**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỒ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A picture containing text

Description automatically generated**

**ĐỒ ÁN HỌC PHẦN**

TÊN HỌC PHẦN: **LẬP TRÌNH ÂM THANH   
 (Audio-Application Programming: AudioPro)**

MÃ SỐ LỚP HP: **MUL14126-01**

Tên đề tài: **LẬP TRÌNH HỆ THỐNG QUẢN LÝ NỀN TẢNG NHẠC GIẢI TRÍ**

Họ tên sinh viên: **Lương Quốc Diễn**

**Mã số sinh viên: N20DCPT009**

**Lớp: D20CQPTUD01-N**

**Ngày nộp: ……./……./2023**

**Ký tên:**

**TP.HCM, ngày 11 tháng 03 năm 2023**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỒ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI ĐỒ ÁN HỌC PHẦN**

**Giảng viên giảng dạy: VÕ XUÂN THỂ**

TÊN HỌC PHẦN: **LẬP TRÌNH ÂM THANH   
 (Audio-Application Programming: AudioPro)**

MÃ SỐ LỚP HP: **MUL14126-01**

Tên đề tài: **LẬP TRÌNH HỆ THỐNG QUẢN LÝ NỀN TẢNG NHẠC GIẢI TRÍ**

Họ tên sinh viên: **Lương Quốc Diễn**

**Mã số sinh viên: N20DCPT009**

**Lớp: D20CQPTUD01-N**

**Tên sản phẩm đề tài: AP03LQDIEN**

**Công cụ sử dụng (phiên bản): Android Studio, Visual Studio Code  
Bộ đóng gói Setup**: **Inno Setup Compiler**

**Thư viên (phiên bản): Framework Flutter (Dart 3.0). IDE Android Studio. Package audioplayers, Package Bloc. Package Google Speech**

**Ngôn ngữ lập trình sử dụng: Dart**

**DataBase lưu thông tin của người dùng và lịch sử tìm kiếm: MySQL**

**Tập dữ liệu thực nghiệm:** [**Zingmp3**](https://www.freesoundeffects.com)

**Nhận xét của giảng viên:**

**………………………………………………………..**

**Điểm đánh giá:………….(…………………)**

**Ngày……./……../2023**

**Giảng viên: Ký tên**

**TP.HCM, ngày 11 tháng 03 năm 2023**

# 

# LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành nhất đến thầy Võ Xuân Thể. Trong quá trình học tập và tìm hiểu bộ môn Lập trình âm thanh, em đã nhận được sự quan tâm giúp đỡ, hướng dẫn rất tận tình, tâm huyết của thầy. Thầy đã giúp em tích lũy thêm nhiều kiến thức để có thể bước đi trên con đường sự nghiệp của em sau này. Từ những kiến thức mà thầy truyền tải, em đã dần trở nên giỏi hơn và tốt hơn trong học phần này. Thông qua bản báo cáo này, em xin trình bày lại những gì mà mình đã làm đối với đồ án học phần môn Lập trình âm thanh với tên đề tài là “**LẬP TRÌNH HỆ THỐNG QUẢN LÝ NỀN TẢNG NHẠC GIẢI TRÍ**”.

Có lẽ kiến thức là vô hạn mà sự tiếp nhận kiến thức của bản thân mỗi người luôn tồn tại những hạn chế nhất định. Do đó, trong quá trình hoàn thành bài tiểu luận, chắc chắn không tránh khỏi những thiếu sót. Bản thân em rất mong nhận được những góp ý đến từ thầy để bài báo cáo của em được hoàn thiện hơn.

Kinh chúc thầy sức khỏe, hạnh phúc thành công trên con đường sự nghiệp giảng day.

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
|  | |
| CSDL hoặc DB | Cơ sở dữ liệu: DataBase |
| NSD = User = Account | Quyền làm việc được cấp cho NGƯỜI SỬ DỤNG trên hệ thống App, quyền này thông thường được đảm bảo bởi tên đăng nhập (username) và mật khẩu (password); cũng có thể gọi là “tài khoản người dùng” [“tài khoản”] |
|  |  |
| Flutter | Framework lập trình đa nền tảng Flutter sử dụng ngôn ngữ Dart được Goolge tạo ra. |
| BE | Backend: Lập trình backend là lập trình phía server để xử lý, lưu trữ và phục vụ dữ liệu cho phía frontend. Thông thường phía backend sẽ bao gồm database (cơ sở dữ liệu) và các chương trình (application hoặc service) chạy trên một hoặc nhiều servers cùng kết nối, tương tác với nhau. |

# DANH MỤC CÁC THUẬT NGỮ

|  |  |
| --- | --- |
| Giao diện người dùng  (User-Interface) | Là hệ thống các màn hình giao tiếp cho phép người sử dụng tương tác với các thành phần phần mềm, điều khiển phần mềm hoạt động theo yêu cầu của người dùng - tương ứng các chức năng hiện có của phần mềm. |
| Người dùng (User)  NSD: Tài khoản (Account) | Là một quyền làm việc trên hệ thống phần mềm được cấp phát cho một cá nhân thông qua tên tài khoản (username) và mật khẩu (password). |
| RESTful API | **RESTful API** là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource |
| JSON (Javascript Object Notation) | Là một kiểu định dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được. [**JSON**](https://topdev.vn/viec-lam-it/json-kt35)là một tiêu chuẩn mở để trao đổi dữ liệu trên web. |
| JWT (Json Web Token) | JWT là cơ chế xác thực người dùng sử dụng các hàm băm để mã hóa và giải mã đoạn String để xác thực người dùng. |

# MỤC LỤC

[LỜI CẢM ƠN 3](#_Toc123600727)

[DANH MỤC CÁC TỪ VIẾT TẮT 4](#_Toc123600728)

[DANH MỤC CÁC THUẬT NGỮ 5](#_Toc123600729)

[MỤC LỤC 6](#_Toc123600730)

[PHIẾU GIAO NHIỆM VỤ ĐỀ TÀI ĐỒ ÁN HỌC PHẦN 8](#_Toc123600731)

[Chương 1: GIỚI THIỆU VỀ ĐỀ TÀI 9](#_Toc123600732)

[**1.1.** Tổng quan về đề tài 9](#_Toc123600733)

[**1.2.** Nội dung chuyên môn chính của đề tài 9](#_Toc123600734)

[**1.3.** Bố cục của báo cáo 9](#_Toc123600735)

[Chương 2: CƠ SỞ LÝ THUYẾT VỀ LẬP TRÌNH ÂM THANH 10](#_Toc123600736)

[**2.1.** Tổng quan về âm thanh học và kỹ thuật âm thanh 10](#_Toc123600737)

[2.1.1. Các khái niệm 10](#_Toc123600738)

[2.1.2. Thực tiễn của âm thanh 10](#_Toc123600739)

[2.1.3. Khuynh hướng kỹ thuật âm thanh 10](#_Toc123600740)

[2.1.4. Các đặc tính cơ bản của âm thanh 10](#_Toc123600741)

[2.1.5. Các đặc tính cơ bản của file lưu trữ âm thanh (Digital) 11](#_Toc123600742)

[2.1.6. Các dạng đuôi (phần mở rộng || Extensions) file âm thanh 11](#_Toc123600743)

[2.1.7. Các dạng nén file âm thanh thông dụng dựa vào BitRate 16](#_Toc123600744)

[2.1.8. Một số dạng xử lý chất lượng âm thanh 16](#_Toc123600745)

[2.1.9. Một số kỹ thuật thu thập âm thanh (Thu âm) 16](#_Toc123600746)

[2.1.10. Các xử lý cơ bản trên tập tin âm thanh kỹ thuật số 16](#_Toc123600747)

[2.1.11. Các xử lý cơ bản với đặc tính âm thanh kỹ thuật số 16](#_Toc123600748)

[**2.2.** Công cụ và các hệ thống thư viện hỗ trợ Audio Pro… 17](#_Toc123600749)

[2.2.1. Phần mềm công cụ sử dụng chính trong học phần 17](#_Toc123600750)

[2.2.2. Các thư viện lập trình chính được sử dụng trong học phần 17](#_Toc123600751)

[2.2.3. Tập các âm thanh mẫu được sử dụng trong đề tài của Học phần 18](#_Toc123600752)

[Chương 3: GIỚI THIỆU CÁC THƯ VIỆN AudioPro SỬ DỤNG 19](#_Toc123600753)

[3.1. Nạp các thư viện xử lý âm thanh (Audio Libraries) 19](#_Toc123600754)

[3.2. Nạp BỘ CÔNG CỤ của các thư viện 20](#_Toc123600755)

[Chương 4: GIỚI THIỆU VÀ CÀI ĐẶT SẢN PHẨM …<tên đề tài>…….. 21](#_Toc123600756)

[4.1. Mô tả sản phẩm 21](#_Toc123600757)

[4.2. Cài CSDL lưu trữ thông tin âm thành 21](#_Toc123600758)

[4.2.1. Lập DB SQL chỉ 1 Table: 21](#_Toc123600759)

[4.2.2. Nhập dữ liệu thực nghiệm: 22](#_Toc123600760)

[4.3. Lập Audio App Winform mới: APsttVXThe 23](#_Toc123600761)

[4.4. Thiết kế và Lập trình Winform màn hình chính: FrMain.cs 23](#_Toc123600762)

[4.4.1. Thiết kế (Design) 23](#_Toc123600763)

[4.4.2. Lập trình (Codes) 25](#_Toc123600764)

[4.5. Thiết kế và Lập trình Winform màn hình quản lý âm thanh dùng WMPLib 27](#_Toc123600765)

[4.5.1. Thiết kế (Design) 27](#_Toc123600766)

[4.5.2. Một số số mã lệnh ban đầu (Codes) 30](#_Toc123600767)

[4.6. Sử dụng công cụ trong Toolbox (đối với một số thư viện có hỗ trợ công cụ) 30](#_Toc123600768)

[4.7. Lập trình (Codes) XỬ LÝ âm thanh sử dụng thư viện 32](#_Toc123600769)

[4.8. Lập trình (Codes) QUẢN LÝ âm thanh sử dụng thư viện 33](#_Toc123600770)

[4.8.1. Giới thiệu 33](#_Toc123600771)

[4.8.2. Nạp thêm file âm thanh (Load) 34](#_Toc123600772)

[4.8.3. Sửa thông tin file âm thanh (Modify) 37](#_Toc123600773)

[4.8.4. Ghi ân file âm thanh (Record) 39](#_Toc123600774)

[4.8.5. Xóa file âm thanh (Delete) 39](#_Toc123600775)

[4.9. Lập trình (Codes) XỬ LÝ âm thanh sử dụng thư viện irrKlang 40](#_Toc123600776)

[4.9.1. Thiết kế Forms 40](#_Toc123600777)

[4.9.2. Lập trình (Codes) 41](#_Toc123600778)

[Chương 5: ĐÓNG GÓI BỘ CÀI ĐẶT SETUP CỦA SẢN PHẨM ĐỀ TÀI 41](#_Toc123600779)

[5.1. Giới thiệu chung 41](#_Toc123600780)

[5.2. Công cụ sử dụng thiết lập bộ cài đặt Setup cho App 42](#_Toc123600781)

[5.3. Thiết lập bộ cài đặt Setup cho App 42](#_Toc123600782)

[Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI 45](#_Toc123600783)

[**6.1.** **Kết luận** 45](#_Toc123600784)

[6.1.1. Những kết quả đạt được 45](#_Toc123600785)

[6.1.2. Hạn chế 46](#_Toc123600786)

[**6.2.** **Hướng phát triển** 46](#_Toc123600787)

[6.2.1. Hướng khắc phục các hạn chế 46](#_Toc123600788)

[6.2.2. Hướng mở rộng đề tài 46](#_Toc123600789)

[DANH MỤC TÀI LIỆU THAM KHẢO 46](#_Toc123600790)

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỒ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

# PHIẾU GIAO NHIỆM VỤ ĐỀ TÀI ĐỒ ÁN HỌC PHẦN

|  |  |
| --- | --- |
| Giảng viên giảng dạy: **VÕ XUÂN THỂ** | |
| Tên học phần: **Lập trình âm thanh (AudioPro)**  Mã số lớp HP: **MUL14126-01** | |
| Tên đề tài: **LẬP TRÌNH HỆ THỐNG QUẢN LÝ NỀN TẢNG NHẠC GIẢI TRÍ** | |
| Sinh viên thực hiện: **Lương Quốc Diễn, N20DCPT009** | |
| Thời gian thực hiện: **03/01/2023**  đến **11/03/2023** | |
| **Yêu cầu của đề tài**  **Lý thuyết:** Vận dụng kiến thức lập trình âm thanh vào việc phát triển một hệ thống ứng dụng quản lý âm thanh dùng trong một lĩnh vực chuyên môn thực tiễn, minh họa với ngôn ngữ lập trình Dart Framework Flutter dùng công cụ Android Studio và Visual Studio Code với một số thư viện hỗ trợ lập trình âm thanh, như: AudioPlayers, Google Speech… Nguồn âm thanh từ: Zingmp3  **Thực hành:** Lập trình phát triển phần mềm cung cấp nền tảng nghe nhạc và giải trí gồm:  + Thư viện chính được dùng: AudioPlayers và Google Speech  + Kiến trúc: Sử dụng kiến trúc Bloc để quản lí event và state trong ứng dụng.  + Nguồn âm thanh thực nghiệm sử dụng: Zingmp3  + Các chức năng chính của sản phẩm:   * Nghe nhạc. * Tải nhạc * Tìm kiếm * Thống kê và xếp hạng nhạc. * Karaoke | |
| **GIẢNG VIÊN** | **Ngày 03 tháng 01 năm 2023**  **SV Thực hiện** |

# GIỚI THIỆU VỀ ĐỀ TÀI

## Tổng quan về đề tài

Tìm hiểu về âm thanh học và kỹ thuật âm thanh, đồng thời dựa trên công cụ lập trình Android Studio với ngôn ngữ lập trình Dart – Framework Flutter sử dụng thư viện hỗ trợ lập trình âm thanh (Audio Program) Audio Players và Bloc (Bussiness logic component) để phát triển Desktop App cho phép quản lý và xử lý các âm thanh sử dụng trong Lập trình phát triển nền tảng nghe nhạc và giải trí sử dụng Framework Flutter (Windows)

Tập dữ liệu thực nghiệm chính đước tham khảo (thu thập) từ Zingmp3 .Chủ yếu là file định dạng mp3.

## Nội dung chuyên môn chính của đề tài

+ Thư viện: Audio Players và Bloc (Bussiness logic component)

+ Công cụ: Sử dụng IDE Android Studio

+ Sản phẩm đề tài: Tên sản phẩm: **LẬP TRÌNH PHÁT TRIỂN NỀN TẢNG NGHE NHẠC VÀ GIẢI TRÍ DÙNG FRAMEWORK FLUTTER (WINDOWS).**

ỨNG DỤNG QUẢN LÝ & XỬ LÝ ÂM THANH: LẬP TRÌNH BỞI **N20DCPT009** HỌC VIÊN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG TP.HCM

## Bố cục của báo cáo

Chương 1: Giới thiệu tổng quan về đề tài, nội dung chuyên môn và bố cục báo cáo

1.1 Tổng quan về đề tài.

1.2 Nội dung chuyên môn chính của đề tài

1.3 Bố cục báo cáo

Chương 2: Cơ sở lý thuyết của học phần: gồm những lý thuyết cơ bản về lập trình âm thanh

2.1 Cơ sở lý thuyết về âm thanh học

2.2 Công cụ lập trình, thư viện và nguồn dữ liệu âm thanh

Chương 3: Cơ sở lý thuyết về các thuật ngữ chuyên môn:

3.1 RESTful API

3.2 JSON (Javascript Object Notation)

3.2 JWT (Json Web Token)

Chương 4: Cơ chế lấy nguồn nhạc từ Zingmp3:

Chương 5: Setup sản phẩm, thiết kế và tiến hành lập trình:

5.1 Setup database MySQL:

5.2 Setup thư viện phía server (BE) sử dụng Nodejs + Express:

5.3 Setup thư viện Audio Players, Bloc và SharedPreferences:

5.4 Thiết kế:

5.5 Lưu trạng thái đăng nhập sử dụng SharedPreferences:

5.6 Phát nhạc thông qua URL sử dụng cấu trúc Bloc (Bussiness logic component):

5.7 Download file nhạc thông qua URL:

5.8 Lấy lời bài hát, hiển thị, ghi âm và cuộn màn hình theo lời bài hát:

5.9 Chấm điểm Karaoke:

Chương 6: Tổng kết đề tài:

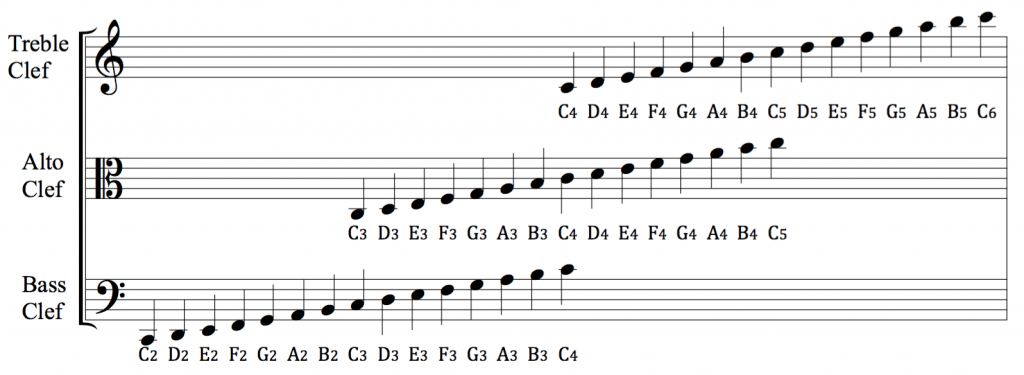
6.1 Kết quả đạt được và những hạn chế

6.2 Tổng kết

# CƠ SỞ LÝ THUYẾT VỀ LẬP TRÌNH ÂM THANH

1. **CĂN BẢN VỀ ÂM THANH HỌC:**
   1. **Một số khái niện cơ bản:**
      1. **Âm thanh là gì:**

* Âm thanh là các [dao động](https://vi.wikipedia.org/wiki/Dao_%C4%91%E1%BB%99ng) [cơ học](https://vi.wikipedia.org/wiki/C%C6%A1_h%E1%BB%8Dc) (biến đổi vị trí qua lại) của các [phân tử](https://vi.wikipedia.org/wiki/Ph%C3%A2n_t%E1%BB%AD), [nguyên tử](https://vi.wikipedia.org/wiki/Nguy%C3%AAn_t%E1%BB%AD) hay các hạt làm nên [vật chất](https://vi.wikipedia.org/wiki/V%E1%BA%ADt_ch%E1%BA%A5t) và lan truyền trong vật chất như các [sóng](https://vi.wikipedia.org/wiki/S%C3%B3ng). Âm thanh, giống như nhiều sóng, được đặc trưng bởi [tần số](https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91), [bước sóng](https://vi.wikipedia.org/wiki/B%C6%B0%E1%BB%9Bc_s%C3%B3ng), [chu kỳ](https://vi.wikipedia.org/wiki/Chu_k%E1%BB%B3), [biên độ](https://vi.wikipedia.org/wiki/Bi%C3%AAn_%C4%91%E1%BB%99) và [vận tốc](https://vi.wikipedia.org/wiki/V%E1%BA%ADn_t%E1%BB%91c) lan truyền ([tốc độ âm thanh](https://vi.wikipedia.org/wiki/T%E1%BB%91c_%C4%91%E1%BB%99_%C3%A2m_thanh)).
* Đối với [thính giác](https://vi.wikipedia.org/wiki/Th%C3%ADnh_gi%C3%A1c) của [người](https://vi.wikipedia.org/wiki/Ng%C6%B0%E1%BB%9Di), âm thanh thường là sự dao động, trong dải tần số từ khoảng 16 [Hz](https://vi.wikipedia.org/wiki/Hz) đến khoảng 20 000 [Hz](https://vi.wikipedia.org/wiki/Hertz), của các [phân tử](https://vi.wikipedia.org/wiki/Ph%C3%A2n_t%E1%BB%AD) không khí, và lan truyền trong không khí, va đập vào [màng nhĩ](https://vi.wikipedia.org/wiki/M%C3%A0ng_nh%C4%A9), làm rung màng nhĩ và kích thích [bộ não](https://vi.wikipedia.org/wiki/B%E1%BB%99_n%C3%A3o). Tuy nhiên âm thanh có thể được định nghĩa rộng hơn, tuỳ vào ứng dụng, bao gồm các tần số cao hơn hay thấp hơn tần số mà tai người có thể nghe thấy, không chỉ lan truyền trong [không khí](https://vi.wikipedia.org/wiki/Kh%C3%B4ng_kh%C3%AD) mà còn truyền trong bất cứ [vật liệu](https://vi.wikipedia.org/wiki/V%E1%BA%ADt_li%E1%BB%87u) nào. Trong định nghĩa rộng này, âm thanh là sóng cơ học và theo [lưỡng tính sóng hạt](https://vi.wikipedia.org/wiki/L%C6%B0%E1%BB%A1ng_t%C3%ADnh_s%C3%B3ng_h%E1%BA%A1t) của vật chất, sóng này cũng có thể coi là dòng lan truyền của các hạt [phonon](https://vi.wikipedia.org/wiki/Phonon), các hạt [lượng tử](https://vi.wikipedia.org/wiki/L%C6%B0%E1%BB%A3ng_t%E1%BB%AD) của âm thanh.
  + 1. **Phân loại âm thanh:**
* Âm thanh Analog: Âm thanh analog là một loại tín hiệu âm thanh có thể truyền đi xa, ở nhiều dạng hình thức khác nhau, có cả dạng con người nghe được và không nghe được. Từ analog trong tiếng anh có nghĩa là liên tục nhưng trong âm thanh có nghĩa là âm thanh lúc sau sẽ tương tự âm thanh lúc trước nhưng chỉ giống về tần số chứ cường độ, độ lớn của tín hiệu thì có sự thay đổi. Tín hiệu analog thường được biểu diễn dưới đồ thị XY (cos sin) có hình parabol liên tục nên được gọi là tín hiệu âm thanh liên tục.
* Âm thanh kỹ thuật số: Khi nói về thiết bị hay bất cứ thứ gì liên quan đến kỹ thuật số bạn có thể nhớ ngay tới những dạy nhị phân. Đó cũng chính là nền tảng của âm thanh digital hay người ta còn gọi là âm thanh kỹ thuật số. Âm thanh digital là tín hiệu âm thanh được chuyển đổi, truyền tải và lưu trữ dưới dạng mã hóa bởi các dãy nhị phân, đơn vị đo là bit, Từ các giao động sóng âm thực tế, qua các công cụ chuyển đổi ban đầu như micro, máy ghi âm,… sẽ được chuyển đổi thành các dãy nhị phân tương đường để truyền tải hoặc lưu trữ lại. Âm thanh digital được ứng dụng dựa trên sự ra đời và phát triển của hệ nhị phân và đây cũng được xem là bước ngoặt trong lĩnh vực âm thanh. Nó đã thay đổi gần như hoàn toàn cách lưu trữ âm thanh truyền thống cũng như thay đổi nhiều cách thức truyền tải thông tin về âm thanh. Nó cũng giải quyết nhiều vấn đề mà tín hiệu âm thanh analog đang gặp phải.
  + 1. **Thực tiễn của âm thanh:**
* Trong thực tiễn chúng ta chỉ nên sử dụng âm thanh Stereo khi giải trí. Cụ thể là khi nghe nhạc, xem phim,…
* Ngược lại ta nên sử dụng âm thanh Mono khi nghe thời sự, nghe video giảng bài, nghe âm thanh độc thoại để tránh bị vang.
  + 1. **Khuynh hướng kỹ thuật âm thanh:**
* Hiện nay hầu hết con người đều sử dụng âm thanh Digital vì nó hiện đại và giải quyết những vẫn đề mà âm thanh Analog không thể làm được.
  1. **Các đặc tính cơ bản của âm thanh:**
     1. **Cao độ (pitch):**
* Cao độ là một đặc tính của [tri giác](https://vi.wikipedia.org/wiki/Tri_gi%C3%A1c) cho phép người ta có thể sắp xếp những âm thanh khác nhau thành một chuỗi những âm có mối tương quan với nhau về [tần số](https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91) dao động, tức cao độ là độ "cao", "thấp" của âm thanh phụ thuộc vào tần số dao động. Tần số dao động càng lớn thì âm thanh càng "cao" và ngược lại. Cao độ cùng với [trường độ](https://vi.wikipedia.org/wiki/Tr%C6%B0%E1%BB%9Dng_%C4%91%E1%BB%99), [cường độ](https://vi.wikipedia.org/w/index.php?title=C%C6%B0%E1%BB%9Dng_%C4%91%E1%BB%99_(%C3%A2m_nh%E1%BA%A1c)&action=edit&redlink=1) và [âm sắc](https://vi.wikipedia.org/wiki/%C3%82m_s%E1%BA%AFc) là 4 thuộc tính chính của âm thanh có nhạc tính.

