

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Xây dựng ứng dụng Web

Nghe nhạc Streaming sử dụng
Flask, Youtube API, pytube
trên Python

GVHD: Từ Lãng Phiêu

SV: Võ Minh Diền - 3121410156
Nguyễn Hữu Đức - 3121410163
Chu Võ Đại - 3121410133

TP. HỒ CHÍ MINH, THÁNG 05/2024

Mục lục

1 Phân giới thiệu	2
1.1 Giới thiệu về ứng dụng	2
1.2 Các ngôn ngữ, các thư viện và cơ sở dữ liệu	2
1.2.1 Các ngôn ngữ đã sử dụng	2
1.2.2 Các thư viện Python đã sử dụng	2
1.2.3 Cơ sở dữ liệu	3
2 Thành viên và phân công công việc	4
3 Chi tiết các chức năng của trang web	5
3.1 Chức năng tìm kiếm bài hát	5
3.2 Chức năng xem danh sách trending	7
3.3 Chức năng nghe nhạc	9
3.4 Chức năng các nút trên thanh điều khiển	11
3.5 Chức năng tải nhạc	11
3.6 Chức năng Đăng nhập, Đăng ký	14
3.7 Chức năng lưu bài hát vào danh sách yêu thích	19
3.8 Chức năng Xem danh sách bài hát yêu thích	21
3.9 Chức năng Xem danh sách bài hát đã nghe	23
4 Tổng kết	26
4.1 Ưu điểm	26
4.2 Nhược điểm	26
4.3 Hướng phát triển	26



1 Phân giới thiệu

1.1 Giới thiệu về ứng dụng

YouMusic là một ứng dụng web đơn giản được phát triển bằng Python và Flask, cho phép người dùng tìm kiếm và phát nhạc trực tuyến từ YouTube. Ứng dụng cũng cho phép người dùng đăng nhập, đăng ký tài khoản để lưu danh sách yêu thích và lịch sử phát nhạc.

Các chức năng cơ bản của trang web

1. Truy cập vào trang web và tìm kiếm bài hát bằng cách nhập từ khóa vào ô tìm kiếm hoặc xem danh sách các bài hát phổ biến.
2. Đăng nhập hoặc đăng ký tài khoản để lưu danh sách yêu thích và lịch sử phát nhạc.
3. Phát nhạc trực tuyến từ YouTube và tải về bài hát yêu thích của bạn.
4. Thao tác với các nút điều khiển cơ bản của một web nghe nhạc
5. Xem lại lịch sử phát nhạc và quản lý danh sách yêu thích của bạn.

1.2 Các ngôn ngữ, các thư viện và cơ sở dữ liệu

1.2.1 Các ngôn ngữ đã sử dụng

- **HTML, CSS, Javascript:** Giao diện và giao tiếp thông tin
- **Python:** Xử lý các chức năng của web, xử lý dữ liệu

1.2.2 Các thư viện Python đã sử dụng

- **Flask**

Thư viện Flask là một framework web siêu nhẹ cho Python, được sử dụng để xây dựng các ứng dụng web. Flask được thiết kế để đơn giản, linh hoạt và dễ dàng để sử dụng, đặc biệt là cho các ứng dụng web nhỏ và trung bình.

Một số đặc điểm của Flask bao gồm:

- Linh hoạt: Flask không bắt buộc sử dụng một cấu trúc dự án cụ thể nào và cho phép bạn tự do tổ chức mã của mình theo cách bạn muốn.
- Ít dư thừa: Flask giữ một lượng ít nhất các thành phần cần thiết để xây dựng ứng dụng web, giúp giảm thiểu sự phức tạp và đảm bảo hiệu suất tốt.
- Mở rộng dễ dàng: Flask cung cấp các extension để mở rộng chức năng của nó, cho phép bạn tích hợp các tính năng như đăng nhập, xác thực, gửi email, và nhiều hơn nữa một cách dễ dàng.
- Jinja2 Templates: Flask sử dụng Jinja2, một ngôn ngữ mấu mạnh mẽ, để tạo ra các trang web động.
- WSGI compatibility: Flask tuân thủ chuẩn WSGI, cho phép tích hợp dễ dàng với các server WSGI như Gunicorn hoặc uWSGI.



- HTTP Request Handling: Flask cung cấp các decorator để định nghĩa các route và xử lý các yêu cầu HTTP GET, POST, PUT, DELETE, vv.
- Mã nguồn mở: Flask là một dự án mã nguồn mở, có nghĩa là bạn có thể xem mã nguồn của nó, đóng góp và sửa lỗi nếu cần.

Cài đặt:

pip install Flask

- **Pytube**

Pytube là một thư viện Python cho phép bạn tải xuống video và âm nhạc từ YouTube. Pytube cung cấp các phương thức đơn giản để tìm kiếm, tải xuống và xử lý video YouTube từ mã Python của bạn.

Cài đặt:

pip install pytube

- **mysql.connector**

mysql.connector là một thư viện Python cho phép kết nối và tương tác với cơ sở dữ liệu MySQL từ mã Python của bạn. Thư viện này cung cấp các công cụ để thực hiện các truy vấn SQL, thực hiện thao tác với cơ sở dữ liệu, và xử lý kết quả trả về từ cơ sở dữ liệu.

Cài đặt:

pip install mysql-connector-python

- **googleapiclient**

googleapiclient là một thư viện Python cung cấp giao diện để tương tác với các dịch vụ của Google thông qua API của họ. Thư viện này giúp bạn gửi yêu cầu và nhận phản hồi từ các dịch vụ như Google Drive, Gmail, Google Calendar và nhiều dịch vụ khác.

Cài đặt:

pip install google-api-python-client

- **pathlib và os**

pathlib và os là hai thư viện trong Python được sử dụng để làm việc với đường dẫn file và thư mục trong hệ thống tệp của bạn.

