

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
VIỆN KHOA HỌC KỸ THUẬT BƯU ĐIỆN**



BÀI TẬP LỚN
MÔN: LẬP TRÌNH JAVASCRIPT
ĐỀ TÀI : NGHIÊN CỨU, PHÁT TRIỂN WEBSITE MUSIC

Giảng viên hướng dẫn	: Vũ Văn Thương
Sinh viên thực hiện	: Nguyễn Trọng Trường
Mã sinh viên	: B23DCCC166
Lớp	: D23CQCC02-B

Hà Nội, 2024

LỜI GIỚI THIỆU

Nhờ vào sự phát triển của công nghệ thông tin, người dùng có thể dễ dàng truy cập vào các website âm nhạc từ bất kỳ đâu và bất kỳ lúc nào chỉ với một thiết bị kết nối Internet. Điều này giúp người dùng có thể tận hưởng âm nhạc mọi lúc, mọi nơi. Các website âm nhạc giúp kết nối người nghe và nghệ sĩ từ khắp nơi trên thế giới. Điều này không chỉ mở rộng phạm vi tiếp cận của nghệ sĩ mà còn giúp người nghe khám phá và thưởng thức nhiều nền văn hóa âm nhạc khác nhau. Trong đó, *Website Music* giúp người dùng biết đến các nguồn âm nhạc mới và đa dạng. Điều này giúp họ mở rộng sở thích âm nhạc, khám phá các nghệ sĩ và thể loại mới mà họ có thể chưa từng nghe đến.

Trước hết thì ta phải nói đến vai trò, ý nghĩa của âm nhạc đối với cuộc sống, xã hội hiện nay. Với cuộc sống, âm nhạc có ý nghĩa vô cùng quan trọng và tác động tích cực đến cuộc sống của chúng ta. Nó không chỉ mang lại niềm vui và sự thư giãn, mà còn giúp chúng ta phát triển tư duy, kỹ năng xã hội và khả năng sáng tạo. Âm nhạc cũng có vai trò quan trọng trong giáo dục và truyền tải những thông điệp xã hội quan trọng. Còn với xã hội, âm nhạc không chỉ là một hình thức nghệ thuật mà còn là một phương tiện giao tiếp xã hội mạnh mẽ. Nó có thể gắn kết cộng đồng và truyền tải những thông điệp xã hội quan trọng. Thông qua âm nhạc, chúng ta có thể thể hiện tình yêu, lòng đoàn kết và hy vọng. Ngoài ra, âm nhạc cũng phản ánh văn hóa địa phương và giúp duy trì và phát triển những giá trị văn hóa của một dân tộc. *Website Music* giúp chúng ta tiếp xúc với âm nhạc dễ dàng thông qua Internet. Website được thiết kế với giao diện trực quan, dễ sử dụng, giúp người dùng dễ dàng tìm kiếm và khám phá các bài hát, album và nghệ sĩ yêu thích chỉ với vài cú nhấp chuột, bạn có thể tìm thấy và nghe những bài hát yêu thích, và có cơ hội khám phá các nghệ sĩ và bài hát mới mà bạn chưa từng nghe đến.

Bằng việc lựa chọn và thực hiện đề tài “Nghiên cứu, xây dựng Website Music”, em muốn tìm hiểu về cách các nền tảng này hoạt động cũng như các yếu tố ảnh hưởng đến trải nghiệm người dùng. Tuy đề tài đã hoàn thành nhưng vì điều kiện thời gian không cho phép cùng như ý tưởng chưa được phong phú nên vẫn còn một số hạn chế nhất định. Rất mong nhận được sự đóng góp ý kiến của thầy và các bạn để sản phẩm ngày càng hoàn thiện hơn.

Trân trọng và chân thành cảm ơn!

I, Tổng quan về dự án

1. Giới thiệu

- Tên dự án: Website Music
- Mô tả ngắn gọn:

Website Music là một trang website được thiết kế trực quan, dễ dàng thao tác sử dụng, giao diện thân thiện dễ nhìn. Website Music giúp người dùng nghe những bài nhạc bài nhạc, album yêu thích của mình. Giúp người dùng khám phá được các bài nhạc, ca sĩ mà họ chưa biết tới để mở rộng thêm sở thích của mình.

