

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP THÁI NGUYÊN
KHOA ĐIỆN TỬ
BỘ MÔN: CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
LẬP TRÌNH PYTHON
ĐỀ TÀI: XÂY DỰNG HỆ THỐNG ĐỀ XUẤT PHIM

Giáo Viên Hướng Dẫn : Đỗ Duy Cốp

Họ và Tên : Nguyễn Thị Mỹ Duyên

MSSV : K205480106033

Lớp : K56KMT.01

Khoa : Điện Tử

Thái Nguyên 2024

PHIẾU GIAO ĐỀ TÀI MÔN LẬP TRÌNH PYTHON

Sinh viên: Nguyễn Thị Mỹ Duyên

MSSV :K205480106033

Lớp: K56KMT.01

Khoá : K56

Ngành học: Kỹ thuật máy tính.

Giáo viên hướng dẫn: Đỗ Duy Cốp

1. Tên đề tài: **Xây dựng hệ thống đề xuất phim**

2. Nội dung thực hiện:

- xây dựng cơ sở dữ liệu

- Bảng:Movies bao gồm :movie_id (pk) , title, release_date , overview , vote_average , vote_count .
- Stored Procedures (SP_): SP_ InsertMovie để lấy danh sách các bộ phim có điểm đánh giá cao nhất từ cơ sở dữ liệu.
- Thuật toán:
- Sử dụng Python và FastAPI gọi API để lấy dữ liệu từ trang web dịch vụ phim trực tuyến TMDB API.
- Xử lý dữ liệu.
- Lưu dữ liệu vào cơ sở dữ liệu MSSQL.
- Sử dụng Node-red
- Gọi API từ Python: Sử dụng node HTTP request để lấy dữ liệu
- Gọi stored procedure: Sử dụng node MSSQL để gọi các stored procedure, truyền tham số vào và lưu dữ liệu vào cơ sở dữ liệu.
- Web
- Xây dựng một ứng dụng web để hiển thị dữ liệu từ cơ sở dữ liệu.
- Hiển thị biểu đồ dữ liệu phim.

3. Ngày giao nhiệm vụ: 15/5/20204

4. Ngày hoàn thành: 25/5/2024

5. Giáo viên hướng dẫn: Đỗ Duy Cốp

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày....tháng.....năm.....

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN CHẤM

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày....tháng.....năm.....

GIÁO VIÊN HƯỚNG DẪN

MỤC LỤC

MỤC LỤC.....	4
LỜI NÓI ĐẦU	5
CHƯƠNG I. TỔNG QUÁT.....	6
1.1.Đặt vấn đề.....	6
1.2.Mục tiêu.....	6
1.2.Hướng giải quyết đề tài.....	7
CHƯƠNG II. CƠ SỞ LÝ THUYẾT	8
2.1. SQL Server Management Studio.....	8
2.2.Python	9
2.3.Pycham.....	10
2.4.Node-red.....	10
CHƯƠNG III.NỘI DUNG THỰC HIỆN.....	12
3.1.FASTAPI.....	12
3.2.NODE-RED.....	13
3.3.SQL	16
3.4.Giao diện	16
CHƯƠNG IV. KẾT LUẬN	20

LỜI NÓI ĐẦU

Công nghệ thông tin đang ngày càng trở nên phổ biến và tiện ích hơn trong cuộc sống hàng ngày của chúng ta. Trong thế giới giải trí, việc tìm kiếm và khám phá các bộ phim mới để xem có thể trở nên phức tạp đối với nhiều người, đặc biệt là trong thời đại của một lượng lớn thông tin trên internet.

Với ý tưởng xây dựng một hệ thống đề xuất phim, em đã đặt ra mục tiêu là tạo ra một công cụ hữu ích, giúp người dùng dễ dàng khám phá và lựa chọn các bộ phim phù hợp với sở thích của mình.

Trong đề tài này, em đã sử dụng một sự kết hợp giữa FastAPI, Node-RED và cơ sở dữ liệu SQL. FastAPI được chọn vì tính năng nhanh chóng và hiệu quả của nó trong việc xây dựng các API RESTful. Node-RED đóng vai trò quan trọng trong việc thiết lập các luồng công việc và tích hợp các thành phần khác nhau của hệ thống, từ việc gửi yêu cầu HTTP đến xử lý dữ liệu. Còn cơ sở dữ liệu SQL được sử dụng để lưu trữ dữ liệu một cách có tổ chức và dễ dàng truy xuất.

Em hy vọng rằng đề tài này sẽ mang lại giá trị và tiện ích cho cộng đồng người dùng, và sẽ mở ra cơ hội cho việc phát triển và mở rộng trong tương lai. Em xin gửi lời cảm ơn đến Thầy Đỗ Duy Cốp đã hỗ trợ em trong quá trình thực hiện đề tài này.

CHƯƠNG I. TỔNG QUÁT

1.1. Đặt vấn đề

Trong thời đại kỹ thuật số ngày nay, nhu cầu giải trí qua các nền tảng trực tuyến đã trở nên phổ biến và ngày càng gia tăng. Người dùng có vô số lựa chọn về các bộ phim và chương trình truyền hình, từ nhiều thể loại và quốc gia khác nhau. Tuy nhiên, với khối lượng nội dung khổng lồ, việc tìm kiếm một bộ phim phù hợp với sở thích cá nhân có thể trở nên khó khăn và tốn thời gian.

Do đó, việc phát triển một hệ thống đề xuất phim cho người dùng là cần thiết và hữu ích. Hệ thống này có thể giúp người dùng dễ dàng tìm thấy những bộ phim mà họ yêu thích, đồng thời hỗ trợ các nền tảng trực tuyến trong việc tăng cường trải nghiệm người dùng và giữ chân khách hàng.

Hiện tại, nhiều hệ thống đề xuất phim đã được triển khai, tuy nhiên, vẫn còn nhiều hạn chế như độ chính xác chưa cao, không thể cá nhân hóa sâu sắc theo từng người dùng, và không tận dụng hết các dữ liệu phong phú từ các nguồn khác nhau. Điều này đòi hỏi cần có một giải pháp mới, kết hợp các phương pháp hiện đại và công nghệ tiên tiến để xây dựng một hệ thống đề xuất phim hiệu quả hơn.

