

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Платформа для обговорення та оцінювання фільмів»

Виконав студент групи ІПЗ-21-1
РИКУН Артем Олександрович

Керівник роботи:
КУШНІР Надія Олександрівна

Рецензент:
ЧИЖМОТРЯ Олена Геннадіївна

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії програмного
забезпечення

Тетяна ВАКАЛЮК

«26» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **РИКУН Артем Олександрович**

Керівник роботи: **КУШНІР Надія Олександрівна**

Тема роботи: **«Платформа для обговорення та оцінювання фільмів»**,
затверджена Наказом закладу вищої освіти від 14.02.2025 р., №61/с

Вихідні дані для роботи: **об'єктом дослідження є інформаційні технології для управління процесами обговорення та оцінювання фільмів, а предметом дослідження є методи та засоби створення платформи для оцінювання та обговорення фільмів.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Кушнір Н. О.	27.02.2025р.	27.02.2025р.
2	Кушнір Н. О.	27.02.2025р.	27.02.2025р.
3	Кушнір Н. О.	27.02.2025р.	27.02.2025р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з платформи для обговорення та оцінювання фільмів та пояснювальної записки. Пояснювальна записка нараховує 82 сторінки друкованого тексту, містить 67 ілюстрацій та 23 таблиці. Джерела інформації включають 30 найменувань і займають 2 сторінки. В роботі міститься 13 сторінок додатків. Загальний обсяг пояснювальної записки становить 95 сторінок.

Виконано розробку веб-платформи для обговорення та оцінювання фільмів. Реалізовано функціонал для створення облікових записів, додавання відгуків, вибору улюблених акторів, спілкування між користувачами та взаємодії з рекомендаціями. Обрано архітектуру програмного забезпечення, структуру бази даних та логіку реалізації прав доступу для різних ролей. Інтеграцію з TMDb API реалізовано для імпорту фільмів та отримання даних про акторів, знімальну групу та іншу інформацію, яка пов'язана з фільмами.

Для реалізації веб-платформи використано фреймворк Laravel, шаблонізатор Blade, фреймворк Tailwind CSS, мову програмування JavaScript та бібліотеку Alpine.js, систему керування базами даних MySQL та середовище XAMPP для локального розгортання. Результати побудови та проектування підтверджені тестуванням усіх основних функцій системи.

Система може бути використана як навчальна, а також як база для створення повноцінного комерційного проєкту, якщо замінити TMDb API, оскільки він призначений тільки для звичайного користування, а не комерційного.

КЛЮЧОВІ СЛОВА: ВЕБ-ПЛАТФОРМА, ОЦІНЮВАННЯ ТА ОБГОВОРЕННЯ, ФІЛЬМ, BLADE, JAVASCRIPT, LARAVEL, MYSQL, XAMPP.

					ІПЗ.КР.Б – 121 – 25 – ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Рикун А.О.			Платформа для обговорення та оцінювання фільмів. Пояснювальна записка	Літ.	Арк.
Перевір.							3
Керівник		Кушнір Н.О.				Аркушів	
Н. контр.		Тичина О.П.				82	
Зав. каф.		Вакалюк Т.А.				ФІКТ Гр. ІПЗ-21-1	

ABSTRACT

The bachelor's thesis consists of a platform for discussing and evaluating films and an explanatory report. The report contains 82 pages of printed text, includes 67 illustrations and 23 tables. The information sources include 30 references and takes 2 pages. The appendices section consists of 13 pages. The total volume of the work is 95 pages.

A web platform for discussing and evaluating films has been developed. Functionality for creating user accounts, adding reviews, selecting favorite actors, communicating between users, and interacting with recommendations has been implemented. The software architecture, database structure, and access control logic for different user roles were defined. Integration with the TMDB API was implemented to import films and retrieve data about actors, crew, and other film-related information.

To implement the web platform, the Laravel framework, Blade templating engine, Tailwind CSS framework, JavaScript programming language, Alpine.js library, MySQL database management system, and the XAMPP environment for local deployment were used. The design and development results were verified through testing of all the main system functions.

The system can be used for educational purposes and also as a foundation for building a fully-fledged commercial project, provided that the TMDB API is replaced, since it is intended for personal, non-commercial use only.

KEYWORDS: BLADE, EVALUATION AND DISCUSSION, FILM, JAVASCRIPT, LARAVEL, MYSQL, XAMPP, WEB PLATFORM.

		Рикун А.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Кушнір Н.О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП	7
1 АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ПЛАТФОРМИ ДЛЯ ОБГОВОРЕННЯ ТА ОЦІНЮВАННЯ ФІЛЬМІВ	8
1.1 Постановка задачі	8
1.2 Аналіз аналогів програмного продукту	9
1.3 Вибір архітектури веб-орієнтованої системи	16
1.4 Обґрунтування вибору інструментальних засобів	18
Висновки до першого розділу	20
2 ПРОЄКТУВАННЯ ПЛАТФОРМИ ДЛЯ ОБГОВОРЕННЯ ТА ОЦІНЮВАННЯ ФІЛЬМІВ	22
2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи	22
2.2 Розробка бази даних	28
2.3 Проектування та реалізація алгоритмів роботи платформи	41
2.4 Реалізація функціоналу платформи	44
Висновки до другого розділу	46
3 ІНТЕРФЕЙС, ПОРЯДОК РОБОТИ ТА ТЕСТУВАННЯ ПЛАТФОРМИ ДЛЯ ОБГОВОРЕННЯ ТА ОЦІНЮВАННЯ ФІЛЬМІВ	48
3.1 Порядок встановлення та налаштування параметрів системи	48
3.2 Структура інтерфейсу платформи для обговорення та оцінювання фільмів	51
3.3 Тестування роботи платформи	67
Висновки до третього розділу	78
ВИСНОВКИ	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
ДОДАТКИ	82

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ - програмне забезпечення

API - прикладний програмний інтерфейс

REST - передача репрезентативного стану

БД - база даних

CRUD - основні функції для управління даними (створення, читання, оновлення, видалення)

MVC - розподілення даних та логіки застосунку на окремі компоненти (Model, View, Controller)

UI - інтерфейс користувача (User Interface)

UX - досвід користувача (User Experience)

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми. Однією з найважливіших речей у повсякденні є створення веб-платформ, які забезпечують ефективний обмін інформацією, інтерактивність та доступність даних у режимі реального часу. Веб-платформи для обговорення та оцінювання фільмів стають все більш популярними, оскільки вони дозволяють користувачам обмінюватися думками, залишати відгуки та створювати власні списки переглянутих фільмів або серіалів. Популярні сервіси, такі як Letterboxd, IMDb використовують персоналізовані рекомендаційні системи для покращення взаємодії користувачів із платформою.

Метою роботи є розробка алгоритмів роботи та реалізація платформи для обговорення та оцінювання фільмів. Ця система має забезпечувати користувачам можливість обирати та оцінювати улюблені фільми, писати відгуки, взаємодіяти з іншими користувачами та отримувати персоналізовані рекомендації.

Для досягнення поставленої мети необхідно вирішити такі **завдання**: провести огляд й аналіз існуючих аналогічних сайтів для обговорення та оцінювання фільмів, визначити ключові вимоги до архітектури та функціоналу системи, обґрунтувати вибір технологій та інструментальних засобів для реалізації веб-платформи, розробити інформаційну та програмну архітектуру системи, реалізувати механізми інтерактивної взаємодії користувачів та персоналізованих рекомендацій.

Об'єктом дослідження є інформаційні технології для управління процесами обговорення та оцінювання фільмів.

Предметом дослідження є методи та засоби створення платформи для оцінювання та обговорення фільмів.

За темою випускної кваліфікаційної роботи бакалавра опубліковані тези: Рикун А.О., Кушнір Н.О. Платформа для обговорення та оцінювання фільмів. Тези доповідей Всеукраїнської науково-практичної on-line конференції здобувачів вищої освіти і молодих учених, присвяченої Дню науки, 12-17 травня 2025 року. Житомир: «Житомирська політехніка», 2025. С. 63.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ПЛАТФОРМИ ДЛЯ ОБГОВОРЕННЯ ТА ОЦІНЮВАННЯ ФІЛЬМІВ

1.1 Постановка задачі

У сучасному світі кінематограф є однією з найпопулярніших форм розваг, і щороку виходять сотні нових фільмів. Глядачі все частіше шукають онлайн-платформи, де можна не тільки переглядати інформацію про фільми, але й ділитися власними враженнями, писати відгуки, ставити оцінки, створювати списки переглянутого контенту та взаємодіяти з іншими користувачами. На ринку вже існують подібні сервіси, такі як Letterboxd, IMDb проте вони мають певні обмеження, зокрема щодо гнучкості персоналізації, інтерактивності та зручності інтерфейсу.

Зважаючи на це, виникає необхідність розробки нової веб-платформи для обговорення та оцінювання фільмів, яка поєднає в собі сучасний дизайн, широкий функціонал та інтерактивність. Основною метою даного проєкту є створення веб-орієнтованої системи, яка дозволить користувачам: переглядати базу даних фільмів, читати короткі описи, відгуки та рейтинги; залишати власні відгуки та оцінки до фільмів; вести персональні списки переглянутих або запланованих до перегляду фільмів; брати участь у дискусіях та обговореннях, залишаючи коментарі та відповіді; отримувати персоналізовані рекомендації на основі вподобань та переглянутого контенту.

Для реалізації даної платформи буде використано сучасні технології. Серверна частина буде розроблена за допомогою фреймворку Laravel, що забезпечить швидку та безпечну роботу веб-застосунку, а також дозволить ефективно працювати з базою даних. Для збереження інформації про фільми, користувачів, оцінки та відгуки буде використовуватися база даних MySQL.

Інтерфейс буде створено за допомогою Blade, шаблонного інструменту Laravel, який допоможе легко працювати зі змінними даними та створювати багаторазові частини, такі як картки фільмів, списки відгуків, форми коментарів

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

тощо. Tailwind CSS використовуватиметься для стилізації веб-інтерфейсу. Це допоможе швидко створити адаптивний і гарний дизайн, не пишучи зайвих власних стилів.

Для досягнення поставленої мети необхідно вирішити такі завдання: провести аналіз аналогічних веб-платформ, виявити їхні сильні та слабкі сторони; спроектувати архітектуру веб-системи, визначивши ключові компоненти та взаємодію між ними; розробити систему реєстрації та аутентифікації користувачів; реалізувати основні функції сайту, а саме перегляд каталогу фільмів, написання відгуків, виставлення оцінок та взаємодію між користувачами; використати Blade для створення шаблонів сторінок та відображення динамічного контенту; застосувати Tailwind CSS для стилізації веб-інтерфейсу та забезпечення адаптивності дизайну; розробити систему рекомендацій на основі оцінок користувачів та їхніх уподобань; інтегрувати API сторонніх сервісів (наприклад, TMDB) для отримання інформації про фільми; провести тестування веб-платформи, усунути можливі помилки та оптимізувати продуктивність.

В результаті розробки буде створено функціональну, швидку та зручну платформу для кіноманів, яка дозволить не лише ділитися думками про фільми, а й взаємодіяти з іншими користувачами, отримувати персоналізовані рекомендації та формувати власну кіноколекцію. Використання Laravel, Blade та Tailwind CSS забезпечить швидку розробку, зручну підтримку та сучасний зовнішній вигляд системи.

1.2 Аналіз аналогів програмного продукту

На сьогодні існує низка популярних веб-платформ для обговорення та оцінювання фільмів, які надають користувачам можливість вести особисті списки переглянутого, залишати відгуки та рейтинги, а також взаємодіяти з іншими кіноманами. Найбільш відомими серед них є Letterboxd, IMDb та TMDb. Аналіз цих платформ дозволяє визначити їхні основні функціональні можливості, а також

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

виявити переваги та недоліки, що допоможе розробити конкурентоспроможний продукт.

Letterboxd - це одна з найпопулярніших платформ для ведення кінощоденників, яка дозволяє користувачам створювати списки фільмів, залишати відгуки, оцінювати переглянуті фільми та взаємодіяти з іншими кіноманами [16]. На рис. 1.1 представлено головну сторінку цієї платформи, звідки можна побачити її основні функції: увійти, створити акаунт, фільми, списки, користувачі та журнал разом з пошуковою системою. Розділ користувачів дає змогу знаходити інших однодумців, які люблять кіно, переглядати їхні профілі та взаємодіяти з ними. Журнал же є особистим щоденником переглядів, де відображаються всі дії користувача на платформі.

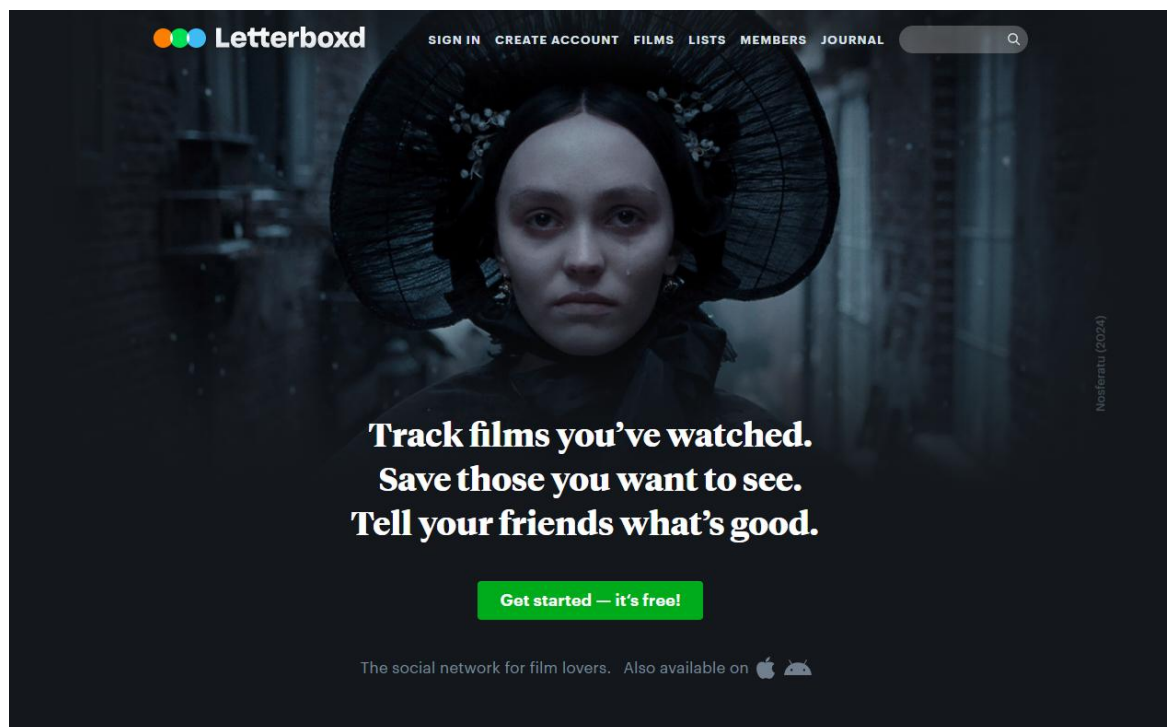


Рисунок 1.1 - Головна сторінка Letterboxd

Також важливим елементом є пошукова система, яка допомагає швидко знаходити потрібні фільми, списки або користувачів, завдяки чому навігація по сайту є простою і зручною. Усі ці функції разом створюють комфортний і привабливий інструмент для любителів кіно, що дозволяє ефективно організувати та підтримувати активну комунікацію в спільноті.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

На рис. 1.2 та рис. 1.3 відповідно представлені сторінка фільму та її продовження.

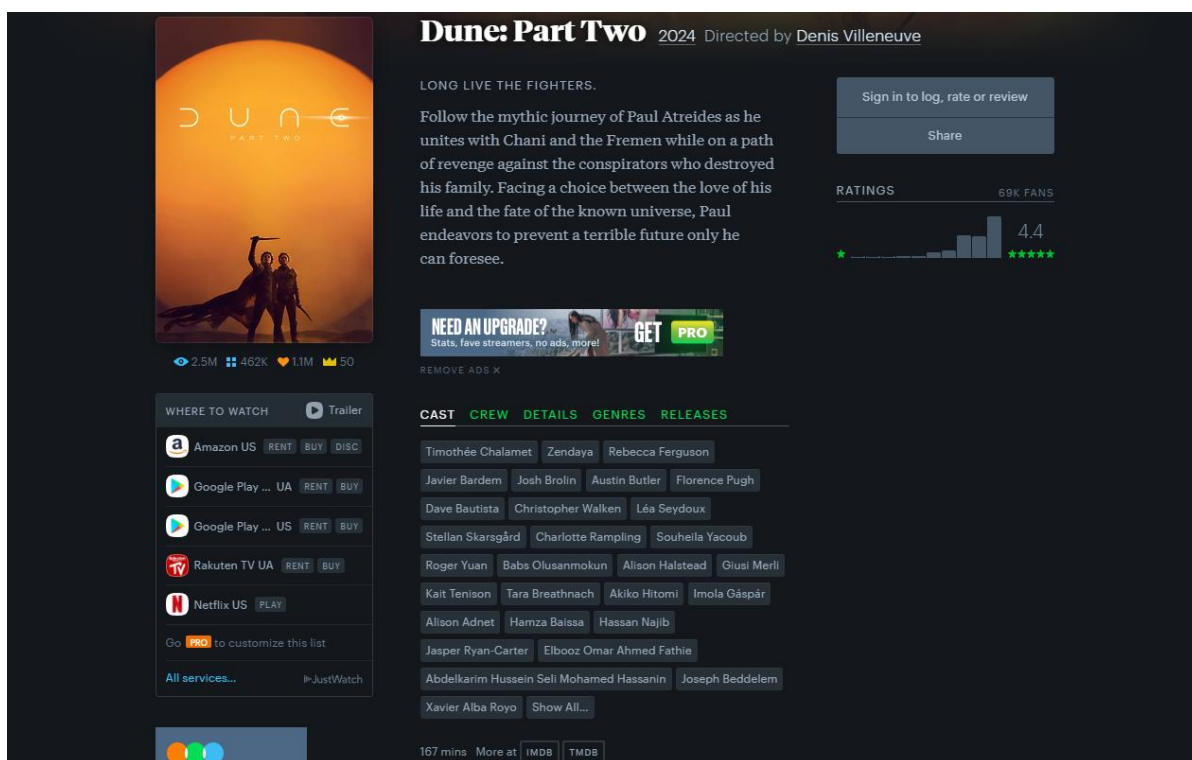


Рисунок 1.2 - Сторінка фільму на Letterboxd

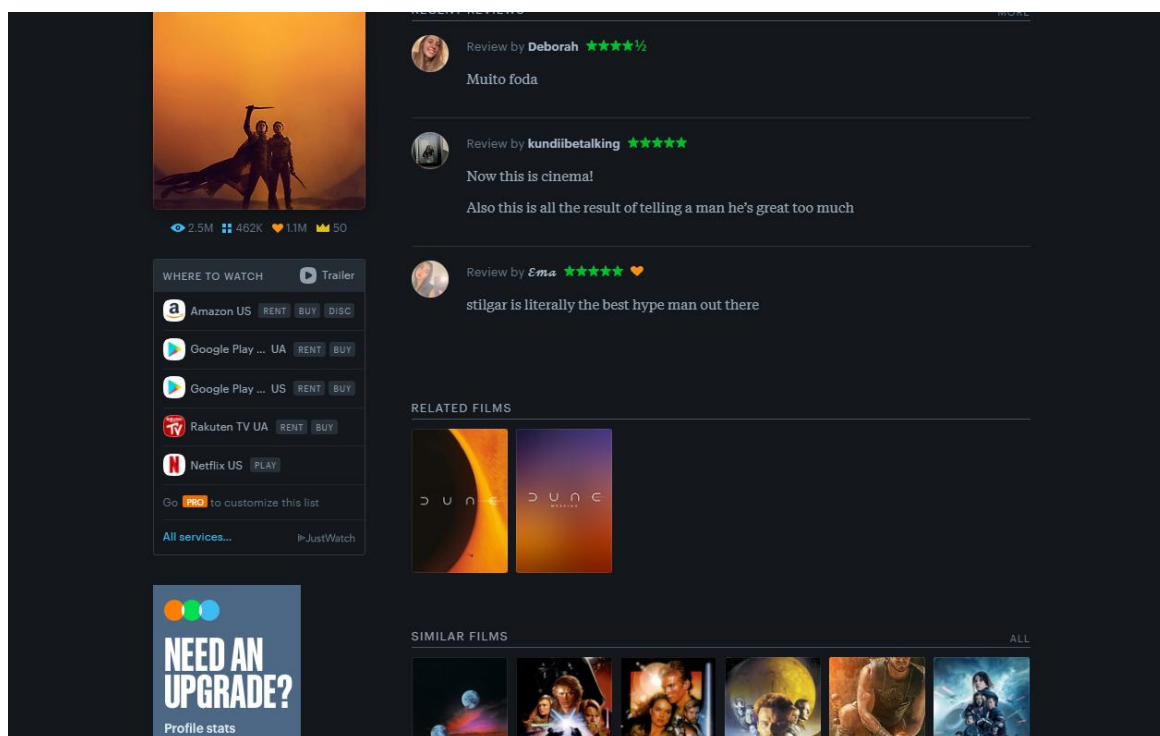


Рисунок 1.3 - Продовження сторінки

Як видно, основні можливості Letterboxd також включають додавання фільмів до списків (переглянуто, заплановано тощо), написання відгуків та

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

виставлення оцінок (за 5-зірковою шкалою), створення власних добірок та рекомендацій, а також соціальні функції: підписки, коментарі, лайки.

