**TRƯỜNG ĐẠI HỌC SÀI GÒN**



**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN HỌC PHẦN**

**CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI**

**TÊN ĐỀ TÀI:**

**Phân tích và xây dựng ứng dụng nghe nhạc trực tuyến sử dụng ReactJS**

**Họ tên thành viên trong nhóm:**

1. **Phan Đồng Thiên Long - 3118410242242**
2. **Hồ Văn Huy - 3117410096**
3. **Nguyễn Ngọc Thái - 3118410390**
4. **You Chí Thành - 3118410396**

**GIÁO VIÊN HƯỚNG DẪN: ThS.TRẦN SƠN HẢI**

**TP. HCM tháng 04/2022**

**LỜI CẢM ƠN**

# **LỜI CẢM ƠN**

Trong thời gian thực hiện đồ án với sự giúp đỡ ,tạo điều kiện của Khoa công nghệ thông– Đại học Sài Gòn, đặt biệt là sự hướng dẫn trực tiếp, chỉ bảo tận tình của giảng viên bộ môn Các công nghệ lập trình hiện đại nhóm em đã hoàn thành đề tài cùng với bản báo cáo đúng thời gian quy định.

Với khả năng và thời gian có hạn nên không tránh khỏi những thiếu sót, nhóm em rất mong nhận được sự quan tâm, giúp đỡ, tạo điều kiện của thầy cô giáo để nhóm em hoàn thành hơn đề tài nghiên cứu trong thởi gian tới.

Một lần nữa em xin chân thành cảm ơn các thầy, cô giáo trong Khoa công nghệ thông tin– Đại học Sài Gòn đã chỉ bảo, dạy dỗ chúng em trong suốt thời gian học. Đặt biệt em xin gửi lời cảm ơn sâu sắc tới thầy giáo Trần Sơn Hải đã hướng dẫn em trong suốt quá trình làm đồ án.

Nhóm em xin chân thành cảm ơn!

# **LỜI MỞ ĐẦU**

Cùng với sự phát triển của Internet tại Việt Nam từ đầu những năm 2000, rất nhiều dịch vụ nghe nhạc trực tuyến đã ra đời trong thời gian này. Có thể kể đến như Nhaccuatui, Zing MP3, Spotify, …

Tuy nhiên do cạnh tranh khốc liệt, nhất là vấn đề bản quyền âm nhạc được làm mạnh tay trong vài năm trở lại đây khiến số lượng dịch vụ nghe nhạc trực tuyến còn hoạt động là khá ít. Chỉ có những trang web được đầu tư mạnh mẽ, chuyên nghiệp, nguồn nội dung có bản quyền và phong phú về số lượng mới có thể tiếp tục phát triển.

Số lượng người bỏ tiền ra nghe và tải nhạc tại Việt Nam ngày một tăng, và có thể nói thị trường nghe nhạc trực tuyến nói chung và Zing MP3 nói riêng đã góp phần thay đổi cách nhìn của người dùng Việt về bản quyền âm nhạc.

Sự bùng nổ về kết nối Internet di động với 3G và 4G hứa hẹn sẽ còn đưa nền âm nhạc trực tuyến trong nước vươn lên tầm cao mới. Hiện Zing MP3 là một trong những ứng dụng nghe nhạc đắt khách với hàng chục triệu lượt tải trên Android và iOS.

Triển vọng của thị trường nghe nhạc trực tuyến trong nước là rất lớn, khi công nghệ 4G được triển khai và đối tượng khách hàng trẻ ngày càng quen thuộc với hình thức nghe nhạc bản quyền. Kho nội dung phong phú, công nghệ hiện đại sẽ là những yếu tố quyết định xem nhà cung cấp nào chiếm ưu thế trong cuộc đua nghe nhạc trực tuyến tại Việt Nam thời gian tới.

Nhìn thấy được nhu cầu của người dùng, giúp người dùng có thể quản lý được kho nhạc cá nhân. Nhóm em quyết định “XÂY DỰNG APP NGHE NHẠC ONLINE BẰNG REACTJS”

*Nhóm em xin chân thành cảm ơn!*

NHẬN XÉT

(Của giảng viên hướng dẫn)

# **MỤC LỤC**

[**LỜI CẢM ƠN** 2](#_Toc101130309)

[**LỜI MỞ ĐẦU** 3](#_Toc101130310)

[NHẬN XÉT 4](#_Toc101130311)

[**MỤC LỤC** 5](#_Toc101130312)

[**Chương 1: TỔNG QUAN** 7](#_Toc101130313)

[**1.1** **Tính cấp thiết của đề tài** 7](#_Toc101130314)

[**1.2** **Mục đích của đề tài** 7](#_Toc101130315)

[**1.3** **Đối tượng nghiên cứu và phạm vi nghiên cứu** 7](#_Toc101130317)

[**1.4** **Phương pháp nghiên cứu** 7](#_Toc101130318)

[**1.5** **Ý nghĩa khoa học và thực tiễn của đề tài** 7](#_Toc101130320)

[**Chương 2: CƠ SỞ LÝ THUYẾT** 8](#_Toc101130321)

[**2.1** **JavaScript là gì?** 8](#_Toc101130322)

[2.2 Giới thiệu về ReactJS 8](#_Toc101130323)

[2.2.1 Đặc tính của ReactJS 8](#_Toc101130324)

[**2.2.2 Thế mạnh của ReactJS** 8](#_Toc101130325)

[2.3 Giới thiệu về Redux 10](#_Toc101130326)

[2.3.1 Redux là gì? 10](#_Toc101130327)

[**2.3.2 Trạng thái (state) của ứng dụng** 10](#_Toc101130328)

[2.3.3 Nguyên lý của Redux 10](#_Toc101130329)

[2.3.4 Tổng quan về Redux và cách vận hành 10](#_Toc101130330)

[2.4 Một số khái niệm, kiến thức khi nghiên cứu ReactJS 12](#_Toc101130331)

[2.5 Kiến thức cần có để làm việc với ReactJS 14](#_Toc101130332)

[2.5.1 Javascript 14](#_Toc101130333)

[2.5.2 NPM 14](#_Toc101130334)

[2.5.3 Javascript “Bundlers” 14](#_Toc101130335)

[2.5.4 ES6 15](#_Toc101130336)

[2.5.5 Routing 15](#_Toc101130337)

[2.5.6 Server-side Rendering 15](#_Toc101130338)

[2.6 Kết luận về ReactJS 16](#_Toc101130339)

[**Chương 3: PHÂN TÍCH XÂY DỰNG APP NGHE NHẠC ONLINE BẰNG REACTJS** 18](#_Toc101130340)

[**3.1 Xác định yêu cầu** 18](#_Toc101130341)

[**3.2 Phân tích yêu cầu** 18](#_Toc101130342)

[**3.3 Mô hình hoá yêu cầu** 18](#_Toc101130343)

[**3.4 Sơ đồ giao diện** 20](#_Toc101130344)

[**Chương 4: XÂY DỰNG APP NGHE NHẠC ONLINE BẰNG REACTJS** 21](#_Toc101130345)

[**4.1 Thiết kế layout** 21](#_Toc101130346)

[**4.1.1 Layout phía người dùng** 21](#_Toc101130347)

[**4.2 Thiết kế giao diện** 23](#_Toc101130348)

[**4.2.1 Giao diện trang chủ** 23](#_Toc101130349)

[**4.2.2 Giao diện tìm kiếm** 23](#_Toc101130350)

[**4.2.3 Giao diện control player và thông nhạc đang chơi** 24](#_Toc101130351)

[**4.2.4 Giao diện danh sách nhạc** 24](#_Toc101130352)

[**4.2.5 Giao diện Trending** 25](#_Toc101130353)

[**4.2.6 Giao diện Discover** 25](#_Toc101130354)

[**4.2.7 Giao diện Dark Mode** 26](#_Toc101130355)

[**4.2.8 Giao diện mobile** 26](#_Toc101130356)

[**Chương 5: BÁO CÁO** 28](#_Toc101130357)

[**5.1 Kết hoạch đạt được** 28](#_Toc101130358)

[**5.2 Hướng phát triển** 28](#_Toc101130359)

[**TÀI liệU THAM THẢO** 29](#_Toc101130360)

# **Chương 1: TỔNG QUAN**

**TÓM TẮT ĐỀ TÀI**

Ứng dụng công nghệ Reactjs để tạo app nghe nhạc giúp người dung dễ dàng quản lý kho nhạc trên laptop một cách nhanh chóng.

