МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

Навчально-науковий інститут фізико-технічних та комп’ютерних наук

Кафедра комп’ютерних наук

**ЗВІТ**  
**про виконання лабораторної роботи № 2**

**з дисципліни**

**«Сучасні технології програмування»**

**на тему: «Створення Django-аплікацій. Робота з базою даних та інтерфейсом адміністратора»**

Виконав

студент 5-го курсу

544м групи

Вікнянський Руслан

Чернівці, 2025

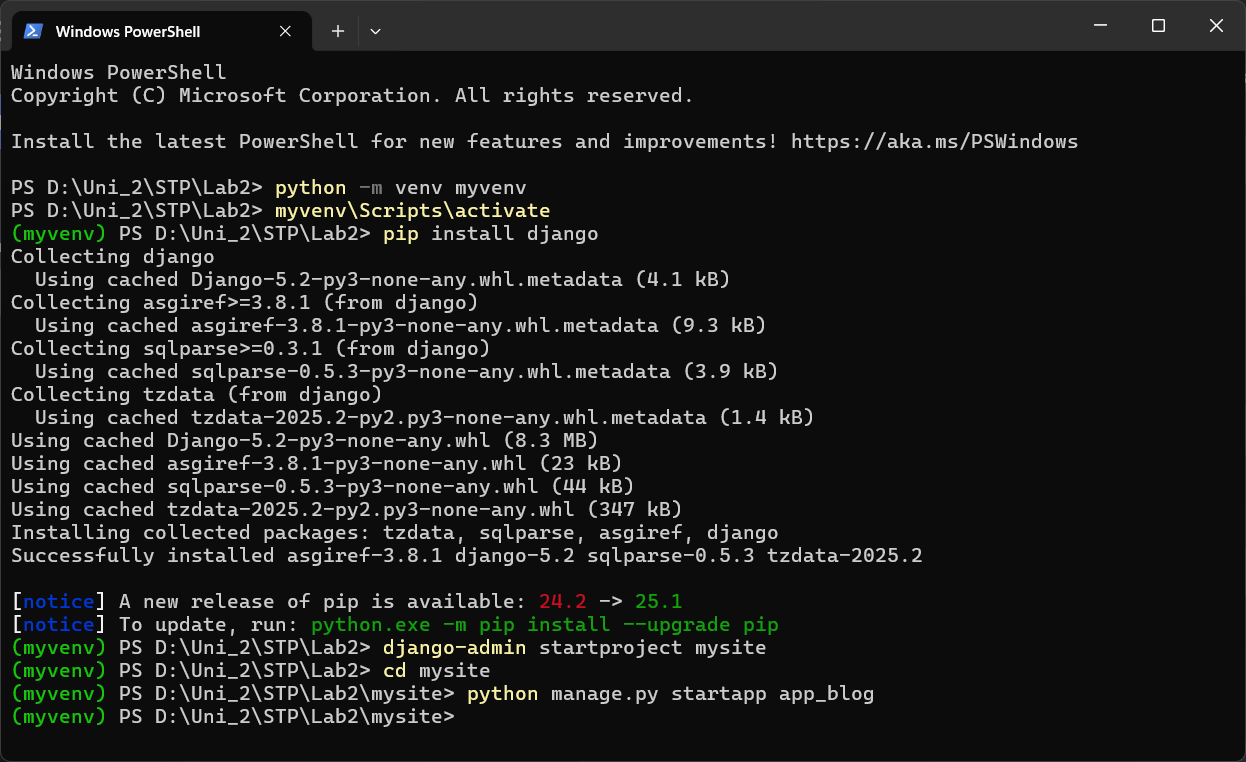
**Мета:** Ознайомитись з процесом створення Django-аплікації та процесом налаштування проекту, створити базу даних для сайту.

