**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**
**ІМЕНІ ТАРАСА ШЕВЧЕНКА**
Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

**Кваліфікаційна робота**
**на здобуття ступеня бакалавра**
за освітньо-науковою програмою «Інформатика»
спеціальності 122 «Комп'ютерні науки»
на тему:

# СТРІМІНГОВИЙ СЕРВІС З ПОШУКУ ВІДЕОКОНТЕНТУ

Виконав студент 4-го курсу бакалавру
Кирил Ампа Джіованні Атассе ДЖОНСОН _____
(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Людмила ОМЕЛЬЧУК _____
(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Роботу розглянуто і допущено до захисту на засіданні кафедри теорії та технології програмування
« 05 » червня 2023 р.,
протокол № 18
Завідувач кафедри
Микола НІКІТЧЕНКО _____
(підпис)

Київ – 2023

# РЕФЕРАТ

Обсяг роботи складає 47 сторінок, з них основний текст складає 42 сторінки, 13 ілюстрацій, 8 таблиць, 26 джерела посилань.

БАЗА ДАНИХ, ВЕБЗАСТОСУНОК, .NET, ANGULAR, ASP.NET CORE, ENTITY FRAMEWORK, ANGULAR, FYGMA, SQL.

Об’єктом даної роботи є процес перегляду інформації про фільми, перегляду відоконтенту та надання можливості нотифікації користувача стосовно попередження про наближення дати виходу фільму, інтерес до якого він виявив. Предметом дослідження є стрімінговий сервіс з пошуку відеоконтенту, призначена для надання можливості користувачам переглядати інформацію про фільми та акторів, а також надавати можливість перегляду доступного матеріалів.

Метою кваліфікаційної роботи є проєктування та розробка сервісу з перегляду доступного відеоконтенту та пошуку інформації про фільми.

Методи розроблення: моделювання та проєктування, розробка на основі ітеративної методології, налагодження та випробування програмного забезпечення. Інструменти розроблення: інтегроване середовище розробки Visual Studio 2022, Visual Studio Code, SQL Server Management Studio.

Результати роботи: розглянуто процес роботи з інформацією про відеоконтент, визначено способи автоматизації даного процесу, проаналізовано наявні сервіси, які призначені для перегляду відеоконтенту та інформації про нього, розроблено інтерфейс користувача, спроєктовано та реалізовано базу даних.

# ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ ..... 5
ВСТУП ..... 6
РОЗДІЛ 1. ОГЛЯД СТРІМІНГОВИХ СЕРВІСІВ ПОШУКУ ВІДЕОКОНТЕНТА ..... 11
1.1. Сервіс «Netflix» ..... 11
1.2. Сервіс «Amazon» ..... 13
1.3. Сервіс «MEGOGO» ..... 14
1.4. Сервіс «SWEET.TV» ..... 15
РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ ..... 17
2.1. Технологія ASP.NET Core ..... 17
2.2. Технологія Entity Framework Core ..... 18
2.3. Фреймворк Angular ..... 20
2.4. Фреймворк Bootstrap ..... 22
РОЗДІЛ 3. ПРИЗНАЧЕННЯ ТА ЦІЛІ СТВОРЕННЯ ВЕБЗАСТОСУНКУ ..... 24
3.1. Призначення застосунку ..... 24
3.2. Вимоги до вебзастосунку ..... 24
3.3. Технічні вимоги до проєкту ..... 26
3.4. Вимоги законодавства України щодо стрімінгового сервісу ..... 27
РОЗДІЛ 4. РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ..... 29
4.1. Вимоги до структури бази даних ..... 29
4.2. Програмна реалізація системи ..... 33
РОЗДІЛ 5. ІНСТРУКЦІЯ КОРИСТУВАЧА ..... 34
5.1 Інструкція для реєстрації ..... 34

ВИСНОВКИ ..... 39

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ..... 40

ДОДАТКИ ..... 43

Додаток А. Use-case діаграма ..... 43

Додаток Б. Моделі проєкту ..... 44

Додаток В. Angular реалізація фронтенду на прикладі реєстрації ..... 46

# СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

**API –** Application programming interface; прикладний програмний інтерфейс;

**CSS –** Cascading Style Sheets; каскадні таблиці стилів;

**EF –** Entity Framework; структура об'єктно-реляційного відображення;

**HTML –** HyperText Markup Language; мова гіпертекстової розмітки;

**HTTPS –** HyperText Transfer Protocol Secure; протокол передачі даних;

**IDE –** Integrated development environment; інтегроване середовище розробки;

**LINQ –** Language Integrated Query; запити, інтегровані в мову;

**MVC –** Model–View–Controller; модель-представлення-контролер;

**ORM –** Object-relational mapping; об'єктно-реляційна проєкція;

**REST –** Representational State Transfer; підхід до архітектури мережевих протоколів;

**SQL –** Structured Query Language; мова структурованих запитів;

**WEB –** World Wide Web; система доступу до пов'язаних між собою документів на різних комп'ютерах, підключених до Інтернету;

**БД –** База даних;

**СКБД –** Система керування базами даних;

**Wi-Fi –** Wireless Fidelity; загальновживана назва для стандарту передавання цифрових потоків даних по радіоканалах;

**HD —** High Density; висока роздільна здатність;

**4k —** 4k resolution; роздільна здатність приблизно в 4 000 пікселів по горизонталі.

# ВСТУП

**Оцінка сучасного стану об’єкта розробки.** На сьогоднішній день онлайн кінотеатри, служби управління відеоконтентом та стрімінгові сервіси мають значний вплив на спосіб, яким українські глядачі споживають відеоконтент. Стан розвитку цих сервісів в Україні можна оцінити як динамічний і постійно зростаючий.

За останні кілька років українські глядачі мали доступ до різноманітних стрімінгових платформ, таких як Netflix [1], Apple TV+ [2], Amazon Prime Video [3], Google Play Movies [4], Megogo [5] та інші. Ці сервіси надають можливість переглядати широкий спектр фільмів, серіалів та інших відеоматеріалів у будь-який час та з будь-якого пристрою з доступом до Інтернету.

Онлайн кінотеатри також стають все популярнішими в Україні [6]. Вони надають можливість переглядати останні фільми у високій якості без необхідності відвідування традиційних кінотеатрів. Глядачі можуть вибрати фільм, встановити зручний для них час перегляду та насолоджуватись кіно в затишку свого домівки.

Однак, варто відзначити, що конкуренція серед стрімінгових сервісів в Україні стає все жорсткішою [6]. Це спонукає провайдерів послуг залучати нових абонентів шляхом пропозицій різноманітного вмісту, оригінальних продукційних проєктів та зручних функцій.

Окрім того, важливо відзначити, що доступність високошвидкісного Інтернету та розвиток технологій мають значний вплив на зростання онлайн кінотеатрів і стрімінгових сервісів.

Забезпечення стабільного та швидкого Інтернет-з'єднання є ключовим фактором для успішного функціонування онлайн кінотеатрів та стрімінгових сервісів. Широкосмуговий доступ до Інтернету стає все більш поширеним українським глядачам, що сприяє зростанню популярності цих сервісів.

Однак, на сьогоднішній день існує кілька викликів, з якими стикаються українські онлайн кінотеатри та стрімінгові сервіси [7]. Перш за все, авторське право та ліцензування є важливими аспектами, які необхідно враховувати при наданні доступу до вмісту. Необхідно забезпечувати дотримання правових норм і укладати ліцензійні угоди з власниками контенту, щоб запобігти порушенню авторських прав.

Також важливим фактором є конкуренція з боку інших медіа-платформ та каналів розповсюдження контенту. Зростаюча кількість стрімінгових сервісів та онлайн платформ, які пропонують відео контент, призводить до збільшення конкуренції за увагу глядачів. Українські онлайн кінотеатри та стрімінгові сервіси повинні постійно вдосконалювати свою пропозицію, надавати якісний вміст та інноваційні функції, щоб залучити та утримати аудиторію.

У підсумку, сучасний стан онлайн кінотеатрів та стрімінгових сервісів в Україні можна охарактеризувати як динамічний та зростаючий. Вони стають все популярнішими серед українських глядачів, які цінують зручність, доступність та різноманіття вмісту, які надають ці сервіси. Розвиток технологій і зростання швидкості Інтернету дозволяють людям з легкістю переглядати фільми та серіали у високій якості без необхідності завантаження файлів або відвідування фізичних кінотеатрів.

Онлайн кінотеатри та стрімінгові сервіси також сприяють поширенню іноземної кіноіндустрії в Україні, дозволяючи глядачам насолоджуватись широким вибором фільмів з усього світу. Вони стимулюють культурний обмін, сприяють розумінню та відкритості до різних культурних впливів.

Проте, існують певні виклики, з якими стикаються онлайн кінотеатри та стрімінгові сервіси в Україні. Одним з них є незаконне поширення авторського контенту та піратство, що може негативно впливати на дохід творців та ліцензіарів. Організації, що надають ці послуги, повинні вживати заходів для захисту авторських прав та забезпечення справедливої оплати за використання контенту.

У підсумку, онлайн кінотеатри та стрімінгові сервіси в Україні мають значний потенціал і продовжують розвиватись. Вони надають глядачам нові можливості для зручного та різноманітного перегляду кіно та відео контенту.

Однак, дотримання правових норм, боротьба з піратством та забезпечення якості та безпеки вмісту залишаються важливими викликами для розвитку онлайн кінотеатрів та стрімінгових сервісів в Україні. Потрібно продовжувати співпрацювати з авторами, студіями та правовласниками, а також впроваджувати технологічні рішення для захисту вмісту від незаконного поширення.

Зростання конкуренції в цій галузі також змушує сервіси шукати нові шляхи привернення глядачів і покращення їх досвіду. Інноваційні функції, персоналізація рекомендацій, введення оригінального контенту та партнерства зі зірками кіноіндустрії можуть стати ключовими факторами в привертанні аудиторії і забезпеченні конкурентоспроможності.

Загалом, стан онлайн кінотеатрів та стрімінгових сервісів в Україні є перспективним, але вимагає постійного вдосконалення та вирішення викликів, пов'язаних з авторськими правами, конкуренцією та якістю вмісту. Ці сервіси мають великий потенціал для задоволення потреб глядачів і розвитку кінематографії в Україні, якщо правильно використовувати технології та забезпечувати справедливу та ефективну співпрацю зі всіма сторонами індустрії.

**Актуальність роботи та підстави для її виконання.** Написання сайту кінотеатру є актуальним в контексті сучасних тенденцій у споживанні контенту і змін у способі, яким люди переглядають фільми та серіали. Нижче приведено обґрунтування актуальності такого сайту на основі попередньо сказаного:

- Зростання популярності онлайн контенту: за останні роки спостерігається зростання популярності онлайн перегляду фільмів та серіалів. Глядачі все більше віддають перевагу зручності та доступності онлайн платформ, де вони можуть вибирати з великого вибору фільмів і переглядати їх у зручний для себе час.
- Розширення аудиторії: наявність онлайн кінотеатру дозволить залучити широку аудиторію, включаючи тих, хто не має можливості або бажання

відвідувати фізичні кінотеатри. Це може бути особливо актуально для людей, які проживають у віддалених регіонах або не мають доступу до багаточисельних кінотеатрів.

- Персоналізований досвід: онлайн кінотеатр дозволяє глядачам налаштувати свій власний персоналізований досвід. Вони можуть створювати профілі, встановлювати вподобання та отримувати рекомендації на основі своїх інтересів. Це дозволяє покращити задоволення від перегляду фільмів та створює можливості для побудови вірного та залученого глядацького співтовариства.
- Можливість розширення бізнесу: онлайн кінотеатр відкриває додаткові можливості для кінотеатральних компаній розширити свої бізнеси.

