

ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

-----□★□-----



NHÓM 4

LÊ MINH PHƯỚC - 18521262

HUỖNH NGỌC QUÂN - 18521284

NGUYỄN LÊ MINH - 18521106

ĐẶNG VĂN NHÂN - 18521172

TRẦN NHẬT NAM - 18521132

ĐỒ ÁN MÔN HỌC

MÔN HỌC: INTERNET VÀ CÔNG NGHỆ WEB

LỚP: IE104.L12.CNCL

WEBSITE NGHE NHẠC TRỰC TUYẾN GOMUSIC

GIẢNG VIÊN HƯỚNG DẪN:

ThS VÕ TẤN KHOA

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

Lời cam đoan

Nhóm chúng tôi cam đoan rằng bài báo cáo đồ án “*Website nghe nhạc trực tuyến GoMusic*” là kết quả tìm hiểu và nghiên cứu của cả nhóm, không sao chép từ bất cứ nguồn nào khác.

Bài báo cáo có sử dụng thông tin trích dẫn và tham khảo từ nhiều nguồn khác nhau, các thông tin tìm được trích dẫn đều ghi rõ nguồn gốc.

TPHCM, ngày 25 tháng 12 năm 2020

Nhóm trưởng

Huỳnh Ngọc Quân

Lời cảm ơn

Lời đầu tiên nhóm chúng tôi xin chân thành cảm ơn ThS. Võ Tấn Khoa đã tận tình hướng dẫn nhóm chúng tôi trong suốt thời gian thực hiện đồ án.

Nhóm chúng tôi bày tỏ lòng biết ơn đến Thầy và các bạn trong lớp học đã tạo điều kiện tốt nhất để nhóm chúng tôi hoàn thiện đồ án này.

Trong thời gian thực hiện đồ án này, nhóm chúng tôi đã sử dụng các kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó vận dụng tối đa những gì đã học được để hoàn thành báo cáo đồ án một cách tốt nhất. Tuy nhiên trong quá trình thực hiện, nhóm chúng tôi không thể tránh khỏi những thiếu sót. Chính vì vậy, nhóm chúng tôi rất mong nhận được sự góp ý chân thành từ thầy và các bạn nhằm hoàn thiện những kiến thức mà nhóm chúng tôi đã học tập và là hành trang để nhóm thực hiện các đồ án khác trong tương lai.

Xin chân thành cảm ơn Thầy và các bạn.

MỤC LỤC

| | |
|---|-----------|
| Lời cam đoan | 2 |
| Lời cảm ơn | 3 |
| MỤC LỤC | 4 |
| Danh mục các hình ảnh | 6 |
| Danh mục các chữ viết tắt | 7 |
| Danh mục các bảng | 8 |
| MỞ ĐẦU | 9 |
| CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI..... | 10 |
| 1.1 Lý do chọn đề tài..... | 10 |
| 1.2 Đối tượng nghiên cứu..... | 10 |
| 1.3 Mục tiêu đề tài..... | 10 |
| 1.4 Phạm vi đề tài | 10 |
| CHƯƠNG 2. CƠ SỞ LÝ THUYẾT | 11 |
| 2.1 Khảo sát thực tế..... | 11 |
| 2.2 Mô hình MVC (Model-View-Control)..... | 17 |
| 2.3 API (Application programming interface- Giao diện lập trình ứng dụng): | 18 |
| 2.4 JWT(JSON WEB TOKEN) | 19 |
| 2.5 Công nghệ ReactJS..... | 22 |
| 2.5.1 Giới thiệu..... | 22 |
| 2.5.2 Component | 22 |
| 2.5.3 DOM (Document Object Model)..... | 22 |
| 2.5.4 Virtual-DOM..... | 23 |
| 2.5.5 JSX..... | 23 |
| 2.5.6 ES6..... | 23 |
| 2.5.7 Redux..... | 24 |
| 2.5.8 Routing | 24 |
| 2.6 Công nghệ Backend | 24 |
| 2.6.1 NodeJS..... | 24 |
| 2.6.2 Express | 25 |
| 2.6.3 Mongoose | 27 |
| 2.6.4 Multer | 27 |
| 2.6.5 Body-parser | 27 |
| 2.6.6 EJS..... | 27 |
| 2.6.7 Nodemon | 27 |
| 2.7 Cơ sở dữ liệu MongoDB | 27 |
| 2.7.1 Tổng quan về MongoDB | 27 |
| 2.7.2 Lý do chọn MongoDB làm CSDL..... | 28 |

| | |
|--|-------------------------------------|
| CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG..... | 29 |
| 3.1 Phân tích yêu cầu hệ thống..... | 29 |
| 3.1.1 Yêu cầu chức năng | 29 |
| 3.1.2 Xác định các đối tượng | 29 |
| 3.2 Phân tích thiết kế | 30 |
| 3.2.1 Sơ đồ hoạt động..... | Error! Bookmark not defined. |
| 3.2.2 Thiết kế hệ thống..... | 30 |
| 3.2.3 Mô tả web server | 31 |
| 3.2.4 Mô tả web client..... | 31 |
| 3.2.5 Thiết kế dữ liệu..... | 32 |
| CHƯƠNG 4. PHÁT TRIỂN TRANG WEB | 35 |
| 4.1 Các giai đoạn phát triển trang Web..... | 35 |
| 4.2 Kiểm tra ứng dụng..... | 36 |
| 4.3 Kết quả đạt được | 37 |
| 4.4 Hạn chế | 37 |
| CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | 38 |
| 5.1 Kết luận..... | 38 |
| 5.2 Kết quả đạt được | 38 |
| 5.3 Hướng phát triển..... | 39 |
| TÀI LIỆU THAM KHẢO | 40 |
| BẢNG PHÂN CHIA CÔNG VIỆC..... | 41 |

Danh mục các hình ảnh

| | |
|--|----|
| HÌNH 2.1.1 - Tỷ lệ giới tính tham gia khảo sát. | 11 |
| HÌNH 2.1.2 - Độ tuổi tham gia khảo sát. | 11 |
| HÌNH 2.1.3 - Nghề nghiệp của người tham gia khảo sát..... | 12 |
| HÌNH 2.1.4 - Tỷ lệ người thường nghe nhạc. | 12 |
| HÌNH 2.1.5 - Trang web nhạc mọi người thường nghe..... | 13 |
| HÌNH 2.1.6 - Tỷ lệ người “có” và “không” tải nhạc. | 13 |
| HÌNH 2.1.7 - Các tính năng được người dùng yêu thích..... | 14 |
| HÌNH 2.1.8 - Tính năng gây khó chịu..... | 14 |
| HÌNH 2.1.9 - Đề xuất bổ sung tính năng của người dùng. | 15 |
| HÌNH 2.1.10 - Giao diện trang web nhạc ưa thích. | 15 |
| HÌNH 2.1.11 - Giao diện website nghe nhạc hiện nay được ưa thích. | 16 |
| HÌNH 2.2.1 - Mô hình MVC..... | 17 |
| HÌNH 2.4.1 - Cấu thành của một JWT..... | 20 |
| HÌNH 2.4.2 - Luồng xử lý của JWT..... | 21 |
| HÌNH 2.6.1 - Mô hình Middleware trong expressjs. | 25 |
| HÌNH 3.2.1 - Mô hình client-server trong website âm nhạc. | 30 |
| HÌNH 3.2.2 - Mô hình mô tả hoạt động của backend server. | 31 |
| HÌNH 3.2.3 - Mô hình mô tả hoạt động web client. | 31 |
| HÌNH 3.2.4 - Mô hình database nhóm thiết kế cho website..... | 32 |

Danh mục các chữ viết tắt

| Chữ viết tắt | Ý nghĩa tiếng Việt |
|---------------------|------------------------------------|
| DOM | Document Object Model |
| API | Application Programming Interface |
| HTTP | HyperText Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| MV | Music video |
| UI | User Interface |

Danh mục các bảng

| | |
|---|----|
| BẢNG 3.2.1 – Mô tả thuộc tính của người dùng..... | 33 |
| BẢNG 3.2.2 – Mô tả thuộc tính của ca sĩ. | 33 |
| BẢNG 3.2.3 – Mô tả thuộc tính của bài hát..... | 33 |

MỞ ĐẦU

Trong thời buổi hiện đại ngày nay, nhu cầu giải trí của con người cũng tăng theo sự phát triển của xã hội. Chúng ta có thể giải trí bằng nhiều cách khác nhau như: chơi game, đi du lịch, đi ăn uống,... Một cách giải trí đơn giản và hiệu quả hơn hết là nghe một bài nhạc phù hợp với tâm trạng. Việc nghe nhạc đã trở thành thú vui, cách giảm stress hiệu quả được nhiều người ở mọi lứa tuổi lựa chọn. Từ đó các website âm nhạc ra đời để đáp án nhu cầu đó.

Với yêu cầu đồ án môn học Internet và công nghệ Web là xây dựng nên một website có tính ứng dụng cao và chạy được nó. Nhóm chúng tôi đã không ngần ngại chọn đề tài “*Website âm nhạc*” để thực hiện. Xét về phương diện kỹ thuật và kiến thức lý thuyết để đáp ứng cho đồ án này là phù hợp với chúng tôi. Website âm nhạc sẽ là ứng dụng mà trong đó nhóm chúng tôi kết hợp những thứ từ frontend (ReactJS, HTML, CSS, Javsscrip), backend (NodeJS) đến Database (MongoDB) vào trong nó.

Chúng tôi kỳ vọng rằng qua đồ án này nhóm sẽ có được thật nhiều kiến thức nhất có thể về lĩnh vực Web để có thể áp dụng cho công việc cũng như những môn học phía sau. Phần chi tiết của đồ án sẽ được trình bày ở bên dưới.

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

Nhóm chúng tôi qua quá trình tự tìm hiểu và trong thời gian học môn **Internet và công nghệ Web** đã tích lũy được những kiến thức cơ bản để có thể hoàn thiện được một trang Web. Qua khảo sát và các yêu cầu thực tế về chức năng của một trang Web mà khả năng nhóm chúng tôi có thể thực hiện được. Nhóm chúng tôi quyết định chọn đề tài tạo dựng một Website nghe nhạc trực tuyến để có thể áp dụng tối đa được những kiến thức đã tích lũy vào đồ án môn học này. Mặt khác nhóm cũng nhìn nhận được hướng phát triển của trang web này về sau cũng góp phần không nhỏ để tiến đến việc chọn đề tài này.

