

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ



**BÀI TẬP LỚN PHÁT TRIỂN
ỨNG DỤNG TRÊN NỀN WEB**

BỘ MÔN : CÔNG NGHỆ THÔNG TIN
**MÔN HỌC: PHÁT TRIỂN ỨNG DỤNG TRÊN
NỀN WEB**

GIÁO VIÊN HƯỚNG DẪN : THS. ĐỖ DUY CỐP
HỌ VÀ TÊN SINH VIÊN : ĐẬU VĂN KHÁNH
LỚP : K58KTP.K01
MSSV : K225480106099

THÁI NGUYÊN - 2025

TRƯỜNG ĐHKTTCN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

PHIẾU GIAO BÀI TẬP LỚN PHÁT TRIỂN ỨNG DỤNG TRÊN NỀN WEB

Sinh viên: Đậu Văn Khánh

MSSV: K225480106099

Lớp: K58KTP.K01

Khoá: 2022-2027

Ngành học: Kỹ thuật phần mềm

Giáo viên hướng dẫn: ThS. Đỗ Duy Cốp

1. Tên đề tài: Bài tập phát triển ứng dụng trên nền web

2. Nội dung thực hiện:

- Bài 1: Xây dựng hệ thống DLL đa nền tảng và tích hợp vào Console App, Windows Form và Web ASP.NET sử dụng IIS (.NET Framework 2.0)

- Bài 2: Xây dựng hệ thống Web Full-Stack sử dụng Apache, Node-RED và SQL Server

- Bài 3: Lập trình ứng dụng web trên nền Linux

3. Ngày giao nhiệm vụ: 15/10/2025

4.. Ngày hoàn thành nhiệm vụ: 25/11/2025

TRƯỜNG KHOA

BỘ MÔN

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

(Ký và ghi rõ họ tên)

(Ký và ghi rõ họ tên)

PHIẾU GHI ĐIỂM

**HƯỚNG DẪN BÀI TẬP LỚN PHÁT TRIỂN ỨNG DỤNG
TRÊN NỀN WEB**

Sinh viên: Đậu Văn Khánh

MSSV: K225480106099

Lớp: K58KTP

GVHD: ThS. Đỗ Duy Cốp

Môn học: Phát triển ứng dụng trên nền web

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....
.....
.....
.....
.....
.....
.....
.....
.....

Xếp loại:

Điểm:

Thái Nguyên, ngày....tháng.....năm 2025

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn sâu sắc đến thầy Đỗ Duy Cốp, giảng viên Trường Đại học Kỹ thuật Công nghiệp Thái Nguyên, người đã tận tình hướng dẫn, chỉ bảo và đồng hành cùng em trong suốt quá trình thực hiện và hoàn thiện báo cáo bài tập lớn này. Nhờ sự giúp đỡ quý báu của thầy, em đã có thể vượt qua những khó khăn, từng bước hoàn thành tốt nội dung đề tài được giao.

Mặc dù đã cố gắng hết sức, nhưng trong quá trình thực hiện báo cáo, chắc chắn sẽ không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được sự thông cảm và những góp ý quý báu từ các thầy, cô để em có thể rút kinh nghiệm, hoàn thiện bản thân hơn trong những chặng đường học tập và công tác sau này.

Em xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển nhanh chóng, các ứng dụng web giữ vai trò trung tâm trong việc xây dựng những hệ thống thông tin hiện đại, linh hoạt và dễ dàng mở rộng. Từ các nền tảng thương mại điện tử, hệ thống quản lý doanh nghiệp cho đến các ứng dụng IoT, web không chỉ là công cụ truyền tải thông tin mà còn là môi trường thực thi nhiều nghiệp vụ phức tạp. Chính vì thế, việc nắm vững quy trình phát triển ứng dụng web toàn diện – từ backend, frontend đến triển khai hạ tầng – là yêu cầu quan trọng đối với mỗi sinh viên công nghệ thông tin.

Bài tập lớn môn Phát triển ứng dụng trên nền web được xây dựng với mục tiêu giúp sinh viên tiếp cận đầy đủ các thành phần của một hệ thống web hoàn chỉnh. Thông qua ba bài thực hành: xây dựng DLL đa nền tảng và tích hợp vào nhiều môi trường .NET Framework; phát triển hệ thống web full-stack dựa trên Apache, Node-RED và SQL Server; và triển khai ứng dụng web trên môi trường Linux kết hợp Docker, sinh viên được rèn luyện kỹ năng làm việc với đa dạng công nghệ, công cụ và mô hình kiến trúc.

Quá trình thực hiện bài tập giúp sinh viên nắm được cách xây dựng API backend, thiết kế giao diện frontend động, xử lý dữ liệu qua hệ quản trị cơ sở dữ liệu, hiểu rõ các hoạt động của web server, reverse proxy và container hóa ứng dụng. Đồng thời, sinh viên có cơ hội tiếp cận tư duy phát triển hệ thống theo mô hình thực tế, từ cài đặt môi trường, cấu hình dịch vụ cho đến tổ chức và vận hành sản phẩm web.

MỤC LỤC

LỜI CẢM ƠN	4
LỜI NÓI ĐẦU	5
BÀI TẬP 1: XÂY DỰNG HỆ THỐNG DLL ĐA NỀN TẢNG VÀ TÍCH HỢP VÀO CONSOLE APP, WINDOWS FORM VÀ WEB ASP.NET SỬ DỤNG IIS (.NET FRAMEWORK 2.0)	6
1.1. Tạo DLL đa năng, keyword: c# window library -> Class Library (.NET Framework)	8
1.2. Tạo Console app, sử dụng .NET Framework 2.0 và sử dụng được DLL	10
1.3. Tạo Windows Form Application(.NET Framework 2.0)	12
1.4. Tạo Web đơn giản, sử dụng .NET Framework 2.0, sử dụng web server là IIS	14
1.5. Tạo cấu hình IIS	17
BÀI TẬP 2: XÂY DỰNG HỆ THỐNG WEB FULL-STACK SỬ DỤNG APACHE, NODE-RED VÀ SQL SERVER	21
2.1. Cài đặt Apache web server:	24
2.2. Cài đặt nodejs và nodered	29
2.3. Tạo cơ sở dữ liệu tuỳ ý trên mssql (sql server 2022)	32
2.4. Cài đặt thư viện trên nodered	34
2.5. Tạo api back-end bằng nodered	38
2.6. Tạo giao diện front-end	39
2.7. Nhận xét bài làm	42
BÀI TẬP 3: LẬP TRÌNH ỦNG DỤNG WEB TRÊN NỀN LINUX	43
3.1. Cài đặt môi trường linux	45
3.2. Cài đặt Docker	48
3.3. Sử dụng 1 file docker-compose.yml	49
3.4. Lập trình web frontend+backend	52
3.5. Nginx làm web-server	68
3.6. Kết luận:	72
KẾT LUẬN	74
TÀI LIỆU THAM KHẢO	75
MÃ QR GITHUB	76

BÀI TẬP 1: XÂY DỰNG HỆ THỐNG DLL ĐA NỀN TẢNG VÀ TÍCH HỢP VÀO CONSOLE APP, WINDOWS FORM VÀ WEB ASP.NET SỬ DỤNG IIS (.NET FRAMEWORK 2.0)

ĐỀ BÀI: TẠO SOLUTION GỒM CÁC PROJECT SAU:

DLL đa năng, keyword: c# window library -> Class Library (.NET Framework) bắt buộc sử dụng .NET Framework 2.0: giải bài toán bất kỳ, độc lạ càng tốt, phải có dấu án cá nhân trong kết quả, biên dịch ra DLL. DLL độc lập vì nó ko nhập, ko xuất, nó nhận input truyền vào thuộc tính của nó, và trả về dữ liệu thông qua thuộc tính khác, hoặc thông qua giá trị trả về của hàm. Nó độc lập thì sẽ sử dụng được trên app dạng console (giao diện dòng lệnh - đen sì), cũng sử dụng được trên app desktop (dạng cửa sổ), và cũng sử dụng được trên web form (web chạy qua iis).

Console app, bắt buộc sử dụng .NET Framework 2.0, sử dụng được DLL trên: nhập được input, gọi DLL, hiển thị kết quả, phải có dấu án cá nhân. keyword: c# window Console => Console App (.NET Framework), biên dịch ra EXE.

Windows Form Application, bắt buộc sử dụng .NET Framework 2.0**, sử dụng được DLL đa năng trên, kéo các control vào để có thể lấy dc input, gọi DLL truyền input để lấy dc kq, hiển thị kq ra window form, phải có dấu án cá nhân; keyword: c# window Desktop => Windows Form Application (.NET Framework), biên dịch ra EXE.

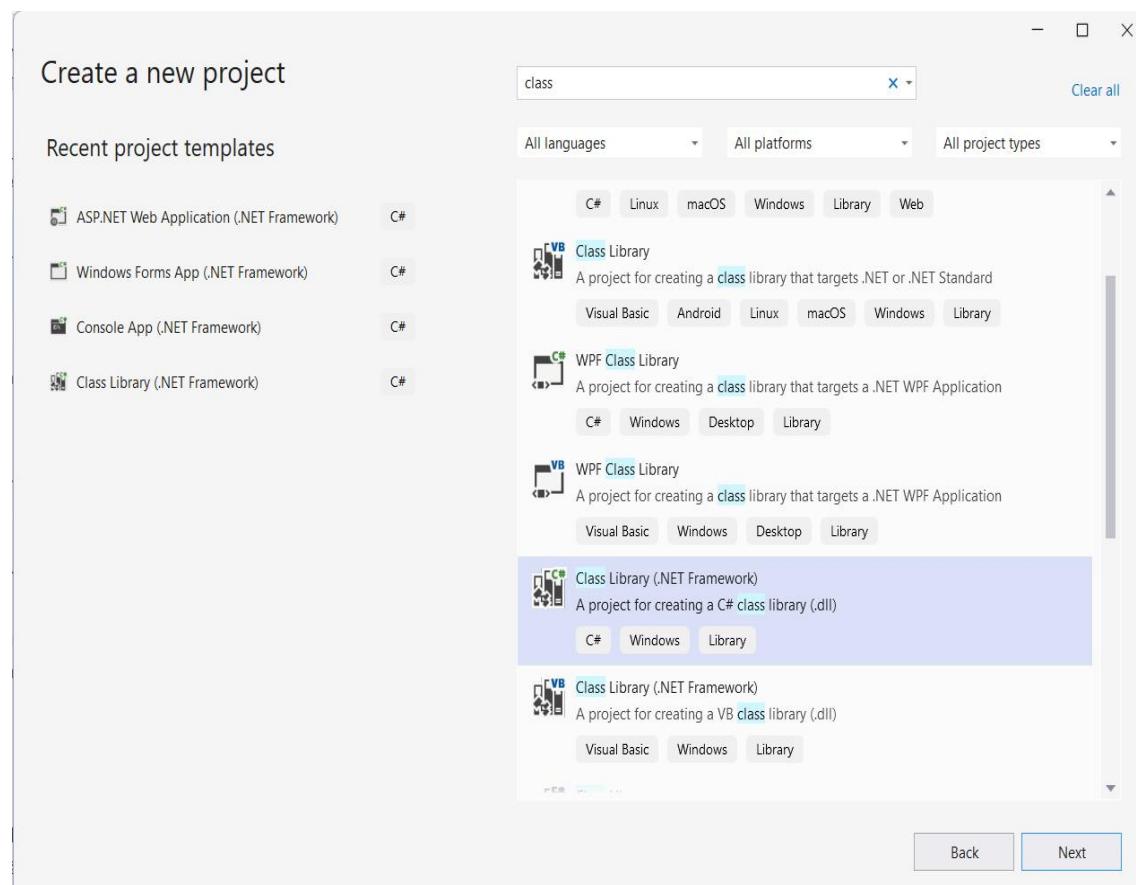
Web đơn giản, bắt buộc sử dụng .NET Framework 2.0, sử dụng web server là IIS, dùng file hosts để tự tạo domain, gắn domain này vào iis, file index.html có sử dụng html css js để xây dựng giao diện nhập được các input cho bài toán, dùng mã js để tiền xử lý dữ liệu, js để gửi lên backend. backend là api.aspx, trong code của api.aspx.cs thì lấy được các input mà js gửi lên, rồi sử dụng được DLL đa năng trên. kết quả gửi lại json cho client, js phía client sẽ nhận được json này hậu xử lý để thay đổi giao diện theo dữ liệu nhận được, phải có dấu án cá nhân. keyword: c# window web => ASP.NET Web Application (.NET Framework) + tham khảo link chatgpt thầy gửi. project web này biên dịch ra DLL, phải kết hợp với IIS mới chạy được.

BÀI LÀM

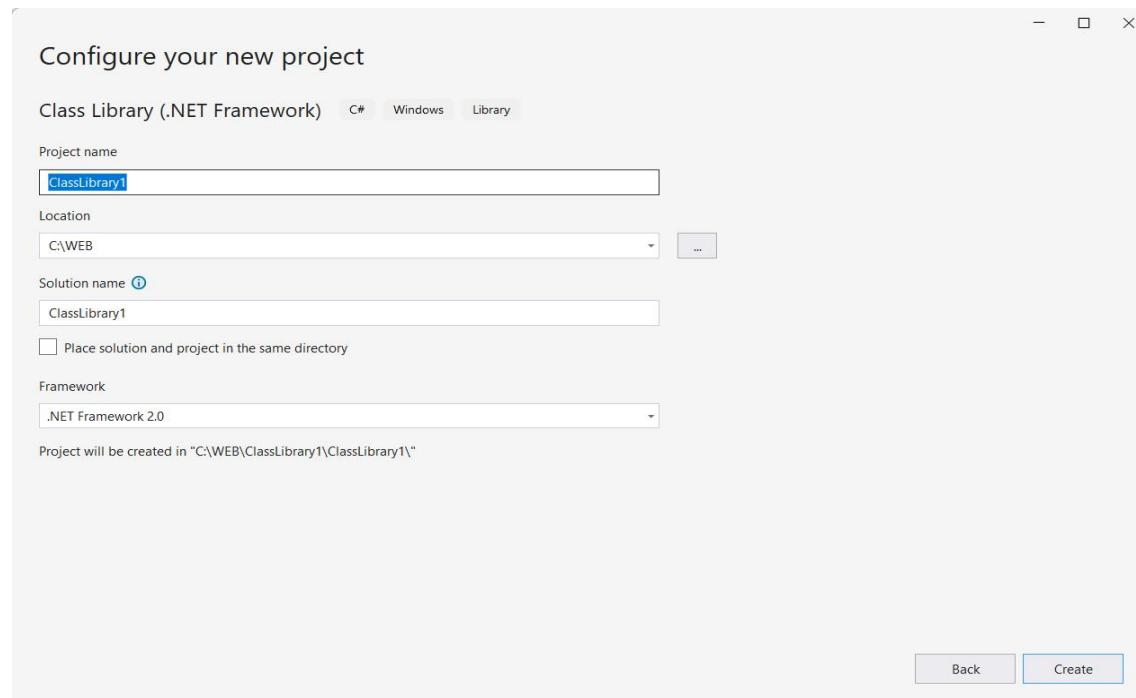
Đề tài: Xây dựng trò chơi Rắn săn mồi (Snake Game) - Trò chơi cho phép người chơi điều khiển rắn di chuyển trong bản đồ để ăn mồi, mỗi lần ăn mồi rắn sẽ dài thêm và cần phải tránh để rắn tự cắn vào thân hoặc va vào tường.

1.1. Tạo DLL đa năng, keyword: c# window library -> Class Library (.NET Framework).

- Mục tiêu: Tạo một DLL để xử lý toàn bộ logic của game Snake: quản lý rắn, thức ăn, di chuyển, tính điểm...
- Trong Visual Studio 2022, tìm và chọn Class Library -> nhấn Next.

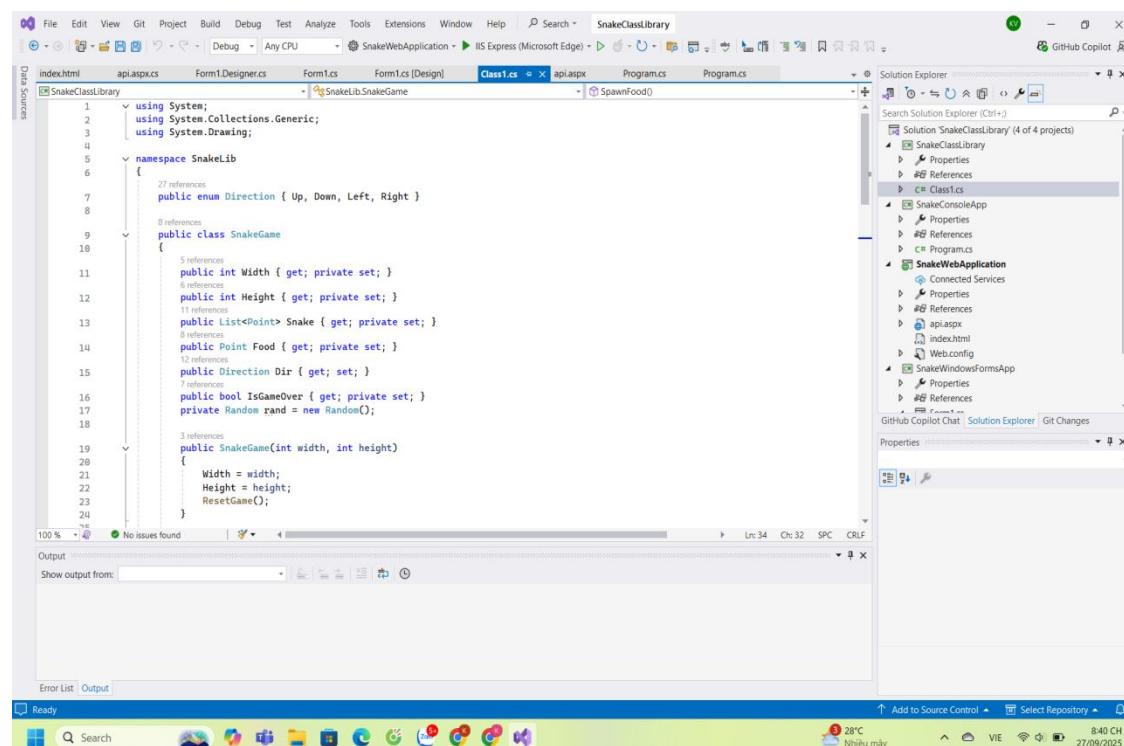


- Sau khi nhấn Next, nó sẽ hiện ra cửa sổ sau:



- Trong cửa sổ cần đặt tên Project và Location (nơi lưu project) và chọn .NET Framework 2.0 -> nhấn Create.

Sau đó Visual Studio sẽ tạo sẵn một file Class1.cs (có thể đổi tên nếu muốn), rồi tiến hành viết code.



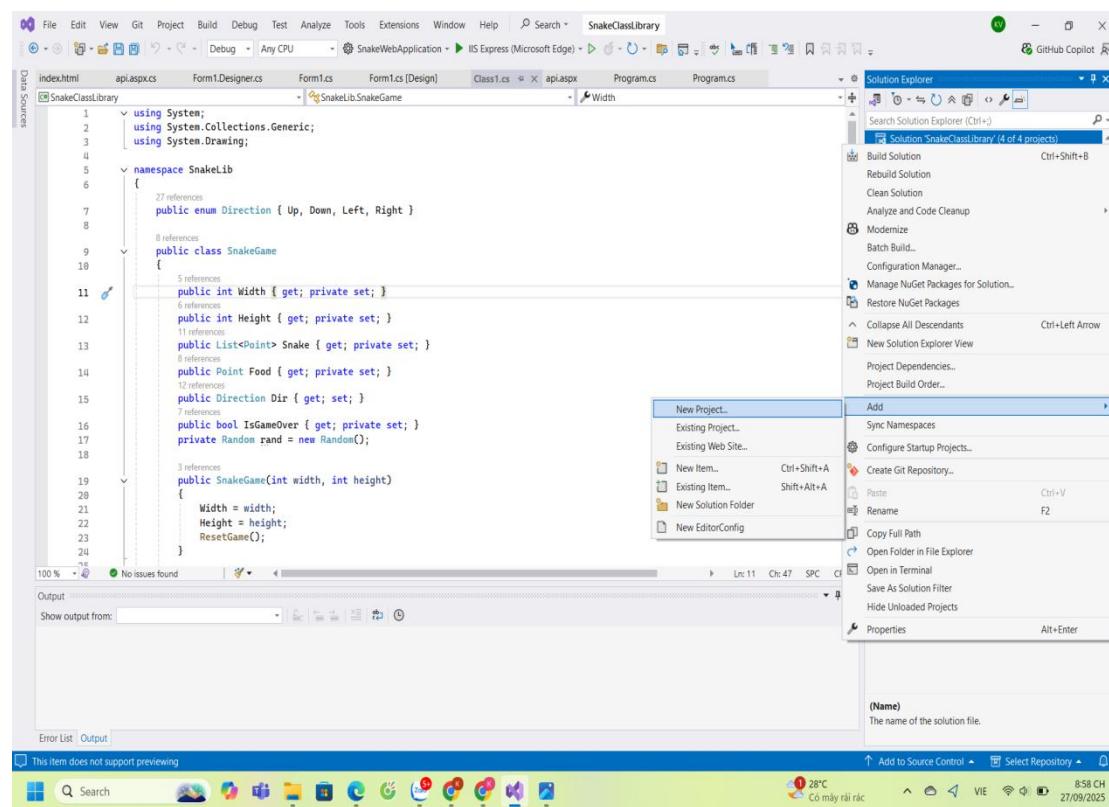
- Chú ý: Tất cả các class và hàm muôn cho project khác sử dụng phải để public.

- Sau khi viết code xong, ta nhấn Build trên thanh công cụ -> Build Solution (hoặc Ctrl + Shift + B).
- Sau khi Build xong, sẽ có file SnakeClassLibrary.dll

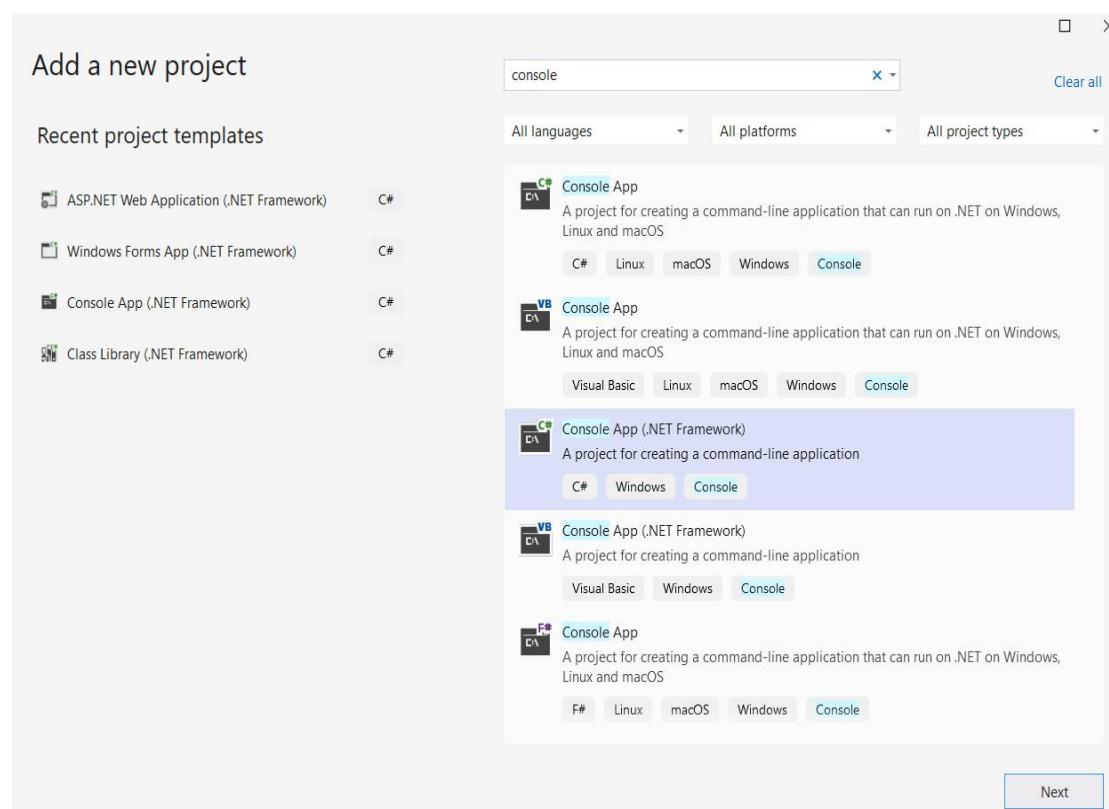


1.2. Tạo Console app, sử dụng .NET Framework 2.0 và sử dụng được DLL trên:

- Mục tiêu: Dùng DLL để chơi Snake trên màn hình dòng lệnh.
- Trong Solution -> chuột phải chọn Add -> chọn New project.



- Sau đó tìm và chọn Console App (.NET Framework) -> nhấn Next.



- Sau đó tiến hành đặt tên project và chọn .NET Framework 2.0 -> nhấn Create.
- Khi tạo xong thì sẽ xuất hiện file Program.cs -> tiến hành viết code trong file Program.cs.

```

using System;
using System.Threading;
using SnakeLib;

namespace ConsoleApp
{
    class Program
    {
        static void Main()
        {
            SnakeGame game = new SnakeGame(20, 10);
            ConsoleKey key = ConsoleKey.RightArrow;

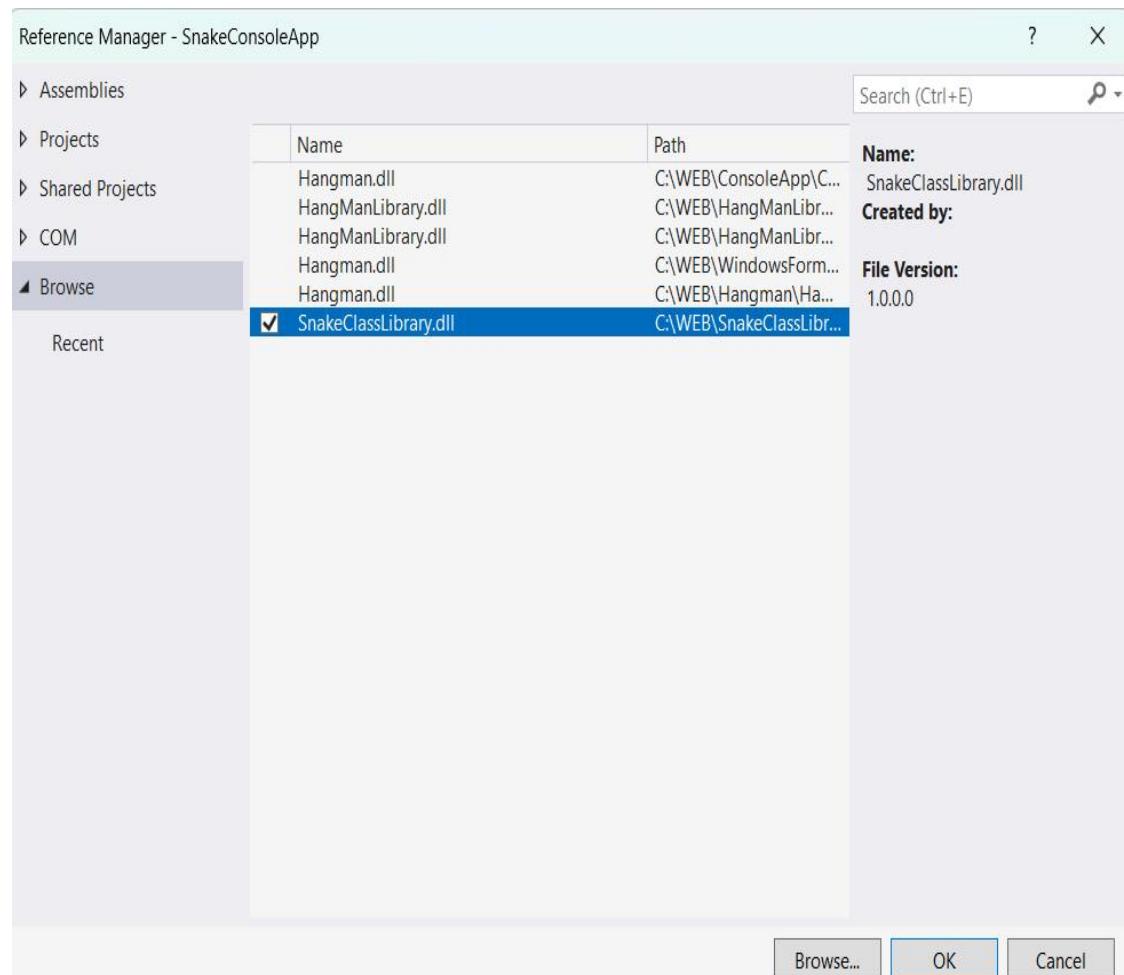
            while (!game.IsGameOver)
            {
                if (Console.KeyAvailable)
                {
                    key = Console.ReadKey(true).Key;
                    switch (key)
                    {
                        case ConsoleKey.UpArrow: game.Dir = Direction.Up; break;
                        case ConsoleKey.DownArrow: game.Dir = Direction.Down; break;
                        case ConsoleKey.LeftArrow: game.Dir = Direction.Left; break;
                        case ConsoleKey.RightArrow: game.Dir = Direction.Right; break;
                    }
                }

                game.Update();
                Draw(game);
                Thread.Sleep(200);
            }
        }

        static void Draw(SnakeGame game)
        {
            // Drawing logic here
        }
    }
}

```

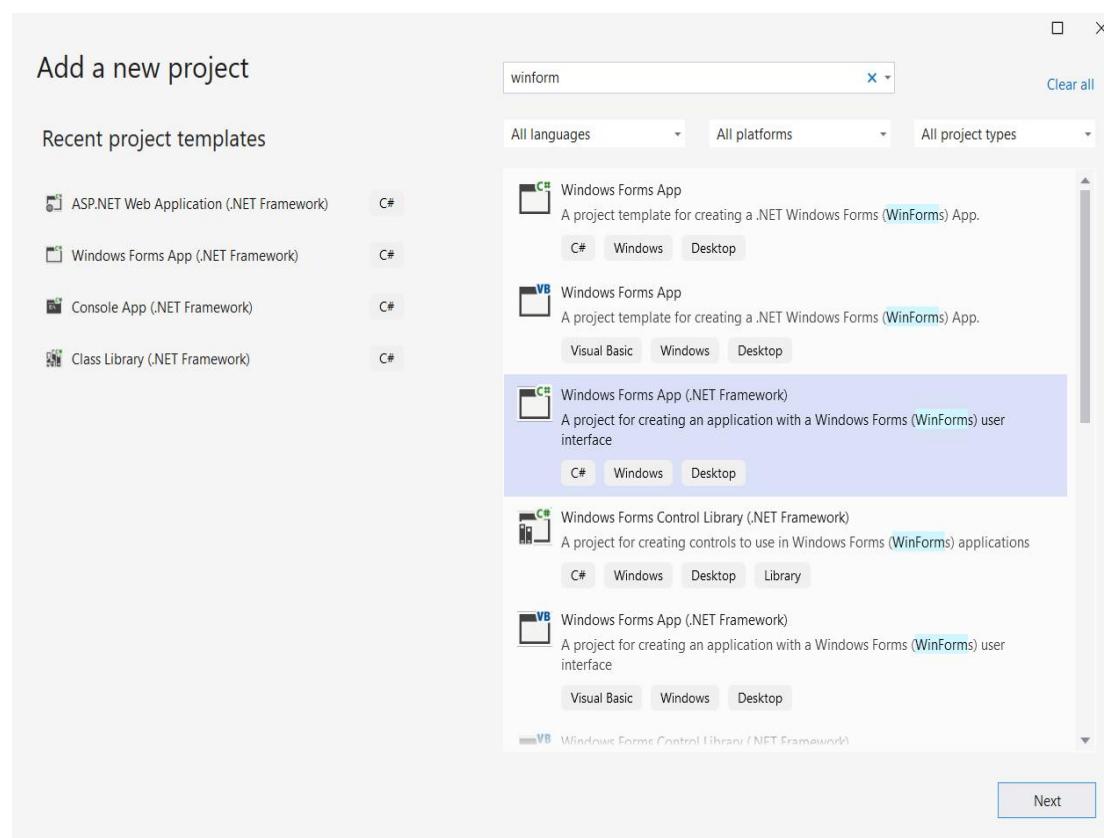
- Khi viết code xong cần tham chiếu đến DLL: nhán chuột phải References trong Console App -> Add Reference... -> Browse -> chọn file SnakeClassLibrary.dll -> nhán OK.