Посилання на Github - <https://github.com/RuslanCHNU/STP_Labs>

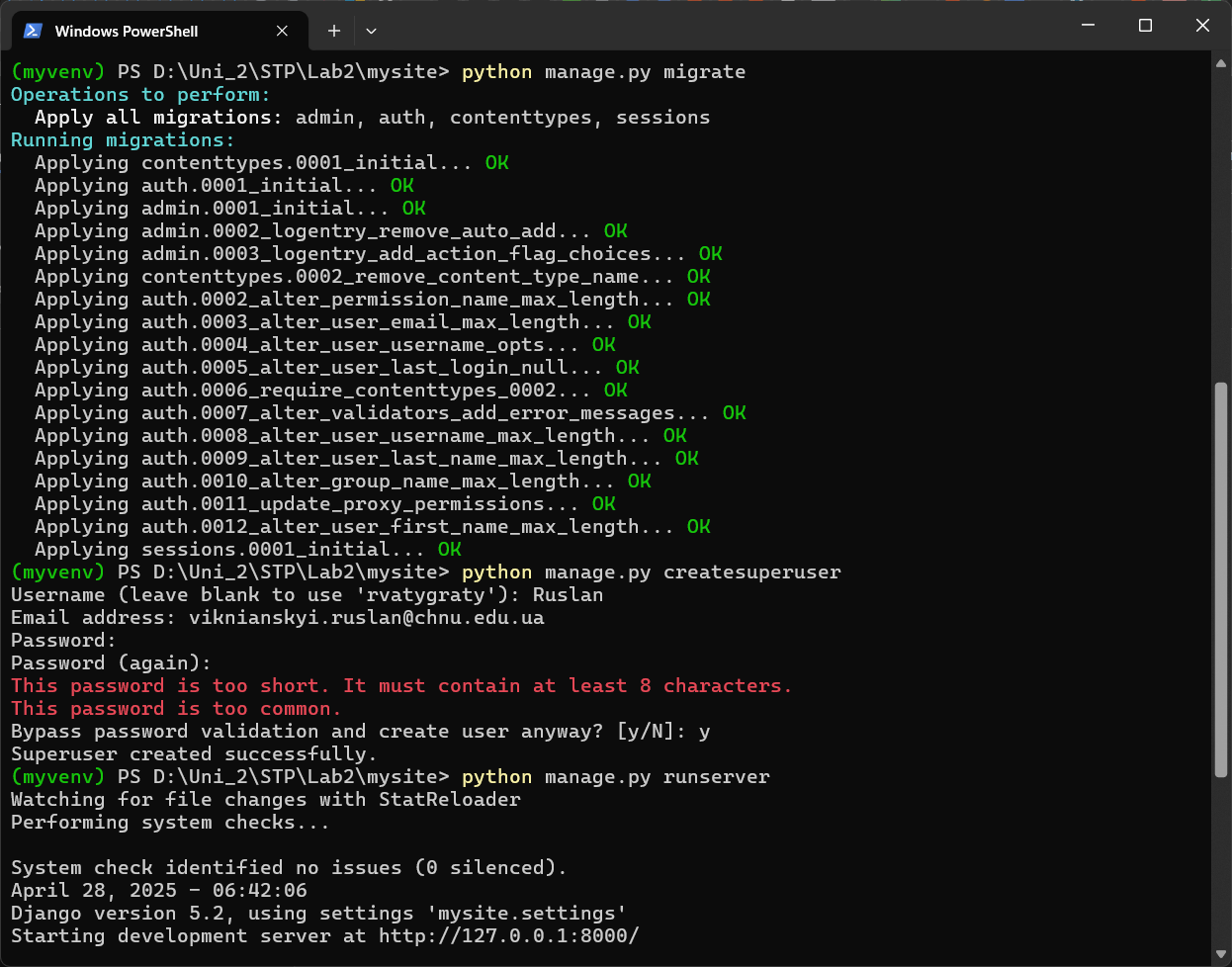
**Порядок виконання роботи**

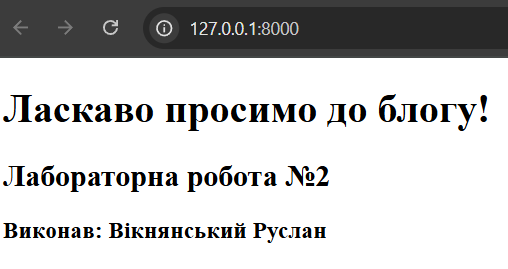
**Частина 1**

1. Створення Django-аплікації, робота з базою даних та адмін-панеллю
   1. Налаштування середовища