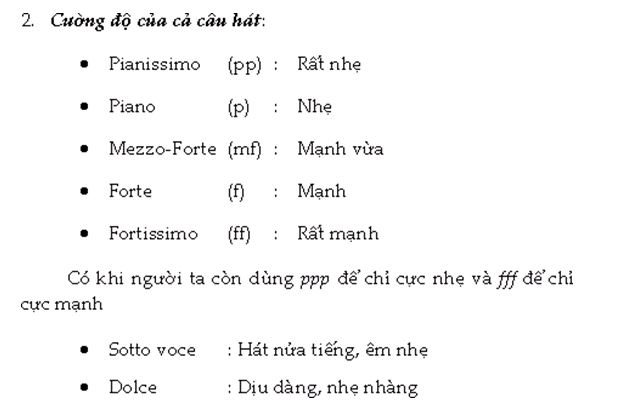


* Cao độ có thể được định lượng như [tần số](https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91), nhưng nó không phải là một tính chất thuần túy khách quan thuộc [vật lý](https://vi.wikipedia.org/wiki/V%E1%BA%ADt_l%C3%BD_h%E1%BB%8Dc), mà nó là một thuộc tính chủ quan thuộc tâm lý âm học (tâm thính học; psychoacoustic) của âm thanh. Việc nghiên cứu về cao độ và sự nhận thức về cao độ đã từng là một vấn đề trọng tâm trong ngành [tâm lý âm học](https://vi.wikipedia.org/w/index.php?title=T%C3%A2m_l%C3%BD_%C3%A2m_h%E1%BB%8Dc&action=edit&redlink=1), và nó từng được dùng làm phương tiện trong sự định hình và kiểm nghiệm những nguyên lý về đặc tả âm thanh, về quá trình xử lý và nhận thức trong hệ thống [thính giác](https://vi.wikipedia.org/wiki/Th%C3%ADnh_gi%C3%A1c).
  + 1. **Trường độ (duration):**
* Trong [âm nhạc](https://vi.wikipedia.org/wiki/%C3%82m_nh%E1%BA%A1c), trường độ là một khoảng thời gian cụ thể. Nó chính là độ dài của một [nốt nhạc](https://vi.wikipedia.org/wiki/N%E1%BB%91t_nh%E1%BA%A1c) tùy thuộc vào thời gian tồn tại của sóng âm trong không khí. Trường độ là một đặc tính của nốt và cũng là một trong những nền tảng của [nhịp điệu](https://vi.wikipedia.org/wiki/Nh%E1%BB%8Bp_%C4%91i%E1%BB%87u).

Diagram

Description automatically generated

* + 1. **Cường độ (amplitude):**
* Cường độ âm thanh là lượng [năng lượng](https://vi.wikipedia.org/wiki/N%C4%83ng_l%C6%B0%E1%BB%A3ng) được [sóng âm](https://vi.wikipedia.org/wiki/%C3%82m_thanh) truyền đi trong một đơn vị thời gian qua một đơn vị diện tích đặt vuông góc với phương truyền âm. Đơn vị cường độ âm là oát trên mét vuông (ký hiệu: W/m²).



Có nhiều cách diễn tấu khác nhau để diễn tả sự mạnh nhẹ của một âm thanh

* Marcato: tấu mạnh và đầy
* Staccato: tấu rời và nhẹ từng nốt.
* Sostenuto: tấu mạnh rồi nhỏ lại Nuơng nhẹ từng [nốt nhạc](https://bloghocpiano.com/not-nhac-la-gi/).



* + 1. **Âm sắc (timbre, tone color):**
* Trong [âm nhạc](https://vi.wikipedia.org/wiki/%C3%82m_nh%E1%BA%A1c), **âm sắc** được biết là phẩm chất của một nốt nhạc hoặc âm thanh. Âm sắc giúp phân biệt làm những loại nhạc cụ khác nhau. Những tính chất vật lý của âm thanh ảnh hưởng đến sự nhận thức âm sắc bao gồm [phổ âm](https://vi.wikipedia.org/wiki/Ph%E1%BB%95_%C3%A2m) và [cường độ](https://vi.wikipedia.org/wiki/C%C6%B0%E1%BB%9Dng_%C4%91%E1%BB%99_%C3%A2m_thanh).
* Trong thanh âm học, âm sắc được biết đến như là "chất lượng" âm thanh hay "màu sắc" của âm thanh. Mỗi giọng hát, mỗi loại đàn cho ta một âm sắc riêng biệt (sáng – tối, trong – đục). Mỗi bài hát cũng có mầu sắc khác nhau (vui – buồn). Đây là đặc tính rất quan trọng của âm nhạc (quality of sound)
* Ví dụ, âm sắc là cái mà người ta có thể dùng để phân biệt một kèn [saxophone](https://vi.wikipedia.org/wiki/Saxophone) và một kèn [trumpet](https://vi.wikipedia.org/wiki/Trumpet) trong một nhóm chơi [nhạc Jazz](https://vi.wikipedia.org/wiki/Nh%E1%BA%A1c_Jazz), ngay cả khi cả hai loại nhạc cụ này đang chơi những nốt nhạc có cùng cường độ và cao độ.

1. **CĂN BẢN VỀ KỸ THUẬT ÂM THANH (AUDIO/SOUND TECHINICAL):**
   1. **Các đặc tính cơ bản của file lưu trữ âm thanh (Digital):**

* Khi nói về thiết bị hay bất cứ thứ gì liên quan đến kỹ thuật số bạn có thể nhớ ngay tới những dạy nhị phân. Đó cũng chính là nền tảng của âm thanh digital hay người ta còn gọi là âm thanh kỹ thuật số. Âm thanh digital là tín hiệu âm thanh được chuyển đổi, truyền tải và lưu trữ dưới dạng mã hóa bởi các dãy nhị phân, đơn vị đo là bit, Từ các giao động sóng âm thực tế, qua các công cụ chuyển đổi ban đầu như micro, máy ghi âm,… sẽ được chuyển đổi thành các dãy nhị phân tương đường để truyền tải hoặc lưu trữ lại.
* Âm thanh digital được ứng dụng dựa trên sự ra đời và phát triển của hệ nhị phân và đây cũng được xem là bước ngoặt trong lĩnh vực âm thanh. Nó đã thay đổi gần như hoàn toàn cách lưu trữ âm thanh truyền thống cũng như thay đổi nhiều cách thức truyền tải thông tin về âm thanh. Nó cũng giải quyết nhiều vấn đề mà tín hiệu âm thanh analog đang gặp phái
* Âm thanh digital con người sẽ không thể nghe và hiểu được nếu không thông qua các thiết bị chuyển đổi kỹ thuật số. Chúng được tồn tại dưới cả dạng bản cứng như các đầu đĩa DVD, VCD mà cũng có thể dưới dạng nhưng bản mềm như các file trên máy tính, các bản nhạc MP3 trên điện thoại,…

Những đặc điểm riêng biệt mà âm thanh digital có đó là:

* Lưu trữ bằng dữ liệu mã hóa nên sẽ không bị biến đổi và tác động bởi môi trường hay khoảng cách
* Con người không thể hiểu hay nghe trực tiếp tín hiệu âm thanh digital
* Truyền đạt hay lưu trữ được nhiều hơn trên một đường dây
* Có thể phát triển hơn nữa ở tương lai
  1. **Các dạng đuôi (PHẦN MỞ RỘNG/Extension) của file âm thanh và đặc tính:**
* **File MP3:**

MP3 là cụm từ viết tắt của MPEG-1 audio Player 3 hay Motion Pictures Expert Group 1 Layer 3, là một định dạng âm thanh kỹ thuật số được tạo ra bằng cách **nén dữ liệu**. Trong quá trình nén, các dãy âm thừa, dãy âm quá cao hay quá thấp đều sẽ bị loại bỏ. Do đó, file MP3 có đặc điểm là rất nhẹ, dễ dàng chia sẻ, tải về nhưng chất lượng âm thanh sẽ bị giảm đi nhiều so với âm từ CD hay phòng thu.

Các file nhạc đuôi .mp3 là loại file âm thanh phổ biến nhất hiện nay, có thể dễ dàng tìm kiếm miễn phí trên mạng internet .

* **File WMA:**

Tương tự như MP3, WMA là một định dạng âm thanh được tạo ra bằng cách nén và mất đi một phần dữ liệu. WMA là viết tắt của Windows Media Audio được phát triển bởi hãng Microsoft, được xem là đối thủ cạnh tranh của MP3. File WMA có dung lượng thậm chí còn nhẹ hơn MP3 (có thể chỉ bằng một nửa) mà chất lượng thì không hề thua kém.

* **File WAV**

WAV là viết tắt của Waveform Audio File Format, một định dạng âm thanh được phát triển bởi Microsoft và IBM. Nếu như WAM và MP3 là những đuôi file âm thanh đã trải qua quá trình nén và mất dữ liệu, thì WAM là một dạng file âm thanh gốc, không nén. Và do đó, file WAV có chất lượng âm thanh tốt (tương đương nhạc CD) nhưng khá nặng do dung lượng lớn.

* **File AMR**

AMR là định dạng âm thanh được sử dụng phổ biến trong các thiết bị âm thanh như: điện thoại, máy nghe nhạc. Định dạng này tối ưu hóa để giải mã giọng nói, được áp dụng cho việc đàm thoại trên điện thoại.

Vào tháng 10/1998 AMR được sử dụng làm codec nói tiêu chuẩn của 3GPP và vẫn đang còn được sử dụng phổ biến trong mạng GSM và UMTS hiện nay.

* **File MIDI**

MIDI (Musical Instrument Digital Interface) bao gồm tiếng đàn, tiếng sáo, tiếng các bộ gõ và cả tiếng hát tồn tại dưới dạng sóng âm thanh hình SIN. MIDI dùng để trao đổi thông tin biểu diễn giữa các nhạc cụ điện tử hoặc giữa các nhạc cụ điện tử với máy tính.

Nhạc MIDI dùng kỹ thuật số (Digital) để lưu lại âm thanh và được mãhóa dưới dạng nhị phân bao gồm các con số 0 và 1.

Hơn nữa, do chỉ ghi lại bản nhạc mà file MIDI có dung lượng rất nhỏ nên thường được ứng dụng rộng rãi trong các thiết bị điện tử như điện thoại di động, Keyboard, guitar điện, kèn saxophone.

* **File ACC:**

AAC – Advanced Audio Coding. Có thể xem nó là anh em song sinh với MP3 khi cùng sử dụng kỹ thuật nén âm thanh mà không làm mất đi nhiều dữ liệu. Ưu điểm của nó so với MP3 là có thể trình diễn các dải tần thấp ở đa kênh mà MP3 hiếm khi làm được.

* **File WMA9**

WMA9 (Window Media Audio) được xem là đối thủ cạnh tranh với MP3, ACC do Microsoft phát triển. Đây là chuẩn nén âm thanh với bit rate thấp hơn một nửa nhưng chất lượng tương đương với MP3.

* **File AC3**

AC3 (Audio Coding 3) được sử dụng hầu hết trên các đĩa DVD giúp mở rộng hệ thống âm thanh vòm. AC3 được tạo ra để tăng độ trung thực so với định dạng trước đó về tiêu chuẩn âm thanh vòm. AC3 cho phép số hóa (coding) âm thanh với tần số thấp hơn (lower sample rate) nhưng vẫn bảo đảm được chất lượng âm thanh.

Hiện nay, AC3 được ứng dụng khá phổ biến trong âm thanh máy tính cũng như trong hầu hết các điện thoại di dộng như nhạc chuông, âm báo cũng được định dạng AC3.

* 1. **Các dạng nén File âm thanh thông dụng dựa và Bitrate:**

- Kbps tên gọi chung là Bit-rate, Bit-rate là số bit mà máy tính cần xử lý trong một giây, đơn cử như nhạc 128 Kbps thì yêu cầu máy tính phải xử lý 128000 bit/giây (vì 1 Kbps = 1000 bit/giây), tương tự với nhạc 320 Kbps thì máy tính phải xử lý ở mức độ cao hơn là 320000 bit/giây. Đi sâu vào Bitrate sẽ có 3 loại:

* **Constant bitrate (CBR)**: CBR chính là những con số 128kbps hoặc 320kbps mà các bạn thấy trên các web hoặc các bài nhạc online đó. Constant bitrate là loại bit-rate cố định, tức với một file nhạc 128 Kbps bất kỳ thì máy tính luôn luôn xử lý 128000 bit/giây xuyên xuốt từ đầu đến cuối bài hát… CBR có hạn chế là bạn thấy đấy, bài hát kể cả đoạn đầu không hát thì máy tính vấn phải xử lý 128000 bit/s nên CPU tự dưng bị nặng nên máy sẽ bị lag, bài hát kém chiều sau và không chân thực. Mà lại tốn dung lượng file nhạc.
* **Average bitrate (ABR)**: ABR sinh ra để giải quyết nhược điểm của CBR bên trên, tức là những đoạn nhạc trầm bổng khác nhau thì số Kbps sẽ được thay đổi cho phù hợp, với những đoạn nhạc nhẹ nhàng thì bit-rate sẽ thấp còn những đoạn nhạc mạnh thì bit-rate sẽ cao, như vậy CPU sẽ rất thoải mái để xử lý các tác vụ khác làm máy của bạn hoạt động tốt hơn. giúp bản nhạc nghe chân thực hơn, ABR ra đời nhằm tạo ra những bản nhạc vừa có tùy biến bit-rate lại vừa có dung lượng tương đương với CBR nên khả năng tùy biến bit-rate của ABR không thực sự tốt, vì có chút tùy biến bit-rate nên dung lượng của ABR nhẹ hơn CBR chút xíu. Cả 2 loại trên phân tích để bạn hiểu bản chất thôi, còn cái bạn nên hiểu và tìm hiểu sâu là cái bên dưới này: VBR
* **Variable bitrate (VBR)**:Variable nghĩa là tùy biến, đúng vậy nghĩa như nào mục đích của nó chính là như vậy. Loại mã hóa VBR là tạo ra những file nhạc có tùy biến bit rate tối ưu nhất đoạn nào cần nhiều bitrate để nâng cao chất lượng và hiệu quả thì nó sẽ tăng cao, còn không cần thì nó sẽ giảm số lượng bit cần xử lý xuống. Qua đó hạn chế tối đa lãng phí CPU và giúp bản nhạc nghe có chiều sâu, độ nảy tốt và tương đối chân thực. Nhờ tối ưu lượng bit-rate nên những file nhạc VBR có dung lượng nhỏ hơn 2 ông CBR và ABR. Bài nhạc dùng VBR bạn nên nghe với các loại [**loa karaoke**](https://amthanhthudo.com/loa-karaoke) và [**cục đẩy công suất**](https://amthanhthudo.com/cuc-day-cong-suat) cao cấp mới có thể cảm nhận được hết chất lượng và sự khác biệt của nó

-Như đã phân tích ở trên, Vì VBR là loại tùy biến bit-rate (Kbps không cố định) nên chúng ta sẽ không gọi tên bài nhạc nén theo chuẩn VRB là 128kbps hay 320 kbps mà chúng ta sẽ gọi loại này là **MP3**–**V (x)** với **(x)** thay bởi các số từ 0 tới 9, đó là thứ tự chất lượng của VBR với –**V 0** = chất lượng cao nhất và –**V 9** = chất lượng thấp nhất.

Dưới đây là thông tin các mức chất lượng của các định dạng nhạc thông dụng được mã hóa bit-rate **VBR**:

* MP3: –**V** 0 tới 9 (-V 0 chất lượng cao nhất, -V 9 chất lượng thấp nhất)
* OGG: –**q** -0.1 đến 1 (-q 1 chất lượng cao nhất, -q -0.1 chất lượng thấp nhất)
* Nero AAC: –**q** 0.05 đến 1.00 (-q 1.00 chất lượng cao nhất, -q 0.05 chất lượng thấp nhất)

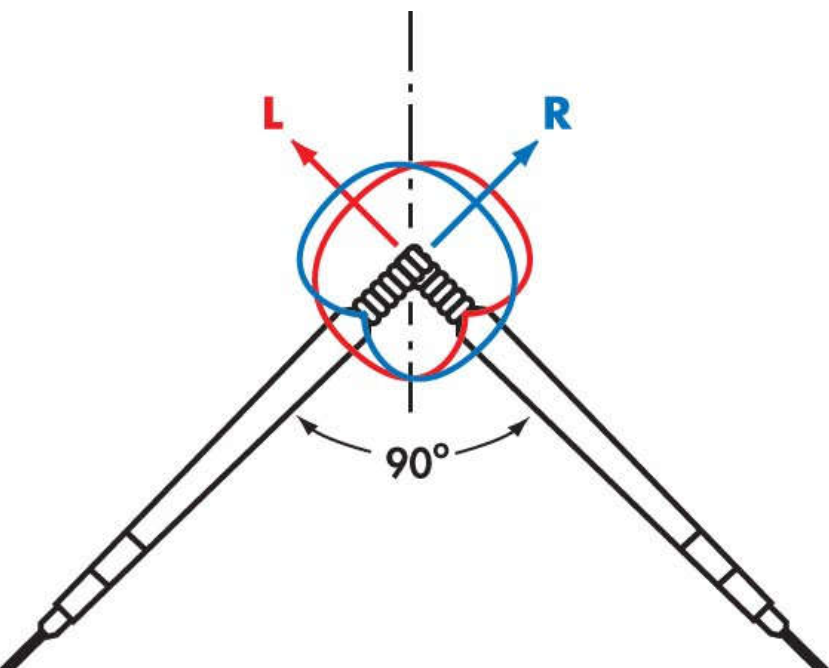
Các định dạng hỗ trợ mã hóa bit-rate **VBR**:

* MP3: **.**mp3
* AAC: **.**aac **.**m4a **.**m4b **.**mp4 **.**m4p
* ALAC: **.**m4a
* OGG: **.**ogg
  1. **Các dạng xử lý chất lượng âm thanh:**
* Các dạng xử lý:
  + Lọc tạp âm
  + Xử lý hợp âm
  + Cộng hưởng âm
  + Lọc âm
  + Tách âm
  + Hòa âm- phối khí
  + …
* Chúng ta có thể xử lý các dạng đó bằng các hình thức thông dụng:
  + 1. **Xử lý bằng công cụ vật lí:**
* Sử dụng amply để thay đổi chất lượng của các loại âm thanh khác nhau.
* Sử dụng môi trường tốt cho chất lượng âm thanh như phòng thu, đầu lọc micro…
  + 1. **Xử lý bằng phần mềm:**
* Sử dụng các công cụ xử lý âm thanh như Adobe Audio, CapCut…
* Sử dụng các công cụ xử lý video cũng có thể tác động lên âm thanh như Pr, CapCut, Adobe After Effect…
  + 1. **Xử lý bằng giải thuật/kỹ thuật mã hóa âm thanh để can thiệp trực tiếp vào chất lượng âm thanh:**
* Can thiệp vào chất lượng âm tham khi xử lý chúng ta có thể sử dụng giải thuật và kỹ thuật mã hóa và nén âm thanh nhằm mục đích làm thay đổi chất lượng âm thanh.
* Kỹ thuật này đòi hỏi rất nhiều về kỹ năng lập trình cũng như khả năng tính toán.
* Một số thuật toán mã hóa âm thanh phổ biến:
  + PCMU
  + PCMA
  + ADPCM
  + LPC
  + CELP
  + G.XINUM
  1. **Một số kỹ thuật thu thập âm thanh (Thu âm):**

## **Ta có 5 Kỹ thuật thu âm Stereo phổ biến (Kỹ thuật thu âm Stereo thường được thực hiện với từ 2 microphone trở lên):**

### Kỹ thuật X-Y:

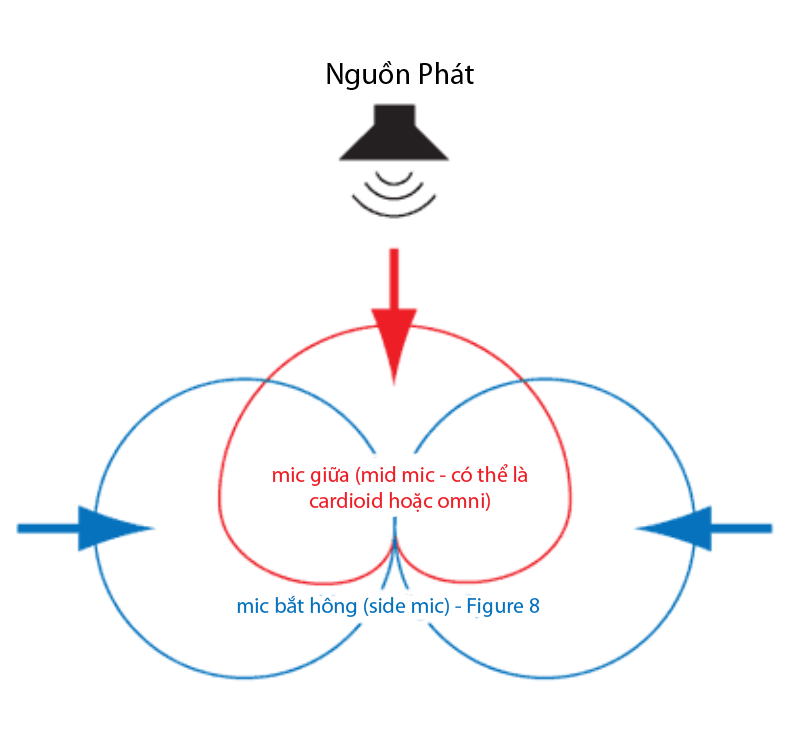
* Kỹ thuật X-Y rất dễ thực hiện và cho âm hình tương đối tốt, tương thích mono tốt. Bạn chỉ cần đảm bảo các yêu cầu sau:
* 2 chiếc microphone giống nhau (cùng model, cùng nhà sản xuất), cùng polar pattern (thường là cardioid)
* Đặt chéo nhau một góc từ 90-135 độ tùy theo độ lớn của nguồn phát (cả dàn nhạc hay drums kit, piano…) hoặc thẩm mỹ. Thiết lập phổ biến nhất là 90 độ. Góc mic này rất hay gặp trong các microphone stereo bán sẵn trên thị trường (ví dụ Audio Technica [AT2022](http://www.audio-technica.com/cms/wired_mics/a2c67abf775c91bf/)).
* 2 đầu mic đặt gần nhau nhất có thể. Khoảng cách tối đa là 12 inch (~ 30.5 cm). Thiết lập phổ biến và an toàn nhất là 2 đầu mic gần chạm nhau như hình.



