

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ



BÀI TẬP LỚN
PHÁT TRIỂN ỨNG DỤNG
TRÊN NỀN WEB

NGÀNH : KỸ THUẬT MÁY TÍNH

HỆ : ĐẠI HỌC CHÍNH QUY

THÁI NGUYÊN - 2025

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ



BÀI TẬP LỚN
PHÁT TRIỂN ỨNG DỤNG
TRÊN NỀN WEB

Sinh viên thực hiện : Lăng Nguyễn Minh Lượng

Lớp : K58KTP

MSSV: K225480106044

Giáo viên hướng dẫn : ThS Đỗ Duy Cốp

THÁI NGUYÊN - 2025

TRƯỜNG ĐHKT&CN
KHOA ĐIỆN TỬ

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÀI TẬP LỚN

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

Sinh viên: Lăng Nguyễn Minh Lượng MSSV:K225480106044

Lớp: K58KTP Ngành: Kỹ Thuật Máy Tính

Giáo viên hướng dẫn : ThS Đỗ Duy Cốp

Ngày giao bài Ngày hoàn thành : 26/11/2025

Tên đê tài : Trình bày các bài tập đã được giao

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

Yêu cầu : Trình bày các bài tập đã được giao (đầu bài và các bước làm)
.....
.....
.....
.....
.....
.....
.....
.....
.....

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Thái Nguyên, ngày....tháng 11 năm 2025

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

MỤC LỤC

| | |
|--|----|
| LỜI CAM ĐOAN..... | 4 |
| Đề bài | 5 |
| Bài 1. Máy tính BMI | 7 |
| 1.Tạo SOLUTION sau đó add các project :..... | 7 |
| 2.Chạy thử console..... | 7 |
| 3.Chạy thử Winforms | 8 |
| 4.Chạy thử webforms | 9 |
| Bài 2: APACHE | 10 |
| Bài 3 LẬP TRÌNH ỨNG DỤNG WEB trên nền linux..... | 17 |
| KẾT LUẬN | 26 |

LỜI CAM ĐOAN

Tôi xin cam đoan rằng bài tập lớn này là kết quả của quá trình nghiên cứu và thực hiện của chính tôi. Tất cả các số liệu và tài liệu được trích dẫn trong bài tập lớn đều có nguồn gốc rõ ràng và được xác thực. Nội dung trong là duy nhất, chưa từng được công bố trong bất kỳ công trình nào trước đây. Tôi xin hoàn toàn chịu trách nhiệm trước pháp luật về tính trung thực và nội dung của bài tập lớn.

Tên sinh viên

Lăng Nguyễn Minh Lượng

Đề bài

Bài tập 1:

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 ấn 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 ấn 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 ấn cá nhân; keyword: c# window Desktop => Windows Form Application (.NET Framework), biên dịch ra EXE

Bài tập 2: APACHE

Xây dựng một website cá nhân có front-end (HTML/CSS/JS) chạy trên Apache Web Server, kết nối tới back-end Node-RED thông qua API, và sử dụng CSDL SQL Server 2022 để lưu trữ dữ liệu. Toàn bộ tiến trình được ghi lại trên GitHub.

Bài tập 3:

Yêu cầu : LẬP TRÌNH ỨNG DỤNG WEB trên nền linux

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: Thông 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.

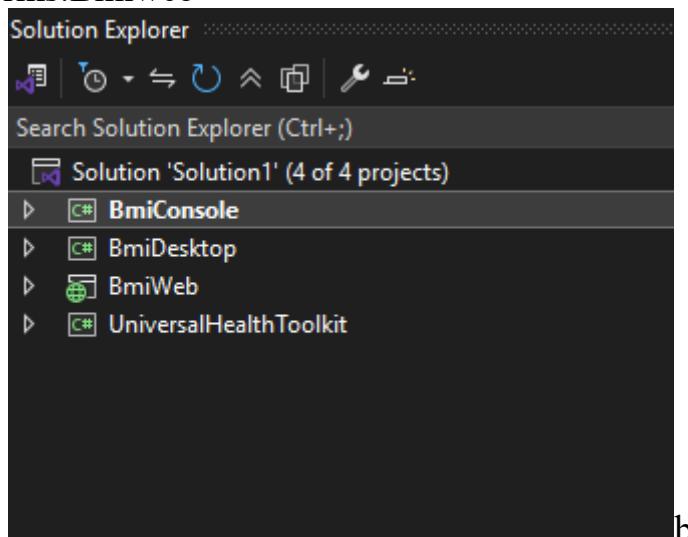
5. Nginx làm web-server

- 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 1. Máy tính BMI

1.Tạo SOLUTION sau đó add các project :

- thư viện library: UniversalHealthToolkit
- -console: Bmiconsole
- -winforms: Bmidesktop
- -webforms:Bmiweb

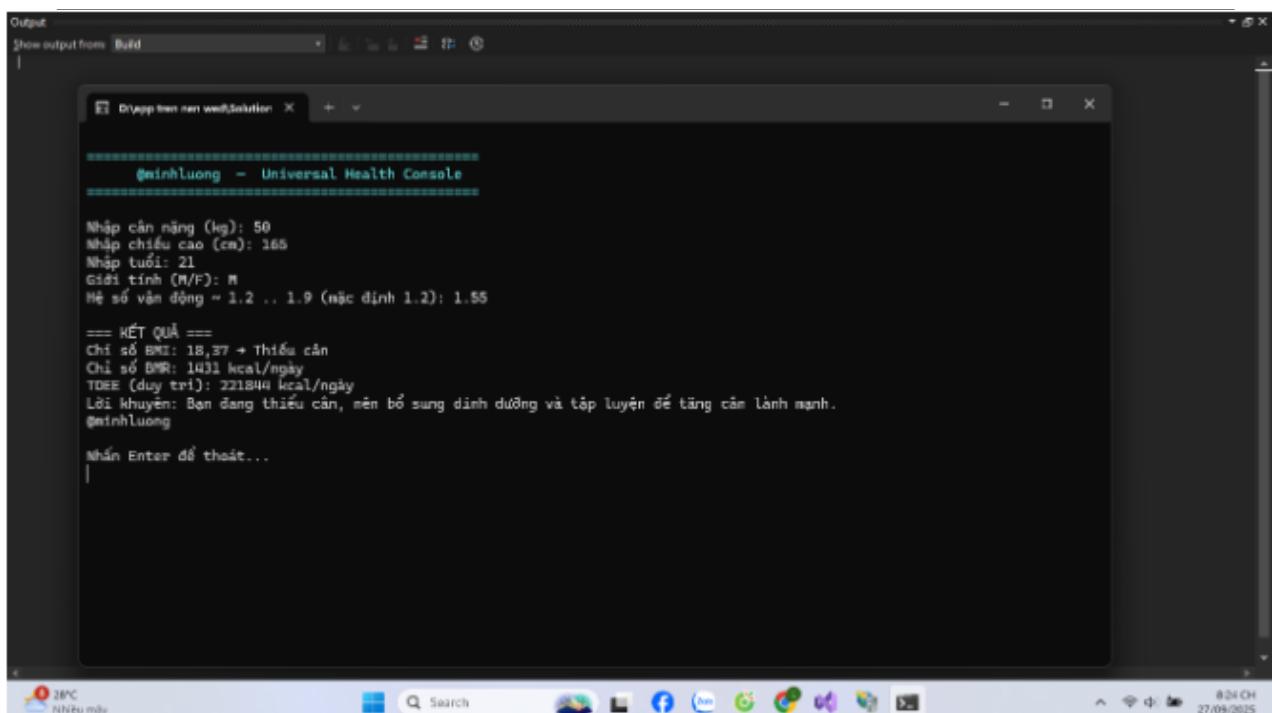


2.Chạy thử console

Sau khi tạo project chuột phải vào references -> add references -> tick chọn thư viện UniversalHealthToolkit -> ok

code file program.cs xong build

chuột phải vào tên project Bmiconsole chọn Set as startup project -> ctrl+f5 để chạy chương trình

