại màn hình trang chủ, người dùng nhấn chuột phải vào một bài hát và để trỏ chuột vào thêm bài hát, sẽ có một màn hình về các danh sách bài hát.



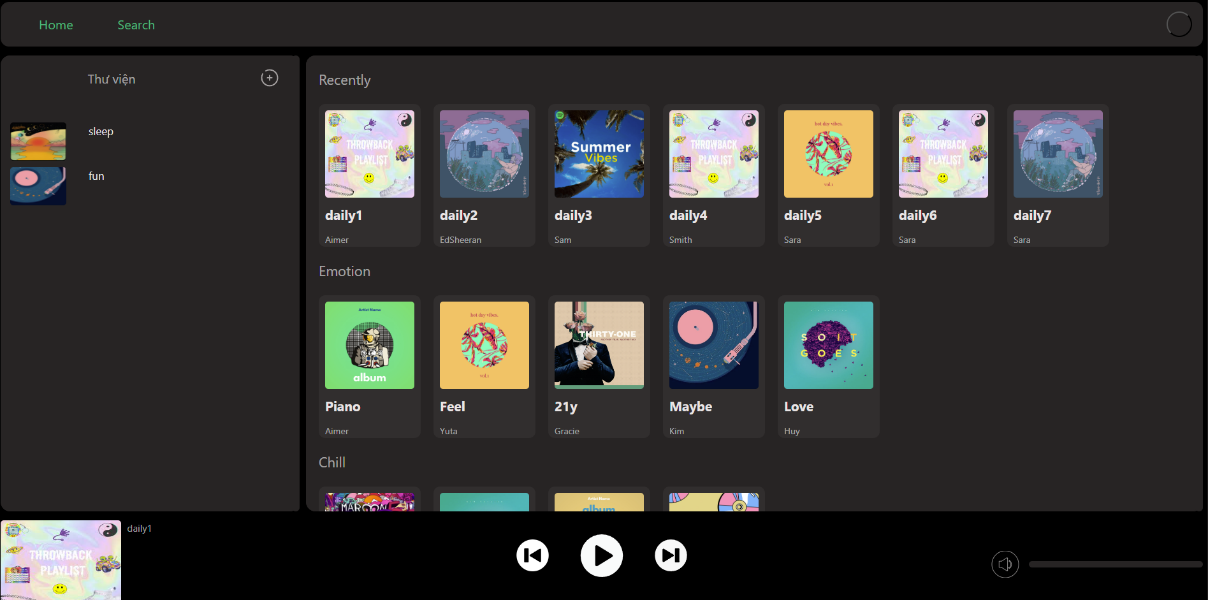
*Hình 4.4: Màn hình chọn tính năng tại trang chủ*

Nếu bấm vào một danh sách nhạc tự tạo thì sẽ thêm vào bài đó, ở đây ta thêm vào sleep thì trong sleep sẽ có bài này

*Hình 4.5: Màn hình danh sách các bài hát sau khi thêm*

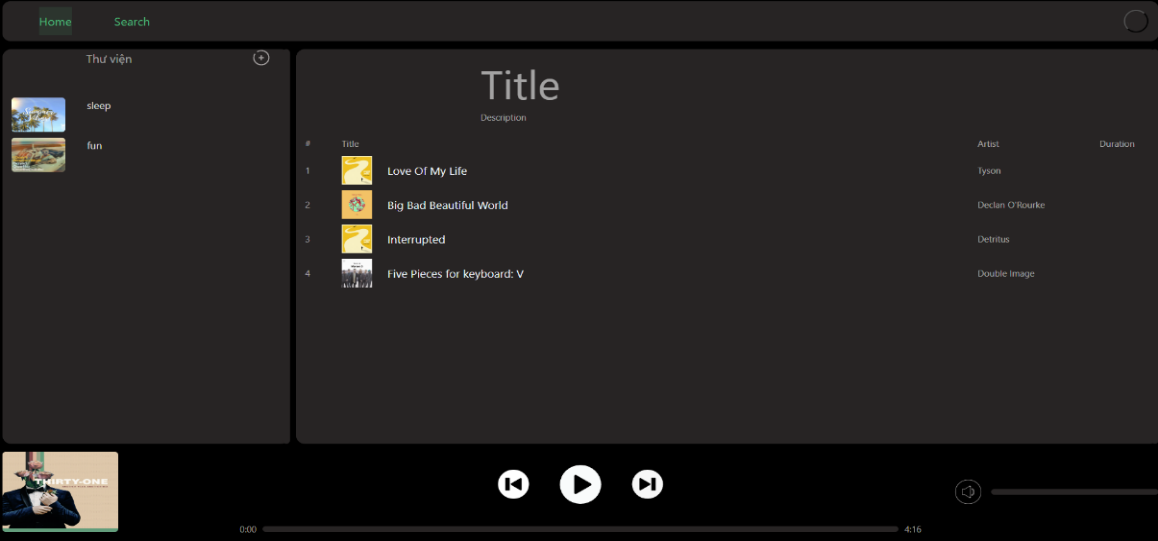
### Lấy các danh sách bài hát của danh sách tự tạo