На рис. 1.2 можна подивитися як платформа інтегрується з базами даних фільмів, наприклад, такими як TMDb.

Однак, платформа Letterboxd має низку недоліків. Серед них можна виділити відсутність можливості перегляду фільмів, обмежений функціонал без преміум-підписки (наприклад, розширена статистика доступна лише платним користувачам), а також відсутність гнучкої системи рекомендацій.

IMDb - одна з найстаріших та найбільших баз даних про кіно, яка містить інформацію про фільми, серіали, акторів, режисерів та інший контент, пов'язаний з кінематографом [12]. На рис. 1.4 представлено головну сторінку IMDb, де є можливість побачити нові новини або статті, зайти в профіль, змінити мову, переглянути популярні трейлери.

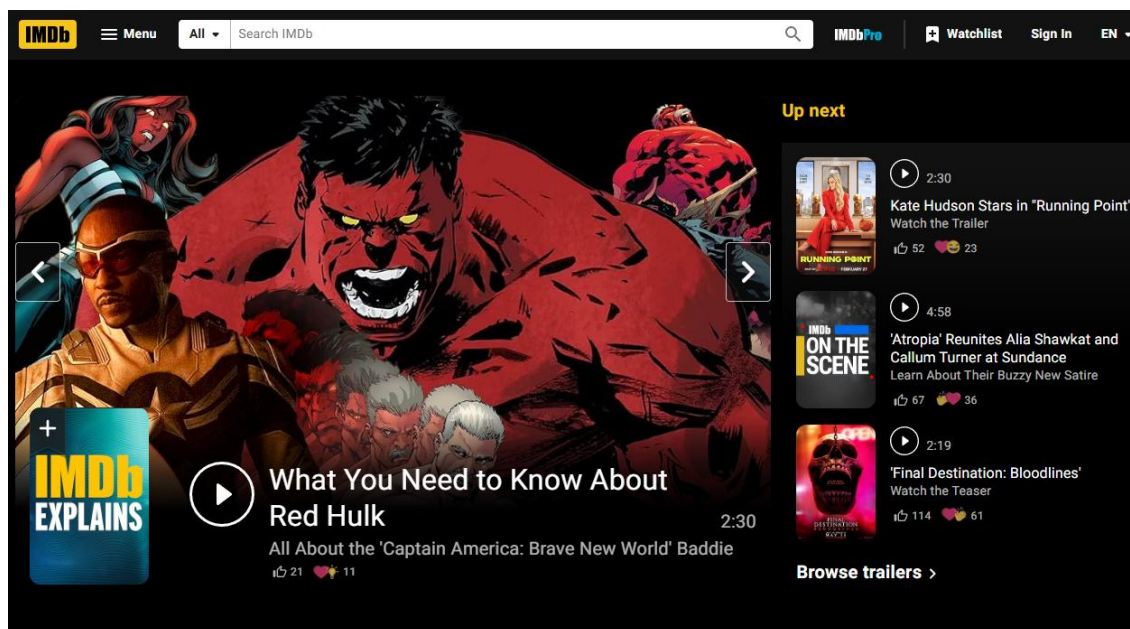


Рисунок 1.4 - Головна сторінка IMDb

Основні можливості IMDb включають глобальну рейтинговану систему на основі голосування користувачів, велику базу даних з детальною інформацією про фільми та серіали, можливість залишати відгуки та коментарі, а також список перегляду.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

На рис. 1.5 можна побачити функції меню: фільми, серіали, знаменитості, заходи та рекомендації щодо перегляду. Однак платформа має обмежену кількість соціальних функцій, таких як ведення кінощоденника або створення добірок, і часто містить багато комерційного контенту. З особливого можливо виділити: розділ нагород та заходів, де виділені найпопулярніші нагороди та заходи з можливістю передивитися всі події; розділ знаменитих людей з новинами про їх життя; розділ спільноти з можливістю брати участь в голосуваннях та розвитку сайту; розділ серіалів, де є можливість оцінити та дізнатися більше про іншу сферу кіно, яка тісно пов'язана з фільмами.

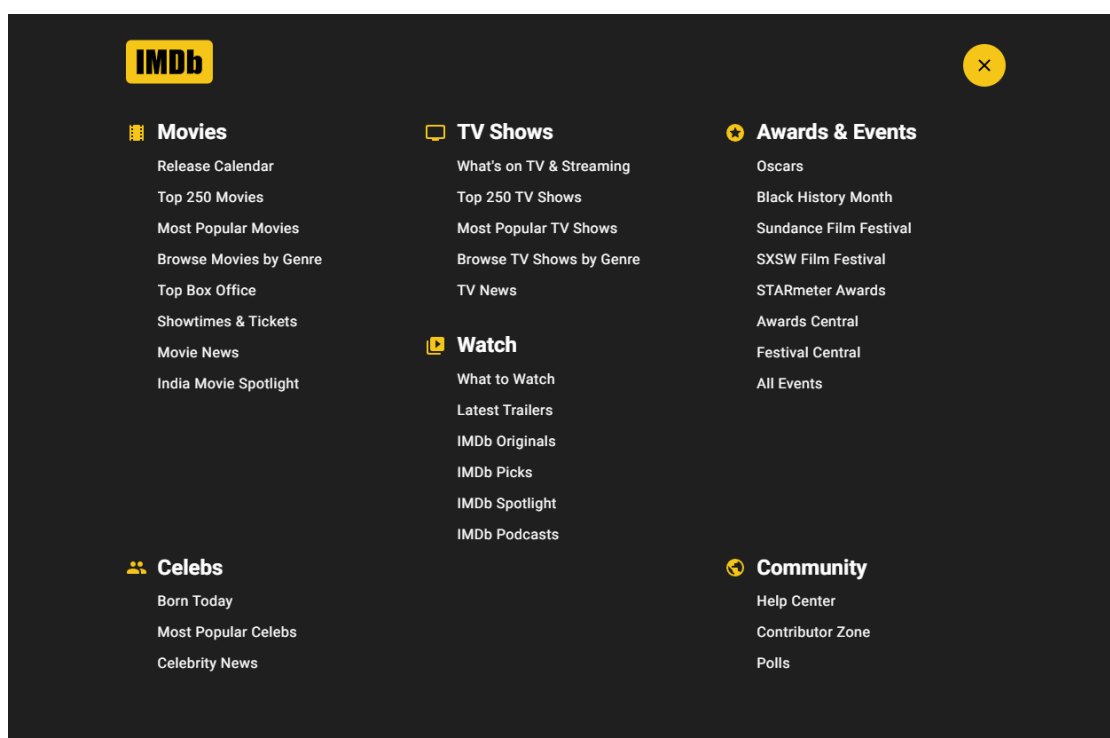


Рисунок 1.5 - Можливості меню

TMDb - це відкрита база даних фільмів та серіалів, яка дозволяє розробникам інтегрувати її API у власні проєкти [26]. На рис. 1.6 можна побачити категорії люди, фільми, серіали та більше. Також ще є вибір мови, реєстрація вхід та пошук. Основні можливості TMDb включають доступ до великої кількості фільмів та серіалів, відгуки, рейтинги та коментарі користувачів, можливість створення списків перегляду, а також відкритий API для використання в інших застосунках.

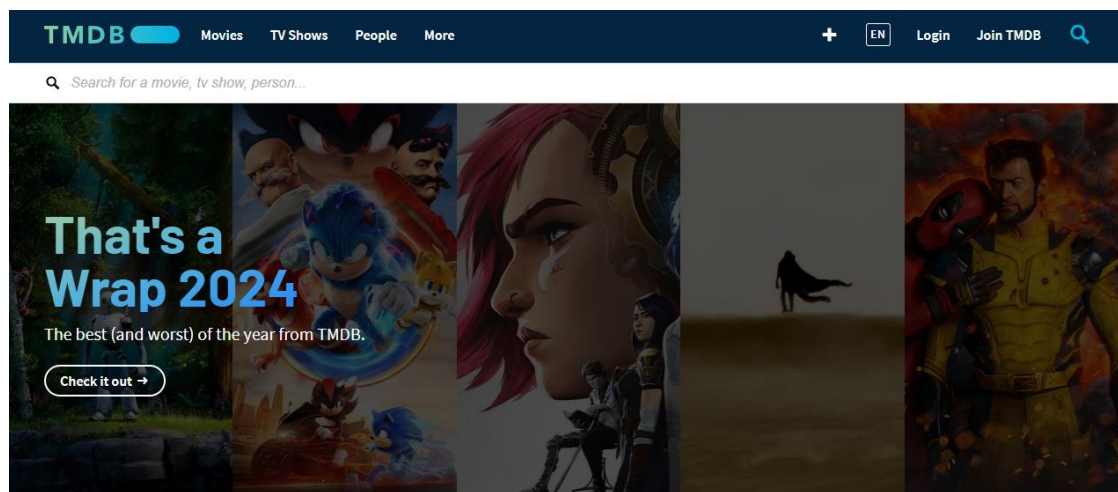


Рисунок 1.6 - Головна сторінка TMDb

На рис. 1.7 зображено сторінку популярних фільмів. На цій сторінці відображено список фільмів в виді карток. Однієї з відмінних характеристик є наявність рейтингу в лівому нижньому кутку кожного фільму та дати виходу під назвою фільму. Також ще є розширена фільтрація і сортування.

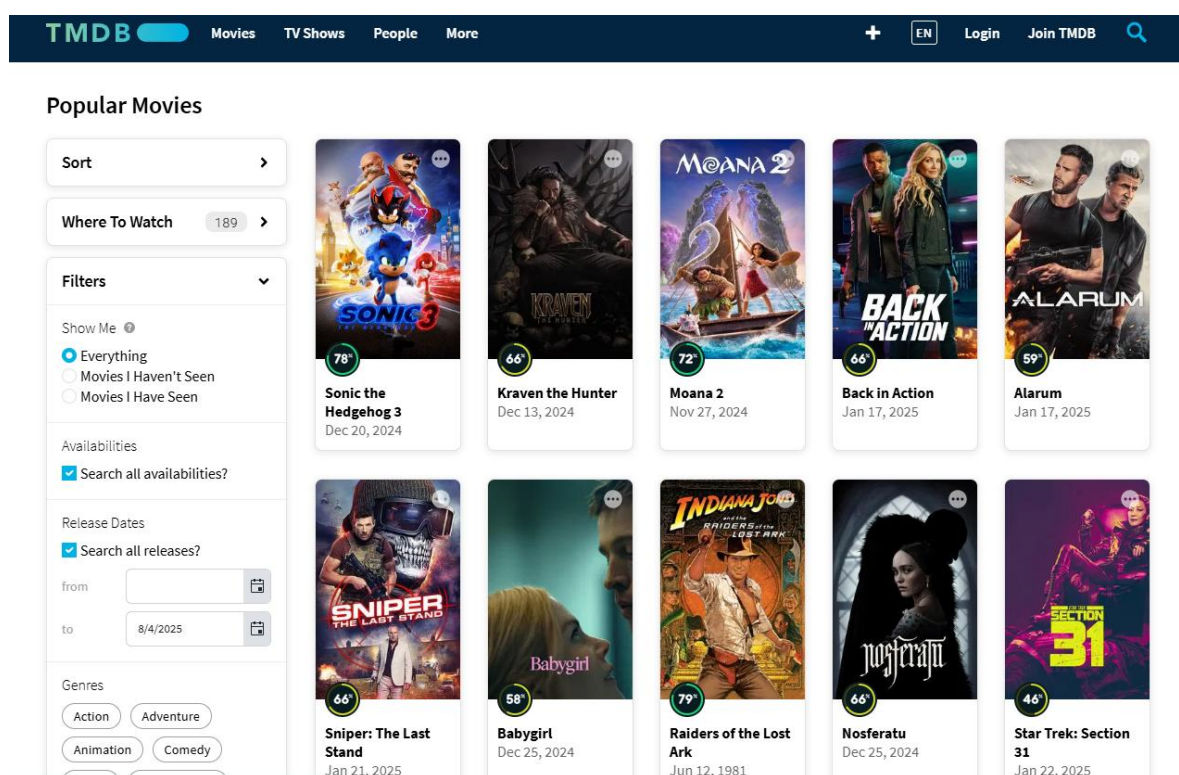


Рисунок 1.7 - Сторінка популярних фільмів

Таким чином, аналіз існуючих аналогів дозволяє визначити основні напрями розвитку веб-платформи та створити конкурентоспроможний продукт, який буде не лише альтернативою популярним сервісам, а й запропонує унікальні можливості

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Кушнір Н.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

для кіноманів. У табл.1.1 представлено порівняльну характеристику розглянутих аналогів програмного продукту за основними параметрами.

Таблиця 1.1

Порівняльна характеристика аналогів програмного продукту

№	Функціонал	Letterboxd	IMDB	TMDB	Розроблювана платформа
1	Перегляд фільмів та серіалів	+/-	+	+	+/-
2	Написання відгуків	+	+	+	+
3	Виставлення оцінок	+	+	+	+
4	Створення списків (переглянуто, заплановано тощо)	+	+	+	+
5	Коментування відгуків	+	—	+	+
6	Лайки / взаємодія з відгуками	+	—	+	+
7	Підписки на інших користувачів	+	—	—	-
8	Стрічка активності	+	+	+	+/-
9	Персоналізовані рекомендації	—	—	+	+
10	Можливість створювати добірки фільмів	+	—	+	-
11	Можливість перегляду трейлерів	—	+	+	+
12	Сортування та фільтрація за жанрами, рейтингом тощо	+	+	+	+
13	Вибір мови інтерфейсу	+	+	+	-
14	API для зовнішніх розробників	—	—	+	—

Порівнюючи існуючі рішення, можна зробити висновок, що хоча кожна з платформ має свої переваги, жодна не забезпечує одночасно всі функції, необхідні для активної взаємодії користувачів. Запропонована веб-платформа буде включати такі ключові особливості: гнучку систему рекомендацій, яка базуватиметься на оцінках користувачів, їхніх уподобаннях та взаємодії з контентом; інтерактивні можливості, такі як коментування, лайки, обговорення відгуків та персоналізовані стрічки активності; зручний адаптивний інтерфейс, створений за допомогою Blade

та Tailwind CSS, що зробить сайт легким у використанні на будь-яких пристроях; швидкість роботи та масштабованість завдяки використанню Laravel для бекенду та MySQL для роботи з базою даних; інтеграцію з API TMDb для отримання актуальної інформації про фільми.

1.3 Вибір архітектури веб-орієнтованої системи

Для створення веб-платформи з обговорення та оцінювання фільмів необхідно обрати ефективну архітектуру, яка забезпечить високу продуктивність, масштабованість і зручність користування. Важливою особливістю такої системи є підтримка великої кількості користувачів, оперативне оновлення контенту та гнучкість у додаванні нових функцій.

В основі веб-платформи лежить багаторівнева архітектура, що складається з трьох основних компонентів: клієнтська частина (Frontend) - інтерфейс, з яким взаємодіє користувач, серверна частина (Backend) - обробка запитів, бізнес-логіка та контроль доступу, база даних (Data Layer) - сховище даних про фільми, користувачів, рейтинги та коментарі.

Кожен із цих рівнів відіграє важливу роль у забезпеченні стабільності роботи системи та її подальшого розвитку.

Клієнтська частина відповідає за відображення інформації та взаємодію з користувачем. Важливо забезпечити швидку та зручну навігацію, щоб користувач міг легко знаходити потрібний контент і брати участь в обговореннях. Для реалізації інтерфейсу використовується шаблонізатор Blade у Laravel, який дозволяє ефективно генерувати сторінки з мінімальним навантаженням на сервер. Blade підтримує архітектурний шаблон MVC (Model-View-Controller), розділяючи логіку застосунка (контролери) та уявлення (шаблони), що покращує структуру коду, робить його більш читабельним і підтримуваним. Це також сприяє швидкому впровадженню змін в інтерфейс користувача без необхідності змінювати логіку. Дизайн реалізується за допомогою Tailwind CSS, що забезпечує зручну адаптивну верстку та гнучке налаштування стилів.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

Серверна частина (Backend) є основним рушієм платформи, який обробляє запити користувачів, виконує бізнес-логіку та забезпечує безпеку системи. Для цього було обрано PHP-фреймворк Laravel, який відомий своєю продуктивністю, зручністю та безпекою. Він містить вбудовані механізми автентифікації, авторизації, кешування та роботи з базами даних, що значно прискорює розробку. Для взаємодії з базою даних використовується ORM Eloquent, який дозволяє працювати з даними у вигляді об'єктів, що спрощує код і покращує його читабельність.

Laravel також забезпечує захист від міжсайтової підміни запитів (CSRF, Cross-Site Request Forgery) шляхом автоматичного генерування унікальних токенів для кожної сесії користувача. Ці токени вбудовуються у всі форми HTML, що надсилаються POST-запитом, і перевіряються при кожному зверненні до сервера. Це унеможливорює виконання запитів сторонніми зловмисниками від імені авторизованих користувачів, тим самим підвищуючи загальну безпеку платформи.

