

**TRƯỜNG ĐẠI HỌC TRÀ VINH  
KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**THIẾT KẾ GIAO DIỆN WEBSITE  
NGHE NHẠC**

**Giáo viên hướng dẫn:**

Nguyễn Ngọc Đan Thanh

**Sinh viên thực hiện:**

Mã số sinh viên: 110122186

Họ tên: Võ Phước Toàn

Lớp: DA22TTC

*Trà Vinh, ngày 21 tháng 5 năm 2024*

**TRƯỜNG ĐẠI HỌC TRÀ VINH**  
**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**  
**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**THIẾT KẾ GIAO DIỆN WEBSITE**  
**NGHE NHẠC**

**Giáo viên hướng dẫn:**

Nguyễn Ngọc Đan Thanh

**Sinh viên thực hiện:**

Mã số sinh viên: 110122186

Họ tên: Võ Phước Toàn

Lớp: DA22TTC

*Trà Vinh, ngày 21 tháng 5 năm 2024*

[illegible]

(Ký và ghi rõ họ tên)

## **LỜI CẢM ƠN**

Em xin chân thành cảm ơn các thầy cô trường Đại học Trà Vinh, đặc biệt là các thầy cô khoa Kỹ thuật & Công nghệ đã tạo cơ hội cho em được thực tập và giao lưu. Em có thể tránh được các vấn đề và nhầm lẫn trong môi trường làm việc trong tương lai.

Em xin cảm ơn cô Nguyễn Ngọc Đan Thanh. Nhờ sự giúp đỡ và hướng dẫn tận tình của cô từ đầu đến cuối đồ án, em đã hoàn thành đồ án đúng thời hạn và tích lũy được một vốn kiến thức quý báu.

Mặc dù đã cố gắng hết sức để hoàn thành đề tài này nhưng do hạn chế về thời gian cũng như kiến thức chuyên môn nên trong quá trình nghiên cứu, đánh giá và trình bày đề tài, em còn nhiều thiếu sót. Em rất mong được sự quan tâm, góp ý của cô bộ môn để đồ án môn học của em được hoàn thành và hoàn thiện hơn. Em xin chân thành cảm ơn.

Võ Phước Toàn.

## **MỤC LỤC**

|   |           |
|---|-----------|
| <b>CHƯƠNG 1. Tổng quan về nội dung nghiên cứu .....</b> | <b>1</b>  |
| 1.1. Quy trình phát triển website .....                 | 1         |
| 1.2. Ngôn ngữ HTML .....                                | 4         |
| 1.2.1. Các thẻ cơ bản trong ngôn ngữ HTML .....         | 5         |
| 1.3. Bảng định kiểu CSS .....                           | 14        |
| 1.3.1. Giới thiệu về bảng định kiểu CSS .....           | 14        |
| 1.3.2. Các loại CSS .....                               | 15        |
| 1.3.3. Các quy tắc về CSS cơ bản .....                  | 16        |
| 1.4. Ngôn ngữ Javascript .....                          | 18        |
| 1.4.1. Giới thiệu về Javascript .....                   | 18        |
| 1.4.2. Khái niệm về Javascript .....                    | 18        |
| 1.4.3. Các thành phần cơ bản của Javascript .....       | 19        |
| 1.4.4. Javascript cơ bản .....                          | 20        |
| 1.5. Kết chương .....                                   | 21        |
| 1.5.1. Quy trình phát triển Website .....               | 21        |
| 1.5.2. Ngôn ngữ HTML .....                              | 23        |
| 1.5.3. Ngôn ngữ JavaScript .....                        | 23        |
| <b>CHƯƠNG 2. Phân tích thiết kế hệ thống .....</b>      | <b>25</b> |
| 2.1. Mô tả bài toán .....                               | 25        |
| 2.1.1. Giới thiệu .....                                 | 25        |
| 2.1.2. Phân tích yêu cầu .....                          | 25        |
| 2.1.3. Thiết kế và triển khai .....                     | 25        |
| 2.1.4. Cải tiến .....                                   | 26        |

|   |           |
|---|-----------|
| 2.1.5. Kết luận .....                             | 27        |
| 2.2. Đặc tả các yêu cầu .....                     | 27        |
| 2.2.1. Yêu cầu chức năng .....                    | 27        |
| 2.2.2. Yêu cầu phi chức năng .....                | 33        |
| 2.3. Thiết kế dữ liệu .....                       | 40        |
| 2.3.1. Mô hình dữ liệu .....                      | 40        |
| 2.3.2. Danh sách các thực thể và mối kết hợp..... | 42        |
| 2.3.3. Chi tiết thực thể và mối kết hợp .....     | 42        |
| 2.4. Thiết kế giao diện .....                     | 47        |
| 2.4.1. Sơ đồ hệ thống .....                       | 47        |
| 2.4.2. Giao diện trang chủ người dùng .....       | 48        |
| 2.4.2.1. Mô tả chi tiết.....                      | 48        |
| 2.4.2.2. Cách bố cục .....                        | 49        |
| 2.4.2.4. Tóm lại.....                             | 50        |
| 2.4.3. Giao diện chức năng Đăng Ký.....           | 50        |
| 2.4.4. Giao diện chức năng Đăng Nhập.....         | 51        |
| 2.4.5. Giao diện chức năng Tìm kiếm.....          | 51        |
| 2.4.6. Kết chương.....                            | 52        |
| <b>CHƯƠNG 3. Cài đặt thực nghiệm .....</b>        | <b>53</b> |
| 3.1. Bộ dữ liệu thử nghiệm .....                  | 53        |
| 3.2. Kết quả thực nghiệm.....                     | 53        |
| 3.2.1. Chức năng Đăng Ký.....                     | 53        |
| 3.2.2. Chức năng Đăng Nhập.....                   | 53        |
| 3.2.3. Chức năng Tìm Kiếm.....                    | 53        |
| 3.2.4. Chức năng Khách .....                      | 53        |

|  |           |
|--|-----------|
| 3.3. Kết chương.....                               | 54        |
| <b>CHƯƠNG 4. Kết luận và hướng phát triển.....</b> | <b>55</b> |
| 4.1. Kết luận.....                                 | 55        |
| 4.2. Hướng phát triển .....                        | 55        |
| 4.2.1. Mở rộng bộ dữ liệu.....                     | 55        |
| 4.2.2. Tích hợp tính năng xã hội.....              | 55        |
| 4.2.3. Tối ưu hóa trải nghiệm di động.....         | 55        |
| 4.2.4. Phát triển tính năng mới .....              | 55        |
| 4.2.5. Tăng cường bảo mật.....                     | 56        |

## **DANH MỤC HÌNH ẢNH**

|   |    |
|---|----|
| Hình 1.1 Ngôn ngữ HTML .....                      | 4  |
| Hình 1.2 Cấu trúc cơ bản .....                    | 5  |
| Hình 1.3 Ngôn ngữ CSS.....                        | 14 |
| Hình 1.4 Ngôn ngữ Javascript.....                 | 18 |
| Hình 2.1 Sơ đồ hệ thống.....                      | 48 |
| Hình 2.2 Giao diện trang chủ của người dùng ..... | 48 |
| Hình 2.3 Giao diện chức năng Đăng Ký .....        | 50 |
| Hình 2.4 Giao diện chức năng Đăng Nhập .....      | 51 |
| Hình 2.5 Giao diện chức năng Tìm Kiếm .....       | 51 |



## **DANH MỤC BẢNG BIỂU**

|  |    |
|--|----|
| Bảng 2.1 Danh sách các thực thể và mối kết hợp ..... | 42 |
| Bảng 2.2 Chi tiết thực thể User .....                | 43 |
| Bảng 2.3 Chi tiết thực thể Artist .....              | 43 |
| Bảng 2.4 Chi tiết thực thể Album .....               | 44 |
| Bảng 2.5 Chi tiết thực thể Song .....                | 44 |
| Bảng 2.6 Chi tiết thực thể Playlist .....            | 45 |
| Bảng 2.7 Chi tiết thực thể PlaylistTrack .....       | 45 |
| Bảng 2.8 Chi tiết thực thể ListeningHist.....        | 45 |
| Bảng 2.9 Chi tiết thực thể Comment.....              | 46 |

## **TÓM TẮT ĐỒ ÁN CƠ SỞ MÔN HỌC**

### **Vấn đề nghiên cứu**

Nghiên cứu về thiết kế giao diện cho website nghe nhạc nhằm tạo ra một trải nghiệm người dùng thú vị, dễ nhìn và thân thiện, giúp thu hút và giữ chân người dùng. Điều này bao gồm việc sử dụng công nghệ thông tin để cải thiện hiệu suất tải trang và quản lý dữ liệu, tối ưu hóa tính năng tìm kiếm và đề xuất nhạc, cùng việc xây dựng tính năng tương tác như bình luận và chia sẻ để tạo cộng đồng người dùng.

### **Phương pháp nghiên cứu**

Nghiên cứu thị trường và phân tích người dùng:

Tiến hành khảo sát thị trường để hiểu về nhu cầu và sở thích của người dùng trong việc nghe nhạc trực tuyến.

Phân tích dữ liệu người dùng hiện có (nếu có) để đánh giá các thói quen nghe nhạc, thiên hướng và mong muốn của họ.

Nghiên cứu về trải nghiệm người dùng (UX Research):

Tiến hành phỏng vấn và nhóm tập trung để hiểu ý kiến và trải nghiệm của người dùng với các trang web nghe nhạc hiện tại.

Sử dụng các phương pháp như thử nghiệm người dùng để đánh giá hiệu quả của giao diện người dùng mới.

Phân tích thiết kế giao diện (UI/UX Design Analysis):

Đánh giá các trang web nghe nhạc phổ biến để nắm bắt các xu hướng thiết kế và phát triển.

Phân tích các yếu tố như cấu trúc trang, màu sắc, kiểu chữ, và trải nghiệm người dùng để xác định những điểm mạnh và yếu của các giao diện hiện tại.

Thử nghiệm A/B:

Sử dụng thử nghiệm A/B để so sánh hiệu suất của các phiên bản giao diện khác nhau, từ đó xác định xem các thay đổi như màu sắc, cấu trúc, hoặc tính năng mới có ảnh hưởng tích cực đến trải nghiệm người dùng hay không.

Nghiên cứu về tối ưu hóa công nghệ:

Nghiên cứu về các công nghệ mới như HTML5, CSS3, JavaScript để tối ưu hóa trang web và tăng trải nghiệm người dùng.

Tìm hiểu về các phương pháp lưu trữ và phát nhạc trực tuyến để cải thiện hiệu suất tải trang và quản lý dữ liệu.

### **Một số kết quả đạt được**

Biết cách sử dụng ngôn ngữ lập trình

Thiết kế một trang web nghe nhạc cơ bản

Giao diện người dùng thân thiện

Tăng cường trải nghiệm của người dùng

Tăng hiệu trang web

Biết cách trình bày báo cáo

### **MỞ ĐẦU**

#### **Lý do chọn đề tài:**

Tôi đã chọn đề tài "Thiết kế giao diện cho website nghe nhạc" vì nó kết hợp sở thích cá nhân của tôi với tiềm năng kinh doanh trong một lĩnh vực đầy sáng tạo và phát triển. Ngành công nghiệp âm nhạc hiện đại đang trải qua một giai đoạn phát triển mạnh mẽ, với người dùng ngày càng chuyển sang nghe nhạc trực tuyến. Việc có một trang web nghe nhạc thú vị, tiện lợi và dễ sử dụng không chỉ là một cơ hội kinh doanh mà còn là một phần không thể thiếu trong cuộc sống hàng ngày của nhiều người.

Đồng thời, việc nghiên cứu và phát triển website nghe nhạc cũng là một cơ hội để tôi áp dụng và phát triển kỹ năng trong lĩnh vực thiết kế giao diện, công nghệ web và SEO. Tôi tin rằng, qua việc tìm hiểu và thực hành trên dự án này, tôi sẽ có cơ hội nâng cao hiểu biết và kỹ năng của mình, đồng thời tạo ra một sản phẩm độc đáo và hấp dẫn cho cả người dùng và các nhà đầu tư.

