

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP



BÀI TẬP LỚN

LẬP TRÌNH PYTHON

Giảng viên hướng dẫn: **ĐỖ DUY CẬP**
Họ tên sinh viên: **Trần Thái Học**
Ngành học: **Kỹ thuật Máy tính**
MSSV: **K205510205187**
Lớp: **K56KMT.01**

Thái Nguyên 2023

PHIẾU GIAO ĐỀ TÀI BÀI TẬP LỚN MÔN HỌC
LẬP TRÌNH PYTHON

I. Thông tin sinh viên

Sinh viên thực hiện: TRẦN THÁI HỌC

Mã SV: K205510205187

II. Tên đề tài

XÂY DỰNG WEBSITE THEO DÕI THỜI GIAN SỬ DỤNG ỨNG DỤNG TRÊN LAPTOP

III. Mục tiêu

- Lấy thông tin ứng dụng đang sử dụng trên laptop
- Sử dụng FastAPI tạo một API trả về tên ứng dụng và thời gian thực
- Xây dựng một flow dùng để gọi FastAPI và lưu trữ vào cơ sở dữ liệu bằng Node-red và SQL Server
- Xây dựng trang web để hiển thị tổng quan về dữ liệu bằng các biểu đồ

IV. Nội dung thực hiện

1. Lấy thông tin ứng dụng đang sử dụng trên laptop bằng python
2. Tạo một cơ sở dữ liệu trong SQL Server để lưu trữ dữ liệu
3. Sử dụng Node-RED để xây dựng các luồng dữ liệu tự động, kết nối và xử lý dữ liệu từ FastAPI đến cơ sở dữ liệu.
4. Sử dụng các công nghệ front-end (HTML, CSS, JavaScript) để xây dựng giao diện người dùng.
5. Tạo biểu đồ thể hiện về phân loại ứng dụng sử dụng, top 5 ứng dụng dùng nhiều nhất và thời gian sử dụng laptop trong tuần.

V. Ngày giao nhiệm vụ: 15/5/20204

VI. Ngày hoàn thành: 25/5/2024

VII. Giáo viên hướng dẫn: Đỗ Duy Cốp

Mục lục

CHƯƠNG 1. TỔNG QUAN CHUNG	5
1.1 ĐẶT VẤN ĐỀ.....	5
1.2 Mục tiêu	5
1.3 Hướng giải quyết.....	5
CHƯƠNG 2. CÔNG CỤ VÀ PHẦN MỀM LẬP TRÌNH	7
2.1 SQL Server Management Studio	7
2.2 Ngôn ngữ lập trình Python	8
2.3 Visual Studio 2022	8
2.4 Node-red.....	10
CHƯƠNG 3. QUÁ TRÌNH THỰC HIỆN	11
3.1 SQL Server.....	11
3.1.1 Tạo cơ sở dữ liệu.....	11
3.1.2 Tạo procedure	12
3.2 FastAPI.....	14
3.2.1 Cài đặt FastAPI và thiết lập API.....	14
3.2.2 Chạy môi trường ảo thực thi FastAPI	15
3.3 Node-red.....	17
3.3.1 Cài đặt và thiết lập	17
3.4 Web	21
CHƯƠNG 4. TỔNG KẾT	25

NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày.....tháng.....năm 2024

CHỮ KÝ CỦA GIÁO VIÊN

(ký, ghi rõ họ tên)

CHƯƠNG 1. TỔNG QUAN CHUNG

1.1 ĐẶT VẤN ĐỀ

Trong thời đại công nghệ số hiện nay, máy tính xách tay (laptop) đã trở thành một công cụ không thể thiếu trong cuộc sống hàng ngày của chúng ta, từ công việc, học tập cho đến giải trí. Việc sử dụng laptop hiệu quả và có kế hoạch là yếu tố quan trọng giúp tăng năng suất lao động và học tập, đồng thời giảm thiểu các tác động tiêu cực như mệt mỏi, căng thẳng và các vấn đề về sức khỏe do sử dụng máy tính quá lâu. Tuy nhiên, nhiều người dùng hiện nay thường không nhận thức được thời gian họ dành cho từng ứng dụng cụ thể, dẫn đến việc phân bổ thời gian không hợp lý và giảm hiệu quả công việc. Do đó, nhu cầu theo dõi và quản lý thời gian sử dụng ứng dụng trên laptop trở nên cấp thiết hơn bao giờ hết. Dự án này nhằm phát triển một hệ thống theo dõi thời gian sử dụng ứng dụng, cung cấp cho người dùng những thông tin chi tiết và chính xác về thói quen sử dụng của họ. Qua đó, người dùng có thể điều chỉnh thời gian sử dụng hợp lý, nâng cao hiệu suất làm việc và cải thiện chất lượng cuộc sống.

1.2 Mục tiêu

Mục tiêu chính của dự án này là phát triển một công cụ theo dõi thời gian sử dụng ứng dụng trên laptop, nhằm cung cấp cho người dùng cái nhìn tổng quan và chi tiết về cách họ sử dụng thời gian trên thiết bị của mình. Cụ thể, dự án hướng tới những mục tiêu sau:

1. Ghi nhận thời gian sử dụng: Xây dựng hệ thống tự động ghi nhận thời gian sử dụng của từng ứng dụng trên laptop.
2. Xử lý luồng dữ liệu và 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.
3. Phân tích và báo cáo: Cung cấp các báo cáo chi tiết về thói quen sử dụng ứng dụng của người dùng, bao gồm phân loại ứng dụng sử dụng, top 5 ứng dụng sử dụng nhiều nhất, và thời gian sử dụng trong tuần. Hệ thống sẽ đưa ra các biểu đồ giúp người dùng dễ dàng theo dõi và phân tích dữ liệu.

1.3 Hướng giải quyết

1. Lấy dữ liệu về thông tin ứng dụng đang sử dụng bằng python, xử dụng fastAPI để tạo một API, trả về chuỗi json về tên ứng dụng và thời gian thực

2. Tạo một cơ sở dữ liệu để có thể lưu trữ về tên ứng dụng, kiểu ứng dụng và lịch sử ứng dụng
3. Sử dụng Node-red để gọi fastAPI, và lưu trữ nó và cơ sở dữ liệu
4. Sử dụng .net để tạo một trang web, hiển thị thống kê lên trang web.