**Мета й завдання роботи.** Метою написання роботи є проєктування та розробка стрімінгового сервісу створення функціонального, ефективного та зручного сервісу, який надає користувачам можливість переглядати кіно та серіалами в онлайн-режимі. Метою є забезпечення зручного і доступного інтерфейсу для користувачів, щоб вони могли знайти і переглянути вміст за їхніми вподобаннями та потребами. Для досягнення мети роботи було поставлено такі завдання:

- Дослідити стрімінгові сервіси, що функціонують в Україні.
- Дослідити технології для проєктування та розробки даного застосунку.
- Розробити структури та архітектури сервісу: створення ефективної та модульної структури проєкту з використанням ASP.NET Core, що дозволить розширювати та підтримувати застосунок у майбутньому.
- Реалізувати функціональності перегляду кінофільмів: розробка функціональності для пошуку, фільтрування та перегляду кінофільмів за різними категоріями, жанрами, акторами, режисерами тощо. Забезпечення можливості перегляду трейлерів та отримання детальної інформації про кожен фільм.

- Реалізувати взаємодію з користувачами: розробка функціоналу реєстрації та авторизації користувачів, що дозволить зберігати їхні вподобання, коментарі, оцінки фільмів та інші персоналізовані дані.
- Реалізувати інтеграцію з платіжними системами: забезпечення можливості оплати підписки або окремих фільмів для доступу до преміум контенту, використовуючи платіжні системи або інтегровані методи оплати.
- Провести оптимізацію продуктивності та безпеки: забезпечення швидкості та ефективності роботи вебзастосунку.

**Об'єкт, методи й засоби розроблення.** Об’єктом розробки є процес перегляду інформації про фільми, перегляду відоконтенту. Предметом дослідження є стрімінговий сервіс з пошуку відеоконтенту, призначена для надання можливості користувачам переглядати інформацію про фільми та акторів, а також надавати можливість перегляду доступного матеріалів.

Для розробки вебзастосунку використовувались такі методи, як моделювання та проєктування, розробка на основі ітеративної методології, розробка, тестування, налагодження та випробування програмного забезпечення.

Засоби розроблення включають Visual Studio 2022 [8] і Visual Studio Code [30] як основні інтегровані середовища розробки, SQL Server Management Studio [9] для управління базою даних, фреймворк Angular для розробки клієнтської частини вебзастосунку та мову SQL для створення та керування базою даних.

Також було використано інструмент дизайну Figma [10] для створення макетів та дизайну інтерфейсу користувача.

# РОЗДІЛ 1.
# ОГЛЯД СТРІМІНГОВИХ СЕРВІСІВ ПОШУКУ ВІДЕОКОНТЕНТА

На ринку доступно досить багато систем та конкурентів, які пропонують різноманітні стрімінгові сервіси в Україні. Кожна з цих систем має свої особливості, унікальний контент та функціональні можливості, спрямовані на задоволення потреб глядачів. Далі ми детальніше та розглянемо проаналізуємо кілька з них.

## 1.1. Сервіс «Netflix»

Netflix [1] є одним з найпопулярніших та найбільших стрімінгових сервісів у світі, з мільйонами підписників по всьому світу. Він пропонує широкий вибір відомих фільмів, популярних серіалів, анімаційних виробів, документальних стрічок та власного виробництва, яке отримало високу оцінку критиків. На рис. 1 наведено вигляд головної сторінки стрімінгового сервісу Netflix.

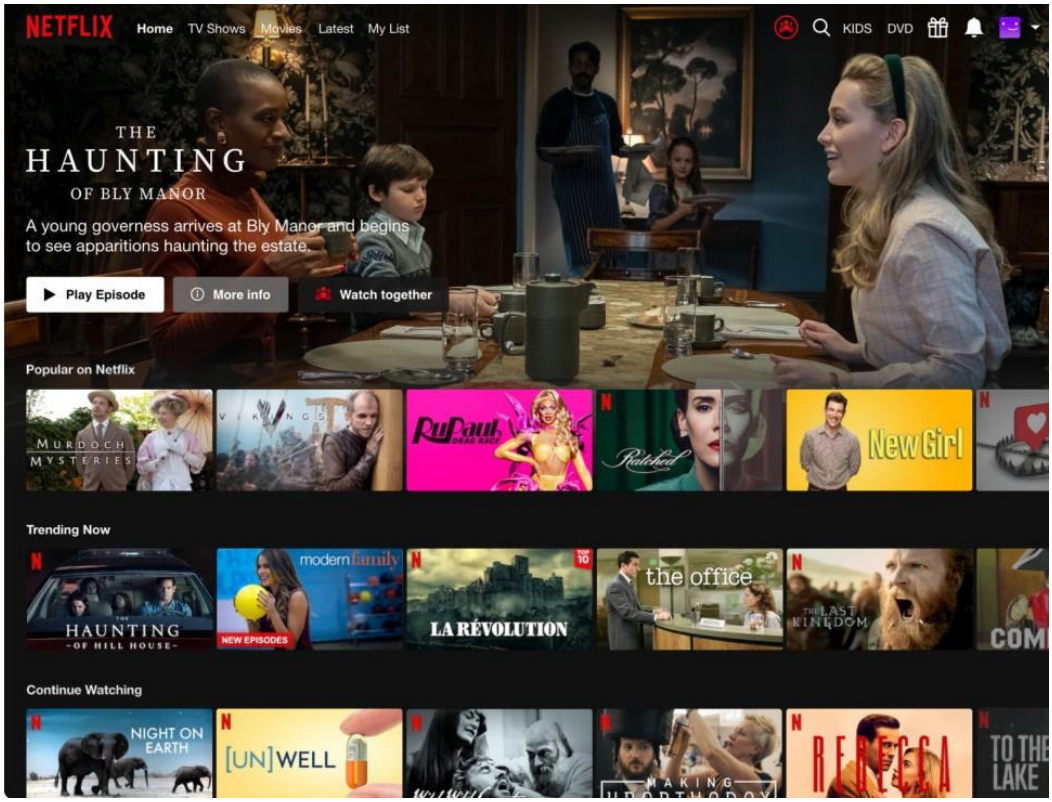



**Рисунок 1 –** Головна сторінка Netflix

Серед переваг сервісу Netflix виділимо наступні:

- Доступність та простота: Netflix – це дешевий, дуже простий і зрозумілий

сервіс. Для користування на комп'ютері або з ноутбуків вам просто потрібно зайти на сайт netflix.com та увійти у свій акаунт. Для мобільних пристроїв вам потрібно завантажити спеціальний застосунок.

- Можливість перегляду на різних пристроях: Netflix пропонує різні плани, включаючи преміум та базовий, які дозволяють одночасний перегляд на кількох пристроях. Особливо преміум-план ідеально підходить для родин, оскільки дозволяє одночасний перегляд на 4 пристроях.
- Офлайн перегляд: Netflix надає можливість завантажити контент і переглядати його в автономному режимі, навіть без доступу до Інтернету. Це дуже зручно, коли ви хочете переглянути фільми або серіали під час подорожей або відсутності мережі Wi-Fi.
- Простота відмови від передплати: Якщо ви вирішите припинити передплату на Netflix, це можна зробити всього за два кліки. Крім того, перший місяць використання сервісу є безкоштовним, що дозволяє оцінити всі його переваги перед прийняттям остаточного рішення.

Недоліки сервісу Netflix:

- Затримка оновлення контенту: нові фільми, серіали та шоу не з'являються на Netflix одразу після їх виходу. Є певна затримка між прем'єрою контенту і його доступністю на платформі.
- Географічні обмеження: доступність відеобібліотеки Netflix залежить від вашого місця проживання. Найбільш розширена вибірка контенту доступна для користувачів у США, тоді як інші країни можуть мати обмежений вибір.
- Обмежена кількість українського контенту: українська версія Netflix має обмежений вибір художніх та документальних фільмів, які мають український дубляж або озвучку. Кількість доступних українських контентів обмежена.
- Помісячна передплата: Netflix пропонує тільки помісячну передплату, що означає, що вам доведеться сплачувати за весь каталог контенту, незалежно від того, чи бажаєте ви переглядати лише один серіал або

фільм.

## 1.2. Сервіс «Amazon»

Незважаючи на те, що Amazon Prime Video [3] має на кілька десятків мільйонів підписників менше, вражає кількість користувачів, яка становить близько 150 мільйонів. В сервісі доступно майже 1000 серіалів і фільмів, які є як оригінальними, так і ліцензійними. Україна отримала доступ до Prime Video у 2016 році, проте він поки що не здобув широкої популярності [6]. На рис. 2 наведено вигляд головної сторінки стрімінгового сервісу Amazon.

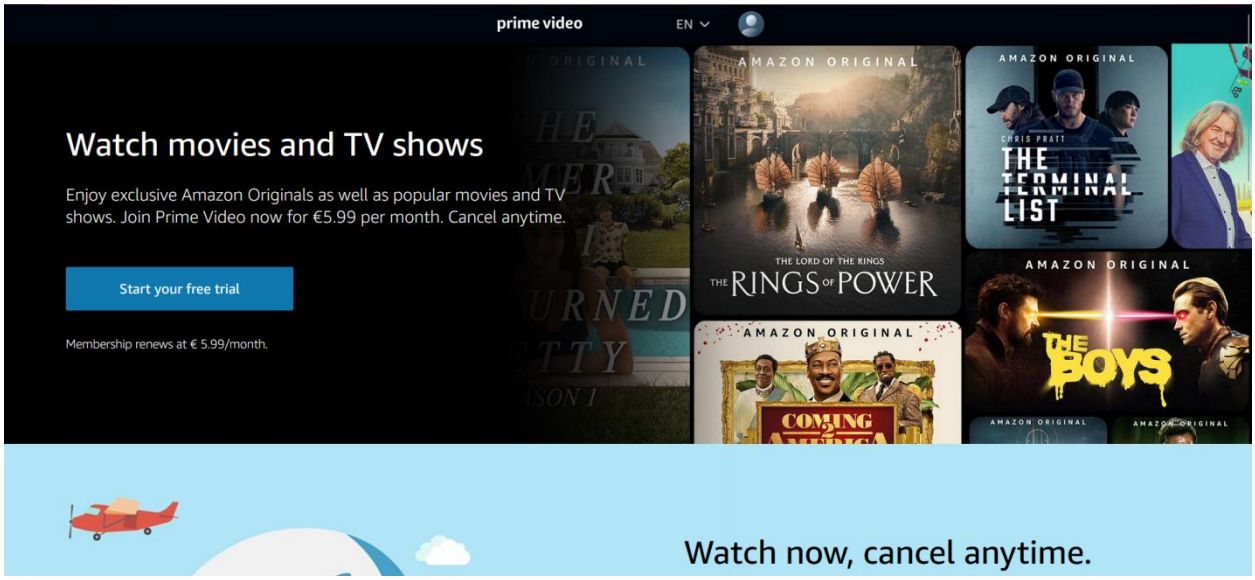



**Рисунок 2 –** Головна сторінка Amazon Prime Video

Переваги сервісу Amazon Prime Video:

- Зрозумілий інтерфейс і сервіс для скасування випадкових витрат усередині кінотеатру: гроші повернуть вам щонайбільше за 14 днів.
- Перевага – дійсно ексклюзивні проєкти, як «Велика мандрівка», або доступ до нових випусків програми Top Gear.

Недоліки сервісу Amazon Prime Video:

- Немає розуміння тарифного плану. Остаточно його можна дізнатися тільки після повної реєстрації у Prime Video, а відтак – лише після прив’язки банківської картки. Саме тому в тарифах значиться єдиний: за

5,99 євро.

- Ще один недолік – відсутній український інтерфейс та фільми з українським дубляжем. Основний контент – комерційні американські телеканали.
- Фільми, серіали, дитячі та розважальні проєкти є, але їх не так багато, як, скажімо, у конкурента Netflix. Похвалитися новинками чи ексклюзивними прем'єрами поки що не може.

## 1.3. Сервіс «MEGOGO»

Щомісячна аудиторія цього сервісу– понад 55 мільйонів користувачів, а каталог ресурсу налічує щонайменше 90 тисяч годин контенту. На рис. 3 наведено вигляд головної сторінки стрімінгового сервісу Megogo [5].