Ngoài ra, đề tài này còn mang lại cơ hội để tôi khám phá những xu hướng mới trong thiết kế giao diện và công nghệ web, cũng như tạo ra một môi trường thú vị và sáng tạo cho các nghệ sĩ và người yêu âm nhạc. Tôi hy vọng rằng dự án này sẽ không chỉ đem lại những kết quả tích cực mà còn là một bước đệm quan trọng cho sự phát triển cá nhân và sự nghiệp của tôi trong tương lai.

## **CHƯƠNG 1. TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU**

### **1.1. Quy trình phát triển website**

#### **1.1.1 Thu thập yêu cầu và phân tích:**

Trước khi bắt đầu phát triển website, quá trình thu thập yêu cầu và phân tích là bước quan trọng nhằm đảm bảo rằng dự án sẽ đáp ứng được mục tiêu và mong muốn của khách hàng. Đây cũng là thời điểm để xác định rõ các yếu tố kỹ thuật và thiết kế. Quy trình này bao gồm:

- Gặp gỡ khách hàng để thảo luận và hiểu rõ yêu cầu.
- Xác định mục tiêu của dự án, đối tượng người dùng, và các yếu tố kỹ thuật như ngôn ngữ lập trình, cơ sở dữ liệu, và tính năng đặc biệt.
- Phân tích yêu cầu để hiểu chi tiết hơn về tính năng, luồng công việc, và giao diện người dùng.

#### **1.1.2. Lập kế hoạch:**

Sau khi yêu cầu đã được thu thập và phân tích, quy trình lập kế hoạch bắt đầu. Ở bước này, chúng ta xác định các công nghệ và quy trình sẽ được sử dụng, cũng như lập ra kế hoạch thời gian và ngân sách. Cụ thể:

- Xác định các công nghệ phù hợp như ngôn ngữ lập trình, framework, hệ quản trị cơ sở dữ liệu.

- Lập kế hoạch sơ đồ trang web và cơ sở dữ liệu để thiết kế cấu trúc tổ chức hợp lý.

- Đặt ra mục tiêu về thời gian hoàn thành và ngân sách dự kiến.

#### 1.1.3. Thiết kế:

Sau khi lập kế hoạch, chúng ta tiến hành thiết kế giao diện và trải nghiệm người dùng của trang web. Quy trình này bao gồm:

- Tạo wireframe và mockup để hiển thị cấu trúc và bố trí của trang web.
- Phát triển thiết kế đồ họa với màu sắc, font chữ, hình ảnh và biểu tượng phù hợp với yêu cầu và thương hiệu của khách hàng.

#### 1.1.4. Phát triển:

Bước phát triển là quá trình thực hiện các yêu cầu kỹ thuật và thiết kế đã được xác định ở các bước trước đó. Quy trình này bao gồm:

- Lập trình trang web sử dụng ngôn ngữ lập trình và framework đã chọn.
- Xây dựng tính năng và chức năng theo yêu cầu, bao gồm cả tương tác với cơ sở dữ liệu và các API nếu cần.
- Kiểm thử và sửa lỗi liên tục để đảm bảo tính ổn định và chất lượng của trang web.

#### 1.1.5. Kiểm thử:

Sau khi phát triển xong, trang web cần được kiểm tra kỹ lưỡng để đảm bảo hoạt động một cách chính xác và mượt mà trên mọi nền tảng và trình duyệt. Quy trình này bao gồm:

- Kiểm tra tính năng, hiệu suất và khả năng tương thích trên nhiều trình duyệt và thiết bị.
- Tiến hành kiểm thử thủ công và tự động để phát hiện và sửa các lỗi.

#### 1.1.6. Triển khai và triển khai:

Sau khi đã hoàn thành và kiểm tra kỹ lưỡng, trang web sẽ được triển khai để công chúng có thể truy cập. Quy trình này bao gồm:

- Đưa trang web lên máy chủ hoặc dịch vụ lưu trữ phù hợp.
- Kiểm tra lại tính năng và khả năng hoạt động trước công chúng trước khi triển khai chính thức.

#### 1.1.7. Bảo trì và cập nhật:

Sau khi trang web đã được triển khai, quá trình bảo trì và cập nhật sẽ tiếp tục để đảm bảo tính ổn định và hiệu suất của nó. Quy trình này bao gồm:

- Theo dõi hiệu suất của trang web và sửa các lỗi phát sinh.

- Cập nhật nội dung và tính năng theo yêu cầu của khách hàng hoặc thị trường để đảm bảo trang web luôn cập nhật và phát triển theo thời gian.

Bằng cách này, quy trình phát triển website được thực hiện một cách có hệ thống và tổ chức để đảm bảo rằng dự án đáp ứng được mục tiêu và mong muốn của khách hàng.

## **1.2. Ngôn ngữ HTML**



*Hình 1.1 Ngôn ngữ HTML*

### **Giới thiệu về ngôn ngữ HTML**

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu được sử dụng để tạo cấu trúc và định dạng nội dung trên các trang web. Được phát triển bởi Tim Berners-Lee vào năm 1990, HTML đã trở thành tiêu chuẩn cho việc phát triển các trang web trên internet. HTML không phải là ngôn ngữ lập trình



mà là ngôn ngữ đánh dấu, nó chỉ định cấu trúc và các thành phần của một trang web, bao gồm văn bản, hình ảnh, liên kết, biểu mẫu và nhiều hơn nữa.

Các thành phần cơ bản của một cấu trúc HTML

Một tài liệu HTML cơ bản bao gồm các thành phần sau:

`<!DOCTYPE html>`: Khai báo loại tài liệu HTML và phiên bản HTML được sử dụng.

`<html>`: Phần bao quanh toàn bộ nội dung của trang web và đặt trong thẻ `<html>`

+ `lang = "en"` : Thuộc tính lang xác định ngôn ngữ của tài liệu.

`<head>`: Chứa các thông tin meta, tiêu đề, và các tệp CSS, JavaScript và các thẻ khác liên quan đến tài liệu HTML.

`<body>`: Chứa nội dung hiển thị trên trình duyệt.

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>Tên Tiêu Đề</title>
5      <meta charset="utf-8" />
6    </head>
7    <body>
8      <!-- Nội dung trang -->
9    </body>
10 </html>
```

*Hình 1.2 Cấu trúc cơ bản*

### *1.2.1. Các thẻ cơ bản trong ngôn ngữ HTML*

- Thẻ `<p>` : dùng để định nghĩa một đoạn văn bản.

`<p>This is a paragraph.</p>`

- Thẻ tiêu đề (heading) trong HTML được sử dụng để xác định tiêu đề của một phần của trang web. HTML cung cấp 6 thẻ tiêu đề, từ <h1> đến <h6>, mỗi thẻ có mức độ quan trọng giảm dần từ <h1> đến <h6>

```
- <h1>This is a heading level 1</h1>
- <h2>This is a heading level 2</h2>
- <h3>This is a heading level 3</h3>
- <h3>This is a heading level 4</h4>
- <h4>This is a heading level 5</h5>
- <h5>This is a heading level 6</h6>
```

+ <h1> - <h6>: Các thẻ này đại diện cho các mức độ tiêu đề khác nhau. <h1> là tiêu đề lớn nhất và có mức độ quan trọng cao nhất, trong khi <h6> là tiêu đề nhỏ nhất và có mức độ quan trọng thấp nhất.

+ Sử dụng: Thẻ tiêu đề thường được sử dụng để định vị và phân loại nội dung trang web. <h1> thường được sử dụng cho tiêu đề chính của trang web, trong khi các tiêu đề nhỏ hơn thường được sử dụng cho các phần tiêu đề phụ và tiêu đề của các phần.

+ SEO (Search Engine Optimization): Các công cụ tìm kiếm như Google thường đánh giá tiêu đề để hiểu nội dung của trang web. Do đó, việc sử dụng các thẻ tiêu đề phù hợp và có ý nghĩa có thể giúp cải thiện vị trí của trang web trong kết quả tìm kiếm.

+ CSS và Styling: Thẻ tiêu đề có thể được tùy chỉnh bằng CSS để thay đổi màu sắc, kích thước và kiểu chữ.

+ Phong cách thêm: Các tiêu đề có thể được cải thiện với sự hỗ trợ của các thư viện CSS và frameworks như Bootstrap, Materialize CSS, etc., để tạo ra các hiệu ứng và kiểu dáng thú vị hơn.

- Thẻ `<strong>`: được sử dụng để in đậm văn bản

```
<p><strong>This text is bold.</strong></p>
```

- Thẻ `<a>`: Dùng để định nghĩa một liên kết.

```
<a href="https://www.example.com">Click here</a>
```

- \*URL tuyệt đối so với URL tương đối:

- Cả hai ví dụ trên đều sử dụng URL tuyệt đối (địa chỉ web đầy đủ) trong thuộc tính href.

- Liên kết cục bộ (liên kết đến một trang trong cùng một trang web) được chỉ định bằng URL tương đối (không có phần "https://www"):

- Ví dụ:

```
<h2>URL tuyệt đối:</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>
```

```
<h2>URL tương đối:</h2>
<p><a href="html_images.asp">Ảnh HTML</a></p>
<p><a href="/css/default.asp">CSS cơ bản</a></p>
```

- \*Sử dụng hình ảnh làm liên kết:

- Để sử dụng hình ảnh làm liên kết, chỉ cần đặt thẻ `<img>` bên trong thẻ `<a>`:

- Ví dụ:

```
<a href="default.asp">

</a>
```

Liên kết HTML có thể được sử dụng để tạo dấu trang để người đọc có thể chuyển đến các phần cụ thể của trang web.

Thẻ `<img>`: Dùng để chèn một hình ảnh vào web.

Hình ảnh không được chèn vào trang web về mặt kỹ thuật; hình ảnh được liên kết đến các trang web. Thẻ `<img>` tạo một không gian lưu giữ cho hình ảnh được tham chiếu.

Thẻ `<img>` là thẻ trống, chỉ chứa các thuộc tính và không có thẻ đóng.

Thẻ `<img>` có hai thuộc tính bắt buộc:

+ `src` - Chỉ định đường dẫn đến hình ảnh.

+ `alt` - Chỉ định văn bản thay thế cho hình ảnh.

```

```

- Lưu ý: Khi một trang web tải, tại thời điểm đó, trình duyệt sẽ lấy hình ảnh từ máy chủ web và chèn nó vào trang. Do đó, hãy đảm bảo rằng hình ảnh thực sự ở cùng một vị trí so với trang web, nếu không khách truy cập của bạn sẽ nhận được biểu tượng liên kết bị hỏng. Biểu tượng liên kết bị hỏng và `alt` văn bản được hiển thị nếu trình duyệt không tìm thấy hình ảnh.

- Ví dụ:

```

```

\*Hình ảnh trong thư mục khác:

- Nếu bạn có hình ảnh của mình trong thư mục con, bạn phải đưa tên thư mục vào thuộc tính `src`:

- Ví dụ:

```

```

\*Hình ảnh trên Máy chủ/Trang web khác:

- Một số trang web trỏ đến một hình ảnh trên một máy chủ khác.
- Để trỏ đến hình ảnh trên máy chủ khác, bạn phải chỉ định URL tuyệt đối (đầy đủ) trong thuộc tính src:

- Ví dụ:

```

```

- Thẻ <form>: Dùng để định nghĩa một biểu mẫu để nhập dữ liệu từ người dùng.

```
<form action="/submit-form" method="POST">  
  <input type="text" name="username"  
placeholder="Username">  
  <input type="password" name="password"  
placeholder="Password">  
  <input type="submit" value="Submit">  
</form>
```

- Thẻ <input>: Dùng để tạo ra một ô nhập liệu trong biểu mẫu.

```
<input type="text" name="username"  
placeholder="Username">
```

- Thẻ <div>: Dùng để định nghĩa một khối dữ liệu nguyên vẹn.

```
<div>This is a division.</div>
```

- Thẻ <span>: Dùng để định nghĩa một phần nhỏ của văn bản.

