

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ПЛАТФОРМИ ДЛЯ ПЕРЕГЛЯДУ ІНФОРМАЦІЇ ПРО ФІЛЬМИ
RESEARCH AND SOFTWARE IMPLEMENTATION OF
A MOVIE INFORMATION PLATFORM

Виконав(ла): студент(ка) 2 курсу, групи КНТ-124м

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ГОРОШКО М.С.

(ПРИЗВИЩЕ та ініціали)

Керівник СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГОРОШКА Микити Сергійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація платформи для перегляду інформації про фільми. Research and Software Implementation of a Movie Information Platform

керівник проєкту (роботи) к.т.н., доцент, СКРУПСЬКИЙ Степан Юрійович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 30 ” вересня 2025 року № 447

2. Строк подання студентом проєкту (роботи) 02 грудня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	СКРУПСЬКИЙ С.Ю., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 30 ” вересня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	8 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Микита ГОРОШКО
(Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Степан СКРУПСЬКИЙ
(Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
81 с., 3 табл., 30 рис., 3 дод., 17 джерел.

JAVASCRIPT, REACT.JS, ЗАСТОСУНОК, МОВА ПРОГРАМУВАННЯ,
МОДЕЛЬ ПРОГРАМИ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ФІЛЬМ.

Об'єкт дослідження – процес розробки програмного забезпечення для перегляду інформації про фільми.

Предмет дослідження – програмні засоби для перегляду інформації про фільми.

Мета роботи – дослідження та розробка програмного забезпечення для перегляду інформації про фільми.

Матеріали, методи та технічні засоби: мова програмування JavaScript, фреймворк React.js.

Результати. Розроблено програмне забезпечення для перегляду інформації про фільми за допомогою мови програмування JavaScript та фреймворку React.js.

Практична цінність роботи полягає у розробці програмного забезпечення для перегляду інформації про фільми.

Висновки. Виконано проєктування програмного забезпечення для перегляду інформації про фільми. Розроблено програмне забезпечення для перегляду інформації про фільми. Здійснено тестування розробленого програмного забезпечення для перегляду інформації про фільми.

Галузь використання – розповсюдження кінопродукції.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 81 pages, 3 tables, 30 figures, 3 appendixes, 17 sources.

JAVASCRIPT, REACT.JS, APPLICATION, PROGRAMMING LANGUAGE, PROGRAM MODEL, SOFTWARE, MOVIE.

The object of research is the process of developing software for viewing information about films.

The subject of the research is software for viewing information about films.

The purpose of this work is to develop the software for viewing information about films.

Materials, methods and technical tools: JavaScript programming language, React.js framework.

Results. The software tools for viewing information about films using the JavaScript programming language and the React.js framework have been developed.

The practical value of the work lies in the development of software for viewing information about films.

Conclusions. The software for viewing information about films has been designed. The software for viewing information about films has been developed. The developed software for viewing information about films has been tested.

Scope of use – film distribution.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Програмне забезпечення для перегляду інформації про фільми	11
1.2 Аналіз програмного забезпечення для перегляду інформації про фільми.....	19
1.3 Висновки за розділом 1	27
2 Матеріали і методи	29
2.1 Вибір мови програмування.....	29
2.2 Вибір фреймворку для створення програмного забезпечення для перегляду інформації про фільми	32
2.3 Розробка моделі інтерфейсу користувача програмного забезпечення для перегляду інформації про фільми	35
2.4 Висновки за розділом 2	37
3 Опис програми	38
3.1 Структура програмного забезпечення для перегляду інформації про фільми.....	38
3.2 Функціонування програмного забезпечення для перегляду інформації про фільми	39
3.3 Розробка бази даних програмного забезпечення для перегляду інформації про фільми	41
3.4 Проєктування інтерфейсу програмного забезпечення для перегляду інформації про фільми	43
3.5 Висновки за розділом 3	46
4 Експлуатація, тестування та експериментальне дослідження програми.....	47
4.1 Призначення й умови застосування програми	47
4.2 Характеристики програми для перегляду інформації про фільми.....	48
4.3 Інструкція по експлуатації програми.....	50

4.3.1	Звернення до програми.....	50
4.3.2	Вхідні й вихідні дані.....	51
4.3.3	Повідомлення.....	51
4.4	Виконання програмного забезпечення для перегляду інформації про фільми.....	51
4.5	Тестування програмного забезпечення для перегляду інформації про фільми.....	54
4.6	Висновки за розділом 4	55
	Висновки.....	56
	Перелік джерел посилання	59
	Додаток А Технічне завдання.....	61
	Додаток Б Фрагмент тексту програми	65
	Додаток В Слайди презентації	74

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

MVC – Model-View-Controller;

БД – база даних;

МД – модель даних;

ПЗ – програмне забезпечення.

ВСТУП

В умовах стрімкого розвитку кіноіндустрії та збільшення обсягу контенту зростає потреба користувачів у швидкому та зручному доступі до достовірної інформації про фільми. Глядачі прагнуть оперативно дізнаватися про новинки кінопрокату, рейтинги, відгуки, акторський склад, жанрову приналежність, нагороди та інші важливі дані, що впливають на вибір фільму для перегляду. Програмне забезпечення для перегляду інформації про фільми дозволяє систематизувати великі масиви даних, спрощує пошук і фільтрацію контенту за різними критеріями та надає інтерактивні можливості для взаємодії користувачів — наприклад, залишення рецензій чи формування власних списків перегляду. Такий програмний інструмент є актуальним як для поодиноких глядачів, так і для кінокритиків, дослідників кіно та індустрії розваг загалом, оскільки сприяє підвищенню поінформованості та культурного розвитку аудиторії [1], [2].

Особливої актуальності програмне забезпечення для перегляду інформації про фільми набуває в умовах активного розвитку стримінгових сервісів та багатоплатформеного споживання контенту. Сучасний глядач має доступ до великої кількості фільмів і серіалів одночасно на різних пристроях. У такій ситуації користувачі потребують програмних інструментів, що допомагають орієнтуватися в різноманітті пропозицій, отримувати рекомендації відповідно до власних уподобань та ефективно планувати свій час для перегляду [3], [4].

Крім того, програмне забезпечення цього типу сприяє популяризації кінокультури та розвитку спільнот за інтересами. Завдяки інтеграції соціальних функцій, можливості оцінювання та обговорення фільмів створюється простір для обміну думками та формування колективної оцінки творів. Це не лише полегшує користувачам вибір, а й стимулює кінематографічну індустрію до підвищення якості продукції, орієнтуючись на вподобання та відгуки аудиторії [5], [6].

Проте деякі вебзастосунки для перегляду інформації про фільми мають переважаність інтерфейсу надлишковою рекламою чи додатковими модулями, що ускладнює пошук потрібних даних і відволікає користувача від основної мети — швидкого доступу до інформації. Окрім того, деякі програми для перегляду інформації про фільми не забезпечують достатнього рівня достовірності контенту: інформація про фільми може бути застарілою або неперевіреною, особливо якщо джерела не модеруються належним чином. Крім того, деякі програми збирають надмірну кількість персональних даних для формування рекомендацій чи реклами, що викликає занепокоєння з точки зору конфіденційності та безпеки користувачів. У результаті частина аудиторії може відмовитися від використання таких сервісів або шукати альтернативні рішення [1]-[6].

Тому актуальною є розробка програмного забезпечення для перегляду інформації про фільми. У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання розробки та дослідження програмного забезпечення для перегляду інформації про фільми.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для перегляду інформації про фільми;
- розробити модель інтерфейсу користувача програмного забезпечення для перегляду інформації про фільми;
- здійснити проєктування програмного забезпечення для перегляду інформації про фільми;
- створити програмне забезпечення для перегляду інформації про фільми;
- виконати тестування розробленого програмного забезпечення для перегляду інформації про фільми.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для перегляду інформації про фільми

Програмне забезпечення для перегляду інформації про фільми можна визначити як спеціалізований ресурс або платформу, що збирає, обробляє й подає користувачам різноманітні відомості про світ кіно. За допомогою таких програмних сервісів користувач отримує доступ до бази даних, де зручно знайти опис сюжету, дізнатися більше про акторський склад, режисера, жанр і всі супутні деталі, які можуть зацікавити перед переглядом. Такі програмні інструменти зазвичай стають надійним джерелом інформації, що дозволяє глибше зануритися у кінематографічну культуру та робити більш усвідомлений вибір, на що варто витратити свій час [1].

Великою перевагою таких програмних рішень є можливість швидко орієнтуватися у великій кількості стрічок, серіалів чи короткометражних проєктів, а також у доступі до думок інших глядачів і професійних критиків. Це поєднання думок допомагає скласти більш об'єктивну картину про якість і атмосферу фільму ще до перегляду. Ще однією вагомою перевагою вважається функція рекомендацій, яка зберігає інтерес аудиторії, пропонуючи нові твори, що відповідають особистим вподобанням. Так формується особливий простір, де можна не лише шукати інформацію, а й ділитися власними враженнями, обговорювати улюблені фільми та знаходити однодумців [2].

Щоб створити подібний програмний продукт, необхідно пройти кілька послідовних етапів: починається все з ретельного планування, де визначаються основні цілі, структура й перелік функцій, які потрібні для зручного користування. Далі важливо продумати дизайн інтерфейсу так, щоб він був зрозумілим і приємним для різної аудиторії. Значну роль відіграє вибір технологій, що забезпечать швидкість роботи, стабільність і можливість обробляти великий потік даних. Важливо інтегрувати систему для додавання та оновлення контенту, підключити інструменти для взаємодії з

користувачами та організувати безперебійну роботу з базами даних, де зберігаються всі описи, рецензії й рейтинги. Такий процес потребує уваги до деталей і постійного вдосконалення, адже з розвитком індустрії зростають і запити глядачів, яким цікаво отримувати якісну інформацію про кіно у зручній формі [1], [2].

Програмне забезпечення для перегляду інформації про фільми стало корисними помічниками для всіх, хто прагне бути в курсі кінопрем'єр та вибирати для перегляду найцікавіше. Такі ресурси допомагають не лише отримати стислий виклад сюжету, а й дізнатися про акторів, режисерів та всі деталі знімального процесу, що робить вибір стрічок більш усвідомленим і цікавим. Крім базової інформації, чимало подібних платформ доповнюються професійними рецензіями та думками звичайних глядачів, завдяки чому формується багатогранне бачення кожного фільму [2], [3].

Коли мова заходить про пошук надійних джерел, що збирають та публікують кінокритику, зазвичай обирають найбільш відомі портали з великою аудиторією. Такі проєкти зазвичай мають значні бази даних, розширені каталоги та спеціалізовані інструменти сортування, що дозволяє швидко знайти цікаві релізи або визначитися з тим, що варто подивитися найближчим часом. Чимало глядачів звертають увагу на те, як саме платформи поєднують професійний аналіз із думками глядачів, адже такий баланс допомагає побачити фільм з різних сторін [2], [3].

Тим, хто замислюється над створенням власного вебресурсу для висвітлення кіно, варто враховувати важливість продуманої структури й зручності у користуванні. Інтерактивні елементи, прості форми для коментарів та обговорень, розділи з рейтингами й добірками за жанрами роблять такі сайти більш привабливими для аудиторії. Правильно організована платформа здатна стати не просто архівом рецензій, а живим місцем для спілкування однодумців, де формується спільнота шанувальників кіномистецтва [3], [4].

Великою перевагою подібних програмних порталів є й те, що вони часто впливають на популярність фільмів у прокаті. Позитивні оцінки

сприяють тому, що дедалі більше людей звертають увагу на стрічки, які могли б залишитися непоміченими. Натомість різко негативна критика іноді здатна зменшити інтерес і навіть вплинути на фінансові збори. Тому такі сервіси не лише інформують, а й часто мають реальний вплив на розвиток кіноіндустрії, що робить їх важливим інструментом у сучасному світі розваг [3], [4].

Крім інформаційної функції, ці програмні платформи об'єднують людей за інтересами, надаючи їм можливість ділитися враженнями й обговорювати фільми разом. Відчуття єдності, яке виникає завдяки таким ресурсам, перетворює перегляд кіно з індивідуальної звички на частину спільного культурного простору, де кожен може висловити думку й знайти одностудців серед тих, хто розділяє любов до якісного кінематографа [3], [4].

У сучасному середовищі, де цифрові ресурси стали звичною частиною повсякденності, спеціалізоване програмне забезпечення, що надає доступ до інформації про кіно, посідає помітне місце серед численних онлайн платформ. Такі сервіси перетворилися на своєрідні центри зібраних знань, де можна знайти практично всі відомості про кінематограф — від коротких описів і розширених синопсисів до деталей знімального процесу, даних про акторів, режисерів, трейлерів і рецензій різних типів. Інтерес до подібних рішень тільки зростає, адже вони дозволяють за лічені хвилини отримати уявлення про будь-яку стрічку чи серіал, не витрачаючи час на довгі пошуки у численних джерелах [4], [5].

Популярність таких програмних продуктів пояснюється тим, що вони об'єднують у собі всі необхідні інструменти для зручного перегляду відомостей про кіно. Чимало користувачів шукають не лише основну інформацію про сюжет чи акторський склад, а й аналітичні огляди, думки професіоналів і персоналізовані добірки. Така комплексність полегшує вибір фільму для вечірнього перегляду, розширюючи горизонти знань про жанри, новинки чи рідкісні роботи. Завдяки зосередженню всього в одному місці глядач може не лише дізнаватися факти, а й брати участь у створенні

контенту, залишаючи власні оцінки й рецензії, що формує активну спільноту прихильників кіномистецтва [4], [5].

Поява й розвиток подібних платформ напряду пов'язані з постійним попитом на якісну, зручну й швидку інформацію про світ кіно. Розширення доступу до онлайн сервісів і зростання популярності стрімінгових послуг лише підкреслили потребу у джерелах, які дозволяють без зайвих зусиль знаходити найактуальніші дані про новинки кінопрокату й оцінювати їх ще до виходу на екрани. Багато хто звертає увагу й на можливість індивідуального підбору рекомендацій на основі попередніх уподобань, що робить використання таких сайтів ще більш привабливим [4], [5].

Важливим аспектом залишається інтерактивність, яка робить взаємодію з такими ресурсами не просто пошуком інформації, а справжнім спілкуванням у колі людей із подібними інтересами. Форуми, персональні рейтинги, коментарі та тематичні дискусії підтримують інтерес користувачів і сприяють обміну думками. Завдяки цьому платформи поступово стають не лише інформаційними майданчиками, а й простором для формування спільнот за інтересами [4], [5].

Щоб створити подібний ресурс, необхідно враховувати чимало факторів. Важливо подбати про наповнення великою базою фільмів і регулярне оновлення інформації, адже саме точність і повнота даних формують довіру до сервісу. Зручна структура та зрозуміла навігація теж мають неабияке значення — інтуїтивно побудований інтерфейс дозволяє швидко знайти потрібну сторінку чи розділ. Додаткові можливості, наприклад, перегляд трейлерів, особисті добірки чи система рекомендацій, роблять використання програмного забезпечення більш приємним і практичним для кожного [4], [5].

Власники подібних рішень прагнуть додавати нові функції, які поглиблюють взаємодію з аудиторією — від інтеграції з соціальними мережами до розділів з інтерв'ю та новинами про знімальні процеси. Так формується цілісне середовище, що поєднує інформацію, спілкування та

натхнення для тих, хто захоплюється кіномистецтвом і хоче відкривати його з нових сторін [4], [5].

Створення програмного продукту для перегляду інформації про фільми сьогодні сприймається як перспективна можливість для тих, хто прагне вийти на світову аудиторію й запропонувати сучасний спосіб доступу до великого каталогу контенту фільмів. Потреба у таких сервісах зростає через те, що глядачам більше не потрібно налаштовуватись під фіксовані сітки телепрограм чи чекати повторів у ефірі. Завдяки зручності Інтернету стало набагато простіше обирати фільми чи серіали за власним смаком і насолоджуватись ними у будь-який момент без зайвих обмежень [4], [5].

Найуспішніші приклади потокових сервісів доводять, що головна сила таких рішень полягає у здатності тонко враховувати вподобання аудиторії та швидко реагувати на зміни попиту. Платформи з великою кількістю підписників не тільки демонструють безперервний ріст, а й формують нові стандарти споживання кіноіндустрії у цифровому вигляді. Саме тому створення власного ресурсу для потокового показу фільмів вимагає зваженого підходу до всіх деталей — від технічної інфраструктури до добору контенту та організації користувацького інтерфейсу [4], [5].

