Міністерство освіти і науки України

Черкаський державний бізнес-коледж

Кафедра комп’ютерної інженерії та інформаційних технологій

Відділення інженерії програмного забезпечення

**Звіт**

з індивідуального завдання

з курсу «Основи інженерії даних»

студента групи Маренича Федора

викладач Немченко В.Ю.

Черкаси 2025

**Індивідуальне завдання**

**Варіант:** 9. Проектування БД з аналізу і контролю успішності здобувачів ВУЗу.

**Проект:** School Todo Management System

**Короткий огляд:** це повноцінний full-stack веб-додаток. Проєкт демонструє сучасні корпоративні підходи та чисту, масштабовану архітектуру. У його основі - бекенд на ASP.NET Core Web API та фронтенд на React, що взаємодіють через REST-запити.

**Посилання на репозиторій:** <https://github.com/Ururu221/SchoolTodoApi>

*(рекомендується також прочитати* ***README.md*** *файл у* ***віддаленому*** *репозиторії* [*за посиланням*](https://github.com/Ururu221/SchoolTodoApi)*)*

**Ключові технології та особливості**

* **ASP.NET Core Web API (.NET 8)**
  + **Dependency Injection (DI)** на кожному рівні: контролери, сервіси, налаштування та навіть клієнт MongoDB - усе впроваджується через вбудований IoC-контейнер. Це розв'язує залежності, підвищує модульність і полегшує юніт-тестування.
  + **Сервісний шар**: окремі класи StudentService, SchoolService, TodoItemService інкапсулюють логіку доступу до даних і бізнес-правила, дотримуючись принципу єдиної відповідальності.
  + **Options Pattern**: налаштування SchoolDatabaseSettings зі strongly-typed binding із appsettings.json через IOptions<T>.
  + **Swagger / OpenAPI**: автогенерована документація з інтерактивним UI для тестування всіх CRUD-операцій.
  + **CORS**: політика, що дозволяє React-додатку без проблем звертатись до API з іншого походження.
  + **Логування**: ILogger<T> у кожному сервісі для відстеження підключень, операцій та помилок - практичний приклад production-рівня логування.
* **MongoDB**
  + **Документна база даних**: окремі колекції Students, Schools, TodoItems.
  + **ObjectId**: усі сутності мають згенеровані 24-символьні hex-ідентифікатори; поле SchoolId у студента опціональне й ігнорується, якщо null.
  + **Асинхронні операції** через офіційний драйвер MongoDB.Driver для високої продуктивності.
* **React-фронтенд**
  + Створено за допомогою Create-React-App + **React Router 6** для SPA:
    - /students - список, додавання, редагування та видалення студентів
    - /schools - управління школами
    - /todos - створення та відстеження завдань, пов'язаних зі студентами чи школами
  + **Axios + сервіс API** (services/api.js) з єдиним baseURL для всіх HTTP-запитів.
  + **Bootstrap 5** для адаптивного та професійного стилю форм і таблиць.
  + **React-хуки** (useState, useEffect, useNavigate) для керування станом і навігацією.
  + **Динамічні форми**: випадаючі списки автоматично заповнюються з API, гарантують цілісність даних.

