|  |
| --- |
| khung doi**ĐẠI HỌC DUY TÂN**  TRƯỜNG KHOA HỌC MÁY TÍNH  KHOA KỸ THUẬT MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG  *🙦*🕮🙤 |
| C:\Users\Administrator\Desktop\logoDT.gif |
| **ĐỒ CÁ NHÂN/ NHÓM**  **MÔN:** ĐỒ ÁN CDIO 4 – **MÃ MÔN:** DS447 |
| **Đề tài:** Ứng dụng nghe Nhạc trực tuyến và Hệ thống đề xuất nhạc sử dụng pLSA |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Nhóm SV thực hiện:** (Nhóm 3) | | | | | | **Mã sinh viên** | | **Họ tên sinh viên** | | | **Điểm** | | 27211921933 | | Trần Nguyễn Hoàng Huy | | |  | | 27211946069 | | Phạm Vũ Long | | |  | | 28211445639 | | Nguyễn Tấn Vĩnh | | |  | | 27211045674 | | Vũ Minh Tiến | | |  | | 27211201978 | | Lê Kim Quang | | |  | | **Lớp môn học:** DS 447 A | | | |  | | | **GVHD:** | | TS. Đặng Việt Hùng  Đà Nẵng, 10/2024 | | | |

# MỤC LỤC

[**CHƯƠNG 1: Ý TƯỞNG THỰC HIỆN** 3](#_Toc179446044)

[1.1. Chọn ý tưởng 3](#_Toc179446045)

[1.2. Tổng quan các vấn đề liên quan đến ý tưởng được chọn 3](#_Toc179446046)

[1.2.1 Phân tích khả năng rời bỏ/ngưng sử dụng dịch vụ telecom 4](#_Toc179446047)

[1.2.2 Phân tích hệ thống đề xuất phim 4](#_Toc179446048)

[1.2.3 Xây dựng hệ thống đề xuất nhạc 4](#_Toc179446049)

[**CHƯƠNG 2: THIẾT KẾ (DESIGN)** 5](#_Toc179446050)

[2.1. Design kiến trúc 5](#_Toc179446051)

[2.2.1. Database 5](#_Toc179446052)

[2.2.2. Context Diagram 7](#_Toc179446053)

[2.2.3. Sơ đồ DFD – Data Flow Diagram 10](#_Toc179446054)

[2.2.4. DFD Đề Xuất 12](#_Toc179446055)

[2.2. Design UI 13](#_Toc179446056)

[2.2.1. base.html 13](#_Toc179446057)

[2.2.2. Trang chủ - index.html 17](#_Toc179446058)

[2.2.3. Trang Album detail – Chi tiết Album 18](#_Toc179446059)

[2.2.4. Trang Lyrics – Lời bài hát 19](#_Toc179446060)

[2.2.5. Trang Create Playlist – Tạo danh sách nhạc 19](#_Toc179446061)

[2.2.6. Trang Playlist Detail – Chi tiết danh sách nhạc 20](#_Toc179446062)

[2.2.7. Trang Favorites – Danh sách nhạc yêu thích 21](#_Toc179446063)

[2.2.8. Trang Login 21](#_Toc179446064)

[2.2.9. Trang Signup 22](#_Toc179446065)

[2.3. Design model: pLSA 23](#_Toc179446066)

[2.3.1. Giới thiệu chung về pLSA 23](#_Toc179446067)

[2.3.2. Nguyên lí hoạt động của pLSA 23](#_Toc179446068)

[2.3.3. Ứng dụng pLSA trong hệ thống đề xuất 26](#_Toc179446069)

[2.3.4. Tokenizer trong pLSA 27](#_Toc179446070)

[2.3.5. Ưu điểm của pLSA so với các phương pháp khác 28](#_Toc179446071)

[2.3.6. Hạn chế của pLSA 28](#_Toc179446072)

[2.3.7. Kiến trúc model pLSA 29](#_Toc179446073)

[**CHƯƠNG 3: TRIỂN KHAI VÀ CÀI ĐẶT (IMPLEMENT)** 39](#_Toc179446074)

[3.1. Kết quả Implement kiến trúc 39](#_Toc179446075)

[3.2. Kết quả Implement UI 41](#_Toc179446076)

[3.2.1. Main (Chưa login) 41](#_Toc179446077)

[3.2.2. Giao diện login 41](#_Toc179446078)

[3.2.3. Main (Đã login) 42](#_Toc179446079)

[3.2.4. Giao diện đăng kí tài khoản 42](#_Toc179446080)

[3.2.5. Giao diện một album 43](#_Toc179446081)

[3.2.6. Giao diện một playlist người dùng tự tạo 43](#_Toc179446082)

[3.2.7. Giao diện tạo playlist 44](#_Toc179446083)

[3.3. Kết quả Implement model 45](#_Toc179446084)

[**CHƯƠNG 4: VẬN HÀNH (OPERATE)** 46](#_Toc179446085)

[4.1. Kết quả 46](#_Toc179446086)

[4.2. Thảo luận 48](#_Toc179446087)

[**KẾT LUẬN** 49](#_Toc179446088)

[Ưu điểm 49](#_Toc179446089)

[Nhược điểm 50](#_Toc179446090)

[Hướng phát triển 51](#_Toc179446091)

[**TÀI LIỆU THAM KHẢO** 53](#_Toc179446092)

# CHƯƠNG 1: Ý TƯỞNG THỰC HIỆN

## Chọn ý tưởng

Ban đầu, chúng em đã cân nhắc và thảo luận về ba đề tài nghiên cứu tiềm năng trong lĩnh vực ứng dụng công nghệ thông tin và trí tuệ nhân tạo. Ba đề tài được lựa chọn gồm: phân tích khả năng rời bỏ/ngưng sử dụng dịch vụ telecom, xây dựng hệ thống đề xuất phim, và xây dựng hệ thống đề xuất nhạc.

Với đề tài **phân tích khả năng rời bỏ/ngưng sử dụng dịch vụ telecom**, chúng em tập trung vào việc phân tích dữ liệu khách hàng của các công ty viễn thông nhằm dự đoán khả năng rời bỏ dịch vụ của họ. Thông qua các thuật toán học máy, hệ thống có thể dự báo khách hàng nào có nguy cơ cao ngưng sử dụng dịch vụ, từ đó giúp doanh nghiệp đưa ra các chiến lược giữ chân khách hàng hiệu quả. Đây là một đề tài rất thực tiễn và có tiềm năng giúp doanh nghiệp giảm chi phí quảng cáo và duy trì khách hàng trung thành. Tuy nhiên, thách thức lớn nhất của đề tài này là yêu cầu dữ liệu chi tiết và chính xác về lịch sử giao dịch, tương tác của khách hàng với dịch vụ, điều mà không phải lúc nào cũng dễ thu thập và xử lý. Hơn nữa, các yếu tố ngoài tầm kiểm soát như chính sách thị trường hoặc sự cạnh tranh cũng có thể ảnh hưởng đến hiệu quả dự đoán của mô hình.

Đề tài thứ hai là **xây dựng hệ thống đề xuất phim**, tập trung vào việc phân tích dữ liệu xem phim của người dùng để đưa ra các gợi ý phù hợp với sở thích cá nhân. Đề xuất phim dựa trên các thuật toán gợi ý nhằm giúp người dùng dễ dàng tìm kiếm phim yêu thích mà không cần tốn quá nhiều thời gian lựa chọn. Mô hình đề xuất có khả năng mang lại trải nghiệm giải trí tốt hơn cho người dùng, đồng thời giúp các nền tảng phim trực tuyến tăng tỷ lệ tương tác. Tuy nhiên, hệ thống này yêu cầu phải xử lý một lượng lớn dữ liệu từ lịch sử xem phim, đánh giá, và thông tin về phim, làm cho quá trình triển khai và xây dựng mô hình trở nên phức tạp.

Cuối cùng, chúng em đã xem xét **xây dựng hệ thống đề xuất nhạc**, một đề tài tương tự với hệ thống đề xuất phim nhưng tập trung vào lĩnh vực âm nhạc. Dự án này nhằm hỗ trợ người dùng các nền tảng âm nhạc trực tuyến tìm kiếm và khám phá những bài hát mới phù hợp với sở thích cá nhân thông qua phân tích dữ liệu nghe nhạc của họ. Đề tài này có ưu điểm là dữ liệu âm nhạc dễ thu thập và trực quan hơn so với dữ liệu phim hoặc mua sắm, đồng thời cũng có tiềm năng ứng dụng cao trong việc mang lại trải nghiệm cá nhân hóa và nâng cao thời gian sử dụng dịch vụ cho người dùng.

## 1.2. Tổng quan các vấn đề liên quan đến ý tưởng được chọn

### 1.2.1 Phân tích khả năng rời bỏ/ngưng sử dụng dịch vụ telecom

* **Ưu điểm**:
  + Giúp doanh nghiệp viễn thông dự đoán khách hàng có nguy cơ rời bỏ.
  + Giúp doanh nghiệp triển khai các chiến lược giữ chân khách hàng, tiết kiệm chi phí.
* **Nhược điểm**:
  + Đòi hỏi dữ liệu chi tiết và có độ chính xác cao.
  + Khó thu thập dữ liệu lịch sử giao dịch và tương tác khách hàng.
  + Môi trường viễn thông biến động mạnh, khiến mô hình không đảm bảo độ chính xác cao.

### 1.2.2 Phân tích hệ thống đề xuất phim

* **Ưu điểm**:
  + Giúp các nền tảng phim trực tuyến đề xuất phim theo sở thích cá nhân của người dùng.
  + Nâng cao trải nghiệm người dùng và gia tăng thời gian sử dụng dịch vụ.
* **Nhược điểm**:
  + Phân tích thói quen xem phim đòi hỏi xử lý dữ liệu từ nhiều nguồn khác nhau.
  + Phức tạp trong việc xác định sở thích người dùng do các yếu tố ngoại cảnh như xu hướng thị trường hoặc thay đổi thị hiếu.

### 1.2.3 Xây dựng hệ thống đề xuất nhạc

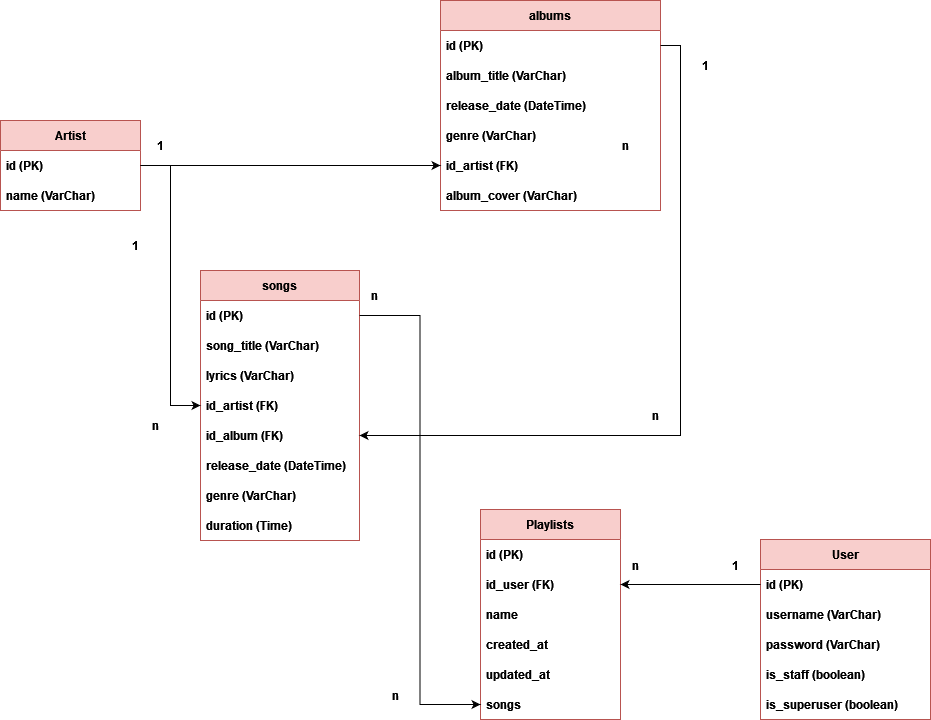
* **Ưu điểm**:
  + Hệ thống giúp cá nhân hóa trải nghiệm âm nhạc, gợi ý các bài hát theo sở thích.
  + Nhạc có tính cá nhân hóa cao, nên dễ dàng trong việc thu thập dữ liệu nghe nhạc.
  + Dễ ứng dụng và triển khai rộng rãi, mang lại giá trị cho cả người dùng và nền tảng âm nhạc.
* **Nhược điểm**:
  + Hệ thống phải linh hoạt do xu hướng âm nhạc thay đổi liên tục.
  + Phải xử lý dữ liệu lớn và phức tạp để đưa ra gợi ý chính xác.

→ **Lựa chọn đề tài 3** vì tính ứng dụng thực tiễn cao và khả năng phát triển công nghệ tốt.

# CHƯƠNG 2: THIẾT KẾ (DESIGN)

## 2.1. Design kiến trúc

### 2.2.1. Database



**Các Thực Thể và Thuộc Tính:**

1. **Artist (Nghệ sĩ)**:

* **id (PK):** Khóa chính
* **name (VarChar):** Tên của nghệ sĩ

1. **Albums (Album)**:

* **id (PK):** Khóa chính
* **album\_title (VarChar):** Tên của album
* **release\_date (DateTime):** Ngày phát hành của album
* **genre (VarChar):** Thể loại của album
* **id\_artist (FK):** Khóa ngoại tham chiếu đến nghệ sĩ đã tạo ra album
* **album\_cover (VarChar):** Đường dẫn của ảnh bìa album

1. **Songs (Bài hát)**:
   * **id (PK):** Khóa chính
   * **song\_title (VarChar):** Tên của bài hát
   * **lyrics (VarChar):** Lời bài hát
   * **id\_artist (FK):** Khóa ngoại tham chiếu đến nghệ sĩ biểu diễn bài hát
   * **id\_album (FK):** Khóa ngoại tham chiếu đến album chứa bài hát
   * **release\_date (DateTime):** Ngày phát hành của bài hát
   * **genre (VarChar):** Thể loại của bài hát (có thể thừa hưởng từ album)
   * **duration (Time):** Thời lượng của bài hát
2. **Playlists (Danh sách phát)**:
   * **id (PK):** Khóa chính
   * **id\_user (FK):** Khóa ngoại tham chiếu đến người dùng đã tạo danh sách phát
   * **name (VarChar):** Tên của danh sách phát
   * **created\_at (DateTime):** Thời điểm danh sách phát được tạo
   * **updated\_at (DateTime):** Thời điểm danh sách phát được cập nhật lần cuối
   * songs: Chỉ ra mối quan hệ nhiều-nhiều với thực thể bài hát, tức là một danh sách phát có thể chứa nhiều bài hát, và một bài hát có thể xuất hiện trong nhiều danh sách phát
3. **User (Người dùng)**:
   * **id (PK):** Khóa chính
   * **username (VarChar):** Tên đăng nhập của người dùng
   * **password (VarChar):** Mật khẩu của người dùng
   * **is\_staff (boolean):** Biến boolean cho biết người dùng có quyền quản trị hay không
   * **is\_superuser (boolean):** Biến boolean cho biết người dùng có quyền siêu quản trị hay không

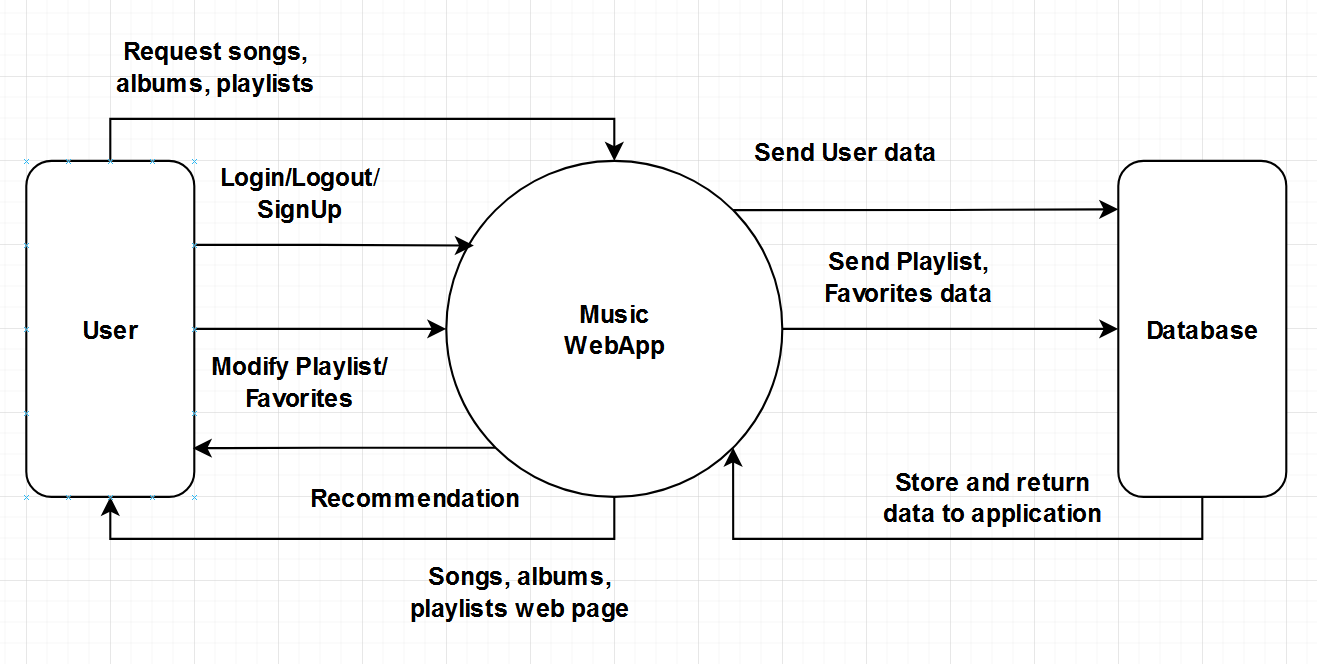
**Mối Quan Hệ:**

* **Một-Nhiều**:
  + Mỗi nghệ sĩ có thể có nhiều album (1 Nghệ sĩ -> n Album)
  + Mỗi nghệ sĩ có thể có nhiều bài hát (1 Nghệ sĩ -> n Bài hát)
  + Mỗi album có thể chứa nhiều bài hát (1 Album -> n Bài hát)
  + Mỗi người dùng có thể có nhiều danh sách phát (1 Người dùng

-> n Danh sách phát)

* **Nhiều-Nhiều**:
  + Một danh sách phát có thể có nhiều bài hát, và một bài hát có thể thuộc nhiều danh sách phát (n Danh sách phát <-> n Bài hát)

### 2.2.2. Context Diagram



Sơ đồ Context Diagram

Đây là phân tích về sơ đồ cấp độ cao giữa ba thực thể: **Người dùng**, **Ứng dụng web âm nhạc**, và **Cơ sở dữ liệu**:

**1. Người dùng:**

* **Yêu cầu bài hát, album, danh sách phát:** Người dùng tương tác với ứng dụng bằng cách yêu cầu các loại phương tiện như bài hát, album và danh sách phát.
* **Đăng nhập/Đăng xuất/Đăng ký:** Người dùng quản lý tài khoản bằng cách đăng nhập, đăng xuất, hoặc đăng ký tài khoản mới.
* **Chỉnh sửa danh sách phát/Yêu thích**: Người dùng có thể chỉnh sửa danh sách phát và thêm bài hát vào mục yêu thích.
* **Đề xuất:** Ứng dụng cung cấp các gợi ý bài hát dựa trên thói quen nghe của người dùng.
* **Trang web hiển thị bài hát, album, danh sách phát**: Người dùng nhận nội dung đã yêu cầu dưới dạng các trang web hiển thị bài hát, album, và danh sách phát.