Tại màn hình Trang chủ, người dùng bấm chuột trái vào một danh sách bất kỳ ở bên trái



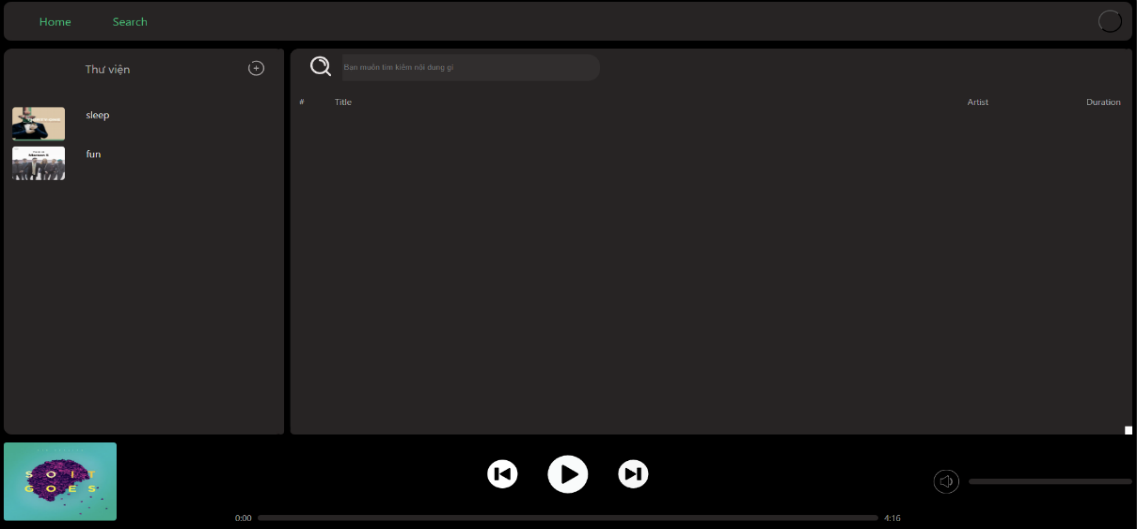
*Hình 4.6: Màn hình chọn tính năng tại trang chủ*