CHƯƠNG 2. CÔNG CỤ VÀ PHẦN MỀM LẬP TRÌNH

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 2022

Microsoft Visual Studio là môi trường phát triển tích hợp (IDE) được thiết kế dành cho giới lập trình viên và các nhà phát triển ứng dụng. Đây là công cụ hỗ trợ phát triển phần mềm mạnh mẽ của Microsoft, cho phép người dùng viết, dịch mã và gỡ lỗi các ứng dụng dựa trên nhiều ngôn ngữ lập trình

khác nhau như C++, C#, Visual Basic, Python, JavaScript... Visual Studio bao gồm một trình biên tập mã nguồn, các công cụ gỡ lỗi và xây dựng ứng dụng đa nền tảng. Nó giúp tăng năng suất và hiệu quả công việc cho các lập trình viên.

Microsoft Visual Studio nổi bật với khả năng hỗ trợ một loạt các ngôn ngữ lập trình, bao gồm JavaScript, TypeScript, Python, C#, Java, Go, Ruby... Điều này biến nó trở thành công cụ lý tưởng cho các nhà phát triển làm việc trên nhiều dự án với đa ngôn ngữ lập trình. Ngoài ra, khả năng hỗ trợ đa ngôn ngữ của Microsoft Visual Studio còn giúp giới lập trình viên dễ dàng chuyển đổi giữa các ngôn ngữ và dự án mà không cần phải thay đổi môi trường làm việc, từ đó tiết kiệm thời gian, đồng thời tối ưu hiệu quả công việc.

Visual Studio 2022 là một phiên bản mới nhất của môi trường phát triển tích hợp (IDE) Visual Studio, được phát triển bởi Microsoft. Được công bố vào tháng 11 năm 2021, Visual Studio 2022 mang đến nhiều cải tiến và tính năng mới so với các phiên bản trước đó.



Visual Studio 2022 đi kèm với hỗ trợ đầy đủ cho .NET 6, bao gồm C# 10 và F# 6. .NET 6 là một phiên bản mới của nền tảng phát triển phần mềm .NET, với nhiều cải tiến về hiệu suất, độ ổn định và tính năng mới. Visual Studio 2022 tích hợp với GitHub Codespaces, cho phép bạn phát triển ứng dụng trực tiếp từ trình duyệt web mà không cần cài đặt môi trường phát triển trên máy cục bộ.

Visual Studio 2022 được tối ưu hóa về hiệu suất, bao gồm tăng tốc khởi động và thời gian phản hồi của các tính năng và công cụ.

Visual Studio 2022 là một bước tiến quan trọng trong việc cung cấp một môi trường phát triển hiệu quả và mạnh mẽ cho các nhà phát triển phần mềm,

với sự hỗ trợ đa nền tảng và tích hợp sâu sắc với các công nghệ và dịch vụ mới.

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.

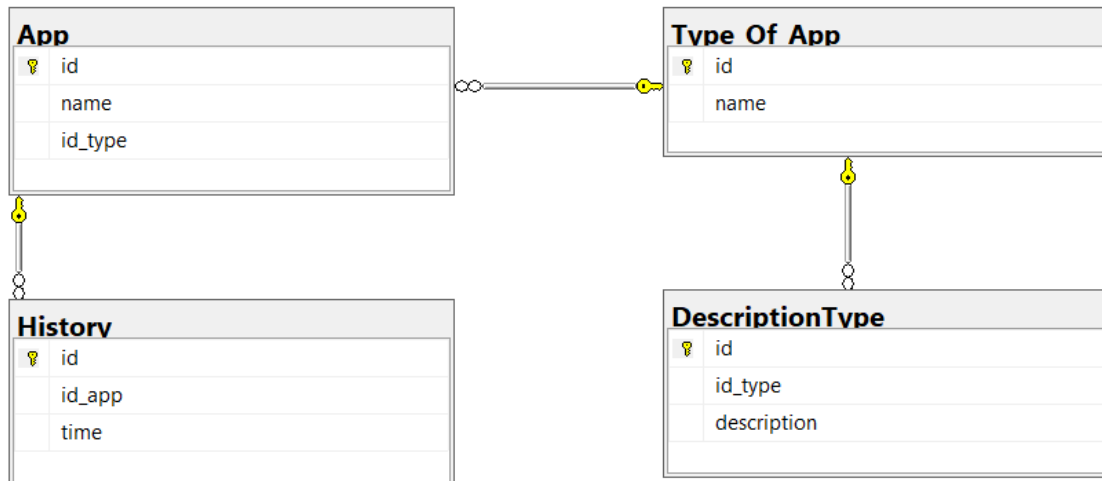
Node-RED cung cấp một cộng đồng lớn và sôi động, với nhiều nút và gói mở rộng được phát triển và chia sẻ miễn phí. Điều này giúp người dùng mở rộng và tùy chỉnh Node-RED theo nhu cầu và yêu cầu cụ thể của họ.

CHƯƠNG 3. QUÁ TRÌNH THỰC HIỆN

3.1 SQL Server

3.1.1 Tạo cơ sở dữ liệu

Tạo một cơ sở dữ liệu ‘SORMonitor’ với các bảng như sau:



Bảng App sẽ lưu thông tin về ứng dụng:

	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	name	nvarchar(500)	<input type="checkbox"/>
	id_type	int	<input type="checkbox"/>

Bảng Type_Of_App lưu thông tin về kiểu của ứng dụng:

	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	name	nvarchar(50)	<input checked="" type="checkbox"/>

Bảng DescriptionType lưu thông tin về mô tả của ứng dụng đó

	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	id_type	int	<input checked="" type="checkbox"/>
	description	nvarchar(500)	<input checked="" type="checkbox"/>

Bảng History lưu thông tin lịch sử ứng dụng

	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	id_app	int	<input checked="" type="checkbox"/>
	time	datetime	<input checked="" type="checkbox"/>

3.1.2 Tạo procedure

Tạo một procedure 'SP_SORMonitor' với các tham số sau:

```
Create PROCEDURE [dbo].[SP_SORMonitor]
@action nvarchar(50),
@name nvarchar(500) = null,
@time datetime = null
```

Khai báo ba biến để lưu kết quả truy vấn và lấy được thứ hai và Chủ Nhật trong tuần với ngày hiện tại

```
declare @json nvarchar(max) = ''
declare @monday date
declare @sunday date
```