Для захисту системи від потенційно небезпечних або некоректних даних у платформі реалізовано серверну валідацію вхідних даних. Laravel надає гнучкий механізм перевірки, який дозволяє визначати обов'язкові поля, обмеження за типом, довжиною, форматом (наприклад, для email-адрес або паролів).

База даних містить усю необхідну інформацію про фільми, користувачів, відгуки та взаємодію між учасниками платформи. Оскільки система потребує швидкого доступу до великої кількості структурованих даних, було обрано реляційну СУБД MySQL. Вона забезпечує ефективне збереження, пошук та обробку інформації, що є важливим для роботи з великою кількістю записів.

Для інтеграції з зовнішніми сервісами буде реалізовано API, який дозволить отримувати інформацію про фільми з The Movie Database (TMDb). Це допоможе автоматично оновлювати дані про нові релізи, рейтинги та описи фільмів, що значно розширить функціонал платформи.

Архітектура системи передбачає можливість масштабування у майбутньому. Наприклад, можна використовувати контейнеризацію за допомогою Docker, що

		Риун А.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Кушнір Н.О.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

полегшить підтримку і розгортання системи. Крім того, можливе додавання алгоритмів персоналізованих рекомендацій, що покращить користувацький досвід.

Таким чином, обрана архітектура забезпечує високу продуктивність, безпеку та зручність використання, а також дозволяє легко розширювати можливості платформи у майбутньому. Це дасть змогу створити сучасний, функціональний та конкурентоспроможний продукт для кіноманів.

1.4 Обґрунтування вибору інструментальних засобів

У процесі розробки веб-платформи для обговорення та оцінювання фільмів важливо правильно обрати технології та апаратні ресурси, які забезпечать її ефективну роботу, безпеку та масштабованість. Оскільки система має обробляти велику кількість запитів, зберігати інформацію про фільми, відгуки, оцінки користувачів та їхню активність, вибір інструментів має ґрунтуватися на їхній продуктивності, стабільності та можливості подальшого розширення функціоналу. Клієнтська частина реалізується з використанням Blade [5], що є механізмом шаблонів у фреймворку Laravel. Це дозволяє легко керувати динамічним контентом сторінок та мінімізувати навантаження на сервер. Для стилізації використовується фреймворк Tailwind CSS [23], що дає змогу швидко створювати адаптивні та зручні інтерфейси.

Серверна частина виконується на основі PHP-фреймворку Laravel [14], який забезпечує безпеку, зручність у розробці та має вбудовані механізми для роботи з базами даних, кешуванням, авторизацією та захистом від атак. Laravel підтримує REST API, що дозволяє легко інтегрувати зовнішні сервіси та забезпечити ефективну взаємодію клієнтської і серверної частин. Використання ORM Eloquent у Laravel спрощує роботу з базою даних та підвищує гнучкість розробки. Для зменшення навантаження на базу даних і пришвидшення роботи серверної частини також можливо застосувати системи кешування Redis або Memcached, які зберігають часто використовувані дані в пам'яті, що значно зменшує час доступу до них. Для підвищення рівня безпеки Laravel забезпечує захист від SQL-ін'єкцій,

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

міжсайтового скриптингу та міжсайтової підробки запитів. Впроваджуються додаткові заходи, такі як хешування паролів за допомогою bcrypt, захист API-токенів, регулярне резервне копіювання бази даних та обмеження частоти запитів для запобігання DDoS-атакам.

У табл. 1.2 присутнє порівняння характеристик різних фреймворки для бекенду. Було обрано Laravel через його простоту, ORM (Eloquent) і інтеграцію з Blade та Tailwind CSS. У порівнянні з Django, Node.js і Spring Boot, Laravel пропонує зручність вивчення та масштабованість для веб-застосунків, що робить його оптимальним вибором для даного проєкту.

Таблиця 1.2

Порівняння характеристик фреймворків

Характеристика	Laravel	Django	Node.js	Spring Boot
Мова програмування	PHP	Python	JavaScript	Java
Вбудований ORM	Так (Eloquent)	Так (Django ORM)	Ні (можна використовувати Sequelize, Mongoose)	Так (Hibernate)
Швидкість роботи	Середня	Висока	Висока	Висока
Гнучкість	Висока	Середня	Висока	Висока
Простота навчання	Середня	Середня	Висока	Низька
Безпека	Висока	Висока	Середня	Висока

Для отримання актуальної інформації про фільми використовується інтеграція з зовнішнім API, зокрема TMDb. Це дозволяє автоматично оновлювати дані про фільми, включаючи рейтинги, трейлери, описи та зображення, що значно покращує користувацький досвід.

База даних містить інформацію про фільми, користувачів, відгуки та рейтинги, тому важливо обрати ефективне рішення для її зберігання та обробки. Для цього використовуються реляційні системи керування базами даних, такі як MySQL або PostgreSQL, що забезпечують швидкість обробки та збереження даних навіть при високих навантаженнях. У даному проєкті обрано СУБД MySQL [19].

У табл. 1.3 наведено порівняльну характеристику баз даних.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

Порівняльна характеристика баз даних

Характеристика	MySQL	PostgreSQL	MongoDB	SQLite
Тип бази даних	Реляційна	Реляційна	NoSQL (документоорієнтована)	Файлова
Швидкість роботи	Висока	Висока	Висока	Висока для невеликих проектів
Підтримка складних запитів	Обмежена	Висока	Немає	Обмежена
Масштабованість	Висока	Висока	Висока	Низька

Зважаючи на потребу у високій продуктивності та масштабованості платформи, визначено мінімальні та рекомендовані апаратні вимоги. Мінімальні параметри включають двоядерний процесор із тактовою частотою від 2.4 ГГц, 4 ГБ оперативної пам'яті, 50 ГБ SSD-дискового простору та сервер на базі операційної системи Ubuntu або Debian. Для стабільної роботи при великій кількості користувачів рекомендовано використовувати сервер із чотириядерним процесором, 8 ГБ оперативної пам'яті, SSD-накопичувачем обсягом від 100 ГБ та підтримкою балансування навантаження.

Завдяки використанню сучасних технологій веб-платформа забезпечуватиме високу продуктивність, безпеку та можливість подальшого розширення. Обрані рішення гарантують стабільну роботу системи, що відповідатиме вимогам користувачів та дозволить їм комфортно взаємодіяти з платформою. Додаткова увага приділяється безпеці платформи: використовуючи поточні методи шифрування, системи перевірки та забезпечує безпечне зберігання інформації користувачів та їх захисту.

Висновки до першого розділу

У першому розділі було розглянуто основні напрями використання інформаційних технологій для створення онлайн-платформи обговорення та оцінювання фільмів. Проведено постановку задачі, де визначено основні функціональні та нефункціональні вимоги до майбутньої системи. Під час аналізу

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

аналогічних програмних продуктів були виявлені їхні переваги та недоліки, що дозволило сформулювати власні критерії для покращення платформи.

Також здійснено вибір архітектури веб-орієнтованої системи, яка забезпечить масштабованість, надійність та ефективність роботи з великою кількістю користувачів, а саме клієнт – серверної. Особливу увагу приділено обґрунтуванню вибору інструментальних засобів розробки, які найкраще відповідають вимогам проєкту та забезпечують можливість реалізації всього необхідного функціоналу: фреймворку Laravel, шаблонізатору Blade, мови програмування JavaScript з бібліотекою Alpine.js та фреймворку Tailwind CSS.

Проведений аналіз створює необхідну теоретичну базу для подальшого етапу розробки онлайн-платформи та дозволяє ефективно планувати реалізацію системи.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ПЛАТФОРМИ ДЛЯ ОБГОВОРЕННЯ ТА ОЦІНЮВАННЯ ФІЛЬМІВ

2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

Варіанти використання описують взаємодію користувачів із системою, чітко визначаючи основні функціональні можливості, які платформа повинна забезпечити. Основні варіанти використання для цього проєкту є такими: реєстрація та автентифікація користувача: користувач без профілю може створити обліковий запис сайту через форму реєстрації. Під час реєстрації вона збирає основну індивідуальну інформацію, включаючи електронне, ім'я користувача та пароль. Після успішної реєстрації, гість може увійти за допомогою своїх ідентифікаційних деталей.

Після авторизації користувач отримує доступ до особистого профілю, де він може: редагувати особисту інформацію (ім'я, аватар, пошту тощо), змінити налаштування облікового запису (наприклад, мотто або фон профілю), дивитися активність платформи (оцінка, список перегляду, відгуки). Ця функція є важливою та основною для досягнення повних можливостей платформи - без входу користувач залишається гостем з обмеженим доступом.

Перегляд та оцінка фільмів: користувач може переглядати інформацію про фільми, включаючи відгуки, рейтинги, жанри та інші деталі. Користувач також може оцінювати фільми за допомогою зірок або іншої системи рейтингу.

Створення та обговорення відгуків: користувач може написати відгук на фільм, а також коментувати відгуки інших користувачів. Це дозволяє створювати інтерактивну атмосферу, де кінолюбители можуть обговорювати свої враження.

Пошукова система та фільтри: користувач може скористатися пошуковою системою для швидкого знаходження фільмів за назвою, жанром, актором чи іншими критеріями. Також є можливість застосовувати фільтри, щоб переглядати

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

лише ті фільми, які відповідають певним вимогам (наприклад, за рейтингом або роком випуску).

Персоналізовані рекомендації: платформа пропонує користувачам фільми, які можуть їх зацікавити, на основі їх оцінок та переглядів. Система використовує алгоритми для обробки цих даних і створює персоналізовані рекомендації.

Чат: користувачі можуть взаємодіяти через приватні чати, обговорюючи фільми або відгуки. Це дає змогу створювати комунікаційні простори для кіноманів і підвищує залученість користувачів.

Адміністрування та модерація: адміністратор має право управляти користувачами, редагувати або видаляти контент, контролювати відгуки та перегляди фільмів, а також модерувати спільноту користувачів.

На рис. 2.1 можна подивитися можливості різних акторів, таких як гість або незареєстрований користувач, користувач, модератор та адміністратор.

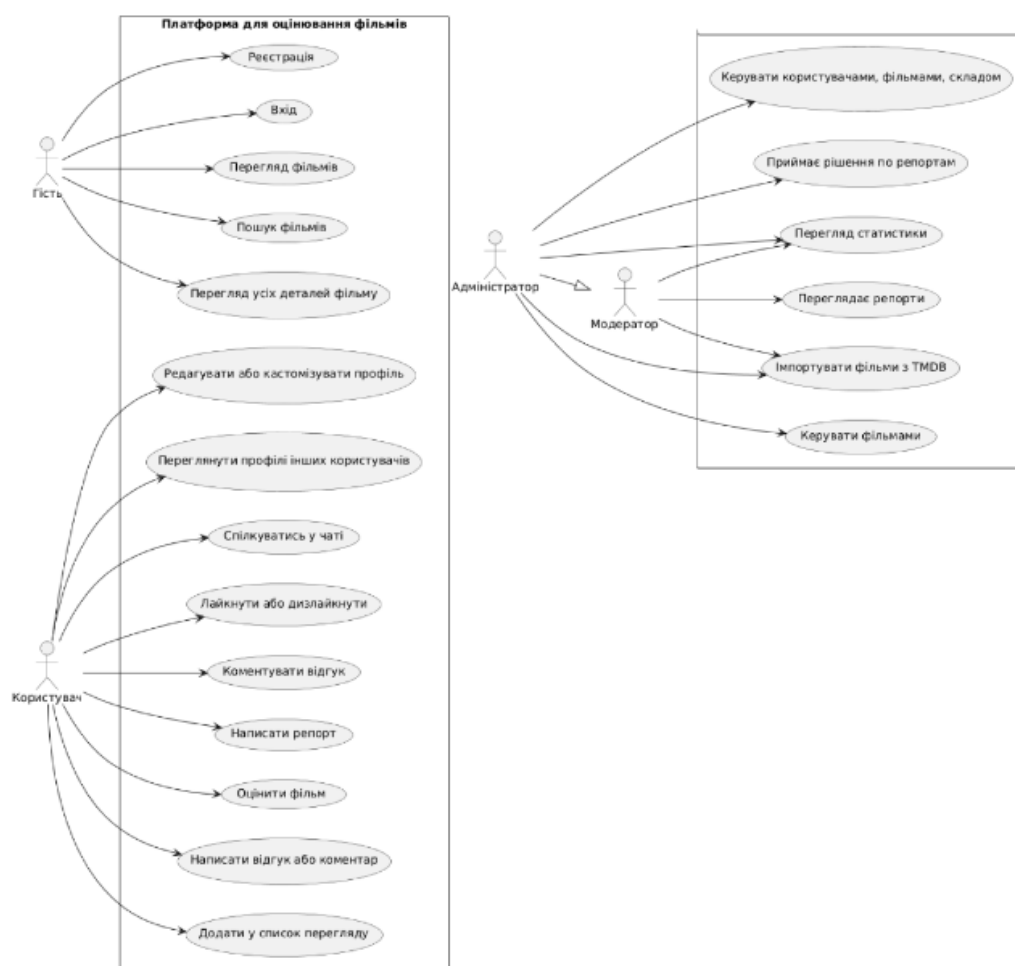


Рисунок 2.1 - Діаграма use-case

Гість має право: зареєструватися на платформі; увійти у свій обліковий запис; переглядати список доступних фільмів; шукати фільми за різними критеріями; переглядати детальну інформацію про фільми.

Користувач після входу може оцінювати фільми, писати відгуки або коментарі до фільмів, додавати фільми до власного списку перегляду, редагувати свій профіль і змінювати аватар, переглядати профілі інших користувачів, спілкуватися з іншими користувачами у чаті, лайкати або дизлайкати відгуки або коментарі, коментувати відгуки інших користувачів, повідомляти про порушення, надсилаючи скарги (репорти).

Модератор має змогу переглядати скарги, залишені користувачами; імпортувати нові фільми з зовнішнього джерела (TMDB); переглядати статистику платформи.

Адміністратор має розширені повноваження на відміну від модератора, а саме керувати базою фільмів (створення, редагування, видалення або оновлення); керувати користувачами, жанрами, акторами, членами знімальної групи тощо; переглядати всі скарги й приймати рішення щодо їх обробки; імпортувати фільми з TMDB; переглядати повну статистику активності на платформі.

Об'єктно-орієнтована структура системи допомагає організувати код та функціональність в об'єкти та класи, що відображають реальні сутності у системі.

На рис. 2.2 відображено діаграму класів.

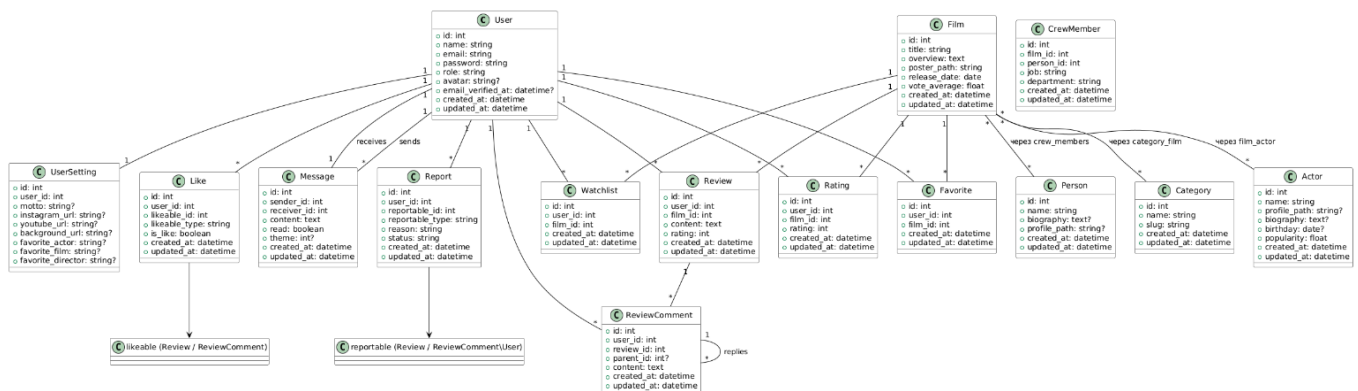


Рисунок 2.2 - Діаграма класів

Опишемо основні класи та їхню взаємодію.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Кушнір Н.О.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Клас User представляє користувача платформи, зберігаючи його дані (ім'я, email, пароль, аватар) та роль (адміністратор, модератор чи звичайний користувач). Користувач може мати список оцінених фільмів, відгуків та рейтингів.

Клас Film містить основну інформацію про фільм - назва, рік випуску, жанр, рейтинг, тощо.

Клас Review відповідає за збереження тексту відгуків, рейтингу та інформації про автора відгука. Це об'єкт, який пов'язаний з конкретним фільмом та користувачем. Кожен відгук може мати декілька коментарів до нього.

Клас Rating відповідає за оцінки, які користувачі ставлять фільмам. Містить значення оцінки (від 1 до 10 або іншу шкалу). Клас також може автоматично оновлювати оцінку, якщо оцінка була поставлена в відгуку і, навпаки, оцінка в відгуку оновлюється, якщо оновити сам рейтинг.

Клас UserSetting включає в себе кастомізацію профілю, а саме можливість змінити фон, додати посилання на соціальні мережі, додати улюбленого актора, тощо.

Клас Message відповідає за чати, де користувачі можуть спілкуватися між собою. Також зберігає статус чи було прочитано повідомлення та тему розмови, якщо тема була обрана.

Клас Watchlist представляє список фільмів, які користувач позначив як такі, що бажає переглянути. Зв'язується з фільмом та користувачем.

Клас Favorite описує фільми, які користувач додав до обраного. Також пов'язаний з користувачем та фільмом.

Клас Actor зберігає дані про акторів, а саме ім'я, дату народження, коротку біографію, фото та рейтинг популярності. Має зв'язок з фільмами через проміжну таблицю film_actor.

Клас Person представляє загальну інформацію про персон - наприклад, режисерів, продюсерів, сценаристів тощо, які є учасниками знімальної групи.

Клас CrewMember реалізує зв'язок між фільмом і персоною з уточненням посади (job) і відділу (department), наприклад, "режисер", "продюсер", "оператор".

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

Клас Category описує жанри фільмів, такі як "драма", "комедія", "фантастика" тощо. Має зв'язок з фільмами через таблицю category_film.

Клас ReviewComment клас реалізує коментарі до відгуків, підтримуючи вкладену структуру (коментар на коментар) через поле parent_id. Коментар також може отримати лайк чи дизлайк.

Клас Like клас описує реакцію користувача у вигляді лайка або дизлайка. Має поліморфну прив'язку до об'єкта, до якого застосовано реакцію (відгук чи коментар).

Клас Report клас реалізує механізм скарг на контент. Є поліморфним, оскільки може бути застосований до різних типів об'єктів. Містить причину, статус та посилання на об'єкт скарги.

У системі також реалізована взаємодія між різними об'єктами. Наприклад, клас User може взаємодіяти з класом Film, додаючи його до списку переглянутих, або з класом Review, щоб створити новий відгук на фільм. Алгоритми рекомендацій можуть використовувати дані з класів Rating та Review, щоб зрозуміти переваги користувача і запропонувати відповідні фільми. Клас Message дозволяє організувати спілкування між користувачами, включаючи можливість встановити тему розмови для обговорення фільмів та відгуків.

Використання об'єктно-орієнтованого підходу в проектуванні системи дозволяє чітко організувати код, забезпечити модульність та зручність у подальшому розвитку проекту. Визначення основних варіантів використання допомагає зрозуміти ключові функціональні можливості, які повинні бути реалізовані для успішної роботи платформи.