1.2 Đối tượng nghiên cứu

Đối tượng nghiên cứu chủ yếu của đề tài xoay quanh các chức năng của trang web nghe nhạc trực tuyến. Nghiên cứu về các yêu cầu của một website nghe nhạc trực tuyến mà người dùng cần. Phân tích các yêu cầu của hệ thống nghe nhạc trực tuyến, xây dựng các chức năng thông qua các ngôn ngữ lập trình NodeJS, ứng dụng ReactJS trong việc xây dựng giao diện người dùng, ngoài ra còn có cách sử dụng database MongoDB trong một ứng dụng thực tế.

1.3 Mục tiêu đề tài

Thiết kế, xây dựng được một website nghe nhạc trực tuyến với các chức năng cơ bản, hoàn chỉnh. Nắm được quy trình hoạt động từ Frontend - Backend - Database của một trang web. Áp dụng được các công nghệ phổ biến với một lập trình viên web như ReactJS, NodeJS, MongoDB. Đề xuất được hướng phát triển của trang web và có thể đưa vào thực tế.

1.4 Phạm vi đề tài

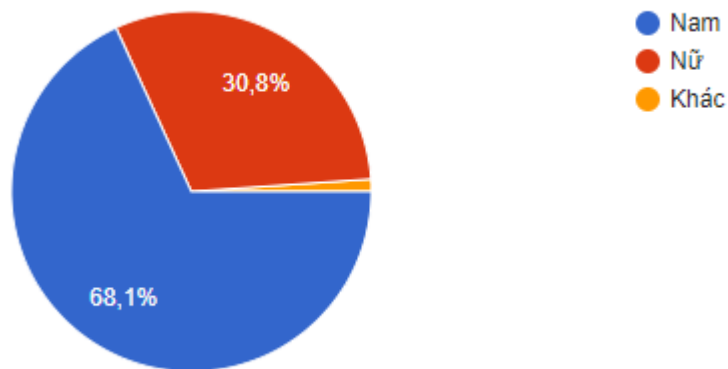
Trong đề tài này nhóm chúng tôi giới hạn một số lượng tín năng nhất định của Website âm nhạc mà trong khả năng của nhóm có thể thực hiện được. Về quy trình hoạt động chỉ nắm được phần cơ bản để có thể chạy được trang web không lỗi. Với các công nghệ áp dụng trong đề tài thì nhóm thực hiện theo phương pháp “*không biết cái nào thì tìm cái đó*”, không thức hiện tìm hiểu hoàn chỉnh từ đầu đến cuối.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Khảo sát thực tế

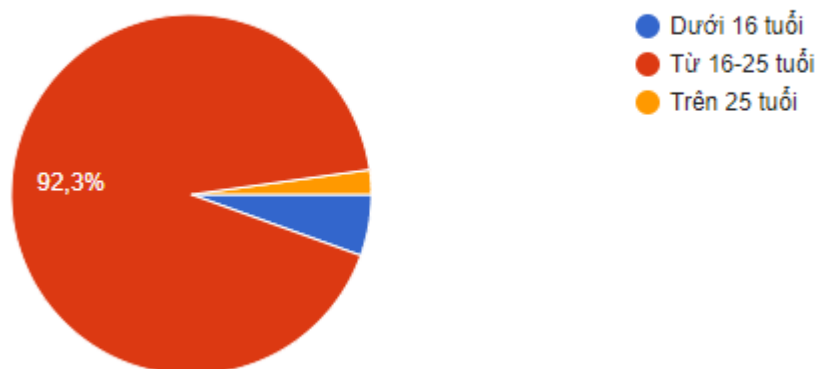
Nhóm tiến hành khảo sát nhu cầu nghe nhạc, và giao diện mọi người yêu thích, để từ đó xây dựng trang web phù hợp. Có tất cả 91 người tham gia khảo sát. Kết quả được thống kê như sau:

Tỉ lệ giới tính:

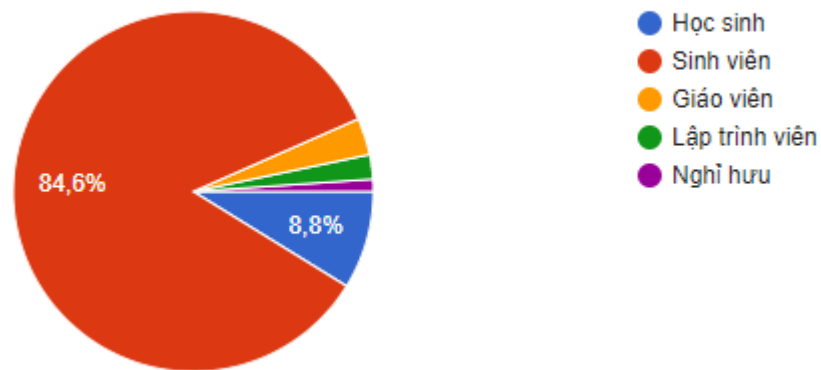


HÌNH 2.1.1 - Tỉ lệ giới tính tham gia khảo sát.

Hầu hết người tham gia khảo sát có độ tuổi từ 16-25 với phần lớn là sinh viên:

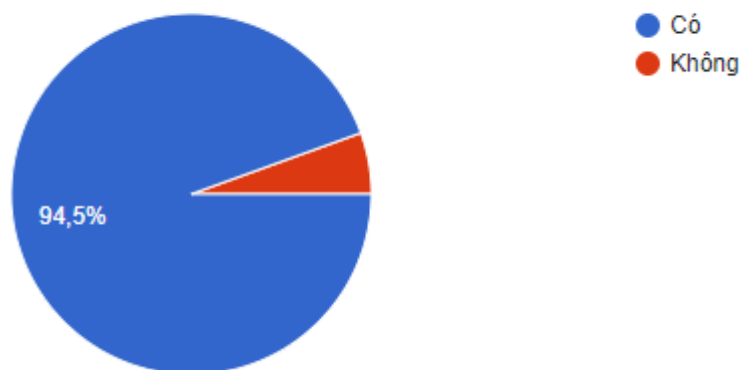


HÌNH 2.1.2 - Độ tuổi tham gia khảo sát.



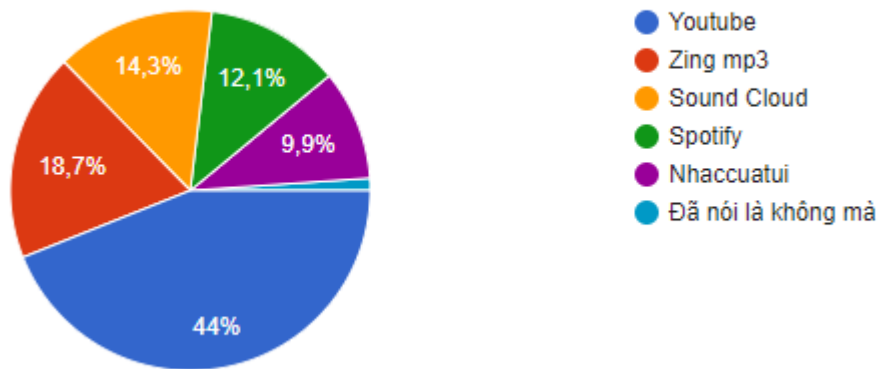
HÌNH 2.1.3 - Nghề nghiệp của người tham gia khảo sát.

Âm nhạc đáp ứng nhu cầu giải trí của con người, cộng với chi phí rẻ nên số lượng người thường nghe nhạc chiếm 94,5 % không quá bất ngờ:



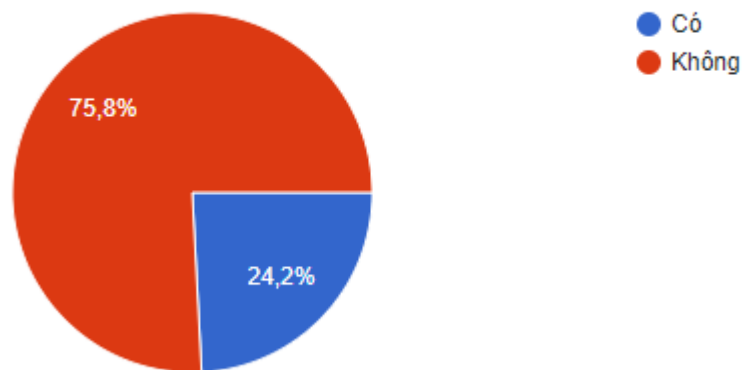
HÌNH 2.1.4 - Tỷ lệ người thường nghe nhạc.

Có đa dạng trang web nghe nhạc chất lượng, cạnh tranh. Nên tùy mỗi người mà thích một trang web khác nhau. Nhưng đa số tập trung vào 5 trang web:



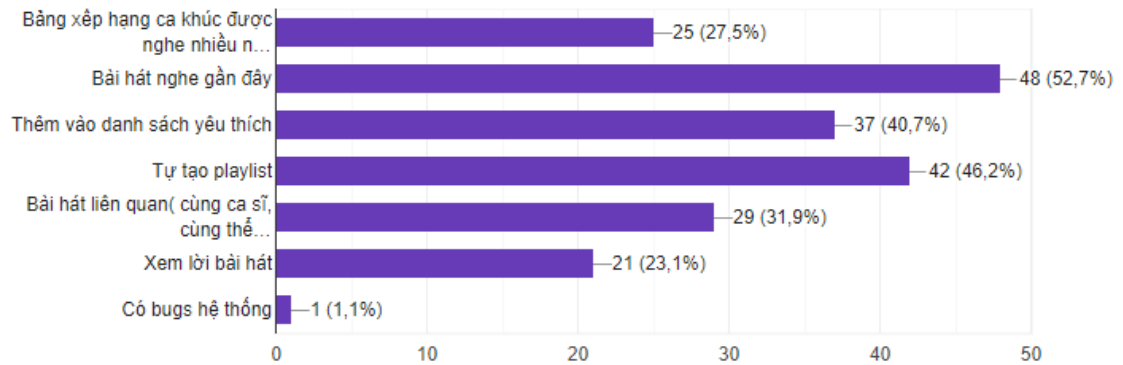
HÌNH 2.1.5 - Trang web nhạc mọi người thường nghe.

