BÁO CÁO CHI TIẾT VỀ VIỆC SỬ DỤNG DOCKER TRIỂN KHAI BACKEND NODE.JS CHO TRANG WEB BÁN HÀNG

# 1. Mục tiêu

Trong dự án xây dựng trang web bán hàng, backend đóng vai trò xử lý logic, tương tác với cơ sở dữ liệu, xác thực người dùng và cung cấp API cho frontend. Việc sử dụng Docker giúp đơn giản hóa quá trình triển khai backend Node.js bằng cách đóng gói toàn bộ mã nguồn, thư viện và cấu hình vào một container. Điều này đảm bảo ứng dụng có thể chạy ổn định trên mọi máy, bất kể hệ điều hành hay môi trường cài đặt.

# 2. Giới thiệu công nghệ sử dụng

- Node.js: Nền tảng backend sử dụng JavaScript, nổi bật với hiệu năng cao và cộng đồng mạnh mẽ.

- Express.js: Framework nhẹ của Node.js giúp xây dựng API RESTful nhanh chóng.

- MongoDB: Cơ sở dữ liệu NoSQL dùng để lưu trữ thông tin sản phẩm, người dùng, đơn hàng,...

- Docker: Nền tảng dùng để đóng gói ứng dụng vào các container, giúp triển khai nhất quán và dễ mở rộng.

# 3. Cấu trúc thư mục dự án

/backend

├── src/ # Chứa mã nguồn (controllers, routes, models, ...)

├── package.json # Danh sách dependencies và script

├── Dockerfile # File cấu hình để build Docker image

├── .dockerignore # Các file/thư mục không copy vào image

├── docker-compose.yml # Khởi tạo nhiều container cùng lúc

# 4. Nội dung Dockerfile và giải thích

```Dockerfile

FROM node:18

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["npm", "start"]

```

- WORKDIR /app: Mọi thao tác sau đó sẽ thực hiện trong thư mục /app của container.

- COPY + RUN npm install: Đảm bảo chỉ cài đặt lại khi package.json thay đổi, giúp tối ưu hóa việc build.

- CMD: Chạy lệnh mặc định để khởi động server Node.js.

# 5. Cấu hình docker-compose.yml và giải thích

```yaml

version: '3'

services:

backend:

build: .

ports:

- "3000:3000"

depends\_on:

- mongo

environment:

- MONGO\_URL=mongodb://mongo:27017/shopdb

mongo:

image: mongo:latest

container\_name: mongo

volumes:

- mongo-data:/data/db

volumes:

mongo-data:

- backend: Dịch vụ Node.js, được build từ Dockerfile, expose cổng 3000.

- mongo: Dịch vụ MongoDB, dùng image chính thức mongo, lưu trữ dữ liệu trong volume mongo-data.

- environment: Truyền biến môi trường MONGO\_URL để backend biết kết nối tới MongoDB.

# 6. Quy trình chạy ứng dụng với Docker

Bước 1: Build và khởi động toàn bộ hệ thống:

docker-compose up --build

Bước 2: Kiểm tra:

- API backend chạy ở: http://localhost:3000

- MongoDB chạy ẩn trong container tên mongo

Bước 3: Dừng toàn bộ container:

docker-compose down

# 7. Lợi ích khi sử dụng Docker

Nhất quán: Docker giúp đảm bảo môi trường chạy giống nhau ở máy cá nhân và server triển khai. |

|Dễ deploy: Chỉ cần một dòng lệnh để build và chạy toàn bộ hệ thống. |

Tách biệt môi trường: Mỗi dịch vụ (backend, database) chạy độc lập, không xung đột với phần mềm cài trong máy

Dễ mở rộng. Có thể thêm Redis, Nginx, hoặc service khác bằng cách khai báo thêm trong docker-compose.

# 8. Kết luận

Docker là công cụ mạnh mẽ giúp triển khai backend Node.js một cách chuyên nghiệp, hiện đại và tối ưu. Việc áp dụng Docker trong dự án trang web bán hàng giúp đảm bảo tính linh hoạt, dễ bảo trì và sẵn sàng mở rộng khi cần thiết.