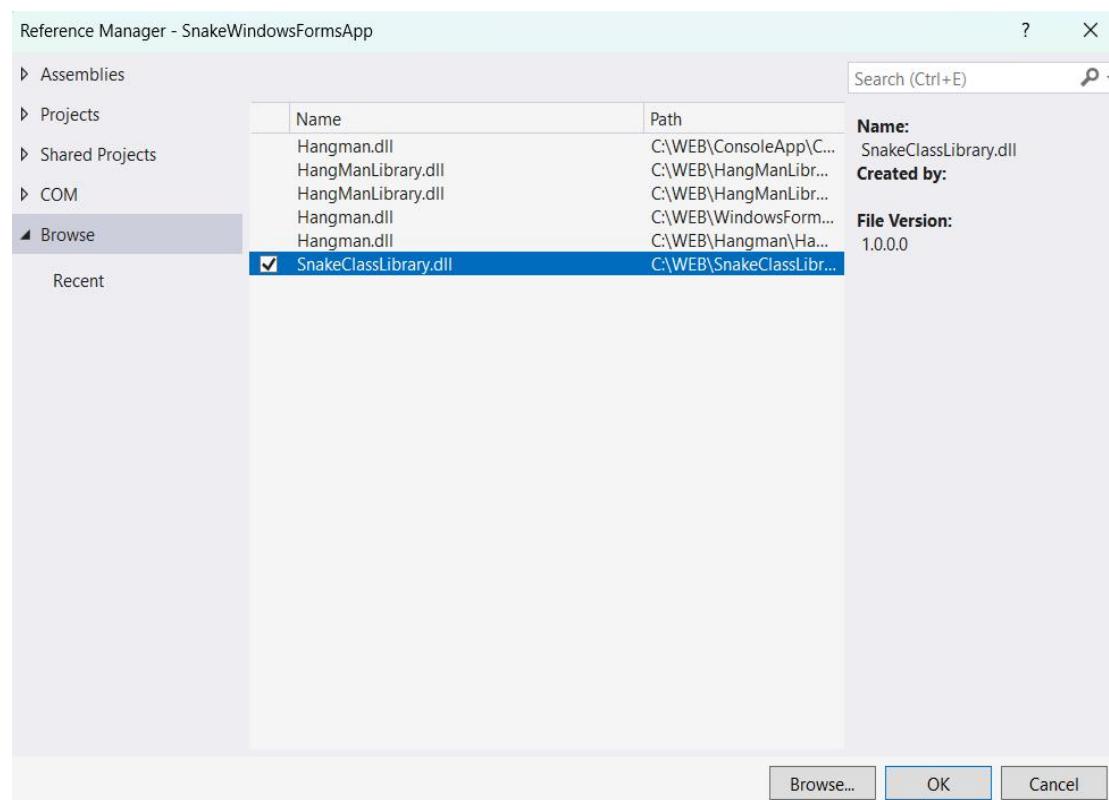
- Sau khi tham chiếu đến DLL xong, thì tiến hành Build -> Build Solution (Ctrl+Shift+B). Chạy bằng F5 (debug) hoặc Ctrl+F5 (run without debug).

1.3. Tạo Windows Form Application(.NET Framework 2.0)

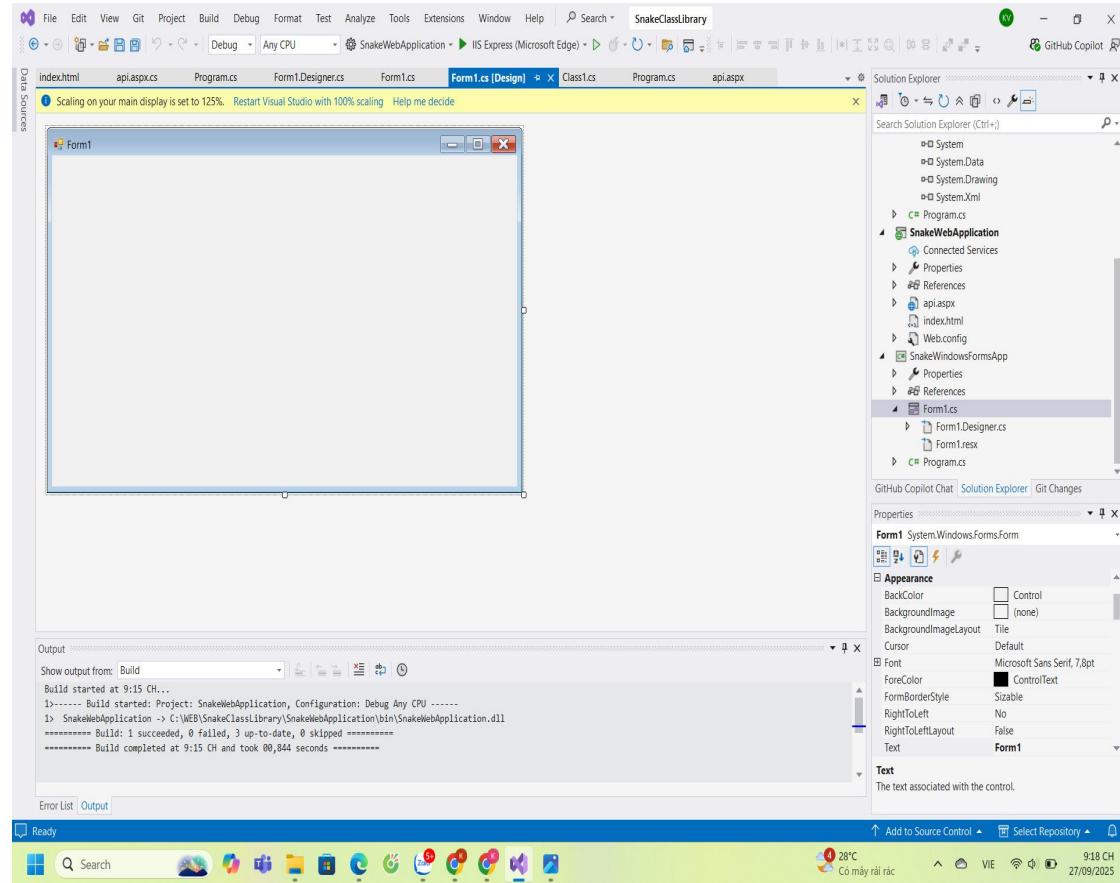
- Mục tiêu: Xây dựng giao diện cửa sổ để chơi Snake.
- Trong Solution -> chuột phải chọn Add -> chọn New project.
- Tạo project: cần tìm và chọn Windows Forms App (.NET Framework) -> Next.



- Sau đó tiến hành đặt tên project và chọn .NET Framework 2.0 -> nhấn Create.
- Sau khi tạo xong, chuột phải References -> Add Reference...-> Browse -> chọn file SnakeClassLibrary.dll -> nhấn OK.



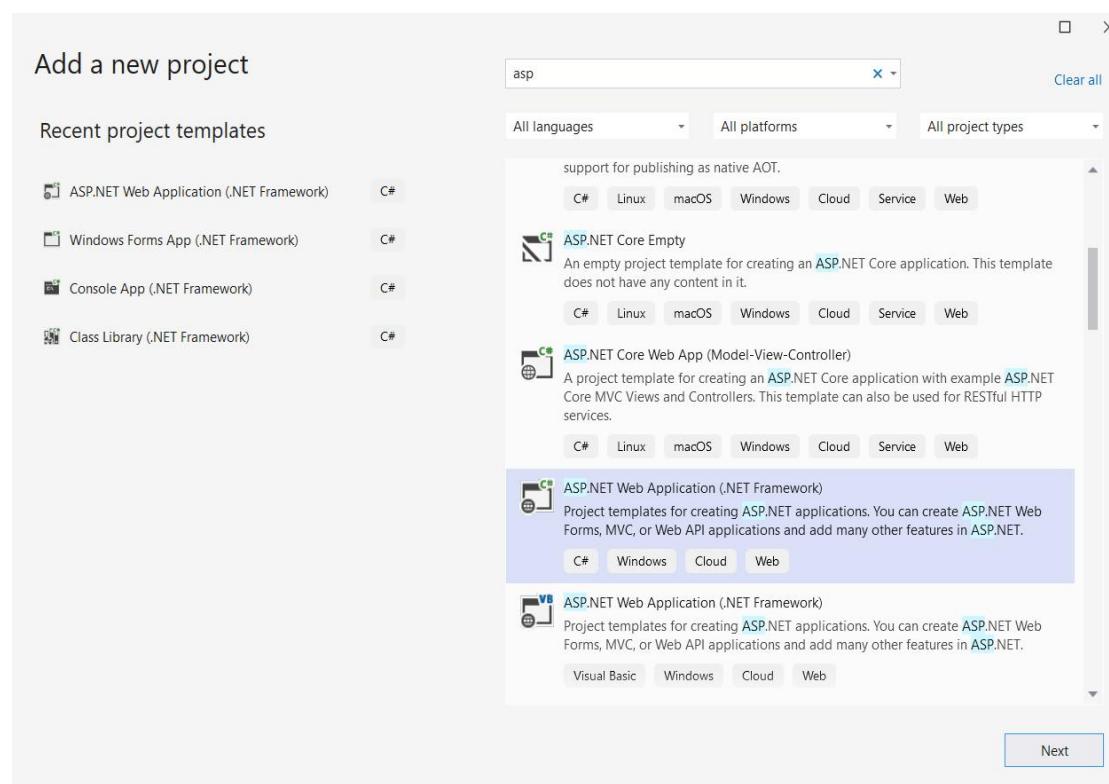
- Để thiết kế giao diện game, mở Form1.cs trong Design: trên thanh công cụ chọn View -> Toolbox rồi kéo Panel (vùng vẽ bàn chơi), Button (Start/Restart), Label (Hiển thị điểm).



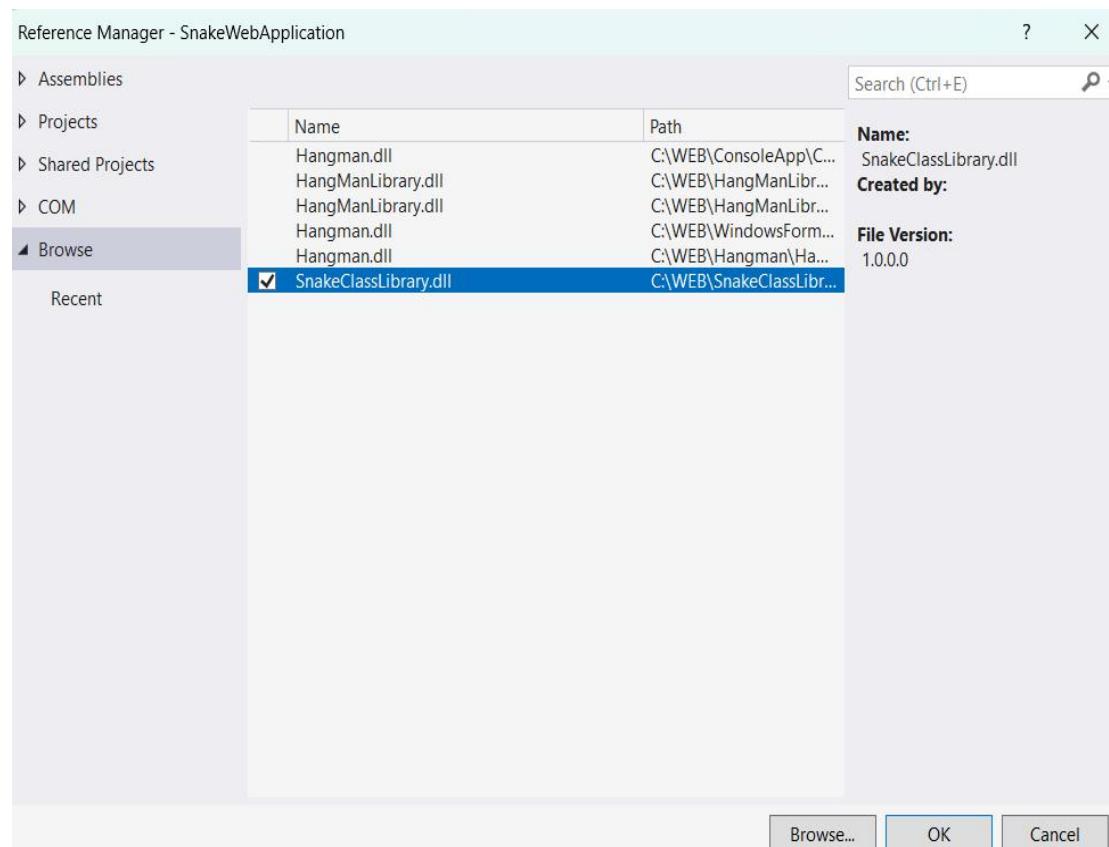
- Nhấn Build → chạy thử.

1.4. Tạo Web đơn giản, sử dụng .NET Framework 2.0, sử dụng web server là IIS.

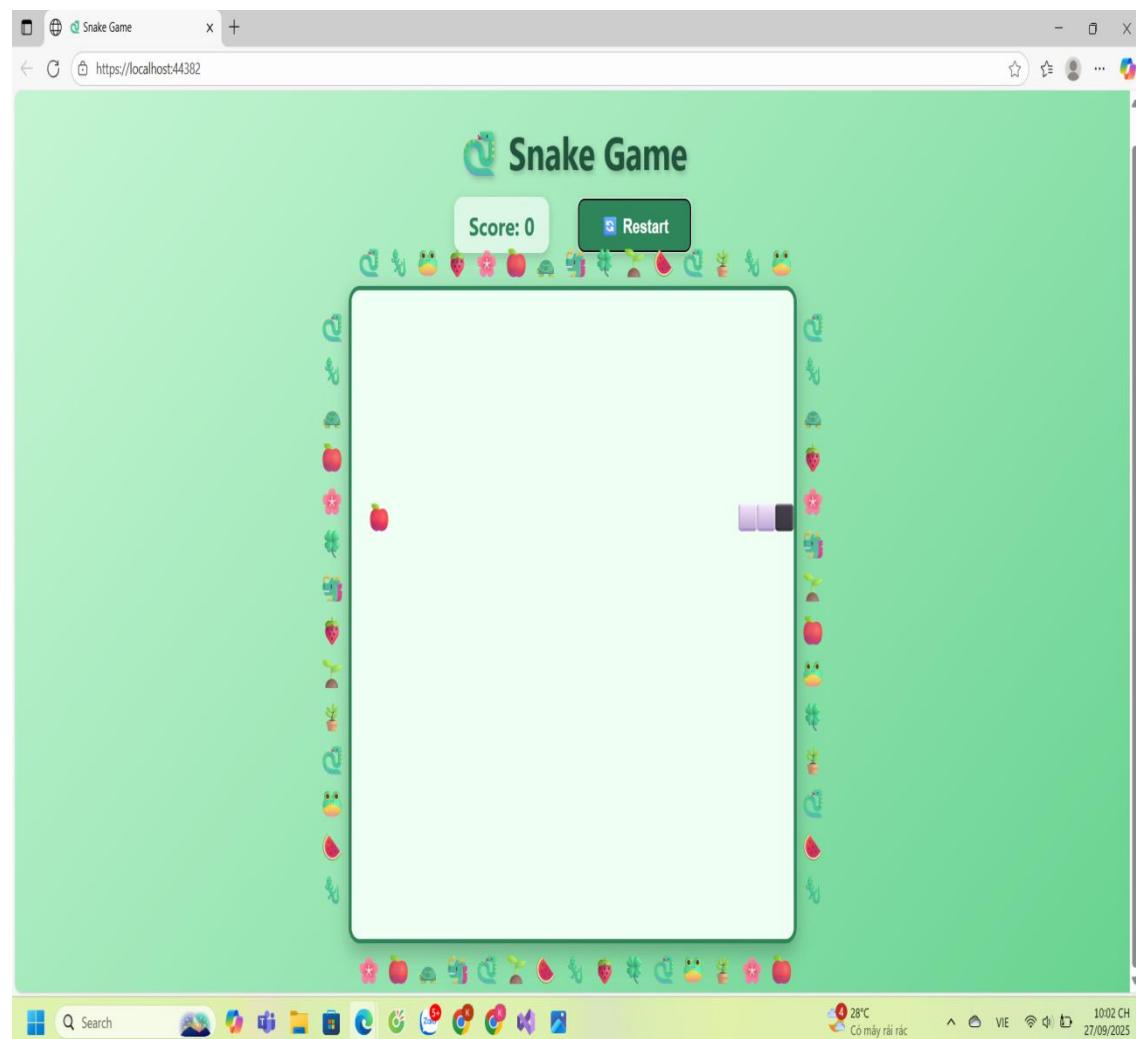
- Mục tiêu: Tạo giao diện web chơi Snake, gọi DLL qua backend API.
- Trong Solution -> chuột phải chọn Add -> chọn New project.
- Sau đó cần tìm và chọn ASP.NET Web Application (.NET Framework) -> Next.



- Tiến hành đặt tên project và chọn .NET Framework 2.0 -> nhấn Create.
- Chuột phải References trong project -> Add Reference...-> Browse -> chọn file SnakeClassLibrary.dll -> nhấn OK.



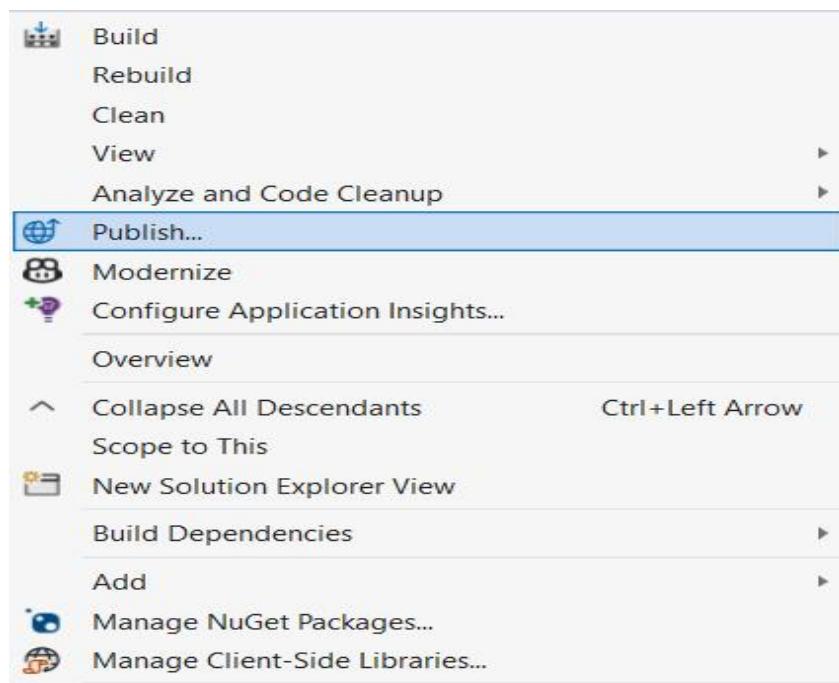
- Thêm file: Chuột phải vào SnakeWebApplication -> Chọn Add -> Chọn New item -> chọn html page.
- index.html → giao diện nhập dữ liệu (bàn phím/di chuyển, nút Start, bảng điểm).
- script.js → điều khiển hiển thị bàn chơi bằng HTML + CSS + Canvas.
- Tạo WebForm api.aspx: Trong SnakeWebApplication -> chuột phải vào SnakeWebApplication -> Chọn Add -> Chọn New item -> chọn Web Form -> Đặt tên file: api.aspx.
- Trong file api.aspx: Bỏ hết code mặc định, chỉ giữ dòng đầu tiên: <%@ Page Language="C#" AutoEventWireup="true" CodeFile="api.aspx.cs" Inherits="api" %> và file api.aspx.cs sẽ tự động được tạo ra.
- File api.aspx.cs: đọc dữ liệu từ client (Request.Form).
- Build và chạy thử kiểm tra kết quả:



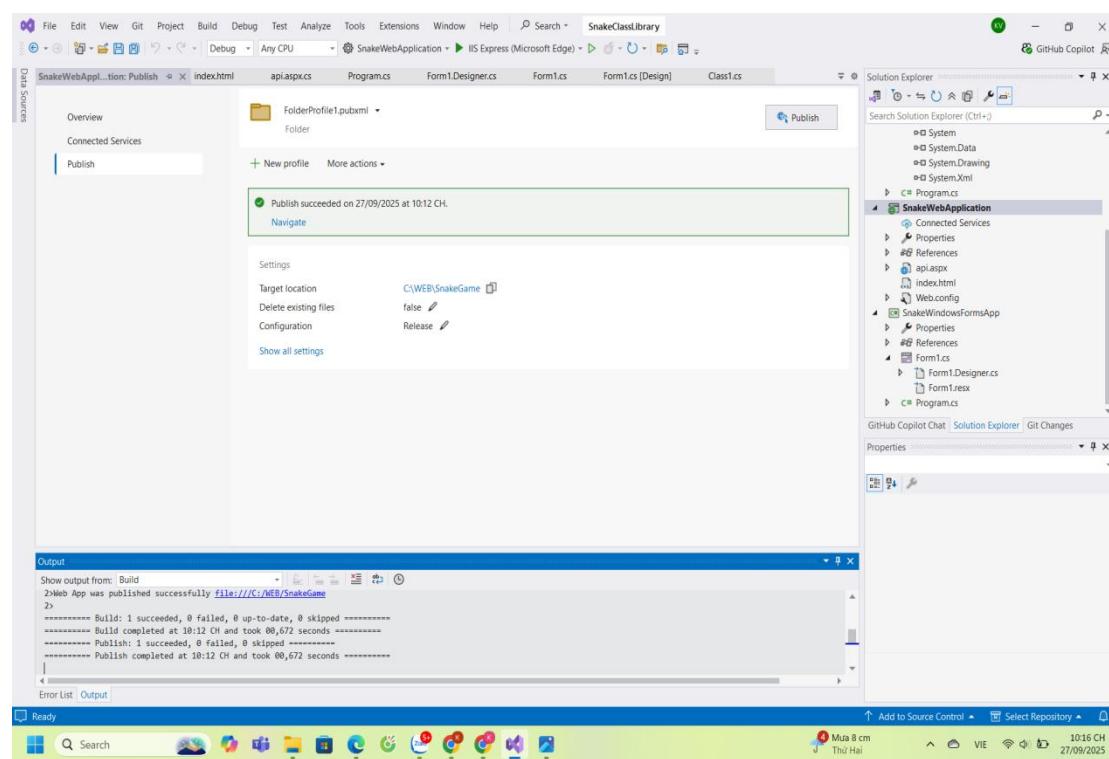
1.5. Tạo cấu hình IIS

1.5.1. Cần tạo 1 folder mới để Publish đến

- Trong Visual studio -> chuột phải SnakeWebApp -> chọn Publish

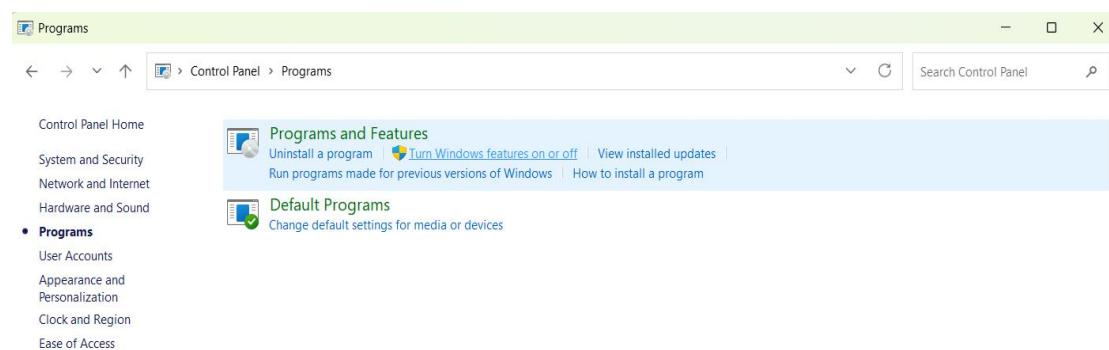


- Sau khi chọn Publish -> chọn Folder -> nhấn Next -> Chọn browse và chọn đến folder mới vừa tạo -> nhấn Finish
- Kết quả sau khi Publish thành công

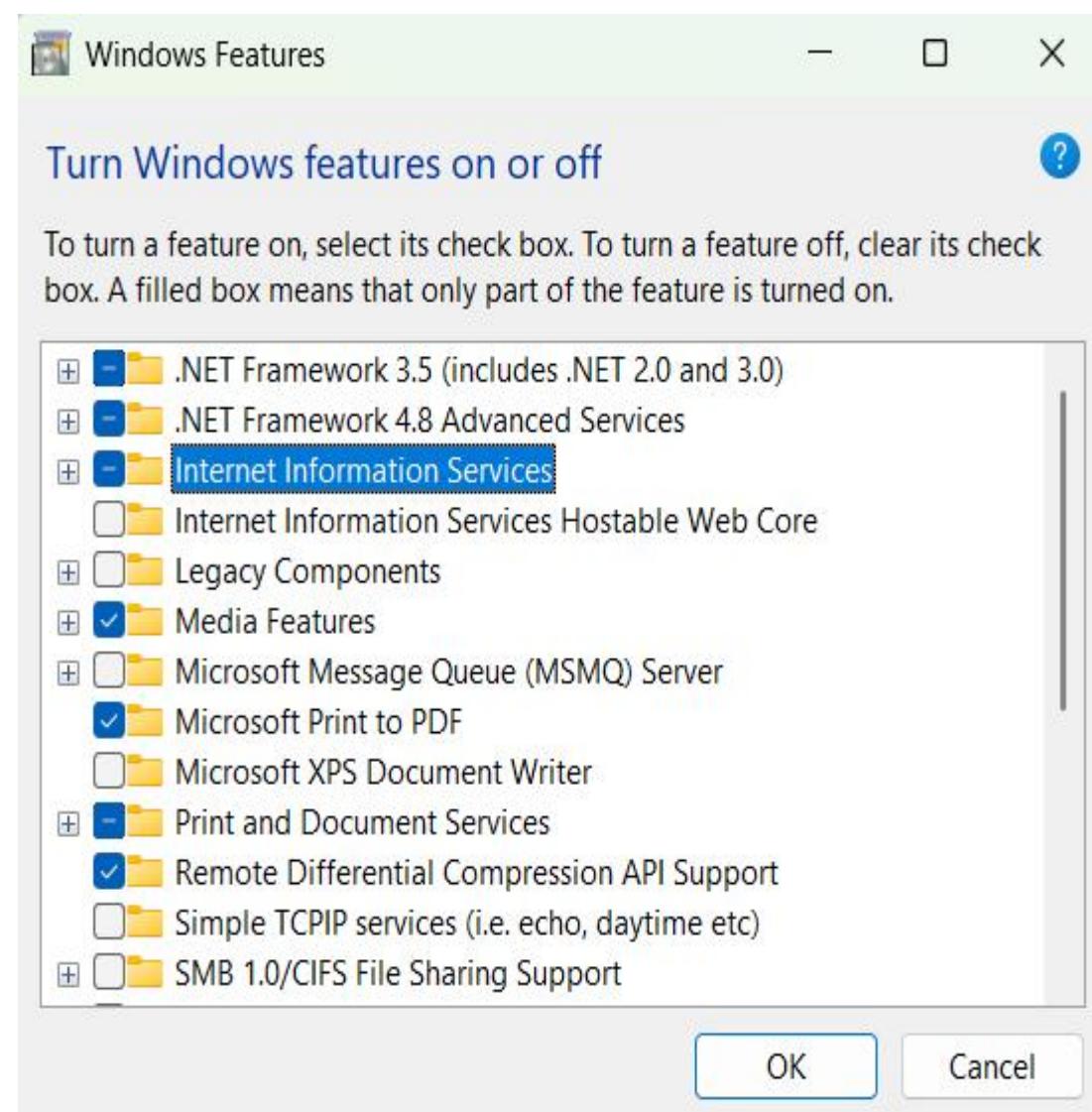


1.5.2. Cài IIS

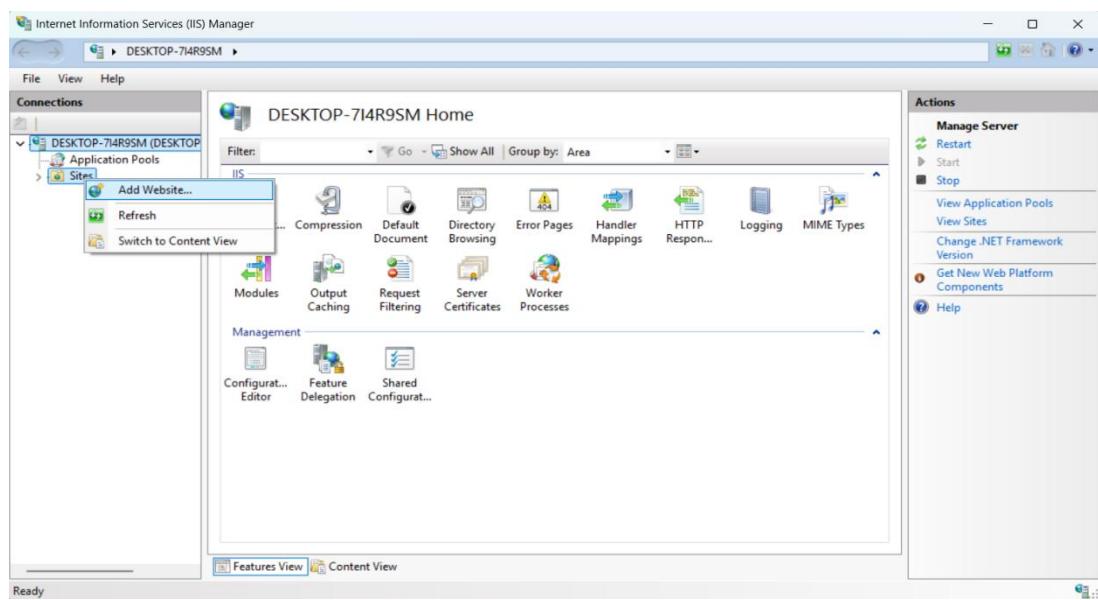
- Mở Control Panel -> Chọn Program -> Chọn Turn Windows features on or off.



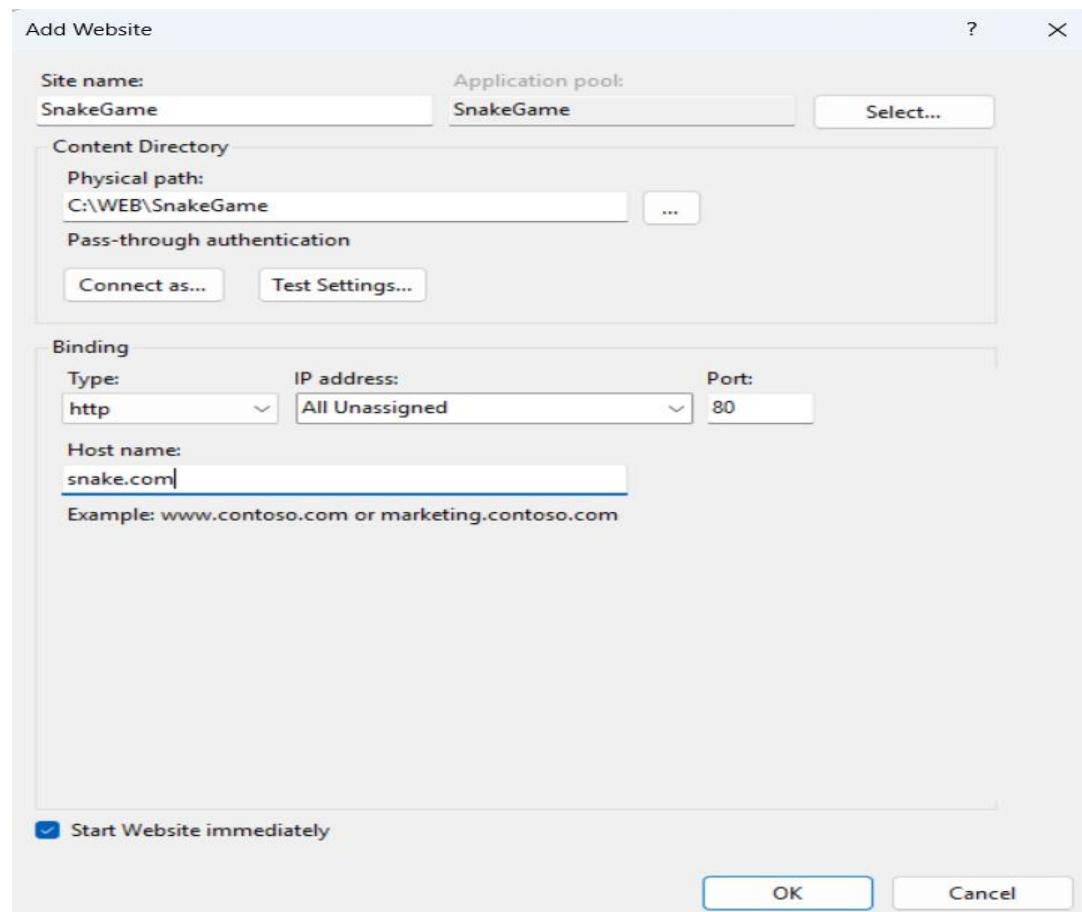
- Nhấn chọn Internet Information Services -> nhấn OK



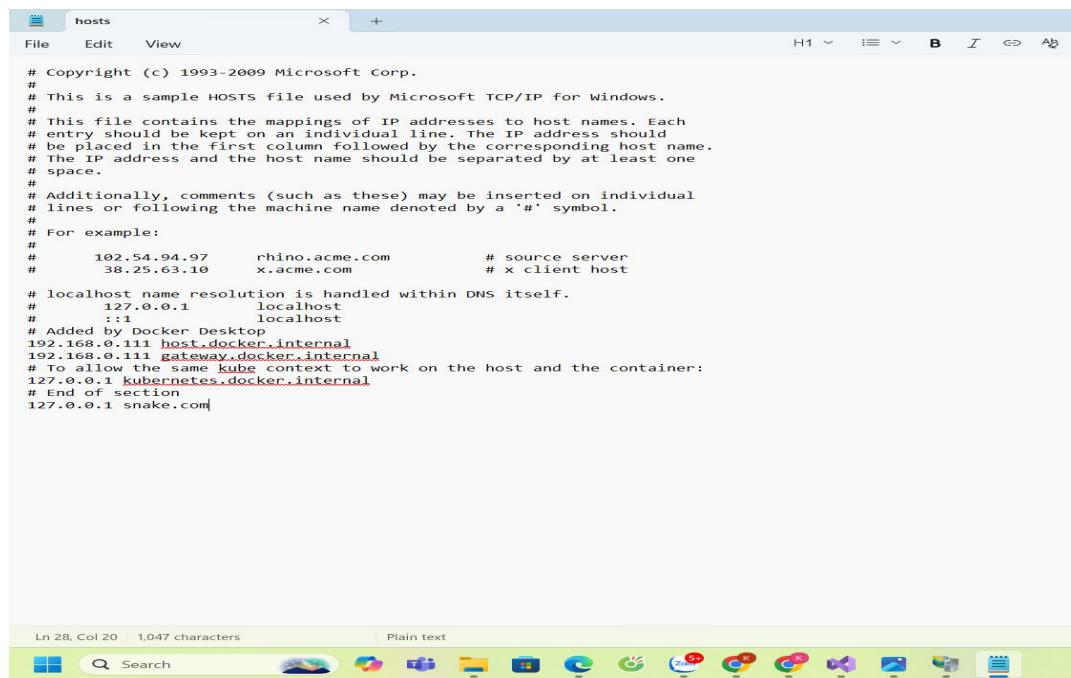
- Mở IIS Manager -> chuột phải vào Sites -> chọn Add Website.



