

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA: ĐIỆN TỬ

Bộ môn: Tin Học Công Nghiệp



BÁO CÁO BÀI TẬP LỚN
MÔN HỌC

Môn : Lập trình Python
Giảng viên hướng dẫn : Đỗ Duy Cốp
Họ và tên nhóm sinh viên : Phommaly Dobee
Ngành học : Kỹ thuật máy tính
Lớp : K56KMT.01

Thái nguyên, 2024

TRƯỜNG ĐHKCN
KHOA ĐIỆN TỬ

CỘNG HÓA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

NHIỆM VỤ BÀI TẬP LỚN LẬP TRÌNH PYTHON

Sinh viên: **Phommaly Dobee**

Mã số sinh viên: **LAOS195098**

Lớp: **K56KMT.01**

Ngành: **Tin học công nghiệp**

Giáo viên hướng dẫn: **Đỗ Duy Cốp**

1. Tên đề tài:

xây dựng website theo dõi bảng xếp hàng bóng rổ NBA

2. Yêu cầu

- **Tìm hiểu về ngôn ngữ Python**
- **Nghiên cứu về Fastapi**
- **Cơ sở dữ liệu**
- **Sử dụng node red để gọi api lưu vào cơ sở dữ liệu**
- **Xây dựng web**

Ngày giao đề: **06/05/2024**. Ngày hoàn thành: **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 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

MỤC LỤC

MỤC LỤC	3
CHƯƠNG 1: TỔNG QUAN VỀ PYTHON	4
1.1. Khái niệm	4
1.2. Lịch sử	4
CHƯƠNG 2: FASTAPI	6
2.1. Khái niệm	6
2.2. Cấu trúc và kiến trúc của FastAPI	7
2.2.1. Asynchronous Programming.....	7
2.2.2. Dependency Injection.....	7
CHƯƠNG 3: MICROSOFT SQL	8
3.1. Khái niệm về ngôn ngữ SQL	8
3.2. SQL Sever Management Studio.	9
3.3. Mối quan hệ giữa SQL Sever và SQL Sever Management Studio.	9
CHƯƠNG 4: NODE-RED	11
4.1. Khái niệm tổng quan	11
4.2. Kiến Trúc và Cách Thức Hoạt Động	11
4.3. Các Thành Phần Chính	11
4.4. Khả Năng Kết Nối và Mở Rộng	11
4.5. Ứng Dụng Thực Tế	12
4.6. Triển Khai và Quản Lý	12
CHƯƠNG 5: LẬP TRÌNH VÀ KIỂM THỬ	13
5.1. Tìm kiếm API dữ liệu	13
5.2. Sử dụng sql	13
5.3. Xây dựng Node-red	14
5.4. Xây dựng giao diện	13
KẾT LUẬN	16

CHƯƠNG 1: TỔNG QUAN VỀ PYTHON

1.1. *Khái niệm*

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, van Rossum đã từ chức lãnh đạo trong cộng đồng ngôn ngữ Python sau 30 năm làm việc.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất.

1.2. *Lịch sử*

Python đã được Guido van Rossum tạo ra vào những năm 1980 tại Trung tâm Toán học – Tin học (Centrum Wiskunde & Informatica, CWI) ở Hà Lan như là một ngôn ngữ kế tục ngôn ngữ ABC – một ngôn ngữ được lấy cảm hứng từ SETL, có khả năng xử lý ngoại lệ và giao tiếp với hệ điều hành Amoeba. Nó bắt đầu được triển khai vào tháng 12 năm 1989. Van Rossum đã tự mình gánh vác trách nhiệm cho dự án, với vai trò là nhà phát triển chính, cho đến ngày 12 tháng 7 năm 2018, khi ông thông báo rằng ông sẽ rời bỏ trách nhiệm của ông và cả danh hiệu "Nhà độc tài nhân từ cho cuộc sống" của Python, một danh hiệu mà cộng đồng Python đã trao tặng cho ông vì sự tận tụy lâu dài của ông với vai trò là người ra quyết định chính cho dự án. Vào tháng 1 năm 2019, các nhà phát triển phân lõi Python đã bầu ra một "Hội đồng chèo lái" gồm năm thành viên để dẫn dắt dự án.