Với mục tiêu cải thiện khả năng quản lý và cung cấp thông tin cho người dùng, việc xây dựng một hệ thống thu thập, xử lý và hiển thị dữ liệu về phim toàn diện sẽ giúp nâng cao trải nghiệm giải trí và đáp ứng nhu cầu ngày càng đa dạng của người dùng.

1.2. Mục tiêu

Để giải quyết vấn đề này, cần thiết phải xây dựng một hệ thống đề xuất phim cho người dùng, thu thập, xử lý và hiển thị dữ liệu phim một cách hiệu quả và chính xác. Hệ thống này không chỉ giúp người dùng dễ dàng tìm kiếm và chọn lựa các bộ phim phù hợp với sở thích cá nhân mà còn hỗ trợ các nhà nghiên cứu và nền tảng trực tuyến trong việc phân tích xu hướng và dự báo nhu cầu giải trí.

Mục tiêu của đề tài này là phát triển một giải pháp toàn diện bao gồm các bước sau:

- Thu thập dữ liệu phim: Dữ liệu được thu thập từ các nguồn uy tín, được cập nhật liên tục để đảm bảo chính xác.
- Xử lý dữ liệu: Sử dụng FastAPI để xây dựng các API dịch vụ và Node-RED để tổ chức các luồng xử lý dữ liệu, giúp dữ liệu được chuẩn hóa.
- Lưu trữ dữ liệu: Dữ liệu sau khi được xử lý sẽ được lưu trữ vào cơ sở dữ liệu SQL, đảm bảo khả năng truy xuất và quản lý dữ liệu một cách hiệu quả.
- Xây dựng trang web: Một trang web thân thiện với người dùng sẽ được phát triển để hiển thị thông tin chi tiết về phim. Trang web sẽ bao gồm các biểu đồ trực quan giúp người dùng dễ dàng theo dõi phim.

Để đảm bảo tính chính xác và hiệu quả trong việc lưu trữ và hiển thị dữ liệu, chúng ta sẽ sử dụng một cơ sở dữ liệu SQL. Để đảm bảo tính chính xác và hiệu quả trong việc lưu trữ và hiển thị dữ liệu, chúng ta sẽ sử dụng một cơ sở dữ liệu SQL. Cơ sở dữ liệu này sẽ lưu trữ các thông tin liên quan đến phim, giúp việc truy xuất và phân tích dữ liệu trở nên

thuận tiện hơn. Ngoài ra, các biểu đồ trực quan sẽ được vẽ dựa trên dữ liệu từ cơ sở dữ liệu, giúp người dùng có cái nhìn tổng quan và dễ hiểu về xu hướng và đánh giá phim.

1.2.Hướng giải quyết đề tài

Để giải quyết vấn đề đề xuất phim cho người dùng và hiển thị thông tin một cách hiệu quả, chúng ta sẽ thực hiện các bước sau:

a. *Dùng request để lấy dữ liệu thô từ API và gửi nó lên endpoint của FastAPI*

Sử dụng Python để gửi các yêu cầu POST đến endpoint của FastAPI nhằm thu thập dữ liệu thô về phim. Thư viện Request trong Python sẽ hỗ trợ chúng ta trong việc này.

b. *ở NODE-RED dùng node http request lấy địa chỉ có chứa endpoint(thường là cổng 127.0.0.1:8000/xxx)*

Thiết lập một flow trong Node-RED để gửi HTTP request tới endpoint địa phương (localhost). Node-RED sẽ đóng vai trò trung gian, giúp ta gửi yêu cầu HTTP và nhận phản hồi từ endpoint.

c. *Lưu vào SQL*

Khi nhận được dữ liệu từ endpoint, chúng ta sẽ lưu trữ dữ liệu này vào cơ sở dữ liệu SQL. Sử dụng ORM (Object Relational Mapping) như SQLAlchemy trong Python để tương tác với cơ sở dữ liệu một cách dễ dàng và hiệu quả.

d. *Xây dựng giao diện người dùng lấy dữ liệu từ SQL để vẽ biểu đồ*

Sử dụng HTML, CSS, JavaScript để xây dựng giao diện người dùng. Dữ liệu từ SQL Server sẽ được truy xuất thông qua ASP.NET (api.aspx), cho phép hiển thị các biểu đồ trực quan.

CHƯƠNG II. CƠ SỞ LÝ THUYẾT

2.1. SQL Server Management Studio

SQL Server Management Studio (SSMS) là một ứng dụng phần mềm của Microsoft, được thiết kế để quản lý và tương tác với các cơ sở dữ liệu SQL Server. Được phát triển từ năm 2005, SSMS là một công cụ quản lý cơ bản và quan trọng cho các quản trị viên cơ sở dữ liệu, nhà phát triển và các chuyên gia dữ liệu.

SQL Server được phát triển lần đầu tiên vào năm 1989 bởi Microsoft, hợp tác với Sybase và Ashton-Tate. Từ đó, nó đã trải qua nhiều phiên bản cải tiến với những tính năng và khả năng mới, trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới.



SSMS cung cấp một giao diện người dùng đồ họa (GUI) thân thiện và dễ sử dụng cho việc quản lý cơ sở dữ liệu SQL Server. Giao diện này cho phép người dùng thực hiện các tác vụ quản lý dữ liệu một cách dễ dàng và hiệu quả. SSMS cung cấp một loạt các công cụ quản lý tích hợp, cho phép người dùng thực hiện các tác vụ như tạo, sửa đổi và xóa cơ sở dữ liệu, bảng, chỉ mục và thủ tục lưu trữ. Nó cũng cho phép quản trị viên sao lưu và phục hồi dữ liệu, kiểm tra và theo dõi hiệu suất, và quản lý bảo mật. SSMS cho phép người dùng thực hiện các truy vấn SQL và xem dữ liệu từ các bảng trong cơ sở dữ liệu. Nó cung cấp một trình soạn thảo truy vấn mạnh mẽ với tính năng gợi ý cú pháp và điều hướng thông minh giúp tăng hiệu suất lập trình.

SQL Server cung cấp cho người dùng các công cụ và tính năng để quản lý, lưu trữ, xử lý các truy vấn dữ liệu, kiểm soát truy cập, xử lý giao dịch và hỗ trợ tích hợp dữ liệu từ nhiều nguồn khác nhau.