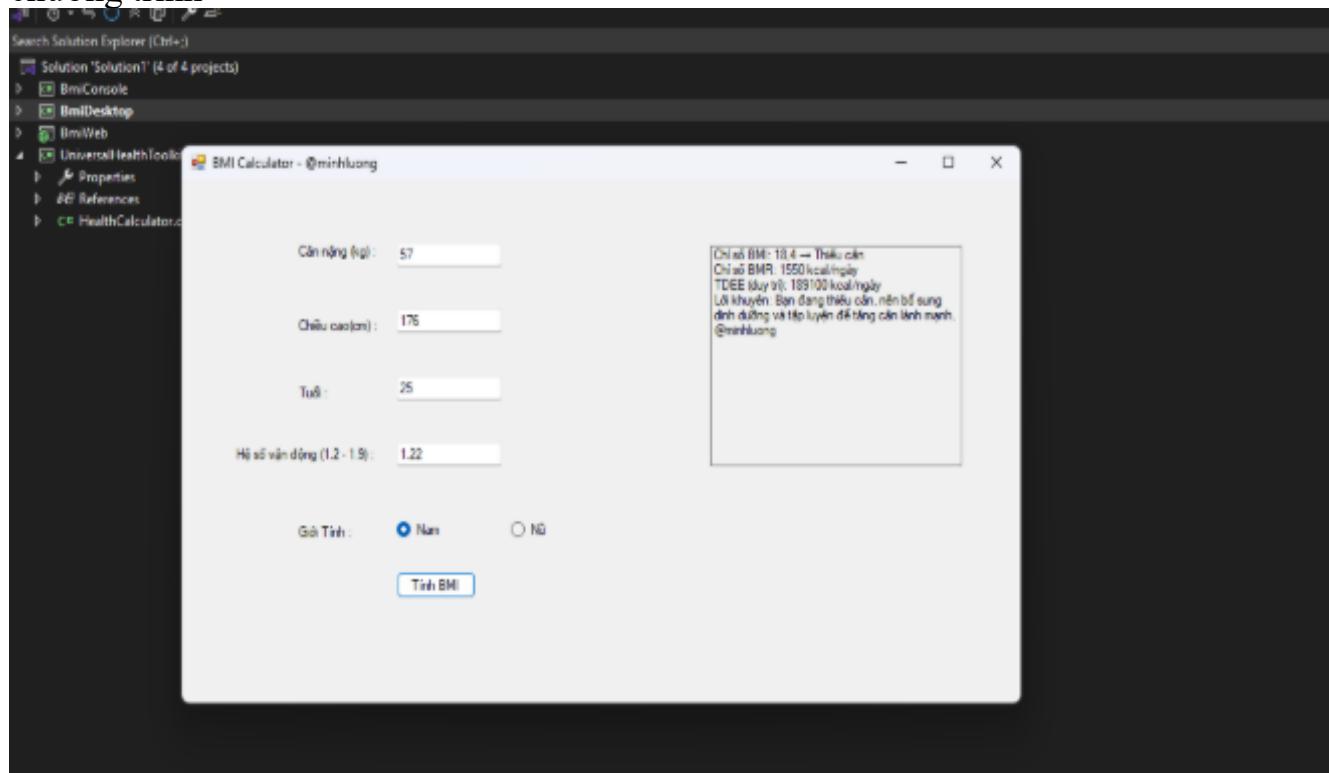


3.Chạy thử Winforms

sau khi tạo project chuột phải vào references -> add references -> tick chọn thư viện UniversalHealthToolkit -> ok

code file mainform.cs -> mở mainform.cs[design] để cấu hình winform -> build project

chuột phải vào tên project Bmidesktop chọn Set as startup project -> ctrl+f5 để chạy chương trình



4.Chạy thử webforms

sau khi tạo project chuột phải vào references -> add references -> tick chọn thư viện UniversalHealthToolkit -> ok

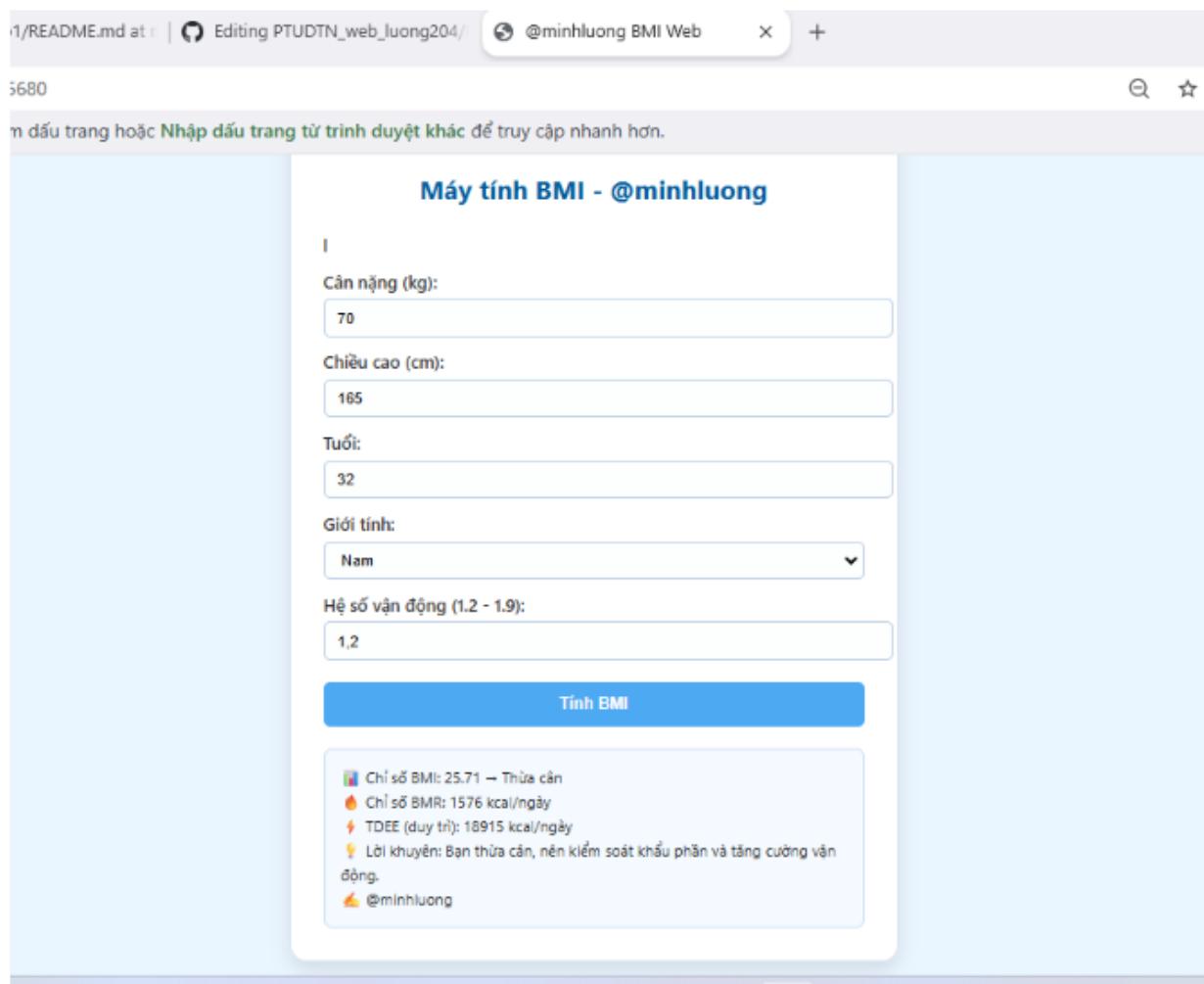
Chuột phải vào tên project Bmiweb chọn add -> new item -> tạo file index.html và api.aspx

code file index.html và api.aspx.cs

vào iis manager -> add website -> đặt mane và host name local -> ok

vào notepad adm nhập host local

vào lại visual studio chuột phải vào tên project Bmiweb chọn Set as startup project -> ctrl+f5 để chạy chương trình



Bài 2: APACHE

1. Tạo repository GitHub

Truy cập: <https://github.com>

Tạo repo tên: Apache-

Chọn Public.

Tạo file README.md để ghi lại quá trình làm việc, chụp màn hình, ghi chú chi tiết các bước.

Cập nhật tiến trình 15–30 phút/lần.

2. Cài đặt Apache Web Server 2.1. Tắt IIS (nếu có)

Mở CMD (adm)

nhập lệnh

```
iisreset /stop
```

2.2. Tải Apache

Tải từ: <https://www.apachelounge.com/download/>

File ví dụ: httpd-2.4.65-250724-win64-VS17.zip

2.3. Giải nén

Giải nén vào: D:\Apache24\

2.4. Sửa file cấu hình

Mở D:\Apache24\conf\httpd.conf

```
Define SRVROOT "D:/Apache24"
ServerName localhost:80
Include conf/extra/httpd-vhosts.conf
```

2.5. Tạo Virtual Host

Mở D:\Apache24\conf\extra\httpd-vhosts.conf, thêm:

```
<VirtualHost *:80>
    ServerAdmin admin@lamnguyenminhluong.com
    DocumentRoot "D:/Apache24/lamnguyenminhluong"
    ServerName lamnguyenminhluong.com
    ErrorLog "logs/lam_error.log"
    CustomLog "logs/lam_access.log" common
</VirtualHost>
```

