

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN

PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Website nghe nhạc

Giáo viên hướng dẫn: Th.s Từ Lăng Phiêu
Dịp Lâm Tuấn - 3119410476
Nguyễn Ngọc Trường Chinh - 3119410046
Ngụy Đoan Toàn - 3120410539

TP. HỒ CHÍ MINH, THÁNG 5/2024

Mục lục

1	Phân chia công việc từng thành viên	3
2	Lời mở đầu	4
3	Tóm tắt	5
4	Website nghe nhạc	6
4.1	Thư viện sử dụng	6
4.2	Ngôn ngữ sử dụng	6
5	Các trang chính	7
5.1	Đăng nhập	7
5.2	Code xử lý Đăng nhập	8
5.3	Đăng Ký	9
5.4	Code xử lý Đăng ký	10
5.5	Xem được Top nghệ sĩ	11
5.6	Code xử lý Top nghệ sĩ	12
5.7	Xem được Top Bài hát	13
5.8	Code xử lý Top bài hát	14
5.9	Trang Nghe nhạc	15
5.10	Code xử lý Trang nghe nhạc	16
5.11	Trang xem Profile nghệ sĩ	17
5.12	Code xử lý Trang profile	18
5.13	Trang tìm kiếm bài hát	19
5.14	Code xử lý Trang tìm kiếm	20
6	Kết luận	21
6.1	Điểm còn hạn chế	21
6.2	Hướng phát triển trong tương lai	21



Danh sách hình vẽ

1	Đăng nhập	7
2	Code xử lý đăng nhập	8
3	Đăng ký	9
4	Code xử lý đăng ký	10
5	Top nghệ sĩ	11
6	Code xử lý Top nghệ sĩ	12
7	Top bài hát	13
8	Code xử lý Top bài hát	14
9	Trang nghe nhạc	15
10	Code xử lý Trang nghe nhạc	16
11	Trang profile nghệ sĩ	17
12	Code xử lý Trang Profile	18
13	Tìm kiếm bài hát	19
14	Code xử lý Trang tìm kiếm	20



1 Phân chia công việc từng thành viên

MSSV	Tên thành viên	Tên công việc	% Công việc
3119410476	Dịp Lâm Tuấn	Trang profile, Trang Music	33,33%
3119410046	Nguyễn Ngọc Trường Chinh	Trang Login, SignUp	33,33%
3120410539	Ngụy Đoan Toàn	Trang search, Trang chủ	33,33%



2 Lời mở đầu

Trong thời đại số hóa ngày nay, việc truy cập và thưởng thức âm nhạc thông qua internet đã trở thành một phần không thể tách rời trong cuộc sống hàng ngày của chúng ta. Từ những bản hit quốc tế đình đám đến những sáng tác nghệ sĩ local tài năng, website nghe nhạc đã trở thành nguồn tài nguyên vô tận, mang lại niềm vui và sự giải trí cho hàng triệu người trên khắp thế giới.

Trong khuôn khổ của đề án này, chúng em đã tập trung vào việc nghiên cứu và phát triển một website nghe nhạc hiện đại và tiện lợi, nhằm mang lại trải nghiệm nghe nhạc tốt nhất cho người dùng. Với sự kết hợp giữa công nghệ web tiên tiến và sự sáng tạo trong thiết kế giao diện, chúng em đã hướng đến mục tiêu tạo ra một nền tảng trực tuyến phong phú, đa dạng, và dễ sử dụng.

Trong phần tiếp theo của báo cáo, chúng em sẽ trình bày chi tiết về quá trình phân tích yêu cầu, thiết kế, và triển khai của dự án. Chúng em cũng sẽ giải thích các tính năng chính của website, cũng như các công nghệ và công cụ được sử dụng trong quá trình phát triển. Bên cạnh đó, chúng em cũng sẽ đề cập đến các thách thức mà chúng em đã đối mặt và cách chúng em đã vượt qua chúng để đạt được kết quả cuối cùng.

Chúng em hi vọng rằng đề án này không chỉ là một sản phẩm kỹ thuật mà còn là một bước đệm quan trọng trong việc hiểu rõ hơn về cách thức xây dựng và phát triển một website nghe nhạc thành công. Hy vọng rằng thông qua báo cáo này, chúng em có thể chia sẻ những kiến thức và kinh nghiệm của mình với cộng đồng, đồng thời góp phần làm giàu thêm nguồn lực kiến thức trong lĩnh vực này.



3 Tóm tắt

Website nghe nhạc này lấy ý tưởng từ trang website nổi tiếng **Spotify**. Bao gồm một vài tính năng như tìm kiếm bài hát, đăng ký / đăng nhập tài khoản, nghe nhạc và có thể tua về trước / sau của bài hát. Do thời gian hạn chế nên còn nhiều tính năng chưa kịp bổ sung. Nhưng đa phần cũng đáp ứng cơ bản một số chức năng của website nghe nhạc. Website sử dụng ngôn ngữ lập trình **Python** với thư viện **Django**. Thư viện Django giúp việc xây dựng website nhanh chóng và dễ dàng hơn.