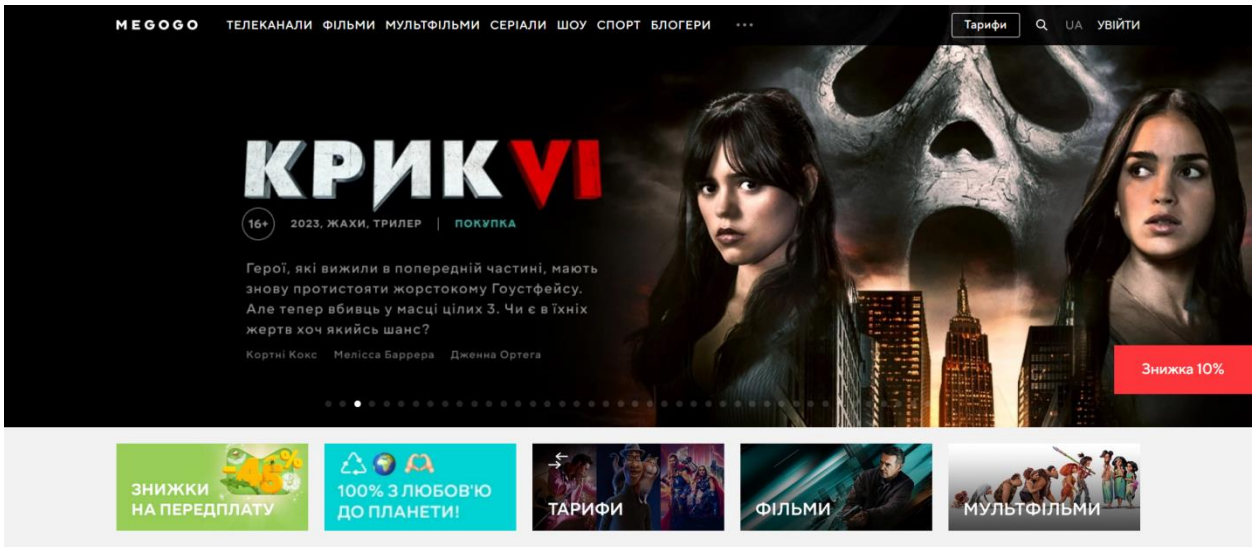



**Рисунок 3 –** Головна сторінка Megogo

Контент у розумінні Megogo – те, що можна і переглядати, і слухати. Сервіс пропонує фільми, серіали, шоу, тб-канали, аудіокниги та подкасти.

Переваги сервісу Megogo:

- Телеканали в записі та наявність фільмів з субтитрами.
- Велика спортивна добірка, що рідкісно для українських сервісів.
- Пробна версія максимальної передплати за 1 гривню на 14 днів.

- Можливість передплатити на 5 пристроїв одночасно.

Недоліки сервісу Megogo:

- Висока вартість та негуманна цінова політика.
- Обмежена кількість україномовного контенту.
- Часті випадки підвисання сервісу.
- Проблеми з відв'язкою банківської картки та непередбачуване списання коштів.

Загалом, на основі відгуків інтернет-користувачів, можна вважати, що у Megogo.net переважають недоліки, ніж плюси. Користувачі часто зазначають високу ціну, обмежений вибір україномовного контенту та проблеми з функціональністю та оплатою.

## 1.4. Сервіс «SWEET.TV»

Sweet TV [11] – національний онлайн-кінотеатр, який надає доступ до телебачення та кінозалу для всієї родини. Сервіс працює скрізь, де є Інтернет. На платформі представлено понад 260 телеканалів, включаючи канали в HD та 4K. Є функція управління прямим ефіром ТБ: пауза, перемотування та запис до 7 днів.

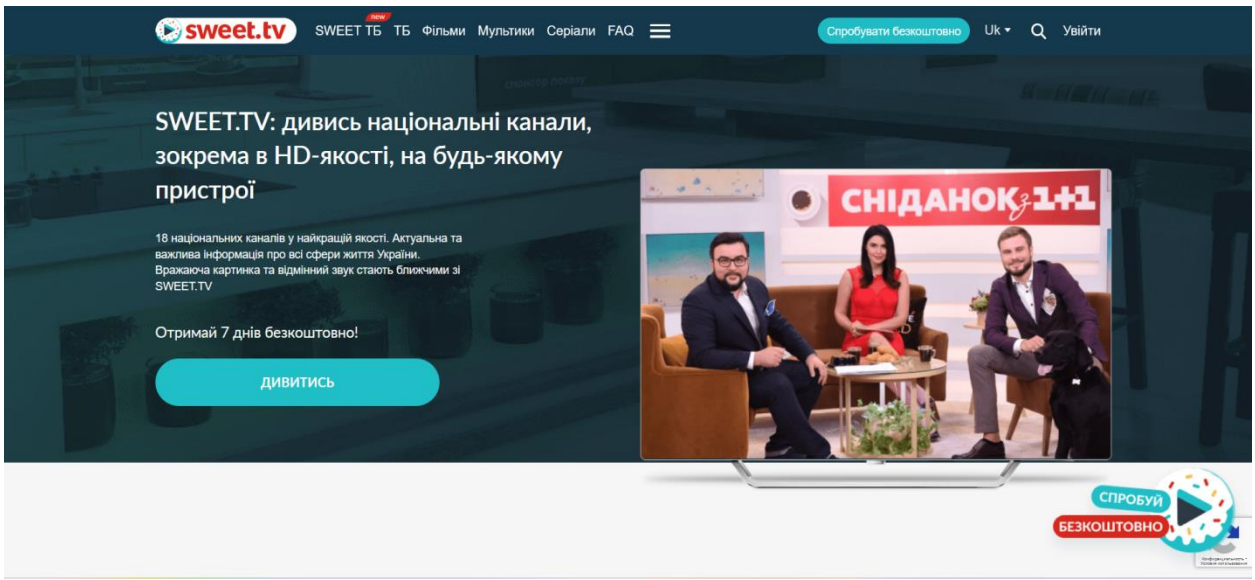



**Рисунок 3 –** Головна сторінка SWEET.TV

Переваги сервісу Sweet.tv:

- Широкий вибір телеканалів, включаючи канали у високій якості (HD і 4K).
- Функція управління прямим ефіром, така як пауза, перемотування і запис програм до 7 днів.
- Доступ до сервісу можливий будь-де, де є Інтернет.
- Можливість перегляду телебачення і кіно для всієї родини.
    - Недоліки сервісу Sweet.tv:
- Вартість передплати може бути високою в порівнянні з іншими подібними сервісами.
- Обмежений вибір україномовного контенту.
- Деякі користувачі можуть відзначати проблеми з функціональністю або стабільністю сервісу.

Незважаючи на наявність конкурентів, актуальність написання та запуску нового стрімінгового сервісу в Україні все ще існує. Ринок стрімінгових сервісів в Україні продовжує рости, і попит на цифровий контент, такий як фільми, серіали та телешоу, є значним. Запуск нового сервісу може привнести свіжість і інновації в цей сегмент ринку, пропонуючи нові можливості та вміст, які можуть привернути увагу аудиторії.

# РОЗДІЛ 2.
# ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

Для розробки та реалізації вебзастосунку використано наступні технології:

- технологія ASP.NET Core 6.0 [12-15];
- технологія Entity Framework Core [16, 17];
- фреймворк Angular [18-20];
- фреймворк Bootstrap [21, 22].

## 2.1. Технологія ASP.NET Core

ASP.NET Core 6.0 [12-15] є вебфреймворком з відкритим вихідним кодом, розробленим компанією Microsoft для створення вебзастосунків, сервісів та API. Він є оновленням популярного фреймворку ASP.NET і надає уніфіковану модель програмування для створення кросплатформних вебзастосунків з високою продуктивністю. Головною мовою програмування в ASP.NET Core 6.0 є C#, але також підтримуються F# та Visual Basic. Фреймворк базується на платформі .NET, яка надає широкий набір бібліотек і інструментів для розробки вебзастосунків. ASP.NET Core 6.0 є модульним, гнучким та масштабованим, і має безліч функцій, що роблять його ідеальним вибором для створення сучасних вебзастосунків.

Основні переваги ASP.NET Core 6.0:

- Кросплатформність: можливість створення застосунків, що працюють на різних операційних системах без змін коду.
- Висока продуктивність: оптимізована для обробки великої кількості паралельних запитів та включає функції кешування відповідей.
- Модульна архітектура: легке додавання та видалення компонентів для збереження легкості програми та спрощення обслуговування та масштабування.

- Відкритий вихідний код: доступ до вихідного коду та можливість сприяти його розвитку.
- Інтеграція з іншими технологіями: легка інтеграція з Microsoft Azure та підтримка широкого спектру фронтенд-технологій.
- Безпека: вбудовані функції безпеки, включаючи аутентифікацію, авторизацію, захист даних та підтримку HTTPS, що допомагають захистити застосунок та дані від атак

ASP.NET Core 6.0 є потужним і актуальним вебфреймворком, який має багато переваг у порівнянні з попередніми версіями. З його кросплатформністю, високою продуктивністю, модульною архітектурою, відкритим вихідним кодом, інтеграцією з іншими технологіями та функціями безпеки, він забезпечує зручні умови для розробки сучасних вебзастосунків та сервісів. Запуск нового стрімінгового сервісу в Україні на основі ASP.NET Core 6.0 може бути актуальним та перспективним кроком, забезпечуючи розробникам потужні інструменти та можливості для створення інноваційних та конкурентоздатних вебзастосунків.

## 2.2. Технологія Entity Framework Core

Технологія Entity Framework Core (EF Core) [16, 17] є ORM (об'єктно-реляційне відображення), яке дозволяє .NET розробникам взаємодіяти з реляційними базами даних за допомогою коду на C#. EF Core розроблено з метою бути легким, розширюваним та кросплатформним. Вона підтримує різноманітні постачальники баз даних, включаючи Microsoft SQL Server, SQLite, MySQL, PostgreSQL та Oracle. Розробники можуть використовувати EF Core для створення моделей баз даних, використовуючи код C# або генеруючи їх з наявних баз даних. EF Core забезпечує можливість створення, читання, оновлення та видалення записів у базі даних, а також виконання складних операцій з даними.

Однією з основних переваг EF Core є те, що вона сприяє написанню

чистого та легкого для обслуговування коду. Завдяки абстрагуванню від деталей роботи з базами даних, розробники можуть сконцентруватися на написанні коду, пов'язаного з бізнес-логікою та функціональністю застосунків. Це дозволяє створювати менш помилковий та більш підтримуваний код з плинним часом [16].

Ще однією перевагою EF Core є можливість роботи з базами даних за допомогою об'єктно-орієнтованих методів програмування. Розробники можуть створювати класи, які відображаються на таблиці бази даних, і використовувати запити LINQ для отримання даних з бази даних. Це спрощує роботу з даними у більш природному та інтуїтивно зрозумілому способі, необхідному для складання складних SQL-запитів [17].

Підтримка широкого спектру постачальників баз даних, кросплатформність та підтримка сучасних практик розробки програмного забезпечення роблять технологію Entity Framework Core популярним вибором для створення надійних та масштабованих застосунків, що працюють з базами даних. EF Core допомагає розробникам працювати з базами даних ефективно, надаючи зручні інструменти та спрощуючи взаємодію з даними. Завдяки його функціональності та гнучкості, EF Core стає потужним інструментом для розробки сучасних застосунків, які використовують бази даних. Структура роботи технології EF Core зображена на рис. 5.