Python 2.0 được ra mắt vào ngày 16 tháng 10 năm 2000, với nhiều tính năng mới mẻ, bao gồm một bộ dọn rác phát hiện theo chu kỳ và khả năng hỗ trợ Unicode.

Python 3.0 được ra mắt vào ngày mùng 3 tháng 12 năm 2008. Đây là một phiên bản lớn của Python không tương thích ngược hoàn toàn. Nhiều tính năng lớn của nó đã được chuyển mã ngược (backport) về loạt phiên bản Python 2.6.x và 2.7.x. Các bản phát hành của Python 3 có đi kèm với công cụ 2to3, có tác dụng tự động hoá việc dịch mã Python 2 sang Python 3.

Python 3.9.2 và 3.8.8 được xúc tiến vì tất cả các phiên bản trước của Python (bao gồm cả 2.7) gặp một số vấn đề bảo mật, có thể dẫn đến thực thi mã từ xa và "đầu độc" bộ nhớ đệm. Trong năm 2022, Python 3.10.4 và 3.9.12 được xúc tiến cùng với 3.8.13 và 3.7.13, nguyên nhân là do một vài vấn đề về bảo mật. Khi Python 3.9.13 được phát hành vào tháng Năm năm 2022, loạt phiên bản 3.9 (cùng với loạt 3.8 và 3.7) được thông báo rằng sẽ chỉ nhận được các bản vá bảo mật trong tương lai. Vào ngày 7 tháng Chín năm 2022, bốn bản cập nhật mới được phát hành do có khả năng xảy ra một cuộc tấn công từ chối dịch vụ: 3.10.7, 3.9.14, 3.8.14 và 3.7.14.

Tính đến tháng 10 năm 2023, Python 3.12 là bản phát hành ổn định mới nhất. Một số thay đổi đáng chú ý từ bản 3.11 bao gồm các thay đổi về ngôn ngữ và thư viện chuẩn.

CHƯƠNG 2: FASTAPI

2.1. *Khái niệm*

FastAPI là một framework web hiện đại và nhanh chóng dành cho python, được thiết kế để dễ dàng sử dụng và có hiệu suất cao. Dưới đây là một số điểm nổi bật về FastAPI

1. Hiệu suất cao

FastAPI được xây dựng dựa trên Starlette và Pydantic, tận dụng tính năng của các thư viện này để cung cấp hiệu suất cao tương tự như các framework hàng đầu như node.js và Go. FastAPI sử dụng công nghệ asyncio của python để xử lý các yêu cầu không đồng bộ, giúp tối ưu hóa tốc độ xử lý.

2. Dễ Dàng Sử Dụng và Phát Triển Nhanh

FastAPI hỗ trợ rất tốt việc tự động tạo tài liệu API bằng cách sử dụng OpenAPI (còn gọi là Swagger). Bạn chỉ cần định nghĩa các endpoint của API bằng Python và FastAPI sẽ tự động sinh ra tài liệu API. Điều này giúp giảm thời gian phát triển và bảo trì.

3. Type Hints và Kiểm Tra Dữ Liệu

FastAPI tận dụng khả năng gợi ý kiểu dữ liệu (type hints) của Python để tự động kiểm tra và xác thực dữ liệu đầu vào. Điều này giúp giảm thiểu lỗi trong quá trình phát triển và nâng cao độ tin cậy của ứng dụng.

4. Dễ Tích Hợp

FastAPI dễ dàng tích hợp với nhiều hệ thống cơ sở dữ liệu, công cụ xác thực và các dịch vụ khác. Nó cung cấp các tính năng tích hợp sẵn như OAuth2, JWT, và hỗ trợ cho nhiều ORM phổ biến như SQLAlchemy và Tortoise-ORM.

5. Tài Liệu và Cộng Đồng Phong Phú

FastAPI có tài liệu rất phong phú và dễ hiểu, giúp người mới bắt đầu dễ dàng tiếp cận. Ngoài ra, cộng đồng người dùng và nhà phát triển FastAPI đang ngày càng phát triển, cung cấp nhiều tài nguyên và hỗ trợ.

