

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики

Курсова робота

За спеціальністю 122 Комп'ютерні науки

на тему:

ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №1

Виконав студент 2 курсу

Бобиль Мирослав Володимирович

Науковий керівник:

асистент

Белова Анна Сергіївна

Засвідчую, що в цій курсовій роботі
немає запозичень з праць інших авторів
без відповідних посилань

Київ – 2024

Зміст

Реферат	3
Скорочення та умовні позначення	4
Вступ	5
Розділ 1. Огляд використаних технологій	7
Розділ 2. Призначення і цілі створення системи	8
Розділ 3. Опис організації інформаційної бази	11
Розділ 4. Реалізація системи	15
Розділ 5. Інструкція користувачу	18
Висновки	20
Перелік використаних джерел	21

Реферат

Ключові слова : ASP.NET CORE, ENTITY FRAMEWORK, MICROSOFT VISUAL STUDIO 2022, .NET, ВЕБ-ЗАСТОСУНОК, ПЗ, РОЗРОБКА ПЗ, ФОРМА, ЗАПИТ, ПРЕДСТАВЛЕННЯ, DATA FIRST, DB.

Об'єктом дослідження є процес реалізації сервісу для управління мережею кінотеатрів, а саме – процесом організації показу сеансів та продажу квитків.

Предметом дослідження є служба управління базою даних мережі кінотеатрів.

Метою даної роботи є розробка служби управління мережею кінотеатрів, а саме показом фільмів, продажем квитків, наповненням фільмотеки.

Методи дослідження: спостереження, порівняння та аналіз.

Інструментальні розроблення: платформа .NET Core, технологія роботи з даними Entity Framework Core, СКБД SQL Management Studio, база даних SQL Server, організація архітектури серверного застосунку на основі ASP.NET Core з використанням багаторівневої архітектури, середовище розробки Microsoft Visual Studio 2022.

Скорочення та умовні позначення

API – application programming interface, прикладний програмний інтерфейс;

ASP – Active Server Pages, технологія створення веб-застосунків і веб-сервісів від компанії Microsoft;

EF Core – Entity Framework Core, об'єктно-орієнтована, легковажна технологія;

HTML – HyperText Markup Language, мова гіпертекстової розмітки;

IDE – Integrated Design Environment, інтегроване середовище розробки;

LINQ – Language Integrated Query, мова запитів для платформи програмування .NET;

.NET – платформа програмування компанії Microsoft;

NuGet – система керування пакетами для платформ Microsoft;

ORM – Object-Relational Mapping, об'єктно-реляційне відображення;

SQL – Structured query language, мова структурованих запитів;

WEB – World Wide Web, всесвітня мережа;

БД – база даних;

Вступ

Актуальність. Розробка інтернет-сервісу керування мережею кінотеатрів та його використання є актуальним питанням на сьогоднішній день. В світі і особливо в Україні, величезними темпами росте кількість користувачів Інтернет і, як наслідок, кількість «електронних» замовників.

Мета дослідження - розробка інтернет-сервісу для керування базою даних та створення продажу квитків.

Завдання дослідження – розробка алгоритму функціонування інтернет-сервісу програмних засобів, проектування структури інтернет-сервісу, керування базою даних мережі кінотеатрів, внесених в базу їх персоналом записів; поглиблення та закріплення знань з технологій ASP.NET MVC та Entity Framework Core – Data First

Предметом дослідження є веб-сервіс, що реалізує ідею сервісу для надання можливості керувати базою даних мережі кінотеатрів.

Методи дослідження, використані у роботі:

- аналіз;
- моделювання;
- спостереження.

Використані інструменти для досягнення остаточного результату:

- платформа .NET;
- інтегроване середовище розробки Visual Studio 2022;
- мова програмування C#;
- додаток для створення веб-застосунків ASP.NET Core;
- WPS Office для створення даного звіту.