**2. Ứng dụng web âm nhạc:**

* Ứng dụng web hoạt động như một tầng trung gian xử lý tất cả các yêu cầu từ người dùng và tương tác với cơ sở dữ liệu để lấy hoặc lưu trữ dữ liệu.
* Gửi dữ liệu người dùng đến cơ sở dữ liệu khi người dùng đăng nhập hoặc đăng ký tài khoản mới.
* Gửi dữ liệu danh sách phát, mục yêu thích đến cơ sở dữ liệu khi người dùng chỉnh sửa danh sách phát hoặc yêu thích.
* Ứng dụng web lấy dữ liệu từ cơ sở dữ liệu và tạo ra các phản hồi (trang web, gợi ý, v.v.) để gửi lại cho người dùng.

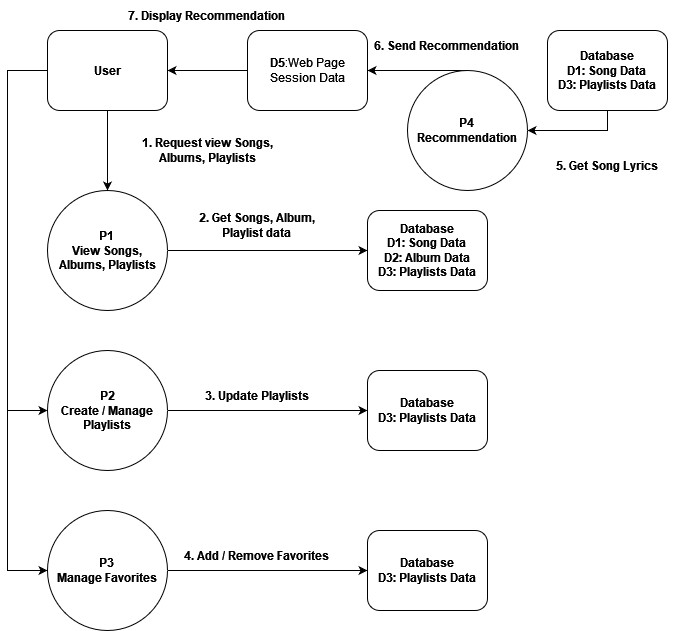
**3. Cơ sở dữ liệu:**

* **Lưu trữ dữ liệu người dùng, danh sách phát, mục yêu thích:** Lưu trữ thông tin về người dùng (như thông tin đăng nhập, sở thích) cùng với các danh sách phát và bài hát yêu thích.
* **Trả về dữ liệu cho ứng dụng:** Khi ứng dụng yêu cầu (ví dụ: lấy danh sách phát hoặc thông tin người dùng), cơ sở dữ liệu sẽ trả về dữ liệu tương ứng.
* Cơ sở dữ liệu quản lý tất cả các thông tin lưu trữ lâu dài của ứng dụng.

**4. Luồng dữ liệu:**

* **Người dùng đến ứng dụng web:** Người dùng gửi các yêu cầu như đăng nhập hoặc lấy bài hát.
* **Ứng dụng web đến cơ sở dữ liệu:** Ứng dụng lưu trữ hoặc truy vấn dữ liệu từ cơ sở dữ liệu theo các hành động của người dùng.
* **Cơ sở dữ liệu đến ứng dụng web:** Cơ sở dữ liệu trả lời các truy vấn từ ứng dụng web.
* **Ứng dụng web đến người dùng:** Ứng dụng web gửi nội dung yêu cầu (bài hát, danh sách phát, gợi ý) lại cho người dùng.

### 2.2.3. Sơ đồ DFD – Data Flow Diagram



Sơ đồ Data Flow

Đây là phân tích chi tiết về sơ đồ luồng dữ liệu cho trang web của chúng em:

**1. Tương tác của Người dùng (P1, P2, P3, P4):**

* **P1: Xem Bài Hát, Album, Danh Sách Phát**
  + **Đầu vào**: Người dùng yêu cầu xem bài hát, album hoặc danh sách phát.
  + **Quy trình**: Ứng dụng truy xuất dữ liệu từ cơ sở dữ liệu liên quan đến bài hát (D1), album (D2) và danh sách phát (D3).
  + **Đầu ra**: Dữ liệu được yêu cầu sẽ được hiển thị cho người dùng.
* **P2: Tạo/Quản lý Danh Sách Phát**
  + **Đầu vào**: Người dùng tạo hoặc cập nhật danh sách phát của họ.
  + **Quy trình**: Ứng dụng cập nhật danh sách phát trong cơ sở dữ liệu (D3).
  + **Đầu ra**: Danh sách phát được quản lý và cập nhật trong hệ thống.
* **P3: Quản lý Yêu Thích**
  + **Đầu vào**: Người dùng thêm hoặc xóa bài hát khỏi danh sách yêu thích.
  + **Quy trình**: Ứng dụng cập nhật bài hát yêu thích của người dùng trong danh sách phát được lưu trữ trong cơ sở dữ liệu (D3).
  + **Đầu ra**: Bài hát yêu thích được thêm vào hoặc xóa khỏi danh sách phát của người dùng.
* **P4: Gợi Ý**
  + **Đầu vào**: Hệ thống tạo ra gợi ý dựa trên hoạt động của người dùng.
  + **Quy trình**: Ứng dụng truy xuất lời bài hát (D1) và dữ liệu danh sách phát (D3), phân tích và tạo ra gợi ý cho người dùng.
  + **Đầu ra**: Gợi ý được gửi đến người dùng dựa trên sở thích của họ.

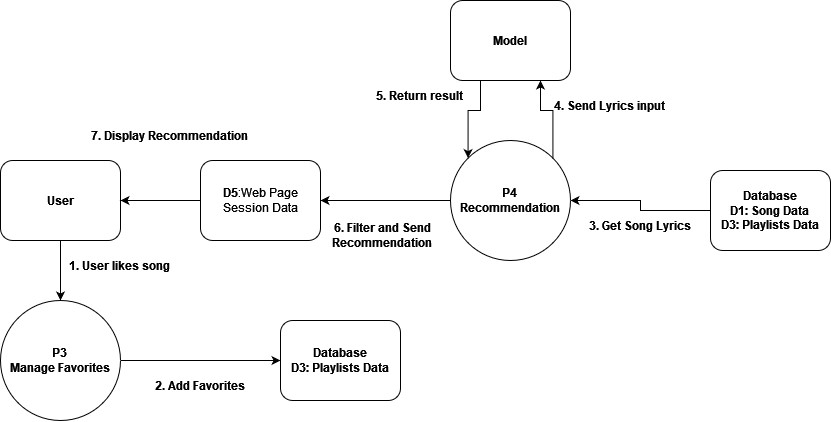
**2. Tương tác với Cơ sở Dữ liệu:**

* **D1: Dữ liệu Bài Hát**: Bao gồm dữ liệu liên quan đến từng bài hát, chẳng hạn như lời bài hát hoặc siêu dữ liệu.
* **D2: Dữ liệu Album**: Lưu trữ thông tin về các album, có thể được truy vấn khi người dùng xem album.
* **D3: Dữ liệu Danh Sách Phát**: Lưu trữ danh sách phát và yêu thích của người dùng. Dữ liệu này được cập nhật mỗi khi người dùng tạo hoặc chỉnh sửa danh sách phát hoặc yêu thích.
* **D5: Dữ liệu Phiên Web**: Dữ liệu phiên được sử dụng để lưu trữ thông tin tạm thời, chẳng hạn như gợi ý, nhằm duy trì trạng thái của phiên làm việc của người dùng.

**3. Luồng Dữ liệu:**

* Người dùng yêu cầu xem bài hát, album, danh sách phát (P1).
* Ứng dụng truy xuất dữ liệu cần thiết từ cơ sở dữ liệu (D1, D2, D3).
* Người dùng tạo hoặc quản lý danh sách phát của họ (P2), và dữ liệu này được cập nhật trong cơ sở dữ liệu danh sách phát (D3).
* Người dùng thêm/xóa bài hát khỏi danh sách yêu thích (P3), và những thay đổi này cũng được phản ánh trong dữ liệu danh sách phát (D3).
* Hệ thống gợi ý (P4) truy xuất lời bài hát (D1) và dữ liệu danh sách phát (D3) để tạo ra gợi ý.
* Các gợi ý được tạo ra sẽ được gửi đến người dùng.
* Gợi ý được hiển thị cho người dùng, với dữ liệu phiên (D5) xử lý trạng thái trang web.

### 2.2.4. DFD Đề Xuất



Hình Data Flow của Quá trình đề xuất

**Đầu tiên** thì người dùng sẽ chọn một bài hát yêu thích, sau đó **P3** sẽ đưa bài hát này vào danh sách bài hát yêu thích của người dùng.

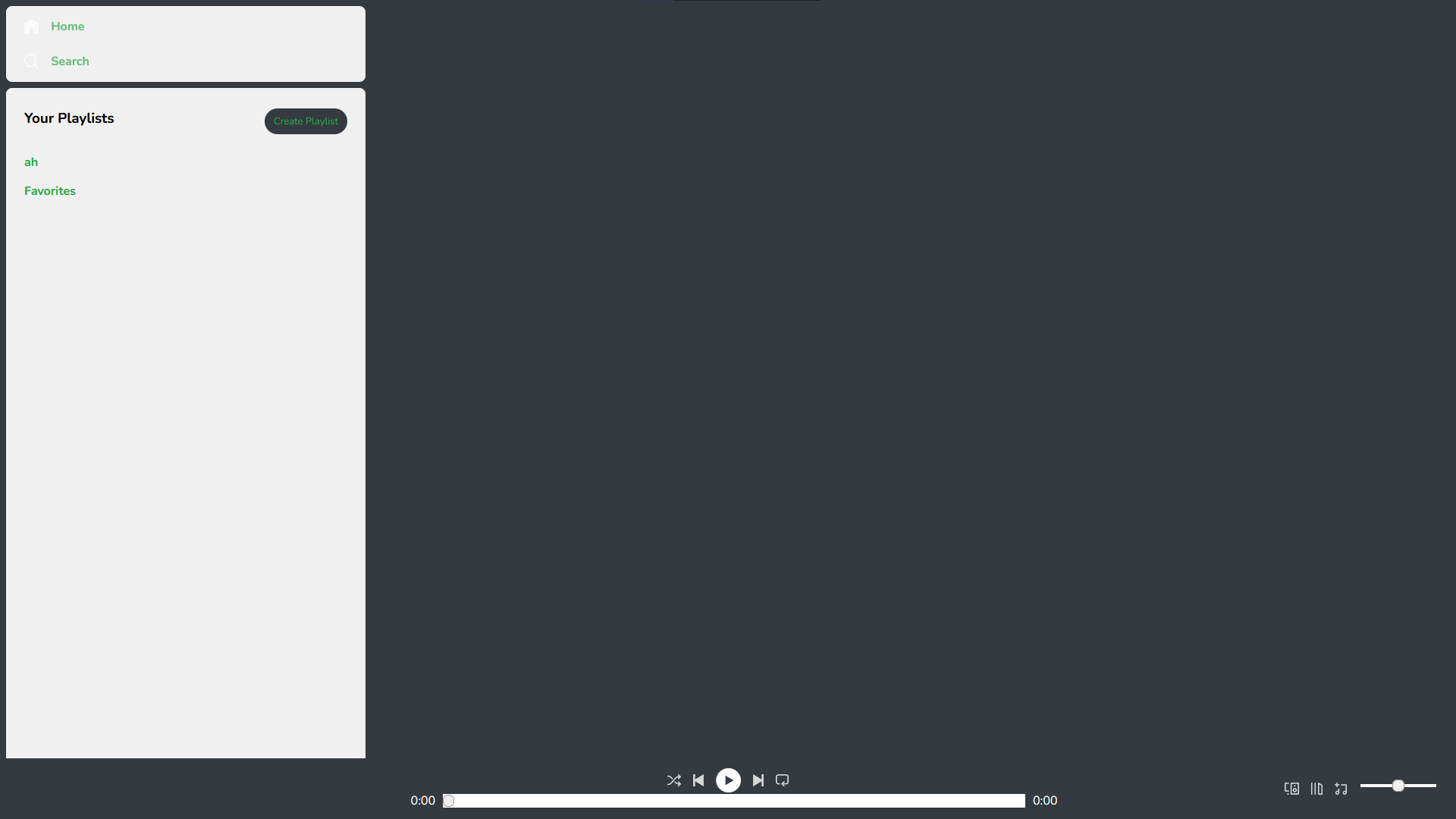
**Tiếp theo**, **P4** sẽ truy cập database và xem danh sách bài yêu thích của người dùng, kiếm tra có bài hát nào và lấy ra dữ liệu lời nhạc và thể loại của bài hát đó. Dữ liệu lời nhạc này được gửi đến Model để dự đoán và trả lại kết quả.

Sau khi nhận được kết quả dự đoán, **P4** sử dụng dữ liệu thể loại đã lấy trước đó, đối chiếu với kết quả dự đoán và chỉ chọn ra những bài hát có thể loại tương ứng. Đây là kết quả đề xuất.

Cuối cùng, kết quả đề xuất được chuyển lại về **D5** chính là nơi lưu dữ liệu của trang web để đưa ra danh sách đề xuất cho người d

## 2.2. Design UI

### 2.2.1. base.html



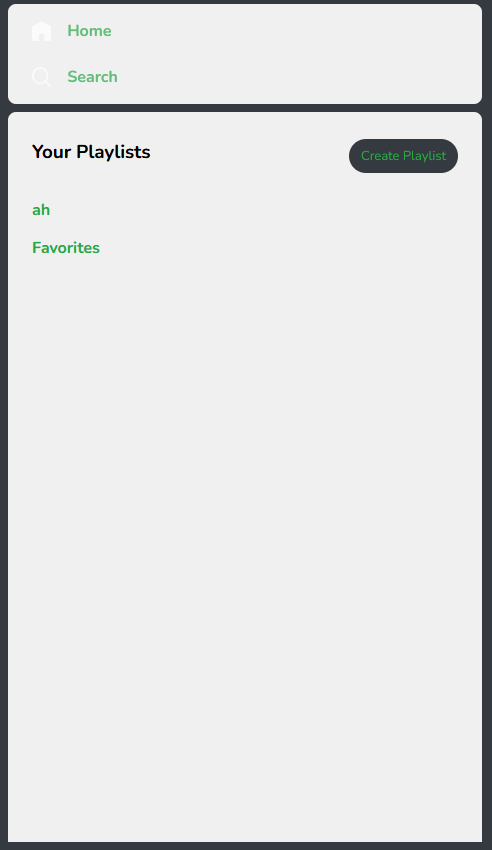
Hình ảnh Giao diện của base.html

Đây là giao diện của **base.html**, là template cơ sở cho các trang web khác của ứng dụng. Mục đích chính của **base.html** là định nghĩa cấu trúc chung (như header, footer, menu điều hướng, v.v.) và cho phép các template con mở rộng từ đó mà không cần lặp lại những phần giống nhau.

Giao diện này được chia ra làm 4 phần chính:

* Sidebar
* Main Content
* Player
* Javascript blocks

#### ****2.2.1.1. Sidebar****

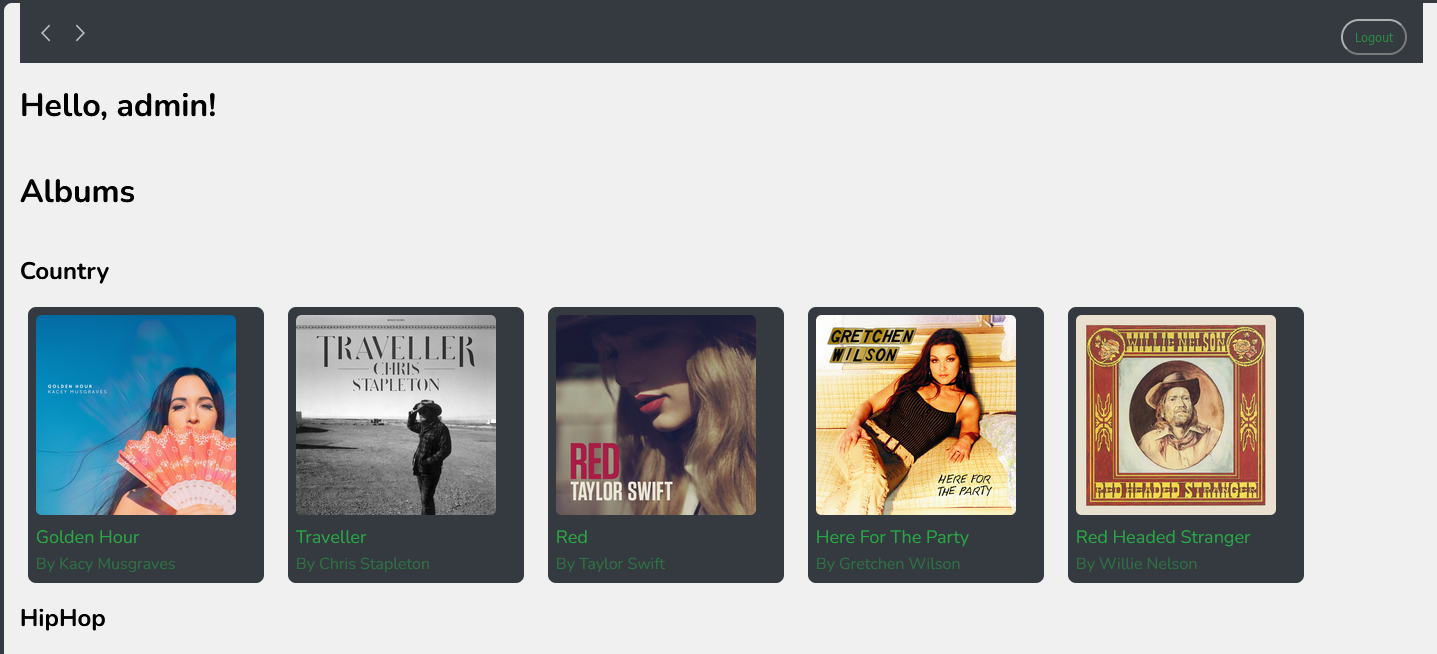


Hình ảnh Giao diện Sidebar

Ở **Sidebar**, phần ô trên thuộc phần điều hướng, gồm nút Home để đưa người dùng về trang chủ

Phần dưới sẽ chưa các danh sách nhạc của người dùng, ở đây người dùng có thể xem các danh sách nhạc của mình, bấm vào từng danh sách để coi chi tiết của danh sách đó cũng như sử dụng chức năng **Create Playlist** để đi tới một trang khác và tạo danh sách nhạc mới. Ngoài ra, người dùng còn có thể coi những bài hát yêu thích của mình thông qua chức năng **Favorites.**

#### ****2.2.1.2. Main Content****



Hình ảnh Giao diện Main Content

****

Hình ảnh Code của Main Content

Main Content là nơi mà các template khác sẽ được đưa vào, quan trọng là ở khu vực {% block content %} - {% endblock %}, 2 từ khóa này có nhiệm vụ điều hướng các trang web khác vào ô trống này để được hiển thị.