- Trong cửa sổ Add Website cần điền thông tin: site name, physical path host name.

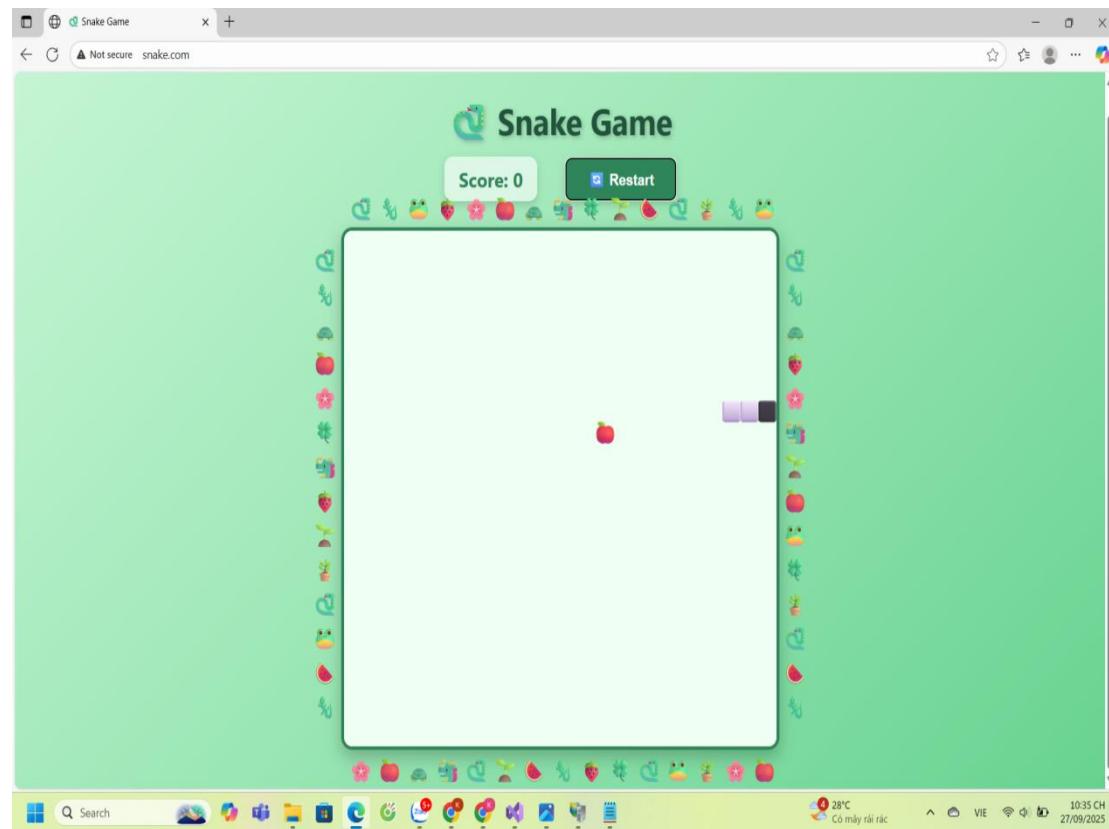


- Cuối cùng chuột phải vào Notepad -> chọn Run as Administrator -> hiển thị file hosts -> thêm 127.0.0.1 snake.com ở cuối file -> nhấn Ctrl+S để lưu file.



```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97    rhino.acme.com        # source server
#      38.25.63.10    x.acme.com            # x client host
#
# localhost name resolution is handled within DNS itself.
#
#       127.0.0.1          localhost
#       ::1               localhost
# Added by Docker Desktop
192.168.0.111 host.docker.internal
192.168.0.111 gateway.docker.internal
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
127.0.0.1 snake.com
```

- Kết quả:



BÀI TẬP 2: XÂY DỰNG HỆ THỐNG WEB FULL-STACK SỬ DỤNG APACHE, NODE-RED VÀ SQL SERVER

2.1. Cài đặt Apache web server:

- Vô hiệu hoá IIS: nếu iis đang chạy thì mở cmd quyền admin để chạy lệnh: iisreset /stop.
- Download apache server, giải nén ra ổ D, cấu hình các file:
 - + D:\Apache24\conf\httpd.conf
 - + D:\Apache24\conf\extra\httpd-vhosts.conf
- Để tạo website với domain: fullname.com
Code web sẽ đặt tại thư mục: `D:\Apache24\fullname` (fullname ko dấu, liền nhau).
- Sử dụng file `c:\WINDOWS\SYSTEM32\Drivers\etc\hosts` để fake ip 127.0.0.1 cho domain này.

Ví dụ sv tên là: 'Đỗ Duy Cốp' thì tạo website với domain là fullname ko dấu, liền nhau: `doduycop.com`.

- Thao tác dòng lệnh trên file 'D:\Apache24\bin\httpd.exe' với các tham số `-k install` và `-k start` để cài đặt và khởi động web server apache.

2.2. Cài đặt nodejs và nodered => Dùng làm backend:

- Cài đặt nodejs:
 - + Download file `https://nodejs.org/dist/v20.19.5/node-v20.19.5-x64.msi` (đây ko phải bản mới nhất, nhưng ổn định)
 - + Cài đặt vào thư mục 'D:\nodejs'
- Cài đặt nodered:
 - + Chạy cmd, vào thư mục 'D:\nodejs', chạy lệnh `npm install -g - unsafe-perm node-red --prefix "D:\nodejs\nodered"`
 - + Download file: <https://nssm.cc/release/nssm-2.24.zip>
giải nén được file nssm.exe
copy nssm.exe vào thư mục 'D:\nodejs\nodered\'
 - + Tạo file "D:\nodejs\nodered\run-nodered.cmd" với nội dung (5 dòng sau):

```
@echo off
```

```
REM fix path  
set PATH=D:\nodejs;%PATH%  
  
REM Run Node-RED  
Node      "D:\nodejs\nodered\node_modules\node-red\red.js"      -u  
"D:\nodejs\nodered\work" %*  
+ Mở cmd, chuyển đến thư mục: 'D:\nodejs\nodered'  
+ Cài đặt service 'a1-nodered' bằng lệnh: nssm.exe install a1-nodered  
"D:\nodejs\nodered\run-nodered.cmd"  
+ Chạy service 'a1-nodered' bằng lệnh: 'nssm start a1-nodered'
```

2.3. Tạo csdl tuỳ ý trên mssql (sql server 2022), nhớ các thông số kết nối: ip, port, username, password, db_name, table_name

2.4. Cài đặt thư viện trên nodered:

- Truy cập giao diện nodered bằng url: http://localhost:1880
- Cài đặt các thư viện: node-red-contrib-mssql-plus, node-red-node-mysql, node-red-contrib-telegrambot, node-red-contrib-moment, node-red-contrib-influxdb, node-red-contrib-duckdns, node-red-contrib-cron-plus.
- Sửa file 'D:\nodejs\nodered\work\settings.js' :

Tìm đến chõ adminAuth, bỏ comment # ở đầu dòng (8 dòng), thay chuỗi mã hoá mật khẩu bằng chuỗi mới

```
adminAuth: {  
    type: "credentials",  
    users: [ {  
        username: "admin",  
        password: "chuỗi_mã_hoa_mật_khẩu",  
        permissions: "*"  
    }]  
},
```

Với mã hoá mật khẩu có thể thiết lập bằng tool:
<https://tms.tnut.edu.vn/pw.php>

- Chạy lại nodered bằng cách: mở cmd, vào thư mục 'D:\nodejs\nodered' và chạy lệnh `nssm restart a1-nodered`.

Khi đó nodered sẽ yêu cầu nhập mật khẩu mới vào được giao diện cho admin tại: http://localhost:1880

2.5. Tạo api back-end bằng nodered:

- Tại flow1 trên nodered, sử dụng node 'http in' và 'http response' để tạo api.
- Thêm node 'MSSQL' để truy vấn tới cơ sở dữ liệu.
- Logic flow sẽ gồm 4 node theo thứ tự sau (thứ tự nối dây):
 1. http in : dùng GET cho đơn giản, URL đặt tùy ý, ví dụ: /timkiem
 2. function : để tiền xử lý dữ liệu gửi đến
 3. MSSQL: để truy vấn dữ liệu tới CSDL, nhận tham số từ node tiền xử lý
 4. http response: để phản hồi dữ liệu về client: Status Code=200, Header add : Content-Type = application/json

Có thể thêm node 'debug' để quan sát giá trị trung gian.

- Test api thông qua trình duyệt, ví dụ: http://localhost:1880/timkiem?q=thị

2.6. Tạo giao diện front-end:

- html form gồm các file : index.html, fullname.js, fullname.css
cả 3 file này đặt trong thư mục: 'D:\Apache24\fullname'
nhớ thay fullname là tên của bạn, viết liền, ko dấu, chữ thường, vd tên là Đỗ Duy Cór thì fullname là `doduycop`
khi đó 3 file sẽ là: index.html, doduycop.js và doduycop.css
- index.html và fullname.css: trang trí tùy ý, có dấu ấn cá nhân, có form nhập được thông tin.
- fullname.js: lấy dữ liệu trên form, gửi đến api nodered đã làm ở bước 2.5, nhận về json, dùng json trả về để tạo giao diện phù hợp với kết quả truy vấn của bạn.

2.7. Nhận xét bài làm của mình:

- Đã hiểu quá trình cài đặt các phần mềm và các thư viện như nào?
- Đã hiểu cách sử dụng nodered để tạo api back-end như nào?
- Đã hiểu cách front-end tương tác với back-end ra sao?

2.1. Cài đặt Apache web server:

❖ Bước 1: Vô hiệu hoá IIS

- Nhấn Start -> gõ cmd -> Chuột phải vào Command Prompt -> chọn Run as administrator -> Sau đó nhập lệnh: iisreset /stop

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>iisreset /stop

Attempting stop...
Internet services successfully stopped

C:\Windows\System32>
```

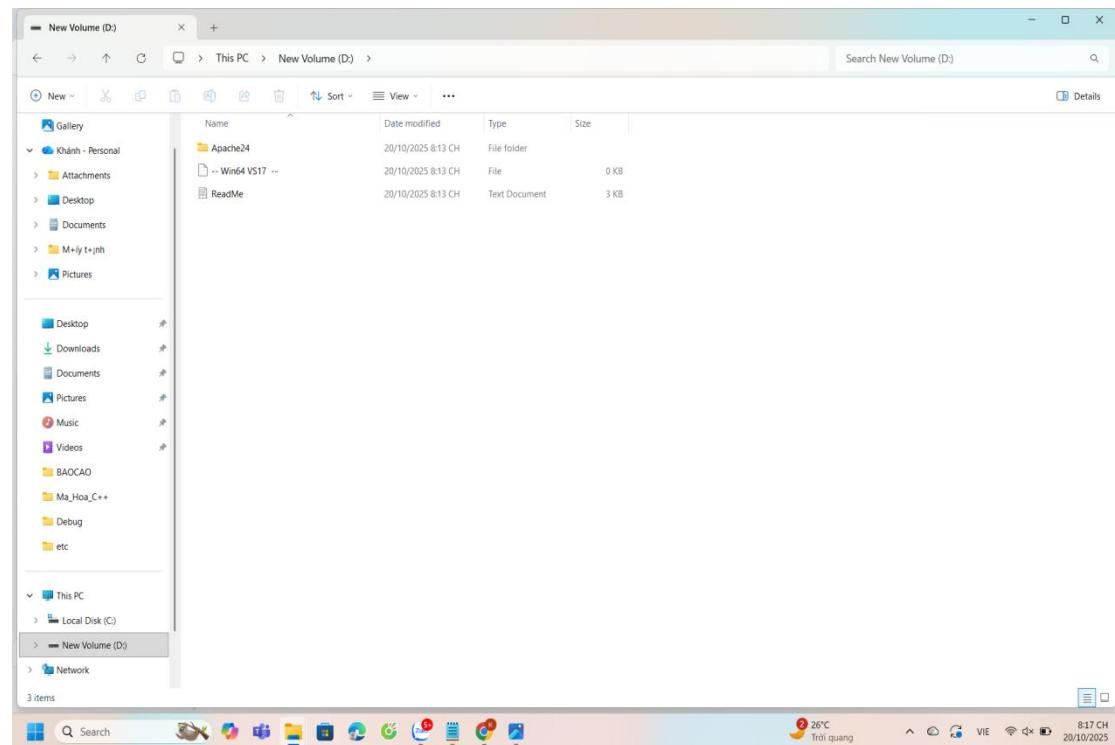
❖ Bước 2. Download apache server

- Truy cập link: <https://www.apachelounge.com/download/> để download apache
- Sau khi tải xong sẽ xuất hiện tệp:



❖ Bước 3: Giải nén ra ổ D:

- Chuột phải vào file vừa tải -> chọn Extract All... -> Chọn nơi muốn giải nén: D:\ -> Nhấn Extract
- Sau khi giải nén xong, sẽ hiện thư mục: D:\Apache24\

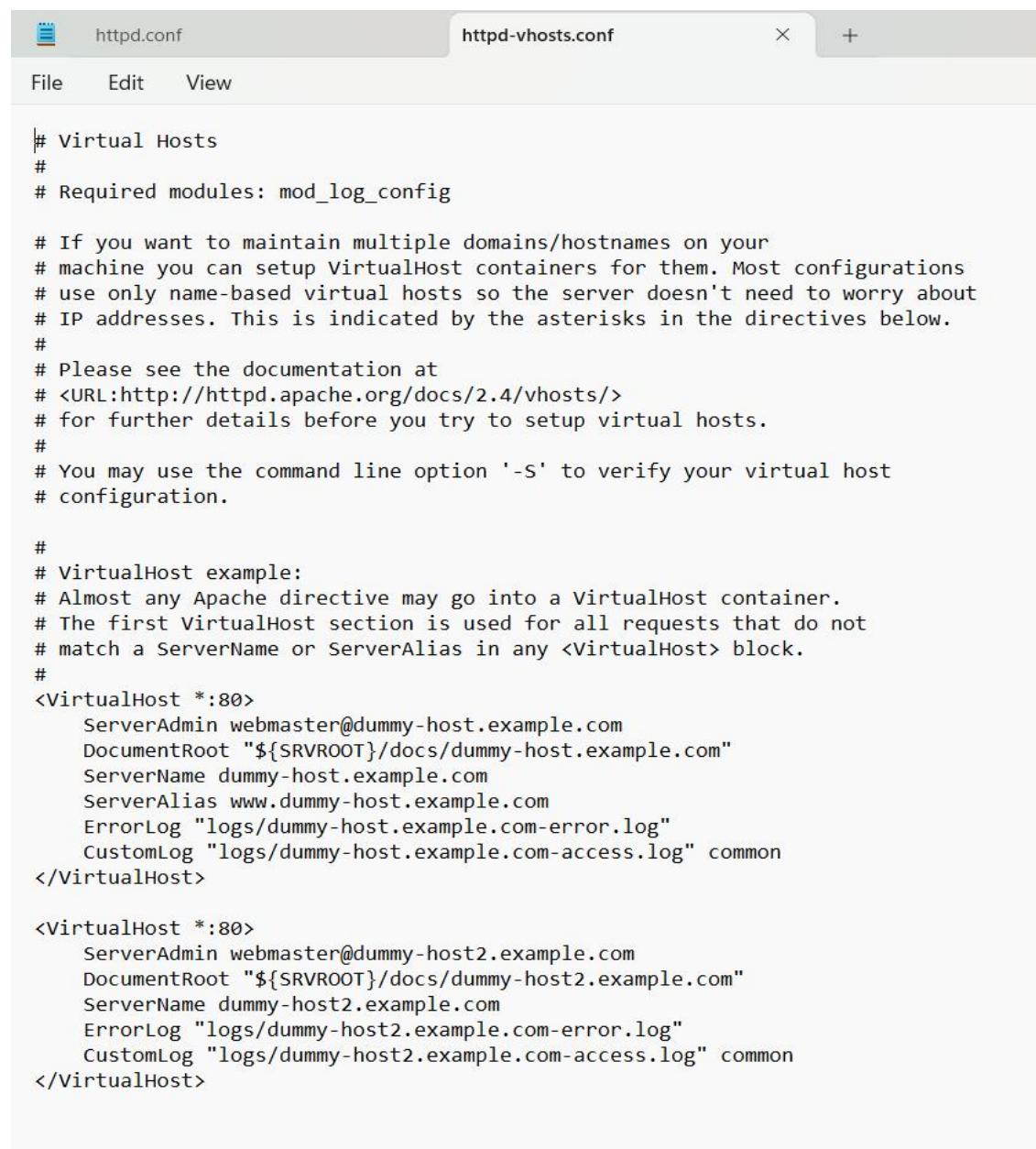


❖ **Bước 4: Cấu hình file: D:\Apache24\conf\httpd.conf**

- Mở file: D:\Apache24\conf\httpd.conf
- Sửa ServerRoot: ServerRoot "c:/Apache24" => ServerRoot "D:/Apache24"
- Sau mở httpd.conf -> Tìm dòng: #Include conf/extra/httpd-vhosts.conf và bỏ dấu # để bật file vhosts.

❖ **Bước 5: Cấu hình file: D:\Apache24\conf\extra\httpd-vhosts.conf**

- Mở file: D:\Apache24\conf\extra\httpd-vhosts.conf
- Sau khi mở file sẽ hiển thị:



```

httpd.conf httpd-vhosts.conf × +
File Edit View

# Virtual Hosts
#
# Required modules: mod_log_config

# If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most configurations
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at
# <URL:http://httpd.apache.org/docs/2.4/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-s' to verify your virtual host
# configuration.

#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ServerName or ServerAlias in any <VirtualHost> block.
#
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot "${SRVROOT}/docs/dummy-host.example.com"
    ServerName dummy-host.example.com
    ServerAlias www.dummy-host.example.com
    ErrorLog "logs/dummy-host.example.com-error.log"
    CustomLog "logs/dummy-host.example.com-access.log" common
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host2.example.com
    DocumentRoot "${SRVROOT}/docs/dummy-host2.example.com"
    ServerName dummy-host2.example.com
    ErrorLog "logs/dummy-host2.example.com-error.log"
    CustomLog "logs/dummy-host2.example.com-access.log" common
</VirtualHost>

```

- Thêm vào cuối file nội dung sau:

```

<VirtualHost *:80>
    ServerAdmin admin@dauvankhanh.com
    DocumentRoot "D:/Apache24/dauvankhanh"
    ServerName dauvankhanh.com
    ServerAlias www.dauvankhanh.com
    ErrorLog "logs/dauvankhanh-error.log"
    CustomLog "logs/dauvankhanh-access.log" common
</VirtualHost>

```

- Trong file D:\Apache24\conf\httpd.conf sửa: DocumentRoot "D:/Apache24/dauvankhanh" và <Directory "D:/Apache24/dauvankhanh">

❖ **Bước 6: Tạo thư mục D:\Apache24\dauvankhanh**

- Trong đó tạo file index.html:

```
<!DOCTYPE html>
<html lang="vi">
<head>
    <meta charset="UTF-8">
    <title>Website cá nhân Đậu Văn Khánh</title>
    <style>
        body {
            font-family: "Segoe UI", Arial, sans-serif;
            background-color: #fafafa;
            text-align: center;
            margin-top: 15%;
        }
        h1 {
            color: #222;
        }
    </style>
</head>
<body>
    <h1>Xin chào! Đây là trang <strong>dauvankhanh.com</strong> chạy trên
    Apache.</h1>
</body>
</html>
```

❖ **Bước 7: Cấu hình file hosts để fake domain**

- Mở file: C:\Windows\System32\drivers\etc\hosts (Mở bằng Notepad quyền Admin (Run as Administrator))
- Sau khi mở, thêm dòng: 127.0.0.1 dauvankhanh.com

```

# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97    rhino.acme.com        # source server
#      38.25.63.10    x.acme.com           # x client host
#
# localhost resolution is handled within DNS itself.
#
#       127.0.0.1          localhost
#       ::1               localhost
# Added by Docker Desktop
192.168.0.111 host.docker.internal
192.168.0.111 gateway.docker.internal
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
127.0.0.1 dauvankhanh.com

```

Ln 28, Col 26 1.053 characters Plain text 100% Windows (CRLF) UTF-8 with BOM

❖ Bước 8: Cài đặt và khởi động Apache

- Mở cmd -> chạy quyền admin (Run as Administrator), rồi chạy:

`cd /d D:\Apache24\bin`

`httpd.exe -k install`

`httpd.exe -k start`

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd /d D:\Apache24\bin

D:\Apache24\bin>httpd.exe -k start
[Mon Oct 20 21:59:08.743856 2025] [mpm_winnt:error] [pid 10760:tid 448] AH00433: Apache2.4: Service is already installed
.

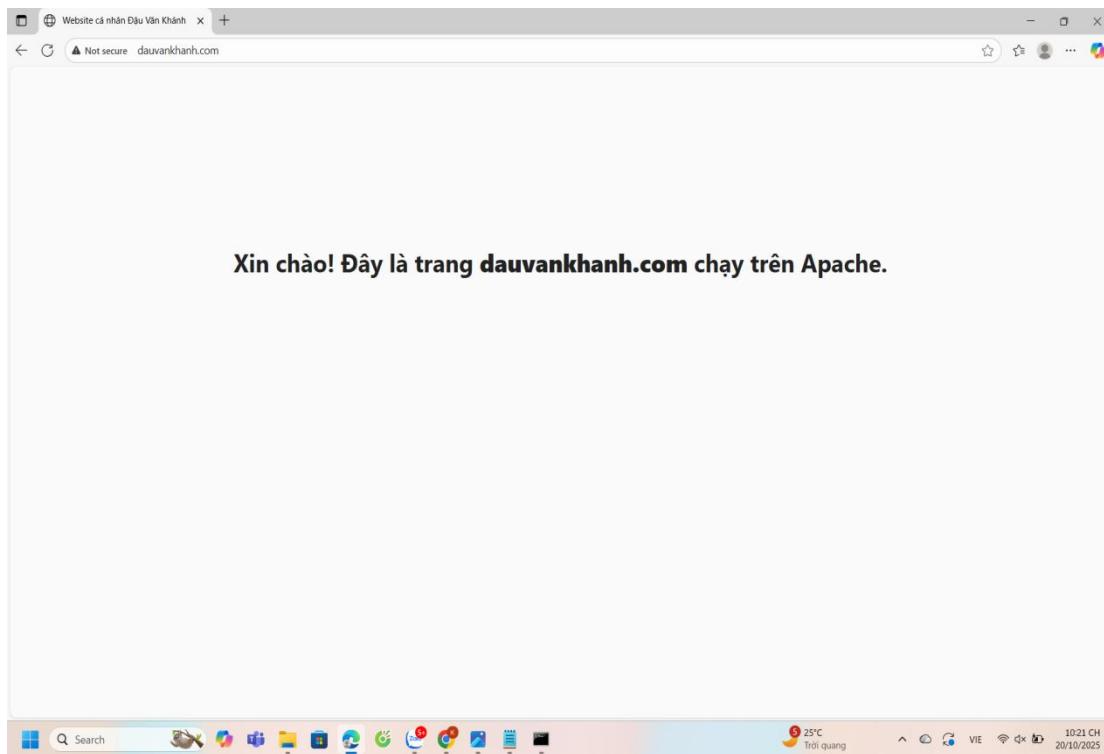
D:\Apache24\bin>httpd.exe -k start
AH00558: httpd.exe: Could not reliably determine the server's fully qualified domain name, using fe80::e6ad:d54:e00:c1cf
. Set the 'ServerName' directive globally to suppress this message
(OS 10048)Only one usage of each socket address (protocol/network address/port) is normally permitted. : AH00072: make_sock: could not bind to address [::]:80
(OS 10048)Only one usage of each socket address (protocol/network address/port) is normally permitted. : AH00072: make_sock: could not bind to address 0.0.0.0:80
AH00451: no listening sockets available, shutting down
AH00015: Unable to open logs

D:\Apache24\bin>

```

❖ Bước 9: Kiểm tra kết quả

- Mở trình duyệt, gõ: <http://dauvankhanh.com>

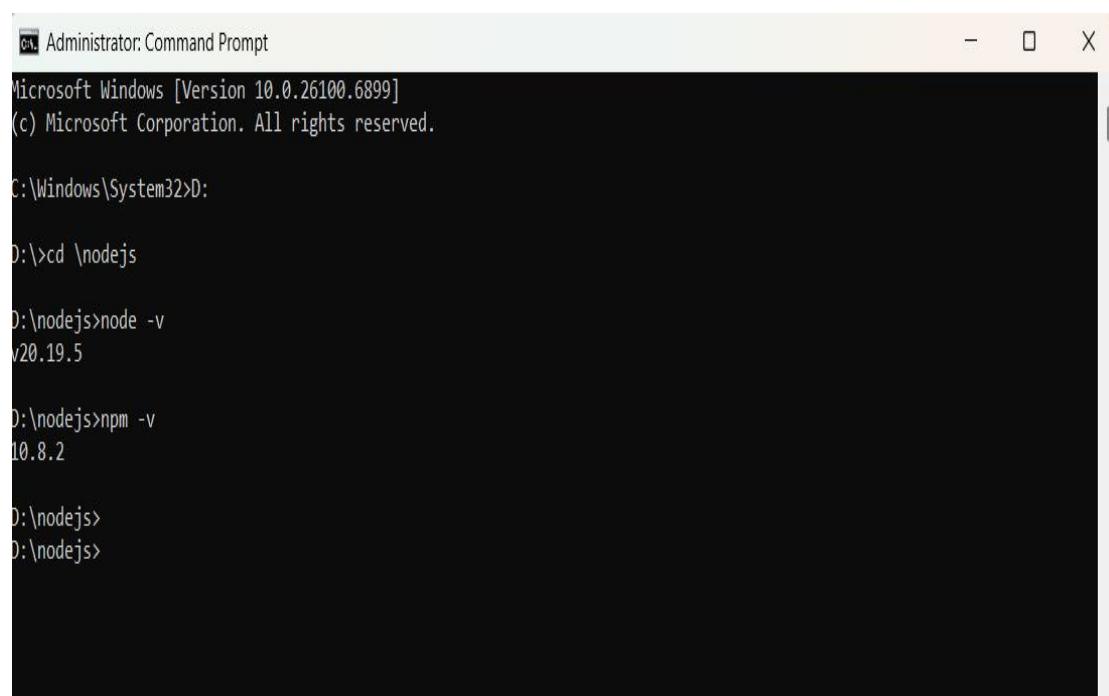


2.2. Cài đặt nodejs và nodered

2.2.1. Cài đặt nodejs

- Tải file: <https://nodejs.org/dist/v20.19.5/node-v20.19.5-x64.msi>
- Cài đặt bằng giao diện (GUI):
 - + Nhấn file: node-v20.19.5-x64.msi
 - + Chọn Next → I Agree → Custom.
 - + Ở phần chọn đường dẫn, đổi thành: D:\nodejs
 - + Bấm Next → Install.
 - + Khi hoàn tất, Node.js sẽ được cài vào D:\nodejs và npm đi kèm.
 - + Kiểm tra: Mở cmd(admin) và chạy:

```
cd \nodejs
node -v
npm -v
```



```
C:\Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>D:

D:>cd \nodejs

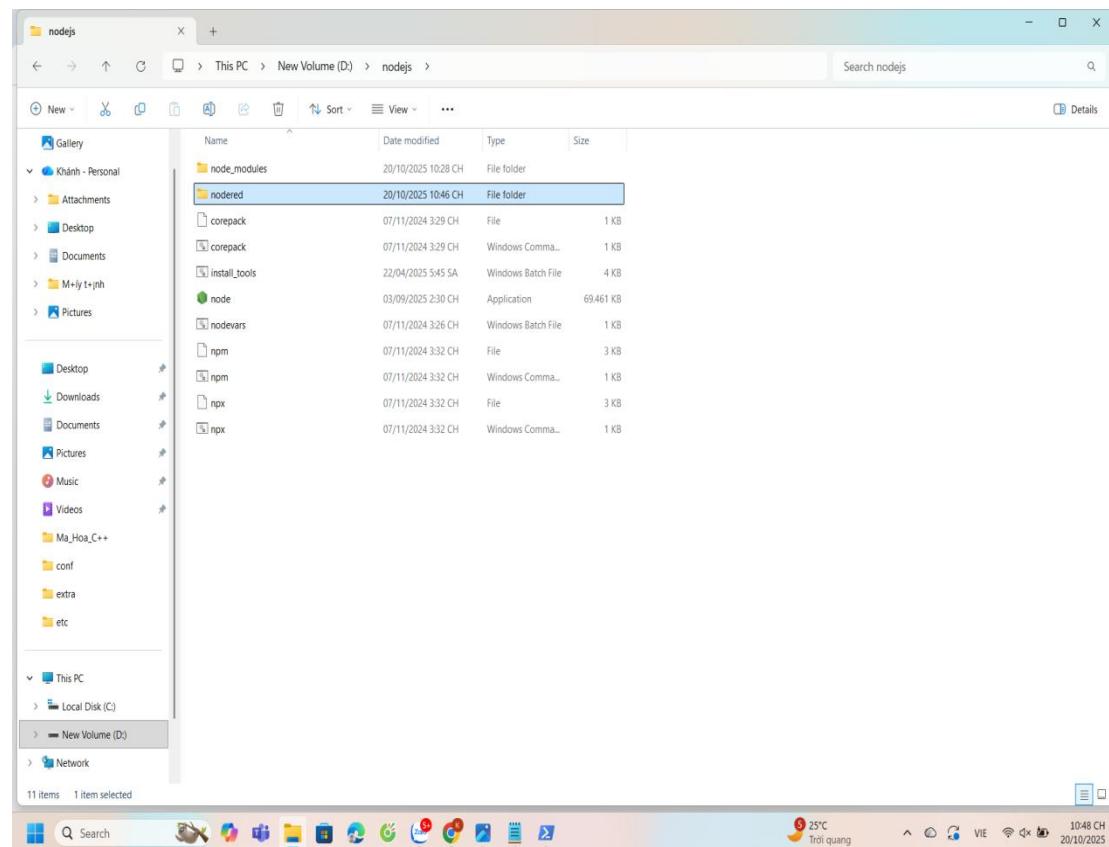
D:\nodejs>node -v
v20.19.5

D:\nodejs>npm -v
10.8.2

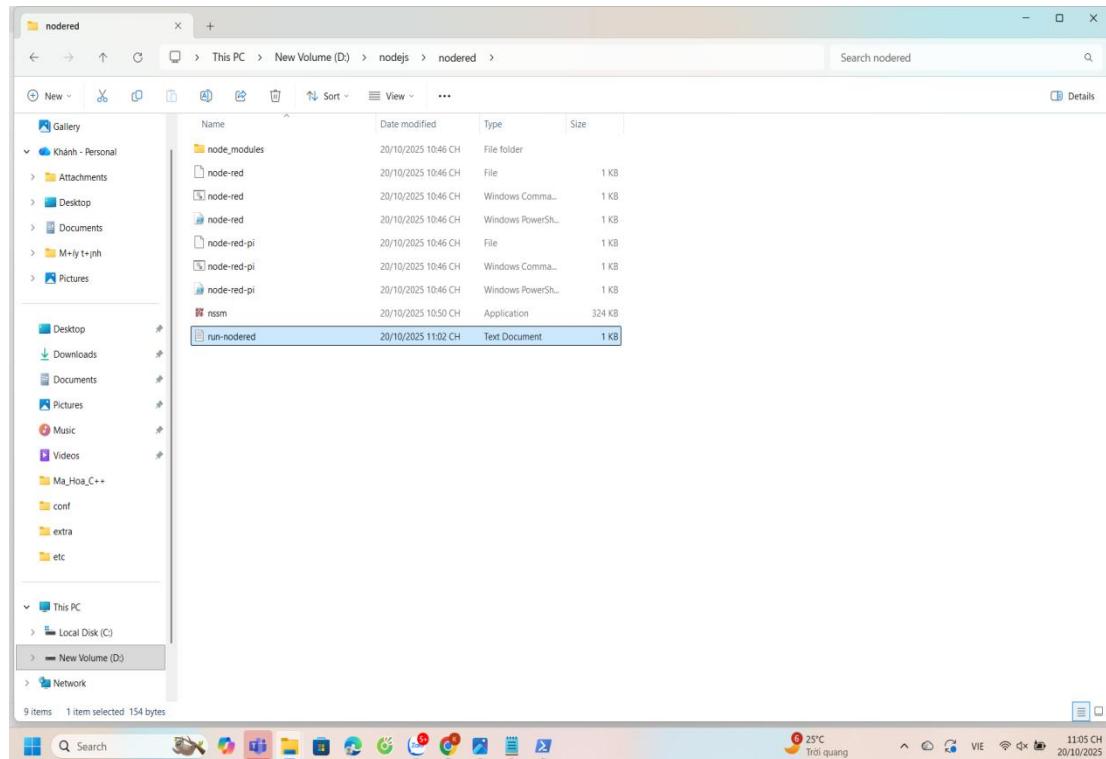
D:\nodejs>
D:\nodejs>
```

2.2.2. Cài đặt nodered

- Chạy cmd (Admin), vào thư mục D:\nodejs, chạy lệnh npm install -g --unsafe-perm node-red --prefix "D:\nodejs\nodered"
- Sau khi chạy cmd, kết quả nodered hiển thị trong thư mục D:\nodejs



- Cài nssm: <https://nssm.cc/release/nssm-2.24.zip>.
- Tạo file "D:\nodejs\nodered\run-nodered.cmd"



- Cài service a1-nodered bằng nssm:
 - + Mở cmd (Admin), chuyển đến thư mục nodered: cd /d D:\nodejs\nodered
 - + Cài đặt service "a1-nodered" bằng lệnh: nssm.exe install a1-nodered "D:\nodejs\nodered\run-nodered.cmd"

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd /d D:\nodejs\nodered

D:\nodejs\nodered>nssm.exe install a1-nodered "D:\nodejs\nodered\run-nodered.cmd"
Service "a1-nodered" installed successfully!

