

Live coding

1. Cài đặt đầy đủ các thành phần bao gồm Nginx, Python, MySQL.

Để cài đặt đầy đủ các thành phần Nginx, Python và MySQL trên Ubuntu, bạn có thể làm theo các bước dưới đây:

Bước 1: Cài đặt Nginx

1. Cập nhật danh sách gói:

```
sudo apt update
```

2. Cài đặt Nginx:

```
sudo apt install nginx
```

3. Khởi động và kiểm tra trạng thái Nginx:

```
sudo systemctl start nginx  
sudo systemctl enable nginx  
sudo systemctl status nginx
```

4. Mở cổng 80 và 443 trong tường lửa:

```
sudo ufw allow 'Nginx Full'
```

5. Kiểm tra Nginx: Mở trình duyệt và truy cập vào địa chỉ `http://<IP của máy chủ>` để kiểm tra xem Nginx đã hoạt động hay chưa.

Bước 2: Cài đặt Python

1. Cài đặt Python 3:

```
sudo apt install python3 python3-pip
```

2. Cài đặt virtualenv (nếu cần thiết):

```
sudo apt install python3-venv
```

3. Cập nhật pip (nếu cần thiết):

```
python3 -m pip install --upgrade pip
```

4. Kiểm tra phiên bản Python:

```
python3 --version
```

Bước 3: Cài đặt MySQL

1. Cài đặt MySQL Server:

```
sudo apt install mysql-server
```

2. Khởi động và kiểm tra trạng thái MySQL:

```
sudo systemctl start mysql  
sudo systemctl enable mysql
```

```
sudo systemctl status mysql
```

3. Cấu hình MySQL:

- Chạy lệnh cấu hình bảo mật MySQL:

```
sudo mysql_secure_installation
```

Thiết lập mật khẩu root và các tùy chọn bảo mật khác.

4. Đăng nhập vào MySQL:

```
sudo mysql -u root -p
```

5. Kiểm tra phiên bản MySQL:

```
mysql --version
```

2. Tạo cơ sở dữ liệu và bảng

- Tạo cơ sở dữ liệu:

```
CREATE DATABASE db_yourname;
```

- Tạo bảng :

```
USE db_yourname;  
CREATE TABLE tb_people (  
  MaSo INT PRIMARY KEY,  
  HoTen VARCHAR(50),
```

```
SoDienThoai VARCHAR(20),  
Diem INT );
```

- Chèn dữ liệu:

```
INSERT INTO tb_people (MaSo, HoTen, SoDienThoai, Diem) VALUES  
(1, 'Nguyen Van A', '0999888777', 9),  
(2, 'Vu Van B', '0999888666', 7),  
(3, 'Bui Thi C', '0999888000', 10);
```

3. Thiết lập kết nối python với mysql

Chúng ta phải cài đặt thư viện mysql-connector, cài đặt trong môi trường ảo :

- Thiết lập môi trường ảo

```
source flask_env/bin/activate
```

- Cài đặt thư viện

```
pip install flask mysql-connector-python
```

Dùng code này được thực hiện trong trình soạn thảo nano như sau : `nano app.py`

```
import mysql.connector  
  
# Thông tin kết nối  
host = 'localhost'      # Địa chỉ của MySQL server  
user = 'root'           # Tên người dùng MySQL  
password = 'Phuc00000@' # Mật khẩu người dùng
```

```
database = 'danhsach' # Tên cơ sở dữ liệu muốn kết nối
```

```
# Kết nối đến MySQL
```

```
try:
```

```
    db_connection = mysql.connector.connect(  
        host=host,  
        user=user,  
        password=password,  
        database=database,  
        ssl_disabled=True # Nếu cần thiết, có thể bỏ dòng này  
    )
```

```
print("Kết nối thành công!")
```

```
# Tạo đối tượng cursor để thực hiện truy vấn
```

```
cursor = db_connection.cursor()
```

```
# Ví dụ truy vấn
```

```
cursor.execute("SELECT * FROM bangthongtin")
```

```
# Lấy kết quả truy vấn
```

```
rows = cursor.fetchall()
```

```
if rows:
```

```
    for row in rows:
```

```
        print(row)
```

```
else:
```

```
    print("Không có dữ liệu.")
```

```
except mysql.connector.Error as err:
```

```
    print(f"Lỗi kết nối: {err}")
```

```
finally:
```

```
    # Đảm bảo đóng kết nối khi kết thúc
```

```
    if 'cursor' in locals() and cursor:
```

```
cursor.close()
if 'db_connection' in locals() and db_connection.is_connected():
    db_connection.close()
print("Đã đóng kết nối.")
```