Một số tải nhạc về thiết bị cá nhân để tiện nghe nhạc khi không có mạng:

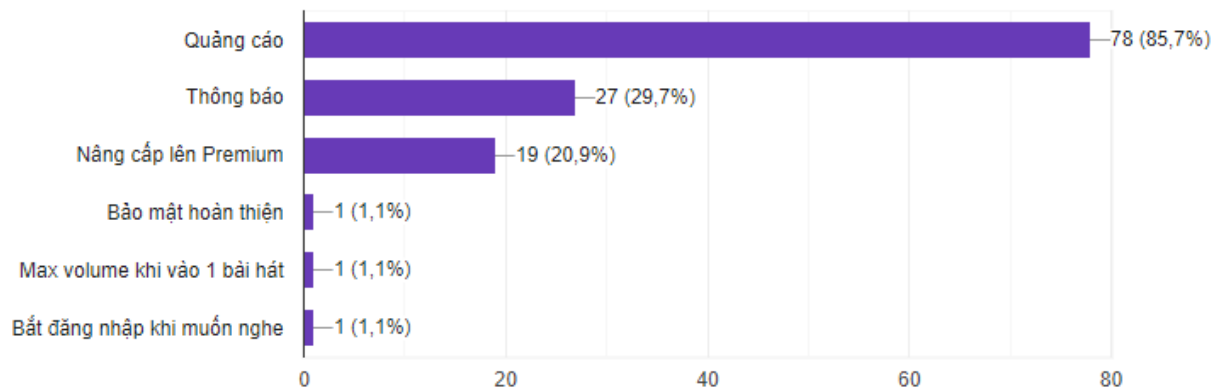


HÌNH 2.1.6 - Tỷ lệ người “có” và “không” tải nhạc.

Để đem đến trải nghiệm tuyệt vời nhất cho người dùng, nhóm muốn khảo sát về các tính năng yêu thích và các tính năng gây khó chịu trên các website nghe nhạc:

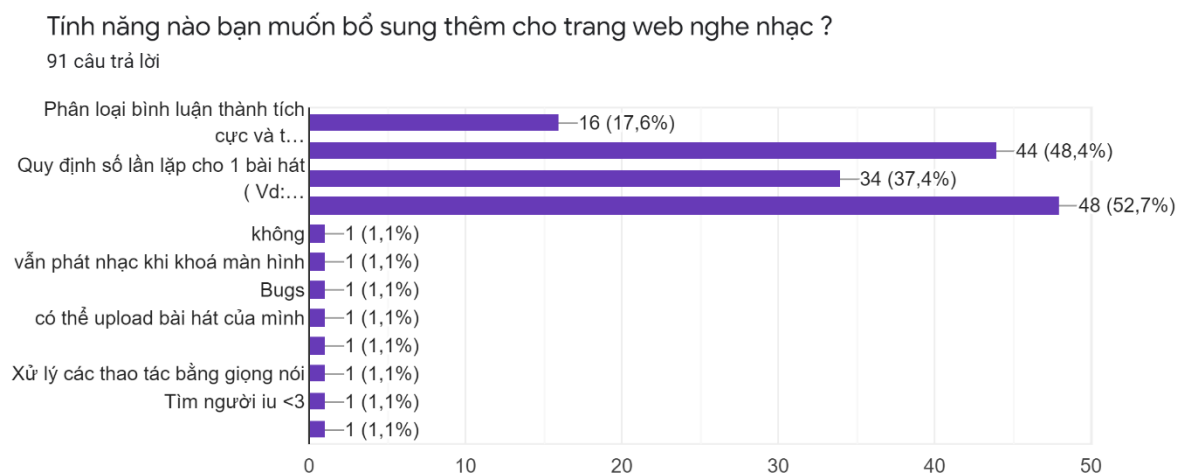


HÌNH 2.1.7 - Các tính năng được người dùng yêu thích.



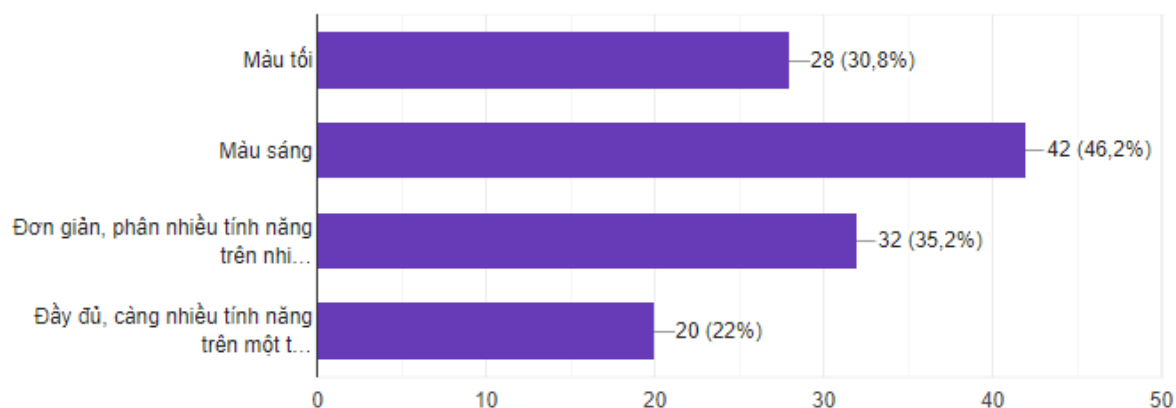
HÌNH 2.1.8 - Tính năng gây khó chịu.

Ngoài các tính năng hiện có của các trang web, nhóm cũng muốn khảo sát các tính năng mà người dùng muốn bổ sung thêm, để tạo ra trang web nghe nhạc tốt nhất:



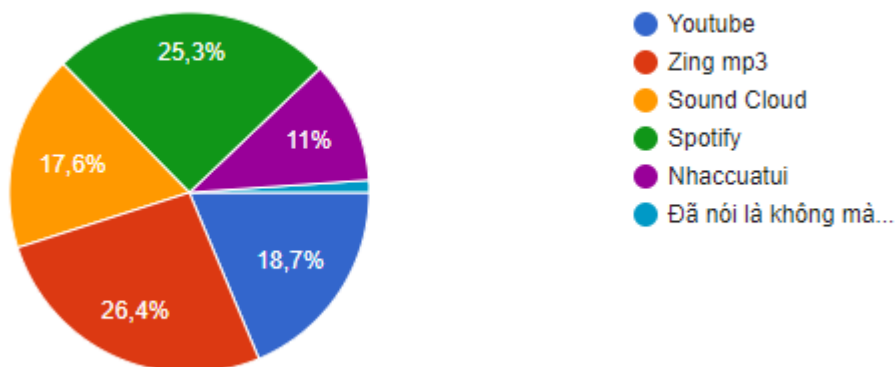
HÌNH 2.1.9 - Đề xuất bổ sung tính năng của người dùng.

Nhìn chung, mỗi người có một cái nhìn khác nhau, nên giao diện trang web mà họ thích cũng khác nhau, không thể đáp ứng cùng lúc nhiều giao diện, nhưng phân đông thích giao diện sáng, đơn giản, các tính năng trên các trang khác nhau:



HÌNH 2.1.10 - Giao diện trang web nhạc ưa thích.

Số người nghe nhạc trên youtube nhiều, nhưng giao diện website âm nhạc gây ấn tượng với họ hơn là Zing MP3 và Spotify:



HÌNH 2.1.11 - Giao diện website nghe nhạc hiện nay được ưa thích.

Từ form khảo sát trên nhóm đúc kết ra giao diện và tính năng cần có của trang web nghe nhạc mà nhóm thiết kế :

Về giao diện:

- Ưu tiên màu sáng, đơn giản, các tính năng phân trên các trang khác nhau.
- Phát triển cả giao diện ở chế độ sáng và tối với nhiều loại hình nền khác nhau để người dùng có thể thoải mái lựa chọn.
- Chọn các hình ảnh chất lượng cao, đẹp để kích thích người nghe.
- Sử dụng hiệu ứng đơn giản, hiệu quả như thanh trượt, xoay tua như các trang web Zing MP3, Spotify.

Về tính năng:

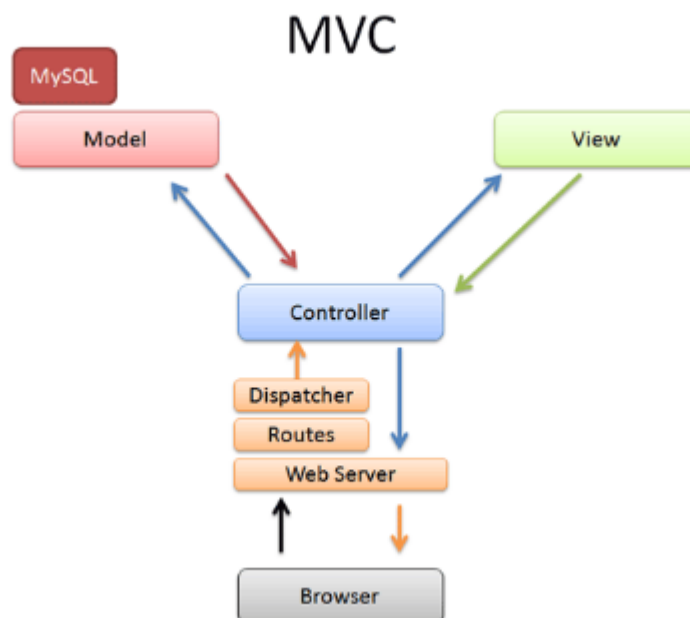
- Đăng ký, đăng nhập, đăng xuất là tính năng cơ bản cần có.
- Tìm kiếm để giúp người dùng dễ tìm ra bài hát mình mong muốn.
- Người dùng có thể tự tạo playlist ưa thích.
- Tính năng hiển thị những bài hát nghe gần nhất của người dùng.
- Thêm vào danh sách yêu thích.
- Đề cử bài hát liên quan.
- Bảng xếp hạng bài hát được nghe nhiều nhất.
- Bình luận.
- Tính năng phân loại, đếm số lượng bình luận tích cực, tiêu cực của bài hát.

