

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

BÁO CÁO ĐỀ TÀI:

XÂY DỰNG ỨNG DỤNG TẢI VIDEO YOUTUBE

GVHD: Từ Lăng Phiêu

SV: Nguyễn Minh Thắng - 3119650056

Ngô Phú Khang - 3119410180

Phạm Ngọc Đông - 3119410094

| | |
|---|----|
| I. PHẦN GIỚI THIỆU..... | 2 |
| ii. NỘI DUNG | 2 |
| 2.1 Mục tiêu..... | 2 |
| 2.2 Công cụ và thư viện sử dụng..... | 2 |
| 2.2.1 Python..... | 2 |
| 2.2.2 Sử dụng Tkinter..... | 2 |
| 2.2.3 Sử dụng thư viện Pytube | 3 |
| 2.2.4 Sử dụng thư viện google-api-python-client. | 5 |
| 2.2.5 Sử dụng Moviepy | 6 |
| III. Thiết kế ứng dụng..... | 9 |
| 3.1. Chức năng tìm kiếm | 9 |
| 3.2. Chức năng phát video..... | 9 |
| 3.3. Thực tế..... | 10 |
| 3.3.1. Tải video..... | 10 |
| 3.3.2. Phát video..... | 20 |
| 3.4. Kết quả..... | 20 |
| IV. Kết luận | 20 |
| Đề xuất..... | 20 |
| V. Phân chia công việc trong nhóm..... | 21 |
| VI. Tài liệu tham khảo: | 21 |

I. PHẦN GIỚI THIỆU

YouTube là một trong những nền tảng chia sẻ video phổ biến nhất thế giới, cung cấp hàng triệu video về nhiều chủ đề khác nhau. Tuy nhiên, việc tải video trực tiếp từ YouTube không được hỗ trợ trên nền tảng này. Vì vậy, nhu cầu phát triển các công cụ và ứng dụng để tải video từ YouTube ngày càng tăng. Trong bài báo cáo này, chúng tôi trình bày một phương pháp tải video từ YouTube sử dụng ngôn ngữ lập trình Python và các thư viện hỗ trợ như `pytube` và `google-api-python-client`

II. NỘI DUNG

2.1 Mục tiêu

Mục tiêu của đề tài này là xây dựng một ứng dụng desktop đơn giản bằng Python cho phép người dùng tìm kiếm và tải video từ YouTube. Ứng dụng sẽ có các chức năng chính:

- Tìm kiếm video trên YouTube dựa trên từ khóa nhập vào.
- Tải video trên Youtube qua đường dẫn (url)
- Hiển thị kết quả tìm kiếm bao gồm tiêu đề và hình ảnh thu nhỏ (thumbnail) của video
- Cho phép người dùng chọn video và tải về máy tính

2.2 Công cụ và thư viện sử dụng

2.2.1 Python

Ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng.

2.2.2 Sử dụng Tkinter

Dễ Học và Sử Dụng:

Tkinter được tích hợp sẵn trong Python, nên không cần phải cài đặt thêm bất kỳ gói nào.

Giao diện lập trình ứng dụng (API) của Tkinter khá đơn giản và dễ hiểu, điều này làm cho việc học và sử dụng Tkinter trở nên dễ dàng cho cả người mới bắt đầu và người có kinh nghiệm.

Cross-platform:

Ứng dụng được phát triển bằng Tkinter có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux mà không cần sửa đổi code. Điều này làm cho Tkinter trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng đa nền tảng.



Thiết Kế Giao Diện Linh Hoạt:

Tkinter cung cấp một loạt các widget (như buttons, labels, entries, checkboxes, v.v.) và layout managers (như pack, grid, place) cho phép bạn thiết kế giao diện đồ họa theo ý muốn.

Bạn có thể tạo ra các giao diện đẹp và linh hoạt thông qua việc sắp xếp, tự động co giãn và thích ứng với kích thước cửa sổ.

Phát triển Nhanh Chóng:

Với Tkinter, bạn có thể phát triển ứng dụng nhanh chóng do sự đơn giản của API và khả năng tích hợp tốt với Python.

Tkinter cung cấp một loạt các widget và tính năng sẵn có, giúp bạn tiết kiệm thời gian và công sức khi xây dựng giao diện người dùng.

Hỗ Trợ Mạnh Mẽ:

Tkinter có một cộng đồng lớn và tích cực, với nhiều tài liệu, hướng dẫn và ví dụ sẵn có.

Có nhiều diễn đàn và cộng đồng trực tuyến bạn có thể tham gia để hỏi và chia sẻ kinh nghiệm về việc sử dụng Tkinter.

Ứng Dụng Đa Dạng:

Tkinter không chỉ dành cho việc phát triển ứng dụng desktop thông thường mà còn cho các ứng dụng như các công cụ quản lý, ứng dụng ghi chú, ứng dụng giáo dục, v.v.

Tóm lại, Tkinter là một lựa chọn tuyệt vời cho việc phát triển ứng dụng desktop Python nhờ vào sự dễ học, đa nền tảng và tính linh hoạt của nó, cùng với sự hỗ trợ mạnh mẽ từ cộng đồng và tài liệu phong phú.