Ngoài ra, ở trên ô trống của **block content**, chúng em có các chức năng điều hướng trang và các thành phần để xử lý công việc đăng nhập. Nếu người dùng chưa đăng nhập thì sẽ hiện thị chức năng **Login**, còn nếu đã có người dùng đăng nhập thì sẽ hiển thị chức năng **Logout**.

Chức năng **Login** sẽ đưa người dùng đến trang Login để đăng nhập vào trang web. Chức năng **Logout** sẽ thoát người dùng khỏi trang web nhưng vẫn đưa về lại “Trang chủ”.

#### ****2.2.1.3. Player****

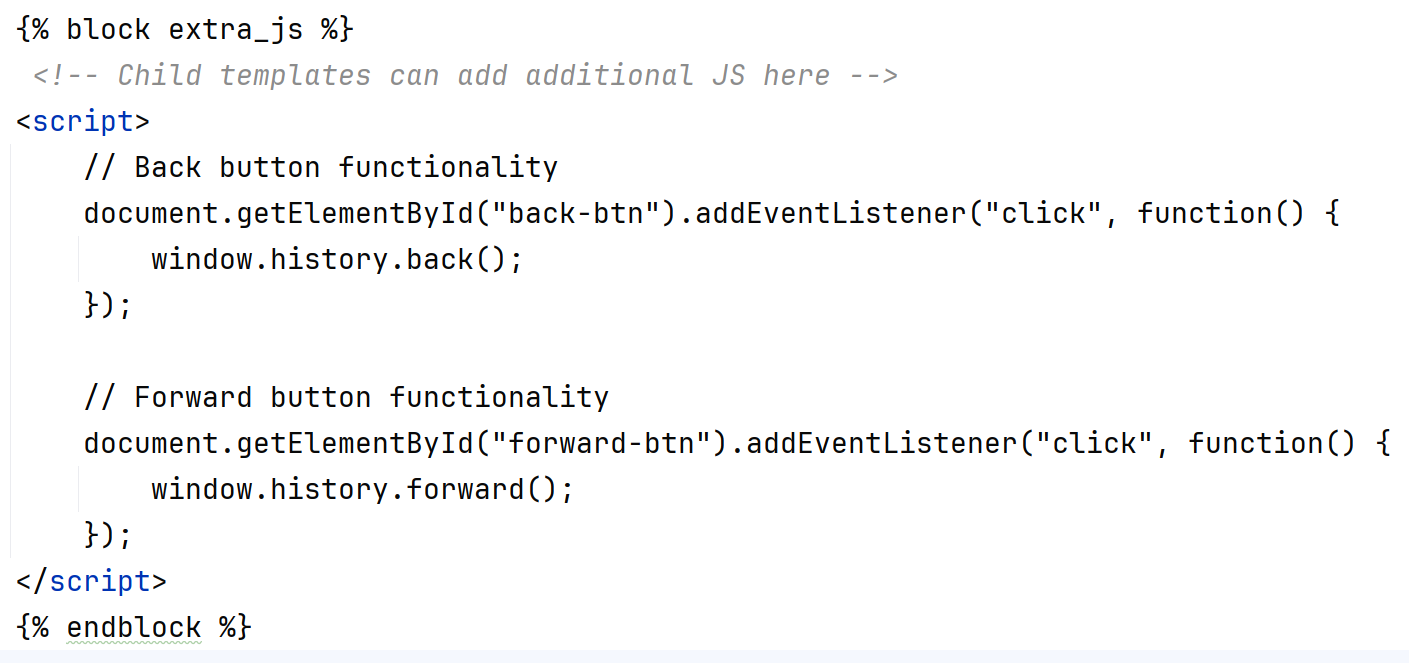


Hình ảnh Giao diện Player

**Player là** phần đại diện cho trình phát nhạc, bao gồm các thanh điều khiển nhạc:

* **Volume Controls**: Điều khiển âm lượng của trình phát.
* **Player Center**: Chứa các nút điều khiển nhạc.
* **Nút Shuffle, Previous, Play, Next, Loop**: Các nút để điều khiển phát nhạc, như phát, tạm dừng, hoặc chuyển bài.
* **Player Timeline**: Hiển thị thời gian hiện tại của bài hát và một thanh trượt để người dùng có thể điều chỉnh vị trí phát bài hát.
* **Volume Slider**: Thanh trượt điều chỉnh âm lượng.

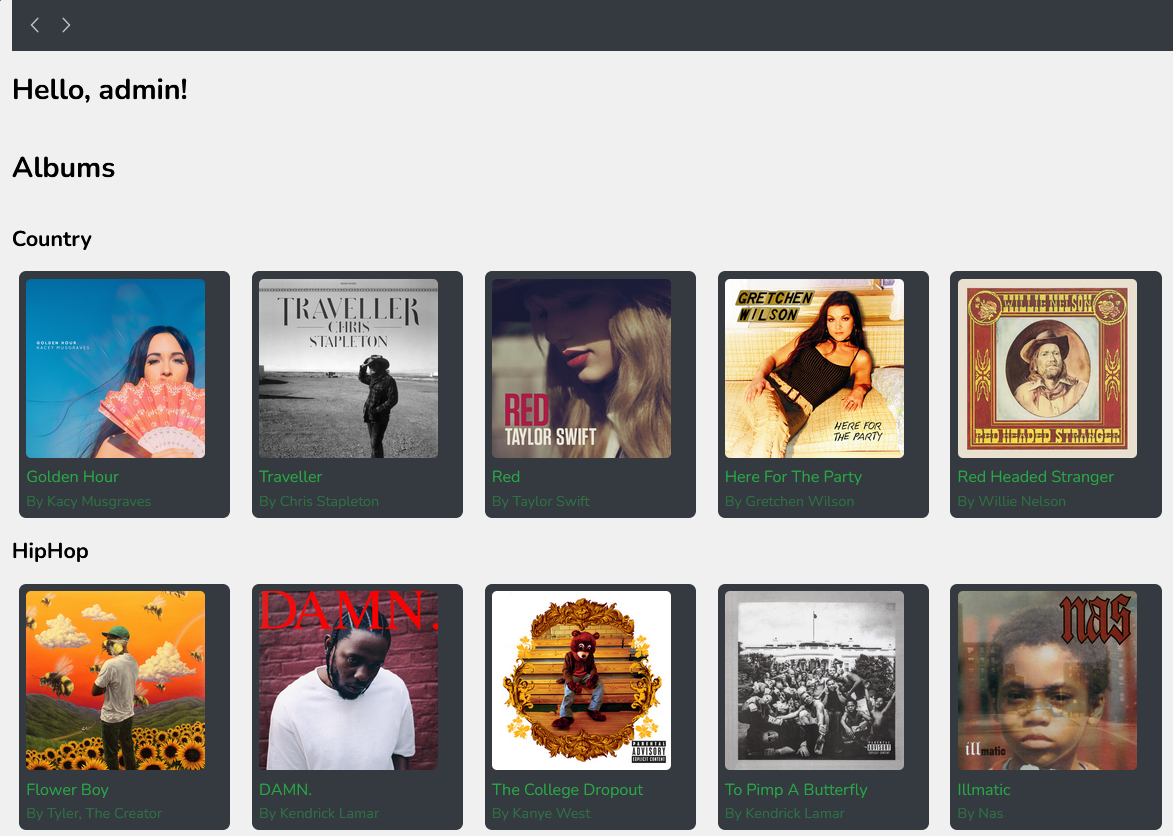
#### ****2.2.1.4.**** ****Javascript blocks****



Hình ảnh Code của Javascript blocks

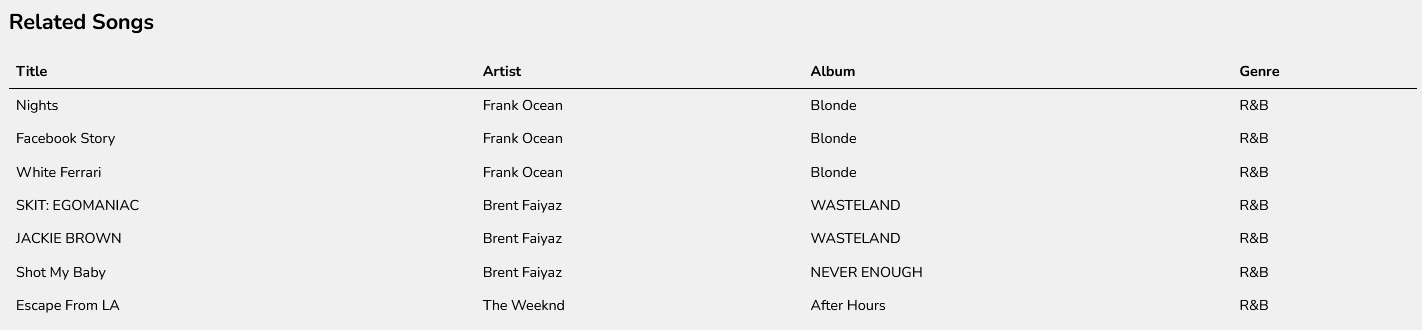
Ta có {% block extra\_js %}-{% endblock %} là ô trống chứa các chức năng javascript của các template con, ngoài ra chúng em chứa các chức năng điều hướng trang của **base.html** ở đây

### 2.2.2. Trang chủ - index.html



Hình ảnh Giao diện Trang chủ

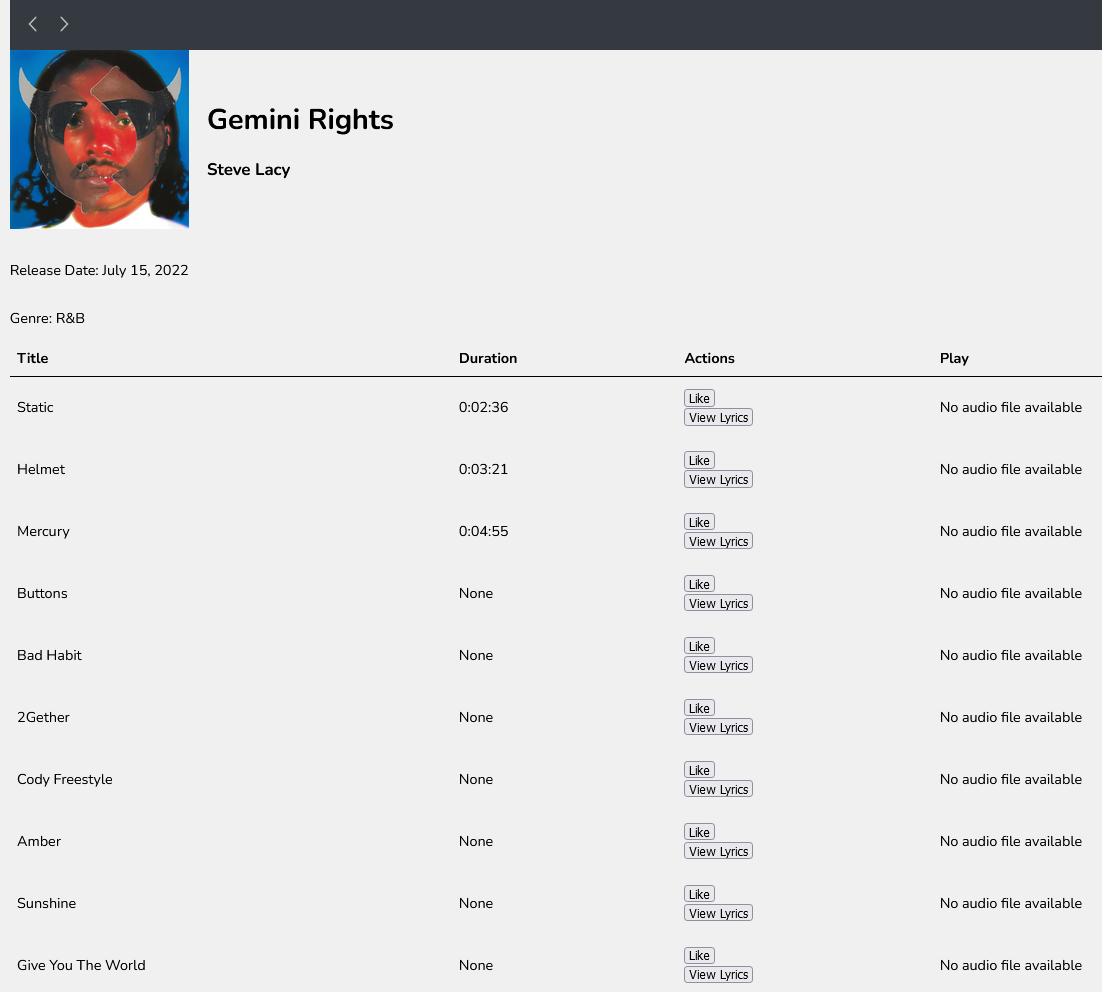
Ở Trang chủ, chúng em sẽ hiện ra các album nhạc hiện có trông database và được sắp xếp theo từng thể loại nhạc. Người dùng có thể bấm vào từng album để coi chi tiết của album đó.



Hình ảnh Giao diện Trang chủ - Phần Đề Xuất

Ở dưới các album nhạc, chúng em có phần đề xuất nhạc gồm các thông tin về bài nhạc được đề xuất. Phần đề xuất nhạc sẽ dựa trên Danh sách nhạc yêu thích của người dùng để đưa có đề xuất nhạc cụ thể.

### 2.2.3. Trang Album detail – Chi tiết Album



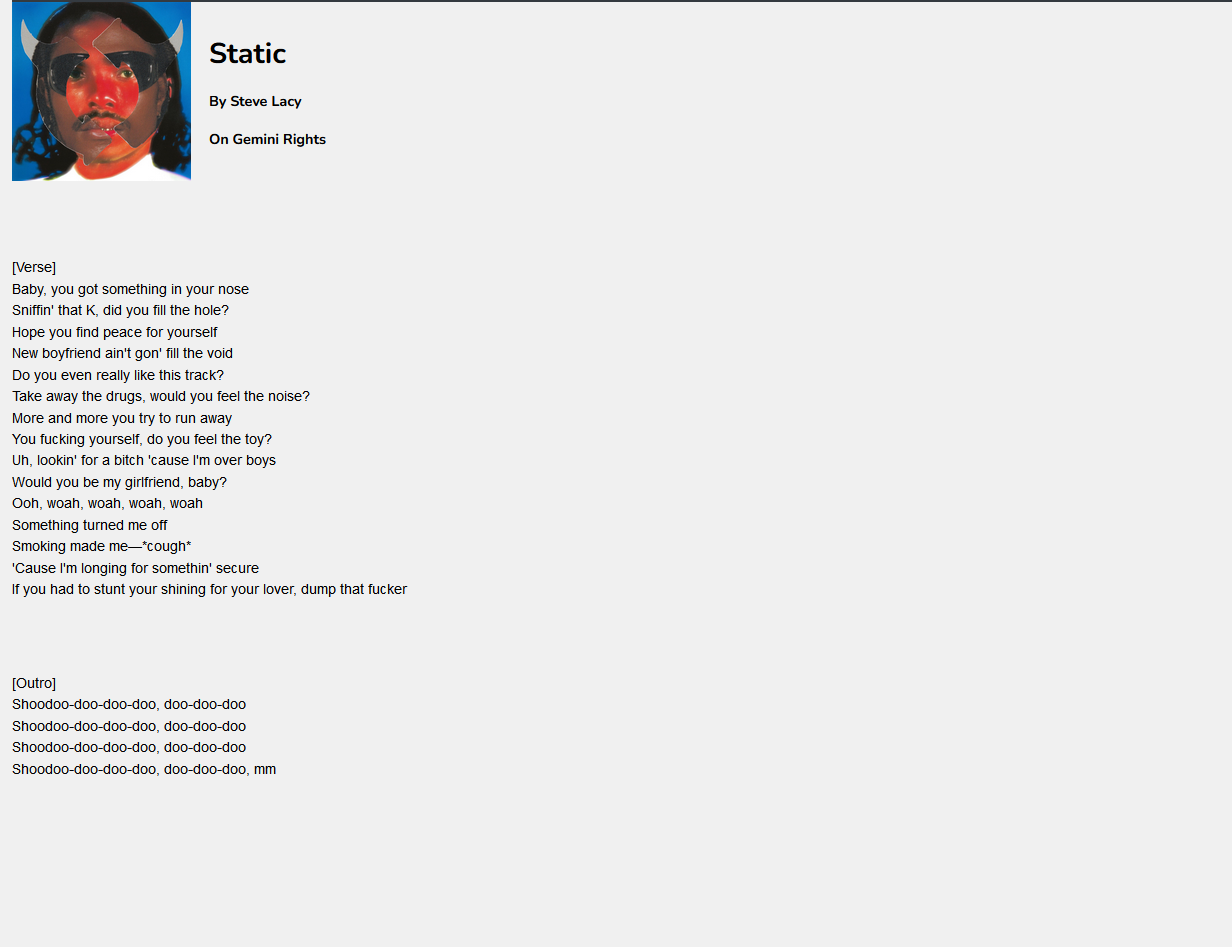
Hình ảnh Giao diện Trang chi tiết Album

Đây là Trang chi tiết Album nhạc, ở trên chúng em sẽ hiện các thông tin của album như ảnh album, tên album, nhạc sĩ sáng tác, ngày ra mắt và thể loại nhạc.

Ở dưới, chúng em hiện ra các bài hát của album này cùng với thời lượng của chúng. Ngoài ra thì chúng em có chức năng **Like** và **View lyrics**:

* Chức năng **Like** sẽ đưa bài hát tương ứng vào danh sách nhạc yêu thích của người dùng.
* Chức năng **View lyrics** sẽ đưa người dùng đến trang **Lyrics** để người dùng có thể them lời của bài hát tương ứng.

