Державний університет інтелектуальних технологій і зв’язку

Факультет: Інформаційних технологій та кібербезпеки

Кафедра: Комп’ютерних наук

**Протокол лабораторної роботи №6**

**з дисципліни**

**«Веб-технології»**

Виконав студент групи ІПЗ-3.02

Жигальов Володимир Олександрович

Перевірив ст. викл. каф. КН

Северин М.В.

Одеса - 2025

**Лабораторна робота № 6**

**Тема:** Механізм взаємодії з базами даних через об'єкти мови Python (Фреймворк Django), технологія ORM.

**Мета роботи:** набуття практичного досвіду налаштування підключення Django до бази даних, створення моделей, виконання міграції, додавання моделей до адміністративної панелі, керування відображенням даних в панелі.

**Завдання**

1. Створити базу даних у PostgreSQL.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

2. Налаштувати клієнтську кодування, часову зону тощо.

*Рішення:*

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

3. Створити користувача PostgreSQL і надати йому права на базу даних.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

4. У `settings.py` налаштувати `DATABASES`, використовуючи дані з п.1 і п.3.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

5. Запустити сервер, переконатися в наявності 18 міграцій (не застосованих).

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

6. Застосувати міграції, перевірити в БД наявність 10 нових таблиць.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

7. Створити суперюзера Django.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

8. Перевірити в PostgreSQL запис суперюзера в таблиці `auth\_user`.

*Рішення:*

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Контент, сгенерированный ИИ, может содержать ошибки.

9. Запустити сервер, увійти в адмінпанель під суперюзером.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

10. Переглянути й за потреби відредагувати дані користувача.

*Рішення:*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

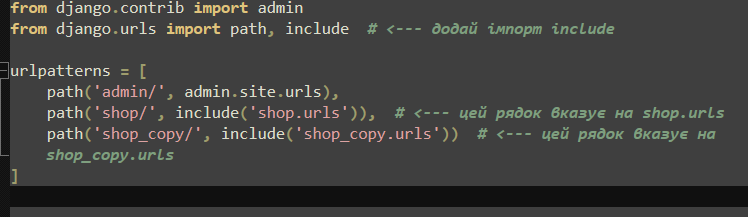
Контент, сгенерированный ИИ, может содержать ошибки.

11. Додати застосунок з лабораторної №5 у `INSTALLED\_APPS`.

*Рішення:*

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.



12. Створити дві моделі з відношенням один-до-багатьох.

*Рішення:*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

13. Створити міграції для цих моделей.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

14. Застосувати міграції.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

15. Перевірити в PostgreSQL створення нових таблиць.

*Рішення:*

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

16. Додати застосунок з моделями до адмінпанелі.

*Рішення:*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

17. Запустити сервер.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

18. Перевірити розділ застосунку в адмінпанелі.

*Рішення:*

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

19. Додати щонайменше по 3 записи в таблиці.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

20. Переглянути відображення списків записів у адмінпанелі.

*Рішення:*

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

21. Забезпечити інформативне відображення записів у списку.

*Рішення:*

Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как снимок экрана, текст, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

22. Вивести дані однієї таблиці на головну сторінку застосунку.

*Рішення:*

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Контент, сгенерированный ИИ, может содержать ошибки.

23. Зафіксувати стан у системі контролю версій.

*Рішення:*

**Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.**

**Відповідь на контрольні питання:**

1. Що таке ORM?

ORM (Object-Relational Mapping) — це технологія, яка дозволяє взаємодіяти з базою даних за допомогою об'єктів мови програмування замість написання SQL-запитів. У Django ORM дозволяє описувати структуру таблиць у вигляді Python-класів (моделей) і виконувати операції (створення, читання, оновлення, видалення) через методи Python.

Приклад:

from django.db import models

class Book(models.Model):

title = models.CharField(max\_length=100)

author = models.CharField(max\_length=100)

2. Як підключити СКБД PostgreSQL до проєкту Django?

Для підключення PostgreSQL потрібно у файлі `settings.py` налаштувати секцію `DATABASES`:

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.postgresql',

'NAME': 'назва\_бази',

'USER': 'користувач',

'PASSWORD': 'пароль',

'HOST': 'localhost',

'PORT': '5432',

}

}

Також потрібно встановити драйвер:

pip install psycopg2

3. Як в проєкті Django створити модель для таблиці бази даних?

У файлі `models.py` створюється клас, який наслідує `models.Model`, і в ньому описуються поля таблиці:

from django.db import models

class Product(models.Model):

name = models.CharField(max\_length=100)

price = models.DecimalField(max\_digits=8, decimal\_places=2)

4. Яка команда використовується для створення міграцій в Django?

Для створення файлів міграцій на основі змін у моделях використовується команда:

python manage.py makemigrations

5. Як додати керування моделями в адміністративну панель проєкта Django?

Потрібно зареєструвати модель у файлі `admin.py` відповідного застосунку:

from django.contrib import admin

from .models import Product

admin.site.register(Product)

Щоб кастомізувати відображення:

class ProductAdmin(admin.ModelAdmin):

list\_display = ('name', 'price')

admin.site.register(Product, ProductAdmin)

**Висновок**

У ході виконання завдань було розглянуто основи роботи з Django та PostgreSQL: підключення СКБД до проєкту, створення моделей, формування та застосування міграцій, а також керування даними через адміністративну панель. Завдяки використанню ORM у Django, робота з базою даних стає зручнішою та не потребує прямого написання SQL-запитів.