

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN
MÔN: PHÁT TRIỂN PHẦN MỀM MÃ
NGUỒN MỞ

ĐỀ TÀI

XÂY DỰNG TRANG WEB CHO PHÉP
DOWNLOAD FILE TỪ VIDEO YOUTUBE

GV hướng dẫn: TÙ LÃNG PHIÊU

Nhóm sinh viên thực hiện:

<i>Họ và tên</i>	<i>MSSV</i>
Nguyễn Ngọc Sang	3121410417
Nguyễn Ngọc Sơn	3121410425

Thành phố Hồ Chí Minh, 2024

Mục lục

1 Giới thiệu	2
1.1 Mục tiêu của đề tài	2
1.2 Phạm vi nghiên cứu	2
2 Tổng quan về công nghệ	3
2.1 FrontEnd - ReactJS	3
2.2 BackEnd - FastAPI	3
2.3 Ứng dụng trong đề tài	4
3 Các thư viện được sử dụng trong ngôn ngữ Python	5
3.1 Pytube	5
3.2 Os	5
3.3 Request	6
3.4 Discord	7
3.5 SQLAlchemy	7
4 Phân tích và thiết kế hệ thống	9
4.1 Phân tích yêu cầu	9
4.2 Kiến trúc hệ thống	9
4.2.1 Kiến trúc tổng thể:	9
4.2.2 Entity Relationship Diagram:	9
4.2.3 Relational Schema:	10
4.2.4 Data Flow Diagram:	10
4.2.5 Thiết kế API:	11
4.2.6 Thiết kế giao diện người dùng:	12
4.2.7 Xây dựng chức năng download file	13
5 Demo trang web	18
6 Kết luận và hướng phát triển	20
6.1 Kết luận	20
6.2 Hướng phát triển trong tương lai	20

1 Giới thiệu

1.1 Mục tiêu của đề tài

Trang web cho phép người dùng tìm kiếm các video từ youtube và lựa chọn định dạng, chất lượng để tải xuống, cũng như có thể tải các file âm thanh hoặc file văn bản được cắt ra từ video đó.

Xây dựng trang web hoàn chỉnh với các thư viện ReactJS (TypeScript) để phát triển giao diện người dùng phía frontend và FastAPI (Python) để xây dựng API phía backend.

Thiết kế hệ thống có khả năng mở rộng và bảo trì dễ dàng để có thể phát triển thêm các tính năng mới trong tương lai

1.2 Phạm vi nghiên cứu

Tìm hiểu về ReactJS (TypeScript) cho frontend và FastAPI (Python) cho backend, cùng với các thư viện và công cụ hỗ trợ như axios, SQLAlchemy, và MySQL. Sử dụng nền tảng Discord như 1 cloud để lưu trữ video và các file audio.

Phạm vi nghiên cứu chỉ bao gồm trong những nội dung sau:

- Đề tài chỉ tập trung vào ứng dụng web và sẽ không phát triển các ứng dụng trên các nền tảng khác.
- Trang web chỉ hỗ trợ tải các video từ Youtube và hiện tại sẽ không tích hợp với các nền tảng khác ngoài Youtube.
- Trang web sẽ tập trung vào chức năng tải video và tạm thời sẽ chưa đi sâu vào việc phân tích hành vi của người dùng.

2 Tổng quan về công nghệ

2.1 FrontEnd - ReactJS

ReactJS là một thư viện JavaScript phổ biến được sử dụng để xây dựng giao diện người dùng cho các ứng dụng web hiệu quả và linh hoạt. Được phát triển bởi Facebook, ReactJS đã trở thành một trong những công cụ phổ biến nhất trong cộng đồng phát triển web.

Đặc điểm nổi bật:

- Sử dụng mô hình component-based, cho phép phân chia giao diện người dùng thành các thành phần độc lập, dễ dàng quản lý và tái sử dụng.
- Sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng web, giúp cập nhật giao diện người dùng một cách nhanh chóng mà không cần render lại toàn bộ trang.
- Thực hiện one-way data binding, giúp quản lý trạng thái ứng dụng một cách hiệu quả và tránh các vấn đề về quản lý trạng thái.
- Có một cộng đồng lớn và mạnh mẽ, đi kèm với nhiều thư viện và công cụ hỗ trợ như Redux, React Router, và Material-UI.

Ưu điểm:

- Hiệu suất cao và cập nhật giao diện nhanh chóng.
- Dễ dàng tái sử dụng và bảo trì code.
- Hỗ trợ một loạt các công cụ và thư viện bên thứ ba.

ReactJS được sử dụng rộng rãi trong việc phát triển các ứng dụng web hiện đại, từ các trang web tĩnh đến các ứng dụng web phức tạp và tương tác cao. Các công ty lớn như Facebook, Instagram, và Airbnb đều sử dụng ReactJS cho sản phẩm của mình.

2.2 BackEnd - FastAPI

FastAPI là một framework web siêu nhanh được viết bằng Python, được thiết kế để xây dựng các API hiệu suất cao một cách dễ dàng và nhanh chóng. Được phát triển bởi Sebastián Ramírez, FastAPI kết hợp giữa tính năng động của Python với hiệu suất tối ưu của các ngôn ngữ lập trình như Go và Node.js.

Đặc điểm nổi bật

- Được thiết kế để cung cấp hiệu suất cao, với tốc độ xử lý và đáp ứng gần như là tiệm cận với Go và Node.js.
- Sử dụng type annotations của Python để tạo ra các API có thể kiểm tra kiểu dữ liệu, giúp tăng tính chắc chắn và dễ bảo trì.
- Tự động tạo tài liệu API dựa trên type annotations, giúp dễ dàng hiểu và sử dụng các endpoint API.

- Hỗ trợ async/await, cho phép xử lý các yêu cầu không đồng bộ một cách hiệu quả và linh hoạt.

Ưu điểm:

- Tốc độ xử lý và đáp ứng cao.
- Dễ dàng sử dụng và tài liệu rõ ràng.
- Hỗ trợ type annotations và async/await, giúp code dễ đọc và bảo trì.

FastAPI phù hợp cho việc phát triển các microservices, các ứng dụng web yêu cầu hiệu suất cao, các hệ thống xử lý dữ liệu lớn và các dịch vụ API yêu cầu bảo mật và xác thực tốt. Được sử dụng bởi các công ty như Microsoft, Netflix, và Uber.

2.3 Ứng dụng trong đề tài

Với đề tài xây dựng website cho phép download các video từ YouTube, ReactJS sẽ chịu trách nhiệm hiển thị giao diện người dùng, tìm kiếm và hiển thị các video. FastAPI sẽ xử lý các yêu cầu từ phía client, tương tác với YouTube API và quản lý việc tải xuống video, tách video thành các file định dạng theo yêu cầu và lưu trữ chúng, đảm bảo an toàn và hiệu quả.