4 Website nghe nhạc

4.1 Thư viện sử dụng

Tailwind CSS là một thư viện CSS hiệu quả và mạnh mẽ, được thiết kế để giúp bạn xây dựng giao diện web một cách nhanh chóng và linh hoạt. Không giống như các framework CSS truyền thống khác, **Tailwind** không định nghĩa sẵn các lớp CSS cụ thể cho các phần tử giao diện, mà thay vào đó nó cung cấp một tập hợp các lớp utility, giúp bạn kiểm soát toàn bộ giao diện của bạn thông qua HTML.

Với **Tailwind**, bạn có thể tùy chỉnh mọi chi tiết từ kích thước phông chữ, màu sắc, đến khoảng cách và căn chỉnh một cách linh hoạt. Điều này giúp bạn tạo ra các giao diện độc đáo và dễ dàng bảo trì. Bằng cách sử dụng lớp utility có sẵn, bạn có thể tạo ra các thành phần giao diện một cách nhanh chóng mà không cần phải viết CSS tùy chỉnh. Điều này giúp giảm thiểu thời gian phát triển và tăng tính linh hoạt trong việc thay đổi giao diện của bạn.

Django là một framework phát triển web mạnh mẽ được xây dựng bằng Python, được thiết kế để giúp các nhà phát triển xây dựng ứng dụng web nhanh chóng và hiệu quả. **Django** cung cấp một loạt các tính năng và công cụ có sẵn để giảm bớt công sức cần thiết cho việc phát triển web, bao gồm cơ sở dữ liệu tích hợp, quản lý định tuyến URL tự động, bảo mật tích hợp, và một hệ thống template mạnh mẽ.

Một trong những điểm mạnh của **Django** là sự tổ chức rõ ràng và cấu trúc mô-đun hóa, giúp các nhà phát triển dễ dàng quản lý dự án của mình và phát triển ứng dụng một cách có tổ chức. Ngoài ra, **Django** cũng có cộng đồng lớn và năng động, cung cấp nhiều tài liệu và hỗ trợ để giúp các nhà phát triển giải quyết các vấn đề và tối ưu hóa quy trình làm việc của họ. Tóm lại, **Django** là một lựa chọn lý tưởng cho việc phát triển các ứng dụng web từ nhỏ đến lớn, với sức mạnh và tính linh hoạt của Python.

4.2 Ngôn ngữ sử dụng

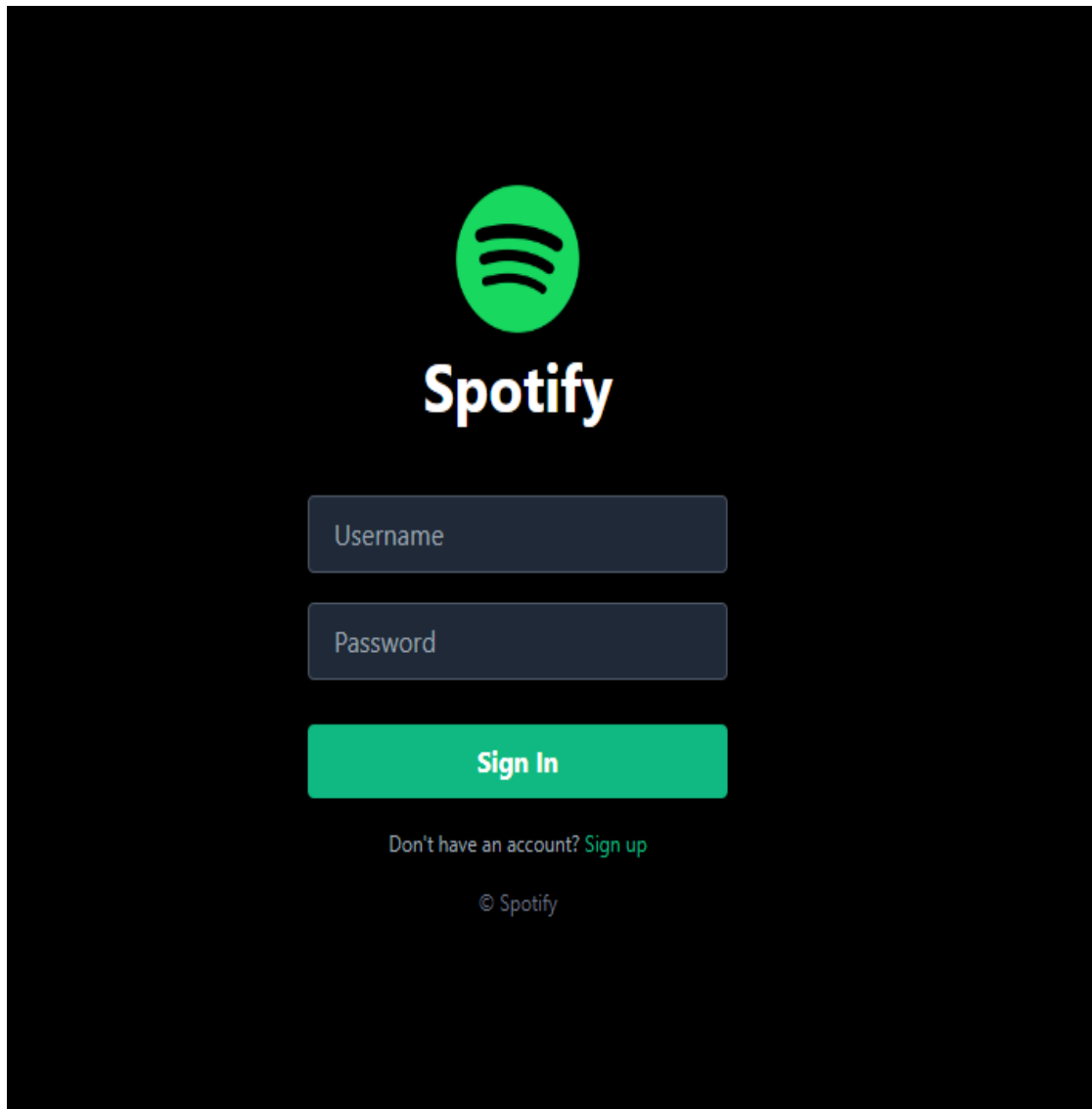
Python là một ngôn ngữ lập trình đa năng, dễ học và mạnh mẽ được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau. Với cú pháp đơn giản và rõ ràng, **Python** là lựa chọn lý tưởng cho cả người mới bắt đầu và các nhà phát triển kinh nghiệm.

