

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: THEO DÕI TIN TỨC CẬP NHẬP TỪ NHIỀU NGUỒN

Giáo Viên Hướng Dẫn : Đỗ Duy Cốp

Họ và Tên : Nguyễn Thị Minh Hoa

MSSV : K205480106054

Lớp : K56KMT.01

Khoa : Điện Tử

Thái Nguyên 2024

PHIẾU GIAO ĐỀ TÀI MÔN LẬP TRÌNH PYTHON

Sinh viên: Nguyễn Thị Minh Hoa

MSSV :K205480106054

Lớp: K56KMT.01

Khoá : K56

Ngành học: Kỹ thuật máy tính.

Giáo viên hướng dẫn: Đỗ Duy Cốp

1. Tên đề tài: Theo dõi tin tức cập nhập từ nhiều nguồn

2. Nội dung thực hiện:

- Dùng FastApi gọi Api để lấy dữ liệu về tin tức từ nguồn dữ liệu NewsAPI:
<https://newsapi.org/docs/get-started>.

- Tạo sơ sở dữ liệu trong SQL Server để lưu dữ liệu đã qua xử lý từ Node-red.

- Dùng Node-red để lấy dữ liệu từ Api và lưu về cơ sở dữ liệu.

- Dùng Flask để xây dựng biểu đồ và hiển thị lên web.

3. Các sản phẩm, kết quả :

- Sản phẩm phần mềm theo yêu cầu của đề án.

4. Ngày giao nhiệm vụ: 15/05/2024.

5. Ngày hoàn thành nhiệm vụ: 26/05/2024.

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	
CHƯƠNG I: TỔNG QUÁT	5
1.1. Đặt vấn đề	5
1.2. Mục tiêu	5
1.3. Hướng giải quyết đề tài	6
CHƯƠNG II. CƠ SỞ LÝ THUYẾT	7
2.1. SQL Server Management Studio	7
2.2. Ngôn ngữ lập trình Python	8
2.3. Visual Studio Code	9
2.4. Node-Red	10
CHƯƠNG III. NỘI DUNG THỰC HIỆN	12
3.1. FastAPI	12
3.2. Node-Red	13
3.3. SQL	17
3.4. Giao diện	19
CHƯƠNG IV. KẾT LUẬN	23

CHƯƠNG I: TỔNG QUÁT

1.1. Đặt vấn đề

Trong thời đại ngày càng phát triển như hiện nay, mỗi ngày sẽ có nhiều nguồn tin đăng tải rất nhiều bài báo với vô số chủ đề thế nên việc cập nhật tin tức nhanh chóng là rất cần thiết.

Do đó, việc phát triển một hệ thống theo dõi tin tức là rất cần thiết. Việc theo dõi tin tức từ nhiều nguồn không chỉ giúp người đọc cập nhật thông tin mới nhất mà còn mang lại sự đa dạng và phong phú trong các nguồn thông tin.

1.2. Mục tiêu

Để giải quyết vấn đề này, cần thiết phải xây dựng một hệ thống thu thập, xử lý và hiển thị dữ liệu số lượng tin tức từ từng nguồn một cách hiệu quả và chính xác. Hệ thống này không chỉ giúp cung cấp thông tin đầy đủ cho người dùng mà còn hỗ trợ người dùng tìm kiếm và lựa chọn tin tức phù hợp.

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

- Thu thập dữ liệu về tin tức: Dùng FastApi gọi Api để thu thập dữ liệu từ nguồn uy tín và được cập nhật liên tục.
- Xử lý dữ liệu: Dùng Node-red để xử lý dữ liệu lấy được từ Api
- Lưu trữ dữ liệu: Sau khi dữ liệu đã được xử lý sẽ được lưu về SQL Server
- Xây dựng trang web: Dùng Flask để tạo giao diện thân thiện cho người dùng. Trang web sẽ bao gồm biểu đồ trực quan được vẽ dựa trên cơ sở dữ liệu, giúp người dùng dễ dàng theo dõi tin tức.

1.3. Hướng giải quyết đề tài

Để giải quyết vấn đề về theo dõi tin tức và hiển thị thông tin một cách hiệu quả, chúng ta sẽ thực hiện các bước sau:

- Dùng FastApi để gọi Api để lấy dữ liệu
 - Sử dụng Python + FastApi gọi 1 Api để dữ liệu. Điều này thường được thực hiện bằng cách sử dụng một thư viện HTTP client như httpx hoặc requests. Sau khi lấy được dữ liệu từ API bên ngoài, ta có thể xử lý và trả về kết quả thông qua endpoint của FastAPI.
- Dùng Node-red để lấy dữ liệu từ Api gọi:
 - Trong Node-red sử dụng HTTP requests tới localhost để lấy dữ liệu mà ta đã lấy qua Api và đẩy lên web. Sau khi lấy được dữ liệu cần phải xử lý để phù hợp với cơ sở dữ liệu ta đã tạo.
- Dùng SQL để lưu dữ liệu:
 - Tạo cơ sở dữ liệu phù hợp với đề tài và viết Stored Procedure để lưu dữ liệu từ Node-red.
- Xây dựng giao diện người dùng từ dữ liệu từ SQL để vẽ biểu đồ:
 - Dùng Flask để tạo giao diện người dùng, thông qua ajax gọi dữ liệu đã lưu để vẽ biểu đồ

Với những bước thực hiện ở trên, ta sẽ có một hệ thống cho phép thu thập dữ liệu, xử lý dữ liệu và hiển thị dữ liệu một cách hiệu quả, đồng thời cung cấp thông tin đầy đủ dễ hiểu cho người dùng

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.

