

TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH  
KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO CUỐI KỲ**  
**CÔNG NGHỆ PHẦN MỀM NÂNG CAO**

**ĐỀ TÀI XÂY DỰNG VÀ PHÁT TRIỂN CÔNG CỤ TẢI**  
**XUỐNG VIDEO TỪ NỀN TẢNG YOUTUBE**

**Giảng viên hướng dẫn** : *Trần Sơn Hải*  
*Nguyễn Tấn Duẩn*  
*Lê Thanh Thọại*

**Mã học phần** : COMP106901

**Tên nhóm** : 5-NEKO

**Thành viên nhóm** :

Trịnh Hoàng Tùng	-	46.01.104.211
Nguyễn Trịnh Thành	-	46.01.104.169
Phạm Quốc Anh Quân	-	46.01.104.146
Lê Quốc Đạt	-	46.01.104.025

**THÀNH PHỐ HỒ CHÍ MINH – 10/2023**

TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO CUỐI KỲ**

**CÔNG NGHỆ PHẦN MỀM NÂNG CAO**

**XÂY DỰNG VÀ PHÁT TRIỂN CÔNG CỤ TẢI XUỐNG  
VIDEO TỪ NỀN TẢNG YOUTUBE**

**Giảng viên hướng dẫn** : *Trần Sơn Hải*  
*Nguyễn Tấn Duẩn*  
*Lê Thanh Thoại*

**Mã học phần** : COMP106901

**Tên nhóm** : 5-NEKO

**Thành viên nhóm** :

Trịnh Hoàng Tùng	-	46.01.104.211
Nguyễn Trịnh Thành	-	46.01.104.169
Phạm Quốc Anh Quân	-	46.01.104.146
Lê Quốc Đạt	-	46.01.104.025

**THÀNH PHỐ HỒ CHÍ MINH – 10/2023**

# MỤC LỤC

<b>CHƯƠNG 1. CƠ SỞ LÝ THUYẾT .....</b>	<b>4</b>
<b>1.1. NỀN TẢNG CƠ SỞ XÂY DỰNG .....</b>	<b>5</b>
1.1.1. Xây dựng ứng dụng web bằng Django.....	5
1.1.2. Xử lý lấy dữ liệu và tải về video từ Youtube.....	5
<b>1.2. CÁC THƯ VIỆN SỬ DỤNG.....</b>	<b>5</b>
1.2.1. Thư viện Django.....	5
1.2.2. Thư viện Youtube-dl.....	7
<b>CHƯƠNG 2. XÂY DỰNG CHƯƠNG TRÌNH.....</b>	<b>10</b>
<b>2.1. XÂY DỰNG CẤU TRÚC WEB-APP DJANGO.....</b>	<b>10</b>
<b>2.2. TÍCH HỢP THƯ VIỆN YOUTUBE-DL VÀ XÂY DỰNG TÍNH NĂNG .....</b>	<b>12</b>
<b>2.3. XÂY DỰNG GIAO DIỆN NGƯỜI DÙNG .....</b>	<b>16</b>
<b>CHƯƠNG 3. THỰC NGHIỆM CHƯƠNG TRÌNH .....</b>	<b>19</b>
<b>3.1. CẤU HÌNH MÁY VÀ MÔI TRƯỜNG CHẠY THỰC NGHIỆM.....</b>	<b>19</b>
3.1.1. Về phần cứng.....	19
3.1.2. Về phần mềm.....	19
<b>3.2. CHẠY TRÊN LOCALHOST .....</b>	<b>19</b>
<b>CHƯƠNG 4. KIỂM THỬ VÀ KẾT LUẬN .....</b>	<b>21</b>
<b>4.1. KIỂM THỬ FORM NHẬP ĐƯỜNG DẪN LIÊN KẾT.....</b>	<b>21</b>
<b>4.2. KIỂM THỬ TẬP TIN TẢI XUỐNG .....</b>	<b>22</b>
4.2.1. Lưu dưới dạng video .....	24
4.2.2. Lưu dưới dạng âm thanh.....	25
<b>4.3. KẾT LUẬN.....</b>	<b>25</b>

## LỜI MỞ ĐẦU

Quá trình thực nghiệm được thực hiện tại Khoa Công nghệ thông tin – Trường Đại học Sư phạm Thành phố Hồ Chí Minh, dưới sự hướng dẫn khoa học của *TS. Trần Sơn Hải, Thầy Nguyễn Tấn Duẩn và Thầy Lê Thanh Thoại*.

Trước tiên chúng em xin bày tỏ lòng biết ơn sâu sắc tới *TS. Trần Sơn Hải, Thầy Nguyễn Tấn Duẩn và Thầy Lê Thanh Thoại* đã đưa chúng em đến với lĩnh vực Công nghệ phần mềm nâng cao. Các thầy đã tận tình giảng dạy, hướng dẫn chúng em tiếp cận và đạt được những kết quả nhất định trong quá trình làm việc của mình. Các thầy đã luôn tận tâm động viên, khuyến khích và chỉ dẫn giúp chúng em hoàn thành bài báo cáo này.

Chúng em xin bày tỏ lòng biết ơn tới các Thầy Cô thuộc Khoa Công nghệ thông tin và cán bộ Phòng Khoa học Công nghệ, khoa Công nghệ Thông tin – Trường Đại học Sư Phạm Thành phố Hồ Chí Minh đã tạo mọi điều kiện thuận lợi giúp đỡ chúng em trong quá trình học tập và nghiên cứu.

Sự hướng dẫn của các thầy đã tận tình hướng dẫn, động viên, cổ vũ của gia đình, bạn bè là nguồn động lực quan trọng để chúng em thực hiện đề tài nghiên cứu. Do kiến thức còn hạn chế, nên đề tài báo cáo của chúng em không tránh khỏi những thiếu sót, kính mong sự thông cảm, chỉ bảo của quý Thầy Cô.

Chúng em xin chân thành cảm ơn.

**Thay mặt nhóm thực hiện.**

Trịnh Hoàng Tùng (*Nhóm Neko*)

## CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

### 1.1. Nền tảng cơ sở xây dựng

#### 1.1.1. Xây dựng ứng dụng web bằng Django

Xây dựng ứng dụng web là một lĩnh vực trong công nghệ phần mềm, trong đó Django là một thư viện nền tảng phục vụ cho việc xây dựng ứng dụng web nhanh chóng và dễ bảo trì

Dự án yêu cầu tối thiểu xây dựng ứng dụng web bằng thư viện Django, với tính năng có thể nhập địa chỉ liên kết tới video trên nền tảng Youtube, sau đó lấy dữ liệu và cho phép tải xuống các định dạng bản sao video cũng như âm thanh dựa trên đoạn liên kết đã được người dùng nhập.