Một trong những điểm mạnh của **Python** là cộng đồng lớn và năng động, cung cấp các thư viện và framework phong phú, giúp tăng tốc độ phát triển và giảm bớt công sức cần thiết cho việc xây dựng ứng dụng. **Python** cũng được sử dụng rộng rãi trong nghiên cứu khoa học, máy học, trí tuệ nhân tạo và phân tích dữ liệu, nhờ vào các thư viện như NumPy, Pandas và TensorFlow.



5 Các trang chính

5.1 Đăng nhập



Hình 1: Đăng nhập

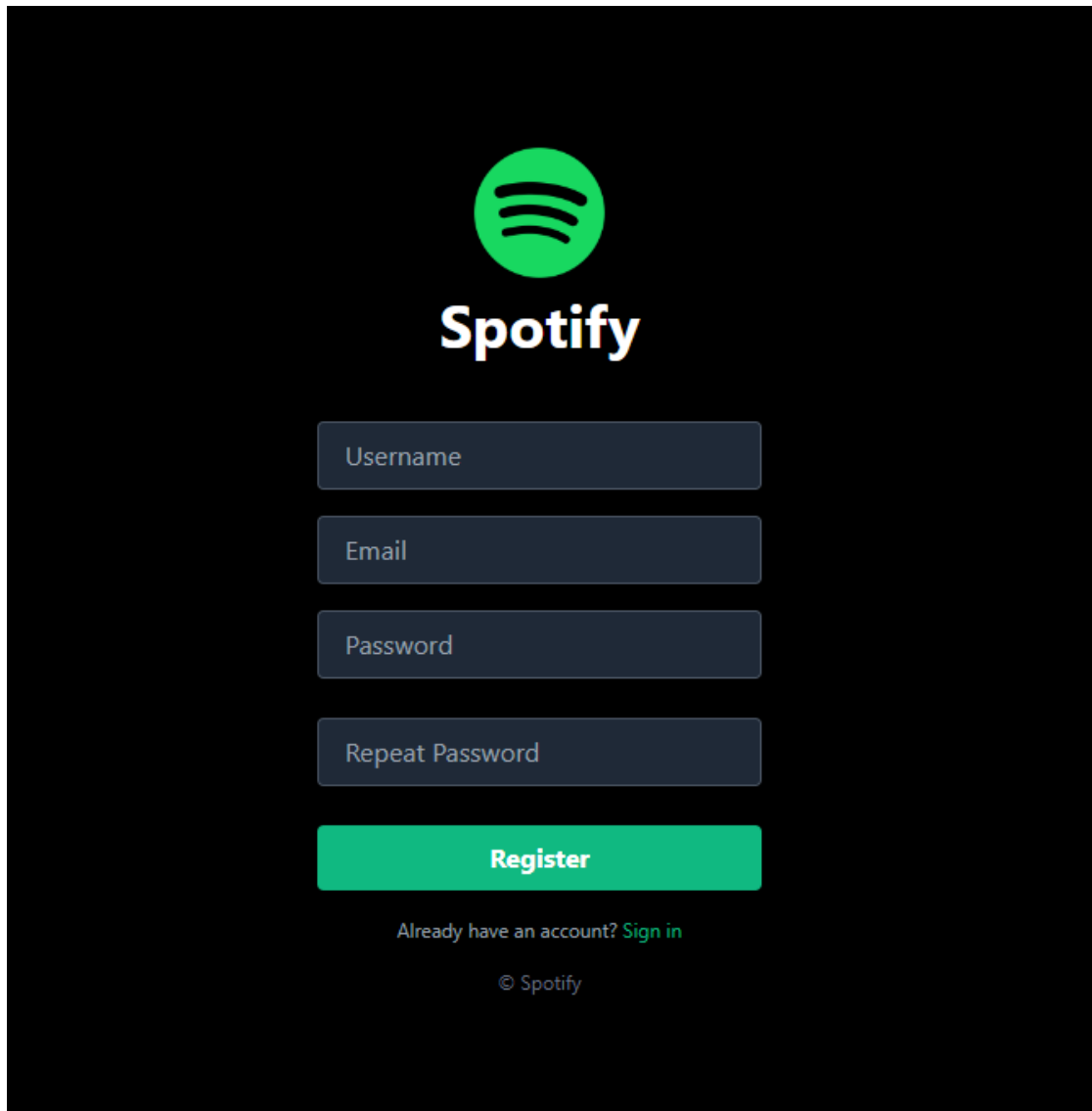
5.2 Code xử lý Đăng nhập

```
def login(request):  
    if request.method == "POST":  
        username = request.POST['username']  
        password = request.POST['password']  
  
        user = auth.authenticate(username=username, password=password)  
  
        if user is not None:  
            auth.login(request, user)  
            return redirect('/')  
        else:  
            messages.info(request, 'Credentials Invalid')  
            return redirect('login')  
    else:  
        return render(request, 'login.html')
```

