

**LÊ MINH TRUNG** **MSSV: 6551071089**

**BÙI GIA PHÚC** **MSSV: 6551071069**

**VÕ HOÀNG PHÚC** **MSSV: 6551071067**

**LÊ MINH CHÍ** **MSSV: 6551071011**

**VŨ ĐĂNG HOÀNG TIẾN**      **MSSV: 6551071067**

## ĐỀ TÀI: THIẾT KẾ TRANG WEB NGHE NHẠC

# BÁO CÁO ĐỒ ÁN MÔN HỌC : THIẾT KẾ WEB

## BÁO CÁO ĐỒ ÁN MÔN HỌC : THIẾT KẾ WEB

Tp. Hồ Chí Minh, năm 2025

[illegible]

**ThS. Nguyễn Thiện Dương**

## LỜI CẢM ƠN

Lời nói đầu tiên, em xin gửi tới Quý Thầy Cô Bộ môn Công nghệ Thông tin Trường Đại học Giao thông vận tải phân hiệu tại thành phố Hồ Chí Minh lời chúc sức khỏe và lòng biết ơn sâu sắc.

Chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Thiện Dương đã tận tình giảng dạy, truyền đạt những kiến thức chuyên môn vững chắc cùng với những bài học kinh nghiệm thực tế quý báu. Những bài giảng tâm huyết, những chỉ dẫn tỉ mỉ, sự tận tâm và trách nhiệm của thầy không chỉ giúp chúng em hiểu rõ hơn về môn Thiết kế web mà còn trang bị cho chúng em những tư duy logic, kỹ năng giải quyết vấn đề và phương pháp làm việc khoa học, đây là hành trang vô cùng quý giá cho con đường học tập và sự nghiệp sau này.

Trong suốt quá trình học tập và thực hiện bài tập lớn môn Thiết kế web, chúng em đã nhận được rất nhiều sự hỗ trợ và động viên quý báu từ thầy Nguyễn Thiện Dương người đã trực tiếp giảng dạy và hướng dẫn chúng em trong suốt học kỳ vừa qua. Tuy đã cố gắng trong quá trình nghiên cứu tìm hiểu tuy nhiên do kiến thức còn hạn chế nên vẫn còn tồn tại nhiều thiếu sót. Vì vậy em rất mong nhận được sự đóng góp ý kiến của Quý thầy cô bộ môn để đề tài của em có thể hoàn thiện hơn.

Chúng em xin chân thành cảm ơn thầy vì không chỉ dừng lại ở việc truyền đạt kiến thức, mà còn luôn động viên, tạo điều kiện kết nối với các anh chị cựu sinh viên để khích lệ tinh thần học hỏi, sáng tạo của chúng em. Sự tận tâm và nhiệt huyết của thầy là nguồn động lực lớn giúp chúng em không ngừng cố gắng và hoàn thiện mình.

Một lần nữa, chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Nguyễn Thiện Dương. Kính chúc thầy luôn mạnh khỏe, hạnh phúc và đạt nhiều thành công trong sự nghiệp trồng người cao quý.

# MỤC LỤC

<b>NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.....</b>	<b>i</b>
<b>LỜI CẢM ƠN .....</b>	<b>ii</b>
<b>MỤC LỤC .....</b>	<b>iii</b>
<b>BẢNG BIỂU, HÌNH VẼ.....</b>	<b>viii</b>
<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI VÀ PHÂN TÍCH YÊU CẦU .....</b>	<b>1</b>
1.1. Giới thiệu đề tài .....	1
1.2. Mục tiêu đề tài .....	1
1.3. Mục đích nghiên cứu .....	1
1.4. Đối tượng nghiên cứu .....	2
1.5. Phạm vi nghiên cứu và giới hạn .....	2
1.6. Phương pháp nghiên cứu .....	3
1.7. Ý nghĩa thực tiễn của đề tài .....	3
1.8. Cấu trúc cuốn báo cáo đồ án .....	3
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ THỰC NGHIỆM .....</b>	<b>5</b>
2.1. Cơ sở lý thuyết .....	5
2.1.1. Ngôn ngữ HTML5 .....	5
2.1.1.1. Khái niệm .....	5
2.1.1.2. Sự khác nhau giữa HTML và HTML5.....	6
2.1.1.3. Các tính năng mới của HTML5.....	7
2.1.1.4 Cấu trúc của một trang HTML5 cơ bản .....	9
2.1.1.5 Ưu và nhược điểm của HTML5 .....	9
2.1.2. Ngôn ngữ CSS.....	10
2.1.2.1. Khái niệm .....	10

2.1.2.2. Vai trò của CSS trong mối liên hệ giữa HTML .....	11
2.1.2.3. Cấu trúc và cú pháp CSS cơ bản .....	12
2.1.2.4. Phân loại CSS .....	13
2.1.2.5. CSS3 Phiên bản tối ưu nhất hiện tại.....	14
2.1.2.6. Ưu điểm khi sử dụng CSS trong thiết kế web.....	15
2.1.3. Ngôn ngữ lập trình JavaScript .....	15
2.1.3.1 Khái niệm .....	15
2.1.3.2. Ứng dụng của JavaScript.....	15
2.1.3.3. Cách thức hoạt động của JavaScript.....	17
2.1.3.4. Ưu và nhược điểm của ngôn ngữ JavaScript.....	17
2.1.4. Khái niệm và nguyên tắc UI/UX, responsive design, accessibility .....	18
2.1.4.1 UI/UX (User Interface / User Experience).....	18
2.1.4.2. Responsive Design (Thiết kế tương thích/phản hồi).....	19
2.1.4.3. Accessibility (Khả năng truy cập).....	19
2.1.5. Một số Framework phổ biến .....	20
2.1.5.1. Tailwind CSS .....	20
2.1.5.2. Bootstrap.....	20
2.1.5.3. Foundation.....	21
2.1.6. Nguyên tắc phối màu, bố cục, typography trong thiết kế web. ....	21
2.1.6.1. Nguyên tắc phối màu trong thiết kế website .....	21
2.1.6.2. Nguyên tắc typography trong thiết kế website.....	22
2.1.7. Khái niệm về hosting, domain, sever.....	23
2.1.7.1. Domain (Tên miền) .....	23
2.1.7.2. Hosting (Web Hosting) .....	24
2.1.7.3. Server (Máy chủ).....	24

<b>2.2. Khảo sát các sản phẩm tương tự.....</b>	<b>25</b>
2.2.1. Phân tích Website cùng lĩnh vực.....	25
2.2.1.1 ZING MP3 (zingmp3.vn).....	25
2.2.1.2. SPOTIFY (spotify.com).....	27
2.2.2. Nhận xét, rút ra bài học, định hướng cho sản phẩm nhóm .....	29
2.2.2.1. Nhận xét về sản phẩm .....	29
2.2.2.2. Rút ra bài học kinh nghiệm .....	30
2.2.2.3. Định hướng phát triển sản phẩm trong tương lai .....	30
<b>2.3. Tổng hợp và định hướng thiết kế.....</b>	<b>30</b>
2.3.1. Tóm tắt các yếu tố sẽ áp dụng cho dự án .....	30
2.3.2. Lý do chọn framework, công cụ và phong cách thiết kế .....	31
2.3.2.1. Lý do chọn framework và công nghệ phát triển.....	31
2.3.2.2. Lý do chọn công cụ hỗ trợ.....	32
2.3.2.3. Lý do chọn phong cách thiết kế.....	32
<b>CHƯƠNG 3: THIẾT KẾ VÀ TRIỂN KHAI WEBSITE .....</b>	<b>33</b>
<b>3.1. Thiết kế giao diện (UI Design).....</b>	<b>33</b>
3.1.1. Phân tích SiteMap .....	33
3.1.2. Bố cục tổng thể.....	35
3.1.3. Wireframe các trang chính.....	36
3.1.4. Bộ màu, font chữ, hình ảnh, biểu tượng.....	38
<b>3.2. Thiết kế trải nghiệm người dùng (UX Design) .....</b>	<b>44</b>
3.2.1. User Flow chính .....	44
3.2.2. Các chức năng chính .....	44
<b>CHƯƠNG 4: KIỂM THỬ, TỐI ƯU HÓA VÀ TRIỂN KHAI.....</b>	<b>50</b>
<b>4.1. Kiểm thử giao diện và chức năng.....</b>	<b>50</b>

4.1.1. Kiểm tra từng chức năng.....	50
4.1.2. Kiểm thử responsive trên nhiều thiết bị .....	50
<b>CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>	<b>54</b>
<b>5.1. Kết quả đạt được .....</b>	<b>54</b>
5.1.1. Những tiêu chí rubric đã hoàn thành.....	54
5.1.2. Mức độ đạt được so với mục tiêu ban đầu .....	55
5.1.3. Những đóng góp và thành tựu nổi bật.....	55
<b>5.2 Hạn chế .....</b>	<b>55</b>
<b>5.3. Hướng phát triển .....</b>	<b>55</b>
5.3.1. Đề xuất cải tiến về backend và cơ sở dữ liệu.....	55
5.3.2. Đề xuất cải tiến về chức năng người dùng.....	56
5.3.3. Đề xuất cải tiến về quản trị hệ thống .....	57
5.3.4. Đề xuất cải tiến về tính năng nâng cao .....	57
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>59</b>
<b>PHỤ LỤC .....</b>	<b>60</b>





## **BẢNG BIỂU, HÌNH VẼ**

Hình 2.1. HTML5 .....	5
Hình 2.2. Mối liên hệ giữa HTML và CSS.....	11
Hình 2.3. Ứng dụng của Javascript .....	17
Hình 2.4. Giao diện Zing MP3 .....	25
Hình 2.5. Giao diện trang web Spotify .....	27
Hình 3.1. SiteMap của đồ án .....	33
Hình 3.2. Trang Index .....	36
Hình 3.3. Trang home .....	37
Hình 3.4. Trang phát nhạc.....	37
Hình 3.5. Trang karaoke .....	38

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI VÀ PHÂN TÍCH YÊU CẦU

**Tóm tắt chương:** Chương 1 giới thiệu đề tài xây dựng website nghe nhạc trực tuyến sử dụng HTML, CSS và JavaScript. Mục tiêu là thiết kế giao diện responsive và phát triển các tính năng cơ bản như phát nhạc, quản lý playlist, tìm kiếm và đăng nhập, hướng đến đối tượng sinh viên và giới trẻ 12-40 tuổi. Đề tài giúp rèn luyện kỹ năng thiết kế web front-end và tạo sản phẩm thực tế cho portfolio.

## 1.1. Giới thiệu đề tài

Trong thời đại công nghệ số phát triển, âm nhạc trực tuyến đã trở thành nhu cầu giải trí thiết yếu của giới trẻ. Các nền tảng như Spotify, Zing MP3 hay YouTube Music thu hút hàng triệu người dùng nhờ giao diện thân thiện và tính năng đa dạng. Tuy nhiên, việc xây dựng một website nghe nhạc với giao diện đẹp, dễ sử dụng và đầy đủ tính năng vẫn là thách thức đối với người học thiết kế web.

Đề tài "Thiết kế và xây dựng website nghe nhạc trực tuyến" được thực hiện nhằm áp dụng kiến thức HTML, CSS và JavaScript để xây dựng một website nghe nhạc hoàn chỉnh, bao gồm các tính năng phát nhạc, tạo playlist, tìm kiếm bài hát, đăng nhập và hiển thị lời bài hát.

## 1.2. Mục tiêu đề tài

- Thiết kế giao diện website nghe nhạc trực quan, thẩm mỹ và responsive
- Xây dựng các tính năng: phát nhạc, tạo và quản lý playlist, tìm kiếm bài hát, đăng nhập người dùng, hiển thị lyrics
- Lập trình front-end sử dụng HTML5, CSS3 và JavaScript
- Tối ưu trải nghiệm người dùng và đảm bảo tương thích đa thiết bị
- Triển khai website lên host để truy cập trực tuyến.

## 1.3. Mục đích nghiên cứu

Nghiên cứu này nhằm củng cố kiến thức về thiết kế web và lập trình front-end thông qua việc xây dựng một dự án thực tế. Đồng thời, phát triển kỹ năng phân tích yêu cầu, thiết kế giao diện, và triển khai các tính năng tương tác phức tạp bằng HTML, CSS và JavaScript.

#### 1.4. Đối tượng nghiên cứu

**Người dùng mục tiêu:** Sinh viên và giới trẻ từ 12 - 40 tuổi, có nhu cầu nghe nhạc trực tuyến và quản lý bài hát yêu thích.

**Nội dung nghiên cứu:** Giao diện người dùng, các tính năng (phát nhạc, playlist, tìm kiếm, đăng nhập, lyrics), và kỹ thuật lập trình front-end.

#### 1.5. Phạm vi nghiên cứu và giới hạn

Đề tài tập trung vào việc xây dựng giao diện và các chức năng cơ bản của một trang web nghe nhạc trực tuyến. Cụ thể gồm các phạm vi sau:

- **Thiết kế các trang giao diện chính:**
  - Trang chủ (Home)
  - Trang Thư viện (Library)
  - Trang Tìm kiếm (Search)
  - Trang phát nhạc (Music Player)
  - Trang Quản lý Playlist
  - Trang Đăng nhập/Đăng ký
- **Phát triển các chức năng chính của hệ thống:**
  - Chức năng phát nhạc: *play, pause, next, previous*.
  - Tạo và quản lý playlist cá nhân.
  - Tìm kiếm bài hát theo tên hoặc nghệ sĩ.
  - Chức năng đăng nhập/đăng ký đơn giản.
  - Hiện thị lời bài hát (lyrics).

#### Về công nghệ

Trong phạm vi môn học, đề tài giới hạn ở các công nghệ front-end cơ bản:

- **Ngôn ngữ sử dụng:**
  - HTML5 để xây dựng cấu trúc trang
  - CSS3 để trình bày và tạo hiệu ứng giao diện
  - JavaScript để xử lý logic và các tương tác trên trang
- **Dữ liệu:**

Sử dụng dữ liệu file JSON đơn giản để mô phỏng danh sách bài hát, thông tin người dùng, playlist,...

### **Giới hạn:**

Dự án có một số giới hạn sau:

- Không thể xây dựng hệ thống phụ trợ phức tạp với cơ sở dữ liệu thực, xử lý phía máy chủ
- Không phát triển công nghệ phát trực tuyến thực tế hay CDN cho việc phát nhạc
- Không đi sâu vào vấn đề bản quyền âm nhạc, chỉ sử dụng mẫu nhạc không vi phạm bản quyền

## **1.6. Phương pháp nghiên cứu**

- **Nghiên cứu tài liệu:** Thu thập tài liệu về thiết kế web, HTML/CSS/JavaScript, nghiên cứu các website nghe nhạc hiện có
- **Phân tích:** Phân tích yêu cầu người dùng, đánh giá các tính năng cần thiết
- **Thiết kế:** Xây dựng wireframe, mockup và user flow
- **Lập trình và triển khai:** Sử dụng HTML5, CSS3, JavaScript để xây dựng giao diện và tính năng, testing trên nhiều trình duyệt

## **1.7. Ý nghĩa thực tiễn của đề tài**

Đề tài giúp sinh viên rèn luyện kỹ năng thiết kế và lập trình web front-end, tạo sản phẩm hoàn chỉnh bổ sung vào portfolio. Sản phẩm có thể phát triển thành website thực tế và đáp ứng nhu cầu giải trí của sinh viên, giới trẻ.

