

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

Bộ môn: Công nghệ thông tin



BÀI TẬP LỚN

MÔN HỌC

LẬP TRÌNH PYTHON

Sinh viên: Lê Việt Tiến

Mssv: K205480106026

Lớp: K56KMT

Giáo viên hướng dẫn: Đỗ Duy Cốp

Thái Nguyên – 2024

PHIẾU GIAO ĐỀ TÀI

Họ và tên: Lê Việt Tiến

Mssv: K205480106026

Lớp: K56KMT

Ngành: Kỹ thuật máy tính

Giáo viên hướng dẫn: Đỗ Duy Cốp

Ngày giao đề tài: 15/05/2024

Ngày hoàn thành: 25/05/2024

1. Tên đề tài: Xây dựng web theo dõi giá vàng.

2. Nội dung yêu cầu làm:

1. Cơ sở dữ liệu: mssql, hoặc mysql (ko dùng sqlite). có các bảng gì? ghi chú đầy đủ PK, FK. Có các SP_ nào.
2. Module đọc dữ liệu: sử dụng Python + FastAPI, lấy dữ liệu từ cảm biến, hoặc từ web, hoặc từ 1 nguồn nào đó, ko sinh random. Mô tả nguồn dữ liệu và thuật toán để lấy được nó. kết hợp python với FastAPI để tạo thành 1 API cho ứng dụng khác khai thác. đóng gói nó thành service với NSSM.
3. Nodered: Chu trình tự động gọi API python để lấy về dữ liệu, có thể phải xử lý thêm, gọi node tương tác với csdl để đẩy được dữ liệu vào database (nên gọi SP và truyền tham số, ko gọi trực tiếp lệnh INSERT).
4. web: thực hiện việc lấy dữ liệu từ db ra để làm gì đó, vd như vẽ biểu đồ, hoặc làm các thứ khác liên quan đến dữ liệu đã lấy được.

GIÁO VIÊN HƯỚNG DẪN
(ký và ghi rõ họ tên)

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the entire width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan đề tài “**Xây dựng web về sức khỏe**” là đề tài nghiên cứu cá nhân của em trong thời gian qua. Mọi số liệu sử dụng phân tích trong đề tài và kết quả nghiên cứu là do em tự tìm hiểu, phân tích một cách khách quan, trung thực, có nguồn gốc rõ ràng và chưa được công bố dưới bất kỳ hình thức nào.

Tên sinh viên

LỜI NÓI ĐẦU

CHƯƠNG I: CÁC NGÔN NGỮ ĐỂ LẬP TRÌNH

1.1. Cài đặt ngôn ngữ lập trình python gồm có các ngôn ngữ để lập trình python như Pycharm, Visual Studio, Visual code...

Ở đây em dùng Pycharm để sử dụng dự án của đề tài này:



Để sử dụng ngôn ngữ này, ta cần cài ứng dụng này như sau:

Bước 1: Lên web search Pycharm download về và cài đặt

Bước 2: Cài đặt thêm Python phiên bản mới nhất và path ứng dụng này vào trong hệ thống có đường dẫn như này ("D:\python\python.exe") ở đây em cài phiên bản 3.10.0.



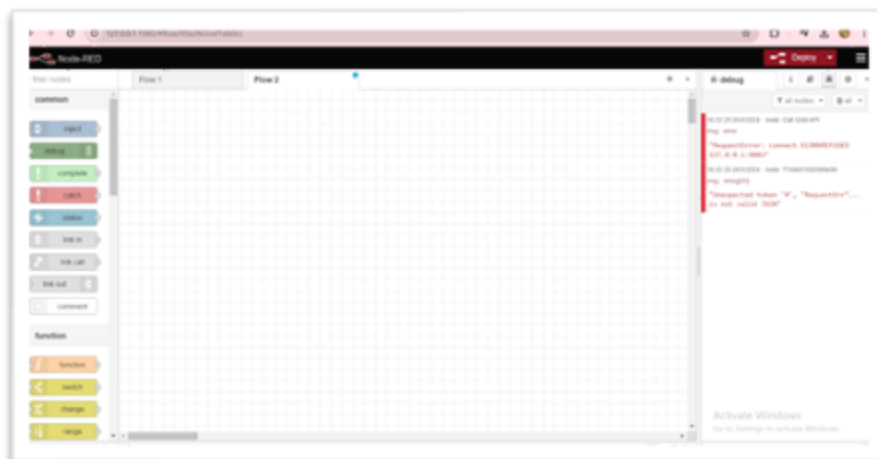
Bước 3: Vào Pycharm tạo New Project để sử dụng lập trình trong dự án.