D:\nodejs\nodered>
```

- Chạy service "a1-nodered" bằng lệnh: nssm start a1-nodered

2.3. Tạo cơ sở dữ liệu tùy ý trên mssql (sql server 2022)

- Tạo username, password:

```
-- USE master;
-- GO

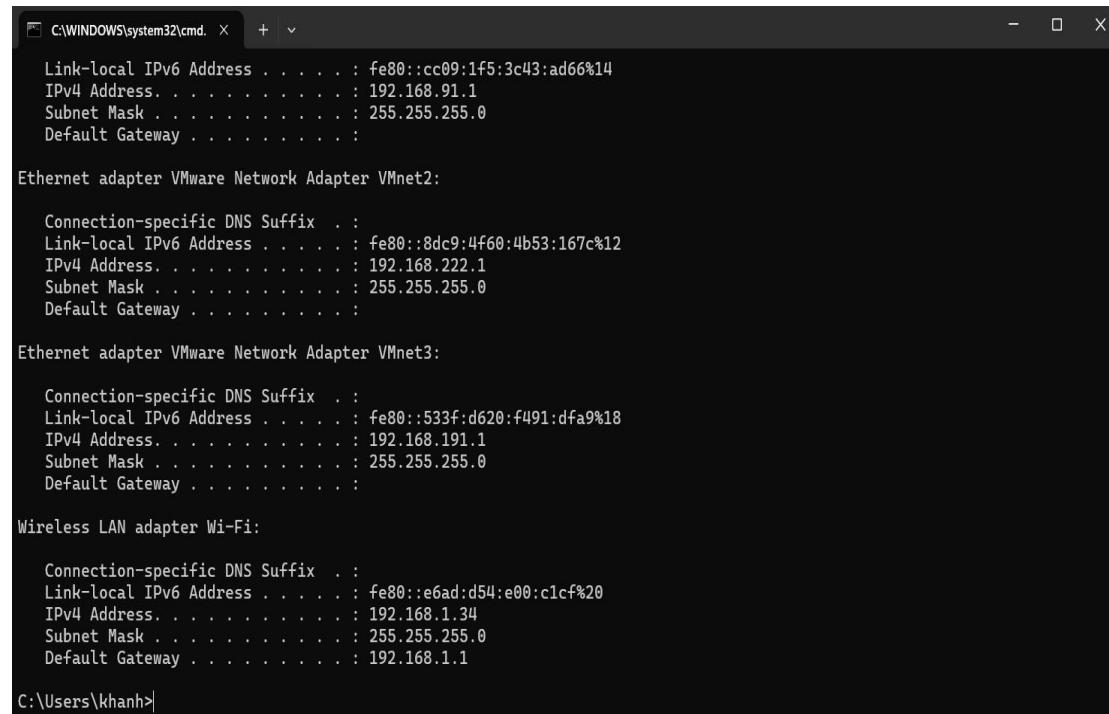
-- IF EXISTS (SELECT * FROM sys.server_principals WHERE name = N'KhanhDV')
--     DROP LOGIN KhanhDV;
-- GO

CREATE LOGIN KhanhDV WITH PASSWORD = '123456';
GO

USE DauVanKhanh;
GO

--CREATE USER KhanhDV FOR LOGIN KhanhDV;
--EXEC sp_addrolemember N'db_owner', N'KhanhDV';
--GO
```

- IP:

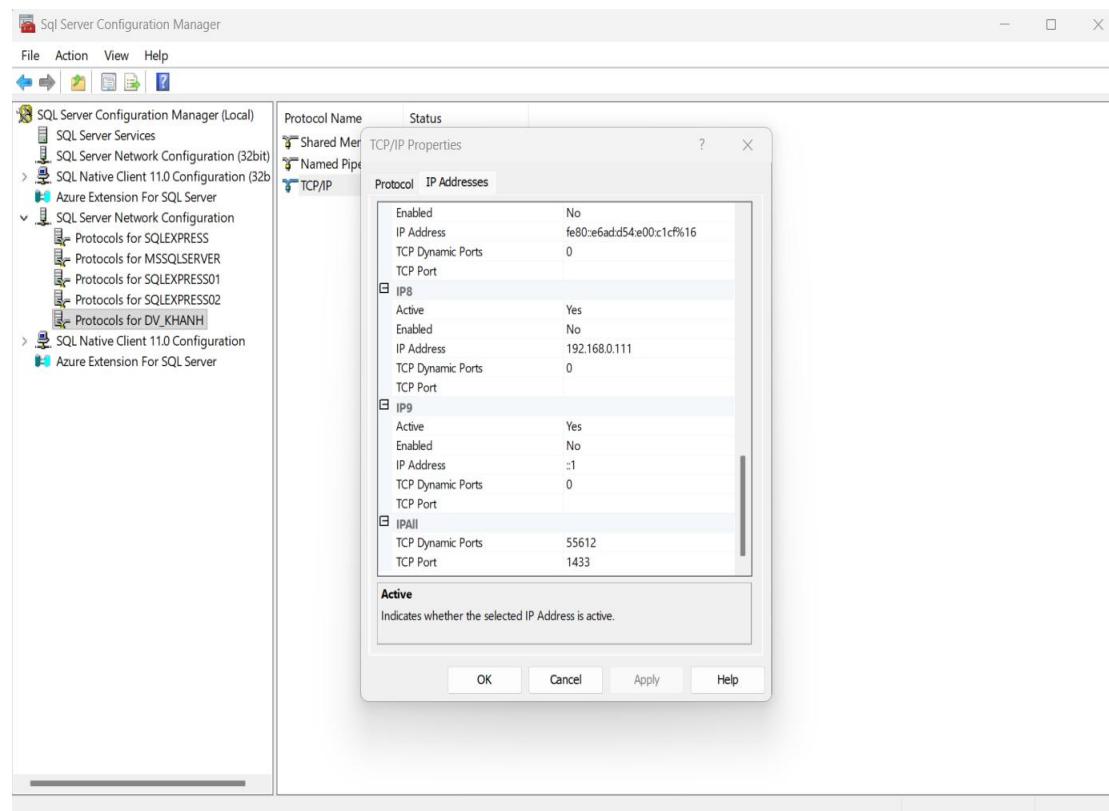


The screenshot shows a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.' with several network adapter configurations listed:

- Ethernet adapter VMware Network Adapter VMnet2:**
 - Connection-specific DNS Suffix : fe80::8dc9:4f60:4b53:167c%12
 - Link-local IPv6 Address : fe80::cc09:1f5:3c43:ad66%14
 - IPv4 Address : 192.168.91.1
 - Subnet Mask : 255.255.255.0
 - Default Gateway :
- Ethernet adapter VMware Network Adapter VMnet3:**
 - Connection-specific DNS Suffix :
 - Link-local IPv6 Address : fe80::533f:d620:f491:dfa9%18
 - IPv4 Address. : 192.168.191.1
 - Subnet Mask : 255.255.255.0
 - Default Gateway :
- Wireless LAN adapter Wi-Fi:**
 - Connection-specific DNS Suffix :
 - Link-local IPv6 Address : fe80::e6ad:d54:e00:c1cf%20
 - IPv4 Address. : 192.168.1.34
 - Subnet Mask : 255.255.255.0
 - Default Gateway : 192.168.1.1

C:\Users\khanh>

- Port:



- Tạo DB_DauVanKhanh:

```
-- =====
-- TẠO CƠ SỞ DỮ LIỆU
-- =====

USE master;
GO

-- Tạo CSDL mới
CREATE DATABASE DauVanKhanh;
GO

-- Sử dụng CSDL
USE DauVanKhanh;
GO
```

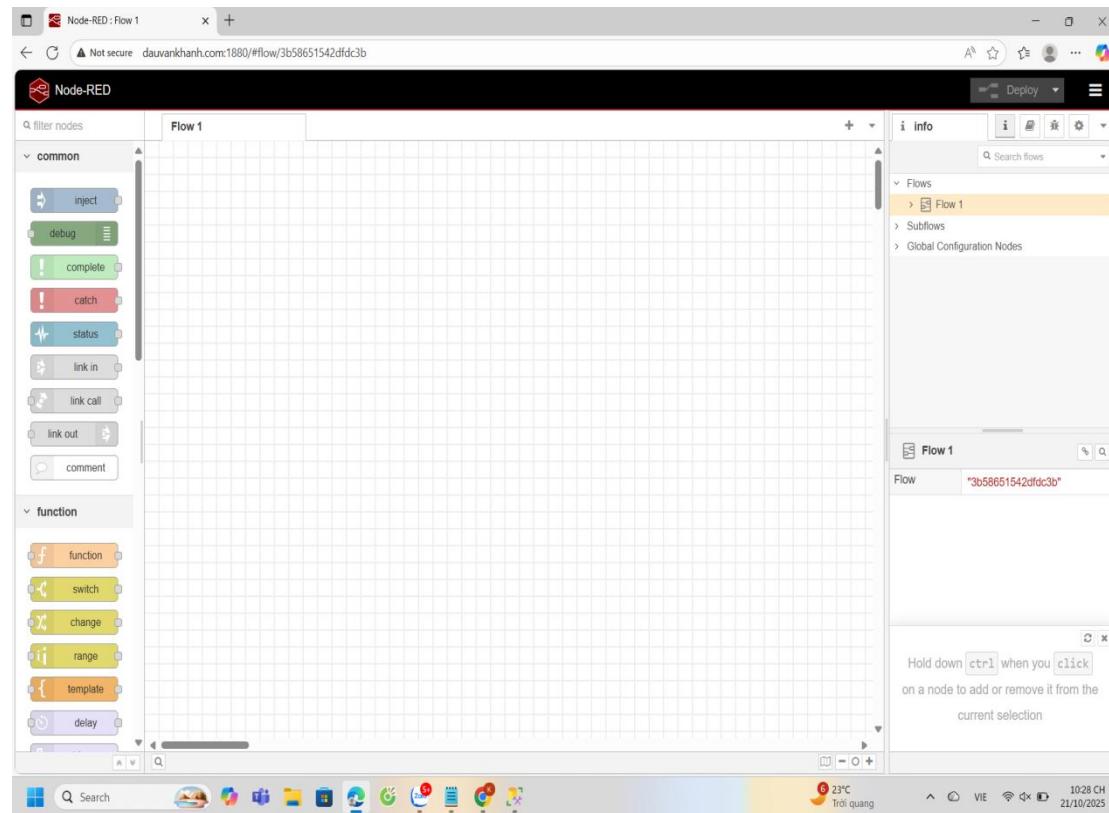
- Tạo table_KhachHang:

```
bt2_web.sql - DESK...anKhanh (sa (115))  X  SQLQuery1.sql - DE...VanKhanh (sa (63))
-- =====
-- TẠO BẢNG KHÁCH HÀNG
-- =====
CREATE TABLE KhachHang (
    MaKH INT IDENTITY(1,1) PRIMARY KEY,
    HoTen NVARCHAR(100) NOT NULL,
    Email NVARCHAR(100),
    SDT VARCHAR(15),
    DiaChi NVARCHAR(200)
);
GO

-- =====
-- THÊM DỮ LIỆU MẪU
-- =====
INSERT INTO KhachHang (HoTen, Email, SDT, DiaChi) VALUES
(N'Đậu Văn Khánh', 'dvkhanh@example.com', '0912345678', N'Bắc Ninh'),
(N'Nguyễn Thị Lan', 'lannt@example.com', '0987654321', N'Hà Nội'),
(N'Phạm Văn Nam', 'nampham@example.com', '0905123123', N'Thái Nguyên');
GO
```

2.4. Cài đặt thư viện trên nodered

- Truy cập giao diện nodered bằng url: <http://dauvankhanh.com:1880>



- Trong giao diện Nodered: Vào Menu (≡ góc phải trên) -> chọn Manage palette
- > Chọn tab Install -> Gõ lần lượt các thư viện và nhấn Install cho từng cái.
- Cài đặt thư viện:

+ node-red-contrib-mssql-plus

The screenshot shows the 'User Settings' interface with the 'Install' tab selected. In the search bar, 'node-red-contrib-mssql-plus' is typed. Below the search bar, the result 'node-red-contrib-mssql-plus' is listed with a checkmark icon, indicating it is installed. The description states: 'A node-red node to execute queries, stored procedures and bulk inserts in Microsoft SQL Server and Azure Databases SQL2000 ~ SQL2022'. The version is 0.13.1, updated 5 months ago, with 1,061 installations. A 'installed' button is visible on the right.

+ node-red-node-mysql

The screenshot shows the 'User Settings' interface with the 'Install' tab selected. In the search bar, 'node-red-contrib-telegrambot' is typed. Below the search bar, the result 'node-red-contrib-telegrambot' is listed with a checkmark icon, indicating it is installed. The description states: 'Telegram bot nodes for Node-RED'. The version is 16.3.3, updated 2 weeks ago, with 1,793 installations. A 'installed' button is visible on the right.

+ node-red-contrib-moment

The screenshot shows the 'User Settings' interface with the 'Install' tab selected. In the search bar, 'node-red-contrib-moment' is typed. Below the search bar, the result 'node-red-contrib-moment' is listed with a checkmark icon, indicating it is installed. The description states: 'Node-Red Node that produces formatted Date/Time output using the Moment.JS library. Timezone, dst and locale aware.'. The version is 5.0.0, updated 2 years, 5 months ago, with 7,348 installations. A 'installed' button is visible on the right.

+ node-red-contrib-influxdb

User Settings

Close

Settings Nodes Install

Palette Node-RED Community catalogue sort: ↗ ↘ ↙ ↖ ↗ ↘ ↙ ↖

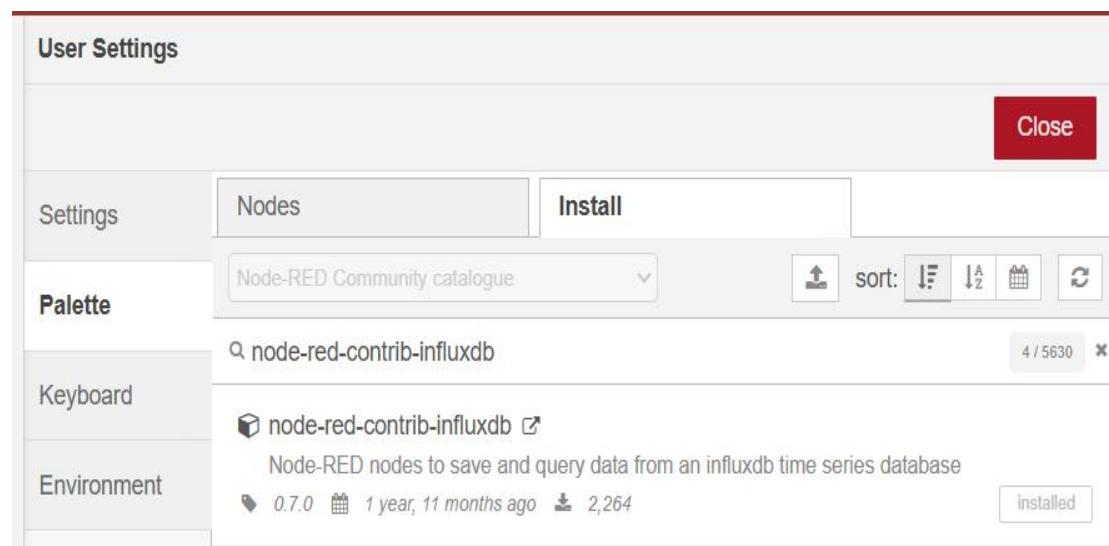
Keyboard

Environment

Q node-red-contrib-influxdb 4 / 5630 ×

node-red-contrib-influxdb Node-RED nodes to save and query data from an influxdb time series database

0.7.0 1 year, 11 months ago 2,264 installed



+ node-red-contrib-duckdns

User Settings

Close

Settings Nodes Install

Palette Node-RED Community catalogue sort: ↗ ↘ ↙ ↖ ↗ ↘ ↙ ↖

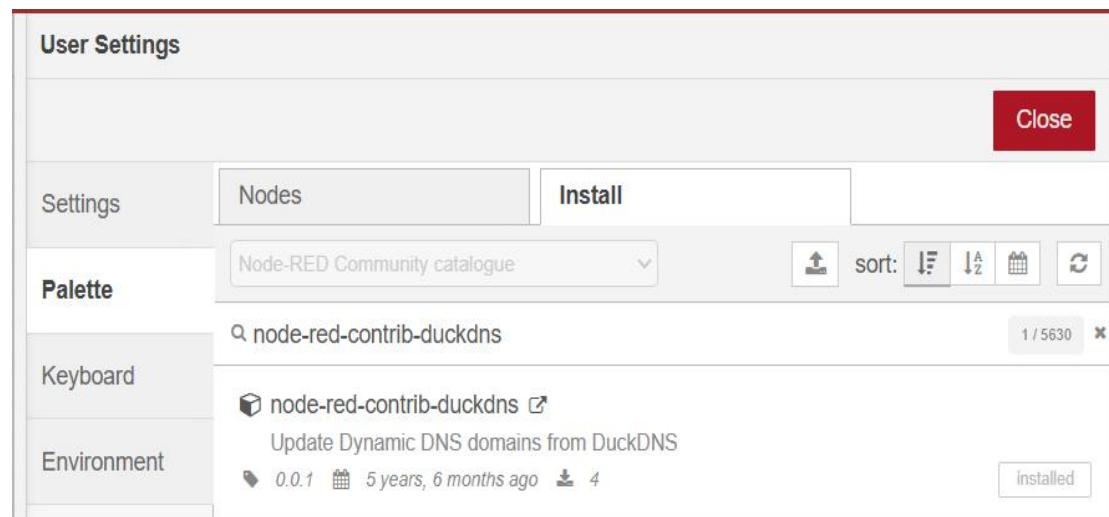
Keyboard

Environment

Q node-red-contrib-duckdns 1 / 5630 ×

node-red-contrib-duckdns Update Dynamic DNS domains from DuckDNS

0.0.1 5 years, 6 months ago 4 installed



+ node-red-contrib-cron-plus

User Settings

Close

Settings Nodes Install

Palette Node-RED Community catalogue sort: ↗ ↘ ↙ ↖ ↗ ↘ ↙ ↖

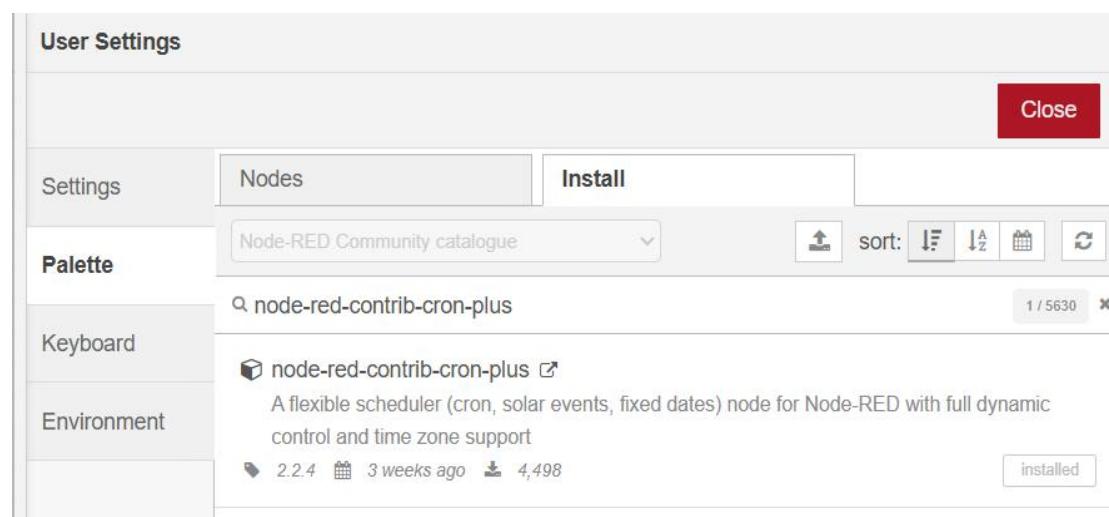
Keyboard

Environment

Q node-red-contrib-cron-plus 1 / 5630 ×

node-red-contrib-cron-plus A flexible scheduler (cron, solar events, fixed dates) node for Node-RED with full dynamic control and time zone support

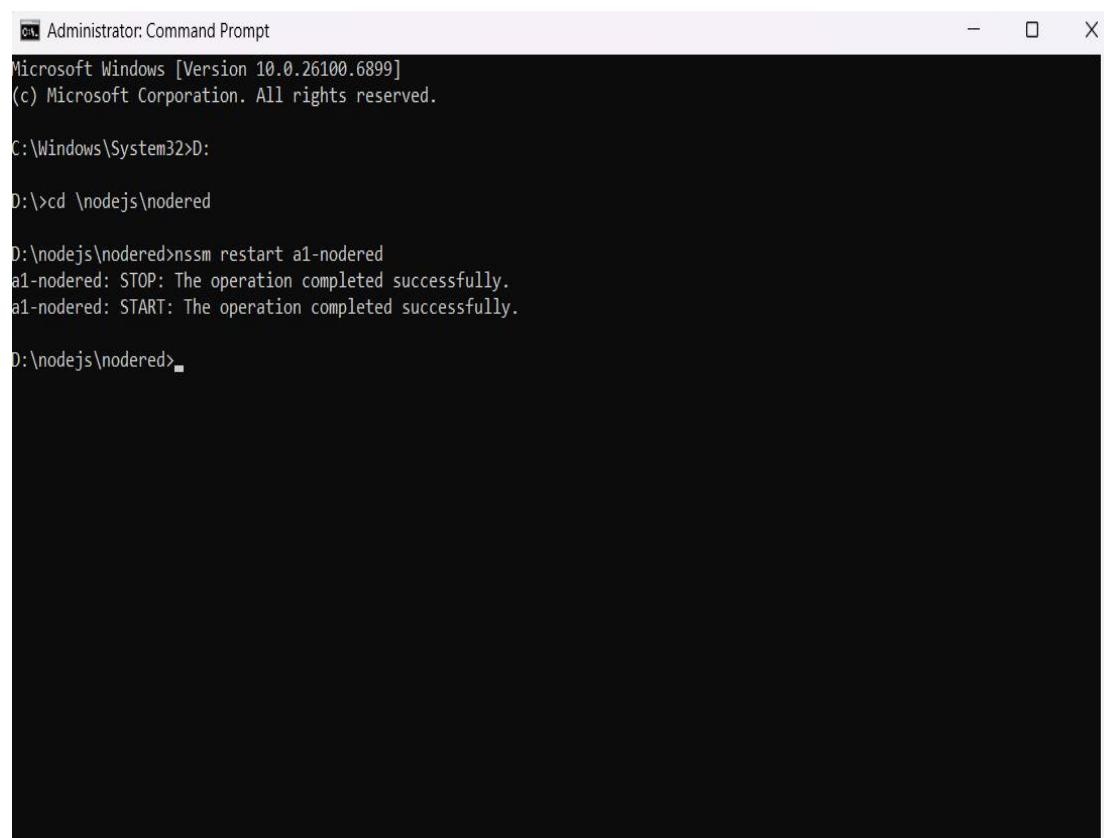
2.2.4 3 weeks ago 4,498 installed



- Sửa file "D:\nodejs\nodered\work\settings.js":

```
/** To password protect the Node-RED editor and admin API, the
following
 * property can be used. See
https://nodered.org/docs/security.html for details.
 */
adminAuth: {
    type: "credentials",
    users: [
        {
            username: "admin",
            password: "$2y$10
$00laXHWJR8LtiVpA8hukp.uzyg5M0ho7H7nMB/182yg78Pq8ISOWu",
            permissions: "*"
        }
    ],
},
```

- Mã hoá mật khẩu có thể thiết lập bằng tool: <https://tms.tnut.edu.vn/pw.php>
- Chạy lại nodered bằng cách: mở cmd (quyền Admin), vào thư mục 'D:\nodejs\nodered' và chạy lệnh 'nssm restart a1-nodered'



The screenshot shows an Administrator Command Prompt window on Windows. The title bar says "Administrator: Command Prompt". The window content is as follows:

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

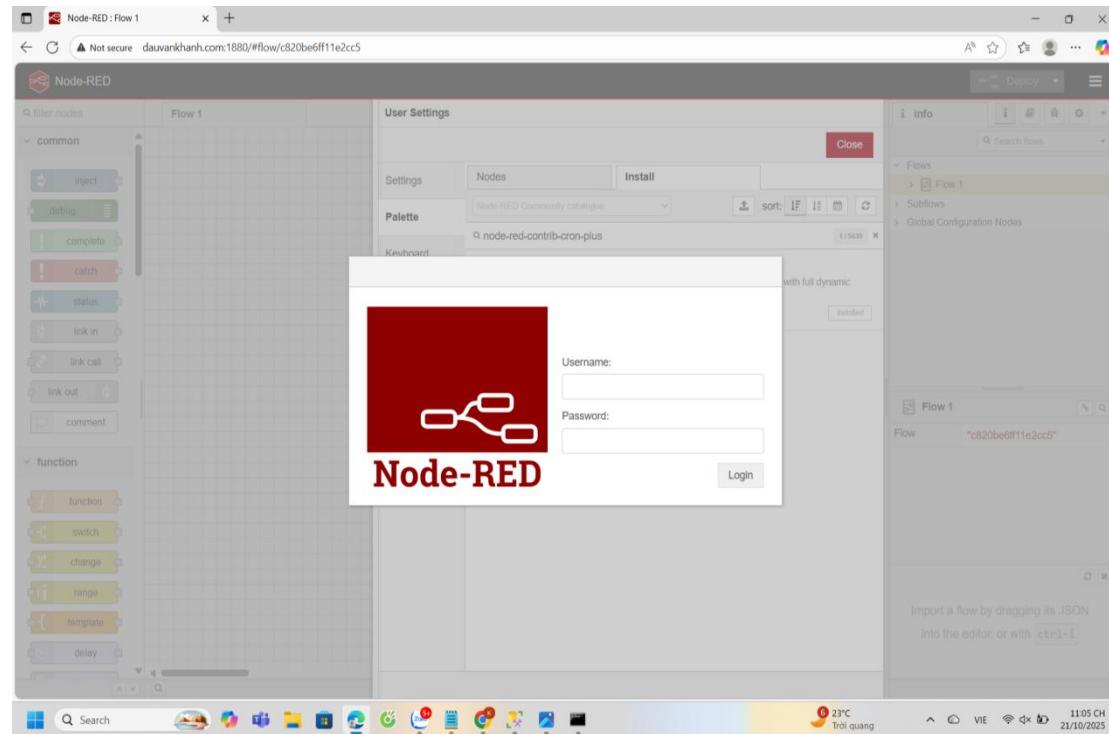
C:\Windows\System32>D:

D:\>cd \nodejs\nodered

D:\nodejs\nodered>nssm restart a1-nodered
a1-nodered: STOP: The operation completed successfully.
a1-nodered: START: The operation completed successfully.

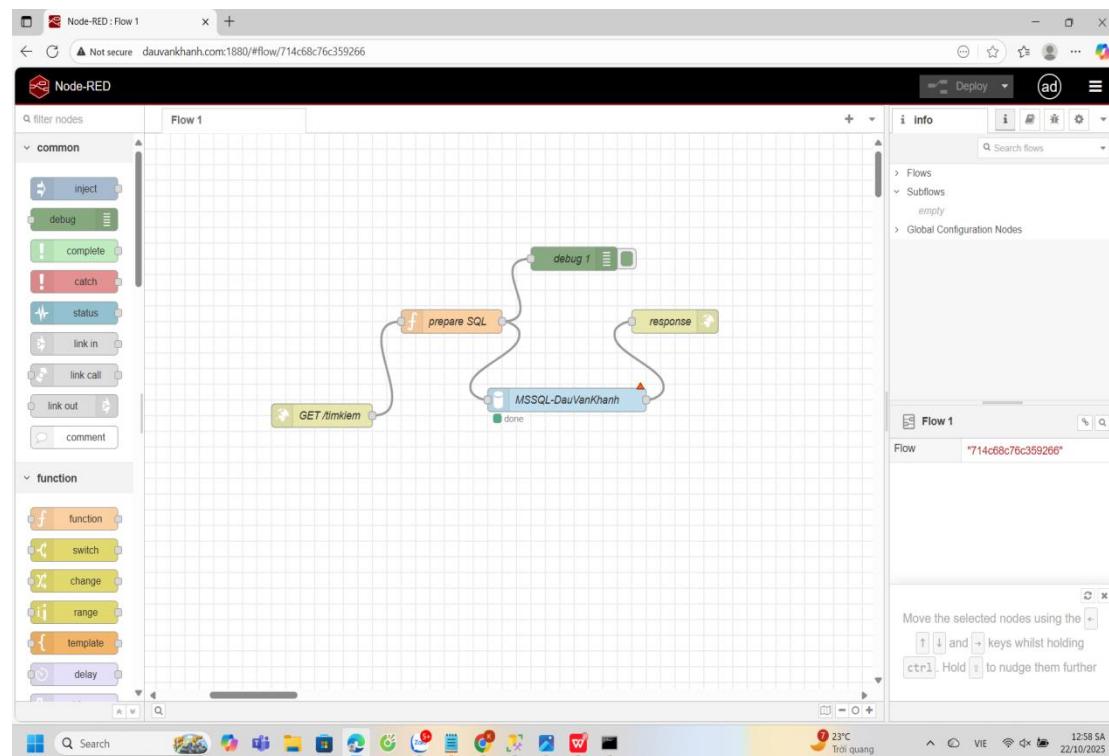
D:\nodejs\nodered>
```

- Nodered sẽ yêu cầu nhập mật khẩu mới vào được giao diện cho admin tại: <http://dauvankhanh.com:1880>



2.5. Tạo api back-end bằng nodered

- Logic flow sẽ gồm 4 node theo thứ tự sau (thứ tự nối dây):



- Test api thông qua trình duyệt: <http://dauvankhanh.com:1880/timkiem?q=thi>

The screenshot shows a browser window with the address bar containing "dauvankhanh.com:1880/timkiem? q=thi". The page content is a JSON object:

```
[{"MaKH": 2, "HoTen": "Nguyễn Thị Lan", "Email": "lannt@example.com", "SDT": "0987654321", "DiaChi": "Hà Nội"}]
```

- Test api thông qua trình duyệt:

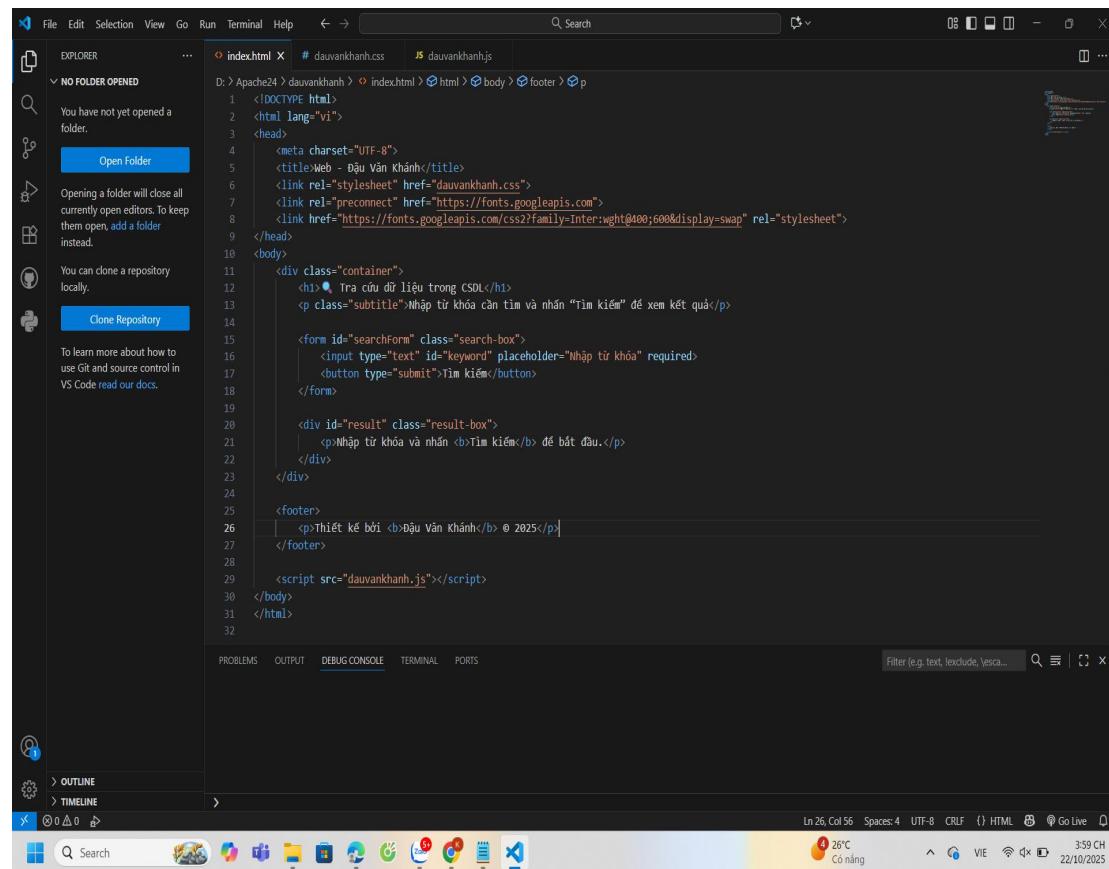
<http://dauvankhanh.com:1880/timkiem?q=Khánh>

The screenshot shows a browser window with the address bar containing "dauvankhanh.com:1880/timkiem? q=Khánh". The page content is a JSON object:

```
[{"MaKH": 1, "HoTen": "Đậu Văn Khánh", "Email": "dvkhanh@example.com", "SDT": "0962213503", "DiaChi": "Bắc Ninh"}]
```

2.6. Tạo giao diện front-end

- Tạo file index.html:



```

File Edit Selection View Go Run Terminal Help ⏪ ⏴ Search
EXPLORER # index.html # dauvankhanh.css JS dauvankhanh.js
D: > Apache24 > dauvankhanh > index.html > html > body > footer > p
1 <!DOCTYPE html>
2 <html lang="vi">
3 <head>
4   <meta charset="UTF-8">
5   <title>Web - Đậu Văn Khanh</title>
6   <link rel="stylesheet" href="dauvankhanh.css">
7   <link rel="preconnect" href="https://fonts.googleapis.com">
8   <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600&display=swap" rel="stylesheet">
9 </head>
10 <body>
11   <div class="container">
12     <h1>Tra cứu dữ liệu trong CSDL</h1>
13     <p class="subtitle">Nhập từ khóa cần tìm và nhấn "Tìm kiếm" để xem kết quả</p>
14
15     <form id="searchForm" class="search-box">
16       <input type="text" id="keyword" placeholder="Nhập từ khóa" required>
17       <button type="submit">Tìm kiếm</button>
18     </form>
19
20     <div id="result" class="result-box">
21       <p>Nhập từ khóa và nhấn <b>Tìm kiếm</b> để bắt đầu.</p>
22     </div>
23
24   </div>
25
26   <footer>
27     <p>Thiết kế bởi <b>Đậu Văn Khanh</b> © 2025</p>
28   </footer>
29
30   <script src="dauvankhanh.js"></script>
31 </body>
32 </html>

```

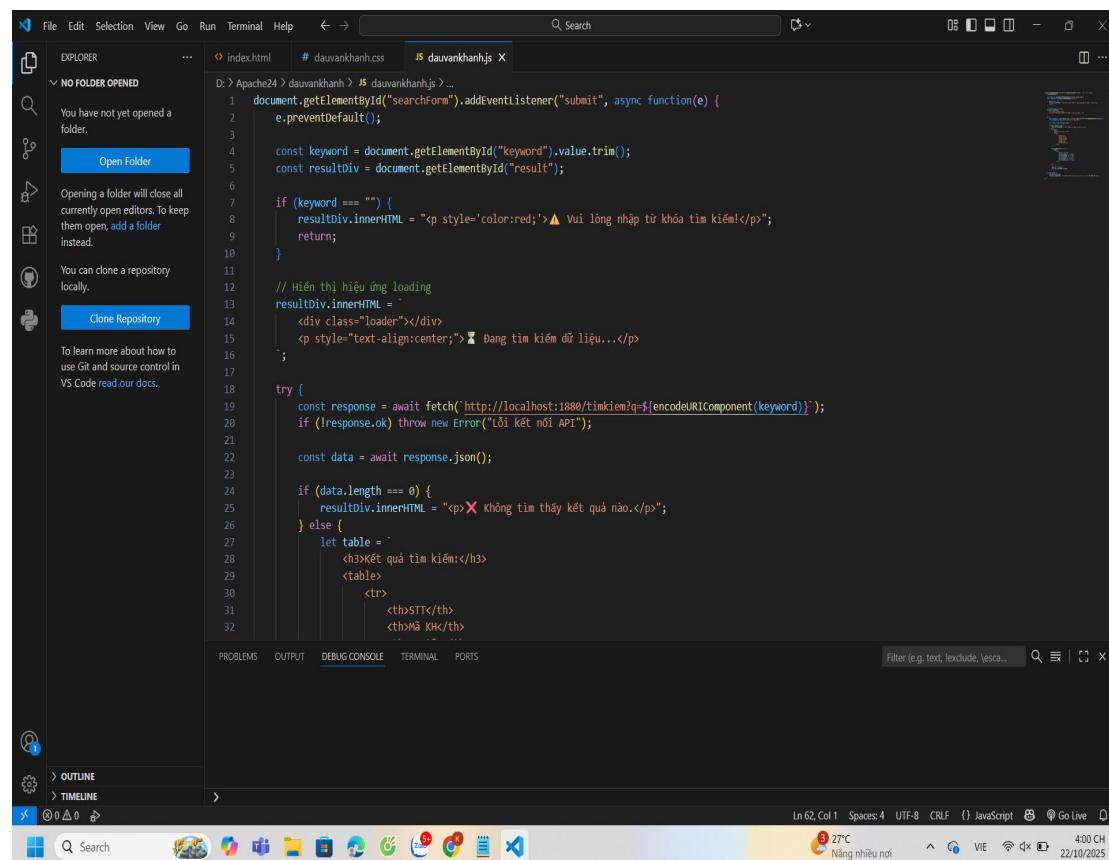
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Filter (e.g. text, exclude, \esc...)