Ngoài ra, SQL Server cũng cung cấp các công cụ để tạo báo cáo, phân tích và quản lý cơ sở dữ liệu trực quan thông qua giao diện người dùng hoặc các script lệnh SQL. SQL Server được xây dựng dựa trên SQL, một ngôn ngữ lập trình tiêu chuẩn để tương tác với

cơ sở dữ liệu quan hệ. SQL Server được liên kết với Transact-SQL hoặc T-SQL, triển khai SQL của Microsoft có bổ sung một tập hợp các cấu trúc lập trình độc quyền.

2.2.Python

Python là một ngôn ngữ lập trình đa mục đích, dễ học và mạnh mẽ, được phát triển bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Với cú pháp đơn giản và gần gũi với ngôn ngữ tự nhiên, Python là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới, được sử dụng rộng rãi trong các lĩnh vực khác nhau từ phát triển web, khoa học dữ liệu đến trí tuệ nhân tạo.



Python không chỉ dễ học mà còn linh hoạt và mở rộng, hỗ trợ nhiều phong cách lập trình và tích hợp tốt với các ngôn ngữ khác như C/C++, Java và .NET. Hệ sinh thái phong phú của Python cung cấp các thư viện và framework đa dạng, giúp lập trình viên dễ dàng phát triển các ứng dụng và dự án.

Điểm nổi bật của Python là cộng đồng lớn mạnh, với hàng triệu lập trình viên trên khắp thế giới, sẵn sàng chia sẻ kiến thức và kinh nghiệm. Nhờ vào điều này, Python không chỉ là một ngôn ngữ lập trình mà còn là một cộng đồng và một triển khai tri thức phong phú, đóng vai trò quan trọng trong việc giải quyết các thách thức hiện đại trong ngành công nghiệp và khoa học.

2.3.Pycharm

PyCharm là một phần mềm được phát triển bởi JetBrains, cung cấp các công cụ cần thiết giúp các lập trình viên Python tăng năng suất làm việc. Ngoài ra, PyCharm còn được tích hợp nhiều yếu tố mở rộng khác như: biên dịch mã, tô sáng cú pháp, điều hướng project nhanh chóng, công cụ cơ sở dữ liệu và trình soạn thảo văn bản có tích hợp lập trình nhằm mục đích thúc đẩy quá trình phát triển Website.

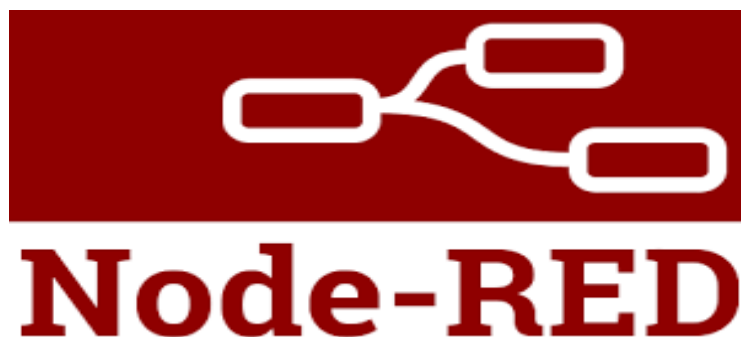


PyCharm cũng tương tự như Python, là một trong những IDE được nhiều người dùng ưa chuộng và sử dụng rộng rãi trong giới lập trình. Nhờ sự tiện lợi và hiệu quả cao, PyCharm đã được nhiều doanh nghiệp có tầm ảnh hưởng lớn sử dụng làm IDE Python như: Symantec, Twitter, Pinterest,...

Một trong những điểm mạnh của việc sử dụng PyCharm là khả năng cung cấp API cho các nhà phát triển, đồng thời cho phép họ viết các plugin riêng để mở rộng tính năng. Hơn nữa, phần mềm này còn tương thích với đa dạng các hệ điều hành như Linux, Windows và macOS. Điều này, giúp các lập trình viên Python tiết kiệm thời gian đáng kể trong quá trình phát triển một ứng dụng hay website nào đó.

2.4.Node-red

Node-RED là một công cụ mã nguồn mở được phát triển bởi IBM và cung cấp một giao diện trực quan để kết nối các thiết bị, dịch vụ và ứng dụng một cách linh hoạt và dễ dàng. Nó được xây dựng dựa trên Node.js và sử dụng một giao diện trực quan dựa trên trình duyệt để tạo, quản lý và triển khai các luồng làm việc (flow) dựa trên sự kết hợp của các "nút" và "luồng".



Các nút trong Node-RED đại diện cho các chức năng hoặc dịch vụ cụ thể, và chúng có thể được kéo và thả vào khung làm việc để tạo ra các luồng làm việc. Mỗi nút thường thực hiện một chức năng nhất định, từ xử lý dữ liệu đến gửi và nhận thông điệp qua các giao thức mạng khác nhau.

Node-RED được sử dụng rộng rãi trong Internet of Things (IoT) và trong các ứng dụng tự động hóa, nơi nó có thể giúp kết nối và tự động hóa các thiết bị và dịch vụ từ nhiều nhà sản xuất khác nhau. Nó cũng thích hợp cho việc xử lý dữ liệu thời gian thực và tích hợp các dịch vụ web khác nhau.

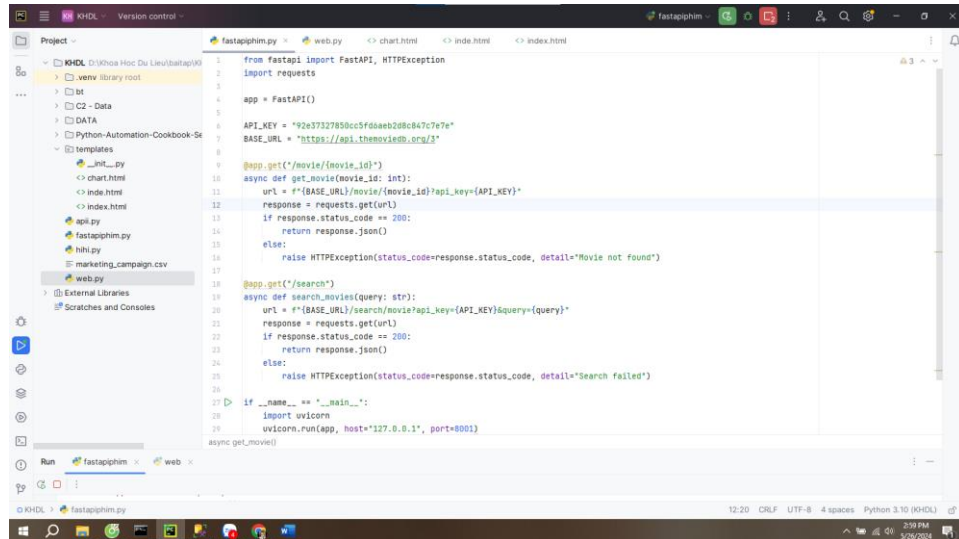
Node-RED cung cấp một cộng đồng lớn và sôi động, với nhiều nút và gói mở rộng được phát triển và chia sẻ miễn phí. Điều này giúp người dùng mở rộng và tùy chỉnh NodeRED theo nhu cầu và yêu cầu cụ thể của họ.