2.2. Cấu trúc và kiến trúc của FastAPI

2.2.1. Asynchronous Programming

FastAPI được thiết kế để hỗ trợ lập trình bất đồng bộ (asynchronous programming). Điều này cho phép xử lý nhiều yêu cầu đồng thời mà không cần chờ đợi các yêu cầu trước đó hoàn thành, giúp cải thiện hiệu suất ứng dụng đáng kể.

2.2.2. Dependency Injection

FastAPI cung cấp một hệ thống Dependency Injection mạnh mẽ, giúp quản lý các phụ thuộc của ứng dụng một cách dễ dàng và hiệu quả. Điều này làm cho mã nguồn dễ kiểm tra và bảo trì hơn.



Hình: Python

CHƯƠNG 3: MICROSOFT SQL

3.1. *Khái niệm về ngôn ngữ SQL*

Người ta phải viết các chương trình ứng dụng để truy cập dữ liệu trong trường hợp của một hệ thống tập tin. Tuy nhiên, để quản lý cơ sở dữ liệu hệ thống có loại lập trình đặc biệt ngôn ngữ được gọi là ngôn ngữ truy vấn có thể được sử dụng để truy cập dữ liệu từ cơ sở dữ liệu. Structured query language (SQL) là ngôn ngữ truy vấn phổ biến nhất được sử dụng bởi các hệ thống quản lý cơ sở dữ liệu quan hệ lớn chẳng hạn như MySQL, ORACLE, SQL Server, v.v. Chúng ta có thể tạo và tương tác với cơ sở dữ liệu bằng SQL một cách hiệu quả và dễ dàng. Lợi ích với SQL là rằng chúng tôi không phải chỉ định cách lấy dữ liệu từ kho dữ liệu. Thay vào đó, chúng tôi chỉ đơn giản xác định những gì sẽ được truy xuất và SQL thực hiện phần còn lại. Mặc dù được gọi là truy vấn ngôn ngữ, SQL có thể làm được nhiều hơn ngoài việc truy vấn. SQL cung cấp các câu lệnh để xác định cấu trúc của dữ liệu, thao tác dữ liệu trong cơ sở dữ liệu, khai báo các ràng buộc và truy xuất dữ liệu từ cơ sở dữ liệu trong nhiều cách khác nhau, tùy thuộc vào yêu cầu của chúng tôi.

SQL là một ngôn ngữ mạnh mẽ và phổ biến trong việc quản lý cơ sở dữ liệu. Nó được sử dụng rộng rãi trong các ứng dụng web, ứng dụng di động và các ứng dụng máy tính khác.

SQL Server hỗ trợ các ngôn ngữ lập trình phổ biến như SQL, T-SQL, C# và Visual Basic .NET. Nó cung cấp một loạt các tính năng quản lý cơ sở dữ liệu, bao gồm quản lý dữ liệu, bảo mật, khả năng mở rộng và sao lưu phục hồi dữ liệu. Nó cung cấp một loạt các tính năng bảo mật, bao gồm xác thực người dùng, kiểm soát truy cập, mã hóa dữ liệu và kiểm tra việc thực thi các tác vụ quản lý cơ sở dữ liệu. Nó cũng hỗ trợ các tính năng khả năng mở rộng, bao gồm khả năng mở rộng theo chiều dọc và chiều ngang, cũng như khả năng tăng tốc độ truy xuất dữ liệu bằng cách sử dụng bộ đệm và index.



SQL Server cũng cung cấp các công cụ phân tích dữ liệu, cho phép người dùng thực hiện các tác vụ như phân tích dữ liệu, định dạng và hiển thị dữ liệu một cách trực quan. Bên cạnh đó, SQL Server cũng cho phép tích hợp với các ứng dụng Microsoft Office và các dịch vụ đám mây như Azure, cho phép người dùng truy cập cơ sở dữ liệu và thực hiện các tác vụ quản lý dữ liệu từ xa.

3.2. *SQL Sever Management Studio.*

SQL Server Management Studio (SSMS) là một công cụ quản lý cơ sở dữ liệu được cung cấp bởi Microsoft để quản lý các cơ sở dữ liệu SQL Server. SSMS là một giao diện đồ họa cho phép người dùng thực hiện các tác vụ quản lý cơ bản như tạo, xóa và chỉnh sửa cơ sở dữ liệu, bảng và chế độ xem. Nó cũng cho phép thực hiện các tác vụ phức tạp hơn như sao lưu và phục hồi dữ liệu, quản lý các chế độ bảo mật và tối ưu hóa hiệu suất cơ sở dữ liệu.