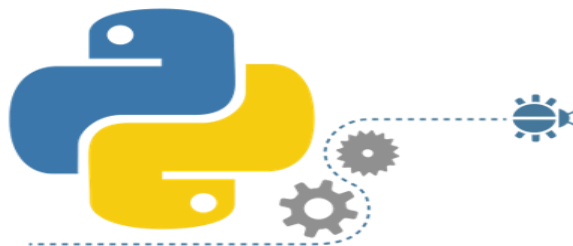
Nó cũng cho phép quản trị viên sao lưu và phục hồi dữ liệu, kiểm tra và theo dõi hiệu suất, và quản lý bảo mật. SSMS cho phép người dùng thực hiện các truy vấn SQL và xem dữ liệu từ các bảng trong cơ sở dữ liệu. Nó cung cấp một trình soạn thảo truy vấn mạnh mẽ với tính năng gợi ý cú pháp và điều hướng 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. Ngôn ngữ lập trình 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. Visual Studio Code

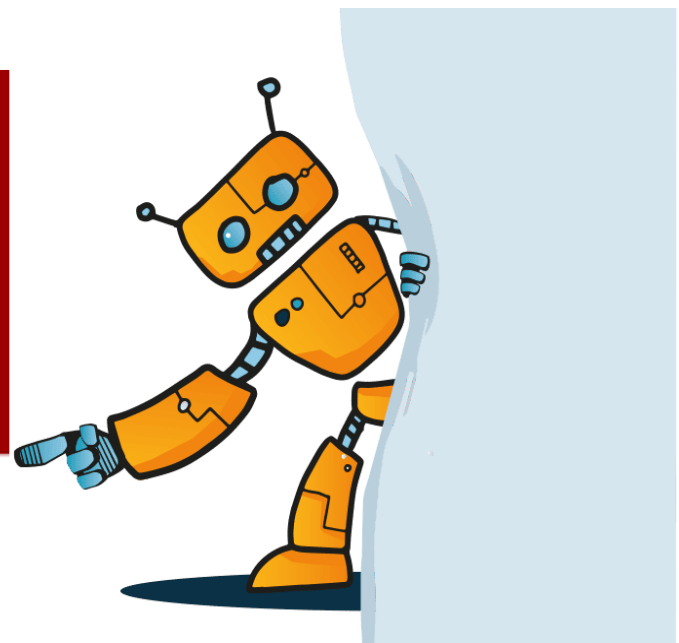
Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở, miễn phí và rất phổ biến, được phát triển bởi Microsoft. Ra mắt lần đầu vào năm 2015, VS Code nhanh chóng trở thành công cụ yêu thích của nhiều lập trình viên nhờ vào sự nhẹ nhàng, mạnh mẽ và linh hoạt của nó.



Visual Studio Code hỗ trợ đa dạng các chức năng Debug, đi kèm với Git, có Syntax Highlighting. Đặc biệt là tự hoàn thành mã thông minh, Snippets, và khả năng cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép các lập trình viên thay đổi Theme, phím tắt, và đa dạng các tùy chọn khác. Mặc dù trình soạn thảo Code này tương đối nhẹ, nhưng lại bao gồm các tính năng mạnh mẽ.

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.

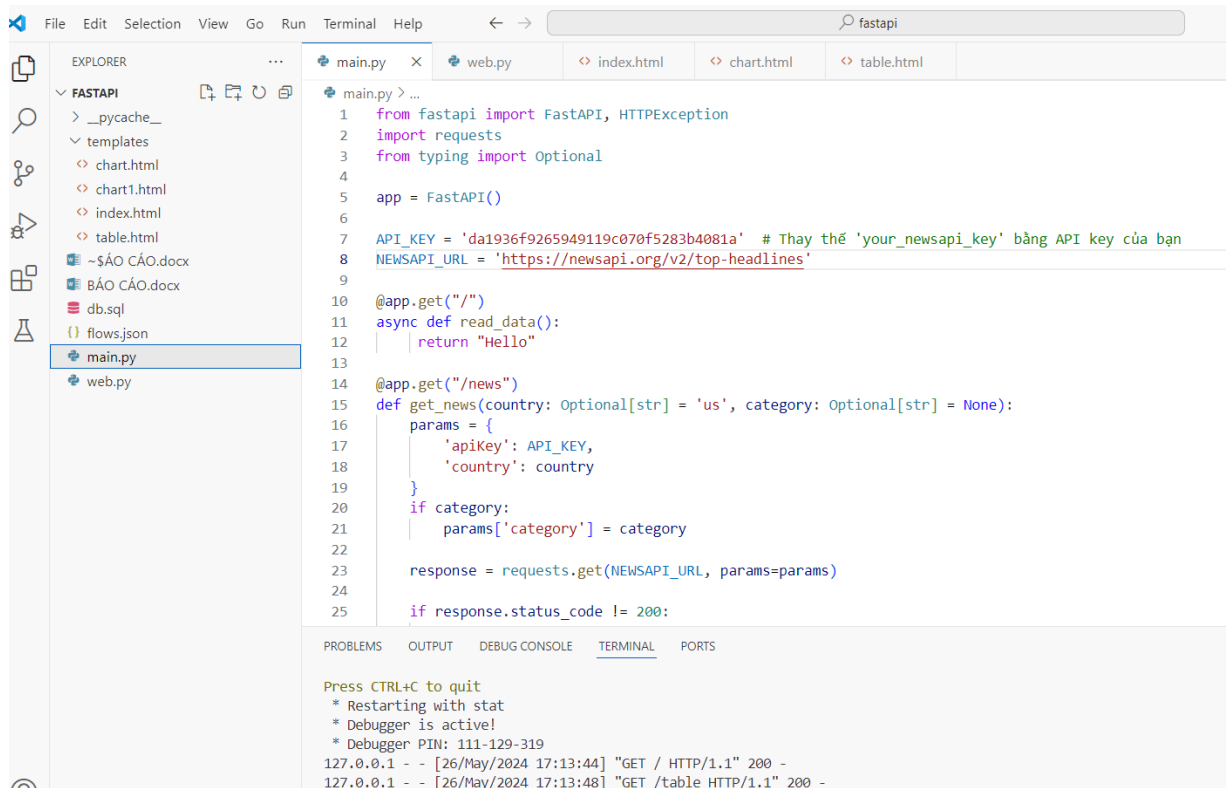
Node-RED được sử dụng rộng rãi trong Internet of Things (IoT) và trong các ứng dụng tự động hóa, nơi nó có thể giúp kết nối và tự động hóa các thiết bị và dịch vụ từ nhiều nhà sản xuất khác nhau. Nó cũng thích hợp cho việc xử lý dữ liệu thời gian thực và tích hợp các dịch vụ web khác nhau.