* 1. **Tính cấp thiết của đề tài**

- Đảm bảo người dùng có thể quản lý nhanh chóng kho nhạc của mình.

* 1. **Mục đích của đề tài**

- Nghiên cứu công nghệ Reactjs để ứng dụng xây dựng app nghe nhạc.

* 1. **Đối tượng nghiên cứu và phạm vi nghiên cứu**

- Đối tượng nghiên cứu: Ứng dụng nghe nhạc

- Phạm vi nghiên cứu: JavaScript, Cơ sở dữ liệu MySQL, công nghệ Reactjs.

* 1. **Phương pháp nghiên cứu**

-Tìm hiểu và nghiên cứu tài liệu trên mạng

* 1. **Ý nghĩa khoa học và thực tiễn của đề tài**

- Ý nghĩa khoa học:

+ Bổ sung tích lũy cho bản thân thêm kiến thức về cơ sở dữ liệu và công nghệ reactjs.

- Ý nghĩa thực tiễn:

+ Giúp người dùng dễ dàng quản lý kho nhạc của mình.

# **Chương 2: CƠ SỞ LÝ THUYẾT**

## **2.1** **JavaScript là gì?**

JavaScript (js) là một ngôn ngữ lập trình hướng đối tượng trọng lượng nhẹ được một số trang web sử dụng để tạo kịch bản cho các trang web. Nó là một ngôn ngữ lập trình chính thức, được thông dịch cho phép tương tác động trên các trang web khi áp dụng cho tài liệu HTML. Nó được giới thiệu vào năm 1995 để thêm chương trình vào các trang web trong trình duyệt Netscape Navigator. Kể từ đó, nó đã được chấp nhận bởi tất cả các trình duyệt web đồ họa khác. Với JavaScript, người dùng có thể xây dựng các ứng dụng web hiện đại để tương tác trực tiếp mà không cần tải lại trang mỗi lần. Trang web truyền thống sử dụng js để cung cấp một số hình thức tương tác và đơn giản.

2.2 Giới thiệu về ReactJS

ReactJS là thư viện Javascript được xây dựng bởi các kỹ sư của Facebook, đang được rất nhiều công ty nổi tiếng sử dụng để phát triển các sản phẩm của họ như Yahoo, Airbnb và tất nhiên là trong nội tại Facebook, Instagram. Nó phù hợp với các dự án lớn có tính mở rộng hơn là các dự án nhỏ.

2.2.1 Đặc tính của ReactJS

Tư tưởng ReactJS là xây dựng lên các **components** có tính tái sử dụng, dễ dàng cho việc chia nhỏ vấn đề, testing. Nó giúp chúng ta dễ dàng quản lý, mở rộng hệ thống, điều này nếu là Angular thì đòi hỏi cấu trúc, cách viết code phải thật tối ưu.

ReactJS luôn giữ các components ở trạng thái stateless khiến ta dễ dàng quản lý bởi nó chẳng khác gì một trang HTML tĩnh. Bản thân các components này không có trạng thái (state) nó nhận đầu vào từ bên ngoài và chỉ hiện thị ra dựa vào các đầu vào đó, điều này lý giải tại sao nó lại mang tính tái sử dụng và dễ dàng cho việc bảo trì.

### **2.2.2 Thế mạnh của ReactJS**

ReactJS là một thư viện **hiển thị view** chú ý đến **hiệu năng** (performance-minded). Rất nhiều đối thủ nặng ký về framework MVVM (Model-View-ViewModel) mất một thời gian lớn để hiển thị những lượng dữ liệu lớn, như trong trường hợp hiển thị những danh sách và tương tự. Nhưng React đó không còn là vấn đề, vì nó chỉ hiển thị những gì thay đổi.

Một trong những điểm mạnh nữa của ReactJS là virtual DOM - thứ nằm ẩn bên trong mỗi view và là lí do khiến cho React đạt được hiệu năng tốt. Khi một view yêu cầu gọi, tất cả mọi thứ sẽ được đưa vào trong một bản sao ảo của DOM. Sau khi việc gọi hoàn thành, React tiến hành một phép so sánh giữa DOM ảo và DOM thật, và thực hiện những thay đổi được chỉ ra trong phép so sánh trên.

Ví dụ: nếu chúng ta đang xem một danh sách có 20 sản phẩm được hiển thị bởi ReactJS, và chúng ta thay đổi sản phẩm thứ 2, thì chỉ sản phẩm đó sẽ được hiển thị lại, và 19 sản phẩm còn lại vẫn giữ nguyên (không cần hiển thị lại hay reload lại trang). React đã dùng cái gọi là “virtual DOM” để tăng hiệu năng bằng cách xuất ra một hiển thị ảo, sau đó kiểm tra sự khác biệt giữa hiển thị ảo và những gì có trên DOM và tạo một bản vá.

2.3 Giới thiệu về Redux

Khi làm việc với React hay các dự án ứng dụng Single Page nói chung, có một vấn đề khá đau đầu là làm sao quản lý được trạng thái của ứng dụng đó. Sau khi xem qua giới thiệu về một số thư viện hỗ trợ công việc này, em đã quyết định chọn Redux.

2.3.1 Redux là gì?

Redux là một thư viện Javascript giúp tạo ra thành một lớp quản lý trạng thái của ứng dụng.

Redux được xây dựng dựa trên nền tảng tư tưởng của ngôn ngữ Elm và kiến trúc Flux do Facebook giới thiệu. Do vậy Redux thường là bộ đôi kết hợp hoàn hảo với React. Tuy nhiên hoàn toàn có thể sử dụng với các framework khác như Angular, Backbone.

**2.3.2 Trạng thái (state) của ứng dụng**

Một ứng dụng web thông thường sẽ nhận dữ liệu từ phía máy chủ (backend), hay nhận những thao tác của người dùng (nhập, click, …), tất cả những thứ này chúng ta gọi đó là trạng thái của ứng dụng. Nếu biết được trạng thái của ứng dụng tại một thời điểm nào đó, chúng ta sẽ biết vào thời điểm đó ứng dụng đã nhận dữ liệu nào, những thao tác nào đã được người dùng truyền lên.

Ví dụ cho trạng thái của ứng dụng: Khi chúng ta click vào nút Back / Forward trên trình duyệt, thì mỗi trang sẽ là một trạng thái của ứng dụng.