Особлива увага приділяється тому, як саме платформа виглядає на різних пристроях і як швидко глядач отримує доступ до потрібного відео. Гнучкість у виборі якості зображення, просте перемикання між мовами, адаптація під швидкість мережі та легка навігація підвищують лояльність аудиторії. Додатковим плюсом стає можливість інтеграції соціальних опцій, що мотивує користувачів ділитися улюбленим контентом, обговорювати новинки та запрошувати друзів приєднуватися до перегляду [4], [5].

Для тих, хто прагне запустити власний проект у цьому напрямку, важливо продумати усі кроки: від вибору надійного середовища для хостингу відео до підбору інструментів для масштабування та монетизації. Зважений підхід до бізнес-моделі допомагає швидше вийти на самоокупність, а використання сучасних рішень дозволяє налаштувати платформу так, щоб

вона відповідала очікуванням глядачів. Завдяки правильному технічному рішенню навіть молодий сервіс здатен конкурувати з потужними гравцями, якщо він пропонує якісний контент, комфорт і багатофункціональність [4], [5].

Щоб розробити програмне забезпечення для перегляду інформації про фільми варто чітко зрозуміти, кому воно адресовано і які очікування у тих, хто користуватиметься таким сервісом. Це допомагає зосередитися на інформації, що дійсно буде потрібною та корисною: від описів і деталей знімальної групи до професійних рецензій і думок глядачів. Для цього програмний продукт має бути простим у використанні, мати зручну структуру та легку навігацію, що дозволяє швидко знаходити потрібні відомості без зайвих кроків [4], [5].

Важливо також подбати про технічну сторону. Програмне забезпечення для перегляду інформації про фільми повинно справлятися з великою кількістю користувачів одночасно, стабільно працювати та швидко завантажувати дані, навіть якщо інформаційна база включає тисячі назв. Для цього зазвичай обираються перевірені хостинг рішення, які гарантують безперебійну роботу сайту. Крім того, для багатьох людей важливим є вигляд і зручність інтерфейсу — тому варто передбачити дизайн, що однаково добре виглядає на будь-яких пристроях, а функції пошуку й фільтрації допомагають відразу знайти конкретний фільм чи рецензію [4], [5].

Ще однією частиною такої програмної системи стає база даних і механізми її оновлення. Без чіткого управління інформацією та постійного поповнення новинами і свіжими рецензіями жоден ресурс довго не буде актуальним. Тому потрібно передбачити зручну панель для адміністрування, через яку можна швидко додавати або редагувати картки фільмів, оновлювати біографії акторів чи публікувати нові інтерв'ю з творцями [5], [6].

Особливо цінним для користувачів програмного забезпечення для перегляду інформації про фільми залишається можливість залишати власні оцінки, коментарі чи обговорювати побачене з іншими шанувальниками. Такі опції роблять ресурс живим, залучають людей до спільноти й мотивують

повертатися знову. Для цього передбачаються розділи для відгуків і форуми, що сприяють активному обміну думками [5], [6].

Щоб програмне забезпечення для перегляду інформації про фільми залишалося цікавим і привабливим, воно повинно відповідати сучасним вимогам захисту інформації, зокрема щодо збереження особистих даних користувачів і запобігання несанкціонованому доступу. Водночас необхідно постійно стежити за якістю роботи ресурсу, збирати аналітику та впроваджувати нові можливості, які зроблять знайомство з кіносвітом ще простішим і цікавішим. Важливо забезпечити, щоб структура та навігація по вебсайту програмного забезпечення для перегляду інформації про фільми були інтуїтивно зрозумілими [5], [6].

Щоб програмне рішення для перегляду інформації про фільми виконувало свою мету максимально ефективно, варто подбати про гнучке наповнення та можливість постійно оновлювати картки фільмів. Така система повинна мати продуманий механізм зберігання й впорядкування відомостей, щоб у потрібний момент кожен користувач міг без зусиль знайти сюжетну лінію, деталі про акторів чи думки кінокритиків. Надійна платформа дозволяє розподіляти великі обсяги інформації швидко й без перебоїв, забезпечуючи стабільну роботу навіть при значному навантаженні [5], [6].

Особливу увагу слід приділяти зовнішньому вигляду та структурі інтерфейсу, адже саме від зрозумілості й привабливості залежить, наскільки охоче людина буде користуватися сервісом. Усе має бути інтуїтивно зрозумілим, а навігація — логічною й зручною. Добре продумане середовище дає можливість переходити між описами фільмів, новинами кіноіндустрії чи розділами з рецензіями без зайвих дій [5], [6].

Інша важлива частина — контроль за якістю та актуальністю контенту. Наявність панелі, що дозволяє гнучко редагувати опис, додавати новини, фільтрувати матеріали й керувати тегами, значно полегшує роботу адміністраторів і тримає базу фільмів у належному стані. Корисною стає і

система збору статистики, яка допомагає зрозуміти, які розділи цікавлять людей найбільше й де потрібно вдосконалити подачу матеріалу [5], [6].

Щоб програмний ресурс був справді живим, не менш важливою є можливість для відвідувачів ділитися своїми враженнями. Розділи для коментарів, опитування, персоналізовані добірки допомагають створити відчуття спільноти й мотивують повертатися за новими рекомендаціями чи обговореннями. Такі інтерактивні можливості роблять простір не лише корисним, а й захопливим [5], [6].

Захист особистих даних користувачів програмного забезпечення для перегляду інформації про фільми і безпека всієї платформи мають стояти окремим пріоритетом. Кожен повинен бути впевнений, що дані надійно зашифровані, а сторонні особи не отримають доступу до конфіденційної інформації [5], [6].

Підтримка зручного доступу з різних пристроїв значно розширює аудиторію й робить використання ресурсу гнучким. Усе це разом створює середовище, де можна легко знайти будь-яку інформацію про кіно, дізнатися про останні прем'єри, перечитати думки критиків чи знайти рекомендації для вечірнього перегляду. Саме такі можливості роблять програмне забезпечення для перегляду інформації про фільми незамінним помічником для тих, хто хоче орієнтуватися у світі кіно впевнено й без зайвих зусиль [5], [6].

Визначено, що програмне забезпечення для перегляду інформації про фільми являє собою спеціалізований ресурс або платформу, що збирає, обробляє й подає користувачам різноманітні відомості про світ кіно. За допомогою таких програмних сервісів користувач отримує доступ до бази даних, де зручно знайти опис сюжету, дізнатися більше про акторський склад, режисера, жанр і всі супутні деталі, які можуть зацікавити перед переглядом. Такі програмні інструменти зазвичай стають надійним джерелом інформації, що дозволяє глибше зануритися у кінематографічну культуру та робити більш усвідомлений вибір, на що варто витратити свій час.

1.2 Аналіз програмного забезпечення для перегляду інформації про фільми

Проаналізуємо програмне забезпечення для перегляду інформації про фільми [6]-[13].

Програмне забезпечення FilmAffinity (рис. 1.1) можна розглядати як платформу, що пропонує особливий підхід до оцінювання й пошуку фільмів, орієнтуючись на потреби кожного, хто прагне знайти щось нове для перегляду. Завдяки продуманій структурі цей сервіс надає можливість не лише переглядати вичерпну добірку кінострічок, а й виставляти власні оцінки, що формують персоналізовану систему рекомендацій, здатну врахувати найменші нюанси смаків глядача. Саме ця можливість індивідуального підбору виділяє ресурс серед інших подібних сайтів [6], [11].

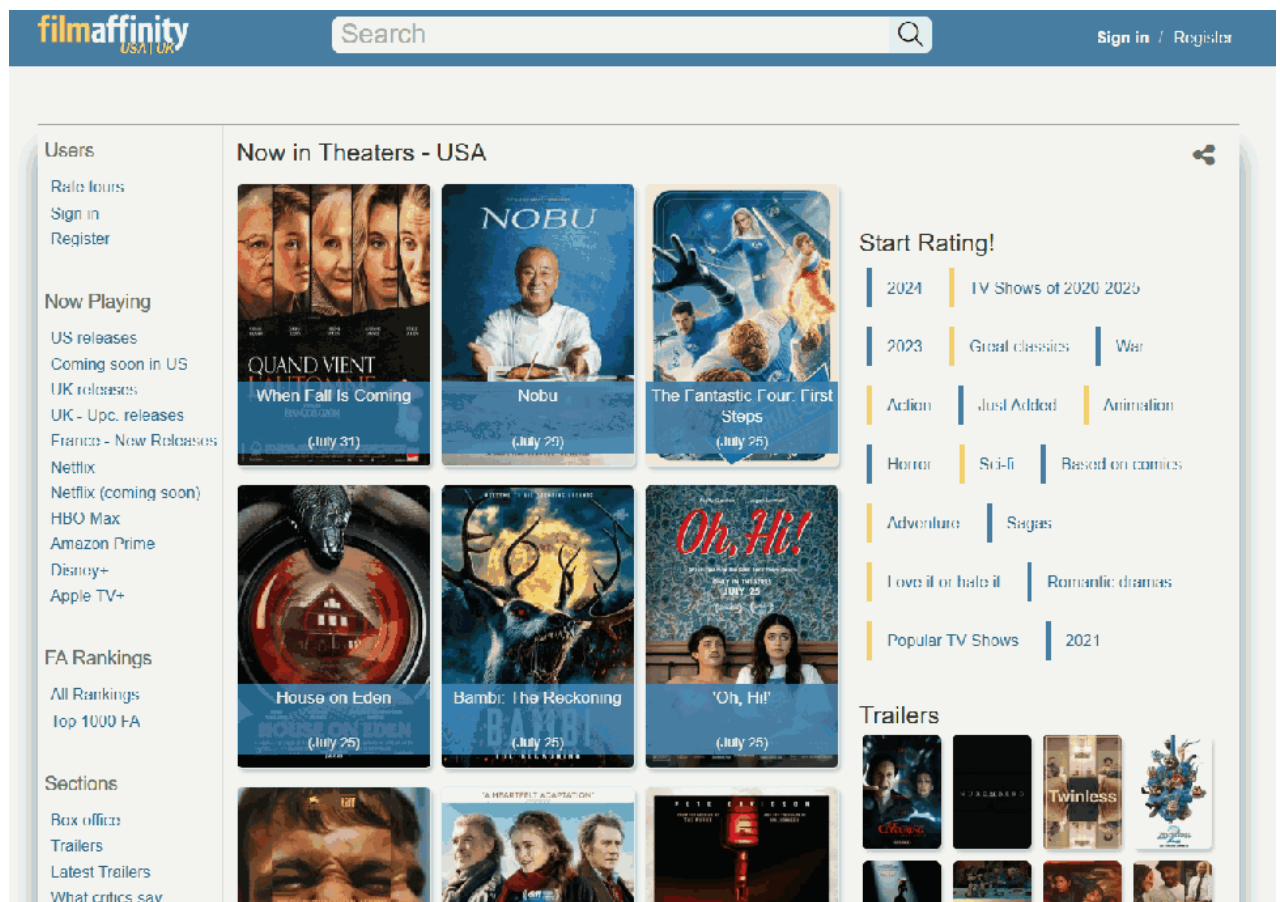


Рисунок 1.1 – Програмне забезпечення FilmAffinity [11]

Інтерфейс ресурсу FilmAffinity спроектовано так, щоб пошук і ознайомлення з інформацією не створювали зайвих труднощів. На сторінках легко знаходяться не лише загальні відомості про фільм, а й деталізовані описи, інформація про акторів і творчу команду, а також оцінки й думки тих, хто вже переглянув стрічку. Це дозволяє користувачеві скласти більш повну картину ще до перегляду [6], [11].

Відмінною рисою FilmAffinity залишається те, що сервіс не обмежується фільмами певного регіону чи мови, а відкриває доступ до різноманітних кінематографічних творів з усього світу. Такий підхід робить його привабливим для тих, хто прагне розширити коло знайомства з іноземними картинами та культурною спадщиною інших країн [6], [11].

Цікавою деталлю платформи FilmAffinity є спеціальна функція, що допомагає знаходити інших користувачів зі схожими вподобаннями. Такий інструмент дозволяє не тільки обмінюватися враженнями, а й відчувати приналежність до спільноти однодумців, які діляться порадами та відкривають маловідомі перлини кінематографа [6], [11].

Завдяки поєднанню детальної інформації, персоналізованих підказок і можливості спілкування FilmAffinity виходить за рамки простого каталогу оцінок і перетворюється на справжній провідник у світ кіно, де кожен має шанс знайти щось близьке до власних смаків і поділитися цим із тими, хто цінує якісний кіно контент [6], [11].

Основними характеристиками програмного забезпечення FilmAffinity є такі [6], [11]:

- персоналізовані рекомендації: програмне забезпечення FilmAffinity пропонує добірки фільмів, які формуються на основі індивідуальних оцінок і вподобань користувача [6], [11];

- велика база фільмів: програмне забезпечення FilmAffinity містить детальні описи, трейлери, списки акторів і знімальної групи для багатьох стрічок з різних країн світу [6], [11];

- простий інтерфейс: структура вебсайту програмного забезпечення FilmAffinity дозволяє швидко орієнтуватися у розділах, переглядати рейтинги та знаходити потрібний контент без зайвих кроків [6], [11];

- соціальні функції: програмне забезпечення FilmAffinity надає можливість спілкування та пошуку користувачів зі схожими кінематографічними смаками через функцію «Споріднені душі» [6], [11];

- міжнародний контент: програмне забезпечення FilmAffinity пропонує широкий вибір фільмів різними мовами, що приваблює користувачів з різних куточків світу [6], [11];

- система оцінювання: кожен зареєстрований користувач програмного забезпечення FilmAffinity може залишати власні оцінки та відгуки, впливаючи на рейтинг фільмів [6], [11].

Серед недоліків програмного забезпечення FilmAffinity можна відзначити певну обмеженість інтерфейсу для користувачів, які звикли до більш сучасного дизайну та інтеграції з іншими мультимедійними платформами. Іноді відчувається нестача додаткових інтерактивних функцій, які могли б розширити можливості персоналізації або зробити взаємодію з іншими користувачами більш зручною. Також окремі користувачі звертають увагу на обмеження у доступі до певного контенту залежно від країни або мовних налаштувань, що може створювати труднощі при пошуку рідкісних чи локальних фільмів [6], [11].

Програмне забезпечення Metacritic (рис. 1.2) являє собою сервіс, що об'єднує рецензії на кінострічки, отримані з великої кількості перевірених джерел, серед яких значну частину становлять професійні кінокритики. Особливістю ресурсу є система, що на основі зібраних оцінок формує єдиний середній показник, який допомагає відвідувачам відразу скласти уявлення про те, як фільм був сприйнятий експертами. Розподіл оцінок за шкалою спрощує вибір і дозволяє користувачам зрозуміти, чи відповідає фільм очікуванням ще до перегляду [8], [12].