3 Các thư viện được sử dụng trong ngôn ngữ Python

3.1 Pytube

Pytube là một thư viện Python nhẹ và không phụ thuộc vào bên thứ ba, được thiết kế để tải video từ YouTube. Nó cung cấp một cách tiếp cận đơn giản nhưng linh hoạt, cho phép người dùng dễ dàng tải các video. Pytube cũng hỗ trợ việc tải danh sách phát đầy đủ và có khả năng xuất các đường dẫn phụ đề dưới dạng .srt (SubRip Subtitle).

Ưu điểm:

- Không cần phụ thuộc bên thứ ba: Điều này làm cho Pytube trở nên dễ cài đặt và ít gặp vấn đề về tương thích.
- Giao diện dòng lệnh: Cho phép tải video ngay từ terminal.
- Hỗ trợ nhiều dạng stream: Tải được cả video dạng progressive và DASH.
- Tài liệu hướng dẫn chi tiết: Giúp người mới bắt đầu có thể dễ dàng sử dụng.

Nhược điểm:

- Hiệu suất: Vì là một thư viện Python, Pytube có thể không nhanh bằng các công cụ tải video được viết bằng ngôn ngữ biên dịch như C hoặc Java.

3.2 Os

Module os trong Python là một thư viện chuẩn cung cấp các hàm cho phép tương tác với hệ điều hành. Nó được sử dụng để xử lý các thao tác liên quan đến tệp và thư mục như tạo, xóa và thay đổi thư mục, cũng như lấy nội dung của chúng. Ngoài ra, nó còn cho phép bạn giao tiếp với hệ điều hành cơ bản theo nhiều cách khác như thực thi các lệnh shell, xử lý các file descriptor, và hơn thế nữa.

Ưu điểm:

- Tính Di động: Module os cung cấp cách thức để sử dụng các chức năng phụ thuộc vào hệ điều hành một cách di động trên các hệ điều hành khác nhau. Các script Python sử dụng module os thường có thể chạy trên Windows, macOS và Linux mà không cần sửa đổi.
- Toàn diện: Nó cung cấp một bộ các hàm phong phú che giấu nhiều phức tạp của các thao tác cấp hệ thống như xử lý tệp, quản lý thư mục và quản lý quá trình.
- Giao diện Trực tiếp với OS: Module này cung cấp nhiều tiện ích để tương tác trực tiếp với hệ điều hành, từ các thao tác đường dẫn đơn giản đến việc thực thi các lệnh trong shell.
- Dễ Sử dụng: Đối với hầu hết người dùng, đặc biệt là những người quen thuộc với các khái niệm lập trình cơ bản, module os là đơn giản và trực quan. Nó đơn giản hóa nhiều thao tác như thao tác đường dẫn thông qua các hàm như os.path.join, os.makedirs, v.v.

- Tích hợp tốt với Các Module Python Khác: Module os hoạt động tốt với các module Python khác như os.path, shutil, và subprocess cho việc xử lý cấp hệ thống toàn diện hơn.

Nhược điểm:

- Không Thực thi JavaScript: os chỉ phân tích cú pháp nội dung tĩnh và không thể thực thi JavaScript. Điều này có nghĩa là nó không thể truy cập nội dung được tạo hoặc thay đổi bởi JavaScript sau khi trang được tải.
- Tốc độ Phân tích Cú pháp Có Thể Chậm: Tùy thuộc vào trình phân tích cú pháp bạn chọn, os có thể không nhanh bằng một số thư viện khác như lxml khi chỉ sử dụng nó mà không thông qua BeautifulSoup.
- Phụ thuộc vào Bên Ngoài: Đối với hiệu suất tối ưu, os cần các trình phân tích cú pháp bên ngoài như lxml hoặc html5lib, có thể yêu cầu cài đặt thêm và quản lý các phụ thuộc.
- Khả năng Nâng cao Bị Hạn chế: Trong khi os tuyệt vời cho các tác vụ phân tích cú pháp cơ bản và trung bình, nó có thể không phù hợp cho các tác vụ phân tích cú pháp rất phức tạp hoặc cần tối ưu hóa cao.

3.3 Request

Requests là một thư viện Python rất phổ biến dùng để gửi các yêu cầu HTTP một cách dễ dàng. Được thiết kế để làm cho các yêu cầu HTTP trở nên đơn giản và trực quan hơn, requests cung cấp một giao diện lập trình ứng dụng (API) rõ ràng và dễ sử dụng.

Ưu điểm:

- Dễ sử dụng: Giao diện của requests rất thân thiện và dễ hiểu, giúp việc gửi các yêu cầu HTTP trở nên dễ dàng hơn nhiều so với urllib - thư viện chuẩn của Python.
- Tự động quản lý phiên: requests có thể tự động xử lý cookie và duy trì phiên, giúp cho việc tương tác với các website yêu cầu xác thực trở nên đơn giản hơn.
- Hỗ trợ nhiều loại xác thực: Thư viện này hỗ trợ các loại xác thực HTTP phổ biến, bao gồm Basic, Digest, và OAuth.
- Hỗ trợ cho các phương thức HTTP: requests hỗ trợ tất cả các phương thức HTTP như GET, POST, PUT, DELETE, HEAD, và OPTIONS một cách trực quan.
- An toàn hơn: requests bao gồm các tính năng để giúp làm cho các yêu cầu của bạn an toàn hơn như xác minh SSL mặc định và tránh các lỗ hổng bảo mật.

Nhược điểm:

- Không hỗ trợ bất đồng bộ: requests không hỗ trợ bất đồng bộ, điều này có thể làm cho nó không phù hợp với một số ứng dụng hiện đại cần xử lý nhiều yêu cầu cùng lúc mà không chặn I/O. Để xử lý bất đồng bộ, bạn có thể cần sử dụng các thư viện khác như aiohttp.

- Sử dụng tài nguyên: Với các ứng dụng lớn hơn cần xử lý số lượng lớn các yêu cầu mạng, requests có thể không hiệu quả bằng các thư viện dựa trên sự kiện như gevent hoặc Tornado
- Phụ thuộc: requests phụ thuộc vào các gói bên ngoài như urllib3, chardet, idna, và certifi. Trong một số trường hợp, điều này có thể gây ra xung đột hoặc các vấn đề về phụ thuộc khi triển khai.

3.4 Discord

Thư viện Discord.py là một thư viện Python mạnh mẽ và linh hoạt được sử dụng để tạo và tương tác với bot Discord. Nó cung cấp các công cụ cho phép bạn kết nối, điều khiển và tương tác với Discord thông qua bot.