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

3.1. FastAPI

Tạo 1 file Python sử dụng FastApi để gọi Api lấy dữ liệu về tin tức và trả về dưới dạng JSON.



```
1 from fastapi import FastAPI, HTTPException
2 import requests
3 from typing import Optional
4
5 app = FastAPI()
6
7 API_KEY = 'da1936f9265949119c070f5283b4081a' # Thay thế 'your_newsapi_key' bằng API key của bạn
8 NEWSAPI_URL = 'https://newsapi.org/v2/top-headlines'
9
10 @app.get("/")
11 async def read_data():
12     return "Hello"
13
14 @app.get("/news")
15 def get_news(country: Optional[str] = 'us', category: Optional[str] = None):
16     params = {
17         'apiKey': API_KEY,
18         'country': country
19     }
20     if category:
21         params['category'] = category
22
23     response = requests.get(NEWSAPI_URL, params=params)
24
25     if response.status_code != 200:
```

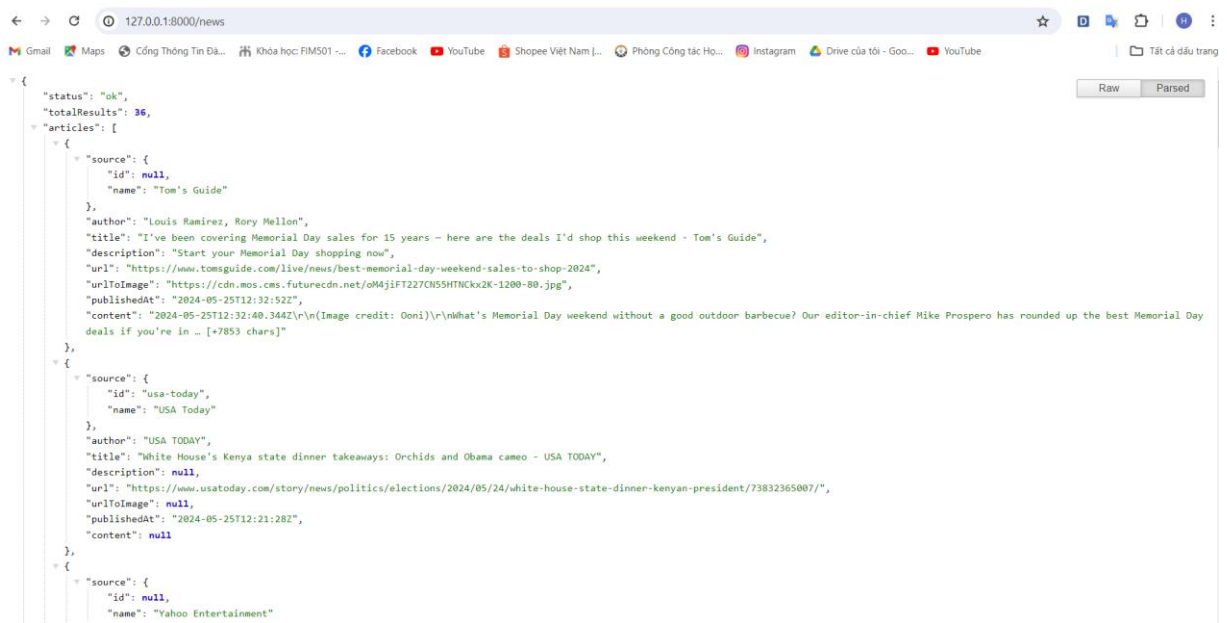
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 111-129-319
127.0.0.1 - - [26/May/2024 17:13:44] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2024 17:13:48] "GET /table HTTP/1.1" 200 -
```

Chạy FastApi:

```
PS D:\python\fastapi> uvicorn main:app
INFO: Started server process [1428]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: 127.0.0.1:52735 - "GET / HTTP/1.1" 200 OK
INFO: 127.0.0.1:52735 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:52736 - "GET /news HTTP/1.1" 200 OK
```

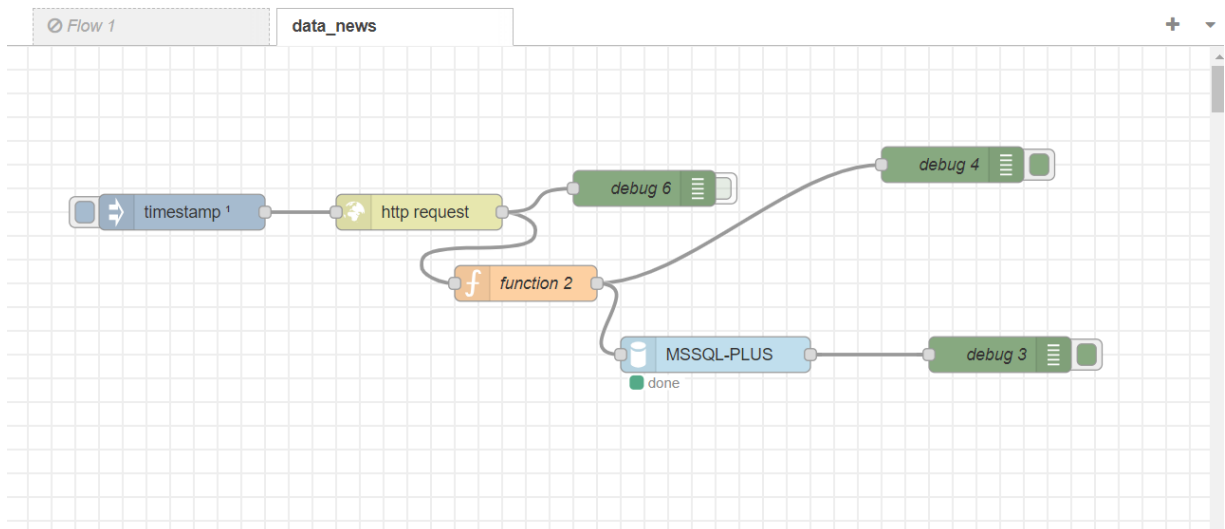
Dữ liệu trả về dạng JSON:



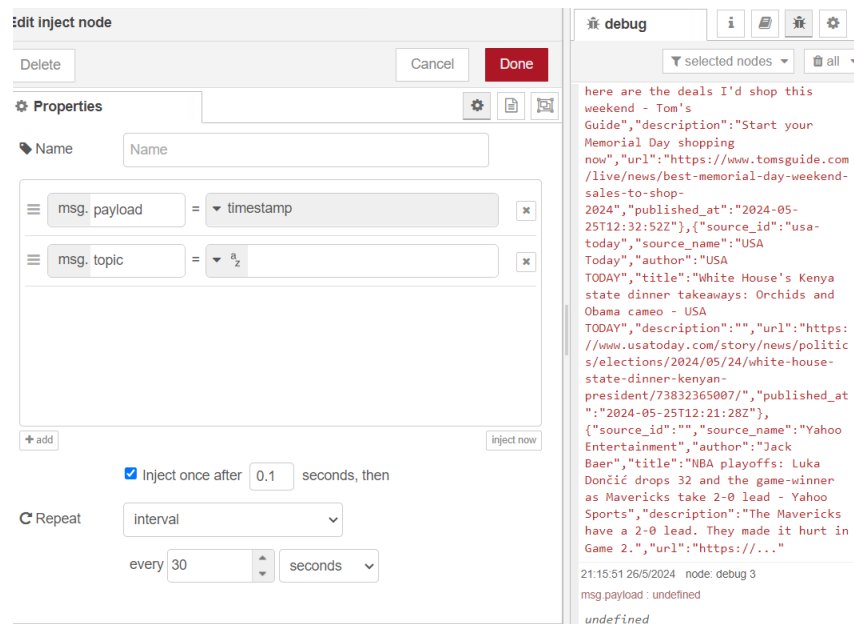
```
{
  "status": "ok",
  "totalResults": 36,
  "articles": [
    {
      "source": {
        "id": null,
        "name": "Tom's Guide"
      },
      "author": "Louis Ramirez, Rory Mellon",
      "title": "I've been covering Memorial Day sales for 15 years - here are the deals I'd shop this weekend - Tom's Guide",
      "description": "Start your Memorial Day shopping now",
      "url": "https://www.tomsguide.com/live/news/best-memorial-day-weekend-sales-to-shop-2024",
      "urlToImage": "https://cdn.mos.cms.futurecdn.net/oM4jIFt227CN55HTNCKx2K-1200-80.jpg",
      "publishedAt": "2024-05-25T12:32:52Z",