2.2.3 Sử dụng thư viện Pytube

ƯU ĐIỂM

Dễ Dàng Tải Video từ YouTube:

Pytube cung cấp một cách dễ dàng và tiện lợi để tải video từ YouTube chỉ với một vài dòng mã Python.

Bạn có thể chỉ cần cung cấp URL của video và Pytube sẽ tự động tải video đó về máy tính của bạn.

Hỗ Trợ Tải Video và Âm Thanh:



Pytube không chỉ hỗ trợ tải video từ YouTube mà còn cho phép bạn tải cả âm thanh (audio) của video nếu cần.

Điều này làm cho Pytube trở thành một công cụ linh hoạt cho việc tải nội dung từ YouTube.

Tính Linh Hoạt và Tùy Chỉnh:

Pytube cung cấp nhiều tùy chọn và cấu hình cho quá trình tải video, bao gồm chọn chất lượng video, định dạng file, vị trí lưu trữ, và nhiều hơn nữa.

Bạn có thể tùy chỉnh quá trình tải video theo nhu cầu của mình.

Hỗ Trợ Phong Phú:

Pytube được phát triển và duy trì một cách chuyên nghiệp, với một cộng đồng sử dụng lớn và nhiều nguồn tài liệu hỗ trợ.

Bạn có thể tìm thấy nhiều ví dụ và hướng dẫn trực tuyến để sử dụng Pytube trong các dự án của mình.

Miễn Phí và Mã Nguồn Mở:

Pytube là một thư viện mã nguồn mở, tức là bạn có thể sử dụng miễn phí và thậm chí cải tiến nó nếu cần.

Điều này giúp bạn tiết kiệm chi phí và tận dụng được lợi ích từ sự cộng tác của cộng đồng.

Tóm lại, việc sử dụng Pytube trong Python là một cách tiện lợi và hiệu quả để tải video và âm thanh từ YouTube, với tính linh hoạt và tùy chỉnh cao, cùng với sự hỗ trợ phong phú từ cộng đồng người dùng.

KHUYẾT ĐIỂM

Mặc dù Pytube là một thư viện hữu ích cho việc tải video từ YouTube trong Python, nhưng cũng có một số hạn chế và khuyết điểm:

Phụ Thuộc vào YouTube:

Pytube hoạt động dựa trên cấu trúc của trang web YouTube. Do đó, nếu YouTube thay đổi cấu trúc trang web của họ, có thể làm cho Pytube không hoạt động đúng cách cho đến khi cập nhật lại.

Hạn Chế về Tính Năng:



Pytube tập trung chủ yếu vào việc tải video từ YouTube và không cung cấp các tính năng phức tạp như tương tác với các thành phần khác của YouTube như bình luận, đăng ký, hay quản lý kênh.

Không Hỗ Trợ Tải Video Bảo Mật:

Pytube không hỗ trợ tải video được bảo mật bằng cách mật khẩu hoặc tài khoản Google để truy cập. Nếu video yêu cầu xác thực để xem, Pytube sẽ không thể tải được video đó.

Giới Hạn Về Số Lượng Yêu Cầu:

Có một số giới hạn về số lượng yêu cầu mà bạn có thể gửi đến YouTube trong một khoảng thời gian nhất định. Điều này có thể ảnh hưởng đến khả năng tải video của bạn nếu bạn cần tải một lượng lớn video trong một thời gian ngắn.

Hạn Chế về Tốc Độ Tải:

Tốc độ tải video từ YouTube bằng Pytube có thể chậm hơn so với các phương pháp tải video khác, đặc biệt là đối với các video có dung lượng lớn hoặc có độ phân giải cao.

Mặc dù có những hạn chế như trên, Pytube vẫn là một công cụ hữu ích và tiện lợi cho việc tải video từ YouTube trong Python, đặc biệt là đối với các dự án đơn giản hoặc yêu cầu tải video cơ bản.

2.2.4 Sử dụng thư viện google-api-python-client.

ƯU ĐIỂM:

Dễ sử dụng:

Thư viện google-api-python-client cung cấp một giao diện Python dễ sử dụng cho việc tương tác với các dịch vụ Google API. Các hàm và lớp được thiết kế một cách rõ ràng và dễ hiểu, giúp bạn thực hiện các yêu cầu API một cách dễ dàng.

Tích hợp sẵn với các dịch vụ Google:

Thư viện này cung cấp hỗ trợ cho nhiều dịch vụ Google API khác nhau, bao gồm Google Drive, Google Calendar, Google Sheets, Gmail, YouTube, và nhiều dịch vụ khác. Điều này giúp bạn tương tác với các dịch vụ này thông qua mã Python của mình.

Tài liệu phong phú:



Google cung cấp tài liệu rất chi tiết và phong phú cho thư viện google-api-python-client, bao gồm hướng dẫn, ví dụ và tài liệu API. Điều này giúp bạn hiểu rõ cách sử dụng thư viện và thực hiện các yêu cầu API.