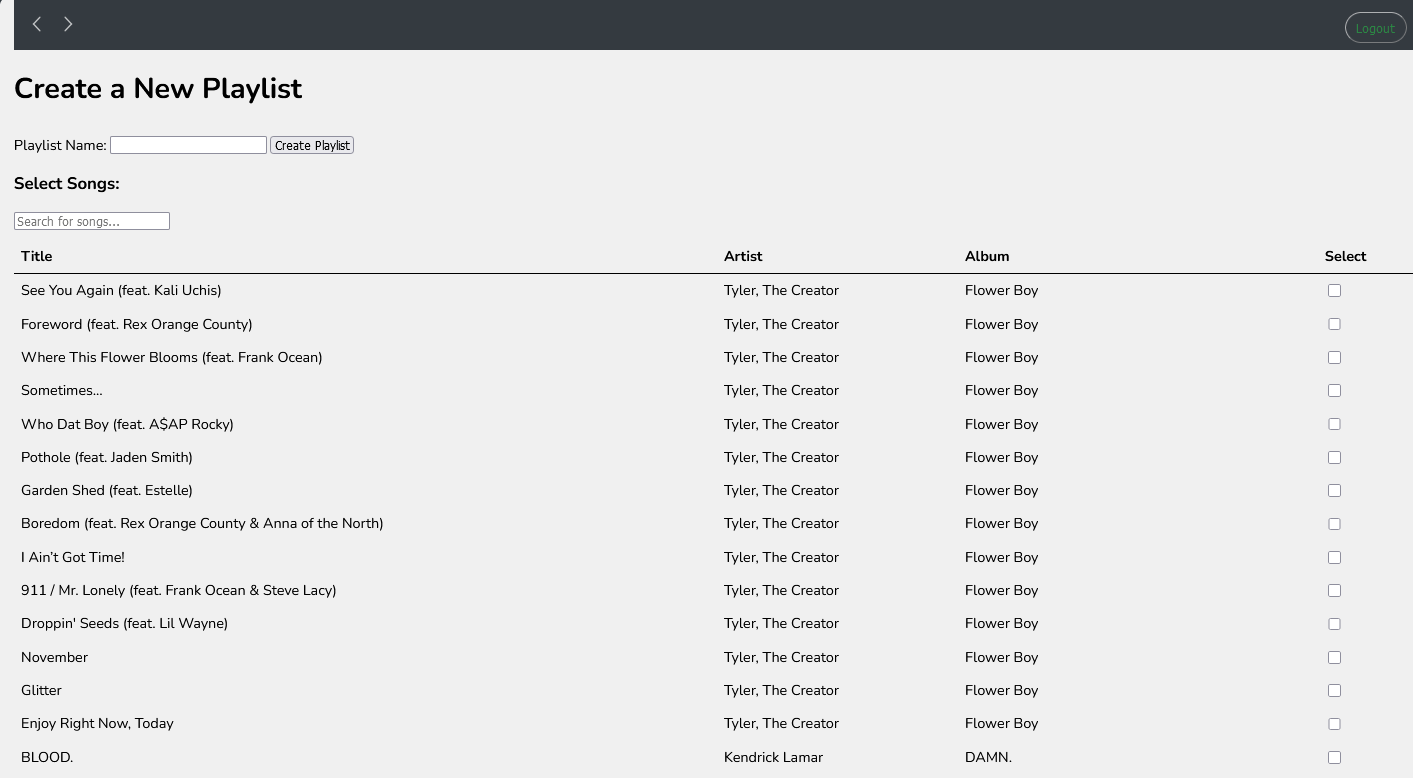
### 2.2.4. Trang Lyrics – Lời bài hát



Hình ảnh Giao diện Trang Lyrics

Ở trang này, chúng em giữ nguyên các về hiện lên một số thông tin của bài hát và hiện lên lời nhạc của bài hát đó.

### 2.2.5. Trang Create Playlist – Tạo danh sách nhạc

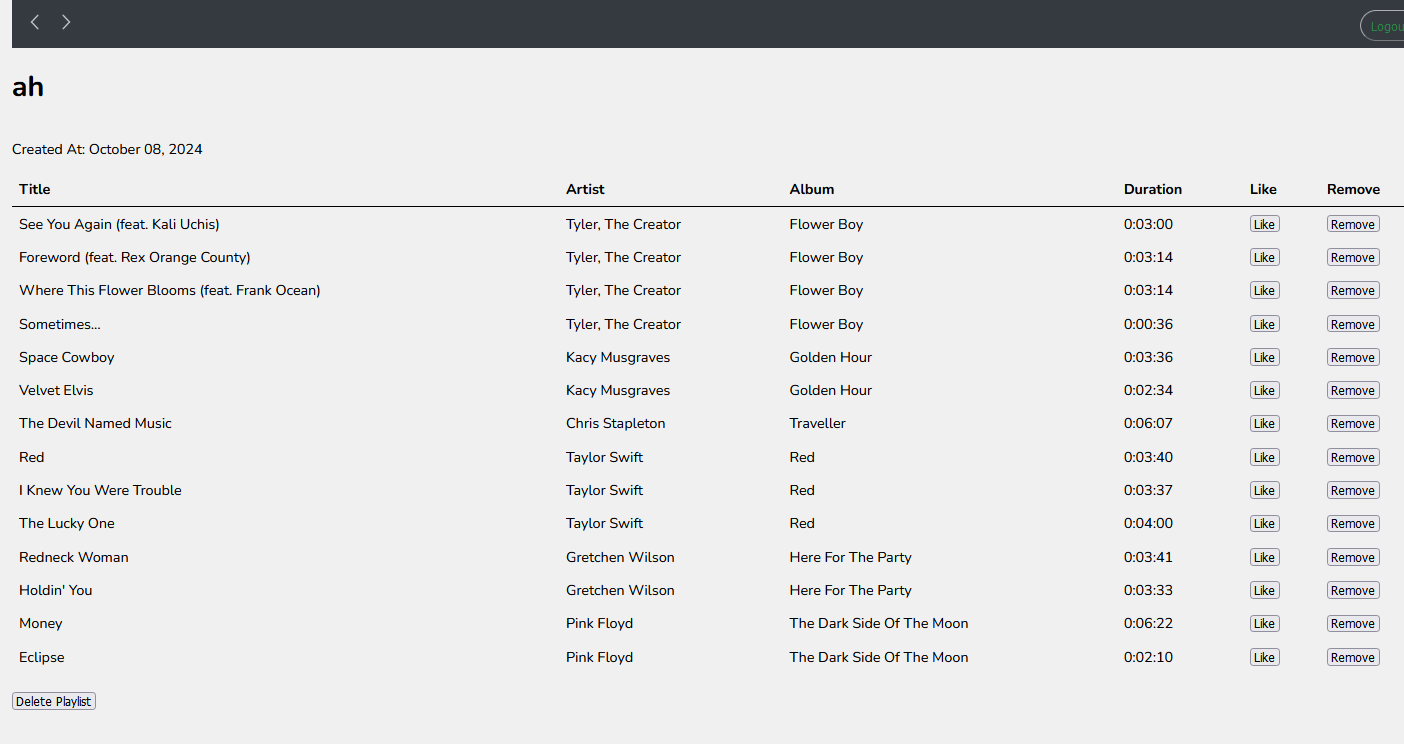


Hình ảnh Giao diện Trang Create Playlist

Trang này sẽ đưa ra danh sách cái bài hát hiện có trong database, ở đây người dùng có thể chọn các bài hát mình muốn thêm vào danh sách nhạc của mình, dùng thanh tìm kiếm để tìm một bài nhạc cụ thể và đặt tên cho danh sách này.

Sau khi bấm nút **Create Playlist**, danh sách nhạc của người dùng sẽ được tạo ra với những bài hát được chọn.

### 2.2.6. Trang Playlist Detail – Chi tiết danh sách nhạc

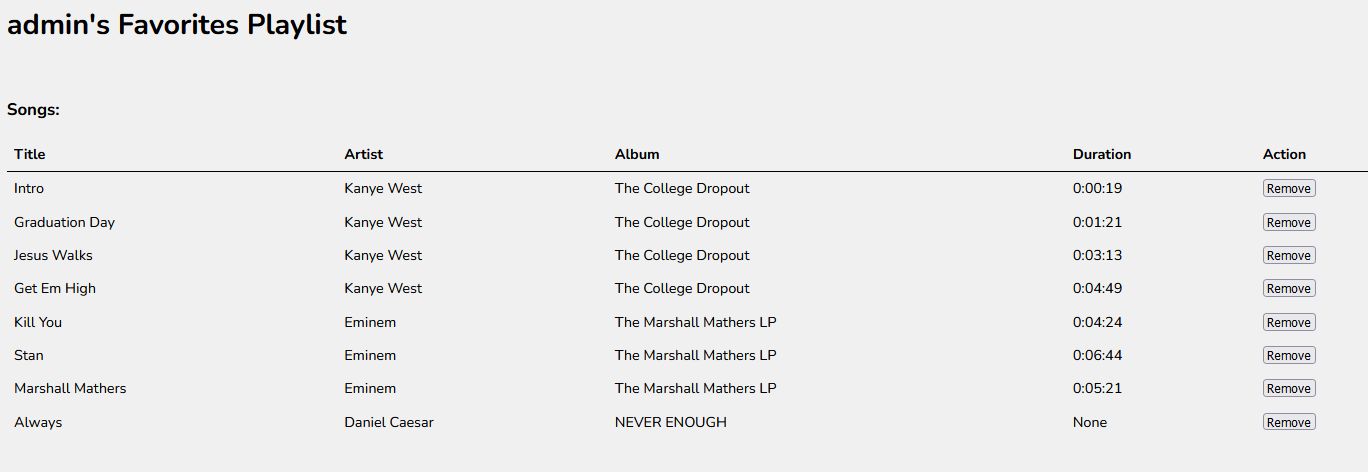


Hình ảnh Giao diện Trang Playlist Detail

Trang này hiện lên thông tin của danh sách nhạc mà người dùng đã tạo, gồm tên của danh sách và ngày tạo. Cùng với thông tin các bài hát người dùng đã chọn.

Ở đây chúng em vẫn giữ chức năng **Like** để nếu người dùng muốn đưa một bài hát nào đó vào danh sách yêu thích. Ngoài ra còn có chức năng **Remove**, sẽ xóa bài hát tương ứng khỏi danh sách nhạc này. Cuối cùng là chức năng **Delete Playlist** có phép người dùng xóa bỏ toàn bộ danh sách nhạc này.

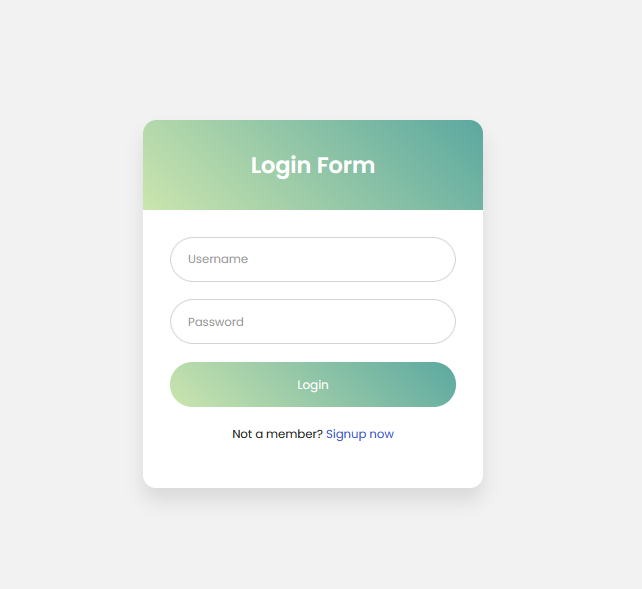
### 2.2.7. Trang Favorites – Danh sách nhạc yêu thích



Hình ảnh Giao diện Trang Favorites

Trang này sẽ hiện các bài và thông tin bài hát của người dùng đã thích khi sử dụng chức năng **Like.** Ngoài ra chúng em có chức năng **Remove** để người dùng xóa bỏ một bài hát nào đó khỏi danh sách này.

### 2.2.8. Trang Login

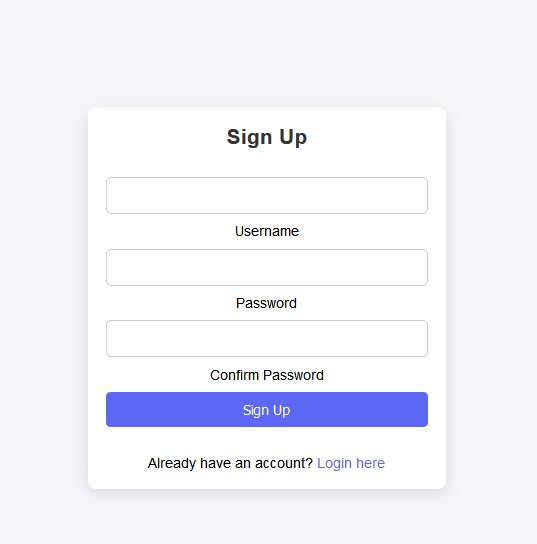


Hình ảnh Giao diện Trang Login

Ở đây chúng em có form login, cho phép người dùng nhập tên tài khoàn và mật khẩu để đăng nhập vào trang web. Nếu chưa có tài khoản thì người dùng có thể bấm vào “Signup now” để được điều hướng tới trang Đăng ký.

Khi bấm vào nút “Login” thì chúng em sẽ kiểm tra nếu thông tin đăng nhập đúng và chuyển hướng tới ”Trang chủ”.

### 2.2.9. Trang Signup



Hình ảnh Giao diện trang Sign Up

Ở đây chúng em có form đăng ký, người dùng có thể nhập tên đăng nhập và mật khẩu để tạo một tài khoản mới. Khi bấm nút “Sign Up”, chúng em sẽ kiểm tra nếu 2 ô mật khẩu trùng nhau, nếu đúng thì sẽ chuyển hướng người dùng đến “Trang chủ”.

## 2.3. Design model: pLSA

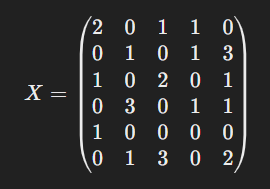
### 2.3.1. Giới thiệu chung về pLSA

pLSA, do Thomas Hofmann giới thiệu vào năm 1999, là một phương pháp dựa trên mô hình xác suất để tìm ra các chủ đề tiềm ẩn trong các tập văn bản. Mục tiêu của pLSA là tìm ra các chủ đề cơ bản trong một tập hợp các tài liệu bằng cách phân tích sự đồng xuất hiện của các từ trong các tài liệu đó. Thay vì xem mỗi từ và tài liệu là một đơn vị riêng biệt, pLSA coi chúng như các biến ngẫu nhiên trong một mô hình xác suất, với các chủ đề là những biến ẩn tiềm tàng mà thuật toán muốn khám phá.

### 2.3.2. Nguyên lí hoạt động của pLSA

#### 2.3.2.1. Mô hình xác suất của pLSA

Giả sử chúng em có một tập dữ liệu gồm 5 tài liệu (D1, D2, D3, D4, D5) và một từ vựng gồm 6 từ (w1, w2, w3, w4, w5, w6). Dữ liệu về sự xuất hiện của các từ trong các tài liệu này được biểu diễn dưới dạng **ma trận từ - tài liệu**:



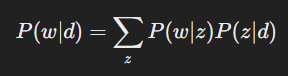
Hình ảnh Ma trận từ - tài liệu X:

Trong đó, mỗi hàng tương ứng với một từ và mỗi cột tương ứng với một tài liệu. Mỗi phần tử của ma trận là số lần xuất hiện của một từ trong một tài liệu. Ví dụ: từ w1 xuất hiện 2 lần trong tài liệu D1 và 1 lần trong D3. Mục tiêu của pLSA là tìm ra các **chủ đề tiềm ẩn** từ các tài liệu này và xác định cách các từ phân bố trong các chủ đề đó.

Mô hình pLSA giả định rằng quá trình sinh ra dữ liệu văn bản có thể được mô tả bởi các bước sau:

* Một tài liệu d được chọn với xác suất P(d).
* Một chủ đề z được chọn với xác suất có điều kiện P(z|d), tức là xác suất chọn một chủ đề z khi biết tài liệu d đã được chọn.
* Một từ w được chọn từ chủ đề z với xác suất P(w|z), tức là xác suất chọn từ w từ chủ đề z.

Tổng quát, mô hình pLSA cho mỗi cặp từ w và tài liệu d có thể được viết dưới dạng công thức xác suất:



Hình ảnh Công thức xác suất **word** theo **document**

Trong pLSA, chúng em muốn ước lượng ba loại phân phối xác suất:

1. **P(w|z):** Xác suất từ w xuất hiện trong chủ đề z.
2. **P(z|d):** Xác suất chủ đề z xuất hiện trong tài liệu d.
3. **P(z):** Xác suất tổng thể của một chủ đề (không phụ thuộc vào tài liệu).

#### 2.3.2.2. Khởi tạo số chủ đề K

Giả sử chúng em giả định rằng có 2 chủ đề K=2. Mục tiêu là phân tích ma trận từ - tài liệu thành các ma trận mô tả sự phân phối của các từ và chủ đề. Hai ma trận chính mà chúng em sẽ tạo ra là:

* **Ma trận từ - chủ đề** P(w|z): Mỗi hàng là một từ và mỗi cột là một chủ đề
* **Ma trận chủ đề - tài liệu** P(z|d): Mỗi hàng là một chủ đề và mỗi cột là một tài liệu.

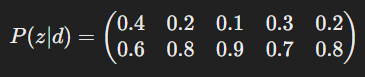
#### 2.3.2.3. Ma trận P(w|z) và P(z|d) ban đầu

Giả sử chúng em khởi tạo ma trận từ - chủ đề P(w|z) và chủ đề - tài liệu P(z|d)) một cách ngẫu nhiên:



Hình ảnh **Ma trận từ - chủ đề** P(w|z)(2 chủ đề)

* Mỗi hàng tương ứng với một từ.
* Mỗi cột là xác suất một từ thuộc về một chủ đề z1​ hoặc z2



**Hình ảnh Ma trận chủ đề - tài liệu** P(z|d) (2 chủ đề)

* Mỗi hàng là một chủ đề.
* Mỗi cột là xác suất một chủ đề xuất hiện trong một tài liệu.

#### 2.3.2.4. Tính xác suất P(w|d)

Mục tiêu là tính xác suất của mỗi từ w xuất hiện trong một tài liệu d. Để tính xác suất của một từ cụ thể xuất hiện trong một tài liệu cụ thể, chúng em chỉ cần lấy tích của xác suất từ đó thuộc về mỗi chủ đề và xác suất chủ đề đó xuất hiện trong tài liệu.