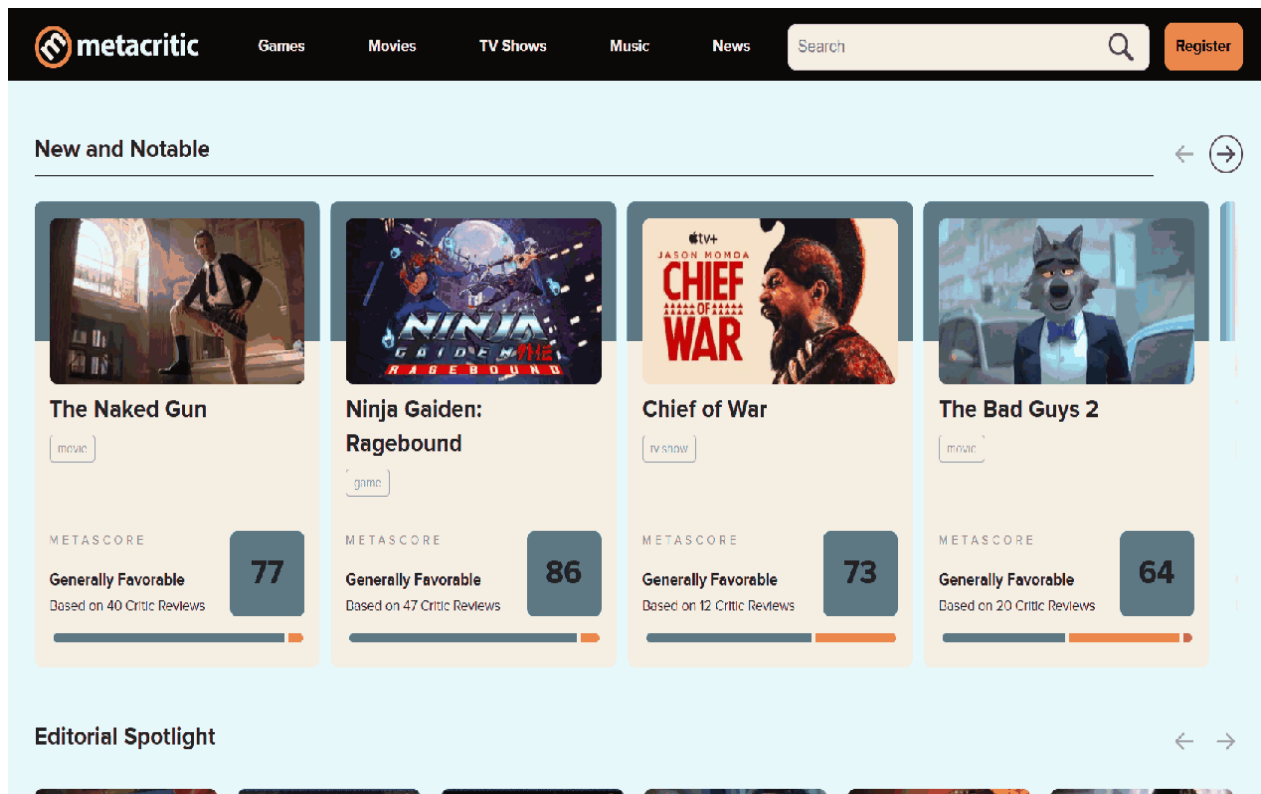


Рисунок 1.2 – Програмне забезпечення Metacritic [12]

Крім аналітичних рецензій від професіоналів, платформа дає змогу звичайним користувачам залишати власні оцінки та коментарі, ділитися враженнями й дискутувати з іншими шанувальниками кіномистецтва на спеціально відведених сторінках для обговорень. Такий підхід поєднує офіційну критичну оцінку з думками глядачів, що робить Metacritic корисним інструментом для тих, хто прагне скласти більш збалансоване враження про фільм ще до його перегляду [8], [12].

Програмне забезпечення Metacritic має такі особливості [8], [12]:

- агрегування рецензій: програмне забезпечення Metacritic збирає огляди з численних авторитетних джерел і формує середню оцінку [8], [12];
- користувацькі оцінки: окремо відображаються оцінки та коментарі глядачів, що дозволяє за допомогою програмного забезпечення Metacritic порівняти думки критиків і аудиторії [8], [12];
- зручна шкала балів: кожен фільм отримує підсумковий бал, що спрощує пошук якісного контенту [8], [12];

– обговорення на форумах: користувачі програмного забезпечення Metacritic можуть обмінюватися враженнями й дискутувати про новинки кіно у спеціальних розділах [8], [12];

– багатофункціональний каталог: окрім фільмів доступні рецензії на ігри, музику та серіали, що розширює сферу використання програмного забезпечення Metacritic [8], [12].

Серед недоліків програмного забезпечення Metacritic можна відзначити певну однобокість у формуванні середньої оцінки, оскільки вона іноді не відображає різноманіття поглядів і може не враховувати важливі нюанси, які помічають окремі користувачі. Деякі глядачі звертають увагу на те, що частина рецензій формується переважно західними виданнями, що обмежує погляд на фільми з інших культурних контекстів. Також можна відчутти брак глибокого обговорення у спільноті, оскільки форуми не завжди активні й не завжди підтримують живий обмін думками між користувачами [8], [12].

Програмне забезпечення IndieWire (рис. 1.3) є інформаційним ресурсом для користувачів, хто цікавиться некомерційним кінематографом і телебаченням. Колись цей проєкт починався як простий майданчик для обміну думками між режисерами та організаторами фестивалів, однак з часом він розрісся до масштабного медіапорталу, що щодня постачає світу актуальні новини зі світу незалежного кіно, свіжі рецензії на прем'єри, аналітику ринку, ексклюзивні інтерв'ю з режисерами, акторами та іншими представниками творчої індустрії. Особливої ваги цей ресурс набув серед тих, хто створює авторське кіно та прагне знайти свою аудиторію за межами масових студійних релізів. IndieWire став не просто віртуальним простором, де можна прочитати огляд чи подивитися рейтинг, а своєрідною платформою для обговорення тенденцій, які формують сучасне незалежне кіно. Тут можна знайти інформацію про найпомітніші фестивалі, дізнатися про проєкти, які лише готуються до прем'єри, і простежити, як змінюється індустрія під впливом нових ідей та підходів до виробництва [10], [13].



Рисунок 1.3 – Програмне забезпечення IndieWire [13]

Матеріали, які публікуються на ресурсі IndieWire, відрізняються ретельно підібраним контентом, що створюється виключно професійними журналістами й кінокритиками. Кожен текст вирізняється глибоким аналізом, іноді з суб'єктивними нотками, що дозволяє глянути на світ кіно з різних ракурсів. Незважаючи на те, що на вебсайті програмного забезпечення IndieWire не передбачено відкритої платформи для користувацького контенту чи форумів для широкого обговорення, ця особливість не применшує його значущості для тих, хто цінує експертну оцінку і прагне отримувати вичерпні матеріали про творчий процес [10], [13].

Програмне забезпечення IndieWire протягом багатьох років залишається точкою тяжіння для тих, хто стежить за змінами у світовому кінематографі. Завдяки публікаціям на сайті багато незалежних робіт і режисерів отримують додаткову увагу, яку складно було б привернути без

такого інформативного та впливового посередника між творцями і глядачами. Водночас платформа не обмежується лише публікацією рецензій чи новин: вона формує контекст, у якому глядачі та фахівці можуть краще зрозуміти, як розвивається культура авторського кіно у світі, які теми стають актуальними й які обличчя стоять за цією творчою хвилею [10], [13].

Програмне забезпечення IndieWire має такі особливості [10], [13]:

- професійні огляди: усі матеріали на вебсайті програмного забезпечення IndieWire створюються досвідченими кінокритиками та журналістами, що гарантує глибину й експертність контенту [10], [13];
- аналітика та інтерв'ю: окрім рецензій на вебсайті програмного забезпечення IndieWire публікуються інтерв'ю з творцями та аналітичні матеріали про тенденції у сфері кіно і телебачення [10], [13];
- відсутність користувацьких рецензій: програмне забезпечення IndieWire не передбачає можливості для глядачів додавати власні відгуки чи брати участь у форумах [10], [13];
- акцент на фестивалях: значна увага на вебсайті програмного забезпечення IndieWire приділяється висвітленню кінофестивалів з усього світу, що допомагає стежити за найважливішими подіями в індустрії [10], [13].

Серед недоліків програмного забезпечення IndieWire можна відзначити обмежені можливості для залучення користувачів до активного обговорення матеріалів, оскільки відсутність функції публікації власних рецензій чи коментарів звужує простір для взаємодії між читачами. Деяким користувачам може бракувати більш різноманітного контенту, що враховував би не лише думку професійних критиків, а й погляди широкої аудиторії. Крім того, через сильний фокус на авторському сегменті частина масового глядача може не знайти тут достатньо інформації про великі студійні релізи, що обмежує охоплення тем і інтересів ширшої кінематографічної спільноти [10], [13].

Порівняльну характеристику програмного забезпечення для перегляду інформації про фільми наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для перегляду інформації про фільми

Критерій порівняння	FilmAffinity [11]	Metacritic [12]	IndieWire [13]
Персоналізовані рекомендації	+	+—	—
Складність інтерфейсу	+—	+	+—
Користувацькі оцінки	+	+	—
Професійні огляди	+—	+	+
Багатомовність	+	+	+—
Інтерв'ю та аналітика	—	+—	+
Соціальна взаємодія	+—	+—	—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для перегляду інформації про фільми. Проте деякі програмні засоби для перегляду інформації про фільми характеризуються перевантаженістю інтерфейсу надлишковою рекламою чи додатковими модулями, що ускладнює пошук потрібних даних і відволікає користувача від основної мети — швидкого доступу до інформації. Окрім того, деякі програми для перегляду інформації про фільми не забезпечують достатнього рівня достовірності контенту: інформація про фільми може бути застарілою або неперевіреною, особливо якщо джерела не модеруються належним чином. Крім того, деякі програми збирають надмірну кількість персональних даних для формування рекомендацій чи реклами, що викликає занепокоєння з точки зору конфіденційності та безпеки користувачів. У результаті частина аудиторії може відмовитися від використання таких сервісів або шукати альтернативні рішення. Тому актуальною є розробка програмного забезпечення для перегляду інформації про фільми.

При розробці програмного забезпечення для перегляду інформації про фільми необхідно забезпечити такі функціональні вимоги:

- підтримка відображення блоку з актуальними фільмами, які вийшли у показ останнім часом;
- підтримка можливості відображення списку фільмів, які є у тренді користувачів вебресурсу;
- можливість відображення переліку фільмів, інформація про які зберігається у базі застосунку;
- можливість відображення детальної інформації про фільм, зокрема, його опис, рік створення, жанр, акторський склад та ін.;
- підтримка можливості відображення сторінки Watchlist зі списком фільмів для подальшого перегляду.

1.3 Висновки за розділом 1

Визначено, що програмне забезпечення для перегляду інформації про фільми являє собою спеціалізований ресурс або платформу, що збирає, обробляє й подає користувачам різноманітні відомості про світ кіно. За допомогою таких програмних сервісів користувач отримує доступ до бази даних, де зручно знайти опис сюжету, дізнатися більше про акторський склад, режисера, жанр і всі супутні деталі, які можуть зацікавити перед переглядом. Такі програмні інструменти зазвичай стають надійним джерелом інформації, що дозволяє глибше зануритися у кінематографічну культуру та робити більш усвідомлений вибір, на що варто витратити свій час.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для перегляду інформації про фільми. Проте деякі програмні засоби для перегляду інформації про фільми характеризуються перевантаженістю інтерфейсу надлишковою рекламою чи додатковими модулями, що ускладнює пошук потрібних даних і відволікає користувача від основної мети — швидкого доступу до інформації. Окрім того, деякі програми для перегляду інформації про фільми не забезпечують достатнього рівня достовірності контенту: інформація про фільми може бути

застарілою або неперевіреною, особливо якщо джерела не модеруються належним чином. Крім того, деякі програми збирають надмірну кількість персональних даних для формування рекомендацій чи реклами, що викликає занепокоєння з точки зору конфіденційності та безпеки користувачів. У результаті частина аудиторії може відмовитися від використання таких сервісів або шукати альтернативні рішення. Тому актуальною є розробка програмного забезпечення для перегляду інформації про фільми.

Сформульовано функціональні вимоги до програмного забезпечення для перегляду інформації про фільми.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для перегляду інформації про фільми було використано мову програмування JavaScript [14], [15].

Мова JavaScript є однією з найгнучкіших мов програмування для створення сучасних вебінтерфейсів, здатних оперативно реагувати на будь-які дії відвідувача платформи. Широка підтримка мовних стандартів і регулярне вдосконалення функціональних можливостей надають розробникам простір для реалізації найрізноманітніших сценаріїв інтерактивної взаємодії, у тому числі через підключення зовнішніх сервісів за допомогою численних API. Завдяки мові JavaScript зручно будувати вебзастосунки, що працюють у форматі єдиної сторінки, де оновлення контенту відбувається динамічно, без потреби повного перезавантаження, що позитивно впливає на швидкодію системи та сприйняття користувачем [14], [15].

Не можна залишати без уваги й ту обставину, що JavaScript підтримується потужною професійною спільнотою, яка щодня розвиває екосистему, наповнюючи її численними бібліотеками, наборами компонентів і додатковими інструментами, які можна задіяти у власних проєктах без потреби створювати все з нуля. Така різноманітність доступних рішень дає змогу помітно скоротити часові та фінансові витрати під час розробки й спростити технічну підтримку та масштабування. Завдяки швидкому впровадженню новітніх підходів та технологій JavaScript утримує провідні позиції серед мов, що дозволяють створювати продукти, які відповідають зростаючим очікуванням користувачів і ринковим тенденціям. Така гнучкість і здатність до швидкого оновлення роблять цю мову ключовим інструментом для команд, які прагнуть реалізувати інноваційні проєкти з мінімальними технічними бар'єрами та максимальною продуктивністю [14], [15].

Використання JavaScript у процесі створення платформи, що надає користувачам доступ до детальної інформації про фільми, можна вважати

обґрунтованим і практичним рішенням. Ця мова програмування відрізняється тим, що її підтримують усі сучасні браузері, що дає змогу одразу охопити широку аудиторію, незалежно від того, на якому пристрої здійснюється вхід — настільному комп'ютері, ноутбуку чи мобільному телефоні. Завдяки цьому немає необхідності змушувати користувача встановлювати додаткові програми чи плагіни, що значно спрощує доступ до сервісу [14], [15].

Особливістю JavaScript є потужна база готових інструментів, серед яких фреймворки та бібліотеки, що допомагають розробникам швидко збирати гнучкі інтерфейси з продуманою навігацією і зручною подачею великої кількості даних. Саме тому ця мова вважається вдалою для побудови платформ, де потрібно організувати безперервний пошук і фільтрацію великого масиву фільмографічної інформації у зрозумілому й привабливому форматі [14], [15].

Не менш важливим є і те, що JavaScript органічно поєднується з середовищами, що відповідають за логіку на сервері. Завдяки цьому зникає потреба комбінувати різні мови програмування для різних частин системи, що робить увесь процес розробки більш послідовним і зрозумілим. Такий підхід скорочує ймовірність технічних помилок, полегшує тестування і відкриває можливості для швидкого оновлення або розширення функціоналу платформи. У результаті вибір JavaScript забезпечує розробницькій команді гнучке середовище, пришвидшує реалізацію проєкту та дає змогу користуватися підтримкою великої професійної спільноти, що накопичила значний досвід і чимало готових рішень для типових задач [14], [15].

Завдяки екосистемі JavaScript розробники мають доступ до численних бібліотек і фреймворків (наприклад, React, Vue.js або Angular), що дозволяють швидко створювати динамічні, зручні та інтуїтивно зрозумілі інтерфейси користувача. Це особливо важливо для сервісів, орієнтованих на велику кількість даних, які потрібно зручно фільтрувати, шукати та відображати у привабливому вигляді [14], [15].

Обґрунтування вибору мови програмування для реалізації програмного забезпечення для перегляду інформації про фільми наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для реалізації програмного забезпечення для перегляду інформації про фільми

Критерій порівняння мов програмування	Мова програмування		
	JavaScript	Python	Java
Підтримка роботи у веббраузері	+	—	—
Зручність створення інтерактивного інтерфейсу користувача	+	+—	+—
Велика кількість готових бібліотек для веброзробки	+	+	+—
Зручність для створення ПЗ для перегляду інформації про фільми	+	—	—
Швидкість прототипування	+	+	+—

Отже, для реалізації програмного забезпечення для перегляду інформації про фільми обрано мову програмування JavaScript, яка є однією з найбільш універсальних мов для створення інтерфейсів, що реагують у реальному часі та забезпечують швидкий відгук на дії користувача. Крім того, JavaScript має потужну спільноту, яка постійно створює і підтримує велике різноманіття плагінів, модулів і готових компонентів, це відкриває широкі можливості для оптимізації витрат часу та ресурсів у процесі розробки. Завдяки активному впровадженню нових стандартів та інструментів мова JavaScript зберігає високу актуальність і дозволяє швидко адаптувати проекти до сучасних вимог користувачів.

2.2 Вибір фреймворку для створення програмного забезпечення для перегляду інформації про фільми

В якості фреймворку для створення програмного забезпечення для перегляду інформації про фільми було обрано React.js [16], [17].

React.js відомий як інструмент, що відкриває широкі можливості для реалізації рішень, здатних витримувати значні навантаження й одночасно забезпечувати стабільну взаємодію з великою кількістю користувачів. Його архітектурні особливості дають змогу організовувати роботу застосунків так, щоб потоки даних оброблялися без затримок і блокувань, що є особливо важливим для платформ, де дані оновлюються безперервно. У випадку сервісів, пов'язаних із фільмами, це означає можливість безперервного відображення змін — нові оцінки, коментарі чи оновлення бази фільмів миттєво з'являються для кожного відвідувача, що підвищує зручність користування ресурсом [16], [17].

Використання великої кількості готових пакетів і модулів, які пропонує екосистема React.js, дозволяє скоротити зусилля, витрачені на з'єднання з різноманітними зовнішніми ресурсами, такими як сховища даних чи сторонні API. Це значно спрощує організацію взаємодії між компонентами системи, прискорює час розробки й мінімізує ризик технічних проблем, пов'язаних із підтримкою сумісності. Особливо вигідним такий підхід виявляється для платформ, що мають обробляти численні паралельні запити, адже неблокувальна модель дозволяє не знижувати швидкість роботи навіть за умов пікового навантаження [16], [17].