#### 1.1.2. Xử lý lấy dữ liệu và tải về video từ Youtube

Xử lý và lấy dữ liệu video là vấn đề chính của dự án. Bài toán này sẽ nhờ vào thư viện bên thứ ba – Youtube-dl – để giải quyết vấn đề truy xuất và lấy về dữ liệu video từ máy chủ Youtube.

### 1.2. Các thư viện sử dụng

#### 1.2.1. Thư viện Django



*Hình 1.2.1. Logo nhận diện Django*

Thư viện Django là một framework phát triển ứng dụng web mạnh mẽ, linh hoạt và được xây dựng bằng Python. Nó giúp phát triển các ứng dụng web phức tạp một cách nhanh chóng và hiệu quả, với sự hỗ trợ của nhiều tính năng sẵn có. Django tập trung vào

việc tối ưu hóa sự phát triển và quản lý dự án bằng cách cung cấp các giải pháp chuẩn cho nhiều nhiệm vụ phổ biến trong phát triển web.

Các đặc điểm nổi bật của Django:

- ORM (Object-Relational Mapping – bản đồ mối quan hệ giữa các đối tượng)
  - Django đi kèm với một ORM mạnh mẽ, cho phép mô hình hóa cơ sở dữ liệu dưới dạng các đối tượng Python. Điều này giúp đơn giản hóa quản lý dữ liệu và truy vấn cơ sở dữ liệu. Tuy nhiên, do dự án này được thiết kế không phụ thuộc vào cơ sở dữ liệu nên tính năng này chưa được triển khai trên dự án.
- Hệ thống quản lý các template mạnh mẽ
  - Django có hệ thống template vô cùng mạnh mẽ, cho phép tách biệt hoàn toàn mã HTML và mã Python. Điều này giúp quản lý giao diện người dùng một cách hiệu quả.
- Hệ thống xử lý yêu cầu (Request-Response Cycle)
  - Django xử lý yêu cầu theo mô hình yêu cầu-đáp ứng (request-response cycle) rõ ràng, giúp tập trung vào việc xử lý logic ứng dụng.
  - Cho phép cập nhật các nội dung và xử lý mới sau mỗi lần lưu lại code.
- Hệ thống xử lý URL và định tuyến
  - Django cung cấp cơ chế mạnh mẽ để xử lý URL và định tuyến yêu cầu đến các view tương ứng, giúp quản lý các đường dẫn của ứng dụng dễ dàng với mô hình xây dựng mã nguồn (design pattern) MVT (model – template – view).
- Bảo mật và quyền người dùng
  - Django đi kèm với các tính năng bảo mật tích hợp, giúp quản lý quyền người dùng, xác thực, và bảo vệ ứng dụng chống lại các lỗ hổng bảo mật phổ biến.
- Tích hợp API và những thư viện Python khác

- Django hỗ trợ tích hợp API một cách dễ dàng thông qua các ứng dụng như Django Rest Framework, giúp xây dựng và duy trì các dịch vụ API linh hoạt.

Tóm lại, Django cho phép bạn xây dựng ứng dụng web nhanh chóng và dễ bảo trì. Qua sự linh hoạt và khả năng mở rộng của mình, Django là một lựa chọn phổ biến và mạnh mẽ cho các dự án phát triển ứng dụng web.

### 1.2.2. Thư viện Youtube-dl



*Hình 1.2.2.1. Logo nhận diện Youtube-dl*

Đây là một thư viện mã nguồn mở được xây dựng để tải về video từ Youtube. Ở dự án này, nhóm xây dựng ứng dụng Django tích hợp thư viện youtube-dl để gọi API đến thư viện này nhằm tải xuống những video từ Youtube nhờ vào đường dẫn liên kết tới video đã được nhập vào.

Các tính năng, lợi ích ứng dụng mà Youtube-dl mang lại khi dự án được tích hợp thư viện này:

- Hỗ trợ đa định dạng: Youtube-dl cho phép người dùng chọn giữa nhiều định dạng video và âm thanh khác nhau khi tải xuống. Điều này bao gồm cả các định dạng video chất lượng cao (với nhiều tùy chọn độ phân giải video) và các định dạng âm thanh như mp3.
- Tải xuống danh sách phát: Có thể sử dụng Youtube-dl để tải xuống toàn bộ danh sách phát YouTube hoặc từ các trang web khác.

- Được cập nhật thường xuyên: Do cộng đồng phát triển sôi nổi, Youtube-dl thường xuyên được cập nhật để nhận những thay đổi mới về tính năng từ các nhà phát triển.
- Cách sử dụng đơn giản: Nếu không tích hợp Django (mà hoàn toàn dùng dòng lệnh thuần), thì cách sử dụng thư viện này hoàn toàn đơn giản. Bắt đầu với việc tải xuống một video Youtube bằng dòng lệnh sau:

```
ydl.download([video_url])
```

Ta có ví dụ cho đoạn mã sau:

```
1 import youtube_dl
2
3 def download_video(video_url, output_path='.'):
4     options = {
5         'format': 'bestvideo+bestaudio/best',
6         'outtmpl': f'{output_path}/%(title)s.%(ext)s',
7     }
8
9     with youtube_dl.YoutubeDL(options) as ydl:
10         ydl.download([video_url])
11
12 if __name__ == "__main__":
13     video_url = input("Enter YouTube video URL: ")
14     output_path = input("Enter the output path (default is the current directory): ") or '.'
15
16     download_video(video_url, output_path)
17
```