## **1.8. Cấu trúc cuốn báo cáo đồ án**

Báo cáo đồ án được cấu trúc thành các chương:

Chương 1: GIỚI THIỆU ĐỀ TÀI VÀ PHÂN TÍCH YÊU CẦU

Chương 2: CƠ SỞ LÝ THUYẾT VÀ THỰC NGHIỆM

Chương 3: THIẾT KẾ VÀ TRIỂN KHAI WEBSITE

Chương 4: KIỂM THỬ, TỐI ƯU HÓA VÀ TRIỂN KHAI WEBSITE LÊN HOST

Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**Kết luận chương:** *Chương 1 đã xác định rõ mục tiêu xây dựng website nghe nhạc trực tuyến với các tính năng cốt lõi và công nghệ front-end cơ bản. Đối tượng người dùng,*

*phạm vi nghiên cứu và phương pháp thực hiện đã được làm rõ, tạo nền tảng cho quá trình thiết kế và phát triển. Những phân tích này sẽ là cơ sở để chương tiếp theo nghiên cứu các khái niệm lý thuyết về HTML, CSS, JavaScript cũng như phân tích các website nghe nhạc hiện có, từ đó đưa ra giải pháp kỹ thuật phù hợp cho dự án.*

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ THỰC NGHIỆM

**Tóm tắt chương:** trình bày các kiến thức nền tảng về HTML, CSS, JavaScript và mô hình Client–Server phục vụ cho việc xây dựng web nghe nhạc. Chương cũng giới thiệu nguyên tắc UI/UX và cách sử dụng thẻ audio để phát nhạc trên trình duyệt. Phần thực nghiệm mô tả các công cụ và kỹ thuật được áp dụng để phát triển giao diện và các chức năng chính của hệ thống.

### 2.1. Cơ sở lý thuyết

#### 2.1.1. Ngôn ngữ HTML5

##### 2.1.1.1. Khái niệm

HTML5 là viết tắt của HyperText Markup Language 5. Đây là phiên bản thứ năm và cũng là phiên bản mới nhất, hoàn thiện nhất của HTML. Về cơ bản, nó không phải là một thứ gì đó hoàn toàn mới, mà là một bản nâng cấp toàn diện từ phiên bản HTML4 trước đó. HTML5 được tạo ra để giải quyết các vấn đề của web hiện đại, giúp các nhà phát triển xây dựng website và ứng dụng web mạnh mẽ hơn, đa phương tiện hơn và thân thiện với thiết bị di động hơn.



Hình 2.1. HTML5

HTML5 ra đời từ sự cần thiết phải cải tiến. Phiên bản HTML4 đã trở nên lỗi thời trước sự bùng nổ của video, âm thanh và các ứng dụng web tương tác. Trước đây, để

xem video, người dùng thường phải cài thêm các plugin như Adobe Flash. Điều này tạo ra nhiều rào cản về bảo mật và tương thích.

HTML5 xuất hiện như một cuộc cách mạng. Nó tích hợp sẵn khả năng xử lý đa phương tiện, đồ họa và lưu trữ dữ liệu ngay trong trình duyệt. Điều này giúp loại bỏ sự phụ thuộc vào plugin, tạo ra một trải nghiệm web mượt mà, đồng nhất trên mọi thiết bị từ máy tính đến điện thoại di động.

#### **2.1.1.2. Sự khác nhau giữa HTML và HTML5**

Sự khác nhau giữa HTML và HTML5 thể hiện rõ rệt ở nhiều khía cạnh, giúp việc phát triển web trở nên đơn giản và hiệu quả hơn rất nhiều.

##### **Sự khác biệt về cú pháp khai báo DOCTYPE**

**DOCTYPE** là dòng khai báo đầu tiên trong một tệp HTML, giúp trình duyệt hiểu phiên bản HTML đang được sử dụng. Đây là khác biệt dễ nhận thấy nhất.

- HTML4: Cú pháp rất dài và phức tạp, khó nhớ.  
`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">`
- HTML5: Cú pháp được rút gọn tối đa, vô cùng đơn giản.  
`<!DOCTYPE html>`

Chỉ với thay đổi nhỏ này, HTML5 đã cho thấy tinh thần hướng đến sự đơn giản và hiện đại.

##### **Khả năng hỗ trợ đa phương tiện**

Đây là một bước tiến vượt bậc. Với HTML, việc chèn âm thanh và video yêu cầu các plugin của bên thứ ba (như Flash). HTML5 đã thay đổi cuộc chơi bằng cách giới thiệu các thẻ `<audio>` và `<video>`, cho phép tích hợp đa phương tiện trực tiếp vào trang web một cách tự nhiên.

##### **Đồ họa Vector và Canvas**

HTML5 mang đến hai cách mạnh mẽ để xử lý đồ họa:

- SVG (Scalable Vector Graphics): Hỗ trợ đồ họa vector, giúp hình ảnh không bị vỡ nét khi phóng to.



- Thẻ <canvas>: Cho phép vẽ đồ họa 2D và 3D động bằng JavaScript. Đây là nền tảng cho nhiều game và ứng dụng đồ họa trên trình duyệt.

### Lưu trữ dữ liệu trên trình duyệt

Trước kia, lập trình viên chỉ có thể dùng Cookies với dung lượng lưu trữ rất hạn chế (khoảng 4KB). HTML5 giới thiệu Web Storage (LocalStorage và SessionStorage) với khả năng lưu trữ lớn hơn nhiều (khoảng 5-10MB), an toàn hơn và không bị gửi đi trong mỗi yêu cầu HTTP, giúp tăng tốc độ tải trang.

Tiêu chí	HTML (HTML4)	HTML5
DOCTYPE	Dài và phức tạp	Ngắn gọn: <!DOCTYPE html>
Audio & Video	Cần plugin (Flash, Silverlight)	Hỗ trợ trực tiếp bằng thẻ <audio>, <video>
Đồ họa	Hạn chế, phụ thuộc vào plugin	Hỗ trợ SVG và thẻ <canvas>
Lưu trữ	Chỉ có Cookies (giới hạn 4KB)	Web Storage (localStorage, sessionStorage) 5-10MB
Ngữ nghĩa	Hạn chế (chủ yếu dùng <div>)	Nhiều thẻ ngữ nghĩa mới (<article>, <section>,...)
Form	Các loại input cơ bản	Nhiều loại input mới (email, date, number...)

Bảng 2.1. Bảng so sánh HTML và HTML5

#### 2.1.1.3. Các tính năng mới của HTML5

HTML5 không chỉ là một bản cập nhật. Nó là một bộ công cụ hoàn toàn mới giúp bạn xây dựng các trang web thông minh và tương tác hơn. Dưới đây là những **tính năng mới của HTML5** mà bất kỳ lập trình viên nào cũng cần nắm vững.

## Các thẻ ngữ nghĩa mới

Trước đây, các lập trình viên thường lạm dụng thẻ <div> cho mọi thứ, khiến mã nguồn trở nên khó đọc và khó cho các công cụ tìm kiếm hiểu được cấu trúc trang. HTML5 đã giới thiệu các thẻ mới trong HTML5 mang ý nghĩa rõ ràng:

- **<header>**: Dành cho phần đầu trang hoặc đầu một khu vực.
- **<footer>**: Dành cho phần chân trang.
- **<nav>**: Chứa các liên kết điều hướng chính.
- **<article>**: Đại diện cho một nội dung độc lập, hoàn chỉnh (ví dụ: một bài blog, một sản phẩm).
- **<section>**: Nhóm các nội dung có liên quan với nhau.
- **<aside>**: Dành cho nội dung phụ, bên lề (ví dụ: sidebar).

## Tích hợp trực tiếp Audio và Video không cần plugin

Thẻ <audio> và <video> là một cuộc cách mạng. Bạn có thể dễ dàng nhúng một tệp âm thanh hoặc video vào trang web của mình chỉ với một dòng mã đơn giản. Bạn cũng có thể điều khiển chúng bằng JavaScript để tạo ra các trình phát media tùy chỉnh.

## Phần tử canvas cho đồ họa 2D/3D động

Thẻ <canvas> tạo ra một “khung vẽ” trên trang web. Sử dụng JavaScript, bạn có thể vẽ hình, tạo hoạt ảnh, xử lý ảnh, và thậm chí xây dựng các trò chơi phức tạp ngay trên trình duyệt mà không cần bất kỳ công nghệ bổ sung nào.

## API lưu trữ Web: LocalStorage và SessionStorage

Đây là giải pháp thay thế hiện đại cho Cookies.

- **LocalStorage**: Dữ liệu được lưu trữ vĩnh viễn trên trình duyệt của người dùng, không có ngày hết hạn. Nó sẽ tồn tại ngay cả khi người dùng đóng trình duyệt.
- **SessionStorage**: Dữ liệu chỉ tồn tại trong một phiên làm việc. Khi người dùng đóng tab hoặc cửa sổ trình duyệt, dữ liệu sẽ bị xóa.

## Các loại input mới và thuộc tính cho Form

HTML5 giúp việc tạo biểu mẫu (form) trở nên dễ dàng và thân thiện hơn với người dùng. Các loại input mới giúp trình duyệt tự động xác thực dữ liệu:

- **type="email"**: Tự động kiểm tra định dạng email.

- `type="number"`: Hiển thị bàn phím số trên di động.
- `type="date"`: Hiển thị công cụ chọn ngày tháng.
- `type="url"`, `type="tel"`, `type="range"`, ...

Ngoài ra, các thuộc tính mới như `placeholder`, `required`, `autofocus` cũng giúp nâng cao trải nghiệm người dùng khi điền form.

#### 2.1.1.4 Cấu trúc của một trang HTML5 cơ bản

##### Các thành phần chính:

Một trang HTML5 hoàn chỉnh luôn có 4 thành phần cốt lõi sau:

- `<!DOCTYPE html>`: Dòng khai báo bắt buộc, cho trình duyệt biết đây là tài liệu HTML5.
- `<html>`: Thẻ gốc bao bọc toàn bộ nội dung của trang web. Bạn có thể thêm thuộc tính `lang` để chỉ định ngôn ngữ, ví dụ: `<html lang="vi">`.
- `<head>`: Phần “bộ não” của trang web. Nó chứa các thông tin meta không hiển thị trực tiếp cho người dùng như:
  - `<title>`: Tiêu đề của trang, hiển thị trên tab trình duyệt.
  - `<meta>`: Cung cấp thông tin về trang (bảng mã ký tự, mô tả, từ khóa).
  - `<link>`: Dùng để liên kết đến các tệp CSS.
  - `<script>`: Dùng để liên kết đến các tệp JavaScript.
- `<body>`: Phần “cơ thể” của trang web. Tất cả nội dung mà người dùng nhìn thấy (văn bản, hình ảnh, video, liên kết...) đều được đặt trong thẻ này.

#### 2.1.1.5 Ưu và nhược điểm của HTML5

Nắm rõ ưu nhược điểm của HTML5 sẽ giúp người dùng sử dụng nó một cách hiệu quả và lường trước được các vấn đề có thể phát sinh.

##### Ưu điểm:

- **Hỗ trợ đa nền tảng:** Code HTML5 hoạt động tốt trên hầu hết các trình duyệt hiện đại và thiết bị, từ máy tính để bàn, laptop đến máy tính bảng và điện thoại thông minh.
- **Mã nguồn sạch sẽ và rõ ràng:** Với các thẻ ngữ nghĩa, cấu trúc trang web trở nên dễ đọc, dễ bảo trì và thân thiện hơn với SEO.
- **Trải nghiệm người dùng tốt hơn:** Các tính năng như form thông minh, hỗ trợ video/audio trực tiếp và khả năng lưu trữ offline giúp tạo ra các trang web nhanh, mượt và tiện lợi hơn.
- **Khả năng tương thích ngược:** HTML5 được thiết kế để các trình duyệt cũ vẫn có thể hiểu và hiển thị được cấu trúc cơ bản, dù không hỗ trợ các tính năng mới.
- **Giảm sự phụ thuộc vào plugin:** Việc loại bỏ Flash giúp tăng cường bảo mật và hiệu suất cho trang web.

#### **Nhược điểm:**

- Trình duyệt cũ không hỗ trợ: Các trình duyệt rất cũ (như Internet Explorer 8 trở về trước) không hỗ trợ hoặc hỗ trợ không đầy đủ các tính năng mới của HTML5. Tuy nhiên, ngày nay, đây không còn là vấn đề lớn khi người dùng đã dần chuyển sang các trình duyệt hiện đại.
- Tính năng không đồng nhất: Đôi khi, cách một tính năng mới hoạt động có thể hơi khác nhau giữa các trình duyệt (ví dụ: giao diện của input type="date" trên Chrome và Firefox). Lập trình viên cần kiểm tra kỹ lưỡng để đảm bảo trải nghiệm đồng nhất.

### **2.1.2. Ngôn ngữ CSS**

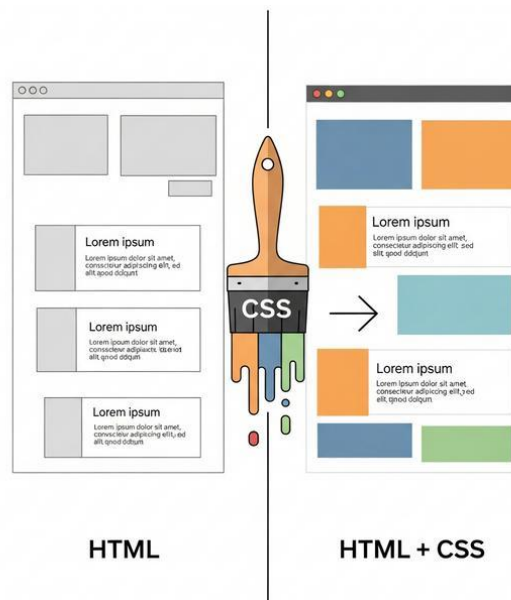
#### **2.1.2.1. Khái niệm**

CSS là viết tắt của cụm từ tiếng Anh **Cascading Style Sheets**. Dịch nôm na, nó có nghĩa là các bảng kiểu xếp chồng. Nghe có vẻ hơi học thuật phải không? Đừng lo, bạn chỉ cần hiểu đơn giản CSS là một ngôn ngữ được dùng để tạo phong cách và định dạng cho các trang web.

Mục đích cốt lõi của CSS là định hình **diện mạo** của một website. Nó quyết định mọi thứ liên quan đến phần nhìn, bao gồm:

- **Màu sắc:** Màu chữ, màu nền, màu của các đường viền.
- **Font chữ:** Kiểu chữ, kích thước, độ đậm nhạt.
- **Bố cục (Layout):** Vị trí các thành phần trên trang, khoảng cách giữa chúng, chia cột.
- **Hiệu ứng:** Đồ bóng, bo góc, các hiệu ứng chuyển động khi di chuột.

Nói cách khác, nếu HTML tạo ra cấu trúc và nội dung thô (như các đoạn văn, tiêu đề, hình ảnh), thì CSS sẽ khoác lên cho chúng một tấm áo đẹp mắt, sắp xếp chúng một cách gọn gàng và làm cho trang web trở nên chuyên nghiệp, thân thiện hơn với người dùng.



Hình 2.2. Mối liên hệ giữa HTML và CSS

#### 2.1.2.2. Vai trò của CSS trong mối liên hệ giữa HTML

HTML và CSS luôn là một cặp bài trùng trong lĩnh vực phát triển web. Chúng có mối quan hệ cộng sinh, không thể tách rời để tạo nên một trang web hoàn chỉnh. Hiểu rõ vai trò của từng ngôn ngữ sẽ giúp bạn xây dựng website hiệu quả hơn.

Để dễ hình dung, bạn hãy nghĩ về việc xây một ngôi nhà:

- HTML (HyperText Markup Language) giống như bộ khung của ngôi nhà. Nó xác định các thành phần cơ bản như nền móng, tường, cột, mái. Trên web, HTML tạo

ra các yếu tố như tiêu đề (<h1>), đoạn văn (<p>), hình ảnh (<img>). Nó chỉ tập trung vào cấu trúc và nội dung.