CHƯƠNG III. NỘI DUNG THỰC HIỆN

Sau đây là các bước thực hiện đề tài của em:

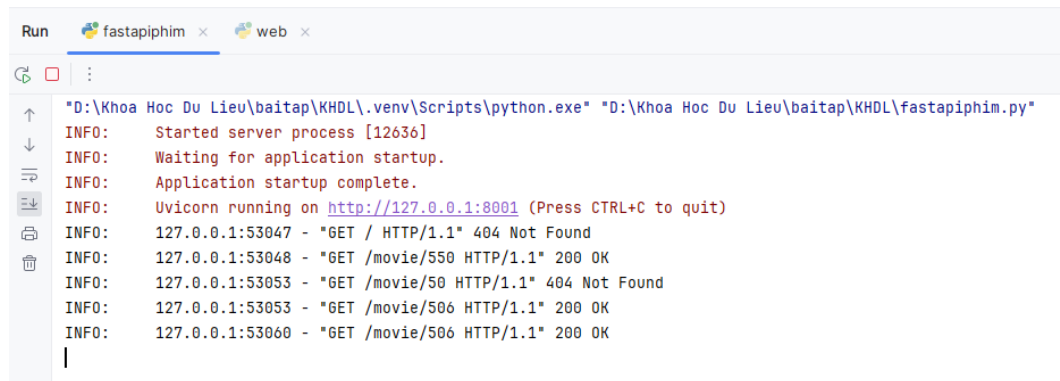
3.1.FASTAPI

Em tạo 1 file Python và sử dụng FastAPI để lấy dữ liệu về PHIM từ một API lấy dữ liệu từ trang web dịch vụ phim trực tuyến TMDB API và trả về dưới dạng JSON thông qua một API.



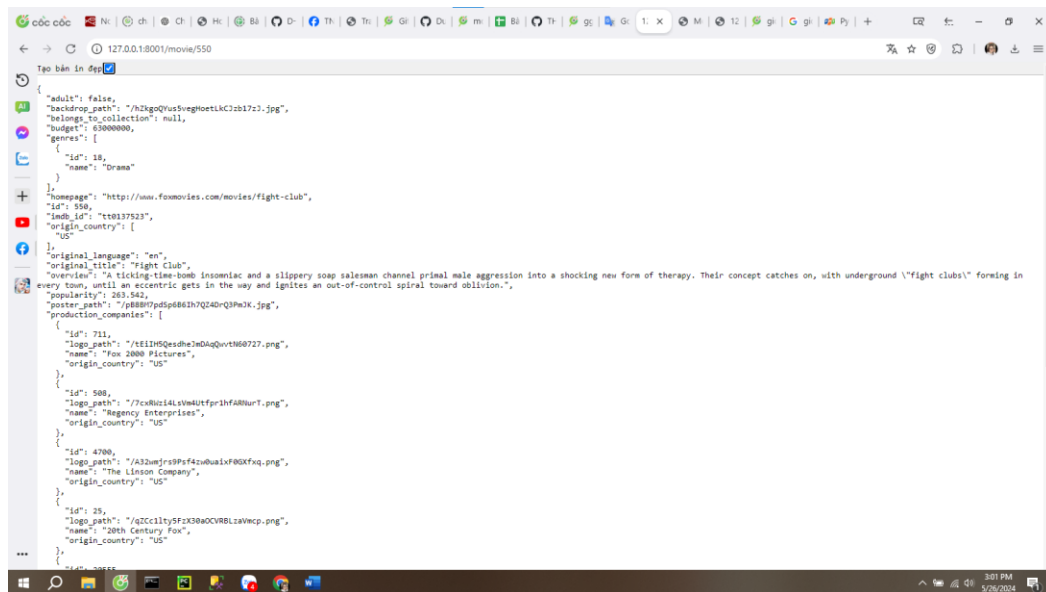
```
1 from fastapi import FastAPI, HTTPException
2 import requests
3
4 app = FastAPI()
5
6 API_KEY = "92e37327850cc5f6aeb20c847c7e7e"
7 BASE_URL = "https://api.themoviedb.org/3"
8
9 @app.get("/movie/{movie_id}")
10 async def get_movie(movie_id: int):
11     url = f"{BASE_URL}/movie/{movie_id}?api_key={API_KEY}"
12     response = requests.get(url)
13     if response.status_code == 200:
14         return response.json()
15     else:
16         raise HTTPException(status_code=response.status_code, detail="Movie not found")
17
18 @app.get("/search")
19 async def search_movies(query: str):
20     url = f"{BASE_URL}/search/movie?api_key={API_KEY}&query={query}"
21     response = requests.get(url)
22     if response.status_code == 200:
23         return response.json()
24     else:
25         raise HTTPException(status_code=response.status_code, detail="Search failed")
26
27 if __name__ == "__main__":
28     import uvicorn
29     uvicorn.run(app, host="127.0.0.1", port=8001)
30
31 async def get_movie():
```

Khởi chạy FastAPI:



```
Run fastapiphim web
"D:\Khoa Hoc Du Lieu\baitap\KHD\venv\Scripts\python.exe" "D:\Khoa Hoc Du Lieu\baitap\KHD\fastapiphim.py"
INFO: Started server process [12636]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://127.0.0.1:8001 (Press CTRL+C to quit)
INFO: 127.0.0.1:53047 - "GET / HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:53048 - "GET /movie/550 HTTP/1.1" 200 OK
INFO: 127.0.0.1:53053 - "GET /movie/50 HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:53053 - "GET /movie/506 HTTP/1.1" 200 OK
INFO: 127.0.0.1:53060 - "GET /movie/506 HTTP/1.1" 200 OK
|
```

Sau khi khởi chạy nó sẽ trả về một chuỗi dạng json trên local của mình:

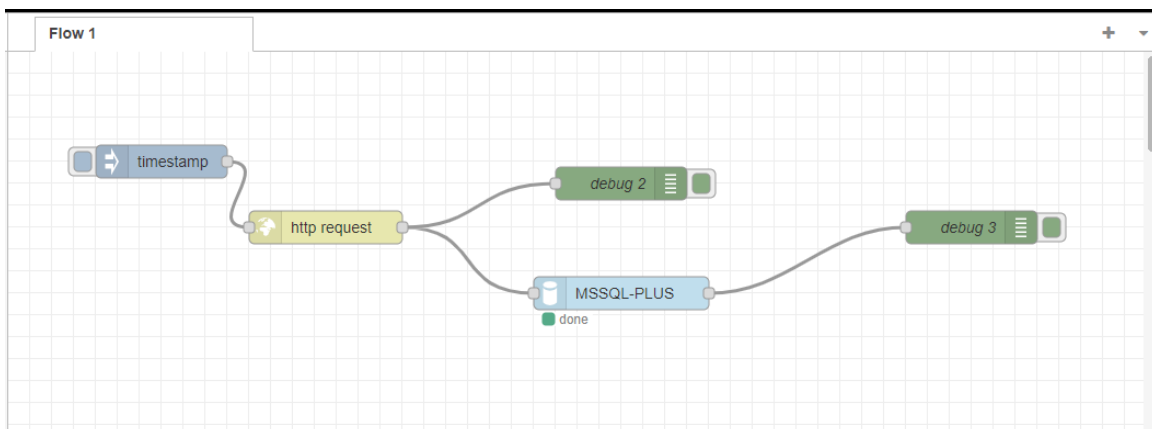


The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8001/movie/550'. The main content area displays a JSON object representing movie information for 'Fight Club'. The JSON includes fields like 'adult', 'backdrop_path', 'belongs_to_collection', 'budget', 'genres', 'id', 'name', 'homepage', 'id', 'iso_639_1', 'iso_639_2', 'origin_country', 'original_language', 'original_title', 'overview', 'popularity', 'poster_path', and 'production_companies'.

```
{
  "adult": false,
  "backdrop_path": "/v3QgQ0vU5v8ghH0tKzCz17z3.jpg",
  "belongs_to_collection": null,
  "budget": 63000000,
  "genres": [
    {
      "id": 18,
      "name": "Drama"
    }
  ],
  "homepage": "http://www.foxmovies.com/movies/fight-club",
  "id": 550,
  "iso_639_1": "en",
  "iso_639_2": "eng",
  "origin_country": [
    "US"
  ],
  "original_language": "en",
  "original_title": "Fight Club",
  "overview": "A ticking-time-bomb insomniac and a slippery soap salesman channel primal male aggression into a shocking new form of therapy. Their concept catches on, with underground \"fight clubs\" forming in every town, until an eccentric gets in the way and ignites an out-of-control spiral toward oblivion.",
  "popularity": 263.542,
  "poster_path": "/pB8B0p0d5p866In7Q40vQ3Pa3K.jpg",
  "production_companies": [
    {
      "id": 711,
      "logo_path": "/fE1H5Qesdhe3mDagQvrtM68727.png",
      "name": "Fox 2000 Pictures",
      "origin_country": "US"
    },
    {
      "id": 508,
      "logo_path": "/7cw8u2i4LxVndutpr1hfA8NurT.png",
      "name": "Regency Enterprises",
      "origin_country": "US"
    },
    {
      "id": 4700,
      "logo_path": "/A32mjr39Psf4z0uailxFOGXfq.png",
      "name": "The Lincon Company",
      "origin_country": "US"
    },
    {
      "id": 25,
      "logo_path": "/q2Ccl1ty5FzX3R0aOCvRBLzavmcp.png",
      "name": "20th Century Fox",
      "origin_country": "US"
    }
  ]
}
```

3.2.NODE-RED

Cấu trúc của một đoạn code node-red sử dụng để lưu dữ liệu vào sql:



ở đây:

- Timestamp : Được sử dụng để chèn thời gian hiện tại vào luồng dữ liệu

Edit inject node

Delete Cancel Done

Properties

Name

msg. payload = timestamp

msg. topic = a_z

+ add inject now

☒ Inject once after 0.1 seconds, then

Repeat interval

every 1 seconds

☐ Enabled

- Http request : Sử dụng node "HTTP Request" để lấy dữ liệu đến API .

Edit http request node

Delete Cancel Done

Properties

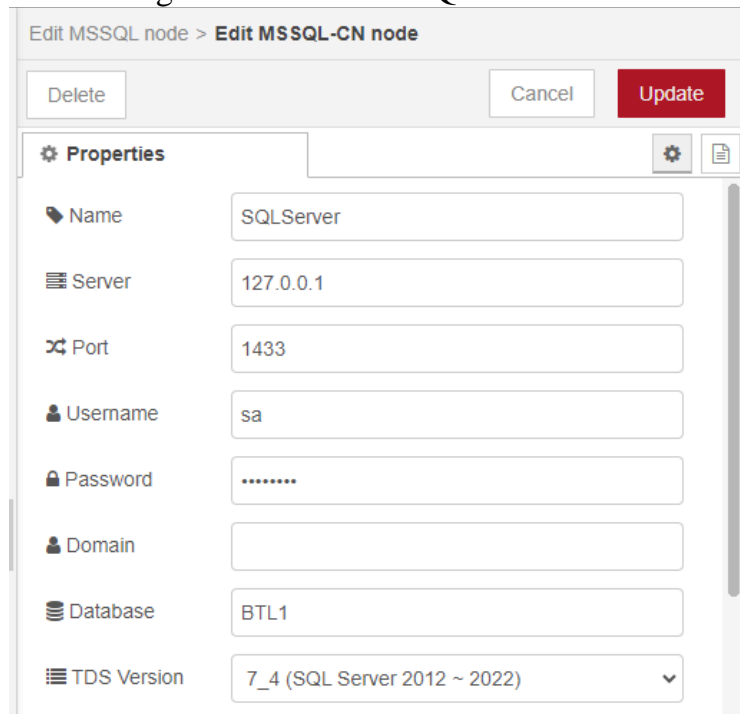
Method GET

URL http://127.0.0.1:8001/movie/506

Payload Ignore

➤ MSSQL-PLUS:

