



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №1
з дисципліни
Робота з даними в хмарних середовищах
Phase 1: REST API для ресурсу з використанням реляційної БД

Виконав:
студент групи ІА-11:
Пасінчук О. В.

Тема роботи: REST API для ресурсу з використанням реляційної БД

Мета роботи: Побудувати CRUD RestAPI для батьківського ресурсу, що використовує реляційну БД в хмарі

Хід роботи:

Для розробки REST API з використанням реляційної бази даних у хмарі було вирішено застосувати Django з Django REST Framework та Docker для створення ізольованого середовища. Проект був структурований за допомогою створення Docker контейнера для Django-додатку та використання зовнішньої бази даних, що розміщена на хост-машині.

Структура проекту

Проект складається з наступних ключових компонентів:

Django-додаток, що реалізує логіку REST API.

Django REST Framework для зручної роботи з REST API.

Dockerfile для створення образу Docker з Django-додатком.

docker-compose.yml для запуску додатку у Docker контейнері.

requirements.txt для визначення залежностей проекту.

Використання зовнішньої бази даних

Було вирішено використовувати зовнішню базу даних, доступ до якої налаштовано через змінні оточення у файлі docker-compose.yml. Це дозволило легко інтегрувати додаток з існуючою інфраструктурою без необхідності запуску додаткових служб баз даних у контейнерах.

Розробка і тестування

Розробка велася ітеративно з використанням принципів TDD (Test-Driven Development). Для кожного ендпоінта REST API були написані інтеграційні тести, що забезпечують високий рівень впевненості у якості та стабільності додатку.

Розгортання

За допомогою Docker та docker-compose було легко налаштувати і запустити додаток у ізольованому середовищі, що спростило процес розгортання та тестування. Використання зовнішньої бази даних додатково знизило складність конфігурації середовища.

Dockerfile

Визначає образ Docker для додатку Django, включаючи налаштування середовища виконання, встановлення залежностей і копіювання коду додатку у контейнер.

docker-compose.yml

Файл для опису і запуску додатку і його сервісів за допомогою Docker Compose. Включає налаштування для Django-додатку та змінні оточення.

requirements.txt

Список залежностей Python для додатку, які потрібно встановити у контейнері Docker.

manage.py

Утиліта командного рядка Django для виконання адміністративних завдань, наприклад, запуску сервера розробки, застосування міграцій бази даних тощо.

myproject/settings.py

Файл налаштувань для проекту Django, включає конфігурацію бази даних, встановлені додатки, середовище середнього програмного забезпечення, шаблони та інші налаштування.

myproject/urls.py

Файл для опису URL-адрес проекту Django, визначає маршрути до в'юшок.

myapp/models.py

Опис моделей даних для Django-додатку. Визначає структуру бази даних.

myapp/views.py

Файл, що містить логіку обробки запитів для Django-додатку. Визначає в'юшки.

myapp/serializers.py

Файл для опису серіалізаторів Django REST Framework, що використовуються для перетворення моделей даних у формат JSON і навпаки.

myapp/urls.py

Локальний файл URL-конфігурації для додатку, визначає маршрути до в'юшок в рамках додатку.

myapp/tests.py

Файл з тестами для Django-додатку, містить тестові випадки для перевірки логіки додатку.

Висновки

Цей проект демонструє ефективність використання сучасних інструментів та підходів у розробці веб-додатків. Django разом з Django REST Framework забезпечили потужну платформу для розробки REST API, а Docker спростив процес розгортання та забезпечив високий рівень ізоляції та контролю над середовищем виконання. Використання зовнішньої бази даних підвищило гнучкість проекту та дозволило легко інтегруватися з існуючою інфраструктурою.