2.3.3 Nguyên lý của Redux

Redux được xây dựng dựa trên 3 nguyên lý như sau:

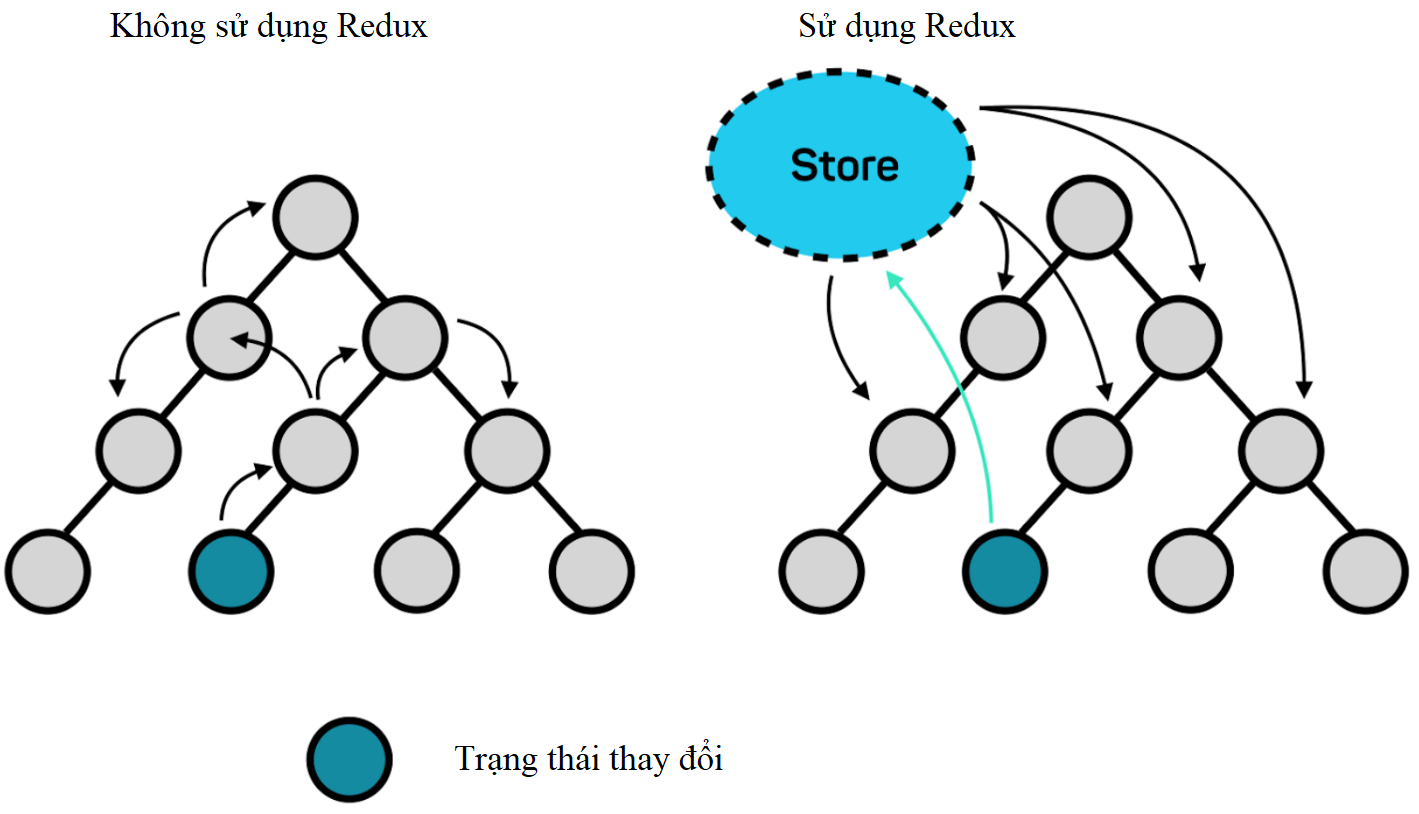
* Trạng thái (state) của toàn bộ ứng dụng được lưu trong trong 1 store duy nhất là 1 Object mô hình tree
* Chỉ có 1 cách duy nhất để thay đổi trạng thái (state) đó là tạo ra một action (là 1 object mô tả những gì xảy ra)
* Để chỉ rõ trạng thái (state) tree được thay đổi bởi 1 action bạn phải viết pure reducers

2.3.4 Tổng quan về Redux và cách vận hành

ReactJS là một thư viện Javascript giúp xây dựng giao diện người dùng, được xây dựng xung quanh các component (thành phần) nhỏ ghép lại với nhau, có hai kiểu dữ liệu trong React đó là state và props. State là trạng thái, mang tính private, chỉ có thể thay đổi ở trong chính component. Còn props mang tính external, không bị kiểm soát bởi component, được truyền từ component cao (cha) đến component thấp (con).

ReactJS xây dựng lên các single-page-app, tức là chỉ render ra 1 trang, và tất cả các thành phần của ứng dụng sẽ được lưu trữ trong đó. Vì thế, nếu ứng dụng phức tạp lên theo thời gian, việc quản lý state của chúng ta sẽ ngày một lớn dần.

Sự khác biệt với việc sử dụng Redux và không sử dụng Redux được miêu tả bằng hình vẽ dưới đây:



Với việc không sử dụng Redux, các component sẽ giao tiếp với nhau bằng props. Nếu chúng ta cần lấy state của một component cách nhau 3 tầng, chúng ta phải gọi 3 lần, điều đó sẽ khiến việc code và quản lý state rất phức tạp, và to dần lên theo thời gian.

Với việc sử dụng Redux, chúng ta sẽ lưu state của các component vào 1 store chung ở bên ngoài. Sau đó nếu muốn dùng ở component nào chúng ta chỉ cần gọi nó và sử dụng.

Kết luận: Redux giúp chúng ta xây dựng một STORE thay vì lưu trữ toàn bộ dữ liệu tại một component thì chúng ta sẽ lưu nó tại store của redux, và nếu có thay đổi, đơn giản là chúng ta sẽ thông qua redux và thay đổi dữ liệu trong store.

2.4 Một số khái niệm, kiến thức khi nghiên cứu ReactJS

2.4.1 JSX là gì?

**JSX** là một cú pháp mở rộng cho JavaScript, là kết hợp của JavaScript và XML. Nó chuyển đổi cú pháp dạng gần như XML về thành JavaScript, giúp chúng ta có thể code ReactJS bằng cú pháp của XML thay vì phải dùng JavaScript.

Thay vì phải truy xuất đến một thẻ HTML bất kì thông qua các thuộc tính như id, class, name... thì ta đã có thể thoải mái sử dụng các thẻ HTML trong code JavaScript rồi. Tất cả những gì ta cần làm đó là viết 1 đống thẻ html và sau đó đặt chúng vào trong một biến, một function hoặc một class chẳng hạn.

Ví dụ:

const name = 'Josh Perez';

const element = <h1>Hello, {name}</h1>;

ReactDOM.render(

element,

document.getElementById('root')

);

2.4.2 Component là gì?

Component là nơi render ra các mã html thuần, một điều quan trọng khác đó là nó cho phép ta chia nhỏ đoạn code của UI thành những phần độc lập với nhau, với mục đích để tiện cho việc quản lý và tái sử dụng. Vậy làm cách nào để có thể gọi đến một Component? Câu trả lời là trong React tên của Component sẽ được dùng để làm tên thẻ và cách sử dụng tương tự với các thẻ tag trong html. Ví dụ:

const App = () => (

<h1>Hello, My name is Bao Vu</h1>

);

ReactDOM.render(

<App />

, document.getElementById('root')

);

Trong đoạn code ở trên, ta có sử dụng cú pháp của ES6 để định nghĩa Component bằng cách khai báo một function tên là App và function này sẽ được kế thừa từ React.Component. Sau khi kế thừa từ React.Component thì ta có thể sử dụng hàm render() mà nó cung cấp để render ra toàn bộ nội dung bên trong. ReactDOM.render sẽ có nhiệm vụ là tạo ra DOM ảo từ component <App /> tại element có id bằng root.

