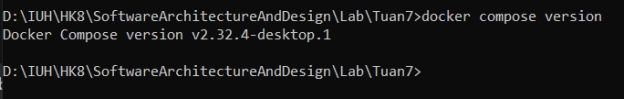
**Họ tên: Trần Ngọc Hưng**

**MSSV: 22711231**

**Phần 1: Một số lệnh Docker Compose cơ bản**

1. **docker compose version**

Kiểm tra phiên bản của Docker Compose đang được cài đặt.

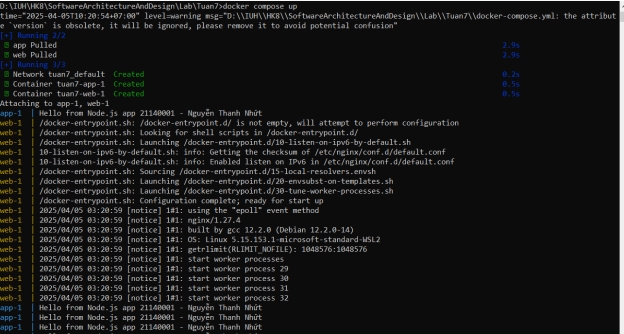
****

1. **docker compose up**

Khởi động và tạo các container dựa trên file docker-compose.yml.

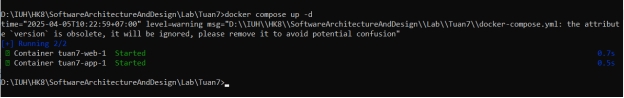
Tạo file docker-compose.yml

****

****

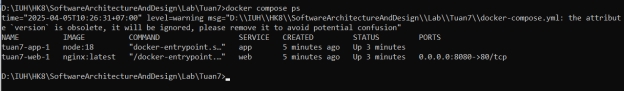
1. **docker compose up -d**

Khởi động container ở chế độ detached (nền).

****

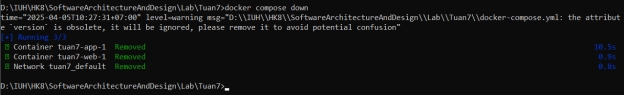
1. **docker compose ps**

Hiển thị danh sách các container đang chạy trong project Docker Compose.

****

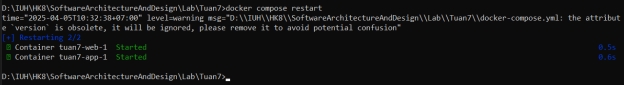
1. **docker compose down**

Dừng và xóa toàn bộ container, mạng, và volumes tạm thời được tạo bởi docker-compose up.

****

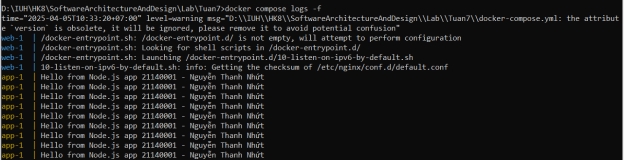
1. **docker compose restart**

Khởi động lại tất cả các container trong project.

****

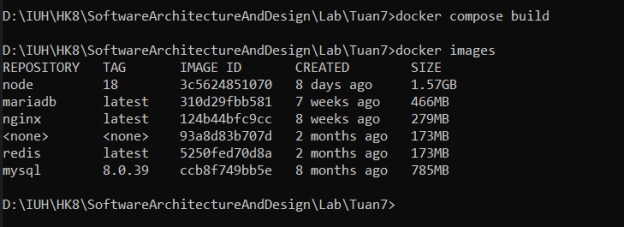
1. **docker compose logs -f**

Theo dõi log của các container trong Docker Compose (tương tự tail -f)

****

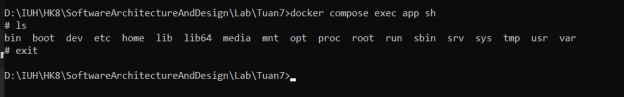
1. **docker compose build**

Build lại image từ Dockerfile trong docker-compose.yml.