      "content": "2024-05-25T12:32:40.344Z\r\n(Image credit: Ooni)\r\nWhat's Memorial Day weekend without a good outdoor barbecue? Our editor-in-chief Mike Prospero has rounded up the best Memorial Day deals if you're in ... [+7853 chars]"
    },
    {
      "source": {
        "id": "usa-today",
        "name": "USA Today"
      },
      "author": "USA TODAY",
      "title": "White House's Kenya state dinner takeaways: Orchids and Obama cameo - USA TODAY",
      "description": null,
      "url": "https://www.usatoday.com/story/news/politics/elections/2024/05/24/white-house-state-dinner-kenyan-president/73832365007/",
      "urlToImage": null,
      "publishedAt": "2024-05-25T12:21:28Z",
      "content": null
    },
    {
      "source": {
        "id": null,
        "name": "Yahoo Entertainment"
      }
    }
  ]
}
```

3.2. Node-Red

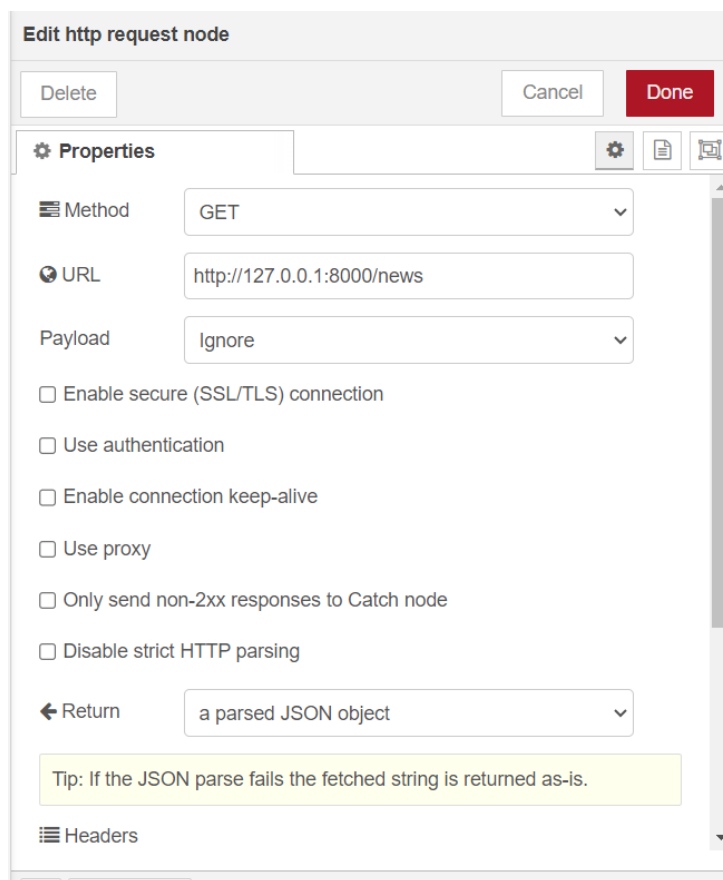
Cấu trúc flow trên node-red để lấy và lưu dữ liệu:



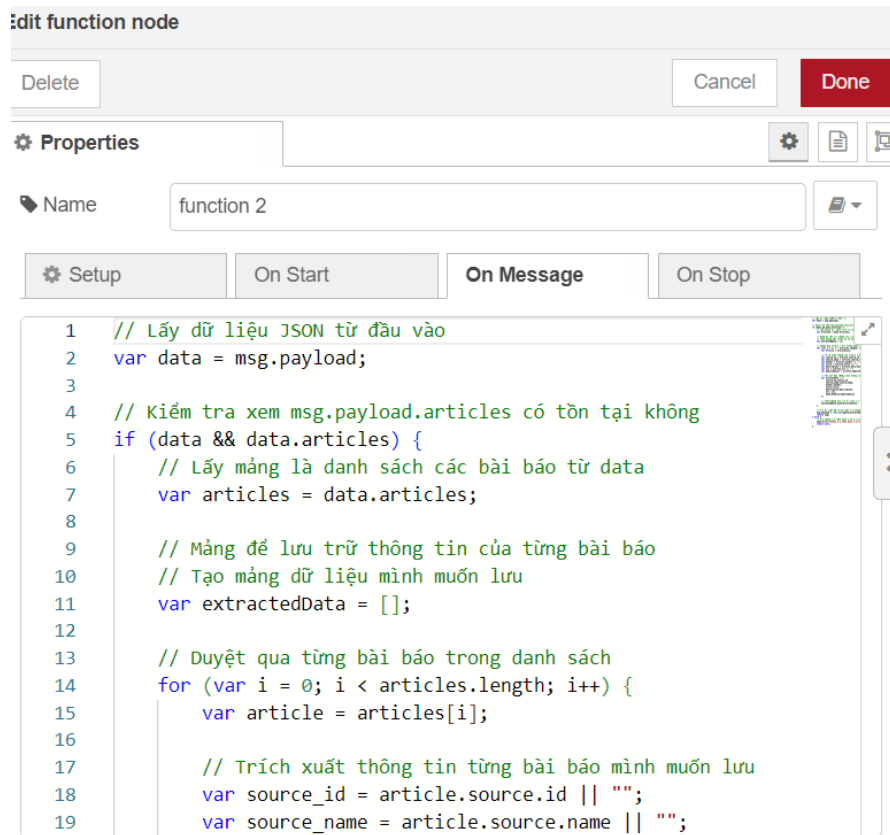
- Timestamp: Được dùng để cài đặt thời gian gọi request lấy dữ liệu:



- HTTP Request: Dùng để lấy dữ liệu từ localhost



- Function: Để xử lý dữ liệu cho phù hợp với SQL