Ưu điểm:

- Discord.py cung cấp một API dễ sử dụng để tạo và quản lý bot Discord. Việc khởi tạo và thiết lập bot Discord có thể được thực hiện một cách dễ dàng.
- Cung cấp một giao diện để tương tác với API Discord, cho phép bạn thực hiện hầu hết các thao tác thông qua bot của mình, bao gồm gửi tin nhắn, thay đổi quyền, tạo kênh, và nhiều hơn nữa.
- Dễ dàng mở rộng bot của mình bằng cách thêm các plugin và module. Discord.py cung cấp các công cụ cho phép bạn tổ chức mã của mình theo cách mà nhà phát triển muốn và tạo ra một bot mạnh mẽ với các tính năng phức tạp.
- Có một cộng đồng lớn và tích cực trên Discord và các diễn đàn trực tuyến khác. Bạn có thể tìm kiếm sự giúp đỡ và tài nguyên từ cộng đồng này khi cần thiết.

3.5 SQLAlchemy

SQLAlchemy là một thư viện mã nguồn mở dành cho Python, cung cấp một công cụ mạnh mẽ và linh hoạt cho làm việc với cơ sở dữ liệu. Được biết đến như một ORM (Object-Relational Mapping), SQLAlchemy cho phép các nhà phát triển ánh xạ các lớp Python tới các bảng cơ sở dữ liệu và ngược lại, giúp việc tương tác với cơ sở dữ liệu trở nên dễ dàng và hiệu quả hơn.

SQLAlchemy không chỉ dừng lại ở việc đơn giản hóa các thao tác CRUD (Create, Read, Update, Delete), mà còn cung cấp một hệ thống trừu tượng hóa cho phép các nhà phát triển xây dựng và quản lý các mối quan hệ phức tạp giữa các bảng một cách trực quan và hiệu quả. Thư viện này hỗ trợ nhiều loại cơ sở dữ liệu, bao gồm nhưng không giới hạn ở PostgreSQL, MySQL, SQLite, Oracle và Microsoft SQL Server.

Ưu điểm:

- cung cấp một hệ thống ORM mạnh mẽ giúp ánh xạ các lớp Python tới các bảng trong cơ sở dữ liệu, cho phép thao tác với cơ sở dữ liệu một cách tự nhiên thông qua các đối tượng Python.
- Không ép buộc các nhà phát triển vào một mô hình duy nhất. Nó cho phép chọn giữa hai cách tiếp cận chính: ORM hoặc Core, mỗi cách tiếp cận có những ưu điểm riêng và phù hợp với các nhu cầu khác nhau.

- hỗ trợ nhiều loại cơ sở dữ liệu khác nhau, bao gồm PostgreSQL, MySQL, SQLite, Oracle, và Microsoft SQL Server, làm cho nó trở thành một công cụ linh hoạt cho các dự án đa nền tảng.
- Cho phép xây dựng các truy vấn SQL phức tạp một cách dễ dàng và trực quan, với sự hỗ trợ đầy đủ cho các mối quan hệ, kết nối và các hàm tổng hợp đồng thời cung cấp một hệ thống quản lý phiên mạnh mẽ, giúp quản lý các giao dịch cơ sở dữ liệu một cách hiệu quả và an toàn.
- Thư viện này hỗ trợ đầy đủ các tính năng của SQL, bao gồm các mệnh đề SELECT, INSERT, UPDATE, DELETE, cũng như các tính năng nâng cao như JOIN, SUBQUERY, và các hàm tổng hợp.
- Có một cộng đồng lớn và tích cực, cùng với tài liệu phong phú và chi tiết, giúp các nhà phát triển dễ dàng học tập và sử dụng thư viện

4 Phân tích và thiết kế hệ thống

4.1 Phân tích yêu cầu

Yêu cầu chức năng

- Người dùng có thể tìm kiếm video trên Youtube thông qua đường dẫn url.
- Người dùng có thể xem kết quả tìm kiếm sau khi nhập url.
- Người dùng có thể chọn định dạng (MP4, MP3, v.v.) và chất lượng (1080p, 720p, v.v.), hoặc file text chức phụ đề (nếu có) của video muốn tải về.

Yêu cầu phi chức năng

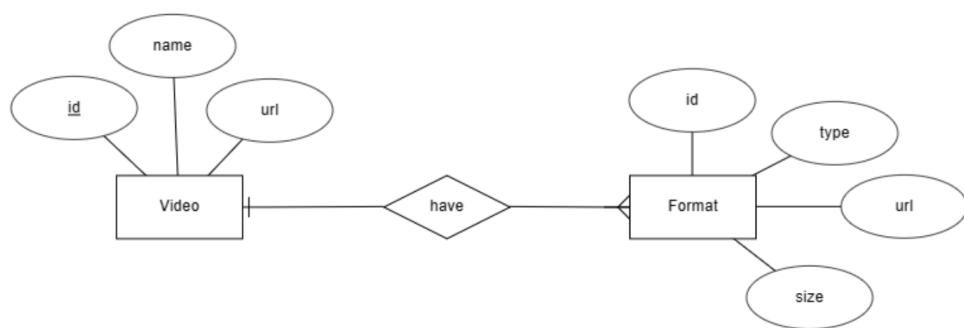
- Ứng dụng phải có giao diện đẹp, dễ sử dụng.
- Hiệu suất cao và thời gian phản hồi nhanh.

4.2 Kiến trúc hệ thống

4.2.1 Kiến trúc tổng thể:

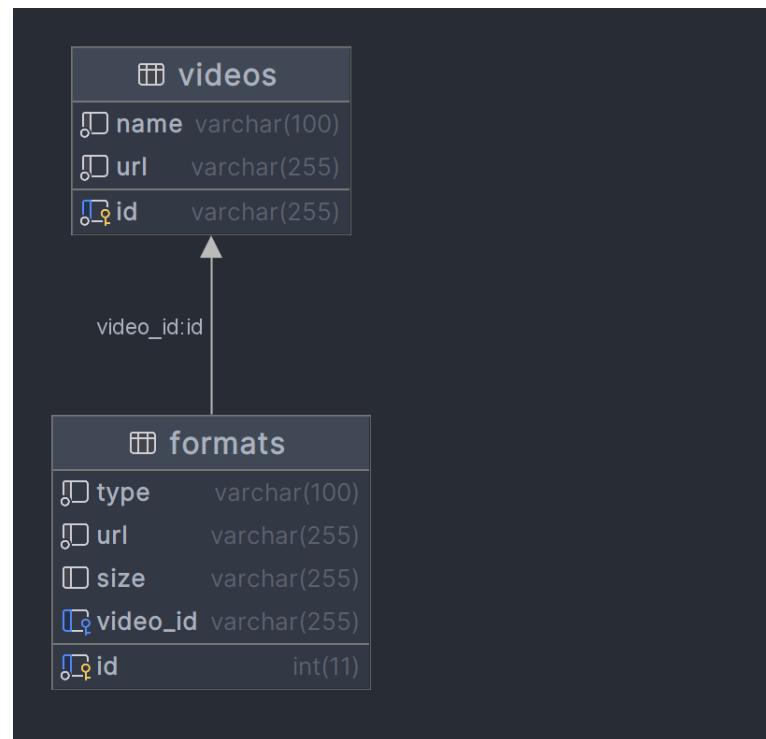
- Frontend: Sử dụng ReactJS để xây dựng giao diện người dùng. Frontend sẽ tương tác với backend thông qua các API.
- Backend: Sử dụng FastAPI để xây dựng các API phục vụ cho các chức năng tìm kiếm và tải video cũng như việc lưu video lên cloud sau đó ghi lại đường dẫn vào cơ sở dữ liệu.
- Database: Sử dụng MySQL để lưu trữ thông tin video đã tải xuống.