Ở bên phải màn hình sẽ xuất hiện danh sách các bài hát thuộc danh sách

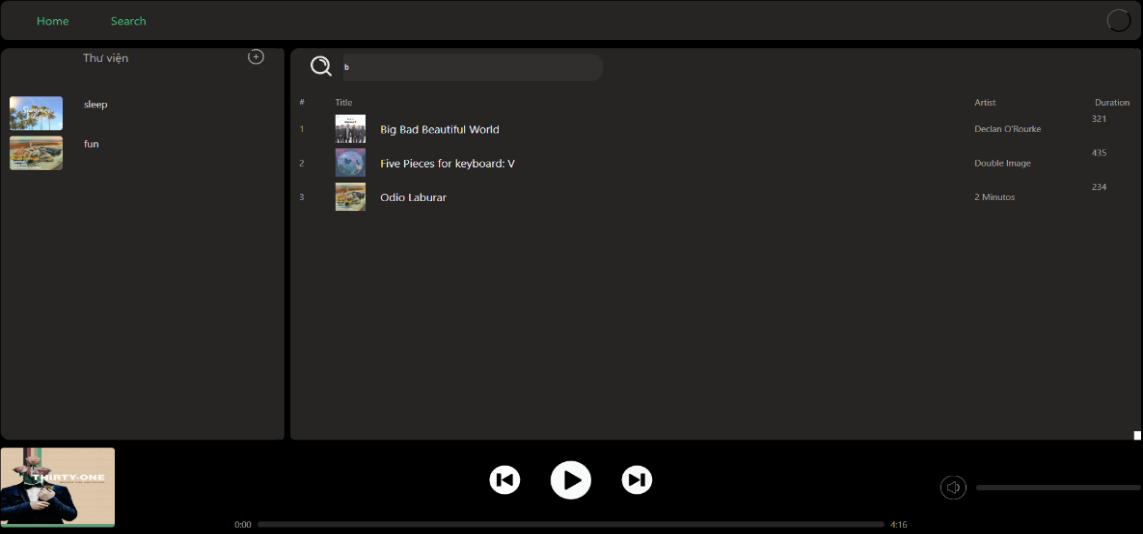


*Hình 4.7: Màn hình danh sách các bài hát tự tạo*

### Chức năng tìm kiếm bài hát

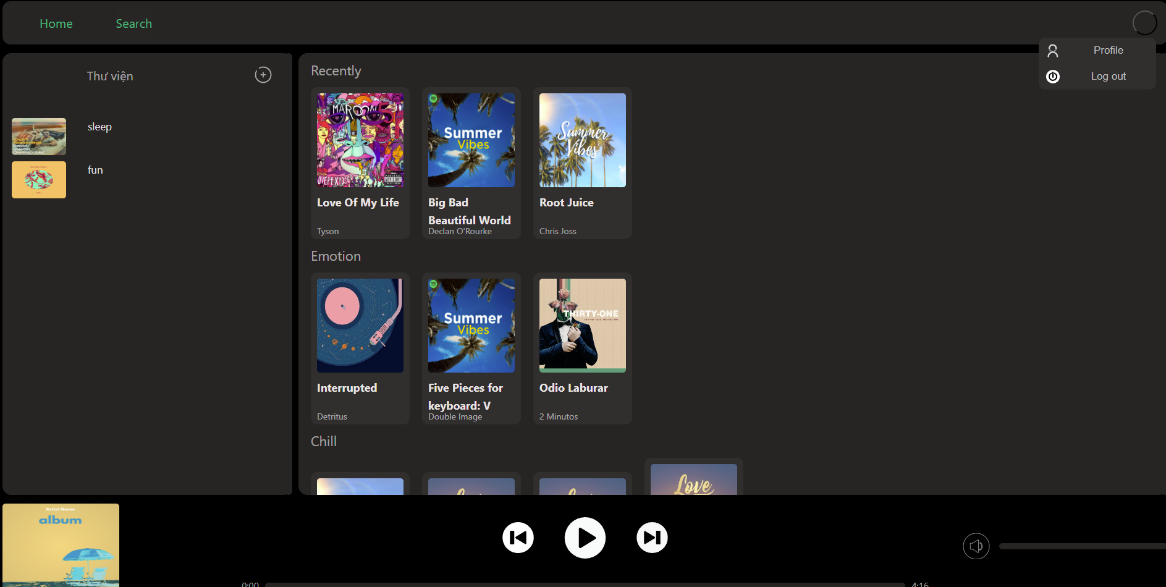
Tại màn hình chính, bấm vào tìm kiếm. Ở thanh tìm kiếm ta có thể điền tên bài hát cần tìm kiếm

*Hình 4.8: Màn hình tìm kiếm*

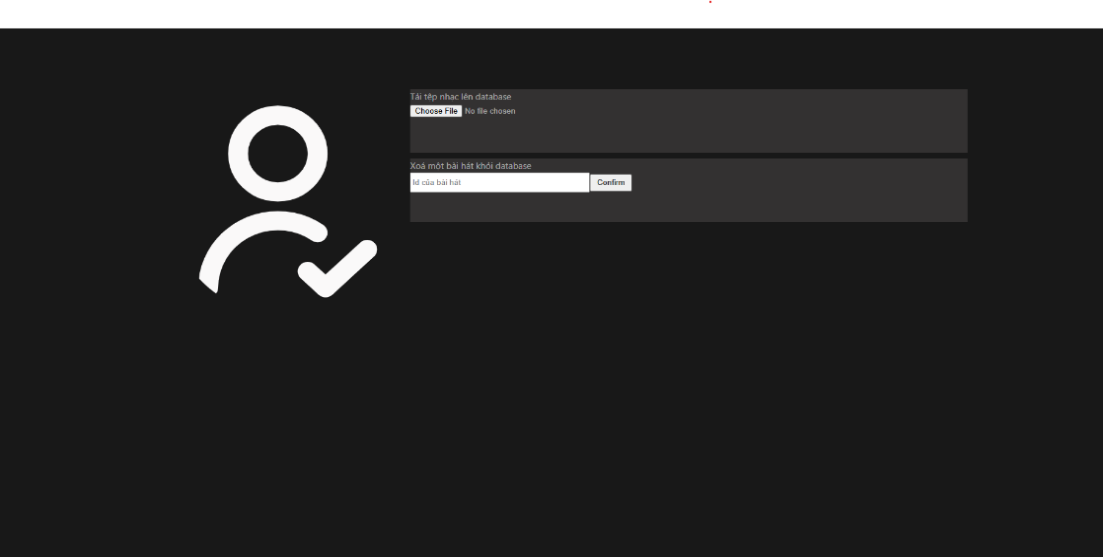
Ta nhập vào một kết quả và được kết quả, sẽ có thể có nhiều kết quả.