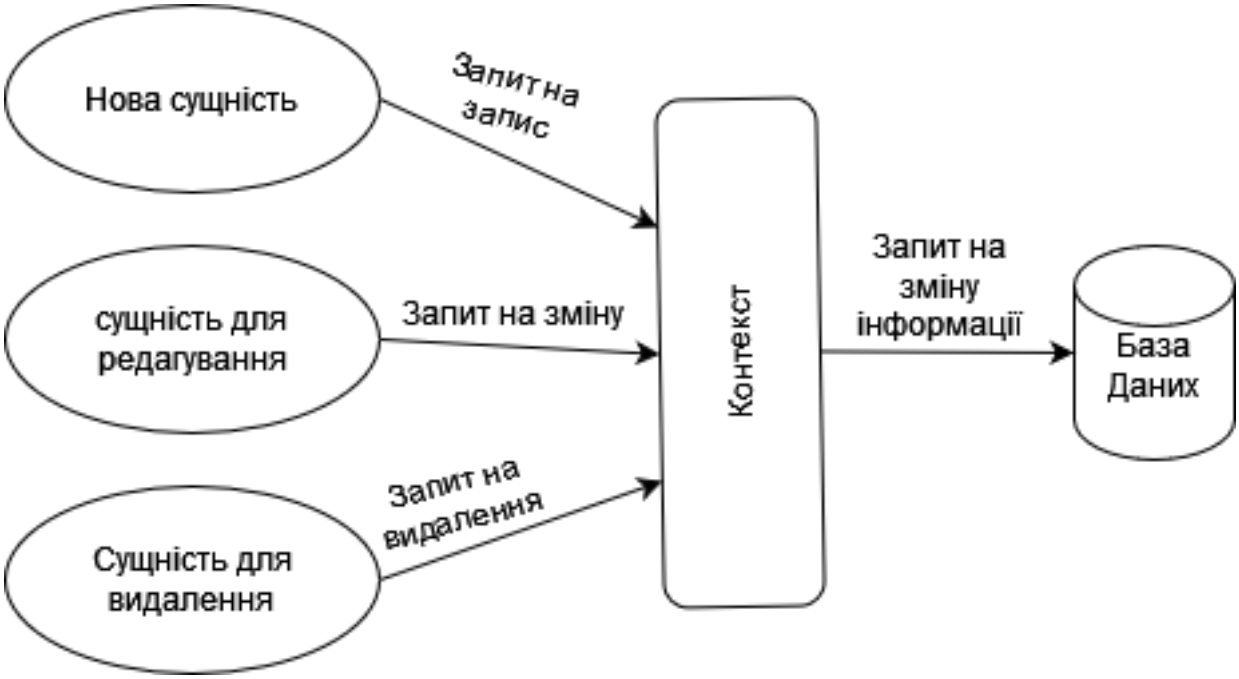



**Рисунок 5 –** Структура роботи технології EF Core

## 2.3. Фреймворк Angular

Angular [18-20] – це відомий фреймворк з відкритим вихідним кодом, розроблений та підтримуваний компанією Google, для створення вебзастосунків. Цей фреймворк має широкі можливості і призначений для розробки динамічних односторінкових вебзастосунків.

Angular надає розробникам засоби для створення складних застосунків, що працюють зі значними обсягами даних. Він включає компоненти, шаблони, сервіси, маршрутизацію, форми та ін'єкцію залежностей. Архітектура фреймворку базується на моделі Model-View-Controller (MVC), що дозволяє розділити застосунок на три окремі шари: модель (дані), представлення (інтерфейс) та контролер (логіка).

Однією з ключових особливостей Angular є використання директив, які є спеціальними атрибутами HTML. Вони дозволяють розробникам прикріпляти певну поведінку до елементів інтерфейсу користувача. Angular також надає набір вбудованих директив, таких як ngIf, ngFor та ngStyle [18].

Фреймворк Angular приділяє значну увагу продуктивності та оптимізації

за допомогою функцій, таких як компіляція Ahead-of-Time (AOT) та ліниве завантаження. Компіляція Ahead-of-Time (AOT) перетворює код Angular-застосунку з TypeScript і HTML в ефективний JavaScript-код під час процесу збірки перед розгортанням застосунку в браузері. AOT має кілька переваг, зокрема поліпшену продуктивність і зменшений розмір застосунку, а також здатність виявляти помилки шаблонів на ранній стадії [19].

Ліниве завантаження в Angular дозволяє завантажувати модулі та компоненти лише тоді, коли вони є необхідними, що допомагає зменшити час початкового завантаження застосунку.

Крім основного фреймворку, Angular має багатий екосистему інструментів та бібліотек. Наприклад, Angular Material надає набір готових компонентів інтерфейсу користувача, що спрощують розробку веб застосунків. Також варто відзначити RxJS, який є потужним інструментом для роботи з асинхронними потоками даних [20].

В цілому, Angular є потужним та всеосяжним фреймворком для створення складних вебзастосунку, які працюють з великими обсягами даних. Він має сильний фокус на продуктивності та оптимізації, а також надає розробникам широкий набір інструментів та бібліотек для полегшення процесу розробки. На рис. 6 можна наведено приблизну структуру Angular застосунку.

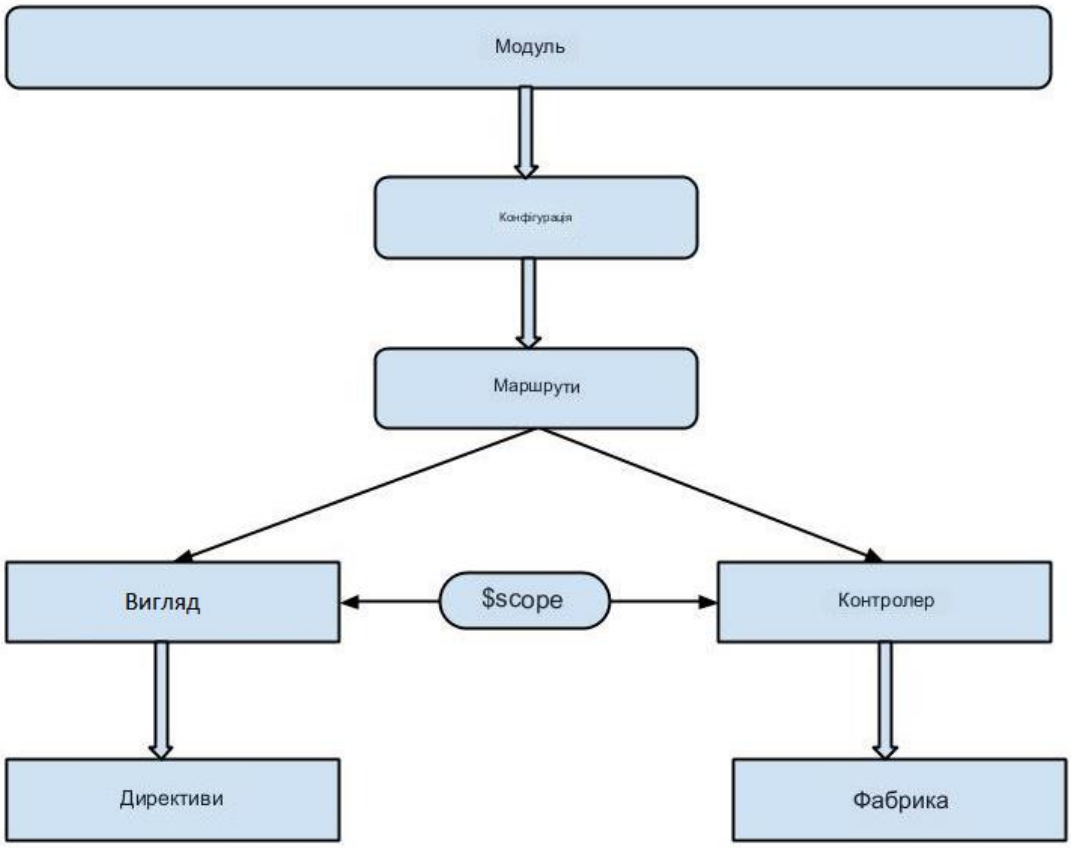



**Рисунок 6** – Схематична архітектура Angular

## 2.4. Фреймворк Bootstrap

Bootstrap [21, 22] є фреймворком для фронтенду з відкритим вихідним кодом, розробленим компанією Twitter та підтримуваним спільнотою розробників. Основна мета Bootstrap полягає в спрощенні створення адаптивних та мобільних веб-сторінок. Він надає набір готових компонентів HTML, CSS та JavaScript, які можна використовувати для швидкого створення загальних елементів користувацького інтерфейсу, таких як навігаційні меню, форми, кнопки та модальні вікна. Ці компоненти розроблені з урахуванням їх взаємодії, що сприяє послідовному та цілісному вигляду користувацького інтерфейсу.

Bootstrap також включає адаптивну систему сітки, яка дозволяє створювати макети, що пристосовуються до різних розмірів екранів та пристроїв. Це особливо корисно для створення застосунків, які повинні добре працювати на

різних пристроях, включаючи як десктопні, так і мобільні пристрої [22].

Однією з ключових переваг використання Bootstrap є можливість економити час та зусилля при створенні вебзастосунків. Замість того, щоб розробляти елементи користувацького інтерфейсу з нуля, цей фреймворк надає готові компоненти та стилі. Це дозволяє розробникам зосередитися на функціональності та змісті застосунку, не витрачаючи багато часу на створення користувацького інтерфейсу.

# РОЗДІЛ 3.
# ПРИЗНАЧЕННЯ ТА ЦІЛІ СТВОРЕННЯ ВЕБЗАСТОСУНКУ

## 3.1. Призначення застосунку

Призначенням застосунку є забезпечення широкого доступу до різноманітного відеоконтенту. Основне завдання цього сервісу полягає в наданні користувачам можливості переглядати фільми та серіали. Він створений, щоб задовольнити різноманітні смаки та інтереси користувачів, дозволяючи їм насолоджуватися розважальним контентом за їхніми власними перевагами. Ви можете побачити Use-case діаграму в додатку А.

Кваліфікаційна робота передбачає:

- вивчення та дослідження різних методів, технологій, моделей та підходів; які використовуються для розробки вебзастосунів;
- дослідження та аналіз процесу створення робочих програм для подальшої автоматизації;
- аналіз наявних програмних засобів;
- проєктування та програмну реалізацію вебзастосунку.

Вебзастосунок створено з метою:

- полегшити користувачам доступ до відеоконтенту;
- Автоматизувати пошук фільмів по жанрам;
- Автоматизувати персоналізований підбір;
- Покращити взаємодію з контентом.

## 3.2. Вимоги до вебзастосунку

Застосунок повинен мати два типи користувачів: звичайні відвідувачі та адміністратори.

Ось кілька загальних вимог, які можуть бути враховані при створенні

успішного стрімінгового сервісу:

- Платформа та інтерфейс: розробка зручної та інтуїтивно зрозумілої платформи, яка дозволяє користувачам легко шукати, вибирати та переглядати фільми. Інтерфейс повинен бути зрозумілим та дружнім до користувача.
- Каталог фільмів: забезпечення широкого вибору фільмів із різних жанрів та країн. Каталог повинен бути оновлюваним та містити нові релізи, класичні фільми та різноманітний контент для різних аудиторій.
- Якість відео та аудіо: забезпечення високоякісного відео та аудіо для перегляду фільмів. Платформа повинна підтримувати відтворення високої роздільної здатності, наприклад, Full HD або навіть 4K, а також забезпечувати чистий іммерсивний звук.
- Адаптивність: можливість адаптувати якість відео залежно від швидкості Інтернет-з'єднання користувача. Надання можливості вибору роздільної здатності або автоматичного регулювання якості, щоб забезпечити безперебійний стрімінг.
- Контроль доступу та захист: забезпечення механізмів контролю доступу до вмісту, включаючи автентифікацію користувачів, систему оплати та захист від незаконного копіювання або поширення контенту.

