

Контейнеризація проєкту VofkPizza

Підготовка проєкту

Структура проєкту:

project/

├─ app/

| └─ __init__.py

| └─ routes/

| | └─ admin.py

| | └─ api.py

| | └─ shop.py

| └─ models.py

├─ data/

├─ templates/

├─ Dockerfile

├─ docker-compose.yml

└─ requirements.txt

Налаштування Docker

Dockerfile

FROM python:3.9-slim

WORKDIR /app

RUN adduser --disabled-password --gecos " appuser

RUN mkdir -p /app/data && \

chown -R appuser:appuser /app/data

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY . .

RUN chown -R appuser:appuser /app

USER appuser

EXPOSE 5000

CMD ["python", "app.py"]

docker-compose.yml

version: '3.8'

services:

web:

build: .

ports:

- "5000:5000"

volumes:

- flask-db-data:/app/data

restart: unless-stopped

volumes:

flask-db-data:

Налаштування бази даних

Модифікований код підключення до БД:

```
def get_db_connection():
```

```
    db_path = os.path.join('data', 'db.sqlite')
```

```
    conn = sqlite3.connect(db_path)
```

```
conn.row_factory = sqlite3.Row
```

```
return conn
```

Тестування

Збірка та запуск контейнера

Збірка образу

```
$ docker build -t flask-shop .
```

Результат:

```
Successfully built 1234567890ab
```

```
Successfully tagged flask-shop:latest
```

Запуск контейнера

```
$ docker run -p 5000:5000 -v flask-db-data:/app/data -d flask-shop
```

Перевірка статусу

```
$ docker ps
```

CONTAINER ID	IMAGE	PORTS	STATUS
--------------	-------	-------	--------

1234567890ab	flask-shop	0.0.0.0:5000->5000/tcp	Up 2 minutes
--------------	------------	------------------------	--------------

Тестування endpoints

- Endpoint: /api/products
- Метод: GET
- Результат: 200 OK
- [Скріншот результату]

Проблеми та їх вирішення

Проблема з правами доступу

Проблема: Контейнер не міг створити файл БД у volume.

Рішення: Додано створення користувача та налаштування прав:

```
RUN adduser --disabled-password --gecos " appuser
```

```
RUN chown -R appuser:appuser /app/data
```

Проблема з ініціалізацією БД

Проблема: База даних не створювалася при першому запуску.

Рішення: Додано скрипт ініціалізації та entrypoint:

```
#!/bin/bash
```

```
python init_db.py
```

```
python app.py
```

Висновки

У ході виконання лабораторної роботи:

1. Успішно контейнеризовано Flask-додаток
2. Налаштовано постійне зберігання даних через Docker Volumes
3. Реалізовано безпечну роботу з БД
4. Отримано практичні навички роботи з Docker

