



# Báo cáo về Bài tập cuối kỳ

## 📘 Tổng quan dự án

### ▼ Thông tin cá nhân 🔎

Người viết: Trần An Thắng

MSSV: 22022525

Ngày viết: 28/11/2023

### ▼ Giới thiệu 📚

- Đây là báo cáo bài tập cuối kì lớp học phần “Lập trình xử lý dữ liệu với Python” ([AIT2003 1](#)).
- Trong dự án này, tôi cần sử dụng các công cụ thu thập nội dung từ các trang MXH Facebook, thu thập dữ liệu của một/nhiều trang và phân tích các dữ liệu thu thập được.
- Notion để dễ dàng hơn trong việc xem báo cáo:

📄 [Báo cáo về Bài tập cuối kỳ](#)

### ▼ Tổng quan 💬

- Fanpage Facebook tôi chọn : [Genshin Impact](#) , page hiện có 2,9 triệu lượt thích và 540 nghìn người theo dõi vào thời điểm viết báo cáo.

<https://www.facebook.com/GenshinImpact.vi>

- Tôi sẽ sử dụng thư viện `facebook_scrapers` để thực hiện thao tác lấy dữ liệu từ page, `pandas` để tổ chức dữ liệu và các thư viện dùng để trực quan hóa dữ liệu như `seaborn`, `matplotlib` cùng các thư viện phụ trợ khác.

## 📘 Nội dung chi tiết dự án

### ▼ Đôi lời muốn nói trước khi vào phần chính

- Dữ liệu được thu vào ngày 24/11/2023 vào khoảng 14h, nên có thể không đúng với thời điểm hiện tại hoặc trong tương lai
- Những phần được đánh số thứ tự trong phần Nội dung chi tiết dự án sau đây được đánh số thứ tự giống với phần Code với Jupyter Notebook được gắn link đi kèm
- Tổng quan về tựa game Genshin Impact:

- là trò chơi nhập vai hành động thế giới mở do miHoYo phát triển
- là trò chơi miễn phí và thu lợi nhuận qua cơ chế gacha, nơi người chơi có thể nhận nhân vật về đội hoặc vũ khí mới.
- Là một người chơi tựa game Genshin Impact, trong những phần diễn giải các phân tích dữ liệu dưới đây có thể có những ý kiến chủ quan của bản thân tôi.

Dự án được hoàn thành bằng ngôn ngữ lập trình Python trên Jupyter Notebook và sẽ được chia thành 3 phần:

- Thu thập dữ liệu
- Xử lý dữ liệu
- Phân tích và trực quan hóa dữ liệu đồng thời đưa ra các lời diễn giải

## ▼ 1. Thu thập dữ liệu

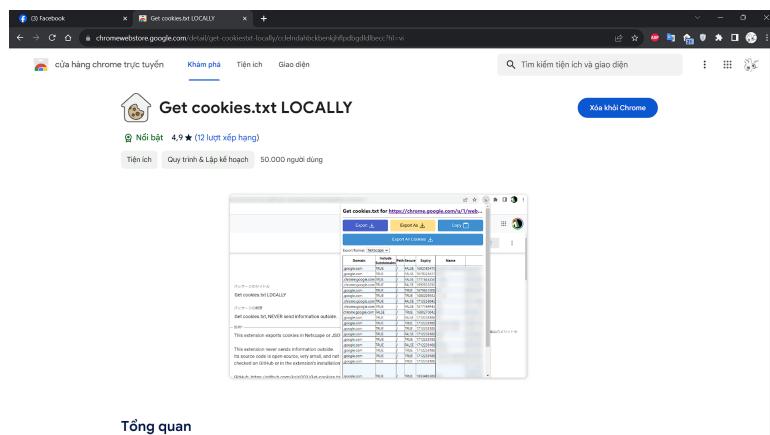
### ▼ 1.1. Cài đặt các thư viện cần thiết và chuẩn bị

- Bước 1 : Vì tôi dùng facebook\_scraper để lấy dữ liệu nên phải cài đặt thư viện tương ứng

```
%pip install facebook_scraper
```

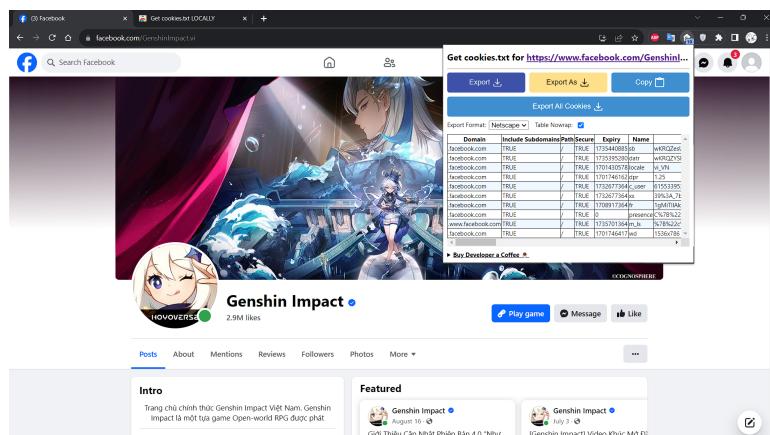
- Bước 2 : Thêm tiện ích Get cookies.txt LOCALLY và lấy cookies tài khoản cào dữ liệu