Варто також звернути увагу на здатність React.js масштабуватися разом із розвитком проєкту. У випадку розширення функцій чи підвищення кількості відвідувачів існує можливість безболісно перенести систему на більш потужні сервери або налаштувати хмарну інфраструктуру, не змінюючи ключових частин програмного коду. Така гнучкість і стабільність роботи роблять React.js одним із найбільш доцільних варіантів для реалізації сучасних вебплатформ,

що повинні працювати швидко, обробляти великі обсяги запитів і безперервно оновлювати контент відповідно до актуальної інформації [16], [17].

Застосування React.js під час створення платформи, що надає доступ до інформації про фільми, обґрунтовується тим, що цей інструмент дає змогу будувати повноцінну архітектуру, використовуючи єдину мову програмування. Такий підхід дозволяє організувати єдину логіку роботи як на стороні клієнта, так і на сервері, що позитивно впливає на узгодженість усіх частин системи та значно полегшує весь цикл розробки — від початкового проектування до подальшого обслуговування та внесення змін. Зникає потреба у складних рішеннях для поєднання різнорідних мов і технологій, що нерідко стає джерелом технічних помилок і затримок під час впровадження нових функцій [16], [17].

Ще однією характерною рисою React.js залишається його здатність витримувати значні навантаження. Для ресурсів, де одночасно велика кількість відвідувачів переглядає описи фільмів, шукає трейлери чи читає відгуки, критично важливо зберігати швидкодію та стабільність навіть у години пікової активності. Асинхронна модель обробки запитів у поєднанні з неблокувальною архітектурою дає можливість миттєво відповідати на численні запити без втрати продуктивності, що робить платформу зручною у користуванні та мінімізує ризик збоїв через перевантаження сервера [16], [17].

Значною перевагою цієї технології є різноманіття готових модулів і бібліотек, доступних у рамках екосистеми React.js. Така відкрита база рішень дозволяє без зайвих витрат часу впроваджувати додаткові сервіси, налаштовувати доступ до баз даних, підключати зовнішні інтерфейси чи системи реєстрації користувачів. Активна спільнота розробників, що постійно вдосконалює середовище та ділиться новими напрацюваннями, сприяє тому, що створення платформи не перетворюється на тривалий і дорогий процес. Натомість можна гнучко реагувати на зміни у вимогах ринку й оперативно додавати функціонал, який підвищує привабливість сервісу для кінцевого користувача. Завдяки такому поєднанню ефективності, швидкої адаптації та

стабільної роботи React.js залишається одним із найбільш практичних рішень для розробки сучасних вебресурсів, які обробляють великі масиви інформації у режимі реального часу [16], [17].

Обґрунтування вибору фреймворку для створення програмного забезпечення для перегляду інформації про фільми наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору фреймворку для створення програмного забезпечення для перегляду інформації про фільми

Критерій порівняння фреймворків	Фреймворки		
	React.js	Next.js	Express.jsp
Широта використання для серверної логіки	+	+—	+—
Гнучкість побудови архітектури	+	+—	+—
Велика кількість доступних модулів і пакетів	+	—	—
Продуктивність при обробці одночасних запитів	+	+—	+—
Гнучкість вибору фреймворків для клієнтської частини	+	+	+—

Таким чином, для створення програмного забезпечення для перегляду інформації про фільми обрано фреймворк React.js, який дозволяє використовувати мову програмування JavaScript як на клієнтській, так і на серверній стороні, що спрощує взаємодію між компонентами програмної системи, пришвидшує розробку та полегшує підтримку проєкту. Завдяки цьому зменшується кількість помилок, які часто виникають під час інтеграції різних технологій. Крім цього, React.js вирізняється високою продуктивністю

при обробці великої кількості одночасних запитів, що є критично важливим для сервісів, де тисячі користувачів можуть шукати, фільтрувати та переглядати інформацію одночасно.

2.3 Розробка моделі інтерфейсу користувача програмного забезпечення для перегляду інформації про фільми

З метою можливості розробки програмного забезпечення для перегляду інформації про фільми створимо відповідну модель інтерфейсу користувача у вигляді прототипу головної вебсторінки програми.

При розробленні моделі інтерфейсу користувача програмного забезпечення для перегляду інформації про фільми враховуємо те, що користувачі програми мають можливості, що реалізують вимоги до програми, зазначені у технічному завданні.

Модель прототипу головної сторінки програмного забезпечення, призначеного для перегляду детальної інформації про фільми, відображає візуальну організацію ключових компонентів і принципи розміщення елементів у межах вебінтерфейсу. На графічному макеті окреслено структуру сторінки, що умовно поділяється на кілька зон, кожна з яких відповідає певній функціональності. У верхній частині передбачено заголовковий блок із логотипом платформи, рядком пошуку та навігаційним меню. Для цих текстових елементів задано розміри шрифтів, які забезпечують помітність і зручне сприйняття — наприклад, назва ресурсу оформлена більшим шрифтом, тоді як пункти меню та пошук мають оптимальний розмір для легкого читання.

Враховуючи особливості взаємодії користувачів програмного забезпечення для перегляду інформації про фільми при роботі з програмою, модель інтерфейсу користувача подамо у вигляді головної вебсторінки програми, наведеної на рис. 2.1.

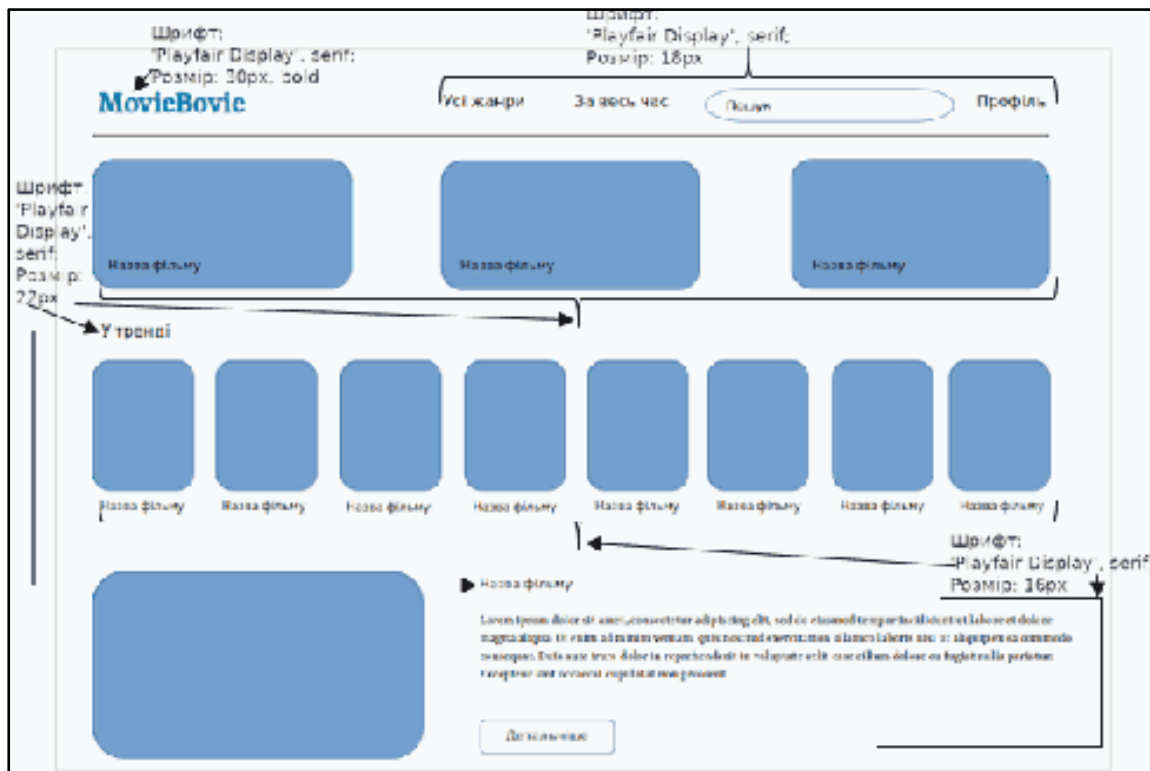


Рисунок 2.1 – Модель прототипу головної сторінки програмного забезпечення для перегляду інформації про фільми

Основну область прототипу головної сторінки програмного забезпечення для перегляду інформації про фільми займає секція з динамічним списком фільмів або афіш, де передбачено схематичне розташування постерів, назв, коротких описів і кнопок для переходу до повної інформації. Для кожного блока визначено пропорції зображень, відступи між картками та типографічні параметри тексту, що дозволяє забезпечити візуальну рівновагу й однаковий стиль подання даних.

На графічній моделі також відмічено допоміжні елементи. Вказано розміри шрифтів для цих елементів та їхнє відносне розташування на сторінці. Загальний вигляд прототипу демонструє баланс між інформативністю та зручністю навігації, а схема шрифтів і блоків допомагає оцінити, як виглядатиме сторінка у фінальній реалізації та наскільки легко користувачеві взаємодіяти з основними функціями платформи.

Таким чином, створено модель інтерфейсу користувача програмного забезпечення для перегляду інформації про фільми, яка подана у вигляді

прототипу вебсторінки, та дозволяє розроблювати програмне забезпечення для перегляду інформації про фільми.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для перегляду інформації про фільми обрано мову програмування JavaScript, яка є однією з найбільш універсальних мов для створення інтерфейсів, що реагують у реальному часі та забезпечують швидкий відгук на дії користувача. Крім того, JavaScript має потужну спільноту, яка постійно створює і підтримує велике різноманіття плагінів, модулів і готових компонентів, це відкриває широкі можливості для оптимізації витрат часу та ресурсів у процесі розробки.

Для створення програмного забезпечення для перегляду інформації про фільми обрано фреймворк React.js, який дозволяє використовувати мову програмування JavaScript як на клієнтській, так і на серверній стороні, що спрощує взаємодію між компонентами програмної системи, пришвидшує розробку та полегшує підтримку проєкту. Завдяки цьому зменшується кількість помилок, які часто виникають під час інтеграції різних технологій. Крім цього, React.js вирізняється високою продуктивністю при обробці великої кількості одночасних запитів, що є критично важливим для сервісів, де тисячі користувачів можуть шукати, фільтрувати та переглядати інформацію одночасно.

Створено модель інтерфейсу користувача програмного забезпечення для перегляду інформації про фільми, яка подана у вигляді прототипу вебсторінки, та дозволяє розроблювати програмне забезпечення для перегляду інформації про фільми.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для перегляду інформації про фільми

Структуру програмного забезпечення для перегляду інформації про фільми подано за допомогою діаграми класів, наведеної на рис. 3.1.

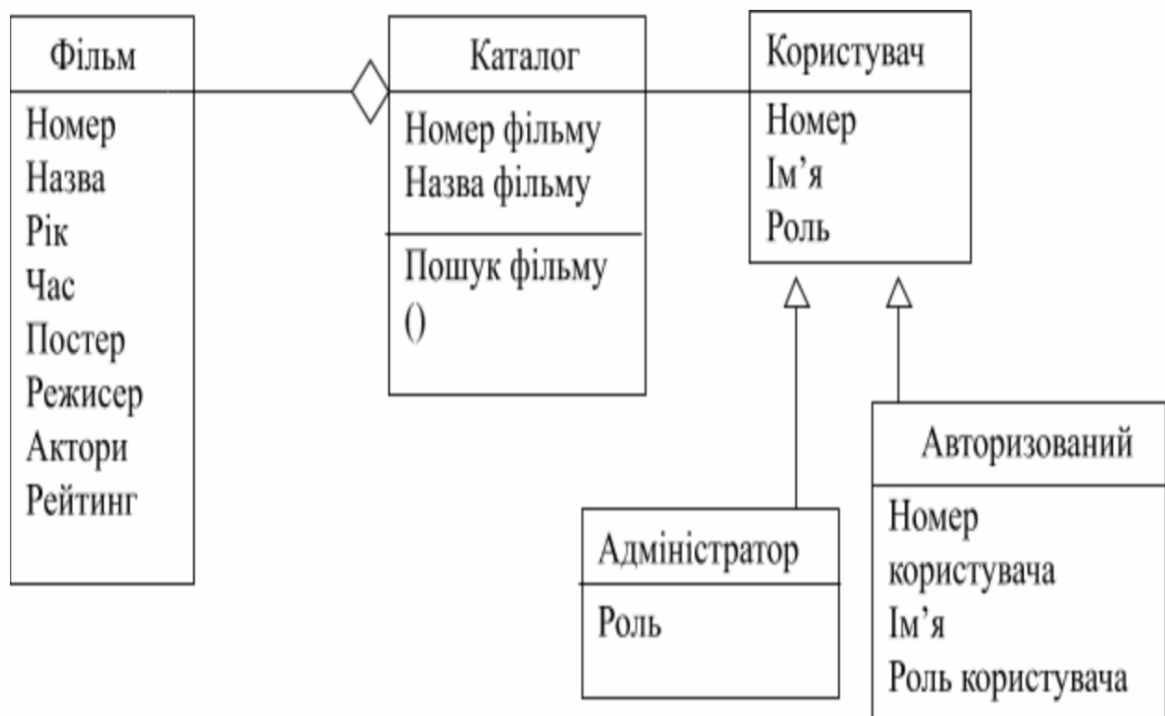


Рисунок 3.1 – Структура програмного забезпечення для перегляду інформації про фільми у вигляді діаграми класів

Як видно з рис. 3.1, основними класами програмного забезпечення для перегляду інформації про фільми є Користувач, Фільм, Каталог та інші.

Діаграма класів, що описує структуру програмного забезпечення для зручного доступу до інформації про фільми, демонструє взаємозв'язки між основними об'єктами системи та принципи організації їхніх властивостей і функцій. На схемі виділено кілька ключових класів, які відображають головних учасників та сутності в роботі платформи.

Центральне місце займає клас «Користувач», що відповідає за зберігання даних про відвідувача сервісу. До цього класу проведено зв'язки від двох інших: «Адміністратор» і «Авторизований», що вказує на те, що обидва є спеціалізованими розширеннями загального типу користувача. Таке успадкування демонструє, що адміністратор має додаткові повноваження для керування контентом, тоді як авторизований користувач отримує можливість залишати рецензії чи створювати власні списки вподобаних фільмів.

Окремо представлено клас «Фільм», який містить характеристики, що описують кожен кінострічку: назву, опис, жанр, рік випуску, рейтинг та інші важливі атрибути. Для впорядкування та групування інформації використовується клас «Каталог», що поєднує кілька об'єктів «Фільм» у загальну колекцію, спрощуючи пошук або фільтрацію за заданими критеріями. На діаграмі простежується зв'язок між каталогом і фільмом, який демонструє, що кожен каталог може містити багато фільмів, тоді як окремий фільм належить щонайменше одному каталогу або категорії.

Уся структура класів показує, як різні ролі користувачів взаємодіють із контентом і між собою, а також підкреслює важливість організації даних для забезпечення швидкого доступу до потрібної інформації. Такий підхід дозволяє створити гнучку та розширювану архітектуру, що може адаптуватися під розвиток платформи та додавання нового функціоналу.

3.2 Функціонування програмного забезпечення для перегляду інформації про фільми

Функціонування програмного забезпечення для перегляду інформації про фільми подамо за допомогою схеми, зображеної на рис. 3.2.