В застосунку наявні дві підсистеми. Підсистеми адміністратора та користувача організовані наступним чином:

Підсистема адміністратора:

- Керування контентом: адміністратор має доступ до панелі керування, де він може додавати, редагувати та видаляти фільми з каталогу. Він також може завантажувати афіші, постери та іншу інформацію про фільми.
- Управління користувачами: адміністратор може керувати користувачами, включаючи реєстрацію нових користувачів, зміну їх профілів, блокування акаунтів та вирішення будь-яких питань щодо облікових записів.
- Підсистема користувача:

- Реєстрація та авторизація: користувач може створити обліковий запис на платформі, заповнивши необхідну інформацію і підтвердивши свою електронну адресу. Після цього він може авторизуватися на платформі, використовуючи свої облікові дані.
- Пошук та перегляд фільмів: користувач може використовувати пошукову функцію, щоб знайти фільми за різними критеріями, такими як жанр, рік виходу, актори тощо. Він може переглядати детальну інформацію про фільм, включаючи опис, трейлер, рейтинги та відгуки інших користувачів.
- Оцінка та відгуки: користувач може залишати відгуки та ставити оцінки фільмам, які він переглянув. Це дозволяє іншим користувачам отримувати рекомендації та впливати на загальний рейтинг фільмів.
- Особистий кабінет: користувач має доступ до особистого кабінету, де він може переглядати свою історію перегляду, управляти підписками або відтворенням списків, змінювати налаштування профілю та отримувати персоналізовані рекомендації.

### 3.3. Технічні вимоги до проєкту

У сучасному світі все більше людей користуються Інтернетом для перегляду фільмів та використовують різноманітні браузери. З цим пов'язана проблема несумісності вебсайтів, оскільки кожен браузер має свої підходи до відображення веб-сторінок.

Онлайн кінотеатр повинен задовольняти наступні технічні вимоги:

- Підтримка браузерів: розробка застосунку повинна забезпечувати сумісність з найбільш популярними браузерами, такими як Google Chrome, Microsoft Edge, Mozilla Firefoxта Opera. Варто тестувати та перевіряти відображення та функціональність проєкту в цих браузерах, щоб забезпечити їхню коректну роботу.
- Адаптивний дизайн: вебсайт онлайн кінотеатру повинен мати адаптивний дизайн, що дозволить користувачам зручно переглядати фільми на різних

пристроях. Дизайн повинен коректно адаптуватись до розмірів екрану, забезпечуючи зручний інтерфейс незалежно від використовуваного пристрою.

- Висока продуктивність: розробка застосунку повинна враховувати оптимізацію швидкодії та завантаження сторінок. Важливо, щоб веб-сторінки завантажувалися швидко, забезпечуючи ефективну навігацію та перегляд фільмів без затримок.
- Підтримка стрімінгу відео: Проєкт повинен мати можливість стрімінгу відео в режимі реального часу, щоб користувачі могли комфортно дивитися фільми без необхідності завантаження їх повністю перед переглядом.
- Інтеграція з СКБД: для зберігання та керування даними про фільми, користувачів та іншої інформації, застосунок повинен використовувати бази даних. Використання Microsoft SQL Server може забезпечити надійну та масштабовану базу даних для онлайн кінотеатру.
- Обробка помилок та повідомлення про помилки: Онлайн кінотеатр повинен мати механізми обробки помилок, які дозволяють користувачам отримувати зрозумілі повідомлення про помилки та поради щодо виправлення неполадок. Це сприяє поліпшенню користувацького досвіду та допомагає уникнути некоректної роботи застосунку.

Загальною метою цих технічних вимог є створення стабільного, безпечного та легко використовуваного онлайн кінотеатру, яка забезпечує зручний перегляд фільмів користувачами з різних пристроїв та браузерів.

### 3.4. Вимоги законодавства України щодо стрімінгового сервісу

Вимоги законодавства України щодо стрімінгового сервісу включають декілька ключових аспектів, які потрібно враховувати при розробці та експлуатації такого сервісу. Нижче наведені деякі з найважливіших законодавчих вимог:

Закон України "Про захист персональних даних": забезпечення

конфіденційності та захисту персональних даних користувачів є обов'язковим. Сервіс повинен дотримуватись вимог щодо збору, зберігання та обробки персональних даних, а також повинен мати політику конфіденційності, яка розкриває, які дані збираються та як вони використовуються [23].

Закон України "Про авторське право та суміжні права": стрімінговий сервіс повинен дотримуватись авторських прав і не порушувати права власників авторських та суміжних прав на контент, що відтворюється або передається через цей сервіс [24].

Закон України "Про телебачення і радіомовлення": якщо стрімінговий сервіс надає доступ до відео- або аудіоматеріалів, які відповідають ознакам телебачення або радіомовлення, можуть застосовуватись додаткові вимоги, включаючи ліцензування та регулювання контенту [25].

Закон України "Про захист прав споживачів": сервіс повинен забезпечувати права споживачів, включаючи доступну та точну інформацію про послуги, право на якісний контент, можливість відмови від послуги тощо [26].

Для дотримання вимог законодавства, пов'язаного з обробкою персональних даних та авторськими правамив вреалізованому в рамках кваліфікаційної роботи сервісі при реєстрації, користувачам надається можливість погодитися з умовами обробки їх персональних даних шляхом встановлення прапорця згоди. Також, в сервісі реалізовано чекбокс, який підтверджує, що користувач має права на розповсюдження фільму, якщо він додає контент до сервісу. Враховуючи ці механізми, в застосунку забезпечено дотримання вимог законодавства.

# РОЗДІЛ 4.
# РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

## 4.1. Вимоги до структури бази даних

Для розробки баз даних було використано інструмент Microsoft SQL Management Studio. База даних використовується в онлайн кінотеатрі для зберігання та керування різноманітною інформацією, необхідною для її правильного функціонування. База даних дозволяє зберігати дані про фільми, користувачів, ролі, оцінки, коментарі, категорії та перегляди, які є основними компонентами. Це надає можливість ефективного управління фільмами, управління користувачами, надання рекомендацій, аналізу даних та забезпечення гладкої та зручної роботи кінотеатру. Далі у табл. 1 наведено перелік головних таблиць бази даних, діаграму якої можна побачити на рис. 8.

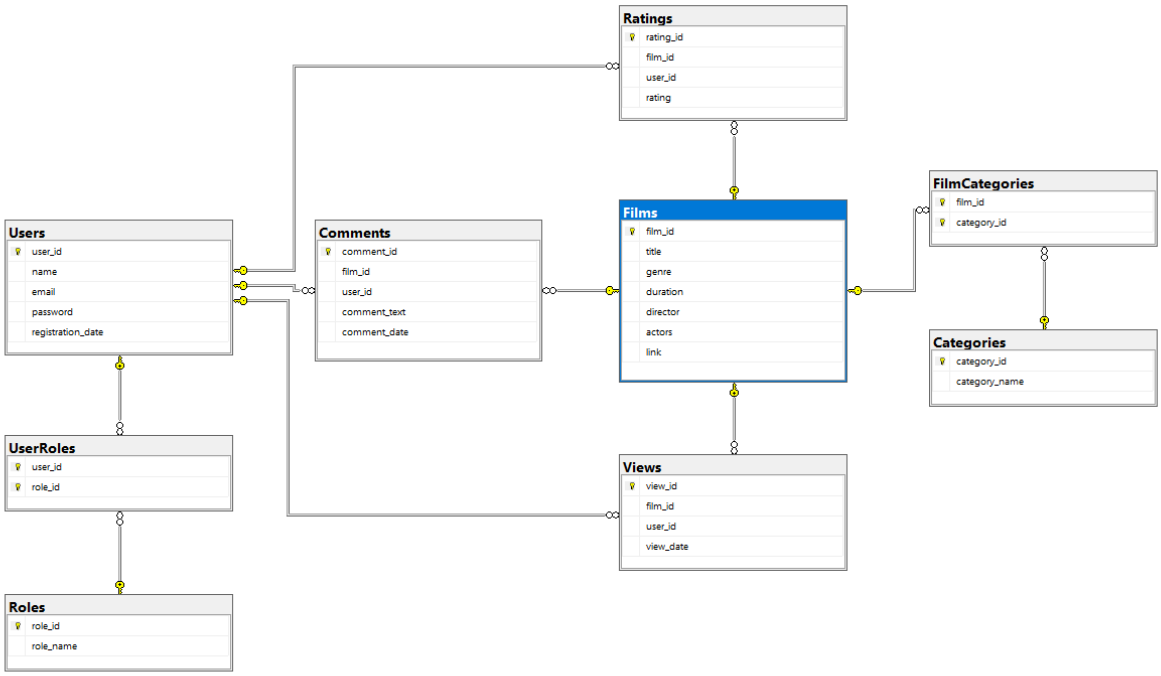



**Рисунок 8** – Діаграма БД

**Таблиця 1** – Короткий опис таблиць БД

| Назва таблиці | Опис |
|---|---|
| Фільми (Movies) | Зберігає інформацію про фільми та посилання на них. |
| Користувачі (Users) | Містить дані про користувачів. |
| Ролі (Roles) | Використовується для зберігання ролей користувачів. |
| Оцінки (Ratings) | Зберігає дані про оцінки фільмів, що ставлять користувачі |
| Коментарі (Comments) | Містить коментарі, що залишаються користувачами під фільмами. |
| Категорії (Categories) | Зберігає інформацію про категорії фільмів. |
| Перегляди (Views) | Використовується для відстеження переглядів фільмів користувачами. |

Наведемо опис таблиць, які були згадані в табл.1:

Таблиця " Films" (див. табл. 2) містить інформацію про фільми, доступні в онлайн кінотеатрі. Вона може включати такі поля, як назва фільму, жанр, тривалість, режисер, актори та посилання на фільм.

**Таблиця 2** – Структура таблиці Фільми (Movies)

| Атрибут | Тип | Опис |
|---|---|---|
| film_id | INT, PRIMARY KEY | Ідентифікатор фільму |
| title | VARCHAR | Назва фільму |
| genre | VARCHAR | Жанр фільму |
| duration | TIME | Тривалість фільму |
| director | VARCHAR | Режисер фільму |
| actors | VARCHAR | Актори фільму |
| link | VARCHAR | Посилання на фільм |

Таблиця " Users " (див. табл. 3) містить дані про користувачів онлайн

кінотеатру. Вона може включати поля, такі як ім'я, прізвище, електронна пошта, пароль, дата реєстрації та інші відомості, необхідні для ідентифікації та керування користувачами.

**Таблиця 3** – Структура таблиці Користувачі (Users)

| Атрибут | Тип | Опис |
|---|---|---|
| user_id | INT, PRIMARY KEY | Ідентифікатор користувача |
| name | VARCHAR | Ім'я користувача |
| email | VARCHAR | Електронна пошта користувача |
| password | VARCHAR | Пароль користувача |
| registration_date | DATE | Дата реєстрації користувача |

Таблиця "Roles " (див. табл. 4) містить дані про ролі користувачів. Вона може містити різні ролі, такі як адміністратор, модератор, звичайний користувач тощо, і дозволяє керувати доступом та привілегіями користувачів.

**Таблиця 4** – Структура таблиці Ролі (Roles)

| Атрибут | Тип | Опис |
|---|---|---|
| role_id | INT, PRIMARY KEY | Ідентифікатор ролі |
| role_name | VARCHAR | Назва ролі |

Таблиця "Ratings " (див. табл. 5) містить дані про оцінки, які користувачі ставлять фільмам. Вона може включати поля, такі як ідентифікатор фільму, ідентифікатор користувача та оцінка, яку користувач виставив фільму.

**Таблиця 5** – Структура таблиці Оцінки (Ratings)

| Атрибут | Тип | Опис |
|---|---|---|
| rating_id | INT, PRIMARY KEY | Ідентифікатор оцінки |
| film_id | INT, FOREIGN KEY | Ідентифікатор фільму, пов'язаний з таблицею Films |

| | | |
|---|---|---|
| user_id | INT, FOREIGN KEY | Ідентифікатор користувача, пов'язаний з таблицею Users |
| rating | DECIMAL | Оцінка фільму |

Таблиця "Comments " (див. табл. 6) містить коментарі, які користувачі залишають під фільмами. Вона може містити поля, такі як ідентифікатор фільму, ідентифікатор користувача, текст коментаря та дата його залишення.

**Таблиця 6** – Структура таблиці Коментарі (Comments)

| Атрибут | Тип | Опис |
|---|---|---|
| comment_id | INT, PRIMARY KEY | Ідентифікатор коментаря |
| film_id | INT, FOREIGN KEY | Ідентифікатор фільму, пов'язаний з таблицею Films |
| user_id | INT, FOREIGN KEY | Ідентифікатор користувача, пов'язаний з таблицею Users |
| comment_text | VARCHAR | Текст коментаря |
| comment_date | DATETIME | Дата коментаря |

Таблиця "Categories " (див. табл. 7) містить дані про категорії фільмів, які використовуються для класифікації та організації фільмів. Вона може містити поля, такі як назва категорії та її опис.