Hỗ trợ cộng đồng tốt:

Thư viện này được sử dụng rộng rãi và có một cộng đồng lớn, nên bạn có thể dễ dàng tìm kiếm câu trả lời cho các vấn đề mà bạn gặp phải thông qua các diễn đàn, stackoverflow và các nguồn tài nguyên khác.

Khuyết điểm:

Khó khăn trong việc cấu hình:

Đôi khi quá trình cấu hình và xác thực với các dịch vụ Google API có thể phức tạp và đòi hỏi nhiều bước. Điều này đặc biệt đúng với các dịch vụ yêu cầu quyền truy cập đặc biệt hoặc quy trình xác thực phức tạp.

Tốc độ truy vấn chậm:

Một số yêu cầu API có thể mất thời gian lâu để hoàn thành, đặc biệt là khi bạn phải truy cập vào dữ liệu lớn hoặc khi dịch vụ API đang gặp vấn đề về hiệu suất.

Giới hạn về quyền truy cập:

Một số dịch vụ Google API yêu cầu quyền truy cập đặc biệt, và bạn có thể cần phải cấp quyền truy cập từ phía người dùng hoặc quản trị viên trước khi bạn có thể sử dụng API.

Tóm lại, google-api-python-client là một công cụ mạnh mẽ cho việc tương tác với các dịch vụ Google API trong Python, nhưng cần phải xem xét kỹ lưỡng các yếu điểm và yêu cầu cấu hình trước khi triển khai.

2.2.5 Sử dụng Moviepy

Sử dụng thư viện MoviePy trong Python mang lại nhiều ưu điểm cho việc xử lý và chỉnh sửa video, nhưng cũng có một số khuyết điểm cần xem xét. Dưới đây là một số điểm cần lưu ý:

Ưu điểm:

Dễ sử dụng:



MoviePy cung cấp một API Python dễ sử dụng và trực quan, giúp bạn dễ dàng tạo, chỉnh sửa và xử lý video trong Python.

Tích hợp với các thư viện khác:

MoviePy tích hợp tốt với các thư viện khác trong hệ sinh thái Python như NumPy, SciPy và Matplotlib, cho phép bạn kết hợp các chức năng của MoviePy với các tính toán và phân tích dữ liệu khác.

Hỗ trợ nhiều định dạng video:

MoviePy hỗ trợ nhiều định dạng video khác nhau bao gồm MP4, AVI, GIF, và nhiều định dạng khác, giúp bạn làm việc với các loại video khác nhau.

Chức năng đa dạng:

MoviePy cung cấp nhiều chức năng để bạn có thể thực hiện các thao tác như cắt, ghép, chỉnh sửa video, thêm hiệu ứng, v.v. Một số chức năng bao gồm cắt, ghép, chuyển đổi định dạng, thêm hiệu ứng video, tạo video từ hình ảnh, v.v.

Tài liệu phong phú:

MoviePy đi kèm với tài liệu chi tiết và ví dụ đa dạng, giúp bạn hiểu rõ cách sử dụng và triển khai các chức năng của nó.

Mã nguồn mở và miễn phí:

MoviePy là một dự án mã nguồn mở, tức là bạn có thể sử dụng nó miễn phí và thậm chí có thể đóng góp vào sự phát triển của nó.

Khuyết điểm:

Hiệu suất không tốt đối với một số thao tác:

Một số thao tác xử lý video có thể yêu cầu nhiều tài nguyên tính toán và có thể không được tối ưu hóa tốt trong MoviePy, dẫn đến hiệu suất không tốt đối với các tập tin video lớn hoặc thời lượng dài.



Khả năng mở rộng hạn chế:

Mặc dù MoviePy cung cấp một số chức năng cơ bản, nhưng không có nhiều tính năng mở rộng hoặc plugin để tùy chỉnh hoặc mở rộng tính năng của nó.

Khó khăn trong việc tương thích với một số định dạng video:

Mặc dù MoviePy hỗ trợ nhiều định dạng video, nhưng vẫn có thể gặp khó khăn khi làm việc với một số định dạng đặc biệt hoặc không tiêu chuẩn.

