

ĐẠI HỌC BÁCH KHOA HÀ NỘI

PROJECT 3

Ứng dụng phát nhạc đa nền tảng

NGUYỄN VĂN LONG

Long.NV215610@sis.hust.edu.vn

Ngành Công nghệ thông tin

Giảng viên hướng dẫn: TS. Đỗ Tiến Dũng

Khoa: Kỹ thuật Máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 12/2024

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	5
1.1 Đặt vấn đề	5
1.2 Mục tiêu và phạm vi đề tài	5
1.3 Định hướng giải pháp.....	5
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	6
2.1 Khảo sát hiện trạng.....	6
2.1.1 Khảo sát từ các ứng dụng tương tự	6
2.1.2 Kết luận	6
2.2 Tổng quan chức năng	6
2.3 Quy trình nghiệp vụ	7
2.3.1 Quy trình nghiệp vụ Tìm kiếm bài hát	7
2.3.2 Quy trình nghiệp vụ Gợi ý nhạc	7
2.3.3 Quy trình nghiệp vụ Quản lý lịch sử	7
2.4 Đặc tả chức năng.....	7
2.4.1 Chức năng Xem thông tin bài hát.....	7
2.4.2 Chức năng Đề xuất nhạc	7
2.4.3 Chức năng Tìm kiếm bài hát	7
2.4.4 Chức năng Quản lý tài khoản	7
2.4.5 Chức năng Phát nhạc	7
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG	8
3.1 Flask	8
3.2 React.....	8
3.3 MongoDB.....	8
3.4 Scikit-learn	8
3.5 YouTube API	8
CHƯƠNG 4. THIẾT KẾ VÀ TRIỂN KHAI.....	9
4.1 Thiết kế kiến trúc hệ thống	9
4.1.1 Tổng quan kiến trúc.....	9
4.1.2 Thành phần hệ thống	9
4.2 Thiết kế giao diện.....	9
4.3 Thiết kế cơ sở dữ liệu.....	11
4.4 Triển khai hệ thống	11

4.4.1	Công cụ và thư viện sử dụng.....	11
4.4.2	Quy trình triển khai	11
4.5	Kết quả đạt được	11
CHƯƠNG 5. GIẢI PHÁP VÀ ĐÓNG GÓP.....		12
5.1	Giải pháp kỹ thuật áp dụng	12
5.1.1	Kỹ thuật clustering K-Means	12
5.1.2	Cosine Similarity	12
5.1.3	Tích hợp API YouTube.....	12
5.2	Đóng góp nổi bật.....	12
5.2.1	Trải nghiệm người dùng.....	12
5.2.2	Tính năng cá nhân hóa	12
5.2.3	Khả năng mở rộng.....	12
5.2.4	Hiệu suất và độ chính xác	12
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		13
6.1	Kết luận	13
6.2	Hướng phát triển	13
TÀI LIỆU THAM KHẢO		14

DANH MỤC HÌNH VẼ

Hình 2.1 Biểu đồ usecase tổng quan	6
Hình 4.1 Giao diện Trang chủ.....	9
Hình 4.2 Giao diện Admin	10
Hình 4.3 Giao diện User.....	10
Hình 4.4 Sơ đồ Database.....	11

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong thời đại số hóa, nhu cầu chia sẻ thông tin và cá nhân hóa trải nghiệm ngày càng tăng, đặc biệt trong lĩnh vực âm nhạc. Hệ thống gợi ý nhạc trở thành công cụ quan trọng, giúp người dùng tìm kiếm và thưởng thức các bài hát phù hợp với sở thích, nâng cao trải nghiệm và tiết kiệm thời gian. Hệ thống Music Recommendation tích hợp các tính năng như quản lý lịch sử nghe nhạc, khám phá bài hát mới và tìm kiếm bài hát nhanh chóng. Với sự hỗ trợ từ YouTube, hệ thống mang đến trải nghiệm âm nhạc phong phú, đáp ứng hiệu quả nhu cầu cá nhân hóa của người dùng.