Cài đặt cấu hình các thông tin cho node MSSQL-PLUS:



Edit MSSQL node > Edit MSSQL-CN node

Delete Cancel Update

Properties

Name SQLServer

Server 127.0.0.1

Port 1433

Username sa

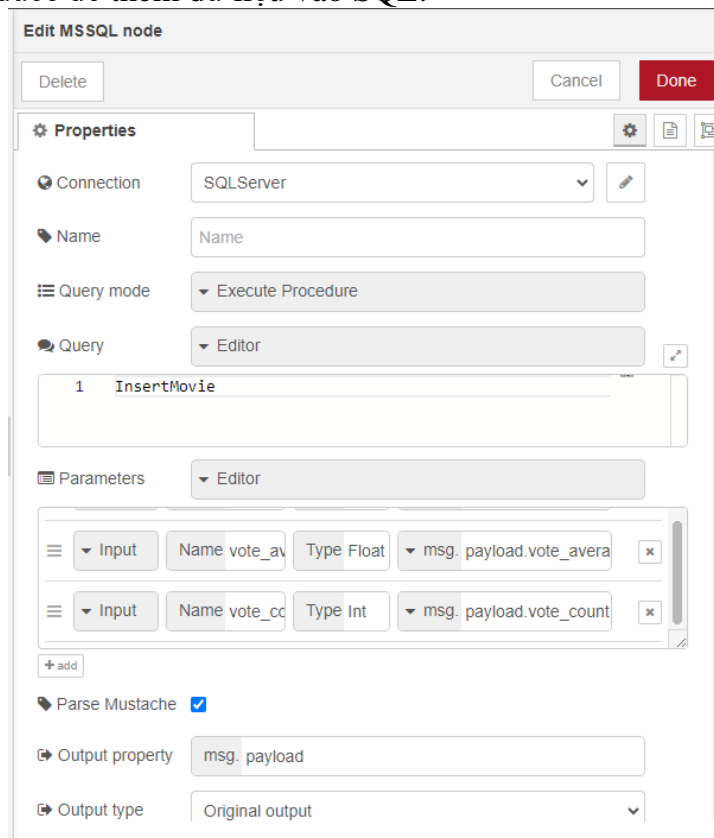
Password

Domain

Database BTL1

TDS Version 7_4 (SQL Server 2012 ~ 2022)

Thực thi procedure để thêm dữ liệu vào SQL:



Edit MSSQL node

Delete Cancel Done

Properties

Connection SQLServer

Name Name

Query mode Execute Procedure

Query Editor

1 InsertMovie

Parameters Editor

Input Name vote_av Type Float msg. payload.vote_avera

Input Name vote_cc Type Int msg. payload.vote_count

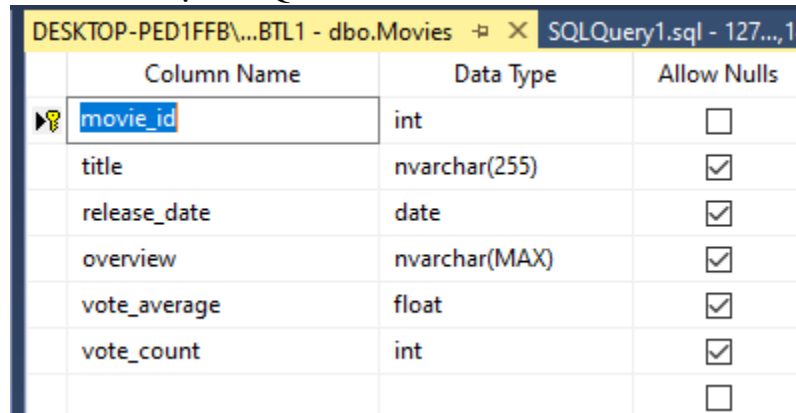
Parse Mustache ☒

Output property msg. payload

Output type Original output

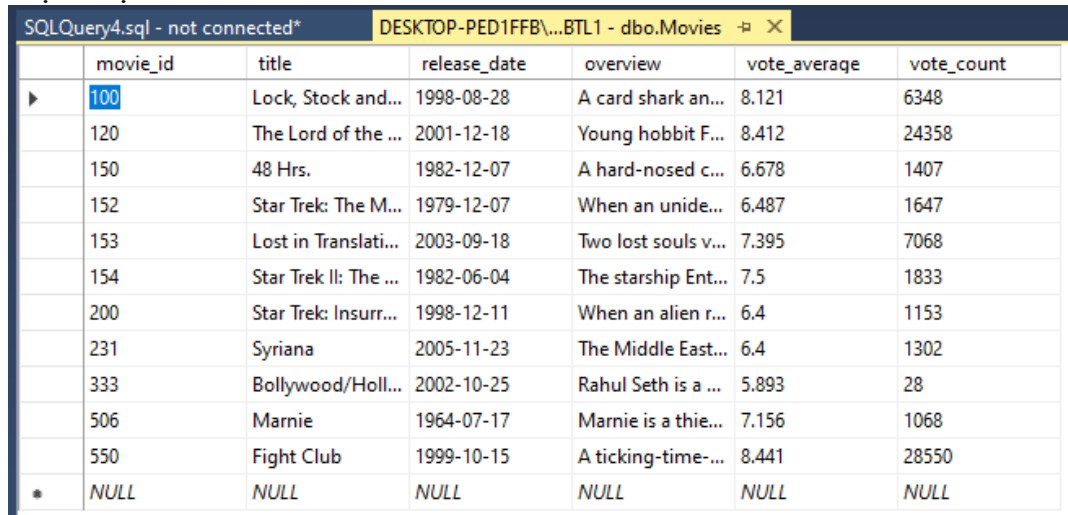
3.3.SQL

- Tạo bảng để lưu trữ dữ liệu ở SQL



Column Name	Data Type	Allow Nulls
movie_id	int	<input type="checkbox"/>
title	nvarchar(255)	<input checked="" type="checkbox"/>
release_date	date	<input checked="" type="checkbox"/>
overview	nvarchar(MAX)	<input checked="" type="checkbox"/>
vote_average	float	<input checked="" type="checkbox"/>
vote_count	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