Практична цінність отриманих результатів полягає у використанні інтернет – сервісу для комерційних цілей. Взаємозв'язок з іншими роботами. В процесі роботи були

використані та поглиблені знання з створення та зміни баз даних за допомогою Data First Workflow, а також набуті знання в роботі з MVC мовою програмування C#

Розділ 1. Огляд використаних технологій

1.1 Entity Framework Core Data First - ORM Framework

Об'єктно-орієнтована, розширювана технологія від компанії Microsoft для доступу до даних. EF Core є ORM-інструментом (object-relational mapping – відображення даних на реальні об'єкти). Тобто EF Core дозволяє працювати базами даних, але є більш високий рівень абстракції: EF Core дозволяє абстрагуватися від самої бази даних та її таблиць та працювати з даними незалежно від типу сховища. Якщо фізично ми оперуємо таблицями, індексами, первинними та зовнішніми ключами, але на концептуальному рівні, який нам пропонує Entity Framework, ми вже працюємо з об'єктами.

1.2 ASP.NET MVC

ASP.NET MVC (Model-View-Controller) – це фреймворк для створення веб-додатків, розроблений компанією Microsoft. Він реалізує патерн MVC, який розділяє додаток на три основні компоненти: модель (Model), уявлення (View) та контролер (Controller). Цей підхід дозволяє розробникам створювати веб-додатки, які легко підтримувати та розширювати.

Основні компоненти ASP.NET MVC

1. Model (Модель):

- Відповідає за бізнес-логіку додатку та роботу з даними.
- Містить дані, що надходять з бази даних, і методи для їхньої обробки.
- Зазвичай моделі є простими об'єктами (Plain Old CLR Objects, POCOs), що представляють структуру даних.

2. View (Уявлення):

- Відповідає за відображення інтерфейсу користувача.
- Уявлення отримує дані з моделі та формує HTML для відправлення на клієнт.
- Часто використовуються Razor шаблони для створення уявлень.

3. Controller (Контролер):

- Відповідає за обробку вхідних запитів від користувача.
- Викликає методи моделі для обробки даних і передає результати уявленню.
- Контролери містять дії (action methods), які відповідають на різні запити (HTTP GET, POST тощо).

1.3 Bootstrap

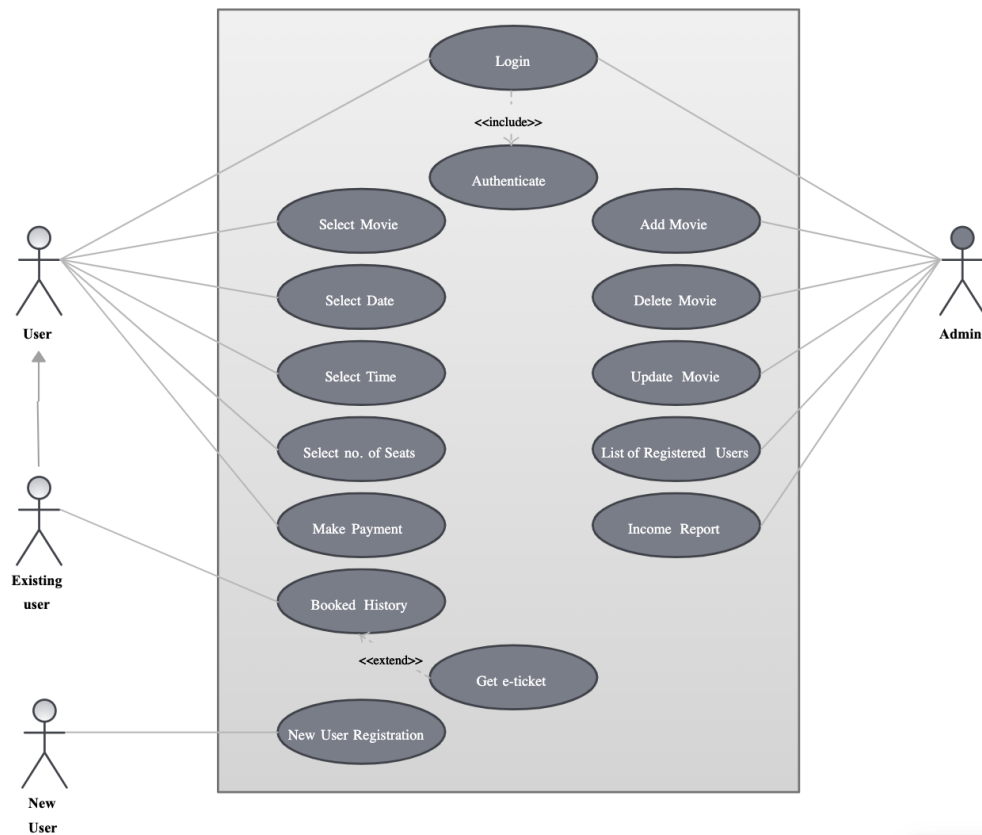
Bootstrap — це популярний фреймворк для розробки адаптивних (responsive) і мобільно-орієнтованих веб-сайтів та веб-додатків. Він містить набір інструментів для створення інтерфейсів користувача, таких як шаблони HTML та CSS для типографії, форм, кнопок, навігаційних елементів та інших компонентів інтерфейсу. Bootstrap також включає JavaScript-плагіни для покращення функціональності.