```
<p>This is <span>highlighted</span> text.</p>
```

- Thẻ `<ul>`, `<ol>`, `<li>`: Dùng để định nghĩa danh sách có thứ tự `<ol>`, danh sách không có thứ tự `<ul>` và một mục trong danh sách `<li>`.

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>

<ol>
  <li>Item 1</li>
  <li>Item 2</li>
</ol>
```

Danh sách mô tả: thẻ `<dl>`, `<dt>`, `<dd>`.

+ HTML cũng hỗ trợ danh sách mô tả.

+ Danh sách mô tả là danh sách các thuật ngữ, kèm theo mô tả cho từng thuật ngữ.

+ Thẻ `<dl>` xác định danh sách mô tả,

+ Thẻ `<dt>` xác định thuật ngữ (tên)

+ Thẻ `<dd>` mô tả từng thuật ngữ:

\* Ví dụ:

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Thẻ `<table>`: Dùng để định nghĩa một bảng.

+ Thẻ `<tr>` : Dùng để định nghĩa một hàng trong bảng.

+ Thẻ `<td>` : Dùng để định một cột trong bảng.

```
<table>
  <tr>
    <td>Row 1, Cell 1</td>
    <td>Row 1, Cell 2</td>
  </tr>
  <tr>
    <td>Row 2, Cell 1</td>
    <td>Row 2, Cell 2</td>
  </tr>
</table>
```

**\*Colspan:**

- Để tạo một ô trải dài trên nhiều cột, hãy sử dụng thuộc tính `colspan`:

- Ví dụ:

```
<table>
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>43</td>
  </tr>
  <tr>
    <td>Eve</td>
```

```
<td>Jackson</td>
<td>57</td>
</tr>
</table>
```

**\*Rowspan:**

- Để tạo một ô trải dài trên nhiều hàng, hãy sử dụng thuộc tính rowspan:

- Ví dụ:

```
<table>
<tr>
  <th>Name</th>
  <td>Jill</td>
</tr>
<tr>
  <th rowspan="2">Phone</th>
  <td>555-1234</td>
</tr>
<tr>
  <td>555-8745</td>
</tr>
</table>
```

**Thẻ <br>:** Dùng để tạo ra một dòng mới trong văn bản.

```
<p>This is the first line.<br>This is the second
line.</p>
```

**Thẻ <hr>:** Dùng để tạo ra một đường kẻ ngang để phân chia nội dung.

```
<p>This is a paragraph.</p>
<hr>
<p>This is another paragraph.</p>
```



Thẻ `<iframe>`: Cho phép nhúng một trang web khác vào trang hiện tại.

```
<iframe src="https://www.example.com"></iframe>
```

Thẻ `<audio>`: Dùng để chèn một phần tử âm thanh vào trang web.

```
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

Thẻ `<video>`: Dùng để một phần tử video vào trang web.

```
<video controls width="320" height="240">
  <source src="video.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

Thẻ `<footer>`: Dùng để định nghĩa chân trang của trang web.

```
<footer>
  <p>&copy; 2024 Example Footer</p>
</footer>
```

### **\*\*Kết luận\*\***

HTML là một ngôn ngữ cơ bản nhưng mạnh mẽ, với nhiều thẻ và thuộc tính khác nhau cho phép bạn xây dựng các trang web đa dạng và phức tạp. Điều này chỉ là một số thẻ cơ bản, trong thực tế còn rất nhiều thẻ và thuộc tính khác để khám phá và sử dụng.

### **1.3. Bảng định kiểu CSS**



*Hình 1.3 Ngôn ngữ CSS*

#### *1.3.1. Giới thiệu về bảng định kiểu CSS*

CSS (Cascading Style Sheets) là ngôn ngữ dùng để mô tả cách mà các phần tử HTML được hiển thị trên trình duyệt. Bằng cách sử dụng CSS, bạn có thể quản lý hiệu ứng, màu sắc, kích thước và vị trí của các phần tử trên trang web. Bảng Định Kiểu CSS là một phần quan trọng của việc phát triển trang web, giúp tạo ra giao diện hấp dẫn và dễ đọc.

Các thành phần cơ bản của bảng định kiểu CSS

Bảng Định Kiểu CSS bao gồm các quy tắc và khai báo để áp dụng cho các phần tử HTML. Cấu trúc cơ bản của một quy tắc CSS bao gồm:

Chọn phần tử (Selector): Phần này xác định phần tử HTML nào sẽ áp dụng các quy tắc CSS.

Thuộc tính (Property): Định nghĩa các thuộc tính mà bạn muốn thay đổi cho phần tử được chọn.

Giá trị (Value): Xác định giá trị cụ thể của thuộc tính.

### 1.3.2. Các loại CSS

#### 1.3.2.1. Inline CSS

- CSS nội tuyến được viết trực tiếp trong thẻ HTML, sử dụng thuộc tính style.

Loại CSS này được sử dụng để định dạng các phần tử HTML cụ thể. Ví dụ:

```
<p style="color: red; font-size: 20px;">Đây là một đoạn  
văn bản được định dạng bằng CSS nội tuyến.</p>
```

#### 1.3.2.2. Internal CSS

- CSS nội bộ được viết trong thẻ <style> nằm trong phần <head> của tài liệu HTML. Loại CSS này được sử dụng để định dạng nhiều phần tử HTML trong cùng một trang web. Ví dụ:

```
<head>  
  <style>  
    p {  
      color: red;  
      font-size: 20px;  
    }  
  </style>  
</head>
```

#### 1.3.2.3. External CSS

- CSS bên ngoài được viết trong một tệp riêng có phần mở rộng .css. Tệp này được liên kết với tài liệu HTML bằng thẻ <link> trong phần <head> của tài liệu HTML. Loại CSS này được sử dụng để định dạng nhiều trang web trong cùng một dự án. Ví dụ:

```
<head>
```

```
<link rel="stylesheet" href="style.css">
</head>
```

### **1.3.3. Các quy tắc về CSS cơ bản**

- Chọn phần tử để đổi màu chữ: Quy tắc này áp dụng cho hầu hết tất cả các thẻ và đổi màu của chữ theo lệnh.

```
p {
    color: blue;
}
```

- Chọn phần tử để thay đổi kích thước văn bản: Quy tắc này áp dụng cho hầu hết tất cả các thẻ và đặt kích thước theo lệnh.

```
h1 {
    font-size: 24px;
}
```

- Chọn phần tử để căn giữa văn bản: quy tắc này áp dụng cho tất cả các phần tử có lớp center và căn giữa văn bản.

```
.center {
    text-align: center;
}
```

- Thay đổi nền và kích thước đường nền: Quy tắc này áp dụng cho tất cả các phần tử mà ta muốn thay đổi, thay đổi màu nền và kích thước đường viền theo lệnh.

```
.box {  
  background-color: #f0f0f0;  
  border: 1px solid #ccc;  
}
```

- Chọn phân tử và thay đổi font: Quy tắc này áp dụng cho toàn bộ nội dung của trang và đặt font chữ là Arial hoặc font không-serif mặc định.

```
body {  
  font-family: Arial, sans-serif;  
}
```

- Chọn phân tử và thay đổi màu nền khi di chuột qua: Quy tắc này áp dụng cho tất cả các liên kết khi di chuột qua và thay đổi màu nền thành màu vàng (#ffff00)

```
a:hover {  
  background-color: #ffff00;  
}
```

### **\*\*Kết Luận\*\***

CSS là một công cụ mạnh mẽ cho phép bạn tùy chỉnh giao diện của trang web một cách linh hoạt và dễ dàng. Với các quy tắc CSS phù hợp, bạn có thể tạo ra giao diện trực quan và thú vị cho người dùng.

## **1.4. Ngôn ngữ Javascript**

### *1.4.1. Giới thiệu về Javascript*



*Hình 1.4 Ngôn ngữ Javascript*

JavaScript là một ngôn ngữ lập trình phía client mạnh mẽ được sử dụng rộng rãi trong phát triển web. Ngôn ngữ này ra đời vào những năm 1990 và từ đó đã trở thành một công cụ không thể thiếu cho việc tạo ra các trang web động và tương tác. JavaScript cho phép thêm tính năng động, xử lý sự kiện, thay đổi nội dung và tương tác với người dùng một cách linh hoạt và hiệu quả.

### *1.4.2. Khái niệm về Javascript*

- JavaScript là ngôn ngữ lập trình bổ sung tính tương tác và hành vi tùy chỉnh cho các trang web của chúng tôi. Nó là ngôn ngữ kịch bản phía máy khách, có nghĩa là nó chạy trên máy của người dùng chứ không phải trên máy chủ, như các ngôn ngữ lập trình web khác như PHP và Ruby.

- Điều đó có nghĩa là JavaScript, JavaScript cũng được biết đến như một ngôn ngữ lập trình động và được gõ lỏng lẻo. JavaScript là một ngôn ngữ kịch bản nhẹ nhưng cực kỳ mạnh mẽ. Chúng ta thường gặp nó nhất thông qua trình

duyet của mình, nhưng JavaScript đã xâm nhập vào mọi thứ, từ ứng dụng gốc, tệp PDF đến sách điện tử.

- Thông thường chúng ta sẽ gặp JavaScript như một cách để thêm tính tương tác vào một trang.

- Tất cả các thành phần, thuộc tính và văn bản trên một trang web đều có thể được truy cập bằng các tập lệnh sử dụng DOM.

- JavaScript cho phép bạn tạo các giao diện có độ phản hồi cao để cải thiện trải nghiệm người dùng và cung cấp chức năng động mà không cần đợi máy chủ tải trang mới.

#### *1.4.3. Các thành phần cơ bản của Javascript*

Javascript bao gồm các khối cơ bản sau:

**Biến (Variables):** Biến trong JavaScript được sử dụng để lưu trữ và tham chiếu đến dữ liệu. Các biến có thể là số, chuỗi, đối tượng hoặc các giá trị khác.

**Hàm (Functions):** Hàm là một khối mã được đặt tên và được sử dụng để thực thi một tập hợp các lệnh cụ thể. JavaScript hỗ trợ khai báo hàm và hàm vô danh.

**Đối tượng (Objects):** Trong JavaScript, đối tượng là một tập hợp của các thuộc tính (properties) và phương thức (methods). Các đối tượng giúp tổ chức và quản lý mã và dữ liệu một cách có tổ chức.

**Điều kiện (Conditional Statements):** Các câu lệnh điều kiện trong JavaScript cho phép bạn thực hiện mã khác nhau dựa trên điều kiện được xác định. Các câu lệnh điều kiện bao gồm if, else, else if, switch.

**Vòng lặp (Loops):** Vòng lặp cho phép bạn lặp lại một khối mã nhiều lần. JavaScript hỗ trợ các vòng lặp như for, while, do...while.

#### 1.4.4. Javascript cơ bản

- Biến (Variable)

```
var message = "Hello, world!";
```

⇒ Biến ‘message’ được khai báo để lưu trữ chuỗi “Hello, world”

- Hàm (Function)

```
function greet(name) {  
    return "Hello, " + name + "!";  
}
```

⇒ Hàm ‘greet’ nhận một tham số ‘name’ và trả về chuỗi ‘Hello’

- Đối tượng (Objects)

```
var person = {  
    name: "John",  
    age: 30,  
    greet: function() {  
        return "Hello, my name is " + this.name + ".";  
    }  
};
```

⇒ Đối tượng ‘person’ có thuộc tính ‘name’ và ‘age’, cùng một phương thức ‘greet’ để trả về ‘Hello, my name is’.

- Điều kiện (Conditinal Statements)

```
var x = 10;  
if (x > 5) {  
    console.log("x is greater than 5");  
}
```



```
} else {  
    console.log("x is less than or equal to 5");  
}
```

⇒ Kiểm tra giá trị của biến ‘x’ và in ra thông báo tườn ứng.

- Vòng lặp (Loops)

```
for (var i = 0; i < 5; i++) {  
    console.log("The value of i is: " + i);  
}
```

⇒ Sử dụng vòng lặp ‘for’ để in ra giá trị của biến ‘i’ từ 0 đến 4.