2. Cấu trúc mã nguồn

Ngôn ngữ lập trình: HTML, CSS, JavaScript

-HTML : HTML là ngôn ngữ đánh dấu tiêu chuẩn để tạo và cấu trúc nội dung trên trang web. Nó sử dụng các thẻ để xác định các phần của nội dung như đoạn văn, tiêu đề, hình ảnh và liên kết.

-CSS: CSS là ngôn ngữ dùng để mô tả cách hiển thị các thành phần HTML. Nó kiểm soát bố cục, màu sắc, phông chữ và nhiều yếu tố thiết kế khác, giúp tạo ra giao diện người dùng hấp dẫn và nhất quán.

-JS: JavaScript là ngôn ngữ lập trình giúp trang web trở nên động và tương tác. Nó cho phép thay đổi nội dung HTML, phản hồi sự kiện của người dùng, xác thực dữ liệu và tạo các hiệu ứng động.

3.Các chức năng của web

3.1 Chức Năng Playlist:

- Danh sách phát: Hiển thị danh sách các bài hát trong playlist hiện tại.
- Last Listening: Hiển thị các bài hát đã nghe gần đây.
- Recommended: Đề xuất các bài hát dựa trên sở thích nghe nhạc của người dùng.

3.2 Thông Tin Nghệ Sĩ

- Nghệ sĩ: Hiển thị thông tin nghệ sĩ đang được chọn, với tên nghệ sĩ và hình ảnh của họ.
- Play/Follow: Nút để phát nhạc của nghệ sĩ hoặc theo dõi nghệ sĩ đó.

3.3 Bài Hát Phổ Biến

- Popular Song: Hiển thị các bài hát phổ biến của nghệ sĩ hoặc trong hệ thống, với các hình ảnh album và tên bài hát.

3.4 Nghệ Sĩ Phổ Biến

- Popular Artists: Hiển thị hình ảnh và tên của các nghệ sĩ phổ biến, cho phép người dùng khám phá thêm âm nhạc từ các nghệ sĩ này.

3.5 Trình Điều Khiển Nhạc

- Phát nhạc: Hiển thị bài hát đang phát, tên bài hát và nghệ sĩ.
- Điều khiển phát lại: Bao gồm các nút chơi, tạm dừng, chuyển bài hát trước/sau, và thanh tiến trình bài hát.
- Điều chỉnh âm lượng: Thanh điều chỉnh âm lượng cho phép người dùng thay đổi âm lượng phát nhạc.

3.6 Tìm Kiếm

- Search Music: Thanh tìm kiếm cho phép người dùng tìm kiếm bài hát, nghệ sĩ, hoặc album trong hệ thống.

3.7 Tab Điều Hướng

- Discover: Tab để khám phá âm nhạc mới.
- My Library: Tab để truy cập vào thư viện âm nhạc cá nhân của người dùng.
- Radio: Tab để nghe các kênh radio dựa trên thể loại, nghệ sĩ, hoặc tâm trạng.

3.8 Ảnh Đại Diện Người Dùng

-Hiển thị ảnh đại diện của người dùng ở góc trên bên phải, cho phép truy cập vào cài đặt tài khoản hoặc thông tin cá nhân.

5. Thiết kế

-Giao diện đơn giản, dễ sử dụng.

-Tập trung vào việc hiển thị danh sách bài hát trong các playlist.

-Sử dụng tông màu chủ đạo là đen và trắng, tạo cảm giác sang trọng và hiện đại.

