

Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра системного аналізу та теорії прийняття рішень

Звіт
з лабораторного проекту
дисципліни «Бази Даних»
На тему: «Система управління студентами факультету
за спеціальностями, курсами та групами»

Виконав
студент групи К-23

Флакей Р.Р.

Прийняв

Івохін Є. В.,

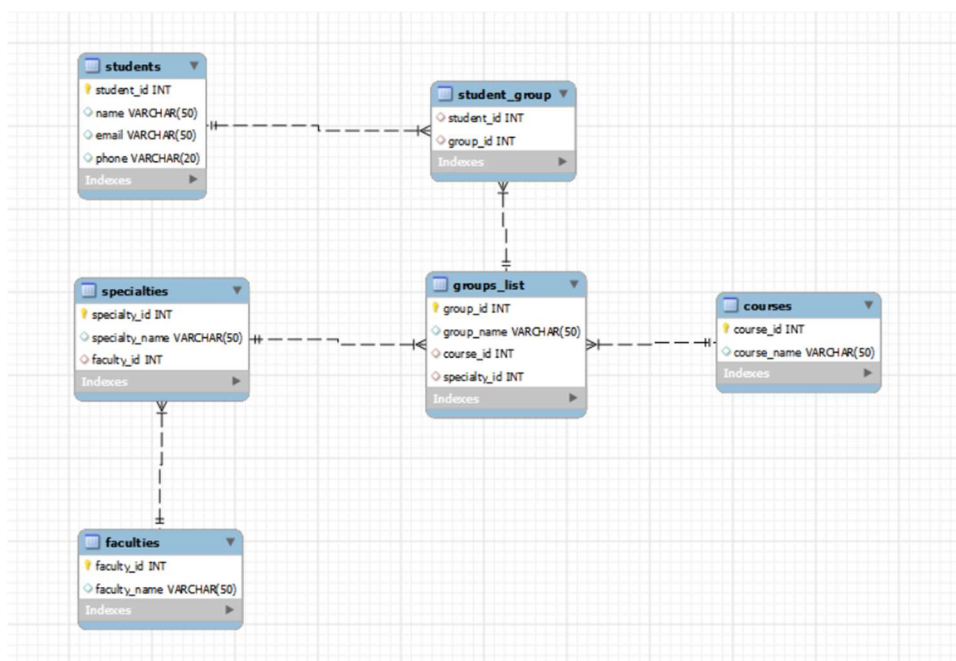
Київ - 2023

Структура проекту

1. База даних - MySQL, яка містить таблиці для зберігання даних про студентів, факультети, спеціальності, курси та групи.
2. Backend - написаний на мові програмування Python, використовує фреймворк Flask.
3. Docker - використовується для контейнеризації проекту та його запуску на будь-якому сервері.

1. База Даних

- Таблиця "courses" містить інформацію про курси, що викладаються в університеті.
- Таблиця "faculties" містить інформацію про факультети університету.
- Таблиця "specialties" містить інформацію про спеціальності, які пропонуються на факультетах.
- Таблиця "groups_list" містить інформацію про групи студентів, які навчаються на певній спеціальності та курсі.
- Таблиця "students" містить інформацію про студентів, які навчаються в університеті.
- Таблиця "student_group" містить інформацію про те, які студенти належать до яких груп.



ER-діаграма

2. Backend

Backend проекту реалізований на мові програмування Python з використанням фреймворку Flask. Основні можливості проекту включають в себе додавання, редагування та видалення записів у базі даних, що стосуються курсів, факультетів, спеціальностей, груп студентів та студентів. Крім того, проект має можливість відображення списків курсів, факультетів, спеціальностей, груп студентів та студентів, а також можливість перегляду детальної інформації про кожен запис. Проект також має можливість додавання та видалення студентів до груп, а також редагування інформації про групи студентів. Всі ці можливості реалізовані через веб-інтерфейс, який забезпечує зручний та простий доступ до функціоналу проекту.