Xử lý biến @action với @action = 'getinfo':

```
declare @temp_id int
if exists (select * from App where [name] like '%' + @name + '%' and LEN([name]) >= 0.9 * LEN(@name))
begin
    select @temp_id = id from App where [name] = @name
    insert into History (id_app, [time])
    values (@temp_id, @time)
end
else
begin
    DECLARE @id_type INT;
    SELECT @id_type = id_type
    FROM DescriptionType
    WHERE @name LIKE '%' + [description] + '%';

    IF @id_type IS NULL
        SET @id_type = 4;

    INSERT INTO App ([name], id_type)
    VALUES (@name, @id_type);

    select @temp_id = id from App where [name] = @name

    INSERT INTO History (id_app, [time])
    VALUES (@temp_id, @time);
end
set @json = FORMATMESSAGE('{"ok":true,"msg": "Thành công"}')
select @json as json
```

Chúng ta sẽ kiểm tra xem ứng dụng này đã có trong cơ sở dữ liệu (cụ thể trong bảng App) chưa, nếu có, thêm dữ liệu vào bảng History, còn không, thiết lập biến @id_type nhằm xác định kiểu dữ liệu của ứng dụng thông qua bảng Description, sau đó thêm vào bảng App và History.

Xử lý biến @action với @action = 'piechart' (thống kê thời gian sử dụng theo kiểu ứng dụng):

```

else if(@action = 'piechart')
begin
    set @monday = DATEADD(DAY, - datediff(day,0, @time) % 7, @time)
    set @sunday = DATEADD(DAY, 6 - datediff(day,0, @time) % 7, @time)
    select @json += FORMATMESSAGE(N'{"name": "%s", "time": %d}', t.[name], ROUND(COUNT(*) / 60, 0))
    FROM History h
    JOIN App a ON h.id_app = a.id
    JOIN Type_Of_App t ON a.id_type = t.id
    WHERE h.time BETWEEN @monday AND @sunday
    GROUP BY t.name;
    if((@json is null)or(@json=''))
        select N'{"ok":false,"msg":"không có dữ liệu","datas":[]}' as json;
    else
    begin
        select @json=REPLACE(@json,'(null)','null')
        select N'{"ok":true,"msg":"ok","datas":['+left(@json,len(@json)-1)+']}' as json;
    end
end

```

Lấy thời điểm thứ hai và chủ Nhật trong tuần của thời điểm hiện tại, hợp các bảng App, Type_Of_App và History, đếm số lượng dòng được nhóm bởi tên kiểu ứng dụng và trả về chuỗi json.

Xử lý biến @action với @action = 'linechart_top5' (thống kê top 5 ứng dụng sử dụng nhiều nhất):

```

else if(@action = 'linechart_top5')
begin
    set @monday = DATEADD(DAY, - datediff(day,0, @time) % 7, @time)
    set @sunday = DATEADD(DAY, 6 - datediff(day,0, @time) % 7, @time)
    select top 5 @json += FORMATMESSAGE(N'{"id": %d, "name": "%s", "time": %d}', h.id_app, a.[name], ROUND(COUNT(*) / 60, 0))
    FROM History h
    JOIN App a ON h.id_app = a.id
    WHERE h.time BETWEEN @monday AND @sunday
    GROUP BY h.id_app, a.name order by COUNT(*) desc;
    if((@json is null)or(@json=''))
        select N'{"ok":false,"msg":"không có dữ liệu","datas":[]}' as json;
    else
    begin
        select @json=REPLACE(@json,'(null)','null')
        select N'{"ok":true,"msg":"ok","datas":['+left(@json,len(@json)-1)+']}' as json;
    end
end

```

Lấy thời điểm thứ hai và chủ Nhật trong tuần của thời điểm hiện tại, hợp các bảng App, và History, đếm số lượng dòng được nhóm bởi id_app, tên ứng dụng và sắp xếp giảm dần.

Xử lý biến @action với @action = 'linechart_usedtime' (thống kê thời gian sử dụng ứng dụng trong tuần):

```

else if (@action = 'linechart_usedtime')
begin
    set @monday = DATEADD(DAY, - datediff(day,0, @time) % 7, @time)
    set @sunday = DATEADD(DAY, 6 - datediff(day,0, @time) % 7, @time)
    select @json += FORMATMESSAGE(N'{"name": "%s", "time": %d}',
    CAST(CAST(h.time AS DATE) AS varchar), ROUND(COUNT(*) / 60, 0))
    FROM History h
    WHERE h.time BETWEEN @monday AND @sunday
    GROUP BY CAST(h.time AS DATE)
    ORDER BY CONVERT(DATE, h.time) ASC;
    if((@json is null)or(@json=''))
        select N'{"ok":false,"msg":"không có dữ liệu","datas":[]}' as json;
    else
    begin
        select @json=REPLACE(@json,'(null)','null')
        select N'{"ok":true,"msg":"ok","datas":['+left(@json,len(@json)-1)+']}' as json;
    end
end

```

Lấy thời điểm thứ hai và chủ Nhật trong tuần của thời điểm hiện tại, đếm số lượng dòng được nhóm bởi thời gian sử dụng và sắp xếp theo thứ tự tăng dần (ngày tăng dần).

3.2 FastAPI

3.2.1 Cài đặt FastAPI và thiết lập API

Để có thể tạo API thông qua FastAPI, chúng ta sẽ cần tải về thư viện FastAPI

```
pip install fastapi
```

Sau đó import các thư viện cần thiết (trong đó có fastAPI)

```

@app.get("/getinfo")
def getstatus():
    data = {}
    try:
        hwnd = win32gui.GetForegroundWindow()
        _, pid = win32process.GetWindowThreadProcessId(hwnd)

        process = psutil.Process(pid)
        executable_name = process.name()

        window_title = win32gui.GetWindowText(hwnd)
        if(executable_name.find('chrome') == 0 or executable_name.find('browser') == 0):
            suffixes = [" - Google Chrome", " - Cốc Cốc"]
            for suffix in suffixes:
                if(window_title.endswith(suffix)):
                    window_title = window_title[:-len(suffix)]
                    break
            data["name"] = window_title
        else:
            data["name"] = executable_name
        data["time"] = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    except Exception as ex:
        data["name"] = False
        data["time"] = False
    return data

```

Chúng ta sẽ lấy cửa sổ của ứng dụng đang hoạt động bằng hàm **win32gui.GetForegroundWindow()**, sau đó lấy tiến trình liên kết với cửa sổ foreground qua hàm **win32process.GetWindowThreadProcessId(hwnd)**. Tiếp theo sử dụng **psutil.Process(pid)** để tạo một đối tượng tiến trình cho pid, sau đó chúng ta có thể lấy được thông tin file thực thi của chương trình thông qua **process.name()**.