2.6. Map domain

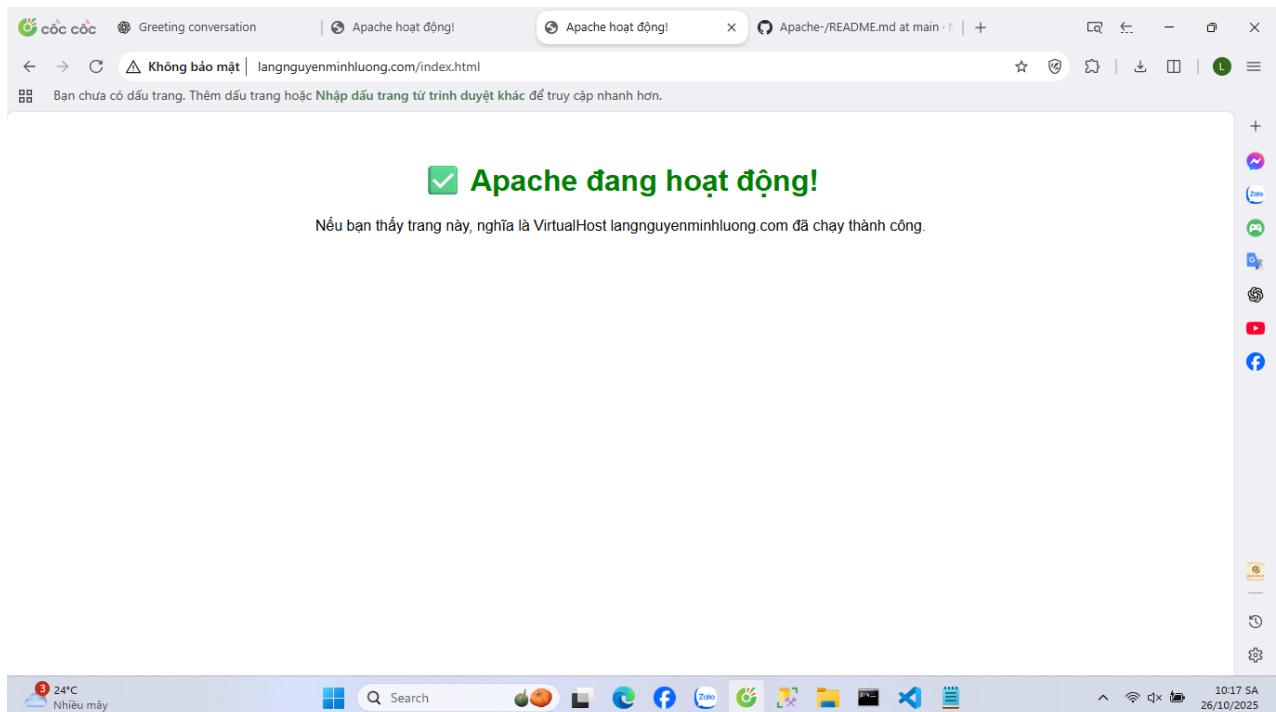
Mở file: C:\Windows\System32\drivers\etc\hosts

```
127.0.0.1 lamnguyenminhluong.com
```

2.7. Cài đặt Apache làm service

```
cd D:\Apache24\bin  
httpd.exe -k install  
httpd.exe -k start
```

→ Mở trình duyệt: <http://lamnguyenminhluong.com>



3. Cài đặt Node.js và Node-RED 3.1. Tải Node.js

Vào <https://nodejs.org/en>

Tải bản LTS: node-v20.19.5-x64.msi

Cài đặt tại: D:\nodejs

3.2. Cài Node-RED

Mở CMD (Administrator):

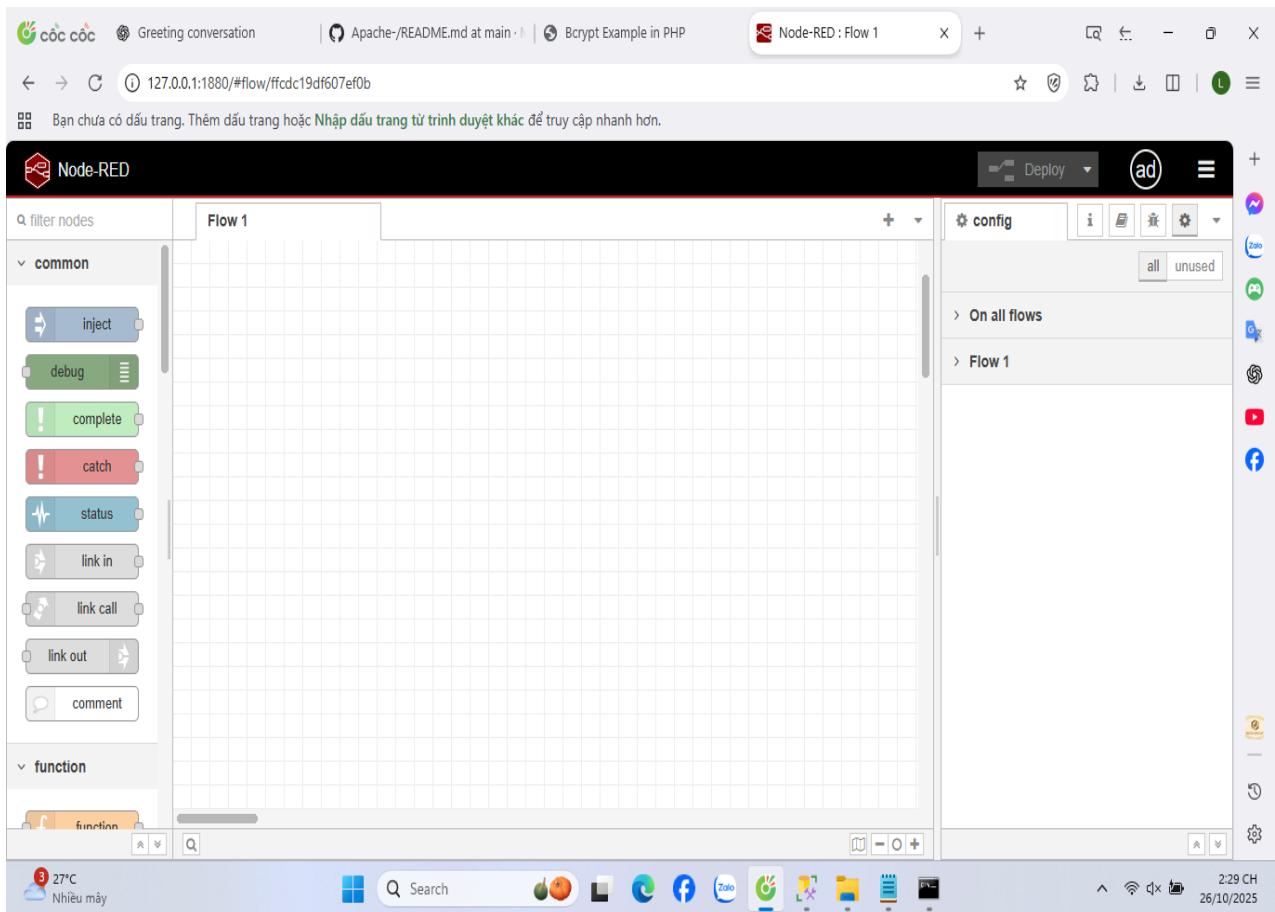
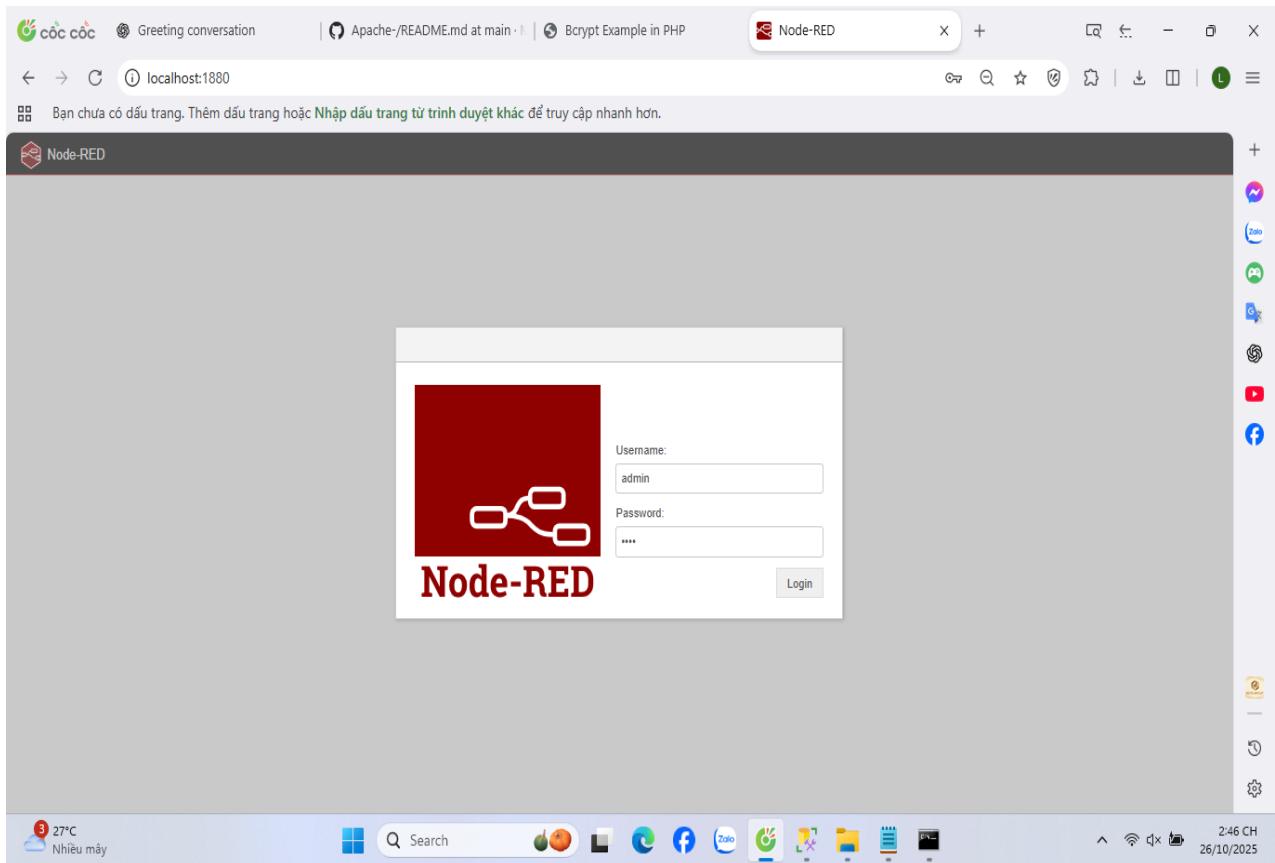
```
npm install -g --unsafe-perm node-red
```