**Таблиця 7** – Структура таблиці Категорії (Categories)

| Атрибут | Тип | Опис |
|---|---|---|
| category_id | INT, PRIMARY KEY | Ідентифікатор категорії |
| category_name | VARCHAR | Назва категорії |

Таблиця "Views " (див. табл. 8) відстежує інформацію про перегляди фільмів користувачами. Вона може включати поля, такі як ідентифікатор фільму, ідентифікатор користувача та дата перегляду.

**Таблиця 8** – Структура таблиці Перегляди (Views)

| Атрибут | Тип | Опис |
|---|---|---|
| view_id | INT, PRIMARY KEY | Ідентифікатор перегляду |
| film_id | INT, FOREIGN KEY | Ідентифікатор фільму, пов'язаний з таблицею Films |
| user_id | INT, FOREIGN KEY | Ідентифікатор користувача, пов'язаний з таблицею Users |
| view_date | DATETIME | Дата перегляду |

## 4.2. Програмна реалізація системи

Для реалізації вебзастосунку використано ASP.NET Core 6.0 [12-15]. Для роботи з даними в проєкті використано технологію Entity Framework Core [16, 17]. Моделі даних, що відповідають наведеним таблицям, можна знайти у Додатку Б. Кожна модель відображає відповідну таблицю з бази даних і містить необхідні властивості та зв'язки з іншими моделями. Схема моделі даних проєкту містить наступні класи: Film, User, Role, Rating, Comment, Category, View та клас контексту FilmContext.

Під час написання фронтенду на Angular [18-20], було використано модульну структуру для організації компонентів, сервісів і маршрутизації.Процес написання фронтенду включав створення компонентів для різних сторінок та функціональності, таких як реєстрація, авторизація, пошук фільмів і коментування. Компоненти були розроблені з урахуванням принципів реактивного програмування і забезпечували взаємодію з сервером через API. Приклад вихідного коду однієї зі сторінок, а саме сторінки реєстрації наведено в додатку В.

# РОЗДІЛ 5.

# ІНСТРУКЦІЯ КОРИСТУВАЧА

## 5.1 Інструкція для реєстрації

Щоб зареєструватися користувачу необхідно зробити наступні кроки:

- Відкрийте домашню сторінку нашого сайту.
- Знайдіть кнопку "Реєстрація" та натисніть на неї.
- Відкриється форма реєстрації, де потрібно ввести свої особисті дані.
- Надайте своє ім'я та прізвище відповідними полями.
- Вкажіть свою електронну адресу, яка буде використовуватися для входу на сайт.
- Виберіть надійний пароль і введіть його у відповідному полі. Пароль повинен складатися з мінімум 8 символів та містити комбінацію цифр, літер верхнього та нижнього регістру, а також спеціальних символів.
- При необхідності, підтвердіть пароль, введений у попередньому кроці.
- Натисніть на чекбокс, щоб підтвердити свою згоду на обробку персональних даних.
- Перевірте всі введені дані ще раз, щоб переконатися в їх правильності.
- Натисніть кнопку "Зареєструватися" або "Створити обліковий запис", для завершення процесу реєстрації.
- Після успішної реєстрації ви отримаєте підтвердження на свою електронну адресу.
- Зараз ви можете увійти на сайт, використовуючи свою електронну адресу та пароль, які ви вказали під час реєстрації.
- Після входу на сайт ви матимете доступ до різноманітного вмісту, включаючи фільми, категорії, коментарі та оцінки.

На рис. 8 ви можете побачити форму реєстрації.

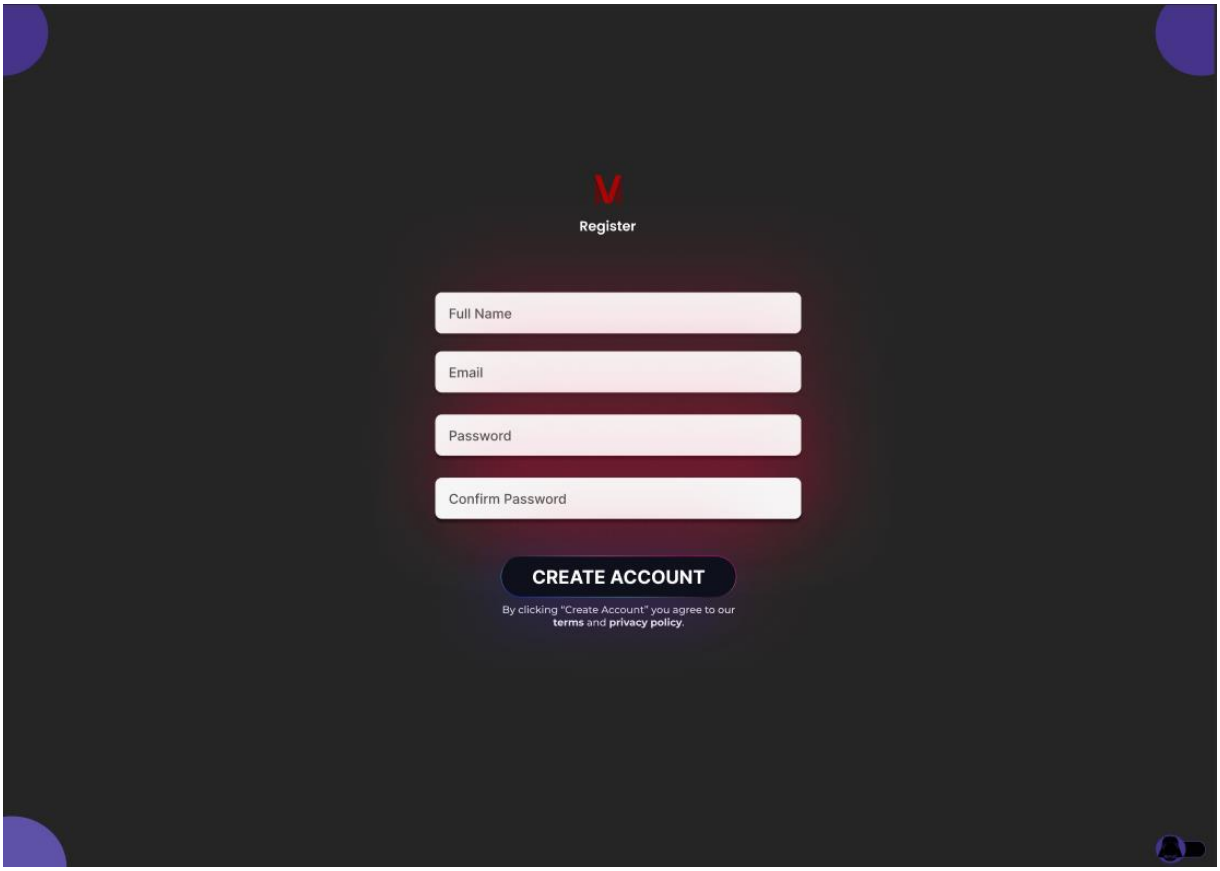



**Рисунок 8** – Форма реєстрації

Інструкція для авторизації:

Знайдіть кнопку "Увійти"або "Авторизацію". Натисніть на неї.

Введіть свою електронну адресу, яка використовувалася при реєстрації.

Введіть пароль.

Після введення облікових даних натисніть кнопку "Увійти".

Якщо введені дані правильні, вас буде успішно авторизовано і перенаправлено на головну сторінку сайту.

На рис.9 наведено вигляд форми для авторизації.

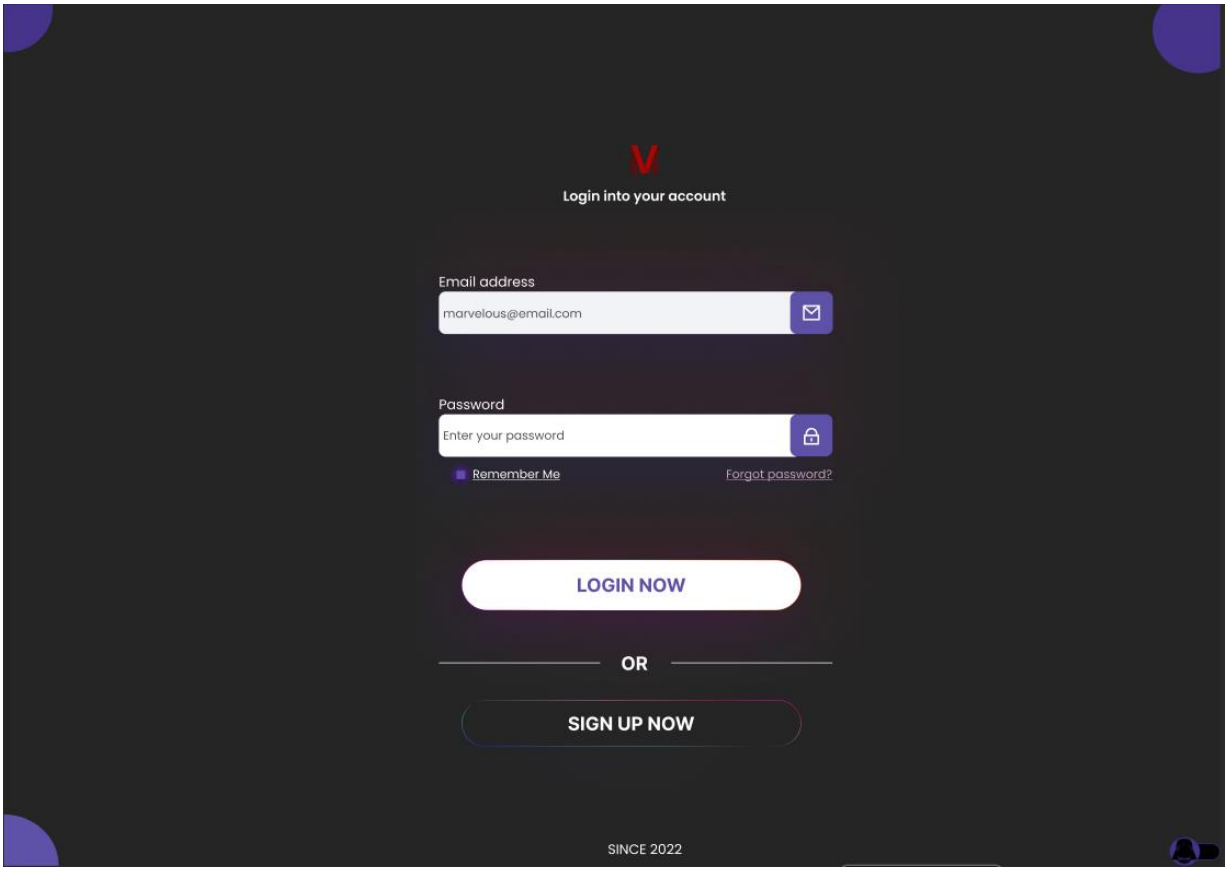



**Рисунок 9** – Форма реєстрації

На рис. 10 ви можете побачити головну сторінку веб-застосунку з якої ви можете переглянути новини кіноіндустрії, популярні фільми, та фільтр списку всіх доступних кінофільмів.

Інструкція по фільтрації фільмів за жанрам, роком та країною походження:

- Перегляньте доступні жанри фільмів, які представлені у вигляді кнопок на головній сторінці
- Щоб обрати жанр натисніть на відповідну кнопку, сайт дозволяє обрати лише один жанр.
- Щоб обрати рік, натисніть на відповідний випадаючий список, виберіть потрібний рік.
- Щоб обрати країну походження фільму також оберіть її з випадаючого списку.

Також на головній сторінці наявний значок пошуку натиснувши на який відкриється поле для введення назви.