Đối với trình duyệt, chúng ta sẽ lấy thông tin về tiêu đề cửa sổ **win32gui.GetWindowText()**, cắt đi phần tiêu đề có 'Google Chrome' hoặc 'Cốc Cốc'.

Cuối cùng sẽ trả về chuỗi json gồm tên ứng dụng và thời gian tại thời điểm đó.

3.2.2 Chạy môi trường ảo thực thi FastAPI

Đi tới đường dẫn như sau:

Name	Date modified	Type	Size
__pycache__	5/19/2024 10:10 AM	File folder	
activate	5/7/2024 11:49 AM	File	3 KB
activate	5/7/2024 11:49 AM	Windows Batch File	1 KB
Activate	5/7/2024 11:49 AM	Windows PowerShell ...	26 KB
deactivate	5/7/2024 11:49 AM	Windows Batch File	1 KB
dotenv	5/7/2024 11:51 AM	Application	106 KB
email_validator	5/7/2024 11:51 AM	Application	106 KB
fastapi	5/7/2024 11:51 AM	Application	106 KB
httpx	5/7/2024 11:51 AM	Application	106 KB
markdown-it	5/7/2024 11:51 AM	Application	106 KB
normalizer	5/13/2024 10:13 PM	Application	106 KB
pip	5/7/2024 11:49 AM	Application	106 KB
pip3.12	5/7/2024 11:49 AM	Application	106 KB
pip3	5/7/2024 11:49 AM	Application	106 KB
pygmentize	5/7/2024 11:51 AM	Application	106 KB
python	5/7/2024 11:48 AM	Application	269 KB
pythonw	5/7/2024 11:48 AM	Application	258 KB
pywin32_postinstall	5/19/2024 10:10 AM	Python Source File	27 KB
pywin32_testall	5/19/2024 10:10 AM	Python Source File	4 KB
typer	5/7/2024 11:51 AM	Application	106 KB
unidecode	5/14/2024 10:49 AM	Application	106 KB
uvicorn	5/7/2024 11:51 AM	Application	106 KB
watchfiles	5/7/2024 11:51 AM	Application	106 KB

Nhấp chuột vào đường dẫn và gõ 'cmd', một cửa sổ cmd hiện ra, sau đó chúng ta thực hiện lệnh 'activate'.

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\stormer\Documents\FasAPI\.venv\Scripts>activate

(.venv) C:\Users\stormer\Documents\FasAPI\.venv\Scripts>

```

Như vậy, chúng ta đã thiết lập được môi trường ảo, bây giờ cần quay về thư mục có chứa file đã code fastAPI bằng lệnh cd 'đường dẫn thư mục'


```
C:\Windows\System32\cmd.exe - uvicorn main:app
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\stormer\Documents\FasAPI\.venv\Scripts>activate

(.venv) C:\Users\stormer\Documents\FasAPI\.venv\Scripts>cd C:\Users\stormer\Documents\FasAPI