Get cookies.txt LOCALLY



Tổng quan

Lấy cookies về



## ▼ 1.2. Thu thập dữ liệu bằng facebook\_scraping

Thực hiện cào dữ liệu cuộn trên 10 trang đầu của page với số page là 30

```
FANPAGE_LINK = "GenshinImpact.vi"
FOLDER_PATH = "Data/"
COOKIE_PATH = "Data/cookies.txt"

PAGES_NUMBER = 30 # Number of pages to crawl

start_url = None
set_cookies(COOKIE_PATH)

def handle_pagination_url(url):
    global start_url
    start_url = url

post_list = []
while True:
    try:
        posts = get_posts(FANPAGE_LINK,
                          options={"comments": True, "reactions": True, "allow_extra_requests": True, "posts_per_page": 500},
                          extra_info= True, pages=PAGES_NUMBER, start_url=start_url, request_url_callback = handle_pagination_url)
        for post in posts:
            post_list.append(post)
            print(post.get('post_url'))
        print('Done')
        break
    except exceptions.TemporarilyBanned:
        print('Banned')
        time.sleep(600)
```

Ta thu được 300 posts khác nhau

```
1 # Số lượng bài viết thu được (Dữ liệu thô)
2 print(f"Số lượng bài viết: {len(post_list)}")
```

Số lượng bài viết: 300

## ▼ 1.3. Lưu dữ liệu thô về

```
# Initialize dataframe to scrape Facebook post
post_df_full = pd.DataFrame(columns=post_list[0].keys(), index=range(len(post_list)), data=post_list)

# To df
path=FOLDER_PATH + FANPAGE_LINK + ".csv"
post_df_full.to_csv(path, index=False)
print(path)
```

## ▼ 2. Xử lý dữ liệu

### ▼ 2.1. Cài đặt thư viện và chuẩn bị dữ liệu thô sang dạng DataFrame để dễ xử lý

### ▼ 2.2. Xử lý dữ liệu

#### ▼ 2.2.1. Xóa các bài viết bị lỗi thu dữ liệu

Sau khi kiểm tra và nhận thấy, những dữ liệu về bài viết bị lỗi thu thập thường bị tình trạng trong trường 'comments\_full' chứa chuỗi không có độ dài hoặc là trong trường 'reaction\_count' chứa giá trị toàn 0

```

Index([123, 124, 125, 126, 127, 128, 129, 130, 131, 132,
      ...
     290, 291, 292, 293, 294, 295, 296, 297, 298, 299],
      dtype='int64', length=177)

```

Ta nhận thấy từ bài viết có index = 123 trở đi bị lỗi thu dữ liệu, để đảm bảo tính liền mạch ta xóa từ bài viết có index = 123 trở đi và giữ nguyên những bài viết đầu

```
cooked_df = raw_df.drop(index=range(123, raw_df.index.max() + 1))
```

#### ▼ 2.2.2. Xóa các trường dữ liệu không chứa bất cứ dữ liệu nào

```
cooked_df = cooked_df.dropna(axis=1, how='all')
```

#### ▼ 2.2.3. Lọc ra các trường dữ liệu cần thiết

```
cooked_df = cooked_df[['post_url', 'post_text', 'time', 'image', 'video', 'comments', 'shares', 'user_id', 'comments_full', 'reac
```

#### ▼ 2.2.4. Xử lí dữ liệu với trường ‘reactions’

- Thêm các trường cảm xúc riêng

```

df1 = cooked_df
df1['reactions'] = df1['reactions'].apply(lambda x : dict(eval(x)))
df2 = df1['reactions'].apply(pd.Series)

cooked_df = pd.concat([df1, df2], axis=1).drop('reactions', axis=1)

```

- Sửa số lượng của các cột cảm xúc không có số lượng cảm xúc (giá trị NaN) thành số lượng là 0 và sửa kiểu dữ liệu

```

for i in range(0, len(cooked_df['thich'].values), 1):
    if pd.isna(cooked_df['thich'].values[i]):
        cooked_df['thich'].values[i] = 0

    if pd.isna(cooked_df['yêu thích'].values[i]):
        cooked_df['yêu thích'].values[i] = 0

    if pd.isna(cooked_df['haha'].values[i]):
        cooked_df['haha'].values[i] = 0

    if pd.isna(cooked_df['wow'].values[i]):
        cooked_df['wow'].values[i] = 0

    if pd.isna(cooked_df['buồn'].values[i]):
        cooked_df['buồn'].values[i] = 0

    if pd.isna(cooked_df['phẫn nộ'].values[i]):
        cooked_df['phẫn nộ'].values[i] = 0

    if pd.isna(cooked_df['thương thương'].values[i]):
        cooked_df['thương thương'].values[i] = 0

    if pd.isna(cooked_df['shares'].values[i]):
        cooked_df['shares'].values[i] = 0

    if pd.isna(cooked_df['comments'].values[i]):
        cooked_df['comments'].values[i] = 0

cooked_df['thich'] = cooked_df['thich'].astype(int)
cooked_df['yêu thích'] = cooked_df['yêu thích'].astype(int)
cooked_df['haha'] = cooked_df['haha'].astype(int)
cooked_df['wow'] = cooked_df['wow'].astype(int)
cooked_df['thương thương'] = cooked_df['thương thương'].astype(int)

```

```
cooked_df['buồn'] = cooked_df['buồn'].astype(int)
cooked_df['phản nộ'] = cooked_df['phản nộ'].astype(int)
```

### ▼ 2.2.5. Xử lí dữ liệu với trường 'time'

Thêm cột thể hiện giờ đăng bài trong ngày để thuận tiện cho việc xử lí

```
cooked_df['time'] = cooked_df['time'].apply(lambda x: datetime.strptime(x, "%Y-%m-%d %H:%M:%S"))
df = cooked_df['posted_hour'] = cooked_df['time'].apply(lambda x: x.hour)
```