1.2.3 Cơ sở dữ liệu

MySQL: Ứng dụng web nghe nhạc YouMusic sử dụng hệ quản trị cơ sở dữ liệu MySQL để quản lý các dữ liệu trong ứng dụng gồm:

- **user:**

Lưu thông tin tài khoản của người dùng

- name: tên đăng nhập của người dùng (khóa chính)

- password: mật khẩu đăng nhập của người dùng

- **favorite:**

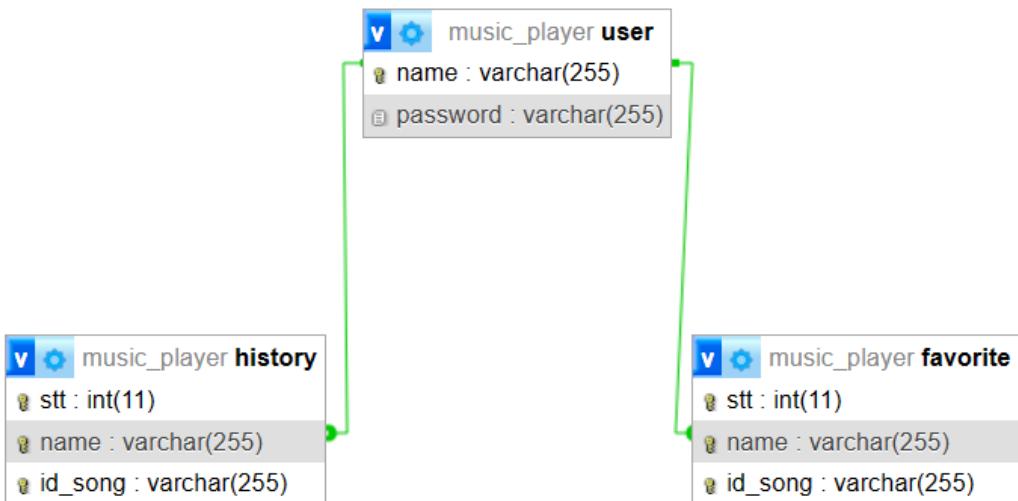
Lưu thông tin các bài hát yêu thích của người dùng

- stt: số thứ tự (khóa chính)
- name: tên đăng nhập của người dùng (khóa ngoại kết nối với name của bảng user)
- idsong: id của bài hát yêu thích

- **history:**

Lưu thông tin các bài hát đã nghe của người dùng

- stt: số thứ tự (khóa chính)
- name: tên đăng nhập của người dùng (khóa ngoại kết nối với name của bảng user)
- idsong: id của bài hát đã nghe



2 Thành viên và phân công công việc



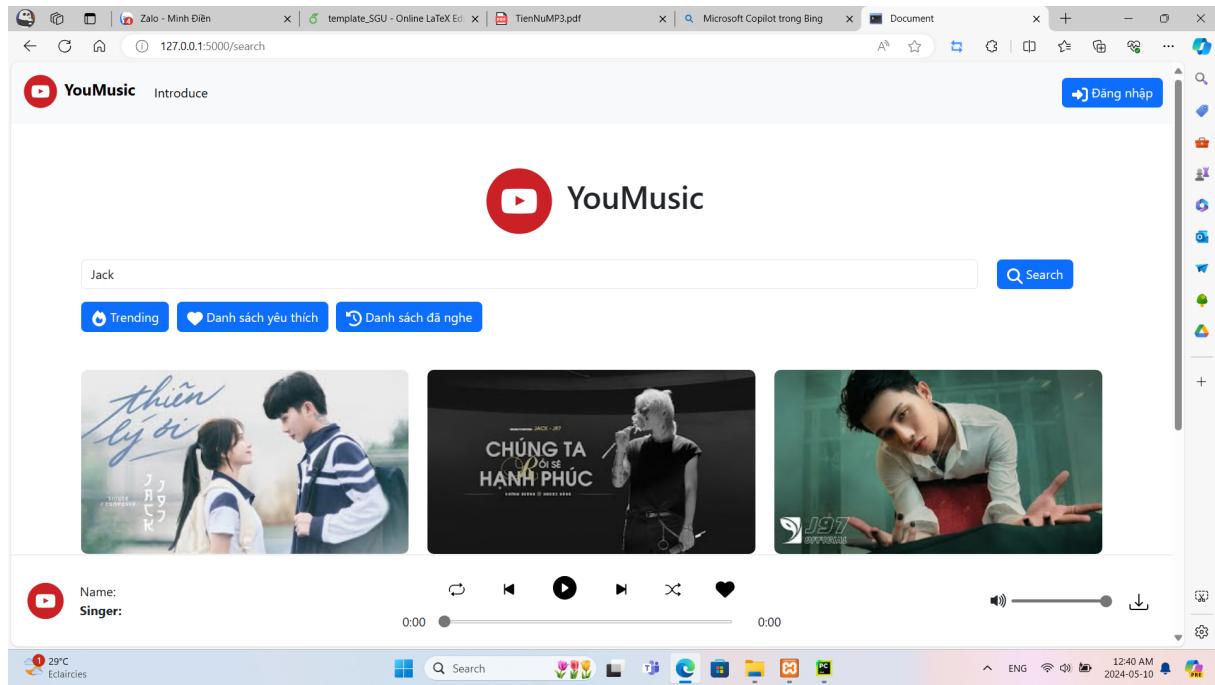
Thành viên	MSSV	Phần trăm công việc	Công việc
Võ Minh Diền	3121410156	35%	Giao diện, chức năng phát nhạc, tải nhạc, tự động chuyển bài hát, lặp lại bài hát, lưu bài hát yêu thích, chuyển bài hát
Nguyễn Hữu Đức	3121410163	35%	Dăng nhập, đăng ký, tìm kiếm, danh sách đã nghe, danh sách yêu thích
Chu Võ Đại	3121410133	30%	Giao diện, danh sách trending, powerpoint