(.venv) C:\Users\stormer\Documents\FasAPI>uvicorn main:app
+ [32mINFO+ [0m:      Started server process [+ [36m2824+ [0m]
+ [32mINFO+ [0m:      Waiting for application startup.
+ [32mINFO+ [0m:      Application startup complete.
+ [32mINFO+ [0m:      Uvicorn running on +[1mhttp://127.0.0.1:8000+ [0m (Press CTRL+C to quit)
```

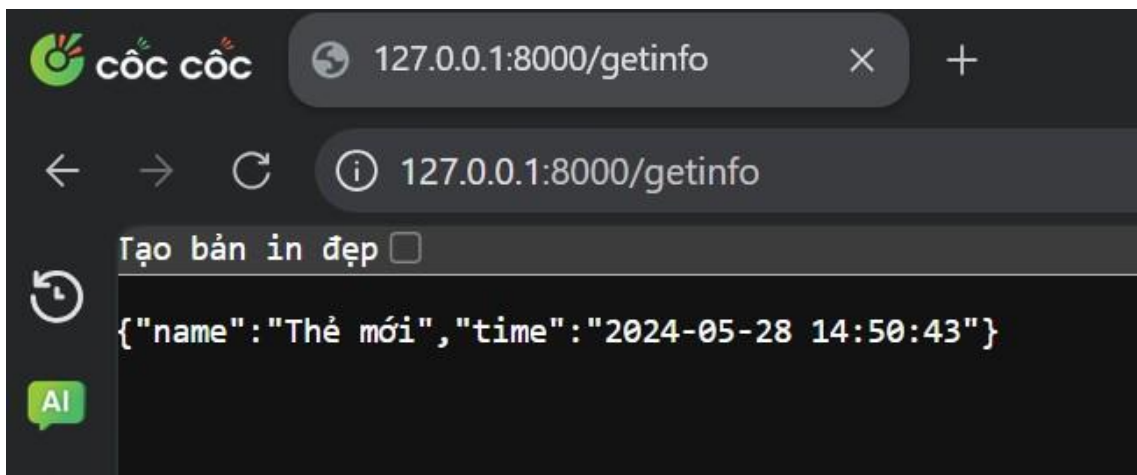
Sau đó chúng ta thực hiện gõ `uvicorn main:app`.

Trong đó: `uvicorn` là thư viện dùng để tạo một server thực thi `fastapi`

Main là tên file chứa code `fastAPI`

App là instance thực thi `fastAPI`

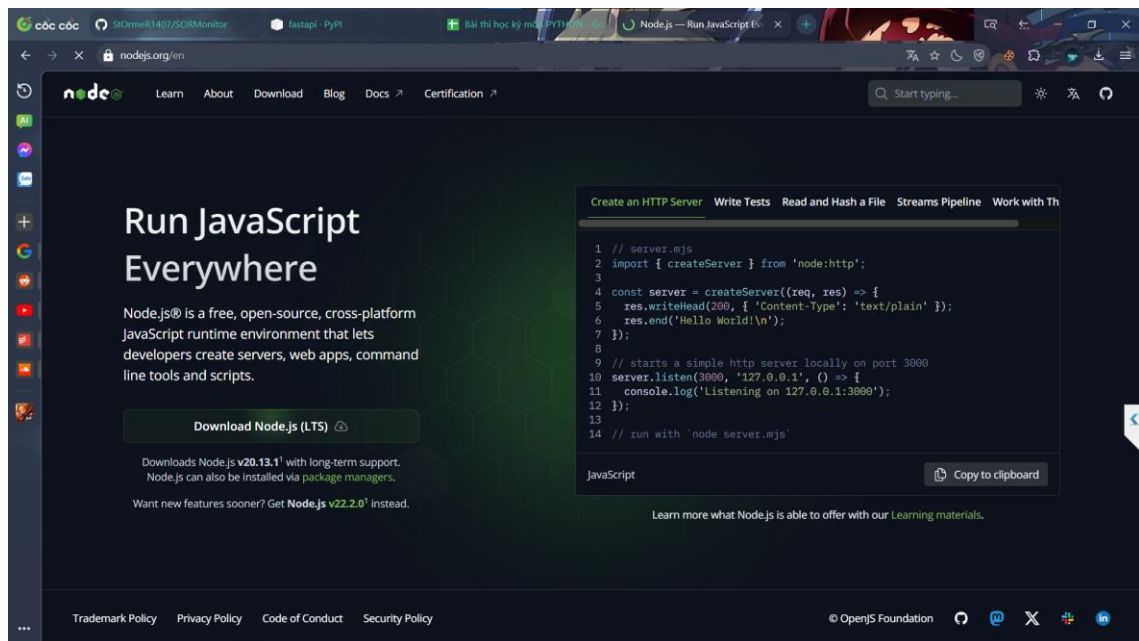
Để kiểm tra xem `fastAPI` đã chạy chưa, mở trình duyệt và nhập đường dẫn <https://1237/.0.0.1:8000/getinfo> (cổng có thể thay đổi tùy thuộc vào người dùng và `getinfo` là một ví dụ về một endpoint)



3.3 Node-red

3.3.1 Cài đặt và thiết lập