## ▼ 2.3. Tổng kết

- So sánh dữ liệu sau khi được xử lí với dữ liệu thô chưa được xử lí

Dữ liệu chưa được xử lí				Dữ liệu đã được xử lí			
#	Column	Non-Null Count	Dtype	#	Column	Non-Null Count	Dtype
0	post_id	300	non-null int64	0	post_url	123	non-null object
1	text	289	non-null object	1	post_text	123	non-null object
2	post_text	289	non-null object	2	time	123	non-null datetime64[ns]
3	shared_text	68	non-null object	3	image	91	non-null object
4	original_text	1	non-null object	4	video	12	non-null object
5	time	300	non-null object	5	comments	123	non-null int64
6	timestamp	300	non-null int64	6	shares	123	non-null int64
7	image	227	non-null object	7	user_id	123	non-null int64
8	image_lowquality	300	non-null object	8	comments_full	123	non-null object
9	images	300	non-null object	9	reaction_count	123	non-null int64
10	images_description	300	non-null object	10	thích	123	non-null int32
11	images_lowquality	300	non-null object	11	yêu thích	123	non-null int32
12	images_lowquality_description	300	non-null object	12	haha	123	non-null int32
13	video	30	non-null object	13	wow	123	non-null int32
14	video_duration_seconds	0	non-null float64	14	thương thương	123	non-null int32
15	video_height	0	non-null float64	15	buồn	123	non-null int32
16	video_id	30	non-null float64	16	phản nộ	123	non-null int32
17	video_quality	0	non-null float64	17	posted_hour	123	non-null int64
18	video_size_MB	0	non-null float64				
19	video_thumbnail	30	non-null object				
...							
51	was_live	300	non-null bool				
52	fetched_time	123	non-null object				
	dtypes:	bool(3), float64(13), int64(7), object(30)					
	memory usage:	118.2+ KB					

Dữ liệu chưa được xử lí

Dữ liệu đã được xử lí

```
[16] 1 print(f"Kích cỡ dữ liệu thô : {raw_df.shape[0]} x {raw_df.shape[1]}")
      ✓ 0.0s
```

... Kích cỡ dữ liệu thô : 300 x 53

```
[17] 1 print(f"Kích cỡ dữ liệu đã được xử lí : {cooked_df.shape[0]} x {cooked_df.shape[1]}")
      ✓ 0.0s
```

... Kích cỡ dữ liệu đã được xử lí : 123 x 18

**Nhân xét :** Như vậy dữ liệu sau khi xử lí đã được thu nhỏ về cả kích cỡ lẫn bộ nhớ, đồng thời chứa nhiều dữ liệu hơn trong việc phân tích và đánh giá

- Lưu dữ liệu đã được xử lí

```
cooked_df.to_csv('Data/GenshinImpact.vi_modified.csv', index=False)
```

### ▼ 3. Phân tích và trực quan hóa dữ liệu đồng thời đưa ra các lời diễn giải

#### ▼ 3.1. Phân tích về reactions của bài viết

##### ▼ 3.1.1. Cài đặt thư viện và chuẩn bị dữ liệu

##### ▼ 3.1.2. Tìm ra bài viết có lượng tương tác nổi bật

###### ▼ 3.1.2.1. Tìm ra bài viết có lượng tương tác lớn nhất (nhiều reaction/ nhiều comment/ nhiều share)

- Trong hợp thay, bài viết có nhiều lượng reactions nhất cũng chính là bài viết có lượng chia sẻ nhiều nhất:

Genshin Impact - Chúc mừng sinh nhật, Kaebedara Kazuha



Giải thích: Kazuha vừa là 1 nhân vật mạnh mẽ trong trò chơi, vừa 1 Ronin Samurai có tính cách hoà nhã, khiêm tốn, vừa có lối sống lang bạt khắp chốn nơi (Theo cốt truyện). Đi kèm theo đó là tạo hình dễ nhìn, ưa mắt khiến cho nhiều người ưa thích, đặc biệt là các fangirl.

⇒ Sự bùng nổ về lượt tương tác và share là yếu tố tất nhiên

- Bài viết có lượng comments nhiều nhất:

Genshin Impact - Các Nhà Lữ Hành thân mến, chương trình đặc biệt



Giải thích: Nội dung bài viết thông báo về việc tổ chức một chương trình hé lộ một phần thông tin phiên bản mới

⇒ Nhiều người chơi hóng chờ thông tin phiên bản mới khiến cho dân tình sôi nổi, điều đó dẫn đến việc bùng nổ số lượng comments

###### ▼ 3.1.2.2. Tìm ra bài viết có lượng cảm xúc tích cực lớn nhất (thích / yêu thích / haha / wow / thương thương)

- Bài viết có lượng like (thích) nhiều nhất

Genshin Impact - Chúc mừng sinh nhật, Sayu!



*Giải thích* : Một nhân vật cute, dễ thương như Sayu thì tại sao đến dịp sinh nhật mà mình không cho 1 like nhỉ?  
⇒ Điều đó dẫn đến lượng like của bài viết tăng lên đáng kể

- Bài viết có lượng yêu thích nhiều nhất đồng thời là bài viết có lượng cảm xúc thương yêu nhiều nhất và là bài viết có lượng reactions đi kèm với lượng share nhiều nhất

Genshin Impact - Chúc mừng sinh nhật, Kaedehara Kazuha



*Giải thích* : Bổ sung cho phần phân tích trên, ta có thể nhận thấy rằng chủ yếu các cảm xúc 'thương thương' và 'yêu thích' chủ yếu đến từ phía các chị em, các fangirl ⇒

- Bài viết có lượng haha nhiều nhất

Genshin Impact sẽ gỡ khỏi CH Play và App Store khu vực Việt Nam

- Bài viết có lượng wow nhiều nhất

Genshin Impact - "...Đội trưởng Chevreuse, một lần nữa tôi xin...