****

1. **docker compose exec <service\_name> <command>**

Chạy lệnh trong container đang chạy.

****

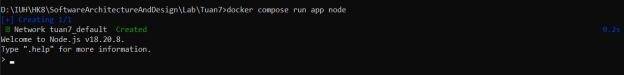
1. **docker compose down -v**

Xóa container và cả volume liên quan.

****

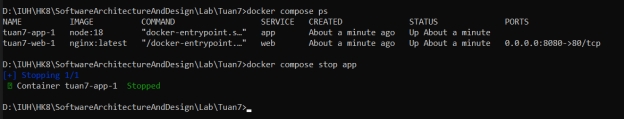
1. **docker compose run <service\_name> <command>**

Tạo container tạm thời để chạy lệnh.

****

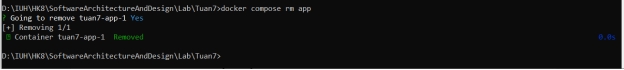
1. **docker compose stop <service\_name>**

Dừng container theo tên service.

****

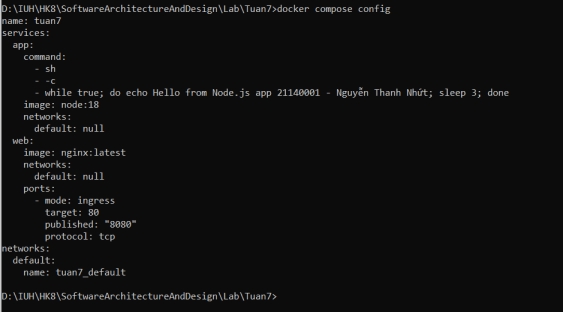
1. **docker compose rm <service\_name>**

Xóa container đã dừng của một service cụ thể.

****

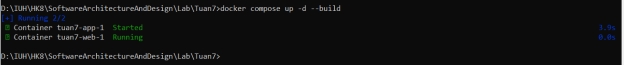
1. **docker compose config**

Kiểm tra file docker-compose.yml sau khi đã được merge, validate cấu trúc.

****

1. **docker compose up -d –build**

Build lại image và chạy container trong chế độ detached.

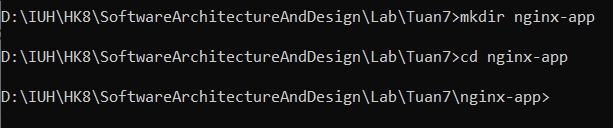
****

**Phần 2: Docker Compose file**

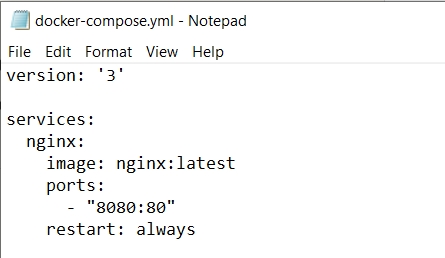
**Bài 1: Chạy một container đơn giản với Docker Compose**

Yêu cầu: Tạo một container chạy Nginx bằng Docker Compose. Map cổng 8080 của máy host với cổng 80 của container.

**Tạo thư mục dự án**

****

**Tạo file docker-compose.yml**

****

**Khởi động và tạo các container dựa trên file docker-compose.yml.**

****

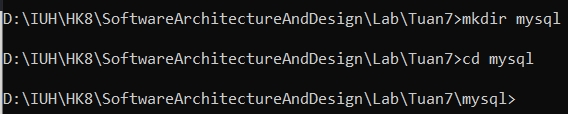
**Truy cập** [**http://localhost:8080**](http://localhost:8080)