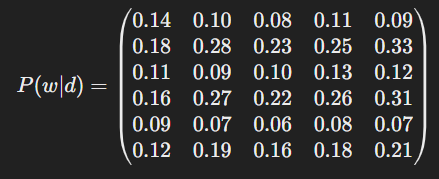
##### Ví dụ: Tính P(w1|D1)

P(w1|D1)=P(w1|z1)P(z1|D1)+P(w1|z2)P(z2|D1)

Thay giá trị từ ma trận P(w|z) và P(z|d):

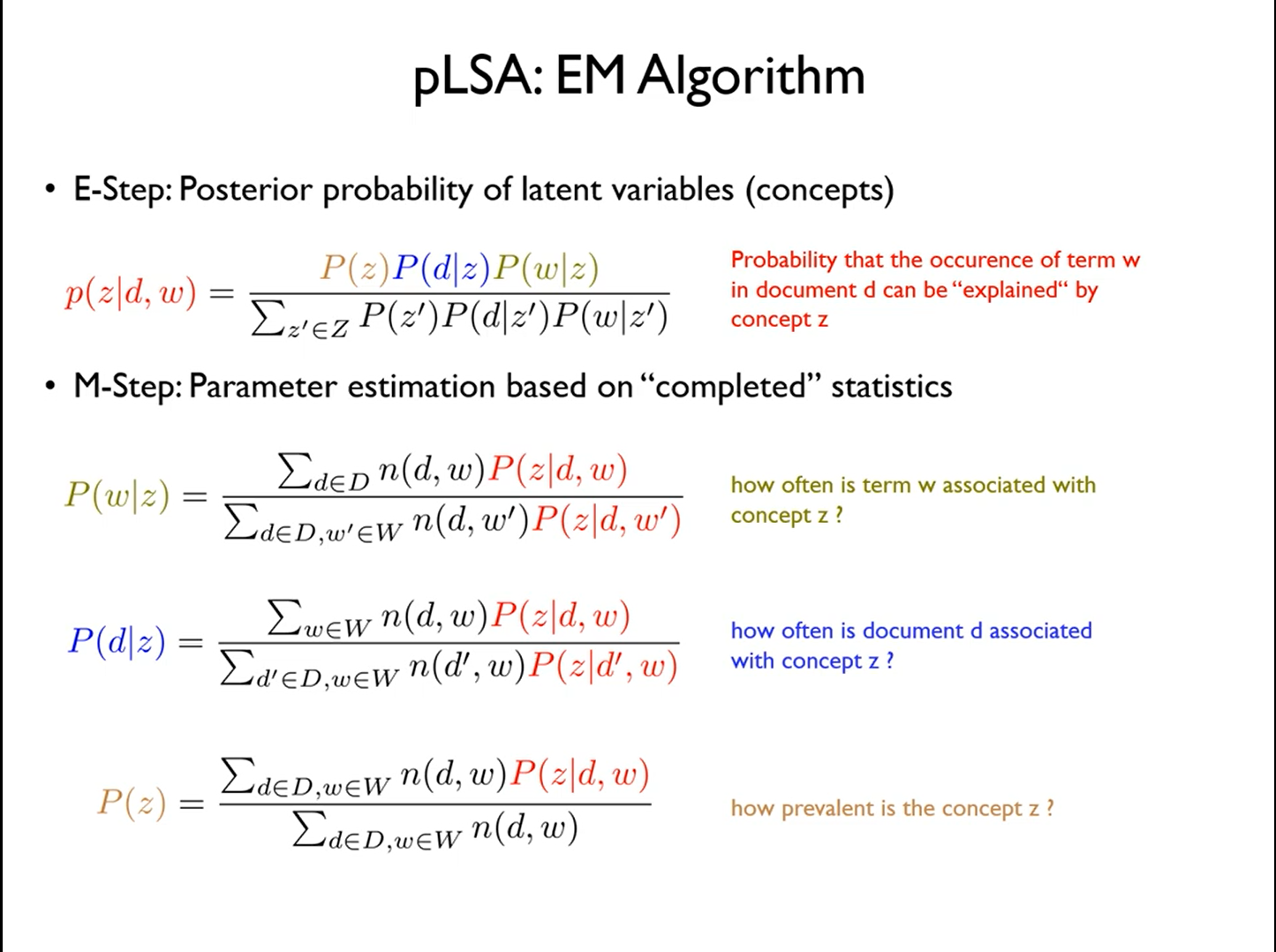
P(w1|D1)=0.2×0.4+0.1×0.6=0.08+0.06=0.14

Tương tự, chúng em có thể tính các giá trị P(w|d) khác để tạo ra **ma trận xác suất từ - tài liệu** được dự đoán:



**Hình ảnh** Ma trận P(w|d)

#### 2.3.2.5. Tối ưu hóa tham số với thuật toán EM



Hình ảnh Thuật toán EM

Để tối ưu hóa các tham số của mô hình P(w|z) và P(z|d), pLSA sử dụng thuật toán EM (Expectation-Maximization).

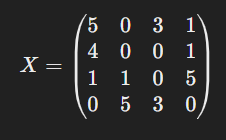
* **Bước E (Expectation)**: Tính xác suất kỳ vọng của các biến ẩn (các chủ đề) với dữ liệu quan sát.
* **Bước M (Maximization)**: Tối ưu hóa các tham số P(w|z) và P(z|d) bằng cách tối đa hóa kỳ vọng được tính từ bước E.

Trong mô hình này, chúng em cập nhật giá trị của P(w|z) và P(z|d) sao cho xác suất dự đoán từ các topic và quan hệ của chúng với bài hát và các từ để đạt được mức tối ưu cao so với input ban đầu.

### 2.3.3. ****Ứng dụng pLSA trong hệ thống đề xuất****

Giả sử chúng em đang xây dựng một hệ thống gợi ý phim. Tương tự như việc sử dụng ma trận từ - tài liệu, chúng em có thể sử dụng **ma trận người dùng - phim**, trong đó các hàng là người dùng, các cột là phim, và giá trị trong ma trận là số điểm (rating) mà người dùng đánh giá cho một bộ phim.

* **pLSA** giúp tìm ra các **chủ đề tiềm ẩn** (ở đây có thể là các thể loại hoặc chủ đề phim) mà một người dùng yêu thích.
* Sau đó, dựa trên các chủ đề này, hệ thống sẽ gợi ý cho người dùng các bộ phim tương ứng với chủ đề mà họ có nhiều khả năng sẽ thích, dựa trên các phim mà họ đã đánh giá trước đó.



Hình ảnh Ví dụ về ma trận người dùng - phim

Ở đây, mỗi hàng tương ứng với một người dùng, và mỗi cột là một bộ phim. Sau khi chạy pLSA, hệ thống sẽ dự đoán các bộ phim khác mà người dùng có thể thích dựa trên các chủ đề tiềm ẩn.

### ****2.3.4. Tokenizer trong pLSA****

Trong quá trình xây dựng hệ thống đề xuất nhạc, bước đầu tiên và quan trọng là xử lý và chuẩn bị dữ liệu văn bản đầu vào để phù hợp với mô hình pLSA. Để làm được điều này, chúng em cần sử dụng **tokenizer** - một công cụ để chuyển đổi văn bản thô thành các đơn vị nhỏ hơn (tokens) mà hệ thống có thể phân tích được. Các đơn vị này có thể là từ, cụm từ, hoặc ký tự phụ thuộc vào mục đích cụ thể của bài toán.

Trong bối cảnh của mô hình pLSA, tokenizer thường được sử dụng để phân tách văn bản (chẳng hạn như mô tả bài hát hoặc thông tin về nghệ sĩ) thành các từ riêng lẻ. Mỗi từ sau đó sẽ được đưa vào ma trận từ - tài liệu (word-document matrix) để pLSA phân tích. Bước tokenizer có vai trò quan trọng trong việc làm sạch và chuẩn hóa dữ liệu trước khi áp dụng mô hình pLSA, giúp loại bỏ các từ không cần thiết như giới từ, đại từ (stop words), và xử lý các dạng từ đồng nghĩa hoặc biến thể của từ gốc (stemming hoặc lemmatization).

#### 2.3.4.1. Quy trình hoạt động của Tokenizer

Quy trình tokenizer trong hệ thống đề xuất nhạc dựa trên mô hình pLSA bao gồm các bước sau:

1. **Xóa các ký tự đặc biệt và dấu câu:** Các ký tự như dấu chấm, dấu phẩy, dấu hỏi, và các ký tự không liên quan đến nội dung của tài liệu sẽ bị loại bỏ.
2. **Chuyển đổi văn bản về dạng chữ thường:** Tất cả các từ trong tài liệu được chuyển thành chữ thường để tránh sự khác biệt về dạng viết hoa, viết thường của từ.
3. **Tách các từ:** Văn bản được chia thành các từ (tokens) dựa trên khoảng trắng hoặc dấu câu.
4. **Loại bỏ các từ không cần thiết (stop words):** Những từ thông dụng không mang nhiều ý nghĩa như "và", "hoặc", "là",... sẽ bị loại bỏ để tăng hiệu quả phân tích.
5. **Lemmatization hoặc stemming:** Từ được đưa về dạng gốc (lemmatization) hoặc dạng cơ bản nhất (stemming) để tránh các biến thể về ngữ pháp làm tăng độ phức tạp của dữ liệu.

Sau khi tokenizer hoàn thành, các từ đã xử lý sẽ được lưu vào ma trận từ - tài liệu, sẵn sàng để áp dụng mô hình pLSA.

#### 2.3.4.2. Ứng dụng Tokenizer trong hệ thống đề xuất nhạc

Trong hệ thống đề xuất nhạc, tokenizer được áp dụng để phân tích mô tả bài hát, tên nghệ sĩ, hoặc lời bài hát. Việc chia nhỏ các đoạn văn bản này thành các từ và chuyển vào ma trận từ - tài liệu giúp pLSA xác định các chủ đề tiềm ẩn, chẳng hạn như các thể loại nhạc hoặc chủ đề âm nhạc mà người dùng quan tâm. Từ đó, hệ thống có thể dự đoán và gợi ý những bài hát phù hợp dựa trên các từ khóa liên quan đến sở thích của người dùng.

### 2.3.5. Ưu điểm của pLSA so với các phương pháp khác

pLSA có một số ưu điểm nổi bật trong việc khám phá các chủ đề tiềm ẩn từ dữ liệu văn bản và được sử dụng rộng rãi trong các hệ thống đề xuất:

* **Tính linh hoạt:** pLSA có thể áp dụng cho nhiều loại dữ liệu khác nhau, từ văn bản (tài liệu, bài hát) đến dữ liệu không phải văn bản như các đánh giá của người dùng đối với sản phẩm hay dịch vụ.
* **Khả năng giải thích:** Một trong những điểm mạnh của pLSA là khả năng giải thích được các chủ đề tiềm ẩn, giúp chúng em hiểu rõ hơn về cách mà các từ (hoặc mục tiêu) được phân phối trong các tài liệu (hoặc tập dữ liệu).
* **Tối ưu hóa dựa trên xác suất:** pLSA dựa trên mô hình xác suất, cung cấp một cách tiếp cận có nền tảng toán học vững chắc để khám phá các chủ đề ẩn trong dữ liệu và xác suất điều kiện của chúng.

### 2.3.6. Hạn chế của pLSA

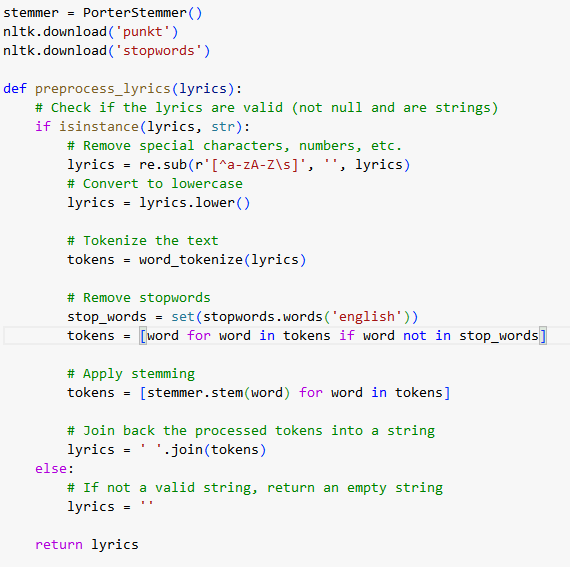
Mặc dù có nhiều ưu điểm, pLSA cũng gặp phải một số hạn chế khi áp dụng trong thực tế:

* **Quá trình huấn luyện phức tạp:** Mô hình pLSA yêu cầu tối ưu hóa nhiều tham số bằng thuật toán EM, quá trình này có thể mất nhiều thời gian và tài nguyên tính toán, đặc biệt khi làm việc với tập dữ liệu lớn.
* **Vấn đề quá khớp (overfitting):** pLSA có thể gặp phải vấn đề quá khớp nếu không kiểm soát tốt số lượng chủ đề K hoặc khi tập dữ liệu quá nhỏ, dẫn đến việc mô hình học thuộc dữ liệu thay vì tìm ra các mẫu chung.
* **Không có sự chia sẻ thông tin giữa các tài liệu:** Mỗi tài liệu có phân phối chủ đề riêng biệt, không có sự chia sẻ thông tin chủ đề giữa các tài liệu, dẫn đến việc khó mở rộng mô hình cho các hệ thống quy mô lớn hoặc khi có dữ liệu mới.

### 2.3.7. Kiến trúc model pLSA

#### 2.3.7.1. Dữ liệu

Dữ liệu đầu vào của chúng em gồm 322 bài hát và lời nhạc của chúng, được chia theo 5 thể loại nhạc Rock, HipHop, R&B, Country, Pop.



Hình ảnh Hàm xử lý lyrics

Hàm **preprocess\_lyrics** được sử dụng để tiền xử lý lời bài hát trước khi phân tích. Quá trình này giúp làm sạch và chuẩn hóa dữ liệu để cải thiện kết quả mô hình.

**Các thư viện và công cụ:**

* **PorterStemmer:** Một công cụ để thực hiện stemming (rút gọn từ về dạng gốc).
* **nltk.download('punkt') và nltk.download('stopwords'):** Tải dữ liệu cần thiết từ thư viện NLTK, bao gồm bộ phân tích từ (tokenizer) và danh sách từ dừng (stopwords).

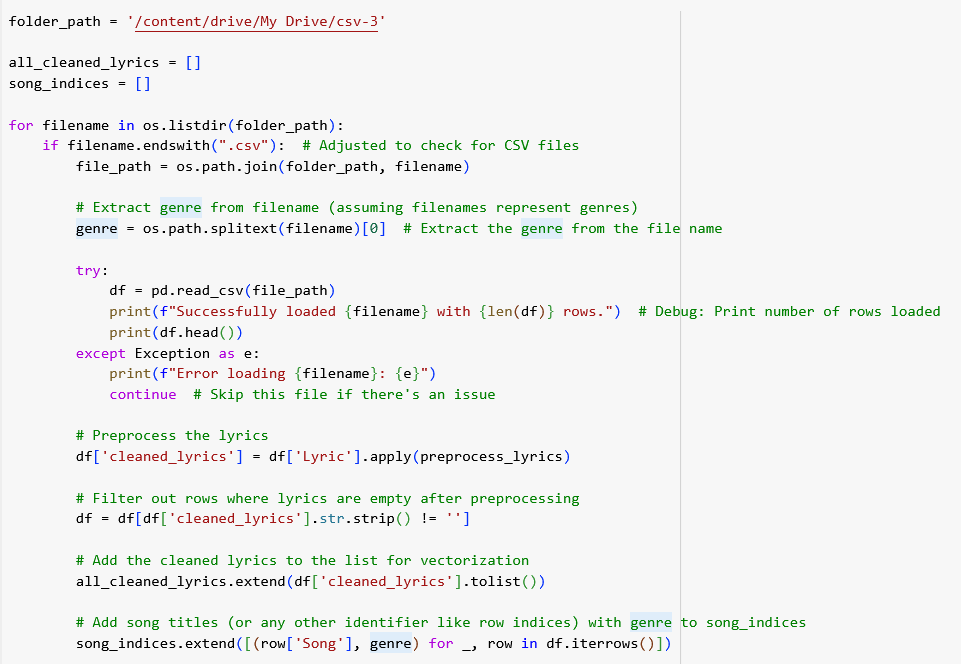
**Input:** Chuỗi lời bài hát đầu vào cần được tiền xử lý (**lyrics)**.

**Output:** Chuỗi lời bài hát đã được tiền xử lý, hoặc một chuỗi rỗng nếu đầu vào không hợp lệ.

**Các bước tiền xử lý:**

* Kiểm tra xem input có phải là một chuỗi ký tự hợp lệ hay không (chuỗi không rỗng).
* Sử dụng biểu thức chính quy **re.sub(r'[^a-zA-Z\s]', '', lyrics)** để loại bỏ mọi ký tự không phải chữ cái và khoảng trắng.
* **lyrics = lyrics.lower()** giúp chuẩn hóa các từ bằng cách chuyển tất cả thành chữ thường.
* Sử dụng hàm **word\_tokenize(lyrics)** để tách lời bài hát thành các (tokens).
* **Stopwords** là các từ phổ biến như "and", "the", "is", thường không mang nhiều ý nghĩa trong phân tích văn bản.
* Bộ từ dừng được lấy từ nltk.corpus.stopwords, và các từ trong danh sách stopwords sẽ được loại khỏi danh sách tokens bằng:
* **tokens = [word for word in tokens if word not in stop\_words].**
* Sử dụng **PorterStemmer** để chuyển mỗi từ về dạng gốc của nó. Ví dụ, "running" sẽ trở thành "run".
* Stemming giúp giảm số lượng từ trong văn bản, giúp mô hình phân tích hiểu rõ hơn về cấu trúc ý nghĩa của văn bản.
* Các từ đã được xử lý sẽ được ghép lại thành một chuỗi: **lyrics = ' '.join(tokens).**

**Output:** Trả về chuỗi lyrics đã được tiền xử lý, hoặc một chuỗi rỗng nếu đầu vào không hợp lệ.



Hình ảnh Quá trình xử lý dữ liệu

**Dữ liệu đã được xử lý:**

* **all\_cleaned\_lyrics:** Danh sách chứa lời bài hát đã qua xử lý và làm sạch từ tất cả các tệp CSV.
* **song\_indices:** Danh sách chứa thông tin về tên bài hát và thể loại của nó. Danh sách này sẽ được dùng để đối chiếu output của model với bài hát tương ứng

**Các bước xử lý:**

**Duyệt qua các tệp CSV trong thư mục:**

* Sử dụng **os.listdir(folder\_path)** để lấy danh sách tất cả các tệp trong thư mục folder\_path.
* Kiểm tra xem tệp có phải là tệp CSV không bằng cách kiểm tra phần mở rộng **(filename.endswith(".csv")).**

**Lấy thể loại từ tên tệp:**

* Giả sử tên tệp đại diện cho thể loại nhạc, chương trình sẽ tách phần tên của tệp bằng os.path.splitext(filename)[0] (bỏ phần mở rộng .csv).
* Ví dụ, nếu tệp tên "rock.csv", chương trình sẽ lấy thể loại là "rock".

**Đọc tệp CSV và xử lý lỗi:**

* Sử dụng pandas.read\_csv(file\_path) để đọc tệp CSV.
* Nếu có lỗi khi tải tệp (ví dụ, tệp hỏng hoặc định dạng không đúng), chương trình sẽ bỏ qua tệp đó và tiếp tục với các tệp khác, đồng thời in ra lỗi gặp phải (except Exception as e).

**Tiền xử lý lời bài hát:**

* Áp dụng hàm preprocess\_lyrics (hàm xử lý được định nghĩa trước) lên cột 'Lyric' trong dữ liệu.
* Sau đó, lọc bỏ các hàng mà sau khi xử lý, lời bài hát bị trống (hàng nào có 'cleaned\_lyrics' chỉ chứa khoảng trắng hoặc rỗng sẽ bị loại bỏ).

**Thêm lời bài hát đã xử lý vào danh sách:**

* Lời bài hát đã qua xử lý (df['cleaned\_lyrics']) sẽ được thêm vào danh sách all\_cleaned\_lyrics để chuẩn bị cho việc vector hóa.