Giải thích: Đây là 1 bài viết giới thiệu về nhân vật mới không có báo từ trước . ⇒ Khiến cho mọi người vô cùng ngạc nhiên và trầm trồ

▼ 3.1.2.3. Tìm ra bài viết có lượng cảm xúc tiêu cực lớn nhất (buồn/ phẫn nộ)

- Bài viết có lượng buồn nhiều nhất cũng chính là bài viết có lượng phẫn nộ nhiều nhất

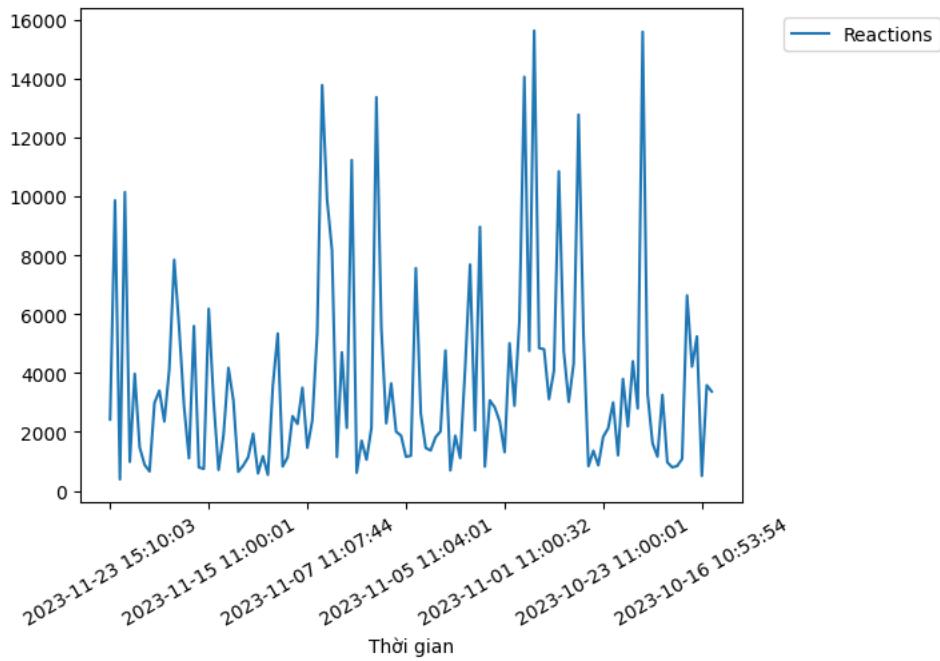
Genshin Impact sẽ gỡ khỏi CH Play và App Store khu vực Việt Nam

Giải thích: Bài viết trên có nội dung đề cập đến việc ứng dụng Genshin Impact trên nền tảng Android và IOS đột nhiên đổi nhà phát hành nên phải xóa phiên bản ban đầu và thay đổi bằng phiên bản khác

⇒ Khiến cho người chơi (nhất là người dùng trên hệ điều hành Android và IOS) vô cùng bức xúc và thất vọng. Điều này khiến chiều hướng dư luận trở nên xấu đi.