1.2 Mục tiêu và phạm vi đề tài

Mục tiêu chính của đề tài là phát triển một hệ thống Music Recommendation, mang đến trải nghiệm nghe nhạc thông minh, tiện lợi và cá nhân hóa cho người dùng. Hệ thống sẽ thực hiện các chức năng chính như:

- Gợi ý nhạc cá nhân hóa: Đưa ra bài hát phù hợp với thói quen và sở thích người dùng.
- Lưu trữ lịch sử nghe nhạc: Quản lý lịch sử nghe, hỗ trợ truy cập lại và cải thiện gợi ý.
- Tìm kiếm bài hát: Tìm kiếm nhanh chóng bài hát.
- Phát nhạc qua YouTube: Tích hợp phát nhạc trực tiếp từ YouTube.

Phạm vi thực hiện bao gồm xây dựng backend bằng Flask và MongoDB để xử lý dữ liệu nhanh chóng và ổn định, cùng frontend sử dụng React kết hợp Material-UI, mang lại giao diện hiện đại, trực quan. Hệ thống không chỉ hướng đến việc cá nhân hóa trải nghiệm âm nhạc mà còn đảm bảo hiệu suất, khả năng mở rộng và tích hợp trong tương lai, đáp ứng tốt nhu cầu nghe nhạc trực tuyến ngày càng đa dạng.

1.3 Định hướng giải pháp

Hệ thống Music Recommendation được xây dựng dựa trên các kỹ thuật và thuật toán hiện đại để cung cấp trải nghiệm nghe nhạc cá nhân hóa. Cụ thể:

- Clustering: Áp dụng thuật toán K-Means để nhóm các bài hát có đặc điểm tương tự, giúp sắp xếp và phân loại bài hát hiệu quả.
- Cosine Similarity: Đo lường độ tương đồng giữa các vector đặc trưng như thể loại hoặc nhịp điệu, từ đó đưa ra các gợi ý bài hát phù hợp hơn.
- API YouTube: Tích hợp tìm kiếm và phát nhạc trực tiếp từ YouTube, đảm bảo nguồn nhạc phong phú và cập nhật.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

2.1.1 Khảo sát từ các ứng dụng tương tự

Các hệ thống gợi ý nhạc hiện nay như Spotify và YouTube Music đã và đang sử dụng các thuật toán tiên tiến để mang đến trải nghiệm người dùng tối ưu. Các ứng dụng này tập trung vào việc gợi ý bài hát dựa trên lịch sử nghe nhạc, tìm kiếm bài hát nhanh chóng, và cung cấp các playlist theo ngữ cảnh.

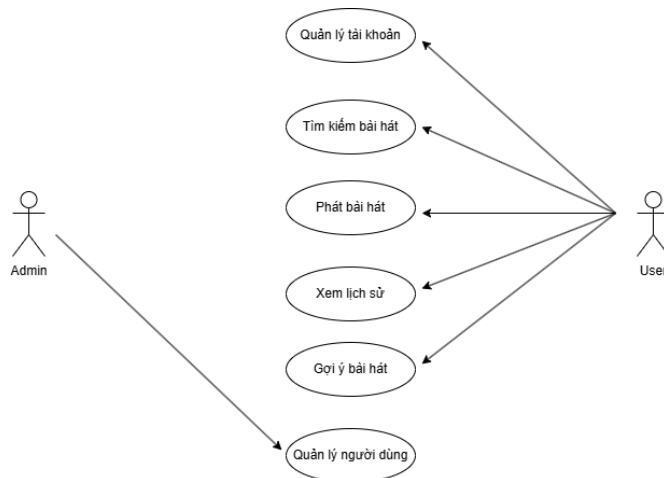
- Spotify: Nổi bật với tính năng Daily Mix, Discover Weekly và khả năng cá nhân hóa mạnh mẽ.
- YouTube Music: Cung cấp tính năng gợi ý dựa trên lịch sử xem và xu hướng.

2.1.2 Kết luận

Khảo sát cho thấy các tính năng như tìm kiếm bài hát, quản lý lịch sử, và đề xuất dựa trên thói quen là cần thiết. Việc triển khai một hệ thống gợi ý nhạc tích hợp các tính năng này sẽ mang lại giá trị thực tế cao.