Hình 2: Code xử lý đăng nhập



5.3 Đăng Ký



The image shows the Spotify registration interface. At the top, the Spotify logo (a green circle with three curved lines) is centered above the word "Spotify" in a white, bold, sans-serif font. Below the logo, there are four dark gray input fields stacked vertically, each with a light gray placeholder text: "Username", "Email", "Password", and "Repeat Password". Below these fields is a prominent red "Register" button with white text. Under the button, there is a link that says "Already have an account? Sign in" in a smaller, lighter font. At the very bottom, the text "© Spotify" is displayed in a small, light gray font.

Hình 3: Đăng ký

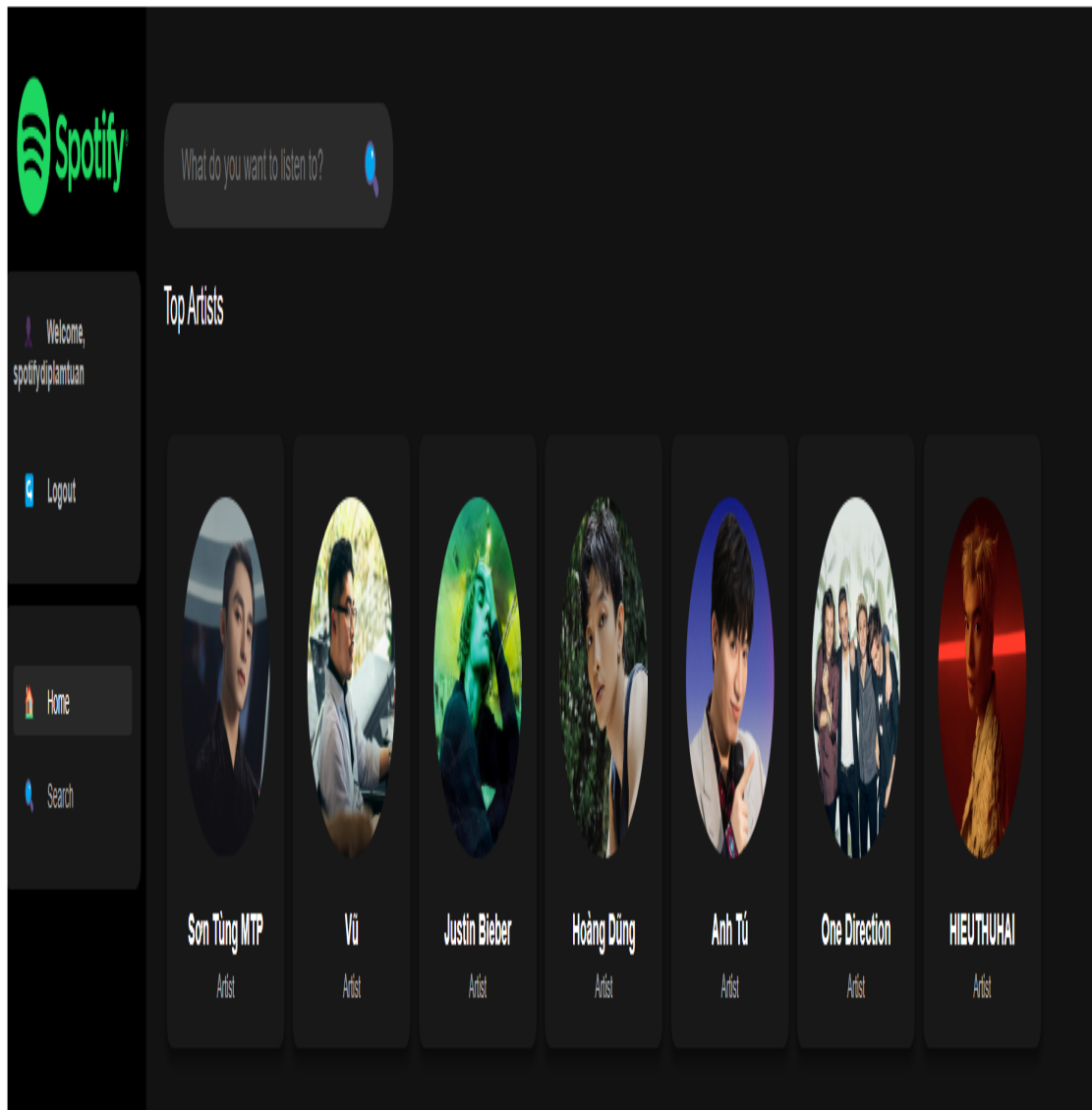
5.4 Code xử lý Đăng ký

```
def signup(request):
    if request.method == 'POST':
        email = request.POST['email']
        username = request.POST['username']
        password = request.POST['password']
        password2 = request.POST['password2']
        if password == password2:
            if User.objects.filter(email=email).exists():
                messages.info(request, 'Email Taken')
                return redirect('signup')
            elif User.objects.filter(username=username).exists():
                messages.info(request, 'Username Taken')
                return redirect('signup')
            else:
                user = User.objects.create_user(username=username, email=email, password=password)
                user.save()

                # login user in
                user_login = auth.authenticate(username=username, password=password)
                auth.login(request, user_login)
                return redirect('/')
        else:
            messages.info(request, 'Password not matching')
            return redirect('signup')
    else:
        return render(request, 'signup.html')
```