*Hình 4.9:Màn hình hiển thị kết quả tìm kiếm*

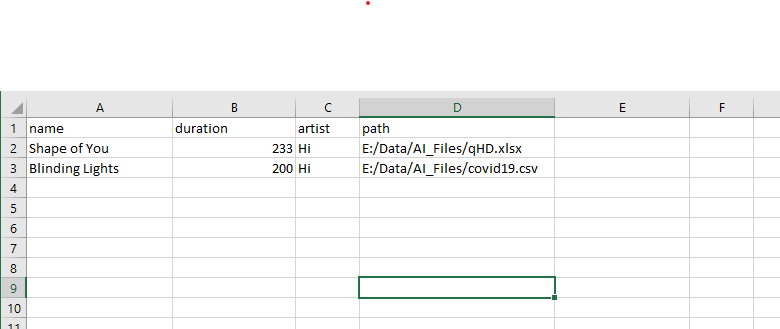
### Chức năng thêm mới dữ liệu

Tại màn hình chính, góc trên bên phải người dùng bấm vào nút tròn, một bảng hiện ra, người dùng bấm vào trang cá nhân để di chuyển tới trang cá nhân.

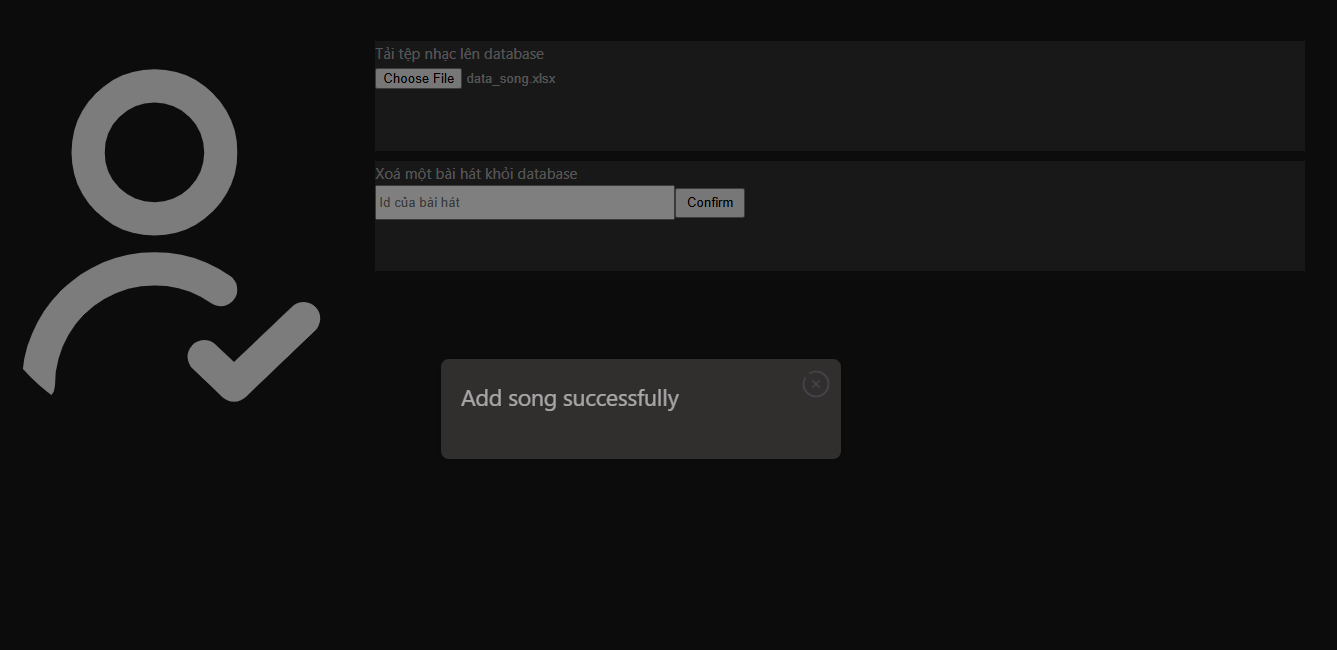
*Hình 4.10: Màn hình chính có nút vào profile*