Bảng 1: Thành viên và phân công công việc

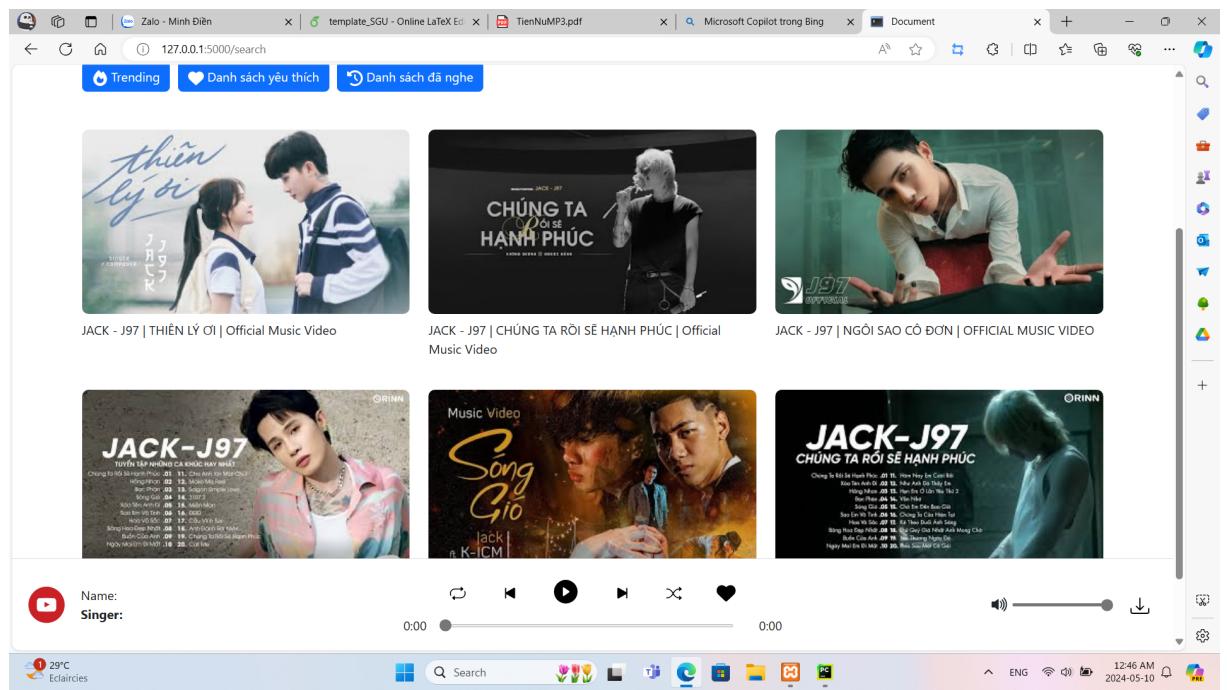
3 Chi tiết các chức năng của trang web

3.1 Chức năng tìm kiếm bài hát

- Người dùng gõ nội dung tìm kiếm bài hát (tên bài hát, tên ca sĩ, v.v) sau đó nhấn Enter hoặc Button "Search"



- Trang web sẽ hiện ra danh sách các bài hát có liên quan đến từ khóa tìm kiếm



- Code xử lý:

```
@app.route('/search', methods=['GET', 'POST'])
def search():

    if request.method == 'POST':
        search_response = youtube.search().list(
            q=request.form['query'],
            part='snippet',
            type='video',
            videoCategoryId='10',
            maxResults=6
        ).execute()

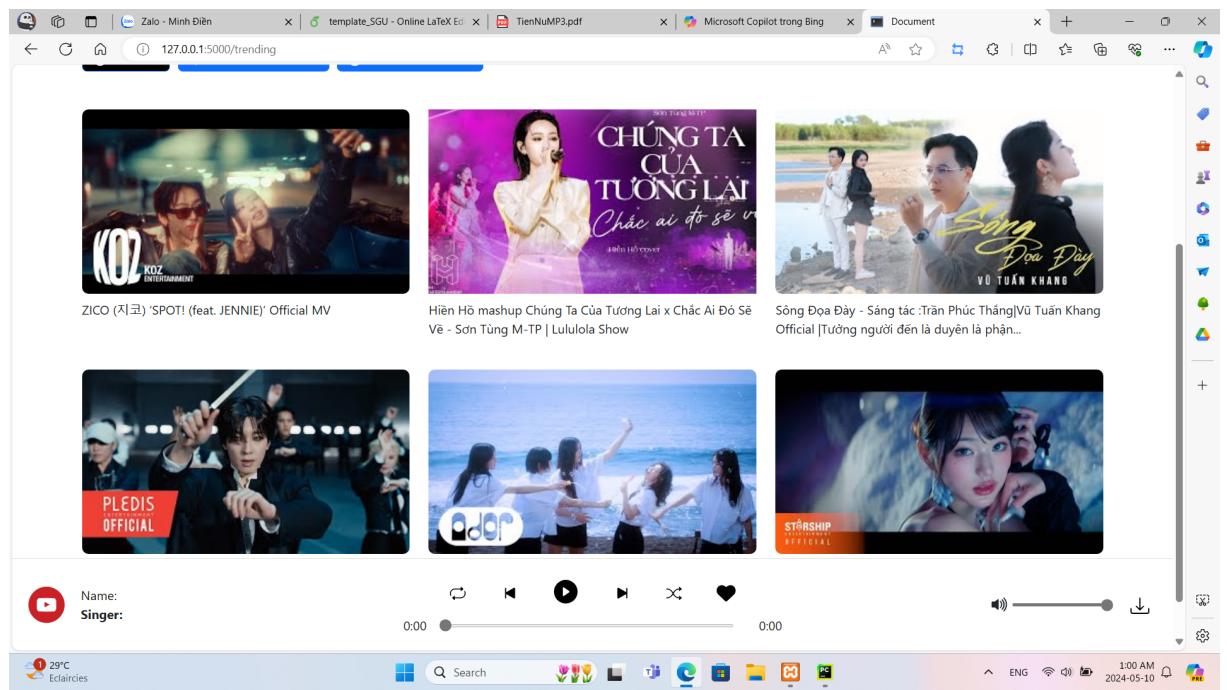
        songs = []

        for search_result in search_response.get('items', []):
            video_id = search_result['id']['videoId']
            song = getInfoSong(video_id)
            songs.append(song)
        global next_page_token
        next_page_token = 'index'
        return render_template_with_session('index.html', songs=songs)
```