- Quy định số lần lặp cho một bài hát.
- Các bài hát dislike sẽ không xuất hiện nữa.

Trong thời gian đầu phát triển trang web, nhóm sẽ tránh việc post quảng cáo, không thông báo những thông tin không cần thiết. Và cho người dùng nghe miễn phí để tăng lượng người nghe trước. Về sau sẽ có những chiến lược lâu dài.

2.2 Mô hình MVC (Model-View-Control)

Trong quá trình xây dựng – phát triển – vận hành và bảo trì một hệ thống hay một ứng dụng – phần mềm các chuyên gia sẽ sử dụng những mô hình nghiệp vụ chung, chúng được tạo nên từ nhiều thành phần độc lập nhau nhưng sẽ có sự tương tác nhau. Những mô hình này, sẽ giúp người sử dụng (manager, user, sponsor, developer,...) có một cái nhìn tổng quan nhất về chức năng và các luồng xử lý có trong dự án, từ đó họ có một điểm chung để trao đổi đưa ra các yêu cầu, dễ dàng tiếp cận – tìm hiểu các chức năng nghiệp vụ đang có và tiến hành xây dựng và phát triển dự án một cách nhanh chóng và hiệu quả nhất. Một mô hình có thể phi chuẩn hoặc hợp chuẩn, nó sẽ có những đặc thù riêng và chỉ áp dụng cho những loại dự án cụ thể, từ đó các chuyên gia sẽ lựa chọn mô hình nào phù hợp.



HÌNH 2.2.1 - Mô hình MVC.

Hiện nay, trong các dự án website các chuyên gia thường áp dụng mô hình chuẩn Model – View – Controller (hay MVC) trong quá trình xây dựng – phát triển, nó đã khá quen thuộc và phổ biến với dân lập trình. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình chia source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

Model:

- Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thao tác với cơ sở dữ liệu như xem, truy xuất, xử lý dữ liệu,...

View:

- Là nơi chứa những giao diện như nút bấm, khung nhập, menu, hình ảnh... nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.

Controller:

- Là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.

Để dễ dàng quản lý các file dữ liệu, đồ án của nhóm xây dựng theo mô hình MVC.

2.3 API (Application programming interface- Giao diện lập trình ứng dụng):

Là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. API cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Để từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

Web API thường được thường được sử dụng trong các hệ thống website. Hầu hết các website đều ứng dụng đến web API cho phép kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu.

Tính năng nổi bật của web API:

- Tự động hóa sản phẩm: Web API giúp tự động hóa quản lý công việc, cập nhật luồng công việc, giúp tăng năng suất và tạo hiệu quả công việc cao hơn.

- Khả năng tích hợp linh động: API cho phép lấy nội dung từ bất kỳ website hoặc ứng dụng nào một cách dễ dàng nếu được cho phép, tăng trải nghiệm người dùng. API hoạt động như một chiếc công, cho phép các công ty chia sẻ thông tin được chọn nhưng vẫn tránh được những yêu cầu không mong muốn.
- Cập nhật thông tin thời gian thực
- Có tiêu chuẩn chung, dễ sử dụng: Bất kỳ người dùng, công ty nào sử dụng cũng có thể điều chỉnh nội dung, dịch vụ mà họ sử dụng. Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

Cách hoạt động của web API:

1. Xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.
2. Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.
3. Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.
4. Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động, dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu...

Trong đồ án này, các API đều do nhóm tự viết.

2.4 JWT(JSON WEB TOKEN)

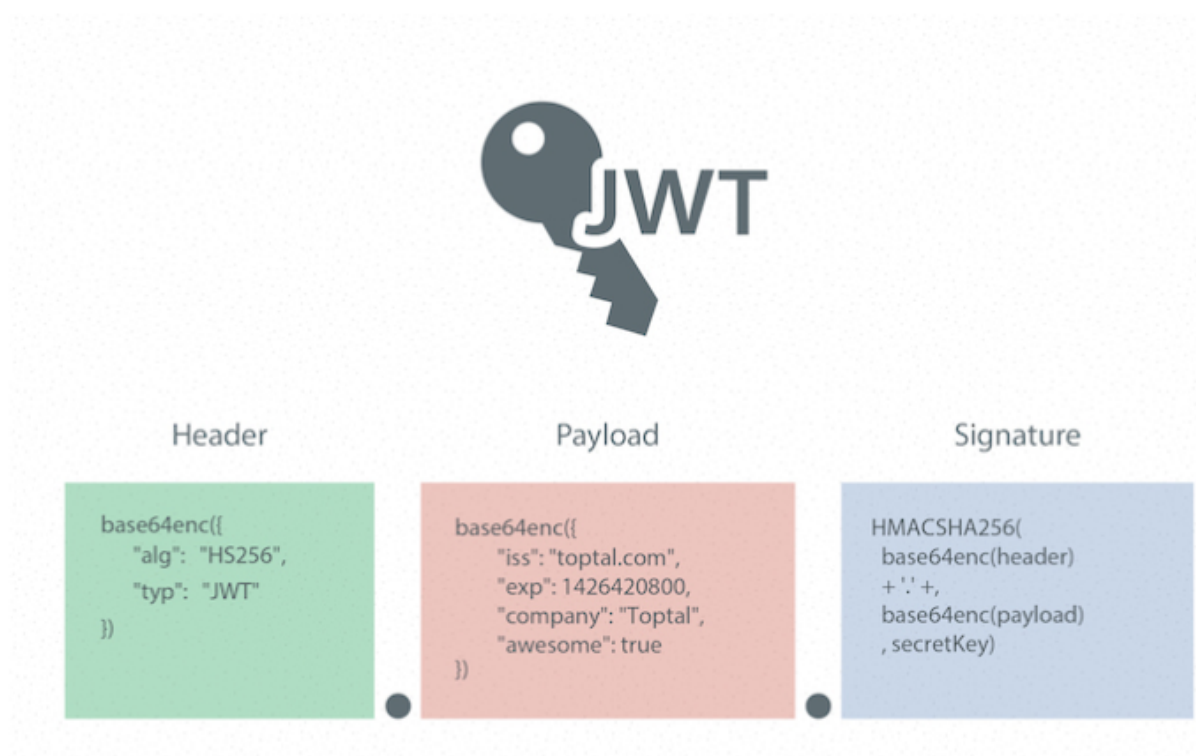
Trong đồ án này có xây dựng tính năng là đăng nhập, đăng xuất. Do đó cần có một phương pháp để xác thực người dùng đã đăng ký chưa, có quyền vào trang chủ của Gomusic hay không. JSON Web Token là công cụ tuyệt vời để giải quyết vấn đề trên.

Token-based authentication là phương thức xác thực bằng chuỗi mã hóa. Một hệ thống sử dụng Token-based authentication cho phép người dùng nhập user/password (hoặc tương tự) để nhận về 1 chuỗi mã token. Mã này được sử dụng để "xác minh" quyền truy cập vào tài nguyên mà không cần phải cung cấp lại username/password nữa.

Trong đề án này, chúng tôi sử dụng Token-based authentication bằng JWT.

JSON Web Token (JWT) là 1 tiêu chuẩn mở (RFC 7519) định nghĩa cách thức truyền tin an toàn giữa các thành viên bằng 1 đối tượng JSON. Thông tin này có thể được xác thực và đánh dấu tin cậy nhờ vào "chữ ký" của nó. Phần chữ ký của JWT sẽ được mã hóa lại bằng HMAC hoặc RSA. Bảo mật JWT là phương pháp xác thực quyền truy cập (Authentication) bằng JSON Web Token.

Trong đó chuỗi Token phải có 3 phần là header , phần payload và phần signature được ngăn bằng dấu “.”



HÌNH 2.4.1 - Cấu thành của một JWT

Header:

- Header bao gồm hai phần chính:
 - typ - Loại token (mặc định là JWT - Thông tin này cho biết đây là một Token JWT).
 - alg - Thuật toán đã dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA).

Payload:

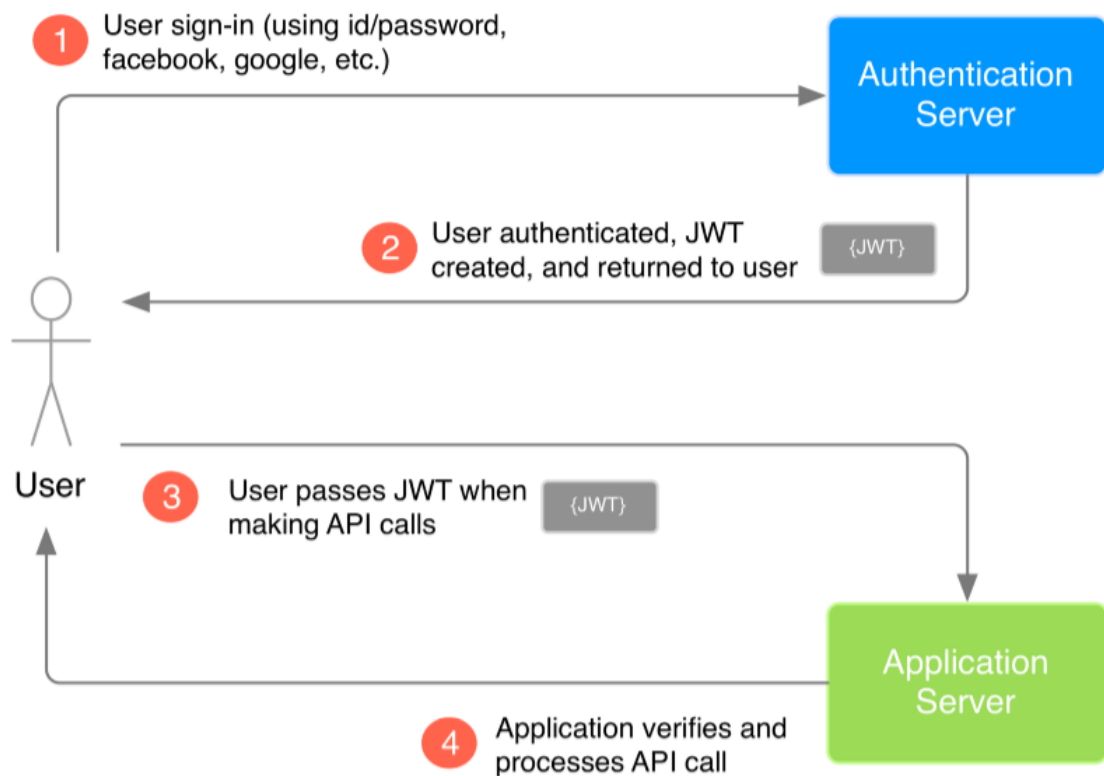
- Payload là nơi chứa các nội dung của thông tin (claim). Thông tin truyền đi có thể là mô tả của 1 thực thể hoặc cũng có thể là các thông tin bổ sung

thêm cho phần Header. Nhìn chung, chúng được chia làm 3 loại: reserved, public và private.