SSMS cung cấp một giao diện đồ họa và các công cụ để quản lý cơ sở dữ liệu, cho phép người dùng thực hiện các tác vụ quản lý bằng cách sử dụng các trình đơn, thanh công cụ và hộp thoại đồ họa. Nó cũng cung cấp một khu vực truy vấn SQL để thực hiện các truy vấn dữ liệu phức tạp.

SSMS cũng hỗ trợ tính năng tối ưu hóa hiệu suất cho cơ sở dữ liệu bằng cách cho phép người dùng tạo chỉ mục, chọn các phương pháp tối ưu hóa và giám sát hiệu suất của các truy vấn. Nó cũng cung cấp tính năng bảo mật để quản lý các chế độ bảo mật của cơ sở dữ liệu và người dùng được cấp quyền truy cập.

SSMS là một công cụ quan trọng cho những người làm việc với cơ sở dữ liệu SQL Server. Nó cung cấp các tính năng quản lý cơ bản và nâng cao để quản lý cơ sở dữ liệu, tối ưu hóa hiệu suất và bảo mật dữ liệu.

3.3. *Mối quan hệ giữa SQL Sever và SQL Sever Management Studio.*

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ, được phát triển bởi Microsoft. Nó cho phép người dùng lưu trữ, truy xuất và quản lý dữ liệu trong môi trường doanh nghiệp. SQL Server Management Studio (SSMS) là một công cụ quản lý SQL Server, cung cấp cho người dùng một giao diện đồ họa để thao tác với cơ sở dữ liệu.

SSMS cung cấp một loạt các tính năng quản lý cơ sở dữ liệu, cho phép người dùng thực hiện các tác vụ như tạo, sửa đổi hoặc xóa cơ sở dữ liệu, bảng dữ liệu, truy vấn và chức năng lưu trữ, cũng như quản lý bảo mật và tài khoản người dùng.

SQL Server và SSMS có mối quan hệ chặt chẽ, vì SSMS được phát triển bởi Microsoft để hỗ trợ người dùng quản lý cơ sở dữ liệu trên SQL

Server. Người dùng có thể sử dụng SSMS để kết nối và làm việc với SQL Server, truy vấn và quản lý cơ sở dữ liệu. Trong SSMS, người dùng có thể tạo các truy vấn SQL và thực hiện các tác vụ quản lý cơ sở dữ liệu một cách dễ dàng thông qua giao diện đồ họa.

Vì vậy, có thể nói SQL Server Management Studio là công cụ quản lý và truy vấn dữ liệu cho SQL Server, cung cấp một giao diện đồ họa để người dùng thao tác với cơ sở dữ liệu, còn SQL Server là hệ quản trị cơ sở dữ liệu quan hệ của Microsoft.

CHƯƠNG 4: NODE-RED

4.1. *Khái niệm tổng quan*

Node-red là một công cụ mã nguồn mở và trực quan được sử dụng để xây dựng các luồng làm việc (workflow) và ứng dụng Internet of Things(IoT).

4.2. *Kiến Trúc và Cách Thức Hoạt Động*

Node-RED cung cấp một giao diện người dùng đồ họa (GUI) để tạo ra các ứng dụng bằng cách kéo và thả các node (nút) vào một canvas (bảng vẽ) và kết nối chúng với nhau. Mỗi node thực hiện một nhiệm vụ cụ thể, chẳng hạn như xử lý dữ liệu, giao tiếp với thiết bị hoặc kết nối đến các dịch vụ web.

Flow-Based Programming

Node-RED sử dụng lập trình theo luồng (flow-based programming), cho phép xây dựng ứng dụng thông qua các luồng dữ liệu. Một luồng bao gồm các node được kết nối với nhau, mỗi node thực hiện một bước trong quá trình xử lý dữ liệu.

4.3. *Các Thành Phần Chính*

- Nodes

Inject: Khởi tạo một luồng dữ liệu bằng cách tạo ra một thông điệp mới.

Debug: Hiển thị thông tin gỡ lỗi trong bảng điều khiển.

Function: Thực hiện mã JavaScript tùy chỉnh để xử lý thông điệp.