- CSS (Cascading Style Sheets) chính là phần nội thất và sơn trang trí. Nó quyết định màu sơn tường, loại gạch lát sàn, kiểu dáng cửa sổ. Tương tự, CSS sẽ quy định màu sắc của tiêu đề, font chữ của đoạn văn, vị trí của hình ảnh. Nó tập trung hoàn toàn vào trình bày và phong cách.

Một trang web có thể tồn tại chỉ với HTML, nhưng nó sẽ trông rất đơn điệu. CSS là yếu tố thổi hồn, mang lại sự sống động và thẩm mỹ cho cấu trúc đó.

### **Tách biệt nội dung và trình bày**

Một trong những vai trò của CSS quan trọng nhất là giúp tách biệt hoàn toàn phần nội dung (HTML) ra khỏi phần trình bày (CSS). Điều này mang lại rất nhiều lợi ích:

- Mã nguồn sạch sẽ: Tập HTML của bạn sẽ gọn gàng hơn, chỉ chứa nội dung và cấu trúc, không bị rối bởi các thẻ định dạng.
- Dễ dàng bảo trì: Khi bạn muốn thay đổi màu sắc chủ đạo của toàn bộ trang web, bạn chỉ cần sửa một vài dòng trong tệp CSS thay vì phải chỉnh sửa từng trang HTML một.
- Tái sử dụng mã: Bạn có thể dùng chung một tệp CSS cho nhiều trang khác nhau, đảm bảo tính nhất quán về giao diện trên toàn bộ website.

### **2.1.2.3. Cấu trúc và cú pháp CSS cơ bản**

#### **Selector**

**Selector** đơn giản là cách bạn “chỉ” cho trình duyệt biết bạn muốn tạo kiểu cho phần tử HTML nào. Đó có thể là một tiêu đề h1, một đoạn văn p, hoặc một phần tử có class hay id đặc biệt. Selector đứng ở vị trí đầu tiên của một quy tắc CSS.

#### **Property và Value**

**Declaration Block** là phần nằm trong cặp dấu ngoặc nhọn {}. Bên trong khối này chứa một hoặc nhiều khai báo (declaration) và mỗi khai báo được kết thúc bằng dấu chấm phẩy ;.

Mỗi khai báo lại gồm hai phần:

- **Property (Thuộc tính):** Là đặc tính bạn muốn thay đổi. Ví dụ: color (màu chữ), font-size (kích thước chữ), background-color (màu nền).
- **Value (Giá trị):** Là giá trị bạn muốn gán cho thuộc tính đó. Ví dụ: blue, 16px, #FFFFFF.

Thuộc tính và giá trị được ngăn cách bởi dấu hai chấm :

#### 2.1.2.4. Phân loại CSS

##### External CSS

Bạn sẽ viết tất cả các quy tắc CSS vào một tệp riêng biệt có đuôi là .css (ví dụ: style.css). Sau đó, bạn liên kết tệp này vào tệp HTML bằng thẻ <link> đặt trong phần <head>.

Ví dụ:

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

- **Ưu điểm:** Dễ dàng quản lý và bảo trì, có thể tái sử dụng cho nhiều trang, giúp tệp HTML gọn gàng.
- **Nhược điểm:** Cần tải thêm một tệp, có thể làm tăng nhẹ thời gian tải trang lần đầu.

##### Internal CSS

Với cách này, bạn sẽ viết mã CSS trực tiếp bên trong cặp thẻ <style> và đặt chúng trong phần <head> của tệp HTML. Cách này hữu ích khi bạn chỉ có một trang duy nhất cần phong cách đặc biệt mà không áp dụng cho các trang khác.

Ví dụ:

```
<head>
  <style>
    body {
      background-color: lightblue;
    }
    h1 {
```

```

        color: navy;
    }
</style>
</head>

```

- **Ưu điểm:** Tất cả mã nguồn nằm trong một tệp, không cần tải tệp CSS riêng.
- **Nhược điểm:** Khó quản lý nếu có nhiều trang, không thể tái sử dụng cho các trang khác.

## Inline CSS

Inline CSS là cách thêm thuộc tính style trực tiếp vào một thẻ HTML cụ thể. Cách này chỉ ảnh hưởng đến duy nhất phần tử đó. Nó thường được dùng để áp dụng một thay đổi nhỏ, nhanh chóng hoặc ghi đè các quy tắc từ External hay Internal CSS.’

### 2.1.2.5. CSS3 Phiên bản tối ưu nhất hiện tại

Không giống như các phiên bản trước, CSS3 không phải là một đặc tả duy nhất. Thay vào đó, nó được chia thành các mô-đun (modules) độc lập. Cách tiếp cận này cho phép các tính năng mới được phát triển và chuẩn hóa nhanh hơn.

Một số mô-đun nổi bật của CSS3 bao gồm:

- **Selectors:** Các bộ chọn mạnh mẽ hơn.
- **Box Model:** Mô hình hộp linh hoạt hơn.
- **Backgrounds and Borders:** Cho phép tạo nền và viền phức tạp hơn (ví dụ: bo tròn góc, đổ bóng).
- **2D/3D Transformations:** Khả năng xoay, co giãn, di chuyển các phần tử.
- **Animations & Transitions:** Tạo hiệu ứng chuyển động mượt mà.
- **Flexible Box Layout (Flexbox) và Grid Layout:** Hai hệ thống bố cục cực kỳ mạnh mẽ, giúp việc sắp xếp các phần tử trên trang trở nên dễ dàng hơn bao giờ hết.

Sự ra đời của CSS3 đã cách mạng hóa lĩnh vực thiết kế web, cho phép các nhà phát triển tạo ra những giao diện phức tạp, linh hoạt và sống động mà không cần đến hình ảnh hay JavaScript phức tạp.



### **2.1.2.6. Ưu điểm khi sử dụng CSS trong thiết kế web**

CSS không chỉ là một tiêu chuẩn mà còn mang lại rất nhiều lợi ích thiết thực cho cả nhà phát triển và người dùng cuối. Dưới đây là những ưu điểm nổi bật nhất.

#### **Tiết kiệm thời gian và công sức**

Với CSS, có thể định nghĩa một kiểu cho một phần tử (ví dụ: tất cả các nút bấm) và áp dụng nó trên toàn bộ website. Khi cần thay đổi, bạn chỉ cần chỉnh sửa ở một nơi duy nhất. Điều này tiết kiệm hàng giờ so với việc phải thay đổi thủ công trên từng trang.

#### **Giúp trang web tải nhanh hơn**

Khi bạn sử dụng External CSS, trình duyệt chỉ cần tải tệp .css một lần. Đối với các lần truy cập trang tiếp theo, tệp này sẽ được lưu vào bộ nhớ đệm (cache). Điều này làm giảm đáng kể dung lượng của các tệp HTML, từ đó giúp tốc độ tải trang được cải thiện rõ rệt.

#### **Dễ dàng bảo trì và cập nhật**

Việc tách biệt mã HTML và CSS giúp mã nguồn trở nên có tổ chức và dễ đọc hơn. Khi bạn cần sửa lỗi hiển thị hoặc nâng cấp giao diện, bạn biết chính xác cần phải tìm ở đâu. Việc bảo trì và cập nhật website trở nên nhanh chóng và ít rủi ro hơn.

#### **Cải thiện khả năng tương thích đa thiết bị**

Với sự trợ giúp của CSS Media Queries, bạn có thể tạo ra các thiết kế đáp ứng (Responsive Design). Trang web của bạn có thể tự động điều chỉnh bố cục để hiển thị tối ưu trên mọi loại màn hình, từ máy tính để bàn, máy tính bảng cho đến điện thoại di động. Điều này cực kỳ quan trọng trong việc cải thiện trải nghiệm người dùng.

## **2.1.3. Ngôn ngữ lập trình JavaScript**

### **2.1.3.1 Khái niệm**

**JavaScript**, hay thường được viết tắt là **JS**, là một ngôn ngữ lập trình kịch bản. Bạn có thể hiểu đơn giản nó là công cụ giúp bạn “ra lệnh” cho trình duyệt web thực hiện các hành động. Mục tiêu chính của nó là tạo ra các trang web có tính tương tác cao.

### **2.1.3.2. Ứng dụng của JavaScript**

#### **Phát triển Front-end**

Đây là ứng dụng gốc và phổ biến nhất. JavaScript được dùng để:

- Tạo hiệu ứng động, slideshow, menu xổ xuống.
- Xác thực dữ liệu trong biểu mẫu (form) ngay trên trình duyệt mà không cần tải lại trang.
- Tải dữ liệu mới từ máy chủ và cập nhật một phần trang web (ví dụ: newfeed trên Facebook).
- Xây dựng các Giao diện Người dùng (UI) phức tạp với các thư viện như React hay Vue.js.

### **Xây dựng máy chủ và Back-end với Node.js**

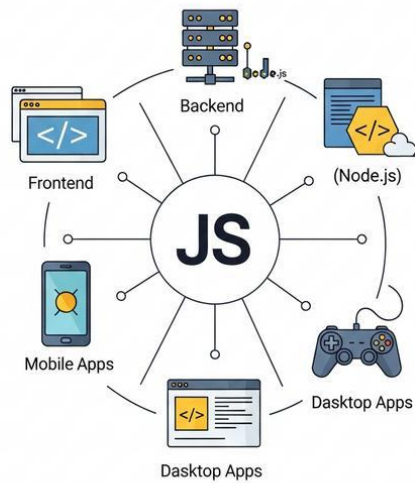
Với sự ra đời của Node.js, JavaScript đã có thể chạy trên máy chủ (server-side). Điều này cho phép các lập trình viên sử dụng cùng một ngôn ngữ cho cả front-end và back-end. Node.js rất mạnh mẽ trong việc xây dựng các API, ứng dụng thời gian thực (như ứng dụng chat), và các microservices.

### **Lập trình ứng dụng di động**

Các framework như React Native (do Facebook phát triển) cho phép lập trình viên xây dựng ứng dụng di động cho cả iOS và Android chỉ từ một mã nguồn JavaScript duy nhất. Điều này giúp tiết kiệm đáng kể thời gian và chi phí phát triển.

### **Phát triển game trên trình duyệt**

Nhiều game 2D và cả 3D đơn giản chạy trên trình duyệt cũng được tạo ra bằng JavaScript, kết hợp với các công nghệ như HTML5 Canvas và WebGL. Các thư viện như Phaser.js hay Three.js giúp việc này trở nên dễ dàng hơn bao giờ hết.



Hình 2.3. Ứng dụng của Javascript

### 2.1.3.3. Cách thức hoạt động của JavaScript

JavaScript có thể hoạt động ở hai nơi chính: trên trình duyệt của người dùng hoặc trên máy chủ. Cách thức hoạt động ở mỗi nơi có sự khác biệt rõ rệt.

#### Phía Client

Đây là cách hoạt động truyền thống. Khi bạn truy cập một website, mã JavaScript sẽ được tải về cùng với HTML và CSS. Sau đó, trình duyệt web của bạn (như Chrome, Firefox) sẽ sử dụng một “engine” tích hợp sẵn để đọc và thực thi đoạn mã này.

#### Phía Server

Nhờ có môi trường Node.js, JavaScript giờ đây có thể chạy trên máy chủ. Trong trường hợp này, mã JS sẽ được thực thi trên server để xử lý các logic phức tạp như truy vấn cơ sở dữ liệu, xác thực người dùng, hoặc xử lý các nghiệp vụ kinh doanh. Sau khi xử lý xong, máy chủ sẽ gửi kết quả (thường là HTML, CSS, và một ít JS phía client) về cho trình duyệt của người dùng hiển thị.

### 2.1.3.4. Ưu và nhược điểm của ngôn ngữ JavaScript

#### Ưu điểm:

- Dễ tiếp cận: Cú pháp của javascript cơ bản khá đơn giản và dễ hiểu, là một lựa chọn tuyệt vời cho người mới bắt đầu.

- Phổ biến và cộng đồng lớn: JS có một cộng đồng người dùng khổng lồ. Bạn có thể dễ dàng tìm thấy tài liệu, các khóa học và sự giúp đỡ khi gặp khó khăn.
- Linh hoạt: Nó có thể chạy trên cả client và server, dùng để phát triển web, mobile, desktop và nhiều hơn nữa.
- Hệ sinh thái phong phú: Có vô số thư viện và framework (React, Angular, Vue.js) giúp đẩy nhanh tốc độ phát triển và xây dựng các ứng dụng mạnh mẽ.
- Tốc độ thực thi nhanh: Vì chạy trực tiếp trên trình duyệt, các tác vụ phía client được thực hiện gần như tức thì, mang lại trải nghiệm tốt cho người dùng.

#### **Nhược điểm:**

- Bảo mật phía client: Mã JavaScript chạy trên trình duyệt có thể bị người dùng xem và chỉnh sửa. Vì vậy, không nên xử lý các thông tin nhạy cảm ở phía client.
- Tính nhất quán giữa các trình duyệt: Mặc dù vấn đề này đã được cải thiện rất nhiều, đôi khi vẫn có sự khác biệt nhỏ trong cách các trình duyệt khác nhau thực thi mã JS.
- Hiệu năng: Với các tác vụ tính toán quá phức tạp, JavaScript có thể không nhanh bằng các ngôn ngữ biên dịch như C++ hay Java.

### **2.1.4. Khái niệm và nguyên tắc UI/UX, responsive design, accessibility**

#### **2.1.4.1 UI/UX (User Interface / User Experience)**

##### **Khái niệm**

**UI (User Interface - Giao diện người dùng):** Là các yếu tố trực quan mà người dùng nhìn thấy và tương tác (màu sắc, font chữ, icon, nút bấm, bố cục). Mục tiêu là làm cho giao diện đẹp mắt, hấp dẫn và dễ sử dụng.

**UX (User Experience - Trải nghiệm người dùng):** Là cảm nhận và cảm xúc tổng thể của người dùng khi sử dụng sản phẩm. Nó liên quan đến tính dễ sử dụng, hiệu quả, logic của sản phẩm. UX tốt giúp người dùng đạt mục tiêu một cách mượt mà.

##### **Nguyên tắc UI/UX:**

- **Lấy người dùng làm trung tâm:** Nghiên cứu khách hàng để hiểu nhu cầu, mong muốn của họ.

- **Tính nhất quán (Consistency):** Giao diện và hành vi phải đồng bộ trên toàn bộ sản phẩm.
- **Phản hồi rõ ràng (Feedback):** Hệ thống phải phản hồi lại hành động của người dùng (ví dụ: nút đổi màu khi bấm).
- **Đơn giản & Rõ ràng (Clarity & Simplicity):** Tránh làm người dùng rối bời với quá nhiều yếu tố không cần thiết.
- **Phân cấp thị giác (Visual Hierarchy):** Sắp xếp thông tin quan trọng nổi bật hơn.

#### 2.1.4.2. Responsive Design (Thiết kế tương thích/phản hồi)

Khái niệm

Kỹ thuật thiết kế giao diện có khả năng tự động co giãn, thay đổi bố cục để hiển thị tối ưu trên mọi kích thước màn hình (desktop, tablet, mobile).

Nguyên tắc:

- **Ưu tiên mobile-first:** Thiết kế cho thiết bị di động trước rồi mở rộng lên các màn hình lớn hơn.
- **Sử dụng lưới (Grid) linh hoạt:** Các cột, khoảng cách tự động điều chỉnh.
- **Hình ảnh & Media co giãn:** Hình ảnh tự điều chỉnh kích thước theo màn hình.
- **Điều hướng linh hoạt:** Menu có thể ẩn/hiện (hamburger menu) trên mobile.