1.2. Cài đặt node-red:

Bước 1: lên web search tìm node.js download về và path ứng dụng này vào trong hệ thống ở máy tính có đường dẫn sau: (“D:\node-red\node.exe”)



Bước 2: Mở Terminal hoặc Command Prompt. ấn dòng lệnh code: node-red, nó sẽ hiện ra đường dẫn <http://127.0.0.1:1880/> copy đường dẫn này lên web google hoặc chrome để sử dụng môi trường lập trình của node-red. Nếu thành công nó sẽ hiện giao diện web như này.



1.3. Cài đặt SQL Server:

Bước 1: Cần cài đặt ngôn ngữ này lên web search download SQL Server, cần thiết thì cài phiên bản mới nhất cũng được, tránh trường hợp lỗi.

Bước 2: Sau khi cài xong thì ta cần Tạo tài khoản user name pass word để kết nối cơ sở dữ liệu kết hợp với ngôn ngữ khác.



1.4 Hướng giải quyết bài toán

Để giải quyết bài toán và hiển thị thông tin, em 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. Thư viện `requests` trong Python sẽ hỗ trợ chúng ta trong việc này.

b. Ở Node-RED dùng node https 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:

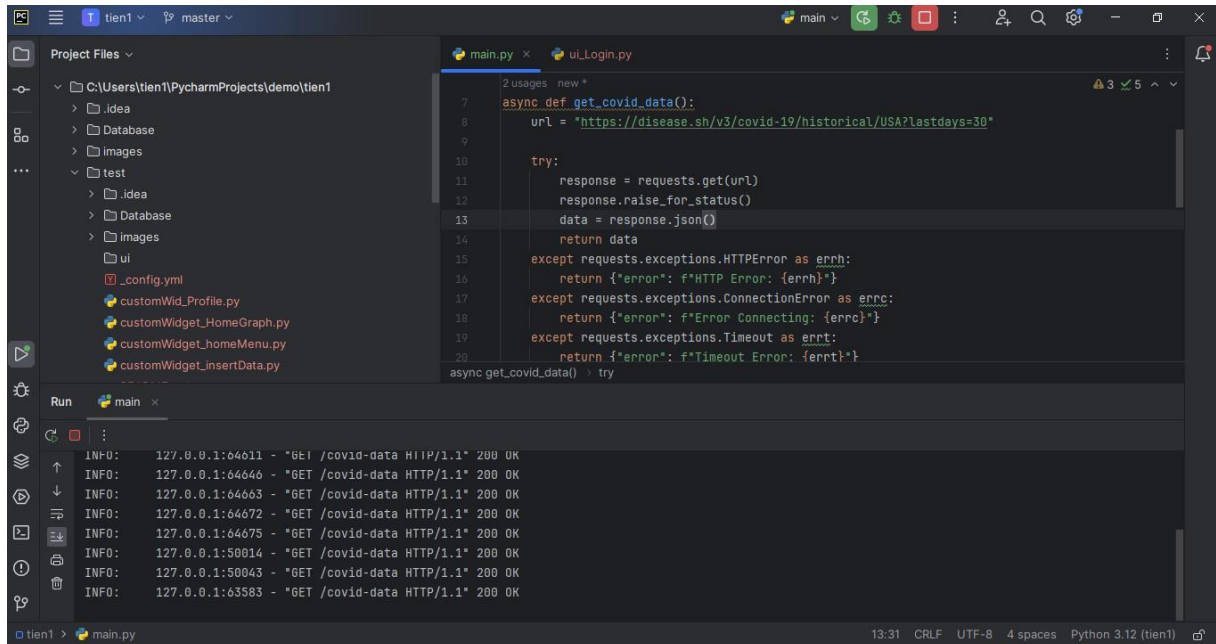
- Sử dụng html,css,js để lấy dữ liệu từ sqlserver thông qua asp.net (api.aspx)