****

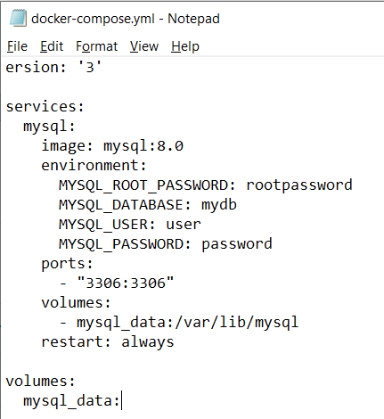
**Bài 2: Chạy MySQL với Docker Compose**

Yêu cầu: Tạo một container chạy MySQL phiên bản 8.0. Đặt username là user, password là password và database là mydb

**Tạo thư mục dự án**

****

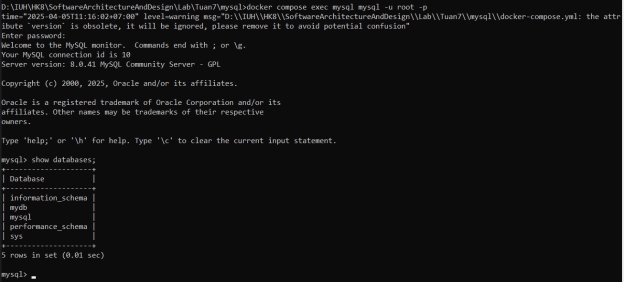
**Tạo file docker-compose.yml**

****

**Khởi động và tạo các container dựa trên file docker-compose.yml.**

****

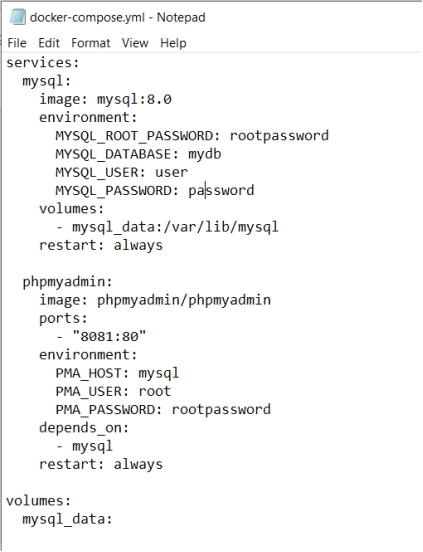
**Kết nối MySQL**

****

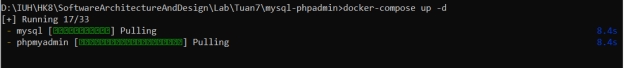
**Bài 3: Kết nối MySQL với PHPMyAdmin**

Yêu cầu: Chạy MySQL và PHPMyAdmin với Docker Compose. PHPMyAdmin chạy trên cổng 8081.

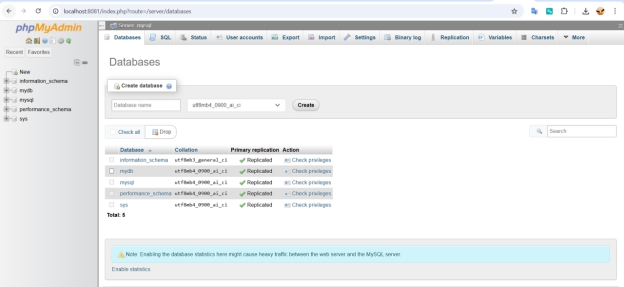
**Tạo file docker-compose.yml**

****

**Chạy lệnh docker-compose up -d**

****

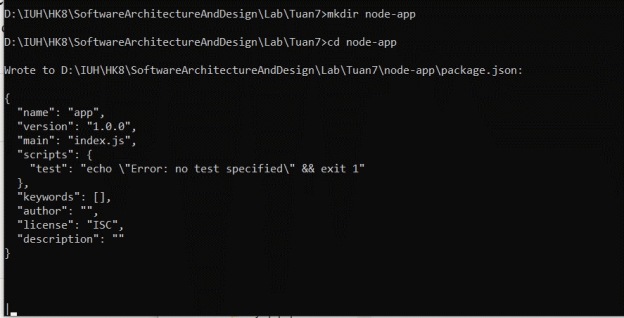
**Truy cập PHPMyAdmin qua** [**http://localhost:8081**](http://localhost:8081)