Hình 1.2.2.2. Ví dụ ứng dụng thư viện Youtube-dl

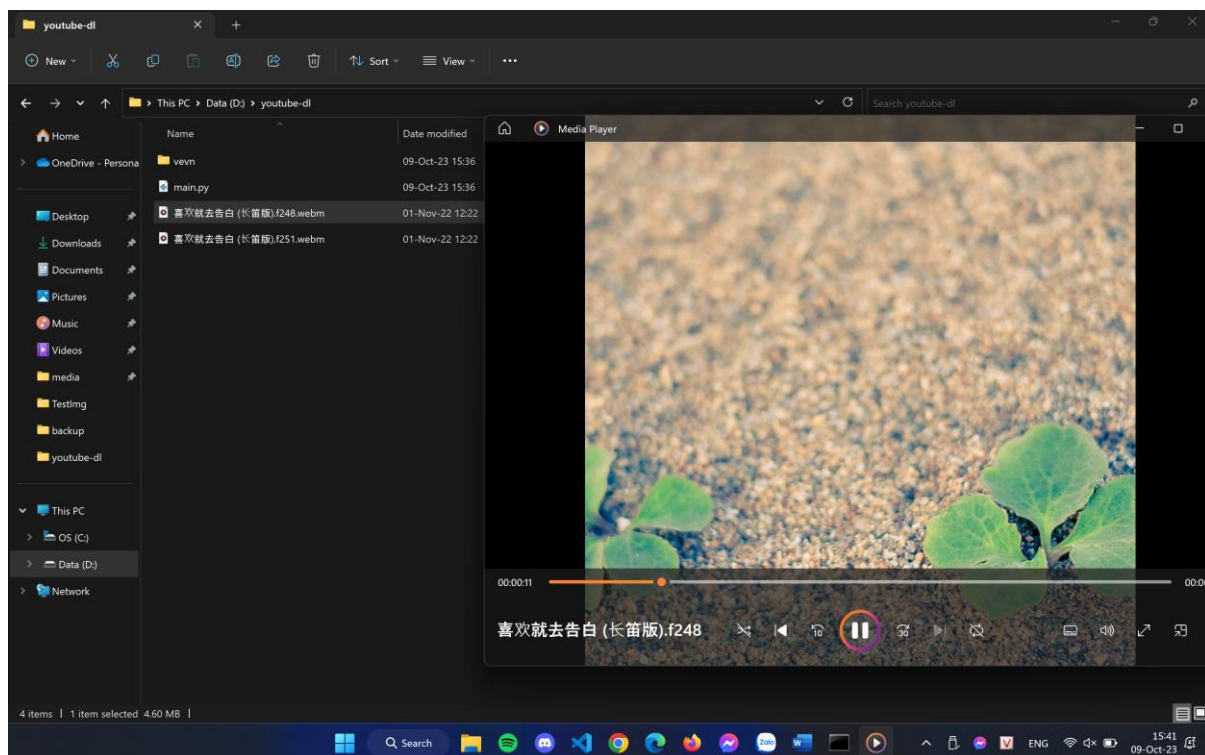
Video sẽ được tự động tải về và lưu ở vị trí thư mục chứa tập tin thực thi dòng lệnh trên.

```
(vevn) D:\youtube-dl>pip install youtube_dl
Collecting youtube_dl
  Using cached youtube_dl-2021.12.17-py2.py3-none-any.whl (1.9 MB)
Installing collected packages: youtube_dl
Successfully installed youtube_dl-2021.12.17

[notice] A new release of pip available: 22.3.1 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(vevn) D:\youtube-dl>python main.py
Enter YouTube video URL: https://youtu.be/Jun804LSPv4
Enter the output path (default is the current directory): .
[youtube] Jun804LSPv4: Downloading webpage
WARNING: You have requested multiple formats but ffmpeg or avconv are not installed. The formats won't be merged.
[download] Destination: 喜欢就去告白 (长笛版).f248.webm
[download] 100% of 4.60MiB in 01:03
[download] Destination: 喜欢就去告白 (长笛版).f251.webm
[download] 100% of 1.09MiB in 00:15
```





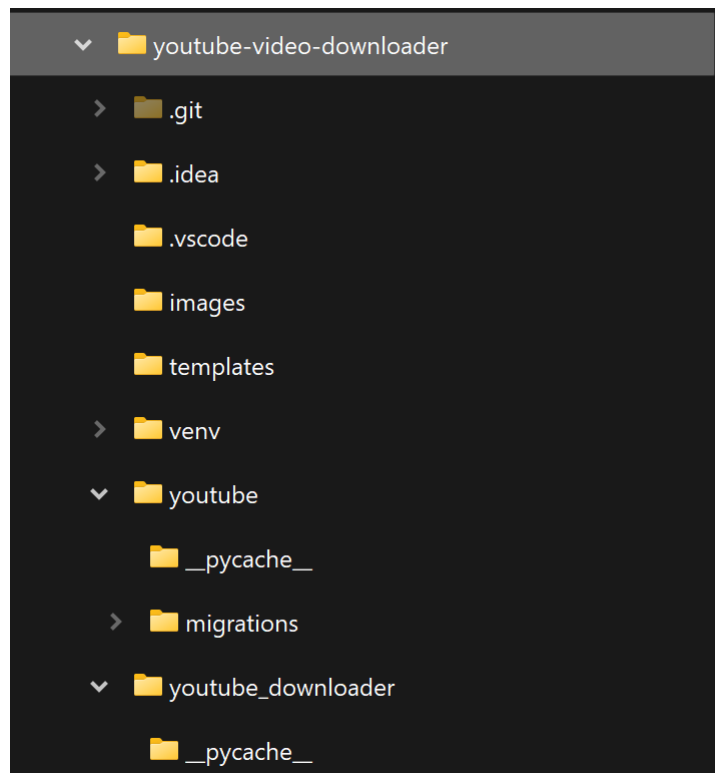
*Hình 1.2.2.3. Kết quả ứng dụng thư viện*

Tuy nhiên, dự án cần một giao diện thân thiện hơn là màn hình dòng lệnh, nên nhóm sẽ tích hợp thêm các thư viện khác đã liệt kê phía trên, đặc biệt là sử dụng Django để viết một ứng dụng web hoàn chỉnh.

## CHƯƠNG 2. XÂY DỰNG CHƯƠNG TRÌNH

### 2.1. Xây dựng cấu trúc web-app Django

Một ứng dụng viết bằng Django khi mới khởi tạo sẽ có cấu trúc thư mục như bên dưới. Ngoài ra ở đây nhóm sử dụng môi trường ảo (venv) được ngôn ngữ Python cung cấp sẵn để thiết lập các thư viện dành riêng cho dự án này, tránh gây ảnh hưởng đến các dự án sử dụng Python khác.