4.2.2 Entity Relationship Diagram:



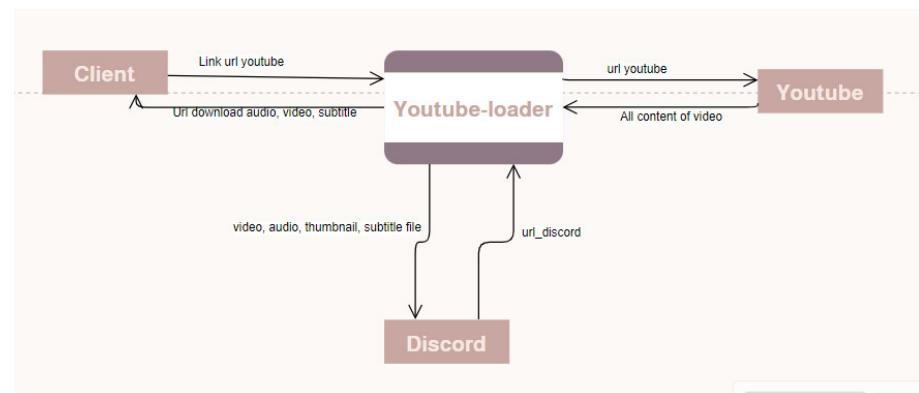
Hình 1: Mô hình thực thể hợp (ERD)

4.2.3 Relational Schema:

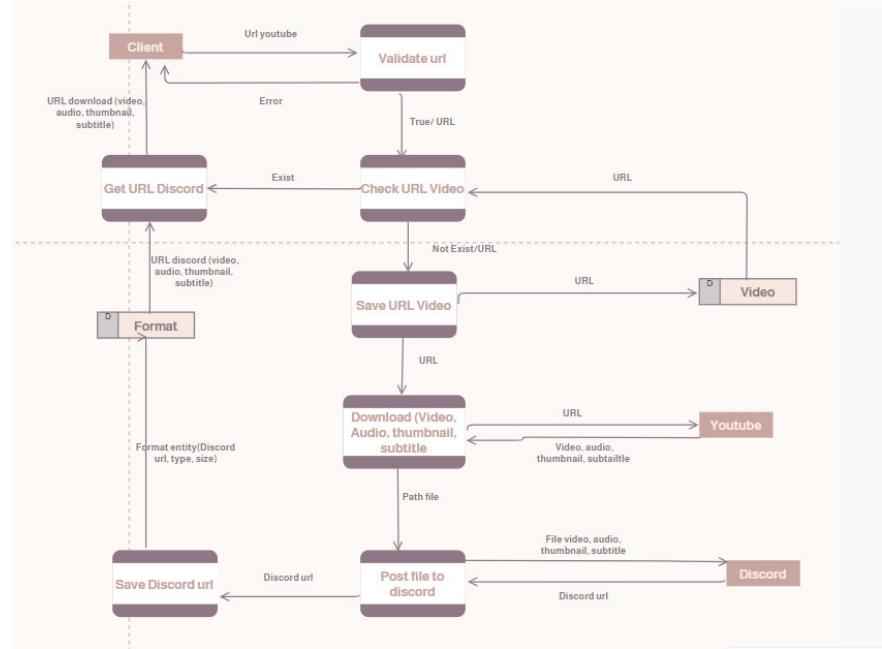


Hình 2: Lược đồ quan hệ (RS)

4.2.4 Data Flow Diagram:



Hình 3: Sơ đồ luồng dữ liệu cấp 0



Hình 4: Sơ đồ luồng dữ liệu cấp 1

4.2.5 Thiết kế API:

API tìm kiếm video trên youtube:

- Endpoint API: 'GET /search?url=keyword'
- Response: Trả về đường dẫn các file được tách từ video mà người dùng muốn tải bao gồm file mp4 có các chất lượng khác nhau, file audio mp3 và file text phụ đề của video.

POST http://127.0.0.1:8000/?url=https://www.youtube.com/watch?v=3Vzc59pps20

Body (raw JSON)

```

1 {
2     "id": "3Vzc59pps20",
3     "name": "Hải Long Cuộc Sông - Trí Ký Cảm Xúc Web5ngay",
4     "url": "https://www.youtube.com/watch?v=3Vzc59pps20",
5     "formats": [
6         {
7             "type": "240p",
8             "url": "https://cdn.discordapp.com/attachments/1238716695636475904/1240269589150044273/Hai_Long_Cuoc_Song_-Tri_Ky_Cam_Xuc_Web5ngay.mp4?ex=6645f291&is=6644a111&hm=3b68aa7415261e6fb9ab74276f84eac1f1b260d1ccb4e878a0be2d97be99256a&"
9         },
10        {
11            "type": "144p",
12            "url": "https://cdn.discordapp.com/attachments/1238716695636475904/1240269610524344331/Hai_Long_Cuoc_Song_-Tri_Ky_Cam_Xuc_Web5ngay.mp4?ex=6645f296&is=6644a116&hm=58520eb2570c29b4f60d0fe85536c644c21ac16325c5edcb820c7de80c9dc8d9&"
13        },
14        {
15            "type": "audio",
16            "url": "https://cdn.discordapp.com/attachments/1238716695636475904/1240269634305785918/Hai_Long_Cuoc_Song_-Tri_Ky_Cam_Xuc_Web5ngay.mp4?ex=6645f29b&is=6644a11b&hm=2ffba7a75b6885ca26933f8d07a9c3798f8f170b6fc1d8f628575a29e310367e&"
17        },
18        {
19            "type": "thumbnail",
20            "url": "https://cdn.discordapp.com/attachments/1238716695636475904/1240269639586676797/Hai_Long_C.jpg?ex=6645f29d&is=6644a11d&hm=bc7ae4042aa153e1d4d430a6cbafcaa1344eea5140d787c96f22b8717c8aac7&"
21        }
22    ]
}

```

Hình 5: Gọi và nhận kết quả của API tìm kiếm video

4.2.6 Thiết kế giao diện người dùng:

Video Downloader - SGU

The screenshot shows the 'Video Downloader - SGU' website with a main title 'Youtube Downloader' and a subtitle 'Download video from your devices with just some few clicks'. Below this is a search bar labeled 'Paste link here...' with a placeholder 'Please provide a valid link' and a 'Start Download' button.

Download your video with 3 steps

1. Paste your link
Paste your link for us to start searching for your video, after pasting the link, you can initiate the search by clicking on the 'Start Download' button or pressing Enter.