**\*\*Kết Luận\*\***

JavaScript là một ngôn ngữ linh hoạt và mạnh mẽ, cho phép bạn tạo ra các tính năng tương tác và động trên trang web một cách dễ dàng. Sử dụng JavaScript, bạn có thể tạo ra trải nghiệm người dùng tốt hơn và làm cho trang web của bạn trở nên linh hoạt và thú vị hơn.

## **1.5. Kết chương**

### *1.5.1. Quy trình phát triển Website*

Quy trình phát triển website là một hành trình phức tạp từ ý tưởng ban đầu đến sản phẩm hoàn chỉnh. Dưới đây là một cái nhìn chi tiết về các bước trong quy trình này:

- Xác định yêu cầu:

Trước khi bắt đầu phát triển, bạn cần hiểu rõ yêu cầu của dự án từ khách hàng. Điều này bao gồm xác định mục tiêu của trang web, đối tượng người dùng, chức năng và yêu cầu kỹ thuật. Bạn cũng cần thu thập thông tin về nội dung, hình ảnh và các yếu tố thiết kế khác.

- Lập kế hoạch:

Sau khi hiểu rõ yêu cầu, bạn cần lập kế hoạch để đảm bảo dự án diễn ra một cách hiệu quả. Bước này bao gồm xác định phạm vi dự án, lên kế hoạch thời gian và ngân sách. Bạn cũng cần tạo sơ đồ trang (sitemap) để xác định cấu trúc của trang web và thiết kế giao diện người dùng (UI/UX) để đảm bảo trải nghiệm người dùng tốt nhất.

- Thiết kế:

Ở bước này, bạn sẽ tạo ra các thiết kế dựa trên yêu cầu và kế hoạch đã xác định. Đầu tiên, bạn có thể tạo wireframe và mockup để mô phỏng trang web. Sau đó, bạn sẽ thiết kế đồ họa, chọn màu sắc và font chữ phù hợp với thương hiệu của khách hàng. Đảm bảo rằng các thiết kế của bạn là responsive, tức là hiển thị tốt trên mọi thiết bị.

- Phát triển:

Ở bước này, bạn sẽ bắt đầu mã hóa trang web dựa trên các thiết kế đã được phê duyệt. Sử dụng HTML, CSS và JavaScript để tạo cấu trúc, kiểu dáng và tính năng cho trang web. Bạn cũng có thể cần tạo cơ sở dữ liệu và kết nối nó với trang web nếu cần thiết. Làm việc một cách kiên nhẫn và tỉ mỉ để đảm bảo rằng trang web hoạt động như mong đợi và tương thích trên mọi trình duyệt và thiết bị.

- Kiểm thử và Đánh giá:

Sau khi phát triển xong, bạn cần kiểm tra và đánh giá trang web để đảm bảo tính ổn định, bảo mật và hiệu suất. Thực hiện kiểm thử tự động và thủ công để tìm lỗi và sửa chúng. Thu thập phản hồi từ người dùng và điều chỉnh trang web dựa trên đó.

- Triển khai và duy trì:

Cuối cùng, sau khi hoàn thành, bạn sẽ triển khai trang web lên một máy chủ hoặc nền tảng hosting để có thể truy cập từ mọi nơi trên thế giới. Đảm bảo sử dụng các công cụ theo dõi hiệu suất và bảo trì để duy trì và cập nhật trang web theo nhu cầu và đảm bảo rằng nó hoạt động một cách mượt mà và an toàn.

#### *1.5.2. Ngôn ngữ HTML*

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu cơ bản của web, được sử dụng để xây dựng cấu trúc của trang web. Dưới đây là một số điểm chi tiết về HTML:

HTML xác định cấu trúc của trang web bằng cách sử dụng các thẻ và thuộc tính.

Các thẻ HTML được sử dụng để định nghĩa các phần tử trên trang web, như văn bản, hình ảnh, liên kết, biểu mẫu, v.v.

Các thuộc tính HTML được sử dụng để cung cấp thông tin bổ sung về các phần tử, chẳng hạn như màu sắc, kích thước và liên kết.

HTML cung cấp một cơ sở vững chắc cho việc xây dựng các trang web hiệu quả và có cấu trúc.

#### *1.5.3. Ngôn ngữ JavaScript*

JavaScript là ngôn ngữ lập trình phía client mạnh mẽ, được sử dụng để tạo ra các tính năng động và tương tác trên trang web. Dưới đây là một số điểm chi tiết về JavaScript:

JavaScript cho phép thêm các hiệu ứng tương tác như đổi màu, di chuyển, và hiển thị/ẩn nội dung.

Nó được sử dụng để xử lý sự kiện như click chuột, hover, và submit biểu mẫu.

## **CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

### **2.1. Mô tả bài toán**

Báo cáo: Xây dựng trang web nghe nhạc

#### *2.1.1. Giới thiệu*

Trong thời đại số hóa ngày nay, việc xây dựng một trang web nghe nhạc là một trong những dự án phổ biến và hấp dẫn. Trang web này cung cấp một nền tảng cho người dùng để thưởng thức và tìm kiếm những bài hát yêu thích của họ, đồng thời cung cấp trải nghiệm nghe nhạc trực tuyến tiện lợi.

#### *2.1.2. Phân tích yêu cầu*

##### *2.1.2.1. Chức năng:*

- Hiển thị danh sách các bài hát với ảnh bìa và tên.
- Cho phép người dùng nhấp vào mỗi bài hát để nghe.
- Cung cấp liên kết đến các trang khác như Trang Chủ, Tìm Kiếm, Đăng Ký và Đăng Nhập.

##### *2.1.2.2. Giao diện:*

- Thiết kế giao diện đẹp mắt và dễ sử dụng.
- Sử dụng Bootstrap để đảm bảo trang web hiển thị một cách thân thiện với nhiều loại thiết bị.

#### *2.1.3. Thiết kế và triển khai*

##### *2.1.3.1. Header (Navbar):*

- Sử dụng thẻ ``<nav>`` để tạo navbar chứa các liên kết.

- Các liên kết bao gồm Trang Chủ, Tìm Kiếm, Đăng Ký, Đăng Nhập và một liên kết dành cho khách.

#### 2.1.3.2. Danh sách các bài hát:

- Sử dụng lưới Bootstrap để chia cột và hiển thị danh sách các bài hát.
- Mỗi bài hát được đặt trong một thẻ `<div class="col-sm-3">`.
- Ảnh bìa và tên bài hát được đặt trong thẻ `<a>`.
- Khi người dùng nhấp vào một bài hát, họ sẽ được chuyển hướng đến trang chi tiết của bài hát đó.

#### 2.1.3.3. Footer:

- Tương tự như header nhưng có một số thay đổi về màu sắc và chứa thông tin liên hệ.

#### 2.1.4. Cải tiến

Bên cạnh những tính năng cơ bản, có thể thực hiện một số cải tiến sau:

- Tối ưu hóa hiệu suất: Đảm bảo trang web hoạt động mượt mà và nhanh chóng trên mọi thiết bị.
- Tính năng tìm kiếm và phân loại: Cho phép người dùng tìm kiếm bài hát theo tiêu chí như nghệ sĩ, thể loại...
- Mô tả chi tiết về bài hát: Thêm thông tin về nghệ sĩ, album, năm phát hành...
- Đăng ký và đăng nhập: Cho phép người dùng tạo tài khoản để lưu danh sách phát, đánh giá bài hát hoặc tạo playlist cá nhân.

### 2.1.5. Kết luận

Trang web nghe nhạc không chỉ là một dự án phổ biến mà còn mang lại trải nghiệm thú vị cho người dùng. Với sự kết hợp giữa thiết kế giao diện hấp dẫn và tính năng linh hoạt, trang web này có thể thu hút nhiều người sử dụng và trở thành một điểm đến lý tưởng cho người yêu âm nhạc.

## 2.2. Đặc tả các yêu cầu

### 2.2.1. Yêu cầu chức năng

#### 2.2.1.1. Hiển thị các danh sách bài hát

Trang web cần hiển thị danh sách các bài hát với thông tin cơ bản như tên và ảnh bìa.

Mỗi bài hát phải có một liên kết để người dùng có thể chuyển đến trang chi tiết của bài hát đó.

Ví dụ, trong đoạn mã HTML dưới đây, các bài hát được hiển thị trong các cột sử dụng lưới Bootstrap. Mỗi cột chứa một ảnh bìa và tên bài hát, đồng thời đặt trong thẻ `<a>` để tạo liên kết đến trang chi tiết của từng bài hát.

#### **Html**

```
<div class="container-fluid bg-3 text-center">
  <div class="row">
    <div class="col-sm-3">
      <a href="chotrongai.html">
        
      </a>
      <a href="chotrongai.html">Chờ Trông ai - HỒ PHI NAL</a>
    </div>
    <!-- Các cột khác -->
```

```
</div>  
</div>
```

#### 2.2.1.2. Trang chi tiết bài hát:

Trang chi tiết bài hát cần cung cấp thông tin chi tiết về bài hát bao gồm tên bài hát, nghệ sĩ, album, năm phát hành, và mô tả.

Người dùng cũng cần có khả năng nghe bài hát trực tuyến.

Ví dụ, trang chi tiết bài hát có thể được thiết kế như sau:

```
html  
<div class="container">  
  <h2>Chờ Trong ai - HỒ PHI NAL</h2>  
  <p>Nghệ sĩ: HỒ PHI NAL</p>  
  <p>Album: [Tên album]</p>  
  <p>Năm phát hành: [Năm]</p>  
  <p>Mô tả: [Mô tả chi tiết về bài hát]</p>  
  <!-- Thẻ audio để người dùng nghe bài hát -->  
  <audio controls>  
    <source src="chotrongai.mp3" type="audio/mpeg">  
    Your browser does not support the audio element.  
  </audio>  
</div>
```

#### 2.2.1.3. Thanh điều hướng:

Thanh điều hướng (navbar) cần chứa các liên kết đến các trang khác như Trang Chủ, Tìm Kiếm, Đăng Ký, và Đăng Nhập.



Liên kết phải dẫn đến các trang tương ứng để người dùng có thể trải nghiệm tính năng một cách dễ dàng.

Ví dụ:

```
html
<nav>
  <ul>
    <li><a href="index.html">Trang Chủ</a></li>
    <li><a href="timkim.html">Tìm Kiếm</a></li>
    <li><a href="DKTK.html">Đăng Ký</a></li>
    <li><a href="DN.html">Đăng Nhập</a></li>
    <li><a href="#">Khách</a></li>
  </ul>
</nav>
```

#### 2.2.1.4. Tìm kiếm:

Trang web cần cung cấp một ô tìm kiếm để người dùng có thể tìm kiếm bài hát theo tên, nghệ sĩ, hoặc thể loại.

Kết quả tìm kiếm phải được hiển thị một cách rõ ràng và dễ đọc.

Ví dụ:

```
html
<form action="/search" method="get">
  <input type="text" placeholder="Tìm kiếm..."
  name="search">
  <button type="submit">Tìm</button>
</form>
```

### 2.2.1.5 Đăng nhập

Đăng nhập là quá trình cho phép người dùng xác thực danh tính của mình trước khi truy cập vào một hệ thống hoặc ứng dụng. Quá trình này thường bao gồm việc cung cấp thông tin đăng nhập như tên người dùng và mật khẩu, sau đó hệ thống sẽ kiểm tra thông tin này để xác định xem người dùng có quyền truy cập hay không.