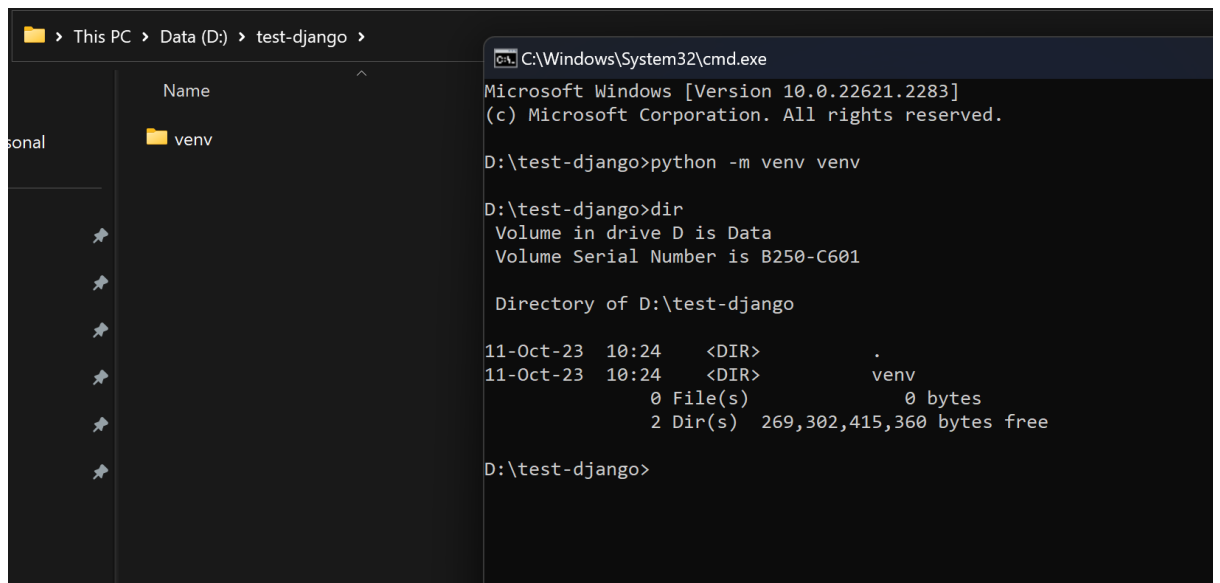


Hình 2.1.1. Cấu trúc một web-app Django

Các bước để thực hiện khởi tạo một ứng dụng như trên, bao gồm:

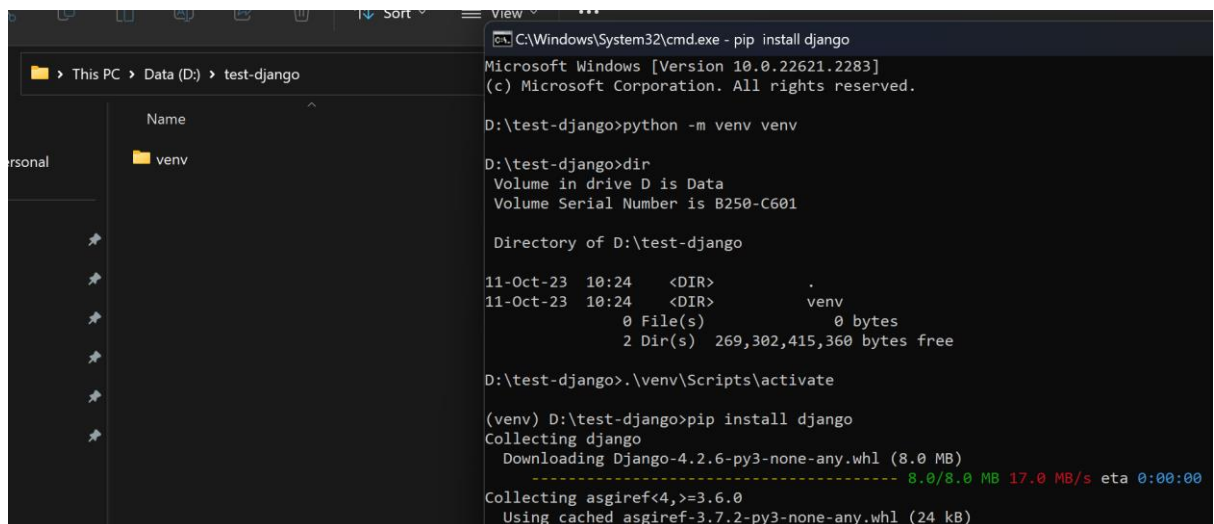
- Khởi tạo môi trường ảo venv:

*python -m venv venv*



*Hình 2.1.2. Khởi tạo môi trường ảo (venv)*

- Truy cập môi trường ảo và cài đặt Django trong môi trường này



*Hình 2.1.3. Cài đặt thư viện Django trong môi trường ảo Python*

- Khởi tạo dự án Django với tên `youtube_downloader`

```
C:\Windows\System32\cmd.exe
Using cached tzdata-2023.3-py2.py3-none-any.whl (341 kB)
collecting typing_extensions=4
Using cached typing_extensions-4.8.0-py3-none-any.whl (31 kB)
Installing collected packages: tzdata, typing_extensions, sqlparse, asgiref, django
Successfully installed asgiref-3.7.2 django-4.2.6 sqlparse-0.4.4 typing_extensions-4.8.0 tzdata-2023.3

[notice] A new release of pip available: 22.3.1 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip

venv) D:\test-django>django-admin startproject youtube-video-downloader
CommandError: 'youtube-video-downloader' is not a valid project name. Please make sure the name is a valid identifier.

venv) D:\test-django>django-admin startproject youtube_downloader

venv) D:\test-django>cd youtube_downloader

venv) D:\test-django\youtube_downloader>dir
Volume in drive D is Data
Volume Serial Number is B250-C601

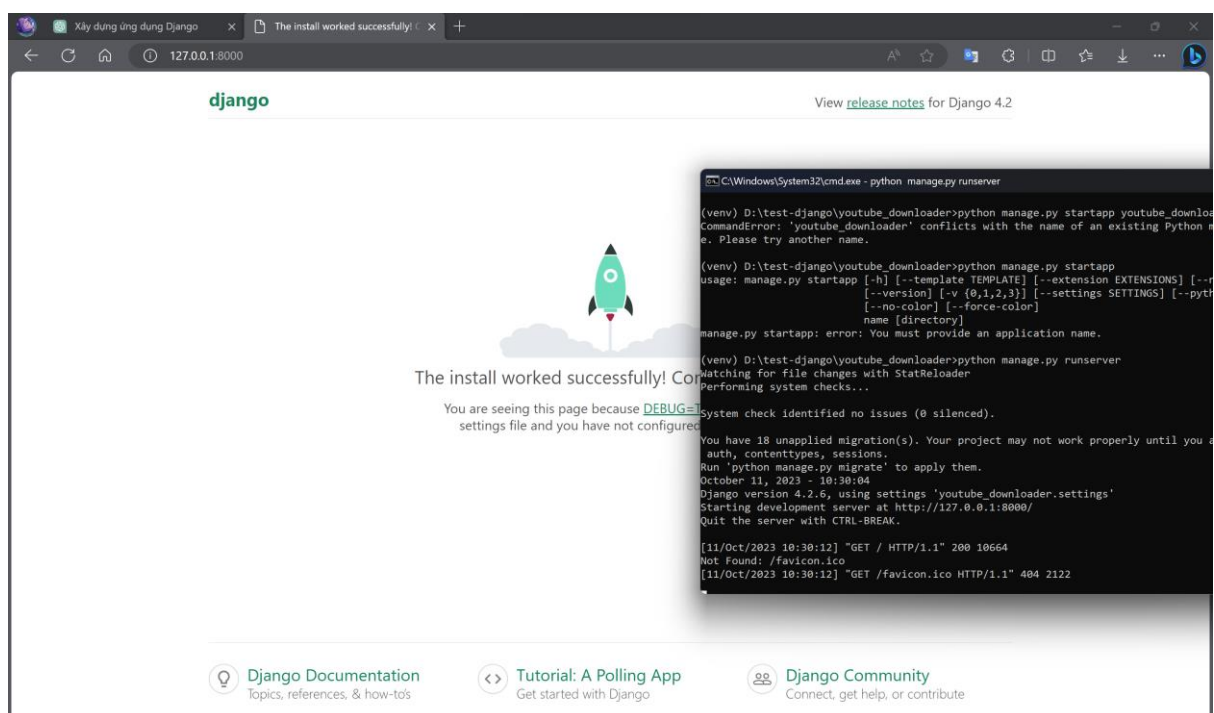
Directory of D:\test-django\youtube_downloader

1-Oct-23 10:28 <DIR>      .
1-Oct-23 10:28 <DIR>      ..
1-Oct-23 10:28          696 manage.py
1-Oct-23 10:28 <DIR>      youtube_downloader
                1 File(s)        696 bytes
                3 Dir(s)  269,260,288,000 bytes free

venv) D:\test-django\youtube_downloader>
```