****

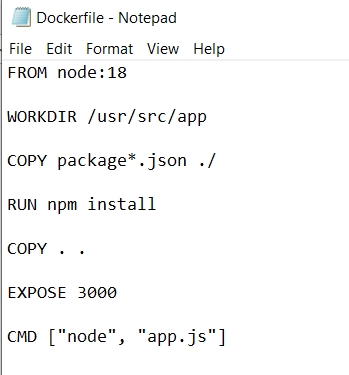
**Bài 4: Chạy ứng dụng Node.js với Docker Compose**

Yêu cầu: Chạy một ứng dụng Node.js đơn giản với Express.

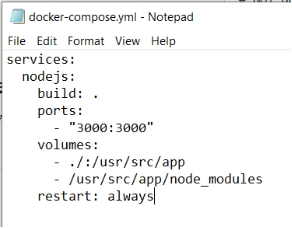
**Tạo thư mục project**

****

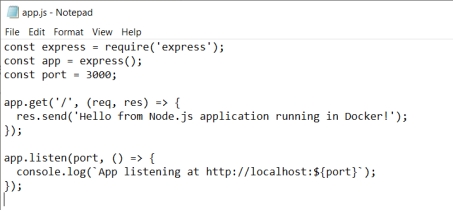
**Tạo Dockerfile**

****

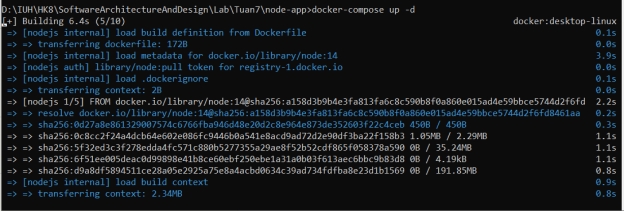
**Tạo file docker-compose.yml**

****

**File app.js**

****

**Chạy file docker compose**

****

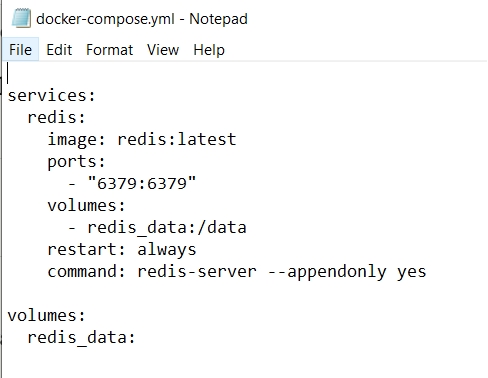
**Truy cập** [**http://localhost:3000**](http://localhost:3000) **để xem kết quả**

****

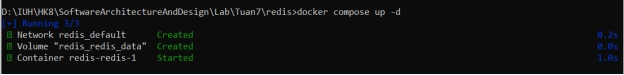
**Bài 5: Chạy Redis với Docker Compose**

Yêu cầu: Chạy một container Redis trên cổng 6379.

**Tạo file docker-compose.yml**

****

**Chạy lệnh docker-compose up -d**

****

**Kết nối Redis bằng Redis CLI:**

****

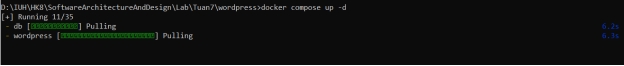
**Bài 6: Chạy WordPress với MySQL**

Yêu cầu: Chạy WordPress với MySQL bằng Docker Compose.