2.2 Tổng quan chức năng

Biểu đồ usecase tổng quan:



Hình 2.1 Biểu đồ usecase tổng quan

Hệ thống Music Recommendation sẽ cung cấp các chức năng chính sau:

- Quản lý tài khoản người dùng: Đăng ký, đăng nhập, và phân quyền (user/admin).
- Xem và tìm kiếm bài hát: Cho phép người dùng tra cứu thông tin bài hát nhanh chóng.
- Gợi ý nhạc cá nhân hóa: Đề xuất bài hát dựa trên thói quen nghe nhạc.
- Quản lý lịch sử nghe nhạc: Lưu trữ và hiển thị lịch sử nghe nhạc của người dùng.
- Phát nhạc qua YouTube: Tích hợp API YouTube để phát nhạc trực tiếp.

2.3 Quy trình nghiệp vụ

2.3.1 Quy trình nghiệp vụ Tìm kiếm bài hát

- Người dùng nhập tên bài hát vào ô tìm kiếm.
- Hệ thống tìm kiếm trong cơ sở dữ liệu và trả về kết quả tương ứng.
- Nếu không tìm thấy, hệ thống sẽ thông báo cho người dùng.

2.3.2 Quy trình nghiệp vụ Gợi ý nhạc

- Hệ thống phân tích lịch sử nghe nhạc của người dùng.
- Áp dụng thuật toán K-Means để nhóm các bài hát có đặc điểm tương tự.
- Gợi ý danh sách bài hát phù hợp với sở thích của người dùng.

2.3.3 Quy trình nghiệp vụ Quản lý lịch sử

- Khi người dùng phát một bài hát, hệ thống tự động lưu vào lịch sử.
- Người dùng có thể xem các bài hát trong lịch sử nghe nhạc.

2.4 Đặc tả chức năng

2.4.1 Chức năng Xem thông tin bài hát

- Mô tả: Hiển thị thông tin chi tiết về bài hát bao gồm tên, nghệ sĩ.
- Input: Tên bài hát.
- Output: Thông tin bài hát dưới dạng danh sách.

2.4.2 Chức năng Đề xuất nhạc

- Mô tả: Gợi ý danh sách bài hát dựa trên thói quen nghe nhạc.
- Input: Dữ liệu lịch sử nghe nhạc của người dùng.
- Output: Danh sách bài hát được đề xuất.

2.4.3 Chức năng Tìm kiếm bài hát

- Mô tả: Tìm kiếm bài hát trong cơ sở dữ liệu và trên YouTube.
- Input: Tên bài hát.
- Output: Danh sách kết quả tìm kiếm.

2.4.4 Chức năng Quản lý tài khoản

- Mô tả: Đăng ký, đăng nhập, và phân quyền người dùng.
- Input: Thông tin tài khoản (username, password).
- Output: Thông báo đăng nhập thành công hoặc thất bại.

2.4.5 Chức năng Phát nhạc

- Mô tả: Phát nhạc trực tiếp qua YouTube.
- Input: ID bài hát từ YouTube.
- Output: Trình phát nhạc nhúng từ YouTube.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 Flask

- Giới thiệu: Flask là một microframework Python nhẹ, linh hoạt, hỗ trợ phát triển ứng dụng web nhanh chóng.
- Mục đích: Xây dựng backend, xử lý logic nghiệp vụ, giao tiếp với MongoDB và cung cấp API cho frontend.
- Ưu điểm: Nhẹ, dễ tích hợp, linh hoạt.
- Nhược điểm: Thiếu tính năng mặc định, cần quản lý nhiều cho dự án lớn.

3.2 React

- Giới thiệu: React là thư viện JavaScript mạnh mẽ, hỗ trợ phát triển giao diện người dùng cho các ứng dụng web đơn trang (SPA).
- Mục đích: Xây dựng frontend, cung cấp giao diện trực quan, phản hồi nhanh.
- Ưu điểm: Nhanh nhờ Virtual DOM, tái sử dụng mã, cộng đồng lớn.
- Nhược điểm: Đòi hỏi kiến thức JavaScript hiện đại, thời gian học tập cao.

3.3 MongoDB

- Giới thiệu: MongoDB là cơ sở dữ liệu NoSQL linh hoạt, lưu trữ dữ liệu JSON/BSON, phù hợp với ứng dụng xử lý dữ liệu lớn.
- Mục đích: Lưu trữ thông tin người dùng, lịch sử nghe nhạc, bài hát.
- Ưu điểm: Linh hoạt, mở rộng tốt, dễ tích hợp với Python.
- Nhược điểm: Hạn chế với giao dịch phức tạp, cần cấu hình để đảm bảo bảo mật.