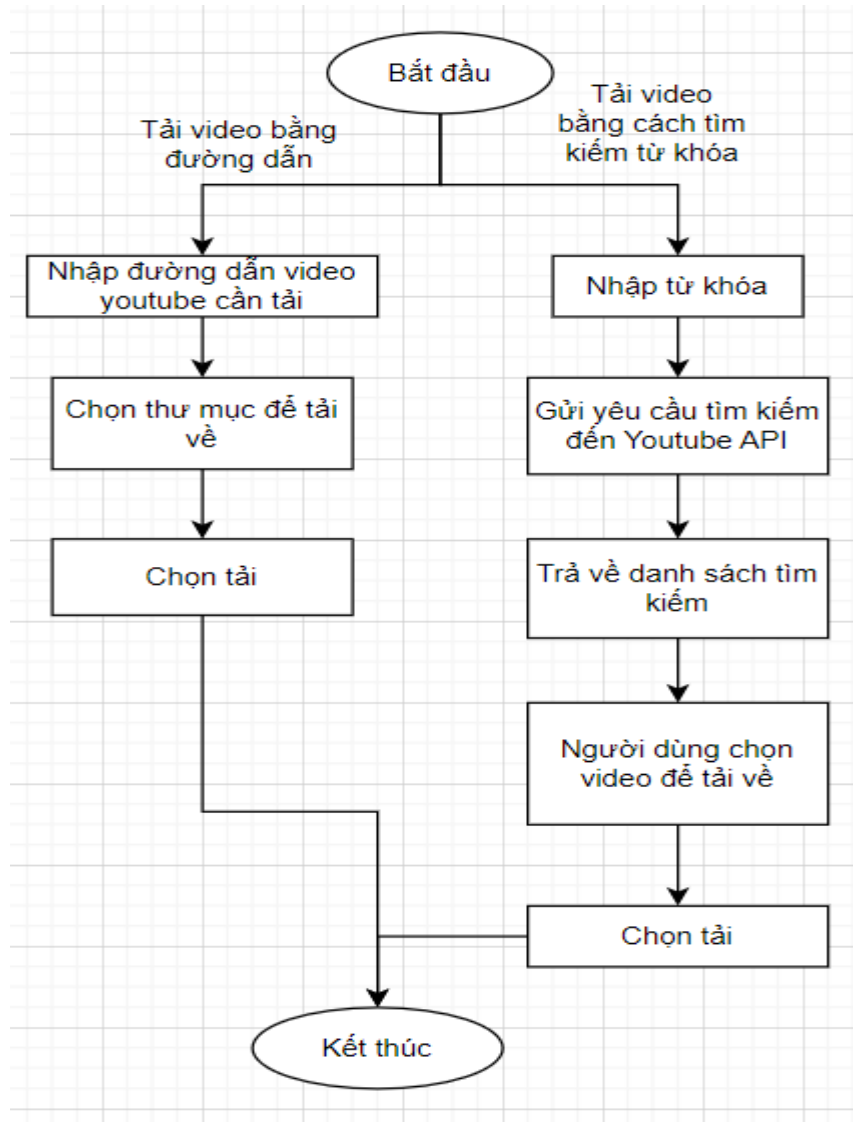
Yêu cầu cài đặt phần mềm bổ sung:

Để sử dụng MoviePy, bạn cần cài đặt một số phần mềm bổ sung như FFMpeg để có thể thực hiện các thao tác xử lý video.

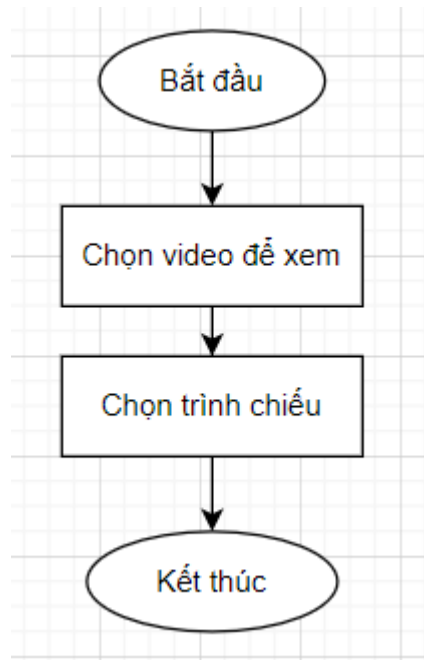
Tóm lại, MoviePy là một thư viện mạnh mẽ và tiện ích cho việc xử lý video trong Python, nhưng vẫn cần xem xét các yếu điểm và yêu cầu cụ thể của dự án trước khi quyết định sử dụng nó.

III. THIẾT KẾ ỨNG DỤNG

3.1. Chức năng tìm kiếm



3.2. Chức năng phát video



3.3. Thực tế

3.3.1. Tải video


- Lấy API của Youtube

Bước 1: Truy cập đường dẫn <https://code.google.com/apis/console>

Bước 2: Tạo project



Select a project

 NEW PROJECT

Search projects and folders

RECENTSTARREDALL

| | Name | ID |
|-----|------------------------------------|----------------------|
| ✓ ☆ | My Project 29416 ? | vocal-day-423518-n3 |
| ☆ | My First Project ? | neon-coast-423517-s5 |

CANCEL

Bước 3: Đặt tên project và create

New Project



You have 10 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *

My Project 96689



Project ID: just-oarlock-423623-g4. It cannot be changed later. [EDIT](#)