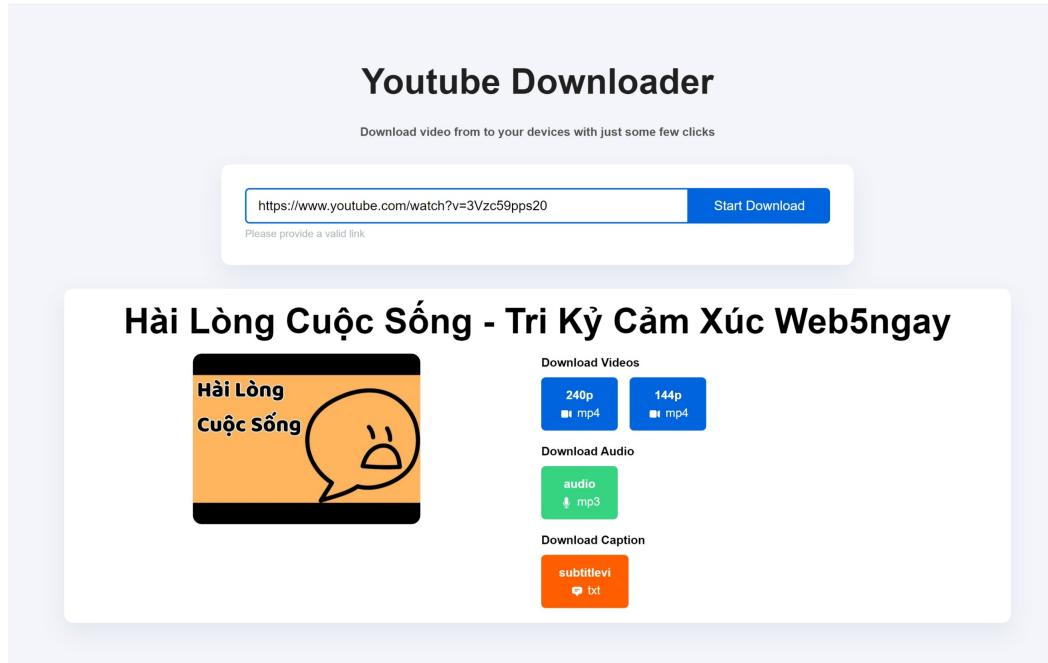
2. Choose your video
Select a format and quality for your video. The format refers to the type of video file you prefer, such as MP4, AVI, or MOV. The quality refers to the resolution of the video, which can range from low (360p) to high definition (720p, 1080p) or even ultra high definition (4K). Once you've made your selections, you can proceed with the next step.

3. Get your video
Click on the 'Download' button to initiate the download process for your video. Please note that the download time may vary depending on the size of the video and your internet speed. Enjoy your video! 😊

Được thực hiện bởi nhóm

Hình 6: Giao diện trang chủ trang web

Video Downloader - SGU



Download your video with 3 steps

- 1. Paste your link**
Paste your link for us to start searching for your video, after pasting the link, you can initiate the search by clicking on the 'Start Download' button or pressing Enter.
- 2. Choose your video**
Select a format and quality for your video. The format refers to the type of video file you prefer, such as MP4, AVI, or MOV. The quality refers to the resolution of the video, which can range from low (360p) to high definition (720p, 1080p) or even ultra high definition (4K). Once you've made your selections, you can proceed with the next step.
- 3. Get your video**
Click on the 'Download' button to initiate the download process for your video. Please note that the download time may vary depending on the size of the video and your internet speed. Enjoy your video! 😊

Được thực hiện bởi nhóm

Hình 7: Giao diện khi người dùng nhập url và tìm kiếm được video

4.2.7 Xây dựng chức năng download file

Nhận yêu cầu từ client

```
34     @app.get("/")
35     @async def root(url: str):
36         result = await youtubeService.download_video(url)
37         return result
```

Hình 8: Hàm sẽ được thực thi khi API được gọi

Bắt đầu xử lý url nhận được:

```

154     async def download_video(self, video_url):
155         parsed_url = urlparse(video_url)
156         # Lấy ra các tham số từ URL
157         query_params = parse_qs(parsed_url.query)
158         # Trả về tham số 'param' nếu tồn tại
159         video_id = query_params.get('v', None)
160         print(video_id)
161         if video_id is None or "www.youtube.com" not in video_url:
162             raise HTTPException(status_code=404, detail="Not Found")
163         else:
164             video_id = video_id[0]
165             video = self.videoRepository.get_video_by_id(video_id)
166             if video:
167                 formats_list = [{type: format.type, url: format.url} for format in video.formats]
168                 formats_dict = {'id': video.id, 'name': video.name, 'url': video.url, 'formats': formats_list}
169                 return formats_dict
170
171             yt = YouTube(video_url)
172             self.videoRepository.add_video(video_id, yt.title, video_url)
173             await self.download_youtube_video(video_url, video_id, yt)
174             await self.download_youtube_audio(video_url, video_id, yt)
175             await self.thumbnail(video_url, video_id, yt)
176             await self.download_subtitle(video_url, video_id)
177
178             get_video = self.videoRepository.get_video_by_id(video_id)
179             formats_list = [{type: format.type, url: format.url} for format in get_video.formats]
180             formats_dict = {'id': get_video.id, 'name': get_video.name, 'url': get_video.url, 'formats': formats_list}
181             return formats_dict

```

Hình 9: Hàm xử lý url nhận được từ client

- Kiểm tra url có phải url của youtube không và có tham số "v" để xác định video không..
- Nếu url đúng định dạng kiểm tra video có tồn tại trong database không.
- Nếu có truy vấn trả về các link download của video, nếu không gọi đến các hàm download thành phần.

Bắt đầu tải các video với nhiều chất lượng khác nhau dưới định dạng mp4:

```

20     async def download_youtube_video(self, url, id, yt):
21         print("Downloading video")
22         try:
23             streams = yt.streams.all()
24             print(streams)
25             url_discords = []
26             for stream in streams:
27                 if stream.mime_type == "video/mp4" and stream.resolution and stream.audio_codec:
28                     # Kiểm tra kích thước của video trước khi tải về
29                     if stream.filesize_mb <= 25: # MAX_FILESIZE là kích thước tối đa cho phép
30                         if not os.path.exists("./downloads"):
31                             os.makedirs("./downloads")
32                         video_path = stream.download(output_path="./downloads")
33                         url_discord = await upload(video_path)
34                         self.formatRepository.add_format(stream.resolution, url_discord, id, stream.filesize_mb)
35                         print("Download complete")
36                         url_discords.append(url_discord)
37                     else:
38                         print(f"Video size exceeds the maximum allowed size: {stream.filesize} bytes")
39             return url_discords
40         except Exception as e:
41             print("error: Không tìm thấy" + e.__str__())
42             return "error: Không tìm thấy" + e.__str__()

```

Hình 10: Kiểm tra và tải các video từ Youtube

- Dùng phương thức all của pytube lấy ra tất cả định dạng của video.
- Dùng if để kiểm tra video có âm thanh và là file mp4.
- Kiểm tra kích thước file dưới 25mb.