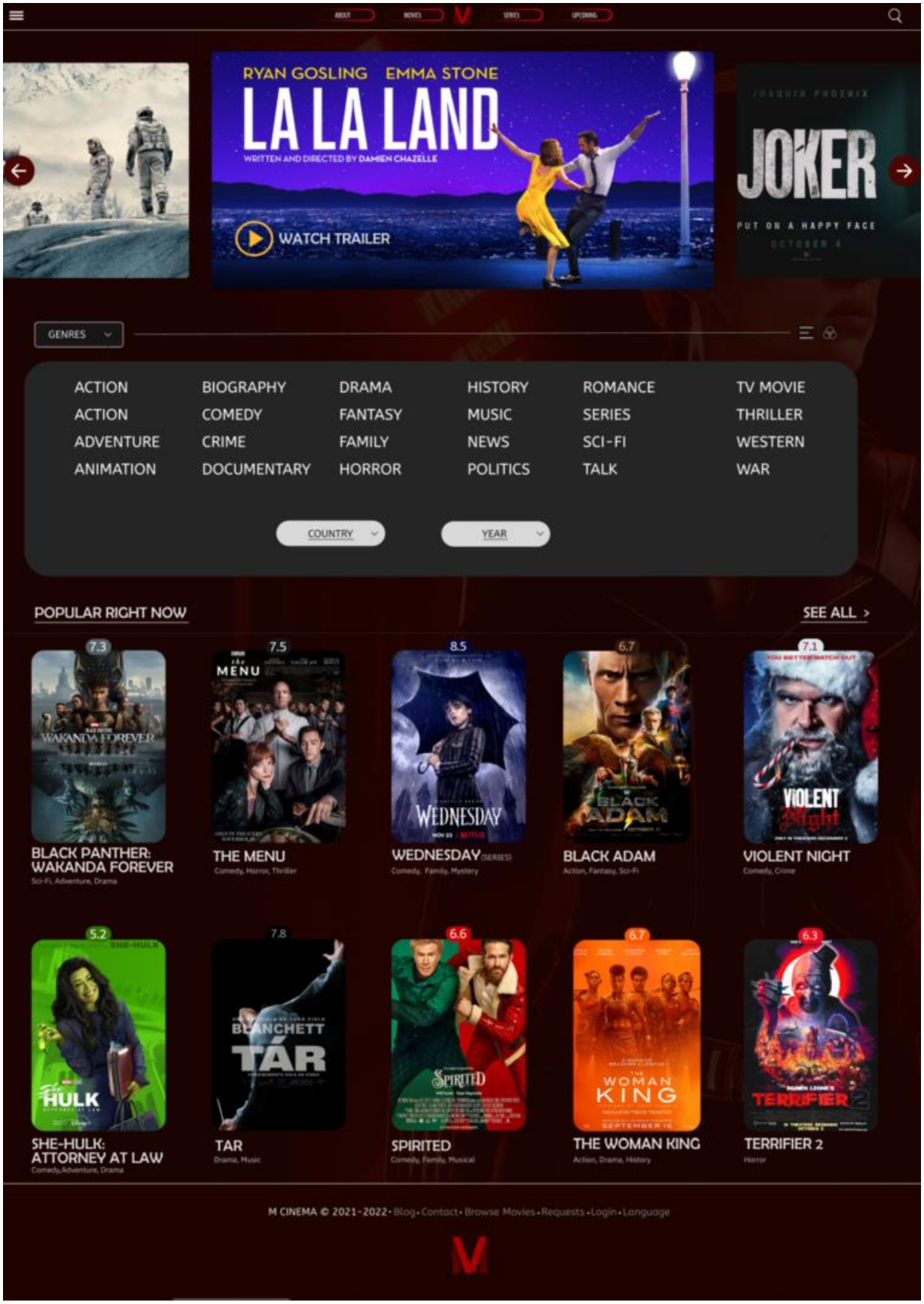



**Рисунок 10** – Головна сторінка

При обранні певного фільму або серіалу користувач попадає на відповідну сторінку, на рис. 11 наведено вигляд сторінки з фільмом, а на рис. 12 з серіалом.

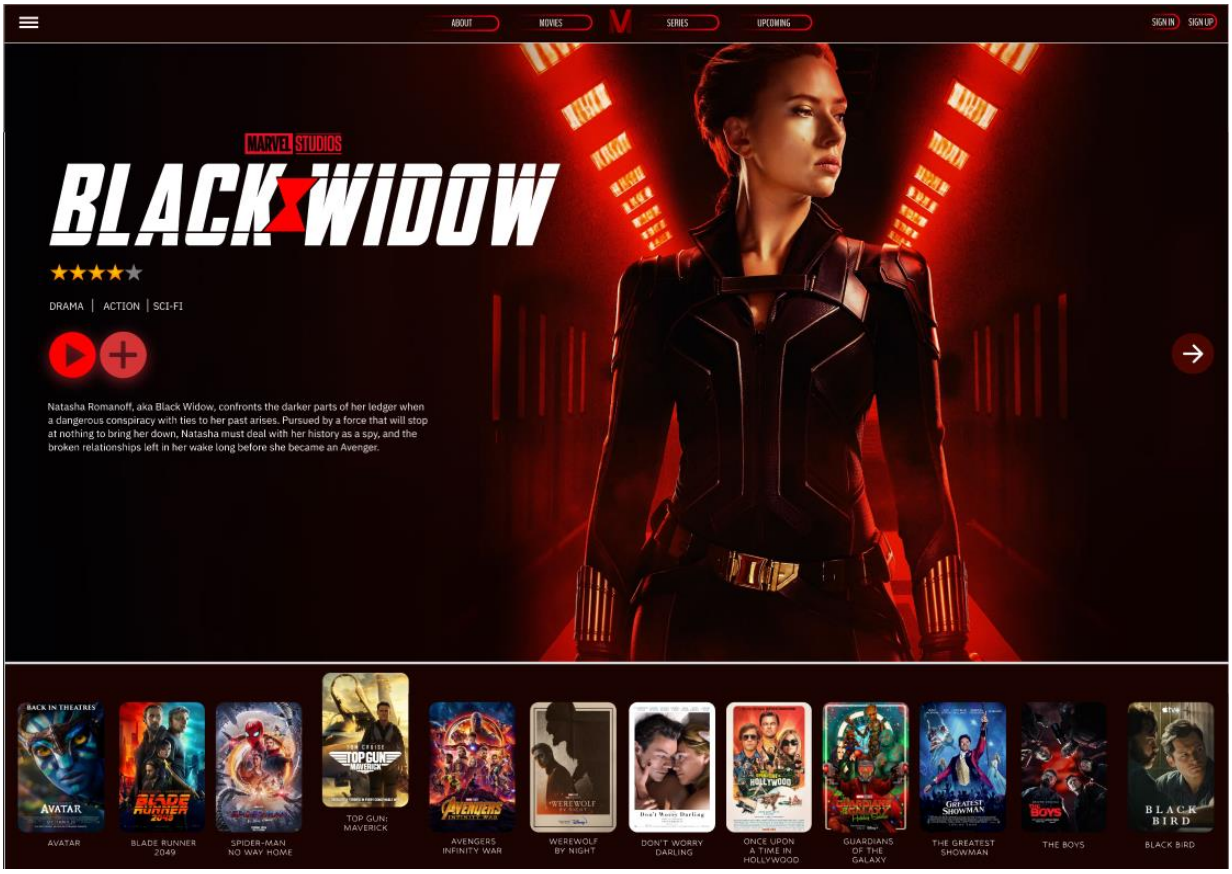



**Рисунок 11** – Сторінка перегляду фільму

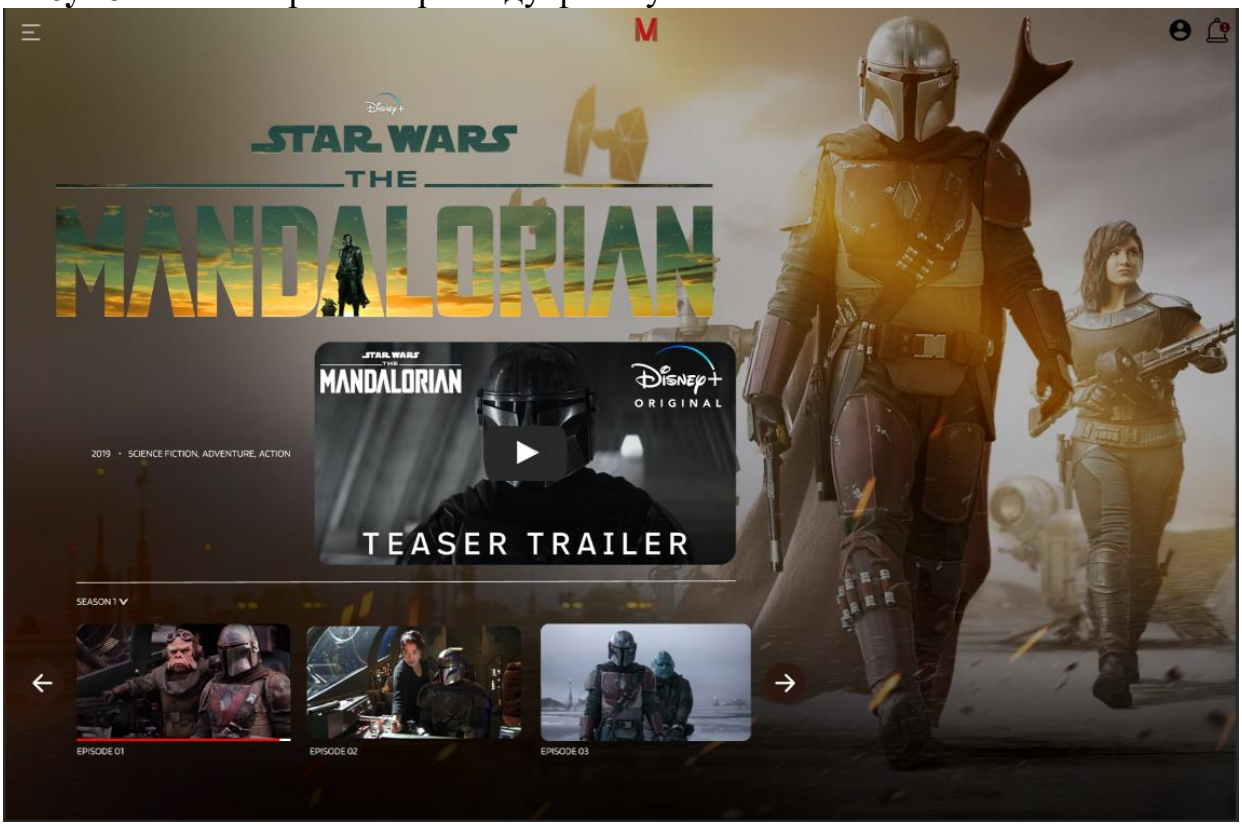



**Рисунок 12** – Сторінка перегляду серіалу

# ВИСНОВКИ

В кваліфікаційній роботі було розроблено стрімінговий сервіс для перегляду кінофільмів та серіалів в онлайн-режимі. Головна мета полягала в створенні функціонального, ефективного та зручного сервісу з інтуїтивно зрозумілим інтерфейсом для користувачів.

Для досягнення цієї мети було виконано наступні завдання:

- Проведено дослідження наявних стрімінгових сервісів в Україні.
- Наведено аналіз технологій розробки вебзастосунків.
- Розроблено структуру та архітектуру сервісу з використанням ASP.NET Core, що забезпечує модульну та розширювану структуру проєкту.
- Реалізовано функціональність пошуку, фільтрування та перегляду кінофільмів з можливістю перегляду трейлерів та отримання детальної інформації про кожен фільм.
- Розроблено реєстрацію та авторизацію користувачів зберігання їхніх вподобань, коментарів та оцінок фільмів.
- Забезпечено інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам знайти та переглянути вміст відповідно до їхніх вподобань та потреб.