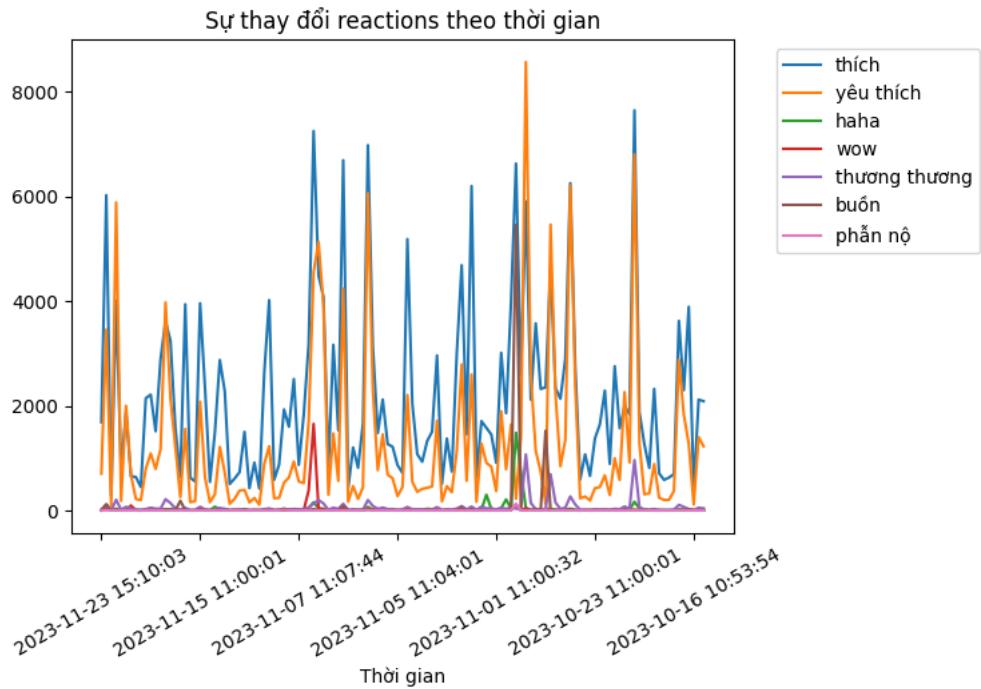
▼ 3.1.3. Phân tích về reactions

▼ 3.1.3.1 Sự thay đổi tổng số reactions theo thời gian

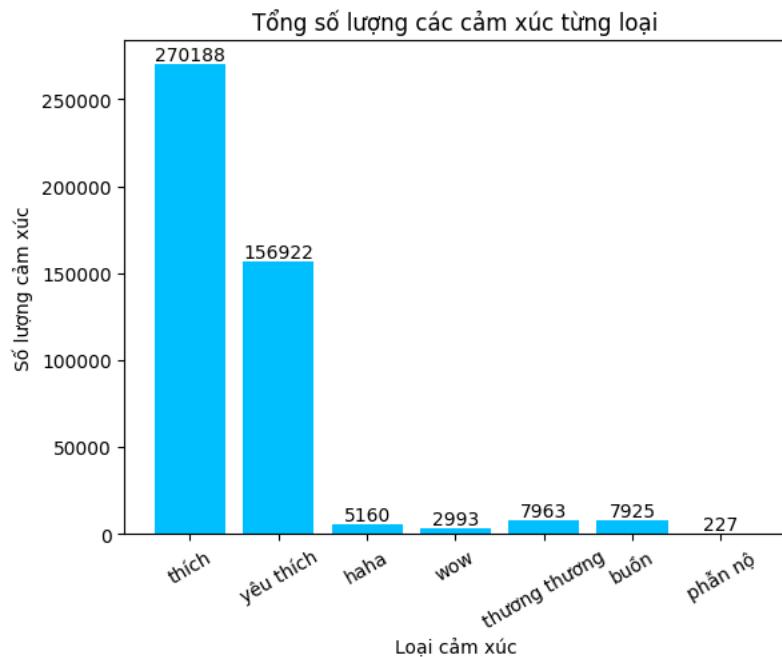


**Nhận xét :** Lượng reactions của page không ổn định theo thời gian nhưng tăng giảm theo chu kỳ

#### ▼ 3.1.3.2 Sự thay đổi từng cảm xúc theo thời gian



#### ▼ 3.1.3.3 Tổng số lượng các cảm xúc các loại

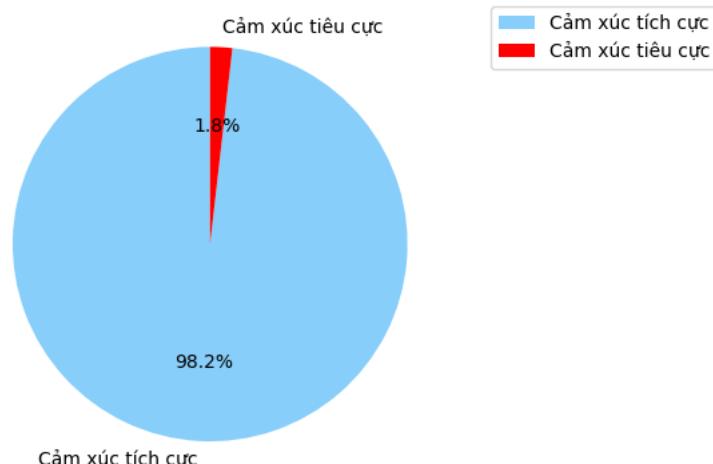


**Nhận xét :** Đa số các reactions là 'yêu thích' và 'thích'

⇒ Genshin Impact được mọi người dành tình cảm yêu quý khá nhiều

#### ▼ 3.1.3.4 Tương quan giữa số lượng cảm xúc tiêu cực và số lượng cảm xúc tích cực

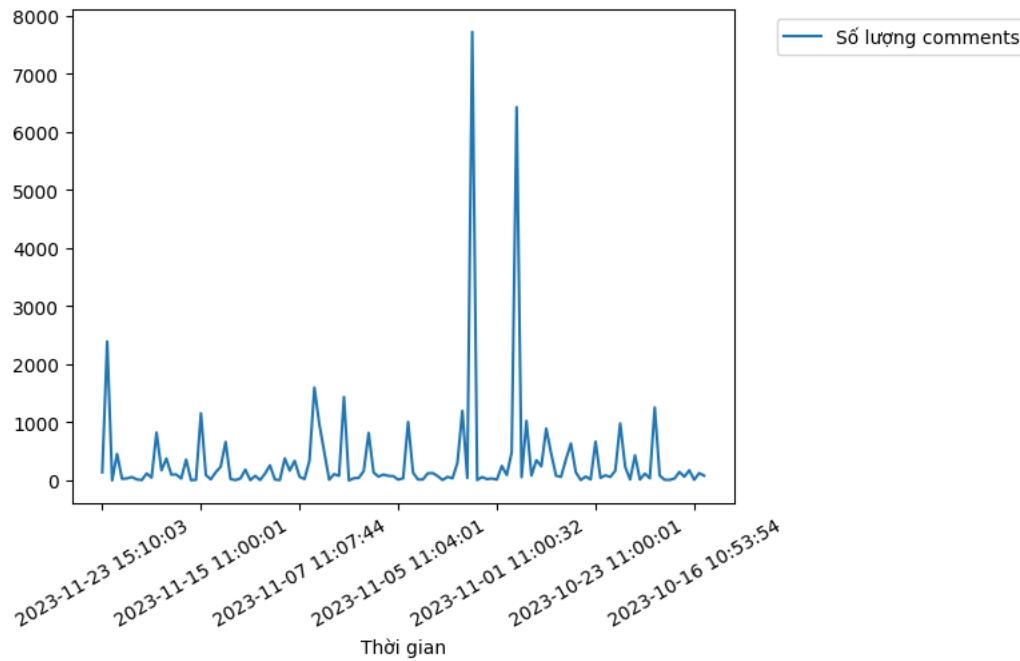
- Cảm xúc tích cực ( thích / yêu thích / haha / wow / thương thương )
- Cảm xúc tiêu cực ( buồn / phẫn nộ )