* Khi thu xong, bạn có thể pan 2 tín hiệu tùy ý để có được âm thanh, độ rộng như ý. Khả năng tương thích Mono của X-Y có thể thay đổi từ khá tốt đến xuất sắc tùy vào khoảng cách giữa 2 đầu mic capsule .

### Kỹ thuật Mid-Side Stereo (M-S)

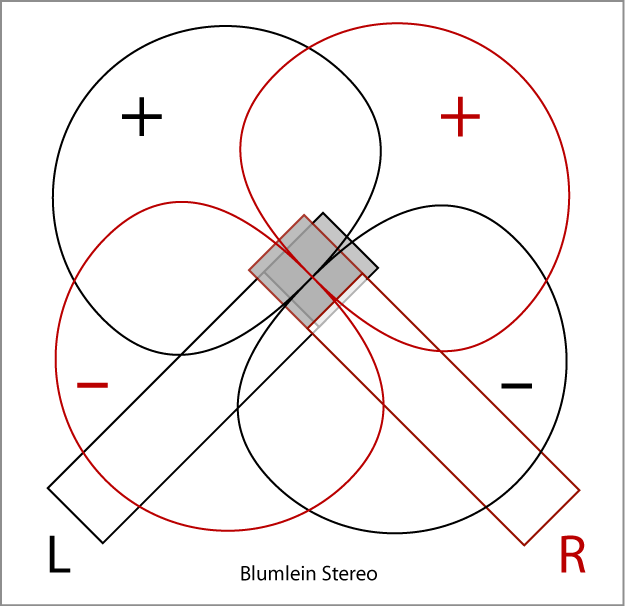
Tương tự như X-Y, đây là kỹ thuật dễ thiết lập. Theo cảm nhận cá nhân của tôi, âm thanh stereo khi sử dụng kỹ thuật M/S rất tự nhiên, không gian rộng hơn XY và tương thích mono tốt. Rất nhiều album nhạc cụ cổ điển độc tấu (đặc biệt là guitar) đã được thu bằng phương pháp này. Tôi cũng có dịp thử nghiệm một vài lần khi thu guitar cổ điển và rất ưng ý với kết quả thu được.



* Yêu cầu kỹ thuật của Mid-Side có hơi phức tạp hơn XY 1 xíu. Chúng ta sẽ cần 2 microphone với đặc tính âm thanh khác nhau.
* Mic A: 01 mic cardioid hoặc omni hướng về nguồn phát, bắt âm thanh trực tiếp. Thường thì người ta sử dụng Cardioid nhưng trong album “[El Diablo Suelto](https://www.amazon.com/El-Diablo-Suelto-Guitar-Venezuela/dp/B0000AQS42)” của John Williams, kỹ sư âm thanh sử dụng mic Omni. Tôi cũng đã thử cả 2 phương án và rất thích lựa chọn thứ 2.
* Mic B: 01 mic Figure-8 (bi-directional mic) hướng về 2 bên trái, phải của Mic A để bắt âm thanh gián tiếp. Mic B đặt ngay dưới hoặc trên Mic A.
* Tín hiệu thu được từ Mic A có thể pan vào giữa (Center). Tín hiệu thu từ Mic B được duplicate (hoặc copy) sang một kênh tín hiệu thứ 2. Hai kênh tín hiệu do Mic B tạo thành được pan trái, phải và 1 bên được đảo cực (invert polarity).
* Nếu bạn muốn giảm độ rộng của âm hình, hạ cường độ tín hiệu của 2 kênh Mic B xuống. Nếu bạn muốn tăng độ rộng? Hãy làm ngược lại. Rất đơn giản phải không.
* Âm thanh thu được khi sử dụng kỹ thuật M-S tương thích mono tuyệt đối! Lý do rất đơn giản. Vì 1 trong 2 kênh tín hiệu của Mic B là phiên bản đảo cực của kênh còn lại, do đó khi về mono, 2 kênh này tự triệt tiêu lẫn nhau. Kết quả bạn sẽ nghe thấy âm thanh chỉ phát từ Mic A mà không gặp bất cứ hệ quả của comb-filtering nào.

### Kỹ thuật Blumlein

* Gần giống như X-Y, kỹ thuật Blumlein dễ thiết lập, cho âm thanh rất tự nhiên, tương thích mono tốt và rất được yêu thích khi thu piano, drums room, ambience.

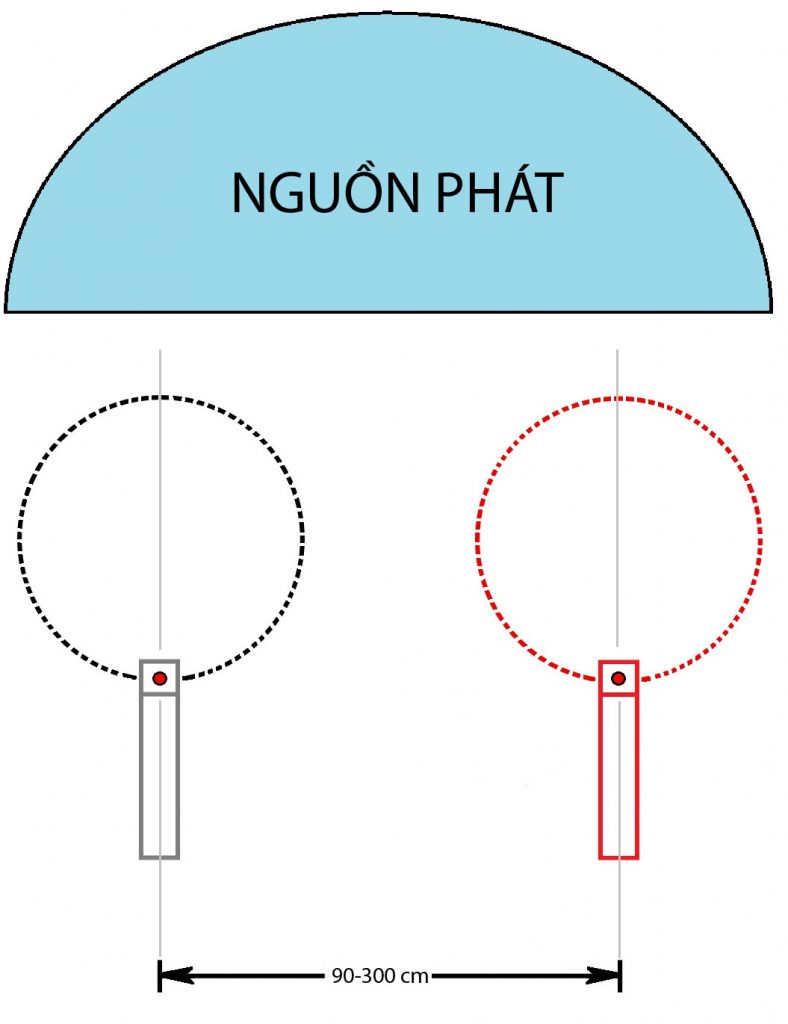


Thiết lập Blumlein:

* Sử dụng 02 microphone giống nhau (cùng model, cùng nhà sản xuất) với polar pattern Figure-8 (bi-directional mic)
* Đặt 2 mic tạo thành 1 góc 90 độ, 2 mic capsule sát nhau nhất có thể
* Tín hiệu thu được từ 2 mic có thể pan trái phải với độ rộng tùy ý để có âm thanh tốt, phù hợp nhất.

### Kỹ thuật Spaced Pair (A/B)

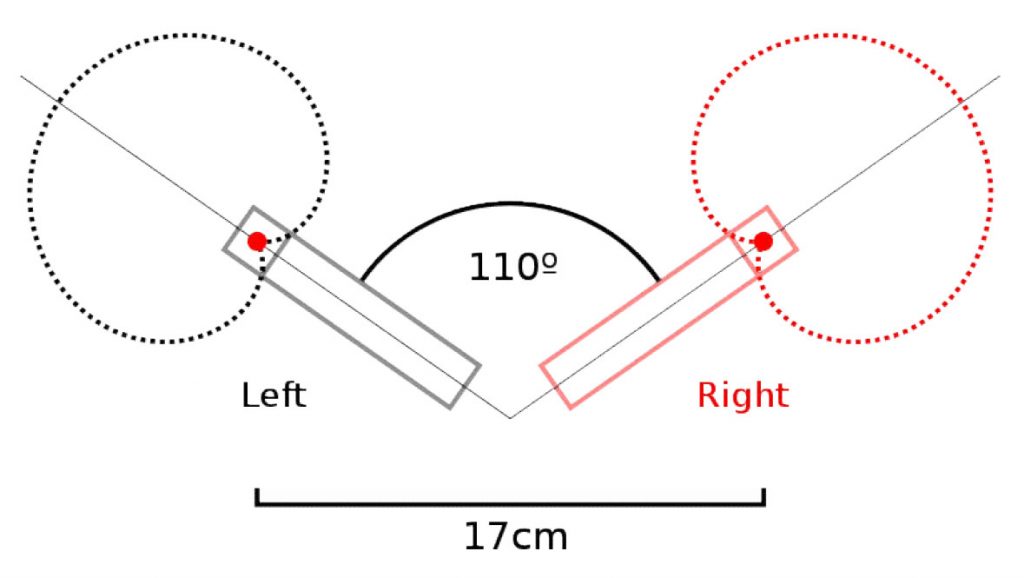
* Đây là kỹ thuật được sử dụng rất phổ biến vì sự linh hoạt, âm hình rộng tuy nhiên lại không thân thiện lắm với người mới tập thu stereo vì dễ gặp vấn đề về combfiltering và khả năng tương thích mono không cao bằng X-Y, M-S.



* Để thiết lập Spaced Pair, chúng ta cần:
* 02 microphone cùng model và polar pattern (cardioid hoặc omni)
* 02 mic này đặt cách nhau từ 90 – 300 cm. Tín hiệu thu được pan trái và phải. Độ rộng khi pan có thể quyết định tùy ý bởi kỹ sư âm thanh
* Kỹ thuật Spaced Pair thường được áp dụng khi thu dàn nhạc (giao hưởng, hợp xướng…) hoặc nhạc cụ có kích thước lớn như drums, piano… Tùy vào độ lớn thực tế của nguồn phát mà bạn có thể thay đổi khoảng cách giữa 2 mic cho phù hợp.
* Như đã nói ở trên, Spaced Pair dễ gặp vấn đề về phase hơn các kỹ thuật trước đó. Lý do chính nằm ở khoảng cách giữa 2 mic lớn hơn, do đó thời gian đi từ nguồn phát tới 2 mic có thể có độ chênh lệch lớn hơn. Ví dụ khi chúng ta thu drums overheads, drummer đánh vào HiHat, chiếc mic nào gần HiHat (mic A) sẽ thu được âm thanh trước chiếc còn lại (mic B). Tuy nhiên, khi drummer đánh vào lá Ride, chiếc mic gần đó (mic B) lại bắt tín hiệu trước mic gần HiHat (mic A). Khi gộp tín hiệu về mono, chính những sự chênh lệch về thời gian như vậy sẽ làm tương tác về phase của 2 nguồn tín hiệu trở nên phức tạp, khó có thể tạo ra kết quả ổn định, nhất quán khi drummer chơi.
* Bởi vậy, một thói quen cần có khi tập thu stereo là nghe kiểm tra trên mono liên tục để đảm bảo vị trí thiết lập của chúng ta hạn chế được nhiều nhất có thể vấn đề về phase.

**Kỹ thuật ORTF**

* Đây là kỹ thuật cho âm thanh tự nhiên, âm hình tốt. Khả năng tương thích mono tốt hơn Spaced Pair nhưng kém hơn X-Y, M-S, Blumlein. Việc thiết lập không quá khó nhưng lại đòi hỏi chính xác:
* Sử dụng 02 microphone Cardioid cùng model, cùng nhà sản xuất
* 02 mic tạo thành một góc 110 độ
* 02 mic capsule cách nhau 17cm



* Kỹ thuật này được sử dụng rất nhiều khi thu drums room và ambience vì âm thanh ấm áp, tự nhiên. Tại sao khả năng tương thích mono lại kém hơn X-Y, M-S, Blumlein? Lý do rất đơn giản: khoảng cách giữa 2 mic capsule lớn hơn (17cm).
* Có thể nói về độ rộng của âm hình, ORTF nằm giữa nhóm X-Y, M-S và Spaced Pair.
  1. **Các xử lý cơ bản trên file âm thanh:**
* Di chuyển file
* Đổi tên file
* Đổi định dạng file (đổi đuôi)
* Nén file âm thanh thành tập nén.

# CƠ SỞ LÝ THUYẾT VỀ CÁC THUẬT NGỮ CHUYÊN M

## RESTful API:

* **RESTful API** là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.
* **API** (**A**pplication **P**rogramming **I**nterface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như [JSON](https://topdev.vn/blog/json-la-gi/).
* **Chức năng chính của của REST là quy định cách sử dụng các phương thức HTTP như GET, POST, PUT, PATCH, DELETE… Tuy nhiên, trong những gì đồ án này thực hiện chỉ sử dụng 2 phương thức là GET và POST nên trong đây chúng ta sẽ chỉ đề cập đến 2 phương thức này.**
* REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng:

**GET (SELECT): Trả về một Resource hoặc một danh sách Resource**

**POST (CREATE): Tạo mới một Resource**

## JSON (Javascript Object Notation):

* **JSON** là viết tắt của **J**ava**S**cript **O**bject **N**otation, là một kiểu định dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được. [**JSON**](https://topdev.vn/viec-lam-it/json-kt35)là một tiêu chuẩn mở để trao đổi dữ liệu trên web.
* Định dạng JSON sử dụng các cặp key – value để dữ liệu sử dụng. Nó hỗ trợ các cấu trúc dữ liệu như đối tượng và mảng. Ví dụ ta có đoạn JSON với nội dung như ở dưới đây:

{

"name" : "AP03LQDien",

"title" : "Lập trình âm thanh",

"description" : "Đồ án sử dụng thư viện AudioPlayers, Bloc và SharedPreferences."

}

Ta có thể thấy cú pháp của JSON có 2 phần đó là key và value:

* Chuỗi JSON được bao lại bởi dấu ngoặc nhọn {}
* Các key, value của JSON bắt buộc phải đặt trong dấu nháy kép {“}, nếu bạn đặt nó trong dấu nháy đơn thì đây không phải là một chuỗi JSON đúng chuẩn. Nếu trường hợp trong value của bạn có chứa dấu nháy kép **"** thì hãy dùng dấu (\) để đặt trước nó, ví dụ  \"json là gì\".
* Nếu có nhiều dữ liệu thì dùng dấu phẩy , để ngăn cách.
* Các key của JSON bạn nên đặt chữ cái không dấu hoặc số, dấu “\_” và không có khoảng trắng., ký tự đầu tiên không nên đặt là số.

File json có thể được lưu với bất kỳ phần mở rộng nào, tuy nhiên thông thường thì nó được lưu dưới phần mở rộng là **.json** hoặc .**js**.

**3.3 JWT (Json Web Token):**

* **JWT** là một phương tiện đại diện cho các yêu cầu chuyển giao giữa hai bên Client – Server , các thông tin trong chuỗi **JWT** được định dạng bằng [JSON](https://topdev.vn/blog/json-la-gi/) . Trong đó chuỗi Token phải có 3 phần là header , phần payload và phần signature được ngăn bằng dấu “.”. Được sử dụng để xác thực người dùng có phải là user được phép truy cập vào resource cho phép hay không?
* Ta có chuỗi Token có dạng: **header . payload . signature**
* Phần **header** sẽ chứa kiểu dữ liệu và thuật toán được sử dụng để mã hóa ra chuỗi JWT chứa 2 cặp key value (hoặc nhiều hơn tùy mục đích):

{

“type” : ”JWT”

“algorithm”: ”SHA256”

}

“type” -> Cho biết đối tượng là chuỗi JWT

“algorithm” -> Cho biết thuật toán mã hóa ra chuỗi JWT. Ở đây ta sử dụng SHA-256

* Phần **payload** sẽ chứa các thông tin về user mà mình muốn lưu trữ. Ví dụ:

{

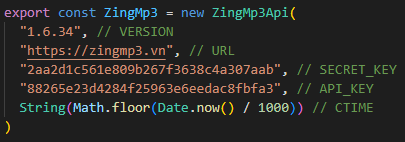
“username”:”ldevmobile”

}

* Phần **signature** được mã hóa bằng các nối 2 đoạn header và payload được mã hóa bằng thuật toán base64 sau đó cuối cùng nối với chuỗi Secret key của server đang nắm giữ và sử dụng giải thuật mã hóa SHA256 để sinh ra chuỗi **signature**.
* Khi phía Client gửi Token lên Server thì Server sẽ biết được rằng user nào đang truy cập vào tài nguyên bằng cách giải mã chuỗi payload đã được mã hóa khi Server gửi cho Client.

# CƠ CHẾ LẤY NHẠC TỪ NGUỒN ZINGMP3

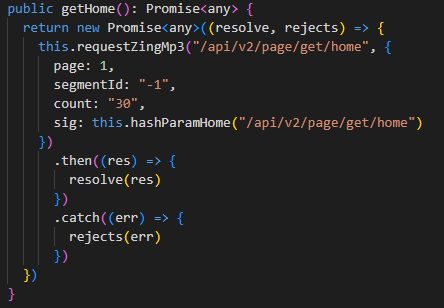
* Zingmp3 hoàn toàn sử dụng cơ chế RESTful API và JWT để xác thực và lấy resource như file nhạc, tên bài hát, nghệ sĩ…
* Zingmp3 sử dụng 5 tham số chính trong cấu trúc request resource từ Client:
  + VESION (ở đây sử dụng version 1.6.34)
  + URL (Endpoint của API)
  + SECRET KEY (Khóa bí mật mà Client cần phải gen ra và lưu trữ nó để gửi yêu cầu lấy resource từ Zingmp3 và Server của Zingmp3 sẽ xác thực).
  + API\_KEY (Khóa API được dùng để Server Zingmp3 xác thực quyền truy cập)
  + CTIME (Thời gian gửi yêu cầu đến Server)



*Hình ảnh minh hoạ*

* Zingmp3 có rất nhiều Endpoint API (URL) để truy cập resource nhạc về cho chúng ta. Vì nếu liệt kê hết tất cả API mà đồ án sử dụng thì sẽ rất nhiều và dài. Nên đồ án này chúng ta sẽ chỉ tìm hiểu một số cấu trúc request Endpoint API mà chúng ta nên biết để nắm rõ về cơ chế băm xác thực để lấy resource từ Zingmp3.

***GET\_HOME:***



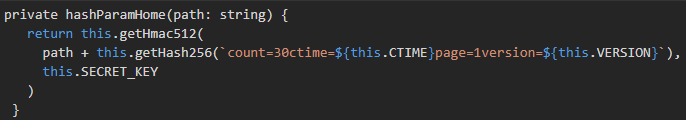
- Trong sesstion request này Zingmp3 yêu cầu Client cung cấp params như sau: “/api/v2/page/get/home” và yêu cầu các hằng query string như sau:

page: 1

segmentId: -1

count: 30

sig: sử dụng thuật toán băm Hmac512. Trong đó bao gồm chuỗi băm của params + chuỗi băm Hash256 của cụm count,ctime,page và version + SECRET\_KEY



=> Từ đó ta có URL (Endpoint API) request có dạng như sau:

[https://zingmp3.vn/api/v2/page/get/home?page=1&segmentId=-1&count=30&sig=<hàm](https://zingmp3.vn/api/v2/page/get/home?page=1&segmentId=-1&count=30&sig=%3chàm) băm như trên đã nói>

Sau đó, Zingmp3 sẽ trả kết quả định dạng JSON về cho Client như sau:

{

    "error": **false**,

    "message": "Success",

    "data": {

“item”:[

{

                "sectionType": "banner",

                "viewType": "slider",

                "title": "",

                "link": "",

                "sectionId": "hSlider",

                "items": [

                    {

                        "type": 1,

                        "link": "/bai-hat/Lam-Vay-Em-Thay-Vui-Khong-Thanh-Dat/Z67U0IE6.html",

                        "banner": "https://photo-zmp3.zmdcdn.me/banner/a/7/8/3/a7835d3a266097b15a166670db550b95.jpg",

                        "cover": "https://photo-resize-zmp3.zmdcdn.me/w94\_r1x1\_jpeg/cover/7/d/6/9/7d69f9012b3ae480141c495b12f28856.jpg",

                        "target": "1",

                        "title": "",

                        "description": "",

                        "ispr": 0,

                        "encodeId": "Z67U0IE6"

                    },

                    {

                        "type": 1,

                        "link": "/bai-hat/When-I-Get-There-P-nk/Z66DFUE7.html",

                        "banner": "https://photo-zmp3.zmdcdn.me/banner/0/7/d/b/07dbd09b741b445994f4c6f1890eee4c.jpg",

                        "cover": "https://photo-resize-zmp3.zmdcdn.me/w94\_r1x1\_jpeg/cover/e/2/a/b/e2ab4fcf94092c5daffeaad342df0653.jpg",

                        "target": "1",

                        "title": "",

                        "description": "",

                        "ispr": 0,

                        "encodeId": "Z66DFUE7"

                    },

                    {

                        "type": 4,

                        "link": "/album/Nhac-Moi-Moi-Ngay-Sam-Smith-Vu-Cat-Tuong-STAYC-Hoang-Thuy-Linh/67WIO6CF.html",

                        "banner": "https://photo-zmp3.zmdcdn.me/banner/8/7/b/c/87bc8a978acb1b03ea14a7d793c67713.jpg",

                        "cover": "https://photo-zmp3.zmdcdn.me/default.jpg",

                        "target": "1",

                        "title": "",

                        "description": "",

                        "ispr": 0,

                        "encodeId": "67WIO6CF"

                    },

                    {

                        "type": 4,

                        "link": "/playlist/Nhac-Tre-Gay-Nghien/ZU89UD8D.html",

                        "banner": "https://photo-zmp3.zmdcdn.me/banner/8/2/2/f/822f8799a239d87e1853354d4c6f13e3.jpg",

                        "cover": "https://photo-resize-zmp3.zmdcdn.me/w94\_r1x1\_jpeg/cover/f/0/5/c/f05c599832086a0a60f1105db362619f.jpg",

                        "target": "1",

                        "title": "",

                        "description": "",

                        "ispr": 0,

                        "encodeId": "ZU89UD8D"

                    },

                    {

                        "type": 4,

                        "link": "/album/Pop-Ballad-Viet-Noi-Bat-Trung-Quan-Idol-Quan-A-P-Ha-Nhi-Orange/ZE99IBE8.html",

                        "banner": "https://photo-zmp3.zmdcdn.me/banner/f/8/b/9/f8b98491d00e59a8be254cde205074da.jpg",

                        "cover": "https://photo-resize-zmp3.zmdcdn.me/w94\_r1x1\_jpeg/cover/4/1/b/1/41b17ef6a600bc71bbf5a1e3cf2da4e6.jpg",

                        "target": "1",

                        "title": "",

                        "description": "",

                        "ispr": 0,

                        "encodeId": "ZE99IBE8"

                    },

                    {

                        "type": 4,

                        "link": "/playlist/Trao-Luu-Nhac-Hot/60B8U0OB.html",

                        "banner": "https://photo-zmp3.zmdcdn.me/banner/d/d/2/4/dd243b3822a181f31e98ed89be31b30d.jpg",

                        "cover": "https://photo-zmp3.zmdcdn.me/default.jpg",

                        "target": "1",

                        "title": "",

                        "description": "",

                        "ispr": 0,

                        "encodeId": "60B8U0OB"

                    }

                ]

            },

            {

                "sectionType": "adBanner",

                "adId": "zmp3\_Top",

                "pageType": "home"

            },

            {

                "sectionId": "hRecent",

                "title": "Gần Đây",

                "sectionType": "recentPlaylist"

            },

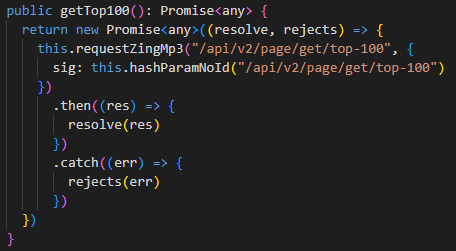
…

]

}

→ Client tiến hành đọc JSON trả về từ Zingmp3 và hiển thị dữ liệu lên UI

***GET\_TOP\_100:***



- Trong sesstion request này Zingmp3 yêu cầu Client cung cấp params như sau: “/api/v2/page/get/top-100” và yêu cầu hằng query string như sau:

Text

Description automatically generatedsig: sử dụng thuật toán băm Hmac512. Trong đó bao gồm chuỗi băm của params + chuỗi băm Hash256 của cụm ctime và version + SECRET\_KEY.