3.4 Scikit-learn

- Giới thiệu: Thư viện Python mạnh mẽ cho học máy, hỗ trợ thuật toán phân loại, hồi quy và clustering.
- Mục đích: Triển khai K-Means clustering để nhóm bài hát có đặc điểm tương tự.
- Ưu điểm: Đa dạng thuật toán, dễ tích hợp với NumPy và Pandas.
- Nhược điểm: Hạn chế với deep learning và dữ liệu rất lớn.

3.5 YouTube API

- Giới thiệu: API cho phép truy xuất thông tin và phát video từ YouTube.
- Mục đích: Tìm kiếm và phát nhạc dựa trên gợi ý của hệ thống.
- Ưu điểm: Truy cập dữ liệu lớn, dễ tích hợp.
- Nhược điểm: Giới hạn truy vấn miễn phí, cần cấu hình API key.

CHƯƠNG 4. THIẾT KẾ VÀ TRIỂN KHAI

4.1 Thiết kế kiến trúc hệ thống

4.1.1 Tổng quan kiến trúc

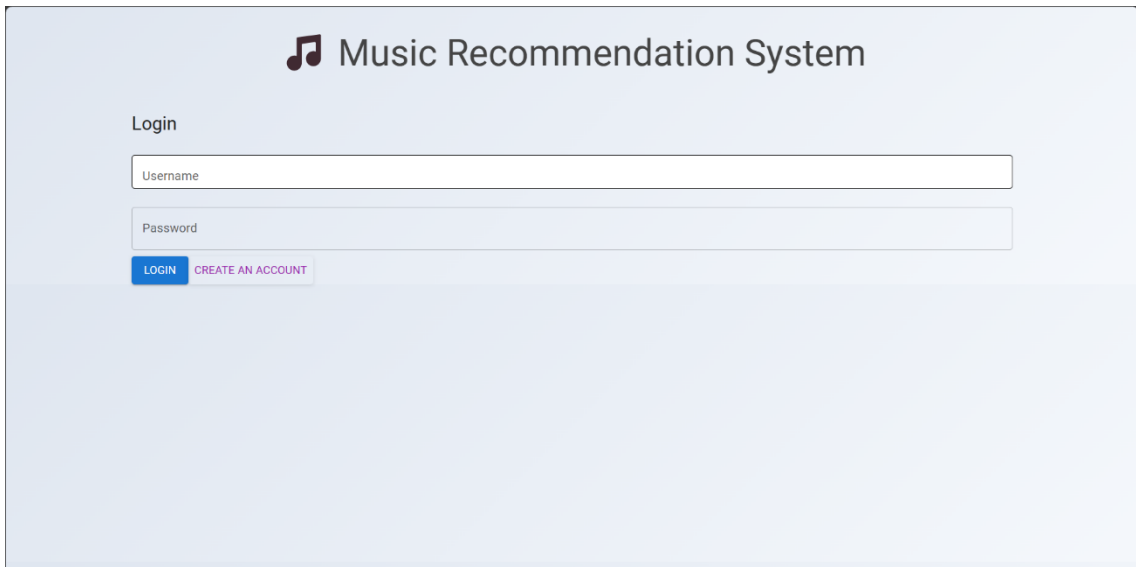
Hệ thống Music Recommendation được thiết kế theo mô hình client-server. Frontend giao tiếp với backend thông qua API RESTful, backend xử lý logic nghiệp vụ và giao tiếp với cơ sở dữ liệu MongoDB để lưu trữ dữ liệu.

4.1.2 Thành phần hệ thống

- Frontend: Xây dựng bằng React, cung cấp giao diện người dùng để thực hiện các chức năng như tìm kiếm, xem lịch sử, và gợi ý nhạc.
- Backend: Xây dựng bằng Flask, chịu trách nhiệm xử lý yêu cầu từ frontend, áp dụng thuật toán gợi ý và giao tiếp với cơ sở dữ liệu.
- Cơ sở dữ liệu: Sử dụng MongoDB để lưu trữ thông tin người dùng, bài hát, và lịch sử nghe nhạc.
- YouTube API: Cung cấp tính năng tìm kiếm và phát nhạc.