2.4.3 Props là gì?

Props là viết tắt của Properties, ta có thể hiểu nó như là những thuộc tính của một Component. Nếu coi Component như là đối tượng trong lập trình OOP thì Props giống như là những miêu tả về tính chất, đặc trưng của 1 đối tượng. Để định nghĩa Props thì ta khai báo chúng ở bên trong dấu {} của Component. Ví dụ:

const App = props => (

<h1>Hello, My name is {props.myName}</h1>

);

Như vậy là Component App đã có 1 thuộc tính là myName và ta sẽ sử dụng nó bằng cách truyền vào Props myName 1 giá trị nào đó, có thể là string, array, function, boolean hoặc object. Trong ví dụ này ta sẽ truyền vào 1 string như sau:

...

ReactDOM.render(

<div>

<App myName="Hoang Bao Vu" />

</div>

,document.getElementById('root')

);

Ngoài ra thì Component còn có 1 Props đặc biệt nữa là children, khác với các Props bình thường thường thì children được truyền vào bằng cách đặt giữa 2 thẻ đóng và thẻ mở của 1 Component. Ví dụ:

...

const App = props => (

<h1>{props.children}, My name is {props.myName}</h1>

);

ReactDOM.render(

<App myName="Hoang Bao Vu">Hello Word</App>

, document.getElementById('root')

);

2.4.4 State

Trong React, State được dùng để biểu diễn trạng thái của Component. Tức là giá trị của State có thể thay đổi được, trái ngược hoàn toàn với Props là một giá trị bất biến. Có vẻ như hơi khó hiểu một chút! Lấy ví dụ, ta có một Component tên là Tivi và trong Component Tivi này có định nghĩa một số Props như là height, width, weight và 1 state là status có giá trị là "Off". Rõ ràng là trong suốt quá trình sử dụng một chiếc Tivi thì bạn không thể nào thay đổi được các thuộc tính của nó như là chiều rộng, chiều cao, cân nặng... Nhưng ngược lại, bạn có thể thay đổi trạng thái của chiếc Tivi đó từ Off sang On hoặc từ On sang Off. Và mỗi lần thay đổi trạng thái như vậy màn hình Tivi cũng sẽ hiển thị thay đổi theo.

2.5 Kiến thức cần có để làm việc với ReactJS

ReactJS là một thư viện giúp render ra phần view, nhưng để xây dựng được ứng dụng hoàn chỉnh, chúng ta cần nhiều thứ hơn nữa. Đây là một số kiến thức khi muốn nghiên cứu ReactJS:

2.5.1 Javascript

Ngôn ngữ xây dựng nên React, yêu cầu mức độ trung bình: các khái niệm “object”, “prototype” là bắt buộc nắm rõ,.

2.5.2 NPM

Công cụ quản lý package của Node.js và là phương pháp phổ biến nhất để các lập trình viên front-end và các nhà thiết kế chia sẻ mã nguồn Javascript. Nó bao gồm một hệ thống quản lý module gọi là CommonJS và cho phép bạn cài đặt bất cứ công cụ command-line nào được viết bằng Javascript.

2.5.3 Javascript “Bundlers”

Hay còn gọi là các công cụ đóng gói JavaScript. Vì một vài lí do kĩ thuật mà module CommonJS không thể sử dụng native trên trình duyệt. Ta cần có một “công cụ đóng gói” Javascript để “đóng gói” các modules này thành các files .js mà có thể include được trên trang web qua thẻ <script>.

2.5.4 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.

2.5.5 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](https://github.com/rackt/react-router).

2.5.6 Server-side Rendering

Server-side rendering thường được gọi là “toàn bộ” hay “đồng bộ” JS. Điều đó có nghĩa là có thể lấy bất cứ React components nào và render chúng thành HTML tĩnh trên server. Điều này cải thiện được tốc độ ban đầu vì người dùng không cần phải chờ JS thực hiện download để có thể thấy được UI ban đầu, và React có thể tái sử dụng HTML đã được render phía server, vì vậy mà không cần xử lý ở phía client.

Chỉ cần sử dụng server-side rendering khi ta thấy việc render ban đầu trở nên quá chậm hoặc muốn cải thiện thứ hạng tìm kiếm. Mặc dù Google bây giờ đã thực hiện index các nội dung render phía client

Việc render phía server vẫn cần nhiều công cụ để làm cho nó trở nên chính xác. Vì việc này mặc định được hỗ trợ trong các React components mà không cần quan tâm tới việc render phía server, chúng ta nên xây dựng ứng dụng trước và quan tâm tới việc render phía server sau. Ta sẽ không cần thiết phải viết lại tất cả các components để xử lý việc đó.

2.6 Kết luận về ReactJS

Ưu và nhược điểm của React

Ưu điểm:

* Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
* Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể them vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
* Nó có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp bạn debug code dễ dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường.
* Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dung sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dung vô hiệu hóa Javascript thì sao? Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.
* Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
* Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

**Nhược điểm:**

Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax

Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.

Khó tiếp cận cho người mới học Web

# **Chương 3: PHÂN TÍCH XÂY DỰNG APP NGHE NHẠC ONLINE BẰNG REACTJS**

## **3.1 Xác định yêu cầu**

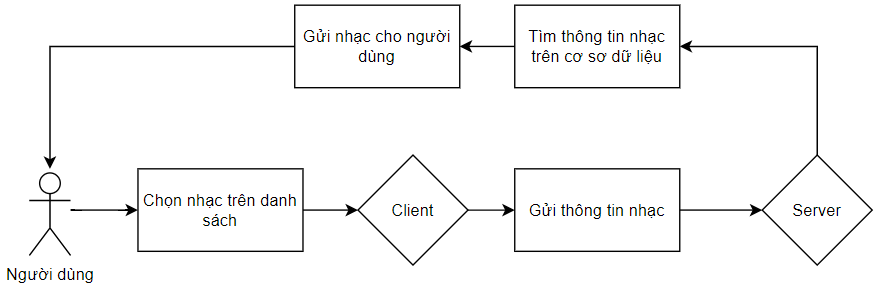
* Tìm kiếm bằng tên ca sĩ hoặc tên bài hát
* Play nhạc
* Tăng giảm âm lượng
* Previous, Next, Shuffle, auto replay
* Timeline
* Đăng ký, đăng nhập thành viên
* Playlist

## **3.2 Phân tích yêu cầu**

Người dùng lựa chọn bài hát qua trên playlist hay bằng cách tìm kiếm, người dùng có thể thực hiện các thao tác điều khiển như play, pause, previous, next, shuffle, auto replay, tua timeline. Người dùng có thể đăng ký/ đăng nhập để có thể lưu playlist của riêng mình