Функціональні вимоги до системи включають реалізацію класичної авторизації користувачів із використанням логіну та пароля. Також має бути забезпечена можливість анонімного перегляду зборів для неавторизованих користувачів. Система повинна підтримувати організацію спілкування між модераторами та користувачами, забезпечувати відображення результатів

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

проведених зборів, зберігати усю інформацію про створені збори, а також надавати користувачам доступ до перегляду статистики щодо їхніх зборів.

Нефункціональні вимоги стосуються таких аспектів, як сприйняття, надійність та продуктивність системи. Зокрема: час, необхідний для освоєння веб-застосунку, має становити від 3 до 5 хвилин, а для досвідчених користувачів - менше 2 хвилин. Час відповіді системи на дії користувача не повинен перевищувати 3 секунд. Інтерфейс веб-застосунку має бути інтуїтивно зрозумілим і не вимагати спеціальної підготовки користувача. Система повинна забезпечувати цілодобову доступність, працюючи 24 години на добу, 7 днів на тиждень. Вебресурс повинен підтримувати одночасну роботу великої кількості користувачів без погіршення продуктивності.

Системні вимоги є такими. Для стабільної та комфортної роботи вебплатформи мінімальні технічні характеристики пристрою користувача повинні включати: наявність не менше 2 ГБ оперативної пам'яті, процесор із тактовою частотою не нижче 1.4 GHz, сучасний веббраузер із підтримкою технологій HTML5, CSS3 та JavaScript (рекомендовано використовувати останні версії Google Chrome, Mozilla Firefox або Microsoft Edge) стабільне підключення до мережі Інтернет зі швидкістю не менше 5 Мбіт/с. Веб-платформа повинна коректно відображатися та функціонувати як на стаціонарних комп'ютерах, так і на мобільних пристроях.

Серверна частина інформаційної системи повинна функціонувати на базі Linux-дистрибутива (рекомендовано Ubuntu Server 20.04 LTS або новішого), із встановленим вебсервером Apache 2.4 або Nginx останньої стабільної версії. Сервер повинен підтримувати PHP версії 8.0 або вище, а також базу даних MySQL 8.0 або сумісні рішення, такі як MariaDB. Для забезпечення безпечного обміну даними необхідна наявність налаштованого SSL-сертифікату та забезпечення підтримки протоколу HTTPS. З метою підвищення продуктивності платформи бажано використовувати технології кешування.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

Інформація про збори, заявки та активність користувачів має зберігатися у реляційній базі даних із забезпеченням цілісності даних та регулярним резервним копіюванням принаймні один раз на добу. Для коректної роботи серверу необхідна підтримка основних PHP-розширень, зокрема PDO, OpenSSL, Mbstring, Tokenizer, JSON та cURL. Також має бути реалізована можливість обробки фонових завдань через механізми черг, наприклад, за допомогою Laravel Queue у випадку використання фреймворку Laravel. Додатково рекомендується встановити системи моніторингу стану серверної інфраструктури для своєчасного виявлення та усунення потенційних збоїв.

2.2 Розробка бази даних

Розробка бази даних для веб-платформи для обговорення та оцінювання фільмів є важливою частиною створення системи. Вона забезпечує збереження та ефективний доступ до великих обсягів даних, які постійно змінюються. Основним завданням є проектування структури бази даних, яка забезпечить зберігання даних про фільми, користувачів, відгуки, рейтинги та інші пов'язані сутності.

Для розробки бази даних у даному проєкті було обрано реляційну модель бази даних, що дозволяє ефективно працювати з різними взаємозв'язками між таблицями та забезпечує високу продуктивність у обробці запитів.

База даних системи складається з 17 таблиць:

users - таблиця, що зберігає інформацію про користувачів платформи (ім'я, email, пароль, роль, аватар, тощо);

user_settings - таблиця, що містить розширені налаштування профілю користувача (посилання на соцмережі, фонове зображення, улюблені актори, режисери тощо);

films - таблиця, що зберігає основну інформацію про фільми (назва, опис, тривалість, рейтинг, дата виходу, бюджет, постер, мова);

reviews - таблиця, що містить текстові відгуки користувачів до фільмів з оцінкою;

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

ratings - таблиця, що зберігає числові рейтинги, які користувачі ставлять фільмам;

review_comments - таблиця, що містить коментарі до відгуків, включаючи вкладену структуру коментарів через parent_id;

likes - таблиця, що зберігає лайки або дизлайки до відгуків і коментарів; реалізована через поліморфний зв'язок;

messages - таблиця, що зберігає приватні повідомлення між користувачами з темами та статусом прочитання;

watchlist - таблиця, що містить список фільмів, які користувач бажає переглянути;

favorites - таблиця, що зберігає обрані фільми користувача;

actors - таблиця, що зберігає дані про акторів: ім'я, дату народження, біографію, фото, популярність;

film_actor - проміжна таблиця для зв'язку між фільмами та акторами, з інформацією про персонажа та порядок;

people - таблиця, що містить дані про персоналій знімальної групи (режисери, сценаристи, оператори тощо);

crew_members - проміжна таблиця, що пов'язує фільми з людьми, які працювали над фільмом (з вказанням посади);

categories - таблиця, що зберігає жанри фільмів (назва, TMDb-ідентифікатор);

category_film - проміжна таблиця для зв'язку фільмів і жанрів;

reports - таблиця, що містить скарги користувачів на відгуки або коментарі; реалізована як поліморфна.

Опис кожної з таблиць бази даних наведений у табл. 2.1 - 2.17.

Таблиця 2.1

Структура таблиці users

Поле	Ключ	Тип	Призначення
id	РК	bigint unsigned	Ідентифікатор користувача

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

Продовження табл. 2.1

name	—	varchar(255)	Ім'я користувача
email	—	varchar(255)	Email користувача
email_verified_at	—	timestamp	Дата підтвердження пошти
password	—	varchar(255)	Хеш пароля
role	—	enum	Роль (admin, moderator, user)
avatar	—	varchar(500)	Аватар користувача
remember_token	—	varchar(100)	Токен “запам'ятати мене”
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.2

Структура таблиці films

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор фільму
tmdb_id	—	bigint	ID фільму з TMDb
title	—	varchar(255)	Назва фільму
overview	—	text	Опис
runtime	—	int	Тривалість
budget	—	bigint	Бюджет

Продовження табл. 2.2

revenue	—	bigint	Касові збори
poster_path	—	varchar(255)	Посилання на постер
backdrop_path	—	varchar(500)	Посилання на фон
release_date	—	date	Дата релізу
vote_average	—	double	Середній рейтинг
vote_count	—	int	Кількість голосів
original_language	—	varchar(255)	Оригінальна мова
production_companies	—	text	Компанії-виробники
production_countries	—	text	Країни-виробники
pg_rating	—	varchar(10)	Віковий рейтинг
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.3

Структура таблиці reviews

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор відгука
user_id	FK	bigint unsigned	Ідентифікатор користувача
film_id	FK	bigint unsigned	Ідентифікатор фільму

Продовження табл. 2.3

content	—	text	Текст відгука
rating	—	tinyint(4)	Оцінка
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.4

Структура таблиці ratings

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор
user_id	FK	bigint unsigned	Користувач
film_id	FK	bigint unsigned	Фільм
rating	—	tinyint(4)	Оцінка
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.5

Структура таблиці watchlist

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор

user_id	FK	bigint unsigned	Користувач
film_id	FK	bigint unsigned	Фільм
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.6

Структура таблиці likes

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор
user_id	FK	bigint unsigned	Користувач, що поставив лайк/дизлайк
likeable_id	—	bigint unsigned	ID об'єкта, до якого застосовано лайк
likeable_type	—	varchar(255)	Тип моделі (наприклад, Review або Comment)
is_like	—	tinyint(1)	1 – лайк, 0 – дизлайк
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.7

Структура таблиці messages

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор повідомлення

Продовження табл. 2.7

sender_id	FK	bigint unsigned	Відправник
receiver_id	FK	bigint unsigned	Отримувач
content	—	text	Текст повідомлення
read	—	tinyint(1)	Ознака прочитання
theme	—	varchar(255)	Тема повідомлення
created_at	—	timestamp	Дата надсилання
updated_at	—	timestamp	Дата оновлення

Таблиця 2.8

Структура таблиці review_comments

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор
user_id	FK	bigint unsigned	Користувач
review_id	FK	bigint unsigned	Відгук, до якої додано коментар
parent_id	—	int	ID батьківського коментаря (для вкладеності)
content	—	text	Текст коментаря
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.9

Структура таблиці actors

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор актора
tmdb_id	—	bigint	ID з TMDb
name	—	varchar(255)	Ім'я
birth_date	—	date	Дата народження
death_date	—	date	Дата смерті (якщо є)
place_of_birth	—	varchar(255)	Місце народження
photo	—	varchar(255)	URL фото
popularity	—	double	Популярність
biography	—	text	Біографія
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.10

Структура таблиці crew_members

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор
person_id	FK	bigint unsigned	ID персони (режисер, сценарист...)

Продовження табл. 2.10

film_id	FK	bigint unsigned	ID фільму
job	—	varchar(100)	Посада (наприклад, режисер)
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.11

Структура таблиці categories

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор жанру
tmdb_id	—	int(11)	ID з TMDb
name	—	varchar(255)	Назва жанру
slug	—	varchar(255)	URL-ідентифікатор (наприклад, "comedy")
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.12

Структура таблиці category_film

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор
film_id	FK	bigint unsigned	ID фільму

Продовження табл. 2.12

category_id	FK	bigint unsigned	ID жанру
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.13

Структура таблиці film_actor

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор
film_id	FK	bigint unsigned	ID фільму
actor_id	FK	bigint unsigned	ID актора
character	—	varchar(255)	Ім'я персонажа
order	—	int(11)	Порядок у титрах
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.14

Структура таблиці favorites

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор
user_id	FK	bigint unsigned	ID користувача

Продовження табл. 2.14

film_id	FK	bigint unsigned	ID фільму
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.15

Структура таблиці people

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор персони
tmdb_id	—	bigint unsigned	ID з TMDb
name	—	varchar(255)	Ім'я
profile_path	—	varchar(255)	Посилання на зображення
biography	—	text	Біографія
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

Таблиця 2.16

Структура таблиці reports

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор скарги
user_id	FK	bigint unsigned	Користувач, що подав скаргу

Продовження табл. 2.16

reportable_id	–	bigint unsigned	ID об'єкта скарги
reportable_type	–	varchar(255)	Тип об'єкта (Review, Comment тощо)
reason	–	varchar(255)	Причина скарги
details	–	text	Додаткові деталі
status	–	enum	Статус скарги ('pending', 'reviewed', 'dismissed')
created_at	–	timestamp	Дата створення
updated_at	–	timestamp	Дата оновлення

Таблиця 2.17

Структура таблиці user_settings

Поле	Ключ	Тип	Призначення
id	PK	bigint unsigned	Ідентифікатор
user_id	FK	bigint unsigned	ID користувача
motto	–	varchar(255)	Девіз профілю
instagram_url	–	varchar(255)	Посилання на Instagram
youtube_url	–	varchar(255)	Посилання на YouTube
background_url	–	varchar(255)	Посилання на фонове зображення
favorite_birth_year_movie	–	varchar(255)	Улюблений фільм року народження
favorite_character	–	varchar(255)	Улюблений персонаж

favorite_actor	—	varchar(255)	Улюблений актор
favorite_director	—	varchar(255)	Улюблений режисер
twitter_url	—	varchar(255)	Посилання на Twitter (X)
created_at	—	timestamp	Дата створення
updated_at	—	timestamp	Дата оновлення

На рис. 2.3 зображено основні сутності бази даних та їх атрибути.

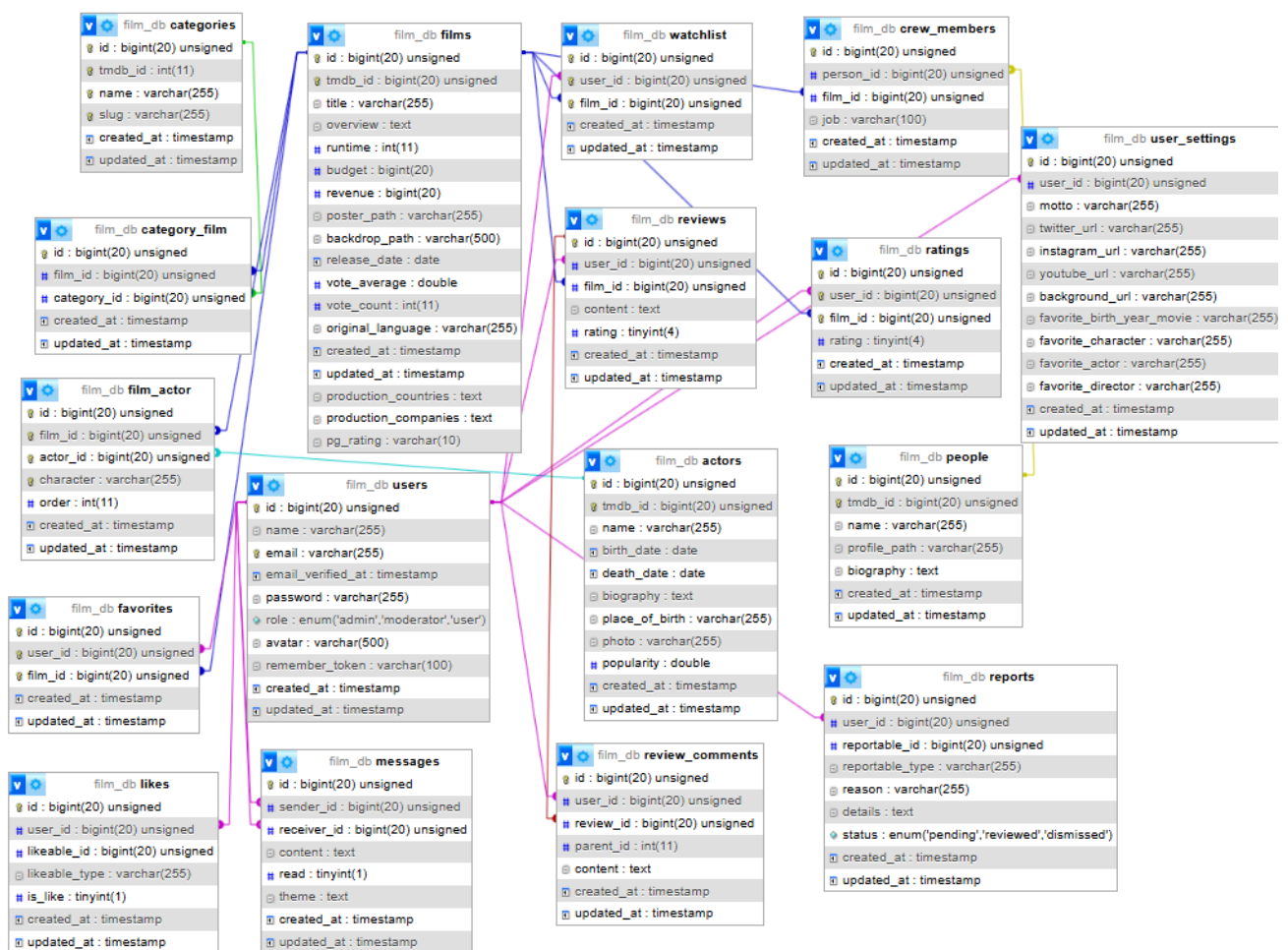


Рисунок 2.3 – ER - діаграма бази даних

Запити до бази даних: для роботи з базою даних будуть використовуватись SQL-запити, які дозволяють виконувати основні операції CRUD. Безпека бази даних: для забезпечення безпеки бази даних важливим аспектом є шифрування

паролів користувачів перед їх збереженням; впровадження механізмів аутентифікації та авторизації для доступу до різних частин системи; використання підготовлених запитів та заходів захисту від SQL-ін'єкцій.

Масштабованість бази даних: згодом, коли система буде зростати, може виникнути потреба в масштабуванні бази даних. Також важливо забезпечити ефективне кешування часто запитуваних даних, таких як рейтинги та популярні фільми.

2.3 Проектування та реалізація алгоритмів роботи платформи

Застосунок є простим у використанні, має всі основні функції для обговорення та оцінювання фільмів, що дозволяє користувачам легко взаємодіяти з платформою. Алгоритм роботи з платформою для звичайного користувача, неавторизованого, адміністратора та модератора зображено на рис. 2.4, рис. 2.5 та рис. 2.6, а також детально описано їх етапи.

Якщо авторизувався адміністратор, то після авторизації адміністратор переходить до адміністративної панелі; адміністратор має можливість переглядати репорти або скарги на відгуки, коментарі, користувачів і змінює або видаляє відповідне порушення; адміністратор управляє категоріями фільмів та самим фільмами відповідно: створює, редагує та видаляє їх; адміністратор здійснює керування користувачами: змінює ролі, змінює або видаляє облікові записи; адміністратор переглядає загальну статистику роботи платформи. Окрім цього, адміністратор також має доступ до людей, які працюють над фільмом та має змогу редагувати або видаляти їх.

Якщо авторизувався модератор, то після авторизації модератор має можливість перейти до панелі модератора; модератор може імпортувати фільми і змінювати їх при потребі перед імпортом як; модератор переглядає статистику активності користувачів; модератор має право перейти до списку репортів, переглянути деталі скарги та змінити статус репорта при потребі або взагалі видалити його зі загального списку, якщо є застарілим або недоречним.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

Якщо платформою користується гість, то гість має можливість переглядати загальний список фільмів, відгуків та оцінок без реєстрації, гість не має змоги залишати відгуки або оцінки без авторизації.

Якщо авторизувався зареєстрований користувач, то користувач переглядає список фільмів та відгуків, особливо свої; залишає власні відгуки, коментарі, лайки, дизлайки та рейтинги; редагує або видаляє свої коментарі, відгуки; переглядає статистику власної активності, має можливість кастомізувати свій профіль.



