Контейнеризація проєкту

# Підготовка проекту

## Структура проекту:

stone-hearts/

├── routes/

├── \_\_init\_\_.py

├── admin.py

├── api.py

├── shop.py

└──models.py

├── data/

└──db.sqlite

├──static/

├── templates/

├── .dockerignore

├── app.py

├── docker-compose.yml

├── Dockerfile

├── init\_db.py

├── models.py

├── README.md

├── requirements.txt

└── seed\_data.py

## Налаштування Docker

**Dockerfile**

FROM python:3.9-slim

WORKDIR /app

COPY requirements.txt .

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

COPY . .

EXPOSE 5000

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

**docker-compose.yml**

version: '3.8'

services:

web:

build: .

volumes:

- ./data:/app/data

ports:

- "5000:5000"

## Налаштування бази даних

Модифікований код підключення до БД:

def get\_db\_connection():

conn = sqlite3.connect('data/db.sqlite')

conn.row\_factory = sqlite3.Row

return conn

# Тестування

## Збірка та запуск контейнера

Збірка образу

docker build -t stone-hearts .

Результат:

Building 10.8s (10/10) FINISHED

Запуск контейнера

docker run -p 5000:5000 -d stone-hearts

Перевірка статусу

docker ps

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES |
| 80ffa9eb35c6 | stone-hearts | "python app.py" | 38 seconds ago | Up 37 seconds | 0.0.0.0:5000->5000/tcp | friendly\_mendeleev |

# Тестування endpoints

* GET /api/products (працює)
* GET /api/orders (працює)
* GET /api/orders/{order\_id} (працює)
* PUT /api/orders/{order\_id} (працює)
* DELETE /api/orders/{order\_id} (працює)
* GET /api/feedback (працює)
* POST /api/feedback (працює)
* DELETE /api/feedback/{feedback\_id} (працює)

# Висновки

У ході виконання лабораторної роботи:

1. Було успішно проведено контейнеризацію Flask-додатку

2. У ході лабораторної роботи було організовано надійне зберігання даних за допомогою технології Docker Volumes.

3. У ході лабораторної роботи була організована безпечна взаємодія з базою даних.

4. Під час лабораторної роботи було набуто практичного досвіду в роботі з технологією Docker.