- Dùng phương thức download của pytube để download video.
- Gọi hàm upload để up video lên discord và lưu đường dẫn discord vào bản format.

Tải bản audio của video:

```

44     async def download_youtube_audio(self, url, id, yt):
45         try:
46             stream = yt.streams.filter(only_audio=True).first()
47             if not os.path.exists("./downloads"):
48                 os.makedirs("./downloads")
49             video_path = stream.download(output_path="./downloads")
50             url_discord = await upload(video_path)
51             self.formatRepository.add_format(type: "audio", url_discord, id, streamfilesize_mb)
52             return url_discord
53         except Exception as e:
54             return "error: Không tìm thấy" + e.__str__()

```

Hình 11: Tải bảng audio của video từ Youtube

- Dùng phương thức filter của pytube để lọc ra bản audio của video.
- Dùng phương thức download để tải audio về.
- Gọi hàm upload để up video lên discord và lưu đường dẫn discord vào bản format.

Tải hình ảnh thumbnail của video:

```

56     async def thumbnail(self, url, id, yt):
57         try:
58             sax = yt.thumbnail_url
59             file_name = yt.title[:10]
60             if not os.path.exists("./thumbnails"):
61                 os.makedirs("./thumbnails")
62             path = os.path.join("./thumbnails", file_name + '.jpg')
63             print(path)
64             with open(path, 'wb') as f:
65                 f.write(urllib.request.urlopen(sax).read())
66                 f.close()
67             print('Image successfully Downloaded: ', path)
68             url_discord = await upload(os.path.abspath(path))
69             self.formatRepository.add_format(type: "thumbnail", url_discord, id, size: None)
70             time.sleep(6)
71             return url_discord
72         except Exception as e:
73             return "error: Không tìm thấy" + e.__str__()

```

Hình 12: Tải thumbnail của video từ Youtube

- Dùng phương thức thumbnail_url của pytube để lấy đường dẫn đến ảnh thumbnail của youtube.
- Tạo đường dẫn file để tải ảnh về.
- Gọi hàm upload để up video lên discord và lưu đường dẫn discord vào bản format.

Tải phụ đề của video:

```

75     @staticmethod
76     def get_subtitle_tracks(url):
77         with urllib.request.urlopen(url) as response:
78             data = response.read()
79             pos = data.find(b'"playerCaptionsTracklistRenderer"')
80             if pos == -1:
81                 print("No captions found")
82                 return 0
83             pos = data.find(b':', pos)
84             pos += 1
85             # Parse until first JSON error
86             track_list = None
87
88             # ncogson741852963
89             def hook(obj):
90                 nonlocal track_list
91                 track_list = obj
92                 return obj
93
94             try:
95                 json.loads(data[pos:], object_hook=hook)
96             except json.JSONDecodeError:
97                 pass
98
99             return track_list['captionTracks']

```

Hình 13: Tải phụ đề của video từ Youtube

- Sử dụng "urllib.request.urlopen(url)" để mở URL và đọc dữ liệu từ trang web. Dữ liệu được lưu trong biến "data".
- Sử dụng 'data.find(b'"playerCaptionsTracklistRenderer")' để tìm vị trí của chuỗi "playerCaptionsTracklistRenderer" trong dữ liệu. Nếu không tìm thấy, in ra thông báo "No captions found" và trả về 0.
- Tiếp tục tìm kiếm từ vị trí đã tìm thấy ở bước trước để định vị vị trí của dấu hai chấm ":" tiếp theo.
- Sử dụng "json.loads(data[pos:], object_hook=hook)" để phân tích cú pháp JSON từ vị trí đã tìm thấy ở bước trước. Mọi đối tượng JSON được trả về sẽ được chuyển vào hàm "hook" , nơi chúng sẽ được gán cho biến "track_list".
- Trả về track_list['captionTracks'], đây là danh sách các đường dẫn phụ đề từ dữ liệu JSON đã được phân tích cú pháp.

Phân tích dữ liệu trả về thành file text:

```

127     @staticmethod
128     def xml_to_text(xml_string):
129         # Phân tích chuỗi XML
130         root = ET.fromstring(xml_string)
131         # Khởi tạo danh sách trống để lưu trữ giá trị
132         text_list = []
133         # Lặp lại tất cả các thành phần văn bản và nối các giá trị của chúng vào danh sách
134         for text_elem in root.findall('.//text'):
135             text_list.append(text_elem.text)
136         # Nối tất cả các giá trị văn bản thành một chuỗi
137         text_result = '\n'.join(text_list)
138
139         return text_result

```

Hình 14: Tải phụ đề của video từ Youtube

- Dùng fromstring để phân tích chuỗi xml.
- Dùng phương thức findall để tìm tất cả phần tử 'text' trong cây xml và thêm vào mảng.
- Sử dụng phương thức join() để kết hợp tất cả các phần tử trong danh sách text_list thành một chuỗi duy nhất, với mỗi phần tử được phân tách bằng dấu xuống dòng.