HTTP Request: Gửi yêu cầu HTTP và xử lý phản hồi.

- Flows

Subflows: Nhóm các node lại với nhau để tái sử dụng trong nhiều luồng khác nhau, giúp quản lý luồng dữ liệu phức tạp dễ dàng hơn.

4.4. *Khả Năng Kết Nối và Mở Rộng*

- Library of Nodes

Node-RED có một thư viện phong phú với hàng ngàn node có sẵn, giúp dễ dàng tích hợp với các thiết bị IoT, cơ sở dữ liệu, dịch vụ web, và nhiều hệ thống khác. Người dùng cũng có thể tạo các node tùy chỉnh để phù hợp với nhu cầu riêng.

- Plugins và Modules

Node-RED hỗ trợ các plugin và module mở rộng, cho phép tích hợp thêm các tính năng và khả năng mới. Các module này có thể được cài đặt thông qua npm (Node Package Manager).

4.5. Ứng Dụng Thực Tế

Node-RED được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

- **Internet of Things (IoT):** Kết nối và điều khiển các thiết bị IoT, thu thập và phân tích dữ liệu từ các cảm biến.
- **Tự động hóa doanh nghiệp:** Tích hợp các hệ thống và dịch vụ khác nhau để tự động hóa quy trình kinh doanh.
- **Xử lý dữ liệu:** Thu thập, xử lý và chuyển đổi dữ liệu từ nhiều nguồn khác nhau.

4.6. Triển Khai và Quản Lý

Node-RED có thể được triển khai trên nhiều nền tảng khác nhau, bao gồm:

- **Máy tính cá nhân:** Cài đặt trên máy tính để bàn hoặc laptop để phát triển và thử nghiệm.
- **Thiết bị nhúng:** Chạy trên các thiết bị như Raspberry Pi để điều khiển và giám sát các thiết bị IoT.
- **Đám mây:** Triển khai trên các nền tảng đám mây như AWS, Azure, hoặc IBM Cloud để tận dụng khả năng mở rộng và tính sẵn sàng cao.

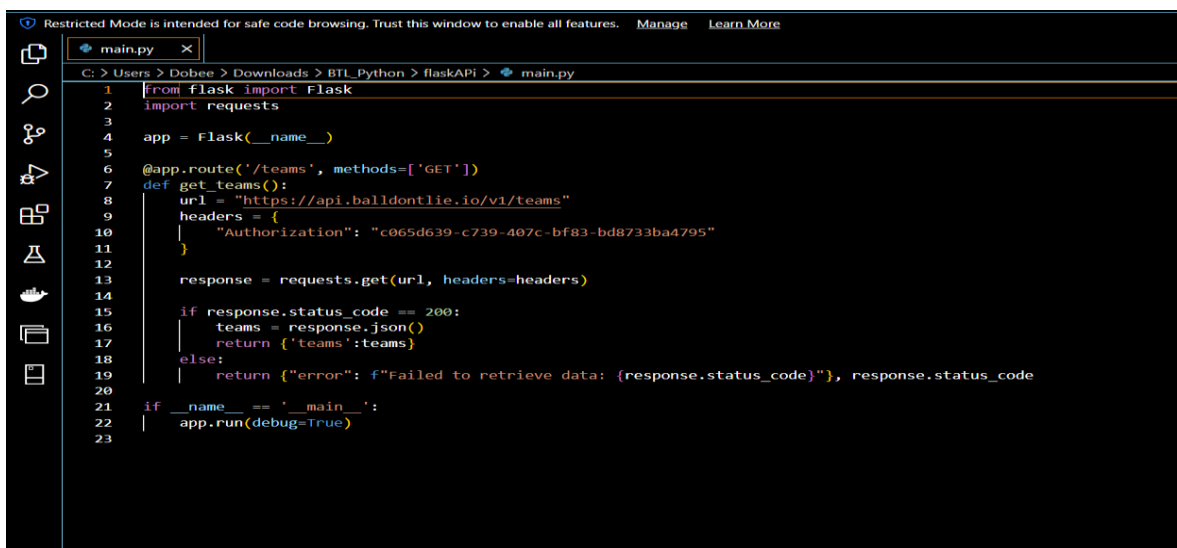
CHƯƠNG 5: LẬP TRÌNH VÀ KIỂM THỬ

5.1. Tìm kiếm API dữ liệu

API mà sử dụng trong việc xây dựng đề tài này tôi đã lấy từ trang web nguồn của BALLDONTLIE. Là một trang web mà người dùng có thể vào để xem các TEAM NBA