Hình 2.1.4. Khởi tạo dự án Django

- Chạy thử lần đầu tiên



Hình 2.1.1. Lần đầu chạy thử trên localhost

## 2.2. Tích hợp thư viện Youtube-dl và xây dựng tính năng

- Tích hợp Youtube-dl

```
# package
import youtube_dl as yt
# check video url
import re
```

*Hình 2.2.1. Tích hợp Youtube-dl*

- Kiểm tra cú pháp của đường liên kết được nhập vào

```
def index(request):
    form = DownloadForm(request.POST or None)
    if form.is_valid():
        url = form.cleaned_data.get("url")
        # check video url
        regex = r'^((http(s)?://)?((w){3}.)?youtu(be|.be)?(\.com)?)/.+'
        if not re.match(regex, url):
            context = {
                'error': "Please enter correct url!",
                'form': form
            }
            return render(request, 'home.html', context)
```

*Hình 2.2.2. Regex đường dẫn video*

- Kiểm tra đường dẫn có đúng là đường dẫn liên kết tới một video Youtube hay không, có kết hợp với thư viện Youtube-dl để kiểm tra độ chính xác của đường dẫn được người dùng nhập vào.

```

yt_opts = {}

with yt.YoutubeDL(yt_opts) as ydl:
    try:
        meta = ydl.extract_info(
            url, download=False)
        video = True
    except:
        context = {
            'error': "Video unavailable. Please enter correct url!",
            'form': form
        }
    return render(request, 'home.html', context)

```

*Hình 2.2.2. Kiểm tra tính chính xác của đường dẫn*

- Trích xuất từ dữ liệu đã lấy sau khi kiểm tra đường dẫn, sau đó phân loại thành các độ phân giải video khác nhau cho người dùng có thể lựa chọn

```

video_audio_streams = []
audioStreams = []
for item in meta['formats']:
    file_size = item['filesize']
    itemFormat = item['format_note']

    resolution = 'Audio'
    if item['height'] is not None:
        resolution = f"{item['height']}x{item['width']}"

    if 'Audio' in resolution:
        audioStreams.append({
            'resolution': resolution,
            'extension': item['ext'],
            'file_size': file_size,
            'itemFormat': itemFormat,
            'url': item['url'],
        })
    else:
        video_audio_streams.append({
            'resolution': resolution,
            'extension': item['ext'],
            'file_size': file_size,
            'itemFormat': itemFormat,
            'url': item['url']
        })
video_audio_streams = video_audio_streams[::-1]

```

*Hình 2.2.3. Phân loại video*

- Lưu trữ các giá trị (context) đã trích xuất và đổ dữ liệu lên giao diện (index.html)

```

context = {
    'form': form,
    'isVideo': video,
    'url': url,
    'title': meta['title'],
    'streams': video_audio_streams,
    'audioStreams': audioStreams,
    'channel': meta['channel'],
    'image': meta['thumbnails'][3]['url'],
    'duration': round(int(meta['duration'])/60, 2),
    'views': f'{int(meta["view_count"]):,}'
}
return render(request, 'home.html', context)
return render(request, 'home.html', {'form': form})

```

Hình 2.2.4. Xử lý dữ liệu

### 2.3. Xây dựng giao diện người dùng

- Form nhập đường liên kết video để người dùng nhập vào:

```

<div class="container d-flex justify-content-center mt-5 text-light">
  <div class="col-md-12 mx-auto">
    <h1 class="text-center display-4 "><strong>Công cụ tải về video Youtube</strong></h1>
    <p class="text-center">Nhóm tác giả: Neko @ HCMUE - Công nghệ phần mềm nâng cao</p>

    {% if error %}
      <div class="alert alert-danger text-center mt-5 display-6">
        {{error}}
      </div>
    {% endif %}

    <form method="POST" class="mt-3">
      {% csrf_token %}
      <div class="form-floating mb-3 mt-5 form-group">
        {{form.url}}
        <label for="floatingInput" class="text-dark">Nhập đường dẫn video: </label>
      </div>
      <div class="d-flex justify-content-center">
        <button type="button" class="btn btn-danger btn-lg mt-3 text-light" data-bs-toggle="mod
      </div>
    </form>

```

Hình 2.3.1. Form nhập đường dẫn

- Hiện thị thông tin của video đã lấy được từ URL:



```

<div class="container text-dark mt-3">
  {% if isVideo %}
  <div class="bd-example mt-5 shadow-lg">
    <h4 class="text-center fw-bold mt-4 mb-3">Xem trước nội dung</h4>
    <div class="card mb-3 border-light bg-transparent border-2 text-light" >
      <div class="row g-0">
        <div class="col-md-4 p-3">
          <div class="img-thumb">
            
          </div>
        </div>
        <div class="col-md-8">
          <div class="card-body">
            <h4 class="card-title"><small class="text-success fw-bold lead">Tiêu đề video</small>
            <h5 class="card-text"><small class="text-success fw-bold lead">Kênh sở hữu: </small>
            <p class="card-text"><small class="text-success fw-bold lead">Thời lượng: </small>
            <p class="card-text"><small class="text-success fw-bold lead">Lượt xem: </small>
          </div>
        </div>
      </div>
    </div>
  </div>