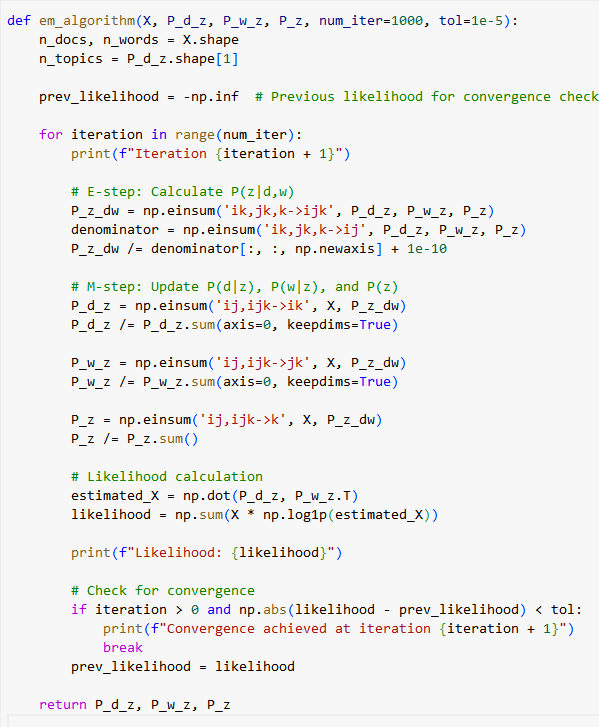
**Thêm chỉ mục bài hát và thể loại:**

* Tạo một danh sách song\_indices chứa tên bài hát (row['Song']) và thể loại tương ứng với từng bài từ mỗi tệp CSV.
* Sử dụng cú pháp [(row['Song'], genre) for \_, row in df.iterrows()] để tạo danh sách các cặp bài hát và thể loại.

**Lưu các kết quả dưới dạng tệp JSON:**

* Lưu chỉ mục bài hát và thể loại: Danh sách **song\_indices** được lưu vào tệp **song\_indices\_with\_genre.json**.
* Lưu lời bài hát đã qua xử lý: Danh sách **all\_cleaned\_lyrics** được lưu vào tệp **all\_cleaned\_lyrics.json**.

#### 2.3.7.2. Hàm EM



Hình ảnh Hàm EM

Hàm em\_algorithm được sử dụng để thực hiện thuật toán kỳ vọng-tối đa (Expectation-Maximization - EM) nhằm tìm các tham số tốt nhất cho mô hình,

**Input:**

* **X:** Ma trận tài liệu-từ **(document-word matrix)**, trong đó X[i, j] đại diện cho tần suất từ j xuất hiện trong tài liệu i.
* **P\_d\_z:** Ma trận xác suất tài liệu cho từng chủ đề, P\_d\_z[i, k] là xác suất tài liệu i thuộc chủ đề k.
* **P\_w\_z:** Ma trận xác suất từ cho từng chủ đề, P\_w\_z[j, k] là xác suất từ j thuộc chủ đề k.
* **P\_z:** Mảng xác suất của các chủ đề, P\_z[k] là xác suất của chủ đề k.
* **num\_iter:** Số vòng lặp tối đa cho thuật toán (mặc định 1000).
* **tol:** Ngưỡng sai khác giữa hai giá trị khả năng hợp lý (likelihood) liên tiếp để kiểm tra hội tụ (mặc định 1e-5).

**Đầu ra:**

Trả về các ma trận cập nhật sau khi chạy thuật toán: P\_d\_z (xác suất tài liệu-chủ đề), P\_w\_z (xác suất từ-chủ đề), và P\_z (xác suất chủ đề).

**Bước 1: Khởi tạo**

* **n\_docs, n\_words = X.shape:** Xác định số lượng tài liệu (n\_docs) và số lượng từ (n\_words).
* **n\_topics = P\_d\_z.shape[1]:** Lấy số lượng chủ đề từ ma trận P\_d\_z.
* **prev\_likelihood = -np.inf:** Khởi tạo giá trị khả năng hợp lý ban đầu bằng một giá trị rất nhỏ.

**Bước 2: Vòng lặp EM**

Vòng lặp sẽ chạy tối đa **num\_iter** lần hoặc cho đến khi đạt được hội tụ dựa trên tol.

**E-step: Tính P(z|d,w)**

**Mục tiêu**: Tính xác suất của chủ đề z cho một tài liệu d và một từ w, ký hiệu là P\_z\_dw.

**Công thức:**

* **P\_z\_dw = P\_d\_z[i, k] \* P\_w\_z[j, k] \* P\_z[k] / Σ (P\_d\_z[i, k] \* P\_w\_z[j, k] \* P\_z[k])**

Chuyển sang code:

* **P\_z\_dw = np.einsum('ik,jk,k->ijk', P\_d\_z, P\_w\_z, P\_z):** Tính tử số.
* **denominator = np.einsum('ik,jk,k->ij', P\_d\_z, P\_w\_z, P\_z):** Tính mẫu số.
* **P\_z\_dw /= denominator[:, :, np.newaxis] + 1e-10:** Chia tử số cho mẫu số (thêm 1e-10 để tránh chia cho 0).

**M-step: Cập nhật các tham số**

Sau khi tính P(z|d,w) trong bước E, bước M sẽ cập nhật lại các ma trận xác suất:

**Cập nhật P(d|z)**:

* **Công thức:** P\_d\_z[i, k] = Σ\_w P(z|d,w) \* X[d, w]
* **Code:** P\_d\_z = np.einsum('ij,ijk->ik', X, P\_z\_dw)
* **Chuẩn hóa để đảm bảo tổng xác suất của một chủ đề bằng 1:** P\_d\_z /= P\_d\_z.sum(axis=0, keepdims=True)

**Cập nhật P(w|z)**:

* **Công thức:** P\_w\_z[j, k] = Σ\_d P(z|d,w) \* X[d, w]
* **Code:** P\_w\_z = np.einsum('ij,ijk->jk', X, P\_z\_dw)
* **Chuẩn hóa tương tự:** P\_w\_z /= P\_w\_z.sum(axis=0, keepdims=True)

**Cập nhật P(z)**:

* **Công thức:** P\_z[k] = Σ\_d Σ\_w P(z|d,w) \* X[d, w]
* **Code:** P\_z = np.einsum('ij,ijk->k', X, P\_z\_dw)
* **Chuẩn hóa:** P\_z /= P\_z.sum()

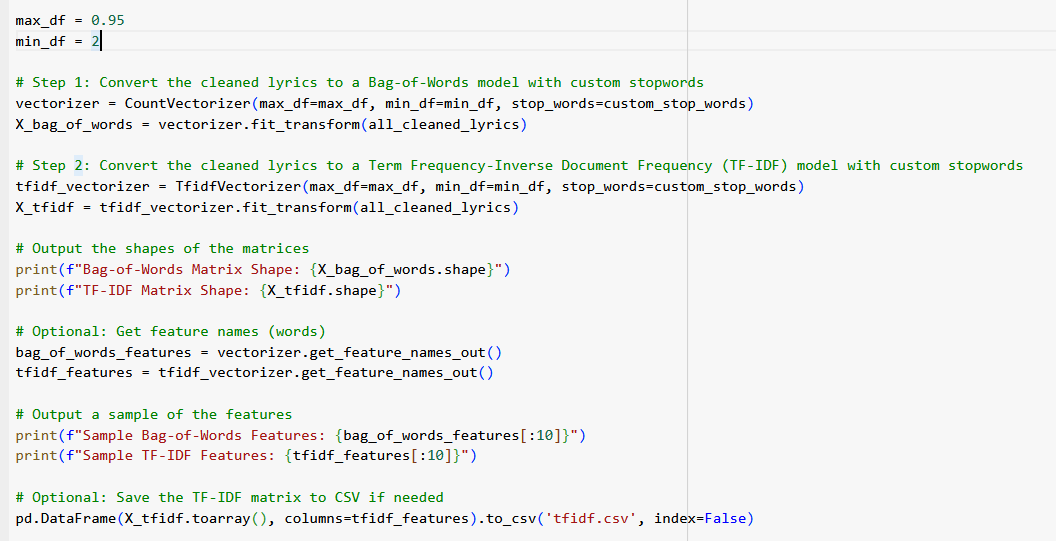
**Tính khả năng hợp lý (Likelihood Calculation):**

* Tính giá trị likelihood để đo mức độ phù hợp giữa mô hình và dữ liệu:
* **estimated\_X = np.dot(P\_d\_z, P\_w\_z.T):** Tính lại ma trận tài liệu-từ được ước tính từ các xác suất đã cập nhật.
* **likelihood = np.sum(X \* np.log1p(estimated\_X)):** Tính log-likelihood (sử dụng log1p để tăng độ ổn định khi giá trị nhỏ).

**Kiểm tra hội tụ (Convergence Check):**

* Nếu sai khác giữa likelihood hiện tại và prev\_likelihood nhỏ hơn ngưỡng tol, vòng lặp sẽ dừng:
* if iteration > 0 and np.abs(likelihood - prev\_likelihood) < tol: Kiểm tra điều kiện hội tụ.
* Nếu hội tụ, in ra thông báo và dừng vòng lặp.

**Bước 3: Trả về kết quả:** Sau khi hội tụ hoặc hoàn thành số vòng lặp tối đa, trả về P\_d\_z, P\_w\_z, và P\_z.



Hình ảnh Chuyển dữ liệu về dạng ma trận Term Frequency-Inverse Document Frequency

**Tham số chính:**

* **max\_df = 0.95:** Loại bỏ các từ xuất hiện trong hơn 95% tài liệu (để tránh các từ quá phổ biến).
* **min\_df = 2:** Chỉ giữ lại các từ xuất hiện ít nhất trong 2 tài liệu (để bỏ qua các từ rất hiếm).
* **custom\_stop\_words:** Danh sách các từ dừng (stopwords) do người dùng định nghĩa sẽ bị loại bỏ trong quá trình tạo vector.

**Các bước thực hiện:**

**Bước 1: Bag-of-Words Model**

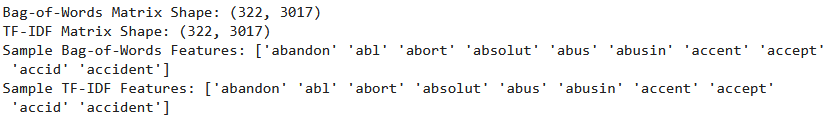
* **CountVectorizer** được sử dụng để chuyển lời bài hát thành mô hình **Bag-of-Words** (túi từ).
* **Bag-of-Words**: Mô hình biểu diễn văn bản dưới dạng ma trận, trong đó mỗi hàng tương ứng với một tài liệu (lời bài hát), và mỗi cột tương ứng với một từ trong từ vựng. Giá trị của mỗi ô là số lần từ đó xuất hiện trong tài liệu.
* **vectorizer.fit\_transform(all\_cleaned\_lyrics):** Chuyển đổi danh sách all\_cleaned\_lyrics thành ma trận túi từ, lưu vào biến X\_bag\_of\_words.

**Bước 2: TF-IDF Model**

* **TfidfVectorizer** được sử dụng để tạo mô hình **Term Frequency-Inverse Document Frequency (TF-IDF)**.
* **TF-IDF**: Là một mô hình biểu diễn văn bản dựa trên tần suất của từ trong một tài liệu so với tần suất của từ đó trong toàn bộ tập hợp tài liệu. Mô hình này giúp giảm trọng số của những từ xuất hiện phổ biến (như các từ dừng) và tăng trọng số cho những từ mang tính chất đặc trưng hơn của tài liệu.
* tfidf\_vectorizer.fit\_transform(all\_cleaned\_lyrics): Chuyển danh sách all\_cleaned\_lyrics thành ma trận TF-IDF, lưu vào biến X\_tfidf.

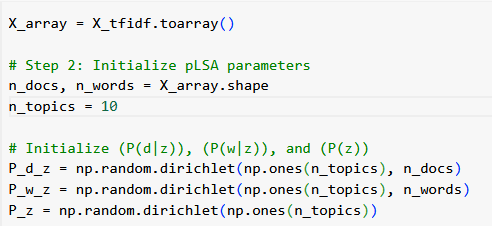
**Xuất thông tin ma trận:**

* **Kích thước ma trận**:
  + X\_bag\_of\_words.shape và X\_tfidf.shape in ra kích thước (số dòng, số cột) của ma trận Bag-of-Words và TF-IDF.
  + Kích thước này tương ứng với (số bài hát, số từ trong từ vựng).



Hình ảnh Kết quả xử lý ma trận TF-IDF

**Kết quả:** Ở đây, chúng em ra được kết 322 document và 3017 từ “duy nhất”



Hình ảnh Các thông số dùng để train model

**X\_tfidf** là ma trận biểu diễn các tài liệu dưới dạng TF-IDF, kết quả từ quá trình vector hóa các lời nhạc. Sử dụng toarrry() để chuyển ma TF-IDF về dạng ma trận thường để model có thể đọc được.

**X\_array.shape**, chúng em lấy số hàng và số cột của ma trận:

* **n\_docs:** Số lượng tài liệu (document) – mỗi tài liệu là một lời nhạc.
* **n\_words:** Số lượng từ duy nhất (unique words) – đại diện cho các đặc trưng (features) trong ma trận TF-IDF.

**n\_topics** là số lượng chủ đề (topics) mà bạn muốn pLSA phân tích và tìm ra trong tập dữ liệu lời nhạc. Ở đây, chúng em chọn 10 topic để phù hợp với dữ liệu.

**P\_d\_z:** Khởi tạo ma trận phân phối chủ đề trên tài liệu (Document-Topic distribution).

**P\_w\_z:** Khởi tạo ma trận phân phối chủ đề trên từ (Topic-Word distribution).

**P\_z:** Phân phối xác suất của chủ đề (Topic distribution), đại diện cho xác suất của mỗi chủ đề trong toàn bộ dữ liệu.

#### 2.3.7.3. Tính topic từ input mới

Sử dụng định lý Bayes, chúng em suy ra chủ đề **z** có khả năng nhất đối với một tập từ **w1,w2,…,wn** ​.

**Áp dụng Định lý Bayes để đưa ra Chủ đề:**

Đối với mỗi từ wiwi​, định lý Bayes cung cấp xác suất của một chủ đề zz cho từ wiwi​, được ký hiệu là **P(z|wi)**

**P(z|wi)=P(wi|z) \* P(z) / P(wi)**

**Trong đó:**

**P(wi|z)** là khả năng: xác suất của từ wiwi​ khi biết chủ đề zz, được học từ dữ liệu huấn luyện.

**P(z)** là xác suất tiên nghiệm: xác suất của chủ đề zz, có thể được coi là phân bố đều (giả sử tất cả các chủ đề có khả năng như nhau) hoặc dựa trên phân bố từ dữ liệu huấn luyện.

**P(wi​)** là bằng chứng: xác suất của từ wiwi​ trên tất cả các chủ đề, đóng vai trò như một hệ số chuẩn hóa.

**Để tìm ra chủ đề có xác suất tốt nhất:**

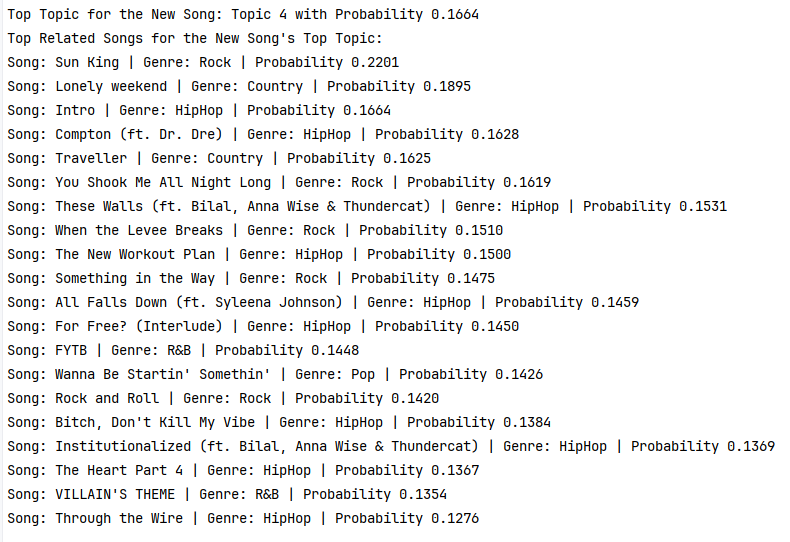
1. **Đối với mỗi chủ đề z:**

* Tính xác suất tiên nghiệm **P(z)**
* Tính khả năng **P(wi∣z)** cho mỗi từ **wi** trong tập từ.

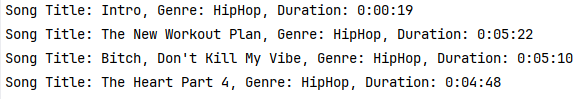
1. **Nhân các xác suất này với nhau**
2. **Chọn chủ đề z có xác suất hậu nghiệm cao nhất.**

# CHƯƠNG 3: TRIỂN KHAI VÀ CÀI ĐẶT (IMPLEMENT)

## 3.1. Kết quả Implement kiến trúc



Hình ảnh Kết quả dự đoán của model



Hình ảnh Kết quả đề xuất

Sau khi model cho ra kết quả dự đoán, các kết quả này sẽ đi qua bước tiếp theo để được lọc theo thể loại của bài hát ban đầu được đùng để đề xuất. Từ kết quả này, chúng em sử dụng query để đối chiếu với dữ liệu có sẵn trong database và đưa ra các đề xuất tương ứng.