Ln 26, Col 56 Spaces: 4 UTF-8 CRLF ⓘ HTML ⓘ Go Live ⓘ

26°C Có nắng 3:59 CH 22/10/2025

- Tạo file dauvankhanh.js:



```

File Edit Selection View Go Run Terminal Help ⏪ ⏴ Search
EXPLORER # index.html # dauvankhanh.css JS dauvankhanh.js
D: > Apache24 > dauvankhanh > JS dauvankhanh.js > ...
1 document.getElementById("searchForm").addEventListener("submit", async function(e) {
2   e.preventDefault();
3
4   const keyword = document.getElementById("keyword").value.trim();
5   const resultDiv = document.getElementById("result");
6
7   if (keyword === "") {
8     resultDiv.innerHTML = "<p style='color:red;'>Vui lòng nhập từ khóa tìm kiếm!</p>";
9     return;
10 }
11
12 // Hiển thị hiệu ứng loading
13 resultDiv.innerHTML =
14   <div class="loader"></div>
15   <p style="text-align:center;">Đang tìm kiếm dữ liệu...</p>;
16
17 try {
18   const response = await fetch(`http://localhost:1888/timkiem?q=${encodeURIComponent(keyword)}`);
19   if (!response.ok) throw new Error("Lỗi kết nối API");
20
21   const data = await response.json();
22
23   if (data.length === 0) {
24     resultDiv.innerHTML = "<p>Không tìm thấy kết quả nào.</p>";
25   } else {
26     let table =
27       <h3>Kết quả tìm kiếm:</h3>
28       <table>
29         <thead>
30           <tr>
31             <th>STT</th>
32             <th>Mã KH</th>

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Filter (e.g. text, exclude, \esc...)

Ln 62, Col 1 Spaces: 4 UTF-8 CRLF ⓘ JavaScript ⓘ Go Live ⓘ

27°C Nắng nhiều nơi 4:00 CH 22/10/2025

- Tao file dauvankhanh.css:

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. The left sidebar has icons for Explorer, Search, Find, Open Folder, and Clone Repository. The main editor area displays a CSS file named 'dauvankhanh.css' with the following code:

```
/* Phông chữ hiện đại */
body {
    font-family: 'Inter', sans-serif;
    background: linear-gradient(120deg, #d4fc79, #96e6a1);
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
    flex-direction: column;
}

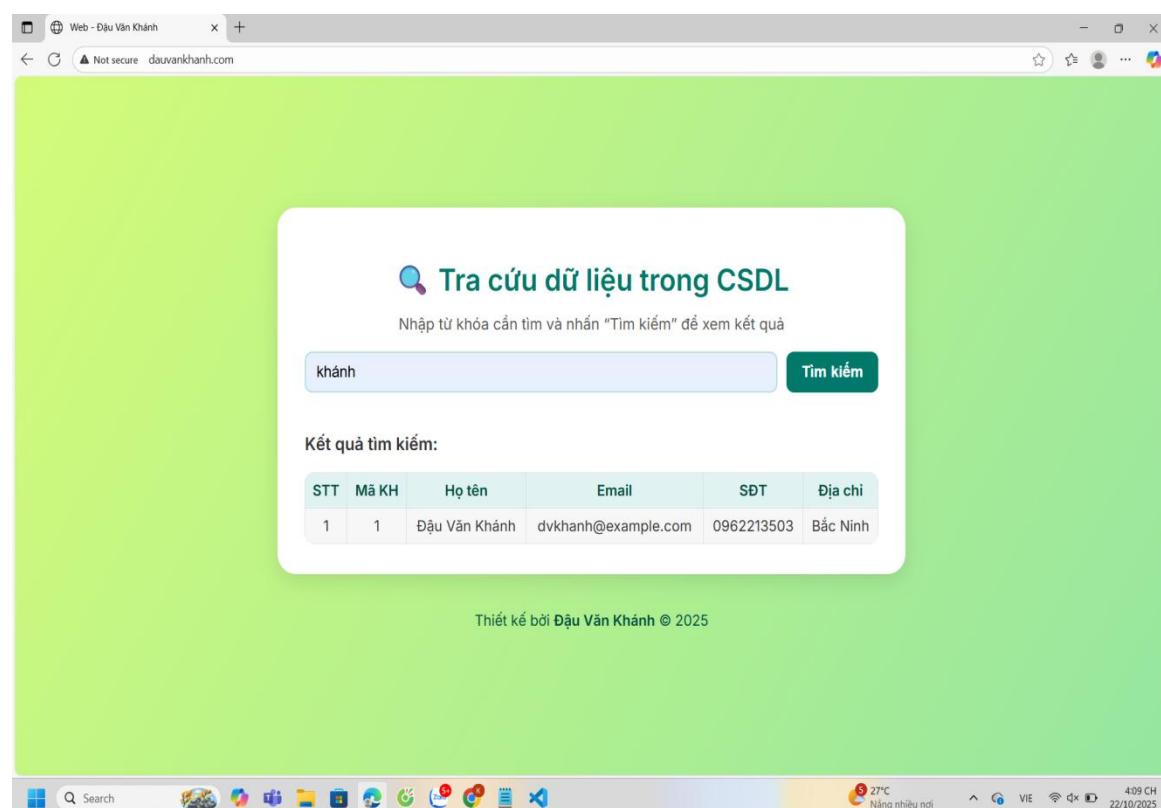
/* Hộp chính */
.container {
    width: 480px;
    background: white;
    padding: 30px 35px;
    border-radius: 20px;
    box-shadow: 0 6px 20px rgba(0, 0, 0, 0.1);
    text-align: center;
    animation: fadeIn 0.8s ease;
    overflow: hidden;
}

@keyframes fadeIn {
    from { opacity: 0; transform: translateY(10px); }
    to { opacity: 1; transform: translateY(0); }
}

h1 {
    color: #00796b;
    font-size: 1.8em;
}
```

The bottom navigation bar includes PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. A status bar at the bottom shows Ln 172, Col 1, Spaces: 4, UTF-8, CRLF, CSS, Go Live, and a date/time stamp of 22/10/2025. The bottom left also shows OUTLINE and TIMELINE.

- Kết quả tìm kiếm:



2.7. Nhận xét bài làm:

2.7.1. Hiểu quá trình cài đặt phần mềm và thư viện

- Biết cách cài Apache trên Windows từ bản ZIP (giải nén) và chỉnh httpd.conf/httpd-vhosts.conf.
- Hiểu ServerRoot / DocumentRoot và cách ánh xạ domain nội bộ qua file hosts.
- Biết cách cài Node.js và Node-RED, cách cài Node-RED global vào thư mục tùy chỉnh.
- Biết dùng nssm để chạy Node-RED như một Windows Service (start/stop/restart).
- Biết cài thêm package vào Node-RED qua Manage Palette.

2.7.2. Hiểu cách sử dụng Node-RED để tạo API backend

- Biết luồng dữ liệu trong Node-RED: http in → xử lý → database → http response.
- Biết cách viết Function nodes để chuẩn bị SQL (chi tiết: msg.topic, msg.params) và để format response (set msg.headers, msg.statusCode).
- Biết cách cấu hình MSSQL node (server, port, database, user/pass), và cách debug kết quả bằng Debug node.
- Biết tạo endpoints REST đơn giản (GET) và xử lý query string (msg.req.query).

2.7.3. Hiểu cách front-end tương tác với back-end

- Hiểu nguyên lý: frontend gửi HTTP request (GET/POST) tới API, nhận JSON, xử lý DOM để hiển thị.
- Biết sử dụng fetch() với encodeURIComponent để gửi tham số, xử lý mã lỗi (response.ok), hiển thị spinner, xử lý trường hợp không có kết quả.
- Biết xử lý CORS (nếu gọi từ host khác) — trong Node-RED đã set header Access-Control-Allow-Origin: * để test nội bộ.
- Hiểu tách trách nhiệm: backend trả dữ liệu (JSON), frontend chịu trách nhiệm render và UX.

BÀI TẬP 3: LẬP TRÌNH ỦNG DỤNG WEB TRÊN NỀN LINUX

ĐỀ BÀI:

1. Cài đặt môi trường linux: SV chọn 1 trong các phương án

- Enable wsl: cài đặt docker desktop
- Enable wsl: cài đặt ubuntu
- Sử dụng Hyper-V: cài đặt ubuntu
- Sử dụng VMware : cài đặt ubuntu
- Sử dụng Virtual Box: cài đặt ubuntu

2. Cài đặt Docker (nếu dùng docker desktop trên windows thì nó có ngay).

3. Sử dụng 1 file docker-compose.yml để cài đặt các docker container sau:

mariadb (3306), phpmyadmin (8080), nodered/node-red (1880), influxdb (8086), grafana/grafana (3000), nginx (80,443)

4. Lập trình web frontend+backend:

SV chọn 1 trong các web sau:

4.1. Web thương mại điện tử

- Tạo web dạng Single Page Application (SPA), chỉ gồm 1 file index.html, toàn bộ giao diện do javascript sinh động.
- Có tính năng login, lưu phiên đăng nhập vào cookie và session.

Thông tin login lưu trong cơ sở dữ liệu của mariadb, được dev quản trị bằng phpmyadmin, yêu cầu sử dụng mã hoá khi gửi login.

Chỉ cần login 1 lần, bao giờ logout thì mới phải login lại.

- Có tính năng liệt kê các sản phẩm bán chạy ra trang chủ.
- Có tính năng liệt kê các nhóm sản phẩm.
- Có tính năng liệt kê sản phẩm theo nhóm.
- Có tính năng tìm kiếm sản phẩm.
- Có tính năng chọn sản phẩm (đưa sản phẩm vào giỏ hàng, thay đổi số lượng sản phẩm trong giỏ, cập nhật tổng tiền).
- Có tính năng đặt hàng, nhập thông tin giao hàng => được 1 đơn hàng.

- Có tính năng dành cho admin: Thống kê xem có bao nhiêu đơn hàng, call để xác nhận và cập nhật thông tin đơn hàng. chuyển cho bộ phận đóng gói, gửi bưu điện, cập nhật mã COD, tình trạng giao hàng, huỷ hàng,...
- Có tính năng dành cho admin: biểu đồ thống kê số lượng mặt hàng bán được trong từng ngày. (sử dụng grafana).
- backend: sử dụng nodered xử lý request gửi lên từ javascript, phản hồi về json.

4.2 Web IOT: Giám sát dữ liệu IOT.

- Tạo web dạng Single Page Application (SPA), chỉ gồm 1 file index.html, toàn bộ giao diện do javascript sinh động.
- Có tính năng login, lưu phiên đăng nhập vào cookie và session

Thông tin login lưu trong cơ sở dữ liệu của mariadb, được dev quản trị bằng phpmyadmin, yêu cầu sử dụng mã hoá khi gửi login.

Chỉ cần login 1 lần, bao giờ logout thì mới phải login lại.

- hiển thị giá trị mới nhất của các thông số đang giám sát, khi click vào thì hiển thị đồ thị lịch sử quá trình thay đổi (gọi grafana iframe để hiển thị).
- backend: Sử dụng nodered để đọc dữ liệu từ các cảm biến (có thể dùng api online để lấy dữ liệu theo giờ gian thực), nodered sẽ lưu dữ liệu mới nhất (dạng update) vào cơ sở dữ liệu mariadb (sử dụng phpmyadmin để tạo table và quản trị lần đầu).

nodered sẽ lưu dữ liệu (insert) vào influxdb để lưu giá trị lịch sử, để cho grafana dùng để hiển thị biểu đồ.

5. Nginx làm web-server

- Cấu hình nginx để chạy được website qua url <http://fullname.com> (thay fullname bằng chuỗi ko dấu viết liền tên của bạn).
- Cấu hình nginx để <http://fullname.com/nodered> truy cập vào nodered qua cổng 80, (dù nodered đang chạy ở port 1880).
- Cấu hình nginx để <http://fullname.com/grafana> truy cập vào grafana qua cổng 80, (dù grafana đang chạy ở port 3000).

BÀI LÀM

3.1. Cài đặt môi trường linuxs

❖ Bước 1: Cài đặt Ubuntu:

- Mở cmd (quyền Admin) và gõ: wsl --install để cài Ubuntu
- Sau khi cài xong gõ: wsl -d Ubuntu để mở
- Sau đó nhập lần lượt: Enter new UNIX username và New password
- Sau khi nhập xong username và password sẽ hiển thị: khanh@DESKTOP-7I4R9SM:/mnt/c/Windows/System32\$

```

khanh@DESKTOP-7I4R9SM:/mnt/c/Windows/System32
C:\Windows\System32>wsl --install
Downloading: Ubuntu
Installing: Ubuntu
Distribution successfully installed. It can be launched via 'wsl.exe -d Ubuntu'

C:\Windows\System32>wsl -d Ubuntu

Provisioning the new WSL instance Ubuntu
This might take a while...

Create a default Unix user account: khanh
New password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

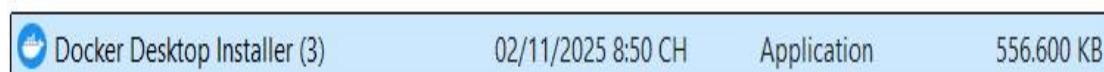
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

```

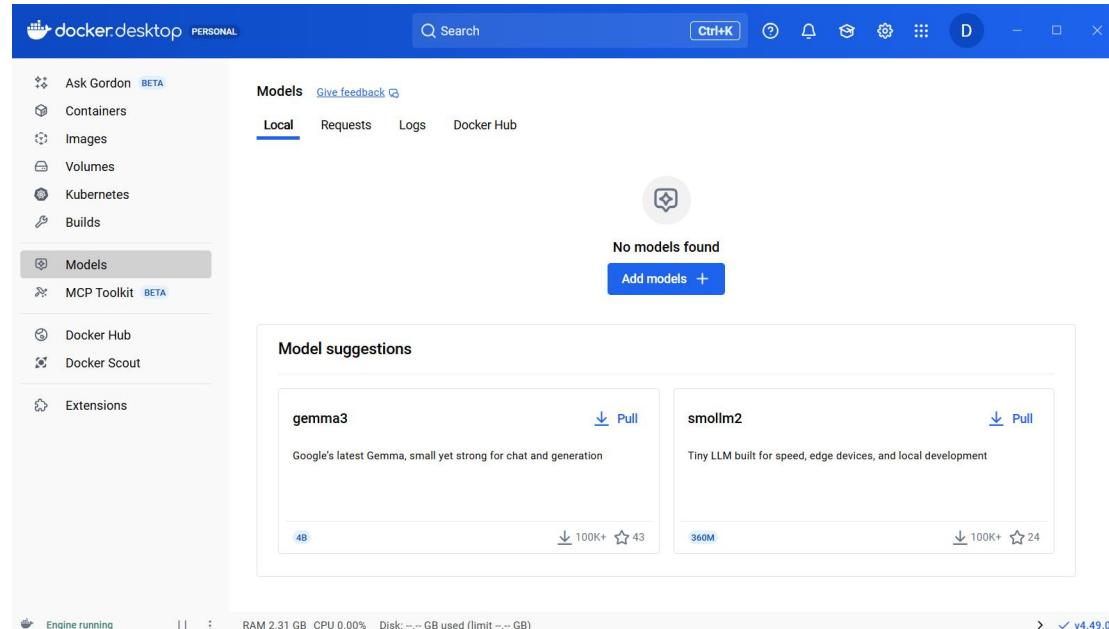
- Sau đó chạy lệnh: sudo apt update và sudo apt upgrade -y

❖ Bước 2: Cài đặt Docker Destop

- Truy cập link: <https://www.docker.com/> -> nhấn Download
- Sau khi tải về sẽ hiển thị file:

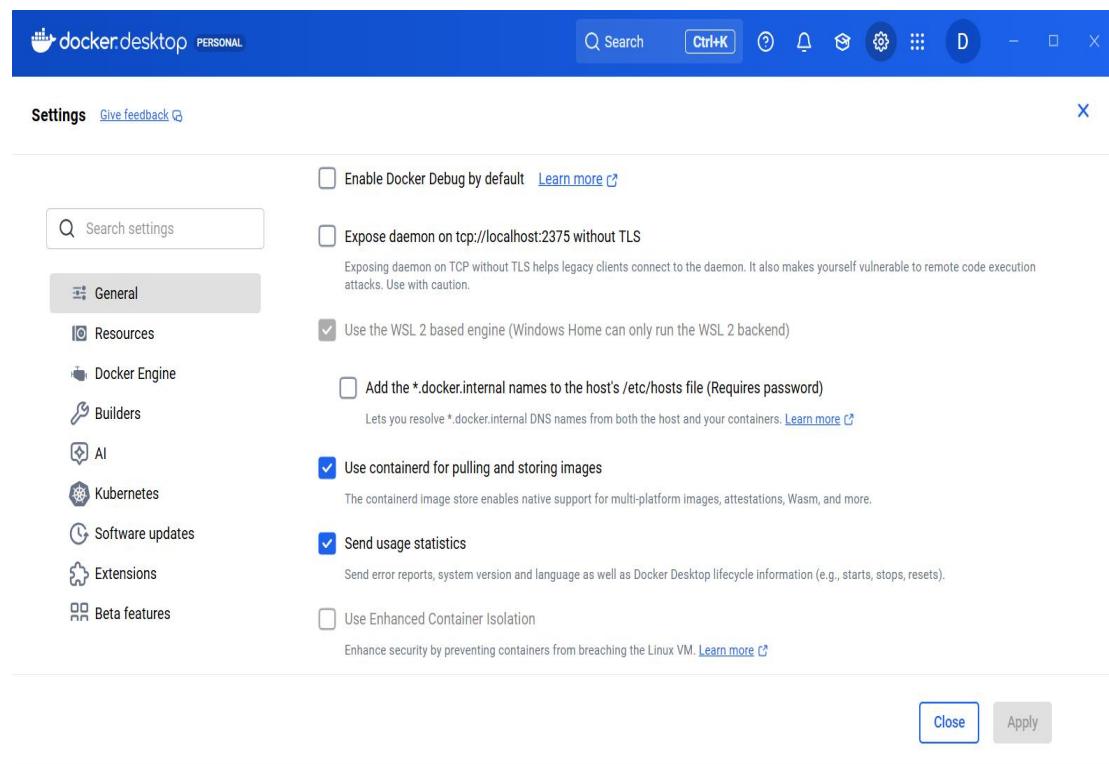


- Nhấp đúp vào tệp cài đặt Docker Desktop -> chọn Yes khi hộp thoại cấp quyền quản trị xuất hiện.
- Sau khi Docker Desktop khởi động, chọn Accept để đồng ý với các điều khoản sử dụng.
- Tiếp theo, đăng nhập bằng tài khoản Google/ GitHub của bạn hoặc đăng ký tài khoản mới nếu chưa có.
- Giao diện Docker Destop:



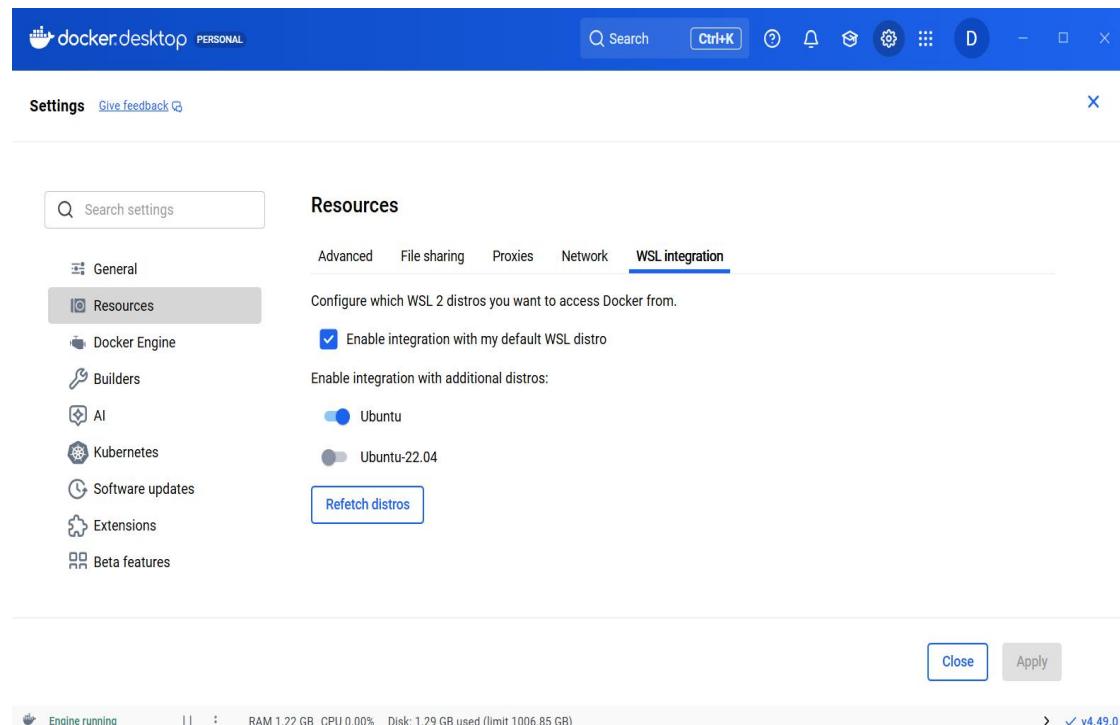
❖ Bước 3: Bật tích hợp Docker với Ubuntu (trong Docker Desktop)

- Mở Docker Desktop
- Vào Settings -> Chọn General: Tick ✓ “Use the WSL 2 based engine”



- Chuyển sang tab Resources -> Chọn WSL Integration:

- + Tick ✓ “Enable integration with my default WSL distro”
- + Bật Ubuntu hoặc Ubuntu-22.04



- Nhấn Apply & Restart -> Sau khi restart, Docker Desktop sẽ tự động kết nối với Ubuntu qua WSL2.

- Kiểm tra Ubuntu:

- + Mở cmd (quyền Admin) và nhập: wsl --list --verbose
- + Kết quả:

A screenshot of a Windows Command Prompt window titled 'Administrator: Command Prompt'. The window shows the following output:

```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>wsl --list --verbose
  NAME      STATE      VERSION
* Ubuntu    Running     2
  docker-desktop  Running     2
  Ubuntu-22.04   Stopped    2

C:\Windows\System32>
```

- Kiểm tra Docker trong Ubuntu

Mở Ubuntu (WSL) mà bạn vừa bật integration → gõ: docker version hoặc docker run hello-world

```

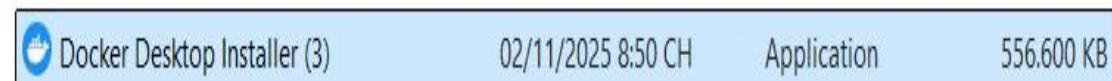
khanh@DESKTOP-7I4R9SM:~ $ docker version
Client:
  Version:          28.5.1
  API version:     1.51
  Go version:      go1.24.8
  Git commit:      e189ab8
  Built:           Wed Oct  8 12:16:30 2025
  OS/Arch:         linux/amd64
  Context:         default

Server: Docker Desktop 4.49.0 (208700)
Engine:
  Version:          28.5.1
  API version:     1.51 (minimum version 1.24)
  Go version:      go1.24.8
  Git commit:      f8215cc
  Built:           Wed Oct  8 12:17:24 2025
  OS/Arch:         linux/amd64
  Experimental:   false
  containerd:
    Version:        1.7.27
    GitCommit:      05044ec0a9a75232cad458027ca83437aae3f4da
  runc:
    Version:        1.2.5
    GitCommit:      v1.2.5-0-g59923ef
  docker-init:
    Version:        0.19.0
    GitCommit:      de40ad0
khanh@DESKTOP-7I4R9SM:~ $

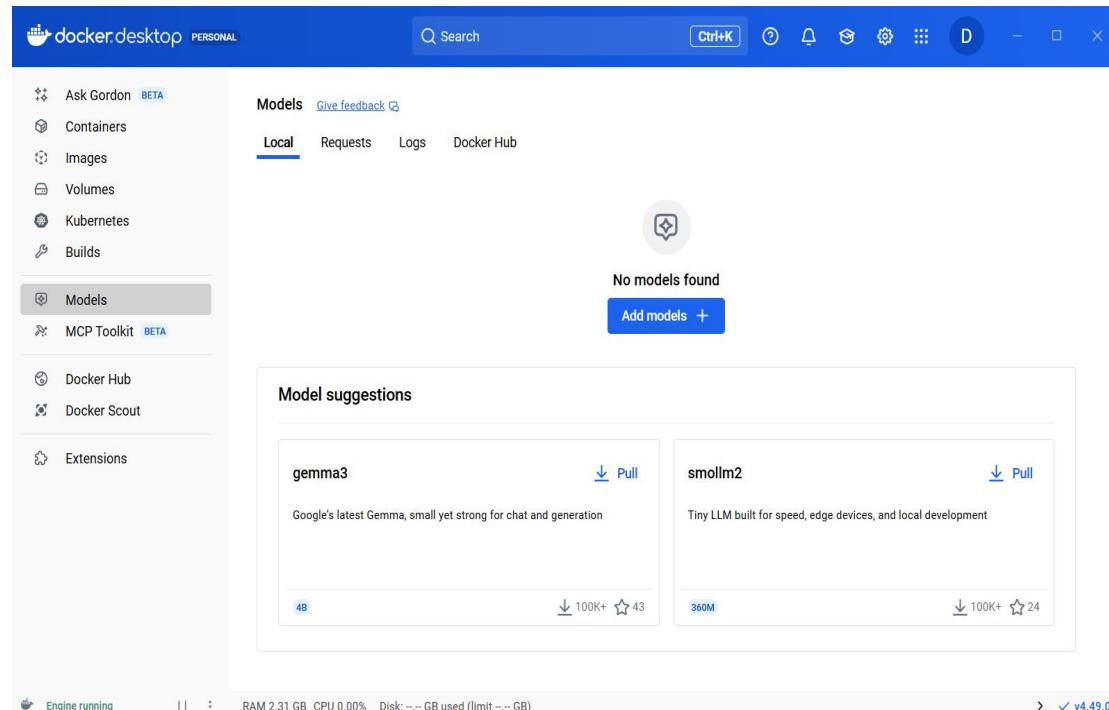
```

3.2. Cài đặt Docker

- Truy cập link: <https://www.docker.com/> -> nhấn Download
- Sau khi tải về sẽ hiển thị file:



- Nhấp đúp vào tệp cài đặt Docker Desktop -> chọn Yes khi hộp thoại cấp quyền quản trị xuất hiện.
- Sau khi Docker Desktop khởi động, chọn Accept để đồng ý với các điều khoản sử dụng.
- Tiếp theo, đăng nhập bằng tài khoản Google/ GitHub của bạn hoặc đăng ký tài khoản mới nếu chưa có.
- Giao diện Docker Destop:



3.3. Sử dụng 1 file docker-compose.yml

- Mục đích của phần này là triển khai nhanh hệ thống IoT web server gồm 6 container hoạt động đồng thời bằng Docker Compose, bao gồm:

Tên dịch vụ	Cổng host	Mục đích
MariaDB	3306	Cơ sở dữ liệu chính
phpMyAdmin	8080	Giao diện web quản lý MariaDB
Node-RED	1880	Nền tảng IoT Flow
InfluxDB	8086	Cơ sở dữ liệu time-series lưu trữ dữ liệu cảm biến
Grafana	3000	Giao diện phân tích, hiển thị dữ liệu InfluxDB
Nginx	80, 443	Máy chủ web front-end (reverse proxy)

- Tất cả được cài đặt thông qua một file docker-compose.yml duy nhất, giúp dễ dàng quản lý và triển khai.

3.3.2. Cấu trúc thư mục dự án

- Tạo cấu trúc thư mục như sau:

```

iot_docker/
├── docker-compose.yml
├── nginx/
│   └── nginx.conf
├── mariadb_data/
├── nodered_data/
├── influxdb_data/
└── grafana_data/
└── README.md

```

- Các thư mục *_data được mount vào container để lưu dữ liệu bền vững (persistent data).

3.3.3. Các bước thực hiện chi tiết

- ❖ Bước 1: Tạo thư mục dự án và di chuyển vào

```
mkdir ~/iot_docker
```

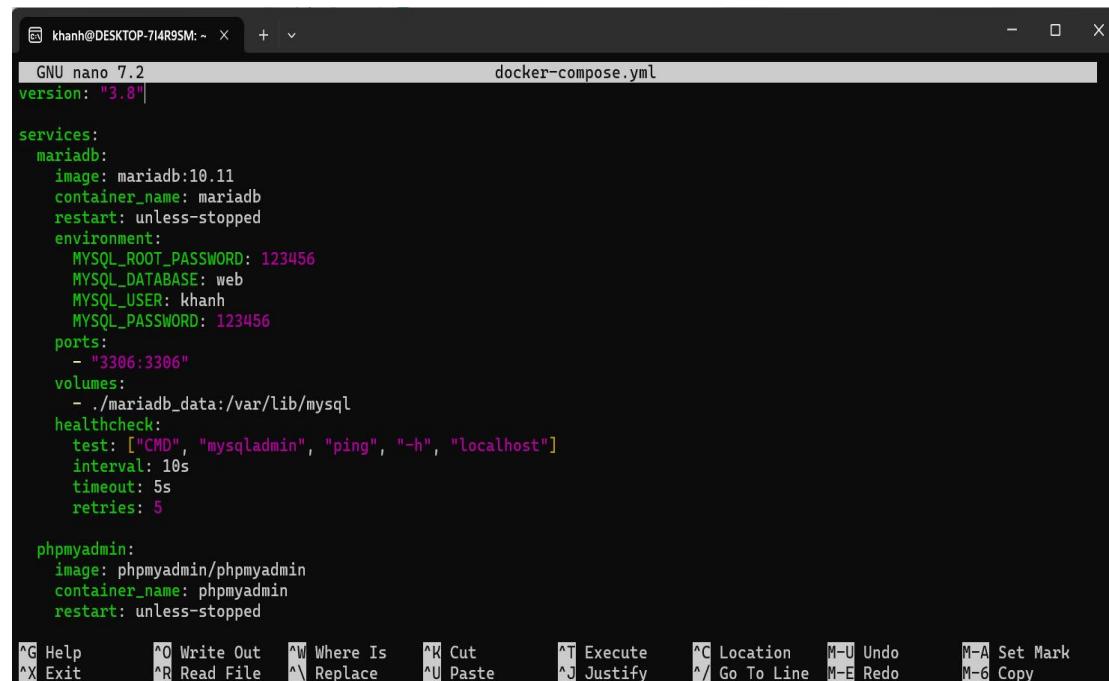
```
cd ~/iot_docker
```

- ❖ Bước 2: Tạo file docker-compose.yml

- Mở Ubuntu.

- Nhập lệnh: nano docker-compose.yml để tạo file.

- Viết code:



```

GNU nano 7.2                               docker-compose.yml
version: "3.8"

services:
  mariadb:
    image: mariadb:10.11
    container_name: mariadb
    restart: unless-stopped
    environment:
      MYSQL_ROOT_PASSWORD: 123456
      MYSQL_DATABASE: web
      MYSQL_USER: khanh
      MYSQL_PASSWORD: 123456
    ports:
      - "3306:3306"
    volumes:
      - ./mariadb_data:/var/lib/mysql
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
      interval: 10s
      timeout: 5s
      retries: 5

  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    container_name: phpmyadmin
    restart: unless-stopped

```

The screenshot shows a terminal window titled 'khanh@DESKTOP-7I4R9SM: ~'. Inside the terminal, the command 'nano docker-compose.yml' has been run, creating a new file named 'docker-compose.yml'. The file contains the configuration for a MySQL database service ('mariadb') and a PHPMyAdmin interface ('phpmyadmin'). The MySQL service uses version 10.11, maps port 3306, and mounts a local directory 'mariadb_data' to its data volume. The PHPMyAdmin service uses the official Docker image and also maps port 3306. The terminal includes standard nano keybindings at the bottom.

-> Nhấn Ctrl + O -> Enter để lưu lại, Ctrl + X để thoát ra

- ❖ Bước 3: Tạo các thư mục dữ liệu: mkdir -p mariadb_data nodered_data influxdb_data grafana_data nginx

❖ Bước 4: Tạo file cấu hình nginx.conf

- Chạy lệnh: nano nginx/nginx.conf để tạo file.

- Chạy code:

```
events { }

http {
    server {
        listen 80;

        location / {
            root /usr/share/nginx/html;
            index index.html;

        }

        location /grafana/ {
            proxy_pass http://grafana:3000/;

        }

        location /nodered/ {
            proxy_pass http://nodered:1880/;

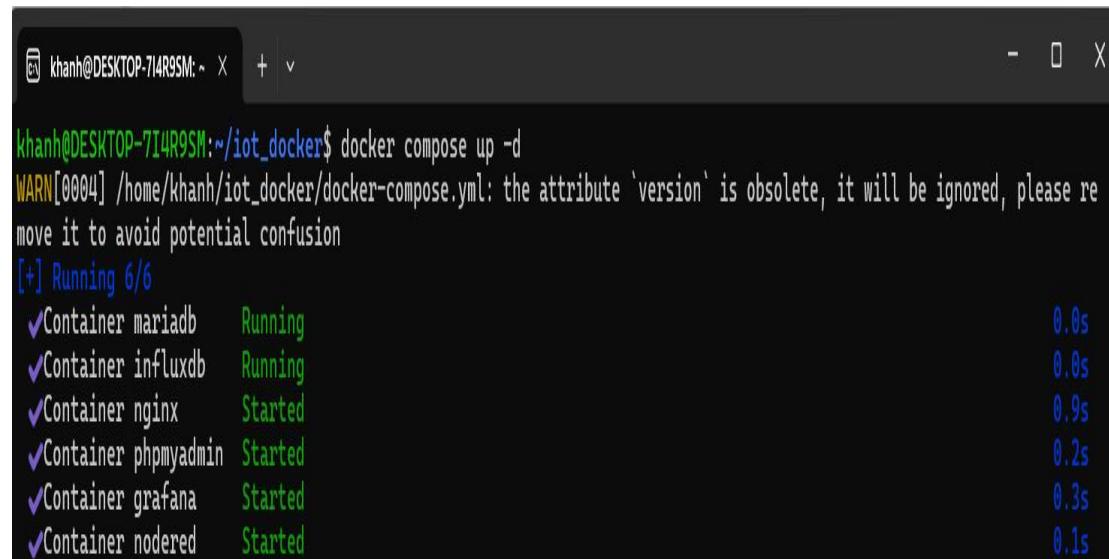
        }

    }
}
```

-> Nhấn Ctrl + O -> Enter để lưu lại, Ctrl + X để thoát ra

❖ Bước 5: Khởi động tất cả container

- Chạy lệnh: docker compose up -d để khởi động



```
khanh@DESKTOP-7I4R9SM:~/iot_docker$ docker compose up -d
WARN[0004] /home/khanh/iot_docker/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 6/6
  ✓ Container mariadb    Running          0.0s
  ✓ Container influxdb   Running          0.0s
  ✓ Container nginx      Started          0.9s
  ✓ Container phpmyadmin Started          0.2s
  ✓ Container grafana    Started          0.3s
  ✓ Container nodered   Started          0.1s
```

❖ **Bước 6: Kiểm tra trạng thái**

- Chạy lệnh: docker ps để kiểm tra trạng thái

```
Khanh@DESKTOP-7I4R9SM:~/iot_docker$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
u77fa1e3f5c1 nodered/node-red "/entrypoint.sh" 31 minutes ago Up 31 minutes (healthy) 0.0.0.0:1880->1880/tcp, [::]:1880->1880/tcp
f23629b55fe1 nginx "/docker-entrypoint..." 55 minutes ago Up 55 minutes 0.0.0.0:8088->80/tcp, [::]:8088->80/tcp, 0.0.0.0:4443->4443/tcp
32f2fc768381 grafana/grafana "/run.sh" 59 minutes ago Up 42 minutes 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp
26239e5259de phpmyadmin/phpmyadmin "/docker-entrypoint..." 59 minutes ago Up 55 minutes 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
3c5db8699778 mariadb:10.11 mariadb "/docker-entrypoint.s..." 59 minutes ago Up 59 minutes 0.0.0.0:3306->3306/tcp, [::]:3306->3306/tcp
3dffadfc423a8 influxdb:2.7 influxdb "/entrypoint.sh infl..." 59 minutes ago Up 59 minutes 0.0.0.0:8086->8086/tcp, [::]:8086->8086/tcp
khanh@DESKTOP-7I4R9SM:~/iot_docker$
```

3.3.4. Kiểm tra kết quả

Node-RED: <http://localhost:1880>

Grafana: <http://localhost:3000>

phpMyAdmin: <http://localhost:8080>

InfluxDB: <http://localhost:8086>

The screenshot shows the Docker Desktop interface with the following details:

- Containers** tab selected.
- Container CPU usage:** 3.00% / 1600% (16 CPUs available).
- Container memory usage:** 507.88MB / 7.43GB.
- Search bar:** Search for "bt3_web".
- Filter:** Only show running containers.
- Table of running containers:**

	Name	Container ID	Image	Port(s)	CPU (%)	Actions
	bt3_web	-	-	-	3%	[Edit] [Stop] [More]
	nodered	43c212f44381	nodered/nodered	1880:1880	0%	[Edit] [Stop] [More]
	mariadb	133c27e447b4	mariadb:10	3306:3306	0.02%	[Edit] [Stop] [More]
	influxdb	07ab047004e	influxdb:2.7	8086:8086	2.61%	[Edit] [Stop] [More]
	grafana	f8b049a045f5	grafana/grafana	3000:3000	0.37%	[Edit] [Stop] [More]
	phpmyadmin	236d14ab1434	phpmyadmin:10.11	8080:80	0%	[Edit] [Stop] [More]
	nginx	a46e9604c4dc	nginx:latest	8088:80	0%	[Edit] [Stop] [More]
- Footer:** Showing 7 items, Engine running, RAM 2.35 GB, CPU 0.00%, Disk: 5.40 GB used (limit 1006.85 GB), v4.49.0.

3.4. Lập trình web frontend+backend:

3.4.1. Tạo CSDL gồm DB: web, các bảng: Categories, Orders, Orders_Items, Products, Users

- Mở phpMyAdmin <http://localhost:8080> và tạo Database: web

The screenshot shows the phpMyAdmin interface for the 'web' database. The left sidebar lists databases: information_schema, mysql, performance_schema, sys, and web. Under 'New' in the 'web' database, there are tables: Categories, Orders, Order_Items, Products, and Users. The 'Categories' table is currently selected, indicated by a blue border. The main area displays the table structure with columns: category_id and name. Data rows include: banchay (Sản Phẩm Bán Chạy), binhnl (Bình Nóng Lạnh), dieuhhoa (Điều Hòa), khac (Sản Phẩm Khác), maygiat (Máy Giặt), nangluongmt (Năng Lượng Mát Trời), tulanh (Tủ Lạnh), and tv (Tivi). Below the table, there are buttons for Check all, With selected:, and a dropdown menu.

- Bảng Categories:

The screenshot shows the phpMyAdmin interface for the 'web' database, specifically the 'Categories' table. The left sidebar shows the same structure as the previous screenshot. The main area displays the results of the query: SELECT * FROM `Categories`. The results are shown in a table with columns: category_id and name. The data is identical to the one in the previous screenshot. At the bottom, there are buttons for Check all, With selected:, Edit, Copy, Delete, and Export.

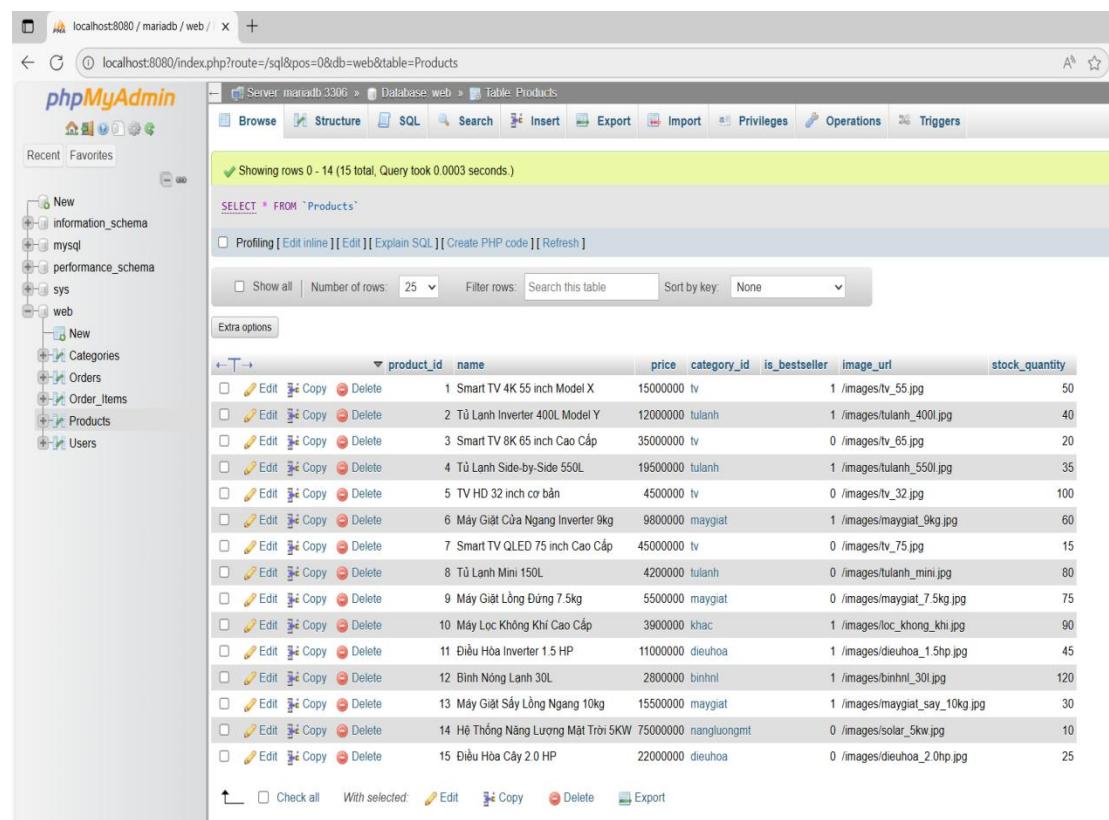
- Bảng Orders:

order_id	user_id	customer_name	customer_phone	customer_address	total_amount	order_date	status	tracking_code
DH-1762385873990	09345243	dv	15000000.00	2025-11-05 23:37:53	pending	NULL		
DH-1762385979093	0962213503	Bắc Ninh	19500000.00	2025-11-05 23:39:39	pending	NULL		
DH-1762386075558	0962213503	Bắc Ninh	30000000.00	2025-11-05 23:41:15	pending	NULL		
DH-1762386610350	09235244	BN	38000000.00	2025-11-05 23:50:10	pending	NULL		
DH-1762388671579	0231343	BN	45000000.00	2025-11-05 23:51:11	pending	NULL		
DH-17623886941560	0231343456	Bắc Ninh	29400000.00	2025-11-05 23:55:41	pending	NULL		
DH-1762411072830	0231343456	bn	15000000.00	2025-11-06 06:37:52	pending	NULL		
DH-1762411704671	0231343456	jj	15000000.00	2025-11-06 06:48:24	pending	NULL		
DH-1762412291568	0231343456	123	35000000.00	2025-11-06 06:58:11	pending	NULL		
DH-1762414700380	0231343	hh	15000000.00	2025-11-06 07:38:20	pending	NULL		
DH-1762415044132	9887	bg	2800000.00	2025-11-06 07:44:04	pending	NULL		
DH-1762415837143	9887	efg	75000000.00	2025-11-06 07:57:17	pending	NULL		
DH-Mẫu-001	0962213503	Đầu Văn Khanh	15000000.00	2025-11-05 23:35:43	delivered	NULL		

- Bảng Orders_Items:

item_id	order_id	product_id	quantity	price_at_order
1	DH-Mẫu-001		1	2 15000000.00
2	DH-Mẫu-001		4	1 19500000.00
3	DH-1762385873990		1	1 15000000.00
4	DH-1762385979093		4	1 19500000.00
5	DH-1762386075558		1	2 15000000.00
6	DH-1762386610350		10	1 3900000.00
7	DH-1762388671579		7	1 45000000.00
8	DH-1762386941560		6	3 9800000.00
9	DH-1762411072830		1	1 15000000.00
10	DH-1762411704671		1	1 15000000.00
11	DH-1762412291568		3	1 35000000.00
12	DH-1762414700380		1	1 15000000.00
13	DH-1762415044132		12	1 2800000.00
14	DH-1762415837143		14	1 7500000.00

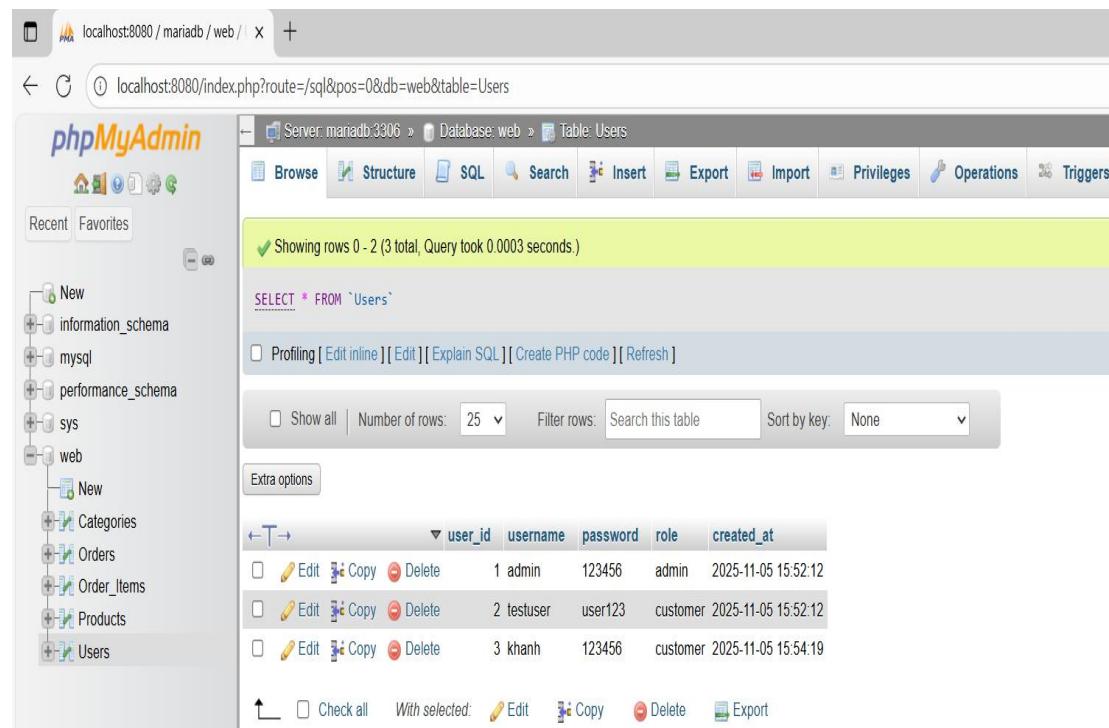
- Bảng Products:



The screenshot shows the phpMyAdmin interface for the 'Products' table in the 'web' database. The table has columns: product_id, name, price, category_id, is_bestseller, image_url, and stock_quantity. The data includes various products like Smart TV, Tủ Lạnh, and May Giặt, each with its specific details and image URL.

	product_id	name	price	category_id	is_bestseller	image_url	stock_quantity
<input type="checkbox"/>	1	Smart TV 4K 55 inch Model X	15000000	tv	1	/images/tv_55.jpg	50
<input type="checkbox"/>	2	Tủ Lạnh Inverter 400L Model Y	12000000	tulanh	1	/images/tulanh_400l.jpg	40
<input type="checkbox"/>	3	Smart TV 8K 65 inch Cao Cấp	35000000	tv	0	/images/tv_65.jpg	20
<input type="checkbox"/>	4	Tủ Lạnh Side-by-Side 550L	19500000	tulanh	1	/images/tulanh_550l.jpg	35
<input type="checkbox"/>	5	TV HD 32 inch cơ bản	4500000	tv	0	/images/tv_32.jpg	100
<input type="checkbox"/>	6	Máy Giặt Cửa Ngang Inverter 9kg	9800000	maygiat	1	/images/maygiat_9kg.jpg	60
<input type="checkbox"/>	7	Smart TV QLED 75 inch Cao Cấp	45000000	tv	0	/images/tv_75.jpg	15
<input type="checkbox"/>	8	Tủ Lạnh Mini 150L	4200000	tulanh	0	/images/tulanh_mini.jpg	80
<input type="checkbox"/>	9	Máy Giặt Lồng Đứng 7.5kg	5500000	maygiat	0	/images/maygiat_7_5kg.jpg	75
<input type="checkbox"/>	10	Máy Lọc Không Khí Cao Cấp	3900000	khac	1	/images/moc_khong_khi.jpg	90
<input type="checkbox"/>	11	Điều Hòa Inverter 1.5 HP	11000000	dieuhoa	1	/images/dieuhoa_1.5hp.jpg	45
<input type="checkbox"/>	12	Bình Nóng Lạnh 30L	2800000	binhnl	1	/images/binhnl_30l.jpg	120
<input type="checkbox"/>	13	Máy Giặt Sấy Lồng Ngang 10kg	15500000	maygiat	1	/images/maygiat_say_10kg.jpg	30
<input type="checkbox"/>	14	Hệ Thống Năng Lượng Mát Trời 5KW	75000000	nangluongmt	0	/images/solar_5kw.jpg	10
<input type="checkbox"/>	15	Điều Hòa Cây 2.0 HP	22000000	dieuhoa	0	/images/dieuhoa_2.0hp.jpg	25

- Bảng Users:

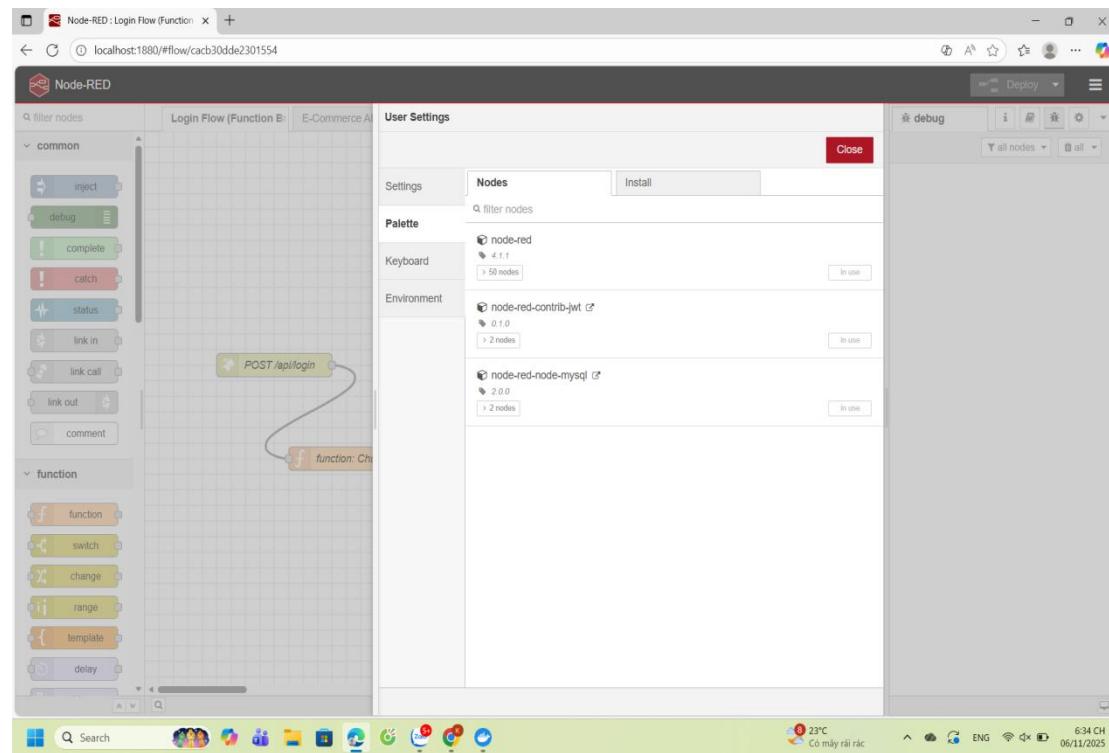


The screenshot shows the phpMyAdmin interface for the 'Users' table in the 'web' database. The table has columns: user_id, username, password, role, and created_at. The data includes users like admin, testuser, and khanh, each with their specific details and creation timestamp.

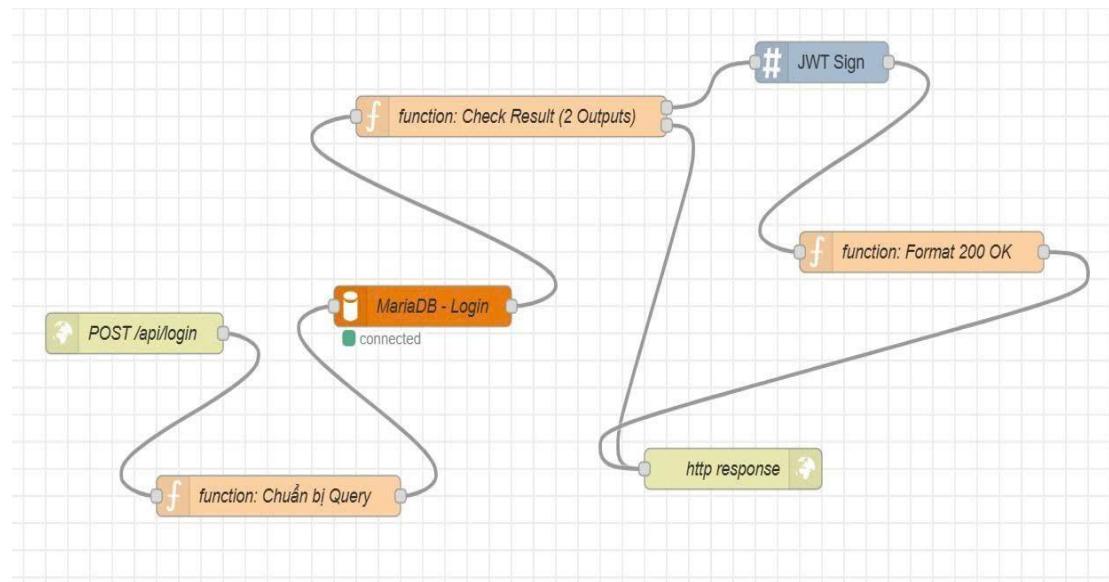
	user_id	username	password	role	created_at
<input type="checkbox"/>	1	admin	123456	admin	2025-11-05 15:52:12
<input type="checkbox"/>	2	testuser	user123	customer	2025-11-05 15:52:12
<input type="checkbox"/>	3	khanh	123456	customer	2025-11-05 15:54:19

3.4.2. Tạo Node-RED backend — import flow (REST API)

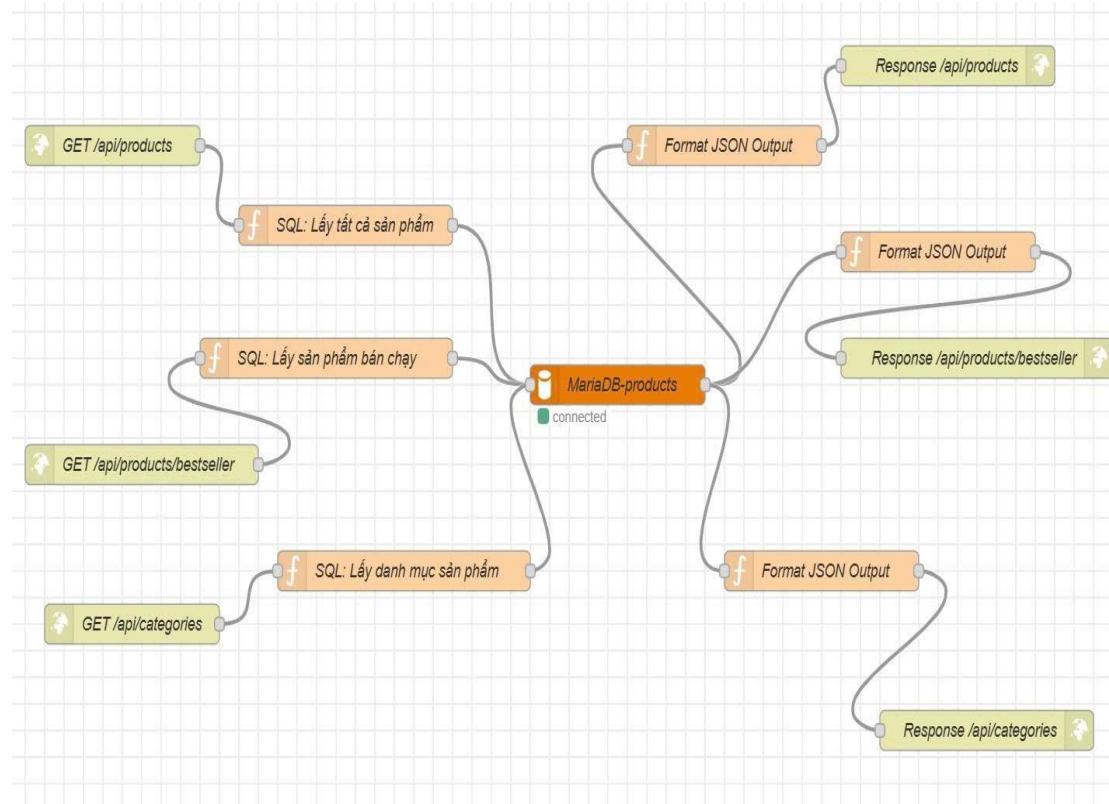
- Mở Node-RED <http://localhost:1880>. Vào Menu → Manage palette install để cài 1 số thư viện:



- Tạo nodered login:



- Tạo nodered liệt kê sản phẩm, các sản phẩm bán chạy nhất, các nhóm sản phẩm



- Kiểm tra API:

+ Liệt kê sản phẩm:

```

{
  "success": true,
  "categories": [
    {
      "id": "binhnl",
      "name": "Bình Nóng Lạnh"
    },
    {
      "id": "maygiat",
      "name": "Máy Giặt"
    },
    {
      "id": "nangluongnt",
      "name": "Năng Lượng Mặt Trời"
    },
    {
      "id": "banchay",
      "name": "Sản Phẩm Bán Chạy"
    },
    {
      "id": "khac",
      "name": "Sản Phẩm Khác"
    },
    {
      "id": "tv",
      "name": "Tivi"
    },
    {
      "id": "tulanh",
      "name": "Tủ Lạnh"
    },
    {
      "id": "dieuhoa",
      "name": "Điều Hòa"
    }
  ]
}
  
```

+ Các sản phẩm bán chạy nhất:

```

{
  "success": true,
  "products": [
    {
      "id": 1,
      "name": "Smart TV 4K 55 inch Model X",
      "price": 15000000,
      "category": "tv",
      "isBestseller": 1,
      "image_url": "/images/tv_55.jpg",
      "stock_quantity": 50
    },
    {
      "id": 2,
      "name": "Tủ Lạnh Inverter 400L Model Y",
      "price": 12000000,
      "category": "tulanh",
      "isBestseller": 1,
      "image_url": "/images/tulanh_400l.jpg",
      "stock_quantity": 40
    },
    {
      "id": 4,
      "name": "Tủ Lạnh Side-by-Side 550L",
      "price": 18000000,
      "category": "tulanh",
      "isBestseller": 1,
      "image_url": "/images/tulanh_550l.jpg",
      "stock_quantity": 35
    },
    {
      "id": 6,
      "name": "Máy Giặt Cửa Ngang Inverter 9kg",
      "price": 9800000,
      "category": "maygiat",
      "isBestseller": 1,
      "image_url": "/images/maygiat_9kg.jpg",
      "stock_quantity": 60
    },
    {
      "id": 10,
      "name": "Máy Lọc Không Khí Cao Cấp",
      "price": 3900000,
      "category": "khac",
      "isBestseller": 1,
      "image_url": "/images/loc_khong_khi.jpg",
      "stock_quantity": 90
    },
    {
      "id": 11,
      "name": "Điều Hòa Inverter 1.5 HP",
      "price": 11000000,
      "category": "dieuhoa"
    }
  ]
}

```

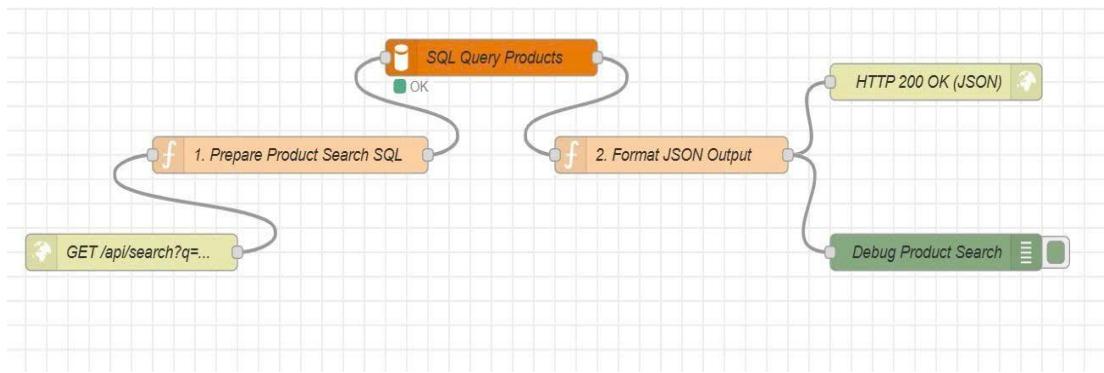
+ Các nhóm sản phẩm:

```

{
  "success": true,
  "categories": [
    {
      "id": "binhnl",
      "name": "Bình Nóng Lạnh"
    },
    {
      "id": "maygiat",
      "name": "Máy Giặt"
    },
    {
      "id": "nangluongt",
      "name": "Năng Lượng Mặt Trời"
    },
    {
      "id": "barichay",
      "name": "Sản Phẩm Bán Chạy"
    },
    {
      "id": "khac",
      "name": "Sản Phẩm Khác"
    },
    {
      "id": "tv",
      "name": "Tivi"
    },
    {
      "id": "tulanh",
      "name": "Tủ Lạnh"
    },
    {
      "id": "dieuhoa",
      "name": "Điều Hòa"
    }
  ]
}

```

- Tạo node red tìm kiếm sản phẩm:

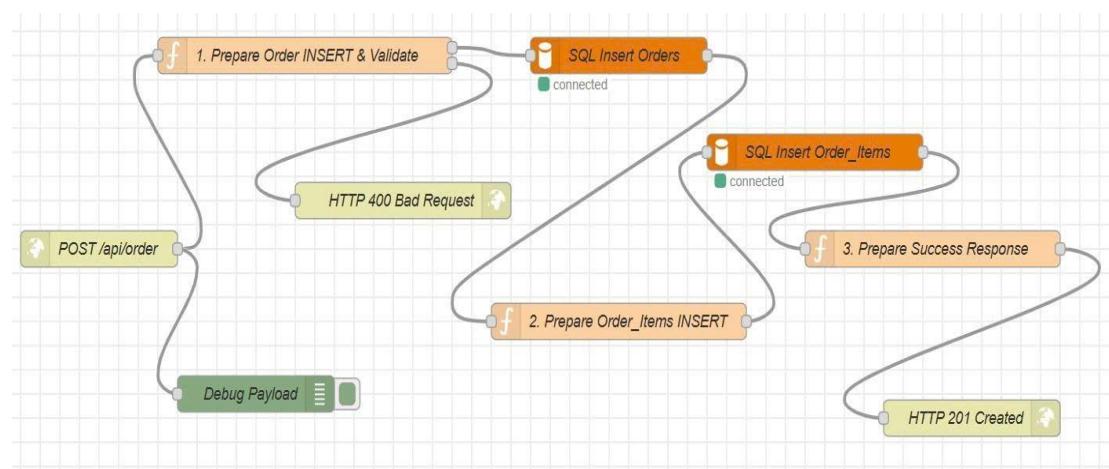


- Kiểm tra API:

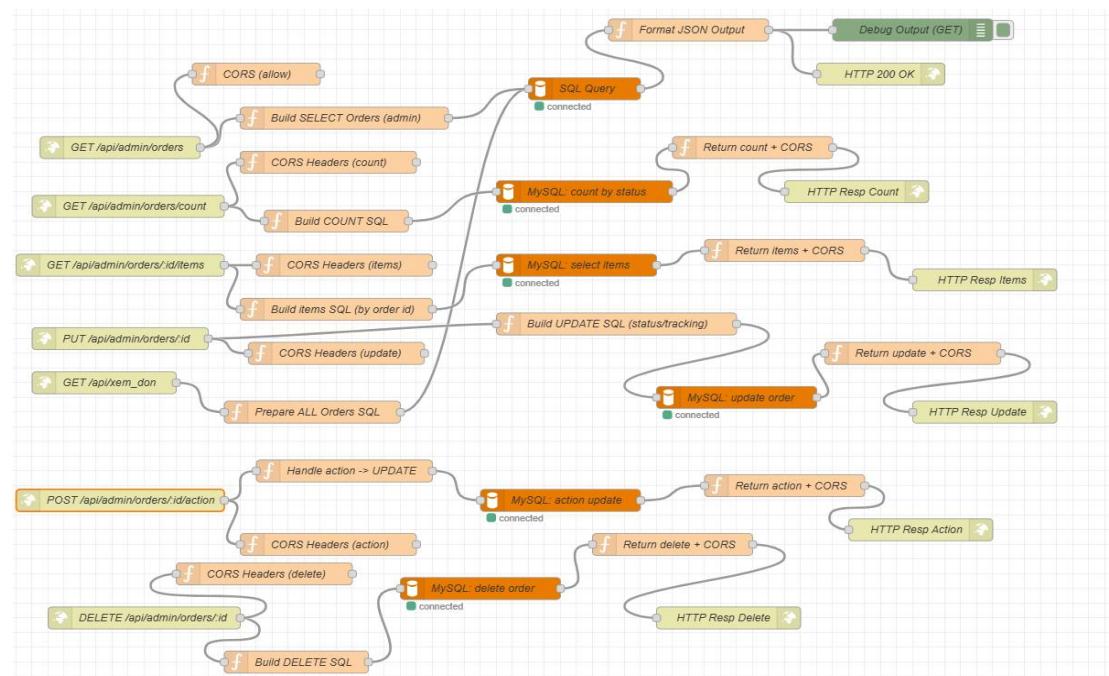
```