Trong trang cá nhân người dùng có thể bấm chọn file để tải lên file chứa thông tin để tải dữ liệu lên.

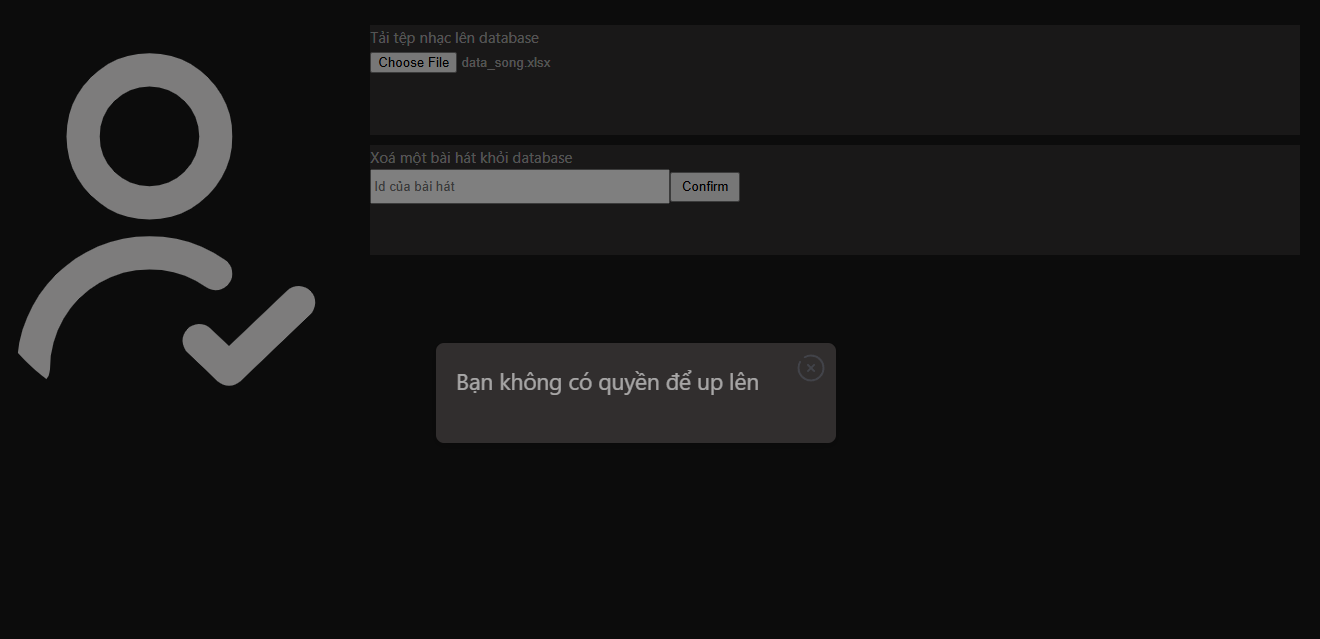
*Hình 4.11: Màn hình trang cá nhân*

Dữ liệu được thêm vào phải có dạng sau.

*Hình 4.12: Hình ảnh cấu trúc của file excel*

Nếu up lên thành công thì sẽ có thông báo thành công, thì sẽ có thông báo đưa ra thành công

*Hình 4.13: Màn hình thành công*

Nếu không có quyền thì bạn sẽ có thông báo.

*Hình 4.14: Màn hình thông báo không có quyền*

# KẾT LUẬN

**Kết quả đạt được:**

Trong quá trình thực hiện đề tài, chúng em đã hoàn thành việc xây dựng hệ thống khuyến nghị bài hát theo sở thích với nhiều chức năng có tính ứng dụng cao, đáp ứng nhu cầu đa dạng của người dùng trong việc tìm kiếm và nghe nhạc như khuyến nghị bài hát dựa trên sở thích cá nhân, tạo playlist tự động, kết nối người dùng có sở thích âm nhạc tương tự,...