II, Tạo Website Music

1, FontEnd

1.1 Sử dụng js để làm giao diện chuyển tiếp ngang:

```
let pop_song_left = document.getElementById('pop_song_left');
let pop_song_right = document.getElementById('pop_song_right');
let pop_song = document.getElementsByClassName('pop_song')[0];

pop_song_right.addEventListener('click', ()=>{
    pop_song.scrollLeft +=330;
});
pop_song_left.addEventListener('click', ()=>{
    pop_song.scrollLeft -=330;
});

let pop_art_left = document.getElementById('pop_art_left');
let pop_art_right = document.getElementById('pop_art_right');
let Artists_bx = document.getElementsByClassName('Artists_bx')[0];

pop_art_right.addEventListener('click', ()=>{
    Artists_bx.scrollLeft +=330;
});
pop_art_left.addEventListener('click', ()=>{
    Artists_bx.scrollLeft -=330;
});
```

1.2 Sử dụng js để chuyển bài hát trước/sau

-Back(trước):

```
let back = document.getElementById('back');
let next = document.getElementById('next');

back.addEventListener('click', () => {
    index -= 1;
    if (index < 1) {
        index = Array.from(document.getElementsByClassName('songItem')).length;
    }
    music.src = `audio/${index}.mp3`;
    poster_master_play.src = `ảnh/${index}.jpg`;
    music.play();
    masterPlay.classList.remove('bi-play-fill');
    masterPlay.classList.add('bi-pause-fill');

    let songTitles = songs.filter((els) => {
        return els.id == index;
    });

    songTitles.forEach(elss => {
        let {songName} = elss;
        title.innerHTML = songName;
    });

    makeAllBackground();
    Array.from(document.getElementsByClassName('songItem'))[index-1].style.background = "rgb(105, 105, 105,.1)";
    makeAllplays();
    el.target.classList.remove('bi-play-circle-fill');
    el.target.classList.add('bi-pause-circle-fill');
    wave.classList.add('active1');
});
```

-Next(sau):

```
next.addEventListener('click', ()=>{
  index ++;
  if (index > Array.from(document.getElementsByClassName('songItem')).length) {
    index = 1;
  }
  music.src = `audio/${index}.mp3`;
  poster_master_play.src = `anh/${index}.jpg`;
  music.play();
  masterPlay.classList.remove('bi-play-fill');
  masterPlay.classList.add('bi-pause-fill');

  let songTitles = songs.filter((els)=> {
    return els.id == index;
  });

  songTitles.forEach(elss =>{
    let {songName} = elss;
    title.innerHTML = songName;
  });

  makeAllBackground();
  Array.from(document.getElementsByClassName('songItem'))[index-1].style.background = "rgb(105, 105, 105,.1)";
  makeAllplays();
  el.target.classList.remove('bi-play-circle-fill');
  el.target.classList.add('bi-pause-circle-fill');
  wave.classList.add('active1');
});
```

1.3 Sử dụng js để nghe/dừng bài hát (play/pause):

```
let masterPlay =document.getElementById('masterPlay');
let wave =document.getElementById('wave');

masterPlay.addEventListener('click', ()=>{
  if (music.paused || music.currentTime <= 0) {
    music.play();
    wave.classList.add('active1');
    masterPlay.classList.remove('bi-play-fill');
    masterPlay.classList.add('bi-pause-fill');
  } else {
    music.pause();
    wave.classList.remove('active1');
    masterPlay.classList.add('bi-play-fill');
    masterPlay.classList.remove('bi-pause-fill');
  }
});
```