Location *

No organization

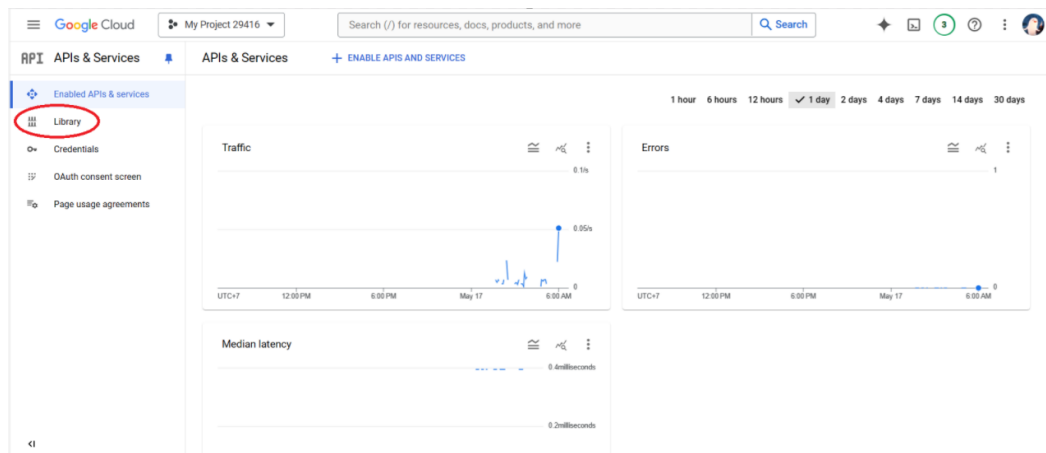
[BROWSE](#)