Câu lệnh để thực hiện:

```
python3 app.py
```

- Thoát môi trường ảo

```
deactivate
```

4. Thiết lập kết nối Nginx và Python server, hiển thị thông tin thí sinh khi truy cập Nginx.

1. Chúng ta phải cài đặt cấu hình cho Nginx

Mở tệp cấu hình Nginx:

```
sudo nano /etc/nginx/sites-available/default
```

Trong tệp cấu hình, bạn thêm hoặc sửa lại phần sau để định tuyến các yêu cầu đến Python server:

```
server {
    listen 80;
    server_name localhost;

    location / {
        proxy_pass http://127.0.0.1:8080; # Chuyển tiếp yêu cầu đến Python server
    }
}
```

```
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;
}
}
```

Ý nghĩa của các trường proxy

1. proxy_set_header Host \$host;

- **Ý nghĩa:** Truyền lại header `Host` từ yêu cầu ban đầu của client.
- **Mục đích:** Đảm bảo server backend biết chính xác tên miền hoặc IP mà client đã yêu cầu (ví dụ: `example.com` hoặc `127.0.0.1`).

2. proxy_set_header X-Real-IP \$remote_addr;

- **Ý nghĩa:** Truyền lại địa chỉ IP thực của client.
- **Mục đích:** Giúp backend biết được địa chỉ IP gốc của client thay vì chỉ thấy địa chỉ IP của Nginx.

3. proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;

- **Ý nghĩa:** Truyền lại chuỗi IP của client và các proxy đã đi qua.
- **Mục đích:** Cung cấp thông tin về hành trình mà yêu cầu đã đi qua, hữu ích cho việc ghi nhận và xử lý dữ liệu từ các proxy trung gian.

4. proxy_set_header X-Forwarded-Proto \$scheme;

- **Ý nghĩa:** Truyền lại giao thức (HTTP/HTTPS) mà client sử dụng khi gửi yêu cầu.
- **Mục đích:** Giúp backend xác định giao thức gốc của yêu cầu (ví dụ: nếu yêu cầu được gửi qua HTTP nhưng backend cần biết client đã dùng HTTPS).