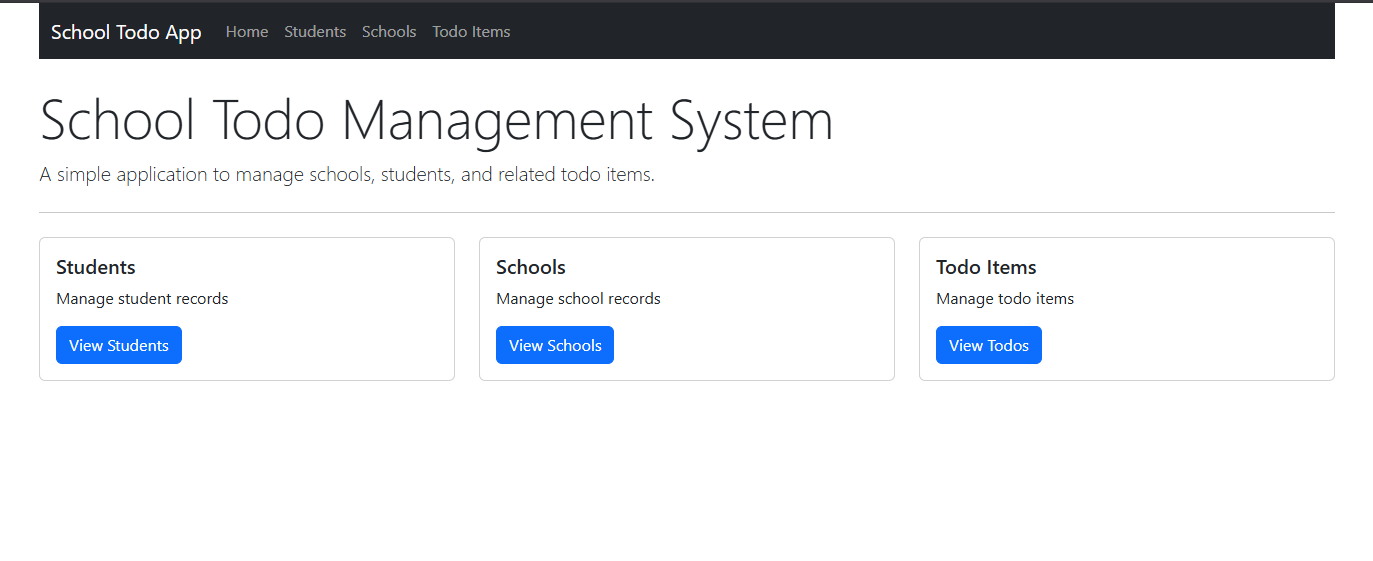
**Архітектура та дизайн**

* **Шарова структура**: чіткий розподіл контролерів, сервісів, моделей даних і компонентів UI.
* **Масштабованість**: нові функції (наприклад, автентифікація або додаткові сутності) легко інтегруються завдяки DI та сервісній архітектурі.
* **Тестованість**: сервіси та контролери можна юніт-тестувати незалежно, мокуючи залежності.

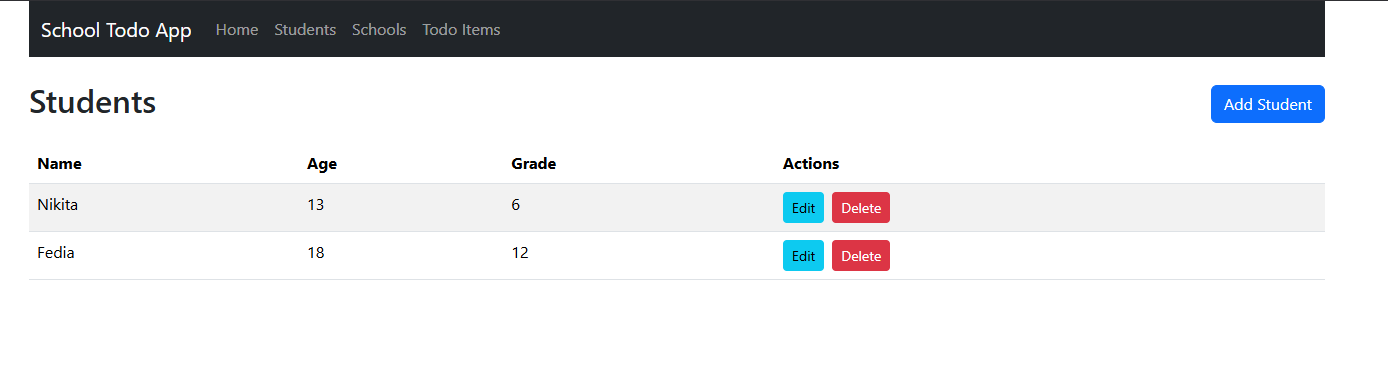
**Чому варто відзначити**

Проєкт не просто демонструє CRUD - він повністю реалізує **Dependency Injection** на кожному рівні, використовує strongly-typed налаштування, автодокументований Swagger та сучасний React-фронтенд, що відповідає принципам реальних корпоративних рішень. Ваш викладач побачить не просто студентську роботу, а професійний full-stack продукт із найкращими практиками інженерії ПЗ.