1.4 JavaScript

Потужна мова програмування, яка може додати веб-сайту інтерактивності. Його винайшов Брендан Айх (співзасновник проекту Mozilla, Mozilla Foundation і Mozilla Corporation). JavaScript є універсальним і зручним для початківців.

Розділ 2. Призначення і цілі створення системи

2.1 Призначення системи



Use-case діаграма

Призначення веб-системи «CMS Web App» є автоматизація процесу управління базою даних мережі кінотеатрів, що дає змогу ефективно керувати усіма задачами, від створення сеансу, продажу квитків, до аналізу попиту на ту чи іншу кінопродукцію в той чи інший час.

Робота передбачає:

- Аналіз методів, методик і моделей, що застосовуються для розв’язання задач створення комплексних вебсистем
- Проектування та програмну реалізацію системи «CMS Web App»

2.2 Цілі створення системи

«CMS Web App» створюється з метою надання користувачам системи управління базою даних, організації її в єдину систему, задля поліпшення та полегшення робочого процесу.

2.3 Вимоги до системи

Система повинна будуватись з використанням підходів ООП, з уніфікацією програмно-технічних засобів розробки прикладної функціональності з використанням сучасних технологій. А також система має бути реалізована з використанням технології Data First.

2.3.1 Вимоги до системи в цілому

Система «CMS Web App» повинна реалізовувати функції створення, редагування та видалення сеансів, фільмів, кінотеатрів, залів та квитків.

В системі передбачається виділити наступні функціональні підсистеми:

- адміністративна, призначена для керування базою даних, зміною записів;
- підсистема користувача, призначена для перегляду вмісту бази даних;

2.3.2 Вимоги до функції, які виконуються системою

Підсистема адміністратора

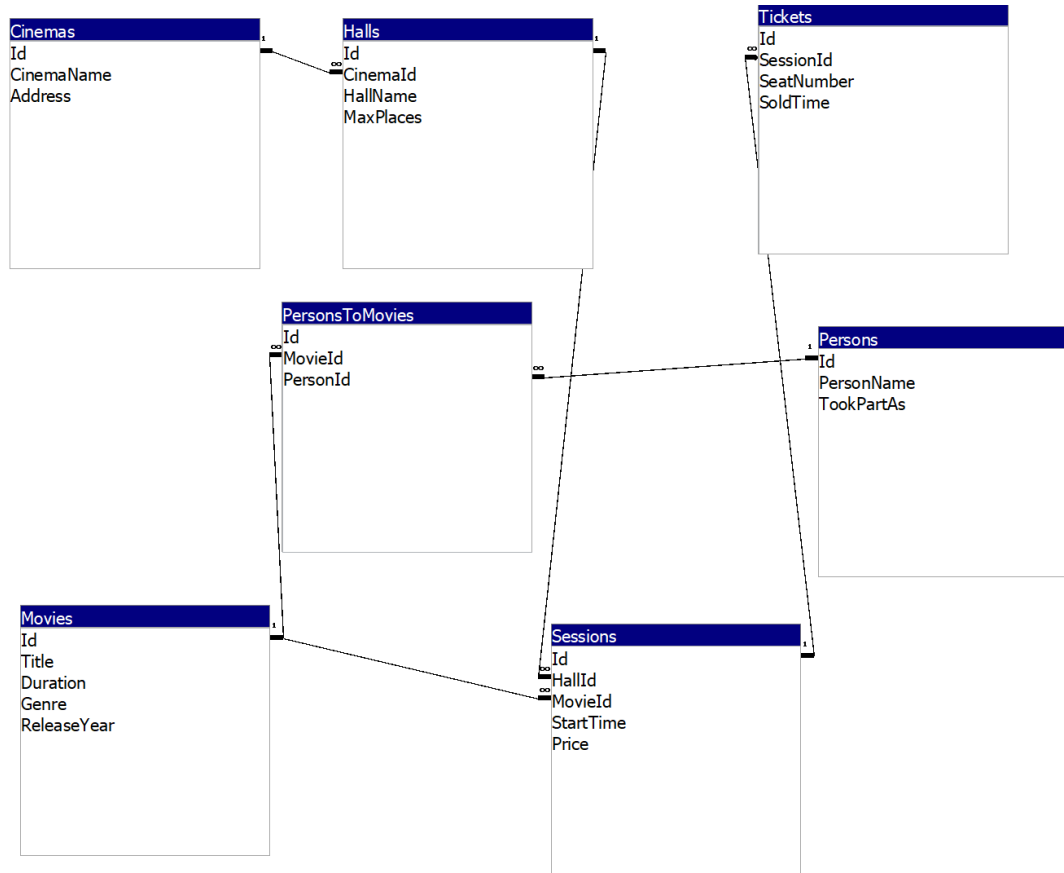
Функція	Задача
Робота з базою даних	Перегляд, створення, оновлення та видалення даних про кінотеатри, сеанси, фільми та квитки
Робота з замовленнями	Перегляд та аналіз попиту