Ví dụ:

```
<body >
  <div id="wrapper">
    <form action="" id="form-login">
      <h1 class="form-heading"> Đăng nhập</h1>
      <div class="form-group">
        <i class="far fa-user"></i>
        <input type="text" class="form-input"
placeholder="Tên đăng nhập">
      </div>
      <div class="form-group">
        <i class="fas fa-key"></i>
        <input type="password" class="form-
input" placeholder="Mật khẩu">
        <div id="eye">
          <i class="far fa-eye"></i>
        </div>
      </div>
      <input type="submit" value="Đăng nhập"
class="form-submit">
    </form>
  </div>
</body>
```

```
<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
<script src="js/app.js"></script>
```

#### 2.2.1.6. Đăng ký tài khoản

Đăng ký tài khoản là quá trình cho phép người dùng tạo ra một hồ sơ cá nhân trên một trang web, ứng dụng hoặc hệ thống. Quá trình này thường bao gồm việc cung cấp thông tin cá nhân như tên, địa chỉ email, mật khẩu, và có thể là các thông tin khác như địa chỉ, số điện thoại, ngày sinh, v.v.

Ví dụ:

```
<body >
  <div id="wrapper">
    <form action="" id="form-login">
      <h1 class="form-heading"> Đăng ký</h1>
      <div class="form-group">
        <i class="far fa-user"></i>
        <input type="text" class="form-input"
placeholder="Tên tài khoản ">
      </div>
      <div class="form-group">
        <i class="fas fa-key"></i>
        <input type="password" class="form-
input" placeholder="Mật khẩu">
        <div id="eye">
          <i class="far fa-eye"></i>
        </div>
      </div>
      <div class="form-group">
        <i class="fas fa-key"></i>
```

```
        <input type="password" class="form-  
input" placeholder="Xác nhận mật khẩu">  
        <div id="eye">  
            <i class="far fa-eye"></i>  
        </div>  
    </div>  
    <input type="submit" value="Đăng ký tài  
khoản " class="form-submit">  
    </form>  
</div>  
  
</body>  
<script src="https://code.jquery.com/jquery-  
3.6.0.js"></script>  
<script src="js/app.js"></script>
```

#### 2.2.1.7. Khách

Tính năng "khách" trong một trang web thường đề cập đến chế độ xem hoặc trạng thái mà người dùng không cần phải đăng nhập để truy cập nội dung hoặc chức năng cụ thể. Điều này cho phép người dùng truy cập các phần của trang web mà không cần tạo tài khoản hoặc đăng nhập.

Dưới đây là một số khái niệm cơ bản về tính năng "khách" trên một trang web:

- Truy cập nội dung cơ bản: Người dùng có thể xem các nội dung cơ bản như trang chủ, thông tin sản phẩm hoặc dịch vụ mà không cần phải đăng nhập.

- Xem danh sách sản phẩm hoặc dịch vụ: Người dùng có thể duyệt qua danh sách các mục sản phẩm hoặc dịch vụ mà không cần phải tạo tài khoản.
- Xem thông tin chung: Thông tin chung như về chúng tôi, liên hệ hoặc câu hỏi thường gặp có thể được truy cập mà không cần đăng nhập.
- Truy cập trang đăng nhập và đăng ký: Người dùng có thể truy cập trang đăng nhập và đăng ký để tạo tài khoản mới hoặc đăng nhập vào tài khoản hiện có.
- Không thể truy cập các tính năng hoặc nội dung đặc biệt: Một số tính năng hoặc nội dung như quản lý đơn hàng, quản lý thông tin cá nhân hoặc tham gia cộng đồng có thể yêu cầu người dùng đăng nhập.

*Ví dụ:* trang web nghe nhạc của tôi có thể cho phép người dùng khách truy cập các bài hát, xem thông tin về nghệ sĩ và album, nhưng để tạo danh sách phát cá nhân hoặc thêm nhận xét, người dùng có thể cần phải đăng nhập vào tài khoản của họ.

### 2.2.2. Yêu cầu phi chức năng

Tính yêu cầu phi chức năng của website này bao gồm:

#### 2.2.2.1. Tính thẩm mỹ và trải nghiệm người dùng:

- Sử dụng màu sắc nhẹ nhàng và hài hòa:

```
CSS  
  
body {margin:0;}  
  
nav, footer {background-color: #9ed9e7;}  
  
nav ul li a:link, nav ul li a:visited {color:  
#de413e;}
```

- Giao diện được thiết kế sao cho dễ nhìn và thân thiện với người dùng:

**CSS**

```
.icon-bar {  
    width: 100%;  
    background-color: #555;  
}  
  
nav ul {  
    display: flex;  
    justify-content: center;  
}
```

**2.2.2.2. Menu điều hướng:**

- Cung cấp menu điều hướng dễ sử dụng:

**html**

```
<nav>  
    <ul>  
        <li><a href="index.html">Trang Chủ</a></li>  
        <li><a href="#">Tìm Kiếm</a></li>  
        <li><a href="DKTK.html">Đăng Ký</a></li>  
        <li><a href="DN.html">Đăng Nhập</a></li>  
        <li><a href="#">Khách</a></li>  
    </ul>  
</nav>
```

**2.2.2.3. Hiện thị danh sách các bài hát:**

- Sắp xếp các bài hát thành các ô có kích thước đồng đều:

```
html
<div class="col-sm-3">
  <!-- Content -->
</div>
```

- Mỗi bài hát được hiển thị với hình ảnh và tiêu đề rõ ràng:

```
html
<div class="col-sm-3">
  <a href="chotrongai.html">
    </a>
    <a href="chotrongai.html">Chờ Trông ai - HỒ
PHI NAL</a>
  </div>
```

```
<div class="col-sm-3">
  <a href="sacngocnga.html">
    </a>
    <a href="sacngocnga.html">Sắc Ngọc Ngà - HỮU
TRÍ </a>
  </div>
```

```
<div class="col-sm-3">
  <a href="diepkhucquehuong.html">
    </a>
    <a href="diepkhucquehuong.html">Điệp Khúc Quê
Hương - TIẾN TỚI </a>
  </div>
```

```
<div class="col-sm-3">
    <a href="thuyenquyen.html">
        </a>
        <a href="thuyenquyen.html">Thuyền Quyên - ĐIỀU
KIÊN </a>
    </div>
```

```
<div class="col-sm-3">
    <a href="khuatloi.html">
        </a>
        <a href="khuatloi.html">Khuất Lỗi - H-
Kray</a>
    </div>
```

```
<div class="col-sm-3">
    <a href="xinmaruoccondau.html">
        </a>
        <a href="xinmaruoccondau.html">Xin Má Rước
Dâu - XIN MÁ RƯỚC DÂU - ĐIỀU KIÊN</a>
    </div>
```



```
<div class="col-sm-3">
    <a href="thuyenhoaxuan.html">
        </a>
        <a href="thuyenhoaxuan.html">THUYỀN HOA XUÂN
- TIẾN TỚI</a>
    </div>
```

```
<div class="col-sm-3">
    <a href="quaphutuong.html">
        </a>
        <a href="quaphutuong.html">Quả Phụ Tướng -
DUNGHOANGPHAM</a>
    </div>
```

```
<div class="col-sm-3">
    <a href="trootraoduyen.html">
        </a>
        <a href="trootraoduyen.html">Trót Trao Duyên -
NB3 Hoài Bảo</a>
    </div>
```

```
<div class="col-sm-3">
    <a href="vanvuong.html">
        </a>
        <a href="vanvuong.html">- VUONG VẤN - - HANA
CẨM TIÊN x TVk x NBORO ( Prod. )</a>
    </div>
```

```
<div class="col-sm-3">
    <a href="tamlongson.html">
        </a>
        <a href="tamlongson.html">Tâm Lòng Son - H-
KRAY</a>
    </div class="col-sm-3">
```

```
<div class="col-sm-3">
    <a href="VTCA.html">
        </a>
        <a href="VTCA.html">PHƯƠNG MỸ CHI x DTAP VŨ
TRỤ CỐ ANH </a>
    </div>
```

```
<div class="col-sm-3">
    <a href="nanchentieusau.html">
        </a>
```

```
<a href="nanchentieusau.html">Nâng Chén Tiêu  
Sầu - Bích Phương</a>  
</div>
```

```
<div class="col-sm-3">  
  <a href="catdoinoisau.html">  
    </a>  
    <a href="catdoinoisau.html"> TẶNG DUY TÂN -  
CẮT ĐÔI NỖI SẦU </a>  
  </div>
```

```
<div class="col-sm-3">  
  <a href="chanai.html">  
    </a>  
    <a href="chanai.html"> Chân Ái</a>  
  </div class="col-sm-3">  
</div>
```

#### 2.2.2.4. Tính linh hoạt và phản hồi:

- Menu và các liên kết có hiệu ứng hover khi di chuột:

##### **CSS**

```
.icon-bar a:hover {background-color: #000;}  
nav ul li a:hover {color: #fff;}
```

Tóm lại, đoạn mã này không chỉ tập trung vào chức năng chính là nghe nhạc mà còn chú trọng vào trải nghiệm người dùng thông qua việc tùy chỉnh

giao diện, cung cấp menu điều hướng dễ sử dụng và hiển thị danh sách các bài hát một cách hấp dẫn.

## **2.3. Thiết kế dữ liệu**

### *2.3.1. Mô hình dữ liệu*

Khái niệm về mô hình ERD

Entity Relationship Diagram được biết đến là một dạng sơ đồ trực quan thực thể, đóng vai trò quan trọng trong việc quản lý và thiết kế cơ sở dữ liệu quan hệ

Dưới đây là một số khái niệm cơ bản về ERD:

Thực thể (Entity): Đại diện cho một nhóm các đối tượng có các thuộc tính chung. Thực thể có thể là người, đồ vật, sự kiện hoặc khái niệm trừu tượng khác.

Ví dụ: Người dùng, Bài hát, Album.

Thuộc tính (Attribute): Là các đặc điểm hoặc thông tin mô tả về một thực thể. Ví dụ: ID, Tên, Email là các thuộc tính của thực thể Người Dùng.

Mối quan hệ (Relationship): Mô tả cách mà các thực thể liên kết với nhau trong cơ sở dữ liệu. Có ba loại mối quan hệ chính là 1-n (một-nhiều), n-n (nhiều-nhiều) và 1-1 (một-một).

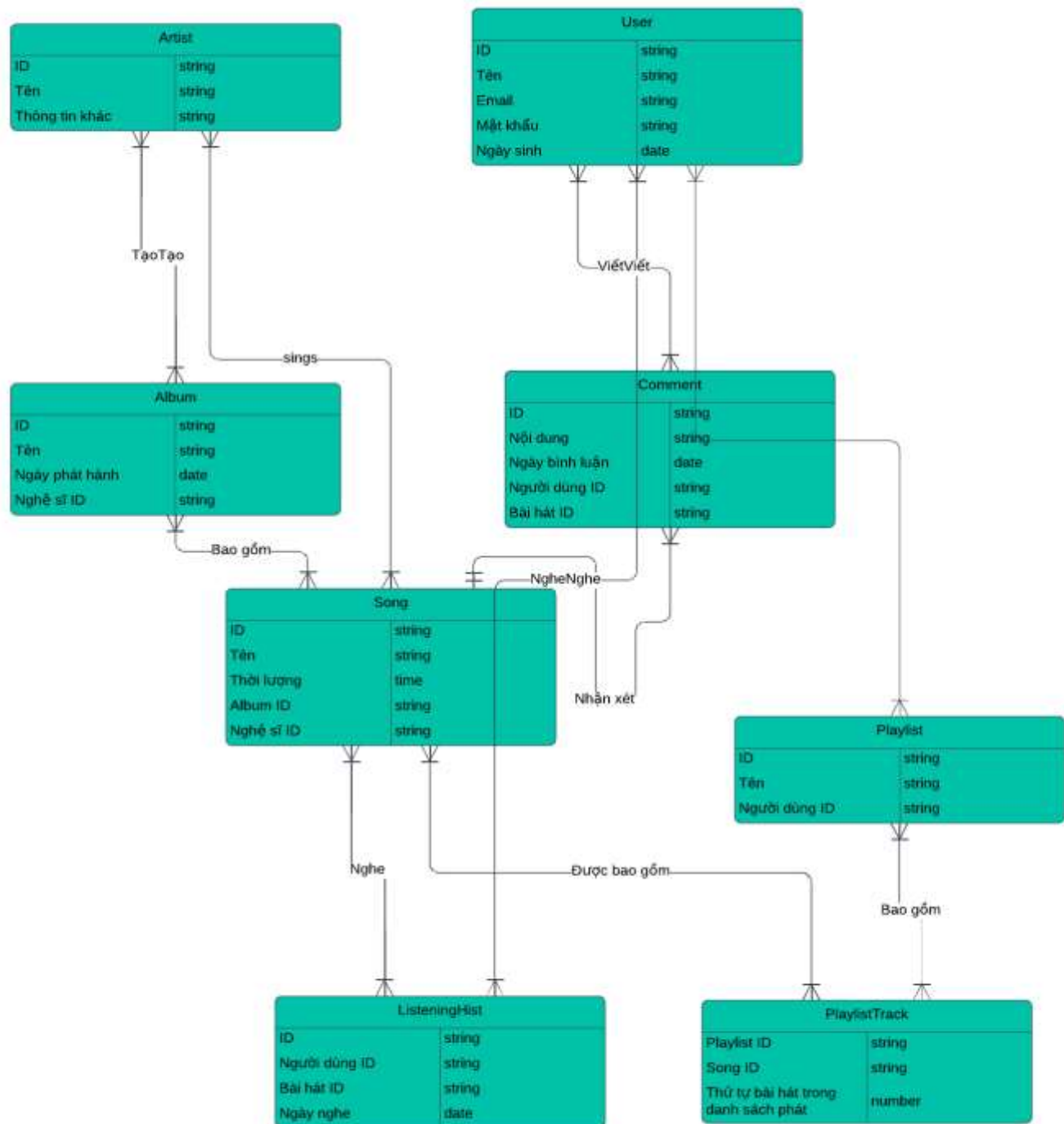
Ví dụ: Mối quan hệ giữa Người Dùng và Danh Sách Phát, mỗi người dùng có thể có nhiều danh sách phát (mối quan hệ 1-n).