Hình 4: Code xử lý đăng ký

5.5 Xem được Top nghệ sĩ



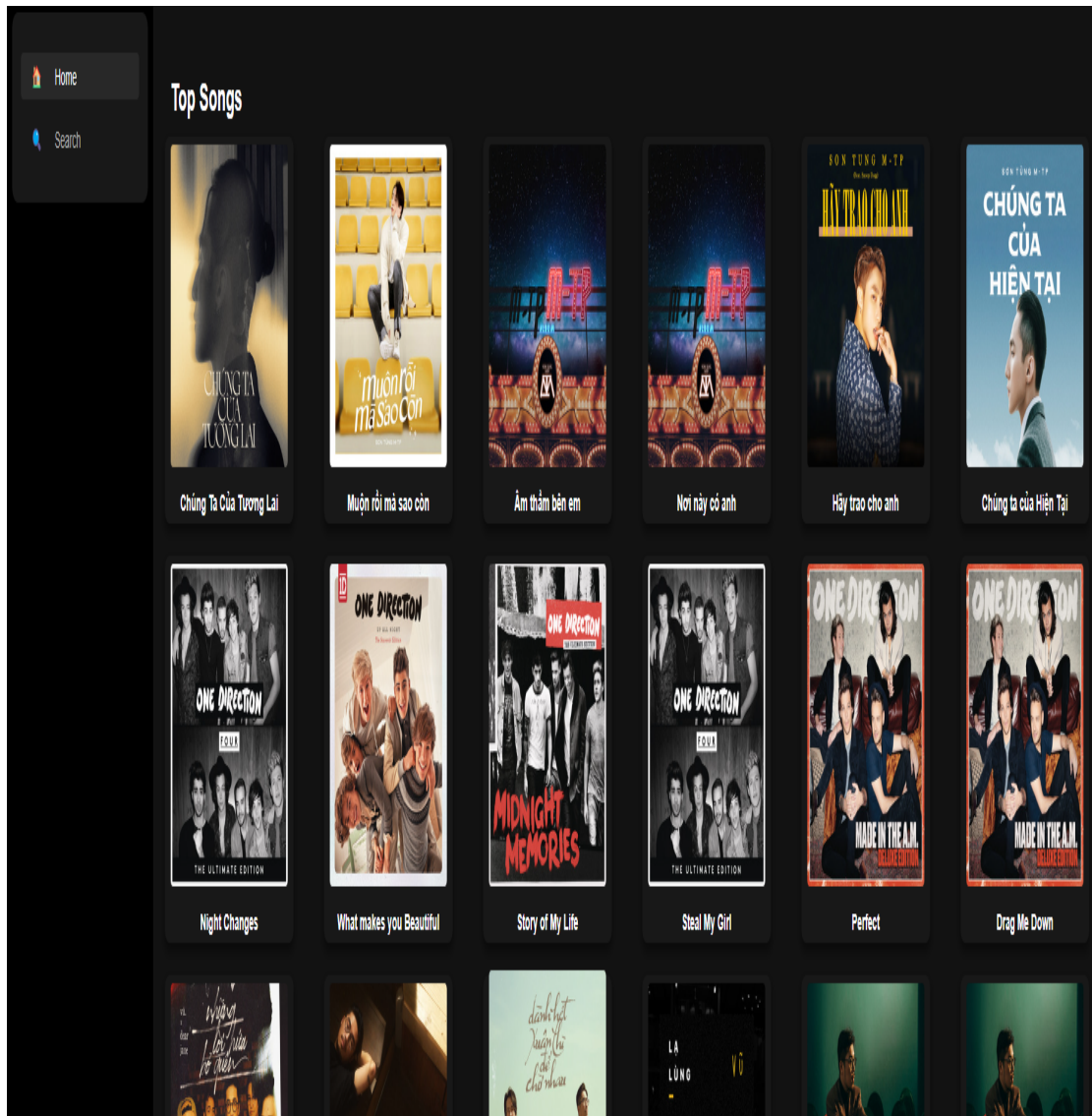
Hình 5: Top nghệ sĩ

5.6 Code xử lý Top nghệ sĩ

```
def top_artists():  
    url = "https://65d16bd9ab7beba3d5e455e6.mockapi.io/todolist/artists"  
    response = requests.get(url)  
    response_data = response.json()  
    artists_info = []  
  
    for artist in response_data:  
        name = artist.get('nameArtist', 'No Name')  
        idArtist = artist.get('idArtist', 'No ID')  
        imageArtist = artist.get('imageArtist', 'No Image')  
        artists_info.append((name, idArtist, imageArtist))  
    return artists_info
```