**Nhận xét :** Đa số lượt cảm xúc (98.2%) là cảm xúc tích cực ⇒ Lập luận "Genshin Impact được mọi người dành tình cảm yêu quý khá nhiều" càng được củng cố

### ▼ 3.2. Phân tích về comments và nội dung của bài viết

- ▼ 3.2.1. Cài đặt thư viện và chuẩn bị dữ liệu
- ▼ 3.2.2. Sự thay đổi số lượng comments theo thời gian



**Nhận xét :** Ta nhận thấy rằng trong khoảng thời gian mà dữ liệu thu được thể hiện có 2 mốc thời gian mà lượng comments tăng đột biến

- Bài viết thu nhiều comments nhất (Đã được đề cập ở phần phân tích trên) :

[Genshin Impact - Các Nhà Lữ Hành thân mến, chương trình đặc biệt](#)

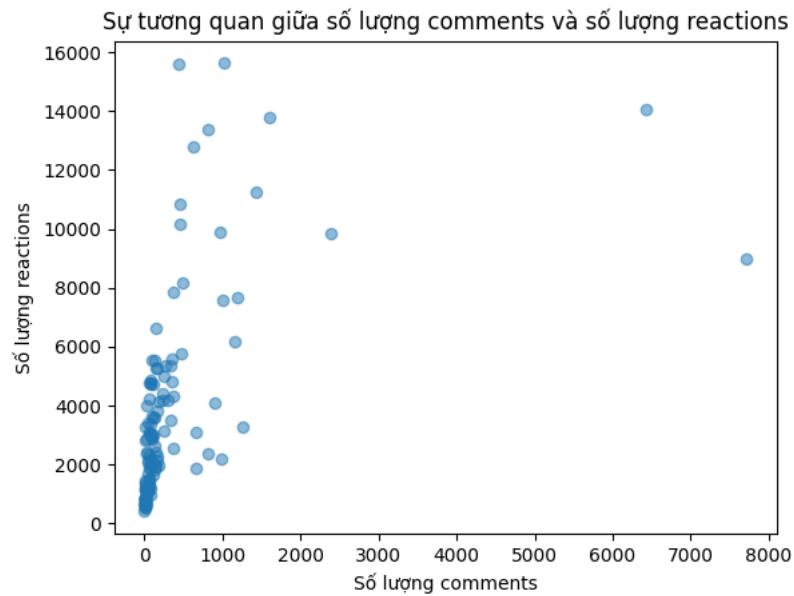


- Bài viết thu nhiều comments thứ hai (Đã được đề cập là bài viết có nhiều cảm xúc tiêu cực nhất ở trên) :

[Genshin Impact sẽ gỡ khỏi CH Play và App Store khu vực Việt Nam](#)

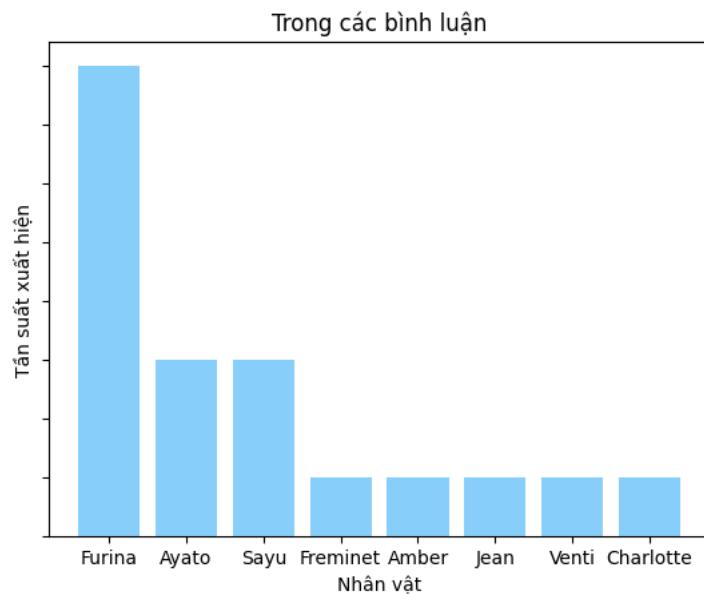
**Giải thích :** Cả hai bài viết đều khiến cho làn sóng dư luận nổi lên. Nhưng một bên là gợi lên làn sóng tích cực, một bên gợi lên làn sóng tiêu cực

### ▼ 3.2.3. Tương quan giữa số lượng comments và số lượng reactions



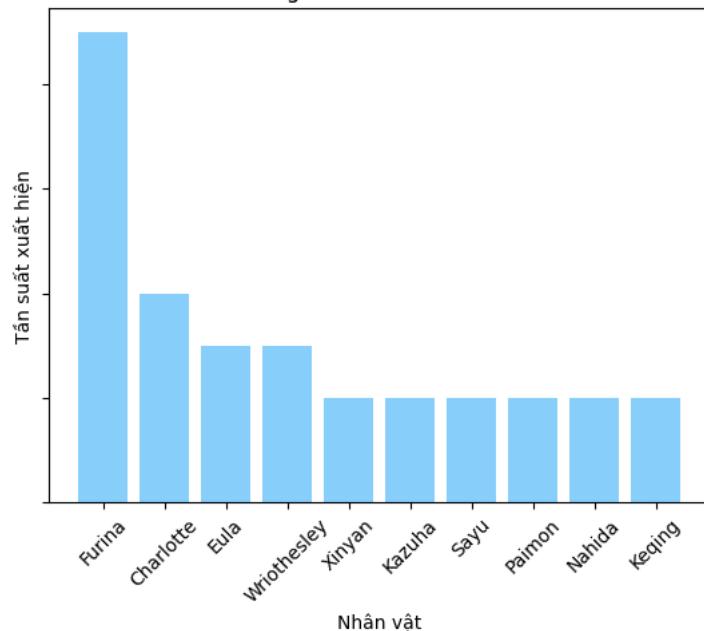
▼ 3.2.4 Tìm kiếm tên nhân vật được nhắc đến nhiều nhất (trong các bình luận/ trong tiêu đề các bài viết)