Signed:

- Phần chữ ký được tạo bằng cách kết hợp 2 phần Header + Payload, rồi mã hóa nó lại bằng 1 giải thuật encode nào đó, càng phức tạp thì càng tốt, ví dụ như HMAC SHA-256.

Luồng xử lý:



HÌNH 2.4.2 - Luồng xử lý của JWT.

1. User thực hiện login bằng cách gửi id/password hay sử dụng các tài khoản mạng xã hội lên phía Authentication Server (Server xác thực)
2. Authentication Server tiếp nhận các dữ liệu mà User gửi lên để phục vụ cho việc xác thực người dùng. Trong trường hợp thành công, Authentication Server sẽ tạo một JWT và trả về cho người dùng thông qua response.
3. Người dùng nhận được JWT do Authentication Server vừa mới trả về làm "chìa khóa" để thực hiện các "lệnh" tiếp theo đối với Application Server.

4. Application Server trước khi thực hiện yêu cầu được gọi từ phía User, sẽ verify JWT gửi lên. Nếu thành công, tiếp tục thực hiện yêu cầu được gọi.

2.5 Công nghệ ReactJS

2.5.1 Giới thiệu

React JS là thư viện Javascript có mã nguồn mở không phải là Framework, được xây dựng bởi các kỹ sư của Facebook. Những năm gần đây đang nổi lên với xu hướng Single Page Application, dùng để xây dựng User Interface – UI (giao diện người dùng) có thể tái sử dụng. Hiện nay đang được rất nhiều công ty nổi tiếng sử dụng để phát triển sản phẩm của họ như Facebook, Instagram, Spotify, Dropbox, Bitbucket,...

2.5.2 Component

React JS được xây dựng với trọng tâm là các component, chứ không dùng Template như các framework khác.

Một component gồm có 3 phần: import, define, export. Khi sử dụng một component cần phải import nó vào với tên đã định nghĩa trước đó. Các component là các thành phần độc lập và có thể tái sử dụng.

2.5.3 DOM (Document Object Model)

Document Object Model thường được gọi là DOM, là một phần quan trọng trong việc làm các website tương tác. Nó như là một giao diện cho phép ngôn ngữ lập trình thao tác nội dung, cấu trúc và định dạng của một website. JavaScript là một ngôn ngữ kịch bản phía client có thể kết nối với DOM trong trình duyệt.

Ở một cấp độ cơ bản nhất, một website tồn tại một tài liệu HTML. Trình duyệt mà bạn sử dụng để xem các website là một chương trình phiên dịch HTML và CSS và tạo ra các định dạng, kiểu dáng, nội dung và cấu trúc bên trong các trang mà bạn nhìn.

Ngoài việc phân tích cú pháp style và cấu trúc của HTML và CSS, trình duyệt tạo ra một đại diện của một tài liệu được biết là Document Object Model. Mô hình này cho phép JavaScript truy cập, thao tác các văn bản nội dung, các phần tử của tài liệu website dưới dạng một Object.

2.5.4 Virtual-DOM

Đối với React JS thì Virtual DOM vừa đóng vai trò Model vừa đóng vai trò View, khi Virtual-DOM thay đổi, ta không cần thao tác trực tiếp với DOM trên View mà vẫn phản ánh được sự thay đổi đó. Do đó nên mọi sự thay đổi trên Model đã kéo theo sự thay đổi trên View và ngược lại điều này làm cho tốc độ View dùng tăng lên đáng kể.

2.5.5 JSX

Là một phần quan trọng trong React, giống như một syntax extension giúp code trở nên mượt, dễ đọc và dễ hiểu hơn rất nhiều.

JSX là tên viết tắt của Javascript XML. Nó transform cú pháp dạng gần như XML về thành Javascript. Giúp người lập trình có thể code ReactJS bằng cú pháp của XML thay vì sử dụng Javascript. Các XML elements, attributes và children được chuyển đổi thành các đối số truyền vào React.createElement.

2.5.6 ES6

Là phiên bản mới nhất của Javascript, vì còn quá mới nên chưa được hỗ trợ trên các trình duyệt, nhưng các công cụ đóng gói sẽ thực hiện phiên dịch với cấu hình đúng để có thể sử dụng được trên trình duyệt.

```
# Viết bằng Javascript
function XeHoi(ten, gia){
    this.name = ten;
    this.price = gia;
}
Const xe1 = new XeHoi('BMW' , '1000$')
Const xe2 = new XeHoi('FORD' , '1000$')

# Viết bằng ES6
class XeHoi {
    constructor(name, price){
        this.name = name;
        this.price = price;
    }
}
Mycar = new XeHoi('MEC' , '2000$')
```

2.5.7 Redux

Là một thư viện Javascript giúp tạo ra một lớp quản lý trạng thái của ứng dụng. Được dựa trên nền tảng của kiến trúc Flux do Facebook giới thiệu. Vì vậy, React và Redux như anh em chung một mẹ, là sự kết hợp, hỗ trợ nhau một cách hoàn hảo. Muốn đi đến đỉnh cao React cần phải có Redux. Ngược lại, phải có React thì Redux mới thể hiện hết khả năng. Hầu hết các React developer đều biết Redux.

2.5.8 Routing

“Các ứng dụng single-page” trở nên khá rầm rộ ngày nay. Đây là những trang web mà chỉ thực hiện khởi tạo một lần, và khi người dùng ấn vào một đường dẫn hay nút ấn thì Javascript trên trang web sẽ thực thi mà trang web không cần phải reload lại. Việc quản lý địa chỉ trên thanh địa chỉ được thực hiện bởi router.

Router được sử dụng phổ biến nhất trong hệ sinh thái React là react-router. Nếu như bạn đang xây dựng một ứng dụng single-page, hãy sử dụng nó trừ khi bạn có lý do mà không thể sử dụng.

2.6 Công nghệ Backend

2.6.1 NodeJS

Nhóm 4 chọn NodeJS để viết phần Backend cho đồ án website lần này. NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.

Lý do chọn NodeJS để xây dựng phần Backend:

- Sau khi sử dụng cân nhắc và tìm hiểu một số công nghệ khác như Spring Boot (Java), Django (Python), ... thì NodeJS lợi thế hơn khi đã biết về JavaScript.
- Nhóm chọn ReactJS được viết bằng JavaScript để xây dựng Frontend. Vì vậy sử dụng NodeJS sẽ dễ dàng đồng bộ hơn và cộng đồng người dùng cũng rất lớn.
- Tốc độ xử lý nhanh nhờ cơ chế bất đồng bộ (non-blocking), NodeJS có thể xử lý hàng ngàn kết nối cùng lúc mà không gặp bất kì khó khăn nào.

- Dễ dàng mở rộng, nâng cấp, phát triển website.

2.6.2 Express

ExpressJS là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs., cung cấp một bộ tính năng mạnh mẽ cho các ứng dụng web và di động. Express cung cấp một lớp mỏng của các tính năng ứng dụng web nền tảng mà không làm mất đi các tính năng Node.js mà người dùng yêu thích. Có nhiều framework phổ biến được xây dựng dựa trên express.

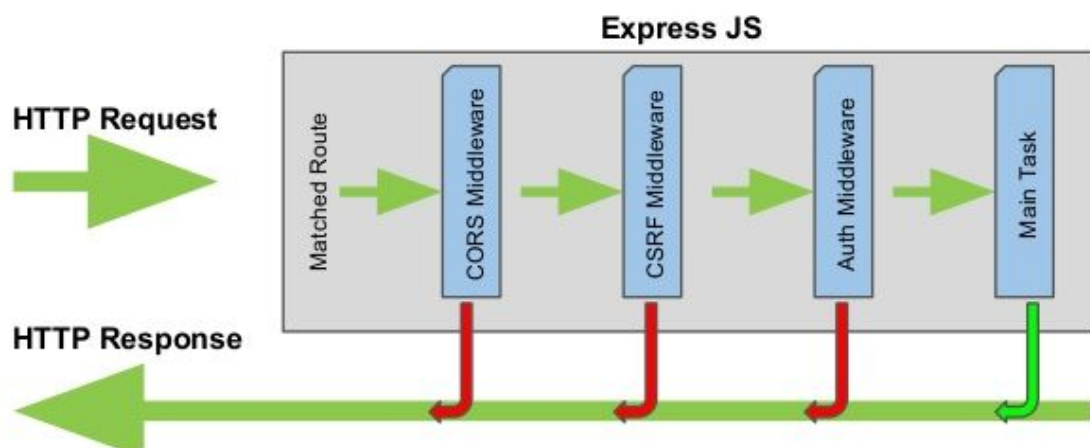
Do đó ExpressJS là một thư viện tuyệt vời để hỗ trợ cho đề án này.

ExpressJS cung cấp các tính năng hữu ích như: định tuyến(Routing), thao tác với phần mềm trung gian(Middleware), ghi đè API Express(Overriding the Express API), xử lý lỗi, sửa lỗi,...