#### 2.1.4.3. Accessibility (Khả năng truy cập)

Khái niệm: Thiết kế sản phẩm để tất cả mọi người, bao gồm cả người khuyết tật (giảm thị lực, nghe, vận động, nhận thức), có thể truy cập và sử dụng dễ dàng.

Nguyên tắc:

- **Cung cấp văn bản thay thế (Alt text) cho hình ảnh:** Giúp người dùng đọc màn hình hiểu nội dung.
- **Độ tương phản màu sắc đủ cao:** Cho người có thị lực kém dễ đọc.
- **Bàn phím điều hướng:** Người dùng phải điều khiển được toàn bộ trang web chỉ bằng bàn phím.
- **Phụ đề/Bản ghi (Captions/Transcripts):** Cho video và audio.
- **Cấu trúc ngữ nghĩa rõ ràng:** Sử dụng các thẻ HTML đúng (h1, h2, p...).

## **2.1.5. Một số Framework phổ biến**

### **2.1.5.1. Tailwind CSS**

Tailwind CSS là một framework tương đối mới (phát hành từ 2017) và được thiết kế cùng sự khác biệt so với các CSS framework trước đây. Nếu như các framework CSS trước đây đi theo hướng sử dụng các thành phần được thiết kế từ trước làm cơ sở để phát triển thêm thì Tailwind cung cấp các lớp tiện ích cấp thấp cho phép bạn xây dựng các thiết kế của riêng mình.

#### **Ưu điểm của Tailwind CSS:**

- Mang lại sự linh hoạt cho người dùng có thể tạo ra các thiết kế độc đáo
- Cú pháp của Tailwind cơ bản giúp mang lại tốc độ phát triển nhanh hơn cho dự án, không cần phải chuyển đổi giữa HTML và CSS khi viết code.
- Tối ưu hóa CSS bằng việc sử dụng PurgeCSS giúp giảm và loại bỏ các class dư thừa cũng như tối giản kích thước file CSS tạo ra
- Khả năng tùy chỉnh cao: dễ dàng ghi đè cấu hình mặc định với file `tailwind.config.js`

### **2.1.5.2. Bootstrap**

Tiếp theo trong danh sách CSS Frameworks tốt nhất 2024 chính là Bootstrap. Bootstrap là cái tên phổ biến và được sử dụng rộng rãi nhất trong các CSS Framework. Được phát triển từ năm 2011, hiện nay nó được sử dụng trên hàng triệu Website và là kỹ năng được xem như mặc định phải có của một Frontend Developers trước đây. Bootstrap framework chứa cả HTML, SASS và JavaScript khiến nó có thể xem như là một Frontend framework hoàn chỉnh; mặc dù vậy việc sử dụng nó cũng khiến kích thước của project tăng lên đáng kể.

#### **Ưu điểm của Bootstrap:**

- Hệ sinh thái khổng lồ: Bootstrap cung cấp một thư viện rộng lớn về bố cục, chủ đề, các components, ... dành cho lập trình viên có thể thoải mái sử dụng. Cộng đồng làm việc, sử dụng và đóng góp cho Bootstrap cũng đông đảo và lâu đời hỗ trợ bạn một cách nhanh chóng

- Tạo bản mẫu nhanh: Nhờ việc sử dụng kết hợp cả HTML và JavaScript nên Bootstrap cung cấp cho bạn các bản mẫu hoàn chỉnh, không mất nhiều thời gian để tạo ra hình dạng đầu tiên của Website một cách trực quan nhất
- Bootstrap hỗ trợ cả SASS và LESS, điều mà không nhiều CSS framework cung cấp

### **2.1.5.3. Foundation**

Foundation là một CSS framework được xây dựng dựa trên hai ngôn ngữ là SASS và SCSS giúp cung cấp các khả năng tiếp cận khác nhau nhằm tối ưu hóa mọi hoạt động thiết kế Web nhanh chóng và hiệu quả. Foundation hướng đến kiến trúc mobile-first, có những nét tương đồng với Bootstrap khi cung cấp một bộ UI Framework khá hoàn chỉnh chứa cả HTML và JavaScript.

#### **Ưu điểm của Foundation:**

- Cung cấp bộ công cụ phong phú giải quyết hầu hết các vấn đề mà anh em lập trình cần để dựng Website.
- Tính linh hoạt cao: Foundation không bắt buộc các nhà phát triển phải sử dụng một phong cách hay ngôn ngữ cụ thể mà hoàn toàn có thể tùy chỉnh, kiểm soát giao diện người dùng
- Hỗ trợ GPU acceleration cho phép thực thi các animation một cách mượt mà và nhanh chóng

### **2.1.6. Nguyên tắc phối màu, bố cục, typography trong thiết kế web.**

#### **2.1.6.1. Nguyên tắc phối màu trong thiết kế website**

Màu sắc và typography trong thiết kế website là bộ đôi quyền lực, và phối màu đúng cách chính là chìa khóa để tạo nên một giao diện vừa đẹp mắt vừa hiệu quả. Một bảng màu được xây dựng dựa trên các nguyên tắc chuyên nghiệp không chỉ nâng cao thẩm mỹ mà còn cải thiện trải nghiệm người dùng, củng cố thương hiệu và tối ưu SEO

#### **Quy tắc 60-30-10**

Để tạo sự cân bằng thị giác, quy tắc 60-30-10 là kim chỉ nam không thể bỏ qua trong thiết kế website chuyên nghiệp. Màu chủ đạo chiếm 60% giao diện (như nền trắng hoặc xám nhạt), tạo khung cảnh chính. Màu phụ chiếm 30%, dùng cho các yếu tố như

tiêu đề, menu (ví dụ: xanh dương nhạt). Cuối cùng, 10% là màu nhấn dành cho các chi tiết nổi bật như nút kêu gọi hành động (như đỏ hoặc cam). Quy tắc này đảm bảo website hài hòa, không quá lòe loẹt hay đơn điệu.

### **Sự tương phản màu sắc**

Tương phản là yếu tố sống còn để đảm bảo nội dung dễ đọc và giao diện thân thiện. Theo chuẩn WCAG 2.1, độ tương phản giữa chữ và nền nên đạt ít nhất 4.5:1 cho văn bản thường. Ví dụ, chữ đen trên nền trắng hoặc chữ trắng trên nền xanh đậm là lựa chọn lý tưởng. Tương phản tốt không chỉ giúp tối ưu UX mà còn hỗ trợ người dùng khuyết tật, đồng thời cải thiện khả năng lập chỉ mục của Google, tăng hiệu quả SEO.

### **Tính nhất quán**

Một website chuyên nghiệp cần duy trì sự nhất quán trong cách sử dụng màu sắc. Từ trang chủ đến các trang con, từ banner đến chân trang, màu sắc phải tuân theo bảng màu thương hiệu đã định. Sự nhất quán giúp người dùng dễ dàng điều hướng và ghi nhớ thương hiệu. Ví dụ, nếu bạn chọn xanh lá làm màu chủ đạo, hãy đảm bảo nó xuất hiện đồng đều trong các yếu tố thiết kế, tránh pha trộn màu sắc ngẫu nhiên gây rối mắt.

#### **2.1.6.2. Nguyên tắc typography trong thiết kế website**

##### **Chọn font chữ phù hợp**

Font chữ là “giọng nói” của website, phản ánh cá tính và mục đích thương hiệu. Font Sans-Serif như Roboto hay Open Sans mang cảm giác hiện đại, dễ đọc trên màn hình kỹ thuật số, lý tưởng cho các website công nghệ hoặc thương mại. Ngược lại, font Serif như Georgia hay Times New Roman gợi nét cổ điển, phù hợp với các trang báo chí hoặc học thuật. Hãy giới hạn trong 2-3 font để tránh rối mắt, đảm bảo sự hài hòa và tối ưu UX.

##### **Kích thước và khoảng cách chữ**

Kích thước chữ ảnh hưởng trực tiếp đến khả năng đọc và sự thoải mái của người dùng. Kích thước tối thiểu cho nội dung chính nên là 16px, đặc biệt trên thiết bị di động. Line – height (khoảng cách giữa các dòng) lý tưởng nằm trong khoảng 1.4 Đến 1.6 để tạo sự thoáng đãng. Ngoài ra, điều chỉnh letter-spacing (khoảng cách giữa các ký



tự) tinh tế giúp văn bản trông gọn gàng hơn. Những yếu tố này không chỉ cải thiện trải nghiệm mà còn hỗ trợ SEO bằng cách giữ chân người dùng lâu hơn.

### **Phân cấp nội dung rõ ràng**

Typography hiệu quả cần tạo ra một hệ thống phân cấp trực quan, giúp người dùng dễ dàng điều hướng nội dung. Sử dụng thẻ heading (H1, H2, H3) hợp lý để phân tách các phần, với kích thước giảm dần từ tiêu đề chính đến nội dung phụ. Ví dụ, H1 có thể dùng font đậm, kích thước 32px, trong khi văn bản thường dùng font nhẹ 16px. Phân cấp tốt không chỉ tăng tính thẩm mỹ mà còn giúp Google hiểu cấu trúc website, cải thiện thứ hạng tìm kiếm.

### **Đảm bảo tính tương thích đa thiết bị**

Trong thời đại responsive design, typography cần hiển thị hoàn hảo trên mọi màn hình, từ điện thoại đến máy tính bảng. Hãy chọn font hỗ trợ tốt trên các trình duyệt và kiểm tra cách chúng hiển thị ở các độ phân giải khác nhau. Công cụ như Google Fonts cung cấp các font tối ưu cho web, giúp tải nhanh và hiển thị mượt mà. Tính tương thích này đảm bảo website luôn chuyên nghiệp, bất kể người dùng truy cập từ đâu.

## **2.1.7. Khái niệm về hosting, domain, sever.**

### **2.1.7.1. Domain (Tên miền)**

Tên miền (Domain) là địa chỉ định danh duy nhất của một website trên Internet, cho phép người dùng truy cập vào trang web thay vì phải ghi nhớ địa chỉ IP dạng số. Domain hoạt động như một “địa chỉ nhà” trong không gian mạng, giúp xác định vị trí của trang web một cách dễ hiểu và dễ sử dụng hơn.

Cấu trúc tên miền bao gồm nhiều cấp độ, phổ biến nhất là:

- TLD (Top-Level Domain): các phần mở rộng như .com, .net, .org, .vn,...
- SLD (Second-Level Domain): phần tên được cá nhân hoặc tổ chức đăng ký, ví dụ như trong “example.com”, phần “example” là SLD.

Tên miền giữ vai trò quan trọng trong việc xây dựng thương hiệu, tính nhận diện và khả năng truy cập của hệ thống web.

### **2.1.7.2. Hosting (Web Hosting)**

Hosting là dịch vụ cung cấp không gian lưu trữ trên máy chủ để vận hành website. Khi một website được triển khai, toàn bộ mã nguồn, cơ sở dữ liệu, hình ảnh và tài nguyên tĩnh đều cần được lưu trữ trên một máy chủ có kết nối Internet liên tục. Dịch vụ hosting đảm nhiệm vai trò này, đảm bảo website có thể truy cập 24/7.

Một số loại hình hosting phổ biến:

- Shared Hosting: nhiều website cùng sử dụng tài nguyên chung trên một máy chủ; chi phí thấp, phù hợp dự án nhỏ.
- VPS (Virtual Private Server): máy chủ ảo hóa, cung cấp tài nguyên riêng biệt; phù hợp dự án vừa và lớn.
- Dedicated Server: máy chủ vật lý riêng biệt dành cho một khách hàng; hiệu suất cao.
- Cloud Hosting: sử dụng hạ tầng đám mây, khả năng mở rộng linh hoạt và độ ổn định cao.

Hosting là nền tảng quan trọng để website hoạt động ổn định và đảm bảo tốc độ truy cập, khả năng chịu tải cũng như tính bảo mật.

### **2.1.7.3. Server (Máy chủ)**

Server là hệ thống máy tính có cấu hình đặc biệt, được thiết kế để cung cấp tài nguyên, dữ liệu và dịch vụ cho các máy tính khác trong mạng (gọi là client). Máy chủ có thể tồn tại dưới dạng:

- Máy chủ vật lý (Physical Server): thiết bị phần cứng chuyên dụng.
- Máy chủ ảo (Virtual Server): được tạo ra thông qua công nghệ ảo hóa.
- Máy chủ đám mây (Cloud Server): tài nguyên phân tán trên hạ tầng đám mây.

Server đảm nhiệm nhiều chức năng như:

- Lưu trữ và phân phối dữ liệu;
- Chạy các ứng dụng web;

- Quản lý cơ sở dữ liệu;
- Đảm bảo an ninh và xử lý các yêu cầu từ client.

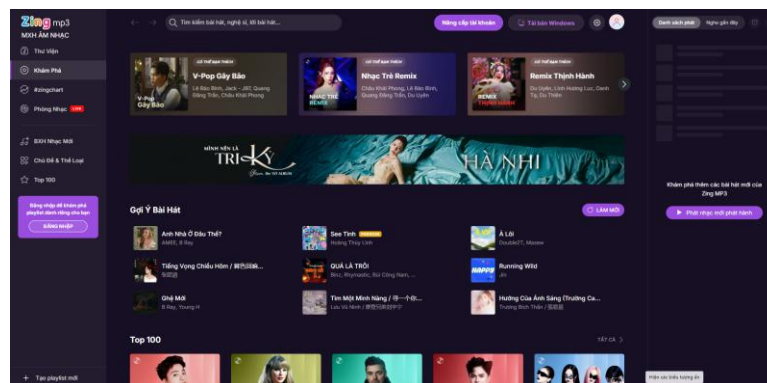
Máy chủ chính là nền tảng vận hành của bất kỳ hệ thống web hoặc dịch vụ trực tuyến nào, giữ vai trò quyết định trong tốc độ, tính ổn định và khả năng mở rộng của hệ thống.

## 2.2. Khảo sát các sản phẩm tương tự

### 2.2.1. Phân tích Website cùng lĩnh vực

Nhằm phục vụ cho quá trình thiết kế và phát triển sản phẩm, nhóm đã tiến hành nghiên cứu, phân tích 2 website tiêu biểu trong lĩnh vực du lịch nghe nhạc trực tuyến, bao gồm: zingmp3.vn, spotify.com. Đây đều là những nền tảng phổ biến, có lượng người dùng lớn và hệ thống dịch vụ đa dạng.