Рисунок 2.4 - Діаграма активності користувачів

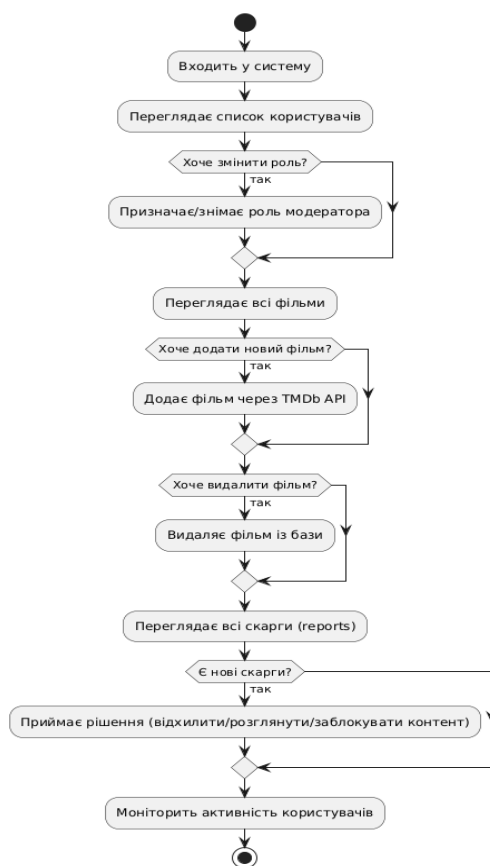


Рисунок 2.5 - Діаграма активності адміністратора

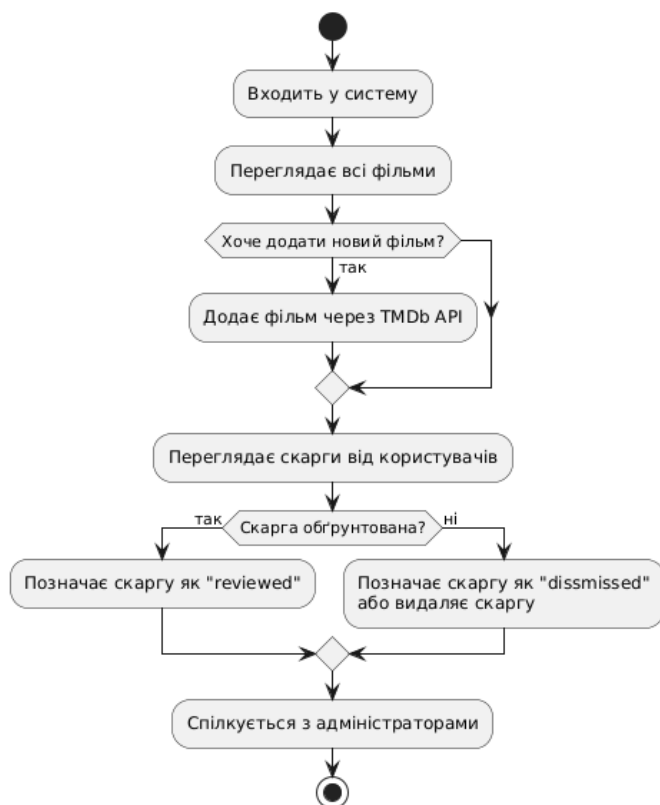


Рисунок 2.6 - Діаграма активності модератора

Також на рис. 2.5 і рис. 2.6 можна помітити, що модератор і адміністратор мають спільні активності, такі як імпорт або зміна фільмів перед імпортом на сайт та перегляд сторінки з репортами та маніпуляція з цими репортами (зміна статусу або видалення при необхідності).

2.4 Реалізація функціоналу платформи

Користувач для взаємодії із платформою спочатку проходить реєстрацію нового акаунту. Після цього він отримує можливість створювати нові відгуки на фільми. Функціонал створення відгуків реалізований через окрему форму на платформі

За інтеграцію з зовнішнім API The Movie Database (TMDb) відповідає представлено клас TmdbService (Додаток А). Цей клас реалізовано за патерном Singleton для забезпечення єдиного екземпляру під час виконання. Основне його призначення - централізовано отримувати дані про фільми, акторів, жанри та деталі конкретного фільму. Реалізовано механізми кешування відповідей (наприклад, окремі фільми кешуються на одну годину), що зменшує кількість зовнішніх запитів і підвищує ефективність. Також реалізовано обробку помилок, логування при невдалих відповідях API, та обробку популярних фільмів і акторів по сторінково. Таким чином, цей клас є ключовим елементом підсистеми імпорту даних із TMDb у платформу.

Реалізує логіку формування персоналізованих рекомендацій фільмів для користувачів контролер RecommendationController (Додаток Б). У процесі обробки враховуються улюблені жанри, країни, режисери, короткометражки, новинки, а також невідомі, але високо оцінені фільми. Джерелом даних слугують відгук користувача з високими оцінками, а результати фільтруються так, щоб виключити вже переглянуті фільми. Контролер формує набір категорій рекомендацій і передає їх у відповідне представлення. Це один із найважливіших функціональних модулів, що підвищує зацікавленість користувача і дає йому нові пропозиції на основі його кінематографічних уподобань або щось нове для ознайомлення.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

За роботу адміністратора з фільмами відповідає AdminController (Додаток В). Контролер дозволяє шукати фільми через API TMDb, переглядати їх детальну інформацію (включно з кастом і знімальною групою), а також імпортувати обрані фільми до бази даних платформи. Під час імпорту виконується збереження жанрів, акторів та режисерів, які обираються вручну. Крім того, реалізовано обробку некоректних значень, обмеження на негативні бюджети чи прибутки, логування помилок та перевірку прав доступу. Таким чином, цей модуль дає змогу адміністратору гнучко керувати наповненням платформи, забезпечуючи якісний і актуальний контент. Також цей контролер є ключовим для платформи, оскільки окрім продемонстрованих методів по наповненню даними, відбувається також контроль майже кожного елемента, наприклад, зміна або видалення неправильно імпортованих елементів, керування та спостереження за користувачами та контентом, який вони додають та приймання рішень на скарги.

Реалізує механізм виставлення оцінок фільмам користувачами контролер RatingController (Додаток Г). Контролер дозволяє створити або оновити оцінку, пов'язану з конкретним фільмом, а також при наявності - синхронізувати її з уже існуючим текстовим відгуком. Після кожного виставлення рейтингу оновлюється середня оцінка фільму (vote_average) та кількість голосів (vote_count). Крім того, реалізовано REST-метод для отримання переліку оцінок з фільтрами, що дозволяє використовувати ці дані в API або асинхронних запитах. Такий функціонал є важливою складовою інтерактивності платформи, оскільки дозволяє кожному користувачеві впливати на формування рейтингу фільмів та системи рекомендацій.

На основі розробленого функціоналу було створено діаграму основних компонентів платформи, яку проілюстровано на рис. 2.7.

Клієнтська частина (Frontend) — включає Blade-шаблони, JavaScript, TailwindCSS та Alpine.js. Вона відповідає за відображення інтерфейсу та взаємодію користувача з платформою.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

Серверна частина (Backend) — обробляє запити через маршрути, контролери та сервіси. Вона працює з моделями, автентифікацією, файловим сховищем і сервісом для взаємодії з TMDb API.

Зовнішні сервіси — база даних MySQL для зберігання даних та зовнішній TMDb API для отримання інформації про фільми.

Усі компоненти логічно пов'язані між собою — від інтерфейсу до бази даних і стороннього API. Вся система побудована із забезпеченням розділення відповідальностей: користувачі створюють контент, адміністрація та модерація здійснює контроль усього, а відгуки та оцінки доступні для перегляду та аналітики у реальному часі. Це дозволяє платформі залишатися відкритою, прозорою і одночасно забезпечувати якісний контент.

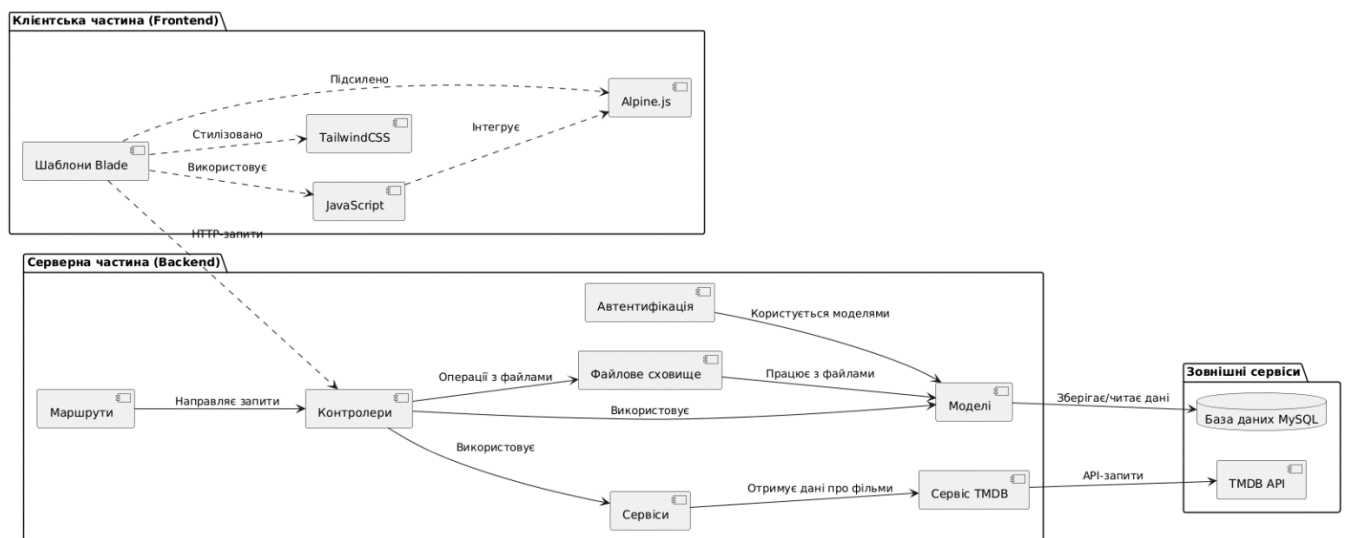


Рисунок 2.7 - Діаграма компонентів

Висновки до другого розділу

У другому розділі було здійснено проєктування веб-платформи для обговорення та оцінювання фільмів. Розглянуто основні варіанти використання системи, визначено ролі користувачів, а також розроблено об'єктно-орієнтовану структуру, що забезпечує логічний поділ функціональності платформи. Було спроектовано базу даних, яка охоплює всі необхідні сутності для ефективного збереження інформації про користувачів, фільми, відгуки, коментарі та оцінки. Також проведено проєктування та реалізацію основних алгоритмів роботи

платформи, що регламентують дії користувачів та адміністратора. Окрема увага була приділена створенню механізмів перевірки, модерації контенту та формуванню статистичних звітів.

В результаті було реалізовано функціонал, який дозволяє користувачам додавати відгуки та рейтинги, переглядати фільми, брати участь в обговореннях, а адміністраторам — здійснювати контроль за вмістом та користувачами платформи.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

З ІНТЕРФЕЙС, ПОРЯДОК РОБОТИ ТА ТЕСТУВАННЯ ПЛАТФОРМИ ДЛЯ ОБГОВОРЕННЯ ТА ОЦІНЮВАННЯ ФІЛЬМІВ

3.1 Порядок встановлення та налаштування параметрів системи

Розгортання веб-платформи для обговорення та оцінювання фільмів потребує попередньої підготовки відповідного програмного середовища, у якому система зможе функціонувати стабільно, безпечно та з високою продуктивністю. Реалізована система побудована на основі PHP-фреймворку Laravel, що забезпечує структуровану організацію коду та підтримку шаблонів проєктування.

Візуальна частина реалізована за допомогою Blade-шаблонізатора та CSS-фреймворку TailwindCSS, а інтерактивна поведінка елементів забезпечується бібліотекою JavaScript. Окрім них, було ще використано Alpine.js для взаємодії з елементами та відображення без перезавантаження сторінки [3]. Alpine.js ідеально підходить для простих реактивних компонентів, таких як випадаючі меню, перемикачі теми або валідація форм. Збереження даних виконується за допомогою системи керування базами даних MySQL. Керування залежностями відбувається через Composer [7]. Всі ці інструменти вимагають відповідної конфігурації перед запуском системи.

Для локального запуску системи використовувалося серверне середовище XAMPP, що включає у свій склад Apache, PHP та MySQL. Для візуального перегляду та маніпуляцій з базою даних було використана функція XAMPP - phpMyAdmin. Це забезпечило можливість швидкого розгортання застосунку без додаткового налаштування кожного з компонентів окремо. Сервіс Apache забезпечував доступ до інтерфейсу веб-застосунку, тоді як MySQL використовувався як система управління базами даних для зберігання відгуків, користувачів та фільмів.

Першочерговим етапом є підготовка середовища, а саме потрібні нам версії різних компонентів програми, яка відповідає вимогам, наведеним у табл. 3.1.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

Версії компонентів платформи

Компонент	Мінімальна версія	Призначення
PHP	8.4	Серверна логіка Laravel
Composer	2.0	Менеджер залежностей
Node.js	18.0	Компіляція ресурсів TailwindCSS
MySQL	8.0	Система керування базами даних
Laravel Framework	12	Структура проєкту і логіка
Blade	—	Шаблонізація сторінок
TailwindCSS	3	Стилізування інтерфейсу
Alpine.js	3	Інтерактивність без важкого JS-фреймворку

У табл. 3.2 можна переглянути усі важливі змінні, які можна знайти у `.env` файлі в корні проєкту. Після налаштування `.env`-файлу генерується ключ шифрування застосунку командою `php artisan key:generate`.

Далі виконується створення структури бази даних за допомогою `php artisan migrate`, яка автоматично створює таблиці: `users`, `films`, `reviews`, `comments`, `likes`, `reports`, `review_comments` і т.д.. Для початкового заповнення тестовими даними може використовуватись `php artisan db:seed`, щоб спростити ранній тестинг функцій.

Задля зменшення кількості запитів до TMDb API (витягування фільмів, акторів і інших даних) реалізовано кешування. При першому зверненні до ресурсу дані зберігаються в кеші, а при повторному - витягуються локально. Кеш керується засобами Laravel через можливість `Cache::remember`.

Змінні файлу .env

Параметр	Значення (приклад)	Опис
APP_NAME	MoviePlatform	Назва системи
APP_URL	<u>http://localhost</u>	URL застосунку
DB_CONNECTION	mysql	Тип бази даних
DB_HOST	127.0.0.1	Локальний хост
DB_PORT	3306	Порт MySQL
DB_DATABASE	Films_db	Назва бази даних
DB_USERNAME	root	Користувач MySQL
DB_PASSWORD	password	Пароль до бази
TMDB_API_KEY	tmdb_api_key_here	Ключ доступу до TMDB API

Усі події, виключення та помилки журналюються у файл storage/logs/laravel.log, що дозволяє виявити потенційні проблеми при роботі системи. Файл сам по собі являється дуже важливим на стадії тестингу, оскільки часто вказує на корінь помилки.

На рис. 3.1 зображено процес розгортання системи.

Клієнтська частина - здійснюється через веб-браузер, який надсилає HTTP-запити до сервера та відображає інтерфейс. Шаблони формуються через Blade, а стилізація - за допомогою TailwindCSS. Інтерактивність реалізовано JavaScript та бібліотекою Alpine.js.'

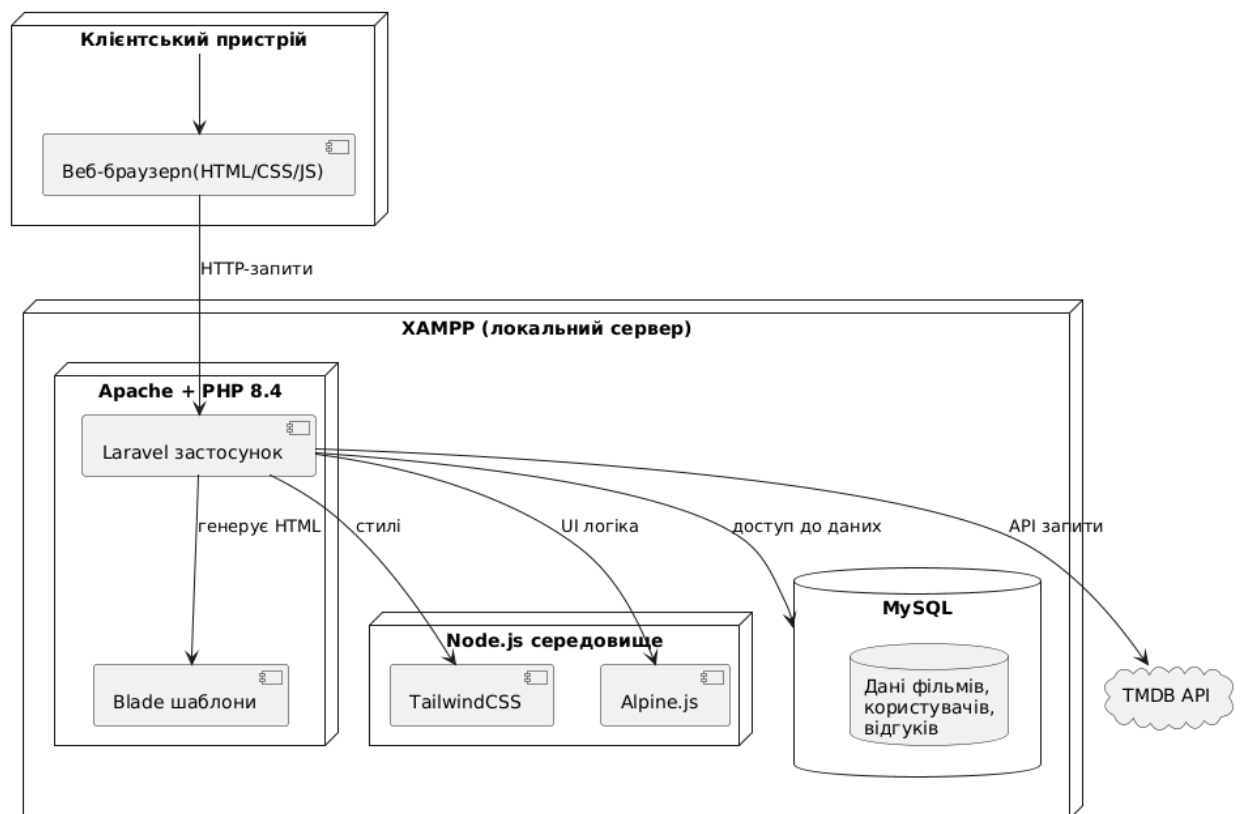


Рисунок 3.1 - Діаграма розгортання

3.2 Структура інтерфейсу платформи для обговорення та оцінювання фільмів

Інтерфейс розробленої системи спроектовано таким чином, щоб забезпечити максимальну зручність взаємодії для користувача та водночас підтримувати майбутнє розширення функціональності. Візуальна частина реалізована з використанням Blade-шаблонів, що дозволило створити чітку структуру сторінок, тоді як TailwindCSS відповідає за гнучке та стильне оформлення інтерфейсу. Для реалізації динамічної поведінки елементів було використано JavaScript бібліотеку Alpine.js.

На рис. 3.2 представлена головна сторінка платформи, яка містить навігаційний таб, який буде супроводжувати по всіх сторінках разом з футером, а також головний фільм на сторінці, який буде різним при кожному оновленні (обирається рандомно з найбільших кількостей голосів і оцінка мінімум сім). В табі у нас присутній пошук по фільмах та акторах, вкладка фільми з усіма фільмами,

вкладка про нас, на якій присутня статистика та основна інформація і можливості увійти або зареєструватися.