```
// Lấy dữ liệu JSON từ đầu vào
var data = msg.payload;

// Kiểm tra xem msg.payload.articles có tồn tại không
if (data && data.articles) {
    // Lấy mảng là danh sách các bài báo từ data
    var articles = data.articles;

    // Mảng để lưu trữ thông tin của từng bài báo
    // Tạo mảng dữ liệu mình muốn lưu
    var extractedData = [];

    // Duyệt qua từng bài báo trong danh sách
    for (var i = 0; i < articles.length; i++) {
        var article = articles[i];

        // Trích xuất thông tin từng bài báo mình muốn lưu
        var source_id = article.source.id || "";
        var source_name = article.source.name || "";
        var author = article.author || "";
        var title = article.title || "";
```

```

var description = article.description || "";
var url = article.url || "";
var published_at = article.publishedAt || "";

// Tạo một đối tượng chứa thông tin của bài báo
var articleInfo = {
    source_id:source_id,
    source_name:source_name,
    author:author,
    title:title,
    description:description,
    url: url,
    published_at:published_at
};

// Thêm thông tin của bài báo vào mảng extractedData
extractedData.push(articleInfo);
}

// Gửi dữ liệu đã trích xuất vào output dạng string
msg.payload = JSON.stringify(extractedData);
return msg;
} else {
    // Nếu không tìm thấy danh sách bài báo, thông báo lỗi
    node.error("Không tìm thấy danh sách bài báo trong dữ liệu
JSON.");
    return null;
}
}

```

- MSSQL-PLUS: Thiết lập cổng kết nối và gọi SP để lấy dữ liệu lưu vào SQL

Edit MSSQL node > **Edit MSSQL-CN node**

Delete Cancel **Update**

Properties

Name data_news

Server 127.0.0.1

Port 1433

Username sa

Password

Domain

Database news

TDS Version 7_4 (SQL Server 2012 ~ 2022)

Use Encryption? ☒

SQL Databases hosted on Azure will need this checked.

Trust Certificate? ☒

Edit MSSQL node

Delete Cancel **Done**

Properties

Connection data_news

Name Name

Query mode Execute Procedure

Query Editor

1 InsertArticle

Parameters Editor


Input Name json Type VarCt msg. payload

+ add

Parse Mustache ☒

3.3. SQL

Tạo bảng để lưu dữ liệu:

	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	source_id	varchar(255)	<input checked="" type="checkbox"/>
	source_name	varchar(255)	<input checked="" type="checkbox"/>
	author	varchar(255)	<input checked="" type="checkbox"/>
	title	varchar(MAX)	<input checked="" type="checkbox"/>
	description	varchar(MAX)	<input checked="" type="checkbox"/>
	url	varchar(MAX)	<input checked="" type="checkbox"/>
	published_at	datetime	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Store procedure lấy dữ liệu từ node-red lưu vào database

```

USE [news]
GO
/***** Object: StoredProcedure [dbo].[InsertArticle]    Script Date: Sun,26,05,2024 9:38:45 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[InsertArticle]
    @json VARCHAR(max)
AS
BEGIN
    INSERT INTO articles (source_id, source_name, author, title, description, url, published_at)
    SELECT json_data.source_id, json_data.source_name, json_data.author, json_data.title, json_data.description, json_data.url, json_data.published_at
    FROM OPENJSON(@json)
    WITH (
        source_id NVARCHAR(1000),
        source_name NVARCHAR(1000),
        author NVARCHAR(1000),
        title NVARCHAR(4000),
        description NVARCHAR(MAX),
        url NVARCHAR(MAX),
        published_at DATETIME
    ) AS json_data
    LEFT JOIN articles ON articles.source_id = json_data.source_id
                        AND articles.source_name = json_data.source_name
                        AND articles.author = json_data.author
                        AND articles.title = json_data.title
                        AND articles.description = json_data.description
                        AND articles.url = json_data.url
                        AND articles.published_at = json_data.published_at
    WHERE articles.source_id IS NULL
END;

```

Dữ liệu đã được lưu vào bảng:

	id	source_id	source_na...	author	title	description	url	published_at
▶	22	cnn	CNN	Deblina Cha...	Boeing Starl...	After report...	https://www...	2024-05-22 ...
	23	nbc-news	NBC News	Patrick Smith	Deadly torn...	Central low...	https://www...	2024-05-22 ...
	24	google-news	Google News	NFL.com	NFL's top 10...		https://news...	2024-05-22 ...
	25		CNET		T-Mobile Is ...	Those on ol...	https://www...	2024-05-22 ...
	26		KTLA Los A...	Tony Kurzw...	SpaceX rock...	A SpaceX ro...	https://ktla...	2024-05-22 ...
	27		Hollywood ...	Lacey Rose	Glen Powell ...	The red hot ...	http://www....	2024-05-22 ...
	28		CBS Sports		How Pacers,...	The ball slip...	https://www...	2024-05-22 ...
	29		NASCAR	NASCAR	Ricky Stenh...		https://www...	2024-05-22 ...
	30	reuters	Reuters	Reuters UK	Sunak says ...		https://www...	2024-05-22 ...
	31	techcrunch	TechCrunch	Paul Sawers	Humane, th...	Humane, th...	https://tech...	2024-05-22 ...
	32		NPR	Will Stone	Microplastic...	The new stu...	https://www...	2024-05-22 ...
	33	fox-news	Fox News	Tracy Wright	Diddy's ex '...	Sean "Didd...	https://www...	2024-05-22 ...
	34	cnn	CNN	Jackie Castil...	'That's Hitle...	President Jo...	https://www...	2024-05-22 ...
	35		Entertainme...	Sophie Schil...	Taylor Swift ...	Taylor Swift ...	https://www...	2024-05-22 ...
	36	abc-news	ABC News	Kevin Shalv...	1 dead, 6 cri...	Singapore ...	https://abcn...	2024-05-22 ...
	37		CNBC	Hakyung Ki...	Stock future...	U.S. stock fu...	https://www...	2024-05-22 ...
	38		NPR	Alina Selyukh	Buy-now, p...	The Consum...	https://www...	2024-05-22 ...
	39		MarketWatch	Tomi Kilgore	Target's stoc...	Target's sale...	https://www...	2024-05-22 ...
	40	google-news	Google News	WIRED	What Scarle...		https://news...	2024-05-22 ...
	41		Atlanta Jour...	Tamar Halle...	Fulton DA ...		https://www...	2024-05-22 ...
	42		Yahoo Enter	Jack Baer	Caitlin Clark	Caitlin Clark	https://snor	2024-05-25

3.4. Giao diện

Thiết lập kết nối đến cơ sở dữ liệu