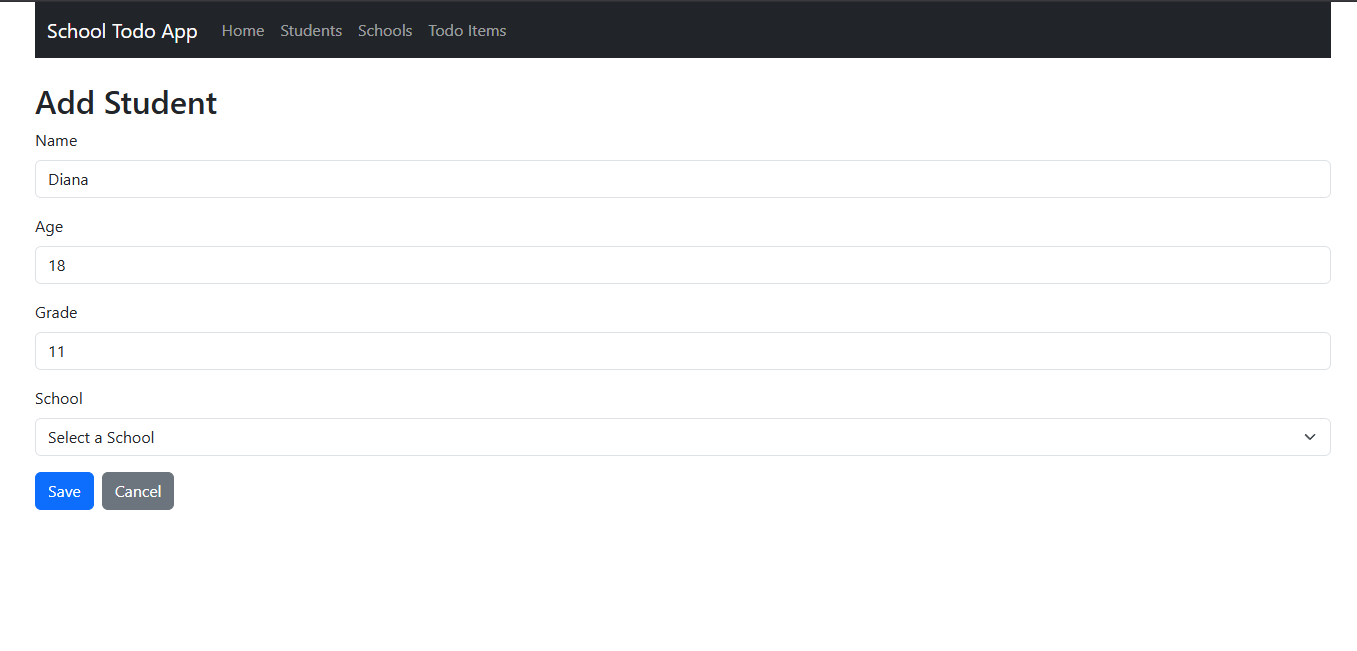
Нижче наведені скриншоти роботи сайту:



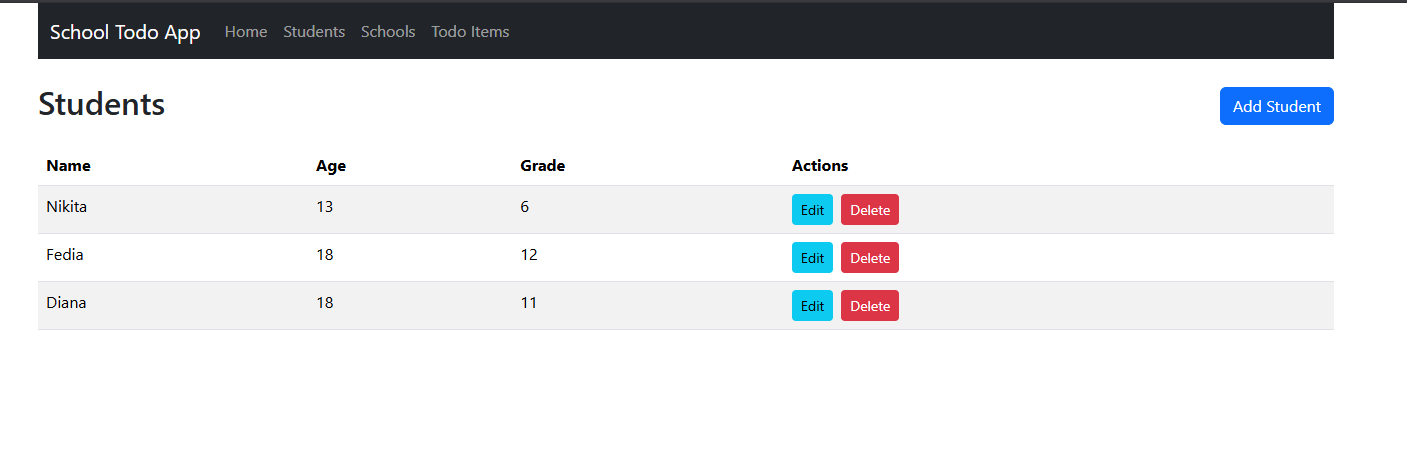
*Рисунок 1. Головна сторінка додатку*



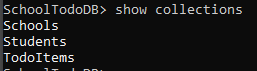
*Рисунок 2. Список студентів до додавання нового запису*