4.2 Thiết kế giao diện

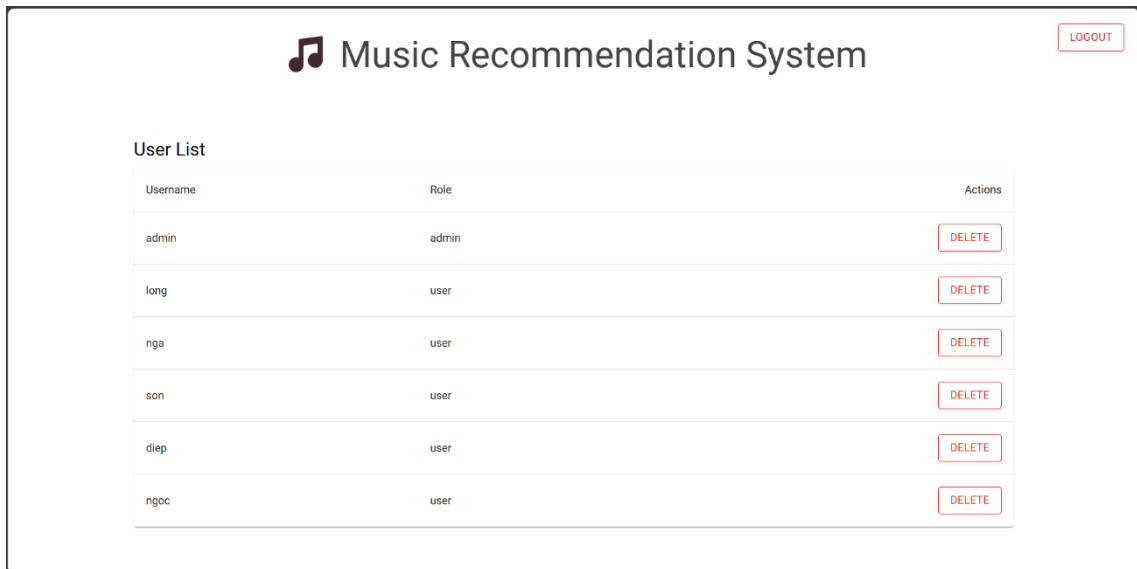
Giao diện Trang chủ:



The screenshot shows the login page of the Music Recommendation System. At the top, there is a header with a music note icon and the text "Music Recommendation System". Below the header, the word "Login" is displayed. There are two input fields: "Username" and "Password". Below these fields are two buttons: "LOGIN" (in blue) and "CREATE AN ACCOUNT" (in purple).