Link api : <https://www.balldontlie.io/v1/teams#teams>

Sau khi đã tìm xong api này tôi đã sử dụng python và fastapi để lấy dữ liệu. Code mẫu như sau:

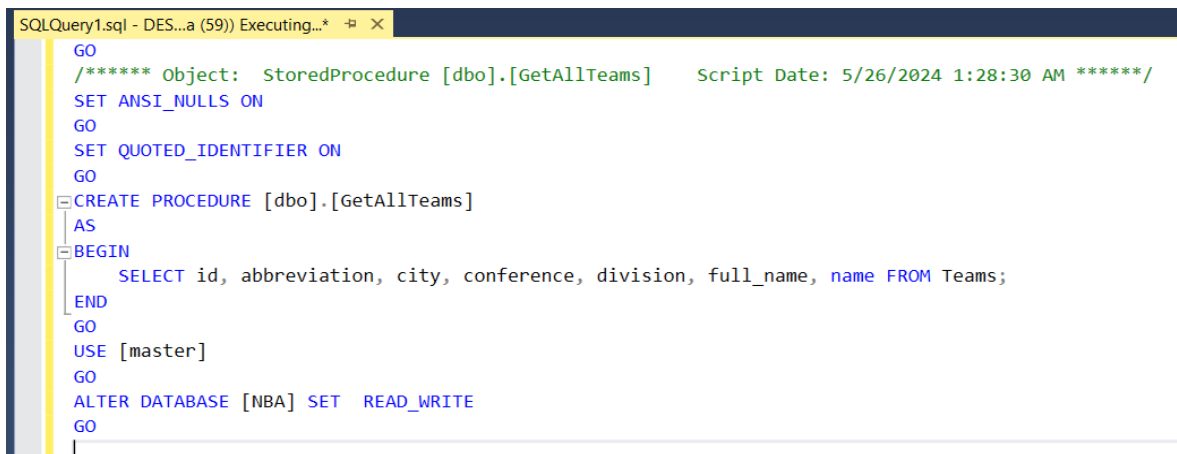


```

1 from flask import Flask
2 import requests
3
4 app = Flask(__name__)
5
6 @app.route('/teams', methods=['GET'])
7 def get_teams():
8     url = "https://api.balldontlie.io/v1/teams"
9     headers = {
10         "Authorization": "c065d639-c739-407c-bf83-bd8733ba4795"
11     }
12
13     response = requests.get(url, headers=headers)
14
15     if response.status_code == 200:
16         teams = response.json()
17         return {'teams': teams}
18     else:
19         return {"error": f"Failed to retrieve data: {response.status_code}"}, response.status_code
20
21 if __name__ == '__main__':
22     app.run(debug=True)
23
  
```

5.2 Sử dụng sql

Đối với những thông tin trên tôi đã tạo được bảng như sau:



```

SQLQuery1.sql - DES...a (59) Executing...
GO
/***** Object: StoredProcedure [dbo].[GetAllTeams]    Script Date: 5/26/2024 1:28:30 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[GetAllTeams]
AS
BEGIN
    SELECT id, abbreviation, city, conference, division, full_name, name FROM Teams;
END
GO
USE [master]
GO
ALTER DATABASE [NBA] SET READ_WRITE
GO
  