2. Kiểm tra cấu hình và khởi động lại Nginx:

- Kiểm tra xem cấu hình Nginx có chính xác không:

```
sudo nginx -t
```

- Khởi động lại Nginx để áp dụng thay đổi:

```
sudo systemctl restart nginx
```

3. Chạy chương trình

- Thiết lập môi trường ảo :

```
source flask_env/bin/activate
```

- Dùng code này được thực hiện trong trình soạn thảo nano như sau : `nano wep_app.py`

```
import mysql.connector
from http.server import BaseHTTPRequestHandler, HTTPServer

# Thông tin kết nối MySQL
host = 'localhost'
user = 'root'
password = 'Phuc00000@'
database = 'danhsach'

class MyHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        # Kết nối với MySQL
        try:
            db_connection = mysql.connector.connect(
                host=host,
                user=user,
                password=password,
                database=database
            )
```



```

cursor = db_connection.cursor()
cursor.execute("SELECT * FROM bangdanhsach") # Thực hiện truy vấn

# Lấy kết quả truy vấn
rows = cursor.fetchall()

# Cấu trúc HTML hiển thị dữ liệu

html = "<html><head><meta charset='UTF-8'></head><body>"
html += "<h1>Thông Tin Thí Sinh</h1><table border='1'>"
html += "<tr><th>ID</th><th>Name</th><th>Age</th><th>Score</th></tr>"

for row in rows:
    html += f"<tr><td>{row[0]}</td><td>{row[1]}</td><td>{row[2]}</td><td>{row[3]}</td></tr>"

html += "</table></body></html>"

# Gửi phản hồi về trình duyệt
self.send_response(200)
self.send_header('Content-type', 'text/html; charset=UTF-8') # Đặt charset UTF-8
self.send_header('Server', 'Nguyen Van A')
self.end_headers()
self.wfile.write(html.encode('utf-8')) # Mã hóa HTML bằng UTF-8

cursor.close()
db_connection.close()
except mysql.connector.Error as err:
    self.send_response(500)
    self.send_header('Content-type', 'text/html; charset=UTF-8') # Đặt charset UTF-8 cho lỗi
    self.end_headers()
    self.wfile.write(f"<html><body><h1>Error: {err}</h1></body></html>".encode('utf-8'))

# Cấu hình server
def run(server_class=HTTPServer, handler_class=MyHandler, port=8080):

```

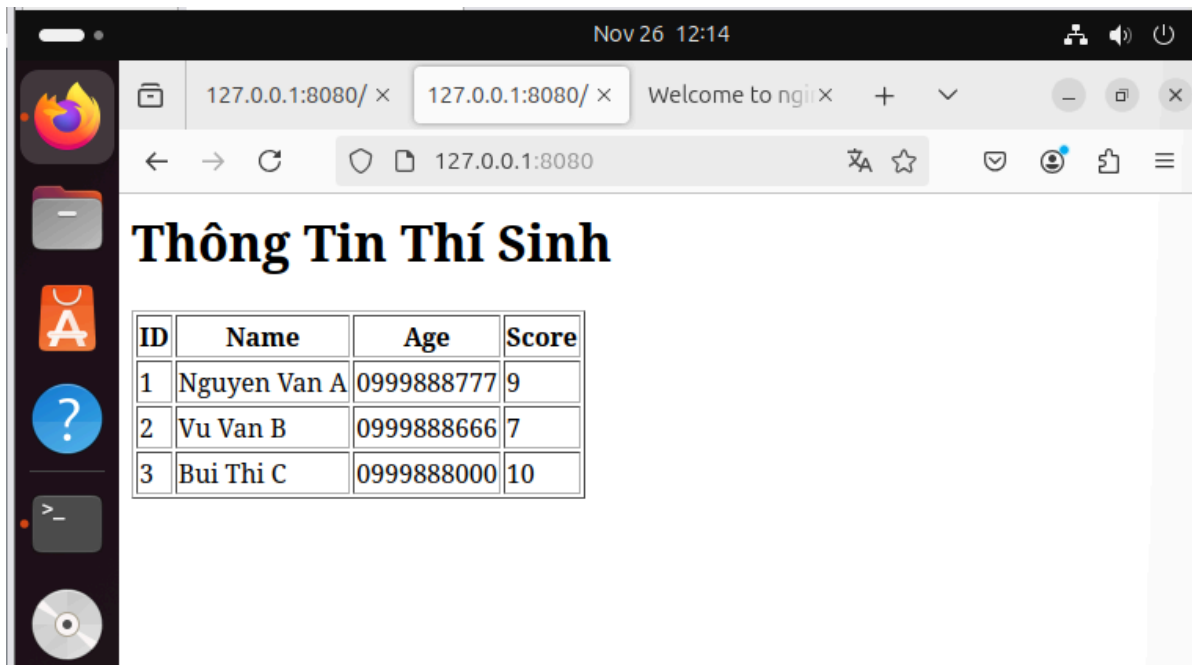
```
server_address = ("", port)
httpd = server_class(server_address, handler_class)
print(f"Starting server on port {port}")
httpd.serve_forever()

if __name__ == "__main__":
    run()
```

- Thoát môi trường ảo

deactivate

Sau khi thực hiện câu lệnh hãy truy cập `http://127.0.0.1:8080/` để kiểm tra kết quả