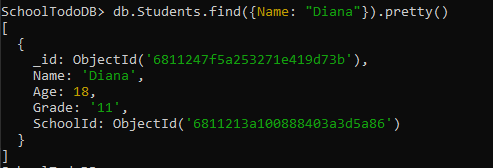
*Рисунок 3. Форма додавання нового студента*



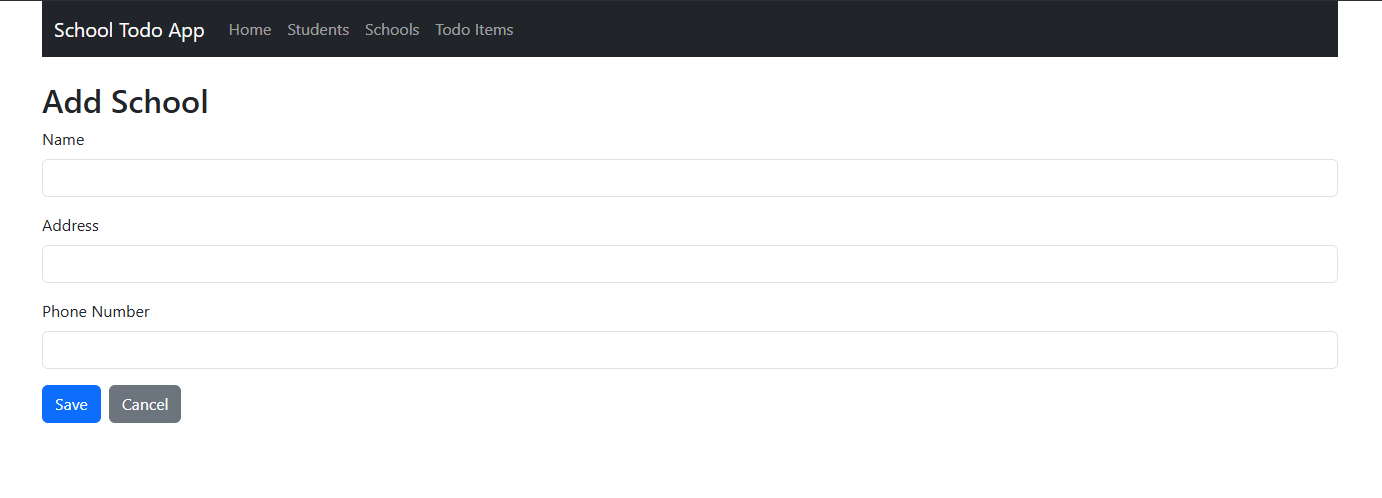
*Рисунок 4. Оновлений список студентів після додавання*



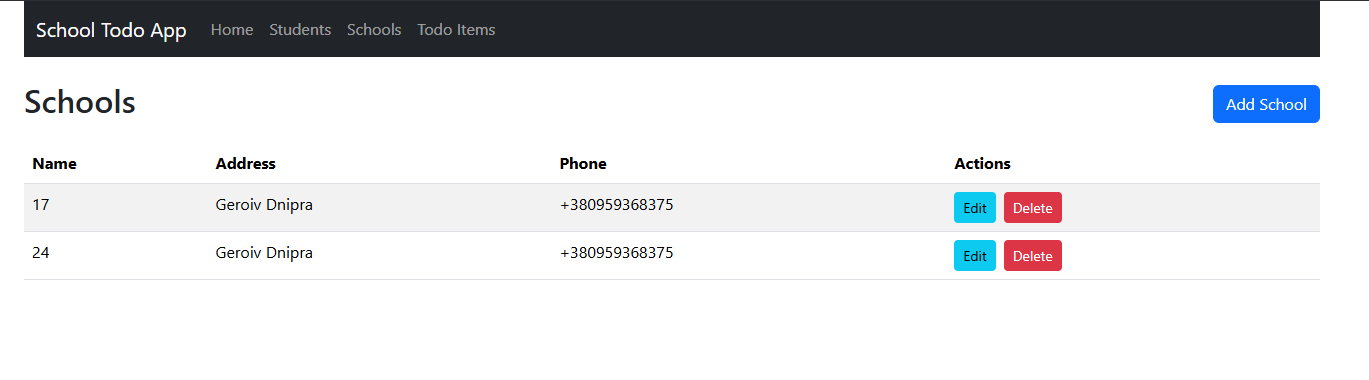
*Рисунок 5. Перевірка колекцій у MongoDB*



