LAB04: Frontend & Backend

This lab exercise is designed to help students understand how to prepare a server environment to support a web system that uses Nginx as the web server and FastAPI (Python framework) as the backend. Students will also create a simple HTML frontend and a simple Python FastAPI backend to see the overall architecture and data flow between components.

# Objectives

- Learn how to install and configure Nginx as a web server.  
- Learn how to set up a Python virtual environment and install FastAPI.  
- Practice writing a simple HTML frontend page.  
- Practice writing a simple FastAPI backend service.  
- Understand how Nginx proxies requests to the backend.

# Prerequisites

- A Linux-based server (Ubuntu/Debian preferred).  
- Basic knowledge of command line usage.  
- Python 3 installed on the server.  
- Internet connection to install packages.

# Step 1: Install Nginx

Run the following commands to install and start Nginx:  
  
*sudo apt update  
sudo apt install -y nginx  
sudo systemctl enable --now nginx*Verify installation by opening http://<server-ip> in your browser. You should see the Nginx welcome page.

# Step 2: Create a Simple HTML Frontend

Create an HTML file to serve as the frontend page. Place it under Nginx's default directory:  
  
*sudo nano /var/www/html/index.html*

<!DOCTYPE html>  
<html>  
<head>  
 <title>FastAPI Demo</title>  
</head>  
<body>  
 <h1>Hello from Frontend!</h1>  
 <p>Click the button to call FastAPI backend:</p>  
 <button onclick="fetch('/api/hello').then(r => r.json()).then(d => alert(d.message))">Call Backend</button>  
</body>  
</html>

# Step 3: Prepare Python Environment

Set up a directory for the FastAPI app and a Python virtual environment:

*sudo mkdir -p /srv/fastapi/app  
sudo chown -R $USER:$USER /srv/fastapi  
cd /srv/fastapi/app  
python3 -m venv .venv  
source .venv/bin/activate*

*(.evnv) $ pip install fastapi "uvicorn[standard]"*

*(.evnv) $ deactivate*

# Step 4: Write FastAPI Backend

Create a Python file `main.py` inside `/srv/fastapi/app/`:

from fastapi import FastAPI, Request

app = FastAPI()

# API แรก: hello world

@app.get("/")

def read\_root():

return {"message": "Hello from FastAPI!"}

# API ที่มีพารามิเตอร์

@app.get("/hello")

def hello(request: Request):

client\_ip = request.headers.get("x-real-ip")

return {"message": f"Hello, your IP is {client\_ip}"}

Run the FastAPI backend using uvicorn:

*source .venv/bin/activate*

*uvicorn main:app --host 127.0.0.1 --port 8000*

# Step 5: Configure Nginx as Reverse Proxy

Tell Nginx to forward requests from /api to the FastAPI backend:  
*sudo nano /etc/nginx/sites-available/web02*

server {  
 listen 80;  
 server\_name \_;  
 root /var/www/html;  
 index.html;  
 location /api/ {  
 proxy\_pass http://127.0.0.1:8000/;  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 }  
}  
  
Enable the site and reload Nginx:  
*sudo ln -sf /etc/nginx/sites-available/web02 /etc/nginx/sites-enabled/web02  
sudo nginx -t  
sudo systemctl reload nginx*

# Step 6: Test the Whole System

1. Open http://<server-ip>/ in your browser. You will see the HTML frontend.  
2. Click the button. The frontend will call the FastAPI backend via Nginx reverse proxy.  
3. You should see an alert box showing the message returned from FastAPI.