Hình 6: Code xử lý Top nghệ sĩ

5.7 Xem được Top Bài hát



Hình 7: Top bài hát

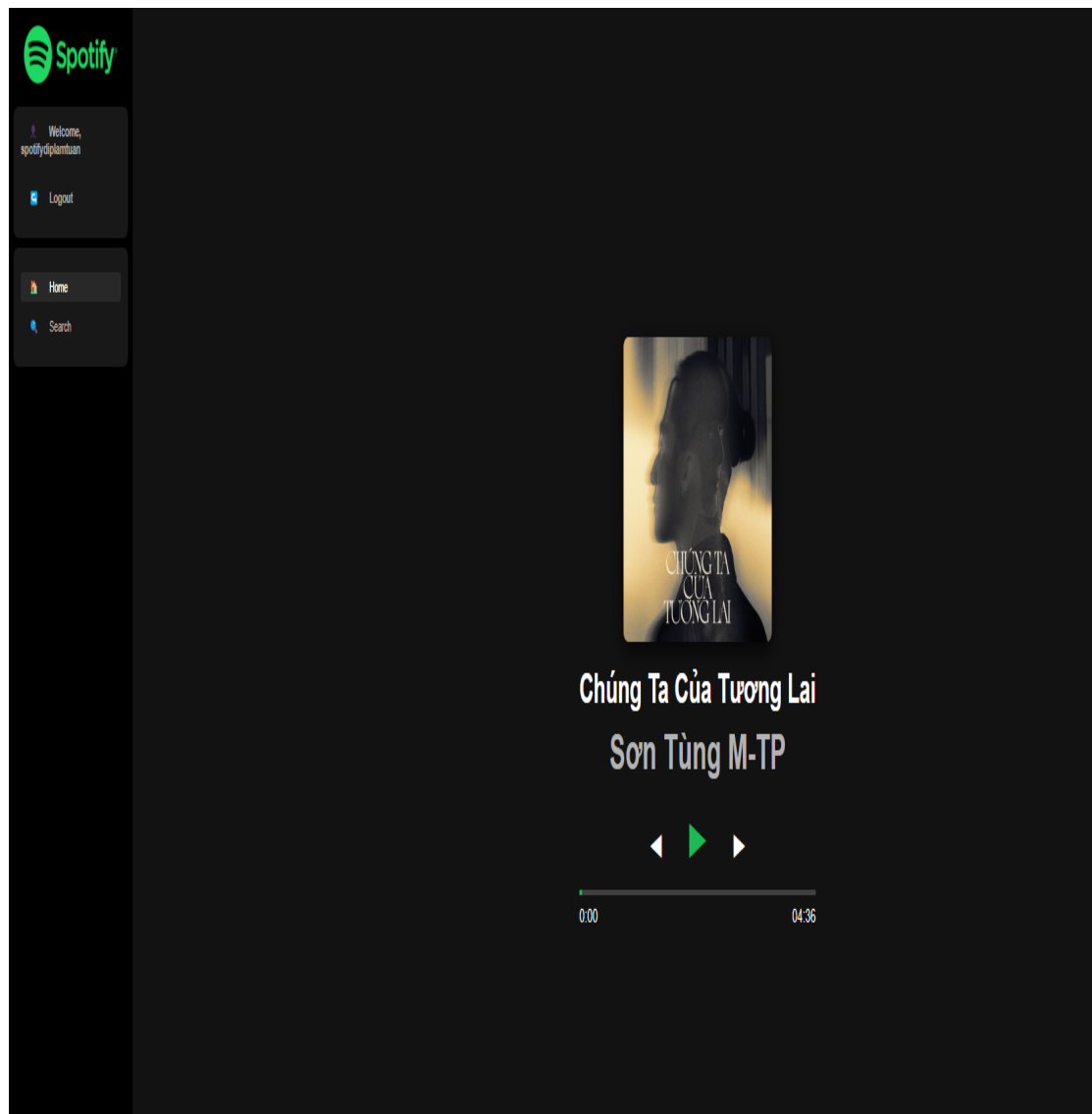
5.8 Code xử lý Top bài hát

```
def top_tracks():  
    url = "https://66160055b8b8e32ffc7c19db.mockapi.io/toptracks"  
    response = requests.get(url)  
    response_data = response.json()  
    track_details=[]  
  
    for track in response_data:  
        track_id = track['track_id']  
        track_name = track['track_name']  
        artist_name = track['artist_name']  
        cover_url = track['cover_url']  
  
        track_details.append({  
            'track_id':track_id,  
            'track_name':track_name,  
            'artist_name':artist_name,  
            'cover_url':cover_url  
        })  
    return track_details
```