Middleware trong expressJS:

- Với tư tưởng chung là cầu nối giữa tương tác của người dùng và phần nhân của hệ thống, trong lập trình Web, Middleware sẽ đóng vai trò trung gian giữa request/response (tương tác với người dùng) và các xử lý logic bên trong web server. Do đó, Middleware trong các Framework lập trình Web (Django, Rails, ExpressJS), sẽ là các hàm được dùng để tiền xử lý, lọc các request trước khi đưa vào xử lý logic hoặc điều chỉnh các response trước khi gửi về cho người dùng.

Middleware Pattern



HÌNH 2.6.1 - Mô hình Middleware trong expressjs.

- Với 3 middleware có trong ExpressJS như hình trên. Một request khi gửi đến Express sẽ được xử lý qua 5 bước như sau :
 1. Tìm Route tương ứng với request.
 2. Dùng CORS Middleware để kiểm tra cross-origin Resource sharing của request.
 3. Dùng CRSF Middleware để xác thực CSRF của request, chống fake request.
 4. Dùng Auth Middleware để xác thực request có được truy cập hay không.
 5. Xử lý công việc được yêu cầu bởi request (Main Task).
- Bất kỳ bước nào trong các bước 2,3,4 nếu xảy ra lỗi sẽ trả về response thông báo cho người dùng, có thể là lỗi CORS, lỗi CSRF hay lỗi auth tùy thuộc vào request bị dừng ở bước nào.
- ExpressJS khi hoạt động, về cơ bản sẽ là một loạt các hàm Middleware được thực hiện liên tiếp nhau. Sau khi đã thiết lập, các request từ phía người dùng khi gửi lên ExpressJS sẽ thực hiện lần lượt qua các hàm Middleware cho đến khi trả về response cho người dùng. Các hàm này sẽ được quyền truy cập đến các đối tượng đại diện cho Request - req, Response - res, hàm Middleware tiếp theo - next, và đối tượng lỗi - err nếu cần thiết.
- Các chức năng mà middleware có thể thực hiện trong ExpressJS sẽ bao gồm:
 - Thực hiện bất cứ đoạn code nào.
 - Thay đổi các đối tượng request và response.
 - Kết thúc một quá trình request-response.
 - Gọi hàm middleware tiếp theo trong stack.
- Trong Express, có 5 kiểu middleware có thể sử dụng :
 - Application-level middleware (middleware cấp ứng dụng).
 - Router-level middleware (middleware cấp điều hướng - router).
 - Error-handling middleware (middleware xử lý lỗi).
 - Built-in middleware (middleware sẵn có).
 - Third-party middleware (middleware của bên thứ ba).

2.6.3 Mongoose

Mongoose là một framework JavaScript thường được sử dụng trong ứng dụng Node.js với database là MongoDB.

Để cài đặt mongoose trên node.js: **npm i mongoose**

2.6.4 Multer

Là phần mềm trung gian của NodeJS chủ yếu để tải lên dữ liệu multipart (image, audio, video, ...).

Cú pháp cài đặt module Multer: **npm i multer**

2.6.5 Body-parser

Là module của NodeJS phân tích và giúp ta thao tác với dữ liệu (ở đề án này chủ yếu là dạng JSON với một số lệnh như req.body).

Cú pháp cài đặt module Body-parser: **npm i body-parser**

2.6.6 EJS

EJS là viết tắt của "Embedded JavaScript templating", đây là một thư viện, được sử dụng để phân tích các tập tin ejs, và tạo ra HTML trả về cho client (Trình duyệt).

Dạng file EJS có đuôi .ejs.

2.6.7 Nodemon

Nodemon là một công cụ giúp phát triển các ứng dụng trên NodeJS một cách hiệu quả bằng việc tự động khởi chạy lại ứng dụng khi phát hiện thay đổi trong các file.

Cú pháp cài đặt module Body-parser: **npm i nodemon**

2.7 Cơ sở dữ liệu MongoDB

2.7.1 Tổng quan về MongoDB

MongoDB là một dạng phần mềm cơ sở dữ liệu sử dụng mã nguồn mở NoSQL. Nó có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế với mục đích hướng đến đối tượng. MongoDB hoạt động dựa vào các khái niệm Collection và Document. Đồng thời, nó có hiệu suất cao cùng với tính khả dụng tốt và dễ dàng mở rộng.

Các Collection trong MongoDB có cấu trúc cực kỳ linh hoạt. Điều này cho phép dữ liệu không cần thiết phải tuân theo bất kỳ một dạng cấu trúc nào. Vì thế, MongoDB có

thể lưu trữ những dữ liệu có cấu trúc đa dạng và phức tạp. Dữ liệu trong MongoDB được lưu bằng định dạng kiểu JSON.

2.7.2 Lý do chọn MongoDB làm CSDL

- Website âm nhạc có database khá phức tạp. MongoDB giúp linh hoạt trong lưu trữ các kích cỡ dữ liệu khác nhau. Nhờ chúng được lưu dưới dạng JSON nên thao tác thuận tiện dễ dàng hơn.
- Có module Mongoose giúp kết nối dễ dàng với NodeJS.
- Ít ràng buộc.
- Dễ dàng mở rộng hệ thống.
- Tốc độ truy vấn nhanh.
- Trường dữ liệu “_id” (đại diện cho giá trị duy nhất trong mỗi document) được tự động đánh chỉ mục nên hiệu suất luôn đạt mức cao nhất.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Phân tích yêu cầu hệ thống

3.1.1 Yêu cầu chức năng

Để một hệ thống nghe nhạc trực tuyến có thể được đưa vào sử dụng thực tế, hệ thống đó cần phải đáp ứng được một số yêu cầu nhất định, cụ thể như sau:

- Nghe nhạc trực tuyến và tải về để nghe nhạc không trực tuyến.
- Gợi ý bài hát cho người dùng
- Đăng nhập, đăng xuất, có trang cá nhân riêng lưu trữ các thông tin về lịch sử nghe, bài nhạc yêu thích, album yêu thích, ca sĩ yêu thích.
- Xếp hạng các bài hát, album thịnh hành.
- Xem được MV của bài hát.
- Hiển thị lời bài hát.
- Nâng cấp để nghe được nhạc chất lượng cao hoặc sở hữu 1 đặc quyền nào đó mà người dùng bình thường không có. (1 trong những nguồn tiền đề duy trì hệ thống ngoài treo quảng cáo)
- Tìm kiếm bài hát theo từ khóa.
- Cho phép người dùng tải file nhạc từ thiết bị của mình vào ứng dụng để nghe.
- Tạo playlist, like bài hát, ca sĩ hoặc album.
- Các chức năng cơ bản của thanh player như: chuyển qua bài nhạc mới, quay lại bài trước, dừng nghe, tiếp tục, lặp lại 1 bài hát, tăng giảm âm lượng...

Vì thời gian có hạn, kiến thức của nhóm còn hạn chế nên trong đề án này, website nhóm xây dựng chỉ đáp ứng được một số chức năng cơ bản sau:

- Đăng ký, đăng nhập.
- Hiển thị bài hát theo các tiêu chí.

3.1.2 Xác định các đối tượng

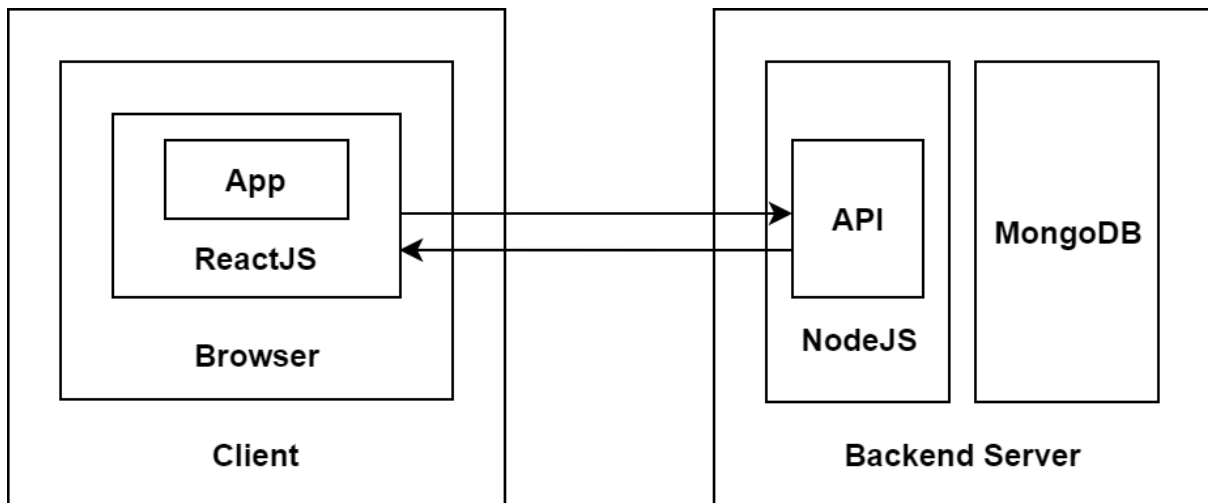
Với khuôn khổ là đề án môn học nên nhóm chúng tôi chỉ thực hiện dựa trên 2 đối tượng là: Người dùng và Lập trình viên (người quản trị hệ thống).

Việc phát triển trang web trong tương lai có thể chúng tôi sẽ phân cấp thêm các nhóm đối tượng như: Người dùng bình thường, Người dùng VIP, Quản trị viên, Lập trình viên, Nhân viên,...

3.2 Phân tích thiết kế

3.2.1 Thiết kế hệ thống

Website âm nhạc được xây dựng với mô hình client-server bao gồm hai thành phần chính sau đây:



HÌNH 3.2.1 - Mô hình client-server trong website âm nhạc.