Với hướng giải quyết này, chúng ta sẽ có một hệ thống toàn diện cho phép thu thập, xử lý và hiển thị dữ liệu số ca nhiễm COVID-19 một cách hiệu quả, đồng thời cung cấp thông tin đáng tin cậy và dễ hiểu cho người dùng.

CHƯƠNG II: NỘI DUNG THỰC HIỆN

1.1 Quá trình làm :

1.1.1 Tạo file python sử dụng fastapi lấy dữ liệu realtime từ web Install các thư viện cần thiết.



The screenshot shows the PyCharm IDE interface. On the left, the 'Project Files' pane displays the project structure for 'demo\tien1', including folders like '.idea', 'Database', 'images', 'test', and files like '_config.yml' and several custom widget Python files. The main editor window shows the code for 'main.py' and 'ui_Login.py'. The 'main.py' file contains an asynchronous function 'get_covid_data()' that uses the 'requests' library to fetch data from 'https://disease.sh/v3/covid-19/historical/USA?lastdays=30'. The function includes error handling for HTTP errors, connection errors, and timeouts. Below the code editor, the 'Run' pane shows the output of the application, displaying a series of log messages indicating successful GET requests to the '/covid-data' endpoint.

```
7 async def get_covid_data():
8     url = "https://disease.sh/v3/covid-19/historical/USA?lastdays=30"
9
10    try:
11        response = requests.get(url)
12        response.raise_for_status()
13        data = response.json()
14        return data
15    except requests.exceptions.HTTPError as errh:
16        return {"error": f"HTTP Error: {errh}"}
17    except requests.exceptions.ConnectionError as errc:
18        return {"error": f"Error Connecting: {errc}"}
19    except requests.exceptions.Timeout as errt:
20        return {"error": f"Timeout Error: {errt}"}
21
22    async get_covid_data() -> try
```

```
INFO: 127.0.0.1:64611 - "GET /covid-data HTTP/1.1" 200 OK
INFO: 127.0.0.1:64646 - "GET /covid-data HTTP/1.1" 200 OK
INFO: 127.0.0.1:64663 - "GET /covid-data HTTP/1.1" 200 OK
INFO: 127.0.0.1:64672 - "GET /covid-data HTTP/1.1" 200 OK
INFO: 127.0.0.1:64675 - "GET /covid-data HTTP/1.1" 200 OK
INFO: 127.0.0.1:50014 - "GET /covid-data HTTP/1.1" 200 OK
INFO: 127.0.0.1:50043 - "GET /covid-data HTTP/1.1" 200 OK
INFO: 127.0.0.1:63583 - "GET /covid-data HTTP/1.1" 200 OK
```