#### 2.2.1.1 ZING MP3 (zingmp3.vn)



Hình 2.4. Giao diện Zing MP3

### Công nghệ & Tính năng kỹ thuật

- **Về chất lượng âm thanh:**
  - Hỗ trợ multiple audio formats: MP3, WAV, FLAC, AAC
  - Chất lượng streaming: Từ 128kbps (miễn phí) đến 320kbps (VIP) và Lossless (.flac)
  - Tích hợp equalizer, bass adjustment, balance, virtualizer và reverb
  - Tối ưu hóa cho các thiết bị khác nhau
- **Nền tảng đa chiều:**
  - Website (zingmp3.vn)
  - Mobile app (iOS, Android)

- Smart TV app
- Hỗ trợ Chromecast và Android Auto
- Widget và music controller trên notification bar và lock screen
- **Tìm kiếm & Khám phá:**
  - Smart voice search hỗ trợ Tiếng Việt
  - Tích hợp Zing Chart (bảng xếp hạng nhạc uy tín)
  - Thường xuyên cập nhật nhạc trending hot
  - Playlist được tạo bởi editors
  - Video music với chất lượng từ 240p đến 1080p

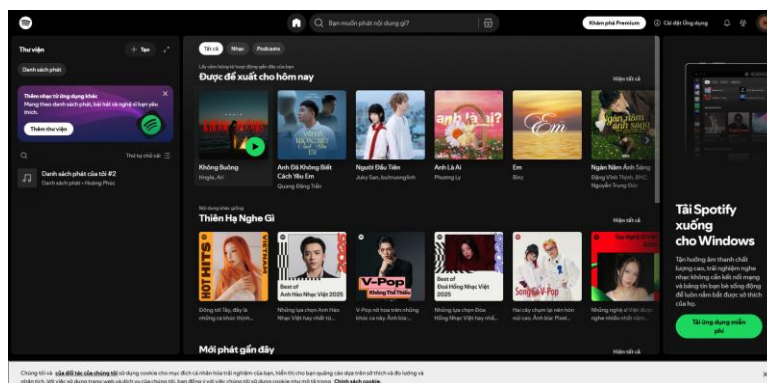
#### **Ưu điểm:**

- **Hoàn toàn miễn phí:** Không yêu cầu đăng ký thanh toán
- **Kho nhạc chất lượng:** Phong phú nhạc Việt với các bài hát độc quyền, hợp tác với các nghệ sĩ nổi tiếng
- **Chất lượng âm thanh:** Hỗ trợ Lossless music, tốt nhất trong các nền tảng Việt
- **Tối ưu hóa cho Việt Nam:** Tìm kiếm bằng Tiếng Việt, giao diện địa phương hóa
- **Tính năng đa nền tảng:** Hoạt động tốt trên nhiều thiết bị
- **Tải nhạc offline:** Người dùng có thể tải nhạc để nghe offline
- **Video music:** Xem video nhạc với chất lượng cao
- **Tốc độ truy cập nhanh**

#### **Nhược điểm:**

- **Tính ổn định:** Một số người dùng báo cáo lỗi ngắt âm nhạc, app crash, đặc biệt khi màn hình tắt
- **Quyền tác quyền:** Một số bài hát bị xóa do vấn đề bản quyền
- **Chính sách hoàn tiền:** Tài khoản VIP/Premium không hoàn tiền
- **Giao diện:** Có thể cần cải thiện tính trực quan trong một số chức năng
- **Khả năng toàn cầu:** Chủ yếu dành cho người dùng Việt Nam

### 2.2.1.2. SPOTIFY (spotify.com)



Hình 2.5. Giao diện trang web Spotify

### Công nghệ & Thuật toán

- **Hệ thống khuyến nghị (Recommendation Engine):**
  - Sử dụng **BaRT (Bandits for Recommendations as Treatments)** - AI recommendation engine
  - Ba công nghệ cơ bản:
    - Collaborative Filtering:** Phân tích hành vi của người dùng, so sánh với những người có sở thích tương tự
    - Natural Language Processing (NLP):** Phân tích từ vựng, lời bài hát, các bài viết trên mạng xã hội để hiểu ngữ cảnh bài hát
    - Audio Analysis:** Phân tích các đặc tính âm thanh (danceability, energy, valence, tempo, loudness, key)
- **Dữ liệu được sử dụng:**
  - Lịch sử nghe nhạc của người dùng (taste profile)
  - Hành vi người dùng: tìm kiếm, skip, lưu, tạo playlist
  - Vị trí địa lý (non-precise), ngôn ngữ, tuổi
  - Người dùng theo dõi
  - Thông tin văn hóa từ internet (trends, sentiment từ mạng xã hội)
- **Những tính năng AI nổi bật (2025):**
  - **Discover Weekly:** Playlist cá nhân hóa hàng tuần dựa trên lịch sử nghe
  - **Release Radar:** Theo dõi bài hát mới từ các nghệ sĩ bạn theo dõi

- **Daylist:** Playlist thay đổi theo thời gian trong ngày (sáng, trưa, tối, đêm)
- **Spotify AI Playlist Generator:** Tạo playlist tùy chỉnh dựa trên prompt từ người dùng (VD: "synthpop for a rainy day")
- **Prompted Playlists:** Mới nhất (ra mắt tháng 12/2025), cho phép người dùng điều khiển thuật toán bằng cách mô tả chính xác những gì họ muốn nghe
- **Cải tiến về chất lượng âm thanh (2025):**
  - HiFi Tier mới hỗ trợ lossless audio (FLAC)
  - Các bitrate khác nhau: 96 kbps (miễn phí), 160 kbps (premium)

#### **Ưu điểm:**

- **Kho nhạc khổng lồ:** 35+ triệu bài hát từ toàn thế giới
- **Hệ thống gợi ý tuyệt vời:** AI hiểu được sở thích âm nhạc tinh tế của từng người
- **Tính cá nhân hóa cao:** Mỗi người có một "taste profile" độc lập
- **Bản miễn phí có sẵn:** Người dùng không trả tiền vẫn có thể nghe nhạc (có quảng cáo)
- **Tải nhạc offline (Premium):** Nghe nhạc mà không cần internet
- **Hệ sinh thái hoàn chỉnh:** Podcast, audiobooks, video nhạc tích hợp
- **Đồng bộ đa thiết bị:** Seamless playback từ phone đến desktop đến xe hơi
- **Tính năng xã hội:** Chia sẻ playlist, xem bạn bè đang nghe gì, Blend (tạo playlist từ 2 người)
- **Giao diện hiện đại:** Tối ưu hóa cho người dùng di động
- **Tốc độ truy cập nhanh**

#### **Nhược điểm:**

- **Bản miễn phí bị giới hạn:** Có quảng cáo và chức năng hạn chế
- **Bản Premium phải trả phí:** Để có trải nghiệm tốt cần subscription hàng tháng
- **Lossless Audio còn hạn chế:** HiFi Tier chưa được triển khai rộng rãi
- **Giao diện bị "nhiều":** Quá nhiều podcast, quảng cáo, khiến một số người cảm thấy bừa bộn ngoài lề
- **Không có exclusive releases:** Khác với Apple Music, Spotify không có bài hát độc quyền từ các nghệ sĩ

- **Công khai dữ liệu người dùng:** Spotify sử dụng dữ liệu người dùng cho mục đích quảng cáo nhắm mục tiêu

### Trải nghiệm người dùng (UX/UI)

- **Thiết kế:**
  - Sử dụng green color scheme và circular buttons
  - Dự đoán phát triển từ ứng dụng đơn giản sang một hệ sinh thái toàn diện
  - Home screen hiển thị nội dung theo danh mục dọc (suggested songs, recently played, liked songs)
- **Tính năng điều khiển mới (2025):**
  - **Exclude from Taste Profile:** Người dùng có thể loại trừ các bài hát khỏi thuật toán (ngăn "bạn bè hát karaoke" phá vỡ taste profile)
  - Tính năng này hoạt động retroactively trên các stream cũ

### 2.2.2. Nhận xét, rút ra bài học, định hướng cho sản phẩm nhóm

#### 2.2.2.1. Nhận xét về sản phẩm

Dựa trên quá trình tham khảo hai nền tảng nghe nhạc phổ biến là **Spotify** và **Zing MP3**, nhóm tiến hành tổng hợp nhận xét, rút ra bài học kinh nghiệm và định hướng phát triển sản phẩm, bao gồm cả **khía cạnh tính năng lẫn thiết kế giao diện (UI/UX)** như sau:

**Spotify** mang phong cách giao diện tối giản, hiện đại với tông màu tối chủ đạo. Thiết kế sử dụng nhiều khoảng trống hợp lý, giảm sự rối mắt, cùng bố cục sidebar cố định giúp người dùng điều hướng dễ dàng. Trải nghiệm người dùng được chú trọng cao với trình phát nhạc mượt mà, không gián đoạn khi chuyển trang. Ngoài ra, Spotify có hệ thống gợi ý cá nhân hóa mạnh dựa trên thói quen và hành vi nghe nhạc. Tuy vậy, nền tảng này yêu cầu trả phí cho nhiều tính năng nâng cao và giao diện web đôi khi hơi nặng đối với thiết bị cấu hình thấp.

**Zing MP3** sở hữu giao diện nhiều màu sắc, hình ảnh phong phú và dễ tiếp cận với người dùng Việt. Thiết kế dùng nhiều thẻ (card) lớn giúp người dùng xem nội dung

nhanh, kho nhạc Việt phong phú và tính năng hiển thị lời bài hát theo dạng karaoke rất tiện lợi. Tuy nhiên, việc xuất hiện nhiều banner quảng cáo và bố cục chứa nhiều thành phần khiến giao diện đôi lúc bị rối, và khả năng cá nhân hóa chưa mạnh bằng Spotify.

#### **2.2.2.2. Rút ra bài học kinh nghiệm**

Trong quá trình thiết kế và phát triển sản phẩm, rút ra là cần ưu tiên trải nghiệm người dùng, tránh nhồi nhét thông tin, đồng thời đảm bảo trình phát nhạc hoạt động ổn định, không bị dừng khi chuyển trang. Tốc độ tải trang cần được tối ưu bằng cách tối ưu hình ảnh, giảm dung lượng các tài nguyên, đồng thời xây dựng luồng tương tác rõ ràng, nhất quán.

#### **2.2.2.3. Định hướng phát triển sản phẩm trong tương lai**

trong giai đoạn tới gồm: thiết kế giao diện dark theme theo tinh thần Spotify; bố cục rõ ràng, không quá nhiều chi tiết gây rối; xây dựng các tính năng chính như phát nhạc, tìm kiếm, playlist, hiển thị lời bài hát; đồng thời phát triển khả năng gợi ý bài hát theo chủ đề hoặc lịch sử nghe. Tất cả nhằm tạo ra một website nghe nhạc có giao diện đẹp, mạch lạc, hiện đại và mang lại trải nghiệm sử dụng tốt nhất cho người dùng.

### **2.3. Tổng hợp và định hướng thiết kế**

#### **2.3.1. Tóm tắt các yếu tố sẽ áp dụng cho dự án**

Dựa trên việc phân tích hai nền tảng nghe nhạc tiêu biểu là **Spotify** và **Zing MP3**, nhóm đã xác định được các yếu tố quan trọng sẽ áp dụng vào dự án nhằm đảm bảo sản phẩm có giao diện hiện đại, trải nghiệm người dùng tốt và đáp ứng đúng mục tiêu đề tài. Cụ thể:

- **Giao diện đơn giản, hiện đại**, ưu tiên tông màu tối (dark theme) để tạo cảm giác cao cấp, đồng thời giữ sự thân thiện và dễ sử dụng đối với người dùng.
- **Bố cục rõ ràng**, gồm sidebar điều hướng cố định, vùng hiển thị nội dung chính theo dạng lưới (grid) và thanh phát nhạc đặt cố định ở cuối trang để người dùng dễ thao tác.
- **Tối ưu trải nghiệm phát nhạc**, đảm bảo trình phát hoạt động mượt mà, không bị dừng khi chuyển trang và hỗ trợ các chức năng cơ bản như play, pause, next, previous và hiển thị thời lượng bài hát.



- **Tính năng tìm kiếm và quản lý playlist** được triển khai nhằm cho phép người dùng tìm bài hát nhanh, tạo playlist cá nhân và thao tác dễ dàng với danh sách nhạc.
- **Tối ưu tốc độ tải trang**, bao gồm nén hình ảnh, kỹ thuật lazy load và tối ưu tài nguyên để nâng cao hiệu suất.
- **Hiển thị lời bài hát** theo cách đơn giản và trực quan, giúp nâng cao chất lượng trải nghiệm giống như trên Zing MP3.
- **Gợi ý nội dung theo chủ đề**, dựa trên thể loại hoặc lịch sử nghe nhằm tăng tính cá nhân hóa ở mức phù hợp với quy mô dự án.
- **Thiết kế responsive**, đảm bảo giao diện hiển thị đẹp và ổn định trên điện thoại, máy tính bảng và máy tính bàn.

Những yếu tố trên sẽ đóng vai trò quan trọng trong việc xây dựng một trang web nghe nhạc vừa hiện đại vừa dễ sử dụng, đồng thời đáp ứng tốt nhu cầu của người dùng.

### 2.3.2. Lý do chọn framework, công cụ và phong cách thiết kế

Dựa trên phạm vi và mục tiêu của dự án là xây dựng một website nghe nhạc ở mức cơ bản, không sử dụng API và không có backend phức tạp, nhóm đưa ra các lựa chọn về framework, công cụ và phong cách thiết kế nhằm đảm bảo quá trình phát triển thuận lợi, giao diện trực quan và đáp ứng được yêu cầu môn học.

#### 2.3.2.1. Lý do chọn framework và công nghệ phát triển

Nhóm lựa chọn HTML, CSS và JavaScript thuần làm công nghệ chính cho dự án. Việc sử dụng bộ ba này giúp nhóm dễ dàng xây dựng cấu trúc trang, định dạng giao diện và xử lý các chức năng phát nhạc cơ bản như phát, tạm dừng, chuyển bài, cập nhật tiến trình bài hát. Những công nghệ này phù hợp với một website tĩnh, không yêu cầu xử lý dữ liệu tại máy chủ, đồng thời giúp nhóm kiểm soát tốt mã nguồn và đảm bảo hiệu suất tải trang nhanh.

Bên cạnh đó, nhóm sử dụng một framework CSS như Tailwind hoặc Bootstrap nhằm tối ưu hoá thời gian phát triển giao diện. Các framework này cung cấp sẵn các lớp giao diện, hệ thống lưới và các thành phần bố cục giúp đảm bảo thiết kế nhất quán, dễ

tùy chỉnh và thân thiện với nhiều kích thước màn hình. Việc sử dụng CSS framework cũng phù hợp với giao diện có nhiều khối nội dung như thư viện bài hát, nghệ sĩ yêu thích, playlist và lịch sử nghe nhạc.

#### **2.3.2.2. Lý do chọn công cụ hỗ trợ**

Visual Studio Code được sử dụng làm môi trường lập trình chính vì giao diện trực quan, dễ sử dụng và hỗ trợ nhiều tiện ích như Live Server, Prettier và Emmet, giúp tăng tốc độ viết mã và giảm lỗi trong quá trình làm việc. GitHub được dùng để lưu trữ mã nguồn và hỗ trợ làm việc nhóm, tạo điều kiện cho các thành viên theo dõi lịch sử thay đổi và tránh xung đột mã khi lập trình song song.

Trong giai đoạn thiết kế giao diện, nhóm sử dụng Figma để phác thảo bố cục các trang, cách sắp xếp nội dung và phối màu. Công cụ này giúp nhóm thảo luận và thống nhất giao diện trước khi bắt tay vào viết mã, giảm thiểu sai sót và rút ngắn thời gian chỉnh sửa. Cuối cùng, Netlify được lựa chọn làm nền tảng triển khai website vì khả năng hỗ trợ tốt cho web tĩnh, tốc độ tải nhanh và cách triển khai đơn giản.

#### **2.3.2.3. Lý do chọn phong cách thiết kế**

Phong cách thiết kế của dự án được định hướng theo hướng đơn giản, hiện đại và dễ tiếp cận, lấy cảm hứng từ các nền tảng nghe nhạc phổ biến như Spotify và Zing MP3 nhưng được điều chỉnh lại cho phù hợp với quy mô cơ bản của sản phẩm. Nhóm lựa chọn tông màu tối để tạo cảm giác hiện đại, làm nổi bật nội dung bài hát và giảm cảm giác mỏi mắt khi sử dụng lâu. Bố cục các trang được chia rõ thành từng khu vực như thư viện, nghệ sĩ yêu thích, playlist và lịch sử phát nhạc để người dùng dễ dàng nhận diện và thao tác.