[{"order_id": "DH-1762438211515", "user_id": 1, "customer_name": "tu", "customer_phone": "093", "customer_address": "vèe", "total_amount": "12000000", "total": 12000000, "order_date": "2025-11-06T14:10:11.000Z", "status": "pending", "tracking_code": "", "items": []}, {"order_id": "DH-1762438101289", "user_id": 1, "customer_name": "tú", "customer_phone": "0993", "customer_address": "bn", "total_amount": "7800000", "total": 7800000, "order_date": "2025-11-06T14:08:21.000Z", "status": "pending", "tracking_code": "", "items": []}, {"order_id": "DH-1762435037251", "user_id": 1, "customer_name": "Đậu Văn Khánh", "customer_phone": "098756452", "customer_address": "Hiệp Hòa, BN", "total_amount": "60000000", "total": 60000000, "order_date": "2025-11-06T13:17:17.000Z", "status": "pending", "tracking_code": "", "items": []}, {"order_id": "DH-1762415837143", "user_id": 1, "customer_name": "Đậu Văn Khánh", "customer_phone": "9887", "customer_address": "efg", "total_amount": "75000000", "total": 75000000, "order_date": "2025-11-06T07:57:17.000Z", "status": "pending", "tracking_code": "", "items": []}]
    
```

- Tạo nodered đặt hàng:



- Tạo nodered xem đơn hàng:



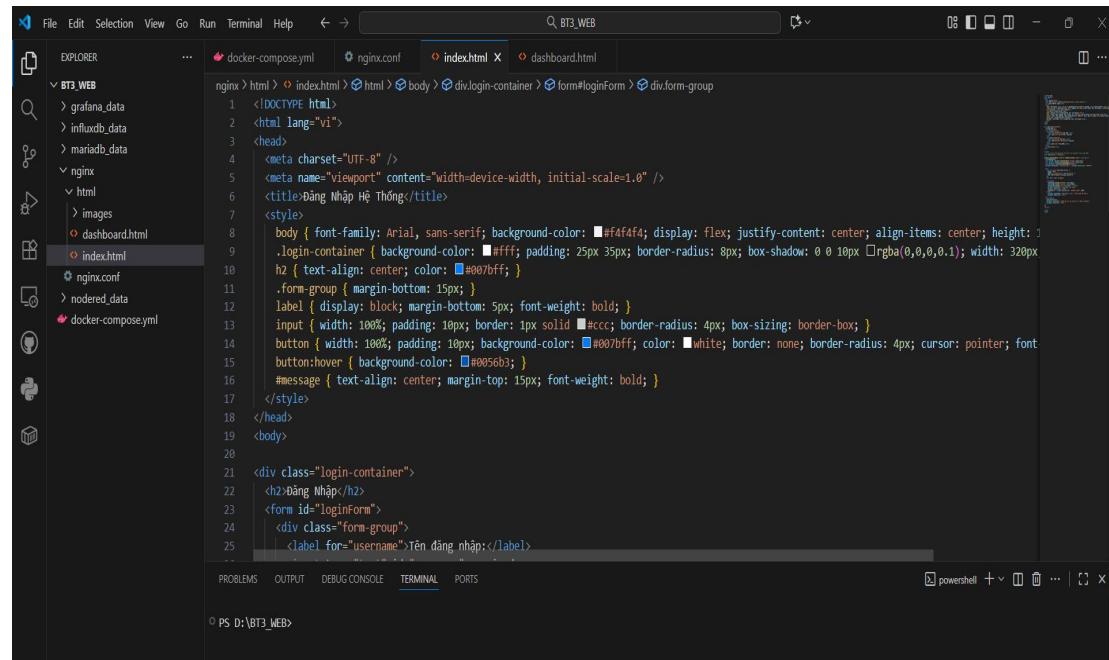
- Kiểm tra API:

```

← ⌂ ⓘ localhost:1880/api/admin/orders
Pretty-print 
[{"order_id": "DH-1762438211515", "user_id": 1, "customer_name": "tú", "customer_phone": "093", "customer_address": "võe", "total_amount": "12000000", "total": 12000000, "order_date": "2025-11-06T14:10:11.000Z", "status": "pending", "tracking_code": "", "items": []}, {"order_id": "DH-1762438101289", "user_id": 1, "customer_name": "tú", "customer_phone": "0993", "customer_address": "bn", "total_amount": "7800000", "total": 7800000, "order_date": "2025-11-06T14:08:21.000Z", "status": "pending", "tracking_code": "", "items": []}, {"order_id": "DH-1762435037251", "user_id": 1, "customer_name": "Đậu Văn Khánh", "customer_phone": "098756452", "customer_address": "Hiệp Hòa, BN", "total_amount": "60000000", "total": 60000000, "order_date": "2025-11-06T13:17:17.000Z", "status": "pending", "tracking_code": "", "items": []}, {"order_id": "DH-1762415837143", "user_id": 1, "customer_name": "Đậu Văn Khánh", "customer_phone": "9887", "customer_address": "efg", "total_amount": "75000000", "total": 75000000, "order_date": "2025-11-06T07:57:17.000Z", "status": "pending", "tracking_code": "", "items": []}]
  
```

3.4.3. Code html

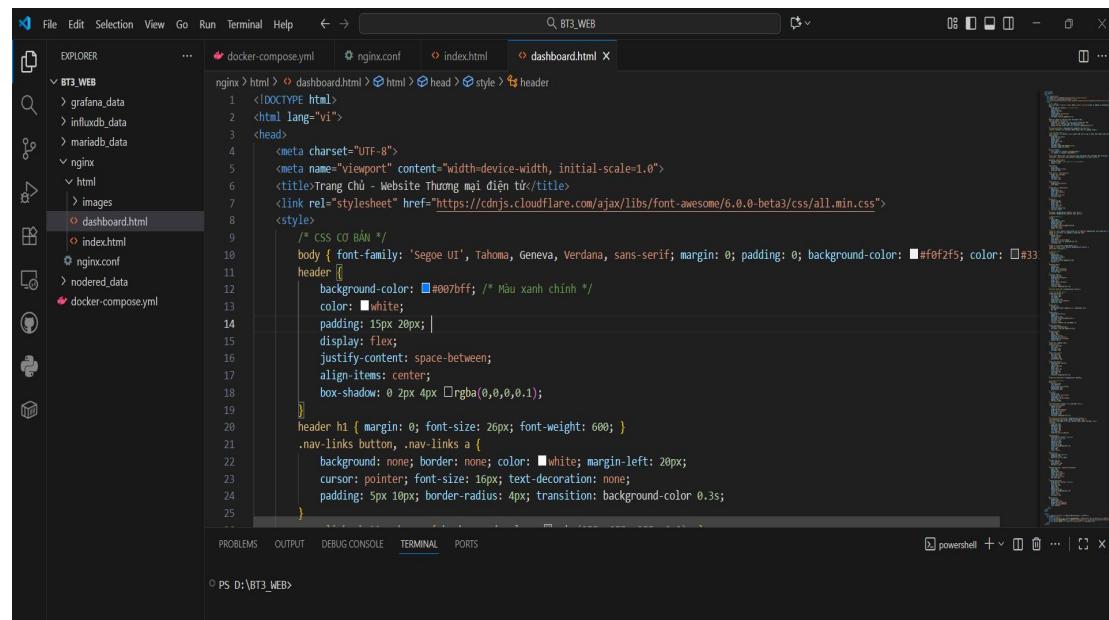
- Code index.html (Giao diện đăng nhập vào hệ thống):



The screenshot shows the VS Code interface with the file 'index.html' open in the center editor pane. The code is an HTML form for user login. It includes a head section with meta tags for charset and viewport, a title 'Đăng Nhập Hệ Thống', and a body section containing a login container with a form and input fields for username and password. The code uses CSS styles for the form elements.

```
<!DOCTYPE html>
<html lang="vi">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Đăng Nhập Hệ Thống</title>
  </head>
  <body>
    <div class="login-container">
      <h2>Đăng Nhập</h2>
      <form id="loginForm">
        <div class="form-group">
          <label for="username">tên đăng nhập:</label>
```

- Code dashboard.html (Giao diện web thương mại điện tử sau khi đăng nhập):

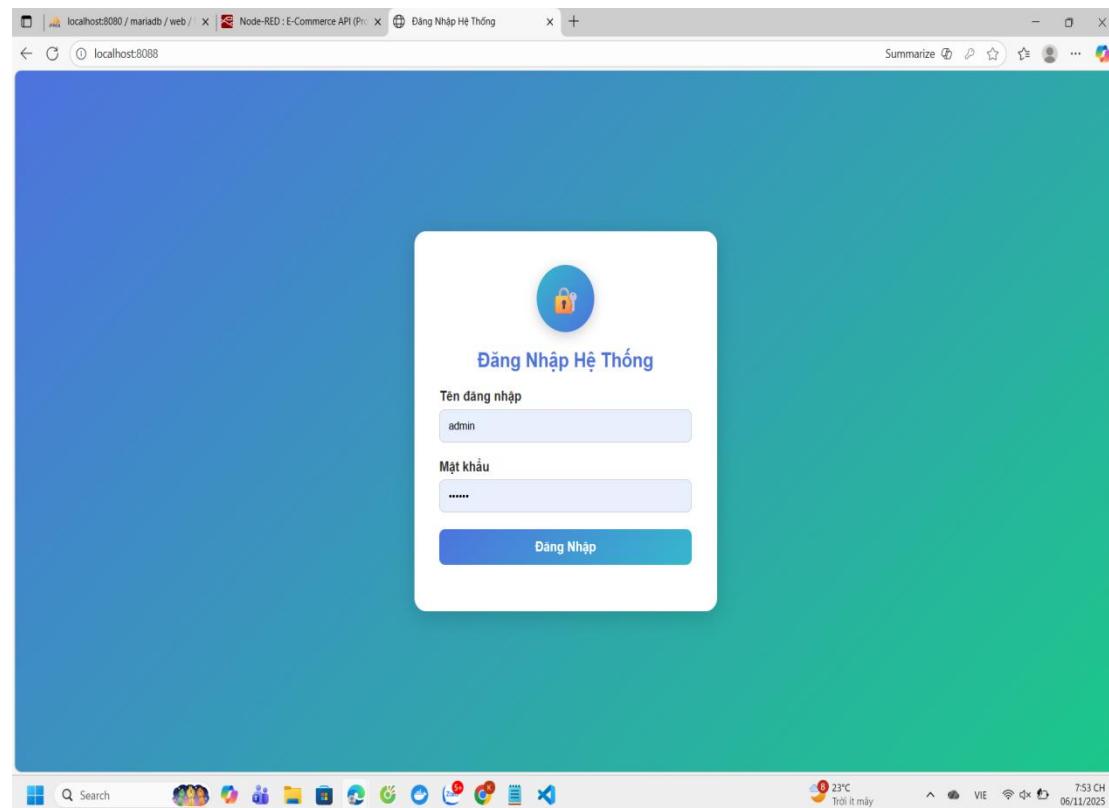


The screenshot shows the VS Code interface with the file 'dashboard.html' open in the center editor pane. The code is an HTML page for a commerce website. It includes a header section with a title 'Trang Chủ - Website Thương mại điện tử' and a link to a CSS file. The main content area contains a header with a background color of #007bff and white text, and a list of navigation links with a background color of #fff and white text.

```
<!DOCTYPE html>
<html lang="vi">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Trang Chủ - Website Thương mại điện tử</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" />
  </head>
  <body>
    <header>
      <div style="background-color: #007bff; /* Màu xanh chính */; color: #fff; padding: 15px 20px; display: flex; justify-content: space-between; align-items: center; box-shadow: 0 2px 4px rgba(0,0,0,0.1);>
```

3.4.4. Giao diện web

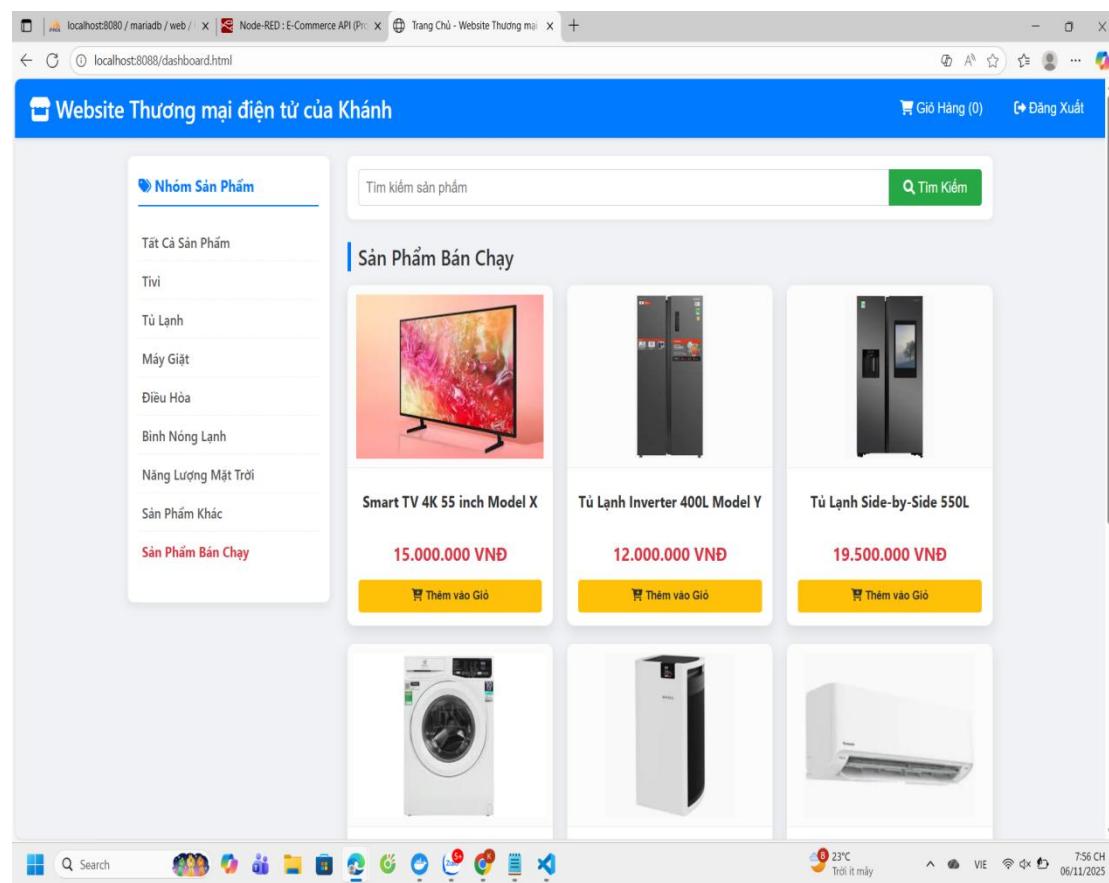
- Giao diện đăng nhập



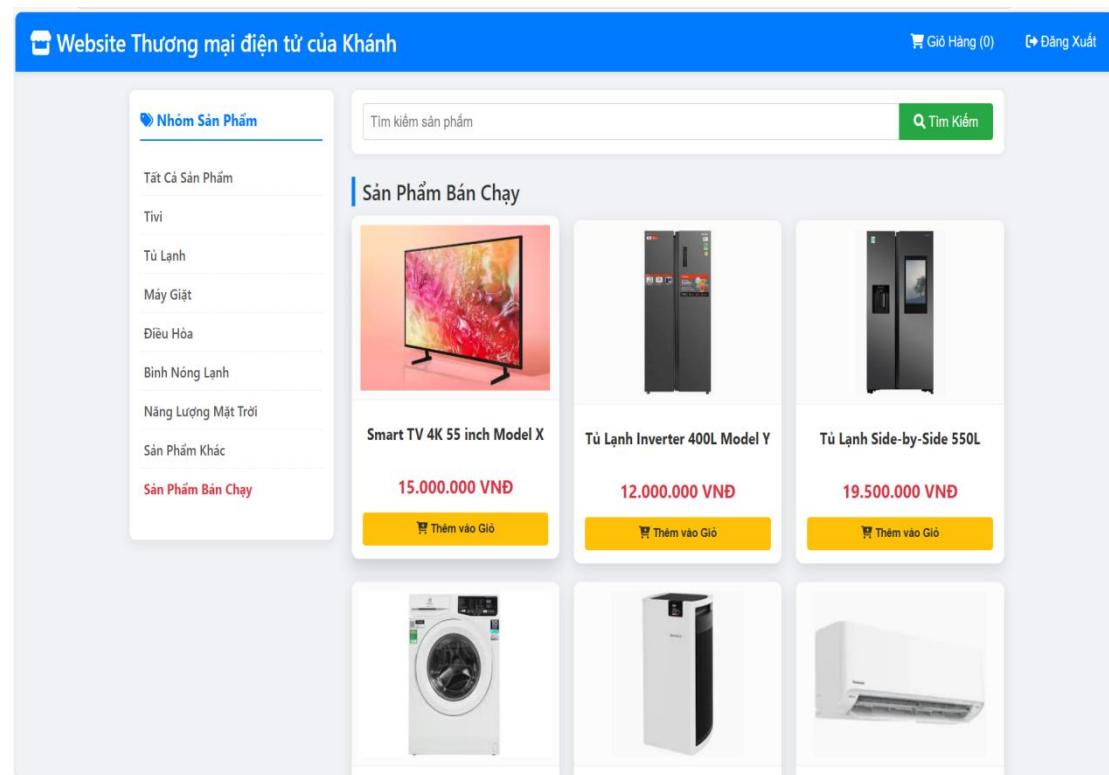
- Giao diện web của Admin

The screenshot shows a web browser window with a blue header bar containing tabs for 'localhost:8080 / mariadb / web' and 'Node-RED : E-Commerce API (P...'. Below the header is a white admin panel interface. At the top right are links for 'Admin Panel', 'Giỏ Hàng (0)', and 'Đăng Xuất'. On the left is a sidebar with a tree view of product categories: 'Nhóm Sản Phẩm' (Tivi, Tủ Lạnh, Máy Giặt, Điều Hòa, Bình Nóng Lạnh, Năng Lượng Mát Trời, Sản Phẩm Khác), and 'Sản Phẩm Bán Chạy'. The main content area features a grid of product cards. One card for a 'Smart TV 4K 55 inch Model X' is highlighted with a red price of '15.000.000 VNĐ'. Other cards include a refrigerator ('Tủ Lạnh Inverter 400L Model Y', 12.000.000 VNĐ), a side-by-side refrigerator ('Tủ Lạnh Side-by-Side 550L', 19.500.000 VNĐ), a washing machine, a dehumidifier, and an air conditioner. Each card has a yellow 'Thêm vào Giỏ' button at the bottom.

- Giao diện web của Customer



- Tính năng liệt kê các sản phẩm bán chạy ra trang chủ



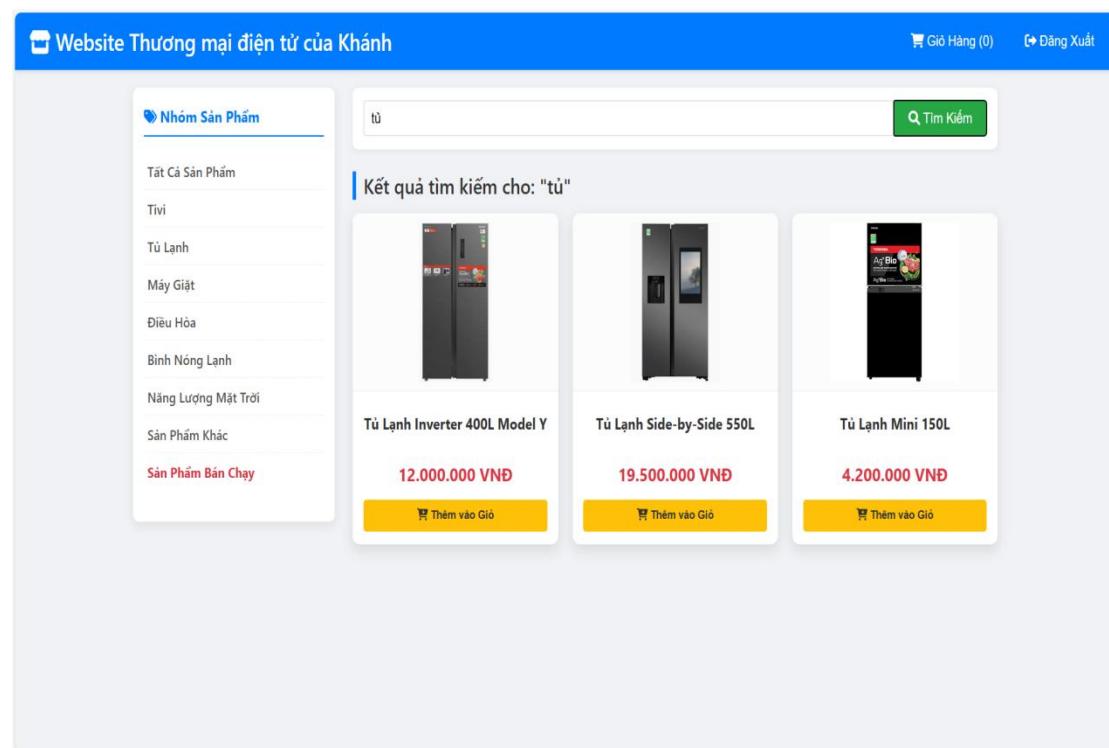
- Tính năng liệt kê các nhóm sản phẩm

The screenshot shows a blue header bar with the text "Website Thương mại điện tử của Khánh". Below it is a sidebar titled "Nhóm Sản Phẩm" containing a list of product categories: Tất Cả Sản Phẩm, Tivi, Tủ Lạnh, Máy Giặt, Điều Hòa, Bình Nóng Lạnh, Năng Lượng Mặt Trời, Sản Phẩm Khác, and Sản Phẩm Bán Chạy. To the right of the sidebar, there is a search bar labeled "Tìm kiếm" and a large image of a smartphone.

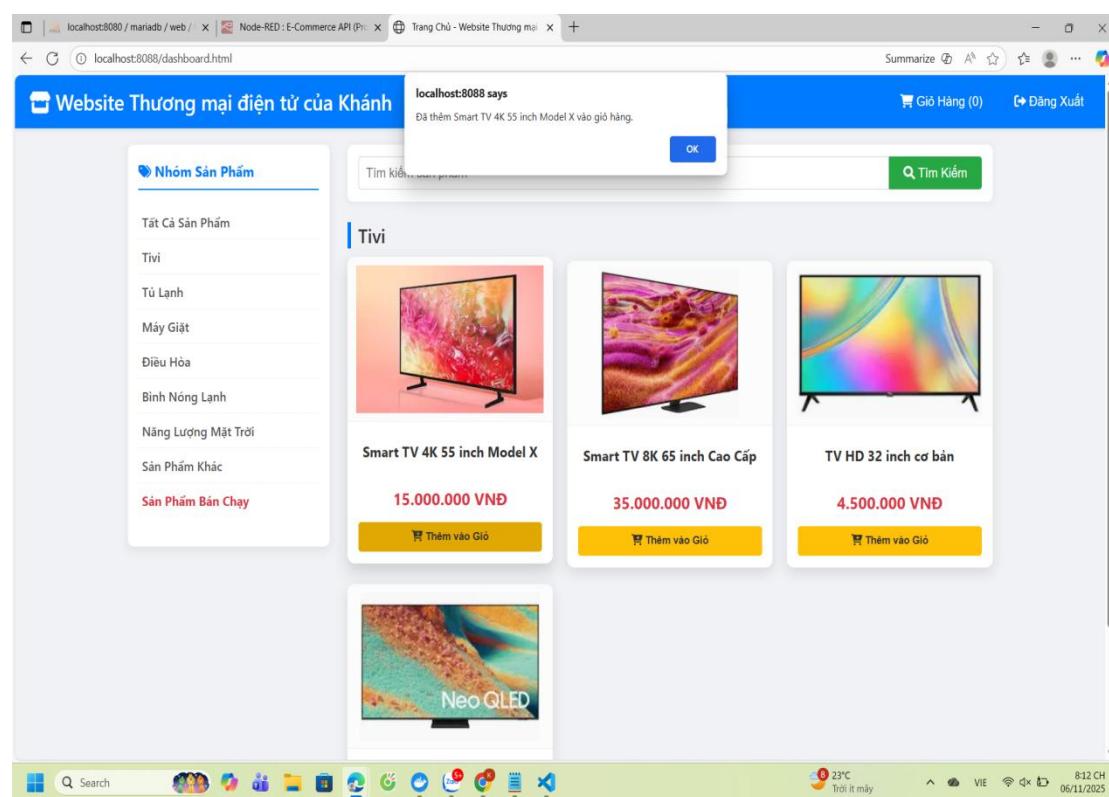
- Tính năng liệt kê sản phẩm theo nhóm

The screenshot shows the same website interface, but the sidebar has "Tủ Lạnh" selected. The main content area displays three refrigerators with their names and prices: Tủ Lạnh Inverter 400L Model Y (12.000.000 VNĐ), Tủ Lạnh Side-by-Side 550L (19.500.000 VNĐ), and Tủ Lạnh Mini 150L (4.200.000 VNĐ). Each item has a yellow "Thêm vào Giỏ" button below it.

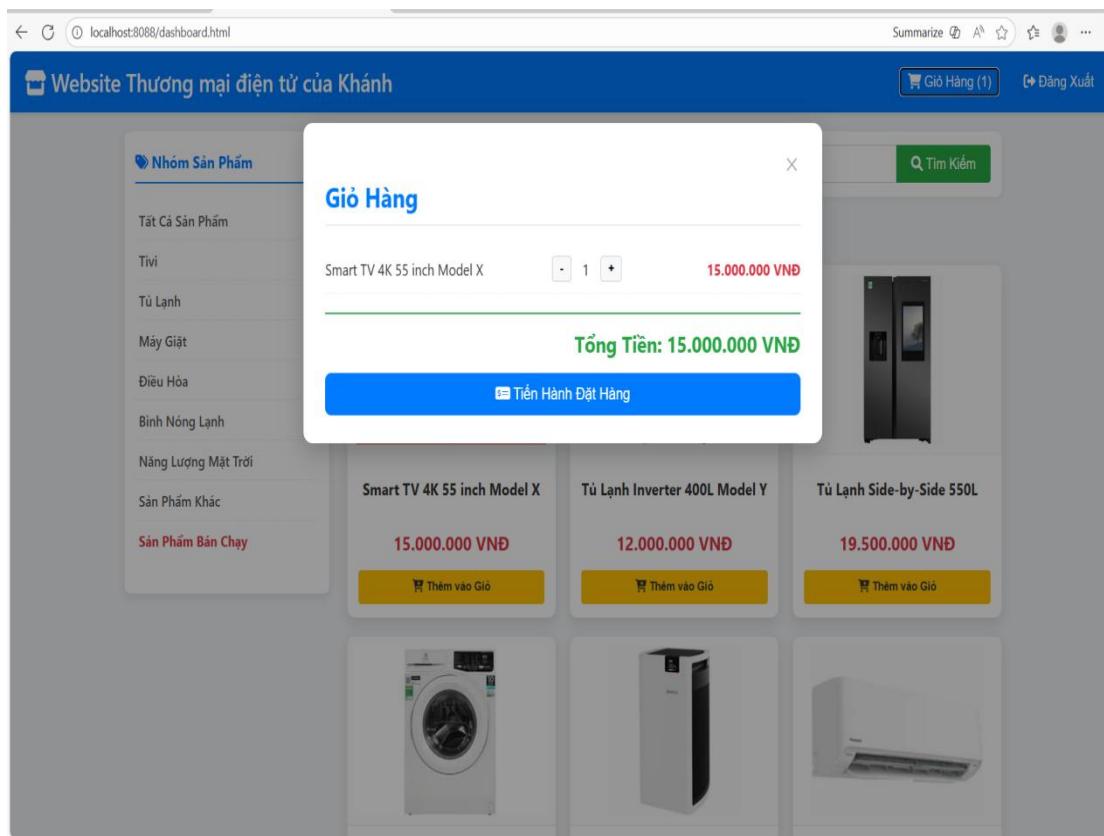
- Tính năng tìm kiếm sản phẩm



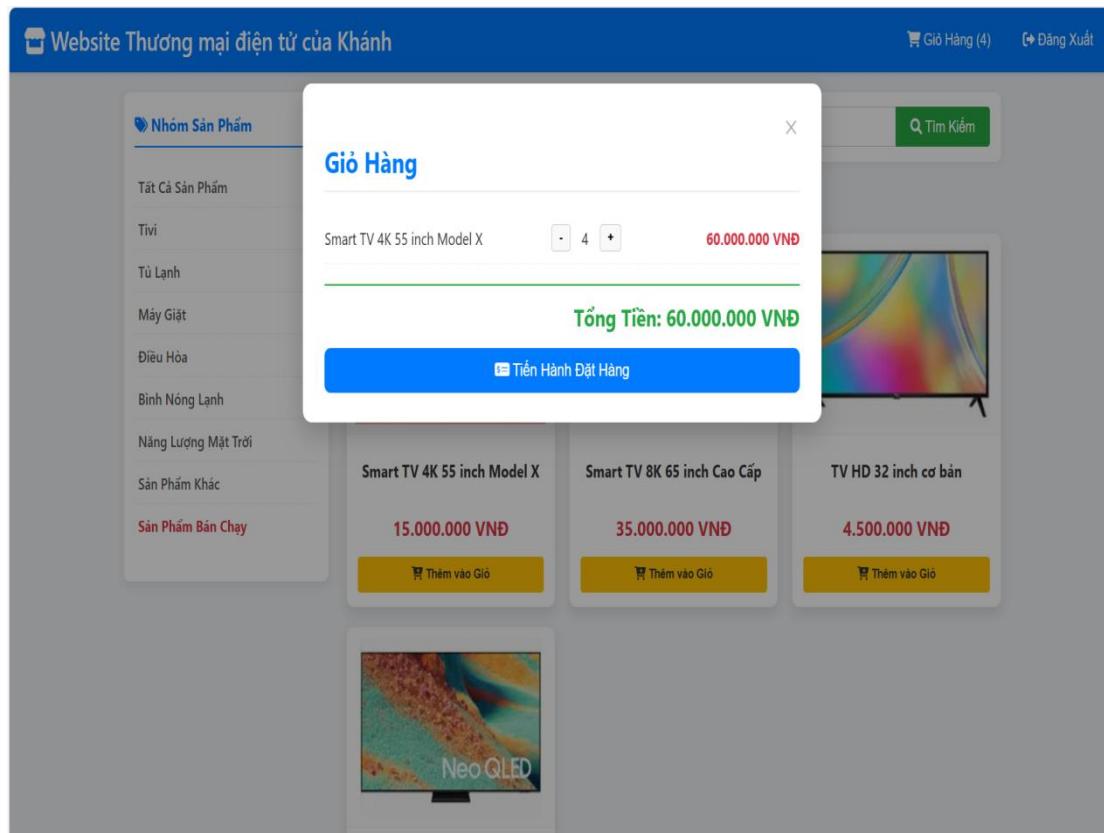
- Tính năng chọn sản phẩm (đưa sản phẩm vào giỏ hàng, thay đổi số lượng sản phẩm trong giỏ, cập nhật tổng tiền)
 - + Thêm sản phẩm vào giỏ hàng



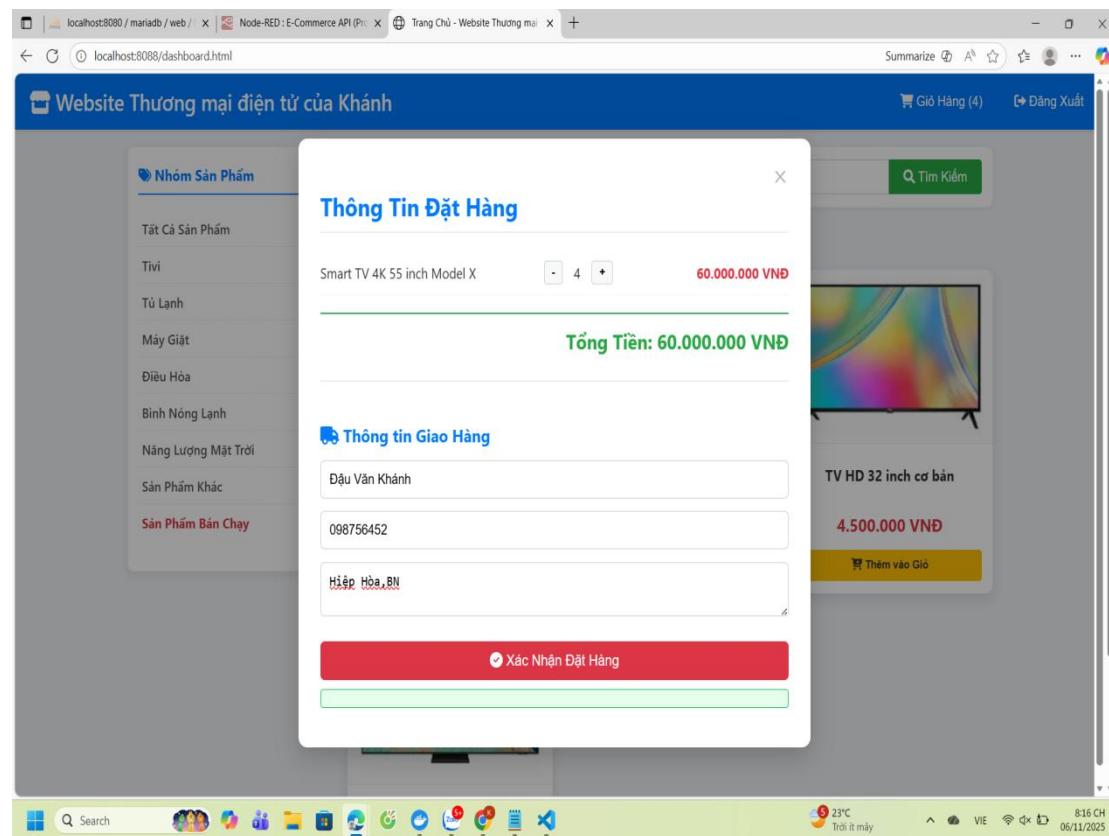
+ Thêm vào giỏ hàng thành công



+ Thay đổi số lượng sản phẩm trong giỏ hàng và cập nhật tổng tiền



+ Tính năng đặt hàng, nhập thông tin giao hàng



-> Sau khi đặt hàng, dữ liệu sẽ tự động hiển thị trong sql

The screenshot shows a browser window with the phpMyAdmin interface. The left sidebar shows the database structure with 'New', 'information_schema', 'mysql', 'performance_schema', 'sys', and 'web' databases. Under 'web', there are 'Categories', 'Orders', 'Order_Items', 'Products', and 'Users'. The main area is titled 'Table: Orders' and shows a list of rows from the 'Orders' table. The columns include 'order_id', 'user_id', 'customer_name', 'customer_phone', 'customer_address', 'total_amount', 'order_date', 'status', and 'tracking_code'. The data listed corresponds to the order placed in the previous screenshot, such as order ID DH-1762385873900 for customer Đậu Văn Khánh.

order_id	user_id	customer_name	customer_phone	customer_address	total_amount	order_date	status	tracking_code
DH-1762385873900	1	Đậu Văn Khánh	09345243	dv	15000000.00	2025-11-05 23:37:53	pending	NULL
DH-1762385979093	1	Đậu Văn Khánh	0962213503	Bắc Ninh	19500000.00	2025-11-05 23:39:39	pending	NULL
DH-1762386075558	1	Đậu Văn Khánh	0962213503	Bắc Ninh	30000000.00	2025-11-05 23:41:15	pending	NULL
DH-1762386610350	1	Khánh	09235244	BN	3900000.00	2025-11-05 23:50:10	pending	NULL
DH-1762386671579	1	Khánh	0231343	BN	45000000.00	2025-11-05 23:51:11	pending	NULL
DH-1762386945160	1	Khánh	0231343456	Bắc Ninh	29400000.00	2025-11-05 23:55:41	pending	NULL
DH-1762411072830	1	Khánh	0231343456	bn	15000000.00	2025-11-06 06:37:52	pending	NULL
DH-17624110740671	1	Khánh	0231343456	jí	15000000.00	2025-11-06 06:48:24	pending	NULL
DH-1762412201568	1	Khánh	0231343456	123	35000000.00	2025-11-06 06:58:11	pending	NULL
DH-1762414700380	1	Đậu Văn Khánh	0231343	hh	15000000.00	2025-11-06 07:38:20	pending	NULL
DH-1762415044132	1	Đậu Văn Khánh	9887	bg	2800000.00	2025-11-06 07:44:04	pending	NULL
DH-1762415837143	1	Đậu Văn Khánh	9887	efg	75000000.00	2025-11-06 07:57:17	pending	NULL
DH-1762435037251	1	Đậu Văn Khánh	098756452	Hiệp Hòa, BN	60000000.00	2025-11-06 13:17:17	pending	NULL
DH-Mẫu-001	1	Đậu Văn Khánh	0962213503	Hiệp Hòa, Bắc Ninh	15000000.00	2025-11-05 23:35:43	delivered	NULL

- Tính năng dành cho admin: Thống kê xem có bao nhiêu đơn hàng, call để xác nhận và cập nhật thông tin đơn hàng. chuyển cho bộ phận đóng gói, gửi bưu điện, cập nhật mã COD, tình trạng giao hàng, huỷ hàng,...

Quản Lý Đơn Hàng (Admin)

Tổng số đơn hàng: 16

Tải lại danh sách

Mã ĐH	Ngày Đặt	Khách Hàng	SĐT/Địa Chỉ	Tổng Tiền	Trạng Thái	Mã COD/Tracking	Thao Tác
undefined	6/11/2025	tu	093 uée	12000000 VND	Đang chờ xác nhận	01	<input checked="" type="button"/> Lưu <input type="button"/> Xóa
undefined	6/11/2025	tú	0993 bn	7800000 VND	Đã xác nhận	02	<input checked="" type="button"/> Lưu <input type="button"/> Xóa
undefined	6/11/2025	Đậu Văn Khanh	098756452 Hiệp Hòa,BN	6000000 VND	Đang đóng gói	03	<input checked="" type="button"/> Lưu <input type="button"/> Xóa
undefined	6/11/2025	Đậu Văn Khanh	9887 efg	7500000 VND	Đã hủy	04	<input checked="" type="button"/> Lưu <input type="button"/> Xóa
undefined	6/11/2025	Đậu Văn Khanh	9887 bg	2800000 VND	Đang đóng gói	05	<input checked="" type="button"/> Lưu <input type="button"/> Xóa
undefined	6/11/2025	Đậu Văn Khanh	0231343 hh	15000000 VND	Đã gửi bưu điện	06	<input checked="" type="button"/> Lưu <input type="button"/> Xóa
undefined	6/11/2025	Khánh	0231343456 123	35000000 VND	Đã gửi bưu điện	07	<input checked="" type="button"/> Lưu <input type="button"/> Xóa
undefined	6/11/2025	Khánh	0231343456 jj	15000000 VND	Đã giao hàng	08	<input checked="" type="button"/> Lưu <input type="button"/> Xóa
undefined	6/11/2025	Khánh	0231343456 bn	15000000 VND	Đã giao hàng	09	<input checked="" type="button"/> Lưu <input type="button"/> Xóa
undefined	6/11/2025	Khánh	0231343456 Bắc Ninh	29400000 VND	Đã hủy	10	<input checked="" type="button"/> Lưu <input type="button"/> Xóa

Đơn hàng undefined đã được cập nhật thành công!

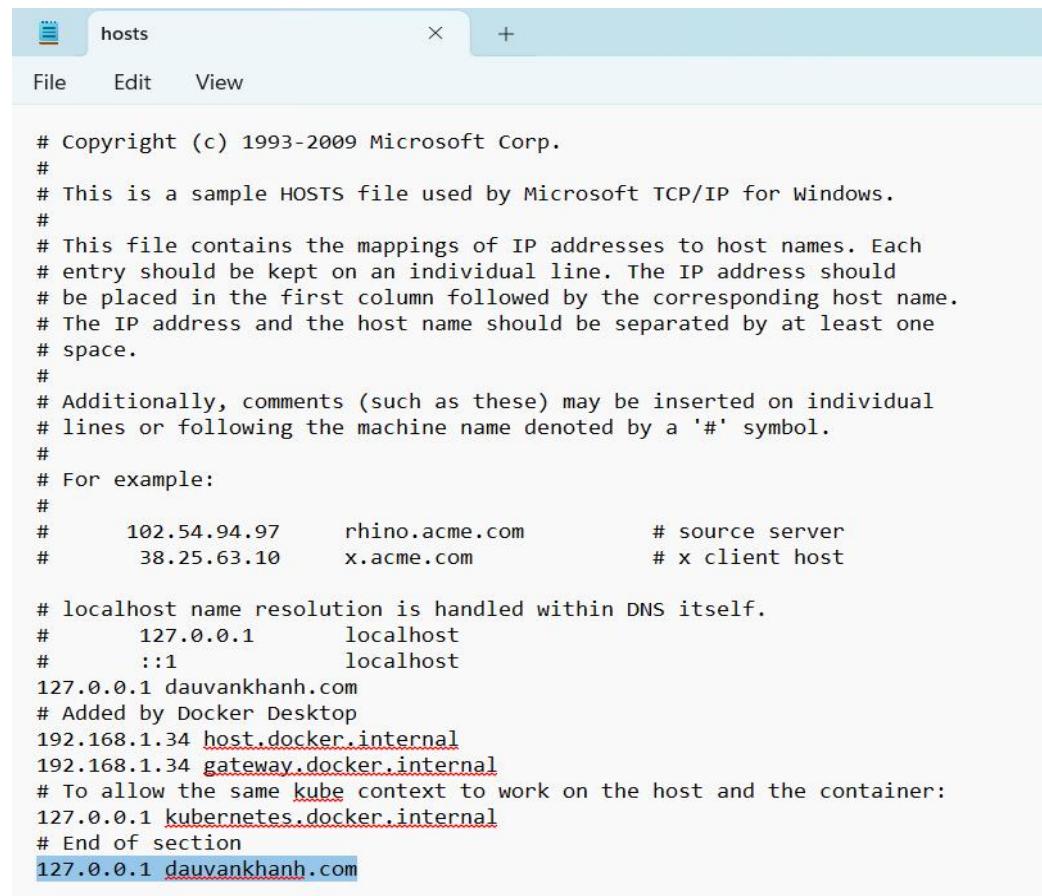
- Tính năng dành cho admin: biểu đồ thống kê số lượng mặt hàng bán được trong từng ngày



3.5. Nginx làm web-server

3.5.1. Cấu hình nginx để chạy được website qua url <http://dauvankhanh.com>

- Truy cập đường dẫn: C:\Windows\System32\drivers\etc\hosts -> thêm dòng 127.0.0.1 dauvankhanh.com