Server là một RESTful Web Services cung cấp các chức năng sau:

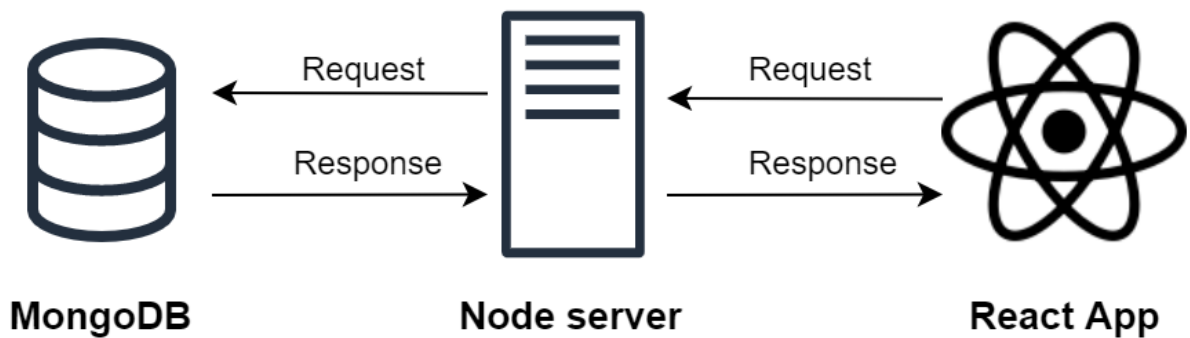
- Thực hiện chức năng quản lý và điều hành hệ thống thông qua nền tảng web nên rất tiện lợi và nhanh chóng.
- Tiếp nhận và xử lý các yêu cầu trực tiếp từ người dùng trong hệ thống.
- Cung cấp các API cho phép thực hiện nhiều tác vụ khác nhau tương ứng với nhu cầu sử dụng của người dùng.
- Sử dụng hệ quản trị cơ sở dữ liệu MongoDB để lưu trữ dữ liệu người dùng.

Client là một web application với các tính năng sau:

- Cung cấp giao diện người dùng trên trình duyệt web một cách hiện đại, đẹp mắt, thân thiện với người dùng.
- Tương tác trực tiếp với server thông qua các API.
- Thực hiện các chức năng liên quan đến dữ liệu người dùng.

- Thực hiện các chức năng liên quan đến dữ liệu bài hát, thông tin mà server tiếp nhận.
- Cung cấp các chức năng cần thiết trong 1 website âm nhạc với giao diện trực quan.

3.2.2 Mô tả web server

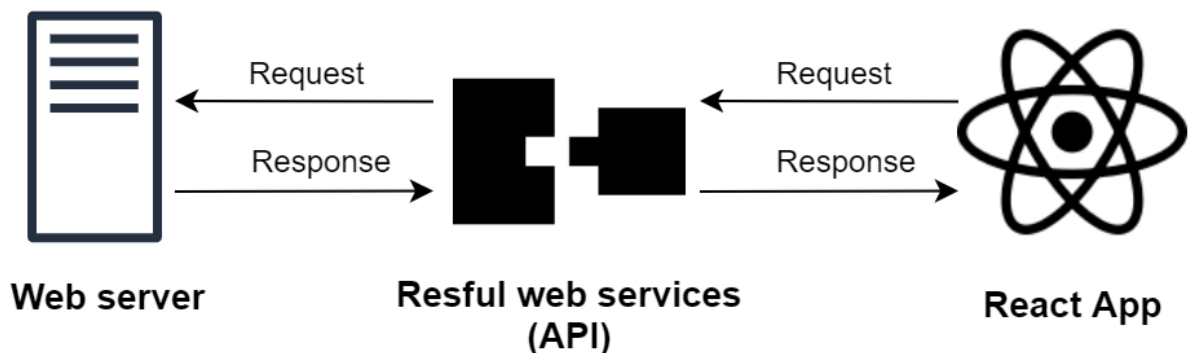


HÌNH 3.2.2 - Mô hình mô tả hoạt động của backend server.

Backend server của hệ thống được xây dựng dựa trên:

- Công nghệ chính: NodeJS.
- Ngôn ngữ: Javascript.
- Hệ quản trị cơ sở dữ liệu: MongoDB.
- Kiểu dữ liệu gửi nhận: Json.

3.2.3 Mô tả web client



HÌNH 3.2.3 - Mô hình mô tả hoạt động web client.

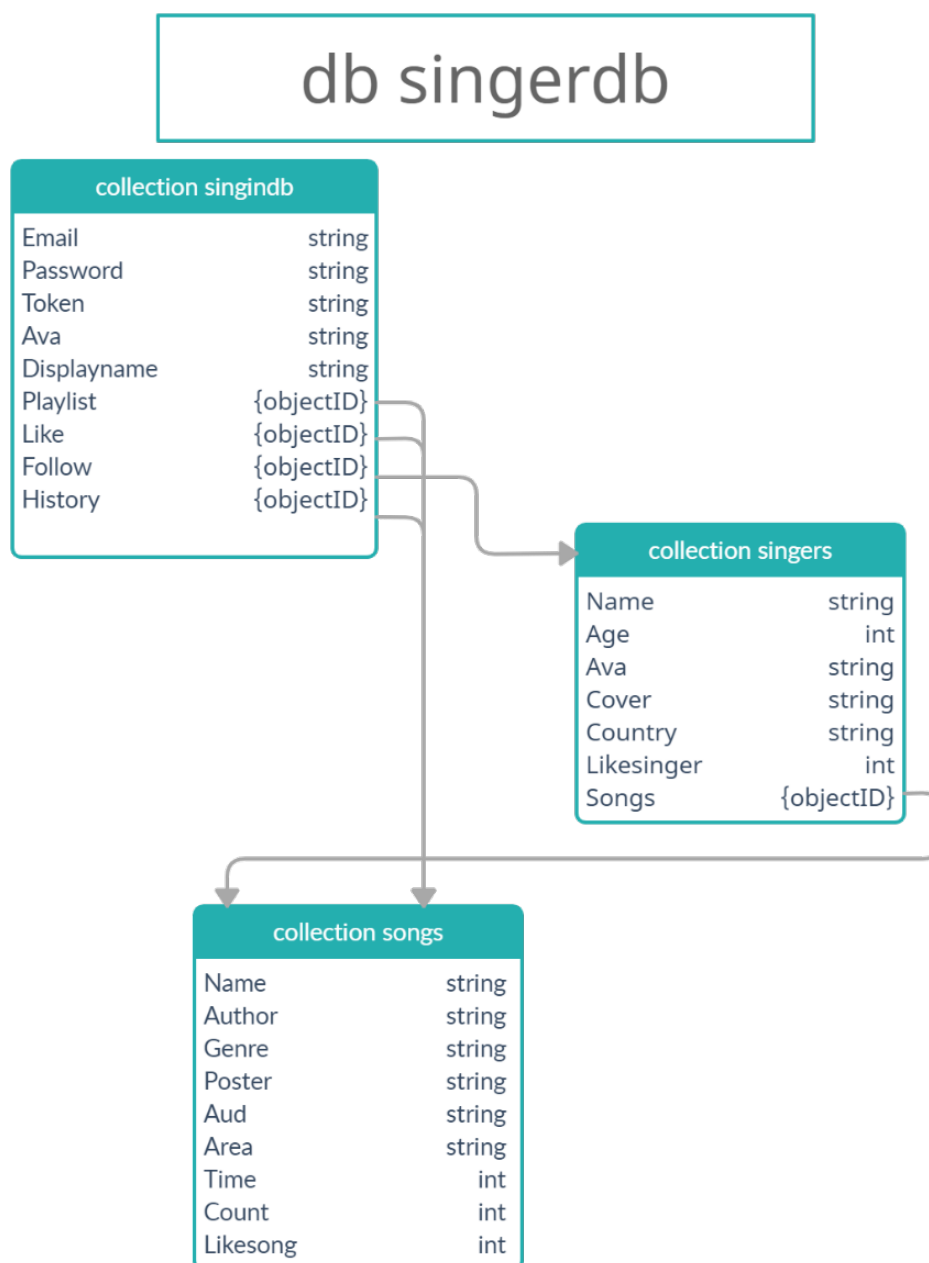
Web client của hệ thống được xây dựng dựa trên:

- Công nghệ chính: ReactJS.
- Ngôn ngữ: HTML, CSS, Javascript, JSX.
- Giao tiếp với server thông qua API.
- Dữ liệu gửi nhận: Json.

- Các công nghệ khác hỗ trợ xây dựng giao diện: JQuery, bootstrap, material-ui, Router, Carousel, SlideShow, Slick, Popup,...

3.2.4 Thiết kế dữ liệu

Database dạng Document file JSON được lưu ở MongoDB:



HÌNH 3.2.4 - Mô hình database nhóm thiết kế cho website.

Database “singerdb” gồm 3 collection:

- Collection singindb:

BẢNG 3.2.1 – Mô tả thuộc tính của người dùng.

| STT | Tên trường | Kiểu | Giải thích |
|-----|-------------|----------|--------------------|
| 1 | Email | String | Email đăng nhập |
| 2 | Password | String | Mật khẩu |
| 3 | Displayname | String | Tên hiển thị |
| 4 | Playlist | objectID | Danh sách nghe |
| 5 | Like | objectID | Danh sách thích |
| 6 | Follow | objectID | Danh sách theo dõi |
| 7 | History | objectID | Lịch sử nghe |

- Collection singers:

BẢNG 3.2.2 – Mô tả thuộc tính của ca sĩ.

| STT | Tên trường | Kiểu | Giải thích |
|-----|------------|----------|-------------------|
| 1 | Name | String | Tên bài hát |
| 2 | Age | int | Tên tác giả |
| 3 | Ava | String | Ảnh Avatar |
| 4 | Cover | String | Ảnh bìa |
| 5 | Country | String | Quê quán |
| 6 | Likesinger | int | Lượt thích |
| 7 | Songs | objectID | Danh sách bài hát |

- Collection songs:

BẢNG 3.2.3 – Mô tả thuộc tính của bài hát.

| STT | Tên trường | Kiểu | Giải thích |
|-----|------------|--------|-----------------|
| 1 | Name | String | Tên bài hát |
| 2 | Author | String | Tên tác giả |
| 3 | Genre | String | Thể loại |
| 4 | Poster | String | Tên ảnh bài hát |
| 5 | Aud | String | File audio |
| 6 | Area | String | Khu vực |

| | | | |
|---|----------|--------|------------|
| 7 | Time | String | Thời lượng |
| 8 | Count | int | Lượt nghe |
| 9 | Likesong | int | Lượt thích |

- Trong singers có thuộc tính Songs là một array objectID, mỗi objectID chính là _id của songs để kết nối 2 collection.
- Tương tự trong singindb cũng có Playlist, Like, Follow, History,... là các array objectID songs nhằm thực hiện các chức năng cùng tên.