3.2 Chức năng xem danh sách trending

- Người dùng chọn nút Trending trên màn hình, trang web sẽ hiển thị các bài hát đang thuộc top thịnh hành âm nhạc của Youtube Việt Nam





- Code xử lý:

```
@app.route('/trending', methods=['GET', 'POST'])
def trending():

    trending_response = youtube.videos().list(
        part='snippet',
        chart='mostPopular',
        regionCode='VN',
        videoCategoryId='10',
        maxResults=6
    ).execute()

    songs = []

    for trending_result in trending_response.get('items', []):
        video_id = trending_result['id']
        song = getInfoSong(video_id)
        songs.append(song)
    global next_page_token
    next_page_token = 'trending'
    return render_template_with_session('index.html', songs=songs)
```

3.3 Chức năng nghe nhạc

- Người dùng chọn một bài hát bất kỳ, chương trình sẽ truyền dữ liệu về giao diện giúp người dùng có thể nghe nhạc, xem tên ca sĩ, thời lượng, ảnh của bài hát, tăng giảm âm lượng bài hát



Sau lời từ khước (OST MAI) | Phan Mạnh Quỳnh | Lyrics



The screenshot shows a web browser window with three tabs open:

- Zalo - Minh Điện
- template_SGU - Online LaTeX Ed.
- TienNuMP3.pdf

The main content area displays the YouMusic search results for "SAU LỜI TỪ KHƯỚC".

YouMusic

Enter search...

Trending | Danh sách yêu thích | Danh sách đã nghe

Three video thumbnails are shown:

- Sau lời từ khước (OST MAI) | Phan Mạnh Quỳnh | Lyrics video**
- Phan Mạnh Quỳnh Playlist | Viet Nam Music Playlist**
- SAU LỜI TỪ KHƯỚC (Theme Song From "MAI", đạo diễn Trần Thành) | Sáng tác & trình bày Phan Mạnh Quỳnh**

A media player at the bottom shows the first video's progress: 2:21 / 4:10. The video title is "Sau lời từ khước (OST MAI) | Phan Mạnh Quỳnh | Lyrics video" and the artist is "Phan Mạnh Quỳnh Official".

System tray icons include weather (29°C Eclaircie), search, file explorer, taskbar, and system status (ENG, 1:12 AM, 2024-05-10).

- **Code xử lý:**

```
class Song:  
    def __init__(self, id, title, artist, thumbnail, songUrl, audioUrl):  
  
        self.id = id  
        self.title = title  
        self.artist = artist  
        self.thumbnail = thumbnail  
        self.songUrl = songUrl  
        self.audioUrl = audioUrl  
  
    def getInfoSong(id):  
        request = youtube.videos().list(part='snippet', id=id).execute()  
        songResponse = request['items'][0]['snippet']  
        thumbnail_url = songResponse['thumbnails']['medium']['url']  
        songUrl = f'https://www.youtube.com/watch?v={id}'  
  
        video = YouTube(songUrl)  
        bestAudio = video.streams.filter(only_audio=True).first()  
        audioUrl = bestAudio.url  
        songInfo = Song(id, songResponse['title'],  
                        songResponse['channelTitle'], thumbnail_url, songUrl, audioUrl)  
        return songInfo
```

3.4 Chức năng các nút trên thanh điều khiển

- Trên thanh điều khiển, người dùng có thể thực hiện các thao tác như lặp lại bài hát, dừng bài hát, chuyển bài hát, tự động chuyển bài hát, tăng giảm âm lượng