- Dữ liệu được lưu vào dbo.Movies



movie_id	title	release_date	overview	vote_average	vote_count
100	Lock, Stock and...	1998-08-28	A card shark an...	8.121	6348
120	The Lord of the ...	2001-12-18	Young hobbit F...	8.412	24358
150	48 Hrs.	1982-12-07	A hard-nosed c...	6.678	1407
152	Star Trek: The M...	1979-12-07	When an unide...	6.487	1647
153	Lost in Translati...	2003-09-18	Two lost souls v...	7.395	7068
154	Star Trek II: The ...	1982-06-04	The starship Ent...	7.5	1833
200	Star Trek: Insurr...	1998-12-11	When an alien r...	6.4	1153
231	Syriana	2005-11-23	The Middle East...	6.4	1302
333	Bollywood/Holl...	2002-10-25	Rahul Seth is a ...	5.893	28
506	Marnie	1964-07-17	Marnie is a thie...	7.156	1068
550	Fight Club	1999-10-15	A ticking-time-...	8.441	28550
*	NULL	NULL	NULL	NULL	NULL

3.4.Giao diện

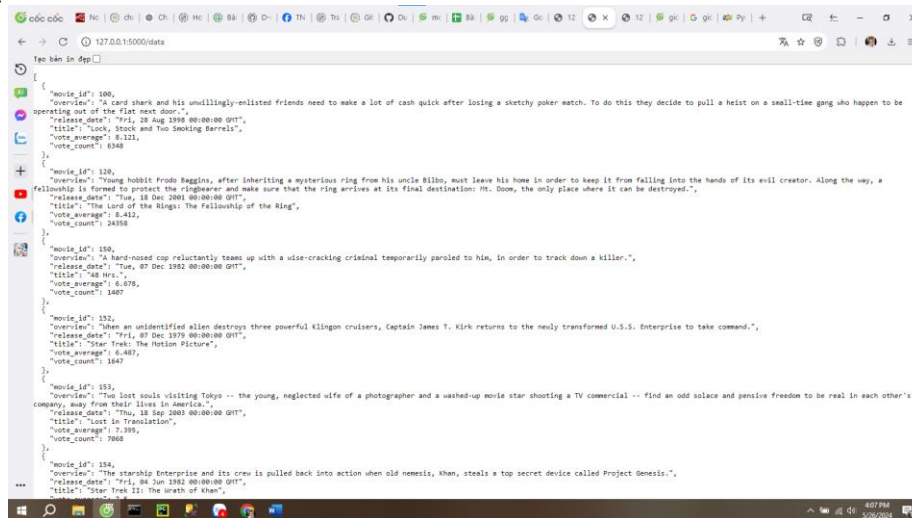
- Lấy dữ liệu từ SQL để vẽ biểu đồ:
- Sử dụng `get_db_connection` để thiết lập và trả về một kết nối đến cơ sở dữ liệu SQL Server.

```
def get_db_connection():  
    return pyodbc.connect('DRIVER={ODBC Driver 17 for SQL Server};SERVER=127.0.0.1,1433;DATABASE=BTL1;UID=sa;PWD=123')
```


- Truy xuất dữ liệu từ cơ sở dữ liệu và trả về dữ liệu đó dưới dạng JSON.

```
@app.route('/data')
def get_data():
    try:
        # Thiết lập kết nối đến cơ sở dữ liệu
        with get_db_connection() as conn:
            cursor = conn.cursor()
            cursor.execute("SELECT * FROM Movies")
            columns = [column[0] for column in cursor.description]
            data = [dict(zip(columns, row)) for row in cursor.fetchall()]
            return jsonify(data)
    except Exception as e:
        # Ghi lại lỗi và trả về thông báo lỗi
        logger.error(f"Error occurred while fetching data: {str(e)}")
        return jsonify({"error": str(e)}), 500
```

- Hiển thị lên web:



- Hiển thị dữ liệu từ cơ sở dữ liệu

```
@app.route('/index')
def show_movies():
    try:
        # Thiết lập kết nối đến cơ sở dữ liệu
        with get_db_connection() as conn:
            cursor = conn.cursor()
            cursor.execute("SELECT * FROM Movies")
            columns = [column[0] for column in cursor.description]
            data = [dict(zip(columns, row)) for row in cursor.fetchall()]
            return render_template(template_name_or_list='index.html', columns=columns, data=data)
    except Exception as e:
        # Ghi lại lỗi và trả về thông báo lỗi
        logger.error(f"Error occurred while fetching data: {str(e)}")
        return render_template(template_name_or_list='error.html', error=str(e))
```