```

```

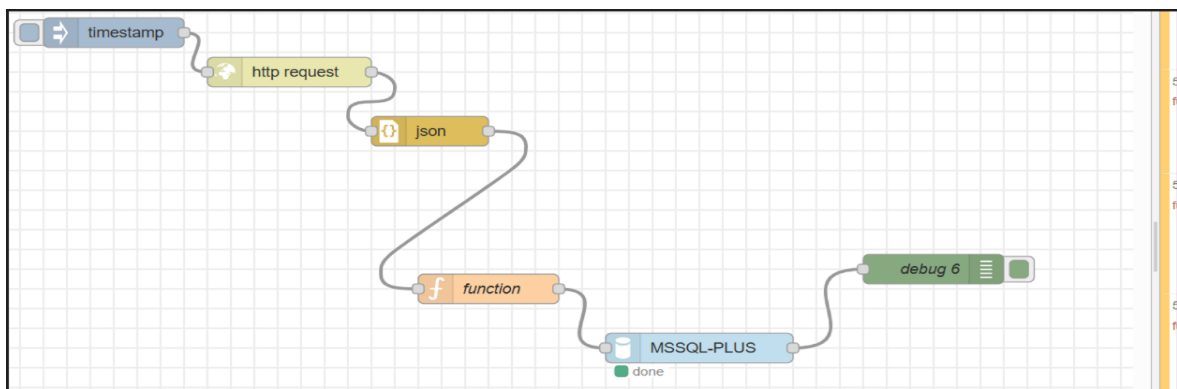
CREATE TABLE [dbo].[Teams](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [abbreviation] [varchar](255) NULL,
    [city] [varchar](255) NULL,
    [conference] [varchar](250) NULL,
    [division] [varchar](250) NULL,
    [full_name] [varchar](100) NULL,
    [name] [varchar](50) NULL,
    [point] [int] NULL,

    CONSTRAINT [PK__Teams__3213E83FABF45F00] PRIMARY KEY CLUSTERED
(

```

5.3 Xây dựng Node-red

Các node để có thể gọi api lưu vào cơ sở dữ liệu có như hình sau:



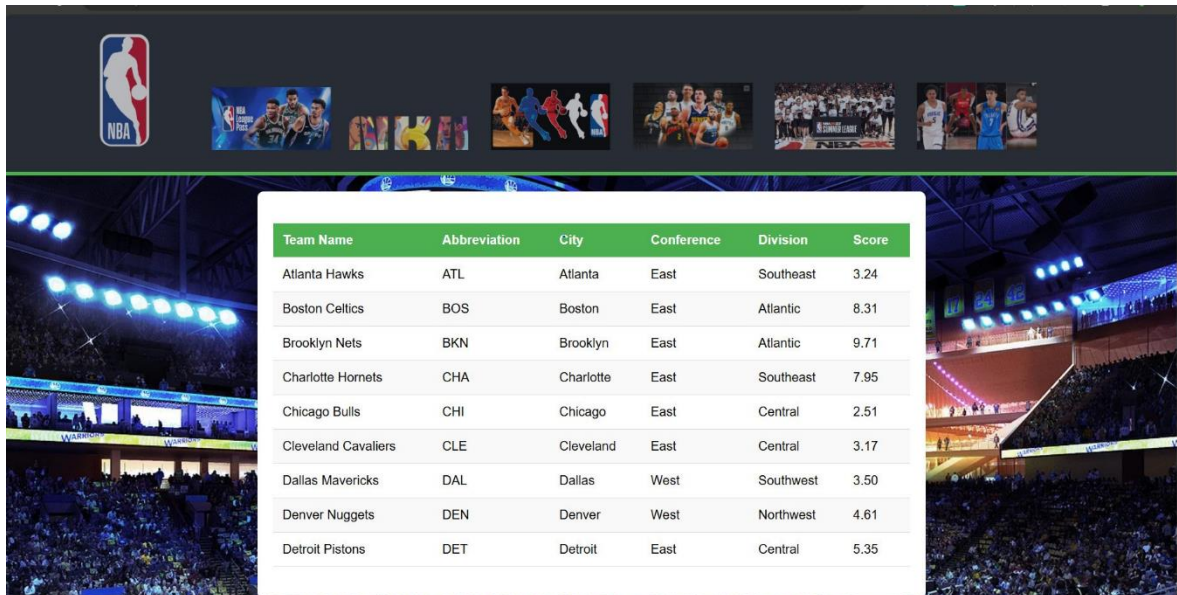
- Node Inject: Node này sẽ Chèn một tin nhắn vào luồng một cách thủ công hoặc theo các khoảng thời gian đều đặn. Dữ liệu tin nhắn có thể
- là nhiều loại khác nhau, bao gồm chuỗi, đối tượng JavaScript hoặc thời gian hiện tại.
- Node http request: cho phép bạn tương tác với các API web, tải xuống dữ liệu từ các trang web, hoặc gửi dữ liệu đến các dịch vụ web. Đây là node chính lấy dữ liệu từ Api mà ta đã tạo trong Python.
- Node Json : Chuyển đổi giữa Json và string.
- Node function: được sử dụng để viết các đoạn mã JavaScript tùy chỉnh để xử lý các tin nhắn (messages).
- Node MSSQL-PLUS : đây là node dùng để liên kết với SQL SEVER. Trong node này sẽ cần tùy chỉnh các thông tin để kết nối với cơ sở dữ liệu và thực hiện các truy vấn hoặc gọi tới Store Procedure.