3.5 Chức năng tải nhạc

- Người dùng chọn một bài hát bất kỳ, nhấn nút tải nhạc trên thanh điều khiển



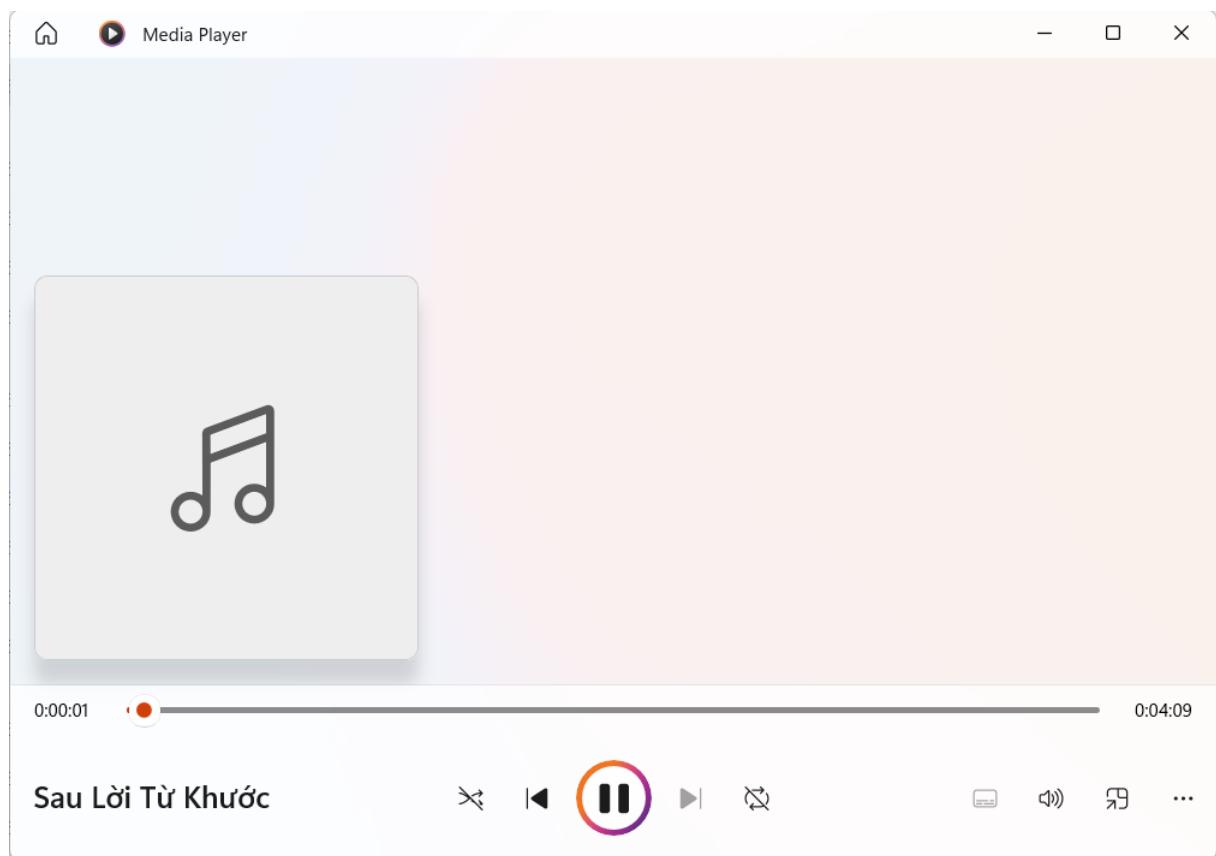
- Hiển thị thông báo thành công

127.0.0.1:5000 cho biết

Tải bài hát thành công!

OK

- Bài hát đã lưu về máy



- Code xử lý:



```
@app.route('/download', methods=['POST'])
def download():
    if request.method == 'POST':
        songUrl = request.form['songUrl']
        yt = YouTube(songUrl)
        video = yt.streams.filter(only_audio=True).first()

        # Get the path to the Downloads folder
        downloads_path = str(Path.home() / "Downloads")

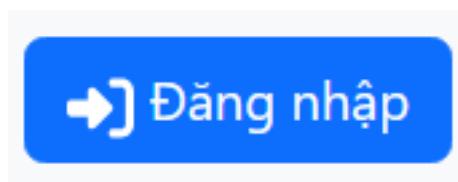
        # Tạo tên file mới
        file_name = video.default_filename
        base, ext = os.path.splitext(file_name)
        file_name = f"{base}.mp3" # Thêm đuôi ".mp3" vào tên file

        # Kiểm tra và điều chỉnh tên file nếu cần thiết
        count = 1
        while os.path.exists(os.path.join(downloads_path, file_name)):
            file_name = f"{base}_{count}.mp3"
            count += 1

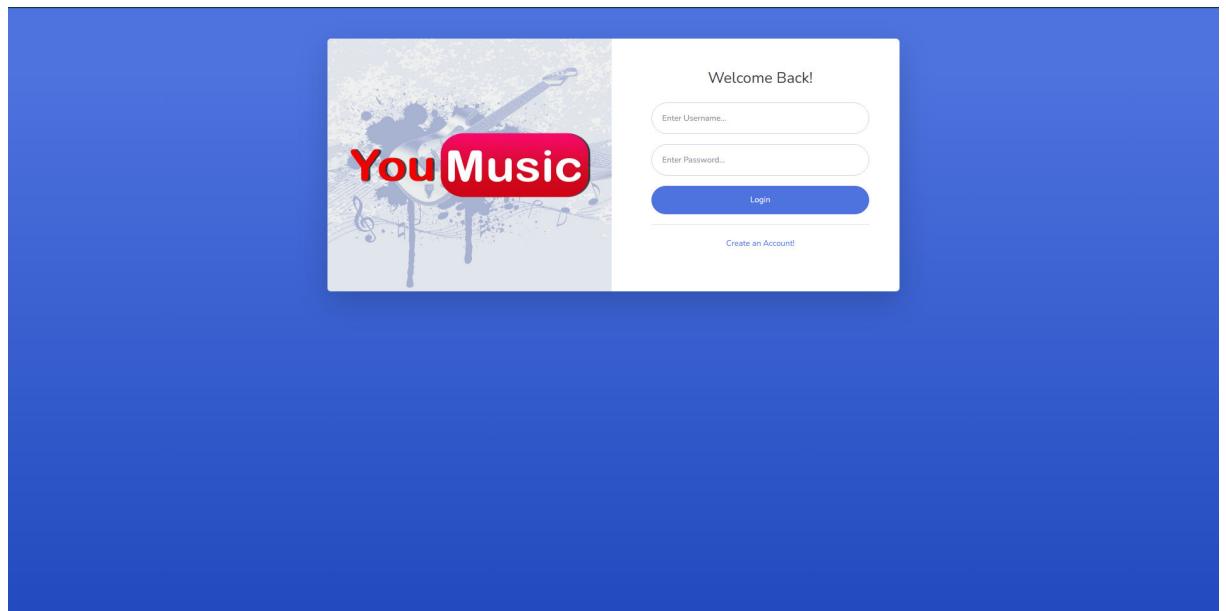
        outfile = video.download(output_path=downloads_path, filename=file_name)
    return jsonify({'message': 'Downloaded successfully!', 'file_name': file_name})
```