```

- Hiện thị giao diện cho phép lưu dưới dạng định dạng video:

```

<thead class="thead-light">
  <tr>
    <th scope="col" class="text-dark">Độ phân giải</th>
    <th scope="col" class="text-dark">Đuôi file</th>
    <th scope="col" class="text-dark">Dung lượng</th>
    <th scope="col" class="text-dark">Lưu về</th>
  </tr>
</thead>
{% for video in streams %}
<tbody>
  <tr>
    <td scope="row"><p>{{video.resolution}} ({{video.itemFormat}})</p> </td>
    <td><p>{{ video.extension }}</p></td>
    <td><p>{{ video.file_size|filesizeformat }}</p></td>
    <td><a download href="{{video.url}}" class="btn"
      target="_blank" style="color: 960FF2;">Tải về PC <i
      class=" fa fa-download"></i></a></td>
  </tr>
</tbody>
{% endfor %}
</table>

```

- Và lưu dưới dạng âm thanh:

```

<table class="table text-light">
  <small class="lead p-3"><h1>Lưu dưới dạng âm thanh</h1></small>
  <thead>
    <tr>
      <th scope="col" class="text-dark">Độ phân giải</th>
      <th scope="col" class="text-dark">Đuôi file</th>
      <th scope="col" class="text-dark">Dung lượng</th>
      <th scope="col" class="text-dark">Lưu về</th>
    </tr>
  </thead>
  {% for audio in audioStreams %}
  <tbody>
    <tr>
      <td scope="row"><p>{{audio.resolution }} ({{audio.itemFormat}})</p> </td>
      <td><p>{{ audio.extension }}</p></td>
      <td><p>{{ audio.file_size|filesizeformat }}</p></td>
      <td><a href="{{audio.url}}" download="" class="btn"
        target="_blank">Lưu về PC <i
          class=" fa fa-download"></i></a></td>
    </tr>
  </tbody>
  {% endfor %}
</table>
</div>

```

- Các thông báo như “Vui lòng chờ. Đang lấy dữ liệu..” trong khi quá trình xử lý được diễn ra:

```

iv class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-
<div class="modal-dialog">
  <div class="modal-content">
    <div class="modal-header">
      <h5 class="modal-title" id="exampleModalLabel">Vui lòng chờ!</h5>
      <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></butt
    </div>
    <div class="modal-body">
      <div class="alert alert-primary alert-dismissible fade show" role="alert" style="font-siz
        <i class="fa-solid fa-file-arrow-down"></i>
        <strong>Vui lòng chờ!</strong> Đang lấy dữ liệu..
      </div>
    </div>
  </div>
</div>
</div>
div>

```

Hình 2.3.2. Hiển thị nội dung

## CHƯƠNG 3. THỰC NGHIỆM CHƯƠNG TRÌNH

Video demo (phiên bản beta): <https://youtu.be/ugRqcRXAVbM>

Source code (Github): <https://github.com/TrHgTung/NekoYoutubeDownloader>

### 3.1. Cấu hình máy và môi trường chạy thực nghiệm

#### 3.1.1. Về phần cứng

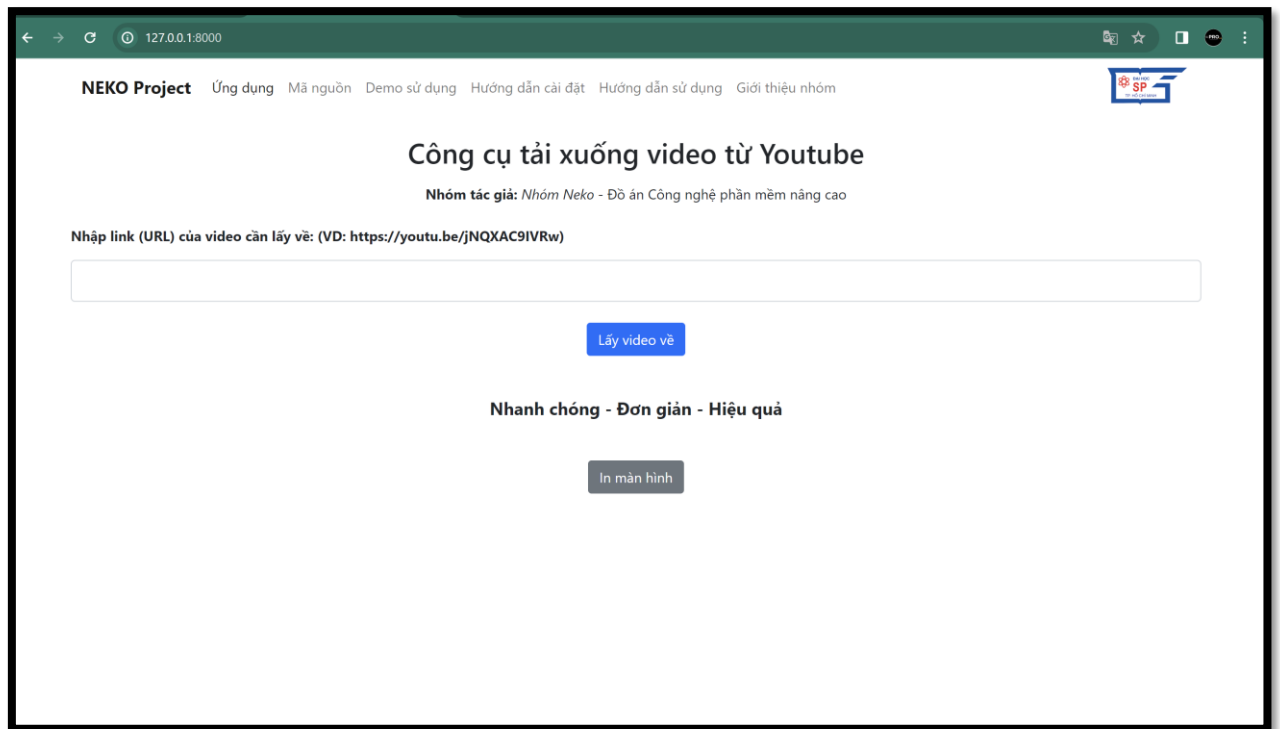
- CPU Intel 12<sup>th</sup> Gen Core i5-1240P (16 nhân) – xung nhịp 1.7 Ghz
- 16 GB RAM
- 512 GB SSD
- GPU Intel Iris Xe Graphics Family 8 GB VRAM

#### 3.1.2. Về phần mềm

- Microsoft Windows 11 Home
- Python 3.10.9
- Visual Studio Code 1.75.1
- Microsoft Edge 117.0.2045.60 (64-bit)