=> Từ đó ta có URL (Endpoint API) request có dạng như sau:

<https://zingmp3.vn/api/v2/page/get/top-100?sig>=<hàm băm như trên đã nói>

* Sau đó, Zingmp3 sẽ trả kết quả định dạng JSON về cho Client như sau:

{

"error": **false**,

"message": "Success",

"data": [

{

"sectionType": "playlist",

             "viewType": "slider",

            "title": "Nổi bật",

            "link": "",

            "sectionId": "",

            "items": [

                {

                    "encodeId": "ZWZB969E",

                    "title": "Top 100 Bài Hát Nhạc Trẻ Hay Nhất",

                    "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w165\_r1x1\_jpeg/cover/0/9/5/4/09542fd83e019d4734587f836bc9bbc0.jpg",

                    "isoffical": **true**,

                    "link": "/album/Top-100-Bai-Hat-Nhac-Tre-Hay-Nhat-MONO-Jack-J97-Phat-Huy-T4-Dunghoangpham/ZWZB969E.html",

                    "isIndie": **false**,

                    "releaseDate": "",

                    "sortDescription": "Top 100 Nhạc Trẻ là danh sách 100 ca khúc hot nhất hiện tại của thể loại Nhạc Trẻ, được Zing MP3 tự động tổng hợp dựa trên thông tin số liệu lượt nghe và lượt chia sẻ của từng bài hát trên phiên bản web và phiên bản Mobile. Dữ liệu sẽ được lấy trong 30 ngày gần nhất và được cập nhật liên tục.",

                    "releasedAt": 0,

                    "genreIds": [

                        "IWZ9Z08I"

                    ],

                    "PR": **false**,

                    "artists": [

                        {

                            "id": "IW7BFU8F",

                            "name": "MONO",

                            "link": "/nghe-si/MONO-Nguyen-Viet-Hoang",

                            "spotlight": **false**,

                            "alias": "MONO-Nguyen-Viet-Hoang",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/4/d/8/7/4d87ca5561df73c8a1b4575eb253d610.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/4/d/8/7/4d87ca5561df73c8a1b4575eb253d610.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6BUFZBCD",

                            "totalFollow": 21287

                        },

                        {

                            "id": "IW6WCI7U",

                            "name": "Jack - J97",

                            "link": "/Jack-J97",

                            "spotlight": **false**,

                            "alias": "Jack-J97",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/0/c/2/1/0c2163f3f85269440046f4d2bb7c4530.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/0/c/2/1/0c2163f3f85269440046f4d2bb7c4530.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "ZUDICAF0",

                            "totalFollow": 2506072

                        },

                        {

                            "id": "IW6ACZZE",

                            "name": "Phát Huy T4",

                            "link": "/Phat-Huy-T4",

                            "spotlight": **false**,

                            "alias": "Phat-Huy-T4",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/6/a/5/b/6a5ba60d63d591fe85bebd32c4410261.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/6/a/5/b/6a5ba60d63d591fe85bebd32c4410261.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "689U9D76",

                            "totalFollow": 21678

                        },

                        {

                            "id": "IW76EDEC",

                            "name": "Dunghoangpham",

                            "link": "/nghe-si/Dunghoangpham",

                            "spotlight": **false**,

                            "alias": "Dunghoangpham",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/1/a/8/8/1a8875fac8dabc964be2bcce8054e133.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/1/a/8/8/1a8875fac8dabc964be2bcce8054e133.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6ABFC9AO",

                            "totalFollow": 15956

                        }

                    ],

                    "artistsNames": "Nhiều nghệ sĩ",

                    "playItemMode": 0,

                    "subType": 24,

                    "uid": 0,

                    "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w320\_r1x1\_jpeg/cover/0/9/5/4/09542fd83e019d4734587f836bc9bbc0.jpg",

                    "isShuffle": **true**,

                    "isPrivate": **false**,

                    "userName": "Zing MP3",

                    "isAlbum": **false**,

                    "textType": "Playlist",

                    "isSingle": **false**

                },

                {

                    "encodeId": "ZWZB96AB",

                    "title": "Top 100 Pop Âu Mỹ Hay Nhất",

                    "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w165\_r1x1\_jpeg/cover/8/5/5/b/855bb71b9bc9a577ea6627df65a2adeb.jpg",

                    "isoffical": **true**,

                    "link": "/album/Top-100-Pop-Au-My-Hay-Nhat-Adele-Sam-Smith-Miley-Cyrus-Troye-Sivan/ZWZB96AB.html",

                    "isIndie": **false**,

                    "releaseDate": "",

                    "sortDescription": "Top 100 Nhạc Pop Âu Mỹ là danh sách 100 ca khúc hot nhất hiện tại của thể loại Top 100 Nhạc Pop Âu Mỹ, được Zing MP3 tự động tổng hợp dựa trên thông tin số liệu lượt nghe và lượt chia sẻ của từng bài hát trên phiên bản web và phiên bản Mobile. Dữ liệu sẽ được lấy trong 30 ngày gần nhất và được cập nhật liên tục.",

                    "releasedAt": 0,

                    "genreIds": [

                        "IWZ9Z08O",

                        "IWZ9Z097"

                    ],

                    "PR": **false**,

                    "artists": [

                        {

                            "id": "IWZ9Z09U",

                            "name": "Adele",

                            "link": "/nghe-si/Adele",

                            "spotlight": **false**,

                            "alias": "Adele",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/c/6/d/3/c6d323a06f7154a8e05056cb435889ea.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/c/6/d/3/c6d323a06f7154a8e05056cb435889ea.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6UAEWAAB",

                            "totalFollow": 25548

                        },

                        {

                            "id": "IWZ9D67E",

                            "name": "Sam Smith",

                            "link": "/nghe-si/Sam-Smith",

                            "spotlight": **false**,

                            "alias": "Sam-Smith",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/a/8/6/6/a86616539b79e4329f895688ac96042e.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/a/8/6/6/a86616539b79e4329f895688ac96042e.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "ZOE0OE0W",

                            "totalFollow": 8334

                        },

                        {

                            "id": "IWZ9ZO96",

                            "name": "Miley Cyrus",

                            "link": "/Miley-Cyrus",

                            "spotlight": **false**,

                            "alias": "Miley-Cyrus",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/d/0/d/2/d0d23ed645be5bbdcaae20b9b554a397.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/d/0/d/2/d0d23ed645be5bbdcaae20b9b554a397.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "ZWZAE9OD",

                            "totalFollow": 20014

                        },

                        {

                            "id": "IWZAWD00",

                            "name": "Troye Sivan",

                            "link": "/nghe-si/Troye-Sivan",

                            "spotlight": **false**,

                            "alias": "Troye-Sivan",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/2/5/f/8/25f86f480eaa6c4c8c8b8f0f2f9b1016.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/2/5/f/8/25f86f480eaa6c4c8c8b8f0f2f9b1016.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "ZOEWEZO0",

                            "totalFollow": 5381

                        }

                    ],

                    "artistsNames": "Nhiều nghệ sĩ",

                    "playItemMode": 0,

                    "subType": 24,

                    "uid": 88,

                    "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w320\_r1x1\_jpeg/cover/8/5/5/b/855bb71b9bc9a577ea6627df65a2adeb.jpg",

                    "isShuffle": **true**,

                    "isPrivate": **false**,

                    "userName": "Zing MP3",

                    "isAlbum": **false**,

                    "textType": "Playlist",

                    "isSingle": **false**

                },

                {

                    "encodeId": "ZWZB96DC",

                    "title": "Top 100 Nhạc Hàn Quốc Hay Nhất",

                    "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w165\_r1x1\_jpeg/cover/8/a/b/c/8abcfc5b85e7c581bf49057d01bfd1d4.jpg",

                    "isoffical": **true**,

                    "link": "/album/Top-100-Nhac-Han-Quoc-Hay-Nhat-TAEYANG-BLACKPINK-IVE-TWICE/ZWZB96DC.html",

                    "isIndie": **false**,

                    "releaseDate": "",

                    "sortDescription": "Top 100 Nhạc Hàn Quốc Hay Nhất là danh sách 100 ca khúc hot nhất hiện tại của thể loại Nhạc Hàn Quốc, được Zing MP3 tự động tổng hợp dựa trên thông tin số liệu lượt nghe và lượt chia sẻ của từng bài hát trên phiên bản web và phiên bản Mobile. Dữ liệu sẽ được lấy trong 30 ngày gần nhất và được cập nhật liên tục.",

                    "releasedAt": 0,

                    "genreIds": [

                        "IWZ9Z08W"

                    ],

                    "PR": **false**,

                    "artists": [

                        {

                            "id": "IWZ9ZUFW",

                            "name": "TAEYANG",

                            "link": "/nghe-si/Tae-Yang",

                            "spotlight": **false**,

                            "alias": "Tae-Yang",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/4/8/2/b/482b89a96b270bdb5012a8a1c0cb3810.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/4/8/2/b/482b89a96b270bdb5012a8a1c0cb3810.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "ZOUUIOEC",

                            "totalFollow": 476010

                        },

                        {

                            "id": "IWZA7DE0",

                            "name": "BLACKPINK",

                            "link": "/nghe-si/BLACKPINK",

                            "spotlight": **false**,

                            "alias": "BLACKPINK",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/7/1/0/3/71031e4984f431fe18cf18cc19af2e9d.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/7/1/0/3/71031e4984f431fe18cf18cc19af2e9d.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "ZOZ9UA7E",

                            "totalFollow": 805659

                        },

                        {

                            "id": "IW76AAWE",

                            "name": "IVE",

                            "link": "/IVE",

                            "spotlight": **false**,

                            "alias": "IVE",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/4/f/1/c/4f1c6667ebbc4fa1a91ab152902817e0.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/4/f/1/c/4f1c6667ebbc4fa1a91ab152902817e0.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "69CI9ZO9",

                            "totalFollow": 12523

                        },

                        {

                            "id": "IWZA6W70",

                            "name": "TWICE",

                            "link": "/TWICE",

                            "spotlight": **false**,

                            "alias": "TWICE",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/8/e/4/d/8e4de186676dcbd389797eed24564ffb.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/8/e/4/d/8e4de186676dcbd389797eed24564ffb.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "ZOZ9OUWE",

                            "totalFollow": 138941

                        }

                    ],

                    "artistsNames": "Nhiều nghệ sĩ",

                    "playItemMode": 0,

                    "subType": 24,

                    "uid": 0,

                    "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w320\_r1x1\_jpeg/cover/8/a/b/c/8abcfc5b85e7c581bf49057d01bfd1d4.jpg",

                    "isShuffle": **true**,

                    "isPrivate": **false**,

                    "userName": "Zing MP3",

                    "isAlbum": **false**,

                    "textType": "Playlist",

                    "isSingle": **false**

                },

                {

                    "encodeId": "ZWZB96AI",

                    "title": "Top 100 Nhạc Rap Việt Nam Hay Nhất",

                    "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w165\_r1x1\_jpeg/cover/2/c/c/6/2cc67e5b1089aaf70f03df8ce43432a1.jpg",

                    "isoffical": **true**,

                    "link": "/album/Top-100-Nhac-Rap-Viet-Nam-Hay-Nhat-HIEUTHUHAI-Phao-B-Ray-G5R-Squad/ZWZB96AI.html",

                    "isIndie": **false**,

                    "releaseDate": "",

                    "sortDescription": "Top 100 Nhạc Rap Việt Nam Hay Nhất là danh sách 100 ca khúc hot nhất hiện tại của thể loại nhạc Rap Việt Nam, được Zing MP3 tự động tổng hợp dựa trên thông tin số liệu lượt nghe và lượt chia sẻ của từng bài hát trên phiên bản web và phiên bản Mobile. Dữ liệu sẽ được lấy trong 30 ngày gần nhất và được cập nhật liên tục.",

                    "releasedAt": 0,

                    "genreIds": [

                        "IWZ9Z08I",

                        "IWZ9Z089"

                    ],

                    "PR": **false**,

                    "artists": [

                        {

                            "id": "IW6ZZ690",

                            "name": "HIEUTHUHAI",

                            "link": "/nghe-si/HIEUTHUHAI",

                            "spotlight": **false**,

                            "alias": "HIEUTHUHAI",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/b/f/9/6/bf969389e9de70560cede36559c8ca4a.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/b/f/9/6/bf969389e9de70560cede36559c8ca4a.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6UW9BFUW",

                            "totalFollow": 56440

                        },

                        {

                            "id": "IW6Z87FU",

                            "name": "Pháo",

                            "link": "/Phao",

                            "spotlight": **false**,

                            "alias": "Phao",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/a/0/7/7/a0775d6cf44120da431ecfb624d47cf6.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/a/0/7/7/a0775d6cf44120da431ecfb624d47cf6.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "6UW9CWZO",

                            "totalFollow": 58959

                        },

                        {

                            "id": "IWZAOAZZ",

                            "name": "B Ray",

                            "link": "/B-Ray-Rapper",

                            "spotlight": **false**,

                            "alias": "B-Ray-Rapper",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/4/1/c/5/41c52d1610adb73f5738b5fcf07e3130.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/4/1/c/5/41c52d1610adb73f5738b5fcf07e3130.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "ZODEAZBA",

                            "totalFollow": 269274

                        },

                        {

                            "id": "IW6ZU880",

                            "name": "G5R Squad",

                            "link": "/G5R-Squad",

                            "spotlight": **false**,

                            "alias": "G5R-Squad",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/1/3/2/c/132c6b8869a6d3696abe5bf69d80b585.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/1/3/2/c/132c6b8869a6d3696abe5bf69d80b585.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "6UW9BEUW",

                            "totalFollow": 180576

                        }

                    ],

                    "artistsNames": "Nhiều nghệ sĩ",

                    "playItemMode": 0,

                    "subType": 24,

                    "uid": 0,

                    "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w320\_r1x1\_jpeg/cover/2/c/c/6/2cc67e5b1089aaf70f03df8ce43432a1.jpg",

                    "isShuffle": **true**,

                    "isPrivate": **false**,

                    "userName": "Zing MP3",

                    "isAlbum": **false**,

                    "textType": "Playlist",

                    "isSingle": **false**

                },

                {

                    "encodeId": "ZWZB96C7",

                    "title": "Top 100 Nhạc Electronic/Dance Âu Mỹ Hay Nhất",

                    "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w165\_r1x1\_jpeg/cover/c/5/f/c/c5fc615c43215c6b72676f42767855ee.jpg",

                    "isoffical": **true**,

                    "link": "/album/Top-100-Nhac-Electronic-Dance-Au-My-Hay-Nhat-Alan-Walker-K-391-Emelie-Hollow-Sabrina-Carpenter/ZWZB96C7.html",

                    "isIndie": **false**,

                    "releaseDate": "",

                    "sortDescription": "Top 100 Nhạc Electronic/Dance Âu Mỹ là danh sách 100 ca khúc hot nhất hiện tại của thể loại Top 100 Nhạc Electronic/Dance Âu Mỹ, được Zing MP3 tự động tổng hợp dựa trên thông tin số liệu lượt nghe và lượt chia sẻ của từng bài hát trên phiên bản web và phiên bản Mobile. Dữ liệu sẽ được lấy trong 30 ngày gần nhất và được cập nhật liên tục.",

                    "releasedAt": 0,

                    "genreIds": [

                        "IWZ9Z08O",

                        "IWZ9Z09A"

                    ],

                    "PR": **false**,

                    "artists": [

                        {

                            "id": "IWZA6CCW",

                            "name": "Alan Walker",

                            "link": "/Alan-Walker",

                            "spotlight": **false**,

                            "alias": "Alan-Walker",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/b/a/6/b/ba6b56ca02dd06f37693d945acfcdad9.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/b/a/6/b/ba6b56ca02dd06f37693d945acfcdad9.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "ZWZCU89E",

                            "totalFollow": 1174587

                        },

                        {

                            "id": "IWZA7OIA",

                            "name": "K-391",

                            "link": "/K-391",

                            "spotlight": **false**,

                            "alias": "K-391",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/4/9/493e761d37d31c5cdef4281c0c0ef6d4\_1458805399.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/4/9/493e761d37d31c5cdef4281c0c0ef6d4\_1458805399.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "6UB6AABC",

                            "totalFollow": 187481

                        },

                        {

                            "id": "IWZD7OZ0",

                            "name": "Emelie Hollow",

                            "link": "/nghe-si/Emelie-Hollow",

                            "spotlight": **false**,

                            "alias": "Emelie-Hollow",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/e/9/e/9/e9e99eb4d4f51f77c96fd6f8a9ea4607.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/e/9/e/9/e9e99eb4d4f51f77c96fd6f8a9ea4607.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6UB6BIFO",

                            "totalFollow": 39608

                        },

                        {

                            "id": "IWZ9CUB9",

                            "name": "Sabrina Carpenter",

                            "link": "/nghe-si/Sabrina-Carpenter",

                            "spotlight": **false**,

                            "alias": "Sabrina-Carpenter",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/7/c/c/1/7cc1f3378ca955619f9389822a20e69f.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/7/c/c/1/7cc1f3378ca955619f9389822a20e69f.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6UB6966U",

                            "totalFollow": 57865

                        }

                    ],

                    "artistsNames": "Nhiều nghệ sĩ",

                    "playItemMode": 0,

                    "subType": 24,

                    "uid": 0,

                    "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w320\_r1x1\_jpeg/cover/c/5/f/c/c5fc615c43215c6b72676f42767855ee.jpg",

                    "isShuffle": **true**,

                    "isPrivate": **false**,

                    "userName": "Zing MP3",

                    "isAlbum": **false**,

                    "textType": "Playlist",

                    "isSingle": **false**

                },

                {

                    "encodeId": "ZWZB969F",

                    "title": "Top 100 Nhạc Trữ Tình Hay Nhất",

                    "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w165\_r1x1\_jpeg/cover/8/9/1/d/891d56e72dde29423e7c2c934727f5ab.jpg",

                    "isoffical": **true**,

                    "link": "/album/Top-100-Nhac-Tru-Tinh-Hay-Nhat-Nhu-Quynh-Manh-Quynh-To-My-Le-Sang/ZWZB969F.html",

                    "isIndie": **false**,

                    "releaseDate": "",

                    "sortDescription": "Top 100 Nhạc Trữ Tình là danh sách 100 ca khúc hot nhất hiện tại của thể loại Nhạc Trữ Tình, được Zing MP3 tự động tổng hợp dựa trên thông tin số liệu lượt nghe và lượt chia sẻ của từng bài hát trên phiên bản web và phiên bản Mobile. Dữ liệu sẽ được lấy trong 30 ngày gần nhất và được cập nhật liên tục.",

                    "releasedAt": 0,

                    "genreIds": [

                        "IWZ9Z08I",

                        "IWZ9Z08B"

                    ],

                    "PR": **false**,

                    "artists": [

                        {

                            "id": "IWZ9ZWA7",

                            "name": "Như Quỳnh",

                            "link": "/Nhu-Quynh",

                            "spotlight": **false**,

                            "alias": "Nhu-Quynh",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/d/c/4/6/dc46b91ae6e40a586283b03f31312baa.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/d/c/4/6/dc46b91ae6e40a586283b03f31312baa.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "ZWZBOZOA",

                            "totalFollow": 305161

                        },

                        {

                            "id": "IWZ9ZWUO",

                            "name": "Mạnh Quỳnh",

                            "link": "/Manh-Quynh",

                            "spotlight": **false**,

                            "alias": "Manh-Quynh",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/d/d/4/8/dd480f10893eb35ade277cbe1bd2bd8a.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/d/d/4/8/dd480f10893eb35ade277cbe1bd2bd8a.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "ZO698D76",

                            "totalFollow": 318407

                        },

                        {

                            "id": "IWZA6D7W",

                            "name": "Tố My",

                            "link": "/nghe-si/To-My",

                            "spotlight": **false**,

                            "alias": "To-My",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/f/3/7/2/f372ff1bee0712828b7ce9e8727614c1.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/f/3/7/2/f372ff1bee0712828b7ce9e8727614c1.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "ZWZCUF70",

                            "totalFollow": 100351

                        },

                        {

                            "id": "IWZ9896F",

                            "name": "Lê Sang",

                            "link": "/nghe-si/Le-Sang",

                            "spotlight": **false**,

                            "alias": "Le-Sang",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/7/3/735b69259492a8ccf8a52f0b26f932e9\_1511494894.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/7/3/735b69259492a8ccf8a52f0b26f932e9\_1511494894.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "ZWZ98O77",

                            "totalFollow": 153661

                        }

                    ],

                    "artistsNames": "Nhiều nghệ sĩ",

                    "playItemMode": 0,

                    "subType": 24,

                    "uid": 0,

                    "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w320\_r1x1\_jpeg/cover/8/9/1/d/891d56e72dde29423e7c2c934727f5ab.jpg",

                    "isShuffle": **true**,

                    "isPrivate": **false**,

                    "userName": "Zing MP3",

                    "isAlbum": **false**,

                    "textType": "Playlist",

                    "isSingle": **false**

                },

                {

                    "encodeId": "ZWZB96EZ",

                    "title": "Top 100 Nhạc Hòa Tấu Nhạc Cụ Guitar Hay Nhất",

                    "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w165\_r1x1\_jpeg/cover/4/9/6/5/4965f11b796243810f2d4a808b0b0e28.jpg",

                    "isoffical": **true**,

                    "link": "/album/Top-100-Nhac-Hoa-Tau-Nhac-Cu-Guitar-Hay-Nhat-Dinh-Quang-Minh-Dong-Hoa-Francis-Goya-Chen-Xiao-Ping/ZWZB96EZ.html",

                    "isIndie": **false**,

                    "releaseDate": "",

                    "sortDescription": "Top 100 Nhạc Hòa Tấu Nhạc Cụ Guitar Hay Nhất là danh sách 100 ca khúc hot nhất hiện tại của thể loại Top 100 Nhạc Hòa Tấu Nhạc Cụ Guitar Hay Nhất, được Zing MP3 tự động tổng hợp dựa trên thông tin số liệu lượt nghe và lượt chia sẻ của từng bài hát trên phiên bản web và phiên bản Mobile. Dữ liệu sẽ được lấy trong 30 ngày gần nhất và được cập nhật liên tục.",

                    "releasedAt": 0,

                    "genreIds": [

                        "IWZ9Z086",

                        "IWZ9Z0A9"

                    ],

                    "PR": **false**,

                    "artists": [

                        {

                            "id": "IWZAZ0CO",

                            "name": "Đinh Quang Minh",

                            "link": "/nghe-si/Dinh-Quang-Minh",

                            "spotlight": **false**,

                            "alias": "Dinh-Quang-Minh",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/b/7/b7f8b43a23a599c3f5b60a3c082d9aee\_1508392798.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/b/7/b7f8b43a23a599c3f5b60a3c082d9aee\_1508392798.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6UW9AEOU",