```
# Hàm này thiết lập và trả về kết nối đến cơ sở dữ liệu
def get_db_connection():
    return pyodbc.connect('DRIVER={ODBC Driver 17 for SQL Server};SERVER=127.0.0.1,1433;DATABASE=news;UID=sa;PWD=1234')
```

Truy xuất vào cơ sở dữ liệu và trả về dữ liệu 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 articles")
            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
```

Dữ liệu hiển thị lên web:

```
{
  {
    "tác giả": "Deblina Chakraborty",
    "mô tả": "Sau khi báo cáo về vụ rò rỉ khí heli trong tàu vũ trụ, NASA đã ấn định ngày mục tiêu tiếp theo cho vụ phóng rất được mong đợi - mà không đặt tên ngay cho một vụ",
    "id": 22,
    "posted_at": "Thứ Tư, ngày 22 tháng 5 năm 2024 14:14:00 GMT",
    "source_id": "cnn",
    "source_name": "CNN",
    "title": "Buổi ra mắt lịch sử của phi hành đoàn Boeing Starliner lại bị trì hoãn vô thời hạn - CNN",
    "url": "https://www.cnn.com/2024/05/22/world/boeing-starliner-crewed-launch-delayed-indefinitely-scn/index.html"
  },
  {
    "tác giả": "Patrick Smith",
    "description": "Miền Trung Iowa chao đảo vào sáng thứ Tư sau khi cảnh sát mô tả là một cơn lốc xoáy "tàn khốc" tàn phá các cộng đồng nông thôn và giết chết vô số người.",
    "id": 23,
    "posted_at": "Thứ Tư, ngày 22 tháng 5 năm 2024 13:49:46 GMT",
    "source_id": "nbc-news",
    "source_name": "Tin tức NBC",
    "title": "Cơn lốc xoáy chết người tàn phá thị trấn Iowa khi thời tiết khắc nghiệt di chuyển về phía đông - NBC News",
    "url": "https://www.nbcnews.com/news/us-news/deadly-tornado-devastates-iowa-town-severe-weather-moves-south-rcna153430"
  },
  {
    "tác giả": "NFL.com",
    "Sự miêu tả": "",
    "id": 24,
    "posted_at": "Thứ tư, ngày 22 tháng 5 năm 2024 13:23:00 GMT",
    "source_id": "google-news",
    "source_name": "Google Tin tức",
    "title": "10 đội hoàn thiện nhất của NFL cho mùa giải 2024: 49ers mạnh mẽ; Browns, Jets có thể cạnh tranh - NFL.com",
    "url": "https://news.google.com/rss/articles/CBMhSIWh0dHBzOiIvd3d3Lm5mbC5jb20vbmV3cy9uZmwtcy10b3AtATAtbW9zdC1jb21wbGV0ZS10ZWZtcy1mb3ItMjAyNC1zZWZzb2Z5SAQA?oc=5"
  },
  {
    "tác giả": "",
    "description": "Những người sử dụng gói cũ hơn có thể mong đợi mức tăng $2 hoặc $5 mỗi dòng và sẽ có thông báo cho khách hàng bị ảnh hưởng vào thứ Tư.",
    "id": 25,
  }
}
```

Lấy dữ liệu và hiển thị bảng lên web:

```

@app.route('/table')
def show_table():
    try:
        # Thiết lập kết nối đến cơ sở dữ liệu
        with get_db_connection() as conn:
            cursor = conn.cursor()
            cursor.execute("SELECT * FROM articles")
            columns = [column[0] for column in cursor.description]
            data = [dict(zip(columns, row)) for row in cursor.fetchall()]
            return render_template('table.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('error.html', error=str(e))

```

Articles

id	source_id	source_name	author	title	description	url
22	cnn	CNN	Deblina Chakraborty	Boeing Starliner historic crewed launch delayed again indefinitely - CNN	After reporting a helium leak in the spacecraft, NASA has taken the next target date for the highly anticipated launch off the table — without immediately naming a new one.	https://www.cnn.com/2024/05/22/world/boeing-starliner-crewed-launch-delayed-indefinitely-scn/index.html
23	nbc-news	NBC News	Patrick Smith	Deadly tornado devastates Iowa town as severe weather moves east - NBC News	Central Iowa was reeling Wednesday morning after what police described as a "devastating" tornado laid waste to rural communities and killed an unknown number of people.	https://www.nbcnews.com/news/us-news/deadly-tornado-devastates-iowa-town-severe-weather-moves-south-rcna153430
				NFL's top 10 most complete teams for 2024		