Trước tiên thì cần phải cài `nodejs`



Sau khi cài đặt xong thì cài đặt tiếp đến nodered:

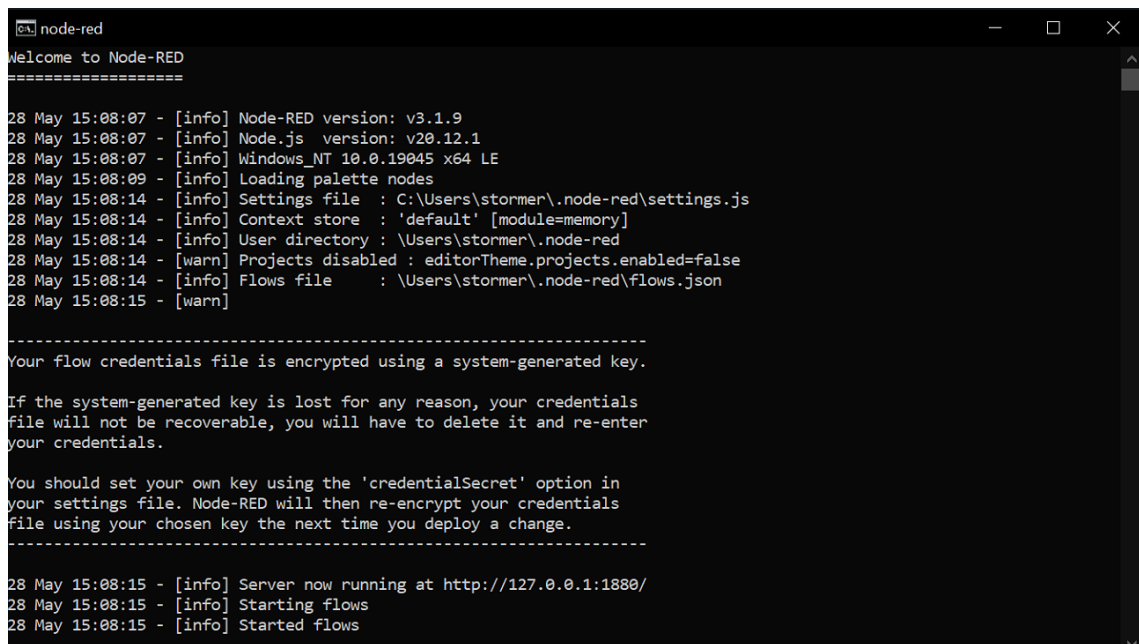
Installing with npm

To install Node-RED you can use the `npm` command that comes with node.js:

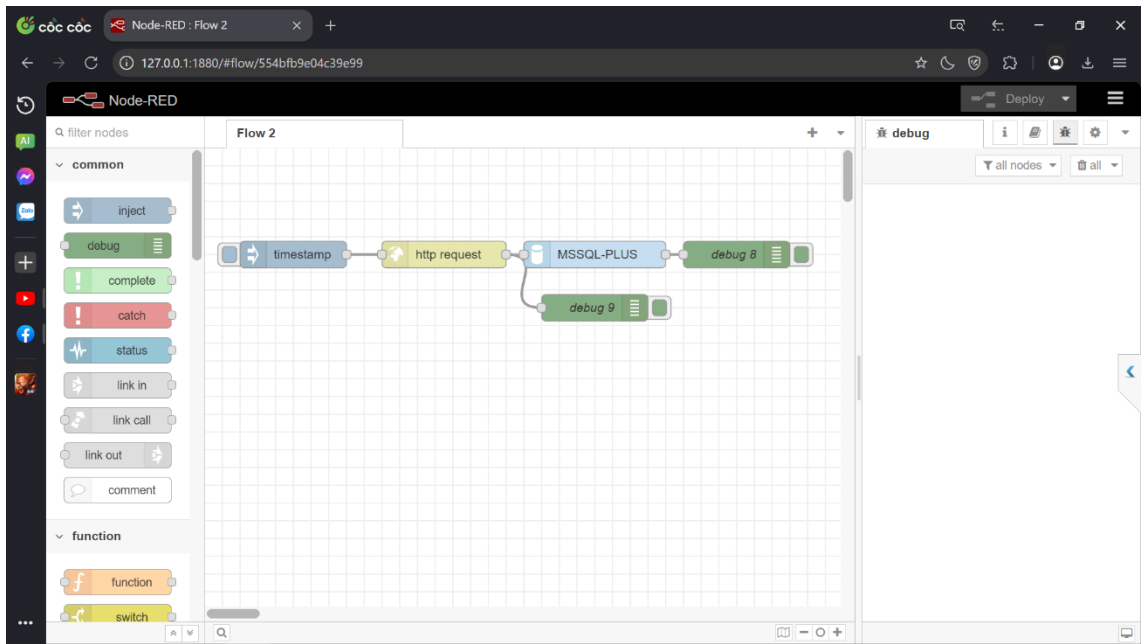
```
sudo npm install -g --unsafe-perm node-red
```

If you are using Windows, do not start the command with `sudo`.

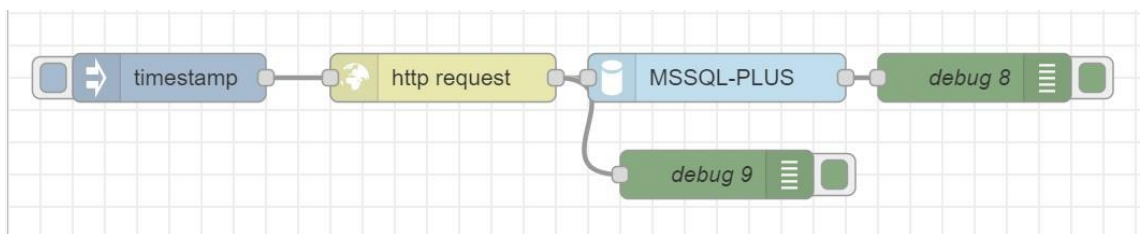
Sau khi cài đặt xong, mở cmd và nhập node-red:



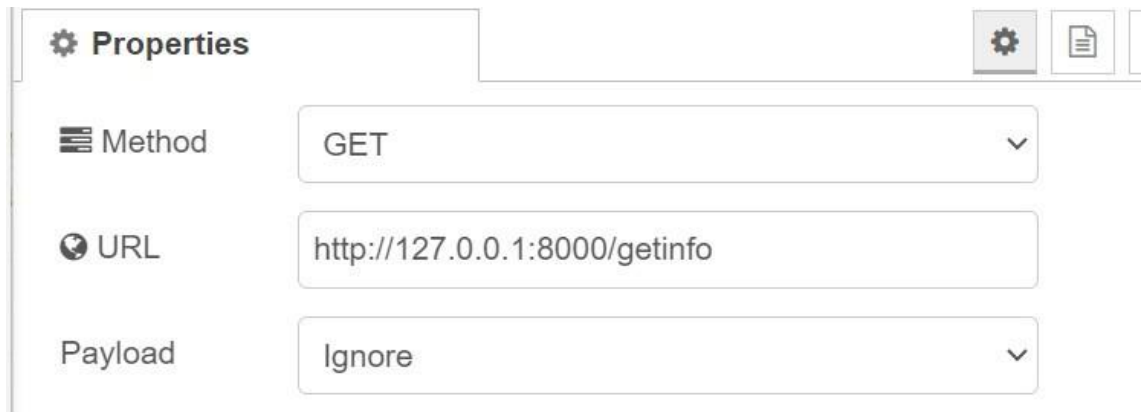
Mở trình duyệt và gõ đường dẫn: 127.0.0.1:1880



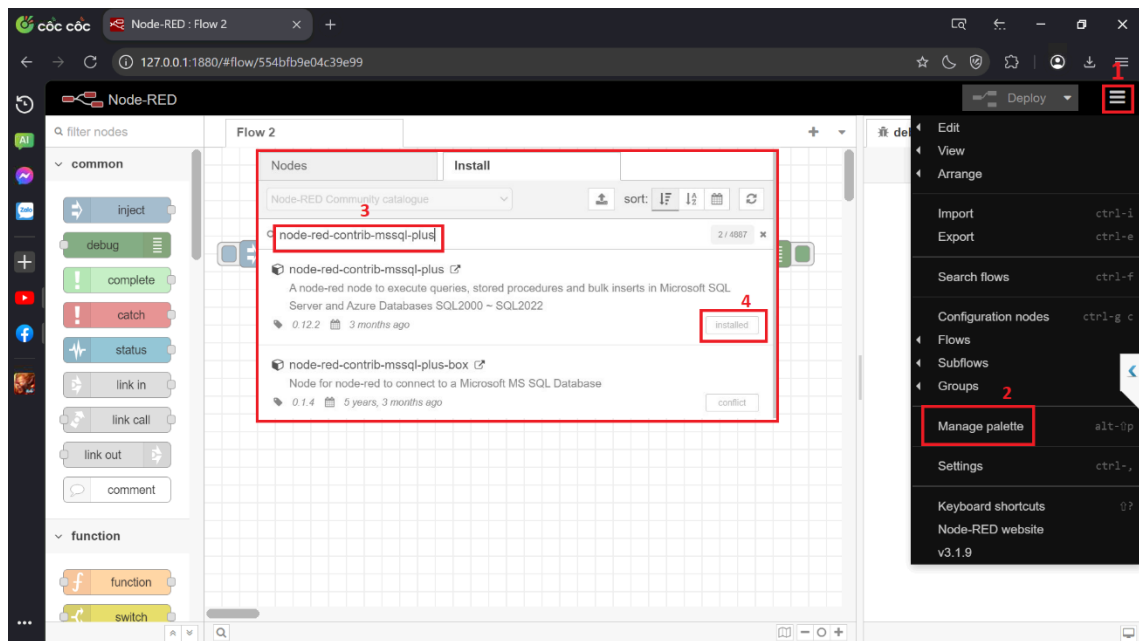
Chúng ta hãy tập trung vào flow chính:



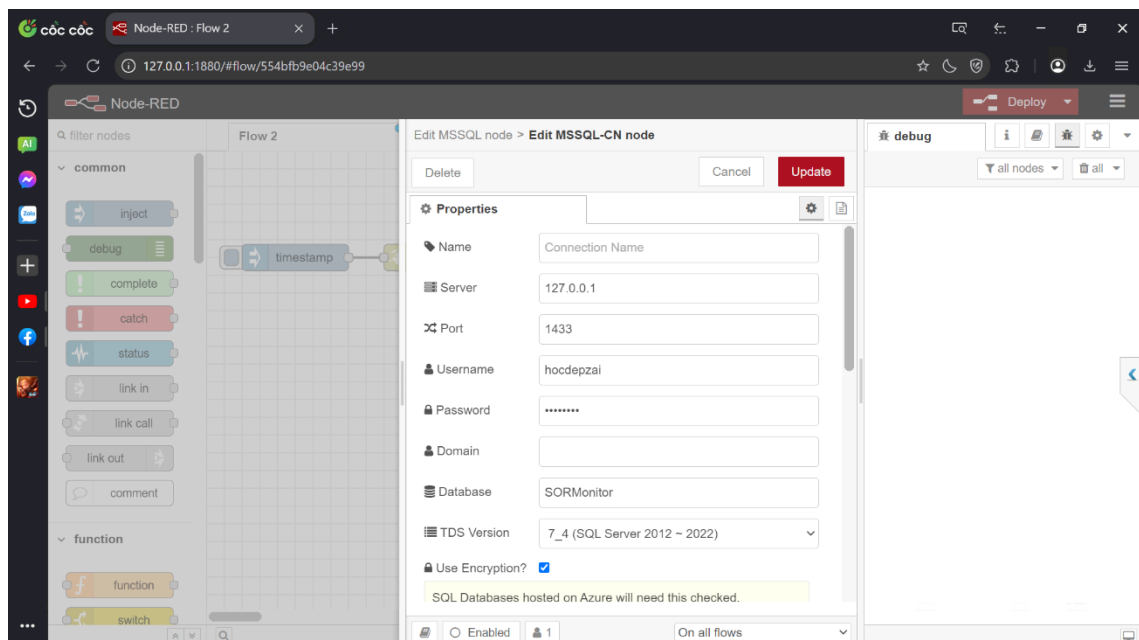
Với mỗi thời điểm một giây, chúng ta sẽ gọi tới FastAPI (thông qua block http request)