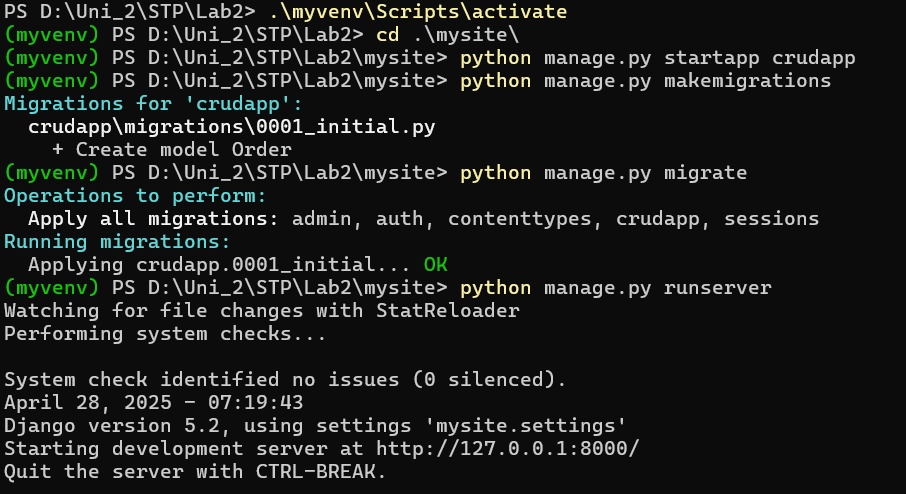
1. Міграція баз даних та запуск сервера



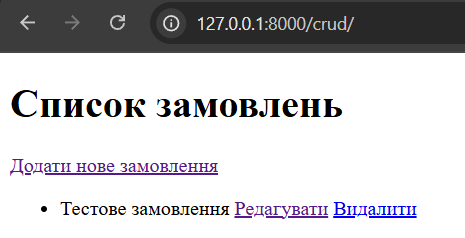


Частина 2

1. Створення додатку crudapp



1. Працюючий додаток crudapp із функцією додавання та видалення замовлення



**Відповіді на контрольні запитання**

1. **Що таке аплікація?**

Аплікація в Django — це модуль, який інкапсулює певну функціональність веб-сайту (наприклад, блог, форум, систему користувачів). Вона містить:

* 1. Моделі даних (models.py).
  2. Логіку обробки запитів (views.py).
  3. Шаблони для відображення (templates/).
  4. URL-шляхи (urls.py).

Аплікації дозволяють структурувати проект, забезпечуючи повторне використання коду.

1. **Яка архітектура використовується в Django Framework?**

Django використовує архітектуру MVT (Model-View-Template):

* 1. Model — відповідає за структуру даних та взаємодію з базою даних.
  2. View — обробляє бізнес-логіку, приймає запити, взаємодіє з моделями та передає дані до шаблонів.
  3. Template — визначає, як дані відображаються на сторінці (HTML-розмітка).

1. **Для чого призначений файл** urls.py**?**

Файл urls.py:

* 1. Визначає **URL-шляхи** проекту або аплікації.
  2. Пов'язує конкретні адреси (наприклад, /about/) з відповідними представленнями (views).
  3. Використовує функції або класи з views.py для обробки запитів.

1. **Для чого призначений файл** views.py**?**

Файл views.py:

* 1. Містить **представлення (views)**, які обробляють HTTP-запити.
  2. Взаємодіє з моделями для отримання даних з бази.
  3. Передає дані до шаблонів для відображення.
  4. Може виконувати CRUD-операції, автентифікацію, валідацію тощо.

1. **Де необхідно вказувати налаштування бази даних для проекту?**

Налаштування бази даних вказуються у файлі settings.py проекту:

* 1. DATABASES — вибір СУБД (SQLite, MySQL тощо).
  2. ENGINE — драйвер бази даних (наприклад, django.db.backends.sqlite3).
  3. NAME, USER, PASSWORD, HOST, PORT — параметри підключення.

1. **Які існують методи HTTP запитів?**

Основні методи HTTP:

* 1. **GET** — отримання даних (наприклад, завантаження сторінки).
  2. **POST** — надсилання даних (наприклад, відправка форми).
  3. **PUT** — оновлення всього ресурсу.
  4. **PATCH** — часткове оновлення ресурсу.
  5. **DELETE** — видалення ресурсу.
  6. **HEAD** — отримання метаданих (без тіла відповіді).
  7. **OPTIONS** — отримання інформації про доступні методи.

1. **Що означають CRUD операції?**

**CRUD** — це базові операції для роботи з даними:

* 1. **Create (Створити)** — додавання нового запису (метод POST).
  2. **Read (Читати)** — отримання даних (метод GET).
  3. **Update (Оновити)** — редагування існуючого запису (методи PUT, PATCH).
  4. **Delete (Видалити)** — видалення запису (метод DELETE).

1. **Для чого призначений файл** forms.py**?**

Файл forms.py:

* 1. Визначає **форми** для введення даних користувачем.
  2. Автоматизує валідацію даних (наприклад, перевірка на правильність email).
  3. Генерує HTML-код для полів форми.
  4. Може бути зв'язаний з моделями (ModelForm), що спрощує роботу з базою даних.