- 3.2.4.1 Thu dữ liệu tên nhân vật từ trang [Characters - Paimon.moe](#)
- ▼ 3.2.4.2 Nhân vật được nhắc đến nhiều trong các bình luận



- ▼ 3.2.4.3 Nhân vật được nhắc đến nhiều trong tiêu đề các bài viết

### Trong các tiêu đề bài viết



#### Nhân xét :

- Nhân vật được nhắc đến nhiều nhất là Furina
- Còn có các nhân vật khác như : Ayato, Charlotte, Wriothesley được nhắc đến khá nhiều
- Ngoài ra còn xuất hiện một số nhân vật nổi bật như : Eula, Sayu

#### Giải thích :

- Furina được nhắc đến nhiều nhất bởi lẽ :
    - Cô ấy là một nhân vật thần thánh mang sức mạnh đáng kể
    - Cô ấy có cốt truyện khá buồn, bi đát, đáng thương khiến cho mọi người đồng cảm
    - Tính cách có phần trẻ con nhưng nội tâm sâu sắc
    - ..... Rất nhiều lí do khác nhau nữa
- ⇒ Furina được cộng đồng đông đảo người chơi yêu mến



- Ayato, Charlotte, Wriothesley được nhắc đến khá nhiều bởi lẽ 3 nhân vật này đang được nhà phát hành mở ra để cho người chơi lấy về đội hình của mình

- Eula, Sayu được nhắc đến bởi lẽ 2 nhân vật này trong khoảng thời gian dữ liệu thu thập thể hiện đã diễn ra sự kiện chúc mừng sinh nhật. Đồng thời đây là 2 nhân vật có lượng người yêu mến khá đông đảo bởi vẻ ngoài và tính cách của mình



#### ▼ 3.2.5. Các từ ngữ nổi bật (Trong Tiêu đề bài viết/ Trong các bình luận)

#### ▼ 3.2.5.1. Các từ ngữ nổi bật trong các bình luận

**Nhận xét**: Từ hình ảnh trên, thông qua từ ngữ ta có thể thấy được cảm xúc của người chơi Genshin Impact dành cho trò chơi chủ yếu là các cảm xúc tích cực, yêu mến =)). Đồng thời thấy được tính chất simp lord của những con dân wibu chúa.

#### ▼ 3.2.5.2. Các từ ngữ nổi bật trong các tiêu đề bài viết



### ▼ 3.3. Phân tích về fanpage

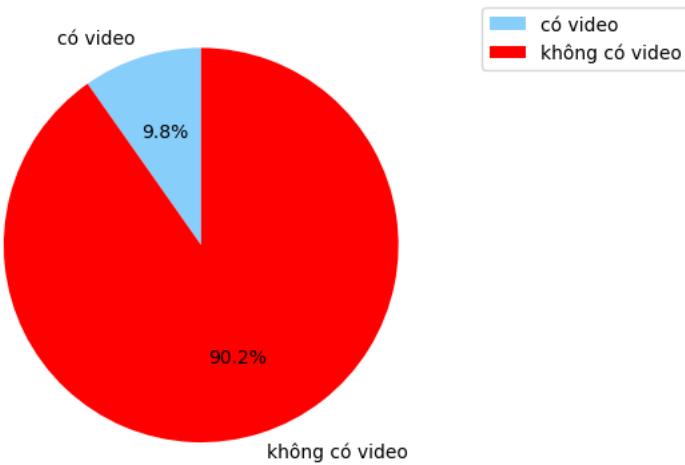
- 3.3.1. Cài đặt thư viện và chuẩn bị dữ liệu

#### ▼ 3.3.2. Tỉ lệ bài viết có hình ảnh đi kèm hoặc video đi kèm

##### ▼ 3.3.2.1. Tỉ lệ bài viết có hình ảnh đi kèm



##### ▼ 3.3.2.2. Tỉ lệ bài viết có video đi kèm

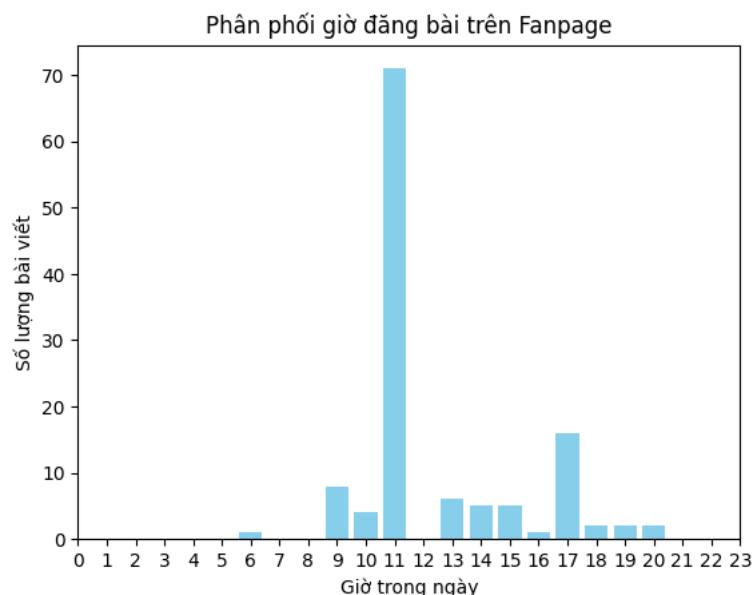


**Nhận xét :** Fanpage thường đăng bài với các hình ảnh/ video đi kèm

⇒ Điều này khiến cho Fanpage ngập tràn màu sắc hơn, thú vị hơn. Từ đó hấp dẫn và thu hút được người dùng hơn, bởi lẽ chủ yếu người dùng trên Fanpage là người trẻ tuổi.