- Node debug được sử dụng để hiển thị các thông tin gỡ lỗi trong quá trình phát triển các luồng (flows).

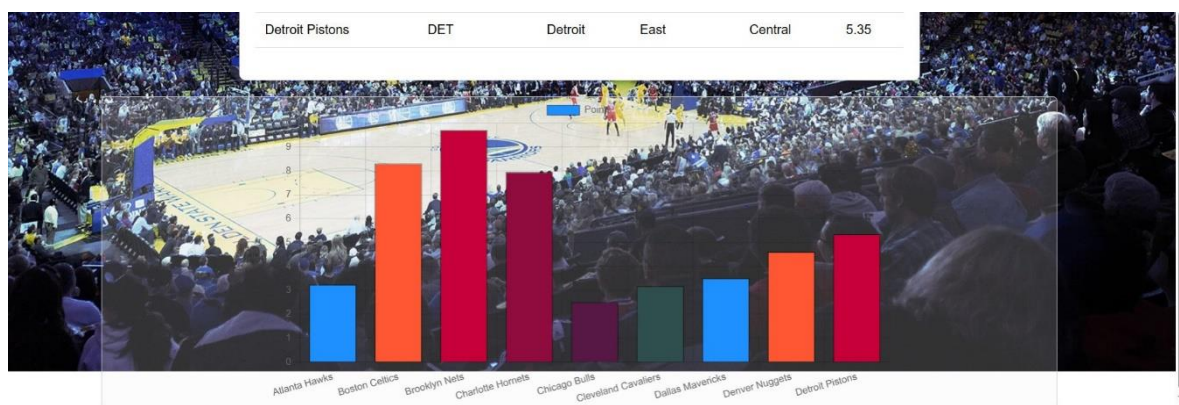
5.2. Xây dựng giao diện

Tôi phần này tôi đã sử dụng HTML, CSS, và Javascript để xây dựng giao diện hiển thị kết quả như trong ảnh sau:

Kết quả:



Team Name	Abbreviation	City	Conference	Division	Score
Atlanta Hawks	ATL	Atlanta	East	Southeast	3.24
Boston Celtics	BOS	Boston	East	Atlantic	8.31
Brooklyn Nets	BKN	Brooklyn	East	Atlantic	9.71
Charlotte Hornets	CHA	Charlotte	East	Southeast	7.95
Chicago Bulls	CHI	Chicago	East	Central	2.51
Cleveland Cavaliers	CLE	Cleveland	East	Central	3.17
Dallas Mavericks	DAL	Dallas	West	Southwest	3.50
Denver Nuggets	DEN	Denver	West	Northwest	4.61
Detroit Pistons	DET	Detroit	East	Central	5.35



KẾT LUẬN

Kết luận, xây dựng một trang web thông tin về vũ trụ và trái đất không chỉ là một dự án thú vị mà còn là một cơ hội để khám phá vô số điều kỳ diệu của vũ trụ. Sự kết hợp của các nguồn thông tin như RATING, TOP RATE không chỉ mở ra cánh cửa cho việc tìm hiểu sâu rộng về vũ trụ và hành tinh của chúng ta mà còn thúc đẩy sự hiểu biết và tò mò trong cộng đồng. Việc sử dụng API cung cấp cơ hội để khám phá và tận dụng nguồn tài nguyên vô tận từ không gian, đồng thời tạo ra những trải nghiệm trực quan và thú vị cho người dùng trên khắp thế giới. Đồng thời giúp cho bản thân em có thêm kiến thức về Python, thư viện FastAPI, Node-red và các kỹ năng tìm kiếm và sửa lỗi trong quá trình hoàn thành đề tài.