


ДОДАТОК А

Графічний матеріал кваліфікаційної роботи



Веб-застосунок з
використанням Web API для
адміністрування кінотеатру

Виконав ст. групи
КІУКІу-22-1
Попов К.І.



Мета проєкту

Мета: створити сучасний веб-застосунок "Кінотеатр" на основі Angular (клієнтська частина) та RESTful Web API (серверна частина), який забезпечить ефективне управління показами, бронюванням квитків та взаємодію між адміністрацією та глядачами (клієнтами). Система має надати зручний інтерфейс для управління фільмами, сеансами, бронювання квитків та контролю доступу, з можливістю подальшого масштабування.




Завдання проєкту

Розробка клієнтської частини (Angular):

- Реалізація інтуїтивного інтерфейсу з різними ролями:
 - Адміністратор: повний доступ до управління фільмами, сеансами, залами, користувачами та квитками;
 - Глядач/Клієнт: перегляд доступних фільмів та сеансів, бронювання та купівля квитків, перегляд історії замовлень.

3



Завдання проєкту

Серверна частина (ASP.NET Core Web API):

- Реалізація CRUD-операцій для сутностей:
 - Фільми;
 - Сеанси;
 - Кінозали;
 - Квитки/Бронювання;
 - Користувачі.
- Підтримка фільтрації даних (фільми за жанром, сеанси за датою, фільмом, залом).

4



База даних проєкту

- Призначення: Централізоване сховище для всіх даних веб-застосунку для кінотеатру на MS SQL Server.
- Взаємодія: База даних взаємодіє з серверною частиною на C# через Entity Framework Core.
- Ключові сутності:
 - Фільми: основна інформація про кінострічки;
 - Жанри та Актори: каталоги для класифікації фільмів;
 - Кінозали: дані про місця показу;
 - Сеанси: розклад показів фільмів;
 - Користувачі: дані глядачів та персоналу;
 - Квитки/Бронювання: записи про придбані місця;
 - Зв'язуючі таблиці: для зв'язків між фільмами та їхніми жанрами/акторами.
- Переваги: Забезпечує цілісність та ефективний доступ до даних.

5



Робота серверної частини

Серверна частина веб-застосунку відповідає за:

- Обробку HTTP-запитів від клієнтського інтерфейсу (Angular) через контролери Web API.
- Реалізацію бізнес-логіки управління кінотеатром: реєстрація користувачів, бронювання та купівля квитків, управління розкладом сеансів, адміністрування контенту та кінозалів.
- Взаємодію з базою даних за допомогою ORM-технології Entity Framework Core, що дозволяє ефективно зберігати, змінювати та отримувати дані.
- Аутентифікацію та авторизацію через ASP.NET Core Identity з використанням JWT-токенів, що забезпечує захищений доступ до ресурсів API.

6

Архітектура серверної частини

Багатошарова архітектура N-tier

Рівні:

- API-рівень – обробка HTTP-запитів.
- Рівень бізнес-логіки – правила обробки даних.
- Рівень доступу до даних – взаємодія з базою даних.



7

Технології що використовувалися



- ASP.NET Core
- Entity Framework Core
- Swagger
- AutoMapper
- JWT для авторизації
- Angular



8

Тестування API за допомогою Swagger

Результат тестування POST-запиту для реєстрації клієнта.

Code	Details
201 <i>Undocumented</i>	Response body <pre>{ "message": "Client created successfully" }</pre>

9

Висновок

- Розроблено веб застосунок для управління кінотеатром.
- Реалізовано повноцінну тривірневу архітектуру з чітким розподілом відповідальності;
- Забезпечено автентифікацію та авторизацію з використанням ASP.NET Core Identity + JWT;
- Впроваджено Swagger для документування та перевірки роботи API;
- Підготовлено основу для подальшого розширення функціоналу (наприклад, інтеграція платіжних систем, система лояльності).

10