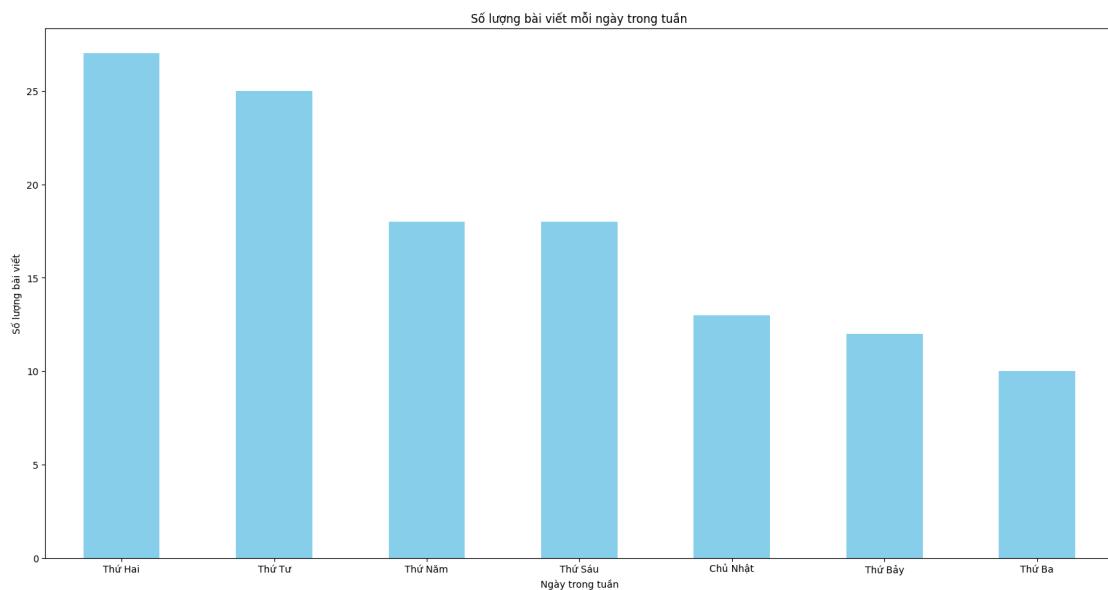
### ▼ 3.3.3. Page với thời gian đăng bài

#### ▼ 3.3.3.1 Các mốc thời gian trong ngày mà Fanpage thường xuyên đăng bài



**Nhận xét :** Máy chủ Genshin Impact được đặt tại Trung Quốc (GMT+8) cho nên từ dữ liệu trên, tại Việt Nam bài viết được đăng nhiều nhất vào khung giờ từ 11h → 12h thì bên Trung Quốc, bài viết nguồn sẽ được đăng vào khoảng 12h → 13h giờ Việt Nam (GMT +7).

#### ▼ 3.3.3.2 Các mốc thời gian trong tuần mà Fanpage thường xuyên đăng bài



Một số bài viết được đăng vào Thứ Hai:

- [Genshin Impact - Chúc mừng sinh nhật, Keqing!](#)
- [Genshin Impact - Chúc mừng sinh nhật, Keqing!](#)
- [Genshin Impact - \[Genshin Impact\] Hành Trình Theo Đuổi Ánh Sáng](#)
- [Genshin Impact - Chi Tiết Cập Nhật Online Client Android](#)
- [Genshin Impact - Chevreuse - Người Thực Thi Công Lý Đội Trưởng Đội](#)
- [Genshin Impact - "...Đội trưởng Chevreuse](#)
- [Genshin Impact - Navia - Bánh Lái Hoa Lê Hội Trưởng](#)
- [Genshin Impact - "Khi còn nhỏ, chúng tôi từng chơi một trò chơi....](#)
- [Genshin Impact - \[Sự Kiện Có Thường\] Sự Kiện Thu Thập Fan Art](#)
- [Genshin Impact - \[Genshin Impact\] Sự kiện web "Thám Tử Đặc Biệt"](#)

## Tổng kết

### ▼ Đôi lời muốn nói

Dự án được hoàn thiện chính thức vào ngày 28/11/2023, trong quá trình hoàn thiện dự án, tôi đã gặp rất nhiều khó khăn. May mắn rằng dự án vẫn được hoàn thiện, mong nhận được nhận xét từ thầy và các bạn! Xin cảm ơn.

### ▼ Links

- Notion để dễ dàng hơn trong việc xem báo cáo:

 [Báo cáo về Bài tập cuối kỳ](#)

- GitHub với đầy đủ mã code:

<https://github.com/IAI-UET/final-project-angWindy>

### ▼ Trích dẫn nguồn

- Dữ liệu fanpage

- [GenshinImpact - Facebook](#)
- Tổng quan về tựa game Genshin Impact
  - [Genshin Impact – Wikipedia tiếng Việt](#)
  - [Genshin Impact - Đi vào thế giới thần kỳ đầy mạo hiểm. \(hoyoverse.com\)](#)
- Dữ liệu về tên nhân vật Genshin Impact
  - [Characters - Paimon.moe](#)
- Cốt truyện nhân vật Genshin Impact
  - [Cốt Truyện | Wiki Genshin Impact | Fandom](#)