### 3.2. Chạy trên localhost

- Truy cập vào thư mục chứa mã nguồn, mở Command Line (Terminal) để chạy các môi trường cần thiết.
- Chạy môi trường ảo (venv): *./venv/Scripts/activate*
- Chạy trên máy chủ cục bộ (localhost) – cổng 8000: *python manage.py runserver 8000*
- Kiểm tra ứng dụng chạy trên trình duyệt Edge



*Hình 3.2.1. Demo trên localhost*

## CHƯƠNG 4. KIỂM THỬ VÀ KẾT LUẬN

### 4.1. Kiểm thử form nhập đường dẫn liên kết

Test case	Trường hợp	Mô tả trường hợp test	Kết quả kiểm thử
1	Để form trống	Không nhập gì vào form và nhấn nút Lấy video về	Ứng dụng bị treo
2	Nhập sai định dạng liên kết (không phải là một liên kết)	Nhập ngẫu nhiên “hgghghgh” và nhấn nút Lấy video về	Trả về thông báo “Please enter correct URL!”
3	Nhập liên kết không phải liên kết đến nền tảng Youtube	Nhập <a href="https://www.tiktok.com/@danpovenmire/video/7279896302782090539">https://www.tiktok.com/@danpovenmire/video/7279896302782090539</a> và nhấn nút Lấy video về	Trả về thông báo “Please enter correct URL!”
4	Nhập liên kết không phải là liên kết đến một video Youtube (một kênh Youtube / một danh sách phát Youtube)	Nhập <a href="https://www.youtube.com/channel/UCPLUuGVDDfjFRAhVq2FnOGA">https://www.youtube.com/channel/UCPLUuGVDDfjFRAhVq2FnOGA</a> và nhấn nút Lấy video về	Ứng dụng bị treo
5	Nhập đúng liên kết tới một video Youtube, nhưng cố tình làm thay đổi đường dẫn trước khi nhấn Lấy video về	Thay vì một đường dẫn đúng <a href="https://youtu.be/GgQFO8dL5XQ">https://youtu.be/GgQFO8dL5XQ</a> cần phải nhập, thì ta nhập <a href="https://youtu.be/GgQFO8dL5Q">https://youtu.be/GgQFO8dL5Q</a> (mất đi chữ cái X ở ký tự gần cuối)	Trả về thông báo “Video unavailable . Please enter correct URL!”
6	Nhập liên kết nhưng	Nhập vào <a href="http://www.youtube.com/watch?v=GgQFO8dL5XQ">www.youtube.com/watch?v=GgQFO8dL5XQ</a> và	Nội dung video được

	không có cú pháp “https://”	nhấn nút Lấy video về	lấy về và hiển thị lên giao diện
7	Nhập liên kết nhưng không có cú pháp “https://www”	Nhập vào <a href="https://www.youtube.com/watch?v=GgQFO8dL5XQ">youtube.com/watch?v=GgQFO8dL5XQ</a> và nhấn nút Lấy video về	Nội dung video được lấy về và hiển thị lên giao diện
8	Nhập đúng yêu cầu (ngoại trừ 7 trường hợp kể trên)	Nhập đúng một liên kết dẫn tới một video trên Youtube và nhấn nút Lấy video về	Nội dung video được lấy về và hiển thị lên giao diện

#### 4.2. Kiểm thử tập tin tải xuống

Sau khi lấy dữ liệu của một video Youtube thành công, ta có các trường hợp cần kiểm tra như sau: (đối với video có hỗ trợ độ phân giải 4K (2160p), có âm thanh và hình ảnh có phổ màu rộng).

NEKO Project

[Ứng dụng](#)
[Mã nguồn](#)
[Demo sử dụng](#)
[Hướng dẫn cài đặt](#)
[Hướng dẫn sử dụng](#)
[Giới thiệu nhóm](#)

127.0.0.1:8000

SP


Công cụ tải xuống video từ Youtube

Nhóm tác giả: Nhóm Neko - Đồ án Công nghệ phần mềm nâng cao

Nhập link (URL) của video cần lấy về: (VD: <https://youtu.be/jNQXAC9IVRw>)

Lấy video về

Xem trước nội dung



Tiêu đề video: [2021 LG OLED I Ink Art 4K HDR 60fps](#)

Kênh sở hữu: LG Global

Thời lượng: 0.75 phút

Lượt xem: 3,544,187

Lưu dưới dạng video

Độ phân giải	Đuôi file	Dung lượng	Lưu về
720x1280 (720p)	mp4	0 bytes	<a href="#">Tải về PC</a>
360x640 (360p)	mp4	0 bytes	<a href="#">Tải về PC</a>
2160x3840 (2160p60 HDR)	webm	146.6 MB	<a href="#">Tải về PC</a>
2160x3840 (2160p60)	webm	37.7 MB	<a href="#">Tải về PC</a>
1440x2560 (1440p60 HDR)	webm	83.5 MB	<a href="#">Tải về PC</a>
1440x2560 (1440p60)	webm	17.6 MB	<a href="#">Tải về PC</a>
1080x1920 (1080p60 HDR)	webm	33.0 MB	<a href="#">Tải về PC</a>
1080x1920 (1080p60)	mp4	14.7 MB	<a href="#">Tải về PC</a>
1080x1920 (1080p60)	webm	5.0 MB	<a href="#">Tải về PC</a>
720x1280 (720p60 HDR)	webm	21.9 MB	<a href="#">Tải về PC</a>

23

720x1280 (720p)	mp4	5.2 MB	Tải về PC
720x1280 (720p60)	webm	3.8 MB	Tải về PC
720x1280 (720p)	webm	2.6 MB	Tải về PC
480x854 (480p60 HDR)	webm	8.8 MB	Tải về PC
480x854 (480p)	mp4	2.8 MB	Tải về PC
480x854 (480p)	webm	1.5 MB	Tải về PC
360x640 (360p60 HDR)	webm	4.4 MB	Tải về PC
360x640 (360p)	mp4	1.6 MB	Tải về PC
360x640 (360p)	webm	1.0 MB	Tải về PC
240x426 (240p60 HDR)	webm	2.1 MB	Tải về PC
240x426 (240p)	mp4	888.9 KB	Tải về PC
240x426 (240p)	webm	671.5 KB	Tải về PC
144x256 (144p60 HDR)	webm	1022.6 KB	Tải về PC
144x256 (144p)	webm	431.9 KB	Tải về PC

240x426 (240p)	webm	671.5 KB	Tải về PC
144x256 (144p60 HDR)	webm	1022.6 KB	Tải về PC
144x256 (144p)	webm	431.9 KB	Tải về PC
144x256 (144p)	mp4	409.4 KB	Tải về PC