```

hosts

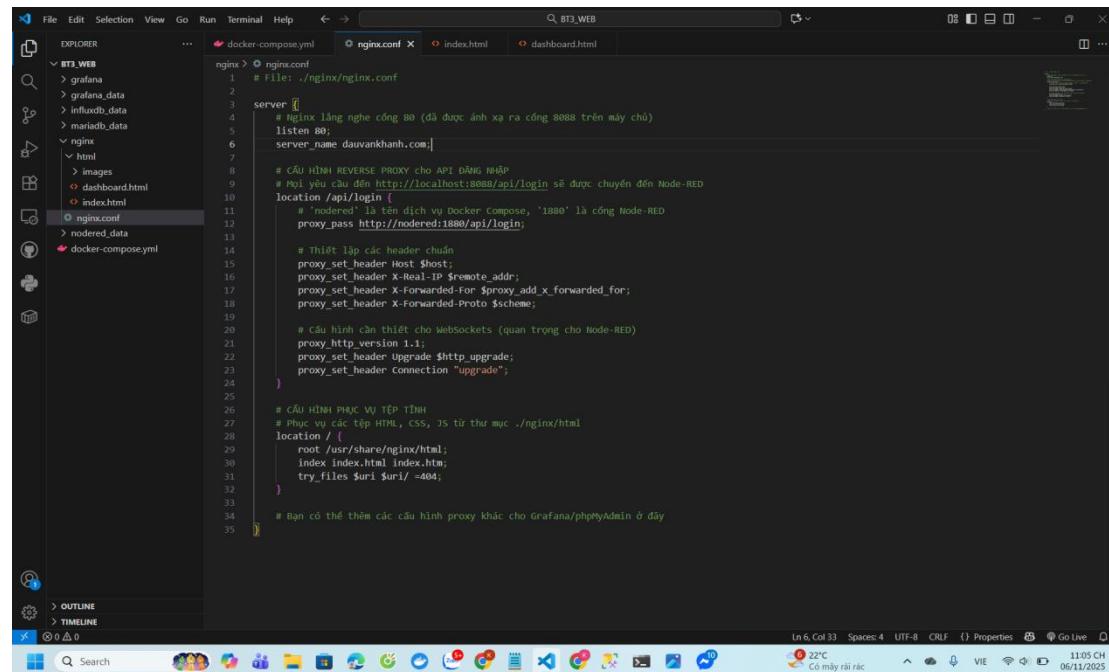
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97    rhino.acme.com        # source server
#      38.25.63.10      x.acme.com            # x client host

# localhost name resolution is handled within DNS itself.
#      127.0.0.1          localhost
#      ::1               localhost
127.0.0.1 dauvankhanh.com
# Added by Docker Desktop
192.168.1.34 host.docker.internal
192.168.1.34 gateway.docker.internal
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
127.0.0.1 dauvankhanh.com

```

3.5.2. Cấu hình nginx để http://fullname.com/nodered truy cập vào nodered qua cổng 80

- Vào file nginx.conf -> sửa server_name localhost thành server_name dauvankhanh.com:



```

nginx > nginx.conf
nginx.conf
server {
    listen 80;
    server_name dauvankhanh.com;

    # CẤU HÌNH REVERSE PROXY CHO API ĐĂNG NHẬP
    # Mọi yêu cầu đến http://localhost:8088/api/login sẽ được chuyển đến Node-RED
    location /api/login {
        proxy_pass http://nodered:1880/api/login;
    }

    # Thiết lập các header chuẩn
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_X_Forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    # CẤU HÌNH CẦN THIẾT CHO WEBSOCKETS (QUAN TRỌNG CHO NODE-RED)
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}

# CẤU HÌNH PHỤC VỤ TỆP TINH
# Phục vụ các tệp HTML, CSS, JS từ thư mục ./nginx/html
location / {
    root /usr/share/nginx/html;
    index index.html index.htm;
    try_files $uri $uri/ =404;
}

# Bạn có thể thêm các cấu hình proxy khác cho Grafana/phpMyAdmin ở đây

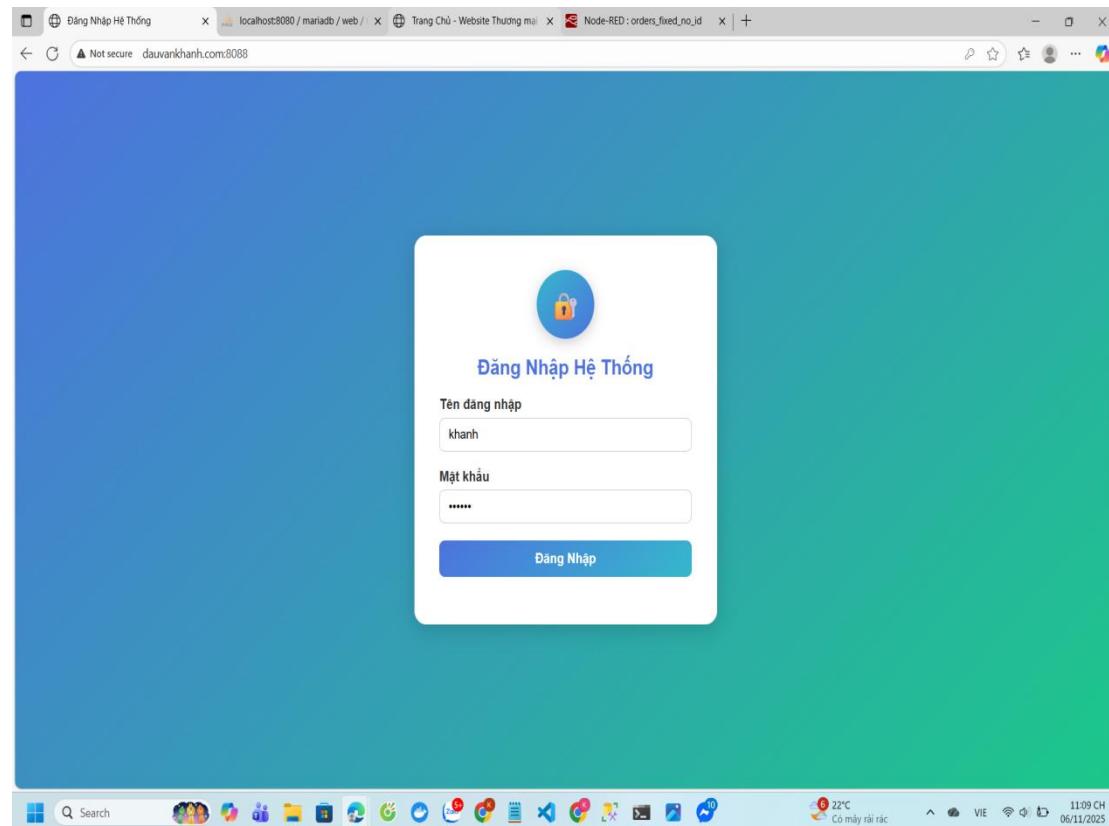
```

3.5.3. Cấu hình nginx để <http://fullname.com/grafana> truy cập vào grafana qua cổng 80

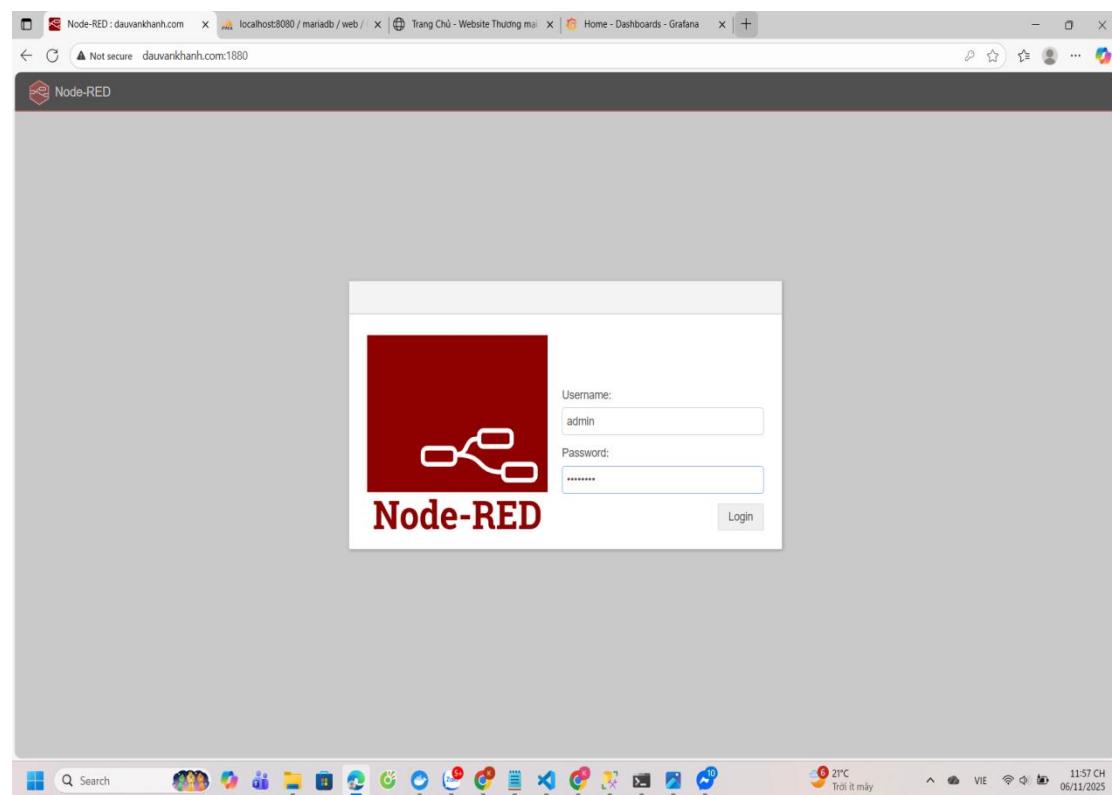
```
▷ Run Service
grafana:
  image: grafana/grafana:latest
  container_name: grafana
  restart: unless-stopped
  ports:
    - "3000:3000"
  depends_on:
    - influxdb
  volumes:
    - ./grafana_data:/var/lib/grafana
  environment:
    GF_SECURITY_ADMIN_USER: admin
    # SỬ DỤNG GIÁ TRỊ CỐ ĐỊNH
    GF_SECURITY_ADMIN_PASSWORD: admin123
    GF_USERS_ALLOW_SIGN_UP: "false"
  healthcheck:
    test: ["CMD", "wget", "-q", "-o", "/dev/null", "http://dauvankhanh.com/api/health"]
    interval: 15s
    timeout: 5s
    retries: 3
  networks:
    - iot_network
```

3.5.4. Kết quả

- Website

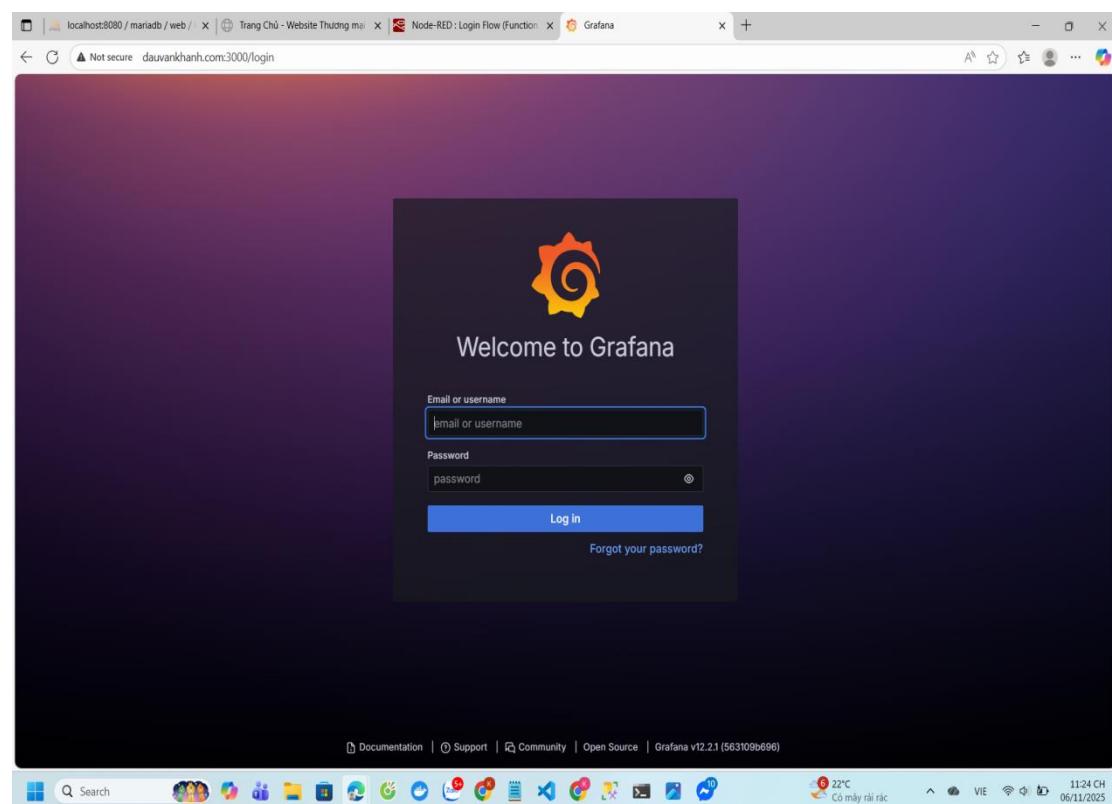


- Nodered

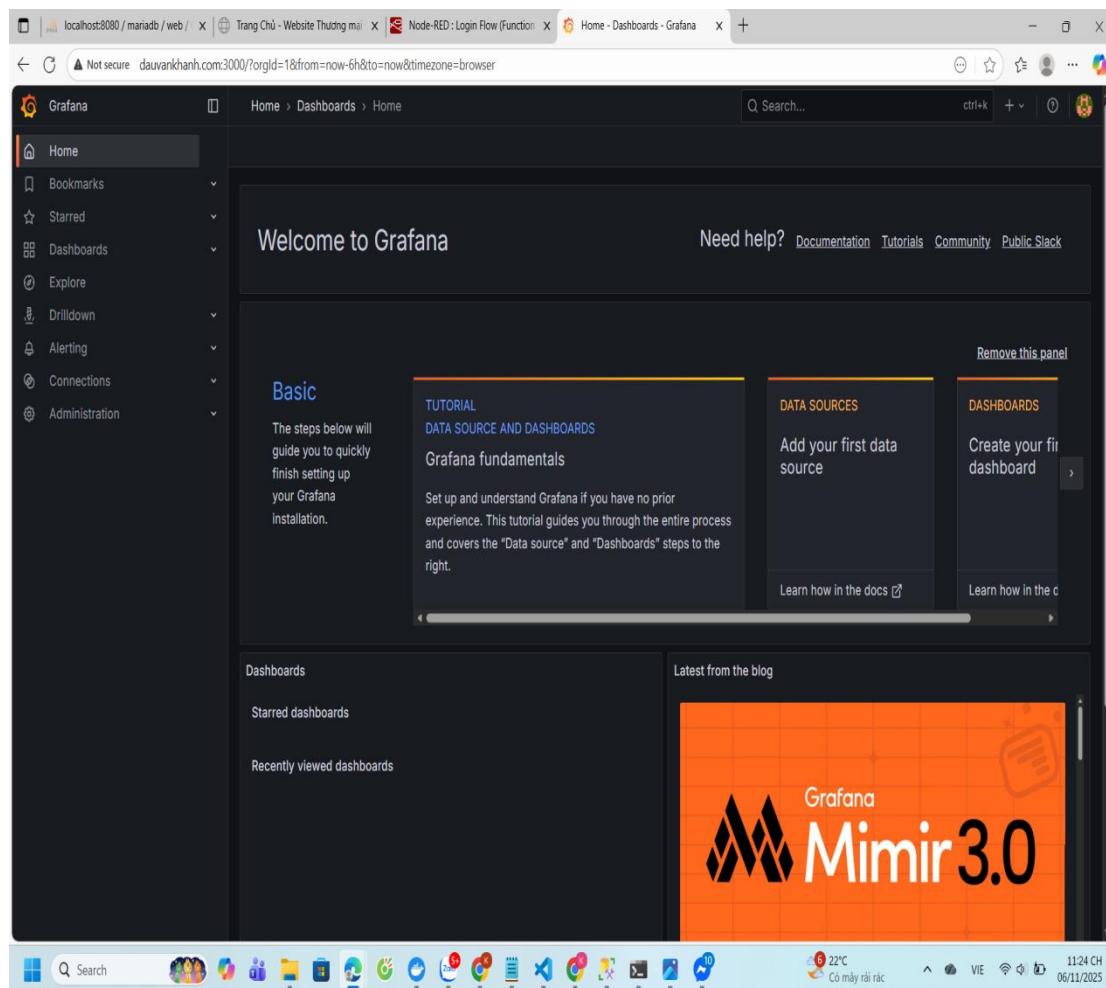


- Grafana

+ Đăng nhập:



+ Giao diện chính:



3.6. Kết luận:

- ❖ Sau khi hoàn thành đề tài, em đã xây dựng thành công một hệ thống web thương mại điện tử theo mô hình Frontend + Backend, đáp ứng đầy đủ các yêu cầu của bài toán đề ra. Cụ thể, hệ thống gồm giao diện người dùng (SPA) được xây dựng hoàn toàn bằng HTML, CSS, JavaScript, và backend sử dụng Node-RED để xử lý logic, truy vấn cơ sở dữ liệu MariaDB, đồng thời phản hồi dữ liệu dưới dạng JSON API.
- ❖ Các kết quả đạt được:
 - Thiết lập thành công môi trường backend Node-RED:
 - + Cài đặt Node-RED bằng Docker, cấu hình adminAuth để bảo mật trang quản trị.

- + Tạo các flow xử lý login, đăng ký, giỏ hàng, đặt hàng, thống kê đơn hàng.
- Kết nối và quản lý cơ sở dữ liệu MariaDB:
 - + Tạo các bảng: Users, Products, Orders, Categories, Orders_Items.
 - + Quản lý dữ liệu bằng phpMyAdmin, đảm bảo tính toàn vẹn và nhất quán dữ liệu.
- Xây dựng giao diện web SPA động:
 - + Sử dụng JavaScript để sinh toàn bộ giao diện từ dữ liệu JSON.
 - + Giao diện có các tính năng: đăng nhập, tìm kiếm, xem sản phẩm theo nhóm, thêm vào giỏ hàng, đặt hàng, xem thống kê.
- Thực hiện cơ chế đăng nhập bảo mật:
 - + Mã hóa thông tin đăng nhập khi gửi lên backend.
 - + Lưu trạng thái đăng nhập bằng cookie và session, chỉ đăng nhập lại khi logout.
- Tích hợp chức năng dành cho admin
 - + Admin có thể xem danh sách đơn hàng, cập nhật tình trạng, mã COD, hoặc hủy đơn.
 - + Xem biểu đồ thống kê số lượng hàng bán ra mỗi ngày bằng Grafana kết nối với Node-RED.
- Hệ thống đã vận hành ổn định, đảm bảo:
 - + Phản hồi nhanh giữa frontend – backend thông qua REST API.
 - + Giao diện trực quan, thân thiện với người dùng.
 - + Cấu trúc mã dễ mở rộng cho các tính năng quản trị sau này.
 - + Tích hợp tốt với cơ sở dữ liệu MariaDB và công cụ giám sát Grafana.
- ❖ Hướng phát triển trong tương lai
 - Tích hợp cổng thanh toán online (Momo, VNPay).
 - Bổ sung trang quản trị nâng cao cho admin (quản lý người dùng, sản phẩm, thống kê).
 - Tối ưu giao diện web cho thiết bị di động (responsive).
 - Nâng cấp bảo mật đăng nhập bằng JWT Token thay cho session truyền thống.

KẾT LUẬN

Qua quá trình thực hiện bài tập lớn môn Phát triển ứng dụng trên nền web, em đã có cơ hội tiếp cận một cách toàn diện các công nghệ quan trọng trong phát triển ứng dụng web. Từ việc xây dựng DLL trên môi trường .NET Framework và tích hợp vào Console App, Windows Form và Web ASP.NET, cho đến triển khai hệ thống Web Full-Stack bằng Apache, Node-RED và SQL Server, mỗi bài thực hành đều giúp em hiểu rõ hơn về cấu trúc, cách vận hành và mối liên kết giữa các thành phần trong một hệ thống web. Bên cạnh đó, quá trình thiết lập môi trường Linux, sử dụng Docker Desktop, và triển khai hoàn chỉnh hệ thống gồm nhiều container như MariaDB, Node-RED, InfluxDB, Grafana và Nginx giúp em có thêm trải nghiệm thực tế về mô hình triển khai web hiện đại dựa trên container hóa.

Thông qua bài tập lớn này, em đã rèn luyện được tư duy hệ thống và khả năng làm việc với nhiều công nghệ khác nhau, từ backend, frontend, cơ sở dữ liệu đến hạ tầng triển khai. Việc tự tay cấu hình máy chủ, xây dựng API, xử lý dữ liệu, phát triển giao diện và kết nối các dịch vụ giúp em hiểu sâu hơn quy trình xây dựng một ứng dụng web hoàn chỉnh, đồng thời nâng cao kỹ năng lập trình, phân tích và giải quyết vấn đề. Đây là nền tảng quan trọng để em tiếp tục phát triển năng lực trong lĩnh vực công nghệ thông tin, đặc biệt là trong hướng phát triển ứng dụng web và các hệ thống tích hợp trong tương lai.

Em xin chân thành cảm ơn thầy Đỗ Duy Cốp, người đã tận tình hướng dẫn, định hướng và hỗ trợ em trong suốt quá trình thực hiện bài tập lớn. Những kiến thức và sự chỉ bảo của thầy đã giúp em vượt qua nhiều khó khăn, hiểu sâu hơn về nội dung học phần và hoàn thành bài báo cáo một cách hiệu quả hơn.

TÀI LIỆU THAM KHẢO

1. W3Schools – *HTML, CSS, JavaScript, SQL* (thư viện tham khảo tổng quát):
<https://www.w3schools.com>
2. Node-RED Official Documentation. *Creating flows, HTTP nodes, database integration*: <https://nodered.org/docs>
3. Võ Hoài Nam (2020). *Giáo trình Lập trình Web* - NXB Đại học Quốc gia TP.HCM.
4. Website Học liệu mở Đại học FPT – Chủ đề *Lập trình .NET Framework* và *Windows Forms*.
5. Nguyễn Thanh Tùng (2021): *Lập trình Web toàn diện với HTML5, CSS3, JavaScript và jQuery* - NXB Thông tin & Truyền thông.
6. Hoàng Đình Cường (2019): *Giáo trình Hệ điều hành Linux* - NXB Bách khoa Hà Nội.
7. Microsoft Documentation: *ASP.NET Web Forms, IIS Deployment, .NET Framework*.
8. Apache HTTP Server Documentation – *VirtualHost, server configuration*:
<https://httpd.apache.org/docs>
9. Docker Documentation – *Docker Engine, Docker Compose, Container deployment*: <https://docs.docker.com>
10. Grafana Documentation – *Dashboards, Graphs, Data Sources (InfluxDB)*:
<https://grafana.com/docs>
11. InfluxDB Documentation – *Time-series database, retention policies, queries*: <https://docs.influxdata.com>

MÃ QR GITHUB

BÀI TẬP 1



BÀI TẬP 2



BÀI TẬP 3



BÁO CÁO