Khóa (Key): Là một thuộc tính hoặc tập hợp các thuộc tính được sử dụng để định danh duy nhất cho mỗi bản ghi trong một thực thể. Khóa chính đại diện cho một cách duy nhất để xác định một bản ghi.

Ví dụ: ID thường được sử dụng làm khóa chính cho các thực thể.

Mô hình ERD: Là biểu đồ đồ họa mô tả các thực thể, thuộc tính và mối quan hệ giữa chúng trong cơ sở dữ liệu. Mô hình ERD giúp hiểu rõ cấu trúc của cơ sở dữ liệu và quan hệ giữa các thành phần của nó.

**\*\* Mô hình ERD**



### 2.3.2. Danh sách các thực thể và mối kết hợp

*Bảng 2.1 Danh sách các thực thể và mối kết hợp*

| <i>STT</i> | <i>Tên thực thể/Mối kết hợp</i> | <i>Diễn giải</i>                        | <i>Ghi chú</i>                                  |
|------------|---------------------------------|---|---|
| 1          | User                            | Thông tin người dùng                    | Định danh người dùng trên trang web             |
| 2          | Artist                          | Thông tin nghệ sĩ                       | Thông tin về nghệ sĩ, ca sĩ                     |
| 3          | Album                           | Thông tin về các album                  | Các album âm nhạc                               |
| 4          | Song                            | Thông tin bài hát                       | Danh sách các bài hát                           |
| 5          | Playlist                        | Danh sách phát của người dùng           | Danh sách phát cá nhân của người dùng           |
| 6          | PlaylistTrack                   | Liên kết giữa danh sách phát và bài hát | Danh sách các bài hát trong mỗi danh sách phát  |
| 7          | ListeningHist                   | Lịch sử nghe nhạc của người dùng        | Các bản ghi về lịch sử nghe nhạc của người dùng |
| 8          | Comment                         | Bình luận của người dùng về các bài hát | Bình luận và phản hồi của người dùng            |

### 2.3.3. Chi tiết thực thể và mối kết hợp

Thực thể

Mô tả

Chi tiết thực thể

*Bảng 2.2 Chi tiết thực thể User*

| <i>STT</i> | <i>Tên thuộc tính</i> | <i>Diễn giải</i>             | <i>Kiểu dữ liệu</i> | <i>Ràng buộc toàn vẹn</i> |
|------------|-----------------------|------------------------------|---------------------|---------------------------|
| 1          | UserID                | ID của người dùng            | int                 | Primary Key               |
| 2          | Username              | Tên đăng nhập của người dùng | varchar(50)         | Unique, Not Null          |
| 3          | Password              | Mật khẩu của người dùng      | varchar(50)         | Not Null                  |
| 4          | Email                 | Địa chỉ email của người dùng | varchar(100)        | Unique, Not Null          |
| 5          | CreatedAt             | Thời điểm tạo tài khoản      | datetime            | Not Null                  |
| 6          | UpdatedAt             | Thời điểm cập nhật thông tin | datetime            | Not Null                  |

*Bảng 2.3 Chi tiết thực thể Artist*

| <i>STT</i> | <i>Tên thuộc tính</i> | <i>Diễn giải</i>     | <i>Kiểu dữ liệu</i> | <i>Ràng buộc toàn vẹn</i> |
|------------|-----------------------|----------------------|---------------------|---------------------------|
| 1          | ArtistID              | ID của nghệ sĩ       | int                 | Primary Key               |
| 2          | Name                  | Tên của nghệ sĩ      | varchar(100)        | Not Null                  |
| 3          | Description           | Mô tả về nghệ sĩ     | text                |                           |
| 4          | Country               | Quốc gia của nghệ sĩ | varchar(50)         |                           |

*Bảng 2.4 Chi tiết thực thể Album*

| <i>STT</i> | <i>Tên thuộc tính</i> | <i>Diễn giải</i>        | <i>Kiểu dữ liệu</i> | <i>Ràng buộc toàn vẹn</i> |
|------------|-----------------------|-------------------------|---------------------|---------------------------|
| 1          | AlbumID               | ID của album            | int                 | Primary Key               |
| 2          | Title                 | Tiêu đề của album       | varchar(100)        | Not Null                  |
| 3          | Year                  | Năm phát hành của album | int                 | Year                      |
| 4          | CoverImage            | Hình ảnh bìa của album  | varchar(255)        |                           |

*Bảng 2.5 Chi tiết thực thể Song*

| <i>STT</i> | <i>Tên thuộc tính</i> | <i>Diễn giải</i>            | <i>Kiểu dữ liệu</i> | <i>Ràng buộc toàn vẹn</i>      |
|------------|-----------------------|-----------------------------|---------------------|--------------------------------|
| 1          | SongID                | ID của bài hát              | int                 | Primary Key                    |
| 2          | Title                 | Tiêu đề của bài hát         | varchar(100)        | Not Null                       |
| 3          | Genre                 | Thể loại của bài hát        | varchar(50)         |                                |
| 4          | AudioURL              | Đường dẫn đến file âm thanh | varchar(255)        | Not Null                       |
| 5          | ArtistID              | ID của nghệ sĩ              | int                 | Foreign Key (Artist(ArtistID)) |
| 6          | AlbumID               | ID của album                | int                 | Foreign Key (Album(AlbumID))   |



*Bảng 2.6 Chi tiết thực thể Playlist*

| <i>STT</i> | <i>Tên thuộc tính</i> | <i>Diễn giải</i>           | <i>Kiểu dữ liệu</i> | <i>Ràng buộc toàn vẹn</i> |
|------------|-----------------------|----------------------------|---------------------|---------------------------|
| 1          | PlaylistID            | ID của danh sách phát      | int                 | Primary Key               |
| 2          | Title                 | Tiêu đề của danh sách phát | varchar(100)        | Not Null                  |
| 3          | UserID                | ID của người dùng          | int                 |                           |
| 4          | Description           | Mô tả về danh sách phát    | text                |                           |

*Bảng 2.7 Chi tiết thực thể PlaylistTrack*

| <i>STT</i> | <i>Tên thuộc tính</i> | <i>Diễn giải</i>                    | <i>Kiểu dữ liệu</i> | <i>Ràng buộc toàn vẹn</i>          |
|------------|-----------------------|-------------------------------------|---------------------|------------------------------------|
| 1          | PlaylistID            | ID của danh sách phát               | int                 | Foreign Key (Playlist(PlaylistID)) |
| 2          | SongID                | ID của bài hát trong danh sách phát |                     |                                    |

*Bảng 2.8 Chi tiết thực thể ListeningHist*

| <i>STT</i> | <i>Tên thuộc tính</i> | <i>Diễn giải</i>  | <i>Kiểu dữ liệu</i> | <i>Ràng buộc toàn vẹn</i> |
|------------|-----------------------|-------------------|---------------------|---------------------------|
| 1          | ListeningID           | ID của bản ghi    | INT                 | Khóa chính                |
| 2          | UserID                | ID của người dùng | INT                 | Khóa ngoại (User)         |

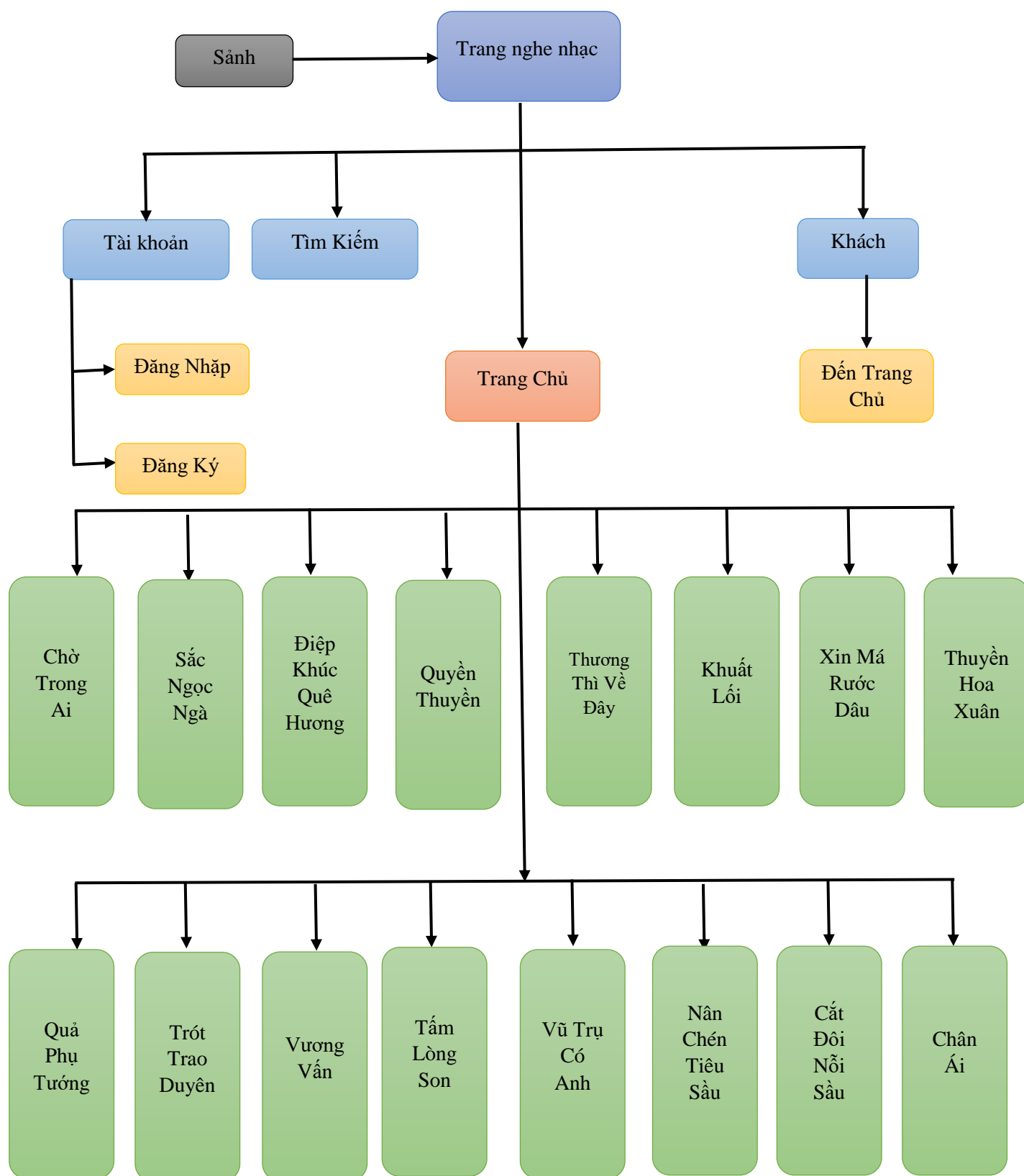
|   |               |                |          |                   |
|---|---------------|----------------|----------|-------------------|
| 3 | SongID        | ID của bài hát | INT      | Khóa ngoại (Song) |
| 4 | ListeningTime | Thời gian nghe | DATETIME |                   |