Sau khi chạy ta được api dạng chuỗi json



The screenshot shows a web browser window displaying the JSON response from the '127.0.0.1:8001/covid-data' API. The JSON data is formatted with syntax highlighting and line numbers. It contains a list of daily new cases, a 'deaths' object with a list of daily deaths, and a 'recovered' object with a list of daily recoveries.

```
23  {
24    "2/24/23": 103378408,
25    "2/25/23": 1033811157,
26    "2/26/23": 103382763,
27    "2/27/23": 103398627,
28    "2/28/23": 103443455,
29    "3/1/23": 103533872,
30    "3/2/23": 103589757,
31    "3/3/23": 103648690,
32    "3/4/23": 103650837,
33    "3/5/23": 103646975,
34    "3/6/23": 103655539,
35    "3/7/23": 103690910,
36    "3/8/23": 103755771,
37    "3/9/23": 103802702,
38  },
39  "deaths": {
40    "2/8/23": 1113429,
41    "2/9/23": 1114385,
42    "2/10/23": 1114492,
43    "2/11/23": 1114529,
44    "2/12/23": 1114537,
45    "2/13/23": 1114709,
46    "2/14/23": 1115160,
47    "2/15/23": 1115741,
48    "2/16/23": 1116851,
49    "2/17/23": 1117572,
50    "2/18/23": 1117589,
51    "2/19/23": 1117590,
52    "2/20/23": 1117663,
53    "2/21/23": 1118025,
54    "2/22/23": 1118886,
55    "2/23/23": 1119521,
56    "2/24/23": 1119573,
57    "2/25/23": 1119587,
58    "2/26/23": 1119588,
59    "2/27/23": 1119582,
60    "2/28/23": 1119917,
61    "3/1/23": 1120897,
62    "3/2/23": 1121658,
63    "3/3/23": 1122165,
64    "3/4/23": 1122172,
65    "3/5/23": 1122134,
66    "3/6/23": 1122181,
67    "3/7/23": 1122516,
68    "3/8/23": 1123246,
69    "3/9/23": 1123836,
70  },
71  "recovered": {
72    "2/8/23": 0,
```

1.1.2 Mở CMD chạy node-red

```
node-red
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\tien1>node-red
26 May 17:01:01 - [info]

Welcome to Node-RED
=====

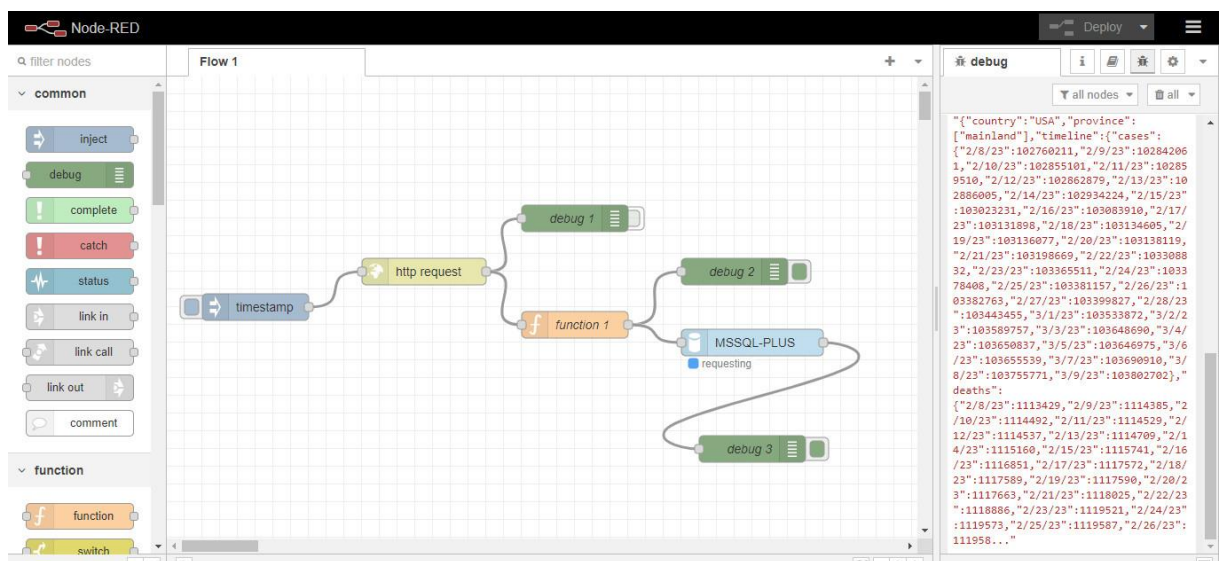
26 May 17:01:01 - [info] Node-RED version: v3.1.9
26 May 17:01:01 - [info] Node.js version: v20.13.1
26 May 17:01:01 - [info] Windows_NT 10.0.19045 x64 LE
26 May 17:01:03 - [info] Loading palette nodes
26 May 17:01:10 - [info] Settings file : C:\Users\tien1\.node-red\settings.js
26 May 17:01:10 - [info] Context store : 'default' [module=memory]
26 May 17:01:10 - [info] User directory : \Users\tien1\.node-red
26 May 17:01:10 - [warn] Projects disabled : editorTheme.projects.enabled=false
26 May 17:01:10 - [info] Flows file : \Users\tien1\.node-red\flows.json
26 May 17:01:10 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
```