                            "totalFollow": 179

                        },

                        {

                            "id": "IWZ9CUDU",

                            "name": "Đông Hòa",

                            "link": "/nghe-si/Dong-Hoa",

                            "spotlight": **false**,

                            "alias": "Dong-Hoa",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/2/8/3/d/283dfdf9638d0c5d4801c1074eccb5fc.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/2/8/3/d/283dfdf9638d0c5d4801c1074eccb5fc.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6UW9A6OC",

                            "totalFollow": 2569

                        },

                        {

                            "id": "IWZ998FO",

                            "name": "Francis Goya",

                            "link": "/nghe-si/Francis-Goya",

                            "spotlight": **false**,

                            "alias": "Francis-Goya",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/f/5/4/a/f54acc430f63d1556318cbb760fc1087.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/f/5/4/a/f54acc430f63d1556318cbb760fc1087.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6UB66U99",

                            "totalFollow": 8111

                        },

                        {

                            "id": "IWZ98A0Z",

                            "name": "Chen Xiao Ping",

                            "link": "/nghe-si/Chen-Xiao-Ping",

                            "spotlight": **false**,

                            "alias": "Chen-Xiao-Ping",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/1/a/2/7/1a27769dccc5d485f84a9b68bd666be0.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/1/a/2/7/1a27769dccc5d485f84a9b68bd666be0.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6B7W7AZA",

                            "totalFollow": 258

                        }

                    ],

                    "artistsNames": "Nhiều nghệ sĩ",

                    "playItemMode": 0,

                    "subType": 24,

                    "uid": 0,

                    "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w320\_r1x1\_jpeg/cover/4/9/6/5/4965f11b796243810f2d4a808b0b0e28.jpg",

                    "isShuffle": **true**,

                    "isPrivate": **false**,

                    "userName": "Zing MP3",

                    "isAlbum": **false**,

                    "textType": "Playlist",

                    "isSingle": **false**

                },

                {

                    "encodeId": "ZWZB96EW",

                    "title": "Top 100 Nhạc Hòa Tấu Cổ Điển Hay Nhất",

                    "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w165\_r1x1\_jpeg/cover/1/5/1/b/151b864e73216ca847eba4caa8dbb527.jpg",

                    "isoffical": **true**,

                    "link": "/album/Top-100-Nhac-Hoa-Tau-Co-Dien-Hay-Nhat-Contempo-Divo-Hawa-Paolo-Olmi/ZWZB96EW.html",

                    "isIndie": **false**,

                    "releaseDate": "",

                    "sortDescription": "Top 100 Nhạc Hòa Tấu Cổ Điển Hay Nhất là danh sách 100 ca khúc hot nhất hiện tại của thể loại Top 100 Nhạc Hòa Tấu Cổ Điển, được Zing MP3 tự động tổng hợp dựa trên thông tin số liệu lượt nghe và lượt chia sẻ của từng bài hát trên phiên bản web và phiên bản Mobile. Dữ liệu sẽ được lấy trong 30 ngày gần nhất và được cập nhật liên tục.",

                    "releasedAt": 0,

                    "genreIds": [

                        "IWZ9Z086",

                        "IWZ9Z0BI"

                    ],

                    "PR": **false**,

                    "artists": [

                        {

                            "id": "IW6ZCU77",

                            "name": "Contempo-Divo",

                            "link": "/nghe-si/Contempo-Divo",

                            "spotlight": **false**,

                            "alias": "Contempo-Divo",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/1/a/2/7/1a27769dccc5d485f84a9b68bd666be0.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/1/a/2/7/1a27769dccc5d485f84a9b68bd666be0.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6B7WAW0O",

                            "totalFollow": 15

                        },

                        {

                            "id": "IW6ZDOC0",

                            "name": "한규원",

                            "link": "/nghe-si/IW6ZDOC0",

                            "spotlight": **false**,

                            "alias": "IW6ZDOC0",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/1/a/2/7/1a27769dccc5d485f84a9b68bd666be0.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/1/a/2/7/1a27769dccc5d485f84a9b68bd666be0.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "totalFollow": 7

                        },

                        {

                            "id": "IWZECBCB",

                            "name": "Hawa",

                            "link": "/nghe-si/Hawa",

                            "spotlight": **false**,

                            "alias": "Hawa",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/e/f/e/7/efe72b3b49fd3823f808f509d8d3f431.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/e/f/e/7/efe72b3b49fd3823f808f509d8d3f431.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "totalFollow": 4

                        },

                        {

                            "id": "IW6ZZE00",

                            "name": "Paolo Olmi",

                            "link": "/nghe-si/Paolo-Olmi",

                            "spotlight": **false**,

                            "alias": "Paolo-Olmi",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/1/a/2/7/1a27769dccc5d485f84a9b68bd666be0.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/1/a/2/7/1a27769dccc5d485f84a9b68bd666be0.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6B7WA0FO",

                            "totalFollow": 6

                        }

                    ],

                    "artistsNames": "Nhiều nghệ sĩ",

                    "playItemMode": 0,

                    "subType": 24,

                    "uid": 0,

                    "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w320\_r1x1\_jpeg/cover/1/5/1/b/151b864e73216ca847eba4caa8dbb527.jpg",

                    "isShuffle": **true**,

                    "isPrivate": **false**,

                    "userName": "Zing MP3",

                    "isAlbum": **false**,

                    "textType": "Playlist",

                    "isSingle": **false**

                },

                {

                    "encodeId": "ZWZB96EU",

                    "title": "Top 100 Nhạc Hòa Tấu Nhạc Cụ Piano Hay Nhất",

                    "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w165\_r1x1\_jpeg/cover/8/5/f/d/85fd9f4dbd3a1df69966f5260da67966.jpg",

                    "isoffical": **true**,

                    "link": "/album/Top-100-Nhac-Hoa-Tau-Nhac-Cu-Piano-Hay-Nhat-Vu-Dang-Quoc-Viet-Relax-Lab-Yiruma-Kevin-Kern/ZWZB96EU.html",

                    "isIndie": **false**,

                    "releaseDate": "",

                    "sortDescription": "Top 100 Nhạc Hòa Tấu Nhạc Cụ Piano Hay Nhất là danh sách 100 ca khúc hot nhất hiện tại của thể loại Top 100 Nhạc Hòa Tấu Nhạc Cụ Piano Hay Nhất, được Zing MP3 tự động tổng hợp dựa trên thông tin số liệu lượt nghe và lượt chia sẻ của từng bài hát trên phiên bản web và phiên bản Mobile. Dữ liệu sẽ được lấy trong 30 ngày gần nhất và được cập nhật liên tục.",

                    "releasedAt": 0,

                    "genreIds": [

                        "IWZ9Z086",

                        "IWZ9Z0B0"

                    ],

                    "PR": **false**,

                    "artists": [

                        {

                            "id": "IWZ9B9IC",

                            "name": "Vũ Đặng Quốc Việt",

                            "link": "/nghe-si/Vu-Dang-Quoc-Viet",

                            "spotlight": **false**,

                            "alias": "Vu-Dang-Quoc-Viet",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/2/5/0/b/250b6d1b98536a5d96ef30c3cfa1bb85.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/2/5/0/b/250b6d1b98536a5d96ef30c3cfa1bb85.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6UW9AUD8",

                            "totalFollow": 20947

                        },

                        {

                            "id": "IW687F6E",

                            "name": "Relax Lab",

                            "link": "/nghe-si/Relax-Lab",

                            "spotlight": **false**,

                            "alias": "Relax-Lab",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/a/e/a/d/aead283aa293caf09d4cfbd9bd0c241f.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/a/e/a/d/aead283aa293caf09d4cfbd9bd0c241f.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6ZUIEUF8",

                            "totalFollow": 210

                        },

                        {

                            "id": "IWZ9Z7F8",

                            "name": "Yiruma",

                            "link": "/nghe-si/Yiruma",

                            "spotlight": **false**,

                            "alias": "Yiruma",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/a/2/4/a/a24a27e454c743d251c39987b7e85875.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/a/2/4/a/a24a27e454c743d251c39987b7e85875.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "ZWZAFO9C",

                            "totalFollow": 146606

                        },

                        {

                            "id": "IWZ96I8W",

                            "name": "Kevin Kern",

                            "link": "/nghe-si/Kevin-Kern",

                            "spotlight": **false**,

                            "alias": "Kevin-Kern",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/9/8/4/5/9845fc205125795decfdb2b74126ff11.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/9/8/4/5/9845fc205125795decfdb2b74126ff11.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6UAEUBD8",

                            "totalFollow": 6720

                        }

                    ],

                    "artistsNames": "Nhiều nghệ sĩ",

                    "playItemMode": 0,

                    "subType": 24,

                    "uid": 0,

                    "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w320\_r1x1\_jpeg/cover/8/5/f/d/85fd9f4dbd3a1df69966f5260da67966.jpg",

                    "isShuffle": **true**,

                    "isPrivate": **false**,

                    "userName": "Zing MP3",

                    "isAlbum": **false**,

                    "textType": "Playlist",

                    "isSingle": **false**

                },

                {

                    "encodeId": "ZWZB96C8",

                    "title": "Top 100 Nhạc R&B Âu Mỹ Hay Nhất",

                    "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w165\_r1x1\_jpeg/cover/c/1/a/1/c1a14402669ed2a7613bb8811e506b4f.jpg",

                    "isoffical": **true**,

                    "link": "/album/Top-100-Nhac-R-B-Au-My-Hay-Nhat-SZA-The-Weeknd-Doja-Cat-Chris-Brown/ZWZB96C8.html",

                    "isIndie": **false**,

                    "releaseDate": "",

                    "sortDescription": "Top 100 Nhạc R&B Âu Mỹ là danh sách 100 ca khúc hot nhất hiện tại của thể loại Top 100 Nhạc R&B Âu Mỹ, được Zing MP3 tự động tổng hợp dựa trên thông tin số liệu lượt nghe và lượt chia sẻ của từng bài hát trên phiên bản web và phiên bản Mobile. Dữ liệu sẽ được lấy trong 30 ngày gần nhất và được cập nhật liên tục.",

                    "releasedAt": 0,

                    "genreIds": [

                        "IWZ9Z08O",

                        "IWZ9Z09D"

                    ],

                    "PR": **false**,

                    "artists": [

                        {

                            "id": "IWZAZ00Z",

                            "name": "SZA",

                            "link": "/nghe-si/SZA",

                            "spotlight": **false**,

                            "alias": "SZA",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/2/7/9/a/279a391af4107610977011d29d5f7cec.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/2/7/9/a/279a391af4107610977011d29d5f7cec.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "ZOEW6DEO",

                            "totalFollow": 1447

                        },

                        {

                            "id": "IWZ98796",

                            "name": "The Weeknd",

                            "link": "/nghe-si/The-Weeknd",

                            "spotlight": **false**,

                            "alias": "The-Weeknd",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/9/4/6/d/946d6011cd47896e15c2cb2a75b60e07.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/9/4/6/d/946d6011cd47896e15c2cb2a75b60e07.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "ZOZBO9CF",

                            "totalFollow": 10076

                        },

                        {

                            "id": "IWZAZ9I9",

                            "name": "Doja Cat",

                            "link": "/nghe-si/Doja-Cat",

                            "spotlight": **false**,

                            "alias": "Doja-Cat",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/1/4/5/0/1450fa09b81fcdcc5a95723ccafbff64.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/1/4/5/0/1450fa09b81fcdcc5a95723ccafbff64.jpg",

                            "isOA": **false**,

                            "isOABrand": **false**,

                            "playlistId": "6UB6A87I",

                            "totalFollow": 18553

                        },

                        {

                            "id": "IWZ9ZO9I",

                            "name": "Chris Brown",

                            "link": "/Chris-Brown",

                            "spotlight": **false**,

                            "alias": "Chris-Brown",

                            "thumbnail": "https://photo-resize-zmp3.zmdcdn.me/w240\_r1x1\_jpeg/avatars/5/6/b/b/56bbe908397823bcff982cfc4eee5d14.jpg",

                            "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w360\_r1x1\_jpeg/avatars/5/6/b/b/56bbe908397823bcff982cfc4eee5d14.jpg",

                            "isOA": **true**,

                            "isOABrand": **false**,

                            "playlistId": "ZOEUZ77Z",

                            "totalFollow": 12252

                        }

                    ],

                    "artistsNames": "Nhiều nghệ sĩ",

                    "playItemMode": 0,

                    "subType": 24,

                    "uid": 0,

                    "thumbnailM": "https://photo-resize-zmp3.zmdcdn.me/w320\_r1x1\_jpeg/cover/c/1/a/1/c1a14402669ed2a7613bb8811e506b4f.jpg",

                    "isShuffle": **true**,

                    "isPrivate": **false**,

                    "userName": "Zing MP3",

                    "isAlbum": **false**,

                    "textType": "Playlist",

                    "isSingle": **false**

                }

            ],

            "genre": {

                "name": "Nổi bật"

            }

        },

}

]

}

→ Client tiến hành đọc JSON trả về từ Zingmp3 và hiển thị dữ liệu lên UI.

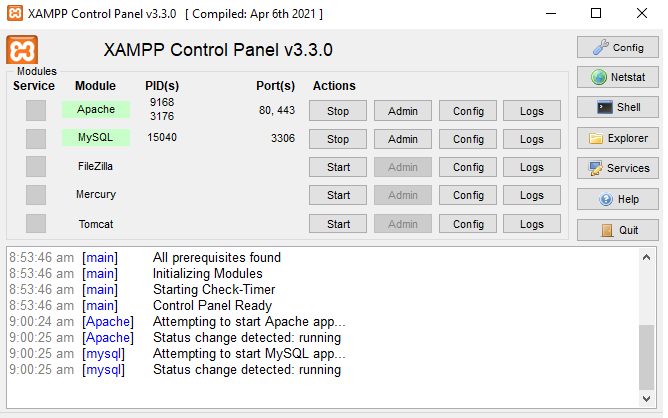
# SETUP SẢN PHẨM, THIẾT KẾ VÀ TIẾN HÀNH LẬP TRÌNH

## Setup database MySQL sử dụng hệ quản trị PhpMyadmin thông qua phần mềm Xampp:

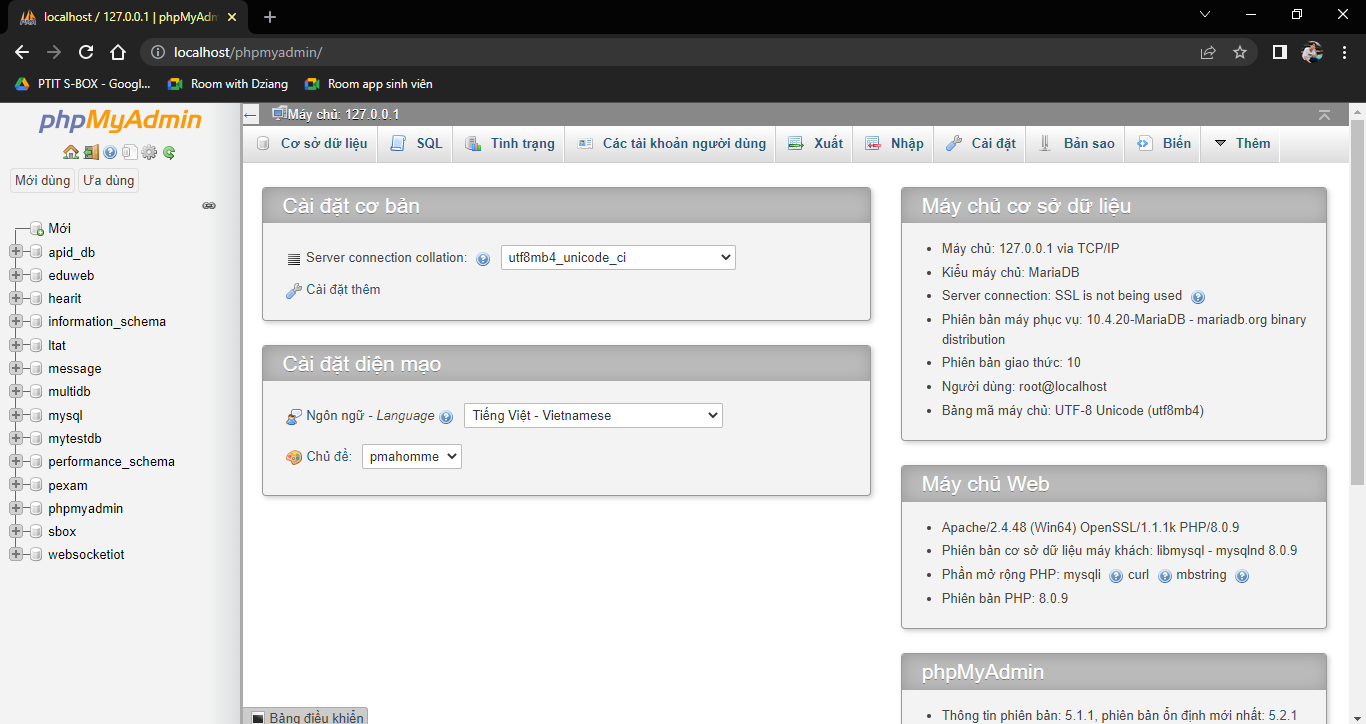
# 5.1.1 Cài đặt phần mềm Xampp:

- Link download: <https://www.apachefriends.org/download.html>

- Sau khi download xong chúng ta tiến hành cài đặt theo hướng dẫn: <https://wiki.matbao.net/xampp-la-gi-cach-cai-dat-va-su-dung-xampp-tren-windows-va-linux/>



* Sau khi cài đặt thành công thì chúng ta ấn “Start” Apache và MySQL để khởi chạy MySQL bản web.
* Để truy cập vào hệ quản trị PhpMyadmin bằng cách truy cập vào đường dẫn: localhost/phpmyadmin

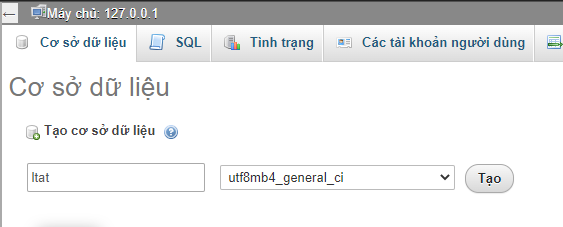


* Lập tức trang quản trị MySQL sẽ hiển thị ra như thế này.

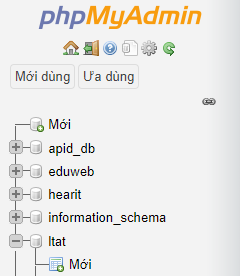
**5.1.2 Tạo database và table:**

- Ấn vào nút “Mới” ở thanh bên trái, nhập tên database là “ltat” và chọn option là “utf8mb4\_general\_ci” và sau đó ấn “Tạo”.

Graphical user interface, text, application

Description automatically generated 

- Để tạo bảng mới, ấn “Mới” ở nhánh con của Database. Sau đó màn hình tạo cấu trúc table sẽ được hiển thị.

Graphical user interface

Description automatically generated

- Tạo table **user** có cấu trúc như sau:



- Tạo table **rcm** có cấu trúc như sau:

Graphical user interface, application

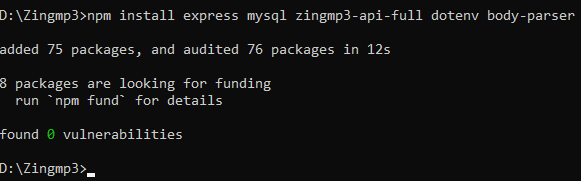
Description automatically generated

**5.2 Setup thư viện phía Server (BE) sử dụng Nodejs + Express:**

- Tạo thư mục mới tên “Zingmp3” hoặc bất kì tên nào mình muốn.

- Bật CMD và trỏ vào thư mục vừa tạo gõ lệnh: “**npm init**” sau đó ấn Enter cho đến khi nào CMD hỏi “Is this OK?” thì mới thôi. Sau đó tiếp tục chạy lệnh “**npm install express mysql zingmp3-api-full dotenv body-parser”** để tiến hành cài đặt các thư viện cần thiết.

Text

Description automatically generated

* Tạo file app.js và paste đoạn code dưới đây để setup API phía Server.
* const express = require('express');
* const port = process.env.PORT || 3002;
* const app = express();
* const { ZingMp3 } = require("zingmp3-api-full");
* const bodyParser = require("body-parser");
* const pool = require("./dbconnection");
* const mylib = require("./mylib");
* app.use(bodyParser.raw({inflate:true,type:'application/json'}))
* app.use(bodyParser.json());
* app.use(bodyParser.urlencoded({extended: true}));
* app.post('/regist', (req,res) => {
* const body = JSON.parse(req.body);
* console.log(body);
* pool.query("SELECT \* FROM user WHERE username = ?",[body.username],(err,resGet)=>{
* if(err) throw err;
* if(resGet.length!=0){
* return res.json({
* error:true,
* message:"Tài khoản này đã tồn tại"
* });
* }
* pool.query("INSERT INTO user VALUES (?,?,?)",
* [body.username,Buffer.from(body.password + process.env.SALT\_PASSWORD).toString('base64'),body.fullname]
* ,(errIns,resIns)=>{
* if(errIns) throw errIns;
* return res.json({
* error:false,
* message:"Đăng ký thành công"
* });
* });
* });
* });
* app.post('/login',(req,res)=>{
* const body = JSON.parse(req.body);
* pool.query("SELECT \* FROM user WHERE username = ? AND password = ?",
* [body.username,Buffer.from(body.password+process.env.SALT\_PASSWORD).toString('base64')],
* (errGet,resGet)=>{
* if(errGet) throw errGet;
* if(resGet.length==0){
* return res.json({
* error:true,
* message:"Sai thông tin đăng nhập"
* });
* } else {
* const row = mylib.parseToJSONFrDB(resGet)[0];
* return res.json({
* error:false,
* message:"Đăng nhập thành công",
* data: {
* fullName:row.fullname,
* accessToken:mylib.generAccessTokenUser(row.username)
* }
* });
* }
* });
* });
* app.get('/search', (req, res) => {
* try {
* const header = req.headers;
* console.log(header.authorization);
* const authState = mylib.verifyAuthorizationUser(header.authorization.split(' ')[1]);
* const user = authState.data.username;
* const qs = req.query;
* const stringSearch = qs.string\_search;
* pool.query("INSERT INTO rcm VALUES (?,?)",[user,stringSearch]);
* ZingMp3.search(stringSearch).then((data)=>{
* res.json({
* error:false,
* message:"Success",
* data:data.data.songs
* });
* });
* } catch (err){
* console.log(err);
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_banner',(req,res)=>{
* ZingMp3.getHome().then((data) => {
* const dataBanner = JSON.parse(JSON.stringify(data)).data.items[0].items;
* res.json({
* error:false,
* message:"Success",
* data:dataBanner
* });
* });
* });
* app.get('/get\_home',(req,res)=>{
* try {
* ZingMp3.getHome().then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:JSON.parse(JSON.stringify(data)).data
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_song',async (req,res)=>{
* try {
* const qs = req.query;
* const id = qs.id;
* const data = await ZingMp3.getSong(id);
* res.json({
* error:false,
* message:"Success",
* data:data.data
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_new\_release',(req,res)=>{
* ZingMp3.getHome().then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:JSON.parse(JSON.stringify(data)).data.items[3].items
* });
* });
* });
* app.get('/get\_detail\_playlist',(req,res)=>{
* try {
* const qs = req.query;
* const id = qs.id;
* ZingMp3.getDetailPlaylist(id).then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_top\_100',(req,res)=>{
* try {
* ZingMp3.getTop100().then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_chart\_home',(req,res)=>{
* try {
* ZingMp3.getChartHome().then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data.RTChart.items
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_new\_release\_chart',(req,res)=>{
* try {
* ZingMp3.getNewReleaseChart().then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data.items
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_inf\_song',(req,res)=>{
* try {
* const qs = req.query;
* const id = qs.id;
* ZingMp3.getInfoSong(id).then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_hot\_artist',(req,res)=>{
* ZingMp3.getHome().then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:JSON.parse(JSON.stringify(data)).data.items[5].items
* });
* });
* });
* app.get('/get\_artist',(req,res)=>{
* try {
* const qs = req.query;
* const name = qs.name;
* ZingMp3.getArtist(name).then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_list\_song\_artist',(req,res)=>{
* try {
* const qs = req.query;
* const id = qs.id;
* const page = qs.page;
* const count = qs.count;
* ZingMp3.getListArtistSong(id,page,count).then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_lyric',(req,res)=>{
* try {
* const qs = req.query;
* const id = qs.id;
* ZingMp3.getLyric(id).then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_list\_mv',(req,res)=>{
* try {
* const qs = req.query;
* const id = qs.id;
* const page = qs.page;
* const count = qs.count;
* ZingMp3.getListMV(id,page,count).then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_category\_mv',(req,res)=>{
* try {
* const qs = req.query;
* const id = qs.id;
* ZingMp3.getCategoryMV(id).then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.get('/get\_video\_mv',(req,res)=>{
* try {
* const qs = req.query;
* const id = qs.id;
* ZingMp3.getVideo(id).then((data) => {
* res.json({
* error:false,
* message:"Success",
* data:data.data
* });
* });
* } catch (error) {
* res.json({
* error:true,
* message:"Có lỗi xảy ra",
* });
* }
* });
* app.listen(port, () => console.log('Server is running on port '+port));
* Tạo file dbconnection.js và paste đoạn code dưới đây để setup kết nối đến hệ quản trị CSDL MySQL:

const mysql = require('mysql');

const dotenv = require('dotenv');

dotenv.config();

const configuration = {

    host: process.env.DB\_HOST,

    user: process.env.DB\_USERNAME,

    password: process.env.DB\_PASSWORD,

    database: process.env.DB\_NAME,

    charset:'utf8mb4'