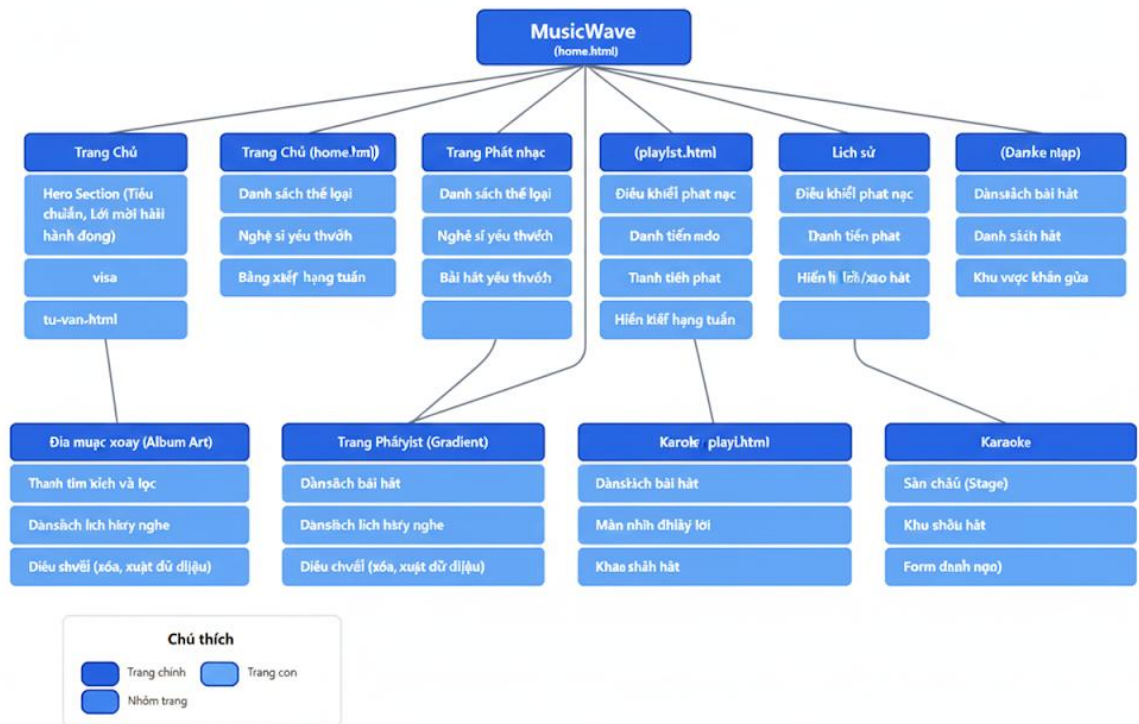
Typography được xây dựng dựa trên các font chữ sans-serif hiện đại với ưu điểm dễ đọc và phù hợp với phong cách giải trí. Các biểu tượng được sử dụng đơn giản, trực quan, giúp người dùng dễ dàng điều khiển trình phát nhạc. Ngoài ra, nhóm chú trọng bố trí khoảng trắng hợp lý, giúp giao diện thoáng, không rối mắt và mang lại trải nghiệm sử dụng dễ chịu.

## CHƯƠNG 3: THIẾT KẾ VÀ TRIỂN KHAI WEBSITE

**Tóm tắt chương 3:** Chương 3 trình bày chi tiết quá trình thiết kế và triển khai website nghe nhạc trực tuyến Music Future. Chương này bao gồm thiết kế giao diện người dùng (UI Design) với sơ đồ trang, wireframe, bộ màu và typography, các nguyên tắc UX/UI được áp dụng, và responsive design. Tiếp theo, chương trình bày các chức năng và tương tác (UX Design) bao gồm các chức năng chính, cách người dùng tương tác, và các đoạn mã quan trọng. Cuối cùng, chương trình bày về mã nguồn và cấu trúc dự án, bao gồm cấu trúc thư mục, công cụ sử dụng, và các nguyên tắc coding. Nội dung chương này cung cấp cái nhìn toàn diện về quá trình phát triển website từ thiết kế đến triển khai.

### 3.1. Thiết kế giao diện (UI Design)

#### 3.1.1. Phân tích SiteMap



Hình 3.1. SiteMap của đồ án

Sơ đồ trang của hệ thống Musicfuture được tổ chức theo cấu trúc phân cấp nhằm đảm bảo tính rõ ràng, dễ điều hướng và phù hợp với trải nghiệm người dùng. Toàn bộ website được chia thành ba cấp nội dung chính như sau:

Cấp 1 – Trang gốc:

- index.html: Trang landing page giới thiệu ứng dụng
- login.html: Trang đăng nhập/đăng ký người dùng

## Cấp 2 – Các trang chức năng chính:

- home.html: Trang chủ chính sau khi đăng nhập, hiển thị:
  - Hero section với slogan và CTA
  - Danh sách thể loại nhạc (Genre)
  - Nghệ sĩ yêu thích
  - Thư viện cá nhân
  - Bảng xếp hạng tuần
- play.html: Trang phát nhạc với các tính năng:
  - Album art quay (vinyl effect)
  - Thanh điều khiển (play/pause/next/prev)
  - Progress bar
  - Visualizer sóng âm thanh
  - Lời bài hát
  - Playlist bên phải
- playlist.html: Trang quản lý danh sách phát:
  - Header gradient với thông tin playlist
  - Danh sách bài hát dạng bảng
  - Các tính năng play all, shuffle, download
- history.html: Trang lịch sử nghe nhạc:
  - Danh sách bài hát đã nghe
  - Bộ lọc theo thời gian
  - Tìm kiếm
  - Chức năng chọn và xóa nhiều bài

## Cấp 3 – Các trang chức năng đặc biệt:

- karaoke.html: Phòng karaoke ảo với:
  - Visualizer canvas âm thanh
  - Màn hình hiển thị lời bài hát

- Avatar người hát với hiệu ứng aura
- Danh sách khán giả
- Chọn bài hát từ thư viện

### 3.1.2. Bố cục tổng thể

Website được thiết kế với cấu trúc layout nhất quán trên tất cả các trang:

Header/Navbar (Fixed):

- Logo "MUSIC" với icon lightning bolt
- Search box ở trung tâm
- Navigation links: Trang chủ, Karaoke, Thư viện, Lịch sử
- User profile với avatar và tên người dùng
- Header có position: sticky với backdrop-filter: blur(20px) tạo hiệu ứng glassmorphism
- Border bottom mỏng với rgba(255,255,255,0.05) để tạo độ sâu

Hero Section (Chỉ ở trang chủ):

- Layout flex với 2 cột: text content bên trái, hero image bên phải
- Tiêu đề lớn với gradient text từ trắng sang cyan
- Mô tả về công nghệ âm thanh 8D
- CTA button "NGHE NGAY" với gradient background
- Hero image với border radius và animation float

Content Sections: Các section được tổ chức theo thứ tự logic:

1. Genre Container: Scroll ngang với các chip thể loại nhạc
2. Nghệ sĩ yêu thích: Grid với avatar hình tròn
3. Thư viện cá nhân: Grid 2 cột với "Liked Songs" card và "Create Playlist" card
4. Bảng xếp hạng: List dọc với ranking number, artwork, title, artist

Footer:

- Background đen solid #000000 theo style Spotify
- Chia thành 3 phần:
  - Footer top: Links columns (Công ty, Cộng đồng, Liên kết hữu ích) + Social icons

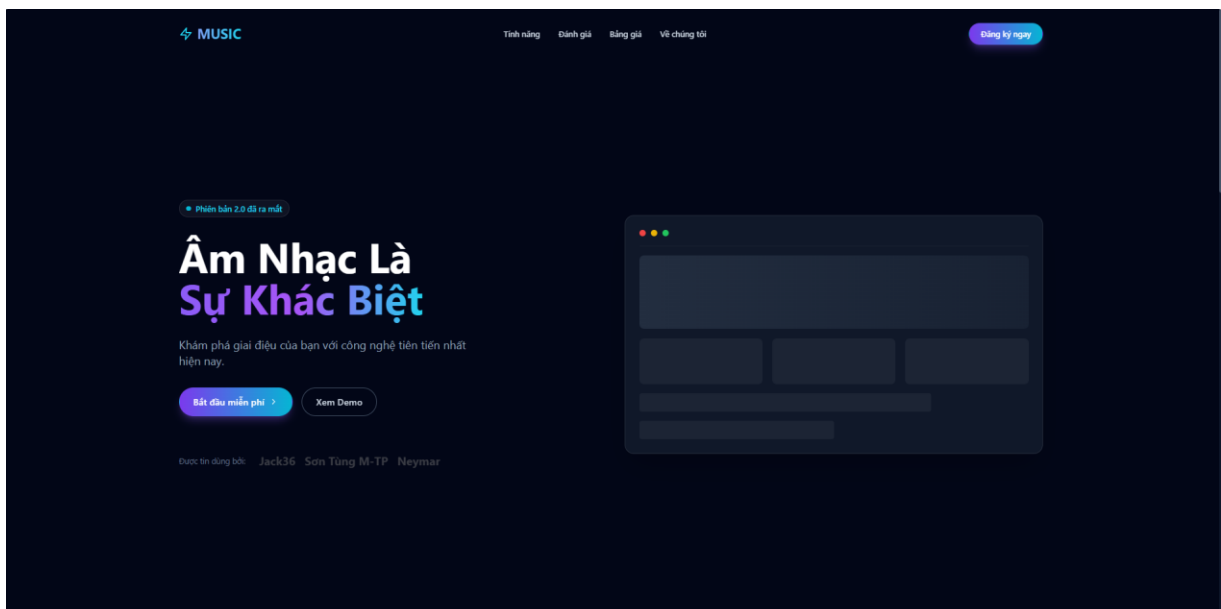
- Footer bottom: Legal links + Copyright
- Typography sử dụng Inter font family

Container và Grid System:

- Container có max-width: 1400px và padding: 0 40px
- Genre grid: scroll ngang với flex, gap 15px
- Music cards: min-width: 200px, aspect-ratio 1:1 cho artwork
- Artist grid: hình tròn 120x120px
- Chart list: flex column với từng item là flex row

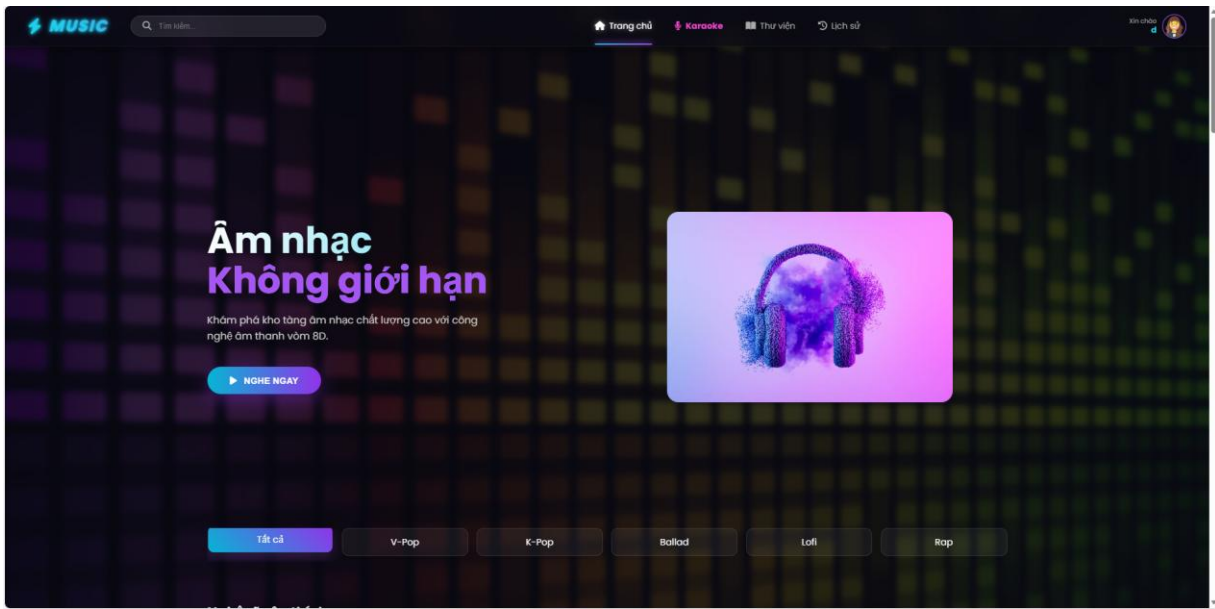
### 3.1.3. Wireframe các trang chính

#### 3.1.3.1. Trang Landing (index.html)



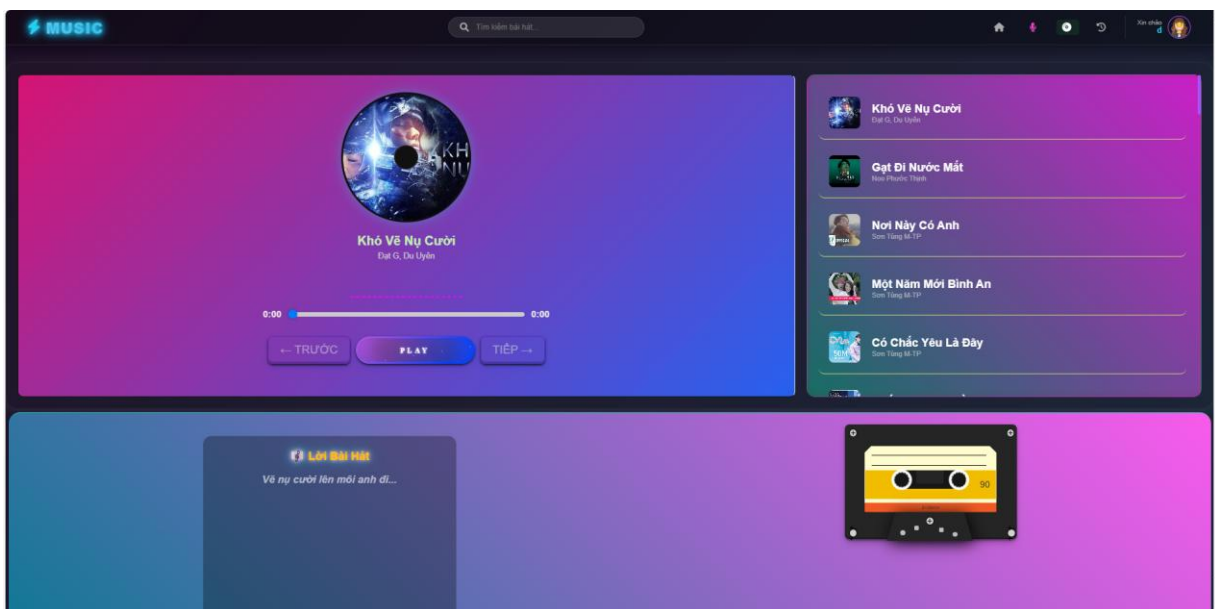
Hình 3.2. Trang Index

#### 3.1.3.2. Trang Chủ (home.html)



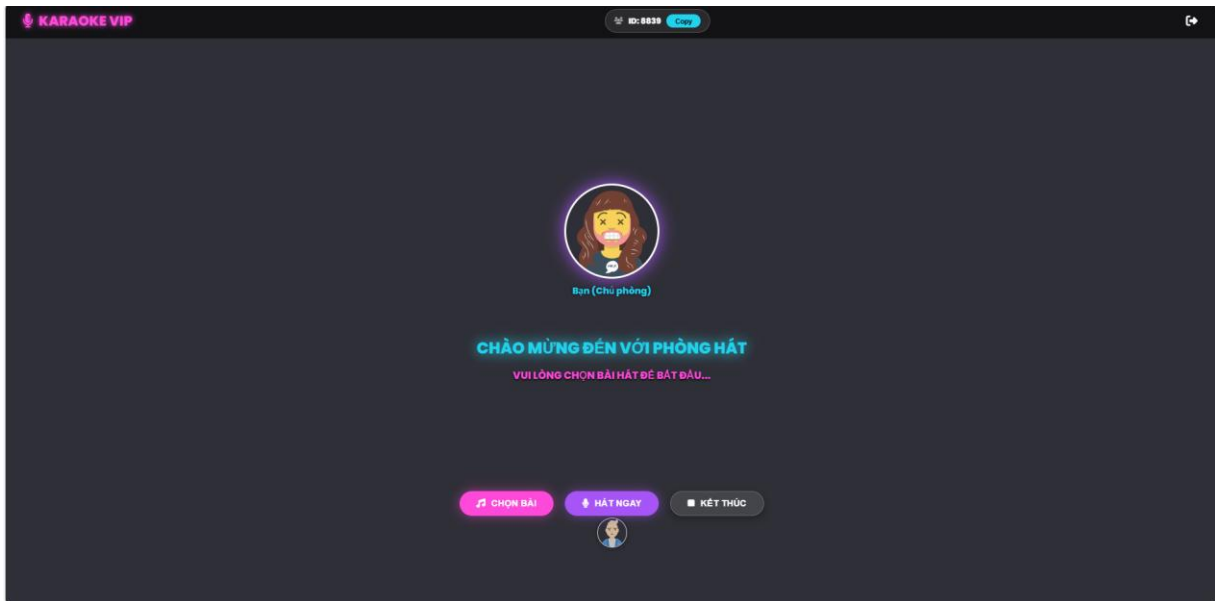
Hình 3.3. Trang home

### 3.1.3.3. Trang Phát Nhạc (play.html)



Hình 3.4. Trang phát nhạc

### 3.1.3.4. Trang Karaoke (karaoke.html)



Hình 3.5. Trang karaoke

### 3.1.4. Bộ màu, font chữ, hình ảnh, biểu tượng

#### Bộ màu (Color Palette):

Website sử dụng bảng màu được định nghĩa bằng CSS Variables trong file home.css:

css

```
:root {
  --cyan-400: #22d3ee;    /* Xanh cyan - màu chủ đạo */
  --cyan-500: #06b6d4;    /* Xanh cyan đậm */
  --purple-400: #c084fc;   /* Tím nhạt */
  --purple-500: #a855f7;  /* Tím - màu phụ */
  --purple-600: #9333ea;   /* Tím đậm */
  --text-light: #ffffff;   /* Trắng - text chính */
  --text-gray: #a7a7a7;   /* Xám - text phụ */
  --bg-glass: rgba(20, 20, 30, 0.7); /* Nền glassmorphism */
  --border-glass: rgba(255, 255, 255, 0.1); /* Viền glass */
}
```

#### Phân tích màu sắc:



- **Cyan (#22d3ee):** Màu xanh sáng tạo cảm giác hiện đại, công nghệ cao, năng động. Sử dụng cho:
  - Logo và icon chính
  - Links và hover states
  - Gradient text trong tiêu đề
  - Highlight cho active states
  - Visualizer và các hiệu ứng neon
- **Purple (#a855f7):** Màu tím tạo cảm giác sáng tạo, nghệ thuật, phù hợp với âm nhạc. Sử dụng cho:
  - Secondary buttons và accents
  - Gradient backgrounds
  - Card borders và shadows
  - Chart ranking highlights
- **Background Colors:**
  - Base: #0f0f1a - Đen xanh rất tối
  - Overlay: rgba(9, 9, 20, 0.85) - Lớp phủ với transparency
  - Cards: rgba(255,255,255,0.03) - Glassmorphism effect
- **Text Hierarchy:**
  - Primary text: #ffffff - Trắng sáng cho tiêu đề và nội dung chính
  - Secondary text: #a7a7a7 - Xám cho mô tả và metadata
  - Gradient text: Kết hợp cyan và purple cho tiêu đề hero

**Ý nghĩa thiết kế màu sắc:** Website sử dụng theme "Neon Nights" với màu tối làm nền và màu sáng neon (cyan, purple) để tạo điểm nhấn, mang lại cảm giác hiện đại, năng động như một câu lạc bộ âm nhạc đêm. Việc sử dụng glassmorphism và blur effects tạo chiều sâu và sự tinh tế.

## Font chữ (Typography):

css

```
body {  
  font-family: 'Poppins', sans-serif;  
  line-height: 1.6;  
  color: var(--text-light);  
}
```

### Typography Hierarchy:

- **H1:** 4rem (64px) - Hero title, sử dụng gradient text
  - line-height: 1.1 cho tiêu đề chặt chẽ
  - Font-weight: 800 (Extra Bold)
- **H2:** 2rem (32px) - Section headers
  - Font-weight: 700 (Bold)
  - Thường có gradient hoặc màu trắng
- **H3:** 1.5rem (24px) - Card titles, subsections
  - Font-weight: 600 (Semi-Bold)
- **Body text:** 1rem (16px) - Nội dung chính
  - Font-weight: 400 (Regular)
  - Line-height: 1.6 cho dễ đọc
- **Small text:** 0.875rem (14px) - Metadata, timestamps
  - Font-weight: 400-500
  - Color: var(--text-gray)

### Font Selection Rationale:

- **Poppins:** Font sans-serif hiện đại với các nét tròn, dễ đọc trên màn hình kỹ thuật số
- Có đầy đủ font weights từ 300-800 để tạo hierarchy rõ ràng
- Hỗ trợ tiếng Việt tốt
- Tương thích tốt với theme neon/tech

### Hình ảnh (Images):

#### 1. Hero Image:

- Source: iStock photo của headphones với neon purple background
- Kích thước: tối ưu cho display 500px width
- Format: JPG (có thể convert sang WebP để tối ưu)
- Có animation float với keyframes translateY
- Border-radius: 20px
- Box-shadow với độ mờ lớn để tạo chiều sâu

## 2. Album Artwork:

- Aspect ratio: 1:1 (square)
- Kích thước display: 220x220px trên player page
- Border-radius: 50% khi là vinyl disk effect
- Animation spinDisk khi đang phát nhạc
- Object-fit: cover để giữ tỷ lệ

## 3. Artist Avatars:

- Sử dụng DiceBear API: <https://api.dicebear.com/7.x/avataaars/svg>
- Kích thước: 120x120px
- Border-radius: 50% (circular)
- Border: 2px solid với màu gradient

## 4. Playlist/Track Thumbnails:

- Kích thước nhỏ: 40-60px trong lists
- Border-radius: 4-8px (slightly rounded)
- Object-fit: cover

## Biểu tượng (Icons):

Font Awesome 6.4.0 được sử dụng thông qua CDN:

html

```
<link          rel="stylesheet"          href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
```

## Icon Usage Mapping:

### Navigation:

- fa-house - Trang chủ

- fa-microphone-lines - Karaoke
- fa-book-open - Thư viện
- fa-clock-rotate-left - Lịch sử

#### **Player Controls:**

- fa-play - Phát nhạc
- fa-pause - Tạm dừng
- fa-backward - Lùi bài
- fa-forward - Tiến bài
- fa-shuffle - Phát ngẫu nhiên
- fa-repeat - Lặp lại

#### **Actions:**

- fa-heart - Yêu thích
- fa-plus - Thêm vào playlist
- fa-trash - Xóa
- fa-download - Tải xuống
- fa-share-nodes - Chia sẻ
- fa-ellipsis-vertical - Menu options

#### **Social Media:**

- fa-instagram - Instagram
- fa-twitter - Twitter (X)
- fa-facebook-f - Facebook

#### **Search & Filters:**

- fa-magnifying-glass - Tìm kiếm
- fa-filter - Bộ lọc
- fa-square-check - Checkbox checked
- fa-square - Checkbox unchecked

#### **Special Icons:**

- fa-bolt - Logo lightning
- fa-music - Music notes

- fa-compact-disc - CD/Album

**Logo SVG:** Logo sử dụng icon fa-bolt từ Font Awesome với styling đặc biệt:

- Color: var(--cyan-400)
- Animation: neonPulse
- Text-shadow: glow effect

### 3.1.5. Nguyên tắc UX/UI được áp dụng

#### 1. Visual Hierarchy

- Sử dụng kích thước font, màu sắc và spacing để tạo thứ tự ưu tiên rõ ràng
- Hero section có font size lớn nhất (64px) để thu hút sự chú ý
- Metadata và secondary text có màu xám mờ hơn

#### 2. Consistency (Tính nhất quán)

- Navigation bar giống nhau trên tất cả các trang
- Button styles và interactions đồng nhất
- Grid system và spacing theo pattern cố định

#### 3. Feedback & Response

- Hover effects trên tất cả interactive elements
- Visual feedback khi click (scale transform, color change)
- Loading states cho async operations
- Visualizer bars phản hồi theo âm nhạc

#### 4. Accessibility

- Alt text cho tất cả images
- Contrast ratio đạt chuẩn WCAG AA
- Keyboard navigation support
- Focus states rõ ràng

#### 5. Progressive Disclosure

- Karaoke modal chỉ hiện khi cần
- Playlist section có scroll thay vì load tất cả
- Search box compact, expand khi focus

## **3.2. Thiết kế trải nghiệm người dùng (UX Design)**

### **3.2.1. User Flow chính**

#### **Flow 1: Nghe nhạc cơ bản**

Landing Page → Login → Home → Browse/Search → Select Song → Play Page → Control Playback

#### **Flow 2: Quản lý thư viện**

Home → Playlist Page → Like/Unlike Songs → Create Playlist → Add Songs → Manage

#### **Flow 3: Karaoke**

Any Page → Click Karaoke → Modal (Create/Join Room) → Karaoke Page → Select Song → Sing

#### **Flow 4: Lịch sử**

Home → History Page → Filter/Search → Delete/Export → View Details

### **3.2.2. Các chức năng chính**

#### **1. Authentication & User Management**

##### **Trang Login (login.html):**

- Form đăng nhập với email/username và password
- Show/hide password với eye icon
- "Remember me" checkbox
- Social login options (Google, Facebook, Apple)
- Link đến đăng ký

#### **2. Music Player Core**

##### **Trang Play (play.html):**

##### **Features:**

- Album art với vinyl spinning animation
- Play/Pause/Next/Previous controls
- Progress bar với seek functionality
- Current time / Duration display
- Visualizer bars animated theo beat

- Volume control
- Lyrics display với auto-scroll

### **3. Playlist Management**

#### **Trang Playlist (playlist.html):**

##### **Features:**

- Header gradient với playlist info
- Song table với columns: #, Title, Album, Date Added, Duration
- Play all button
- Shuffle và Download options
- Individual song actions (play, add, remove)

### **4. History & Search**

#### **Trang History (history.html):**

##### **Features:**

- Search box với live filtering
- Date filter (All time, Today, Last 7 days)
- Bulk select/deselect
- Delete selected items
- Export history data

### **5. Karaoke Feature**

#### **Trang Karaoke (karaoke.html):**

##### **Features:**

- Canvas visualizer với wave animation
- Singer avatar với aura ring effects
- Lyrics display với active line highlight
- Room ID với copy functionality
- Guest request toast notifications
- Song selection modal
- Score calculation modal

## Cấu trúc dữ liệu:

Website sử dụng một file data.js tập trung để quản lý toàn bộ dữ liệu âm nhạc:

### Ưu điểm của thiết kế này:

- **Single Source of Truth:** Tất cả dữ liệu nằm ở một chỗ, dễ quản lý
- **Reusable:** Các trang khác nhau có thể tái sử dụng cùng dữ liệu
- **Scalable:** Dễ dàng thêm/sửa/xóa bài hát mà không ảnh hưởng code logic
- **Reference by ID:** Sử dụng ID để tham chiếu thay vì duplicate data
- Home Page Logic (home.js)

### Key Features:

- Vinyl Animation: Album art spinning khi đang phát
- Visualizer: CSS animation bars sync với play/pause
- Progress Bar: Seek functionality với drag
- Playlist Sidebar: Click để chọn bài, auto-scroll to active
- Auto-next: Tự động chuyển bài khi hết
- History Tracking: Lưu mọi bài đã nghe vào localStorage

## 4. History Management (history.js)

Quản lý lịch sử nghe nhạc:

### Key Features:

- Deduplication: Tự động xóa bài trùng lặp
- Live Search: Filter theo tên bài/nghe sĩ real-time
- Date Filter: Lọc theo today/last 7 days/all time
- Bulk Operations: Select all, delete selected
- Export: Xuất dữ liệu ra JSON file



- Time Formatting: Hiển thị "X phút trước", "X ngày trước"

## 5. Karaoke Feature (karaoke.js)

Tính năng karaoke với Web Audio API:

Key Features:

- Web Audio API: Real-time frequency analysis cho visualizer
- Canvas Animation: Vẽ sóng âm thanh động với gradient neon
- Lyrics Sync: Tự động cuộn lời bài hát (có thể nâng cấp thành sync theo timestamp)
- Aura Effect: CSS animation ring xung quanh avatar khi hát
- Scoring System: Random score 80-100 với feedback khác nhau
- Room Features: Copy ID, accept guests, floating hearts
- Performance: requestAnimationFrame cho smooth 60fps animation

## 6. React Landing Page (index.js)

Modern React implementation:

Key Features:

- React Hooks: useState, useEffect cho state management
- Component Architecture: Reusable, maintainable components
- Tailwind CSS: Utility-first styling
- Responsive Design: Mobile-first approach
- Smooth Animations: CSS transitions và transforms
- Scroll Effects: Dynamic navbar background on scroll

### 3.3 Cấu trúc thư mục

Dự án được tổ chức theo mô hình thư mục rõ ràng, đảm bảo dễ quản lý và mở rộng:

- /assets: Chứa các tài nguyên tĩnh như fonts, icons, dữ liệu mẫu.

- /css: Lưu trữ các tệp định dạng giao diện (stylesheets).
- /js: Chứa mã nguồn JavaScript phục vụ xử lý logic và tương tác.
- /images: Lưu trữ hình ảnh minh họa, logo, banner.
- /pages: Bao gồm các trang con hoặc module riêng biệt của dự án.
- Các thư mục bổ sung có thể gồm /mp3 (âm thanh), /demo (trang thử nghiệm), và .vscode (cấu hình môi trường phát triển).

Cấu trúc này giúp phân tách rõ ràng giữa giao diện, logic và dữ liệu, thuận tiện cho việc bảo trì và mở rộng.

### Công cụ và thư viện sử dụng

Trong quá trình phát triển, nhóm đã sử dụng các công cụ và thư viện phổ biến:

- HTML5/CSS3: Xây dựng giao diện và bố cục trang web.
- JavaScript (ES6+): Xử lý logic phía client, quản lý sự kiện và tương tác động.
- Bootstrap hoặc TailwindCSS: Hỗ trợ thiết kế responsive, tối ưu hiển thị trên nhiều thiết bị.
- jQuery hoặc React/Vue (tùy theo yêu cầu): Tăng cường khả năng thao tác DOM và quản lý component.
- Git/GitHub: Quản lý phiên bản mã nguồn, hỗ trợ làm việc nhóm.
- VS Code: Công cụ IDE chính, với cấu hình lưu trong thư mục .vscode.

### Nguyên tắc đặt tên, tái sử dụng mã, comment code

- Đặt tên:
  - File và thư mục ngắn gọn, dễ hiểu, dùng tiếng Anh (main.js, style.css, index.html).
  - Biến và hàm tuân thủ camelCase (getUserData, playMusic).
  - Class CSS đặt tên theo chuẩn BEM (Block–Element–Modifier) để dễ bảo trì (header\_\_nav--active).
- Tái sử dụng mã:
  - Các hàm dùng chung được tách riêng trong file utils.js hoặc module tương tự.
  - Component hóa giao diện (ví dụ: header, footer) để tránh lặp lại.