Khởi tạo file text (.txt và lưu trữ)

```

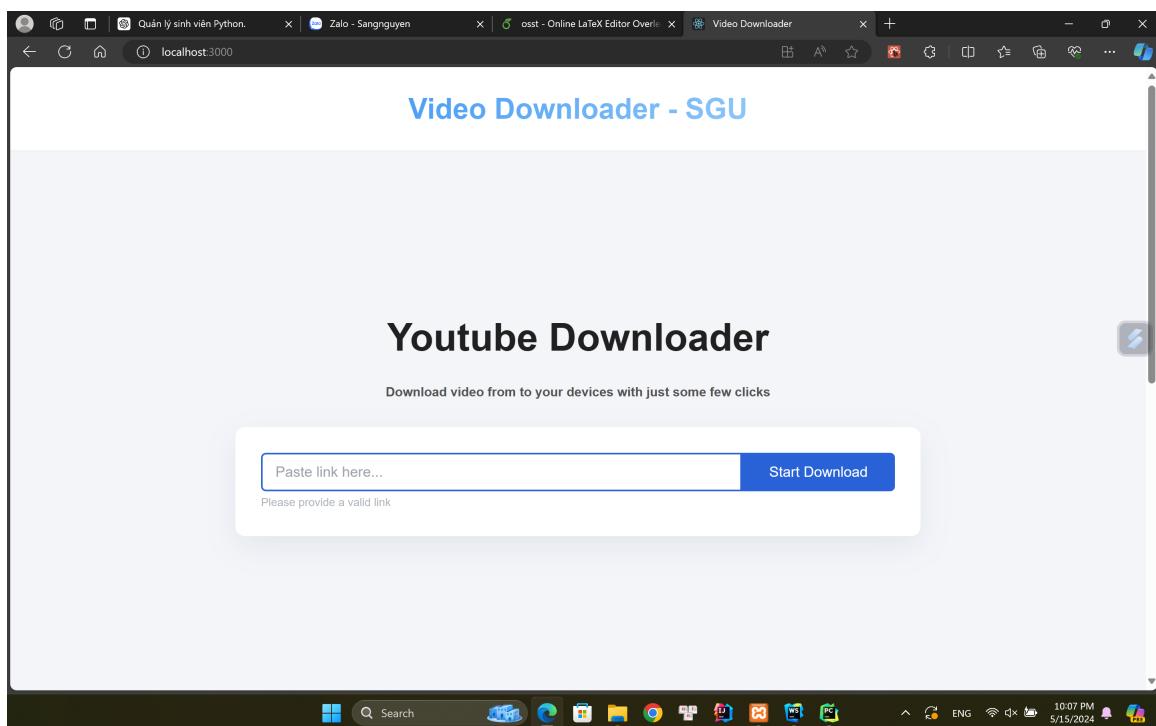
100    v   usage  ~ ngocson741852963
101      async def download_subtitle(self, url, id):
102          tracks = self.get_subtitle_tracks(url)
103
104          if not tracks:
105              print("No", "captions found")
106              return
107
108          # ngocson741852963
109          def isASR(track):
110              return 'kind' in track and track['kind'] == 'asr'
111
112          url_discords = []
113          if not os.path.exists("./captions"):
114              os.makedirs("./captions")
115          for track in tracks:
116              with urllib.request.urlopen(track['baseUrl']) as response:
117                  xml_data = self.xml_to_text(response.read())
118                  filename = urllib.parse.urlsplit(url).query + (
119                      "-asr" if isASR(track) else "") + f"-{track['languageCode']}"+ ".txt"
120                  with open(os.path.join("./captions", filename), 'wb') as f:
121                      f.write(xml_data.encode('utf-8'))
122                      f.close()
123
124          url_discord = await upload(os.path.abspath(os.path.join("./captions", filename)))
125          url_discords.append({'name': track['languageCode'], 'url': url_discord})
126          self.formatRepository.add_format("subtitle" + track['languageCode'], url_discord, id, size: None)
127          print("Written", filename, track['languageCode'], "ASR" if isASR(track) else "")
128
129      return url_discords

```

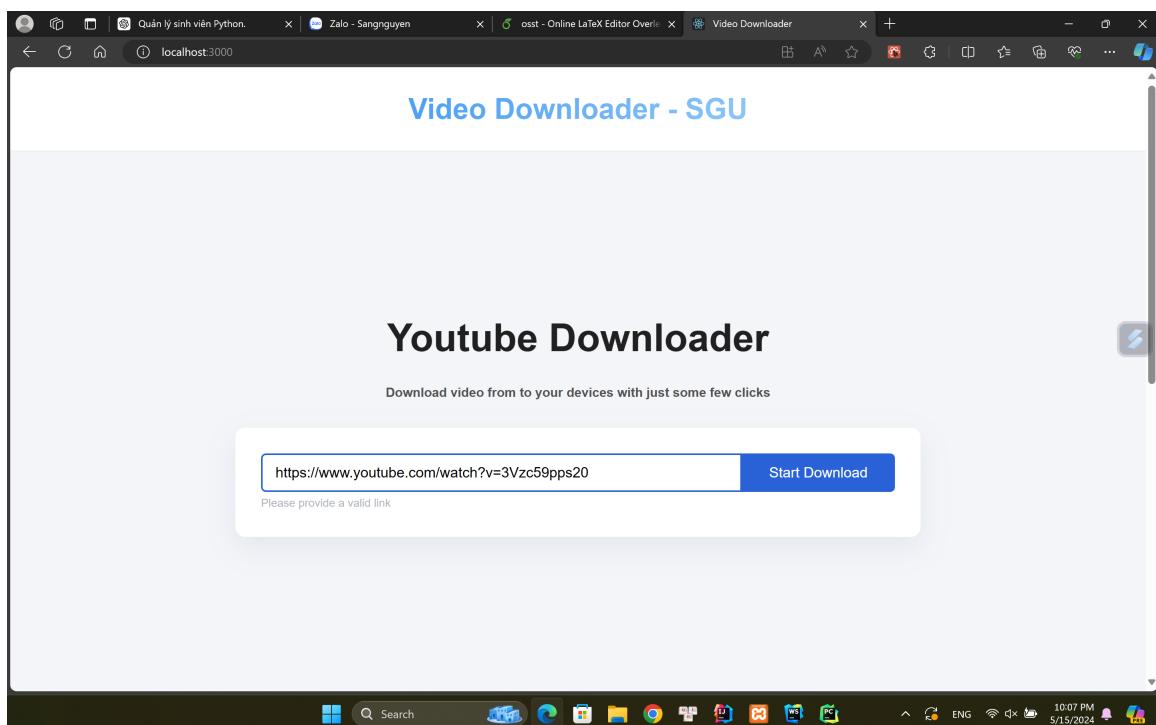
Hình 15: Tạo file text chứa phụ đề của video từ Youtube

- Sử dụng phương thức "get_subtitle_tracks" để lấy danh sách object caption
- Kiểm tra "tracks" có tồn tại không
- Duyệt qua từng đối tượng trong "tracks"
- Sử dụng "urllib.request.urlopen(track['baseUrl'])" để mở url của thuộc tính baseUrl trong track
- Sử dụng phương thức "xml_to_text(response.read())" để chuyển giá trị trả về từ xml sang text
- Dùng 'filename = urllib.parse.urlsplit(url).query + ("-asr" if isASR(track) else "") + f"-track['languageCode']" + ".txt"' để tạo tên file theo id video, languageCode và dùng hàm isASR để kiểm tra xem có phải là văn bản nhận dạng tự động (Automatic Speech Recognition) không.
- Gọi hàm upload để up video lên discord và lưu đường dẫn discord vào bản format.