Kiểm tra:

```
node-red
```

Truy cập: <http://localhost:1880>

Bài tập lớn An Toàn Bảo Mật



3.3. Cài Node-RED làm service (tùy chọn)

Tải nssm từ <https://nssm.cc/download>

Mở CMD:

```
nssm install NodeRED
```



Điền đường dẫn Node.js + đổi số node-red

Start service.

4. Tạo CSDL SQL Server

Mở SQL Server Management Studio

Tạo database: BT2

Tạo bảng ví dụ:

```
USE BT2;
GO
CREATE TABLE dbo.ThietBi (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    TenThietBi NVARCHAR(100),
    TrangThai NVARCHAR(50),
    GiaTri FLOAT,
    ThoiGian DATETIME DEFAULT GETDATE()
);
GO
```



Nhập dữ liệu mẫu

5. Kết nối Node-RED với SQL Server

Mở Node-RED: <http://localhost:1880>

Cài node:

```
node-red-contrib-mssql-plus  
node-red-node-mysql  
node-red-contrib-telegrambot  
node-red-contrib-moment  
node-red-contrib-cron-plus
```

Kéo node http in → function → mssql → http response

Cấu hình mssql:

```
{  
  "user": "sa",  
  "password": "123",  
  "server": "127.0.0.1,1433",  
  "database": "BT2",  
  "port": 1433  
}
```

Trong node function:

```
var ten = msg.req.query.q || '';  
msg.topic = `SELECT * FROM ThietBi WHERE TenThietBi LIKE N'%${ten}%'`;  
return msg;
```

Đặt http in URL = /timkiem

Deploy → Test:

<http://localhost:1880/timkiem?q=cảm>

Bài tập lớn An Toàn Bảo Mật

The screenshot shows the Node-RED interface with a flow titled "Flow 1". The flow consists of the following nodes:

- [get] timkiem
- Xử lý thêm mã liên kết
- MSSQL-PLUS
- Http (200)
- debug

A feedback loop connects the output of the "Http (200)" node back to the "debug" node.

The browser window displays the URL <http://localhost:1880/timkiem?q=cảm>. The page content is a JSON array of two objects representing sensor data:

```
[{"ID": 1, "TenThietBi": "Cảm biến nhiệt độ", "TrangThai": "Hoạt động", "GiaTri": 25.7, "ThoiGian": "2025-10-26T14:55:34.577Z"}, {"ID": 2, "TenThietBi": "Cảm biến ánh sáng", "TrangThai": "Ngừng hoạt động", "GiaTri": 0, "ThoiGian": "2025-10-26T14:55:34.577Z"}]
```

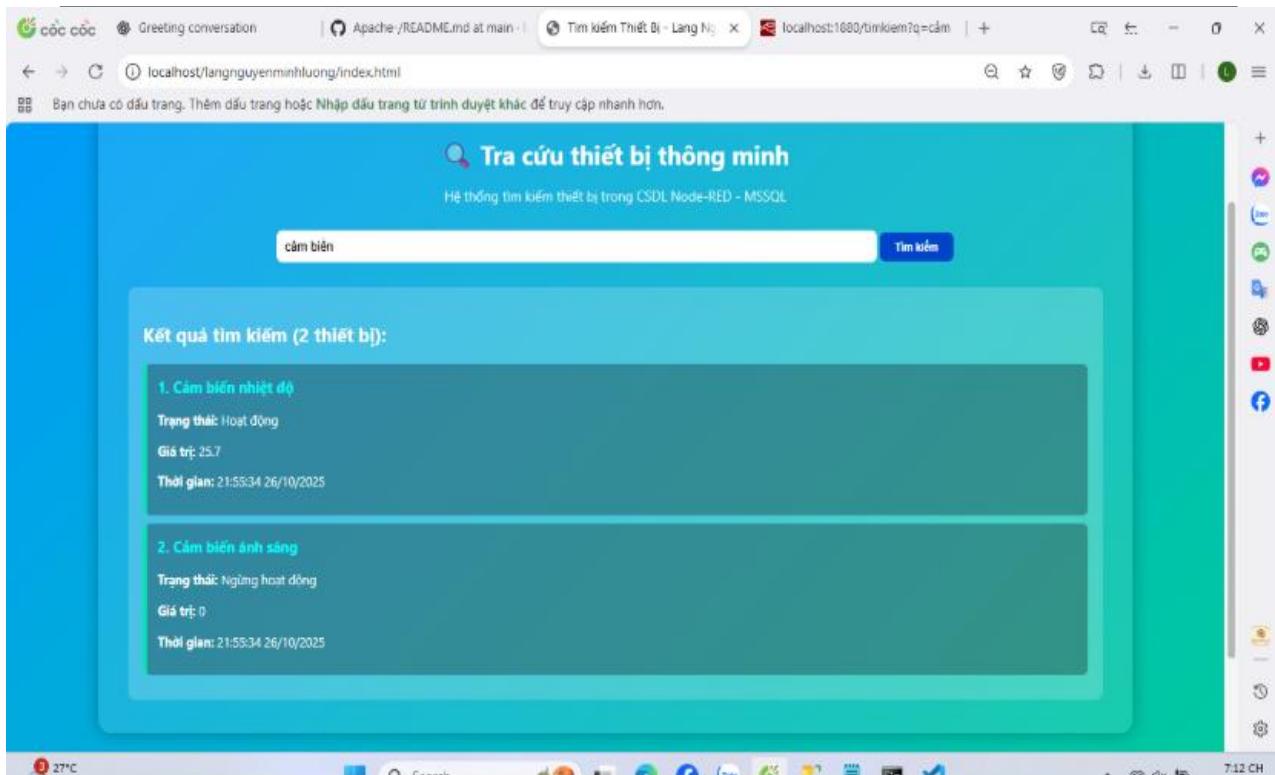
6. Tạo Front-End HTML/JS/CSS

Tạo thư mục:

D:\Apache24\htdocs\langnguyenminhluong\

chạy <http://localhost/langnguyenminhluong/>

Bài tập lớn An Toàn Bảo Mật



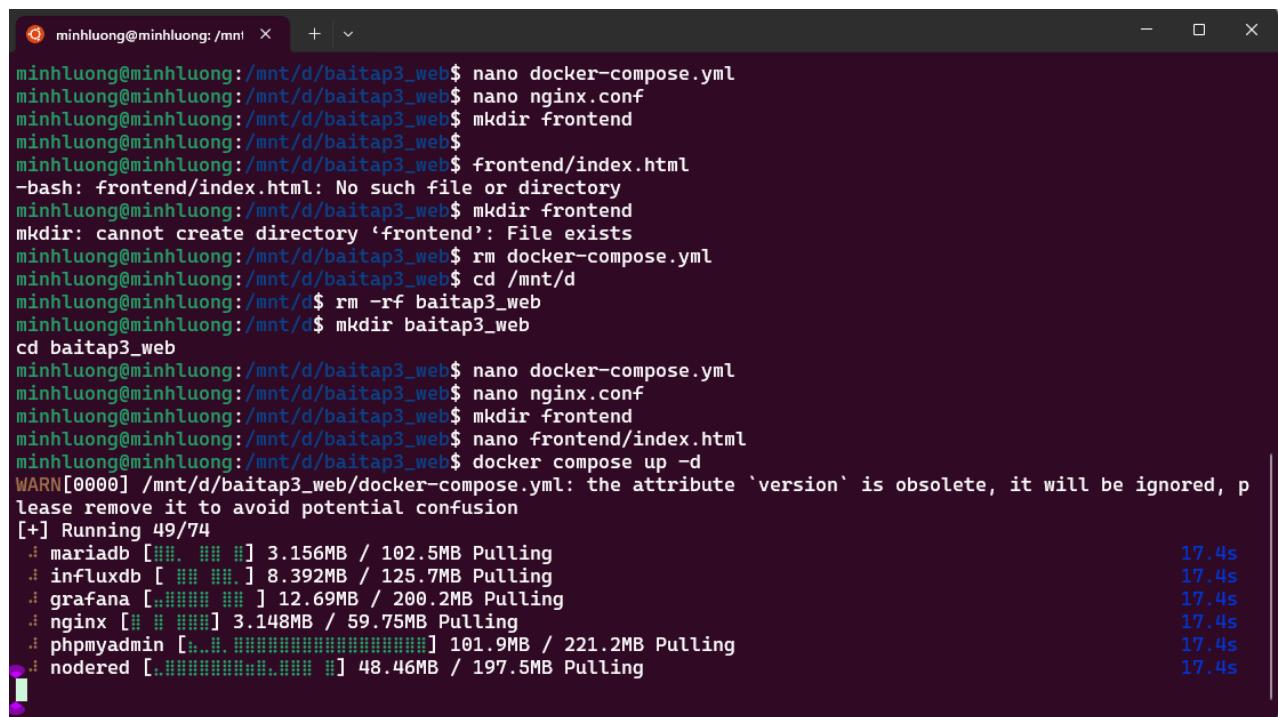
→ Mở trình duyệt: <http://localhost/langnguyenminhluong/index.html>