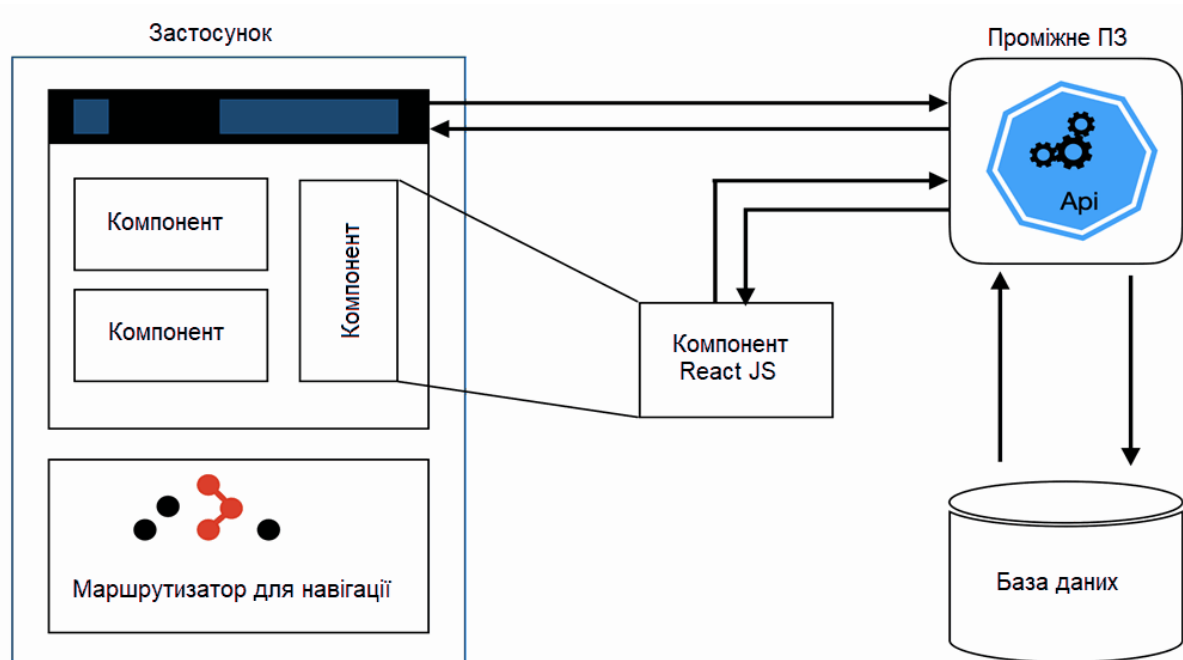


Рисунок 3.2 – Схема функціонування програмного забезпечення для перегляду інформації про фільми

Схема функціонування програмного забезпечення для перегляду інформації про фільми відображає взаємодію між основними складовими системи. На ній показано, як кожен елемент виконує свою роль у загальному процесі обробки запитів і передачі даних.

Ключову позицію займає Застосунок, який слугує вихідною точкою для користувача: саме через нього надсилаються запити, наприклад, для пошуку певного фільму чи фільтрації за жанром або рейтингом. Від Застосунку інформація потрапляє у Проміжне ПЗ, блок який відповідає за обробку всіх запитів, забезпечуючи логіку виконання, безпеку та перевірку даних перед тим, як вони підуть далі. Проміжне програмне забезпечення є своєрідним посередником між інтерфейсом і сховищем даних, воно регулює послідовність дій і забезпечує правильний обмін інформацією.

Важливим вузлом є База даних, де зберігаються всі відомості про фільми: назви, описи, афіші, рейтинги та супутня інформація. Саме звідти Проміжне ПЗ отримує дані у відповідь на запити Застосунку, формуючи структуровану відповідь.

Окремо на схемі виділено Компонент React JS, який відповідає за побудову користувацького інтерфейсу у браузері. Саме цей компонент отримує дані, що повертаються через Проміжне ПЗ, і динамічно відображає їх у зручному для користувача вигляді. Завдяки React JS сторінка швидко реагує на введення, без потреби повного перезавантаження — оновлюються лише ті частини інтерфейсу, що змінилися, що робить взаємодію плавною й ефективною.

Таким чином, схема наочно демонструє послідовність руху даних від запиту до результату: від моменту звернення у Застосунку до обробки проміжним шаром, отримання відповіді з Баз даних і фінального виведення через React JS. Усі елементи працюють як єдиний механізм, забезпечуючи безперервний обмін інформацією та стабільну роботу сервісу незалежно від кількості користувачів.

3.3 Розробка бази даних програмного забезпечення для перегляду інформації про фільми

Схема бази даних програмного забезпечення для перегляду інформації про фільми відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Основними сутностями бази даних обрано такі: Користувач, Фільми, Рейтинг, WatchList, Жанри та інші.

Структура бази даних, що обслуговує програмне забезпечення для перегляду детальної інформації про фільми, відображає взаємозв'язки між основними таблицями та сутностями, що формують ядро всієї системи. На схемі відображається, як окремі елементи пов'язуються між собою для забезпечення повноти й точності інформації, яку платформа надає своїм користувачам.

Схему бази даних програмного забезпечення для перегляду інформації про фільми наведено на рис. 3.3.

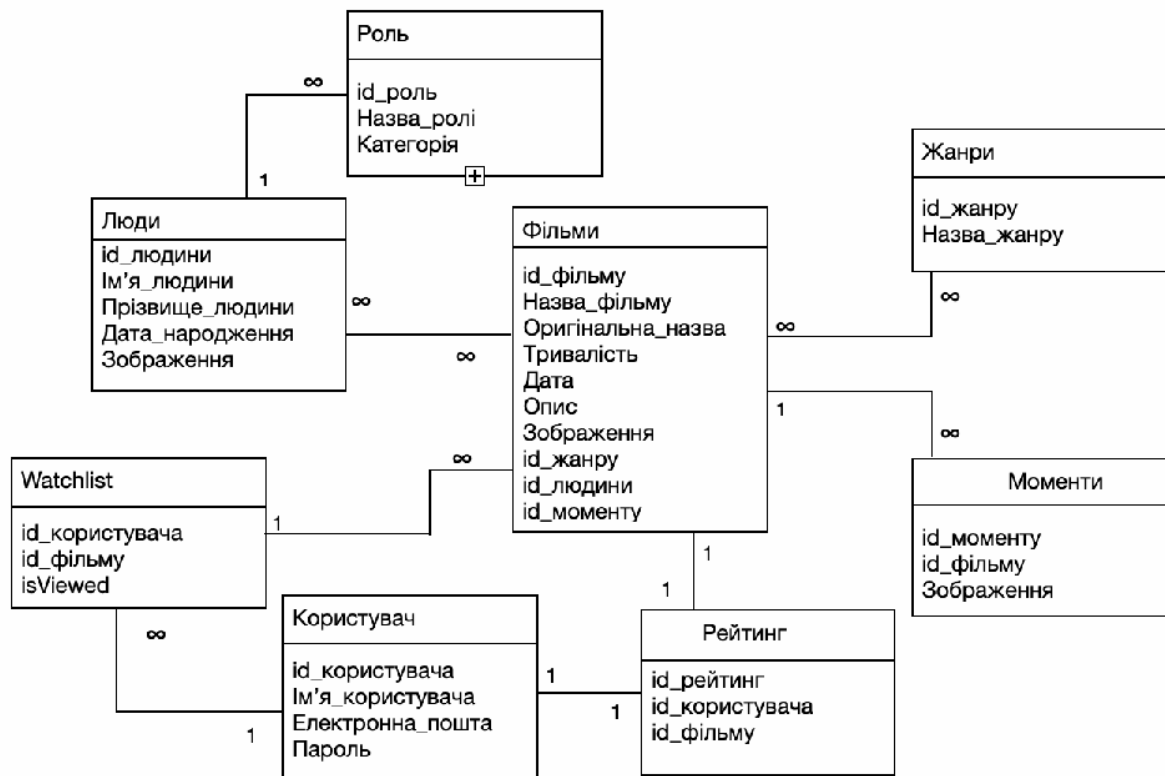


Рисунок 3.3 – Схема бази даних програмного забезпечення для перегляду інформації про фільми

В основі лежить таблиця Користувач, де зберігаються всі необхідні дані про людей, які взаємодіють із сервісом: від базових контактних відомостей до унікальних ідентифікаторів для входу. Ця таблиця пов'язана з іншими об'єктами, що дозволяє персоналізувати контент і зберігати історію взаємодії кожного відвідувача.

Важливе місце у базі посідає таблиця Фільми, де зберігаються всі відомості про кінострічки — назви, описи, дати випуску, обкладинки та інші атрибути, необхідні для повноцінного наповнення каталогу. Від неї тягнуться кілька важливих зв'язків. Один веде до таблиці Моменти, яка відповідає за фіксацію ключових сцен чи фрагментів — це можуть бути уривки, які користувач хоче переглянути окремо або позначити для повернення пізніше. Інший зв'язок іде до сутності Люди, де зберігаються дані про осіб, які брали участь у створенні фільмів: акторів, режисерів, сценаристів та інших учасників знімального процесу.

Від таблиці Люди прокладено ще один зв'язок до таблиці Ролі, що дозволяє деталізувати, хто і яку функцію виконував у межах конкретного фільму. Така структура дає змогу швидко визначати, які саме ролі виконували ті чи інші актори або хто займався постановкою картини.

Також база даних включає таблицю Жанри, яка систематизує фільми за тематикою, допомагаючи користувачам швидко фільтрувати контент відповідно до їхніх уподобань. Для зручності особистих добірок передбачено таблицю WatchList, де фіксується перелік стрічок, які користувачі планують переглянути. Це створює персоналізований досвід і полегшує повернення до вподобаного контенту.

Не менш важливим елементом залишається таблиця Рейтинг, у якій зберігаються оцінки, що їх користувачі виставляють фільмам. Ця інформація формує середню оцінку стрічок і впливає на їхню популярність у каталозі.

Уся ця система зв'язків формує єдину узгоджену модель даних, що дозволяє швидко отримувати всі потрібні відомості для відображення на вебсторінках, забезпечуючи користувачеві зручний доступ до структурованої й повної інформації про світ кіно.

3.4 Проєктування інтерфейсу програмного забезпечення для перегляду інформації про фільми

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для перегляду інформації про фільми виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для перегляду інформації про фільми враховані функціональні вимоги до програми:

- підтримка відображення блоку з актуальними фільмами, які вийшли у показ останнім часом;
- підтримка можливості відображення списку фільмів, які є у тренді користувачів вебресурсу;

- можливість відображення переліку фільмів, інформація про які зберігається у базі застосунку;
- можливість відображення детальної інформації про фільм, зокрема, його опис, рік створення, жанр, акторський склад та ін.;
- підтримка можливості відображення сторінки Watchlist зі списком фільмів для подальшого перегляду.

Макет сторінки про фільми програмного забезпечення для перегляду інформації про фільми наведено на рис. 3.4.

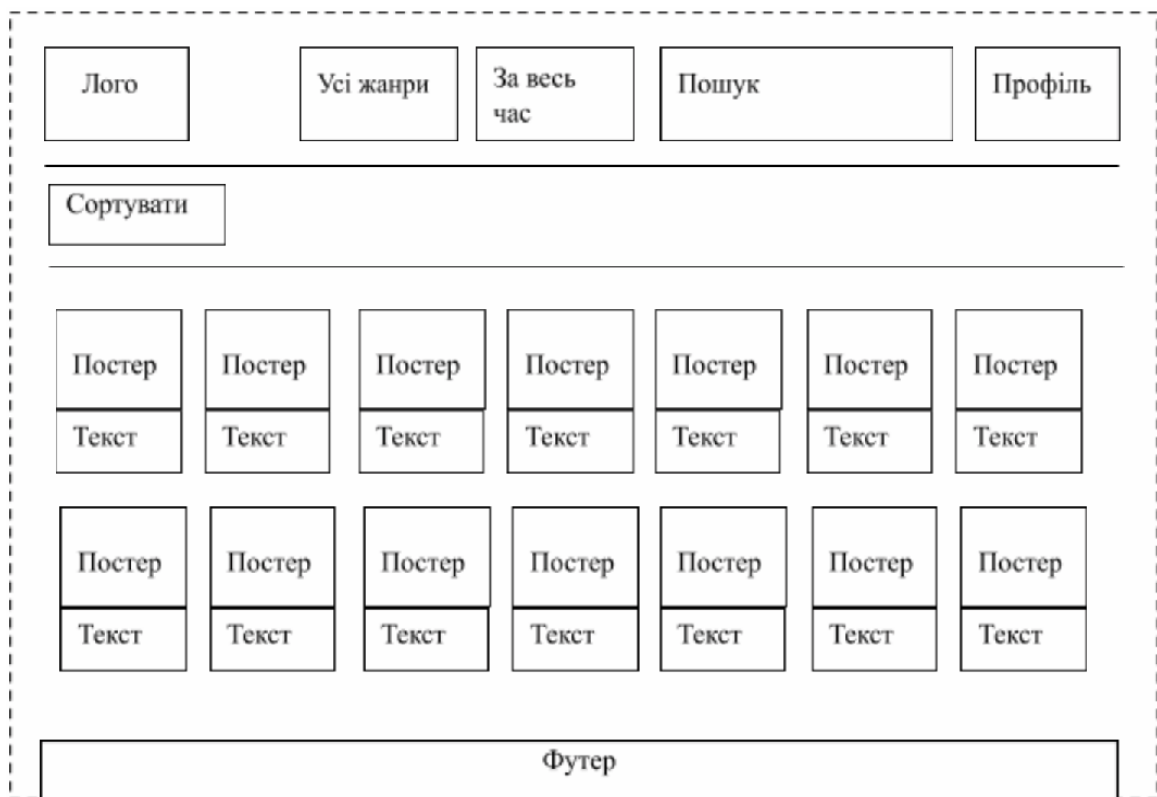


Рисунок 3.4 – Макет сторінки про фільми програмного забезпечення для перегляду інформації про фільми

Макет сторінки Checklist програмного забезпечення для перегляду інформації про фільми наведено на рис. 3.5.

Макет сторінки відображення інформації про фільм наведено на рис. 3.6.

Лого	Усі жанри	За весь час	Пошук	Профіль
Кількість	Сортувати			
Постер	Інформація про фільм			X
Постер	Інформація про фільм			X
Футер				

Рисунок 3.5 – Макет сторінки Checklist програмного забезпечення для перегляду інформації про фільми

Лого	Усі жанри	За весь час	Пошук	Профіль
Постер		Інформація		
Моменти кіно				
Зображення	Зображення	Зображення	Зображення	Зображення
Акторський склад				
Фото	Фото	Фото	Фото	Фото
Текст	Текст	Текст	Текст	Текст
Режисер	Текст			
Схожі фільми				
Фото	Фото	Фото		
Текст	Текст	Текст		
Футер				

Рисунок 3.6 – Макет сторінки відображення інформації про фільм

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для перегляду інформації про фільми.

Запропоновано структуру програмного забезпечення для перегляду інформації про фільми. Визначено, що основними програмного забезпечення для перегляду інформації про фільми є Користувач, Фільм, Каталог та інші.

Побудовано схему функціонування програмного забезпечення для перегляду інформації про фільми, яка відображає взаємодію між основними складовими системи, на ній показано, як кожен елемент виконує свою роль у загальному процесі обробки запитів і передачі даних.

Розроблено базу даних програмного забезпечення для перегляду інформації про фільми, яка відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Визначено, що основними сутностями бази даних обрано такі: Користувач, Фільми, Рейтинг, WatchList, Жанри та інші. Структура бази даних, що обслуговує програмне забезпечення для перегляду детальної інформації про фільми, відображає взаємозв'язки між основними таблицями та сутностями, що формують ядро всієї системи. На схемі відображено, як окремі елементи пов'язуються між собою для забезпечення повноти й точності інформації, яку платформа надає своїм користувачам.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для перегляду інформації про фільми.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для перегляду інформації про фільми. Програма написана на мові JavaScript, яка є однією з найбільш універсальних мов для створення інтерфейсів, що реагують у реальному часі та забезпечують швидкий відгук на дії користувача. В якості фреймворку для створення програмного забезпечення для перегляду інформації про фільми обрано React.js, який дозволяє використовувати мову програмування JavaScript як на клієнтській, так і на серверній стороні, що спрощує взаємодію між компонентами програмної системи, пришвидшує розробку та полегшує підтримку проєкту.

Програмне забезпечення для перегляду інформації про фільми забезпечує виконання таких функцій:

- підтримка відображення блоку з актуальними фільмами, які вийшли у показ останнім часом;
- підтримка можливості відображення списку фільмів, які є у тренді користувачів вебресурсу;
- можливість відображення переліку фільмів, інформація про які зберігається у базі застосунку;
- можливість відображення детальної інформації про фільм, зокрема, його опис, рік створення, жанр, акторський склад та ін.;
- підтримка можливості відображення сторінки Watchlist зі списком фільмів для подальшого перегляду.

Для роботи програмного забезпечення для перегляду інформації про фільми потрібні такі технічні та програмні ресурси: комп'ютер або сервер з процесором із тактовою частотою не нижче 2,4 ГГц, що гарантує достатню швидкодію під час обробки одночасних запитів користувачів і виконання серверних операцій. Рекомендується використовувати багатоядерні

процесори, щоб забезпечити стабільність роботи у випадку високих навантажень. Оперативна пам'ять повинна становити не менше ніж 4 ГБ для базових конфігурацій або 8 ГБ і більше для стабільної роботи при активному використанні платформи великою кількістю відвідувачів одночасно. З боку програмного середовища потрібна сучасна операційна система, наприклад Windows 10 чи новіша версія, одна з актуальних збірок Linux або macOS. Для функціонування серверної частини необхідно мати встановлене середовище Node.js відповідної версії, а також налаштовану систему керування базами даних. Для повноцінної роботи користувацького інтерфейсу у браузері потрібен сучасний вебоглядач з підтримкою JavaScript і React JS, що забезпечує коректне відображення сторінок і швидке оновлення динамічного контенту без затримок.