Підсистема користувача

Функція	Задача
Робота з базою даних	Перегляд, даних про фільми, сеанси, багатокритеріальний пошук
Робота з замовленнями	Покупка квитків

Розділ 3. Опис організації інформаційної бази

3.1 Логічна структура бази даних



Діаграма бази даних.

Перелік таблиць системи «Restaurant API»

Номер	Таблиця	Опис
1	Cinemas	Таблиця для збереження інформації кінотеатри
2	Halls	Таблиця для збереження інформації про зали кінотеатрів
3	Sessions	Таблиця для збереження інформації про сеанси
4	Movies	Таблиця для збереження інформації про фільми
5	Persons	Таблиця для збереження інформації про

		важливих персон, які брали участь в зйомці фільму
6	PersonsToMovies	Таблиця для збереження інформації про зв'язок між учасниками зйомки та фільмами
7	Tickets	Таблиця для збереження інформації про куплені квитки

3.2 Опис таблиць

Таблиця Cinemas

Атрибут	Тип	Опис
Id	int	Ідентифікатор кінотеатру
CinemaName	Nvarchar(255)	Назва кінотеатру
Adress	Nvarchar(255)	Адреса кінотеатру

Таблиця Halls

Атрибут	Тип	Опис
Id	int	Ідентифікатор зали
CinemaId	int	Зовнішній ключ
HallName	Nvarchar(255)	Назва зали
MaxPlaces	int	Кількість доступних місць у залі

Таблиця Sessions

Атрибут	Тип	Опис
Id	int	Ідентифікатор сеансу
HallId	int	Зовнішній ключ

MovieId	int	Зовнішній ключ
StartTime	DateTime	Початок сеансу
Price	Currency	Ціна квитка

Таблиця Tickets

Атрибут	Тип	Опис
Id	int	Ідентифікатор квитка
SessionId	int	Зовнішній ключ
SoldTime	DateTime	Час продажу
SeatNumber	int	Номер місця

Таблиця Movies

Атрибут	Тип	Опис
Id	int	Ідентифікатор фільма
Title	Nvarchar(50)	Назва
Genre	Nvarchar(50)	Жанр
ReleaseYear	int	Рік випуску
Duration	int	Тривалість у хвилинах

Таблиця Persons

Атрибут	Тип	Опис
Id	int	Ідентифікатор персони
PersonName	Nvarchar(50)	Ім'я
TookPartAs	Nvarchar(50)	В якій ролі брав участь під час зйомки