Nếu hiện dữ liệu từ SQL ==> Hoàn tất

7. Viết phần nhận xét cá nhân Qua bài tập này, em đã hiểu rõ quy trình cài đặt và cấu hình các phần mềm như Apache, Node.js, Node-RED và SQL Server, cũng như cách thêm các thư viện cần thiết để hệ thống hoạt động. Em nắm được cách sử dụng Node-RED để tạo API back-end, xử lý dữ liệu và kết nối với cơ sở dữ liệu SQL Server. Đồng thời, em hiểu cách front-end (HTML/CSS/JS) giao tiếp với back-end thông qua API, gửi yêu cầu tìm kiếm và nhận dữ liệu JSON để hiển thị lên giao diện. Bài tập giúp em củng cố kiến thức về mô hình client-server và quy trình xây dựng một ứng dụng web hoàn chỉnh.

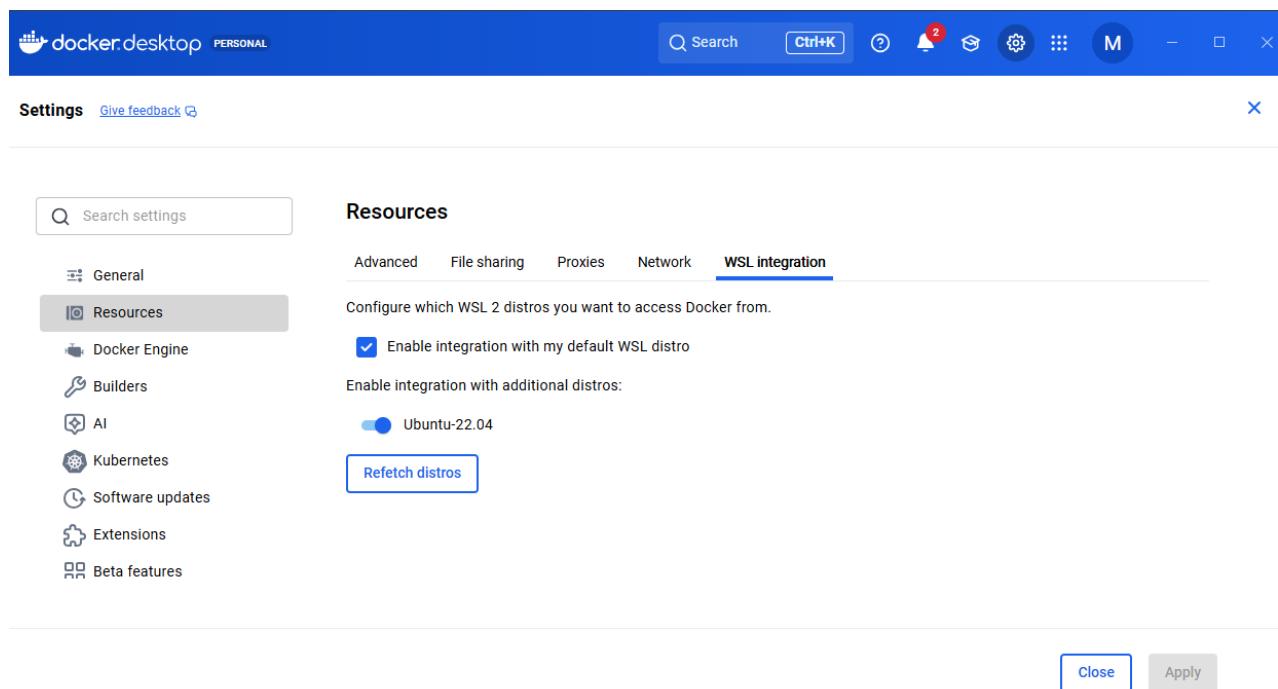
Bài 3 LẬP TRÌNH ỦNG DỤNG WEB trên nền linux

Mở cmd chạy lệnh wsl --install

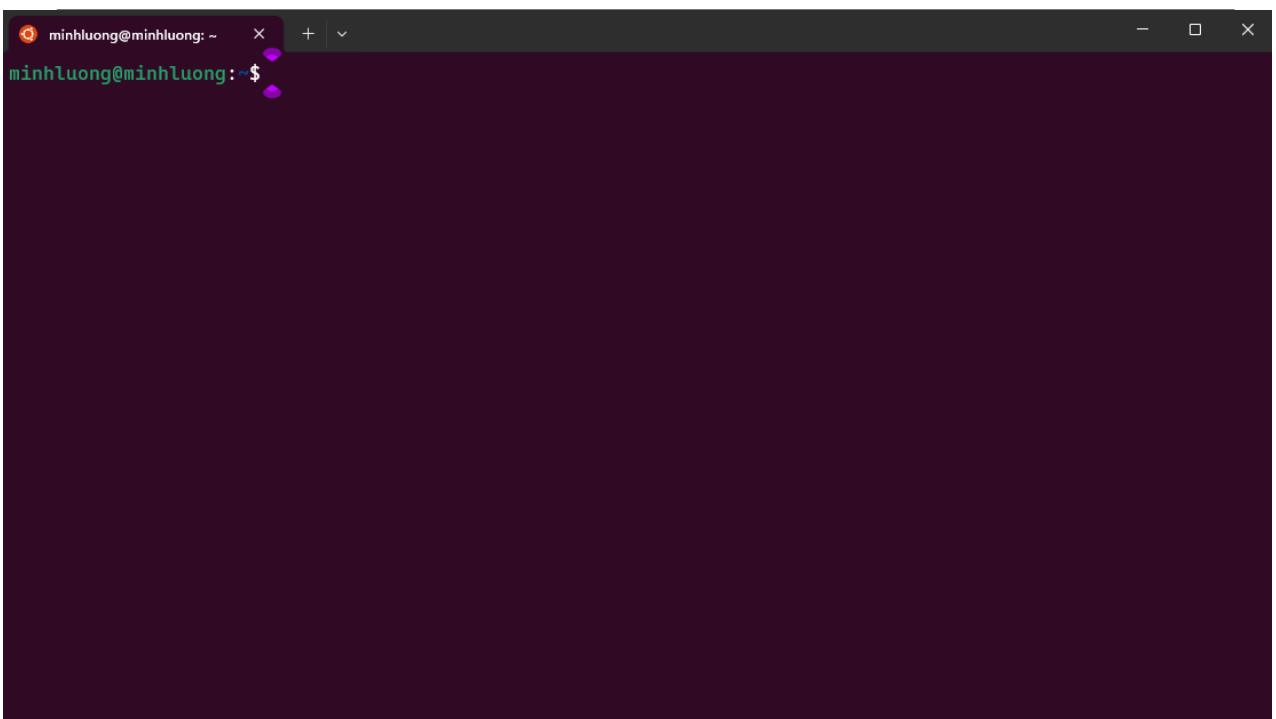


```
minhluong@minhluong:/mnt/d/baitap3_web$ nano docker-compose.yml
minhluong@minhluong:/mnt/d/baitap3_web$ nano nginx.conf
minhluong@minhluong:/mnt/d/baitap3_web$ mkdir frontend
minhluong@minhluong:/mnt/d/baitap3_web$ 
minhluong@minhluong:/mnt/d/baitap3_web$ frontend/index.html
-bash: frontend/index.html: No such file or directory
minhluong@minhluong:/mnt/d/baitap3_web$ mkdir frontend
mkdir: cannot create directory ‘frontend’: File exists
minhluong@minhluong:/mnt/d/baitap3_web$ rm docker-compose.yml
minhluong@minhluong:/mnt/d/baitap3_web$ cd /mnt/d
minhluong@minhluong:/mnt/d$ rm -rf baitap3_web
minhluong@minhluong:/mnt/d$ mkdir baitap3_web
cd baitap3_web
minhluong@minhluong:/mnt/d/baitap3_web$ nano docker-compose.yml
minhluong@minhluong:/mnt/d/baitap3_web$ nano nginx.conf
minhluong@minhluong:/mnt/d/baitap3_web$ mkdir frontend
minhluong@minhluong:/mnt/d/baitap3_web$ nano frontend/index.html
minhluong@minhluong:/mnt/d/baitap3_web$ docker compose up -d
WARN[0000] /mnt/d/baitap3_web/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 49/74
  ⠄ mariadb [██████████] 3.156MB / 102.5MB Pulling                         17.4s
  ⠄ influxdb [███ █████] 8.392MB / 125.7MB Pulling                         17.4s
  ⠄ grafana [██████████] 12.69MB / 200.2MB Pulling                         17.4s
  ⠄ nginx [███ ███ ███] 3.148MB / 59.75MB Pulling                         17.4s
  ⠄ phpmyadmin [███████████████████] 101.9MB / 221.2MB Pulling                         17.4s
  ⠄ nodered [███████████████████] 48.46MB / 197.5MB Pulling                         17.4s
```