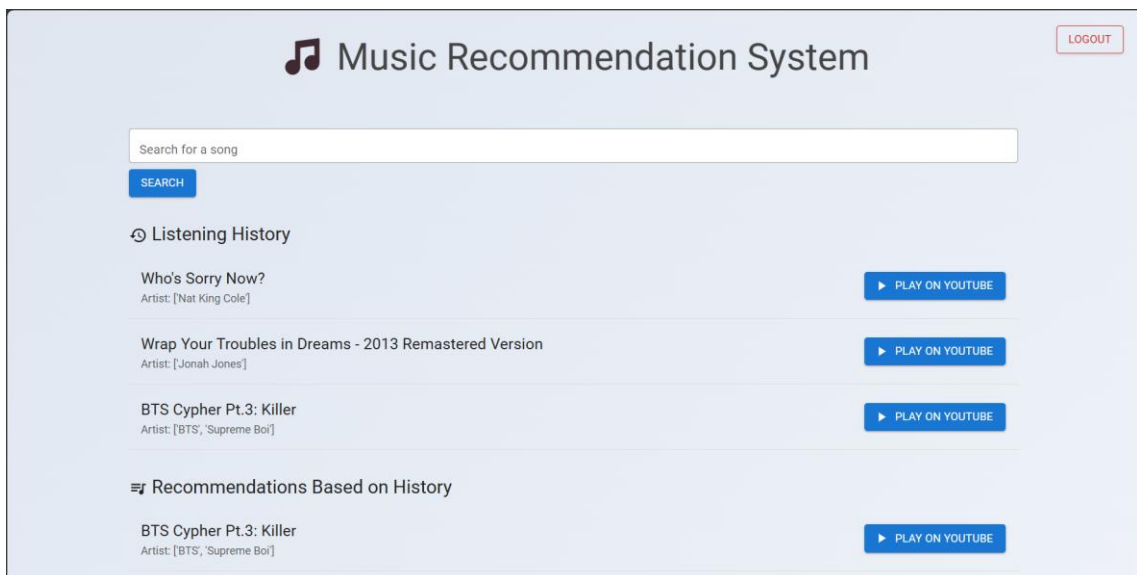
Hình 4.1 Giao diện Trang chủ

Giao diện Admin:



Hình 4.2 Giao diện Admin

Giao diện Users:



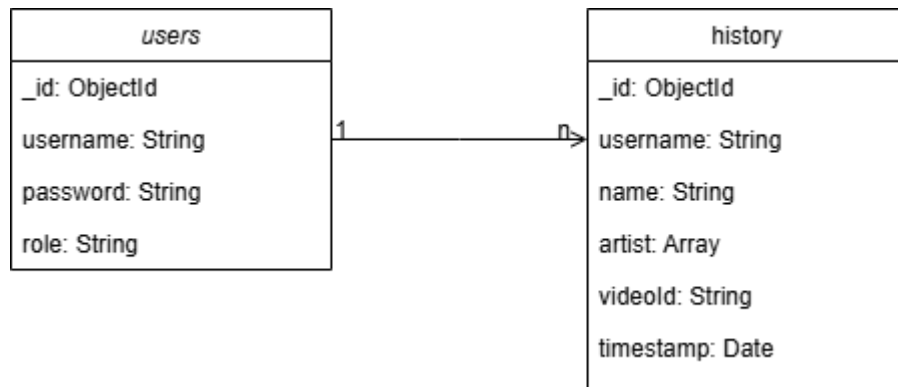
Hình 4.3 Giao diện User

Giao diện được thiết kế trực quan và thân thiện với người dùng. Các thành phần chính bao gồm:

- Trang chủ: Hiển thị thông tin tổng quan và các chức năng chính.
- Tìm kiếm: Cho phép người dùng nhập từ khóa và hiển thị kết quả.
- Gợi ý nhạc: Hiển thị danh sách bài hát được đề xuất dựa trên lịch sử nghe nhạc.
- Lịch sử nghe nhạc: Hiển thị các bài hát mà người dùng đã nghe trước đó.
- Quản lý tài khoản: Đăng ký và đăng nhập.

4.3 Thiết kế cơ sở dữ liệu

MongoDB Schema Diagram:



Hình 4.4 Sơ đồ Database

Cơ sở dữ liệu được thiết kế với các bảng (collection) chính:

- **users:** Lưu trữ thông tin người dùng (`_id`, `username`, `password`, `role`).
- **history:** Lưu trữ lịch sử nghe nhạc của người dùng (`_id`, `username`, `song`, `name`, `timestamp`).
- **data.csv:** Lưu trữ thông tin bài hát (`name`, `artist`, `album`, `year`, `features`).

4.4 Triển khai hệ thống

4.4.1 Công cụ và thư viện sử dụng

- **Flask:** Xây dựng API backend.
- **React:** Phát triển giao diện frontend.
- **MongoDB:** Quản lý dữ liệu.
- **Scikit-learn:** Áp dụng thuật toán học máy.
- **YouTube API:** Tìm kiếm và phát nhạc.

4.4.2 Quy trình triển khai

- **Bước 1:** Xây dựng cơ sở dữ liệu MongoDB và cấu hình kết nối.
- **Bước 2:** Phát triển backend với Flask, tích hợp thuật toán gợi ý và API YouTube.
- **Bước 3:** Phát triển frontend với React và Material-UI.
- **Bước 4:** Kiểm thử các chức năng và tối ưu hóa hiệu suất.

4.5 Kết quả đạt được

Hệ thống đã được triển khai với các tính năng chính hoạt động ổn định. Giao diện người dùng thân thiện, hệ thống gợi ý nhạc chính xác và thời gian phản hồi nhanh chóng.