The screenshot displays a web application interface with a navigation bar at the top containing links: Home, Faculties, Courses, Specialties, Groups, Students, and Student Groups. The interface is divided into four main sections:

- Faculties:** A panel with a search bar, an 'Add Faculty' button, and a table listing faculties.
- Groups:** A panel with a search bar, an 'Add Group' button, and a table listing groups.
- All Courses:** A panel with a search bar, a 'Search' button, an 'Add Course' button, and a table listing courses.
- Edit Group:** A panel for editing a group, featuring input fields for Group Name, Course, and Specialty, and a 'Save Changes' button.

Faculties Table:

Name	Actions
Факультет Комп'ютерних Наук та Інформатики	Edit Delete
Факультет Інформаційних Технологій	Edit Delete

Groups Table:

Group Name	Course	Specialty	Actions
K-24	Другий курс	124 Системний аналіз	Edit Delete
K-13	Перший курс	124 Системний аналіз	Edit Delete
K000	Четвертий курс	124 Системний аналіз	Edit Delete
K-23	Четвертий курс	122 Квантиторні науки	Edit Delete

All Courses Table:

Course Name	Actions
Другий курс	Edit Delete
Перший курс	Edit Delete
Третій курс	Edit Delete
Четвертий курс	Edit Delete

Edit Group Form:

Group Name:

Course:

Specialty:

3. Docker

Docker контейнер у проєкті використовується для забезпечення ізольованого середовища для запуску додатку. У контейнері встановлюється необхідне програмне забезпечення, включаючи Python, Flask та MySQL, а також копіюється весь код проєкту.

Запуск:

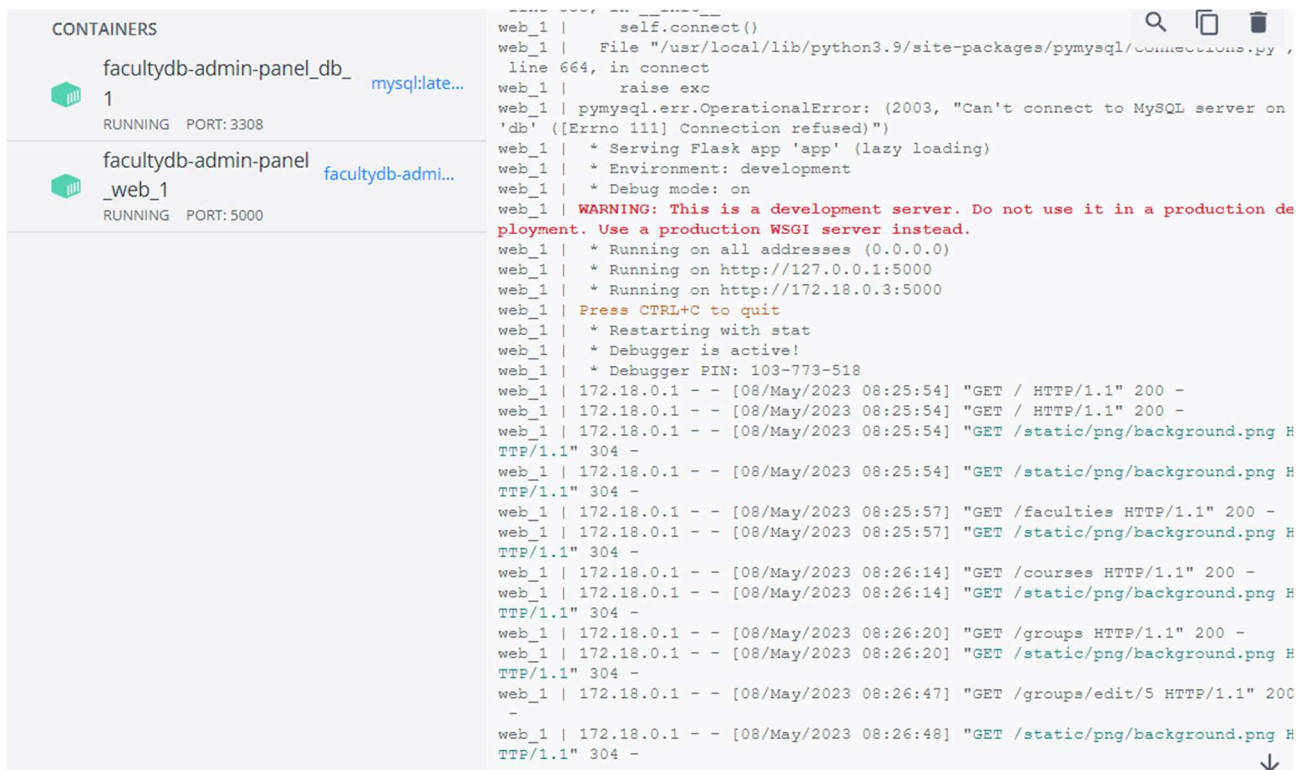
`docker-compose up --build`

Зупинка:

`docker-compose down`

Підключення:

`localhost:5000`



The screenshot shows the Docker Desktop interface. On the left, under 'CONTAINERS', there are two containers: 'facultydb-admin-panel_db_1' (mysql:late...) and 'facultydb-admin-panel_web_1' (facultydb-admi...). The 'web_1' container is selected, and its logs are displayed on the right. The logs show the container starting, connecting to the database, and serving the Flask app. It includes a warning about using a development server and a list of HTTP requests and responses.

```
web_1 | self.connect()
web_1 | File "/usr/local/lib/python3.9/site-packages/pymysql/connections.py",
web_1 | line 664, in connect
web_1 |     raise exc
web_1 | pymysql.err.OperationalError: (2003, "Can't connect to MySQL server on
web_1 | 'db' ([Errno 111] Connection refused)")
web_1 | * Serving Flask app 'app' (lazy loading)
web_1 | * Environment: development
web_1 | * Debug mode: on
web_1 | WARNING: This is a development server. Do not use it in a production de
web_1 | ployment. Use a production WSGI server instead.
web_1 | * Running on all addresses (0.0.0.0)
web_1 | * Running on http://127.0.0.1:5000
web_1 | * Running on http://172.18.0.3:5000
web_1 | Press CTRL+C to quit
web_1 | * Restarting with stat
web_1 | * Debugger is active!
web_1 | * Debugger PIN: 103-773-518
web_1 | 172.18.0.1 - - [08/May/2023 08:25:54] "GET / HTTP/1.1" 200 -
web_1 | 172.18.0.1 - - [08/May/2023 08:25:54] "GET / HTTP/1.1" 200 -
web_1 | 172.18.0.1 - - [08/May/2023 08:25:54] "GET /static/png/background.png H
web_1 | TTP/1.1" 304 -
web_1 | 172.18.0.1 - - [08/May/2023 08:25:54] "GET /static/png/background.png H
web_1 | TTP/1.1" 304 -
web_1 | 172.18.0.1 - - [08/May/2023 08:25:57] "GET /faculties HTTP/1.1" 200 -
web_1 | 172.18.0.1 - - [08/May/2023 08:25:57] "GET /static/png/background.png H
web_1 | TTP/1.1" 304 -
web_1 | 172.18.0.1 - - [08/May/2023 08:26:14] "GET /courses HTTP/1.1" 200 -
web_1 | 172.18.0.1 - - [08/May/2023 08:26:14] "GET /static/png/background.png H
web_1 | TTP/1.1" 304 -
web_1 | 172.18.0.1 - - [08/May/2023 08:26:20] "GET /groups HTTP/1.1" 200 -
web_1 | 172.18.0.1 - - [08/May/2023 08:26:20] "GET /static/png/background.png H
web_1 | TTP/1.1" 304 -
web_1 | 172.18.0.1 - - [08/May/2023 08:26:47] "GET /groups/edit/5 HTTP/1.1" 200
web_1 | -
web_1 | 172.18.0.1 - - [08/May/2023 08:26:48] "GET /static/png/background.png H
web_1 | TTP/1.1" 304 -
```

Файли проєкту:

<https://github.com/MinTins/FacultyDB-admin-panel>