}

module.exports = mysql.createPool(configuration);

* Tạo file mylib.js và paste đoạn code dưới đây để tạo một số hàm được tái sử dụng nhiều lần:
* const saltedSha256 = require('salted-sha256');
* var parseToJSONFrDB = function(a){
* return JSON.parse(JSON.stringify(a))
* }
* var generRandString = function(length) {
* var text = "";
* var possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
* for (var i = 0; i < length; i++)
* text += possible.charAt(Math.floor(Math.random() \* possible.length));
* return text;
* }
* var generAccessTokenUser = function(username){
* const headerServer = {
* "typ": "JWT",
* "alg": "HS256",
* "for": "user"
* };
* const payloadServer = {
* username:username
* };
* const base64HeaderStr = Buffer.from(JSON.stringify(headerServer)).toString("base64");
* const base64PayloadStr = Buffer.from(JSON.stringify(payloadServer)).toString("base64");
* const dataCombinHeadPay = base64HeaderStr + '.' + base64PayloadStr;
* const hashedData = saltedSha256(dataCombinHeadPay, process.env.SECRET\_KEY);
* const signature = Buffer.from(hashedData).toString("base64");
* const accesstokenServer = base64HeaderStr + '.' + base64PayloadStr + '.' + signature;
* return accesstokenServer.replace(/[;'"-]/g,'');
* }
* var decodeFromBase64 = function(base64String){
* return Buffer.from(base64String,'base64').toString('ascii');
* }
* var verifyAuthorizationUser = function(accesstokenClient){
* const headerServer = {
* "typ": "JWT",
* "alg": "HS256",
* "for": "user"
* };
* const payloadAccessClient = JSON.parse(decodeFromBase64(accesstokenClient.split('.')[1]));
* if(typeof payloadAccessClient.username === undefined) return {authState:0,data:null};
* const base64HeaderStr = Buffer.from(JSON.stringify(headerServer)).toString("base64");
* const base64PayloadStr = Buffer.from(JSON.stringify(payloadAccessClient)).toString("base64");
* const dataCombinHeadPay = base64HeaderStr + '.' + base64PayloadStr;
* const hashedData = saltedSha256(dataCombinHeadPay, process.env.SECRET\_KEY);
* const signature = Buffer.from(hashedData).toString("base64");
* const accesstokenServer = (base64HeaderStr + '.' + base64PayloadStr + '.' + signature);
* return {authState:accesstokenClient == accesstokenServer,data:payloadAccessClient};
* };
* module.exports = {
* parseToJSONFrDB:parseToJSONFrDB,
* generRandString:generRandString,
* verifyAuthorizationUser:verifyAuthorizationUser,
* generAccessTokenUser:generAccessTokenUser,
* }

**5.3 Setup thư viện phía Client sử dụng Audio Players, Bloc, Record, Google Speech và SharedPreferences:**

**-** Mở CMD và trỏ vào thư mục mà mình muốn tạo project Flutter.

- Gõ lệnh “flutter create <tên project>” và sau đó hệ thống sẽ tạo project Flutter cho chúng ta.

- Sử dụng Android Studio IDE và open project mà chúng ta vừa mới tạo. Mở file **pubspec.yaml** (đây là file import thư viện của project chúng ta) và paste đoạn mã dưới đây vào mục **dependencies** sau đó ấn **Pub get** nằm ở góc phải màn hình để import thư viện vào project.

audioplayers: ^3.0.1  
flutter\_bloc: ^8.1.2  
record: ^4.4.4  
google\_speech: ^2.1.2  
shared\_preferences: ^2.0.17

* Audio Players: Thư viện chuyên quản lí và xử lí âm thanh. Trong project này chúng ta sẽ sử dụng nó trong việc phát nhạc từ URL.
* Flutter Bloc: Thư viện được sử dụng để quản lí sự kiện và trạng thái. Chia tách 2 thành phần là UI và Bussiness logic code giúp lập trình viên có thể quản lí dễ dàng các sự kiện và trạng thái được phát đi trong ứng dụng.
* Record: Thư viện giúp chúng ta tiến hành ghi âm người dùng trong module hát karaoke.
* Google Speech: Một thư viện giúp nhận dạng file âm thanh và chuyển hóa voice trong âm thanh đó thành text để tiến hành đối chiếu với lyric của bài hát cung cấp.
* SharedPreferences: Thư viện giúp quản lí Local Storage. Lưu dữ liệu hoặc các trạng thái của ứng dụng dưới local. Thư viện này chúng ta sử dụng để duy trì trạng thái đăng nhập và lưu session của người dùng phía Client.

**5.4 Thiết kế UX/UI:**

**-** Lấy màu đỏ chủ đạo của logo PTIT làm màu chủ đạo của ứng dụng. Bản thiết kế của chính tác giả tự nghĩ ra và làm nên.

- Các bố cục của UI giúp cho người dùng có trải nghiệm tốt trên nền giao diện đẹp mắt nhất có thể.

**a) Trang đăng nhập:**

Graphical user interface

Description automatically generated

*Hình ảnh minh họa cho trang đăng nhập*

- Sử dụng 2 logo bao gồm logo của PTIT và logo của ngành Công nghệ Đa phương tiện làm biểu tượng của ứng dụng.

- Gồm 2 ô nhập liệu tài khoản và mật khẩu.

- 2 nút “Đăng nhập” và “Đăng ký” cho người dùng lựa chọn.

- Dưới đây là đoạn code design trang đăng nhập:

import 'package:ap\_03\_luongquocdien/network/route/auth/auth\_service.dart';  
import 'package:ap\_03\_luongquocdien/page/main\_page.dart';  
import 'package:ap\_03\_luongquocdien/page/regist\_page.dart';  
import 'package:flutter/material.dart';  
import 'package:dio/dio.dart' as dio;  
import 'package:ap\_03\_luongquocdien/main.dart';  
  
class LoginPage extends StatefulWidget {  
 const LoginPage({super.key});  
  
 @override  
 State<LoginPage> createState() => \_LoginPage();  
}  
  
class \_LoginPage extends State<LoginPage> {  
  
 final \_textUsernameController = TextEditingController();  
 final \_textPasswordController = TextEditingController();  
 AuthService? service;  
  
 Future<void> login() async {  
 showDialog(  
 context: context,  
 barrierDismissible: false,  
 builder: (BuildContext context) {  
 return Dialog(  
 backgroundColor: Colors.*transparent*,  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: const [  
 CircularProgressIndicator()  
 ],  
 )  
 );  
 },  
 );  
 Map<String,dynamic> body = {  
 "username":\_textUsernameController.text,  
 "password":\_textPasswordController.text  
 };  
 final response = await service!.login(body);  
 Navigator.*pop*(context);  
 if(response.error){   
 showDialog(  
 context: context,  
 builder: (context) {  
 return AlertDialog(  
 title: const Text('Trạng thái đăng nhập:'),  
 content: SingleChildScrollView(  
 child: ListBody(  
 children: const <Widget>[  
 Text('Tài khoản hoặc mật khẩu của bạn không chính xác.'),  
 ],  
 ),  
 ),  
 actions: <Widget>[  
 TextButton(  
 child: const Text('Đã rõ',style: TextStyle(color: Colors.*orange*),),  
 onPressed: () {  
 Navigator.*of*(context).pop();  
 },  
 ),  
 ],  
 );  
 }  
 );  
 } else {  
 shared.setString("accessToken", response.data!.accessToken);  
 shared.setString("fullName", response.data!.fullName);  
 Navigator.*push*(context, MaterialPageRoute(builder: (context)=> const MainPageProvider()));  
 }  
 }  
  
 @override  
 void initState() {  
 // *TODO: implement initState* super.initState();  
 service = AuthService(dio.Dio());  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 body: Center(  
 child: SizedBox(  
 width: 600,  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Container(  
 decoration: const BoxDecoration(  
 borderRadius: BorderRadius.all(Radius.circular(100))  
 ),  
 child: const Image(image: AssetImage("images/logo\_ptit.png"),width: 90,height: 90,),  
 ),  
 const Padding(padding: EdgeInsets.only(right: 50)),  
 Container(  
 decoration: const BoxDecoration(  
 borderRadius: BorderRadius.all(Radius.circular(100))  
 ),  
 child: const Image(image: AssetImage("images/logo\_multi.png"),width: 100,height: 100,),  
 )  
 ],  
 ),  
 const Text(  
 "Đăng nhập",  
 style: TextStyle(  
 color: Colors.*red*,  
 fontSize: 60,  
 fontWeight: FontWeight.*bold* ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 20)),  
 TextField(  
 controller: \_textUsernameController,  
 decoration: const InputDecoration(  
 hintText: "Nhập tài khoản",  
 labelText: "Tài khoản",  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 30)),  
 TextField(  
 controller: \_textPasswordController,  
 onSubmitted: (stringSearch){  
 login();  
 },  
 textInputAction: TextInputAction.go,  
 obscureText: true,  
 decoration: const InputDecoration(  
 hintText: "Nhập mật khẩu",  
 labelText: "Mật khẩu",  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 50)),  
 GestureDetector(  
 onTap: (){  
 login();  
 },  
 child: Container(  
 padding: const EdgeInsets.symmetric(vertical: 8),  
 decoration: const BoxDecoration(  
 color: Colors.*red*,  
 borderRadius: BorderRadius.all(Radius.circular(8))  
 ),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: const [  
 Text(  
 "Đăng nhập",  
 style: TextStyle(  
 color: Colors.*white*,  
 fontSize: 20  
 ),  
 )  
 ],  
 ),  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 12)),  
 const Text(  
 "Bạn chưa có tài khoản?",  
 style: TextStyle(  
 color: Colors.*grey*,  
 fontSize: 18  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 12)),  
 GestureDetector(  
 onTap: (){  
 Navigator.*push*(context, MaterialPageRoute(builder: (context) => const RegistPage()));  
 },  
 child: Container(  
 padding: const EdgeInsets.symmetric(vertical: 8),  
 decoration: BoxDecoration(  
 color: Colors.*white*,  
 border: Border.all(color: Colors.*red*),  
 borderRadius: const BorderRadius.all(Radius.circular(8))  
 ),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: const [  
 Text(  
 "Đăng ký",  
 style: TextStyle(  
 color: Colors.*red*,  
 fontSize: 20  
 ),  
 )  
 ],  
 ),  
 ),  
 ),  
 ],  
 ),  
 ),  
 ),  
 );  
 }  
}

**b) Trang đăng ký:**

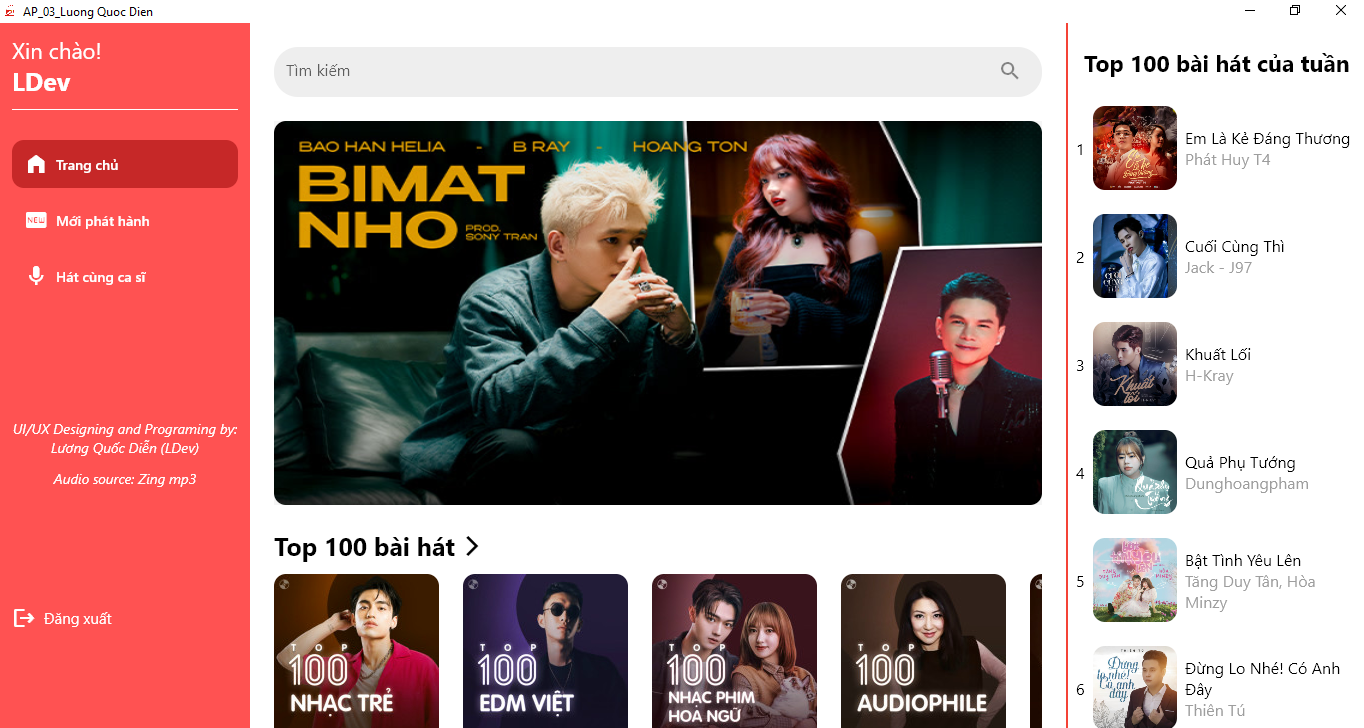
**Graphical user interface, application

Description automatically generated**

*Hình ảnh minh họa cho trang đăng ký*

* Sử dụng 4 ô nhập liệu bao gồm Họ tên, tài khoản, mật khẩu và nhập lại mật khẩu.
* 1 nút kích hoạt lệnh đăng ký tài khoản
* Xử lí bắt ngoại lệ và ràng buộc: Tài khoản và họ tên không được trống, mật khẩu phải có tối thiểu 6 ký tự.
* Dưới đây là đoạn code dùng để design trang đăng ký:
* import 'package:dio/dio.dart' as dio;  
  import 'package:ap\_03\_luongquocdien/network/route/auth/auth\_service.dart';  
  import 'package:flutter/material.dart';  
    
  class RegistPage extends StatefulWidget{  
   const RegistPage({super.key});  
    
   @override  
   State<RegistPage> createState() => \_RegistPage();  
  }  
    
  class \_RegistPage extends State<RegistPage> {  
    
   final \_textUsernameController = TextEditingController();  
   final \_textPasswordController = TextEditingController();  
   final \_textConfirmPasswordController = TextEditingController();  
   final \_textFullnameController = TextEditingController();  
   bool isContainUsername = false;  
   bool isIncorrectPassword = false;  
   bool isFullnameEmpty = false;  
   bool isInvalidUsername = false;  
   bool isInvalidPassword = false;  
   AuthService? service;  
    
   Future<void> register() async {  
   if(\_textFullnameController.text.isEmpty ||  
   \_textUsernameController.text.isEmpty ||  
   \_textPasswordController.text.length<6 ||  
   \_textPasswordController.text != \_textConfirmPasswordController.text  
   ){  
   return;  
   }  
   showDialog(  
   context: context,  
   barrierDismissible: false,  
   builder: (BuildContext context) {  
   return Dialog(  
   child: Row(  
   mainAxisSize: MainAxisSize.min,  
   children: const [  
   CircularProgressIndicator(),  
   ],  
   ),  
   );  
   },  
   );  
   Map<String,dynamic> body = {  
   "username":\_textUsernameController.text,  
   "password":\_textPasswordController.text,  
   "fullname":\_textFullnameController.text  
   };  
   final response = await service!.registAccount(body);  
   Navigator.*pop*(context);  
   if(response.error){  
   setState(() {  
   isContainUsername = true;  
   });  
   } else {  
   showDialog(  
   context: context,  
   builder: (context) {  
   return AlertDialog(  
   title: const Text('Đăng ký tài khoản thành công'),  
   content: SingleChildScrollView(  
   child: ListBody(  
   children: const <Widget>[  
   Text('Bạn đã đăng ký tài khoản thành công. Hệ thống sẽ điều hướng bạn sang trang đăng nhập'),  
   ],  
   ),  
   ),  
   actions: <Widget>[  
   TextButton(  
   child: const Text('Đã rõ',style: TextStyle(color: Colors.*orange*),),  
   onPressed: () {  
   Navigator.*of*(context).pop();  
   Navigator.*of*(context).pop();  
   },  
   ),  
   ],  
   );  
   }  
   );  
   }  
   }  
    
   @override  
   void initState() {  
   // *TODO: implement initState* super.initState();  
   service = AuthService(dio.Dio());  
   \_textFullnameController.addListener(() {  
   setState(() {  
   isFullnameEmpty = \_textFullnameController.text.isEmpty;  
   });  
   });  
   \_textUsernameController.addListener(() {  
   setState(() {  
   isInvalidUsername = \_textUsernameController.text.isEmpty;  
   });  
   });  
   \_textPasswordController.addListener(() {  
   isInvalidPassword = (\_textPasswordController.text.length<6)?true:false;  
   });  
   \_textConfirmPasswordController.addListener(() {  
   isIncorrectPassword = (\_textConfirmPasswordController.text != \_textPasswordController.text)?true:false;  
   });  
   }  
    
   @override  
   Widget build(BuildContext context) {  
   return Scaffold(  
   body: Center(  
   child: SizedBox(  
   width: 600,  
   child: Column(  
   mainAxisAlignment: MainAxisAlignment.center,  
   crossAxisAlignment: CrossAxisAlignment.center,  
   children: [  
   const Text(  
   "Đăng ký",  
   style: TextStyle(  
   color: Colors.*red*,  
   fontSize: 60,  
   fontWeight: FontWeight.*bold* ),  
   ),  
   const Padding(padding: EdgeInsets.only(top: 30)),  
   TextField(  
   controller: \_textFullnameController,  
   decoration: const InputDecoration(  
   hintText: "Nhập họ tên",  
   labelText: "Họ tên",  
   ),  
   ),  
   Visibility(  
   visible: isFullnameEmpty,  
   child: Row(  
   mainAxisAlignment: MainAxisAlignment.end,  
   children: const [  
   Text(  
   "Họ tên không được để trống!",  
   style: TextStyle(  
   color: Colors.*red*,  
   fontSize: 12  
   ),  
   ),  
   ],  
   )  
   ),  
   const Padding(padding: EdgeInsets.only(top: 30)),  
   TextField(  
   controller: \_textUsernameController,  
   decoration: const InputDecoration(  
   hintText: "Nhập tài khoản",  
   labelText: "Tài khoản",  
   ),  
   ),  
   Visibility(  
   visible: (isInvalidUsername)?true:(isContainUsername),  
   child: Row(  
   mainAxisAlignment: MainAxisAlignment.end,  
   children: [  
   if(isContainUsername) const Text(  
   "Tài khoản này đã tồn tại! Vui lòng sử dụng tài khoản khác",  
   style: TextStyle(  
   color: Colors.*red*,  
   fontSize: 12  
   ),  
   ),  
   if(isInvalidUsername) const Text(  
   "Tài khoản không được để trống!",  
   style: TextStyle(  
   color: Colors.*red*,  
   fontSize: 12  
   ),  
   ),  
   ],  
   )  
   ),  
   const Padding(padding: EdgeInsets.only(top: 20)),  
   TextField(  
   controller: \_textPasswordController,  
   obscureText: true,  
   decoration: const InputDecoration(  
   hintText: "Nhập mật khẩu",  
   labelText: "Mật khẩu",  
   ),  
   ),  
   const Padding(padding: EdgeInsets.only(top: 20)),  
   TextField(  
   controller: \_textConfirmPasswordController,  
   obscureText: true,  
   decoration: const InputDecoration(  
   hintText: "Nhập mật khẩu",  
   labelText: "Nhập lại mật khẩu",  
   ),  
   ),  
   Visibility(  
   visible: (isInvalidPassword)?true:(isIncorrectPassword),  
   child: Row(  
   mainAxisAlignment: MainAxisAlignment.end,  
   children: [  
   if(isIncorrectPassword) const Text(  
   "Mật khẩu nhập lại không chính xác!",  
   style: TextStyle(  
   color: Colors.*red*,  
   fontSize: 12  
   ),  
   ),  
   if(isInvalidPassword) const Text(  
   "Mật khẩu phải có ít nhất 6 ký tự!",  
   style: TextStyle(  
   color: Colors.*red*,  
   fontSize: 12  
   ),  
   ),  
   ],  
   )  
   ),  
   const Padding(padding: EdgeInsets.only(top: 30)),  
   GestureDetector(  
   onTap: (){  
   register();  
   },  
   child: Container(  
   padding: const EdgeInsets.symmetric(vertical: 8),  
   decoration: const BoxDecoration(  
   color: Colors.*red*,  
   borderRadius: BorderRadius.all(Radius.circular(8))  
   ),  
   child: Row(  
   mainAxisAlignment: MainAxisAlignment.center,  
   crossAxisAlignment: CrossAxisAlignment.center,  
   children: const [  
   Text(  
   "Đăng ký",  
   style: TextStyle(  
   color: Colors.*white*,  
   fontSize: 20  
   ),  
   )  
   ],  
   ),  
   ),  
   ),  
   const Padding(padding: EdgeInsets.only(top: 25)),  
   GestureDetector(  
   onTap: (){  
   Navigator.*pop*(context);  
   },  
   child: Row(  
   mainAxisAlignment: MainAxisAlignment.start,  
   children: const [  
   Icon(Icons.*arrow\_back*),  
   Text(  
   "Quay lại",  
   style: TextStyle(  
   fontSize: 17  
   ),  
   )  
   ],  
   ),  
   )  
   ],  
   ),  
   ),  
   ),  
   );  
   }  
  }

**c) Trang chủ:**

****

*Hình ảnh minh họa về trang chủ của ứng dụng.*

- Mục trang chủ của ứng dụng có thanh tìm kiếm, các danh mục như banner, top 100 bài hát, nghệ sĩ đang “hot” và top 100 bài hát của tuần.