1.4 Sử dụng js để tạo thời gian của bài hát, tua đến đoạn muốn nghe :

```
let currentStart = document.getElementById('currentStart');
let currentEnd = document.getElementById('currentEnd');
let seek = document.getElementById('seek');
let bar2 = document.getElementById('bar2');
let dot = document.getElementsByClassName('dot')[0];

music.addEventListener('timeupdate', ()=> {
  let music_curr = music.currentTime;
  let music_dur = music.duration;

  let min1 = Math.floor(music_dur / 60);
  let sec1 = Math.floor(music_dur % 60);

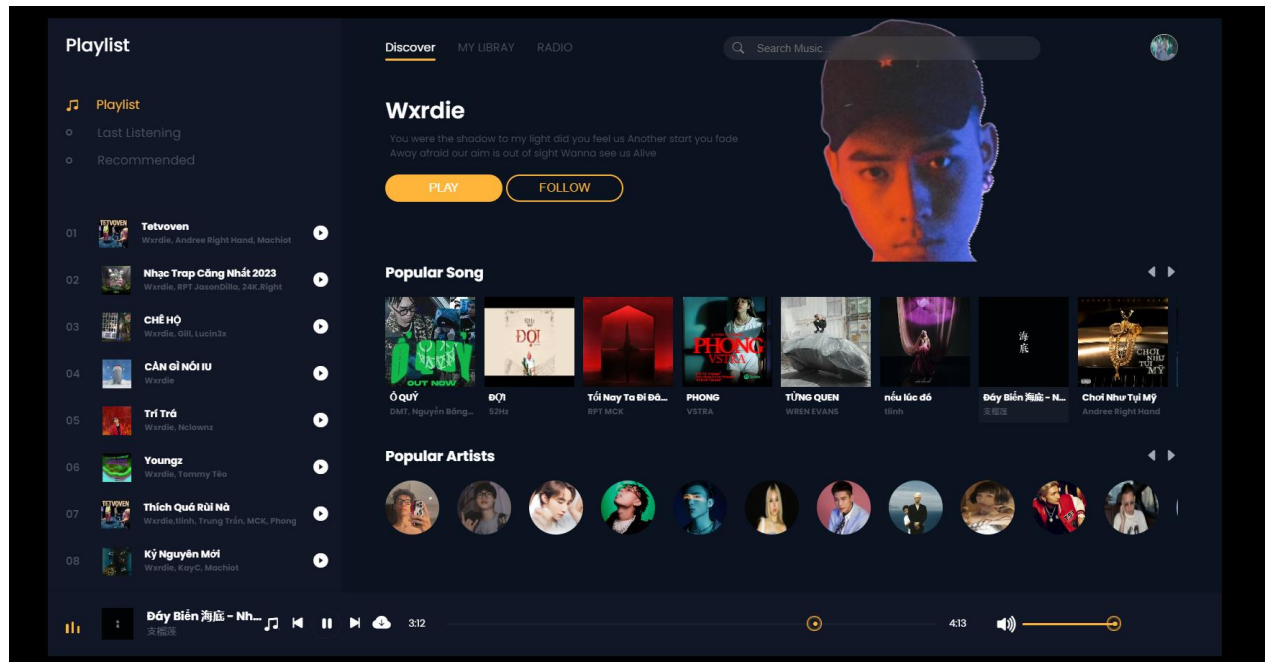
  // console.log(min1);
  if (sec1 < 10) {
    sec1 = `0${sec1}`;
  }
  currentEnd.innerText = `${min1}:${sec1}`;

  let min2 = Math.floor(music_curr / 60);
  let sec2 = Math.floor(music_curr % 60);
  if (sec2 < 10) {
    sec2 = `0${sec2}`;
  }
  currentStart.innerText = `${min2}:${sec2}`;

  let progressBar = parseInt((music_curr / music_dur) * 100);
  seek.value = progressBar;
  // console.log(seek.value);
  let seekbar = seek.value;
  bar2.style.width = `${seekbar}%`;
  dot.style.left = `${seekbar}%`;
});

seek.addEventListener('change', ()=>{
  music.currentTime = seek.value * music.duration / 100;
```


1.5 Giao diện chính



III, Những khó khăn, hạn chế gặp phải và phát triển sản phẩm trong tương lai

1. Khó khăn, hạn chế

1.1 Sau khi hoàn thành xong Website, em nhận thấy vẫn còn nhiều hạn chế sau:

- Vẫn còn ít chức năng, hiện tại website chỉ mới thỏa mãn được những yêu cầu cần thiết nhất.
- Cách khai báo, code v.v... còn thiếu gọn gàng, khoa học.

1.2 Khó khăn trong phần BackEnd :

-Do điều kiện thời gian quy định không có hạn và cũng như ý tưởng chưa được phong phú nên không có cơ hội làm và hoàn thiện phần BackEnd.

2. Phát triển trong tương lai

2.1 Giao diện người dùng (UI) và Trải nghiệm người dùng (UX):

-Thiết kế phản hồi (Responsive Design): Đảm bảo trang web của bạn có thể hiển thị tốt trên mọi thiết bị từ máy tính bàn đến điện thoại di động.

-Dễ dàng điều hướng: Cấu trúc trang web rõ ràng, dễ dàng tìm kiếm và điều hướng.