Sau khi gọi tới FastAPI, dữ liệu sẽ trả về json, chúng ta sẽ lấy dữ liệu đó thực hiện lưu vào sql bằng khối mssql, cài bên ngoài bằng cách như sau:



Sau khi đã cài đặt sau, chúng ta sẽ edit kết nối tới sql server:



Với các tham số: Server : Địa chỉ kết nối tới sql server (ở đây là localhost)

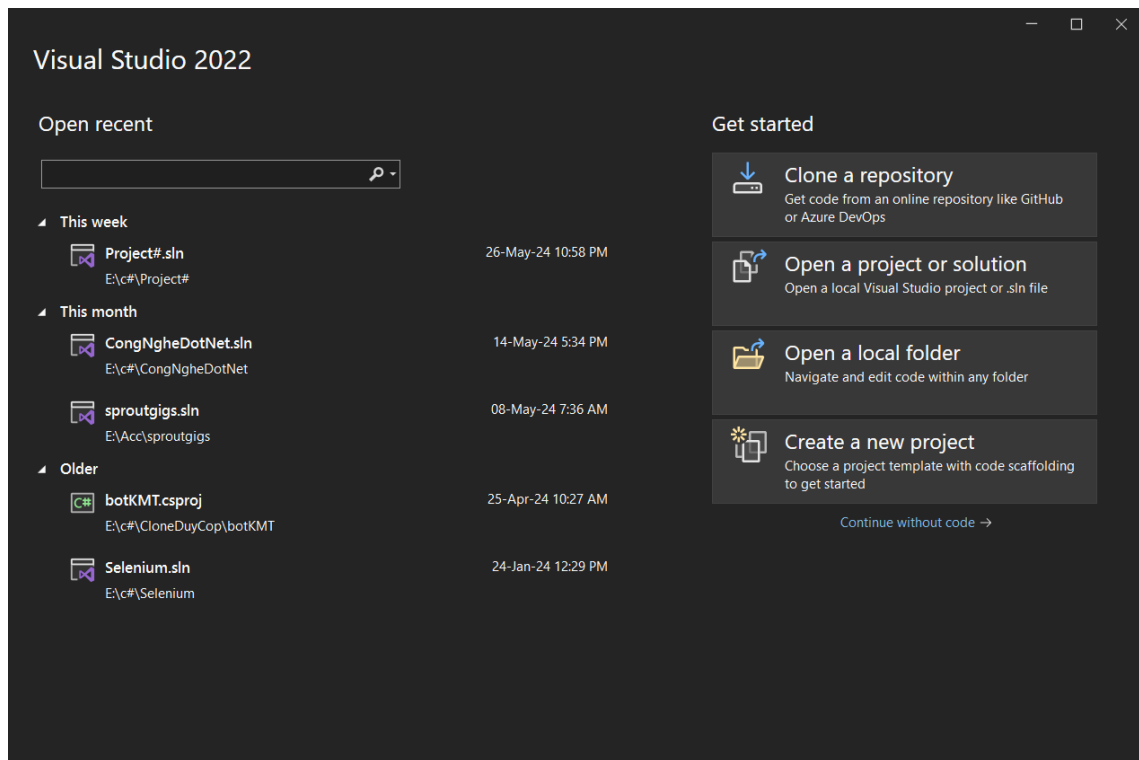
Port: 1433 (có thể thay đổi trong phần SQL Server Configuration Manager -> SQL Server Network Configuration -> Protocols for MSSQLServer -> Properties -> IP Addresses)

Username: tên tài khoản đăng nhập

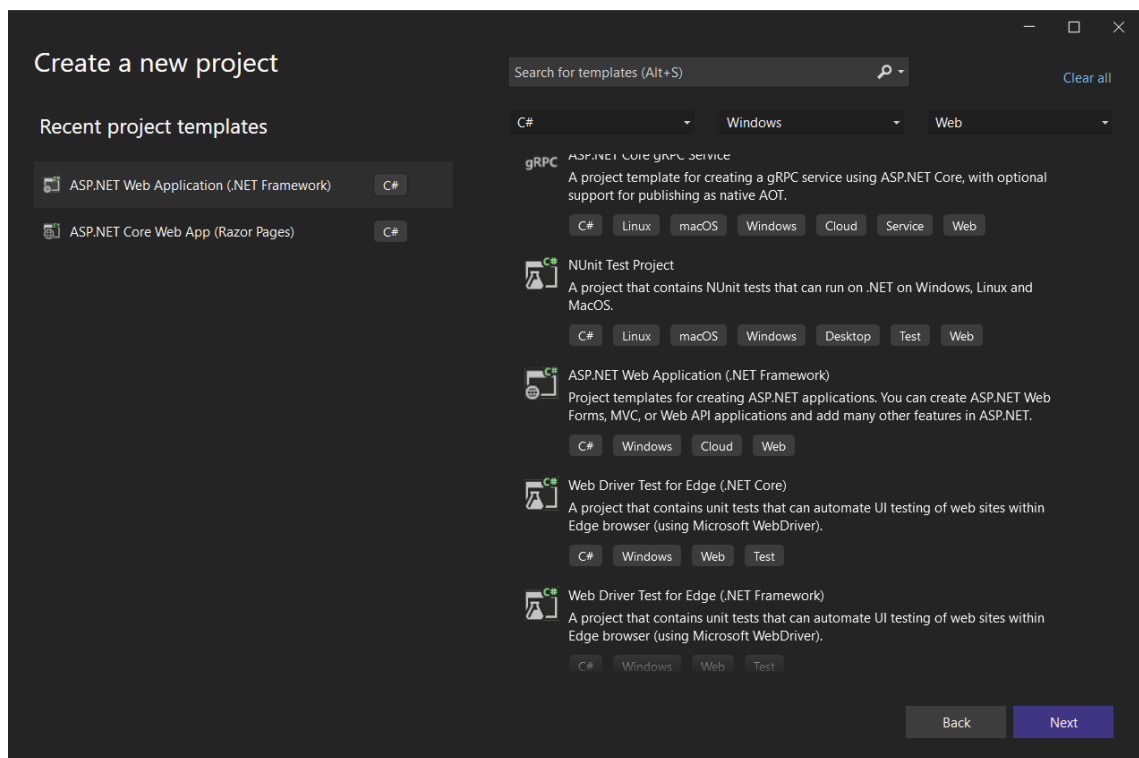
Password: mật khẩu đăng nhập

Database: Cơ sở dữ liệu chúng ta kết nối tới

Với procedure là SP_Monitor, các tham số truyền vào là action, name, và time



Chọn ASP.NET Web Application (.NET Framework):



Nhấp chuột trái vào Project -> Add -> Web Form

Chúng ta sử dụng 3 lần post, mỗi lần post đều tới api.aspx với từng action khác nhau:

```

$(document).ready(function () {
    $.post("api.aspx",
        {
            action : 'piechart',
            time: getCurrentDate()
        }
        , function (data)...)

    $.post("api.aspx",
        {
            action: 'linechart_top5',
            time: getCurrentDate()
        }
        , function (data)...)

    $.post("api.aspx",
        {
            action: 'linechart_usedtime',
            time: getCurrentDate()
        }
        , function (data)...)
})

```

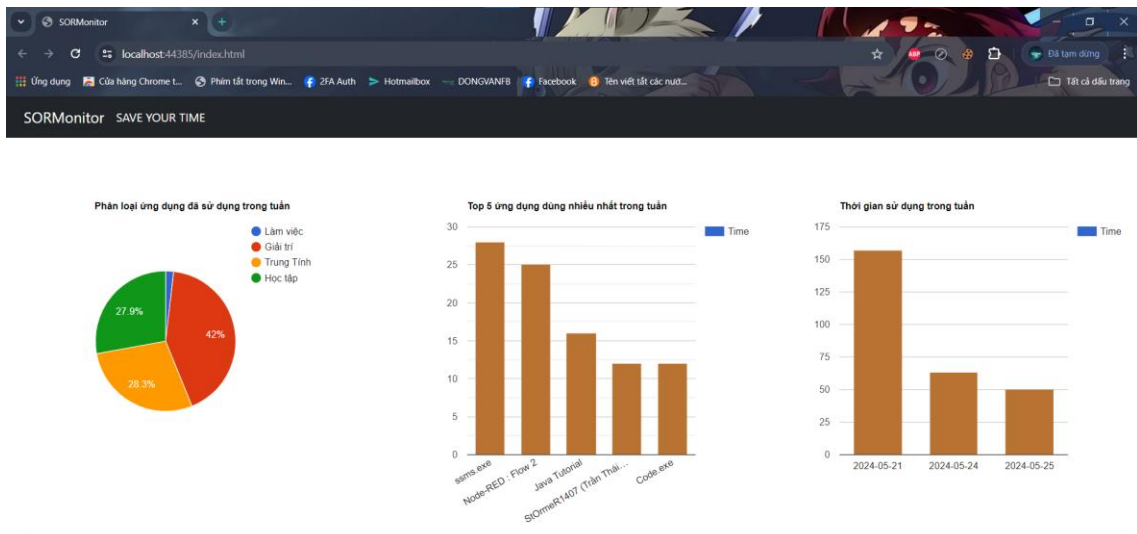
Api.aspx.cs sẽ nhận được các tham số được gửi lên và thực thi truy xuất dữ liệu của SQL Server:

```

string action = Request["action"];
try
{
    SqlCommand cm = GetCmd("SP_SORMonitor", action);
    switch (action)
    {
        case "piechart":
        case "linechart_top5":
        case "linechart_usedtime":
            cm.Parameters.Add("@time", SqlDbType.Date).Value = Request["time"];
            break;
    }
    json = (string)Scalar(cm);
}
catch (Exception ex)
{
    reply.ok = false;
    reply.msg = ex.Message;
    json = JsonConvert.SerializeObject(reply);
}
Response.Write(json);

```

Sau khi đã lấy được dữ liệu, xử lí một vài thông tin và có kết quả như hình:



CHƯƠNG 4. TỔNG KẾT

Qua bài tập lớn này, có thể thấy rằng python là một ngôn ngữ lập trình dễ hiểu, dễ học cho những người mới bắt đầu, và với một cộng đồng lớn mạnh, có rất nhiều thư viện phục vụ cho nhiều mục đích khác nhau, python đã trở thành một ngôn ngữ phổ biến trong giới lập trình.

Node-red là một công cụ dễ học với các khối logic, thuận tiện cho những người mới bắt đầu và có thể với những học sinh nhỏ tuổi.

ASP.NET là một framework gần như là thuận tiện nhất khi liên kết 3 thành phần của web: cơ sở dữ liệu (SQL Server) – Backend (.aspx) – Frontend (html, css và javascript)

Tuy nhiên, bên cạnh việc sử dụng một ngôn ngữ, một nền tảng để lập trình, chúng ta cần phải kết hợp được nhiều những nền tảng, công cụ khác nhau để có thể tạo nên một bộ khung kiến thức chuyên môn.

Việc kết hợp các kiến thức về mạng máy tính, lập trình web, lập trình về các mảng khác giúp chúng ta mở rộng về tư duy hệ thống cũng như tăng kiến thức về các mảng khác.