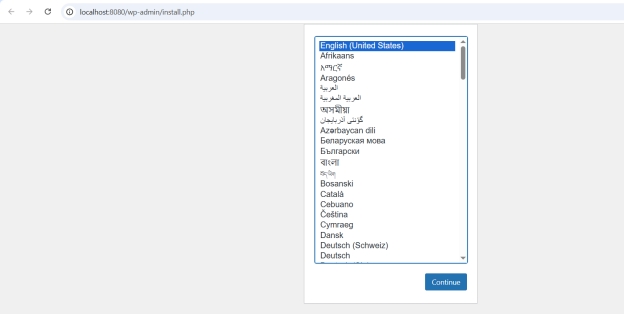
**Tạo file docker-compose.yml**

****

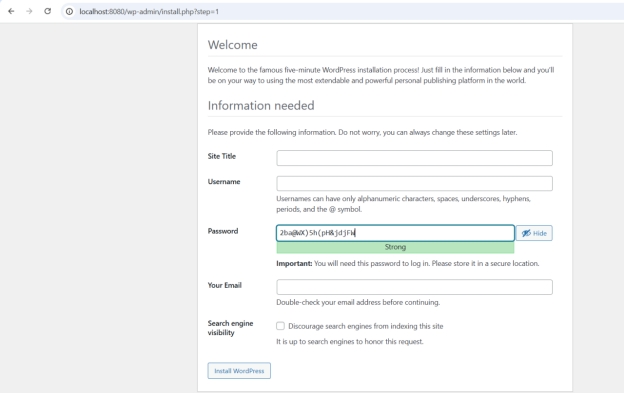
**Chạy lệnh docker-compose up -d**

****

**Truy cập WordPress tại** [**http://localhost:8080**](http://localhost:8080)

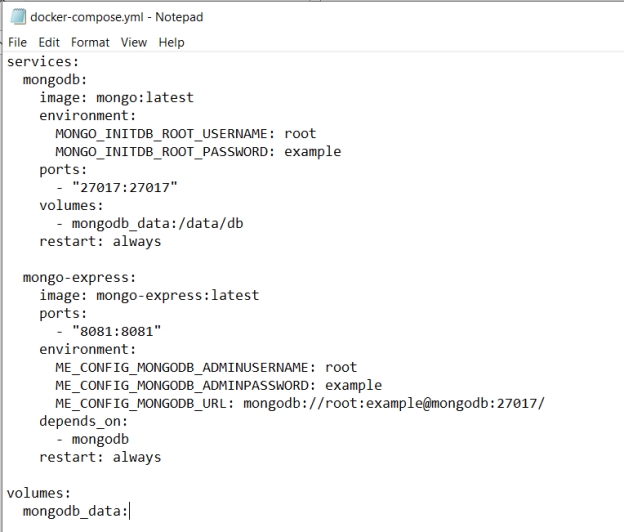
****

**Làm theo hướng dẫn**

****

**Bài 7: Chạy MongoDB với Docker Compose**

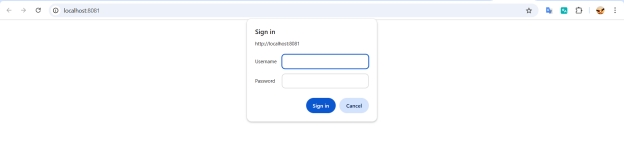
**Tạo file docker-compose.yml**

****

**Chạy lệnh docker-compose up -d**

****

**Truy cập Mongo Express tại** [**http://localhost:8081**](http://localhost:8081)

****

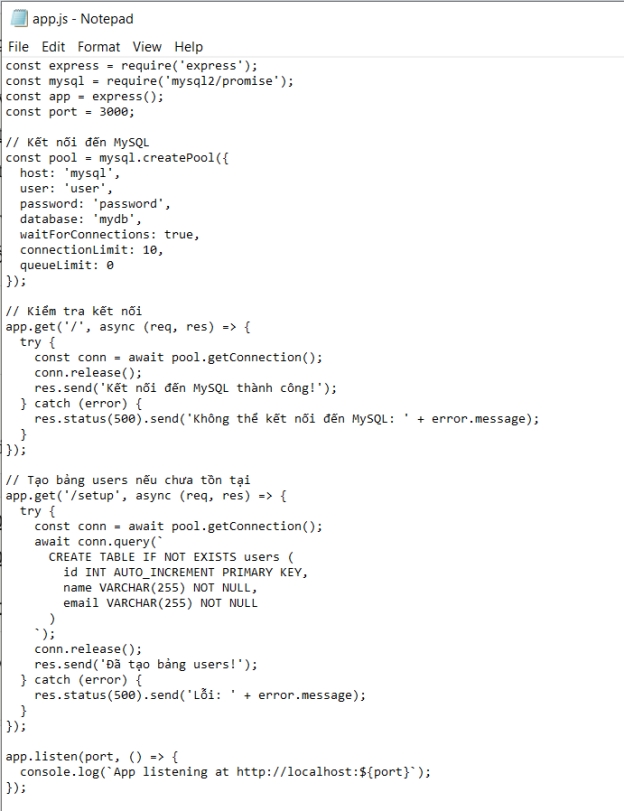