tải docker desktop => vào cài đặt bật ubuntu => ally



sau khi tạo user và pw thì được giao diện ubuntu như này



chạy lệnh nano docker-compose.yml => sau đó code

```
version: "3.8"

services:
  # =====
  # 🗃 MariaDB Database
  # =====
  mariadb:
    image: mariadb:latest
    container_name: mariadb
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: 123456
      MYSQL_DATABASE: iotdb
      MYSQL_USER: root
      MYSQL_PASSWORD: 123456
    ports:
      - "3306:3306"
    volumes:
      - ./db:/var/lib/mysql
    networks:
      - backend

  # =====
  # 📈 phpMyAdmin (DB GUI)
  # =====
```

```
phpmyadmin:  
  image: phpmyadmin:latest  
  container_name: phpmyadmin  
  restart: always  
  environment:  
    PMA_HOST: mariadb  
    PMA_PORT: 3306  
  ports:  
    - "8080:80"  
  depends_on:  
    - mariadb  
  networks:  
    - backend  
  
# =====  
# ⏳ InfluxDB (Time-Series DB)  
# =====  
influxdb:  
  image: influxdb:latest  
  container_name: influxdb  
  restart: always  
  ports:  
    - "8086:8086"  
  environment:  
    - DOCKER_INFLUXDB_INIT_MODE=setup  
    - DOCKER_INFLUXDB_INIT_USERNAME=admin  
    - DOCKER_INFLUXDB_INIT_PASSWORD=12345678      # >= 8 ký tự  
    - DOCKER_INFLUXDB_INIT_ORG=iotorg  
    - DOCKER_INFLUXDB_INIT_BUCKET=iotdata  
    - DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=my-super-token  
  volumes:  
    - ./influxdb_data:/var/lib/influxdb2  
  networks:  
    - backend  
  
# =====  
# 🛡 Node-RED (Logic & API)  
# =====  
nodered:  
  image: nodered/node-red:latest  
  container_name: nodered  
  restart: always  
  ports:  
    - "1880:1880"
```

```
environment:
  - TZ=Asia/Ho_Chi_Minh
volumes:
  - ./nodered_data:/data
depends_on:
  - mariadb
  - influxdb
networks:
  - frontend
  - backend

# =====
# 📈 Grafana (Dashboard)
# =====

grafana:
  image: grafana/grafana:latest
  container_name: grafana
  restart: always
  ports:
    - "3000:3000"
  environment:
    - GF_SECURITY_ADMIN_USER=admin
    - GF_SECURITY_ADMIN_PASSWORD=12345678
    - GF_INSTALL_PLUGINS=marcusolsson-json-datasource
  volumes:
    - ./grafana_data:/var/lib/grafana
  depends_on:
    - influxdb
networks:
  - frontend
  - backend

# =====
# 🌐 Nginx (Frontend Web)
# =====

nginx:
  image: nginx:latest
  container_name: nginx
  restart: always
  ports:
    - "8088:80"
  volumes:
    - ./nginx/conf.d:/etc/nginx/conf.d
    - ./nginx/www:/usr/share/nginx/html
```

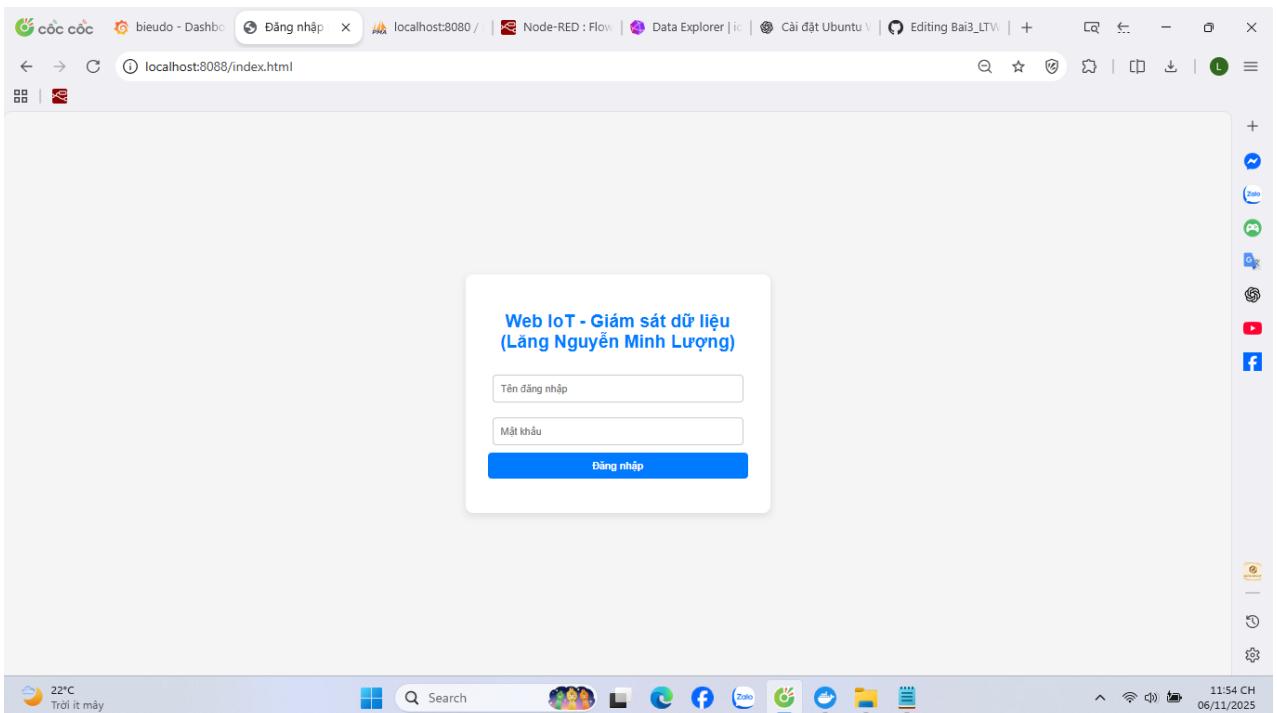
```
depends_on:
  - nodered
  - grafana
networks:
  - frontend
  - backend

# =====
# 🌐 Network Configuration
# =====

networks:
  frontend:
    driver: bridge
  backend:
    driver: bridge
```

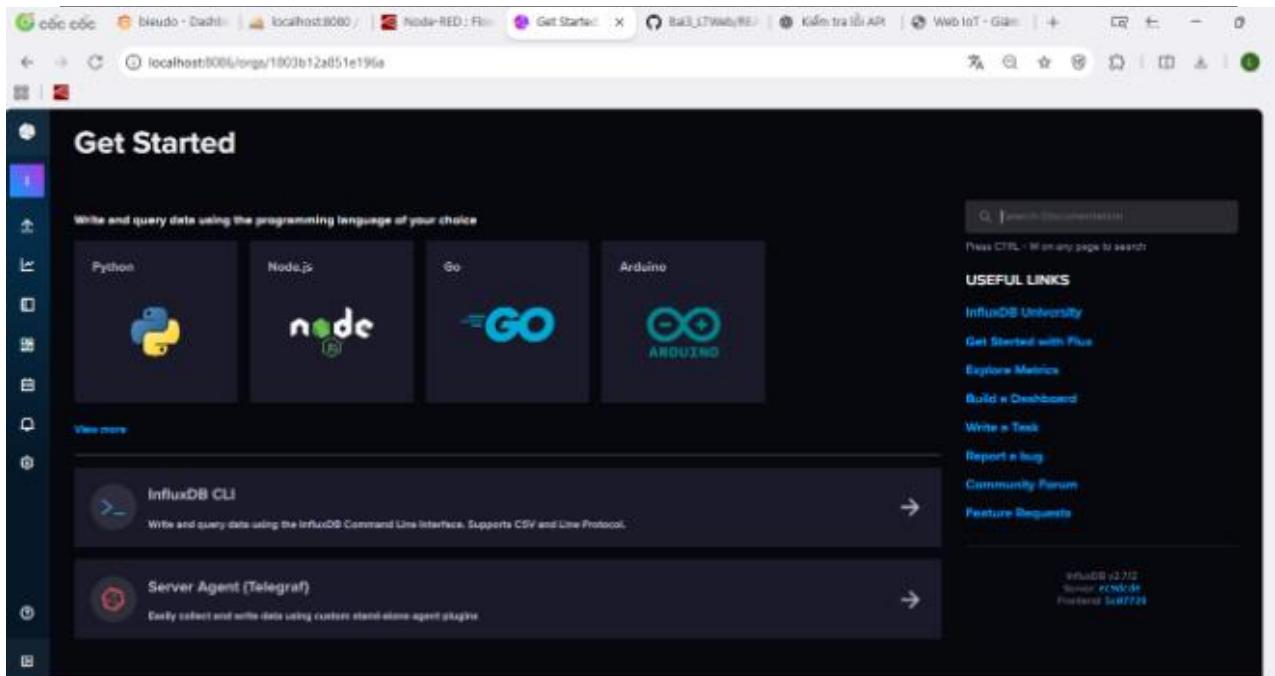
ctrl+O + enter để lưu ctrl+x để trở về ubuntu sau khi load xong thì mariadb (3306), phpmyadmin (8080), nodered/node-red (1880), influxdb (8086), grafana/grafana (3000), nginx (80,443) sẽ được cài đặt