Функціональні характеристики розробленого програмного забезпечення для перегляду інформації про фільми наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для перегляду інформації про фільми наведено у додатку Б.

4.2 Характеристики програми для перегляду інформації про фільми

Програмне забезпечення для перегляду інформації про фільми вирізняється доступністю через звичайний вебінтерфейс, що не потребує встановлення окремих клієнтів чи складного налаштування. Інтерфейс побудований так, щоб користувач міг легко знаходити потрібні фільми, користуватися фільтрами за жанрами або рейтингами й переглядати детальні описи, фото, трейлери та рецензії. Завдяки використанню сучасних технологій сторінки швидко оновлюються, реагують на дії без перезавантаження, що підвищує зручність навігації.

Платформа здатна обробляти запити багатьох користувачів одночасно без помітних затримок. Завдяки гнучкій структурі бази даних легко

зберігається велика кількість інформації про фільми, акторів, режисерів і супутні матеріали. Система забезпечує можливість інтеграції з різними сторонніми сервісами, що дозволяє додавати новий контент або автоматично оновлювати вже наявні відомості.

Файлову структуру клієнтської частини програми наведено на рис. 4.1.

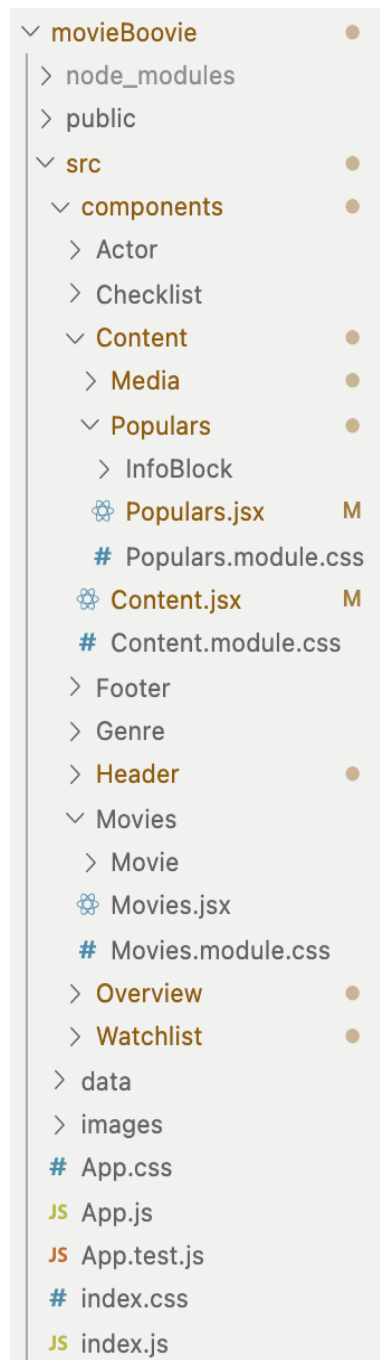


Рисунок 4.1 – Файлова структура клієнтської частини програмного забезпечення для перегляду інформації про фільми

Файлову структуру серверної частини програми наведено на рис. 4.2.

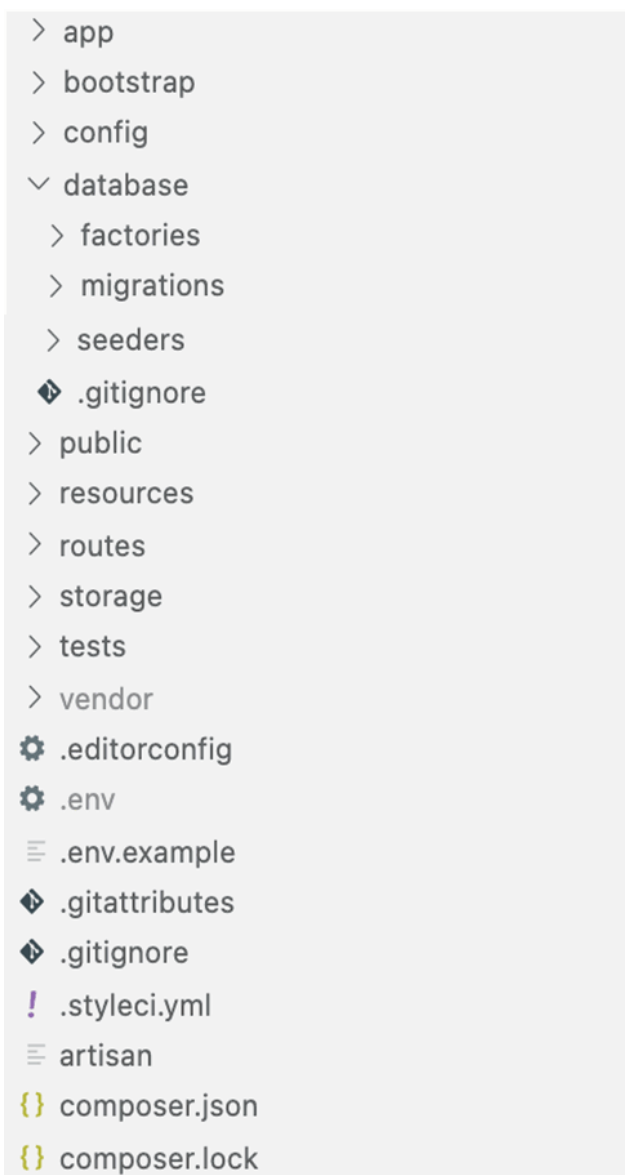


Рисунок 4.2 – Файлова структура серверної частини програмного забезпечення для перегляду інформації про фільми

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для перегляду інформації про фільми на серверній частині необхідно запустити відповідний файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для перегляду інформації про фільми є результати дій користувача, зокрема, вибір фільму для перегляду інформації про нього.

Вихідними даними програмного забезпечення для перегляду інформації про фільми є відображені вебсторінки порталу, зокрема, вебсторінки з інформацією про обрані користувачем фільми.

4.3.3 Повідомлення

Користувач програмного забезпечення для перегляду інформації про фільми може отримати такі повідомлення: повідомлення про відсутність інформації про обраний фільм.

4.4 Виконання програмного забезпечення для перегляду інформації про фільми

Після запуску програмного забезпечення для перегляду інформації про фільми користувачу відображається головна сторінка (рис. 4.3). На головній сторінці відображено актуальні фільми, які вийшли у показ останнім часом, а також фільми, які є у тренді користувачів вебресурсу. Користувачу надається можливість навігації по вебзастосунку за допомогою головного меню, зокрема, можливість переходу до сторінок, де відображаються усі жанри фільмів та список фільмів для перегляду (Watchlist).

Знизу головної сторінки користувач може побачити перелік фільмів, які будуть доступними для перегляду у найближчий час (рис. 4.4).

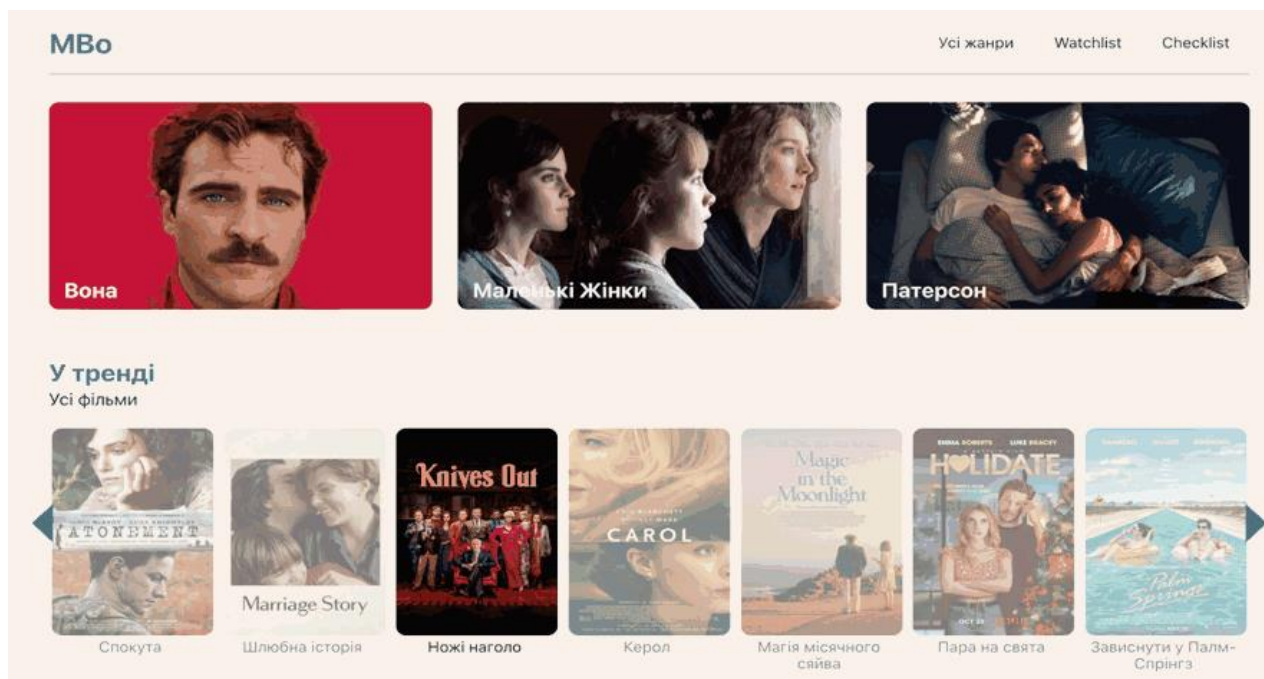


Рисунок 4.3 – Інтерфейс головної сторінки програмного забезпечення для перегляду інформації про фільми



Рисунок 4.4 – Інтерфейс нижньої частини головної сторінки програмного забезпечення для перегляду інформації про фільми

При виборі пункту меню Усі фільми відображається сторінка з фільмами, де наведено зображення фільмів та їх назви (рис. 4.5).

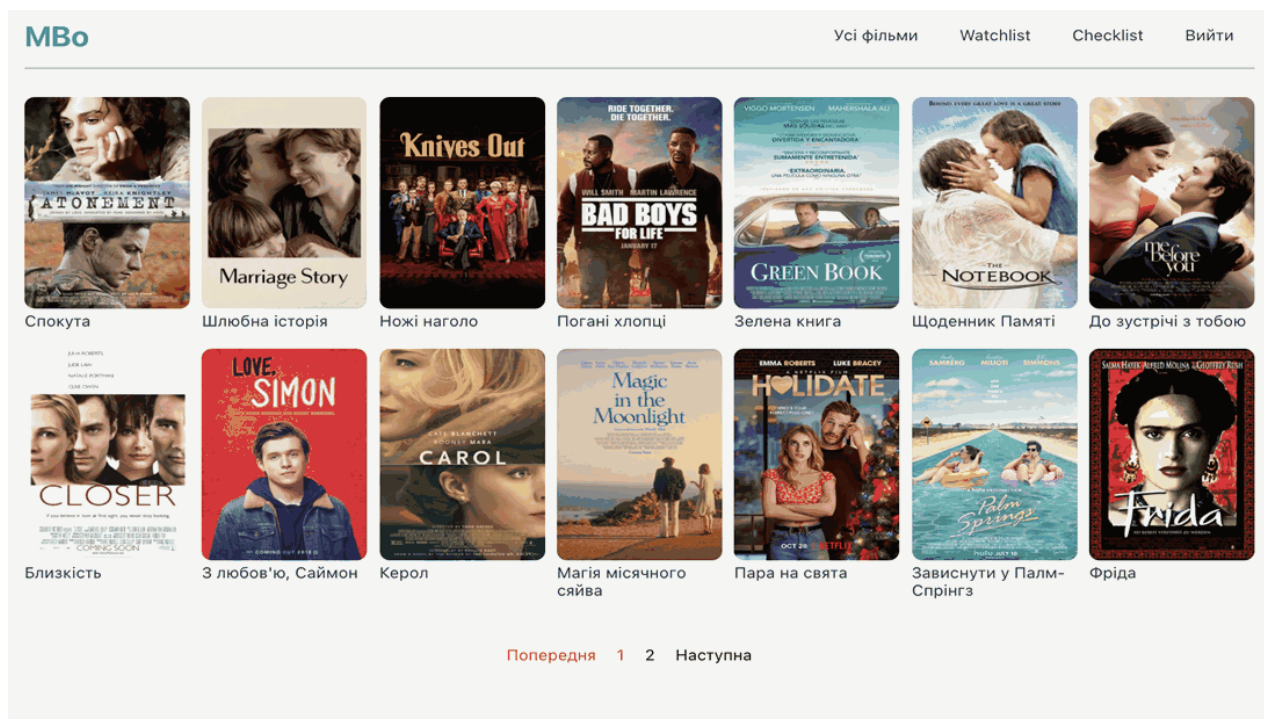


Рисунок 4.5 – Сторінка з жанрами фільмів

При виборі фільму відображається сторінка (рис. 4.6), де наведено детальну інформацію про фільм, зокрема, його опис, рік створення, жанр, основні зображення, а також акторський склад фільму.

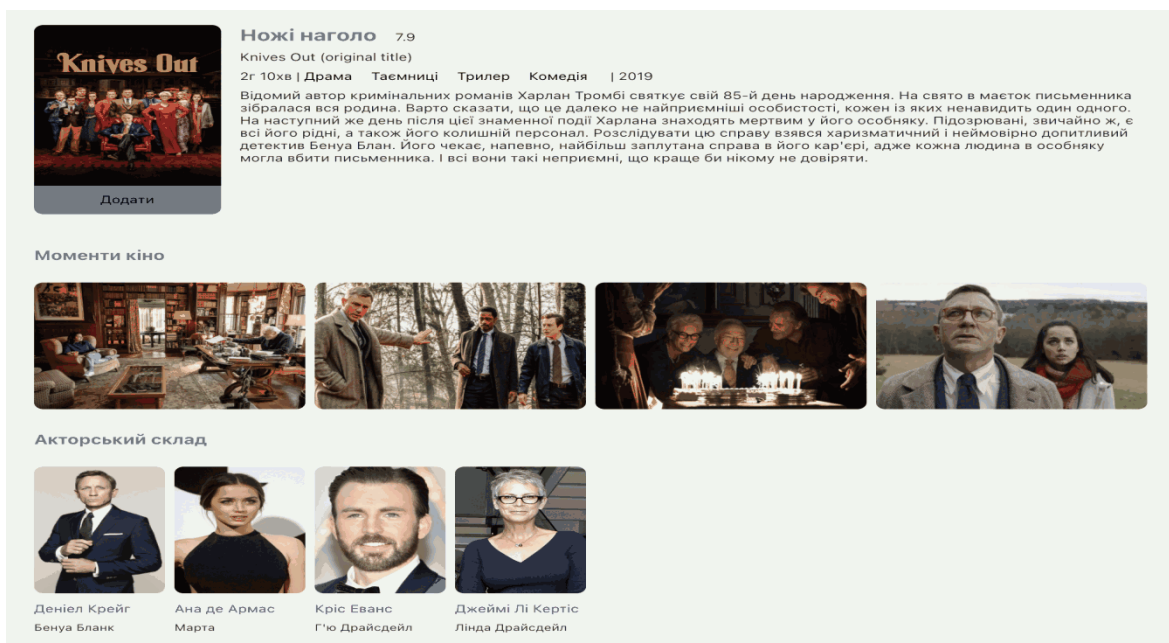


Рисунок 4.6 – Сторінка з інформацією про фільм

При виборі пункту меню Watchlist відображається сторінка (рис. 4.7) зі списком фільмів для перегляду.