3.6 Chức năng Đăng nhập, Đăng ký

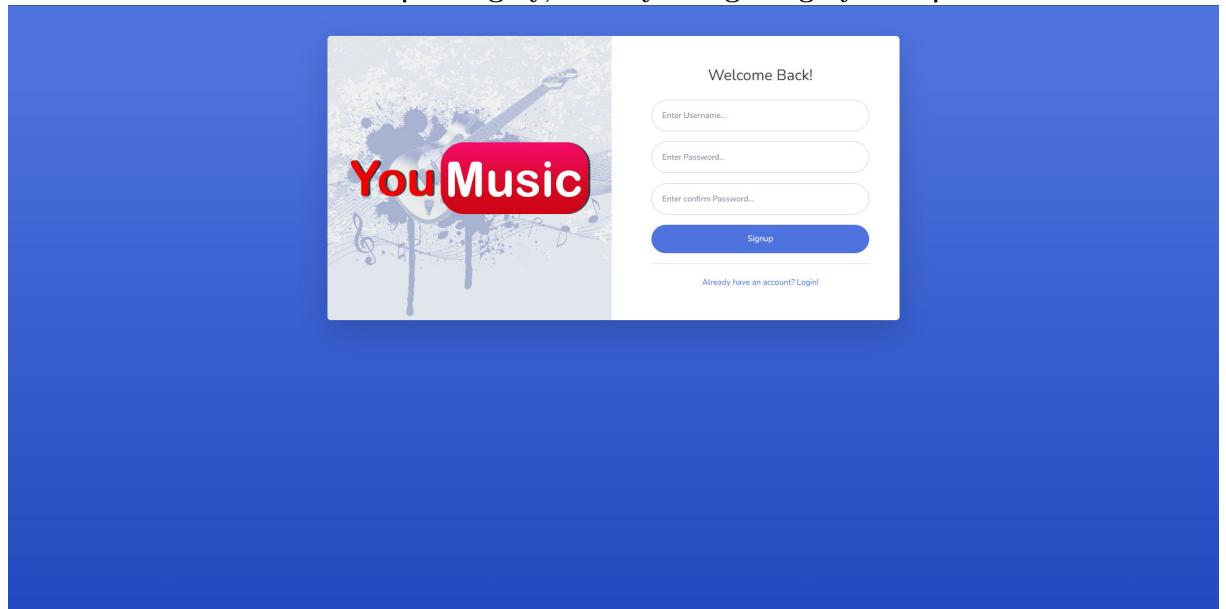
- Trang web cho phép người dùng đăng nhập, đăng ký để thực hiện các thao tác lưu bài hát vào danh sách yêu thích, danh sách lịch sử nghe
- Người dùng nhấn nút đăng nhập ở trang chủ



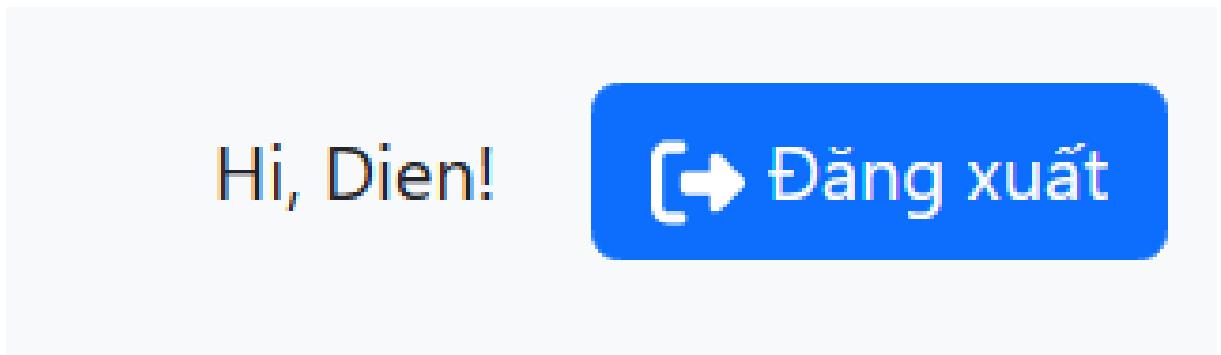
- Trang đăng nhập



- Nếu chưa có tài khoản thì chọn đăng ký, lúc này trang đăng ký sẽ hiện ra



- Nếu đăng nhập thành công, người dùng sẽ trở về trang chủ



- Code xử lý:

Chuyển hướng trang Đăng nhập

```
@app.route('/Login_page')
def login_page():
    return render_template('login.html')
```

Xử lý chức năng đăng nhập



```
@app.route('/login', methods=['POST'])
def login():
    username = request.form['username']
    password = request.form['password']

    mycursor = mydb.cursor()
    sql = "SELECT * FROM user WHERE name = %s AND password = %s"
    val = (username, password)
    mycursor.execute(sql, val)
    result = mycursor.fetchone()

    if result:
        session['loggedin'] = True
        session['username'] = username
        return redirect(url_for(next_page_token))
    else:
        flash('Tên đăng nhập hoặc mật khẩu không đúng!', 'error')
        return redirect(url_for('login_page'))
```

Trạng thái đăng nhập của trang chủ

```
@app.route('/')
def index():
    global next_page_token
    next_page_token = 'index'
    if 'loggedin' in session:
        username = session['username']
        return render_template('index.html', username=username)
    else:
        return render_template('index.html', username=None)
```

Chuyển hướng trang Đăng ký



```
@app.route('/signup_page')
def signup_page():
    return render_template_with_session('signup.html')
```

Xử lý chức năng đăng ký

```
@app.route('/signup', methods=['POST'])
def signup():
    username = request.form['username']
    password = request.form['password']
    confirm_password = request.form['confirm-password']

    # Kiểm tra độ dài của mật khẩu
    if len(password) < 6:
        flash('Mật khẩu phải chứa ít nhất 6 ký tự!', 'error')
        return redirect(url_for('signup_page'))

    # Kiểm tra xem mật khẩu và xác nhận mật khẩu có khớp nhau không
    if password != confirm_password:
        flash('Mật khẩu không khớp!', 'error')
        return redirect(url_for('signup_page'))
```