5. Cấu hình để sửa giá trị HTTP Response Header "Server" trả về từ server thành tên thí sinh

Để thực hiện trên Nginx chúng ta phải thực hiện các bước sau:

Bước 1:

- Mở file `/etc/nginx/sites-available/default` bằng lệnh

```
sudo nano /etc/nginx/sites-available/default
```

- Thêm hoặc thay đổi như sau:

```
# Server block mặc định
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # Đặt tên server trong header HTTP (sử dụng module headers_more)
    more_set_headers "Server: Nguyen Van A"; # Sử dụng module headers_more

    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        try_files $uri $uri/ =404;
    }
}

# Proxy server
```

```
server {
    listen 80;
    server_name localhost;

    # Đặt tên server trong header HTTP (sử dụng module headers_more)
    more_set_headers "Server: Nguyen Van A"; # Sử dụng module headers_more

    location / {
        proxy_pass http://127.0.0.1:8000;
        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;
    }
}

# Chuyển hướng từ HTTP sang HTTPS
server {
    listen 80;
    server_name 192.168.248.131;

    return 301 https://$host$request_uri;
}

# Server HTTPS
server {
    listen 443 ssl;
    server_name your_domain_or_ip;

    # SSL Configuration
    include snippets/self-signed.conf;
    include snippets/ssl-params.conf;

    # Đặt tên server trong header HTTP (sử dụng module headers_more)
```

```
more_set_headers "Server: Nguyen Van A"; # Sử dụng module headers_more
```

```
root /var/www/html;  
index index.html index.htm index.nginx-debian.html;
```

```
location / {  
    try_files $uri $uri/ =404;  
}  
}
```

Bước 2: Cài đặt headers_more trên Ubuntu:

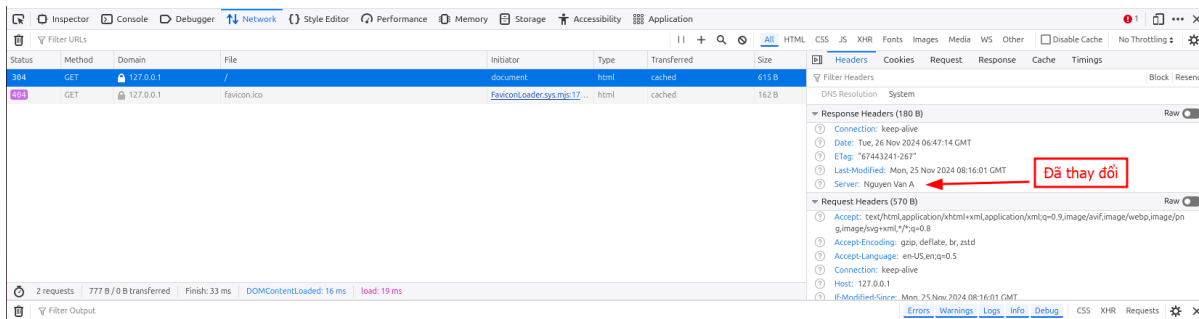
```
sudo apt-get install nginx-extras
```

Bước 3: Kiểm tra cấu hình và khởi động lại Nginx

```
# Kiểm tra cú pháp cấu hình Nginx  
sudo nginx -t
```

```
# Nếu không có lỗi, khởi động lại Nginx  
sudo systemctl restart nginx
```

Kết quả khi truy cập <http://127.0.0.1/>:



6. Thiết lập chứng chỉ SSL tự ký cho Nginx và tự động điều hướng tới HTTPS khi truy cập HTTP

Để thiết lập chứng chỉ SSL tự ký cho Nginx và tự động điều hướng từ HTTP sang HTTPS, bạn làm theo các bước sau:

Bước 1: Tạo chứng chỉ SSL tự ký

1. Tạo thư mục lưu chứng chỉ SSL:

```
sudo mkdir /etc/nginx/ssl
```

2. Tạo chứng chỉ SSL tự ký:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \  
-keyout /etc/nginx/ssl/nginx-selfsigned.key \  
-out /etc/nginx/ssl/nginx-selfsigned.crt
```

Bạn sẽ được yêu cầu nhập thông tin chứng chỉ. Điền như sau:

Country **Name** (2 letter code) [XX]:VN
State or Province **Name** (full name) []:Your Province
Locality **Name** (eg, city) [Default City]:Your City
Organization **Name** (eg, company) [Default Company Ltd]:Your Organization
Organizational Unit **Name** (eg, section) []:Your Department
Common **Name** (eg, your name or your server's hostname) []:Your Domain/IP
Email Address []:Your Email

Thêm nội dung sau:

```
ssl_certificate /etc/nginx/ssl/nginx-selfsigned.crt;  
ssl_certificate_key /etc/nginx/ssl/nginx-selfsigned.key;
```

2. Tạo tệp cấu hình các tham số SSL:

```
sudo nano /etc/nginx/snippets/ssl-params.conf
```

Thêm nội dung sau:

```
ssl_protocols TLSv1.2 TLSv1.3;  
ssl_prefer_server_ciphers on;  
ssl_ciphers HIGH:!aNULL:!MD5;  
ssl_dhparam /etc/nginx/ssl/dhparam.pem;  
ssl_session_timeout 1d;  
ssl_session_cache shared:SSL:10m;  
ssl_session_tickets off;  
ssl_stapling on;  
ssl_stapling_verify on;  
resolver 8.8.8.8 8.8.4.4 valid=300s;  
resolver_timeout 5s;  
add_header X-Frame-Options DENY;  
add_header X-Content-Type-Options nosniff;  
add_header X-XSS-Protection "1; mode=block";
```

Bước 3: Cập nhật cấu hình Nginx

1. Sửa file cấu hình Nginx:


```
sudo nano /etc/nginx/sites-available/default
```

Thêm/chỉnh sửa nội dung như sau:

```
server {  
    listen 80;  
    server_name your_domain_or_ip;  
  
    # Tự động chuyển hướng sang HTTPS  
    return 301 https://$host$request_uri;  
}  
  
server {  
    listen 443 ssl;  
    server_name 192.168.248.131;  
  
    include snippets/self-signed.conf;  
    include snippets/ssl-params.conf;  
  
    root /var/www/html;  
    index index.html index.htm index.nginx-debian.html;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

Giải thích :

```
return 301 https://$host$request_uri;
```

1. \$host:

- Đây là một **biến trong Nginx**, đại diện cho **tên miền (domain)** hoặc **địa chỉ IP** trong yêu cầu HTTP mà máy chủ nhận được.

2. \$request_uri:

- Đây là một **biến trong Nginx**, đại diện cho phần **URI** của yêu cầu, bao gồm cả đường dẫn và chuỗi truy vấn (query string).

3. https://*host**request_uri*:

- Kết hợp cả domain/IP (`$host`) và đường dẫn URI (`$request_uri`), chuyển hướng toàn bộ yêu cầu từ HTTP sang HTTPS mà không làm mất thông tin.

Bước 4: Kiểm tra và khởi động lại Nginx

1. Kiểm tra tính hợp lệ của cấu hình Nginx:

```
sudo nginx -t
```

2. Tải lại cấu hình Nginx:

```
sudo systemctl reload nginx
```

Lệnh reload :

```
sudo systemctl reload nginx
```

- Tải lại cấu hình mà không làm gián đoạn dịch vụ hiện tại hoặc các phiên đang hoạt động.

Lệnh restart :

```
sudo systemctl restart nginx
```

- Dừng và khởi động lại Nginx, làm gián đoạn tất cả các phiên kết nối hiện tại.

Bước 5: Kiểm tra kết quả

- Mở trình duyệt và truy cập vào địa chỉ `http://<IP>` hoặc `http://<domain>` , bạn sẽ được tự động điều hướng sang `https://<IP>` hoặc `https://<domain>` .
- Trình duyệt sẽ cảnh báo "Kết nối không an toàn" vì chứng chỉ tự ký. Bạn có thể chọn "Tiếp tục" để truy cập.

