

# HW4--Refactor a 3-Tier WebApp with Spec-Driven

## Development (Spec Kit + AI)

### 1. What changed (from HW3) and why (reference spec sections)

- 資料庫從 PostgreSQL 改為 SQLite (spec: 資料庫簡化, 便於部署與測試)
- 移除 db 服務, docker-compose.yml 只保留 backend (Flask+Gunicorn+SQLite) 與 frontend (Nginx)
- backend 連線字串、初始化、SQL 語法全面改為 SQLite 相容
- 資料持久化改用 Docker volume 綁定主機 names.db
- requirements.txt 移除 PostgreSQL 相關依賴
- 自動建立資料表, 確保測試與啟動不會因沒有表無法啟動

### 2. 重構過程遇到的關鍵問題與解法

- SQLite 檔案權限與 volume 綁定問題  
問題: 容器啟動時出現 unable to open database file  
解法: 主機端預先建立 names.db 並給予寫入權限, 確保 Docker volume 綁定時檔案可用
- SQL 語法不相容  
問題: PostgreSQL 的 RETURNING、NOW()、IDENTITY 等語法在 SQLite 不支援  
解法: 改用 SQLite 的 AUTOINCREMENT、CURRENT\_TIMESTAMP, 查詢最後一筆資料用 SELECT ... ORDER BY id DESC LIMIT 1
- 資料表初始化  
問題: 測試或啟動時若資料表不存在會失敗  
解法: Flask 啟動時自動執行 CREATE TABLE IF NOT EXISTS

### 3. 如何在本地重現結果

- 準備環境

安裝 Docker 與 Docker Compose

- 建立資料庫檔案

在專案 backend 目錄下建立空的 names.db

```
```sh
```

```
cd backend
```

```
touch names.db && chmod 666 names.db
```

```
cd ..
```

```
```
```

- 啟動服務

```
```sh
```

```
docker compose up -d --build
```

```
```
```

- 前端操作

瀏覽 <http://localhost:8080>

可新增、查詢、刪除姓名

- 執行測試

```
```sh
```

```
docker compose exec backend pytest
```

```
```
```