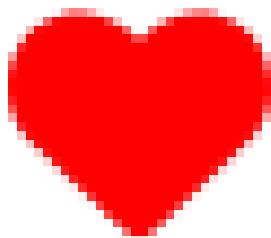
```
# Kiểm tra xem tên người dùng đã tồn tại trong cơ sở dữ liệu chưa
mycursor = mydb.cursor()
sql = "SELECT * FROM user WHERE name = %s"
val = (username,)
mycursor.execute(sql, val)
existing_user = mycursor.fetchone()
if existing_user:
    flash('Tên người dùng đã tồn tại, vui lòng chọn một tên khác!', 'error')
    return redirect(url_for('signup_page'))

# Nếu không có người dùng tồn tại và mật khẩu hợp lệ, thực hiện thêm người dùng mới vào cơ sở dữ liệu
insert_sql = "INSERT INTO user (name, password) VALUES (%s, %s)"
insert_val = (username, password)
mycursor.execute(insert_sql, insert_val)
mydb.commit()

# Đăng ký thành công, chuyển hướng người dùng đến trang đăng nhập
flash('Đăng ký thành công! Vui lòng đăng nhập để tiếp tục.', 'success')
return redirect(url_for('login_page'))
```

3.7 Chức năng lưu bài hát vào danh sách yêu thích

- Người dùng sau khi đăng nhập có thể lưu bài hát vào danh sách yêu thích bằng cách chọn một bài hát để phát sau đó nhấn vào button yêu thích trên thanh điều khiển, bài hát được yêu thích thì button yêu thích có màu đỏ, hủy yêu thích bằng cách nhấn thêm một lần nữa vào button
- Người dùng nhấn nút button yêu thích sau khi chọn một bài hát





- **Code xử lý:**

Thao tác với database

```
@app.route('/toggle_favorite', methods=['POST'])
def toggle_favorite():
    if 'loggedin' in session:
        username = session['username']
        data = request.json
        song_id = data['songId']

        # Kiểm tra xem bài hát đã được lưu trong danh sách yêu thích của người dùng hay chưa
        mycursor = mydb.cursor()
        sql = "SELECT * FROM favorite WHERE name = %s AND id_song = %s"
        val = (username, song_id)
        mycursor.execute(sql, val)
        result = mycursor.fetchone()

        if result:
            # Nếu đã lưu, xóa bài hát khỏi danh sách yêu thích
            delete_sql = "DELETE FROM favorite WHERE name = %s AND id_song = %s"
            delete_val = (username, song_id)
            mycursor.execute(delete_sql, delete_val)
            mydb.commit()
        else:
            # Nếu chưa lưu, thêm bài hát vào danh sách yêu thích
            insert_sql = "INSERT INTO favorite (name, id_song) VALUES (%s, %s)"
            insert_val = (username, song_id)
            mycursor.execute(insert_sql, insert_val)
            mydb.commit()
    return 'OK', 200
else:
    return 'Unauthorized', 401
```

Check bài hát đã được yêu thích hay chưa

```
@app.route('/check_favorite', methods=['POST'])
def check_favorite():
    if 'loggedin' in session:
        username = session['username']
        data = request.json
        song_id = data['songId']

        # Kiểm tra xem bài hát đã được lưu trong danh sách yêu thích của người dùng hay chưa
        mycursor = mydb.cursor()
        sql = "SELECT * FROM favorite WHERE name = %s AND id_song = %s"
        val = (username, song_id)
        mycursor.execute(sql, val)
        result = mycursor.fetchone()

        if result:
            # Nếu đã lưu, trả về kết quả là True
            return jsonify({'isFavorite': True}), 200
        else:
            # Nếu chưa lưu, trả về kết quả là False
            return jsonify({'isFavorite': False}), 200
    else:
        # Trả về mã lỗi 401 nếu người dùng chưa đăng nhập
        return 'Unauthorized', 401
```

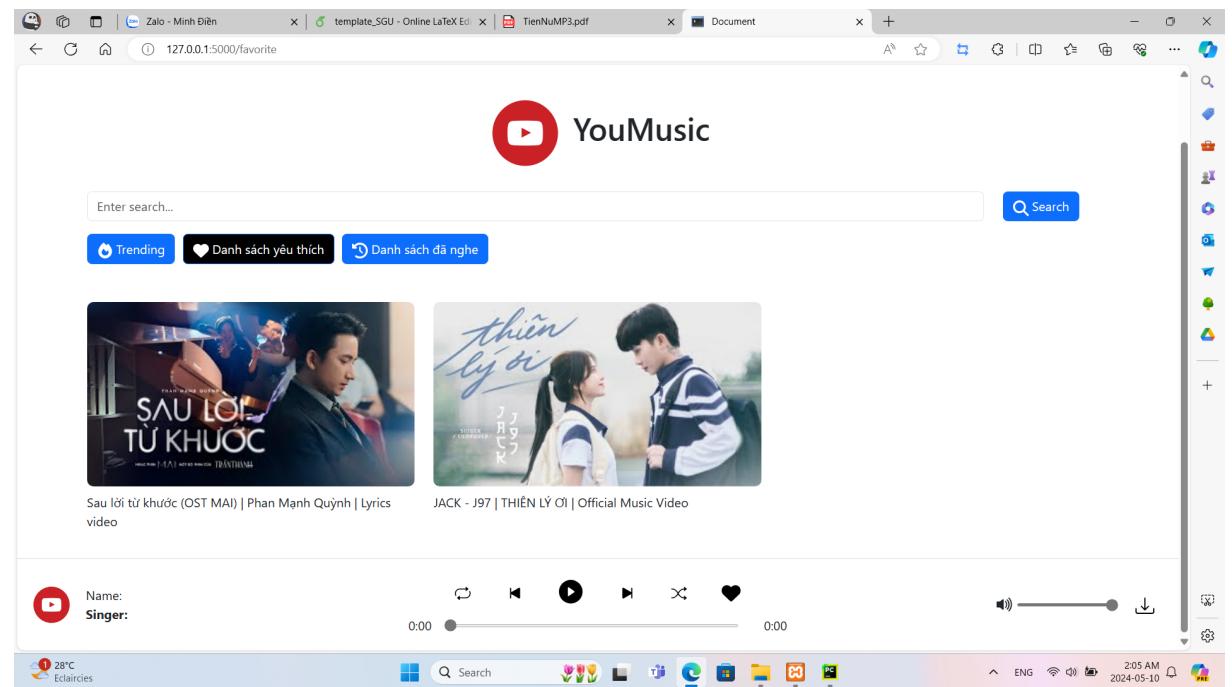
3.8 Chức năng Xem danh sách bài hát yêu thích