-Tốc độ tải trang: Tối ưu hóa hình ảnh và mã nguồn để tăng tốc độ tải trang

2.2 Nội dung chất lượng:

- Cập nhật thường xuyên: Nội dung phải được cập nhật thường xuyên với thông tin mới và hữu

ích.

-SEO (Tối ưu hóa công cụ tìm kiếm): Tối ưu hóa từ khóa, mô tả meta, và tiêu đề để cải thiện vị trí trên các công cụ tìm kiếm.

2.3 Tương tác và gắn kết người dùng:

-Mạng xã hội: Tích hợp các nút chia sẻ mạng xã hội và tạo nội dung tương tác để khuyến khích người dùng chia sẻ.

-Email Marketing: Xây dựng danh sách email và gửi các bản tin để giữ liên lạc với người dùng

2.4 Tối ưu hóa hiệu suất:

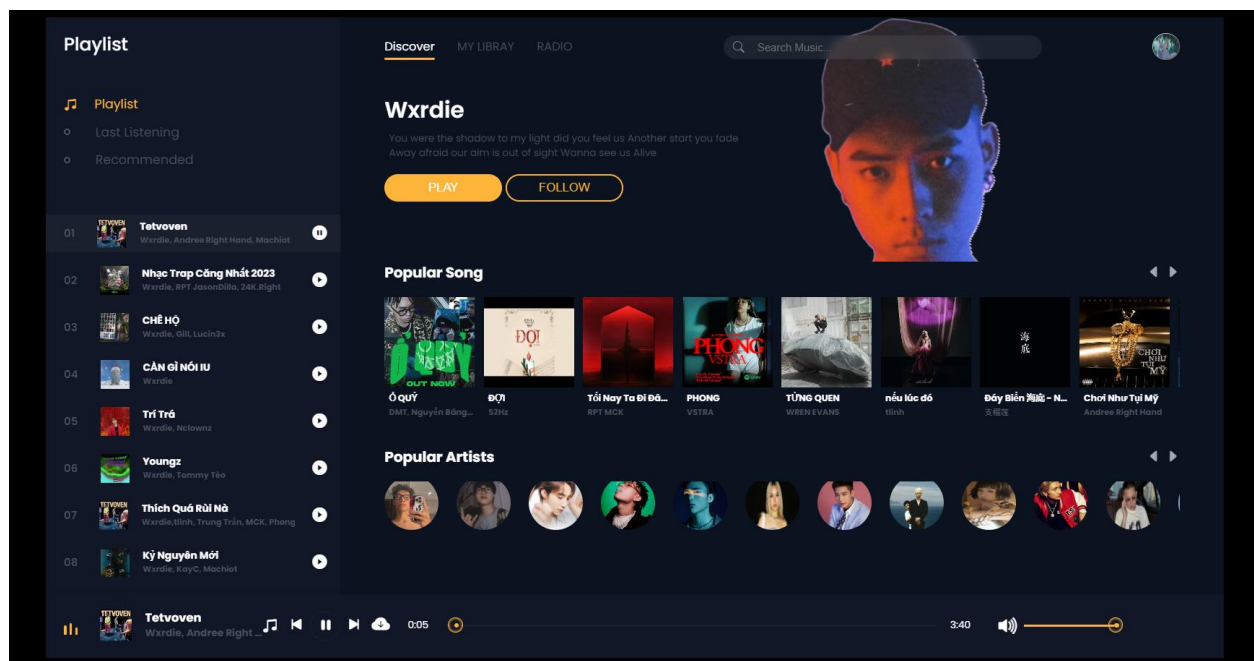
-Mạng phân phối nội dung (CDN): Sử dụng CDN để giảm thời gian tải trang cho người dùng ở các vị trí địa lý khác nhau.

-Caching: Sử dụng các kỹ thuật caching để giảm tải cho máy chủ và tăng tốc độ tải trang

2.5 Hợp tác và đối tác:

-Liên kết với các đối tác: Hợp tác với các trang web hoặc dịch vụ khác để mở rộng phạm vi tiếp cận và cung cấp giá trị gia tăng cho người dùng.

Demo



Tài khảo tham khảo

-YouTube

-Chat GPT

-W3schoole