giao diện login

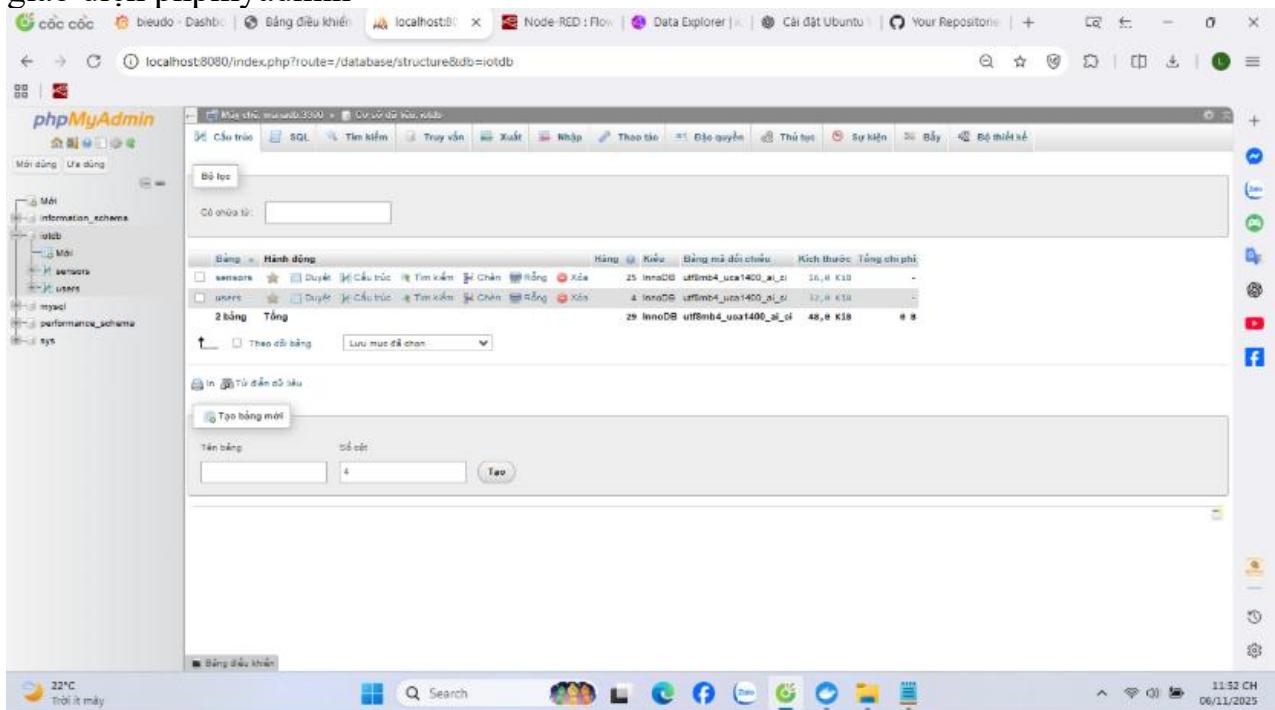


giao diện influxdb

Bài tập lớn An Toàn Bảo Mật

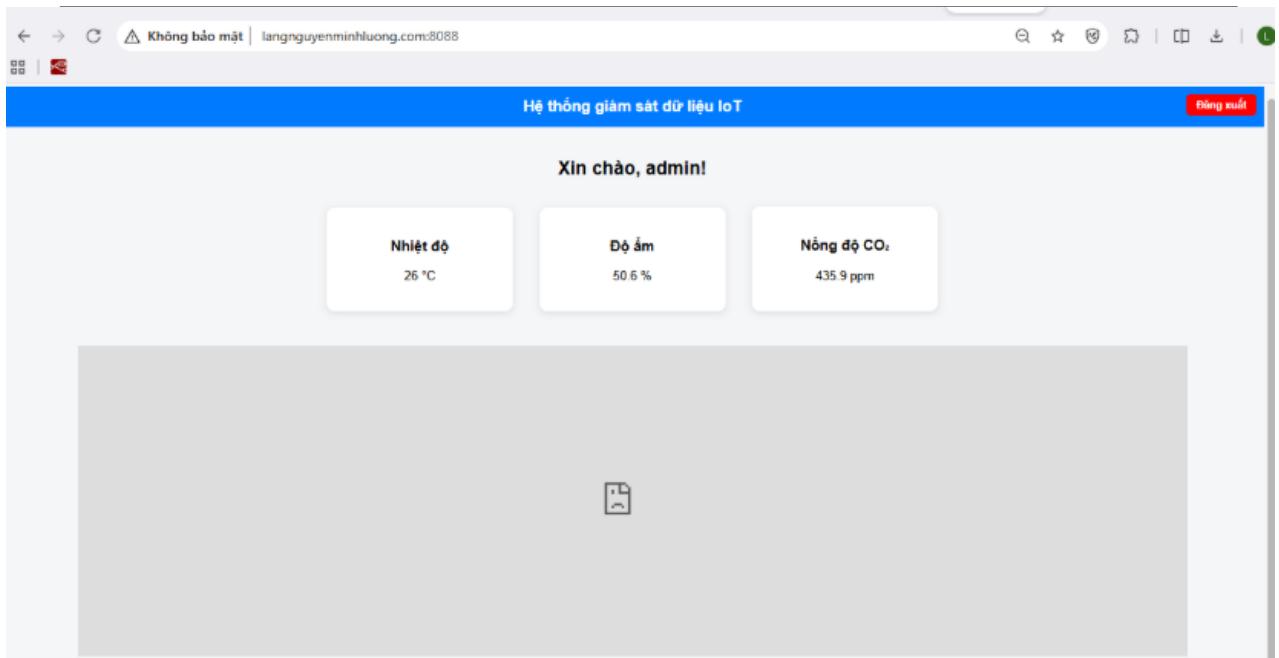


giao diện phpmyadmin

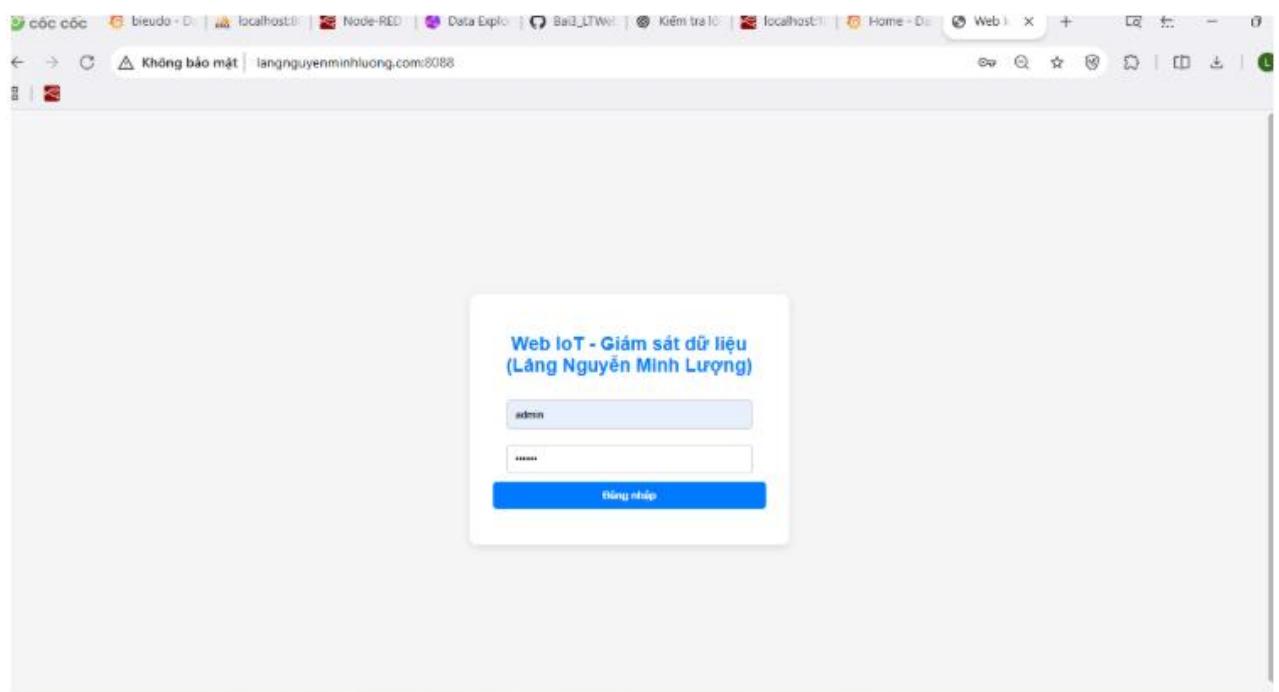


giao diện nginx

Bài tập lớn An Toàn Bảo Mật



Kết quả

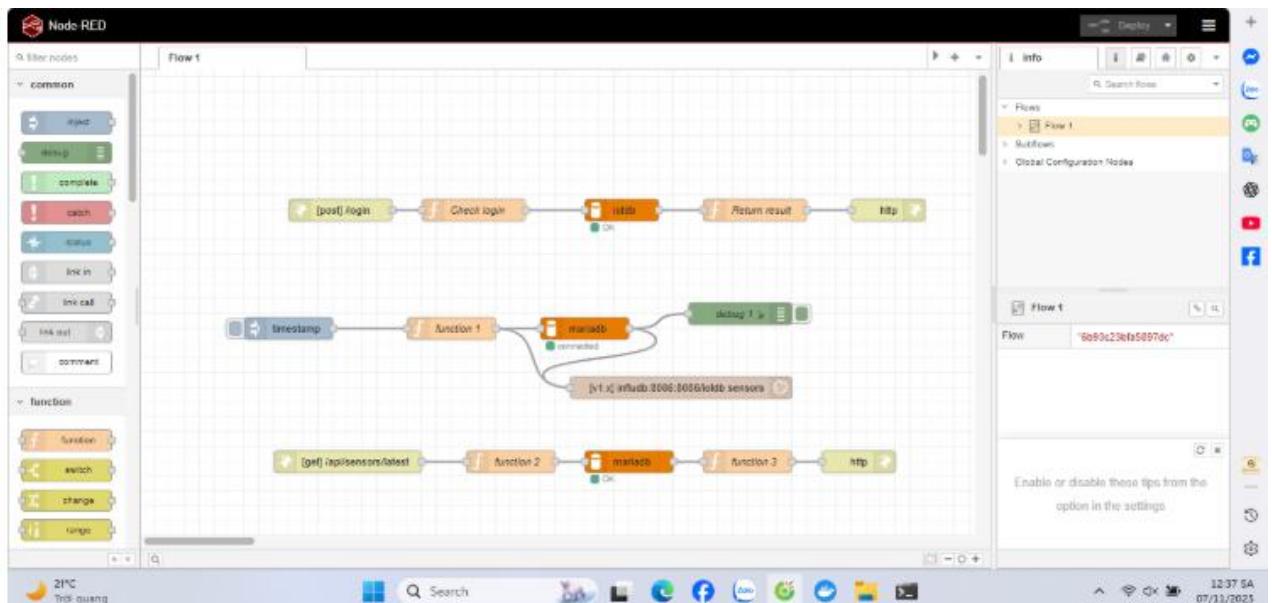
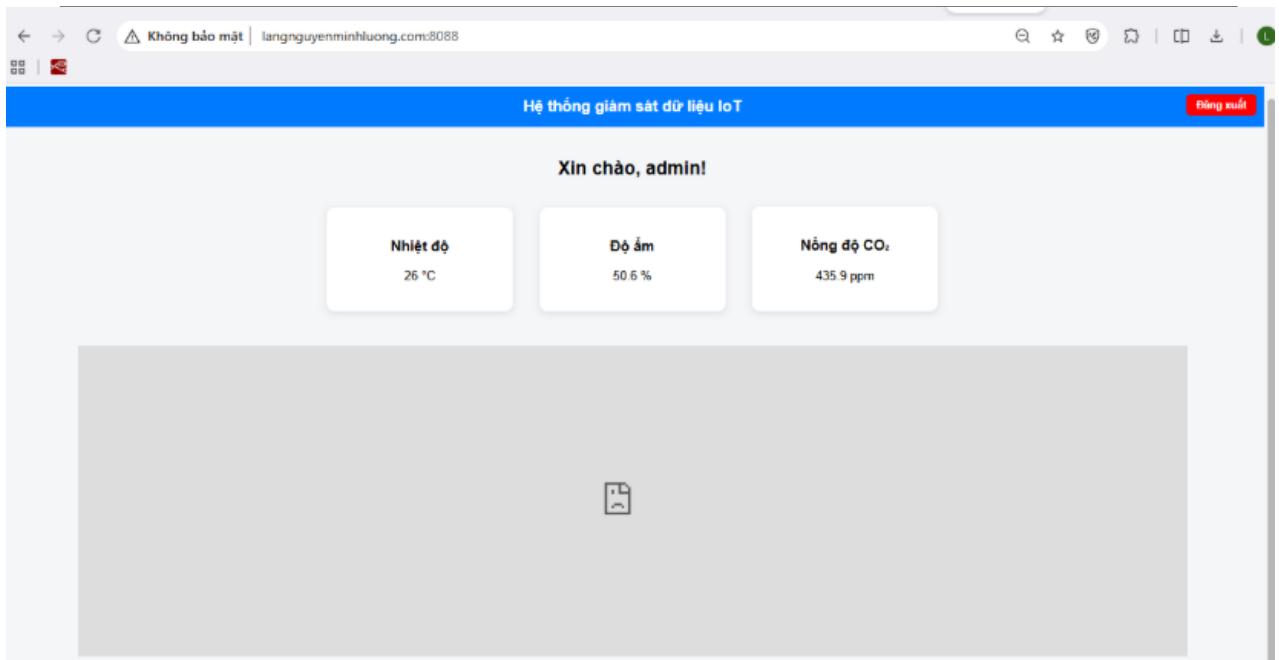


Bài tập lớn An Toàn Bảo Mật

The screenshot shows the phpMyAdmin interface for a MySQL database named 'iotdb'. The left sidebar lists databases: information_schema, iotdb, mysql, performance_schema, and sys. The 'iotdb' database is selected, showing tables: sensors and users. The 'sensors' table has 25 rows and the 'users' table has 4 rows. A modal window titled 'Tạo bảng mới' (Create new table) is open, prompting for 'Tên bảng' (Table name) and 'Số cột' (Number of columns), with a value of 4 entered.

The screenshot shows the Grafana home page. The left sidebar includes 'Home', 'Bookmarks', 'Started', 'Dashboards', 'Explore', 'Drilldown', 'Alerting', 'Connections', and 'Administration'. The main area displays a 'Welcome to Grafana' message with sections for 'Basic', 'Tutorial', 'Data source and dashboards', 'Grafana fundamentals', 'COMPLETE', 'Add your first data source', and 'COMPLETE', 'Create your first dashboard'. Below this, there's a 'Dashboards' section with 