Tạo cấu trúc của node-red để lưu dữ liệu về SQL:



Muốn lưu về mình cần phải điền địa chỉ api chạy fast api

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🔍

☰ Method

GET

▼

🌐 URL

http://127.0.0.1:8001/covid-data

Payload

Ignore

▼

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

☐ Only send non-2xx responses to Catch node

☐ Disable strict HTTP parsing

⬅️ Return

a UTF-8 string

▼

☰ Headers

📄


○ Enabled


Cài đặt node-red-contrib-mssql-plus ---> cài đặt cấu hình cho node


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


Delete Cancel **Update**


Properties


 Name Connection Name


 Server DESKTOP-LJMHIB7\SQLEXPRESS


 Port 1433


 Username sa

 Password

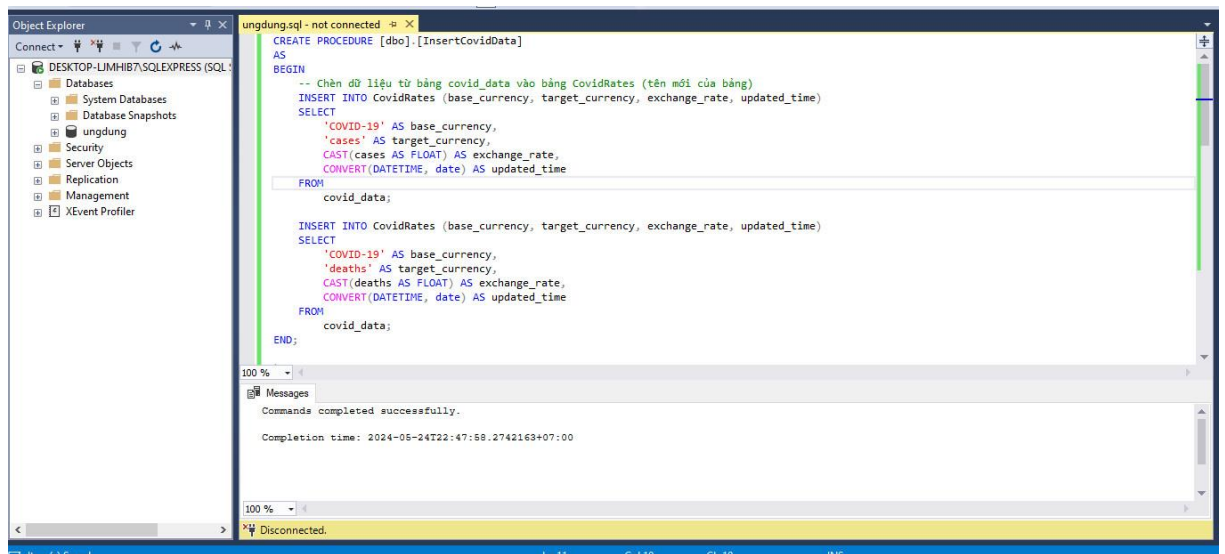
 Domain

 Database ungdung

 TDS Version 7_4 (SQL Server 2012 ~ 2022) ▼

 Use Encryption? ☒

Sau đó em viết SP trả dữ liệu dạng json rồi đẩy lên web.