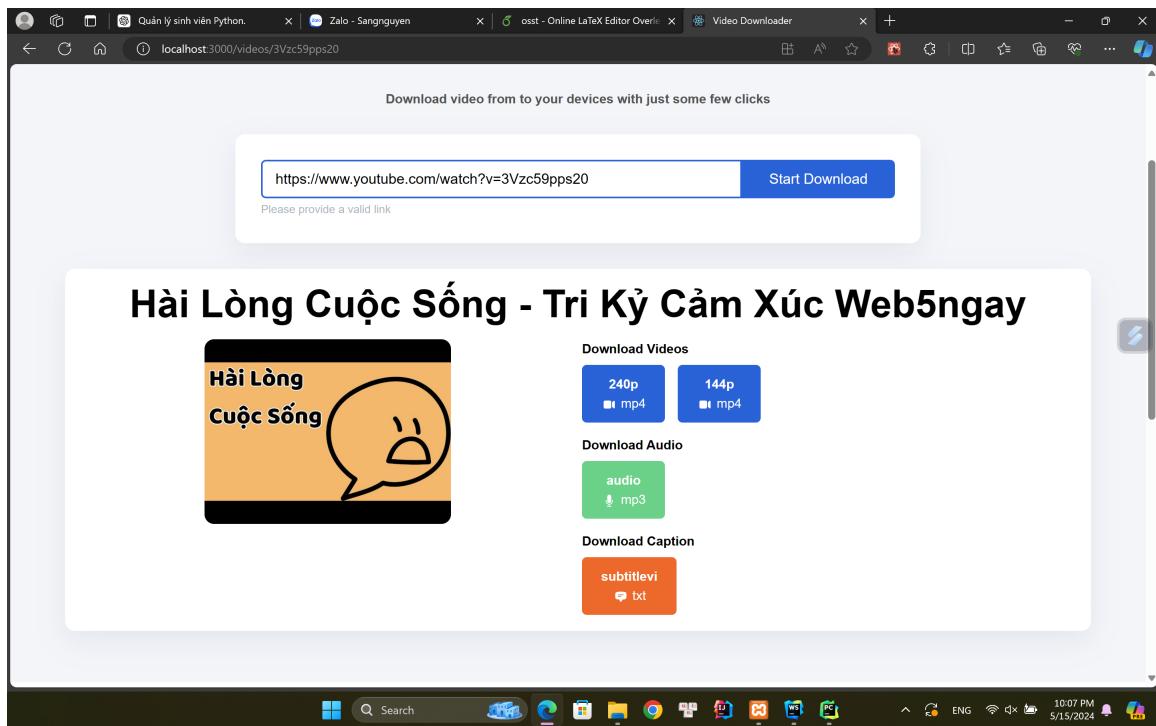
5 Demo trang web



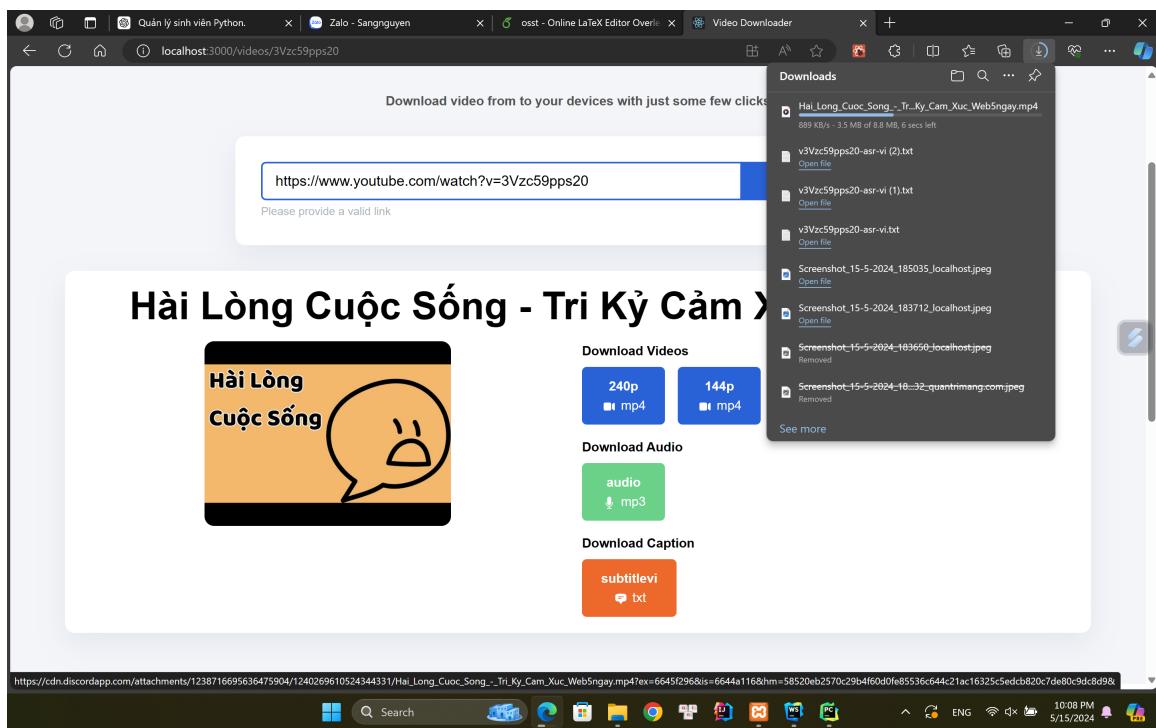
Hình 16: Truy cập trang chủ trang web



Hình 17: Dán đường dẫn đến video vào ô input



Hình 18: Giao diện hiển thị ra các file có thể tải



Hình 19: Án nút và tải về

6 Kết luận và hướng phát triển

6.1 Kết luận

Sau quá trình nghiên cứu và phát triển, đề tài "Xây dựng trang web cho phép download file từ video YouTube" đã đạt được những kết quả quan trọng. Ứng dụng đã được triển khai thành công với các tính năng chính là tìm kiếm và tải xuống các file video, audio từ video YouTube.

Trang web được xây dựng bằng 2 công nghệ đó là ReactJS (frontend) và FastAPI (backend), đây là 2 công nghệ được sử dụng rộng rãi trong lĩnh vực phát triển web. Nhờ khả năng tái sử dụng các thành phần (components), ReactJS giúp việc phát triển và bảo trì giao diện trở nên dễ dàng hơn, trong khi đó FastAPI cung cấp hiệu suất cao và khả năng mở rộng tốt, hỗ trợ tự động sinh tài liệu API, giúp ích trong việc tích hợp và mở rộng hệ thống.

6.2 Hướng phát triển trong tương lai

Nâng cấp tính năng

- Hiện tại, ứng dụng hỗ trợ tải xuống video với các định dạng phổ biến và kích thước nhỏ. Trong tương lai, có thể mở rộng để hỗ trợ nhiều định dạng hơn, nâng cao dung lượng cho phép tải để có thể đáp ứng nhu cầu đa dạng của người dùng.
- Bổ sung tính năng cho phép người dùng tải xuống nhiều video cùng lúc, giúp tiết kiệm thời gian và nâng cao trải nghiệm sử dụng.
- Xây dựng hệ thống lưu trữ và quản lý lịch sử tải xuống của người dùng, giúp người dùng dễ dàng theo dõi và quản lý các video đã tải.

Cải thiện hiệu suất và bảo mật

- Tối ưu hóa hiệu suất: Nâng cấp và tối ưu hóa các thuật toán xử lý video, giảm thời gian tải xuống và tăng tốc độ xử lý.
- Triển khai các biện pháp bảo mật mạnh mẽ để bảo vệ dữ liệu người dùng như mã hóa dữ liệu, xác thực hai yếu tố và quản lý phiên làm việc.
- Xây dựng hệ thống giám sát và kiểm soát truy cập, đảm bảo rằng các hoạt động tải xuống video đều hợp lệ và không vi phạm các quy định của YouTube và các dịch vụ khác.