Таблиця PersonsToMovies

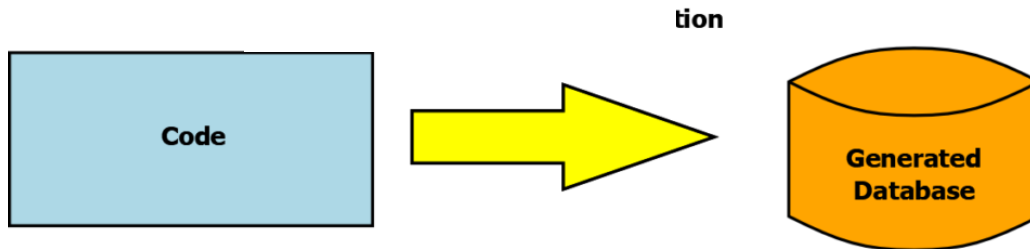
Атрибут	Тип	Опис
---------	-----	------

Id	int	Ідентифікатор зв'язку
PersonId	int	Ідентифікатор персони
MovieId	int	Ідентифікатор фільму

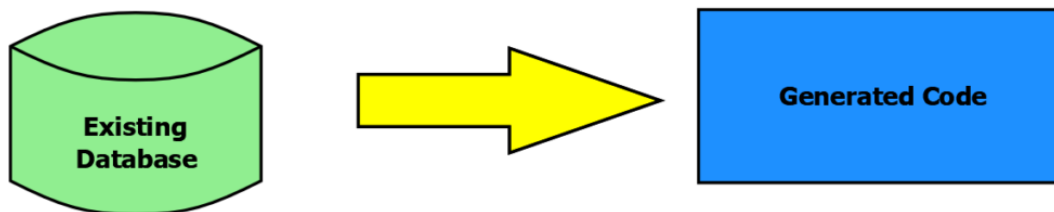
Розділ 4. Реалізація системи

4.1.Бази даних

Весь процес кодування виконувався в середовищі програмування Visual Studio 2022 на мові програмування C# із застосуванням технології Entity Framework Core. Був застосований підхід Data First.



Database First or Reverse Engineering



Для того щоб використовувати БД у програмі, необхідно під'єднатися до SQL-сервера, вказавши `ConnectionString`, і створити необхідні моделі та контекст з цими моделями.

4.2 Створення моделей

За замовчуванням всі типи сутностей, для яких визначені в контексті даних набори `DbSet`, включаються в модель і надалі зіставляються з таблицями в базі даних. Але крім того, в модель також включаються типи, на які є посилання в сутності, які вже включені в модель, наприклад, через властивості `DbSet`.

За замовчуванням кожна сутність зіставляється з таблицею, яка називається за іменем властивості `DbSet <T>` в контексті даних, що представляє дану сутність. Якщо в контексті даних подібного властивості не визначено, то для назви таблиці використовується ім'я класу сутності.

```

namespace CMSWebAppLab1.Models;

27 references
public partial class Session
{
    27 references
    public int Id { get; set; }

    9 references
    public int HallId { get; set; }

    9 references
    public int MovieId { get; set; }

    21 references
    public DateTime StartTime { get; set; }

    17 references
    public decimal Price { get; set; }

    26 references
    public virtual Hall Hall { get; set; } = null!;

    37 references
    public virtual Movie Movie { get; set; } = null!;

    3 references
    public virtual ICollection<Ticket> Tickets { get; set; } = new List<Ticket>();
}

```

Модель для зберігання сезону

```

public partial class ApplicationDbContext : IdentityDbContext
{
    0 references
    public ApplicationDbContext()
    {
    }

    0 references
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
        : base(options)
    {
    }

    17 references
    public virtual DbSet<Cinema> Cinemas { get; set; }

    13 references
    public virtual DbSet<Hall> Halls { get; set; }

    19 references
    public virtual DbSet<Movie> Movies { get; set; }

    13 references
    public virtual DbSet<Person> Persons { get; set; }

    7 references
    public virtual DbSet<PersonsToMovie> PersonsToMovies { get; set; }

    16 references
    public virtual DbSet<Session> Sessions { get; set; }

    10 references
    public virtual DbSet<Ticket> Tickets { get; set; }
}