Lưu dưới dạng âm thanh

Độ phân giải	Đuôi file	Dung lượng	Lưu về
Audio (tiny)	webm	296.6 KB	Lưu về PC
Audio (tiny)	webm	390.8 KB	Lưu về PC
Audio (tiny)	m4a	719.6 KB	Lưu về PC
Audio (tiny)	webm	773.6 KB	Lưu về PC

Nhanh chóng - Đơn giản - Hiệu quả

Hình 4.2.1. Kết quả kiểm thử

#### 4.2.1. Lưu dưới dạng video

Test case	Trường hợp	Mô tả trường hợp test	Kết quả kiểm thử
1	338x640 (360p) – mp4	Lưu video (nhấn Tải về PC)	OK
2	2026x3840 (2160p) – webm		OK
3	2026x3840 (2160p) - mp4		OK



4	1350x2560 (1440p) – webm		OK
5	1350x2560 (1440p) – mp4		OK
6	1012x1920 (1080p) – mp4		OK
7	1012x1920 (1080p) - webm		OK
8	676x1280 (720p) – webm		OK
9	676x1280 (720p) – mp4		OK
10	450x854 (480p) – webm		OK
11	450x854 (480p) – mp4		OK
12	338x640 (360p) – webm		OK
13	338x640 (360p) – mp4		OK
14	224x426 (240p) – webm		OK
15	224x426 (240p) – mp4		OK
16	136x256 (144p) - webm		OK
17	136x256 (144p) – mp4		OK

*Bảng 1. Trường hợp kiểm thử cho lưu video*

#### 4.2.2. Lưu dưới dạng âm thanh

Test case	Trường hợp	Mô tả trường hợp test	Kết quả kiểm thử
1	Audio (tiny) – webm (1.6 MB)	Lưu âm thanh (nhấn Tải về PC)	OK
2	Audio (tiny) – webm (2.1 MB)		OK
3	Audio (tiny) – m4a (3.7 MB)		OK
4	Audio (tiny) – webm (4.1 MB)		OK

*Bảng 2. Trường hợp kiểm thử cho lưu âm thanh*

#### 4.3. Kết luận

Cùng với sự nỗ lực và hướng dẫn nhiệt tình của giảng viên hướng dẫn. Nhóm đã thành công ứng dụng công nghệ Python, cụ thể là thư viện Django, để tạo ra một chương trình tải về các định dạng video từ nền tảng Youtube.

Hướng phát triển: Tuy đã hoàn thành nhưng vẫn còn nhiều thiếu sót, nhóm đã rút ra kinh nghiệm và sẽ là tiền đề cho nhóm phát triển các dự án lớn hơn trong tương lai.

## Phân công công việc nhóm

**Nhóm:** Neko (nhóm 5)

Tên thành viên	Nội dung công việc	Đánh giá hoàn thành	Nhận xét công việc
Trịnh Hoàng Tùng	Lên và phân chia kế hoạch làm việc nhóm; viết source (phần xử lý); quay và dựng video demo; đại diện trình bày báo cáo dự án.	10/10	Hoàn thành tốt
Nguyễn Trịnh Thành	Lên ý tưởng dự án; Viết & thiết kế nội dung Word (Chương 1), PowerPoint & thiết kế giao diện source (trang hướng dẫn).	10/10	Hoàn thành tốt
Phạm Quốc Anh Quân	Viết; thiết kế nội dung Word (Chương 2), PowerPoint & Soạn nội dung kịch bản báo cáo & thiết kế giao diện source (trang index, about).	10/10	Hoàn thành tốt
Lê Quốc Đạt	Kiểm thử source → Viết Word (Chương 3 & 4); & Kiểm tra lỗi Word (Hoàn thiện và kiểm tra bố cục, chính tả) và kiểm tra PowerPoint.	10/10	Hoàn thành tốt

## **Mục lục hình ảnh và bảng biểu**

Hình 1.2.1.1. Thư viện Django

Hình 1.2.2.1. Thư viện Asgiref

Hình 1.2.4.1. Ví dụ ứng dụng thư viện Youtube-dl

Hình 1.2.4.2. Kết quả ứng dụng thư viện

Hình 2.1.1. Cấu trúc một web-app Django

Hình 2.1.2. Khởi tạo môi trường ảo (venv)

Hình 2.1.3. Cài đặt thư viện Django trong môi trường ảo Python

Hình 2.1.4. Khởi tạo dự án Django

Hình 2.1.5. Lần đầu chạy thử trên localhost

Hình 2.2.1. Tích hợp Youtube-dl

Hình 2.2.2. Regex đường dẫn video

Hình 2.2.3. Kiểm tra tính chính xác của đường dẫn

Hình 2.2.4. Phân loại video

Hình 2.2.5. Xử lý dữ liệu

Hình 2.3.1. Form nhập đường dẫn

Hình 2.3.2. Hiển thị nội dung

Hình 3.2.1. Demo trên localhost

Hình 4.2.1. Kết quả kiểm thử

Bảng 1. Trường hợp kiểm thử cho lưu video

Bảng 2. Trường hợp kiểm thử cho lưu âm thanh

## Tài liệu tham khảo

1. Youtube-dl là gì? <https://viblo.asia/p/tai-va-upload-video-tu-dong-len-youtube-voi-youtube-dl-va-youtube-api-bai-1-gioi-thieu-ve-youtube-dl-tai-video-tu-dong-gEmzOxEAGpv>
2. Django là gì? <https://aws.amazon.com/vi/what-is/django/>
3. Django documents <https://www.djangoproject.com/start/overview/>
4. Typing-extension documents <https://pypi.org/project/typing-extensions/>
5. Request Lifecircle <https://viblo.asia/p/tim-hieu-ve-request-lifecycle-rQOvPKzZkYj>
6. Youtube link regular expression <https://stackoverflow.com/questions/3717115/regular-expression-for-youtube-links>
7. Python regex convert youtube url to youtube video – <https://stackoverflow.com/questions/4705996/python-regex-convert-youtube-url-to-youtube-video>
8. How do I download a specific section of a video – <https://stackoverflow.com/questions/73516823/using-yt-dlp-in-a-python-script-how-do-i-download-a-specific-section-of-a-video>
9. Youtube-dl output template – <https://github.com/yt-dl-org/youtube-dl#output-template>
10. Error – Unable to extract uploader\_id – <https://github.com/yt-dl-org/youtube-dl/issues/31563> (<https://github.com/yt-dl-org/youtube-dl/issues/31563#issuecomment-1435954775>)