Vẽ biểu đồ:

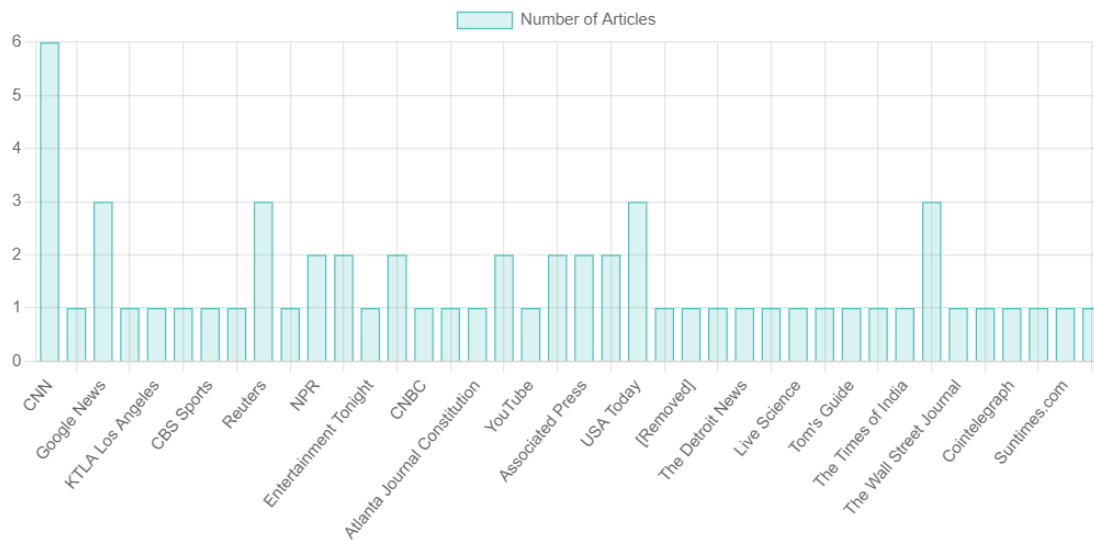
```

2   <html lang="en">
14  <body>
19    <a href="/">Back to Home</a>
20    <script>
21      $(document).ready(function() {
22        function fetchData() {
23          $.ajax({
24            url: '/data',
25            method: 'GET',
26            success: function(data) {
27              console.log('Data fetched:', data); // Debug: Log dữ liệu nhận được
28
29              const sourceCounts = data.reduce((acc, article) => {
30                acc[article.source_name] = (acc[article.source_name] || 0) + 1;
31                return acc;
32              }, {});
33
34              console.log('Source counts:', sourceCounts); // Debug: Log đếm nguồn
35
36              const labels = Object.keys(sourceCounts);
37              const dataCounts = Object.values(sourceCounts);
38
39              console.log('Labels:', labels); // Debug: Log nhãn
40              console.log('Data Counts:', dataCounts); // Debug: Log dữ liệu đếm
41
42              const ctx = document.getElementById('newsSourceChart').getContext('2d');
43
44              new Chart(ctx, {
45                type: 'bar',
46                data: {
47                  labels: labels,
48                  datasets: [{
49                    label: 'Number of Articles',
50                    data: dataCounts,
51                    backgroundColor: 'rgba(75, 192, 192, 0.2)',
52                    borderColor: 'rgba(75, 192, 192, 1)',
53                    borderWidth: 1

```

Ln 23, Co

Biểu đồ nguồn tin tức



CHƯƠNG IV. KẾT LUẬN

Đề tài "Theo dõi tin tức cập nhật từ nhiều nguồn" đã minh chứng sự hiệu quả của việc tích hợp các công nghệ hiện đại như Python, FastAPI, Node-Red, và SQL Server trong việc xây dựng một hệ thống thu thập, xử lý và hiển thị thông tin từ nhiều nguồn khác nhau. Qua quá trình thực hiện, em đã tiếp thu được khá nhiều kiến thức bổ ích như:

- Việc sử dụng Python kết hợp với FastAPI đã tạo ra một backend mạnh mẽ và linh hoạt, có khả năng xử lý và cung cấp dữ liệu theo thời gian thực. FastAPI không chỉ cung cấp các endpoint nhanh chóng và an toàn, mà còn hỗ trợ async/await giúp tối ưu hóa hiệu suất.
- Node-Red, với giao diện kéo thả trực quan, đã đơn giản hóa quy trình thu thập và xử lý dữ liệu từ nhiều nguồn. Khả năng tích hợp với các dịch vụ và API khác nhau của Node-Red giúp chúng ta nhanh chóng thiết lập các luồng dữ liệu phức tạp mà không cần viết quá nhiều mã.
- SQL Server đã cung cấp một cơ sở dữ liệu mạnh mẽ và ổn định cho hệ thống, giúp lưu trữ và truy xuất dữ liệu một cách hiệu quả. Việc sử dụng các truy vấn SQL để tổng hợp và phân tích dữ liệu đã cải thiện khả năng cung cấp thông tin chính xác và kịp thời.

Sau khi hoàn thành đề tài này đã giúp em hiểu thêm được rất nhiều về ngôn ngữ lập trình Python cùng với các công nghệ đã sử dụng trong đề tài.

Cuối cùng em xin cảm ơn Thầy Đỗ Duy Cốp đã hướng dẫn nhiệt tình để em có thể hoàn thành đề tài này.