```

Контекст

4.3 Створення контролера

Контролери можна створити автоматично при наявності коректної моделі даних та працюючого контексту з'єднання з БД – цього достатньо для автогенерації не тільки контролера, а і вигляду для перегляду і редагування даних – так званих CRUD-операцій.

```
[Authorize(Roles = "Admin,User")]
0 references
public async Task<IActionResult> SearchSessions(SessionSearchViewModel searchModel)
{
    var query = _context.Sessions
        .Include(s => s.Movie)
        .ThenInclude(m => m.PersonsToMovies)
        .ThenInclude(ptm => ptm.Person)
        .Include(s => s.Hall)
        .ThenInclude(h => h.Cinema)
        .AsQueryable();

    if (!string.IsNullOrEmpty(searchModel.Title))
    {
        query = query.Where(s => s.Movie.Title.Contains(searchModel.Title));
    }

    if (!string.IsNullOrEmpty(searchModel.ActorName))
    {
        query = query.Where(s => s.Movie.PersonsToMovies.Any(ptm => ptm.Person.TookPartAs
    }

    if (!string.IsNullOrEmpty(searchModel.DirectorName))
    {
        query = query.Where(s => s.Movie.PersonsToMovies.Any(ptm => ptm.Person.TookPartAs
    }

    var result = await query.Select(s => new SessionResultViewModel
    {
        Title = s.Movie.Title,
        ActorName = s.Movie.PersonsToMovies.Where(ptm => ptm.Person.TookPartAs == "Actor"),
        DirectorName = s.Movie.PersonsToMovies.Where(ptm => ptm.Person.TookPartAs == "Dire
        StartTime = s.StartTime,
        Price = s.Price,
```

Контролер для обробки запитів для пошуку сеансу по назві фільму, імені актора або режисера з реалізацією авторизації та розділення ролей

Розділ 5. Інструкція користувачу

Користувач має доступ до всіх функцій з головної сторінки.
Додатково, імплементовано функцію реєстрації та авторизації.

Система Управління Кінотеатрами

Головна

Політика приватності

Привіт myroslavboby@gmail.com! Вийти

Вітаємо в Системі Управління Кінотеатрами

Кінотеатри

Зали

Сеанси

Пошук Сеансу

Квитки

Фільми

Персони

Персони у фільмах

Available Sessions

Фільм	Кінотеатр	Зала	Початок	Ціна	
Lock, stock andtwo smoking barrels	Cinema_1	Hall_1	6/3/2024 1:00:00 PM	\$2.00	Купити квиток
Lock, stock andtwo smoking barrels	Cinema_1	Hall_1	6/3/2024 2:45:00 PM	\$2.00	Купити квиток
Terminator 2: Judgement Day	Cinema_2	Hall_1	7/3/2024 5:00:00 PM	\$3.00	Купити квиток
Avengers	Cinema_1	Hall_1	9/3/2024 9:00:00 AM	\$10.00	Купити квиток
Die Hard	Cinema_2	Hall_3	10/3/2024 9:00:00 AM	\$5.00	Купити квиток
John Wick: Chapter 4	Cinema_3	Hall_1	3/20/2024 10:00:00 AM	\$25.00	Купити квиток
Lock, stock andtwo smoking barrels	Cinema_3	Hall_1	6/3/2024 1:00:00 PM	\$3.00	Купити квиток

Головна сторінка

Система Управління Кінотеатрами

Головна

Політика приватності

Зареєструватися

Війти

Register

Create a new account.

Email

Password

Confirm Password

Register

Use another service to register.

There are no external authentication services configured. See this [article about setting up this ASP.NET application to support logging in via external services](#).

© 2024 - Система Управління Кінотеатрами - [Політика приватності](#)

Форма реєстрації

Log in

Use a local account to log in.

☐ Remember me?[Forgot your password?](#)[Register as a new user](#)[Resend email confirmation](#)

Use another service to log in.

There are no external authentication services configured. See this [article about setting up this ASP.NET application to support logging in via external services](#).

© 2024 - Система Управління Кінотеатрами - [Політика приватності](#)

Форма входу

У сервісі, кожен користувач має 1 з двох ролей – Admin, який має право на будь які операції, та User – який може виконувати лише операції читання. Єдине виключення – це операція покупки квитків, яка міняє значення в БД.

Для кожної із 7 таблиць реалізовані CRUD-операції: Create, Read, Update, Delete.

Додатково, імплементовано генерацію звітів у форматі .xlsx та .docx по таблиці Cinemas: отримати список всіх сеансів із відповідним даними по кожному кінотеатру.