CHƯƠNG 4. PHÁT TRIỂN TRANG WEB

4.1 Các giai đoạn phát triển trang Web

Bước 1: Khảo sát lập kế hoạch xây dựng trang Web âm nhạc

- Khảo sát các trang Web âm nhạc hiện hành hiện nay như: Zing mp3, Nhaccuatui, Spotify, Soundcloud,... từ đó tham khảo cấu trúc tổng quát, cách phân chia các layout, components trong một trang Web.
- Để có thể thiết kế trang Web phù hợp, tối ưu nhất nhóm chúng tôi đã lựa chọn cũng như khảo sát các đối tượng đa dạng bao gồm: Sinh viên, lập trình viên, học sinh, giáo viên.
- Để nắm bắt được nhu cầu cũng như mong muốn của người nghe đối với một Web âm nhạc thì nhóm chúng tôi cũng đưa vào form khảo sát những câu hỏi vừa liên quan tới thói quen nghe nhạc của người sử dụng, những tính năng hay chức năng mà người sử dụng cho là gây phiền phức tới trải nghiệm nghe nhạc và cuối cùng là mong muốn cách trình bày, bố cục, màu sắc của người sử dụng đối với một trang Web âm nhạc.

Bước 2: Phân tích

- Dựa vào form khảo sát người dùng nhóm chúng tôi đã tổng hợp và phân tích từ những câu hỏi chúng tôi đưa ra cho người sử dụng, từ đó có thể loại bỏ những chức năng mà nhóm chúng tôi cho là không cần thiết.
- Phân tích các chức năng mà các trang Web hiện nay đang có, từ đó phân tích các luồng chạy của dữ liệu, các mối quan hệ giữa các khu vực chính của trang Web.
- Sau đó chúng tôi lập kế hoạch thực hiện, phân chia công việc phù hợp ra sao cho từng thành viên trong nhóm.

Bước 3: Giai đoạn thiết kế

- Tổng hợp và phân loại những nội dung có trong trang Web mà nhóm xây dựng.
- Thiết kế outline cho trang chính cũng như các trang phụ đi kèm.
- Quyết định màu sắc, font chữ cho các phân vùng của Web.

- Thiết kế một UI đơn giản bằng wireframes từ đó có thể hình dung được cũng như tham khảo để đảm bảo nhóm có thể xây dựng Web đúng theo như kế hoạch được đề ra.

Bước 4: Giai đoạn xây dựng

- Xây dựng Frontend dựa trên các thiết kế ban đầu từ đó quyết định xây dựng cơ sở dữ liệu phù hợp dựa trên thiết kế.
- Lập trình và tinh chỉnh các hàm chức năng cho từng components.
- Tích hợp hệ thống, tạo sự kết nối hoàn hảo giữa Frontend và Backend.

Bước 5: Giai đoạn chạy thử

- Kiểm tra và sửa lỗi.
- Tối ưu và tinh chỉnh những thứ cần thiết trên các components.
- Kiểm tra hiệu suất của trang Web trên các nền tảng thiết bị khác nhau.
- Đảm bảo hệ thống chạy mượt mà, không có lỗi xảy ra.

Bước 6: Giai đoạn nghiệm thu cũng như công bố sản phẩm

- Đào tạo cũng như huấn luyện cho người sử dụng quản trị Web một cách hiệu quả.
- Đánh giá, bảo trì.

4.2 Kiểm tra ứng dụng

1. Test link

- Không có link trang web nào bị chết bao gồm:
 - Liên kết ngoài trang web.
 - Liên kết nội bộ.
 - Liên kết tới vị trí cùng trang.

2. Test forms

- Kiểm tra forms đăng nhập của trang đảm bảo đã hoạt động như mong đợi, bao gồm các yêu cầu sau:
 - Khi nhập không đúng như thông tin mà khách hàng đã đăng ký thì sẽ báo lỗi.
 - Yêu cầu nhập đúng các validate của các trường thông tin.
 - Các form thao tác dễ nhìn.

3. Test khả năng sử dụng

- Các bài hát được nghe một cách ổn định, mượt mà.
- Việc tìm và lọc ra những bài hát được nghe theo top cũng hoạt động một cách hợp lý, chính xác.
- Tất cả các nút chức năng trang Web đều thuận tiện cho người sử dụng.
- Người sử dụng đều có quyền truy cập vào menu chính từ các trang khác.

4.3 Kết quả đạt được

Web âm nhạc được xây dựng từ ReactJS và NodeJS và một số framework hỗ trợ.

4.4 Hạn chế

Nhóm đã thiết kế cũng như hoàn thành các tính năng yêu cầu cơ bản của trang Web như đăng nhập, nghe nhạc. Nhưng vì lượng kiến thức nhiều, thời gian ngắn cho nên một số tính năng dự định ban đầu nhưng nhóm chúng tôi chưa có thời gian hoàn thành.

ReactJS đối với nhóm chúng tôi thực sự không hề dễ và đơn giản, cần sự tìm hiểu rất lâu mới có thể hoàn thành một cách căn bản. Về giao diện web là do nhóm tham khảo từ nhiều nguồn và form khảo sát chưa có đánh giá cụ thể một cách bài bản hoặc tìm hiểu sâu về giao diện người dùng nên có một vài phân bố trí vẫn chưa phù hợp.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Sau hơn 2 tháng nghiên cứu, định hướng, thiết kế và xây dựng thì hệ thống website âm nhạc đã hoàn thành được khoản 60% về mặt yêu cầu so với ban đầu đề ra. Mặc dù kết quả hoàn thành không cao nhưng nhóm chúng tôi tin rằng với những kiến thức và khó khăn trong quá trình làm đồ án sẽ giúp ích rất nhiều về mặt kinh nghiệm cho nhóm chúng tôi thực hiện các đồ án môn học khác ở các học kỳ tới. Với sự hướng dẫn tận tình từ phía Thầy và hỗ trợ từ các bạn trong lớp nhóm chúng tôi có thể hoàn thành tốt đồ án này. Website giải trí hay cụ thể hơn là website âm nhạc không phải là một đề tài mới mẻ nhưng để xây dựng và hoàn thiện được đề tài này đòi hỏi người thực hiện cần phải nỗ lực tìm kiếm và học hỏi rất nhiều kiến thức mới, đồng thời phải áp dụng những công nghệ mới trong quá trình xây dựng và phát triển các chức năng. Có thể nói đây là thành quả trong suốt quá trình học tập và nỗ lực nghiên cứu thực hiện của nhóm, trong tương lai hệ thống sẽ có nhiều cải tiến mới và đáp ứng được các nhu cầu của người dùng.

5.2 Kết quả đạt được

Sau khi hoàn thành đồ án Website âm nhạc nhóm chúng tôi đã đạt được các kết quả sau:

- Hiểu rõ hơn về các yêu cầu của người dùng với một website giải trí đặc biệt là với chủ đề âm nhạc.
- Nắm được cơ bản các kỹ thuật quan trọng trong lập trình ứng dụng web, cũng như kết hợp hai công nghệ nổi tiếng là NodeJS vào việc xây dựng Backend và ReactJS vào việc xây dựng Frontend để tạo một Website âm nhạc.
- Các chức năng thực hiện được chưa được nhiều, nhưng về mặt tổng thể thì cũng đã hoàn thành được trang web mở mức độ cơ bản nhất.
- Với sự hỗ trợ từ phía giảng viên cũng như góp ý từ bạn bè trong lớp nhóm đã nắm bắt được những thiếu sót và khắc phục kịp thời cũng như hoàn thiện hơn hệ thống.

5.3 Hướng phát triển

Nhằm hoàn thiện hơn các tính năng của website âm nhạc trong tương lai, nhóm đề xuất một số hướng phát triển như sau:

- Hoàn thiện các chức năng chưa làm được cũng như các chức năng chưa hoàn thiện.
- Tích hợp thêm các cách bảo mật một trang web.
- Phát triển web trên nền tảng di động.
- Bổ sung thêm nhiều chức năng nâng cao như thanh toán online, chuyển web sang nền tảng app trên các thiết bị di động và app máy tính, stream nhạc trực tiếp với người khác,...

TÀI LIỆU THAM KHẢO

- [1] reactrouter.com, “*REACT TRAINING / REACT ROUTER*”, 2020. [Trực tuyến]. Địa chỉ: <https://reactrouter.com/web/example/basic>. [Truy cập 25/12/2020].
- [2] reactjs.org, “*Tutorial: Intro to React*”, 2020. [Trực tuyến]. Địa chỉ: <https://reactjs.org/tutorial/tutorial.html>. [Truy cập 25/12/2020].
- [3] w3schools.com, “*Node.js Tutorial*”, 2020. [Trực tuyến]. Địa chỉ: <https://www.w3schools.com/nodejs/>. [Truy cập 25/12/2020].
- [4] tutorialspoint.com, “*MongoDB Tutorial*”, 2020, [Trực tuyến]. Địa chỉ: <https://www.tutorialspoint.com/mongodb/index.htm>. [Truy cập 25/12/2020].

BẢNG PHÂN CHIA CÔNG VIỆC

| STT | HỌ VÀ TÊN | MSSV | CÔNG VIỆC | HOÀN THÀNH |
|-----|----------------------------------|----------|--|------------|
| 1 | Huỳnh Ngọc Quân (Nhóm trưởng) | 18521284 | Fontend: phân tích thiết kế chức năng, viết giao diện., phân chia và giám sát công việc. | 9.5/10 |
| 2 | Nguyễn Lê Minh | 18521106 | Backend: trọng tâm về data và các hàm chức năng, giao diện admin. | 9.75/10 |
| 3 | Lê Minh Phước | 18521262 | Frontend: phân tích thiết kế giao diện, các chức năng, trình bày file báo cáo. | 9.5/10 |
| 4 | Đặng Văn Nhân | 18521172 | Backend: trọng tâm vào user và các hàm chức năng, thiết kế powerpoint. | 9.5/10 |
| 5 | Trần Nhật Nam | 18521132 | Tìm kiếm, xử lý dữ liệu, import dữ liệu. | 7/10 |