*Bảng 2.9 Chi tiết thực thể Comment*

| <i>STT</i> | <i>Tên thuộc tính</i> | <i>Diễn giải</i>    | <i>Kiểu dữ liệu</i> | <i>Ràng buộc toàn vẹn</i> |
|------------|-----------------------|---------------------|---------------------|---------------------------|
| 1          | CommentID             | ID của bình luận    | INT                 | Khóa chính                |
| 2          | UserID                | ID của người dùng   | INT                 | Khóa ngoại (User)         |
| 3          | SongID                | ID của bài hát      | INT                 | Khóa ngoại (Song)         |
| 4          | Content               | Nội dung bình luận  | TEXT                | -                         |
| 5          | CommentTime           | Thời gian bình luận | DATETIME            | -                         |

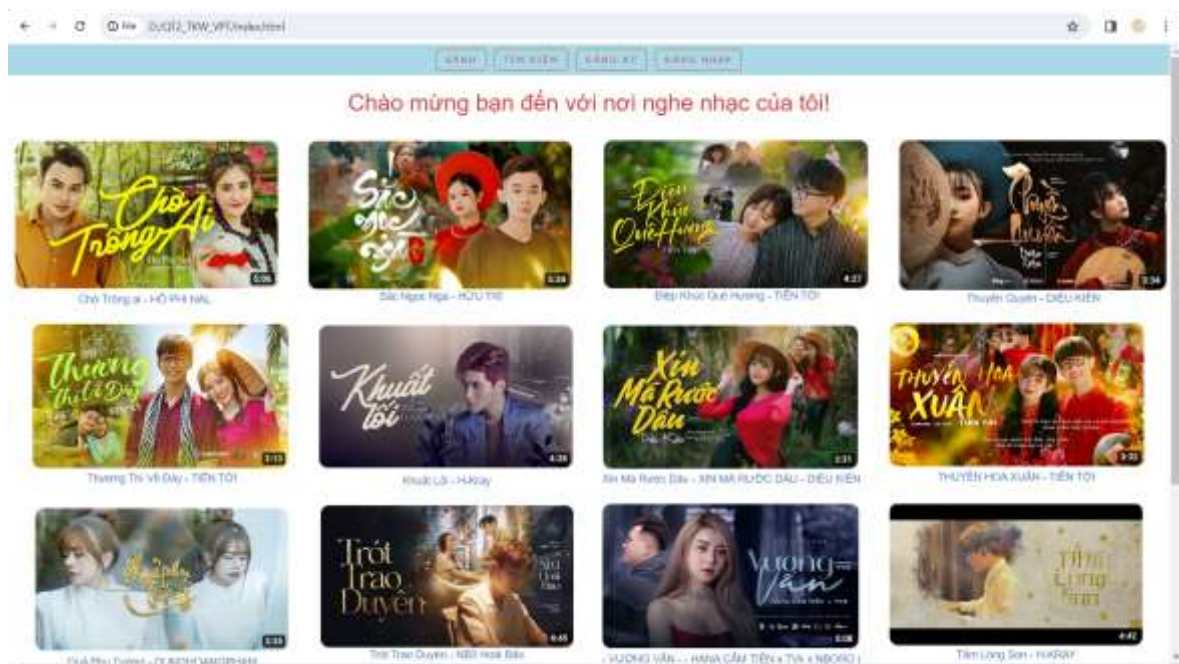
## **2.4. Thiết kế giao diện**

### *2.4.1. Sơ đồ hệ thống*



*Hình 2.1 Sơ đồ hệ thống*

#### 2.4.2. Giao diện trang chủ người dùng



*Hình 2.2 Giao diện trang chủ của người dùng*

##### 2.4.2.1. Mô tả chi tiết

- Thanh điều hướng (Navigation Bar)
  - + Vị trí: Phía trên cùng của trang web.
  - + Màu sắc: Màu xanh nhạt (#9ed9e7).
  - + Nội dung: Bao gồm các nút điều hướng đến các trang khác của trang web:
    - Sản phẩm (trang chủ)
    - Tìm Kiếm
    - Đăng Ký

- Đăng Nhập
- Bố cục: Các nút điều hướng được căn giữa theo chiều ngang và được bao quanh bởi các đường viền mỏng. Khi di chuột qua các liên kết, màu chữ thay đổi từ đỏ (#de413e) sang màu đen (#080505).
- Tiêu đề chào mừng
  - + Vị trí: Ngay dưới thanh điều hướng.
  - + Nội dung: "Chào mừng bạn đến với nơi nghe nhạc của tôi!"
  - + Màu sắc chữ: Đỏ (#de413e).
  - + Định dạng: Phong chữ in đậm (bold).
- Danh sách các bài hát
  - + Bố cục: Các bài hát được hiển thị trong một lưới (grid) gồm nhiều hàng và cột, với mỗi hàng gồm 4 bài hát.
  - + Chi tiết từng bài hát:

Ảnh bìa: Mỗi bài hát có một hình ảnh bìa (album cover) hiển thị ở phía trên cùng.

- Tên bài hát và nghệ sĩ: Được đặt ngay dưới hình ảnh bìa.
- Tên bài hát và nghệ sĩ được liên kết (hyperlink) tới trang chi tiết của bài hát đó.
- Thời lượng: Thời lượng của mỗi bài hát được hiển thị ở góc dưới bên phải của hình ảnh bìa.

#### 2.4.2.2. Cách bố cục

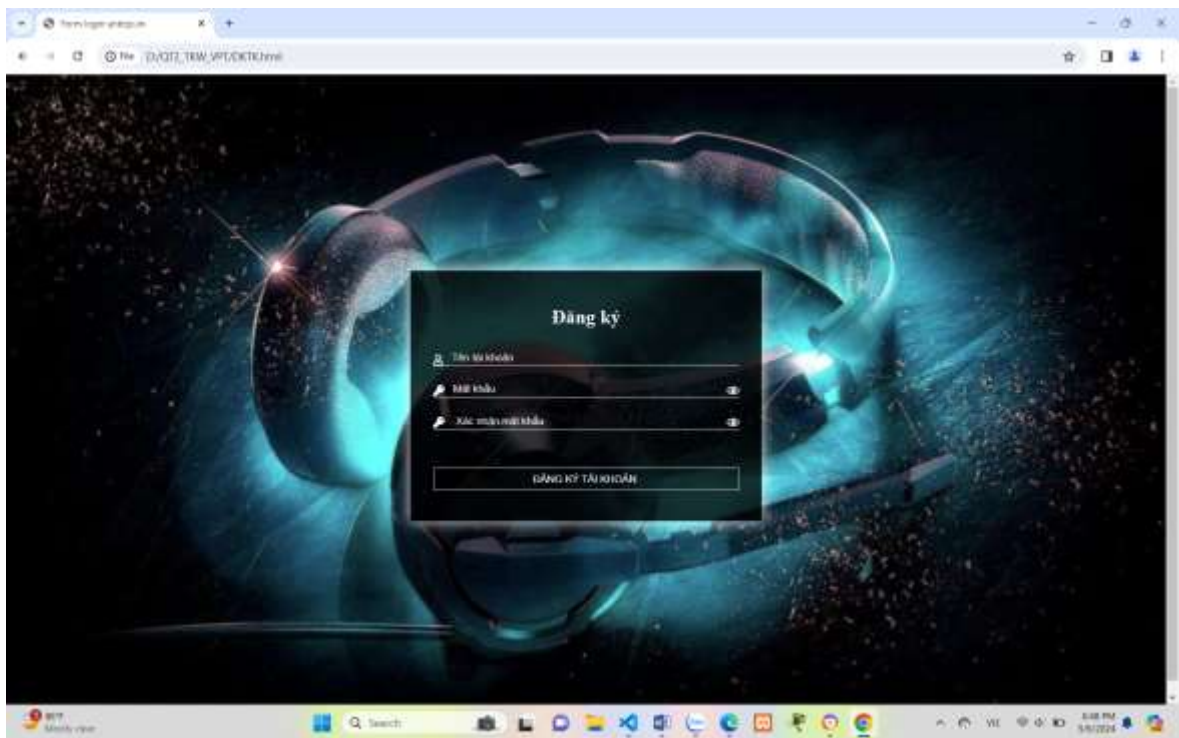
- Container chính: Toàn bộ nội dung của trang được đặt trong một container có lớp container-fluid bg-3 text-center, đảm bảo rằng nội dung được căn giữa và hiển thị tốt trên mọi thiết bị (responsive design).

- Hàng và cột: Sử dụng hệ thống lưới của Bootstrap để sắp xếp các bài hát. Mỗi hàng (row) chứa 4 cột (col-sm-3), mỗi cột chứa một bài hát.
- Hình ảnh bài hát: Hình ảnh của các bài hát được thiết lập với lớp img-responsive, đảm bảo rằng hình ảnh sẽ tự động điều chỉnh kích thước phù hợp với chiều rộng của cột.
- Liên kết (Links): Mỗi tên bài hát và nghệ sĩ được bọc trong thẻ <a> để người dùng có thể nhấp vào và được chuyển đến trang chi tiết của bài hát.

#### 2.4.2.4. Tóm lại

Trang web nghe nhạc này có thiết kế đơn giản và trực quan, với thanh điều hướng dễ sử dụng và một bố cục lưới rõ ràng giúp người dùng dễ dàng tìm kiếm và truy cập các bài hát yêu thích. Hình ảnh và thông tin về bài hát được trình bày rõ ràng, tạo trải nghiệm người dùng thân thiện.