Parent organization or folder

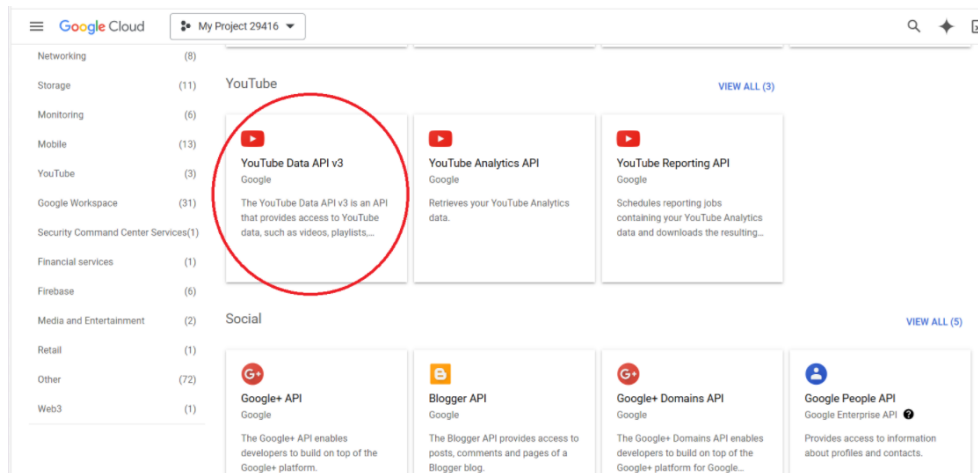
CREATE

CANCEL

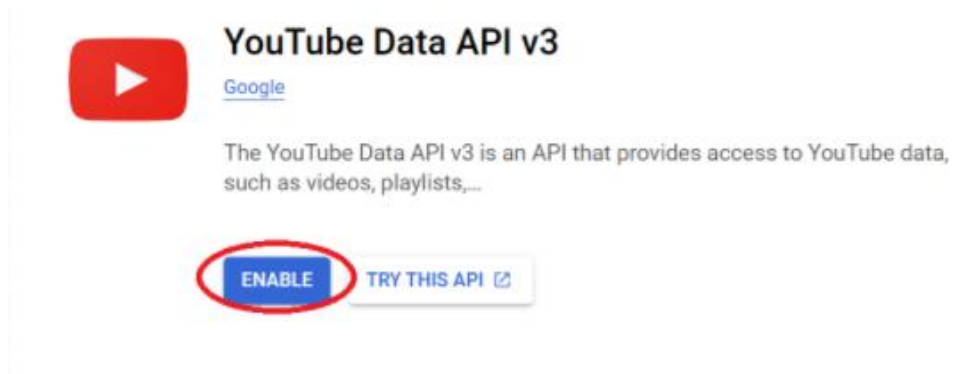
Bước 4: Vào Library



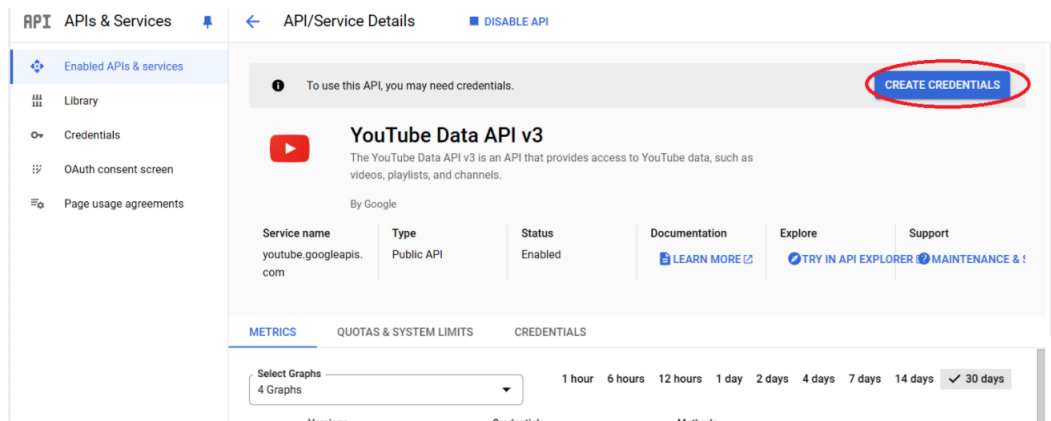
Bước 5: Chọn YouTube Data API v3



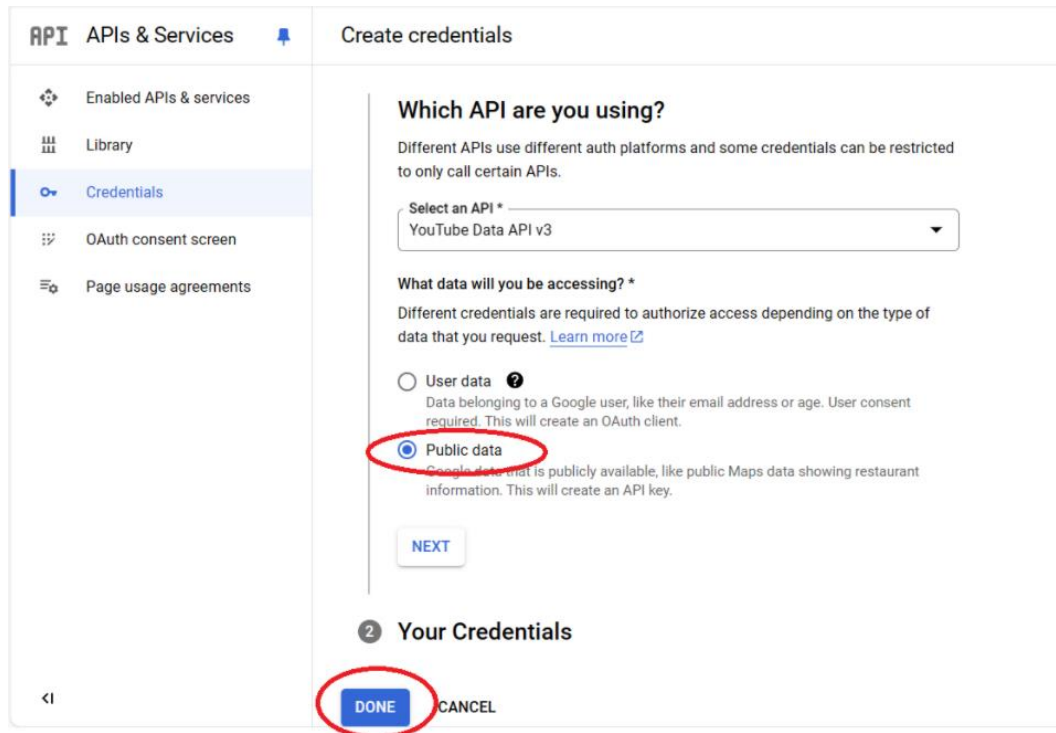
Bước 6: nhấn Enable



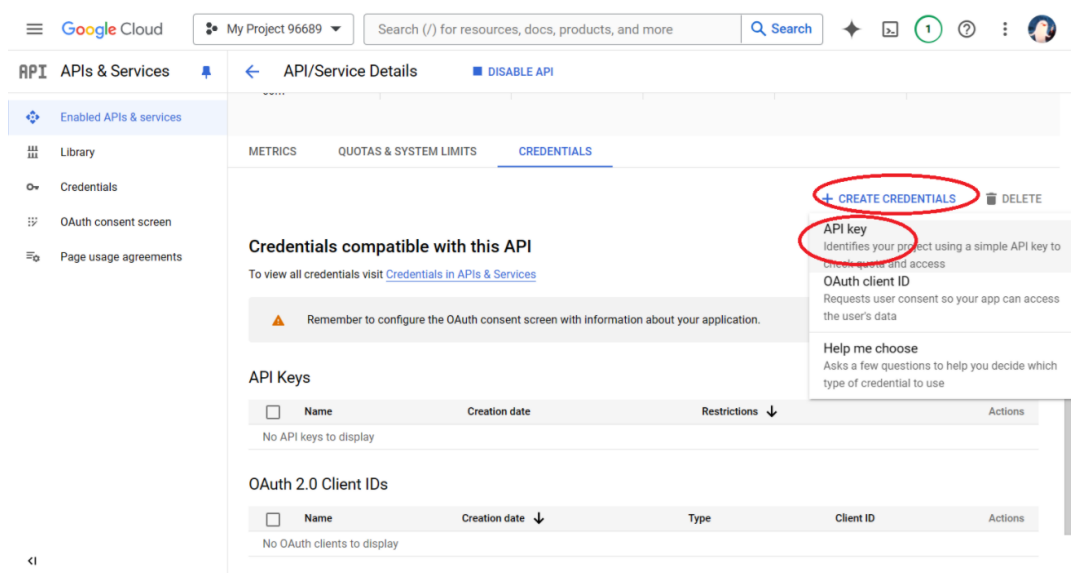
Bước 7: Chọn create credentials



Bước 8: Chọn public data và nhấn done



Bước 9: chọn create credentials sau đó chọn API Key



Bước 10: copy API Key vừa tạo

API key 1

Use this key in your application by passing it with the `key=API_KEY` parameter.