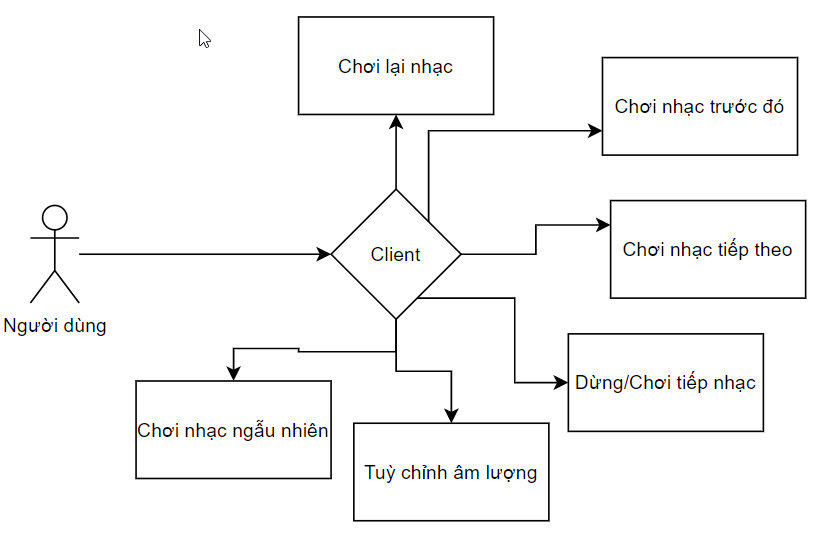
## **3.3 Mô hình hoá yêu cầu**

**Mô hình chọn nhạc**



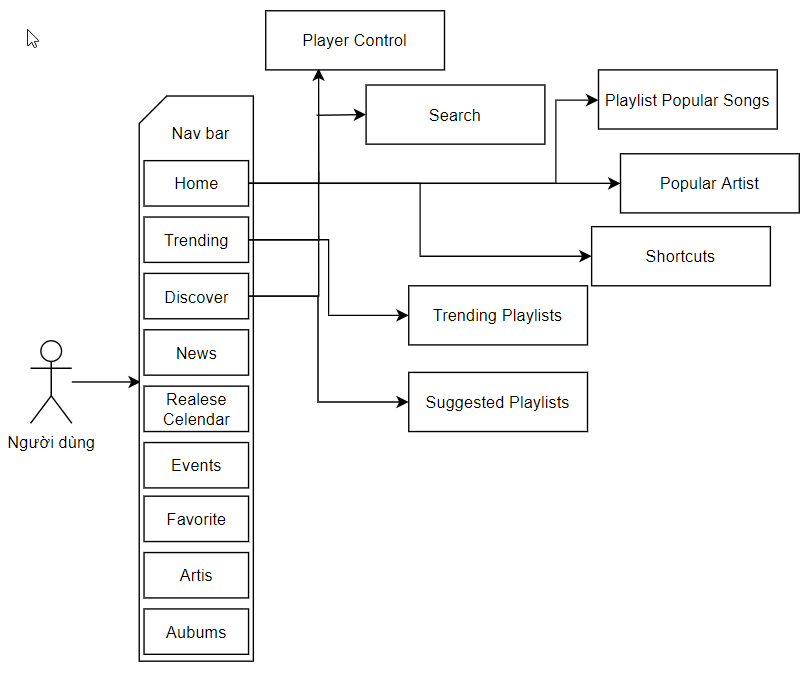
Mô hình tổng quan việc chọn nhạc trên danh sách