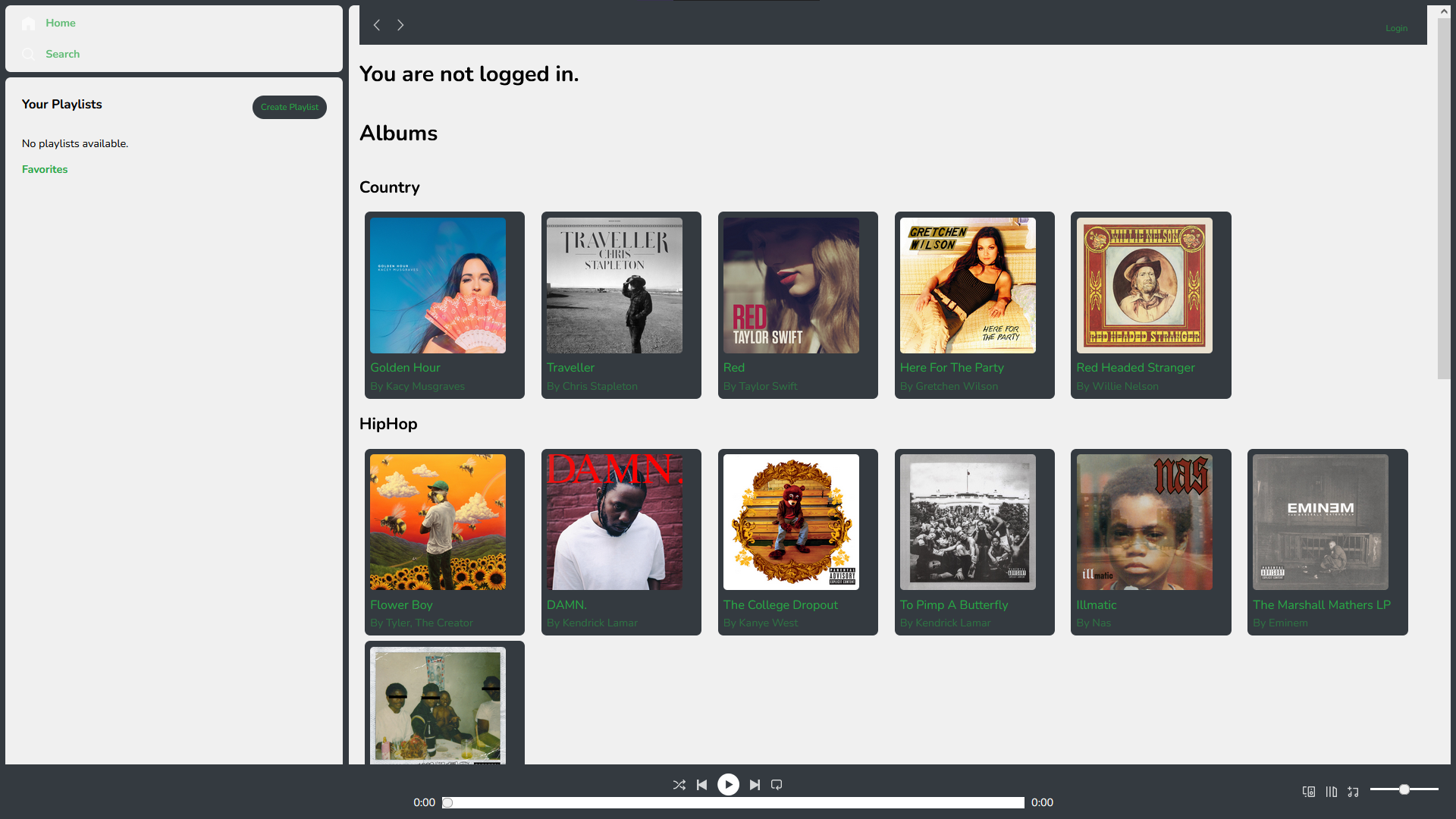


Hình ảnh Kết quả khi tạo Playlist thành công

Khi tạo Playlist thành công, người dùng sẽ được chuyển hướng về Trang chủ. Ứng dụng sẽ lấy thông tin danh sách từ database và hiện lên ở sidebar trên trái.

## 3.2. Kết quả Implement UI

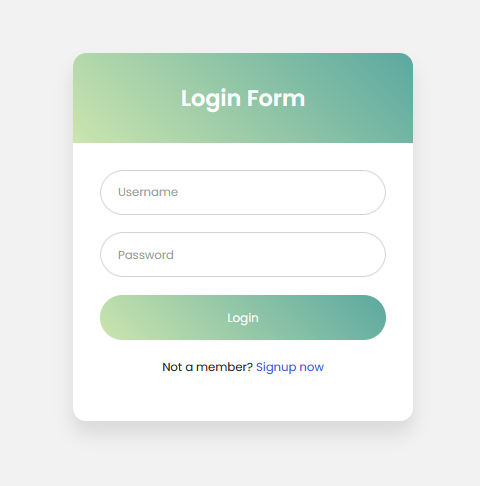
### 3.2.1. Main (Chưa login)



Hình ảnh Trang Chủ khi chưa đăng nhập

Người dùng có thể xem trang web khi chưa đăng nhập nhưng sẽ không có các danh sách nhạc ở bên trái vì chưa tạo tài khoản.

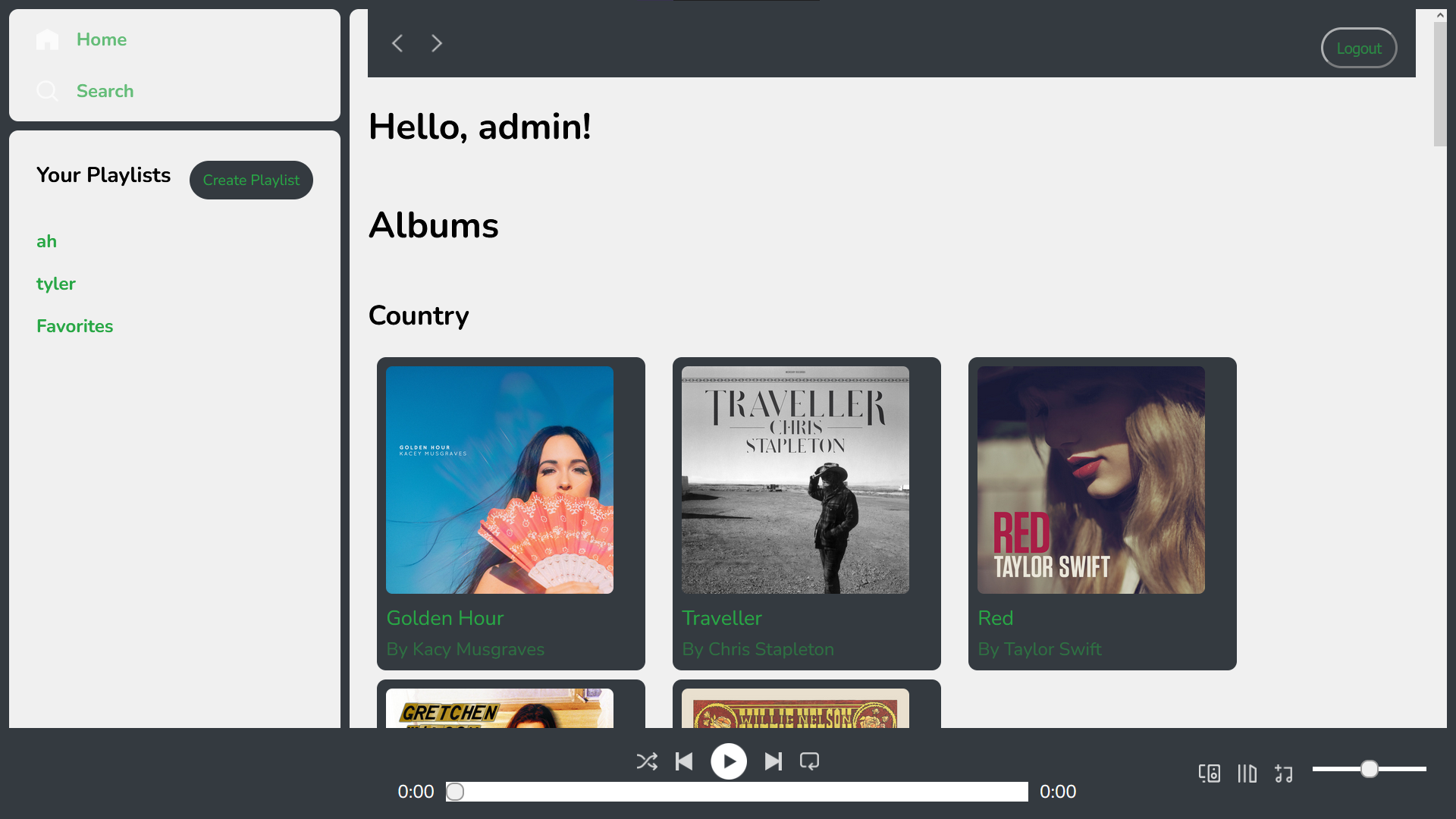
### 3.2.2. Giao diện login



Hình ảnh Trang Login

Người dùng có thể đăng nhập nếu đã có tài khoản, nếu không thì có thể sử dụng nút **Signup now** để đưa tới trang đăng ký và tạo tài khoản mới.

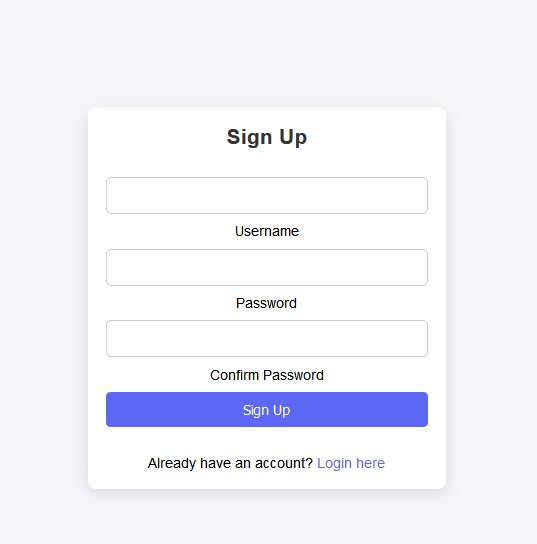
### 3.2.3. Main (Đã login)



Hình ảnh Trang chủ khi đã đăng nhập

Khi đăng nhập thành công thì người dùng có thể xem trang web như bình thường, ngoài ra thì còn có thể coi danh sách nhạc thuộc về người dùng ở thanh bên trái.

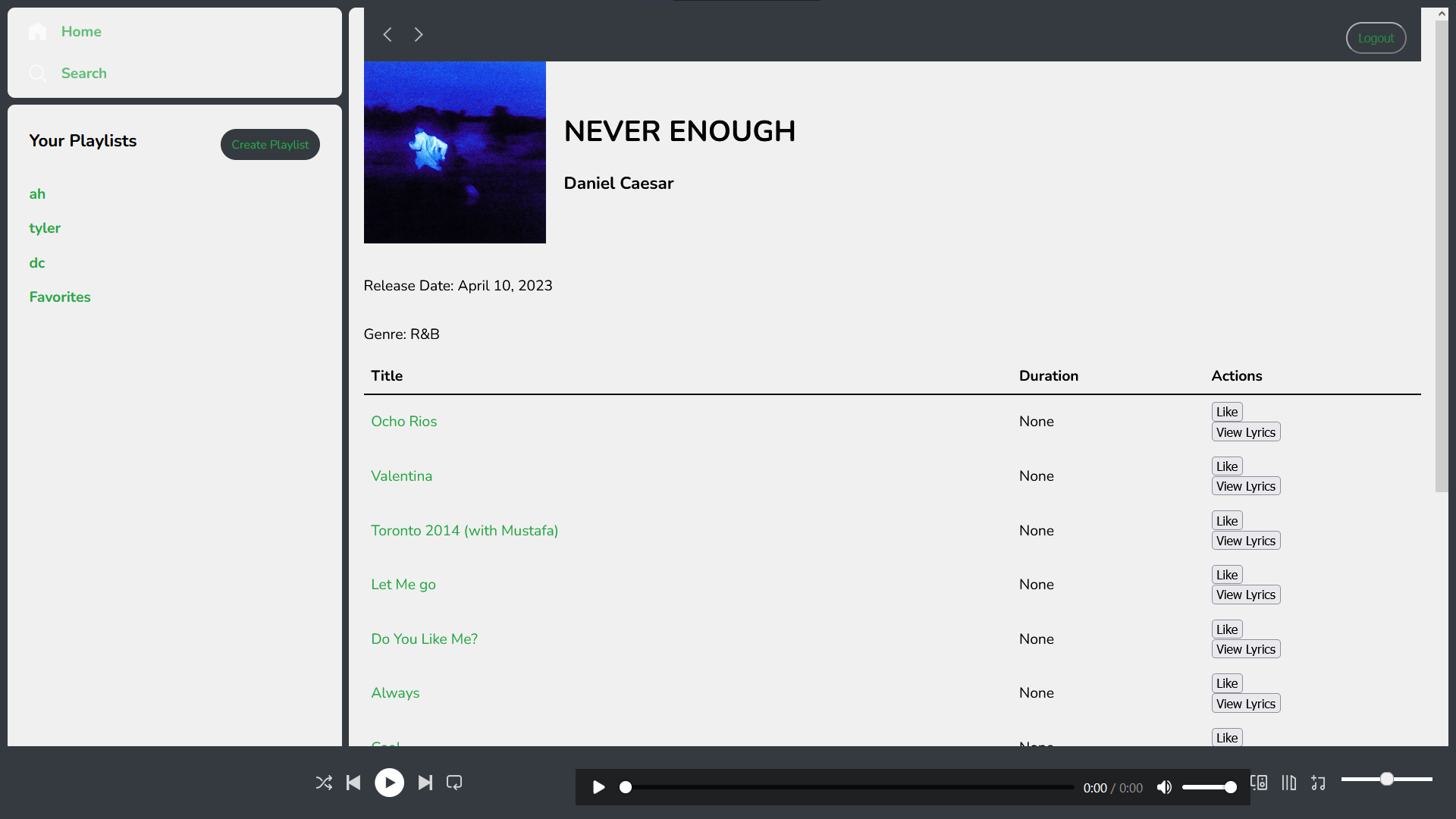
### 3.2.4. Giao diện đăng kí tài khoản



Hình ảnh Trang Đăng ký

Người dùng có thể đăng ký tài khoản bằng cách nhập vào tên đăng nhập và mật khẩu.

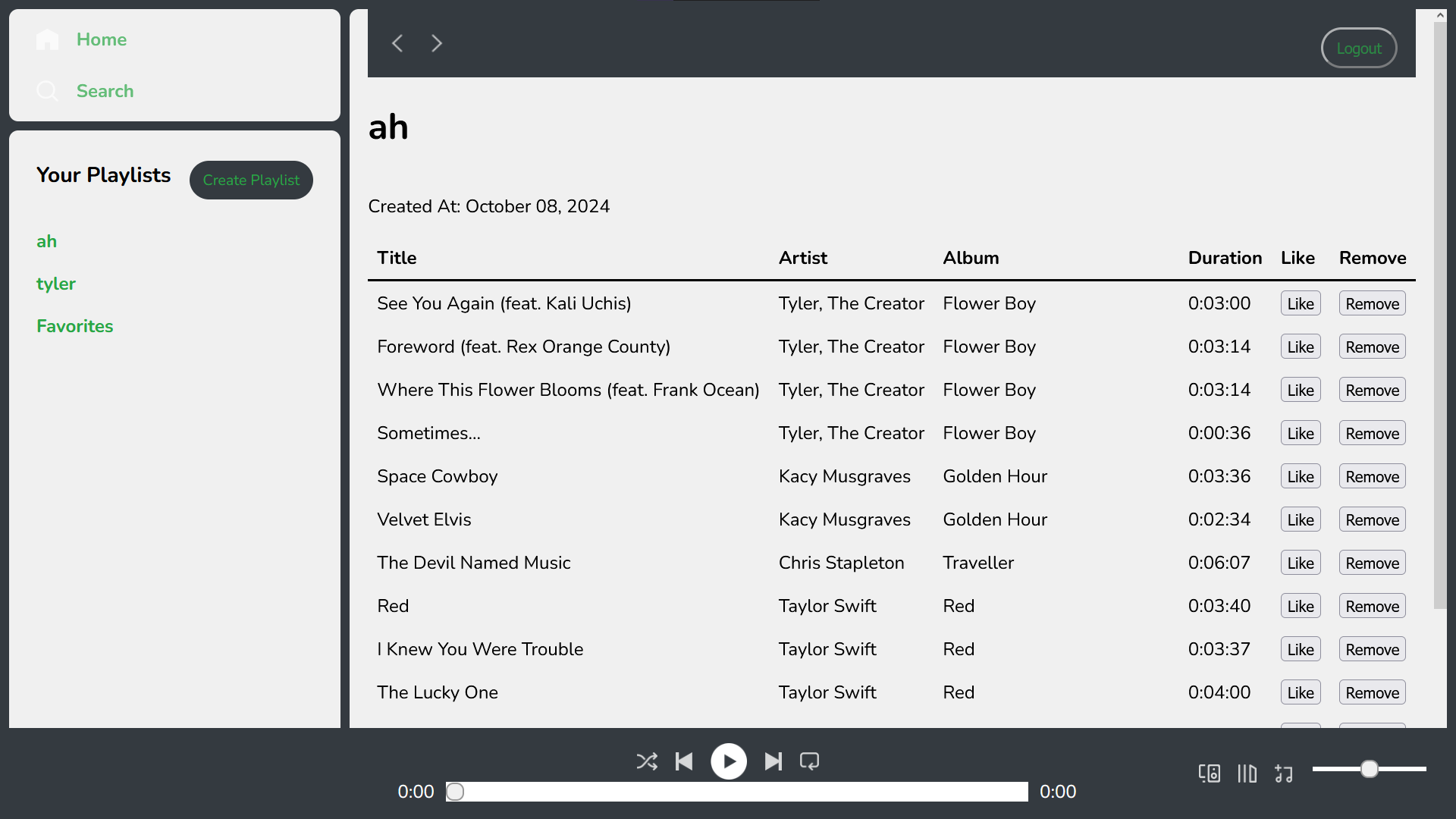
### 3.2.5. Giao diện một album



Hình ảnh Trang Album detail

Người dùng có thể xem các bài hát của album trên, sử dụng nút **Like** để đưa bài hát vào danh sách nhạc thích hoặc dùng nút **View Lyrics** để coi lời bài hát. Ngoài ra thì bấm vào bài hát nào đó sẽ bắt bầu chơi bài hát đó.