Your API key

AIzaSyAaNFraDAI36Y0nTCtrRaeJr5nLus-Gx08



⚠ This key is unrestricted. To prevent unauthorized use, we recommend restricting where and for which APIs it can be used. [Edit API key](#) to add restrictions. [Learn more](#)

CLOSE

- Tải video bằng đường dẫn (url) youtube

```
def download_video(url):
    try:
        yt = YouTube(url)
        stream = yt.streams.get_highest_resolution()
        filename = stream.default_filename
        stream.download(output_path=download_path)
        if tab_control.index(tab_control.select()) == 0:
            status_label.config(text="Download completed!", fg="green")
        elif tab_control.index(tab_control.select()) == 1:
            status_label_url.config(text="Download completed!", fg="green")
        update_downloaded_videos_list()
    except Exception as e:
        if tab_control.index(tab_control.select()) == 0:
            status_label.config(text="Error: " + str(e), fg="red")
        elif tab_control.index(tab_control.select()) == 1:
            status_label_url.config(text="Error: " + str(e), fg="red")
```

-Hàm download_video(url):

Đây là một hàm Python với một tham số đầu vào url, đại diện cho đường dẫn của video cần tải xuống.

-Thử và Xử lý Ngoại lệ (try-except):

Một khối try-except được sử dụng để xử lý các ngoại lệ có thể xảy ra trong quá trình tải xuống video. Nếu có bất kỳ lỗi nào xảy ra trong khối try, nó sẽ được bắt bởi khối except và mã bên trong nó sẽ được thực thi.

Tạo đối tượng YouTube và lấy đối tượng Stream:

Hàm YouTube(url) được sử dụng để tạo một đối tượng YouTube từ URL được cung cấp. Sau đó, yt.streams.get_highest_resolution () được sử dụng để lấy đối tượng Stream có độ phân giải cao nhất của video.

-Tải xuống video và cập nhật trạng thái:

Sau khi có được đối tượng Stream, mã sẽ gán tên mặc định của video vào biến filename. Tiếp theo, hàm stream.download() được sử dụng để tải xuống video vào một đường dẫn được chỉ định (download_path).

Sau khi quá trình tải xuống hoàn thành, trạng thái sẽ được cập nhật. Nếu tab hiện tại đang ở tab thứ nhất (tab_control.index(tab_control.select()) == 0), status_label sẽ được cập nhật với thông báo "Download completed!" và màu chữ màu xanh lam. Nếu tab hiện tại đang ở tab thứ hai, status_label_url sẽ được cập nhật tương tự.

Cuối cùng, hàm update_downloaded_videos_list () được gọi để cập nhật danh sách các video đã được tải xuống.

Xử lý Ngoại lệ:

Trong trường hợp có lỗi, ngoại lệ sẽ được bắt và mã sẽ cập nhật trạng thái tương ứng với lỗi xảy ra. Nếu tab hiện tại đang ở tab thứ nhất, status_label sẽ được cập nhật với thông báo lỗi và màu chữ đỏ. Nếu tab hiện tại đang ở tab thứ hai, status_label_url sẽ được cập nhật tương tự.

- Tải video bằng cách tìm từ khóa

```
def search_youtube():
    query = search_entry.get()
    youtube = build("youtube", "v3", developerKey=API_KEY)
    request = youtube.search().list(
        q=query,
        part="snippet",
        type="video",
        maxResults=10
    )
    response = request.execute()
    search_results.delete(0, tk.END)
    for item in response["items"]:
        search_results.insert(tk.END, item["snippet"]["title"])
```

- Tính năng tìm kiếm:
 - + Hàm build() được sử dụng để tạo một đối tượng YouTube API với cấu hình cụ thể. Trong trường hợp này, chúng ta đang sử dụng YouTube Data API v3 và cung cấp khóa phát triển (API_KEY).
 - + Đối tượng YouTube API này được sử dụng để tạo yêu cầu tìm kiếm thông qua phương thức search().list().
 - + Các tham số được truyền cho yêu cầu tìm kiếm bao gồm:
 - q: Từ khóa tìm kiếm (query) được lấy từ biến query.
 - part: Một phần bắt buộc, chỉ định các phần dữ liệu bạn muốn bao gồm trong kết quả. Ở đây, chỉ cần thông tin mô tả (snippet).
 - type: Loại dữ liệu bạn muốn tìm kiếm. Ở đây, chỉ tìm kiếm video.
 - maxResults: Số lượng kết quả tìm kiếm tối đa muốn trả về.

```
def select_video_to_download():
    selected_index = search_results.curselection()
    if selected_index:
        selected_video_title = search_results.get(selected_index)
        query = search_entry.get()
        youtube = build("youtube", "v3", developerKey=API_KEY)
        request = youtube.search().list(
            q=query,
            part="snippet",
            type="video",
            maxResults=10
        )
        response = request.execute()
        for item in response["items"]:
            if item["snippet"]["title"] == selected_video_title:
                selected_video_id = item["id"]["videoId"]
                selected_video_url = f"https://www.youtube.com/watch?v={selected_video_id}"
                download_video(selected_video_url)
```

- Tính năng tải về:

+ Lấy Chỉ số của Video Được Chọn:

- * Hàm `curselection()` được sử dụng để lấy chỉ mục của mục được chọn trong widget `search_results`.