Hình 8: Code xử lý Top bài hát



5.9 Trang Nghe nhạc



Hình 9: Trang nghe nhạc

5.10 Code xử lý Trang nghe nhạc

```
@login_required(login_url='login')
def music(request,pk):
    track_id = pk
    url = "https://spotify-scraper.p.rapidapi.com/v1/track/metadata"
    querystring = {"trackId":track_id}
    headers = {
        "X-RapidAPI-Key": "d01f11a307mshf1c4745d6bd7c6fp1d217fjsn80edc9228357",
        "X-RapidAPI-Host": "spotify-scraper.p.rapidapi.com"
    }
    response = requests.get(url, headers=headers, params=querystring)
    if response.status_code == 200:
        data = response.json()
        track_name = data.get("name")
        artists_list = data.get("artists",[])
        first_artist_name = artists_list[0].get('name') if artists_list else "No artist found"
        audio_details_query=track_name + first_artist_name
        audio_details = get_audio_details(audio_details_query)
        audio_url= audio_details[0]
        duration_text = audio_details[1]
        image_url = audio_details[2]
        context = {
            'track_name':track_name,
            'artist_name':first_artist_name,
            'audio_url':audio_url,
            'duration_text':duration_text,
            'image_url':image_url,
        }
    return render(request,'music.html',context)
```

Hình 10: Code xử lý Trang nghe nhạc



5.11 Trang xem Profile nghệ sĩ

Spotify

Welcome,
spotifydiphuan

Logout

Home

Search

Verified Artist

Sơn Tùng M-TP
1,405,165 monthly listeners

Songs

	Chúng Ta Của Tương Lai Sơn Tùng M-TP	11,437,796	04:09
	Chúng Ta Của Hiện Tại Sơn Tùng M-TP	44,662,643	05:01
	Muộn Rồi Mà Sao Còn Sơn Tùng M-TP	37,423,272	04:35
	Âm Thầm Bên Em Sơn Tùng M-TP	20,633,321	04:53
	Nơi Này Có Anh Sơn Tùng M-TP	21,094,712	04:20
	Buông Đôi Tay Nhau Ra Sơn Tùng M-TP	12,470,660	03:47
	Có Chắc Yêu Là Đây Sơn Tùng M-TP	24,271,537	03:22
	Hãy Trao Cho Anh Sơn Tùng M-TP	25,407,827	04:05

Hình 11: Trang profile nghệ sĩ

5.12 Code xử lý Trang profile

```
@login_required(login_url='login')
def profile(request,pk):
    artist_id = pk
    url = "https://spotify-scraper.p.rapidapi.com/v1/artist/overview"
    querystring = {"artistId":artist_id}
    headers = {
        "X-RapidAPI-Key": "d01f11a307mshf1c4745d6bd7c6fp1d217fjsn80edc9228357",
        "X-RapidAPI-Host": "spotify-scraper.p.rapidapi.com"
    }
    response = requests.get(url, headers=headers, params=querystring)
    if response.status_code == 200:
        data = response.json()

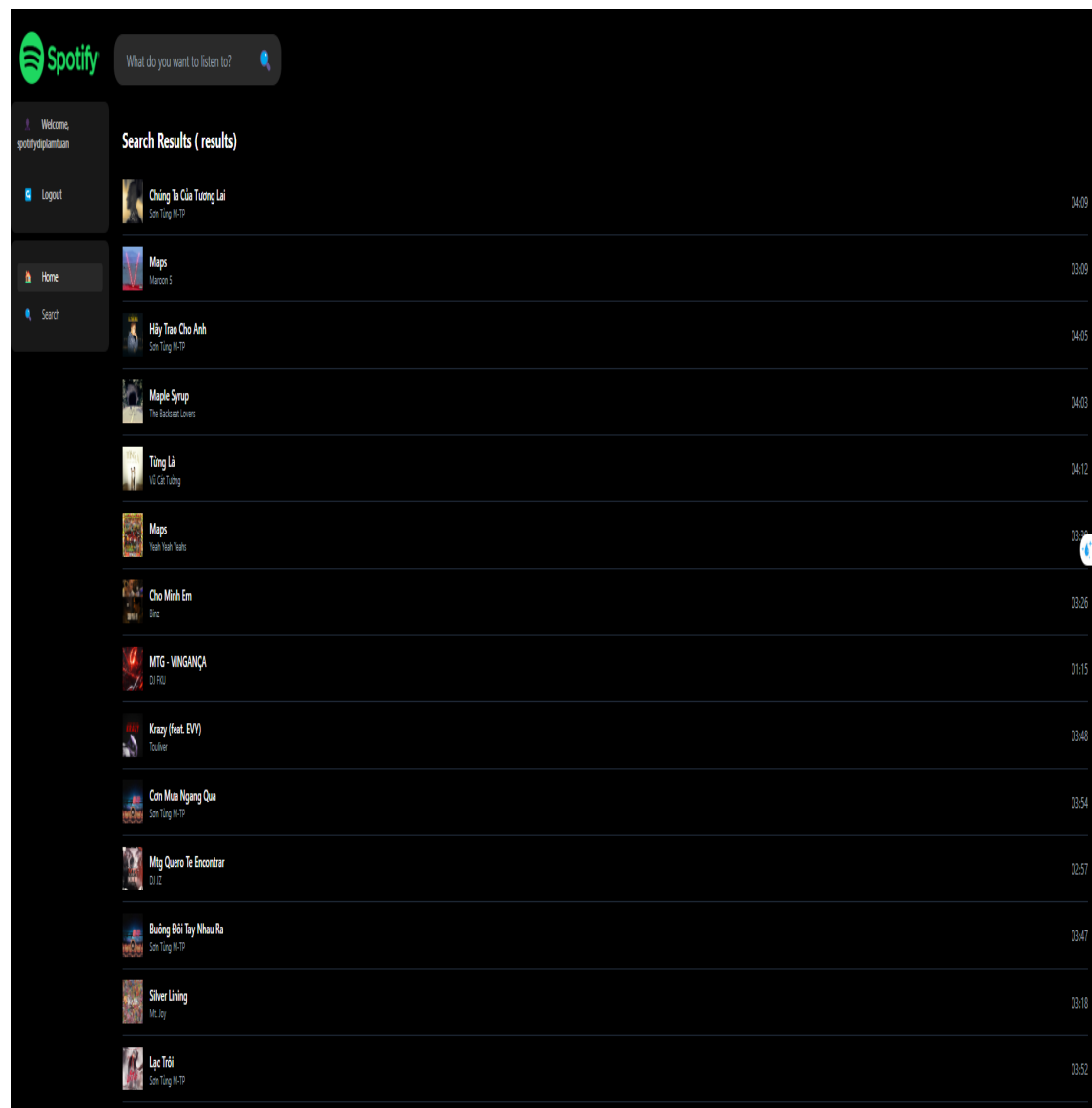
        name = data['name']
        monthly_listeners = int(data['stats']['monthlyListeners'])
        formatMonthlyListeners = f'{monthly_listeners:,}'
        header_url = data['visuals']['header'][0]['url']
        top_tracks = []
        for track in data['discography']['topTracks']:
            trackId = str(track['id'])
            trackName = str(track['name'])
            trackImage = str(track['album']['cover'][0]['url'])
            formatPlayCount = f'{track["playCount"]:,}'
            track_info = {
                "id":trackId,
                "name":trackName,
                "track_image":trackImage,
                "durationText":track['durationText'],
                "playCount":formatPlayCount
            }
            top_tracks.append(track_info)

        artist_data = {
            'name':name,
            'monthly_listeners':formatMonthlyListeners,
            'header_url':header_url,
            'top_tracks': top_tracks
        }
    else:
        artist_data={}
    return render(request,'profile.html',artist_data)
```