**Đăng nhập với username: root, password: example**

**Bài 8: Kết nối nhiều dịch vụ với Docker Compose**

Yêu cầu: Chạy Node.js kết nối với MySQL.

**Tạo thư mục nodejs-mysql với các file như sau:**

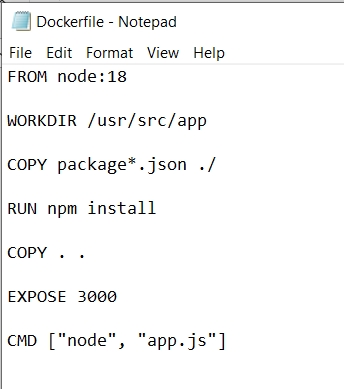
**File app.js**

****

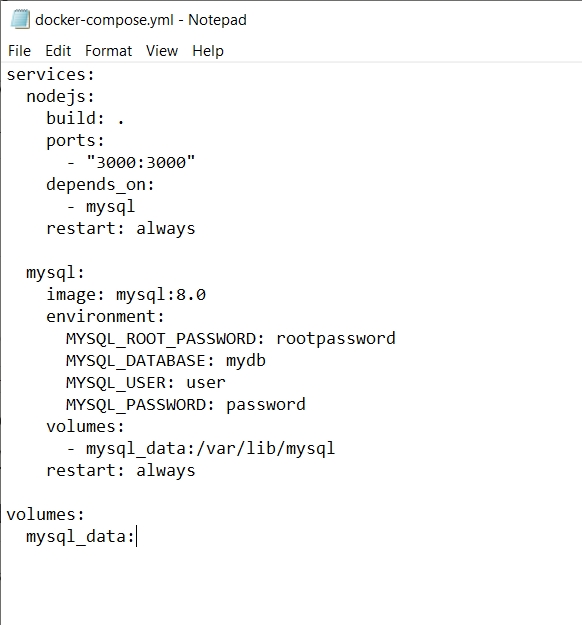
**Package.json**

****

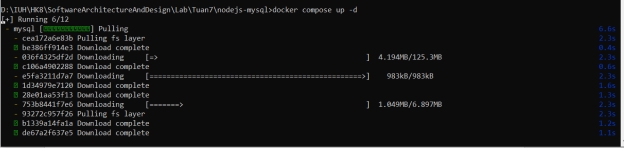
**Dockerfile**

****

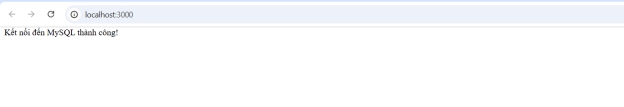
**File docker-compose.yml**

****

**Chạy lệnh docker-compose up -d**

****

**Truy cập** [**http://localhost:3000**](http://localhost:3000) **để kiểm tra kết nối**

****

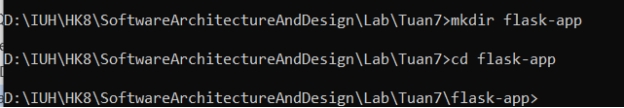
**Truy cập** [**http://localhost:3000/setup**](http://localhost:3000/setup) **để tạo bảng users**