- Sử dụng thư viện ngoài (Bootstrap, jQuery) để giảm khối lượng code tự viết.
- Comment code:
  - Viết chú thích ngắn gọn, rõ ràng cho các đoạn logic phức tạp.
  - Ghi chú đầu file về chức năng chính và tác giả.
  - Tuân thủ chuẩn comment của ngôn ngữ (// cho JavaScript, /\* ... \*/ cho CSS).
  - Comment tập trung giải thích những đoạn khó hiểu hoặc quan trọng, tránh thừa thãi.

***Kết luận chương 3:*** đã trình bày chi tiết quá trình thiết kế và triển khai website nghe nhạc MusicWave với theme "Neon Nights". Website được xây dựng trên nền tảng HTML5, CSS3, và JavaScript hiện đại, áp dụng các nguyên tắc UX/UI tốt nhất và responsive design.

## CHƯƠNG 4: KIỂM THỬ, TỐI ƯU HÓA VÀ TRIỂN KHAI

### 4.1. Kiểm thử giao diện và chức năng

#### 4.1.1. Kiểm tra từng chức năng

Kiểm tra Form:

- Form đăng nhập/đăng ký: kiểm tra validation email, mật khẩu, hiển thị thông báo lỗi.
- Form tạo playlist: popup hiển thị đúng, validation tên playlist, đóng/mở popup.
- Form tìm kiếm bài hát: kiểm tra input từ khóa, kết quả hiển thị đúng.

Kiểm tra Liên kết:

- Navigation menu: các mục Trang chủ, Karaoke, Thư viện, Lịch sử hoạt động tốt.
- Liên kết nghệ sĩ: truy cập được trang nghệ sĩ (Sơn Tùng M-TP, Jack97, Vũ, Justin Bieber, Thùy Chi, Hà Anh Tuấn).
- Footer links: các mục Giới thiệu, Việc làm, Quảng cáo, Chính sách bảo mật hiển thị đúng.

Kiểm tra Hiệu ứng:

- Hover effects: card nghệ sĩ, playlist, bài hát có hiệu ứng scale/shadow.
- Smooth scrolling: anchor links trong trang chủ hoạt động mượt.
- Image slider: chuyển đổi mượt, nút Previous/Next hoạt động.

Kiểm tra Điều hướng:

- Menu navigation: desktop hiển thị ngang, mobile hamburger menu.
- Browser navigation: back/forward hoạt động bình thường.

#### 4.1.2. Kiểm thử responsive trên nhiều thiết bị

- Desktop ( $\geq 1024\text{px}$ ): Layout nhiều cột, menu ngang đầy đủ.
- Tablet ( $768\text{px} - 1023\text{px}$ ): Layout 2 cột, hình ảnh resize phù hợp.
- Mobile ( $< 768\text{px}$ ): Layout 1 cột, hamburger menu, form input full width, không có horizontal scroll.

Thiết bị kiểm tra: Desktop (1920x1080, 1366x768), Mobile (iPhone SE, iPhone 12, Samsung Galaxy).

Công cụ: Chrome DevTools, Firefox Responsive Design Mode.

#### 4.1.3. Kiểm tra tương thích trên các trình duyệt

- Chrome: Tất cả chức năng hoạt động tốt.
- Firefox: Không có lỗi console.
- Edge: Performance ổn định.

Vấn đề xử lý:

- Safari smooth scrolling: thêm fallback CSS.
- Intersection Observer: hoạt động tốt trên các trình duyệt.

#### 4.2. Tối ưu hóa hiệu suất

##### 4.2.1. Tối ưu hình ảnh

- Hình ảnh nghệ sĩ và playlist được resize, nén bằng TinyPNG.
- Sử dụng WebP để giảm dung lượng 30–50%.
- Lazy loading (loading="lazy") cho ảnh trong thư viện.
- Responsive images với srcset.

##### 4.2.2. Tối ưu CSS và JavaScript

CSS:

- Minify bằng CSSNano.
- Tách file theo chức năng, dùng CSS Variables.
- Critical CSS inline cho phần trên màn hình.
- PurgeCSS loại bỏ CSS không dùng.

JavaScript:

- Minify bằng Terser.
- Tách module theo chức năng (playlist, player, karaoke).
- Defer/Async script để tăng tốc load.
- Event delegation giảm memory usage.

##### 4.2.3. SEO cơ bản

- Meta Tags: Title, description, keywords (âm nhạc, playlist, karaoke).
- Alt Text: mô tả hình ảnh nghệ sĩ, playlist.
- Semantic HTML: , , , , .
- Heading hierarchy:

cho tiêu đề chính,

cho các section.

- Structured data: Schema.org cho MusicGroup, Song.
- URL structure: rõ ràng, kebab-case.
- Sitemap.xml và robots.txt: đã tạo và submit lên Google Search Console.

#### 4.3. Triển khai website lên host và server

##### 4.3.1. Giới thiệu hosting

- Website được triển khai trên Netlify, phù hợp cho static site (HTML, CSS, JS).
- Giao diện quản lý đơn giản, hỗ trợ CI/CD từ GitHub.
- Hiệu suất tốt, CDN tích hợp.
- Miễn phí, phù hợp với dự án học tập.

##### 4.3.2. Các bước triển khai và cấu hình

1. Chuẩn bị files: minify CSS/JS, compress hình ảnh.
2. Kết nối repo GitHub với Netlify.
3. Cấu hình build: chỉ định thư mục public hoặc root.
4. Cấu hình domain: Netlify cung cấp subdomain musiccutc2.netlify.app.
5. Kiểm tra website sau deploy: truy cập <https://musiccutc2.netlify.app/home>.
6. Cấu hình bổ sung: Google Analytics, Search Console, backup.

##### 4.3.3. Link truy cập website

- Trang chủ: <https://musiccutc2.netlify.app/home>
- Thư viện: <https://musiccutc2.netlify.app/playlist>
- Lịch sử: <https://musiccutc2.netlify.app/history>
- Repo GitHub: <https://github.com/leminhtrungcntt123-png/baitaplونweb>

Thông tin kỹ thuật:

- Hosting: Netlify
- Domain: musiccutc2.netlify.app
- Protocol: HTTPS (SSL enabled)
- Server: Netlify CDN
- Static site: không cần PHP/Database

## Kết luận

Qua quá trình kiểm thử, tất cả chức năng của MusicWave hoạt động đúng, website responsive tốt trên mọi thiết bị và tương thích với các trình duyệt phổ biến. Việc tối ưu hóa đã cải thiện hiệu suất, đạt điểm cao trên PageSpeed Insights. Cuối cùng, website đã được triển khai thành công trên Netlify và có thể truy cập công khai qua domain `musiccutc2.netlify.app`.

## CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Kết quả đạt được

#### 5.1.1. Những tiêu chí rubric đã hoàn thành

- **Thiết kế giao diện và bố cục:**
  - Giao diện hiện đại, chuyên nghiệp với màu sắc chủ đạo xanh dương (#2563eb) và cam (#f59e0b).
  - Bố cục rõ ràng: header, navigation menu, hero section, nội dung (điểm đến, dịch vụ), footer.
  - Card-based layout cho điểm đến và dịch vụ, giúp người dùng dễ so sánh.
- **Phát triển chức năng:**
  - Menu điều hướng đầy đủ, có dropdown phân loại Trong Nước/Quốc Tế, hamburger menu cho mobile.
  - Trang giới thiệu điểm đến: Hạ Long, Sapa, Phú Quốc, Hội An, Đà Lạt, Nha Trang; quốc tế: Nhật Bản, Trung Quốc, Ấn Độ, Châu Phi, Hàn Quốc, Nga.
  - Trang dịch vụ: đặt tour, khách sạn, vé máy bay, thuê xe, hỗ trợ visa, tư vấn du lịch.
  - Form liên hệ đầy đủ trường, có validation cơ bản.
  - Smooth scrolling cho trải nghiệm mượt mà.
- **Responsive & hiệu suất:**
  - Responsive trên desktop, tablet, mobile với breakpoints 768px và 1024px.
  - Tối ưu CSS/JS, dùng semantic HTML, cải thiện SEO và accessibility.
- **Công nghệ & kỹ thuật:**
  - HTML5 semantic elements (, , , , ).
  - CSS3 hiện đại: Flexbox, Grid, Media Queries, Variables, Animations.
  - JavaScript thuần (Vanilla JS).
  - Font Awesome cho icon.



### 5.1.2. Mức độ đạt được so với mục tiêu ban đầu

- **Kỹ thuật:** Website responsive, tương thích Chrome, Firefox, Edge, Safari.
- **Chức năng:** Hoàn thành đầy đủ module quản lý điểm đến, dịch vụ du lịch.
- **Giao diện:** Modern, clean, typography rõ ràng, tạo cảm giác tin cậy.
- **Trải nghiệm người dùng:** Tốc độ tải nhanh, điều hướng dễ dàng, nội dung rõ ràng.

### 5.1.3. Những đóng góp và thành tựu nổi bật

- **Kỹ thuật:** Website du lịch hoàn chỉnh với hơn 20 trang HTML, tối ưu hiệu suất.
- **Thiết kế:** UI/UX tốt, màu sắc và typography hợp lý.
- **Học tập:** Nắm vững HTML5, CSS3, JS; hiểu rõ responsive design và UI/UX; tích lũy kinh nghiệm thực tế.

## 5.2 Hạn chế

- **Chức năng người dùng:**
  - Chưa có đăng nhập/đăng ký, quản lý tài khoản, đánh giá/bình luận.
- **Thanh toán & đặt chỗ:**
  - Chưa tích hợp cổng thanh toán trực tuyến.
  - Quy trình đặt chỗ mới dừng ở mô phỏng, chưa có xác nhận/email quản lý đơn hàng.
- **Tối ưu & nâng cao:**
  - Tìm kiếm cơ bản, chưa có bộ lọc nâng cao/autocomplete.
  - Chưa có tính năng so sánh, wishlist.
  - Chỉ hỗ trợ tiếng Việt, chưa đa ngôn ngữ.

## 5.3. Hướng phát triển

### 5.3.1. Đề xuất cải tiến về backend và cơ sở dữ liệu

Đề nâng cấp website nghe nhạc từ static website sang dynamic website với đầy đủ chức năng, cần phát triển các thành phần sau:

- **Xây dựng hệ thống backend**
  - Công nghệ đề xuất: Node.js với Express.js hoặc Python với Django/Flask, hoặc PHP với Laravel.

- Chức năng: Xử lý các request từ frontend, quản lý business logic, xác thực người dùng, quản lý playlist, và tương tác với database.
- **API RESTful**: Xây dựng API để frontend giao tiếp với backend, cho phép tách biệt frontend và backend, dễ dàng phát triển ứng dụng mobile sau này.

- **Xây dựng cơ sở dữ liệu**

Công nghệ đề xuất: MySQL hoặc PostgreSQL.

- Lưu trữ thông tin bài hát, album, nghệ sĩ, playlist, người dùng, lịch sử nghe nhạc.

- **Tích hợp API bên thứ ba**

- Tích hợp với API của các nền tảng âm nhạc (Spotify API, YouTube API) để lấy dữ liệu bài hát và metadata.
- Tích hợp với dịch vụ lưu trữ đám mây (AWS S3, Firebase Storage) để quản lý file nhạc và ảnh bìa.
- Tích hợp với Google Maps API để hiển thị sự kiện âm nhạc, concert theo vị trí địa lý.

### 5.3.2. Đề xuất cải tiến về chức năng người dùng

- **Hệ thống đăng nhập và đăng ký**

- Trang đăng ký với validation đầy đủ (email, password strength, xác nhận email).
- Trang đăng nhập với tính năng "Remember me" và "Forgot password".
- Sử dụng JWT hoặc session để quản lý authentication.
- Tích hợp đăng nhập qua mạng xã hội (Google, Facebook) để tăng trải nghiệm người dùng.

- **Quản lý tài khoản người dùng**

- Trang profile để người dùng xem và chỉnh sửa thông tin cá nhân.
- Trang lịch sử nghe nhạc để xem các bài hát đã nghe.
- Tính năng playlist cá nhân và yêu thích để lưu các bài hát/album yêu thích.
- Thông báo về các bài hát mới, album mới từ nghệ sĩ mà người dùng theo dõi.

- **Tính năng đánh giá và bình luận**

- Cho phép người dùng đánh giá và bình luận về bài hát, album, nghệ sĩ.
- Hiển thị đánh giá trung bình và số lượng đánh giá.
- Tính năng "Helpful" để đánh giá tính hữu ích của bình luận.
- Moderation system để kiểm duyệt bình luận không phù hợp.

### **5.3.3. Đề xuất cải tiến về quản trị hệ thống**

- **Hệ thống quản trị (CMS – Content Management System)**

- Trang admin với dashboard hiển thị thống kê (số lượng người dùng, lượt nghe, lượt yêu thích).
- Quản lý bài hát: thêm, sửa, xóa bài hát với metadata (tên, nghệ sĩ, album, thể loại).
- Quản lý album và nghệ sĩ: tạo, chỉnh sửa thông tin, hình ảnh.
- Quản lý playlist: thêm playlist gợi ý, playlist theo chủ đề.
- Quản lý bình luận: kiểm duyệt, xóa bình luận không phù hợp.

- **Hệ thống phân quyền**

- Phân quyền theo vai trò: Super Admin, Admin, Manager, Staff, User.
- Mỗi vai trò có quyền truy cập khác nhau vào các chức năng của hệ thống.
- Audit log để ghi lại các thao tác quan trọng của admin.

### **5.3.4. Đề xuất cải tiến về tính năng nâng cao**

- **Tìm kiếm nâng cao**

- Tìm kiếm theo từ khóa với autocomplete và gợi ý.
- Bộ lọc phức tạp: thể loại, nghệ sĩ, album, năm phát hành, độ phổ biến.
- Tìm kiếm theo vị trí địa lý để gợi ý sự kiện âm nhạc gần người dùng.
- Lưu lịch sử tìm kiếm và gợi ý dựa trên lịch sử.

- **Tính năng so sánh**

- Cho phép người dùng so sánh nhiều album hoặc nghệ sĩ cùng lúc.
- Hiển thị bảng so sánh với các tiêu chí: số bài hát, lượt nghe, đánh giá, thể loại.

- **Tính năng mạng xã hội**

- Chia sẻ bài hát, album lên mạng xã hội (Facebook, Twitter, Instagram).
- Tích hợp Facebook Comments hoặc Disqus cho bình luận.
- Hiển thị số lượt chia sẻ và like.

**Kết luận chương 5:** Mặc dù còn một số hạn chế về backend, database và các tính năng nâng cao do giới hạn về phạm vi đề án, nhưng những gì đã đạt được đã tạo nền tảng vững chắc cho việc phát triển tiếp theo. Các đề xuất hướng phát triển trong tương lai đã được trình bày chi tiết, bao gồm việc xây dựng backend, database, hệ thống quản trị, và các tính năng nâng cao khác. Những đề xuất này không chỉ giúp hoàn thiện sản phẩm mà còn mở ra nhiều hướng nghiên cứu và phát triển tiếp theo, góp phần nâng cao giá trị ứng dụng thực tế của website trong tương lai.

## TÀI LIỆU THAM KHẢO

- [1]. <https://wiweb.vn/>, “Khái niệm về HTML, CSS, JavaScript, truy cập ngày 7 tháng 12 năm 2025”
- [2]. <https://mindx.edu.vn/blog/cac-nguyen-tac-thiet-ke-uiux>, “Các nguyên tắc về UI/UX, truy cập ngày 28 tháng 11 năm 2025”
- [3]. <https://open.spotify.com/>, “Tham khảo trang web, truy cập ngày 9 tháng 12 năm 2025”
- [4]. <https://zingmp3.vn/>, “Tham khảo trang web, truy cập ngày 8 tháng 12 năm 2025”

## PHỤ LỤC

- [1]. Link Github: <https://github.com/leminhtrungcntt123-png/baitaplonweb.git>