CHƯƠNG 5. GIẢI PHÁP VÀ ĐÓNG GÓP

5.1 Giải pháp kỹ thuật áp dụng

5.1.1 Kỹ thuật clustering K-Means

Hệ thống áp dụng thuật toán K-Means để nhóm các bài hát thành các cụm dựa trên đặc điểm âm nhạc (valence, danceability, energy, acousticness, v.v.). Điều này giúp hệ thống dễ dàng đề xuất các bài hát tương tự cho người dùng.

5.1.2 Cosine Similarity

Đo độ tương đồng giữa các vector đặc điểm bài hát giúp xác định mức độ phù hợp giữa bài hát đã nghe và các bài hát trong cơ sở dữ liệu. Từ đó, hệ thống chọn ra các bài hát có điểm tương đồng cao nhất để gợi ý.

5.1.3 Tích hợp API YouTube

API YouTube được sử dụng để tìm kiếm và phát nhạc. Mỗi bài hát gợi ý sẽ được liên kết với một video trên YouTube, giúp người dùng phát nhạc trực tiếp mà không cần rời khỏi giao diện.

5.2 Đóng góp nổi bật

5.2.1 Trải nghiệm người dùng

Hệ thống cung cấp giao diện trực quan, thân thiện với người dùng. Các tính năng như lịch sử nghe nhạc, tìm kiếm nhanh, và phát nhạc trực tiếp qua YouTube giúp nâng cao trải nghiệm người dùng.

5.2.2 Tính năng cá nhân hóa

Thuật toán K-Means và Cosine Similarity giúp cá nhân hóa gợi ý nhạc dựa trên sở thích và lịch sử nghe nhạc của từng người dùng.

5.2.3 Khả năng mở rộng

Hệ thống được thiết kế linh hoạt, dễ dàng mở rộng và tích hợp thêm các tính năng mới như hỗ trợ đa nền tảng, tích hợp thêm các nguồn phát nhạc khác ngoài YouTube.

5.2.4 Hiệu suất và độ chính xác

Việc sử dụng Scikit-learn và các kỹ thuật tối ưu hóa giúp hệ thống đạt được độ chính xác cao trong việc gợi ý bài hát, đồng thời đảm bảo thời gian phản hồi nhanh chóng.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Dự án Music Recommendation đã hoàn thiện các chức năng chính như quản lý tài khoản, tìm kiếm bài hát, gợi ý nhạc cá nhân hóa và phát nhạc qua YouTube. Hệ thống đáp ứng tốt nhu cầu người dùng, mang lại trải nghiệm hiệu quả nhờ tích hợp API mạnh mẽ và thuật toán clustering, Cosine Similarity. Việc ứng dụng các thuật toán học máy và kết nối với nền tảng YouTube đã chứng minh tính khả thi và tiềm năng mở rộng của hệ thống.

6.2 Hướng phát triển

Hệ thống có thể được nâng cấp với các hướng sau:

- Cải thiện thuật toán gợi ý: Áp dụng học sâu (deep learning) để nâng cao độ chính xác.
- Hỗ trợ đa nền tảng: Phát triển ứng dụng trên Android và iOS.
- Mở rộng nguồn dữ liệu: Tích hợp với Spotify, Apple Music, và SoundCloud.
- Tăng cường bảo mật: Mã hóa dữ liệu và quản lý quyền truy cập tốt hơn.
- Phân tích nâng cao: Xây dựng công cụ phân tích thói quen nghe nhạc.
- Cải thiện giao diện: Tích hợp playlist cá nhân, chế độ offline, và chia sẻ qua mạng xã hội.

Những cải tiến này sẽ giúp hệ thống trở thành một nền tảng toàn diện, đáp ứng tốt hơn nhu cầu người dùng trong tương lai.

TÀI LIỆU THAM KHẢO

- [1] [1] Vatsal Mavani, "Music Recommendation System using Spotify Dataset," Kaggle, [Online]. Available: Kaggle Notebook. [Accessed: Dec. 2024].
- [2] Spotify, "Open Spotify," [Online]. [Accessed: Dec. 2024].
- [3] YouTube, "YouTube Platform," [Online]. [Accessed: Dec. 2024].