- Khi người dùng ấn vào bài hát bất kì thì ứng dụng sẽ phát ra nhạc và trình điều khiển âm nhạc sẽ xuất hiện ở phía dưới màn hình.

Graphical user interface, application

Description automatically generated

*Ảnh minh họa trình điều khiển phát nhạc.*

- Dưới đây là đoạn code thiết kế:

import 'package:ap\_03\_luongquocdien/bloc/play\_music/bloc.dart';  
import 'package:ap\_03\_luongquocdien/bloc/play\_music/event.dart';  
import 'package:ap\_03\_luongquocdien/bloc/play\_music/state.dart';  
import 'package:ap\_03\_luongquocdien/main.dart';  
import 'package:ap\_03\_luongquocdien/page/home\_page.dart';  
import 'package:ap\_03\_luongquocdien/page/karaoke\_page.dart';  
import 'package:ap\_03\_luongquocdien/page/new\_release\_page.dart';  
import 'package:audioplayers/audioplayers.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter\_bloc/flutter\_bloc.dart';  
  
class MainPageProvider extends StatelessWidget {  
 const MainPageProvider({Key? key}) : super(key: key);  
  
 @override  
 Widget build(BuildContext context) {  
 return BlocProvider(  
 create: (context) => PlayMusicBloc(PlayMusicState()),  
 child: const MainPage(),  
 );  
 }  
}  
  
  
class MainPage extends StatefulWidget{  
 const MainPage({super.key});  
  
 @override  
 State<MainPage> createState() => \_MainPage();  
}  
  
class \_MainPage extends State<MainPage> {  
  
 final pageController = PageController(initialPage: 0);  
 int indexPage = 0;  
  
 @override  
 void initState() {  
 // *TODO: implement initState* super.initState();  
 BlocProvider.*of*<PlayMusicBloc>(context).add(InitEvent());  
 }  
  
 @override  
 void dispose() {  
 // *TODO: implement dispose* super.dispose();  
 pageController.dispose();  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 scrollBehavior: MyCustomScrollBehavior(),  
 debugShowCheckedModeBanner: false,  
 theme: ThemeData(  
 primarySwatch: Colors.*red*,  
 scaffoldBackgroundColor: Colors.*white*,  
 ),  
 home: Scaffold(  
 body: Stack(  
 children: [  
 Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 mainAxisSize: MainAxisSize.max,  
 children: [  
 Container(  
 width: 250,  
 height: double.*infinity*,  
 decoration: const BoxDecoration(  
 color: Colors.*redAccent*,  
 ),  
 child: Padding(  
 padding: const EdgeInsets.all(12),  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.stretch,  
 children: [  
 const Text(  
 "Xin chào!",  
 style: TextStyle(  
 color: Colors.*white*,  
 fontSize: 22  
 ),  
 ),  
 Flexible(  
 child: Text(  
 "${shared.getString("fullName")}",  
 style: const TextStyle(  
 color: Colors.*white*,  
 fontSize: 25,  
 fontWeight: FontWeight.*bold* ),  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 12)),  
 const Divider(color: Colors.*white*,height: 1,),  
 const Padding(padding: EdgeInsets.only(top: 30)),  
 GestureDetector(  
 onTap: (){  
 setState(() {  
 indexPage=0;  
 pageController.animateToPage(indexPage, duration: const Duration(milliseconds: 500), curve: Curves.*easeInOut*);  
 });  
 },  
 child: Container(  
 padding: const EdgeInsets.all(12),  
 decoration: BoxDecoration(  
 color: (indexPage==0)?Colors.*red*[800]:Colors.*redAccent*,  
 borderRadius: const BorderRadius.all(Radius.circular(12))  
 ),  
 child: Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: const [  
 Icon(Icons.*home\_filled*, color: Colors.*white*,),  
 Padding(padding: EdgeInsets.only(right: 8)),  
 Text(  
 "Trang chủ",  
 style: TextStyle(  
 color: Colors.*white*,  
 fontWeight: FontWeight.*w600* ),  
 )  
 ],  
 ),  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 8)),  
 GestureDetector(  
 onTap: (){  
 setState(() {  
 indexPage=1;  
 pageController.animateToPage(indexPage, duration: const Duration(milliseconds: 500), curve: Curves.*easeInOut*);  
 });  
 },  
 child: Container(  
 padding: const EdgeInsets.all(12),  
 decoration: BoxDecoration(  
 color: (indexPage==1) ? Colors.*red*[800]: Colors.*redAccent*,  
 borderRadius: const BorderRadius.all(Radius.circular(12))  
 ),  
 child: Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: const [  
 Icon(Icons.*fiber\_new\_rounded*, color: Colors.*white*,),  
 Padding(padding: EdgeInsets.only(right: 8)),  
 Text(  
 "Mới phát hành",  
 style: TextStyle(  
 color: Colors.*white*,  
 fontWeight: FontWeight.*w600* ),  
 )  
 ],  
 ),  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 8)),  
 GestureDetector(  
 onTap: (){  
 setState(() {  
 indexPage=2;  
 pageController.animateToPage(indexPage, duration: const Duration(milliseconds: 500), curve: Curves.*easeInOut*);  
 });  
 },  
 child: Container(  
 padding: const EdgeInsets.all(12),  
 decoration: BoxDecoration(  
 color: (indexPage==2) ? Colors.*red*[800] : Colors.*redAccent*,  
 borderRadius: const BorderRadius.all(Radius.circular(12))  
 ),  
 child: Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: const [  
 Icon(Icons.*mic*, color: Colors.*white*,),  
 Padding(padding: EdgeInsets.only(right: 8)),  
 Text(  
 "Hát cùng ca sĩ",  
 style: TextStyle(  
 color: Colors.*white*,  
 fontWeight: FontWeight.*w600* ),  
 )  
 ],  
 ),  
 ),  
 ),  
 const Spacer(),  
 const Text(  
 "UI/UX Designing and Programing by: Lương Quốc Diễn (LDev)",  
 textAlign: TextAlign.center,  
 style: TextStyle(  
 color: Colors.*white*,  
 fontStyle: FontStyle.italic,  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 12)),  
 const Text(  
 "Audio source: Zing mp3",  
 textAlign: TextAlign.center,  
 style: TextStyle(  
 color: Colors.*white*,  
 fontStyle: FontStyle.italic  
 ),  
 ),  
 const Spacer(),  
 InkWell(  
 onTap:(){  
 shared.setString("fullName", "");  
 Navigator.*pushReplacement*(context, MaterialPageRoute(builder: (context) => const MyApp()));  
 },  
 child: Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 mainAxisAlignment: MainAxisAlignment.start,  
 children: const [  
 Icon(Icons.*logout*,color: Colors.*white*,),  
 Padding(padding: EdgeInsets.only(right: 8)),  
 Text(  
 "Đăng xuất",  
 style: TextStyle(  
 color: Colors.*white*,  
 fontSize: 15  
 ),  
 )  
 ],  
 ),  
 )  
 ],  
 ),  
 )  
 ),  
 Expanded(  
 child: PageView(  
 physics: const NeverScrollableScrollPhysics(),  
 scrollDirection: Axis.vertical,  
 controller: pageController,  
 children: const [  
 HomePage(),  
 NewReleasePage(),  
 KaraokePage()  
 ],  
 ),  
 )  
 ],  
 ),  
 BlocListener<PlayMusicBloc,PlayMusicState>(  
 listenWhen: (prev,curr){  
 return curr is EndKaraokeState || curr is KaraokePointingState;  
 },  
 child: Container(),  
 listener: (context,state){  
 if(state is EndKaraokeState) {  
 Navigator.*pop*(context);  
 showDialog(  
 context: context,  
 builder: (context) {  
 return AlertDialog(  
 title: const Text('KẾT QUẢ:'),  
 content: SingleChildScrollView(  
 child: ListBody(  
 children: [  
 Text('Điểm số: ${state.point}'),  
 ],  
 ),  
 ),  
 actions: <Widget>[  
 TextButton(  
 child: const Text('OK',style: TextStyle(color: Colors.*orange*),),  
 onPressed: () {  
 Navigator.*of*(context).pop();  
 },  
 ),  
 ],  
 );  
 }  
 );  
 } else if (state is KaraokePointingState){  
 showDialog(  
 context: context,  
 barrierDismissible: false,  
 builder: (BuildContext context) {  
 return Dialog(  
 backgroundColor: Colors.*transparent*,  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: const [  
 CircularProgressIndicator()  
 ],  
 )  
 );  
 },  
 );  
 }  
 },  
 ),  
 BlocBuilder<PlayMusicBloc,PlayMusicState>(  
 buildWhen: (previous, current) {  
 return current is PlaySongState;  
 },  
 builder: (context,state){  
 if(state is PlaySongState) {  
 return Positioned(  
 bottom: 0,  
 child: Container(  
 padding: const EdgeInsets.all(12),  
 height: 90,  
 width: 1370,  
 decoration: BoxDecoration(  
 border: Border.all(color: Colors.*red*,width: 0.3),  
 color: Colors.*white*,  
 ),  
 child: Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 mainAxisAlignment: MainAxisAlignment.start,  
 children: [  
 //*TODO: Button play* BlocBuilder<PlayMusicBloc,PlayMusicState>(  
 buildWhen: (prev,curr){  
 return curr is PlaySongState || curr is PauseSongState || curr is ResumeSongState;  
 },  
 builder: (context,state){  
 return GestureDetector(  
 onTap: (){  
 BlocProvider.*of*<PlayMusicBloc>(context).add((state is PlaySongState || state is ResumeSongState)?PauseSongEvent():ResumeSongEvent());  
 },  
 child: Icon((state is PlaySongState || state is ResumeSongState)?Icons.*pause\_circle*:Icons.*play\_circle\_fill*,size: 40,)  
 );  
 }  
 ),  
 const Padding(padding: EdgeInsets.only(right: 8)),  
 //*TODO: Image thumb music* BlocBuilder<PlayMusicBloc,PlayMusicState>(  
 buildWhen: (prev,curr){  
 return curr is PlaySongState;  
 },  
 builder: (context,state){  
 if(state is PlaySongState){  
 return SizedBox(  
 width: 300,  
 child: Row(  
 children: [  
 ClipRRect(  
 borderRadius: const BorderRadius.all(Radius.circular(12)),  
 child: Image(image: NetworkImage((state).linkThumb)),  
 ),  
 const Padding(padding: EdgeInsets.only(right: 12)),  
 Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 mainAxisAlignment: MainAxisAlignment.start,  
 children: [  
 const Padding(padding: EdgeInsets.only(top: 4)),  
 Flexible(  
 child: Text(  
 state.title,  
 overflow: TextOverflow.clip,  
 style: const TextStyle(  
 fontWeight: FontWeight.*bold*,  
 fontSize: 15,  
 ),  
 ),  
 ),  
 const Spacer(),  
 Flexible(  
 child: Text(  
 state.artist,  
 style: const TextStyle(  
 fontSize: 14  
 ),  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 4)),  
 ],  
 ),  
 ]  
 ),  
 );  
 } else {  
 return Row(  
 children: [  
 const Padding(padding: EdgeInsets.only(right: 12)),  
 Image.asset("images/logo\_ptit.png")  
 ],  
 );  
 }  
 }  
 ),  
 const Padding(padding: EdgeInsets.only(right: 4)),  
 BlocBuilder<PlayMusicBloc,PlayMusicState>(  
 buildWhen: (prev,curr){  
 return curr is SeekState || curr is PlaySongState || curr is AudioPositionChangeState;  
 },  
 builder: (context,state){  
 if(state is PlaySongState){  
 return Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 const Text(  
 "00:00"  
 ),  
 SizedBox(  
 width: 500,  
 height: 10,  
 child: Slider(  
 max: state.total.inSeconds.toDouble(),  
 value: 0,  
 onChanged: (val) {  
 BlocProvider.*of*<PlayMusicBloc>(context).add(SeekEvent(Duration(seconds: val.toInt())));  
 }  
 ),  
 ),  
 Text(  
 "${state.total.inMinutes.toInt()}:${state.total.inSeconds.toInt()%60}"  
 )  
 ]  
 );  
 } else if(state is SeekState){  
 return Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 Text(  
 "${state.newDuration.inMinutes}:${state.newDuration.inSeconds%60}"  
 ),  
 SizedBox(  
 width: 500,  
 height: 10,  
 child: Slider(  
 max: state.total.inSeconds.toDouble(),  
 value: state.newDuration.inSeconds.toDouble(),  
 onChanged: (val) {  
 BlocProvider.*of*<PlayMusicBloc>(context).add(SeekEvent(Duration(seconds: val.toInt())));  
 }  
 ),  
 ),  
 Text(  
 "${state.total.inMinutes.toInt()}:${state.total.inSeconds.toInt()%60}"  
 )  
 ]  
 );  
 } else if(state is AudioPositionChangeState){  
 return Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 Text(  
 "${state.currPosition.inMinutes}:${state.currPosition.inSeconds%60}"  
 ),  
 SizedBox(  
 width: 500,  
 height: 10,  
 child: Slider(  
 min: 0,  
 max: state.total.inSeconds.toDouble(),  
 value: state.currPosition.inSeconds.toDouble(),  
 onChanged: (val) {  
 BlocProvider.*of*<PlayMusicBloc>(context).add(SeekEvent(Duration(seconds: val.toInt())));  
 }  
 ),  
 ),  
 Text(  
 "${state.total.inMinutes.toInt()}:${state.total.inSeconds.toInt()%60}"  
 )  
 ]  
 );  
 } else {  
 return SizedBox(  
 height:20,  
 child: Slider(  
 min: 0,  
 max: 0,  
 value: 0,   
 ),  
 );  
 }  
 }  
 ),  
 const Spacer(),  
 SizedBox(  
 width: 320,  
 child: Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 const Icon(Icons.*volume\_up*),  
 const Padding(padding: EdgeInsets.only(right: 4)),  
 BlocBuilder<PlayMusicBloc,PlayMusicState>(  
 buildWhen: (prev,curr){  
 return curr is ChangeVolumeState;  
 },  
 builder: (context,state){  
 if(state is ChangeVolumeState){  
 return Slider(  
 max: 1,  
 value: state.volume,  
 onChanged: (val) {  
 BlocProvider.*of*<PlayMusicBloc>(context).add(ChangeVolumeEvent(val));  
 }  
 );  
 } else {  
 return Slider(  
 max: 1,  
 value: 1,  
 onChanged: (val) {  
 BlocProvider.*of*<PlayMusicBloc>(context).add(ChangeVolumeEvent(val));  
 }  
 );  
 }  
 }  
 ),  
 BlocBuilder<PlayMusicBloc,PlayMusicState>(  
 buildWhen: (prev,curr){  
 return curr is DownloadSongState;  
 },  
 builder: (context,state){  
 if(state is DownloadSongState) {  
 return InkWell(  
 onTap: (){  
 BlocProvider.*of*<PlayMusicBloc>(context).add(DownloadSongEvent());  
 },  
 child: Container(  
 padding: const EdgeInsets.all(12),  
 decoration: const BoxDecoration(  
 borderRadius: BorderRadius.all(Radius.circular(50)),  
 ),  
 child: (!state.isDownloadedFinish)?const CircularProgressIndicator():const Icon(Icons.*download*),  
 ),  
 );  
 } else {  
 return InkWell(  
 onTap: (){  
 BlocProvider.*of*<PlayMusicBloc>(context).add(DownloadSongEvent());  
 },  
 child: Container(  
 padding: const EdgeInsets.all(12),  
 decoration: const BoxDecoration(  
 borderRadius: BorderRadius.all(Radius.circular(50)),  
 ),  
 child: const Icon(Icons.*download*),  
 ),  
 );  
 }  
 }  
 ),  
 ],  
 ),  
 ),  
 ]  
 ),  
 ),  
 );  
 } else {  
 return Container();  
 }  
 }  
 ),  
 ]  
 ),  
 ),  
 );  
 }  
}

**d) Mới phát hành:**

**Graphical user interface, application, chat or text message

Description automatically generated**

*Hình ảnh minh họa trang Mới phát hành*

- Tại trang này chúng ta sẽ chia ra làm 2 mục chính:

+ Mục đầu tiên là nhạc mới phát hành. Tại đây ta sẽ chia ra làm 3 dạng nhạc: Tất cả, Việt Nam và các quốc gia “Khác”.

+ Mục thứ 2 là ranking nhạc mới phát hành.

- Dưới đây là code design:

import 'package:ap\_03\_luongquocdien/bloc/play\_music/bloc.dart';  
import 'package:ap\_03\_luongquocdien/bloc/play\_music/event.dart';  
import 'package:ap\_03\_luongquocdien/network/entity/chart/song\_chart\_entity.dart';  
import 'package:flutter/material.dart';  
import 'package:ap\_03\_luongquocdien/main.dart';  
import 'package:flutter\_bloc/flutter\_bloc.dart';  
import '';  
  
class NewReleasePage extends StatefulWidget{  
 const NewReleasePage({super.key});  
  
 @override  
 State<NewReleasePage> createState() => \_NewReleasePage();  
}  
  
class \_NewReleasePage extends State<NewReleasePage> with AutomaticKeepAliveClientMixin<NewReleasePage>{  
  
 List<SongChartEntity?> all = [];  
 List<SongChartEntity?> vPop = [];  
 List<SongChartEntity?> other = [];  
  
 int indexType = 0;  
 List<SongChartEntity> tmp = [];  
 List<SongChartEntity> listSongChart = [];  
  
 Future<void> getNewRelease() async {  
 final response = await service.getNewRelease();  
 setState(() {  
 all = response.data.all;  
 vPop = response.data.vPop;  
 other = response.data.others;  
 for(int i=0;i<all.length;++i){  
 tmp.add(all[i]!);  
 }  
 });  
 }  
  
 Future<void> getNewReleaseChart() async {  
 final response = await service.getNewReleaseChart();  
 setState(() {  
 listSongChart = response.data;  
 });  
 }  
  
 @override  
 void initState() {  
 // *TODO: implement initState* super.initState();  
 getNewRelease();  
 getNewReleaseChart();  
 }  
 void emitEvent(String id, String quality,String title,String artist,String linkThumb,double secondDuration){  
 BlocProvider.*of*<PlayMusicBloc>(context).add(PlaySongEvent(id, quality,title,artist,linkThumb,Duration(seconds: secondDuration.toInt())));  
 }  
 @override  
 Widget build(BuildContext context) {  
 super.build(context);  
 return Scaffold(  
 body: Padding(  
 padding: const EdgeInsets.all(24),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.start,  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Expanded(  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.start,  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 const Text(  
 "Mới phát hành",  
 style: TextStyle(  
 color: Colors.*black*,  
 fontSize: 30,  
 fontWeight: FontWeight.*bold* ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 50)),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.start,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 GestureDetector(  
 onTap: (){  
 setState(() {  
 indexType = 0;  
 tmp.clear();  
 for(int i=0;i<all.length;++i){  
 tmp.add(all[i]!);  
 }  
 });  
 },  
 child: Container(  
 width: 100,  
 padding: const EdgeInsets.symmetric(vertical: 8),  
 decoration: BoxDecoration(  
 borderRadius: const BorderRadius.all(Radius.circular(24)),  
 border: Border.all(color: (indexType==0)?Colors.*red*:Colors.*grey*),  
 color: (indexType==0)?Colors.*red*:Colors.*white* ),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Text(  
 "Tất cả",  
 style: TextStyle(  
 color: (indexType==0)?Colors.*white*:Colors.*grey*,  
 fontSize: 15  
 ),  
 )  
 ],  
 ),  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(right: 12)),  
 GestureDetector(  
 onTap: (){  
 setState(() {  
 indexType = 1;  
 tmp.clear();  
 for(int i=0;i<vPop.length;++i){  
 tmp.add(vPop[i]!);  
 }  
 });  
 },  
 child: Container(  
 width: 100,  
 padding: const EdgeInsets.symmetric(vertical: 8),  
 decoration: BoxDecoration(  
 borderRadius: const BorderRadius.all(Radius.circular(24)),  
 border: Border.all(color: (indexType==1)?Colors.*red*:Colors.*grey*),  
 color: (indexType==1)?Colors.*red*:Colors.*white* ),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Text(  
 "Việt Nam",  
 style: TextStyle(  
 color: (indexType==1)?Colors.*white*:Colors.*grey*,  
 fontSize: 15  
 ),  
 )  
 ],  
 ),  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(right: 12)),  
 GestureDetector(  
 onTap: (){  
 setState(() {  
 indexType = 2;  
 tmp.clear();  
 for(int i=0;i<other.length;++i){  
 tmp.add(other[i]!);  
 }  
 });  
 },  
 child: Container(  
 width: 100,  
 padding: const EdgeInsets.symmetric(vertical: 8),  
 decoration: BoxDecoration(  
 borderRadius: const BorderRadius.all(Radius.circular(24)),  
 border: Border.all(color: (indexType==2)?Colors.*red*:Colors.*grey*),  
 color: (indexType==2)?Colors.*red*:Colors.*white* ),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Text(  
 "Khác",  
 style: TextStyle(  
 color: (indexType==2)?Colors.*white*:Colors.*grey*,  
 fontSize: 15  
 ),  
 )  
 ],  
 ),  
 ),  
 ),  
 ],  
 ),  
 const Padding(padding: EdgeInsets.only(top: 40)),  
 Expanded(  
 child: ListView.builder(  
 scrollDirection: Axis.vertical,  
 itemCount: tmp.length,  
 itemBuilder: (context,index){  
 return GestureDetector(  
 onTap:() async {  
 String id = tmp[index].encodeId!;  
 String linkThumb = tmp[index].thumbnailM!;  
 final response = await service.getSong(id);  
 emitEvent(id,(response.data.quality320=="VIP")?"320":"128",tmp[index].title!,tmp[index].artistsNames!,linkThumb,tmp[index].duration!);  
 },  
 child: Container(  
 height: 120,  
 margin: const EdgeInsets.only(bottom: 12),  
 child: Row(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 mainAxisAlignment: MainAxisAlignment.start,  
 children: [  
 ClipRRect(  
 borderRadius: const BorderRadius.all(Radius.circular(12)),  
 child: Image(image: NetworkImage(tmp[index].thumbnailM!)),  
 ),  
 const Padding(padding: EdgeInsets.only(right: 8)),  
 Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Text(  
 tmp[index].title!,  
 style: const TextStyle(  
 color: Colors.*black*,  
 fontSize: 20,  
 fontWeight: FontWeight.*bold* ),  
 ),  
 const Spacer(),  
 Text(  
 tmp[index].artistsNames!,  
 style: const TextStyle(  
 color: Colors.*black*,  
 fontSize: 14,  
 ),  
 ),  
 const Spacer(),  
 ],  
 )  
 ],  
 ),  
 ),  
 );  
 },  
 ),  
 )  
 ],  
 ),  
 ),  
 SizedBox(  
 width: 400,  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.start,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 const Text(  
 "Xếp hạng mới phát hành",  
 style: TextStyle(  
 color: Colors.*black*,  
 fontSize: 23,  
 fontWeight: FontWeight.*bold* ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 12)),  
 Expanded(  
 child: ListView.builder(  
 scrollDirection: Axis.vertical,  
 itemCount: listSongChart.length,  
 itemBuilder: (context,index){  
 return GestureDetector(  
 onTap:() async {  
 String id = listSongChart[index].encodeId!;  
 String linkThumb = listSongChart[index].thumbnailM!;  
 final response = await service.getSong(id);  
 emitEvent(id,(response.data.quality320=="VIP")?"320":"128",listSongChart[index].title!,listSongChart[index].artistsNames!,linkThumb,listSongChart[index].duration!);  
 },  
 child: Container(  
 margin: const EdgeInsets.only(bottom: 12),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.start,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 Text(  
 "${index+1}",  
 style: const TextStyle(  
 color: Colors.*black*,  
 fontSize: 17,  
 fontWeight: FontWeight.*w500* ),  
 ),  
 const Padding(padding: EdgeInsets.only(right: 8)),  
 ClipRRect(  
 borderRadius: const BorderRadius.all(Radius.circular(12)),  
 child: Image(image: NetworkImage(listSongChart[index].thumbnailM!),width: 100,height: 100,),  
 ),  
 const Padding(padding: EdgeInsets.only(right: 8)),  
 Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Text(  
 listSongChart[index].title!,  
 style: const TextStyle(  
 color: Colors.*black*,  
 fontSize: 15,  
 fontWeight: FontWeight.*bold* ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 12)),  
 Text(  
 listSongChart[index].artistsNames!,  
 style: const TextStyle(  
 color: Colors.*grey*,  
 fontSize: 14.5,  
 ),  
 )  
 ],  
 )  
 ],  
 ),  
 ),  
 );  
 },  
 ),  
 ),  
 ],  
 ),  
 )  
 ],  
 )  
 )  
 );  
 }  
 @override  
 bool get wantKeepAlive => true;  
}

**e) Hát cùng ca sĩ:**

**Graphical user interface

Description automatically generated with medium confidence**

*Hình ảnh minh họa trang Karaoke*

* Tại trang này chúng ta sẽ có 1 thanh tìm kiếm nhạc để karaoke.
* Và có một listview có thể scroll theo lời bài hát
* Sau khi hát xong và sẽ có 1 dialog hiện điểm lên để tính điểm cho người dùng.