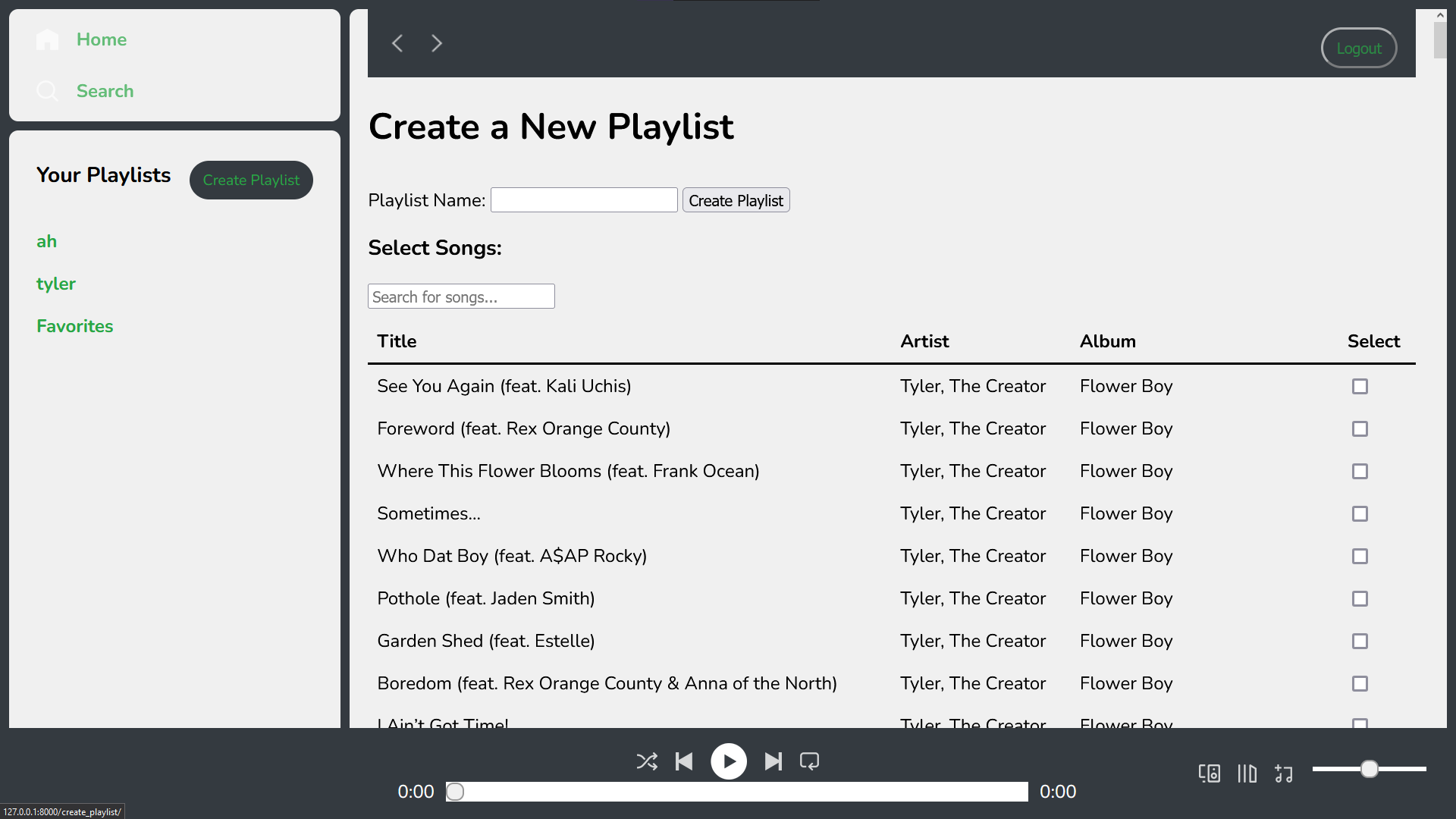
### 3.2.6. Giao diện một playlist người dùng tự tạo



Hình ảnh Trang Playlist Detail

Người dùng có thể coi các bài hát trong danh sách nhạc mà họ đã tạo, còn có thể đưa các bài hát đó vào danh sách nhạc thích qua nút **Like** hoặc xóa bỏ bài hát đó khỏi danh sách đang coi bằng nút **Remove.**

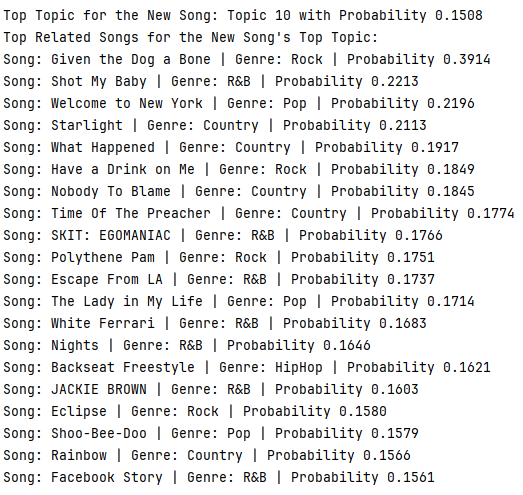
### 3.2.7. Giao diện tạo playlist



Hình ảnh Trang Create Playlist

Người dùng có thể chọn bài hát tùy thích để thêm vào danh sách, và dùng thanh tìm kiếm để tìm theo tên bài hát, tên nhạc sĩ hoặc tên album mong muốn.

## 3.3. Kết quả Implement model

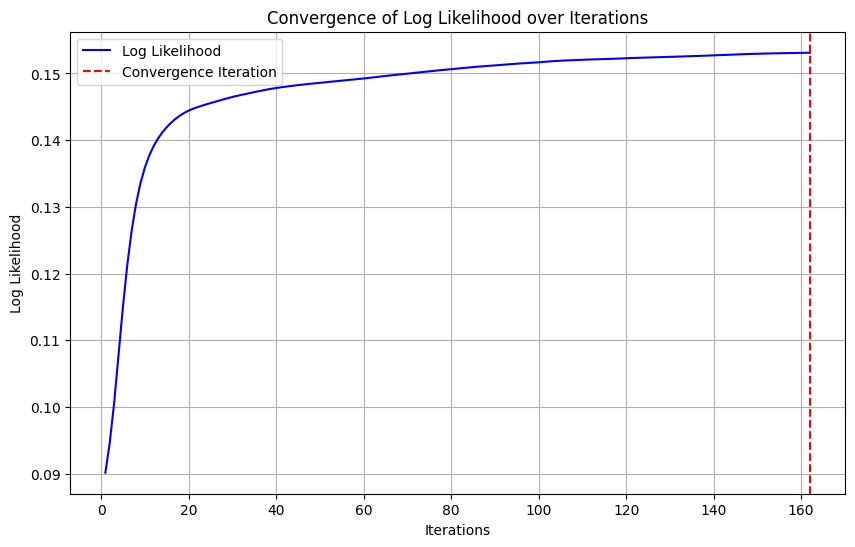


Hình ảnh Kết quả dự đoán của model

Sau khi lấy xử lý lời nhạc của bài hát, model đã trả về các bài hát tương ứng cùng với thể loại và xác suất của chúng. Từ các kết quả trên, chúng em có thể tiếp tục lọc các bài hát dựa trên thể loại để đưa ra đề xuất phù hợp.

# CHƯƠNG 4: VẬN HÀNH (OPERATE)

## 4.1. Kết quả

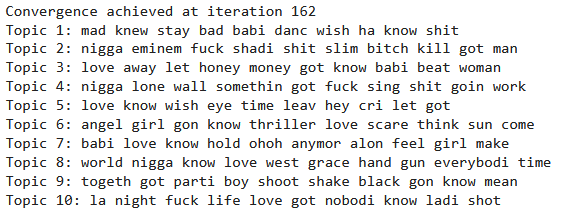


Hình ảnh Quá trình hội tụ

**Likelihood** ở đây là một thước đo cho thấy mức độ phù hợp của mô hình đối với dữ liệu đã quan sát. Nó biểu thị khả năng rằng dữ liệu được quan sát sẽ xảy ra với các tham số hiện tại của mô hình. Qua biểu đồ trên, quá trình hội tụ của likelihood qua từng vòng lặp ở các điểm:

* **Giai đoạn đầu (0-20 lần lặp)**: Giá trị khả năng tăng nhanh cho thấy mô hình đang bắt đầu học từ dữ liệu.
* **Giai đoạn giữa (từ lần lặp 20 đến 60)**: Giai đoạn này sự quá trình cải thiện chậm lại. Mô hình tiếp tục cải thiện và điều chỉnh các tham số để phù hợp với dữ liệu tốt hơn.
* **Giai đoạn cuối (từ lần lặp 60 đến 162)**: Mức tăng của giá trị khả năng bắt đầu giảm dần. Đến gần cuối (sau lần lặp 150), giá trị khả năng gần như ổn định, cho thấy mô hình đã tiến tới hội tụ.
* **Hội tụ**: Hội tụ xảy ra sau lần lặp thứ 162, khi giá trị khả năng không còn thay đổi đáng kể.

Tóm lại, mô hình đạt đến hội tụ sau 162 lần lặp, với sự cải thiện dần dần trong giá trị khả năng ở những vòng lặp đầu và sau đó ổn định khi gần đạt tới sự hội tụ.



Hình ảnh Kết quả train Model

Sau iteration 162 thì model đã đạt được sự hội tụ, model đạt trạng thái tối ưu và không còn cải thiện thêm.

Có **10 chủ đề** được rút trích từ các bài hát, với mỗi chủ đề được mô tả bằng những từ quan trọng nhất trong các bài hát thuộc về chủ đề đó. Mỗi chủ đề chứa các từ đặc trưng, và bạn có thể phân tích từng chủ đề như sau:

* **Topic 1**: Liên quan đến cảm xúc mạnh mẽ, các từ như "mad," "knew," "bad," "dance," "shit" có thể đại diện cho các bài hát liên quan đến sự nổi loạn, căng thẳng.
* **Topic 2**: Chứa các từ tục tiũ liên quan đến rap hoặc hip-hop, đặc biệt là phong cách của Eminem.
* **Topic 3**: Các từ như "love," "away," "honey," "money," cho thấy đây có thể là chủ đề tình yêu và mối quan hệ.
* **Topic 4**: Chứa các từ tục tiũ có thể phản ánh các bài hát hip-hop hoặc rap với chủ đề về cuộc sống và thách thức.
* **Topic 5**: Có nhiều từ liên quan đến cảm xúc ("love," "cry," "time"), thể hiện các bài hát về tình cảm và sự mất mát.
* Các chủ đề khác cũng tương tự, mỗi chủ đề thể hiện một nhóm ý nghĩa hoặc cảm xúc trong tập bài hát.

## 4.2. Thảo luận

Giá trị về nghiên cứu mà pLSA mang lại:

* Phát hiện chủ đề ẩn: pLSA giúp khám phá các chủ đề ẩn trong lời bài hát, từ đó tìm ra những mẫu nội dung chung, các cảm xúc hoặc thông điệp chính trong một tập hợp các bài hát. Điều này hữu ích trong việc phân loại âm nhạc theo nội dung hoặc cảm xúc, hỗ trợ các hệ thống gợi ý nhạc dựa trên sở thích cá nhân.
* Phân tích ngữ nghĩa: Trong phân tích lời nhạc, pLSA có khả năng học ra các mô hình từ các cụm từ liên quan đến nhau trong lời bài hát. Điều này giúp phát hiện các mối liên hệ giữa từ ngữ và chủ đề, từ đó đưa ra những kết quả mang tính ngữ nghĩa sâu sắc, giúp hiểu hơn về cách mà các nghệ sĩ diễn đạt các ý tưởng hay cảm xúc qua lời nhạc.
* Gợi ý âm nhạc và nội dung dựa trên ngữ nghĩa: pLSA có thể được ứng dụng trong các hệ thống gợi ý âm nhạc. Ví dụ, dựa trên việc phân tích các chủ đề từ lời nhạc, hệ thống có thể gợi ý những bài hát có nội dung tương tự hoặc phù hợp với tâm trạng hoặc sở thích của người dùng.
* Phân tích tác động cảm xúc: Thông qua việc khám phá các chủ đề cảm xúc trong lời bài hát (ví dụ: tình yêu, cô đơn, vui vẻ, giận dữ), pLSA giúp hiểu rõ hơn tác động của bài hát đối với người nghe. Các nhà nghiên cứu có thể sử dụng các chủ đề này để nghiên cứu cách âm nhạc ảnh hưởng đến cảm xúc con người.

# KẾT LUẬN

## Ưu điểm

**1. Cá nhân hóa trải nghiệm:**

Phân tích sở thích âm nhạc: pLSA có thể phân tích các bài hát mà người dùng thường nghe và xác định những chủ đề tiềm ẩn trong lời bài hát hoặc các thuộc tính khác của bài hát (thể loại, nghệ sĩ, album). Từ đó, hệ thống sẽ xây dựng một hồ sơ về sở thích âm nhạc của người dùng. Điều này giúp cá nhân hóa trải nghiệm âm nhạc, cung cấp các đề xuất bài hát và thể loại phù hợp với gu âm nhạc của từng cá nhân.

Đề xuất theo thời gian thực: Dựa trên lịch sử nghe và các hành động của người dùng như thêm bài hát vào playlist, hệ thống sẽ tự động cập nhật các đề xuất để giữ cho chúng luôn phù hợp và mới mẻ.

**2. Khám phá bài hát mới:**

Gợi ý bài hát mới: Hệ thống có thể không chỉ gợi ý những bài hát mà người dùng đã nghe hoặc thích, mà còn khám phá các bài hát mới thuộc về những chủ đề tiềm ẩn mà họ chưa tiếp cận. Điều này mở ra cơ hội cho người dùng khám phá nhiều phong cách và thể loại âm nhạc khác nhau.

Chủ đề tiềm ẩn: Dựa trên các chủ đề tiềm ẩn từ pLSA, hệ thống có thể phát hiện các bài hát liên quan đến những chủ đề mà người dùng có thể yêu thích nhưng chưa từng nghe qua. Điều này giúp tạo ra trải nghiệm âm nhạc đa dạng và phong phú hơn.

## Nhược điểm

**1. Dataset nhỏ:**

Phát hiện nhiều chủ đề hơn: Mặc dù dataset nhỏ có thể hạn chế trong việc tìm ra các chủ đề phong phú, nhưng vẫn có khả năng phát hiện các chủ đề đặc biệt hoặc ít phổ biến hơn. pLSA vẫn có thể tìm ra các mối liên kết giữa các bài hát thuộc những chủ đề tiềm ẩn, giúp người dùng khám phá những khía cạnh âm nhạc ít được biết đến nhưng có giá trị.

Tinh chỉnh chủ đề: Với một dataset nhỏ, quá trình tinh chỉnh các chủ đề có thể trở nên rõ ràng hơn, giúp tập trung vào các nhóm bài hát có nội dung hoặc phong cách âm nhạc rất đặc trưng. Điều này có thể giúp hệ thống gợi ý những bài hát với chủ đề đặc biệt phù hợp với sở thích của từng người dùng.

**2. Thiếu cá nhân hóa sâu:**

Chưa phân tích ngữ cảnh hoặc cảm xúc: Một điểm hạn chế của hệ thống là chưa phân tích được ngữ cảnh hoặc cảm xúc trong các bài hát, dẫn đến những đề xuất có thể chưa hoàn toàn phù hợp với tình trạng cảm xúc hoặc tình huống của người dùng. Việc chỉ dựa vào các chủ đề tiềm ẩn có thể không đủ để nắm bắt toàn bộ sắc thái của âm nhạc và sở thích cá nhân.

Cải thiện phân tích ngữ nghĩa: Một cách mở rộng tiềm năng là tích hợp thêm mô hình phân tích cảm xúc (sentiment analysis) hoặc các mô hình ngữ nghĩa sâu hơn để đánh giá tâm trạng của bài hát, từ đó cung cấp đề xuất chính xác hơn dựa trên cảm xúc của người dùng tại một thời điểm cụ thể.

**3. Thiếu chức năng và giao diện chưa hoàn thành:**

Tập trung vào phần cốt lõi: Do hạn chế về thời gian, dự án đã dành phần lớn nỗ lực cho việc phát triển các chức năng cốt lõi, chẳng hạn như việc phân tích và đề xuất bài hát dựa trên các chủ đề tiềm ẩn. Tuy nhiên, giao diện người dùng và các tính năng bổ sung khác vẫn chưa được hoàn thiện, dẫn đến trải nghiệm người dùng có thể chưa được mượt mà hoặc đầy đủ.

Kế hoạch phát triển giao diện: Giao diện có thể được mở rộng trong tương lai với các tính năng như tùy chọn lọc theo cảm xúc, thêm bảng điều khiển trực quan để người dùng theo dõi các đề xuất và thông tin chi tiết hơn về các chủ đề tiềm ẩn mà họ đã tương tác.

## Hướng phát triển

**1. Hoàn thiện giao diện và chức năng:**

**Thêm các chức năng cần thiết**: Để hoàn thiện ứng dụng, việc bổ sung một số chức năng chính là điều cần thiết, như:

**Tùy chọn lọc nâng cao:** Cho phép người dùng lọc bài hát theo nhiều tiêu chí như tâm trạng, thể loại, nghệ sĩ, hoặc chủ đề.

**Chức năng tìm kiếm:** Nâng cao tính năng tìm kiếm để người dùng dễ dàng khám phá thêm nghệ sĩ, album, hoặc các bài hát mới.

**Lịch sử nghe:** Lưu trữ và hiển thị lịch sử nghe của người dùng, từ đó có thể đề xuất các bài hát tương tự hoặc gợi ý quay lại những bài hát họ yêu thích trước đó.

**Hoàn thiện giao diện:** Để tạo ra một ứng dụng bắt mắt và thân thiện với người dùng:

**Giao diện hiện đại và trực quan:** Tối ưu hóa bố cục, sử dụng các màu sắc và hiệu ứng trực quan như chuyển động mượt mà khi duyệt playlist, chuyển cảnh nhẹ nhàng giữa các trang.

**Cá nhân hóa giao diện:** Cho phép người dùng tùy chỉnh giao diện theo sở thích, chẳng hạn như đổi màu nền, kiểu phông chữ, hoặc bố cục màn hình.

**Hiển thị thông tin trực quan:** Thêm biểu đồ hoặc bảng xếp hạng để hiển thị các bài hát được nghe nhiều nhất, các thể loại phổ biến, hoặc mức độ tương tác của người dùng với các chủ đề âm nhạc.

**2. Dataset lớn hơn:**

**Mở rộng dataset:** Để cải thiện khả năng phân tích và gợi ý của mô hình pLSA, việc thu thập thêm nhiều bài hát và lời nhạc là cần thiết. Một dataset lớn hơn có thể giúp hệ thống:

**Phát hiện nhiều chủ đề hơn:** Khi có thêm dữ liệu, mô hình pLSA sẽ phát hiện ra thêm nhiều chủ đề tiềm ẩn và phong phú hơn. Điều này giúp gợi ý âm nhạc đa dạng hơn và phù hợp hơn với sở thích của người dùng.

**Tăng độ chính xác của đề xuất:** Dataset lớn hơn giúp mô hình pLSA hiểu rõ hơn về mối quan hệ giữa các bài hát, thể loại và cảm xúc. Từ đó, chất lượng đề xuất sẽ được cải thiện, mang đến cho người dùng những gợi ý sát với sở thích cá nhân hơn.

**Mở rộng ra các ngôn ngữ và thể loại âm nhạc khác:** Thêm dữ liệu từ các nền tảng âm nhạc quốc tế, các bài hát bằng nhiều ngôn ngữ khác nhau, và các thể loại âm nhạc đa dạng (như rock, pop, jazz, rap) để đáp ứng nhu cầu của người dùng toàn cầu.

**Thu thập và làm sạch dữ liệu:** Khi thu thập dữ liệu từ nhiều nguồn khác nhau, cần có quy trình làm sạch dữ liệu để đảm bảo rằng các bài hát được đưa vào dataset có chất lượng tốt, thông tin đầy đủ và không bị trùng lặp. Điều này cũng giúp mô hình hoạt động chính xác hơn.

**3. Mở rộng:**

**Tích hợp thêm các API âm nhạc:** Sử dụng các API như Spotify, Musixmatch, hoặc Last.fm để thu thập dữ liệu lời nhạc, thông tin nghệ sĩ và album. Điều này không chỉ giúp mở rộng dataset mà còn tăng tính phong phú và đa dạng của hệ thống.

**Phân tích chuyên sâu:** Với dataset lớn hơn, mô hình có thể không chỉ phân tích lời bài hát mà còn các yếu tố khác như thời lượng bài hát, mức độ phổ biến, hoặc các xu hướng âm nhạc đang thịnh hành. Điều này giúp hệ thống đề xuất âm nhạc theo phong cách hiện đại và sát với thị hiếu của người dùng.

**Giao diện tương tác thời gian thực:** Tích hợp chức năng đề xuất ngay lập tức khi người dùng tương tác với một bài hát, playlist, hoặc nghệ sĩ cụ thể, giúp trải nghiệm nghe nhạc trở nên liền mạch và hấp dẫn hơn.

# TÀI LIỆU THAM KHẢO

Playlist cơ bản về Django (Tiếng Anh): <https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBlmzzFcLgDhKTTfNLfX1IK>

Tìm hiểu về pLSA:

<https://www.youtube.com/watch?v=J1ri0EQnUOg>

Mẫu giao diện Web

<https://github.com/AVIVASHISHTA29/spotify-clone-html>

Mẫu kiến trúc pLSA

<https://github.com/laserwave/plsa>