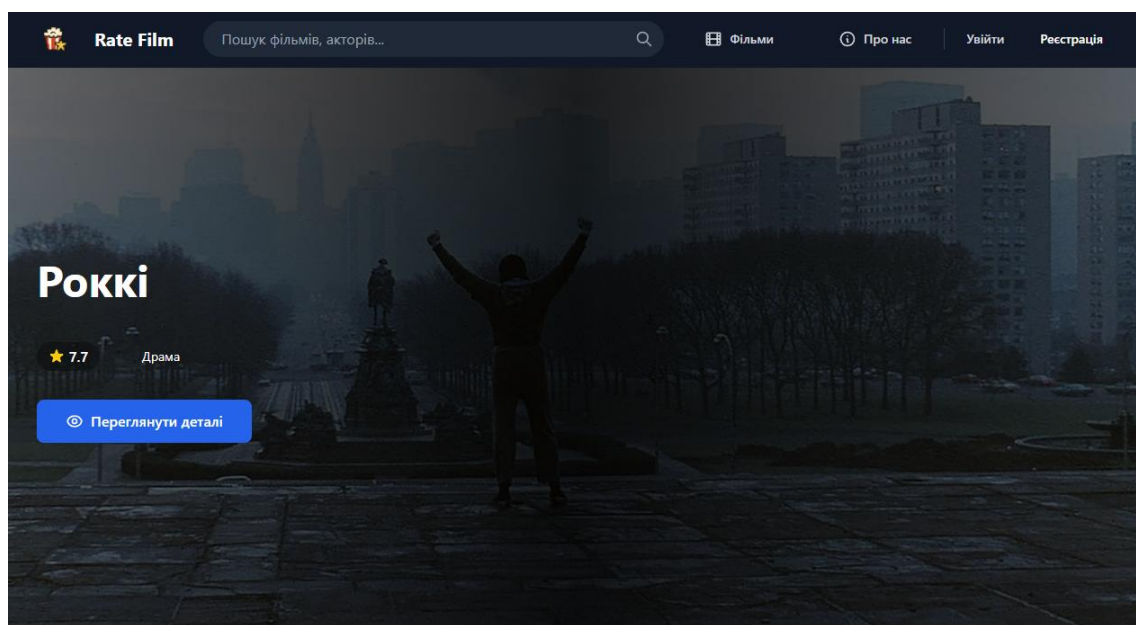


Рисунок 3.2 - Головна сторінка з головним фільмом

Далі, на рис. 3.3 проілюстровано продовження головної сторінки, де можна побачити нещодавно додані фільми з можливістю перейти на всі фільми та можливість побачити всі жанри, використовуючи слайдер. Якщо натиснути на будь-який жанр або фільм перейдемо на сторінки фільмів з цих жанрів або на сторінку фільму відповідно.

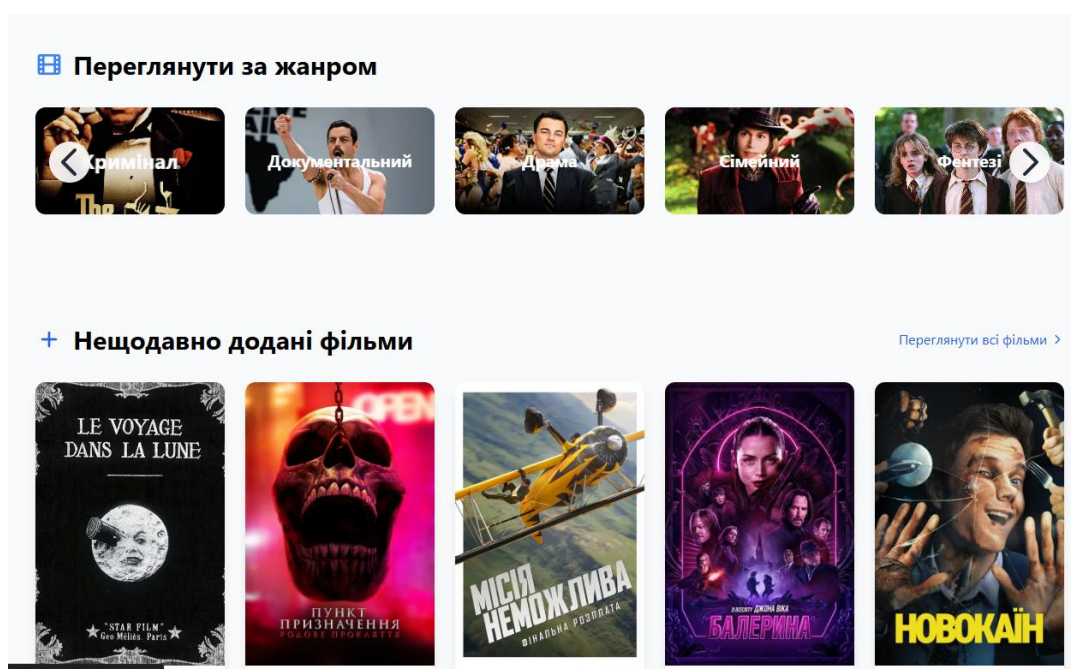


Рисунок 3.3 - Продовження головної сторінки

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

На рис. 3.4 представлено останню частину головної сторінки з відгуками та футером. З футером не можна інтерактувати, а з відгуками є такі функції, як можливість перейти на профіль того, хто написав відгук або перейти на фільм на який було написано відгук.

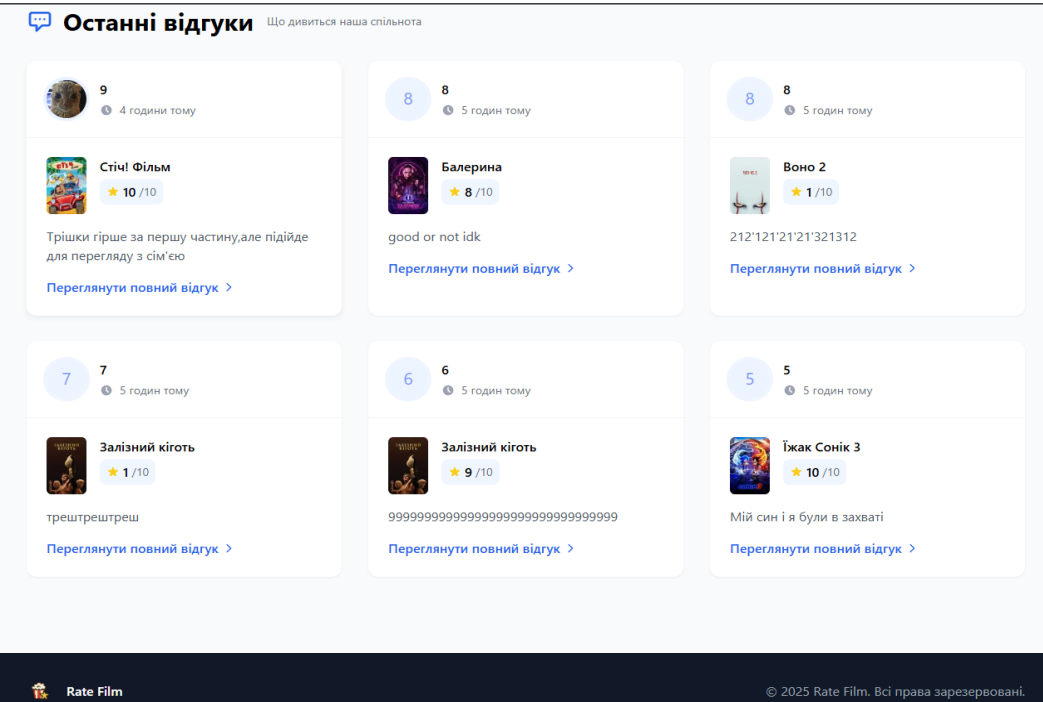


Рисунок 3.4 - Продовження головної сторінки з відгуками та футером

На рис. 3.5 та рис. 3.6 показані сторінки з навігаційного табу - логін та реєстрація акаунта. Пароль має бути мінімум 8 символів.

Створити обліковий запис

Приєднуйтесь до нашої спільноти кіноманів

Ім'я

Email адреса

Пароль

Підтвердження пароля

Зареєструватися

Вже маєте обліковий запис? [Увійти](#)

Рисунок 3.5 - Сторінка реєстрації

Ласкаво просимо назад

Увійдіть до свого облікового запису

Email адреса

Пароль

Увійти

Немаєте облікового запису? [Зареєструватися](#)

Рисунок 3.6 - Сторінка входу

Після того як в акаунт було здійснено вхід, вкладки логіна і реєстрації зникають, натомість з'являється ім'я, картинка профілю та роль, якщо така є. Всі різниці між ролями відображено на рис. 3.7, рис. 3.8 та рис. 3.9.

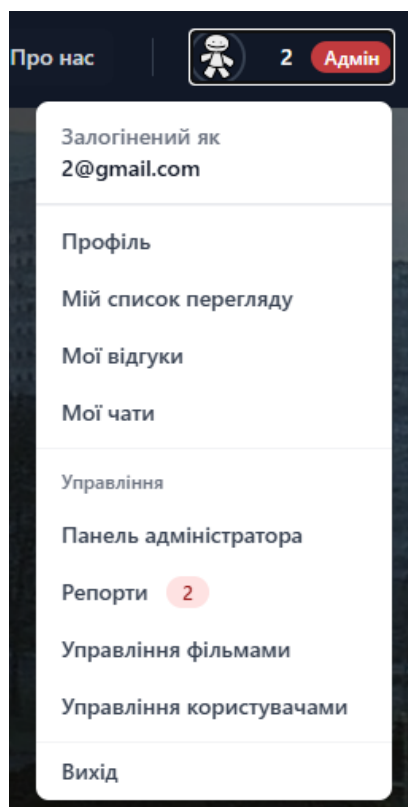


Рисунок 3.7 - Можливості адміна

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

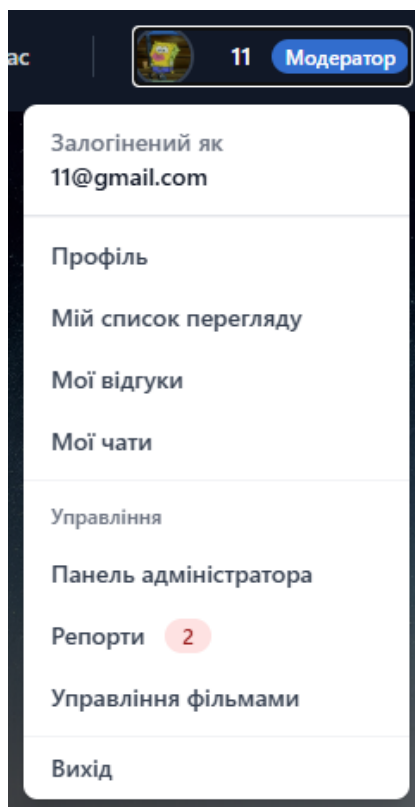


Рисунок 3.8 - Можливості модератора

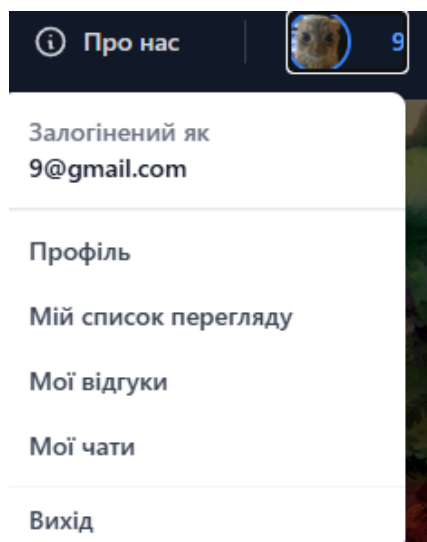


Рисунок 3.9 - Можливості звичайного користувача

На рис. 3.10 проілюстровано нову вкладку із одним із декілька персоналізованих списків, який з'являється після входу в акаунт - Рекомендації. На сторінці присутні декілька списків, більшість з яких не є сталими та змінюються в залежності від декількох факторів - від дати виходу фільмів до оцінок користувача. Один із списків завжди буде рандомним набором неоцінених фільмів користувача з назвою Фільми, які можуть сподобатися.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

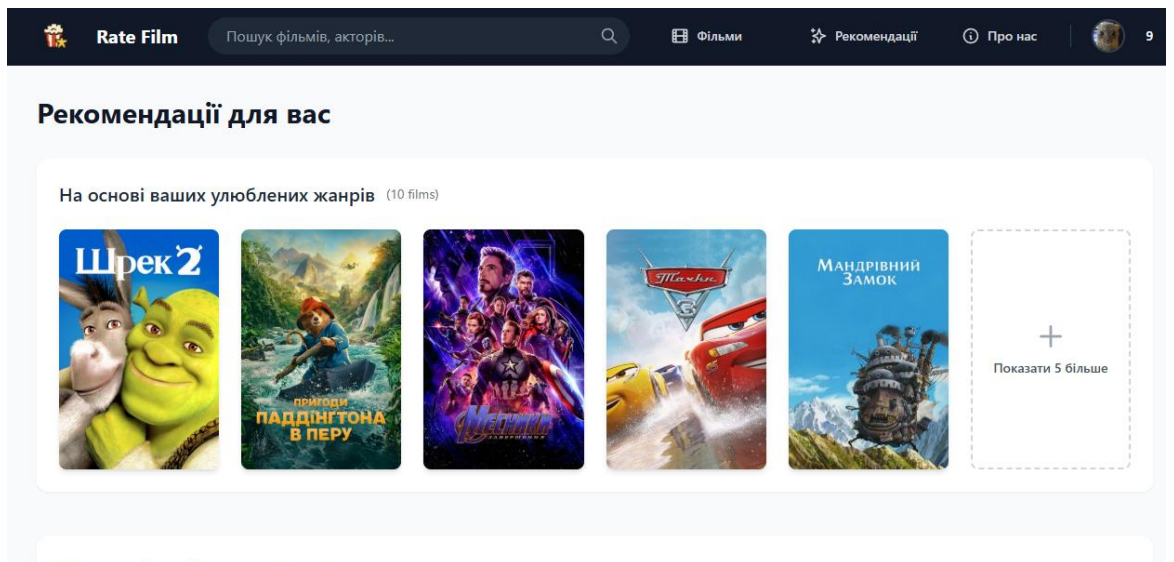


Рисунок 3.10 - Нова вкладка

Вкладка Про нас, яка проілюстрована на рис. 3.11, є простою сторінкою з статистикою і невеликою інформацією про сайт.

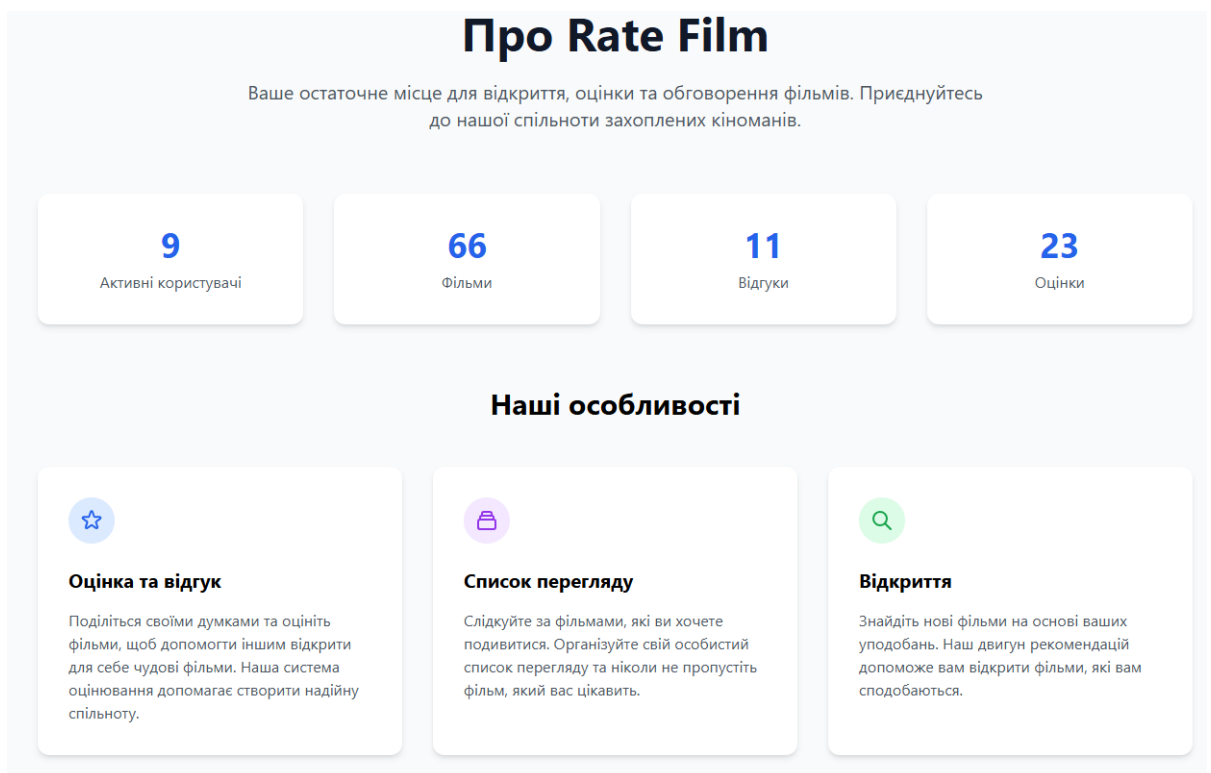


Рисунок 3.11 - Вкладка Про нас

На рис. 3.12 відображена сторінка із фільмами з можливістю здійснити фільтр по жанрам, мовам, рокам та рейтингом з опцією просортувати за алфавітом, датою виходу та ін. Натиснувши на фільм, можливо перейти на його сторінку.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

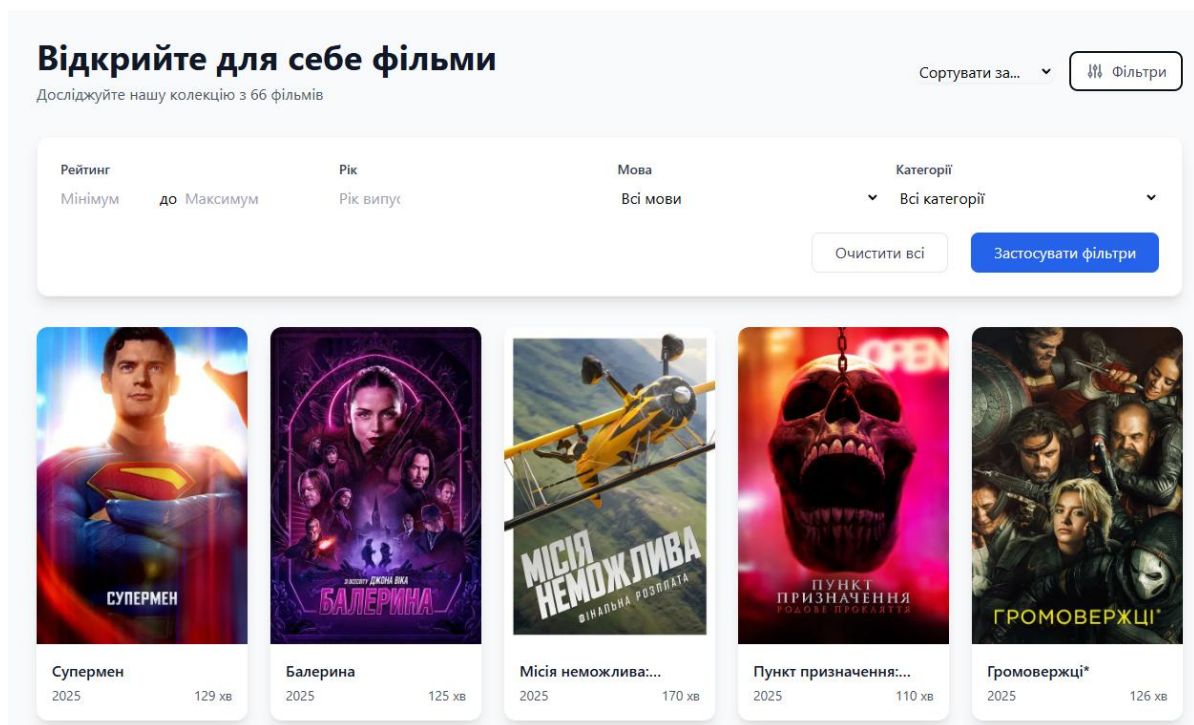


Рисунок 3.12 - Вкладка Фільми

На рис. 3.13 зображена загальна інформація про фільм: постер, жанри, на які можна перейти, бюджет, дата виходу, скільки заробив фільм (якщо заробив), короткий опис, рейтинг з можливістю оцінити, додати або прибрати з списку перегляду.