Sau khi hoàn thiện đề tài, chúng em cũng tích lũy được cho mình thêm nhiều kiến thức và kỹ năng trong phát triển phần mềm như:

* Phân tích yêu cầu nghiệp vụ
* Tìm kiếm và đưa ra các giải pháp tối ưu cho các vấn đề của bài toán.
* Xử lý lỗi cả trong thiết kế lẫn lập trình
* Kỹ năng báo cáo, quản lý công việc.

Ngoài ra, chúng em cũng thu hoạch thêm các kinh nghiệm sử dụng ngôn ngữ Python, JavaScript, các framework và các mẫu thiết kế. Đồng thời, chúng em cũng học được cách làm việc với nền tảng quản lý phiên bản mã nguồn là Git và rất nhiều các hệ thống hỗ trợ khác. Đây đều là những kinh nghiệm vô cùng quý báu là những hành trang không thể thiếu giúp tôi tiếp tục theo đuổi con đường phát triển phần mềm trong tương lai.

Tuy nhiên, do thời gian và khả năng có hạn nên đề tài vẫn chưa hoàn toàn hoàn thiện sản phẩm, với một vài chức năng chưa hoàn thành như:

* Mô hình chưa kết hợp được nhiều phương pháp khuyến nghị
* Chức năng chưa tối ưu, có thể gây ra lỗi nhỏ
* Thiết kế trang web chưa đạt được trải nghiệm người dùng tốt

**Hướng phát triển:**

Trong tương lai, chúng em muốn tiếp tục phát triển hệ thống này để bổ sung thêm các tính năng, công nghệ mới như sau:

* Tối ưu hóa thuật toán khuyến nghị: Tiếp tục nghiên cứu và phát triển thuật toán để cải thiện độ chính xác và sự linh hoạt.
* Kết hợp nhiều phương pháp khuyến nghị khác nhau để cải thiện độ chính xác, nâng cao trải nhiệm người dùng.
* Triên khai được hệ thống mà có thể thay thế và quản lý mô hình dễ dàng
* Phát triển thêm được nhiều chức năng mới, tăng trải nghiệm người dùng

Chúng em rất mong nhận được những ý kiến góp ý của thầy cô trong Hội đồng để sản phẩm được hoàn thiện hơn và được ứng dụng vào thực tế.

Chúng em xin chân thành cảm ơn!

# TÀI LIỆU THAM KHẢO

1. "Lý thuyết trí tuệ nhân tạo, " [Online]. Available: https://aws.amazon.com/vi/what-is/artificial-intelligence [Accessed 20 01 2024].
2. "Tổng quan về Recommender System," [Online]. Available: <https://viblo.asia/p/tong-quan-ve-recommender-system-recommender-system-co-ban-phan-1-924lJGBb5PM> [Accessed 22 01 2024].
3. "Content-based Filtering RS," [Online]. Available: https://viblo.asia/p/xay-dung-content-based-filtering-rs-recommender-system-co-ban-phan-2-bWrZnVovZxw [Accessed 22 01 2024].
4. "Collaborative Filtering RS," [Online]. Available: <https://viblo.asia/p/xay-dung-collaborative-filtering-rs-recommender-system-co-ban-phan-3-Az45bMqolxY> [Accessed 22 01 2024].
5. "JavaScript (ngôn ngữ lập trình)," [Online]. Available: https://vi.wikipedia.org/wiki/JavaScript (ngôn\_ngữ\_lập\_trình). [Accessed 28 01 2024].
6. "Python (ngôn ngữ lập trình)," [Online]. Available: https://vi.wikipedia.org/wiki/Python\_(ngôn\_ngữ\_lập\_trình). [Accessed 16 01 2024].
7. "FastAPI (web framework)" [Online]. Available: https://fastapi.tiangolo.com/. [Accessed 03 02 2024].
8. "VueJs (web framework)" [Online]. Available: https://en.wikipedia.org/wiki/Vue.js. [Accessed 13 01 2024].
9. "Amazon S3" [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html [Accessed 27 02 2024].
10. "MongoDB (Database)" [Online]. Available: https://www.mongodb.com/docs/manual. [Accessed 20 01 2024].
11. "Microsoft SQL server (Database)" [Online]. Available:   
    https://learn.microsoft.com/en-us/sql/sql-server/what-s-new-in-sql-server-2022?view=sql-server-ver16 [Accessed 13 01 2024].