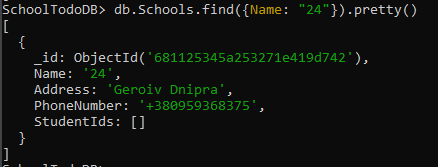
*Рисунок 6. Новий документ студента в MongoDB*



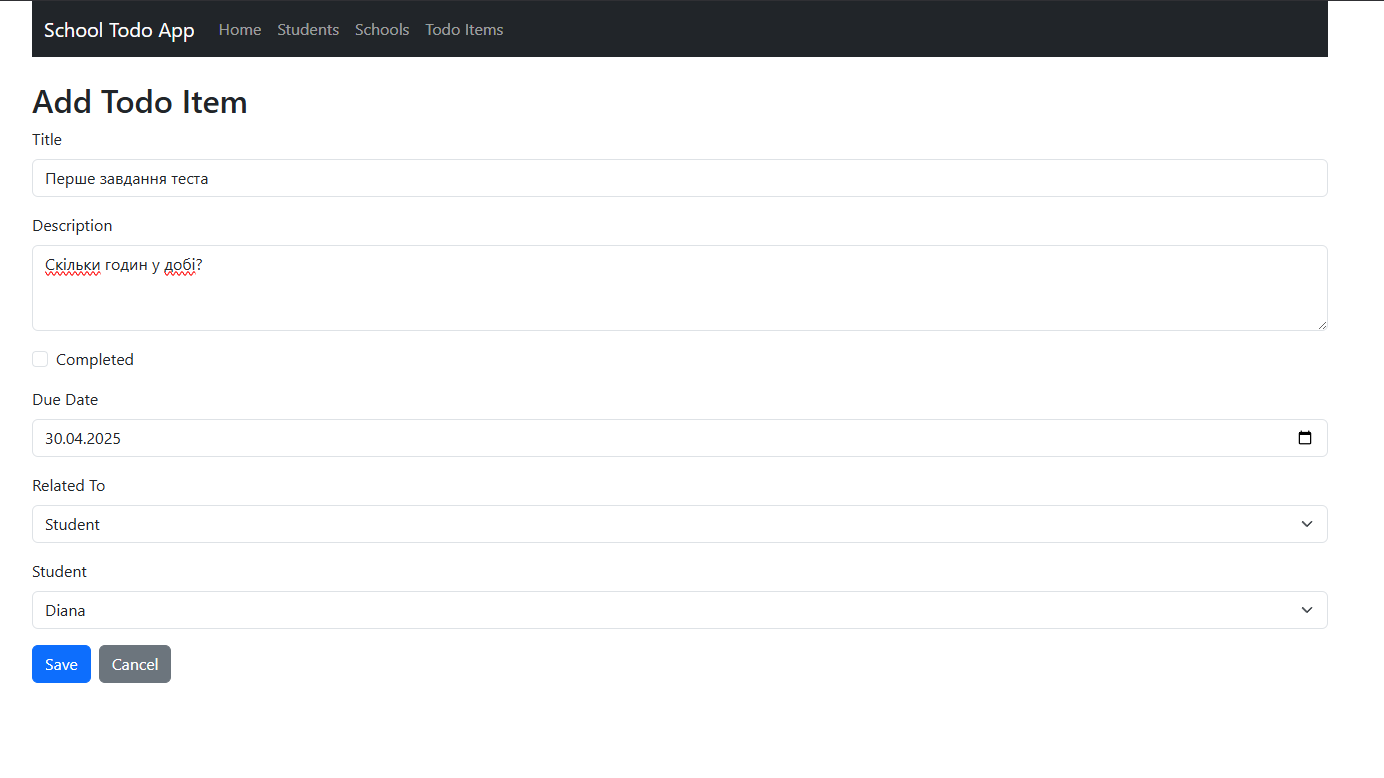
*Рисунок 7. Форма додавання нової школи*



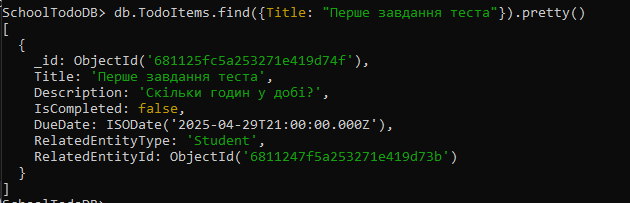
*Рисунок 8. Оновлений список шкіл після додавання*



*Рисунок 9. Документ нової школи в MongoDB*



*Рисунок 10. Форма додавання нового Todo-завдання*



*Рисунок 11. Документ Todo-завдання в MongoDB*