- * Nếu `selected_index` khác rỗng, nghĩa là một video đã được chọn.

- * Lấy Tiêu Đề Video Được Chọn:

- * Tiêu đề của video được chọn được lấy từ danh sách kết quả tìm kiếm sử dụng phương thức `get(selected_index)`.

+ Tạo Yêu Cầu Tìm Kiếm:

- * Tương tự như trong hàm `search_youtube()`, một yêu cầu tìm kiếm được tạo để lấy thông tin về các video có liên quan đến từ khóa tìm kiếm.

- * Phần này có thể thực hiện một cách tối ưu hơn bằng cách truyền danh sách kết quả tìm kiếm từ hàm `search_youtube()`.

+ Lặp qua Kết Quả và Xác Định Video Được Chọn:

- * Vòng lặp được sử dụng để lặp qua mỗi mục trong `response["items"]`.

- * Nếu tiêu đề của một mục trùng khớp với tiêu đề của video được chọn, `selected_video_id` được lấy từ `item["id"]["videoId"]` và sau đó tạo ra URL của video được chọn từ `selected_video_id`.

* Cuối cùng, hàm `download_video(selected_video_url)` được gọi để tải xuống video được chọn.

- Tính năng xem video

```
def play_video():
    selected_index = downloaded_videos_listbox.curselection()
    if selected_index:
        selected_video = downloaded_videos_listbox.get(selected_index)
        video_path = os.path.join(download_path, selected_video)
        try:
            video = VideoFileClip(video_path)
            video.preview()
        except Exception as e:
            status_label.config(text="Error: " + str(e), fg="red")
```

- Lấy Chỉ số của Video Được Chọn:

+ Hàm `curselection()` được sử dụng để lấy chỉ mục của mục được chọn trong widget `downloaded_videos_listbox`.

+ Nếu `selected_index` khác rỗng, nghĩa là một video đã được chọn.

- Tên của Video Được Chọn:

+ Tên của video được chọn được lấy từ danh sách `downloaded_videos_listbox` sử dụng phương thức `get(selected_index)`.

- Xây dựng Đường Dẫn Video:

+ Đường dẫn đến video được chọn được tạo bằng cách kết hợp `download_path` (thư mục nơi video đã được tải xuống) với tên của video được chọn.

3.3.2. Phát video

Trong khối try-except, một VideoFileClip được tạo từ video được chọn bằng cách sử dụng video_path như một đối số.

Sau đó, phương thức preview () được gọi để chơi video trong trình xem video mặc định trên hệ thống.

3.4. Kết quả

Ứng dụng đã hoàn thành các chức năng đề ra:

- Tìm kiếm và hiển thị danh sách video theo từ khóa.
- Hiển thị thumbnail của video khi được chọn.
- Tải video về máy tính khi người dùng chọn tải.

IV. KẾT LUẬN

Ứng dụng tải video từ YouTube bằng Python đã chứng minh tính khả thi và hiệu quả của việc sử dụng các thư viện như pytube, google-api-python-client, và Pillow. Ứng dụng này không chỉ hỗ trợ tìm kiếm và tải video mà còn cung cấp giao diện thân thiện với người dùng, giúp việc sử dụng trở nên dễ dàng hơn.

Đề xuất

Để cải thiện và mở rộng ứng dụng này, chúng tôi đề xuất một số hướng phát triển sau:

- Thêm tính năng tải danh sách phát (playlist): Cho phép người dùng tải toàn bộ danh sách phát từ YouTube.
- Hỗ trợ định dạng và chất lượng video khác nhau: Cho phép người dùng chọn định dạng và chất lượng video trước khi tải.
- Tối ưu hóa giao diện người dùng: Cải thiện giao diện để trực quan và dễ sử dụng hơn, đặc biệt là khi hiển thị nhiều kết quả tìm kiếm.

V. PHÂN CHIA CÔNG VIỆC TRONG NHÓM

| Mssv | Tên | Công việc | % Công việc | Email |
|------------|-------------------|---|-------------|---------------------------|
| 3119560065 | Nguyễn Minh Thắng | Tìm kiếm, tải video | 33,33% | nm.thang994@gmail.com |
| 3119410180 | Ngô Phú Khang | Tải video bằng url | 33,33% | ngophukhang2001@gmail.com |
| 3119410094 | Phạm Ngọc Đông | Trình phát video đã tải, đường dẫn tải về | 33,33% | Tompham2901@gmail.com |

VI. TÀI LIỆU THAM KHẢO:

- YouTube Data API v3 : <https://code.google.com/apis/console>
- pytube Documentation
- Tkinter Documentation