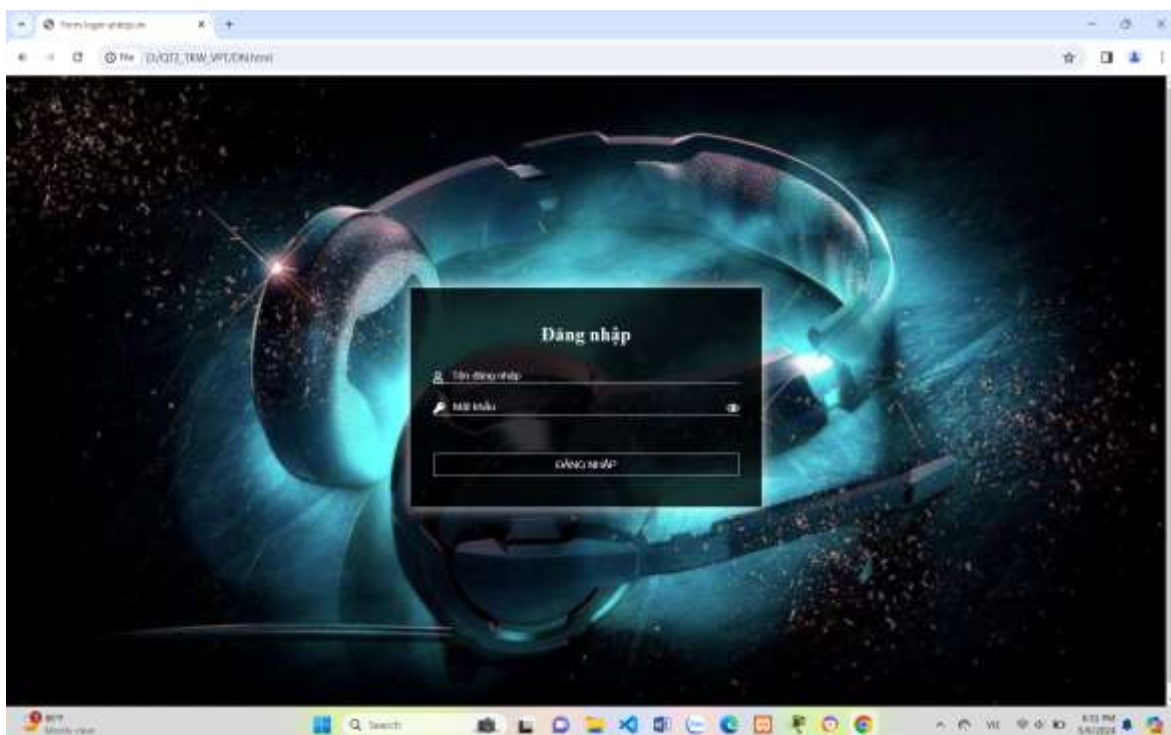
#### 2.4.3. *Giao diện chức năng Đăng Ký*



*Hình 2.3 Giao diện chức năng Đăng Ký*

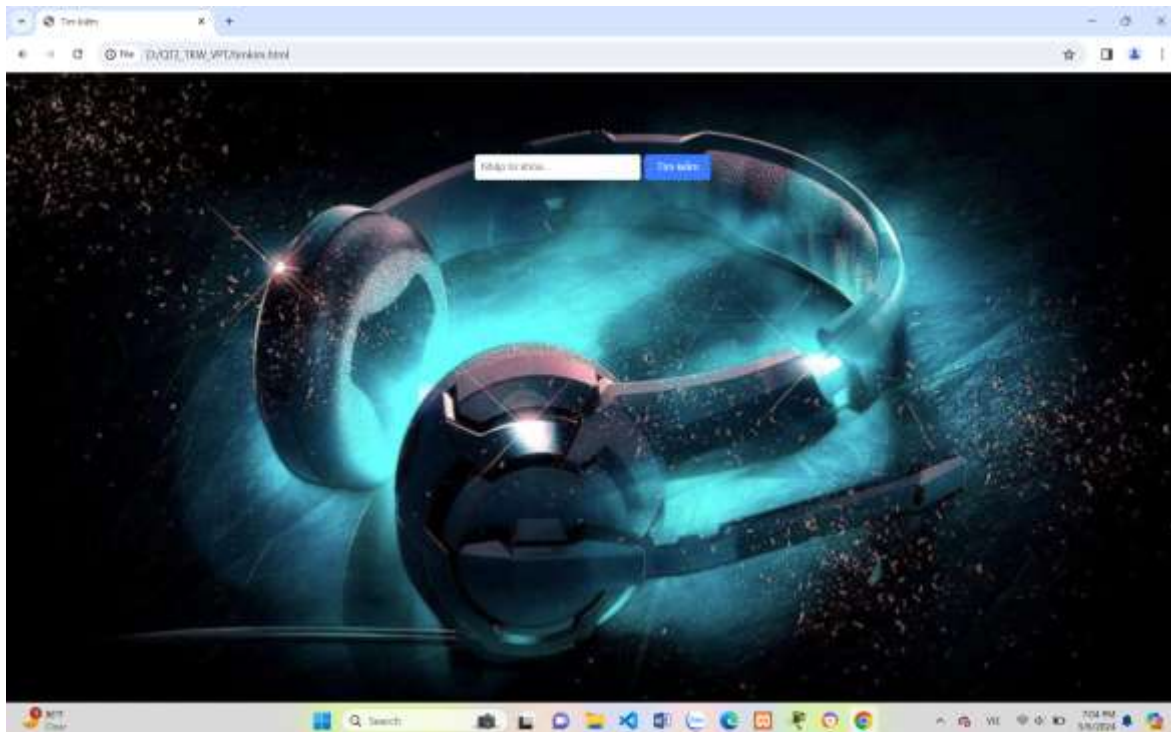


#### 2.4.4. *Giao diện chức năng Đăng Nhập*



*Hình 2.4 Giao diện chức năng Đăng Nhập*

#### 2.4.5. *Giao diện chức năng Tìm kiếm*



*Hình 2.5 Giao diện chức năng Tìm Kiếm*

#### *2.4.6. Kết chương*

Trong trang web này, giao diện được thiết kế đơn giản nhưng vẫn rất hấp dẫn và dễ sử dụng. Navbar ở đầu trang giúp người dùng dễ dàng điều hướng giữa các trang. Các nút điều hướng có màu sắc và kiểu dáng rất bắt mắt. Phần chào mừng nổi bật giữa trang với hình nền đặc biệt và tiêu đề lớn. Phần danh sách các bài hát được trình bày một cách gọn gàng và hấp dẫn, mỗi bài hát được kèm theo một hình ảnh và link điều hướng. Tổng thể, giao diện của trang web này tạo cảm giác thân thiện và dễ dàng cho người dùng khi sử dụng.



## **CHƯƠNG 3. CÀI ĐẶT THỰC NGHIỆM**

### **3.1. Bộ dữ liệu thử nghiệm**

Trong quá trình thử nghiệm, chúng tôi đã sử dụng một bộ dữ liệu gồm các bài hát được lựa chọn từ các thể loại nhạc khác nhau như pop, rock, rap, và nhạc ballad. Bộ dữ liệu này bao gồm các thông tin về tên bài hát, nghệ sĩ, hình ảnh đại diện, và các liên kết điều hướng đến trang chi tiết của từng bài hát.

### **3.2. Kết quả thực nghiệm**

#### *3.2.1. Chức năng Đăng Ký*

Chức năng Đăng Ký cho phép người dùng tạo tài khoản mới trên trang web để truy cập vào các tính năng khác như lưu trữ yêu thích. Kết quả thực nghiệm cho thấy chức năng này hoạt động một cách trơn tru và linh hoạt, cho phép người dùng nhập thông tin cá nhân và tạo tài khoản một cách dễ dàng.

#### *3.2.2. Chức năng Đăng Nhập*

Chức năng Đăng Nhập cho phép người dùng truy cập vào tài khoản đã đăng ký trước đó. Kết quả thực nghiệm cho thấy chức năng này hoạt động một cách nhanh chóng và đáng tin cậy, người dùng có thể nhập thông tin đăng nhập và truy cập vào tài khoản của họ một cách dễ dàng.

#### *3.2.3. Chức năng Tìm Kiếm*

Chức năng Tìm Kiếm cho phép người dùng tìm kiếm bài hát theo tên hoặc nghệ sĩ. Kết quả thực nghiệm cho thấy chức năng này có thể tìm kiếm và hiển thị kết quả một cách chính xác và nhanh chóng, giúp người dùng dễ dàng tìm thấy bài hát mong muốn.

#### *3.2.4. Chức năng Khách*

Chức năng Khách cho phép người dùng truy cập vào trang web mà không cần đăng nhập. Kết quả thực nghiệm cho thấy chức năng này cho phép người

dùng trải nghiệm trang web và nghe nhạc một cách thuận tiện mà không cần phải tạo tài khoản.

### **3.3. Kết chương**

Kết quả của các thử nghiệm đã chứng minh rằng giao diện và các chức năng trên trang web của chúng tôi hoạt động một cách hiệu quả và cung cấp trải nghiệm người dùng tốt. Tính linh hoạt và sự dễ sử dụng của các chức năng đã được chứng minh thông qua quá trình thử nghiệm. Điều này cung cấp cơ sở để tiếp tục phát triển và cải thiện trang web để mang lại trải nghiệm người dùng tốt nhất.

## **CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **4.1. Kết luận**

Trang web nghe nhạc của chúng tôi đã được phát triển với mục tiêu cung cấp trải nghiệm nghe nhạc tốt nhất cho người dùng. Tính linh hoạt và dễ sử dụng của giao diện đã được chứng minh thông qua quá trình thử nghiệm. Các chức năng như Đăng Ký, Đăng Nhập, Tìm Kiếm, Phát Nhạc và Chức năng Khách đều hoạt động một cách mượt mà và đáp ứng được nhu cầu của người dùng.

### **4.2. Hướng phát triển**

#### *4.2.1. Mở rộng bộ dữ liệu*

Cần mở rộng bộ dữ liệu với nhiều thể loại nhạc và bài hát hơn, cũng như cập nhật thông tin về các nghệ sĩ và album mới. Điều này giúp người dùng có thêm nhiều lựa chọn hơn và trải nghiệm nhạc sự đa dạng hơn.

#### *4.2.2. Tích hợp tính năng xã hội*

Tích hợp tính năng cho phép người dùng chia sẻ bài hát yêu thích, tạo danh sách phát, và tương tác với nhau qua các mạng xã hội như Facebook, Twitter, và Instagram. Điều này giúp tạo ra một cộng đồng người yêu nhạc chia sẻ và tương tác với nhau.

#### *4.2.3. Tối ưu hóa trải nghiệm di động*

Tối ưu hóa trang web cho các thiết bị di động như điện thoại di động và máy tính bảng để đảm bảo trải nghiệm tốt nhất cho người dùng trên các nền tảng này. Giao diện phải linh hoạt và dễ dàng sử dụng trên mọi loại thiết bị.

#### *4.2.4. Phát triển tính năng mới*

Thêm các tính năng mới như tạo playlist cá nhân, gợi ý bài hát dựa trên sở thích của người dùng, và tích hợp các dịch vụ âm nhạc khác như podcast và

radio. Điều này giúp tăng cường tính tương tác và mang lại trải nghiệm nghe nhạc đa dạng.

#### *4.2.5. Tăng cường bảo mật*

Cải thiện hệ thống bảo mật để đảm bảo an toàn thông tin cá nhân của người dùng và ngăn chặn các vấn đề liên quan đến việc truy cập trái phép. Các biện pháp bảo mật như mã hóa dữ liệu, xác thực hai yếu tố và giám sát hệ thống cần được triển khai và cập nhật định kỳ.

*Tóm lại*, việc tiếp tục phát triển và cải thiện trang web sẽ giúp nâng cao trải nghiệm người dùng và đáp ứng được nhu cầu ngày càng đa dạng của cộng đồng người sử dụng. Chúng tôi cam kết tiếp tục làm việc để mang lại trải nghiệm nghe nhạc tốt nhất cho mọi người.

## **DANH MỤC TÀI LIỆU THAM KHẢO**

- [1]. Niederst Robbins, Learning Web Design, 5th ed. [Online]. Available: [Learning Web Design \(booksfree.org\)](https://booksfree.org) [Accessed: 21-May-2024].
- [2]. Hà Thị Thúy Vi, Tài Liệu Giảng Dạy Môn CSDL, [Trường đại học Trà Vinh 2013]
- [3]. Kim, Y., Lee, J., & Kim, S. (2019). A Study on User Interface Design for Music Recommendation Service. *2019 12th International Conference on Human System Interaction (HSI)*.
- [4]. OpenAI. (2022). OpenAI GPT. [Online]. Available: <https://openai.com/gpt>.
- [5]. W3Schools. (n.d.). Bootstrap Introduction. [Online]. Available: [W3Schools Online Web Tutorials](https://www.w3schools.com/bootstrap/)
- [6]. Mozilla Developer Network. (n.d.). CSS Styling. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
- [7]. Bài giảng Cơ sở dữ liệu, Khoa CNTT - ĐH KHTN TPHCM
- [8]. [Online]. Available: [Documents \(lucid.app\)](https://lucid.app) [Accessed: 21-May-2024].
- [9]. Mozilla Developer Network, "HTML: HyperText Markup Language," [Online]. Available: [HTML: HyperText Markup Language | MDN \(mozilla.org\)](https://developer.mozilla.org/en-US/docs/Web/HTML) [Accessed: 21-May-2024].
- [10]. Bootstrap, "Introduction," [Online]. Available: [Getting started · Bootstrap \(getbootstrap.com\)](https://getbootstrap.com/) [Accessed: 21-May-2024].