Graphical user interface, application

Description automatically generated

*Hình ảnh minh họa hiển thị điểm.*

* Dưới đây là đoạn code design:

import 'dart:io';  
  
import 'package:ap\_03\_luongquocdien/bloc/play\_music/bloc.dart';  
import 'package:ap\_03\_luongquocdien/bloc/play\_music/event.dart';  
import 'package:ap\_03\_luongquocdien/bloc/play\_music/state.dart';  
import 'package:ap\_03\_luongquocdien/network/entity/lyric/lyric\_entity.dart';  
import 'package:ap\_03\_luongquocdien/page/search\_result\_dialog.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter\_bloc/flutter\_bloc.dart';  
import 'package:scrollable\_positioned\_list/scrollable\_positioned\_list.dart';  
  
class KaraokePage extends StatefulWidget{  
 const KaraokePage({super.key});  
  
  
 @override  
 State<KaraokePage> createState() => \_KaraokePage();  
}  
  
class \_KaraokePage extends State<KaraokePage> {  
  
 final \_textSearchController = TextEditingController();  
 final ItemScrollController \_scrollController = ItemScrollController();  
  
 @override  
 void initState() {  
 // *TODO: implement initState* super.initState();  
  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 body: Padding(  
 padding: const EdgeInsets.all(24),  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.start,  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 GestureDetector(  
 onTap:() {  
 BlocProvider.*of*<PlayMusicBloc>(context).add(EndKaraokeEvent());  
 },  
 child: const Text(  
 "Karaoke",  
 style: TextStyle(  
 color: Colors.*black*,  
 fontWeight: FontWeight.*bold*,  
 fontSize: 40  
 ),  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 12)),  
 Container(  
 padding: const EdgeInsets.symmetric(horizontal: 12),  
 decoration: BoxDecoration(  
 color: Colors.*grey*[200],  
 borderRadius: const BorderRadius.all(Radius.circular(50))  
 ),  
 height: 50,  
 child: TextField(  
 onSubmitted: (stringSearch){  
 showDialog(context: context, builder: (BuildContext context) => SearchResultDialog(\_textSearchController.text,"kara"));  
 },  
 textInputAction: TextInputAction.go,  
 controller: \_textSearchController,  
 decoration: const InputDecoration(  
 hintText: "Tìm kiếm",  
 border: InputBorder.*none*,  
 suffixIcon: Icon(Icons.*search*)  
 ),  
 ),  
 ),  
 const Padding(padding: EdgeInsets.only(top: 12)),  
 Expanded(  
 child: BlocBuilder<PlayMusicBloc,PlayMusicState>(  
 buildWhen: (prev,curr){  
 return curr is KaraokeState;  
 },  
 builder: (context,stateKaraoke){  
 if(stateKaraoke is KaraokeState){  
 return ScrollablePositionedList.builder(  
 itemScrollController: \_scrollController,  
 itemCount: stateKaraoke.sentences.length,  
 itemBuilder: (context, index) {  
 String sentence = "";  
 for(int i=0;i<stateKaraoke.sentences[index].words.length;++i){  
 sentence += "${stateKaraoke.sentences[index].words[i].data} ";  
 }  
 sentence = sentence.trim();  
 return BlocBuilder<PlayMusicBloc,PlayMusicState>(  
 buildWhen: (prev,curr){  
 return curr is KaraokeChangeLyricState;  
 },  
 builder: (context,stateLyricChange){  
 if(stateLyricChange is KaraokeChangeLyricState){  
 List<Word> words = stateKaraoke.sentences[index].words;  
 \_scrollController.scrollTo(index: stateLyricChange.indexScroll, duration: const Duration(milliseconds: 200));  
 return Container(  
 margin: const EdgeInsets.only(bottom: 8),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 Flexible(  
 child: Text(  
 sentence,  
 style: TextStyle(  
 color: (stateLyricChange.currDuration.inMilliseconds > words[0].startTime)?Colors.*red*:Colors.*grey*,  
 fontSize: (stateLyricChange.currDuration.inMilliseconds > words[0].startTime)?20:15,  
 fontWeight: FontWeight.*bold* ),  
 )  
 )  
 ],  
 ),  
 );  
 } else {  
 return Container();  
 }  
 }  
 );  
 },  
 );  
 } else {  
 return Container();  
 }  
 },  
 )  
 )  
 ],  
 ),  
 )  
 );  
 }  
}

**5.5 Lưu trạng thái đăng nhập sử dụng SharedPreferences:**

- Trong file main.dart. Sử dụng hàm main() để khởi tạo đối tượng SharedPreferences

void main() async {  
 WidgetsFlutterBinding.*ensureInitialized*();  
 shared = await SharedPreferences.*getInstance*();  
 service = MusicService(dio.Dio());  
 runApp((shared.getString("fullName")!=null && shared.getString("fullName")!="" )?const MainPageProvider():const MyApp());  
}

* Ở đoạn code trên, khi check key “fullName” mà bằng null hoặc là chuỗi rỗng thì lập tức chuyển sang màn hình login. Ngược lại sẽ vào trang chủ.
* Bởi vì nếu như user đã đăng nhập rồi thì chúng ta sẽ lưu tên của người dùng vào local storage nên chúng ta hoàn toàn có thể biết được rằng người dùng đã đăng nhập chưa.
* Dưới đây là đoạn code lưu giá trị của key “fullName” và “accessToken” trong hàm login() của trang đăng nhập:

shared.setString("accessToken", response.data!.accessToken);  
shared.setString("fullName", response.data!.fullName);  
Navigator.*push*(context, MaterialPageRoute(builder: (context)=> const MainPageProvider()));

**5.6 Phát nhạc thông qua URL sử dụng cấu trúc Bloc (Bussiness Logic Code):**

* Bloc trong Flutter hoạt động theo cơ chế: Khi người dùng tương tác với thành phần giao diện, chúng ta sẽ tiến hành phát sự kiện (event) vào bloc. Bloc sẽ xử lí và trả trạng thái (state) ra cho giao diện người dùng.
* Tạo 3 file đặt tên là **bloc.dart**, **event.dart** và **state.dart**. Sau đó paste toàn bộ code nhận event, xử lí logic và trả state về cho người dùng:
  + bloc.dart:
* import 'dart:io';  
    
  import 'package:ap\_03\_luongquocdien/network/entity/lyric/lyric\_entity.dart';  
  import 'package:ap\_03\_luongquocdien/network/route/music/music\_service.dart';  
  import 'package:bloc/bloc.dart';  
  import 'package:audioplayers/audioplayers.dart';  
  import 'package:dio/dio.dart' as dio;  
  import 'package:google\_speech/google\_speech.dart';  
  import 'package:record/record.dart';  
  import 'package:google\_speech/generated/google/cloud/speech/v1/cloud\_speech.pb.dart' hide RecognitionConfig;  
    
  import 'event.dart';  
  import 'state.dart';  
    
  class PlayMusicBloc extends Bloc<PlayMusicEvent, PlayMusicState> {  
    
   bool isPlaying = false;  
   AudioPlayer player = AudioPlayer();  
   late String id;  
   late String url;  
   late String quality;  
   late String title;  
   late String artist;  
   late MusicService service;  
   late Duration total;  
   late double volume;  
   late List<Sentence> lyrics;  
   late Duration curr;  
   late int timeTmp;  
   late int durationDiff;  
   late int indexScroll;  
   final record = Record();  
   final serviceAccount = ServiceAccount.fromFile(File('stt-flutter-6b7e79229bb6.json'));  
   var speechToText;  
   final config = RecognitionConfig(  
   encoding: AudioEncoding.FLAC,  
   model: RecognitionModel.basic,  
   enableAutomaticPunctuation: true,  
   sampleRateHertz: 44100,  
   enableWordTimeOffsets: true,  
   languageCode: 'vi-VN'  
   );  
    
   int indexSentence = 0, indexWord = 0;  
    
   Future<List<int>> \_getAudioContent(String name) async {  
   final path = 'D:\\AndroidProject\\ap\_03\_LuongQuocDien\\record\\$name';  
   return File(path).readAsBytesSync().toList();  
   }  
    
   void startRecord() async {  
   if (await record.hasPermission()) {  
   // Start recording  
   await record.start(  
   path: 'record/myFile.flac',  
   numChannels: 1,  
   encoder: AudioEncoder.flac, // by default  
   bitRate: 128000, // by default  
   samplingRate: 44100, // by default  
   );  
   }  
   }  
    
   PlayMusicBloc(PlayMusicState initial) : super(initial) {  
   on<InitEvent>((event,emit){  
   service = MusicService(dio.Dio());  
   volume = 1.0;  
   });  
    
   on<SeekEvent>((event,emit) async {  
   player.seek(event.duration);  
   add(ResumeSongEvent());  
   emit(SeekState(event.duration,total));  
   });  
    
   on<AudioPositionChangeEvent>((event,emit) async {  
   emit(AudioPositionChangeState(event.currPosition,total));  
   });  
    
   on<PlaySongEvent>((event,emit) async {  
   id = event.id;  
   title = event.title;  
   artist = event.artist;  
   total = event.total;  
   String quality = event.quality;  
   String linkThumb = event.linkThumb;  
   url = "http://api.mp3.zing.vn/api/streaming/audio/$id/$quality";  
   player.play(UrlSource(url));  
   player.setReleaseMode(ReleaseMode.loop);  
   emit(PlaySongState(id, quality, title, artist,linkThumb,total));  
   player.onPositionChanged.listen((newPosition) {  
   add(AudioPositionChangeEvent(newPosition));  
   });  
   });  
    
   on<ResumeSongEvent>((event,emit) async {  
   player.resume();  
   emit(ResumeSongState());  
   });  
    
   on<PauseSongEvent>((event,emit) async {  
   player.pause();  
   emit(PauseSongState());  
   });  
    
   on<DownloadSongEvent>((event,emit) async {  
   emit(DownloadSongState(false));  
   await dio.Dio().download(url,"download/$title.mp3");  
   emit(DownloadSongState(true));  
   });  
    
   on<ChangeVolumeEvent> ((event,emit) {  
   volume = event.newVolume;  
   player.setVolume(volume);  
   emit(ChangeVolumeState(event.newVolume));  
   });  
    
   on<KaraokeEvent> ((event,emit) async {  
   indexScroll = 0;  
   total = event.total;  
   final id = event.id;  
   quality = event.quality;  
   url = "http://api.mp3.zing.vn/api/streaming/audio/$id/$quality";  
   player.play(UrlSource(url));  
   final response = await service.getLyric(id);  
   lyrics = response.data;  
   emit(KaraokeState(event.title, event.artist, event.linkThumb, event.total, lyrics));  
   player.onPositionChanged.listen((newPosition) {  
   add(KaraokeChangeLyricEvent(newPosition));  
   });  
   startRecord();  
   });  
    
   on<EndKaraokeEvent> ((event,emit) async {  
   emit(KaraokePointingState());  
   speechToText = SpeechToText.viaServiceAccount(serviceAccount);  
   player.stop();  
   await record.stop();  
   final audio = await \_getAudioContent('myFile.flac');  
   RecognizeResponse response = await speechToText.recognize(config, audio);  
   print("get response done: $response}");  
   final List<WordInfo> wordUsers = response.results[0].alternatives.toList()[0].words;  
   int wordRightTotal = 0;  
   int total = 0;  
   bool isFound = false;  
   for(int i=0;i<lyrics.length;++i) total += lyrics[i].words.length;  
    
   for(int i=0;i<wordUsers.length;++i){  
   for(int j=indexSentence;j<lyrics.length;++j){  
   for(int k=indexWord;k<lyrics[j].words.length;++k){  
   double millisecondWord = int.*parse*(wordUsers[i].startTime.seconds.toString())\*1000 + int.*parse*(wordUsers[i].startTime.nanos.toString())/1000000;  
   if((lyrics[j].words[k].startTime - millisecondWord).abs() <= 200 && lyrics[j].words[k].data == wordUsers[i].word){  
   print("word right: ${wordUsers[i].word}, in time: ${lyrics[j].words[k].startTime}");  
   wordRightTotal = wordRightTotal + 1;  
   indexSentence = j+1;  
   indexWord = k+1;  
   isFound = true;  
   break;  
   }  
   }  
   if(isFound) {  
   isFound = false;  
   break;  
   }  
   }  
   }  
   print("total right: $wordRightTotal. total word: $total");  
   double point = (wordRightTotal/total)\*10;  
   emit(EndKaraokeState(point));  
   });  
    
   on<KaraokeChangeLyricEvent> ((event,emit) async {  
   for(int i=0;i<lyrics.length;++i){  
   if(event.currDuration.inMilliseconds <= lyrics[i].words[0].startTime) {  
   indexScroll = i-1;  
   break;  
   }  
   }  
   emit(KaraokeChangeLyricState(event.currDuration,(indexScroll==-1)?0:indexScroll));  
   });  
   }  
  }
  + state.dart:
* import 'package:ap\_03\_luongquocdien/bloc/play\_music/event.dart';  
  import 'package:ap\_03\_luongquocdien/network/entity/lyric/lyric\_entity.dart';  
    
  class PlayMusicState {}  
    
  class PlaySongState extends PlayMusicState{  
   final String id;  
   final String quality;  
   final String title;  
   final String artist;  
   final String linkThumb;  
   final Duration total;  
    
   PlaySongState(this.id, this.quality, this.title, this.artist,this.linkThumb,this.total);  
  }  
    
  class PauseSongState extends PlayMusicState{  
  }  
    
  class ResumeSongState extends PlayMusicState{  
  }  
    
  class SeekState extends PlayMusicState {  
   final Duration newDuration;  
   final Duration total;  
    
   SeekState(this.newDuration, this.total);  
  }  
    
  class AudioPositionChangeState extends PlayMusicState {  
   final Duration currPosition;  
   final Duration total;  
    
   AudioPositionChangeState(this.currPosition, this.total);  
  }  
    
  class DownloadSongState extends PlayMusicState {  
   final bool isDownloadedFinish;  
    
   DownloadSongState(this.isDownloadedFinish);  
  }  
    
  class ChangeVolumeState extends PlayMusicState {  
   final double volume;  
    
   ChangeVolumeState(this.volume);  
  }  
    
  class KaraokeState extends PlayMusicState {  
   final String title;  
   final String artist;  
   final String linkThumb;  
   final Duration total;  
   final List<Sentence> sentences;  
    
   KaraokeState(this.title, this.artist,this.linkThumb,this.total, this.sentences);  
  }  
    
  class KaraokeChangeLyricState extends PlayMusicState {  
   final Duration currDuration;  
   final int indexScroll;  
    
   KaraokeChangeLyricState(this.currDuration, this.indexScroll);  
  }  
    
  class EndKaraokeState extends PlayMusicState{  
   final double point;  
    
   EndKaraokeState(this.point);  
  }  
    
  class KaraokePointingState extends PlayMusicState {}
  + event.dart:
* abstract class PlayMusicEvent {}  
    
  class InitEvent extends PlayMusicEvent {}  
    
  class PlaySongEvent extends PlayMusicEvent {  
   final String id;  
   final String quality;  
   final String title;  
   final String artist;  
   final String linkThumb;  
   final Duration total;  
    
   PlaySongEvent(this.id, this.quality, this.title, this.artist,this.linkThumb,this.total);  
  }  
    
  class ResumeSongEvent extends PlayMusicEvent {}  
    
  class PauseSongEvent extends PlayMusicEvent {}  
    
  class SeekEvent extends PlayMusicEvent {  
   final Duration duration;  
    
   SeekEvent(this.duration);  
  }  
    
  class AudioPositionChangeEvent extends PlayMusicEvent {  
   final Duration currPosition;  
    
   AudioPositionChangeEvent(this.currPosition);  
  }  
    
  class DownloadSongEvent extends PlayMusicEvent {}  
    
  class ChangeVolumeEvent extends PlayMusicEvent {  
   final double newVolume;  
    
   ChangeVolumeEvent(this.newVolume);  
  }  
    
  class KaraokeEvent extends PlayMusicEvent {  
   final String id;  
   final String quality;  
   final String title;  
   final String artist;  
   final String linkThumb;  
   final Duration total;  
    
   KaraokeEvent(this.id, this.quality, this.title, this.artist,this.linkThumb,this.total);  
  }  
    
  class EndKaraokeEvent extends PlayMusicEvent {}  
    
  class KaraokeChangeLyricEvent extends PlayMusicEvent {  
   final Duration currDuration;  
    
   KaraokeChangeLyricEvent(this.currDuration);  
  }
* Tại hàm on<PlaySongEvent> chúng ta sẽ lắng nghe event chơi nhạc từ người dùng và tiến hành xử lí nó trong callBack và trả trạng thái về cho user thông qua hàm “emit(PlaySongState)”:

Text

Description automatically generated

**5.7 Download nhạc thông qua URL:**

**-** Chúng ta lắng nghe sự kiện download nhạc thông qua hàm on<DownloadSongEvent>. Tiến hành gọi vào url và download nhạc từ URL đó rồi lưu file vào thư mục “download” với tên bài hát với phần mở rộng là **mp3**. Cuối cùng, trả trạng thái về cho user thông qua hàm “emit(DownloadSongState)”.

Text

Description automatically generated

**5.8 Lấy lời bài hát, ghi âm, hiển thị và cuộn màn hình theo lời bài hát:**

- Khi karaoke. Chúng ta sẽ lắng nghe sự kiện karaoke từ người dùng trong hàm on<KaraokeEvent>. Tiến hành setup url, lấy lời bài hát thông qua API Zingmp3 và phát nhạc sau đó bắt đầu ghi âm trong hàm “startRecord()” trong hàm này chúng ta sẽ cấu hình các thông số cần thiết trước khi tiến hành thu âm như kênh, tốc độ bit, tốc độ lấy mẫu… (hình minh họa). Cuối cùng, trả trạng thái karaoke về cho người dùng. Trong trạng thái đó có những thông tin về lời bài hát để hiển thị ra cho người dùng.

Text

Description automatically generated

*Ảnh minh họa việc cấu hình thông số thu âm*

Text

Description automatically generated

*Ảnh minh họa xử lí.*

**5.9 Chấm điểm Karaoke:**

- Sau khi hát xong, chúng ta lắng nghe sự kiện hết nhạc trong hàm on<EndKaraokeEvent> và tiến hành xử lí như sau:

Text

Description automatically generated

*Ảnh minh họa*

* Hệ thống sẽ đọc file ghi âm và sử dụng thư viện google speech để tiến hành phân tích và chuyển đổi âm thanh thành text. Trong data mà thư viện trả về nó sẽ bao gồm có startTime, endTime và word. Nhưng chúng ta chỉ lấy 2 field là startTime và word để tiến hành chấm điểm.
* Tiêu chí cộng điểm:
  + Hát đúng lời.
  + Độ lớn chênh lệch giữa thời gian của lyric và của người dùng hát vào file âm thanh không lớn hơn 200ms.
* Sau khi mọi thứ hoàn tất. Tiến hành duyệt tuyến tính danh sách các từ trong lyrics của bài hát rồi tính điểm theo tiêu chí trên.
* Cuối cùng trả kết quả điểm về cho user thông qua hàm “emit(EndKaraokeState)”.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

## **Kết luận**

### Những kết quả đạt được

Đề tài đã:

+ Tìm hiểu về âm thanh học và kỹ thuật âm thanh

+ Thiết kế, lập trình và phát triển thành công ứng dụng nghe nhạc và giải trí. Thành công đáng chú ý là đã khai thác tốt tài nguyên nhạc từ Zingmp3.

* + - * Main\_page.dart
      * Home\_page.dart
      * New\_release\_page.dart
      * Login\_page.dart
      * Regist\_page.dart
      * Search\_result\_dialog.dart
      * Karaoke\_page.dart

+ Đóng gói bộ cài Setup bằng tool Inno Setup Compiler

### Hạn chế

Do thời gian hạn chế, nên App sản phẩm của đề tài còn một số hạn chế sau:

+ Ứng dụng chưa phát nhạc theo playlist.

## **Hướng phát triển**

### Hướng mở rộng đề tài

* Sẽ nghiên cứu và phát triển thêm một số tính năng như phát MV nhạc từ Zingmp3 trả về

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Võ Xuân Thể (2021), *Bài giảng học phần Lập trình âm thanh*, Học viện Công nghệ Bưu chính – Viễn thông, cơ sở TP.HCM (lưu hành nội bộ).
2. Rémi Audfray, Jean-Marc Jot, and Sam Dicker (Magic Leap, Inc), “*Audio Application Programming Interface for Mixed Reality*” in Audio Engineering Society 145th Convention, 2018 October 17 – 20, New York, NY, USA.
3. <http://www.ambiera.com/irrklang> : Thư viện irrKlang
4. <https://zingmp3.vn>: Tài nguyên Zingmp3
5. [https://www.openal.org](https://www.openal.org/) : thư viện OpenAL
6. [https://www.hackaudio.com](https://www.hackaudio.com/getting-started/introduction/): Audio Computer Programming = hướng dẫn lập trình âm thanh máy tính
7. [http://www.music-software-development.com](http://www.music-software-development.com/) : hướng dẫn phát triển các phần mềm âm nhạc
8. [http://www.aes.org](http://www.aes.org/) : Audio Engineering Society = Hội kỹ thuật âm thanh (e-library)
9. [https://www.freesoundeffects.com](https://www.freesoundeffects.com/): Tập âm thanh mẫu (sử dụng rất nhiều ứng dụng)
10. http://thammyamnhac.com (forum) diễn đàn về Âm thanh học / Kỹ thuật âm thanh.
11. https://viblo.asia : thư viện Tài liệu về Âm thanh học

1. <https://dotnet.microsoft.com/> và <https://dotnet.microsoft.com/apps/aspnet> : Công nghệ ASP.NET
2. [https://visualstudio.microsoft.com/downloads](https://visualstudio.microsoft.com/downloads/) : Các bộ công cụ hỗ trợ của Microsoft
3. [https://docs.microsoft.com/en-us](https://docs.microsoft.com/en-us/) : Các tài liệu hướng dẫn lập trình của Microsoft (tiếng Anh)