**Mô hình công cụ chơi nhạc trên danh sách nhạc.**



Mô hình tổng quan công cụ chơi nhạc

## **3.4 Sơ đồ giao diện**



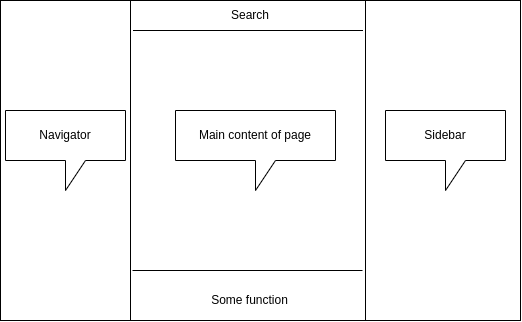
Mô hình giao diện

# **Chương 4: XÂY DỰNG APP NGHE NHẠC ONLINE BẰNG REACTJS**

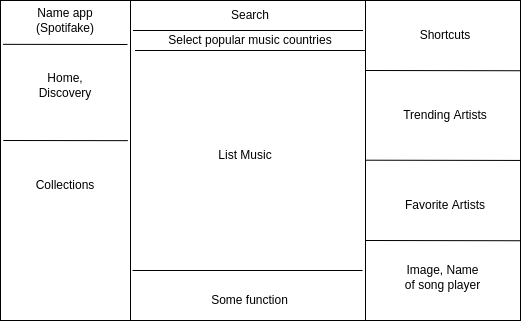
## **4.1 Thiết kế layout**

### **4.1.1 Layout phía người dùng**

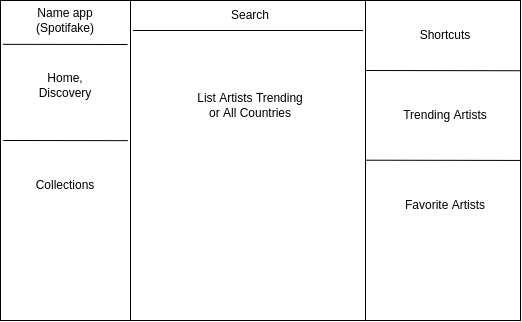
* Phần Mainlayout



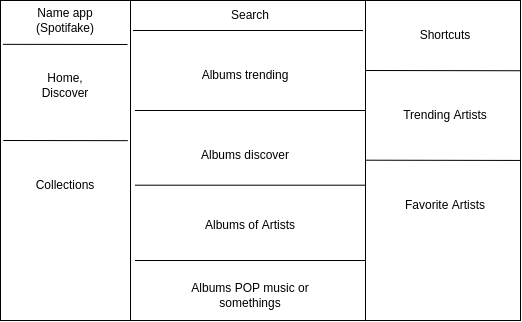
* Trang chính



* Danh sách nhạc sĩ

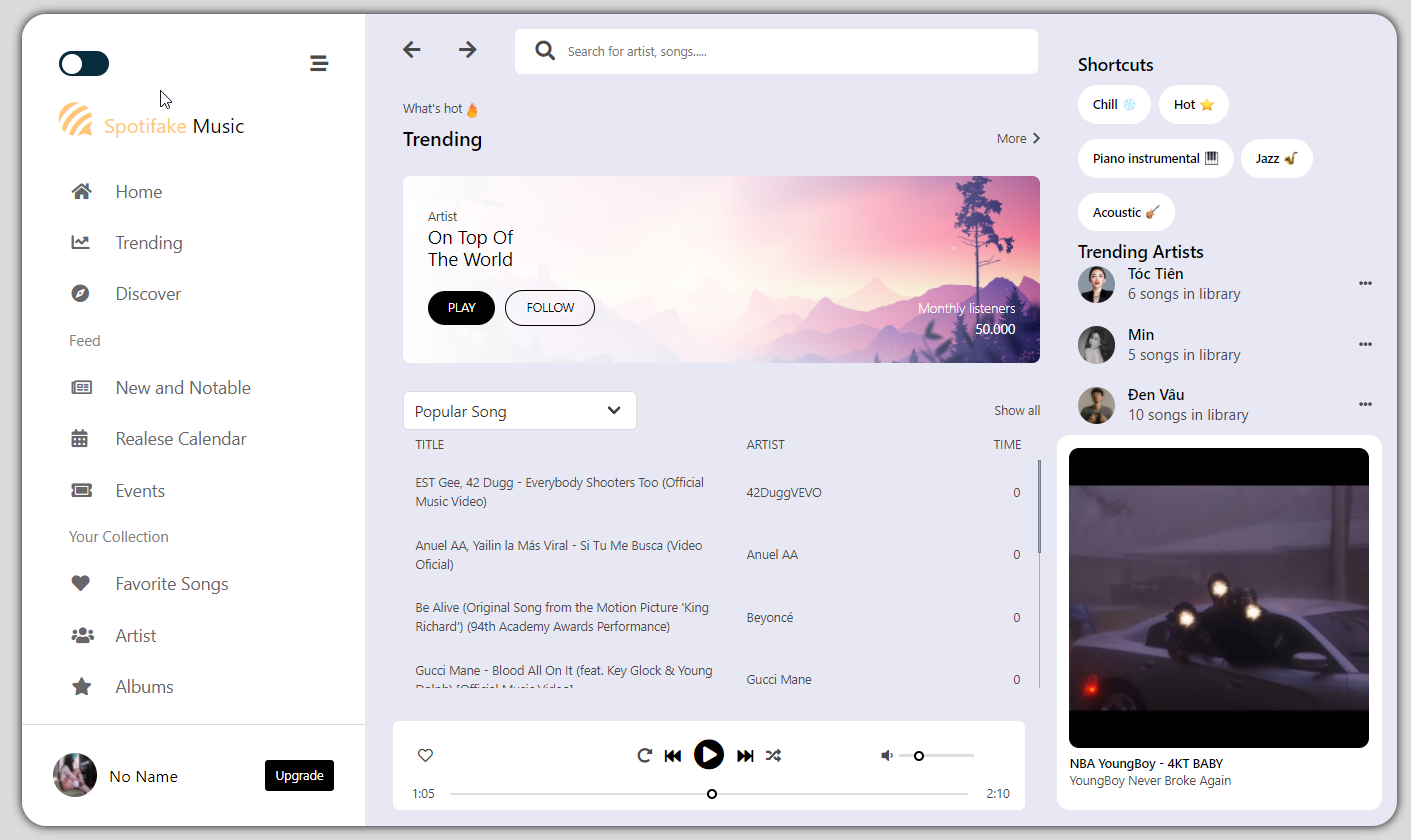


* Album

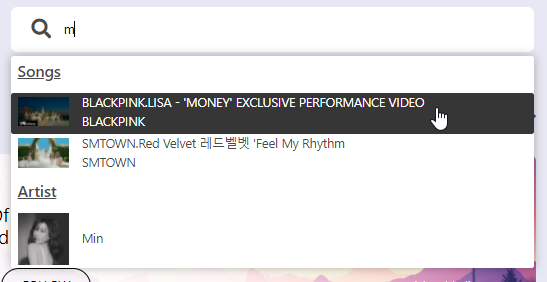


## **4.2 Thiết kế giao diện**

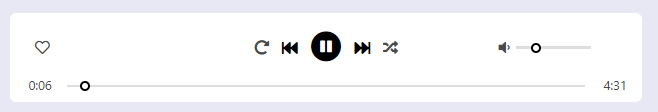
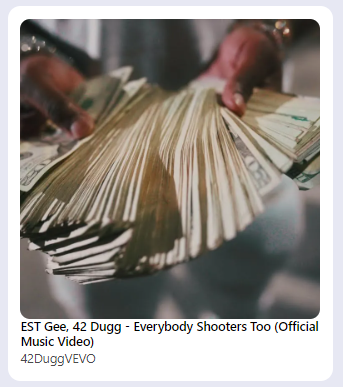
### **4.2.1 Giao diện trang chủ**



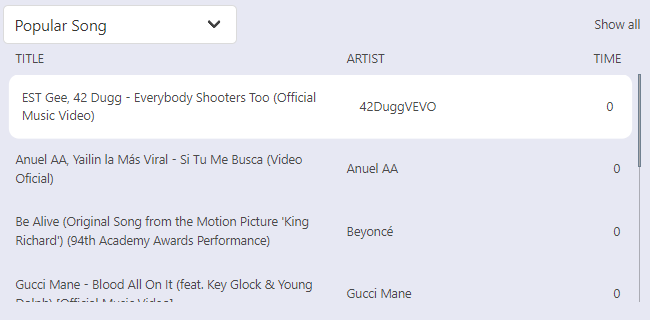
### **4.2.2 Giao diện tìm kiếm**



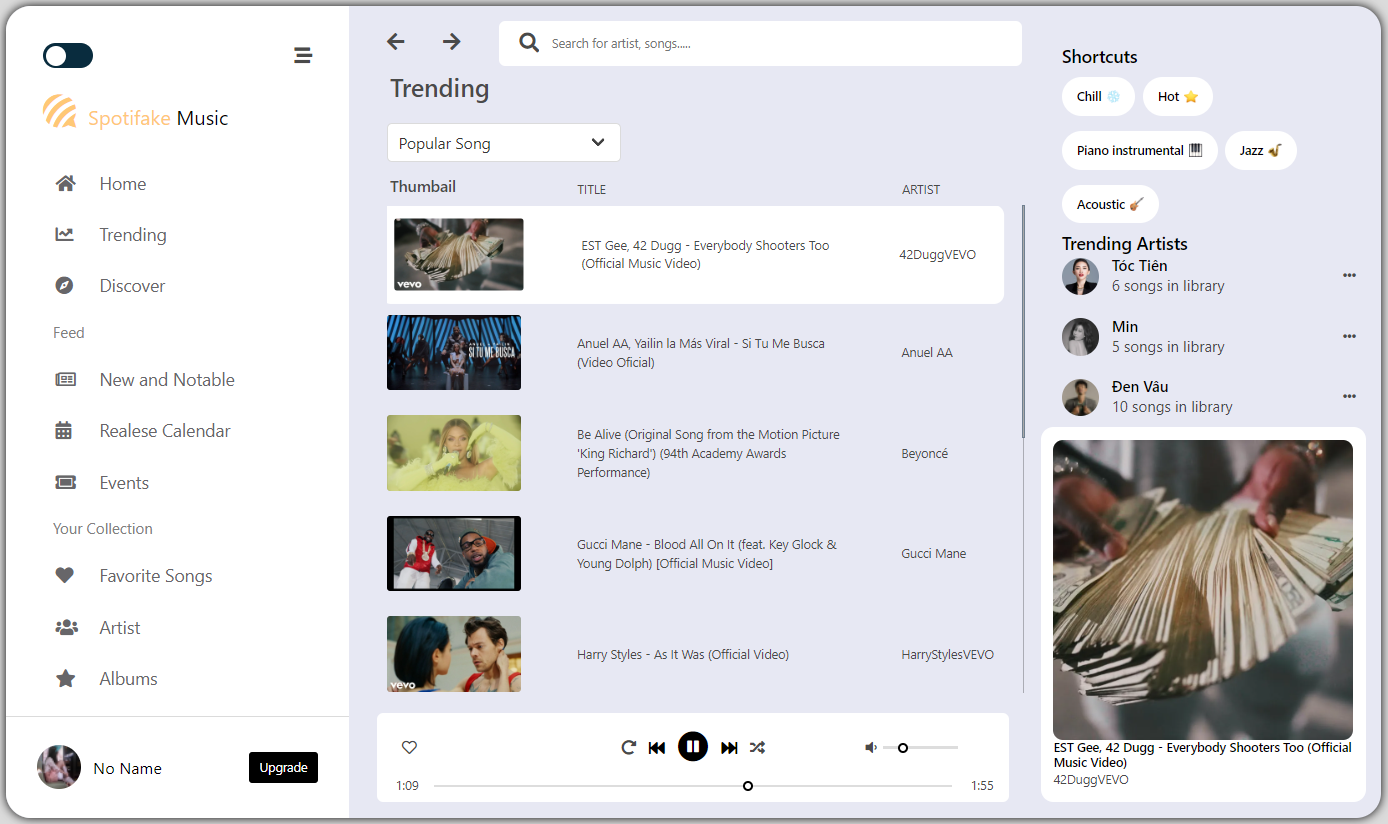
### **4.2.3 Giao diện control player và thông nhạc đang chơi**