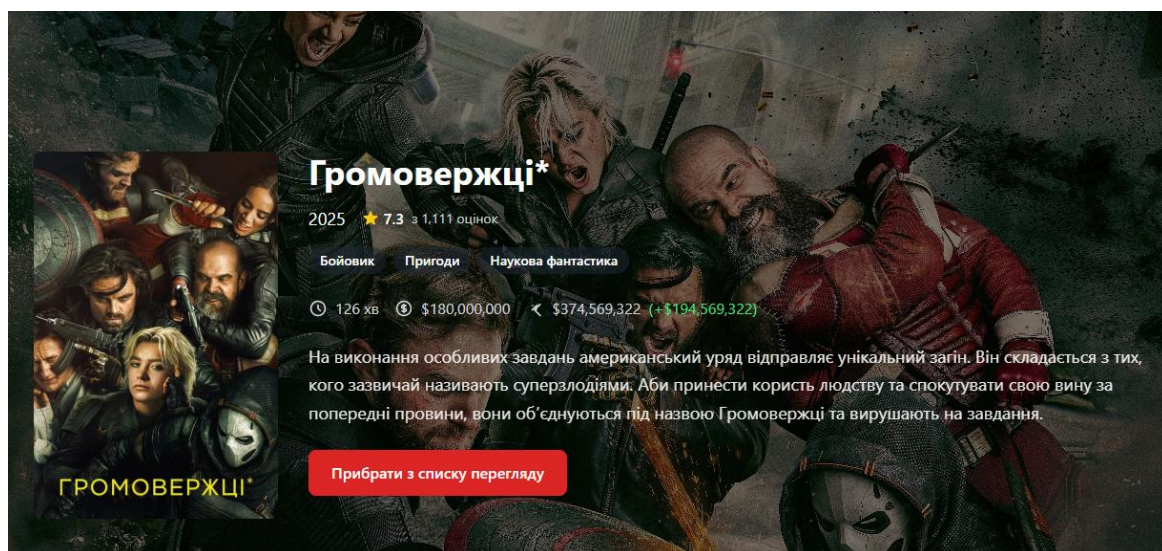


Рисунок 3.13 - Інформаційна частина фільма

Пролистнувши вниз, з'являється новий блок із деталями фільму, як проілюстровано на рис. 3.14. Натиснувши на будь-якого члена ключової команди, можна перейти на сторінку із біографією та фільмами в яких була взята участь.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

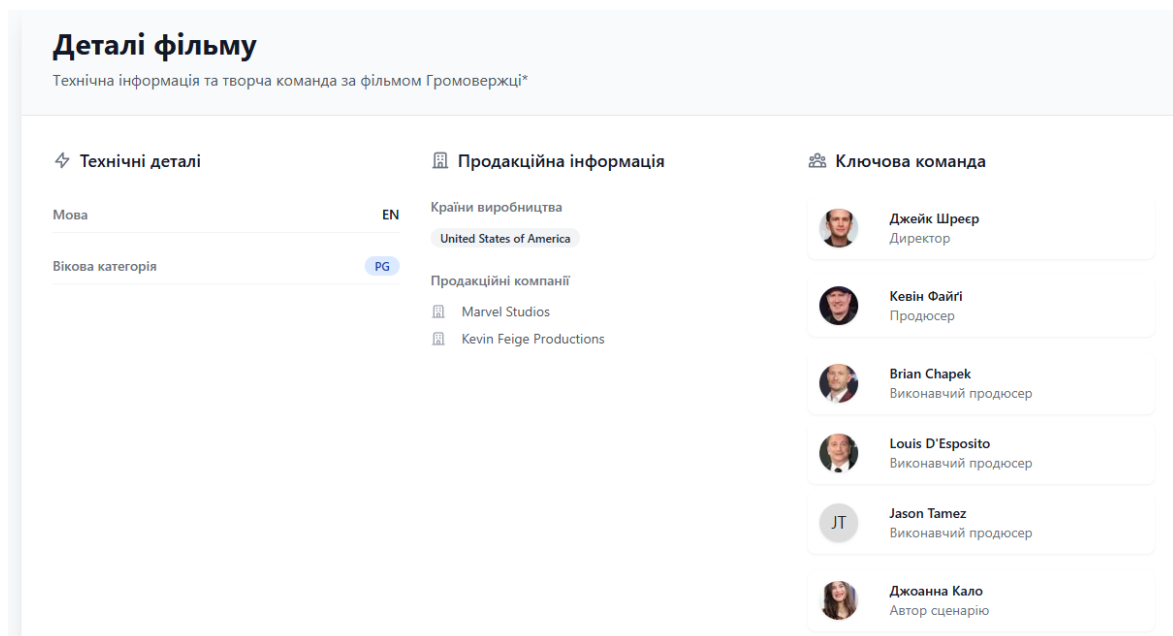


Рисунок 3.14 - Частина із деталями фільму

На рис. 3.15 відображено офіційний трейлер фільму та офіційні джерела, де є можливість подивитися фільм, якщо джерела існують. Інформація про трейлер не зберігається в базі даних, а береться інформація про нього з API, використовуючи збережений ідентифікатор фільма в БД.

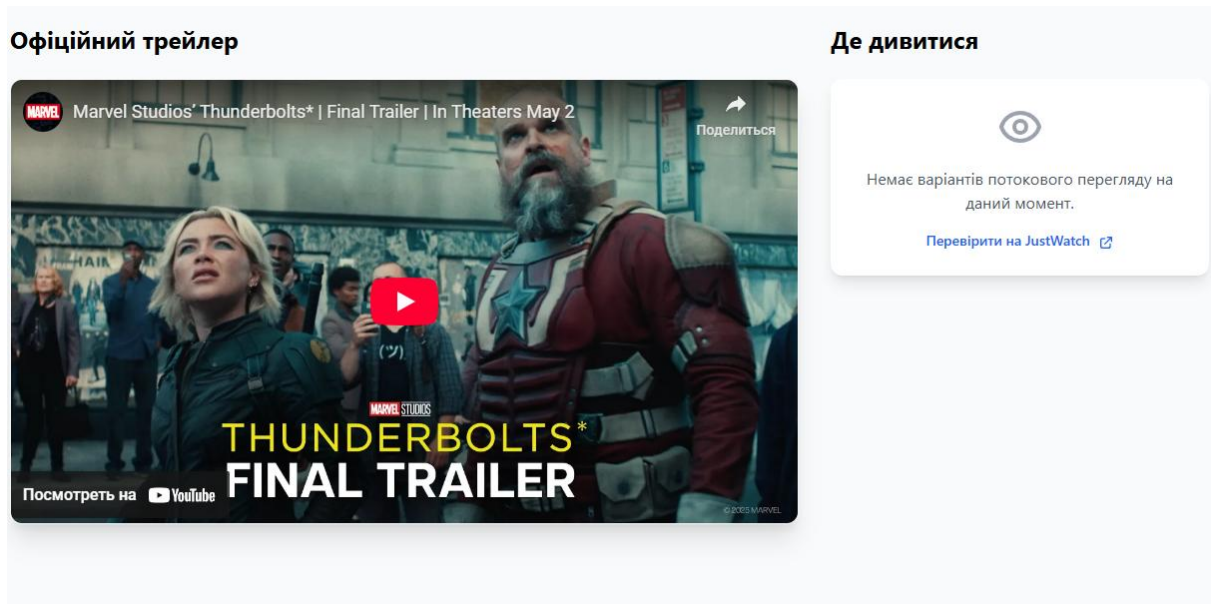


Рисунок 3.15 - Частина із трейлером та джерелами

На рис. 3.16 зображено частину із найпопулярнішими відгуками та відгуком нинішнього залогіненого користувача. Якщо користувач не зробив ще відгук для цього фільму, в нього показано форму для створення відгука.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

Відгуки

Найкращі відгуки на основі вподобань та коментарів

Дивитися всі відгуки

Ваша оцінка

Оберіть оцінку

Ваш відгук

Поділіться своїми думками про фільм (мінімум 10 символів)...

Мінімум 10 символів обов'язково

Надіслати відгук

Немає відгуків для цього фільму.

Рисунок 3.16 - Частина із відгуками для фільма

Остання частина сторінки фільму - головні актори, яких зображено на рис. 3.17. Якщо натиснути на будь-якого, можна подивитися сторінку із інформацією про актора: дата народження, біографія, фільмографія та дата смерті, якщо актор помер. Проілюстрована сторінка на рис. 3.18. Якщо натиснути будь-який фільм із списку в розділі Фільмографія, то можливо буде перейти на сторінку цього фільму.

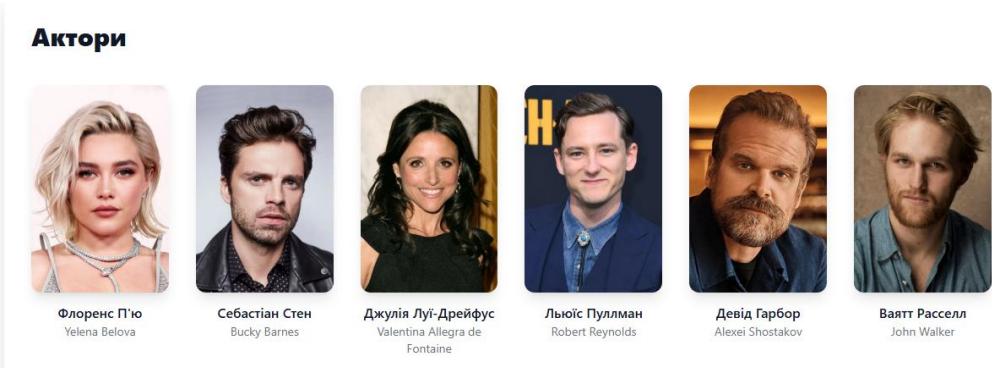


Рисунок 3.17 - Частина із акторами фільму

НАРОДИВСЯ

16 жовтня 1958

Вест-Ковіна, Каліфорнія, США

Вік: 66 років

Біографія

Тім Роббінс народився у Вест-Ковіні (штат Каліфорнія, США) і виховувався у католицькій та надзвичайно побожній родині. Його батьками були Мері Сеселія (до шлюбу Бледсо), музикантка, і Гіберт Лі Роббінс, співак, актор і менеджер The Gaslight Cafe. У Роббінса є дві сестри, Аделі і Габріель, і брат, композитор Девід Роббінс. Вирвавшись нарешті на свободу та почавши вчитися в університеті штату Нью-Йорк, Роббінс перші два роки занять перетворив на одну суцільну вечірку. Кінокар'єру починав з епізодичних ролей у телесеріалах. Помітні ролі у кінофільмах почав отримувати з другої половини 1980-х років. Можливо, кульмінацією його кінокар'єри була головна роль у пригодницькій драмі «Втеча з Шоушенка» (1994). 2004 року удостоєний премії «Оскар» за найкращу роль другого плану (фільм «Тамничка ріка», реж. Клінт Іствуд). Два роки потому знявся у Стівена Спілберга у «Війні світів». Відомий антиглобаліст.

Фільмографія

1 фільмів

Рисунок 3.18 - Сторінка актора

На рис. 3.19 зображено, як мають виглядати відгуки, а саме оцінка, текст відгука, лайки та дизлайки, коментарі, можливість відповісти на коментар або на відгук та дві точки біля кожного із коментарів та відгука. Натиснувши на дві точки, відкриється меню: якщо це елемент, який додав нинішній користувач, то буде можливість видалити або змінити контент елемента, якщо це елемент іншого користувача, то можливість поскаржитися.

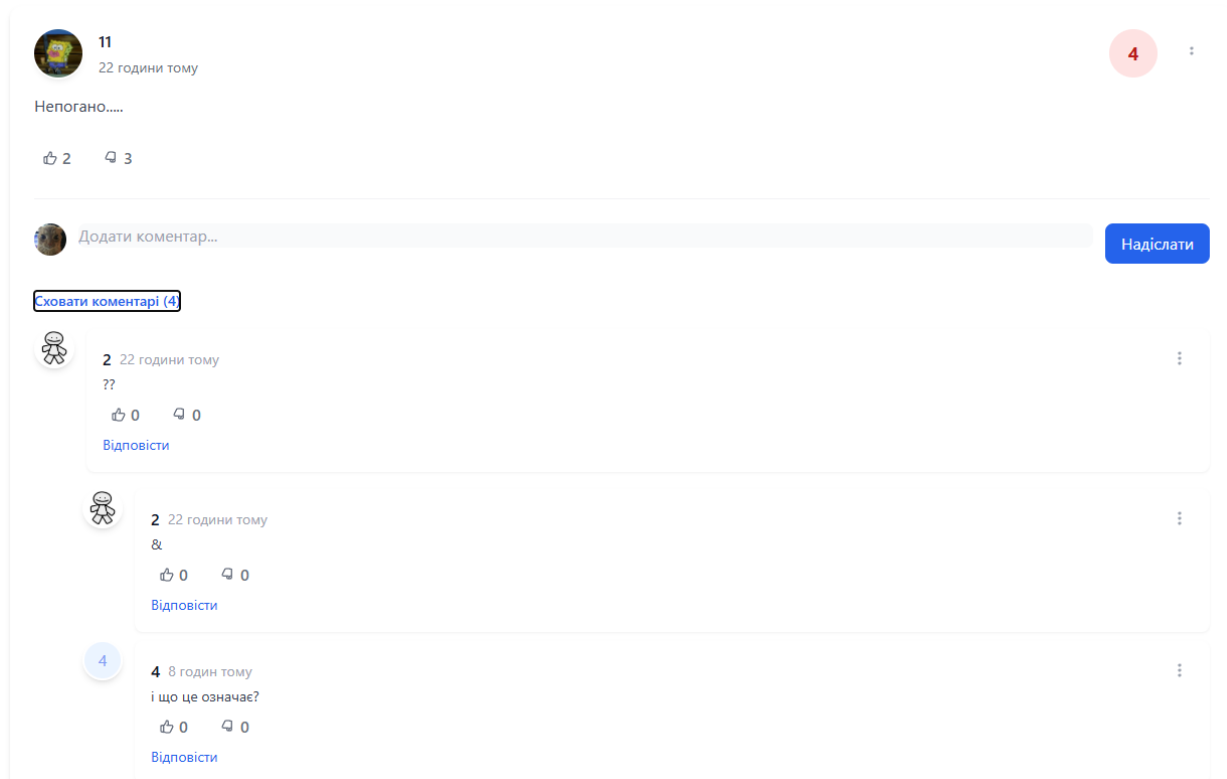


Рисунок 3.19 - Сторінка відгуків

На рис. 3.20 зображена сторінка користувача, яка має великий каталог для кастомізації та статистику, а саме наявні можливості додання моттто або слогана, додавання персональних соціальних мереж (YouTube, Twitter або X, Instagram), зміна фону, зміна профілю (імені, паролю, картинки профілю, електронної пошти), зміна або додання улюблених актора, режисера, фільму за роком народження та персонажа.

На рис. 3.21 проілюстровано нижню частину профілю, елемент з можливістю переключання між відгуками, списком перегляду, оцінками та улюбленими фільмами користувача.

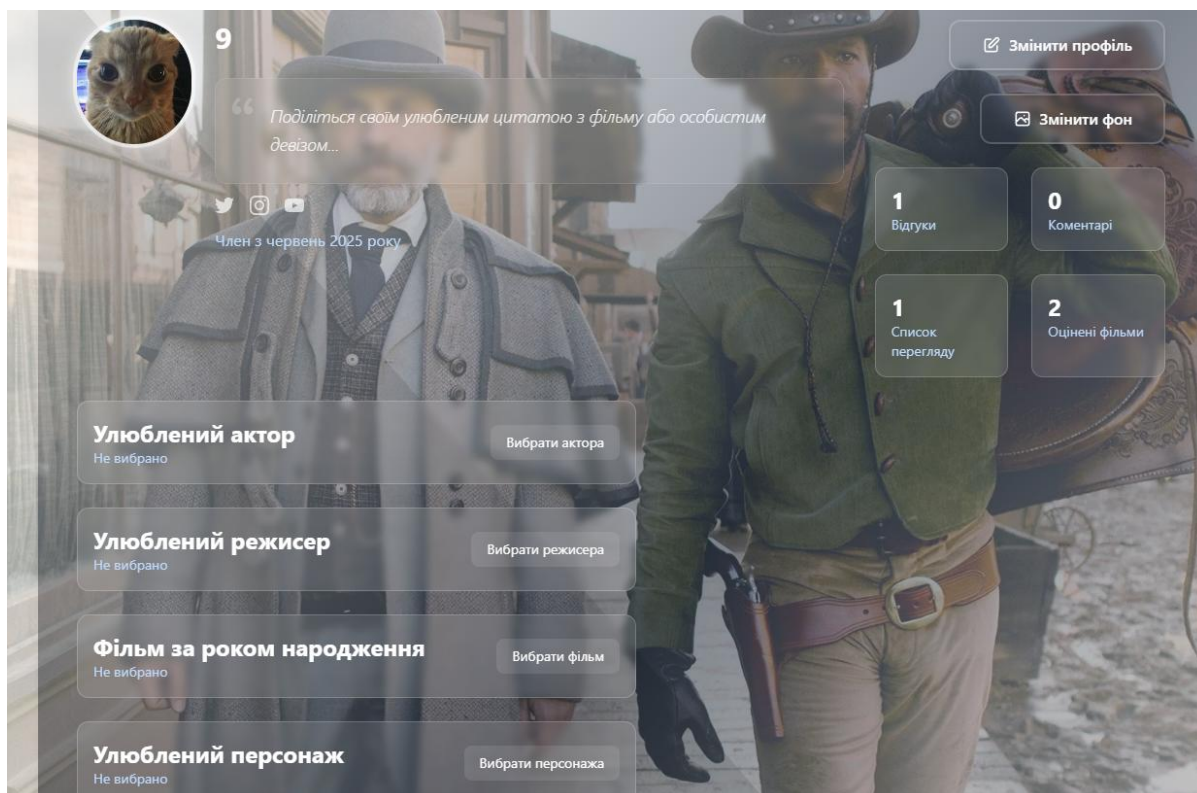


Рисунок 3.20 - Перша частина сторінки профіля

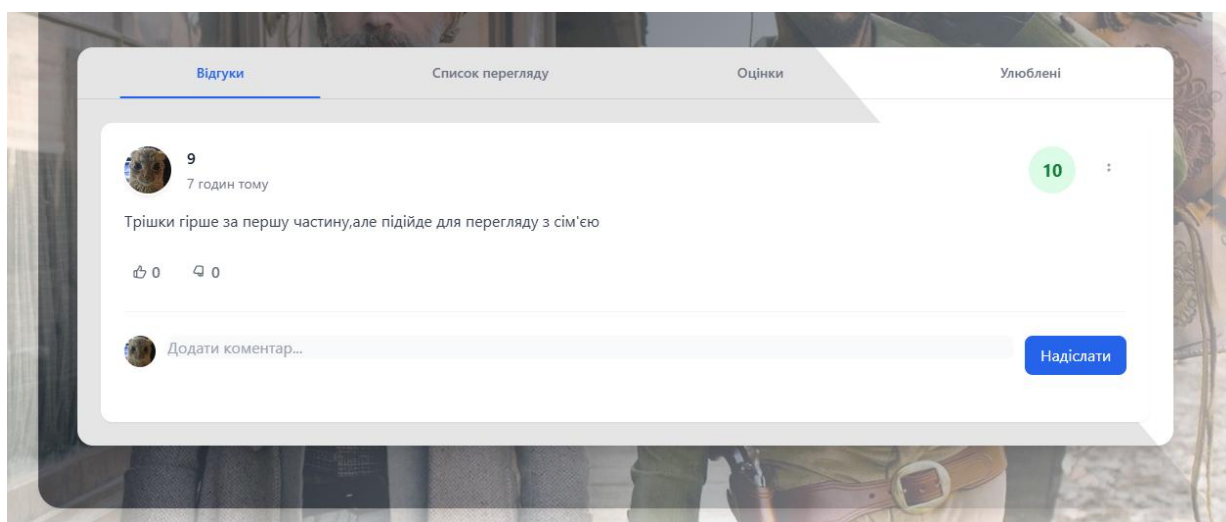


Рисунок 3.21 - Друга частина сторінки профілю

Платформа також надає можливість здійснити пошук. Результат виконання пошуку відображено на рис. 3.22. Пошук одночасно здійснюється по фільмам і акторам, але біля кнопки пошук можна обмежити результат до чогось одного. При натисненні на результат пошуку є можливість перейти на сторінку знайденого, щоб дізнатися більше інформації.