Результатом роботи є функціональний та зручний стрімінговий сервіс, що надає користувачам можливість насолоджуватись переглядом кінофільмів та серіалів у зручний час та в зручному форматі.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ


1. Netflix [Електронний ресурс] - Джерело інформації: https://www.netflix.com/ua/
2. Apple TV+ [Електронний ресурс] - Джерело інформації: https://tv.apple.com/ua
3. Amazon Prime Video [Електронний ресурс] - Джерело інформації: https://www.primevideo.com/offers/nonprimehomepage/ref=dvm_pds_amz_UA_lb_s_g_mkw_sY09BtrhT-dc_pcrid_507190918478?mrntrk=slid__pgrid_123397680870_pgeo_9061017_x__adext__ptid_kwd-2850171361
4. Google Play Movies [Електронний ресурс] - Джерело інформації: https://play.google.com/store/movies/
5. Megogo [Електронний ресурс] - Джерело інформації: https://megogo.net/ua
6. 10 найпопулярніших стрімінгових сервісів [Електронний ресурс] - Джерело інформації: https://dev.ua/news/valid-article-1661772935-1661772935
7. COVID-19 accelerates streaming challenges as well as demand [Електронний ресурс] - Джерело інформації: https://info.kpmg.us/news-perspectives/technology-innovation/covid-19-accelerates-streaming-challenges-and-demand.html
8. Visual Studio Community 2022 [Електронний ресурс] - Програмне забезпечення: https://visualstudio.microsoft.com/vs/
9. Download SQL Server Management Studio [Електронний ресурс] - Програмне забезпечення: https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16
10. Figma [Електронний ресурс] - Програмне забезпечення: https://www.figma.com
11. Sweet.tv [Електронний ресурс] - Джерело інформації: https://sweet.tv/

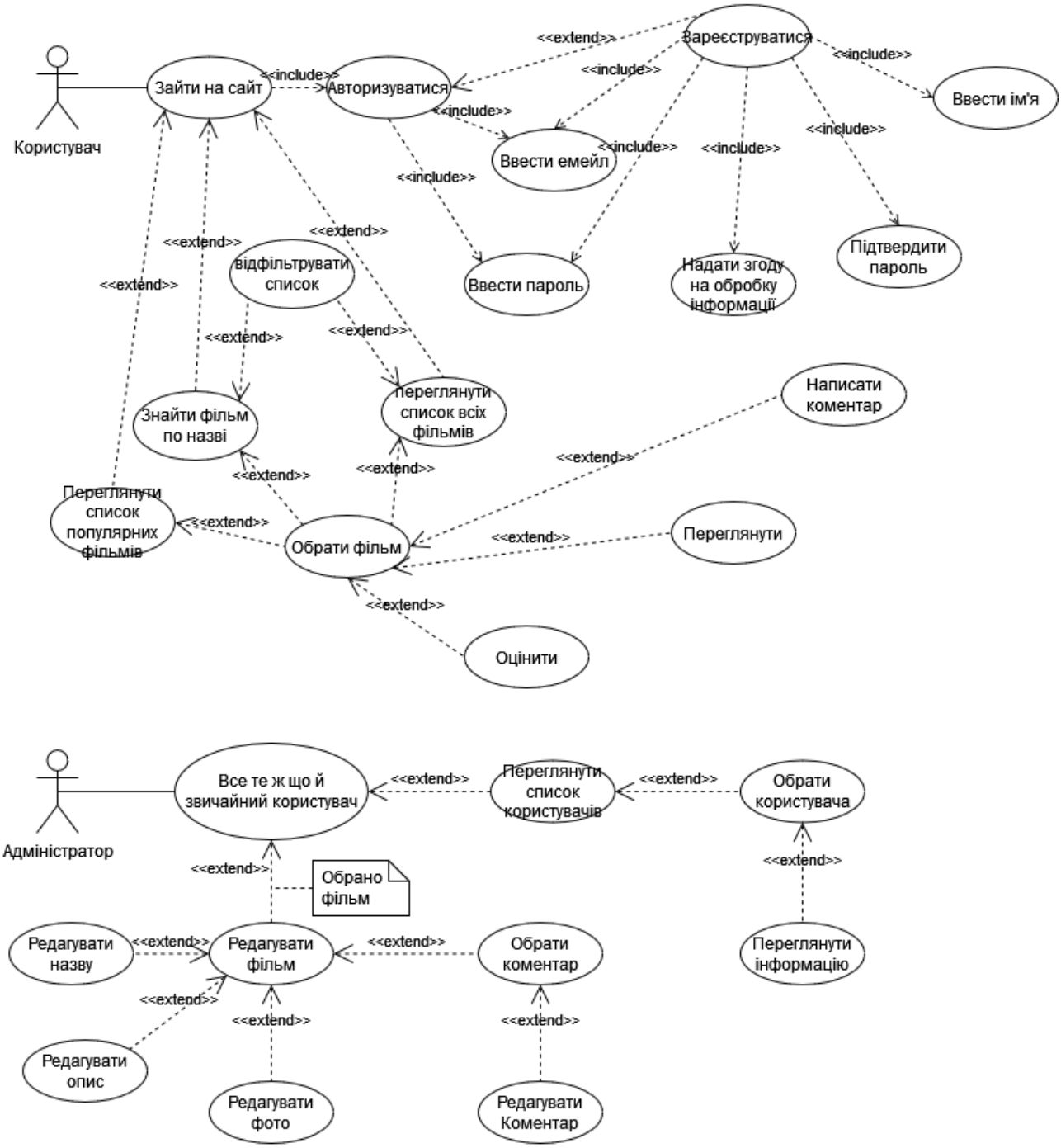
12. Overview of ASP.NET Core [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-6.0.
13. ASP.NET documentation [Електронний ресурс] – Режим доступу до ресурсу: https://docs.microsoft.com/en-us/aspnet/core/.
14. Code Maze - ASP.NET Core [Електронний ресурс] – Режим доступу до ресурсу: https://code-maze.com/net-core-series/.
15. What's new in ASP.NET Core 6.0 [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://learn.microsoft.com/en-us/aspnet/core/release-notes/aspnetcore-6.0?view=aspnetcore-6.0.
16. Entity Framework Core [Електронний ресурс] – Режим доступу до ресурсу: https://docs.microsoft.com/en-us/ef/core/.
17. Entity Framework Core [Електронний ресурс] – Режим доступу до ресурсу: https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx.
18. Introduction to Angular concepts [Електронний ресурс] – Режим доступу до ресурсу https://angular.io/guide/architecture.
19. Using Angular in Visual Studio Code [Електронний ресурс] – Режим доступу до ресурсу https://code.visualstudio.com/docs/nodejs/angular-tutorial.
20. Code Maze - Angular [Електронний ресурс] – Режим доступу до ресурсу: https://code-maze.com/angular-series/.
21. Build fast, responsive sites with Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: https://getbootstrap.com/.
22. Bootstrap documentation [Електронний ресурс] – Режим доступу до ресурсу: https://getbootstrap.com/docs/4.1/getting-started/introduction/.
23. Закон України "Про захист персональних даних" - Правовий комплекс: https://zakon.rada.gov.ua/laws/show/2297-17#Text
24. Закон України "Про авторське право та суміжні права" - Правовий комплекс: https://zakon.rada.gov.ua/laws/show/3792-12#Text

25. Закон України "Про телебачення і радіомовлення" - Правовий комплекс: https://zakon.rada.gov.ua/laws/show/3759-12#Text

26. Закон України "Про захист прав споживачів" - Правовий комплекс: https://zakon.rada.gov.ua/laws/show/1023-12#Text

# ДОДАТКИ

## Додаток А. Use-case діаграма



## Додаток Б. Моделі проєкту

```
public class Film
{
    public int Id { get; set; }
    public string Title { get; set; }
    public string Link { get; set; }
    // Additional properties for films

    public List<Rating> Ratings { get; set; }
    public List<Comment> Comments { get; set; }
    public List<Category> Categories { get; set; }
    public List<View> Views { get; set; }
}
public class User
{
    public int Id { get; set; }
    public string Name { get; set; }
    // Additional properties for users

    public List<Rating> Ratings { get; set; }
    public List<Comment> Comments { get; set; }
    public List<Role> Roles { get; set; }
}
public class Role
{
    public int Id { get; set; }
    public string Name { get; set; }
    // Additional properties for roles

    public List<User> Users { get; set; }
}
public class Rating
{
    public int Id { get; set; }
    public int UserId { get; set; }
    public int FilmId { get; set; }
    public int Value { get; set; }
    // Additional properties for ratings

    public User User { get; set; }
    public Film Film { get; set; }
}
public class Comment
{
    public int Id { get; set; }
    public int UserId { get; set; }
    public int FilmId { get; set; }
    public string Text { get; set; }
```

```
    // Additional properties for comments

    public User User { get; set; }
    public Film Film { get; set; }
}
public class Category
{
    public int Id { get; set; }
    public string Name { get; set; }
    // Additional properties for categories

    public List<Film> Films { get; set; }
}
public class View
{
    public int Id { get; set; }
    public int UserId { get; set; }
    public int FilmId { get; set; }
    public DateTime ViewTime { get; set; }
    // Additional properties for views

    public User User { get; set; }
    public Film Film { get; set; }
}
```

## Додаток В. Angular реалізація фронт-енду на прикладі реєстрації

В Html файлі ми маємо таку розмітку:

```
<form (ngSubmit)="registerUser()" #registrationForm="ngForm">
  <div>
    <label for="name">Name</label>
    <input type="text" id="name" name="name" ngModel required>
  </div>
  <div>
    <label for="email">Email</label>
    <input type="email" id="email" name="email" ngModel required pattern="[a-z0-
9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$">
    <div *ngIf="registrationForm.controls.email.invalid &&
(registrationForm.controls.email.dirty ||
registrationForm.controls.email.touched)">
      <div *ngIf="registrationForm.controls.email.errors.required">
        Email is required.
      </div>
      <div *ngIf="registrationForm.controls.email.errors.pattern">
        Please enter a valid email address.
      </div>
    </div>
  </div>
  <div>
    <label for="password">Password</label>
    <input type="password" id="password" name="password" ngModel required>
  </div>
  <div>
    <label for="confirmPassword">Confirm Password</label>
    <input type="password" id="confirmPassword" name="confirmPassword" ngModel
required>
    <div *ngIf="registrationForm.controls.confirmPassword.invalid &&
(registrationForm.controls.confirmPassword.dirty ||
registrationForm.controls.confirmPassword.touched)">
      <div *ngIf="registrationForm.controls.confirmPassword.errors.required">
        Confirm Password is required.
      </div>
      <div *ngIf="registrationForm.controls.confirmPassword.value !==
registrationForm.controls.password.value">
        Password and Confirm Password do not match.
      </div>
    </div>
  </div>
  <button type="submit"
[disabled]="registrationForm.invalid">Register</button></form>
```

В файлі TypeScript ми маємо:

```
import { Component } from '@angular/core';import { UserService } from 'path-to-your-user-service';
@Component({
  selector: 'app-registration',
  templateUrl: './registration.component.html',
  styleUrls: ['./registration.component.css']
})export class RegistrationComponent {
  constructor(private userService: UserService) { }

  registerUser() {
    if (registrationForm.valid) {
      const name = registrationForm.value.name;
      const email = registrationForm.value.email;
      const password = registrationForm.value.password;

      // Call the user service to register the user
      this.userService.registerUser(name, email, password).subscribe(
        (response) => {
          // Registration successful, handle the response or navigate to another page
          console.log('Registration successful:', response);
        },
        (error) => {
          // Registration failed, handle the error
          console.error('Registration failed:', error);
        }
      );
    }
  }
}
```

В файлі сервісів описано функцію registerUser:

```
import { Injectable } from '@angular/core';import { HttpClient } from '@angular/common/http';
@Injectable({
  providedIn: 'root'
})export class UserService {
  private apiUrl = 'https://your-api-url.com';

  constructor(private http: HttpClient) { }

  registerUser(name: string, email: string, password: string) {
    const user = { name, email, password };

    // Send a POST request to your backend API to register the user
    return this.http.post(`${this.apiUrl}/register`, user);
  }
}
```