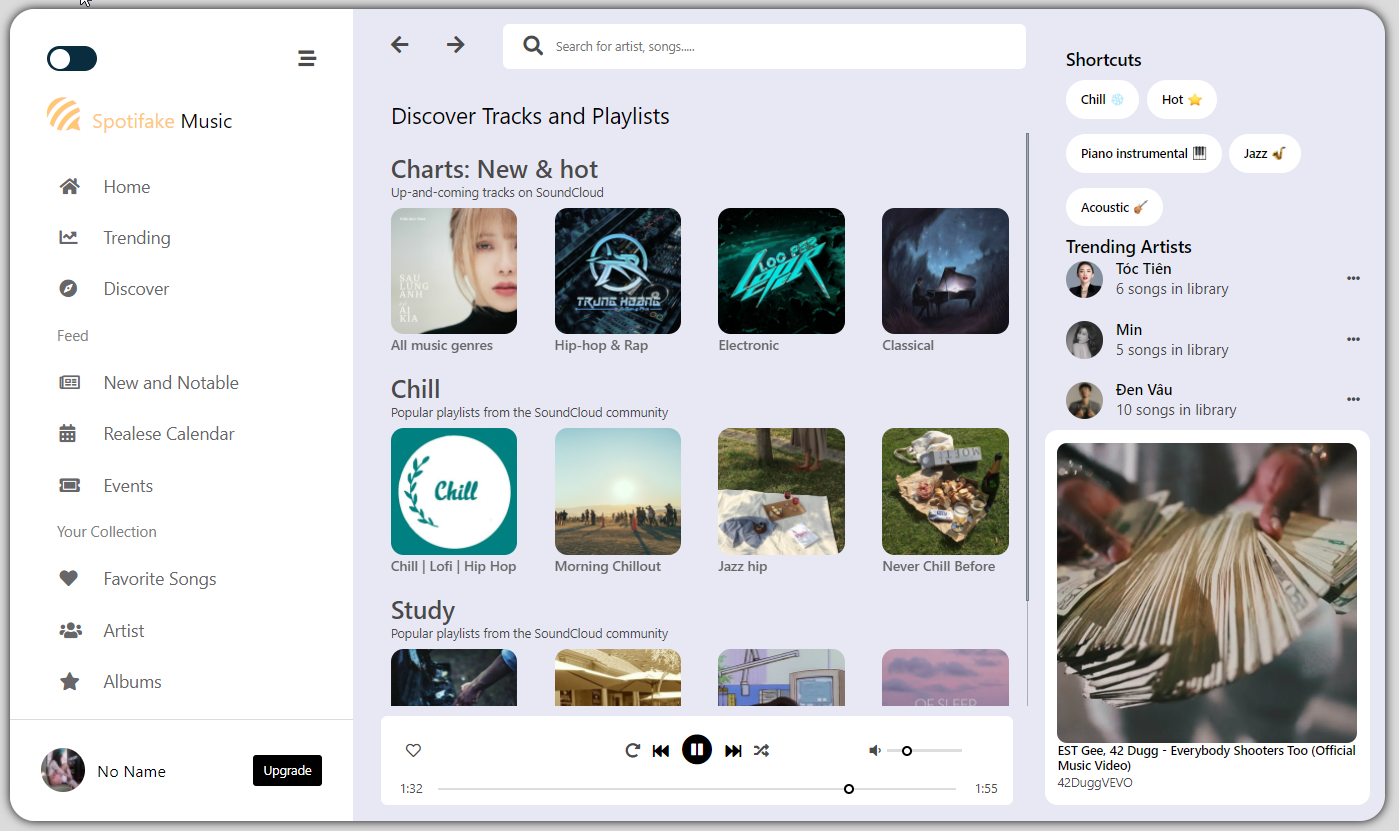
### **4.2.4 Giao diện danh sách nhạc**



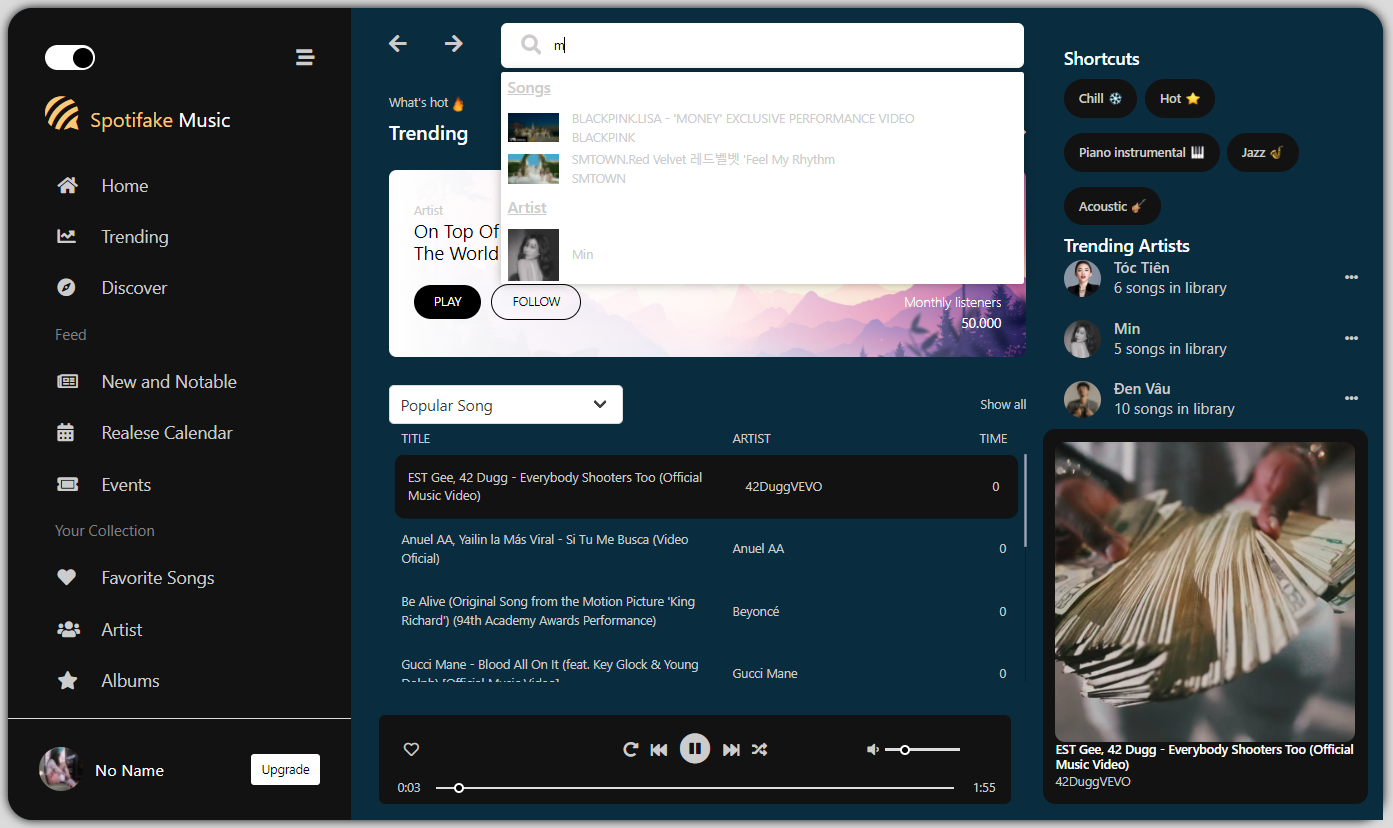
### **4.2.5 Giao diện Trending**



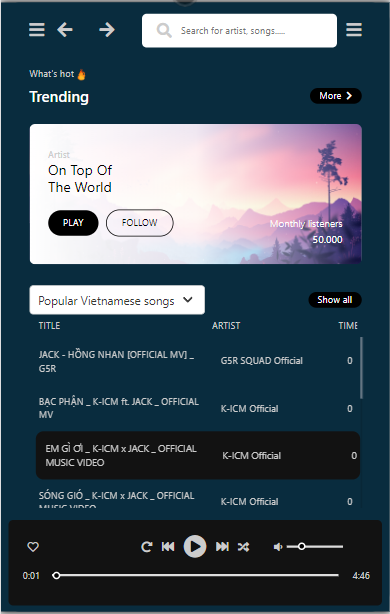
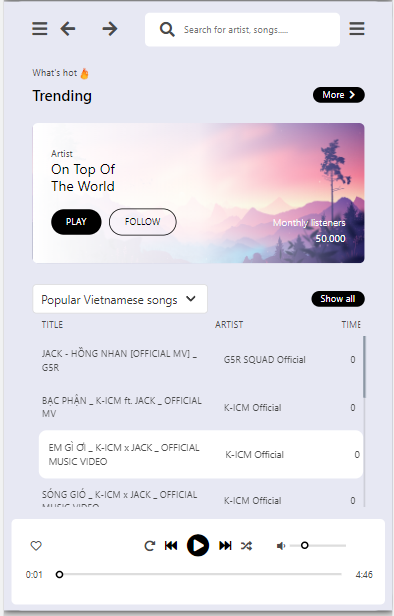
### **4.2.6 Giao diện Discover**

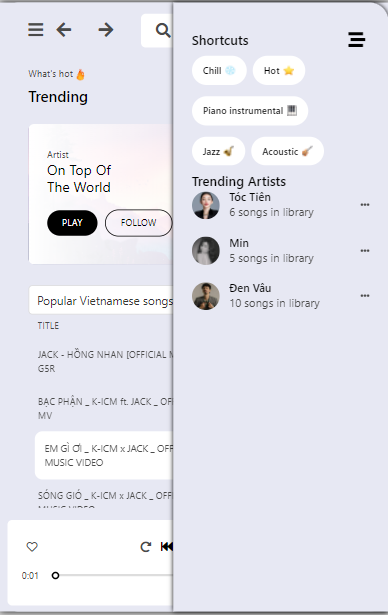
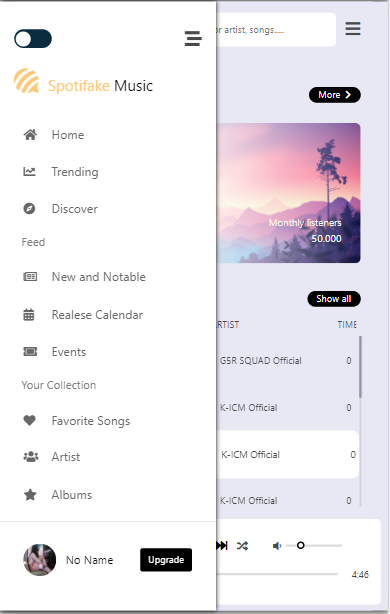


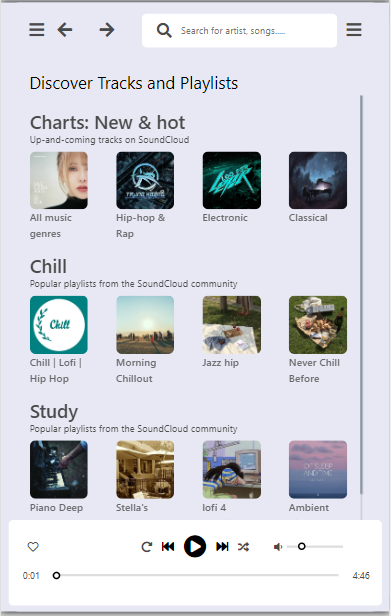
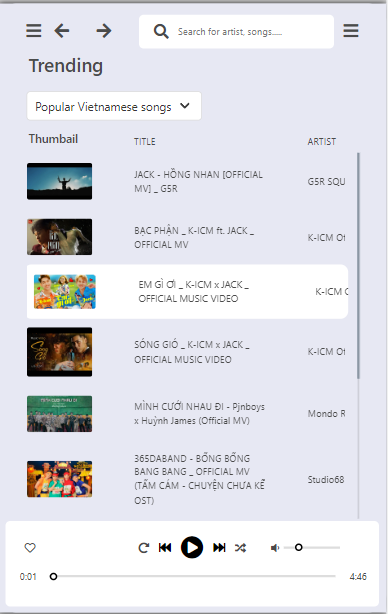
### **4.2.7 Giao diện Dark Mode**



### **4.2.8 Giao diện mobile**







# **Chương 5: BÁO CÁO**

## **5.1 Kết hoạch đạt được**

Trong quá trình thực hiện đề tài, em đã có thêm cơ hội củng cố, tích lũy thêm kiến thức chuyên môn về lập trình web cũng như kinh nghiệm về thiết kế, lên kế hoạch cho đề tài, dự án, …. Nắm rõ hơn về quy trình phát triển phần mềm cũng như cấu trúc một dự án về React.

Sau khoảng thời gian tìm tòi, vận dụng những gì đã học được, sản phẩm của em đã hoàn thành được những chức năng sau:

* Tạo ra giao diện người dùng từ bản thiết kế Photoshop.
* Hỗ trợ đa ngôn ngữ

Ngoài ra, em còn cũng cố thêm kiến thức về:

* Javascript
* ReactJS
* Redux
* Bootstrap
* Áp dụng reactstrap vào trang web, giúp việc tạo giao diện trên React trở lên đơn giản và tối ưu hơn khi loại bỏ jQuery
* Code splitting, tối ưu tốc độ load javascript

## **5.2 Hướng phát triển**

Dù đã tạo được trang web tương đối đầy đủ các chức năng nhưng với thời gian có hạn, ứng dụng vẫn chưa đáp ứng được mọi yêu cầu người dùng.

Trong tương lai em sẽ cố gắng khắc phục, chỉnh sửa một vài sai sót của trang web:

* Khách hàng có thể thêm nhạc, ca sĩ yêu thích cho riêng mình
* Hỗ trợ tải nhạc
* Tìm kiếm nhạc theo yêu cầu

# **TÀI liệU THAM THẢO**

<https://123docz.net/document/1055884-tai-lieu-bao-cao-de-tai-xay-dung-website-nghe-nhac-truc-tuyen-pdf.htm>

<https://babeljs.io/docs/learn-es2015/>

<http://es6katas.org/>

<https://github.com/lukehoban/es6features>

<https://github.com/rse/es6-features>

<http://exploringjs.com/es6/>