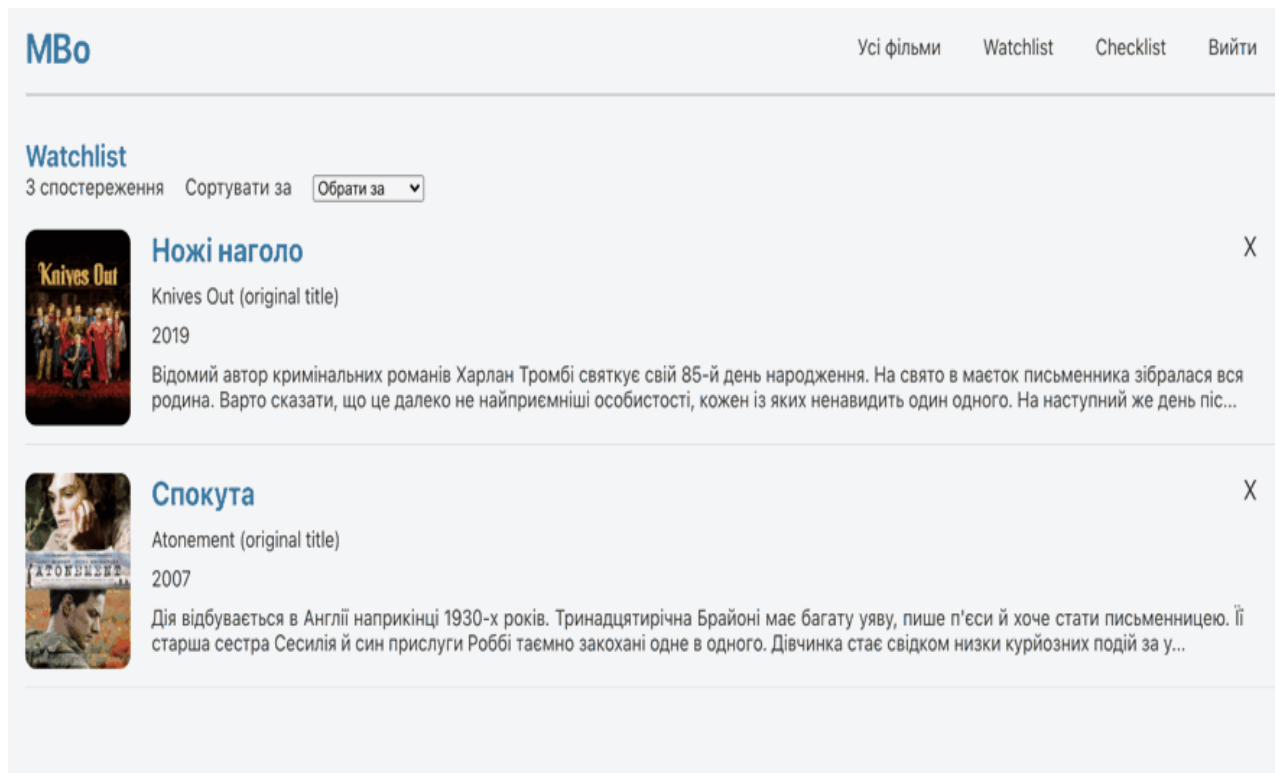


Рисунок 4.7 – Сторінка списку фільмів для перегляду (Watchlist)

4.5 Тестування програмного забезпечення для перегляду інформації про фільми

Виконано тестування програмного забезпечення для перегляду інформації про фільми.

Виявлено, що програма для перегляду інформації про фільми функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу організації харчування домашніх тварин.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою організації харчування домашніх тварин.

4.6 Висновки за розділом 4

Описано програмне забезпечення для перегляду інформації про фільми. Виконано тестування розробленого програмного забезпечення для перегляду інформації про фільми. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою організації харчування домашніх тварин.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для перегляду інформації про фільми.

Визначено, що програмне забезпечення для перегляду інформації про фільми являє собою спеціалізований ресурс або платформу, що збирає, обробляє й подає користувачам різноманітні відомості про світ кіно. За допомогою таких програмних сервісів користувач отримує доступ до бази даних, де зручно знайти опис сюжету, дізнатися більше про акторський склад, режисера, жанр і всі супутні деталі, які можуть зацікавити перед переглядом. Такі програмні інструменти зазвичай стають надійним джерелом інформації, що дозволяє глибше зануритися у кінематографічну культуру та робити більш усвідомлений вибір, на що варто витратити свій час.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для перегляду інформації про фільми. Проте деякі програмні засоби для перегляду інформації про фільми характеризуються перевантаженістю інтерфейсу надлишковою рекламою чи додатковими модулями, що ускладнює пошук потрібних даних і відволікає користувача від основної мети — швидкого доступу до інформації. Окрім того, деякі програми для перегляду інформації про фільми не забезпечують достатнього рівня достовірності контенту: інформація про фільми може бути застарілою або неперевіреною, особливо якщо джерела не модеруються належним чином. Крім того, деякі програми збирають надмірну кількість персональних даних для формування рекомендацій чи реклами, що викликає занепокоєння з точки зору конфіденційності та безпеки користувачів. У результаті частина аудиторії може відмовитися від використання таких сервісів або шукати альтернативні рішення. Тому актуальною є розробка програмного забезпечення для перегляду інформації про фільми.

Сформульовано функціональні вимоги до програмного забезпечення для перегляду інформації про фільми.

Для реалізації програмного забезпечення для перегляду інформації про фільми обрано мову програмування JavaScript, яка є однією з найбільш універсальних мов для створення інтерфейсів, що реагують у реальному часі та забезпечують швидкий відгук на дії користувача. Крім того, JavaScript має потужну спільноту, яка постійно створює і підтримує велике різноманіття плагінів, модулів і готових компонентів, це відкриває широкі можливості для оптимізації витрат часу та ресурсів у процесі розробки.

Для створення програмного забезпечення для перегляду інформації про фільми обрано фреймворк React.js, який дозволяє використовувати мову програмування JavaScript як на клієнтській, так і на серверній стороні, що спрощує взаємодію між компонентами програмної системи, пришвидшує розробку та полегшує підтримку проєкту. Завдяки цьому зменшується кількість помилок, які часто виникають під час інтеграції різних технологій. Крім цього, React.js вирізняється високою продуктивністю при обробці великої кількості одночасних запитів, що є критично важливим для сервісів, де тисячі користувачів можуть шукати, фільтрувати та переглядати інформацію одночасно.

Новизна роботи полягає у тому, що створено модель інтерфейсу користувача програмного забезпечення для перегляду інформації про фільми, яка подана у вигляді прототипу вебсторінки, та дозволяє розроблювати програмне забезпечення для перегляду інформації про фільми.

Практична цінність полягає у розробці програмного забезпечення для перегляду інформації про фільми.

Запропоновано структуру програмного забезпечення для перегляду інформації про фільми. Визначено, що основними програмного забезпечення для перегляду інформації про фільми є Користувач, Фільм, Каталог та інші.

Побудовано схему функціонування програмного забезпечення для перегляду інформації про фільми, яка відображає взаємодію між основними

складовими системи, на ній показано, як кожен елемент виконує свою роль у загальному процесі обробки запитів і передачі даних.

Розроблено базу даних програмного забезпечення для перегляду інформації про фільми, яка відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Визначено, що основними сутностями бази даних обрано такі: Користувач, Фільми, Рейтинг, WatchList, Жанри та інші. Структура бази даних, що обслуговує програмне забезпечення для перегляду детальної інформації про фільми, відображає взаємозв'язки між основними таблицями та сутностями, що формують ядро всієї системи. На схемі відображено, як окремі елементи пов'язуються між собою для забезпечення повноти й точності інформації, яку платформа надає своїм користувачам.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для перегляду інформації про фільми.

Виконано тестування розробленого програмного забезпечення для перегляду інформації про фільми. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою організації харчування домашніх тварин.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Building a Movie website [Electronic resource]. – Access mode: <https://medium.com/@feven.25yi/building-a-movie-website-5ab5e34d13de>.
2. The Ultimate Movie Review Website Guide [Electronic resource]. – Access mode: <https://www.strikingly.com/blog/posts/ultimate-movie-review-website-guide>.
3. What is the Best Website for Movie Information [Electronic resource]. – Access mode: <https://movrq.com/what-is-the-best-website-for-movie-information/>.
4. How To Create A Movie Streaming App Or Website [Electronic resource]. – Access mode: <https://www.contus.com/blog/build-movie-streaming-website-like-netflix-hulu/>.
5. A Guide to Create a Movie Streaming Platform [Electronic resource]. – Access mode: <https://www.muvi.com/blogs/create-a-movie-streaming-platform/>.
6. Best Sites for Rating Movies [Electronic resource]. – Access mode: <https://filmlifestyle.com/best-sites-for-rating-movies/>.
7. The Ultimate Guide to Free Movie Websites [Electronic resource]. – Access mode: <https://techbeamers.com/best-websites-to-watch-movies-free-online/>.
8. Top 10 Best Movie Review Sites [Electronic resource]. – Access mode: <https://www.thetoptens.com/movies/movie-reviewing-sites/>.
9. Best Movie Review Websites [Electronic resource]. – Access mode: <https://blog.favoree.io/articles/best-movie-review-websites/>.
10. Best Movie Review Sites [Electronic resource]. – Access mode: <https://fictionhorizon.com/best-movie-review-sites/>.
11. FilmAffinity [Electronic resource]. – Access mode: <https://www.filmaffinity.com/en/main.html>.
12. Metacritic [Electronic resource]. – Access mode: <https://www.metacritic.com/>.
13. IndieWire [Electronic resource]. – Access mode: <https://www.indiewire.com/>.

14. Namaste JavaScript [Electronic resource]. – Access mode: <https://namastedev.com/learn/namaste-javascript>.

15. What is JavaScript [Electronic resource]. – Access mode: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/What_is_JavaScript.

16. React Introduction [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/reactjs/reactjs-introduction/>.

17. React [Electronic resource]. – Access mode: <https://react.dev/>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для перегляду інформації про фільми.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація платформи для перегляду інформації про фільми», затверджене наказом Національного університету «Запорізька політехніка» № 447 від 30 вересня 2025 р.

A.2 Призначення розробки

Програмний продукт призначений для перегляду інформації про фільми.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для перегляду інформації про фільми забезпечує виконання таких функцій:

- підтримка відображення блоку з актуальними фільмами, які вийшли у показ останнім часом;
- підтримка можливості відображення списку фільмів, які є у тренді користувачів вебресурсу;
- можливість відображення переліку фільмів, інформація про які зберігається у базі застосунку;
- можливість відображення детальної інформації про фільм, зокрема, його опис, рік створення, жанр, акторський склад та ін.;

– підтримка можливості відображення сторінки Watchlist зі списком фільмів для подальшого перегляду.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для перегляду інформації про фільми повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для перегляду інформації про фільми повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для перегляду інформації про фільми необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи програмного забезпечення для перегляду інформації про фільми потрібні такі технічні та програмні ресурси: комп'ютер або сервер з процесором із тактовою частотою не нижче 2,4 ГГц, що гарантує достатню швидкодію під час обробки одночасних запитів користувачів і виконання серверних операцій. Рекомендується використовувати багатоядерні процесори, щоб забезпечити стабільність роботи у випадку високих навантажень. Оперативна пам'ять повинна становити не менше ніж 4 ГБ для

базових конфігурацій або 8 ГБ і більше для стабільної роботи при активному використанні платформи великою кількістю відвідувачів одночасно. З боку програмного середовища потрібна сучасна операційна система, наприклад Windows 10 чи новіша версія, одна з актуальних збірок Linux або macOS. Для функціонування серверної частини необхідно мати встановлене середовище Node.js відповідної версії, а також налаштовану систему керування базами даних. Для повноцінної роботи користувацького інтерфейсу у браузері потрібен сучасний вебглядач з підтримкою JavaScript і React JS, що забезпечує коректне відображення сторінок і швидке оновлення динамічного контенту без затримок.

A.3.6 Вимоги до маркування і пакування

Програма для перегляду інформації про фільми може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для перегляду інформації про фільми».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для перегляду інформації про фільми є результати дій користувача, зокрема, вибір фільму для перегляду інформації про нього.

Вихідними даними програмного забезпечення для перегляду інформації про фільми є відображені вебсторінки порталу, зокрема, вебсторінки з інформацією про обрані користувачем фільми.

ДОДАТОК Б
Фрагмент тексту програми

```

function crtFlm(id, nm, drctr, yr, gnrs, rttnng) {
  let obj = {
    id: id,
    nm: nm,
    drctr: drctr,
    yr: yr,
    gnrs: gnrs,
    rttnng: rttnng
  };
  flmLst.push(obj);
}

for (let i = 0; i < 100; i++) {
  crtFlm(
    i,
    "FLM_" + i,
    "DRCTR_" + i,
    "YR_" + (2000 + i % 20),
    ["CTN", "DRM", "THR"].slice(0, i % 3 + 1),
    Math.floor(Math.random() * 10) + 1
  );
}

function rndrLst() {
  lst.innerHTML = "";
  let fnd = flmLst.filter(x =>
x.nm.toLowerCase().includes(fltr.value.toLowerCase()));
  fnd.forEach(f => {
    let cd = document.createElement("div");
    cd.style.border = "1px solid #ddd";
    cd.style.borderRadius = "4px";
    cd.style.width = "200px";
    cd.style.margin = "10px";
    cd.style.padding = "10px";
    cd.style.boxShadow = "0 2px 5px rgba(0,0,0,0.1)";
    let nm = document.createElement("h3");
    nm.innerText = f.nm;
    cd.appendChild(nm);
    let dr = document.createElement("p");
    dr.innerText = "DRCTR: " + f.drctr;
    cd.appendChild(dr);
    let yr = document.createElement("p");
    yr.innerText = "YR: " + f.yr;
    cd.appendChild(yr);
    let gn = document.createElement("p");
    gn.innerText = "GNRS: " + f.gnrs.join(", ");
    cd.appendChild(gn);
  });
}

```

```

        let rt = document.createElement("p");
        rt.innerText = "RTTNG: " + f.rttng;
        cd.appendChild(rt);
        lst.appendChild(cd);
    });
}

fltr.addEventListener("input", () => {
    rndrLst();
});

rndrLst();

function flmMng() {
    for (let i = 0; i < 100; i++) {
        let id = i;
        let nm = "FLM_MNG_" + i;
        let dr = "DRCTR_MNG_" + i;
        let yr = "YR_MNG_" + i;
        let gnrs = ["CTN", "THR"];
        let rttng = Math.floor(Math.random() * 10) + 1;
        flmLst.push({
            id: id,
            nm: nm,
            drctr: dr,
            yr: yr,
            gnrs: gnrs,
            rttng: rttng
        });
    }
}

function flmUpdt() {
    flmLst.forEach(f => {
        f.rttng = Math.floor(Math.random() * 10) + 1;
    });
}

function flmRmv() {
    flmLst = flmLst.filter(f => f.rttng > 5);
}

function flmAddX() {
    for (let i = 0; i < 100; i++) {
        flmLst.push({
            id: i + 200,
            nm: "FLM_X_" + i,
            drctr: "DRCTR_X_" + i,

```

```

        yr: "YR_X_" + i,
        gnrs: ["CTN"],
        rttng: Math.floor(Math.random() * 10) + 1
    });
}
}

function flmFlt() {
    return flmLst.filter(f => f.rttng > 5);
}

function flmSrt() {
    return flmLst.sort((a, b) => b.rttng - a.rttng);
}

function dbCnct() {
    let db = {};
    db.st = false;
    db.ctn = [];
    db.cn = function () {
        db.st = true;
    };
    db.ds = function () {
        db.st = false;
    };
    return db;
}

let dbHndl = dbCnct();
dbHndl.cn();

function dbSv(flm) {
    if (!dbHndl.st) return false;
    dbHndl.ctn.push(flm);
    return true;
}

function dbLd() {
    if (!dbHndl.st) return [];
    return dbHndl.ctn;
}

function dbCln() {
    if (!dbHndl.st) return;
    dbHndl.ctn = [];
}

function dbFndByNm(nm) {
    if (!dbHndl.st) return [];

```

```

        return dbHndl.ctn.filter(f
f.nm.toLowerCase().includes(nm.toLowerCase()));
    }

    function dbRm(id) {
        if (!dbHndl.st) return false;
        dbHndl.ctn = dbHndl.ctn.filter(f => f.id !== id);
        return true;
    }

    function dbUpdt(id, dt) {
        if (!dbHndl.st) return false;
        let idx = dbHndl.ctn.findIndex(f => f.id === id);
        if (idx === -1) return false;
        dbHndl.ctn[idx] = {...dbHndl.ctn[idx], ...dt};
        return true;
    }

    function dbCshWr() {
        let dt = JSON.stringify(dbHndl.ctn);
        localStorage.setItem("flm_csh", dt);
    }

    function dbCshRd() {
        let dt = localStorage.getItem("flm_csh");
        if (!dt) return [];
        return JSON.parse(dt);
    }

    function dbSync() {
        let sv = dbLd();
        dbCshWr();
        return sv;
    }

    function dbBkp() {
        let dt = dbLd();
        let ts = new Date().getTime();
        localStorage.setItem("flm_bkp_" + ts, JSON.stringify(dt));
    }

    function dbRstr(ts) {
        let dt = localStorage.getItem("flm_bkp_" + ts);
        if (!dt) return [];
        dbHndl.ctn = JSON.parse(dt);
    }

    function dbExp() {
        let dt = JSON.stringify(dbHndl.ctn);
=>