Hình 12: Code xử lý Trang Profile



5.13 Trang tìm kiếm bài hát



Hình 13: Tìm kiếm bài hát

5.14 Code xử lý Trang tìm kiếm

```
@login_required(login_url='login')
def search(request):
    if request.method == 'POST':
        seach_query = request.POST['search_query']
        url = "https://spotify-scraper.p.rapidapi.com/v1/search"

        querystring = {"term":seach_query,"type":"track"}

        headers = {
            "X-RapidAPI-Key": "d01f11a307mshf1c4745d6bd7c6fp1d217fjsn80edc9228357",
            "X-RapidAPI-Host": "spotify-scraper.p.rapidapi.com"
        }
        response = requests.get(url, headers=headers, params=querystring)
        track_list = []
        if response.status_code == 200:
            data = response.json()
            seach_result_count = data['tracks']['totalCount']
            tracks = data['tracks']['items']

            for track in tracks:
                track_name = track['name']
                artist_name = track['artists'][0]['name']
                duration = track['durationText']
                trackId = track['id']
                trackImage = track['album']['cover'][0]['url']

                track_list.append({
                    'track_name':track_name,
                    'artist_name':artist_name,
                    'duration':duration,
                    'trackId':trackId,
                    'trackImage':trackImage,
                })
            context = {
                'seach_result_count':seach_result_count,
                'track_list':track_list,
            }
            return render(request, 'search.html', context)
        else:
            return render(request, 'search.html')
```

Hình 14: Code xử lý Trang tìm kiếm



6 Kết luận

Đề tài xây dựng "**Website nghe nhạc**" cũng xuất phát sở thích của nhóm, thông qua việc xây dựng website nghe nhạc này giúp cả nhóm hiểu được cách một website nghe nhạc hoạt động như thế nào.

6.1 Điểm còn hạn chế

Mặc dù đã cố gắng tìm hiểu nhưng do khả năng cũng như thời gian hạn chế nên việc thực hiện website gặp nhiều khó khăn. Dẫn đến chưa có nhiều chức năng mong muốn cũng như còn một số lỗi .

6.2 Hướng phát triển trong tương lai

Tìm hiểu sâu hơn về các chức năng của website nghe nhạc, như **download nhạc về máy**, **Tạo playlist nhạc theo sở thích của cá nhân**,..

Để hoàn thành được đề tài này em xin cảm ơn đến thầy **Thạc Sĩ Từ Lăng Phiêu** người đã giúp đỡ và quan tâm đến nhóm em trong thời gian qua. Em xin bày tỏ lòng biết ơn sâu sắc đến thầy.