****

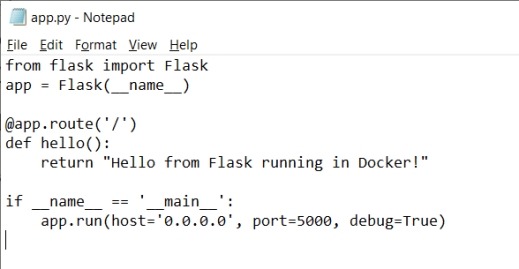
**Bài 9: Chạy ứng dụng Python Flask với Docker Compose**

Yêu cầu: Chạy ứng dụng Flask đơn giản với Docker Compose.

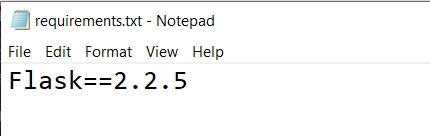
**Tạo thư mục flask-app và các file như sau:**

****

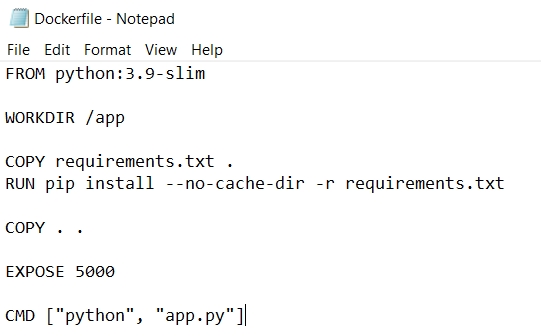
**Nội dung file app.py**

****

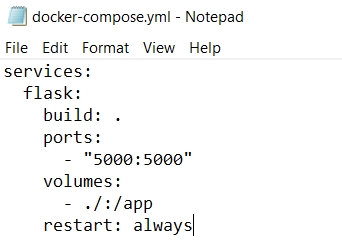
**Nội dung file requirements.txt**

****

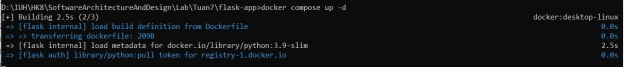
**Nội dung file Dockerfile**

****

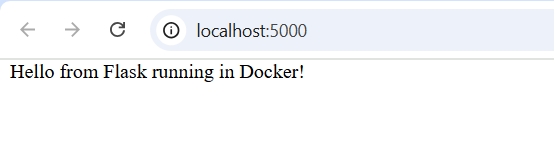
**Nội dung file docker-compose.yml**

****

**Chạy lệnh docker-compose up -d**

****

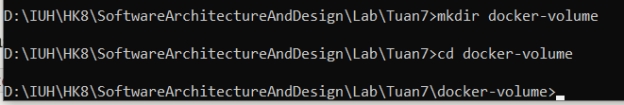
**Truy cập** [**http://localhost:5000**](http://localhost:5000) **để thấy kết quả**

****

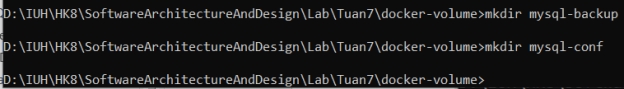
**Bài 10: Lưu trữ dữ liệu với Docker Volumes**

Yêu cầu: Chạy MySQL và gắn volume để dữ liệu không bị mất.