1.1.3 Làm giao diện web

Liên kết SQL sever với VS studio lấy dữ liệu và đưa lên web bằng APS.NET



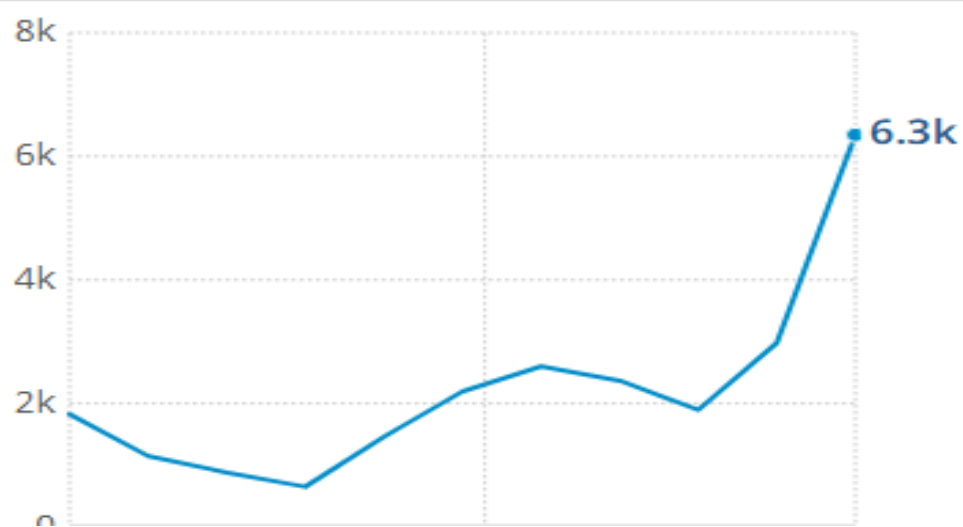
Code js và html

```
C:\> Users > Admin > Desktop > api2.aspx.cs
1  using System;
2  using System.Net.Http;
3  using System.Threading.Tasks;
4  using Newtonsoft.Json.Linq;
5
6  namespace YourNamespace
7  {
8      public class CovidDataService
9      {
10         private static readonly HttpClient client = new HttpClient();
11
12         public async Task<JObject> GetCovidData()
13         {
14             try
15             {
16                 // URL endpoint của Node-RED
17                 string url = "http://localhost:1880/getcoviddata";
18
19                 // Gọi HTTP GET để lấy JSON từ Node-RED
20                 HttpResponseMessage response = await client.GetAsync(url);
21                 response.EnsureSuccessStatusCode();
22                 string responseBody = await response.Content.ReadAsStringAsync();
23
24                 // Parse JSON response
25                 JObject json = JObject.Parse(responseBody);
26
27                 return json;
28             }
29             catch (HttpRequestException e)
30             {
31                 Console.WriteLine("\nException Caught!");
32                 Console.WriteLine("Message :{0} ", e.Message);
33                 return null;
34             }
35         }
36     }
37 }
```

```
1  @{{
2      ViewData["Title"] = "Home Page";
3  }}
4
5  <!DOCTYPE html>
6  <html>
7  <head>
8      <title>COVID-19 Data</title>
9      <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
10     <script src="~/js/covidChart.js"></script>
11 </head>
12 <body>
13     <div style="width: 75%;">
14         <canvas id="covidChart"></canvas>
15     </div>
16 </body>
17 </html>
```


Kết quả :

Các trường hợp nhiễm COVID-19 được báo cáo



Chương III: KẾT LUẬN

Trong bài tập lớn này, em đã biết thực hiện việc sử dụng FastAPI, một framework Python hiệu suất cao, để tạo một ứng dụng web API. Và em đã có thể kết nối với một nguồn dữ liệu API và cung cấp dữ liệu đó cho các client qua các endpoint.

- Xây dựng một ứng dụng FastAPI đơn giản với các endpoint.
- Sử dụng thư viện requests để tương tác với API từ một nguồn dữ liệu bên ngoài.
- Xử lý dữ liệu trả về từ API và trả về cho client qua các endpoint FastAPI.

Hiểu thêm về nodejs cách cài đặt nodejs , chạy node-red , cấu hình cho nodered để nhận api từ fastapi rồi lại chuyển về trong sql sever , biết cách tạo các bảng Sp để chuyển và chứa dữ liệu.

Em cảm ơn thầy Đỗ Duy Cốp đã giúp em hiểu và hoàn thành bài tập lớn này.