- Hiện thị lên web

127.0.0.1:5000/index

Movies

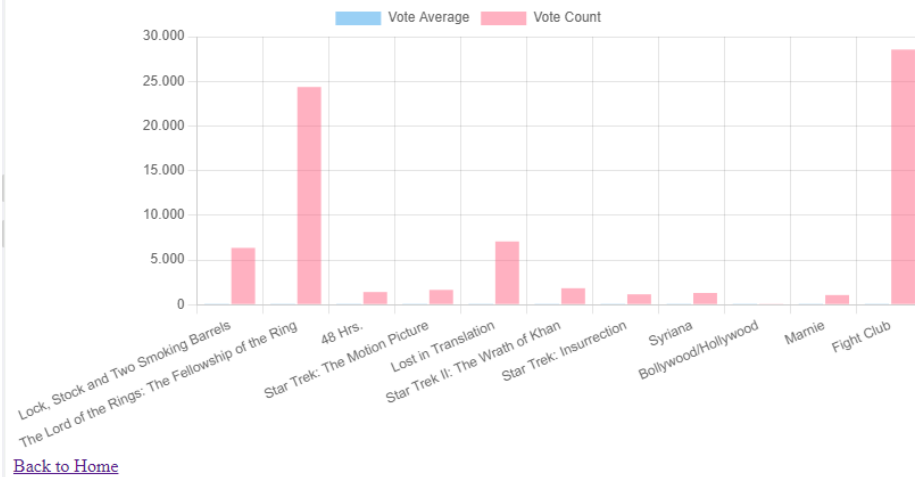
movie_id	title	release_date	overview	vote_average	vote_count
100	Lock, Stock and Two Smoking Barrels	1998-08-28	A card shark and his unwillingly-enlisted friends need to make a lot of cash quick after losing a sketchy poker match. To do this they decide to pull a heist on a small-time gang who happens to be operating out of the flat next door.	8.121	6348
120	The Lord of the Rings: The Fellowship of the Ring	2001-12-18	Young hobbit Frodo Baggins, after inheriting a mysterious ring from his uncle Bilbo, must leave his home in order to keep it from falling into the hands of its evil creator. Along the way, a fellowship is formed to protect the ringbearer and make sure that the ring arrives at its final destination: Mt. Doom, the only place where it can be destroyed.	8.412	24358
150	48 Hrs.	1982-12-07	A hard-nosed cop reluctantly teams up with a wise-cracking criminal temporarily paroled to him, in order to track down a killer.	6.678	1407
152	Star Trek: The Motion Picture	1979-12-07	When an unidentified alien destroys three powerful Klingon cruisers, Captain James T. Kirk returns to the newly transformed U.S.S. Enterprise to take command.	6.487	1647
153	Lost in Translation	2003-09-18	Two lost souls visiting Tokyo -- the young, neglected wife of a photographer and a washed-up movie star shooting a TV commercial -- find an odd solace and pensive freedom to be real in each other's company, away from their lives in America.	7.395	7068
154	Star Trek II: The Wrath of Khan	1982-06-04	The starship Enterprise and its crew is pulled back into action when old nemesis, Khan, steals a top secret device called Project Genesis.	7.5	1833
200	Star Trek: Insurrection	1998-12-11	When an alien race and factions within Starfleet attempt to take over a planet that has "regenerative" properties, it falls upon Captain Picard and the crew of the Enterprise to defend the planet's people as well as the very ideals upon which the Federation itself was founded.	6.4	1153
231	Syriana	2005-11-23	The Middle Eastern oil industry is the backdrop of this tense drama, which weaves together numerous story lines. Bennett Holiday is an American lawyer in charge of facilitating a dubious merger of oil companies, while Bryan Woodman, a Switzerland-based energy analyst, experiences both personal tragedy and opportunity during a visit with Arabian royalty. Meanwhile, veteran CIA agent Bob Barnes uncovers an assassination plot with unsettling origins.	6.4	1302
333	Bollywood Hollywood	2002-10-25	Rahul Seth is a dashing young millionaire who believes he is "western" enough to rebel against his mother and grandmother. They are not too keen about his Caucasian girlfriend Kimberly who, to make matters worse, is a pop star. Before you can say "karmic intervention," Kimberly dies in a freak accident and Rahul is devastated. Instead of allowing him to mourn in peace, Rahul's mother sees the opportunity she's been waiting for. She threatens to call off his sister's wedding unless he finds himself a "nice Indian girl." Rahul enlists the services of Sue, a fiercely independent escort whom he believes to be Hispanic, and therefore not "married" to the conventions taught to young Indian women. With a wink in her eye, Sue accepts the deal to pose as his Indian bride-to-be. She needs the money and having never been a fan of the typical Indian male, she feels her heart is safe. The chaos begins.	5.893	28
506	Marnie	1964-07-17	Marnie is a thief, a liar, and a cheat. When her new boss, Mark Rutland, catches on to her routine kleptomaniac, she finds herself being blackmailed.	7.156	1068
550	Fight Club	1999-10-15	A ticking-time-bomb insomniac and a slippery soap salesman channel primal male aggression into a shocking new form of therapy. Their concept catches on, with underground "fight clubs" forming in every town, until an eccentric gets in the way and ignites an out-of-control spiral toward oblivion.	8.441	28550

- Vẽ biểu đồ

```
@app.route('/chart')
def show_chart():
    return render_template('chart.html')
```

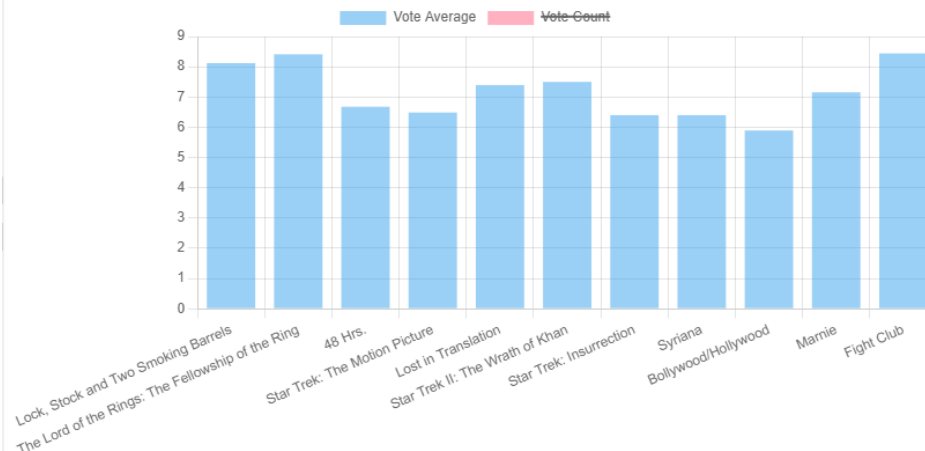
- Hiện thị trên web

Movies Chart



- Do biểu đồ của vote count quá cao nên không thể nhìn thấy cote Average nên em đã tắt biểu đồ của vote count để hiện rõ hơn:

Movies Chart



[Back to Home](#)

CHƯƠNG IV. KẾT LUẬN

Đề tài "Xây dựng hệ thống đề xuất phim" đã được triển khai, kết hợp sự tích hợp giữa FastAPI, Node-RED và cơ sở dữ liệu SQL, tạo ra một giải pháp mạnh mẽ và linh hoạt. FastAPI đã giúp dễ dàng tạo ra các API RESTful, Node-RED đã đóng vai trò quan trọng trong việc thiết lập luồng công việc và tích hợp các thành phần của hệ thống, trong khi cơ sở dữ liệu SQL giúp lưu trữ dữ liệu một cách có tổ chức và dễ dàng truy xuất.

Giao diện người dùng đã được thiết kế để hiển thị dữ liệu một cách trực quan và thân thiện, giúp người dùng dễ dàng nắm bắt thông tin quan trọng về các bộ phim đề xuất. Hệ thống cũng có tiềm năng mở rộng và phát triển trong tương lai, có thể tích hợp thêm các tính năng mới và xử lý nhiều loại dữ liệu khác nhau.

Thông qua bài tập này, em đã có cơ hội hiểu rõ hơn về ngôn ngữ lập trình Python cũng như một số công cụ hữu ích như Node-RED và SQL. Cuối cùng, việc hoàn thành dự án cũng là nhờ vào sự hỗ trợ nhiệt tình của thầy Đỗ Duy Cốp, và em muốn gửi lời cảm ơn tới thầy về điều đó.