CRUD: Кінотеатри

[Створити новий запис](#)

Назва	Адреса	
Cinema_1	Location_1	Редагувати Деталі Видалити Показати зали Показати сеанси
Cinema_2	Location_2	Редагувати Деталі Видалити Показати зали Показати сеанси
Cinema_3	Location_3	Редагувати Деталі Видалити Показати зали Показати сеанси
Cinema_4	Location_4	Редагувати Деталі Видалити Показати зали Показати сеанси
Cinema_5	Location_5	Редагувати Деталі Видалити Показати зали Показати сеанси
Cinema_6	Location_6	Редагувати Деталі Видалити Показати зали Показати сеанси

Оберіть формат звіту:

Format Painter

Clipboard

Font

Alignment

Nur

A1

Назва зали

	A	B	C	D	E	F	G	H	I	J	K
1	Назва зали	Назва філь	Початок се	Ціна							
2	Hall_1	Lock, stock	2024-06-03	\$2.00							
3	Hall_1	Lock, stock	2024-06-03	\$2.00							
4	Hall_1	Avengers	2024-09-03	\$10.00							
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											

Cinema_1

Cinema_2

Cinema_3

Cinema_4

Cinema_5

Cinema_6

+

Ready Accessibility: Good to go

Normal

No Spacing

Heading 1

Editing

Dictate

Editor

Add-ins

Paragraph

Styles

Voice

Editor

Add-ins

Double-click to hide white space

Cinema_1

Назва зали	Назва фільму	Початок сесії	Ціна
Hall_1	Lock, stock andtwo smoking barrels	6/3/2024 1:00 PM	\$2.00
Hall_1	Lock, stock andtwo smoking barrels	6/3/2024 2:45 PM	\$2.00
Hall_1	Avengers	9/3/2024 9:00 AM	\$10.00

Також, імплементовано можливість багатокритеріального пошуку сеансу і покупки квитка:

Пошук сеансу

Фільм:

Ter

Актор:

Режисер:

Пошук

Результат пошуку:

Фільм	Актор	Режисер	Початок	Ціна	Кінотеатр	Зал
Terminator 2: Judgement Day	Arnold Shwarzenegger	James Cameron	7/3/2024 5:00 PM	\$3.00	Cinema_2	Hall_1
John Wick: Chapter 4	Keanu Reaves	Chad StahelskiChad Stahelski	3/20/2024 10:00 AM	\$25.00	Cinema_3	Hall_1

Available Sessions

Фільм	Кінотеатр	Зала	Початок	Ціна	
Lock, stock andtwo smoking barrels	Cinema_1	Hall_1	6/3/2024 1:00:00 PM	\$2.00	Купити квиток
Lock, stock andtwo smoking barrels	Cinema_1	Hall_1	6/3/2024 2:45:00 PM	\$2.00	Купити квиток
Terminator 2: Judgement Day	Cinema_2	Hall_1	7/3/2024 5:00:00 PM	\$3.00	Купити квиток
Avengers	Cinema_1	Hall_1	9/3/2024 9:00:00 AM	\$10.00	Купити квиток
Die Hard	Cinema_2	Hall_3	10/3/2024 9:00:00 AM	\$5.00	Купити квиток
John Wick: Chapter 4	Cinema_3	Hall_1	3/20/2024 10:00:00 AM	\$25.00	Купити квиток
Lock, stock andtwo smoking barrels	Cinema_3	Hall_1	6/3/2024 1:00:00 PM	\$3.00	Купити квиток

Висновки

Було розроблено систему, що надає користувачу можливість керування базою даних мережі кінотеатрів, зокрема переглядати, видаляти, створювати та редагувати кожну таблицю даних.

У процесі роботи над розробкою веб-додатку навчився новій технології з розробки веб-додатків, поглибив знання з HTML та CSS, а також ознайомився з технологією Data First.

В результаті роботи над проектом було окремо розроблено backend та frontend частину, що в майбутньому дозволить реалізувати повноцінний програмний продукт «CMS Web App», що буде відповідати усім технічним вимогам. Цей досвід дуже важливий для підвищення технічних навичок, покращення знань у сфері інформаційних технологій та оформленні необхідної документації до проекту.

Перелік використаних джерел

1. Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://bootstrap.com/>
2. ASP.NET MVC documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started/>
3. Entity Framework documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/ef/>
4. JavaScript documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://devdocs.io/javascript/>