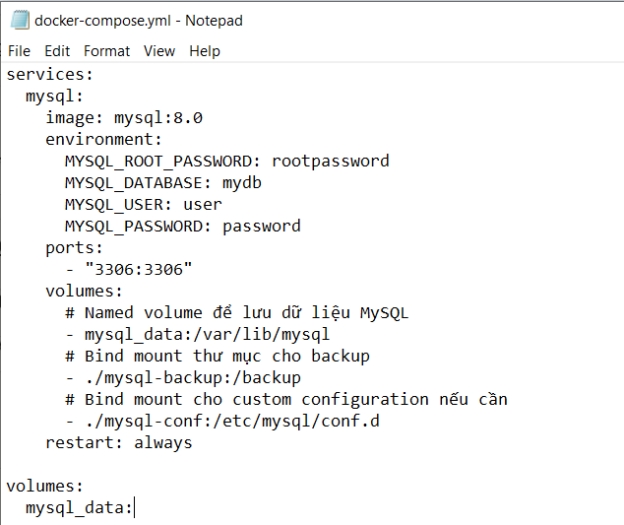
**Tạo thư mục cho project**

****

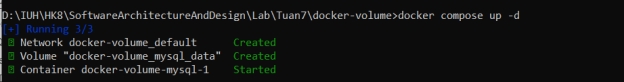
**Tạo thư mục con mysql-backup và mysql-conf trong thư mục project**

****

**Tạo file docker-compose.yml**

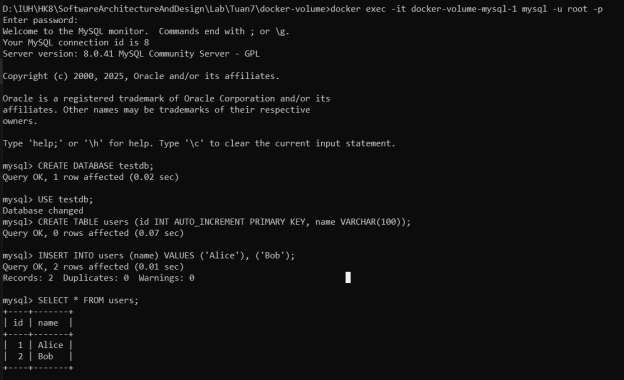
****

**Chạy docker-compose up -d**

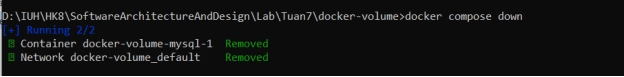
****

**Để kiểm tra volume hoạt động:**

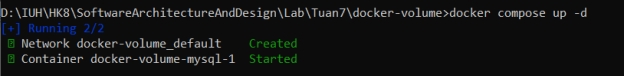
* Tạo dữ liệu trong database

****

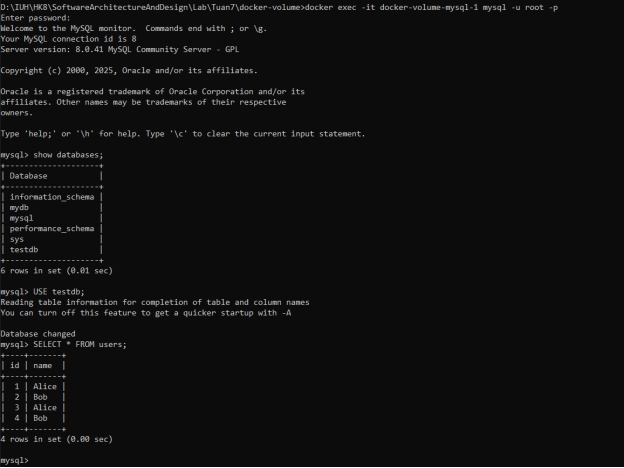
* Dừng và xóa container: docker-compose down

****

* Khởi động lại: docker-compose up -d

****

* Dữ liệu sẽ vẫn còn vì được lưu trong volume

****