```

```

        let ln = "data:text/json;charset=utf-8," +
encodeURIComponent(dt);
        let lnk = document.createElement("a");
        lnk.setAttribute("href", ln);
        lnk.setAttribute("download", "flm_exp.json");
        document.body.appendChild(lnk);
        lnk.click();
        document.body.removeChild(lnk);
    }

function dbImp(dt) {
    try {
        let js = JSON.parse(dt);
        dbHndl.ctn = js;
    } catch (e) {
        return false;
    }
    return true;
}

function dbMckRqst(nm, cb) {
    setTimeout(() => {
        let rs = dbFndByNm(nm);
        cb(rs);
    }, Math.random() * 1000 + 500);
}

function dbAuth(k) {
    return k.split("").reverse().join("") === "kyd";
}

function dbLck() {
    dbHndl.st = false;
}

function dbUnlck(k) {
    if (dbAuth(k)) {
        dbHndl.st = true;
    }
}

function dbMnt() {
    if (dbHndl.ctn.length > 1000) {
        dbHndl.ctn = dbHndl.ctn.slice(0, 1000);
    }
}

function dbArchv() {
    let dt = dbLd();

```

```

    localStorage.setItem("flm_archv", JSON.stringify(dt));
    dbCln();
}

function dbDrp() {
    dbCln();
    localStorage.removeItem("flm_csh");
    localStorage.removeItem("flm_archv");
}

function dbChkIn(id) {
    return dbHndl.ctn.some(f => f.id === id);
}

function dbLngstNm() {
    return dbHndl.ctn.reduce((a, b) => a.nm.length > b.nm.length
? a : b, {}).nm || "";
}

function dbStats() {
    return {
        ttl: dbHndl.ctn.length,
        maxRt: Math.max(...dbHndl.ctn.map(f => f.rttng)),
        minRt: Math.min(...dbHndl.ctn.map(f => f.rttng))
    };
}

function svrStrt() {
    svr.dt = [];
    svr.usr = [{nm: "dmn", ps: "s crt"}];
    svr.tk = [];
}

function svrAuth(nm, ps) {
    let fnd = svr.usr.find(u => u.nm === nm && u.ps === ps);
    if (fnd) {
        let tkn = Math.random().toString(36).substr(2);
        svr.tk.push(tkn);
        return tkn;
    }
    return null;
}

function svrChkTkn(tkn) {
    return svr.tk.includes(tkn);
}

function svrRqstTkn(tkn, act, dt) {
    if (!svrChkTkn(tkn)) return {err: "nvd"};

```

```

    if (act === "lst") return svr.dt;
    if (act === "add") {
        svr.dt.push(dt);
        return {ok: true};
    }
    if (act === "rmv") {
        svr.dt = svr.dt.filter(f => f.id !== dt.id);
        return {ok: true};
    }
    return {err: "nkn"};
}

function apiRqst(act, dt, cb) {
    setTimeout(() => {
        let rsp = {};
        if (act === "get") rsp = dbLd();
        if (act === "put") {
            let s = dbSv(dt);
            rsp = s ? {ok: true} : {err: "fl"};
        }
        if (act === "del") {
            let r = dbRm(dt.id);
            rsp = r ? {ok: true} : {err: "fl"};
        }
        cb(rsp);
    }, Math.random() * 500 + 200);
}

function apiLogin(nm, ps, cb) {
    setTimeout(() => {
        let tkn = svrAuth(nm, ps);
        cb(tkn ? {tkn: tkn} : {err: "fl"});
    }, Math.random() * 500 + 200);
}

function apiTknRqst(tkn, act, dt, cb) {
    setTimeout(() => {
        let rsp = svrRqstTkn(tkn, act, dt);
        cb(rsp);
    }, Math.random() * 500 + 200);
}

function apiPng(cb) {
    setTimeout(() => {
        cb({st: "ok"});
    }, Math.random() * 200 + 100);
}

function apiUpdtTkn(tkn, dt, cb) {

```

```

    setTimeout(() => {
        if (!svrChkTkn(tkn)) return cb({err: "nvd"});
        svr.dt = svr.dt.map(f => f.id === dt.id ? {...f, ...dt} :
f);
        cb({ok: true});
    }, Math.random() * 500 + 200);
}

function apiSyncDb(cb) {
    setTimeout(() => {
        dbSync();
        cb({ok: true});
    }, Math.random() * 500 + 200);
}

function apiFltNm(tkn, nm, cb) {
    setTimeout(() => {
        if (!svrChkTkn(tkn)) return cb({err: "nvd"});
        let          rs          =          svr.dt.filter(f          =>
f.nm.toLowerCase().includes(nm.toLowerCase()));
        cb(rs);
    }, Math.random() * 500 + 200);
}

function apiAddUsr(nm, ps, cb) {
    setTimeout(() => {
        svr.usr.push({nm: nm, ps: ps});
        cb({ok: true});
    }, Math.random() * 300 + 200);
}

function apiChkUsr(nm, cb) {
    setTimeout(() => {
        let ex = svr.usr.some(u => u.nm === nm);
        cb({ex: ex});
    }, Math.random() * 300 + 200);
}

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дослідження та програмна реалізація платформи для перегляду інформації про фільми

Виконав: ст. гр. КНТ-124м

Микита ГОРОШКО

Керівник:
к.т.н., доцент

Степан СКРУПСЬКИЙ¹

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета дослідження

Об'єкт дослідження – процес розробки програмного забезпечення для перегляду інформації про фільми.

Предмет дослідження – програмні засоби для перегляду інформації про фільми.

Метою роботи є дослідження та розробка програмного забезпечення для перегляду інформації про фільми.

Рисунок В.2 – Слайд 2

Задачи роботи:

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для перегляду інформації про фільми;
- розробити модель інтерфейсу користувача програмного забезпечення для перегляду інформації про фільми;
- здійснити проектування програмного забезпечення для перегляду інформації про фільми;
- створити програмне забезпечення для перегляду інформації про фільми;
- виконати тестування розробленого програмного забезпечення для перегляду інформації про фільми.

3

Рисунок В.3 – Слайд 3

Дослідження програмного забезпечення для перегляду інформації про фільми

Критерій порівняння	FilmAffinity [11]	Metacritic [12]	IndieWire [13]
Персоналізовані рекомендації	+	+-	-
Складність інтерфейсу	+-	+	+-
Користувацькі оцінки	+	+	-
Професійні огляди	+-	+	+
Багатомовність	+	+	+-
Інтерв'ю та аналітика	-	+-	+
Соціальна взаємодія	+-	+-	-

4

Рисунок В.4 – Слайд 4

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	JavaScript	Python	Java
Підтримка роботи у веббраузері	+	—	—
Зручність створення інтерактивного інтерфейсу користувача	+	+—	+—
Велика кількість готових бібліотек для веброзробки	+	+	+—
Зручність для створення ПЗ для перегляду інформації про фільми	+	—	—
Швидкість прототипування	+	+	+—

5

Рисунок В.5 – Слайд 5

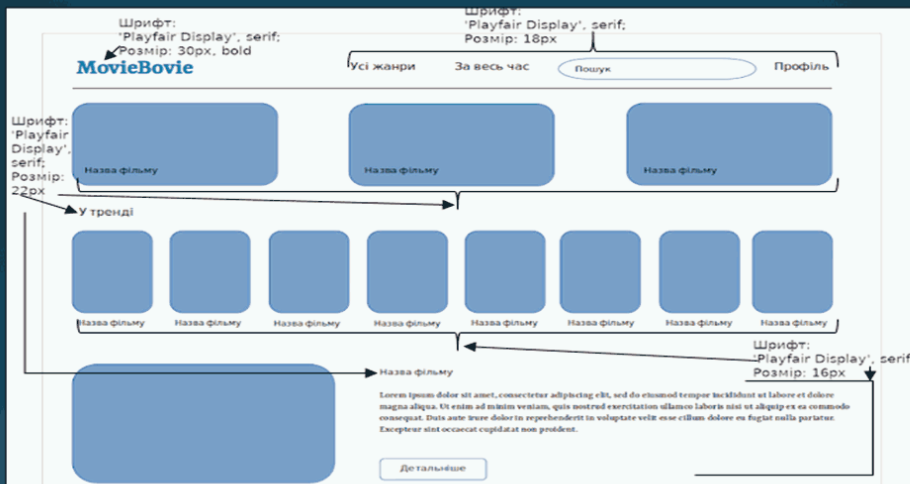
Вибір фреймворку

Критерій порівняння фреймворків	Фреймворки		
	React.js	Next.js	Express.jsp
Широта використання для серверної логіки	+	+—	+—
Гнучкість побудови архітектури	+	+—	+—
Велика кількість доступних модулів і пакетів	+	—	—
Продуктивність при обробці одночасних запитів	+	+—	+—
Гнучкість вибору фреймворків для клієнтської частини	+	+	+—

6

Рисунок В.6 – Слайд 6

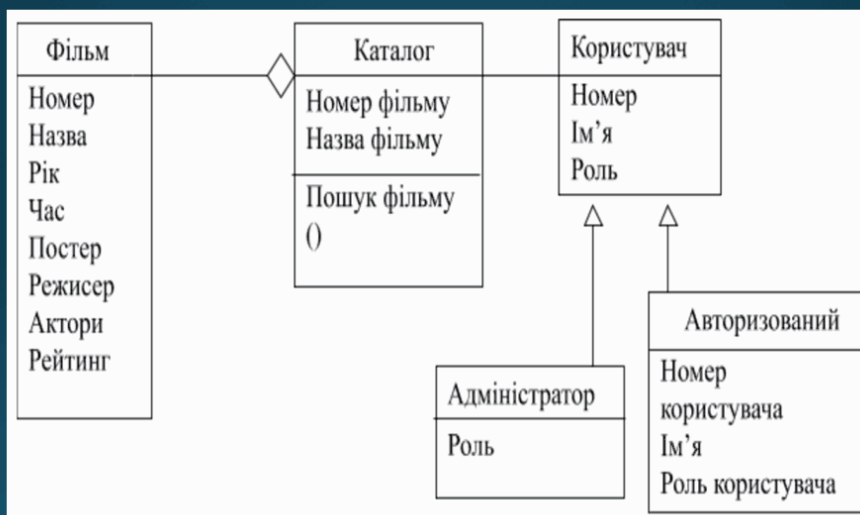
Модель прототипу головної сторінки ПЗ для перегляду інформації про фільми



7

Рисунок В.7 – Слайд 7

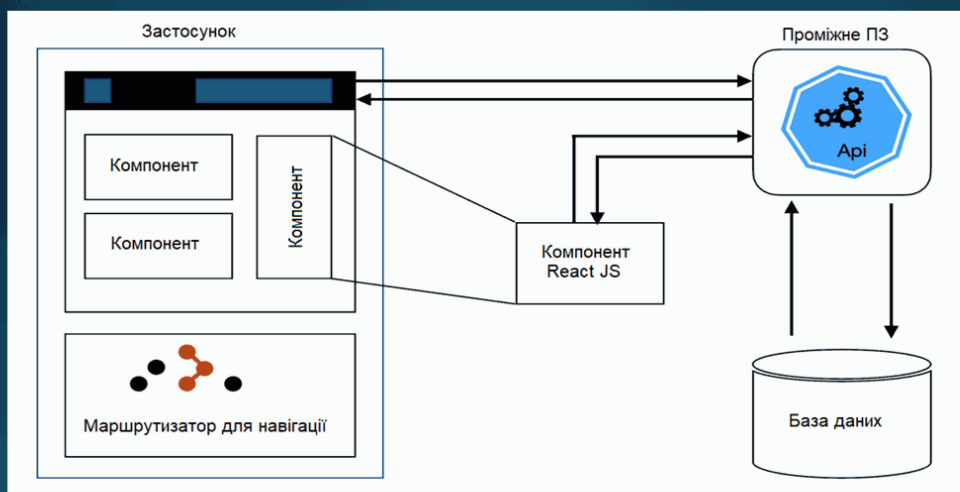
Діаграма класів програмного забезпечення для перегляду інформації про фільми



8

Рисунок В.8 – Слайд 8

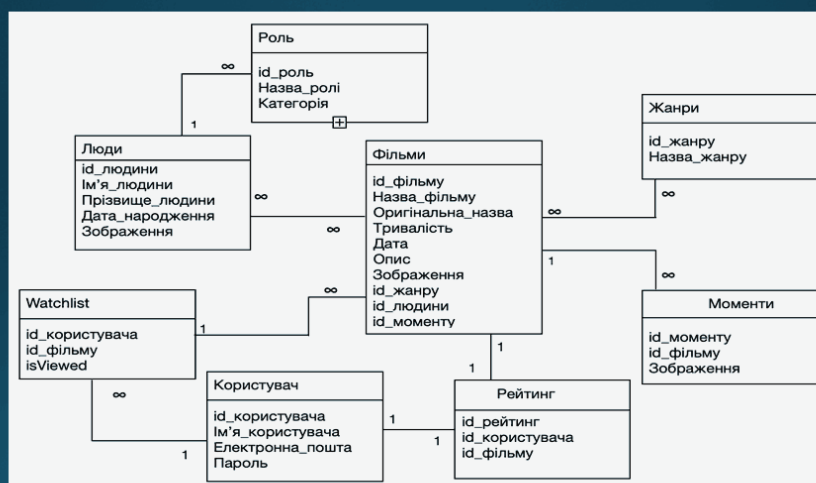
Схема функціонування ПЗ для перегляду інформації про фільми



9

Рисунок В.9 – Слайд 9

Схема бази даних програмного забезпечення для перегляду інформації про фільми



10

Рисунок В.10 – Слайд 10

Робота з програмою. Реєстрація. Авторизація. Список заявок

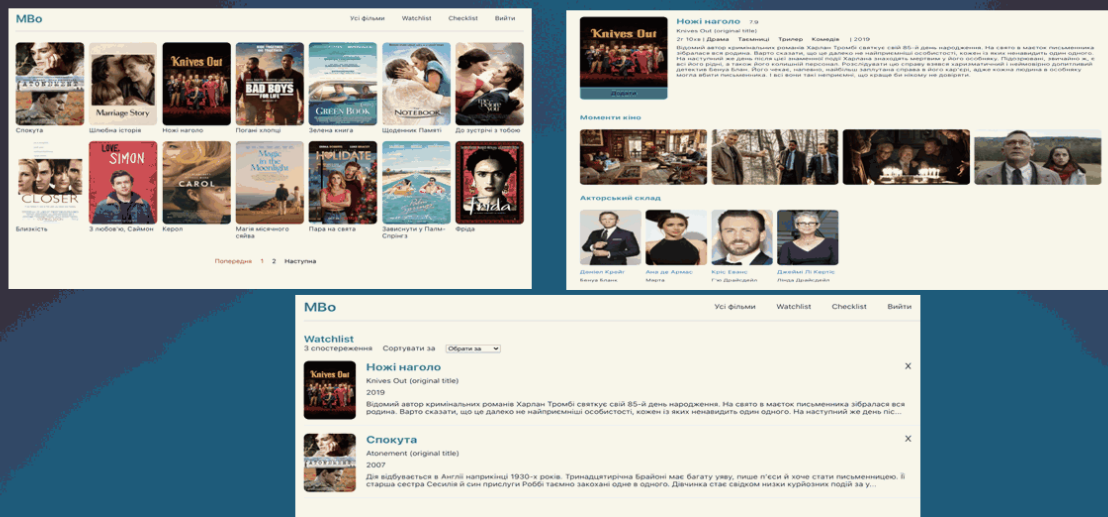


Рисунок В.12 – Слайд 12

Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для перегляду інформації про фільми.

Новизна роботи полягає в тому, що запропоновано модель інтерфейсу користувача програмного забезпечення для перегляду інформації про фільми, яка подана у вигляді прототипу вебсторінки, та дозволяє розроблювати програмне забезпечення для перегляду інформації про фільми.

Практична цінність полягає у розробці програмного забезпечення для перегляду інформації про фільми.

Результати тестування програмного забезпечення показали, що розроблена програма дозволяє розв'язувати поставлені завдання.

13

Рисунок В.13 – Слайд 13