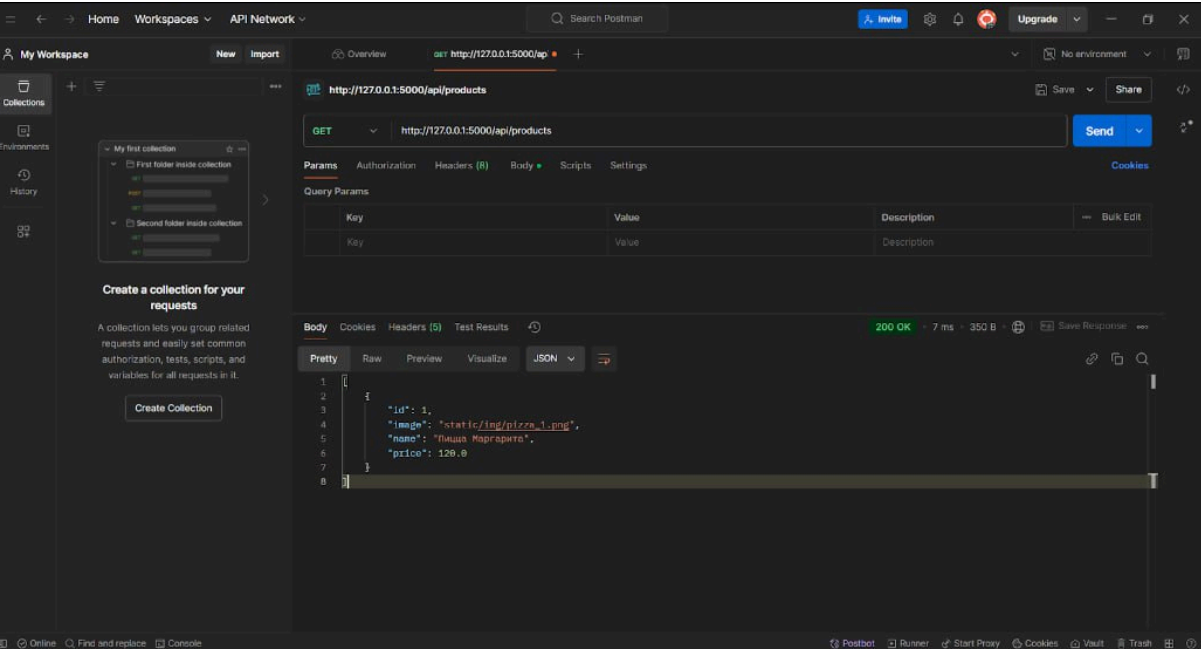
Додатки

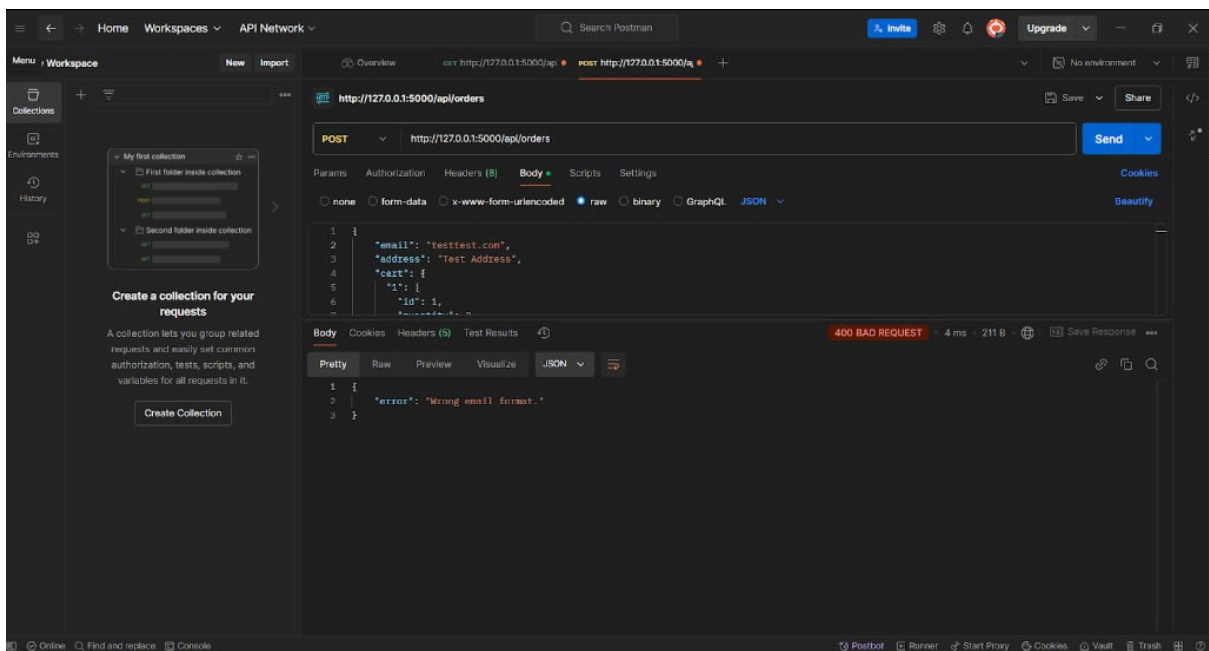
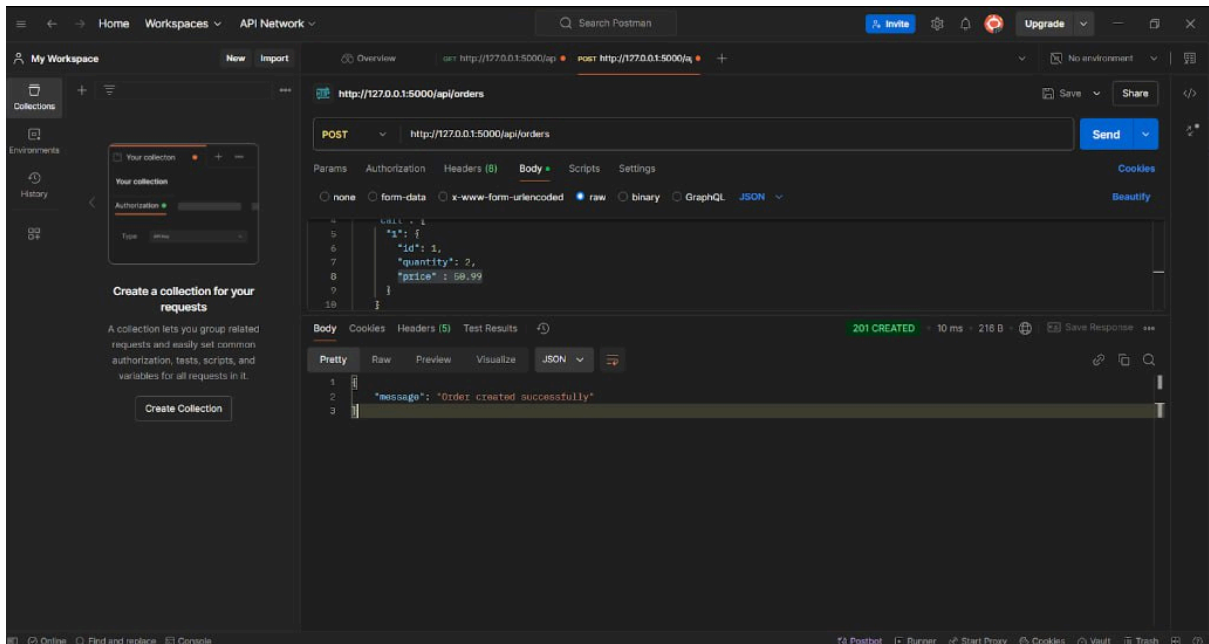
Скріншоти роботи

Процес збірки Docker образу

-

Тестування API endpoints





Перевірка даних у Docker Volume

-