- Người dùng sau khi đã đăng nhập, chọn button Danh sách yêu thích trên trang chủ



Danh sách yêu thích

- người dùng được chuyển hướng đến danh sách yêu thích



- **Code xử lý:**

```
@app.route('/favorite', methods=['GET', 'POST'])
def favorite():
    global next_page_token
    next_page_token = 'favorite'
    if 'loggedin' in session:
        username = session['username']
        mycursor = mydb.cursor()
        sql = "SELECT id_song FROM favorite WHERE name = %s"
        val = (username,)
        mycursor.execute(sql, val)
        result = mycursor.fetchall()
        songs = []

        for row in result:
            song_id = row[0] # Lấy giá trị của cột id_song từ kết quả truy vấn
            song = getInfoSong(song_id) # Truyền giá trị của cột id_song vào hàm getInfoSong
            songs.append(song)
    return render_template_with_session('index.html', songs=songs)
else:
    return redirect(url_for('login_page'))
```



3.9 Chức năng Xem danh sách bài hát đã nghe

- Người dùng sau khi đã đăng nhập, chọn button Danh sách đã nghe trên trang chủ

Danh sách đã nghe

- người dùng được chuyển hướng đến danh sách đã nghe

- người dùng có thể xóa tất cả bài hát đã nghe bằng nút xóa tất cả

Xóa tất cả

- Code xử lý:

Thao tác với cơ sở dữ liệu



```
@app.route('/save_played_song', methods=['POST'])
def save_played_song():
    if 'loggedin' not in session:
        return 'Unauthorized', 401

    user = session['username']
    song_id = request.form['songId']

    mycursor = mydb.cursor()

    # Xóa bản ghi cũ nếu đã tồn tại
    delete_sql = "DELETE FROM history WHERE name = %s AND id_song = %s"
    delete_val = (user, song_id)
    mycursor.execute(delete_sql, delete_val)

    # Thêm bản ghi mới
    insert_sql = "INSERT INTO history (name, id_song) VALUES (%s, %s)"
    insert_val = (user, song_id)
    mycursor.execute(insert_sql, insert_val)

    mydb.commit()

    return 'OK', 200
```

Lấy danh sách đã nghe



```
@app.route('/history', methods=['GET', 'POST'])
def history():
    global next_page_token
    next_page_token = 'history'
    if 'loggedin' in session:
        username = session['username']
        mycursor = mydb.cursor()
        sql = "SELECT id_song FROM history WHERE name = %s"
        val = (username,)
        mycursor.execute(sql, val)
        result = mycursor.fetchall()
        songs = []

        for row in result:
            song_id = row[0]
            song = getInfoSong(song_id)
            songs.append(song)

    return render_template_with_session('index.html', songs=songs)
else:
    return redirect(url_for('login_page'))
```

Xóa danh sách đã nghe



```
@app.route('/delete_history', methods=['POST', 'GET'])
def delete_history():
    if 'loggedin' in session:
        username = session['username']
        mycursor = mydb.cursor()
        sql = "DELETE FROM history WHERE name = %s"
        val = (username,)
        mycursor.execute(sql, val)
        mydb.commit()
        return redirect(url_for('history'))
```

4 Tổng kết

4.1 Ưu điểm

- Dễ dàng sử dụng: Trang web cho người dùng thao tác trên giao diện thân thiện, dễ dàng tương tác với các chức năng.
- Dữ liệu kho bài hát lớn từ Youtube.
- Chức năng đa dạng:
 - Có khá đầy đủ những chức năng cơ bản của nhu cầu nghe nhạc như: play nhạc, chuyển bài, lặp lại, tự động chuyển, tăng giảm âm lượng.
 - Chức năng tải nhạc.
 - Chức đăng nhập, đăng ký.
 - Chức năng lưu bài hát vào danh sách yêu thích và xem danh sách bài hát đã xem.

4.2 Nhược điểm

- Tốc độ phản hồi khá chậm khi thực hiện các thao tác như tìm kiếm, xem danh sách.
- Không thể nghe và tải một số bài hát do bản quyền.
- Chức năng đăng nhập, đăng ký còn đơn giản.

4.3 Hướng phát triển

- Cải thiện tốc độ phản hồi của trang web để trải nghiệm của người dùng trở nên tốt hơn.
- Tìm hiểu thêm nhiều chức năng mới để phục vụ nhu cầu của người dùng.



Tài liệu

- [1] Flask Tutorial “link: <https://www.geeksforgeeks.org/flask-tutorial/>”, *Flask Tutorial*, lần truy cập cuối: 15/04/2024.
- [2] Pytube | Python library to download youtube videos “link: <https://www.geeksforgeeks.org/pytube-python-library-download-youtube-videos/>”, *Pytube / Python library to download youtube videos*, lần truy cập cuối: 15/04/2024.
- [3] How to Get a YouTube API Key [Tutorial + Examples] “link: <https://blog.hubspot.com/website/how-to-get-youtube-api-key>”, *How to Get a YouTube API Key [Tutorial + Examples]*, lần truy cập cuối: 15/04/2024.