'Starred dashboards' and 'Recently viewed dashboards', and a 'Latest from the blog' section featuring an article about 'Grafana Mimir 3.0 release: performance improvements, a new query engine, and more'.

Bài tập lớn An Toàn Bảo Mật



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 và vận dụng thực tế các kiến thức quan trọng liên quan đến việc xây dựng, thiết kế và triển khai một ứng dụng web hoàn chỉnh. Từ những nội dung cơ bản như cấu trúc HTML, định dạng giao diện bằng CSS, lập trình tương tác với JavaScript, cho đến các nội dung nâng cao như xử lý dữ liệu phía server, kết nối cơ sở dữ liệu, xây dựng API và tổ chức kiến trúc ứng dụng web theo mô hình hiện đại, tất cả đã giúp em hiểu sâu hơn về quy trình phát triển một hệ thống web từ ý tưởng đến triển khai thực tế. Ngoài ra, việc thực hành với những bài tập và dự án như xây dựng trang quản lý sinh viên, hệ thống bán hàng trực tuyến hay trang thông tin cá nhân đã rèn luyện cho em kỹ năng tư duy lập trình, khả năng phân tích – thiết kế giao diện và làm việc với các công nghệ web phổ biến. Bài tập lớn không chỉ cung cấp kiến thức lý thuyết mà còn tạo điều kiện để em trau dồi kỹ năng thực hành, giúp em tự tin hơn khi phát triển các ứng dụng web trong học tập, nghiên cứu và công việc thực tế sau này.

Link Github của BTL :

https://github.com/Minhluong999/BTL_PTUDTNW