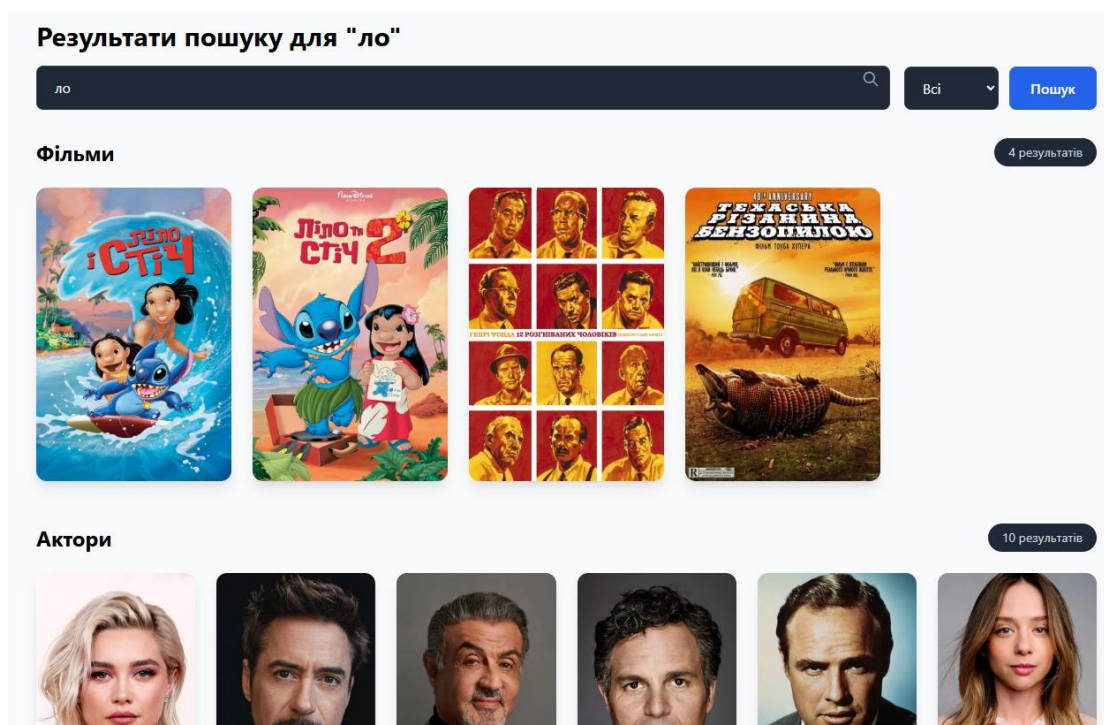


Рисунок 3.22 - Сторінка пошуку

На рис. 3.23 проілюстровано список перегляду нинішнього користувача із можливостями видалити із списку та фільтрувати. При натисненні на фільм є можливість прямо перейти на сторінку фільма та прибрати зі списку перегляду вже там.

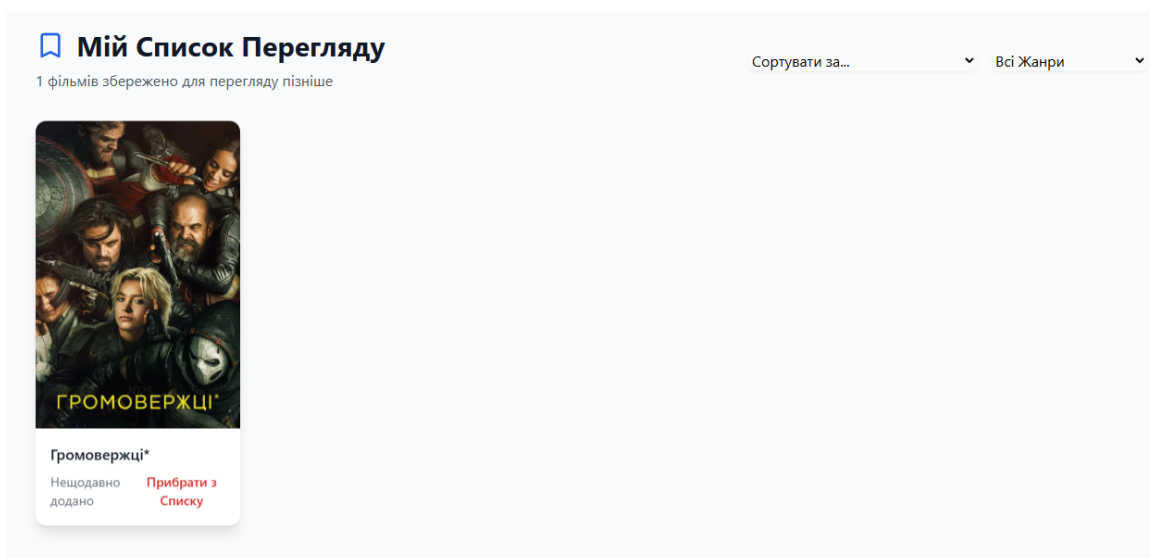


Рисунок 3.23 - Список перегляду користувача

Окрім списку перегляду є ще список усіх відгуків, який відображено на рис. 3.24, користувача, де можна подивитися, редагувати або видалити відгук при необхідності.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

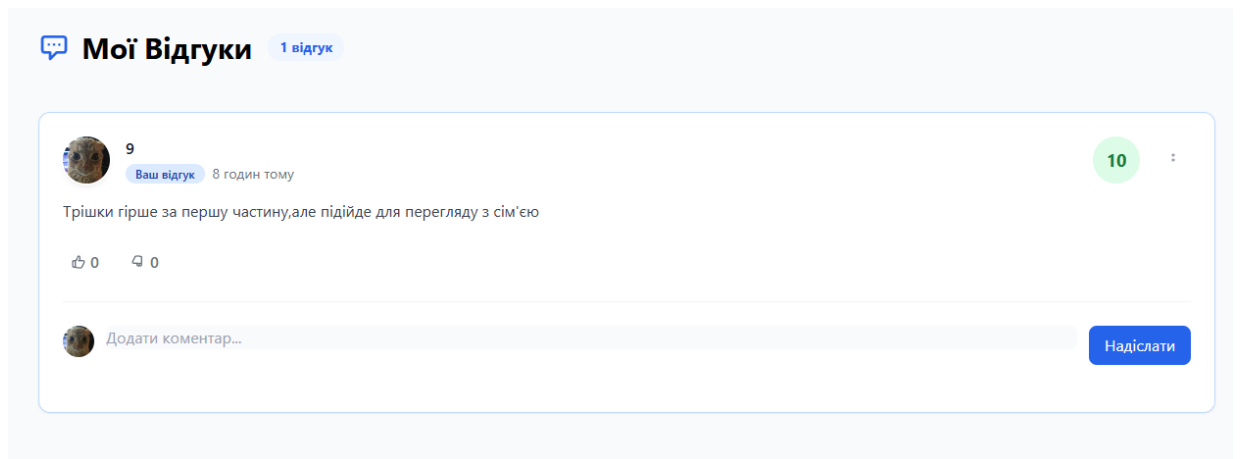


Рисунок 3.24 - Список відгуків користувача

На рис. 3.25 проілюстровані усі користувачі, з якими можна поспілкуватися, а якщо потрібно знайти конкретного користувача - можна скористатися пошуком. Також показується, якщо є непереглянуті повідомлення та вибрана тема чату.

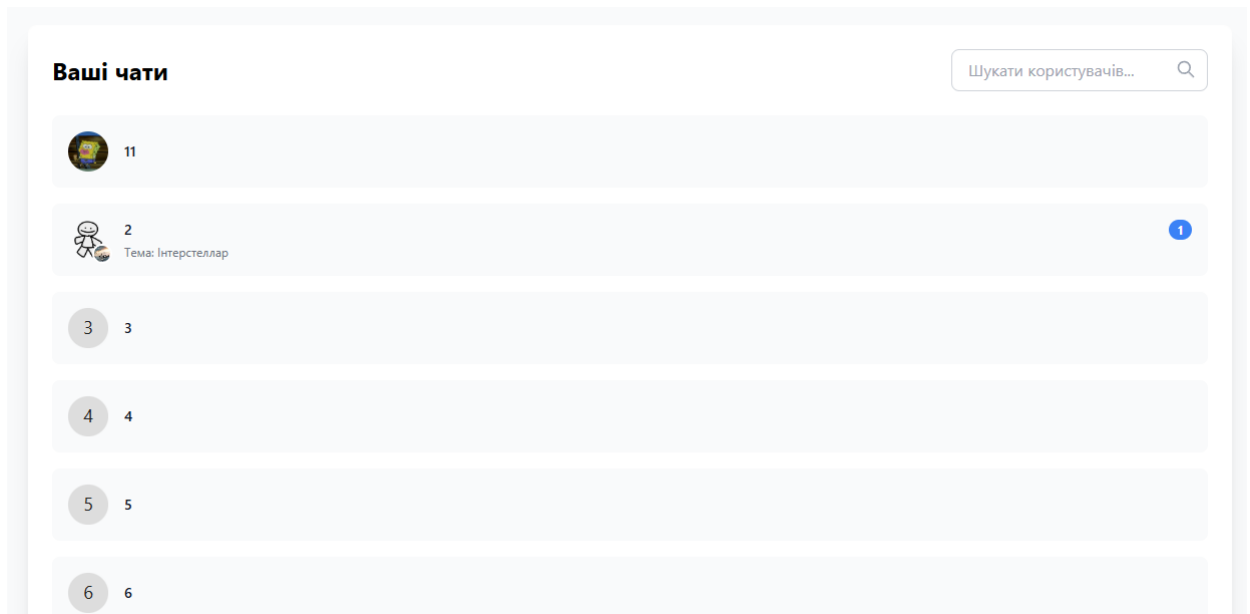


Рисунок 3.25 - Список чатів

Натиснувши на якийсь з доступних чатів, буде здійснено перехід на сторінку чата, яку відображено на рис. 3.26. У вікні чату відображається історія повідомлень між двома користувачами, у верхній частині інтерфейсу можна переглянути ім'я співрозмовника та перейти на його профіль для ознайомлення з додатковою інформацією, також користувач має можливість змінити тему обговорення. Ще передбачено поле для введення повідомлень із підтримкою емодзі, що дозволяє урізноманітнити комунікацію.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		63

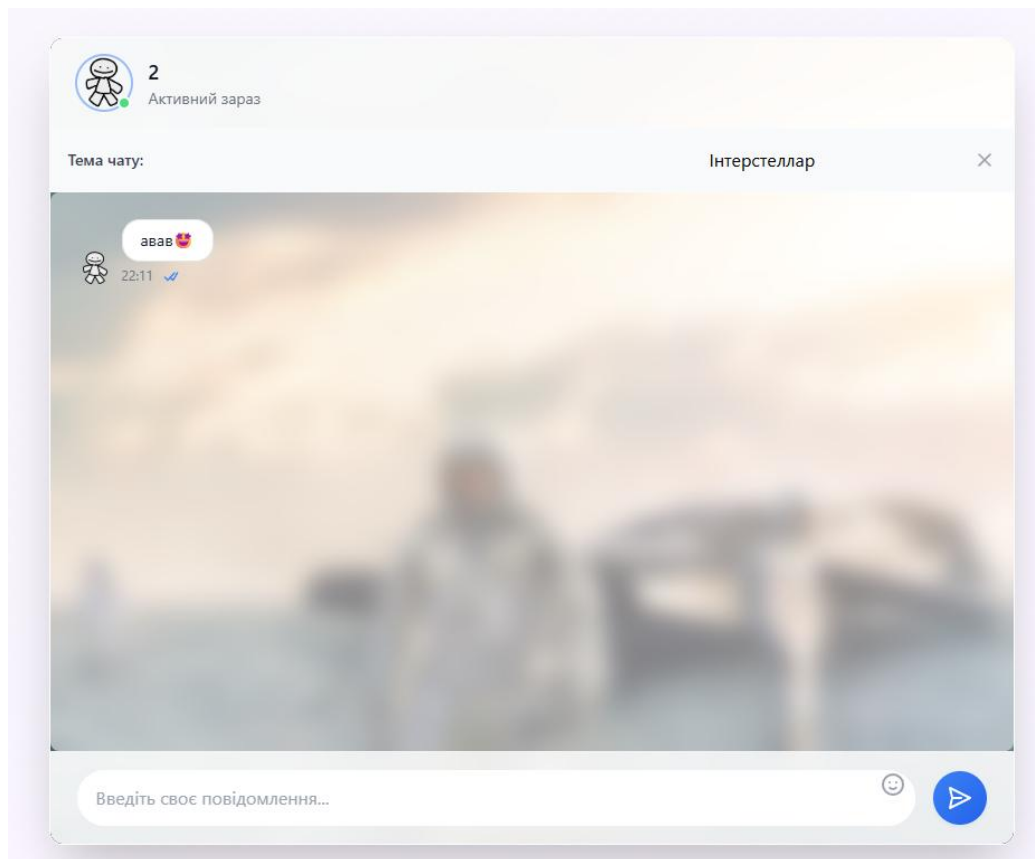


Рисунок 3.26 - Чат

На рис. 3.27 проілюстрована панель адміна із статистикою, можливістю подивитися на всі фільми, відгуки, користувачів, знімальні групи, акторів та подальшими функціями зміни або видалення їх усіх.

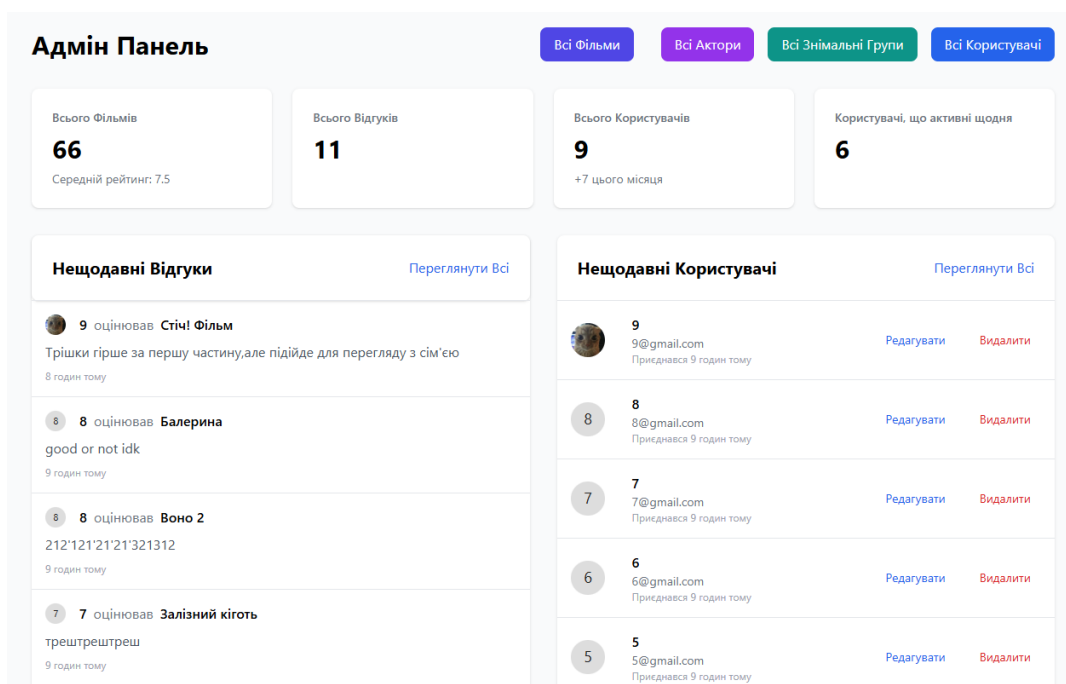


Рисунок 3.27 - Адмін панель

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

Далі, для прикладу, буде розглянуто один із списків панелі адміна - фільмів. На рис. 3.28 зображено невелику інформацію про кожен фільм, можливість пошуку та видалення або редагування фільму.

Управління Фільмами				Назад до Панелі Керування
Шукати фільми за назвою...		<button>Шукати</button>		
ФІЛЬМ	ДАТА ВИХОДУ	РЕЙТИНГ	ДІЇ	
 12 розгніваних чоловіків Вісімнадцятирічного юнака звинувачують у вбивстві власного батька, йому загрожує електричний стілець...	Apr 10, 1957	8.5 (9163 відгуків)	Редагувати Видалити	
 Аутсайтери Сюжет оповідає про групу американських інженерів і дизайнерів. У 1960-х під керівництвом конструктор...	Nov 13, 2019	8.0 (8328 відгуків)	Редагувати Видалити	
 Балерина Сюжет розгортається навколо Єви, старанної учениці балетної школи, а також безжалюїної вбивці з кримі...	Jun 04, 2025	8.0 (1 відгуків)	Редагувати Видалити	
 Бетмен Бетмен розслідує детективну справу — деякі люди починають вмирати дивним чином. Бетмен повинен загли...	Mar 01, 2022	10.0 (1 відгуків)	Редагувати Видалити	

Рисунок 3.28 - Список усіх фільмів в панелі

На рис. 3.29 проілюстровано усю інформацію про фільм і є можливість редагувати усе. Пролістнувши вниз, на рис. 3.30 зображено можливість змінити зв'язки та ролі акторів та знімальної групи. Також видно останній пункт зі зміни усієї інформації про фільм - огляд фільму або короткий опис про сюжет. Після усіх змін зберігаються зміни або скасовуються дії при потребності.

Редагувати Фільм		Назад до Фільмів
Назва	12 розгніваних чоловіків	
Дата Виходу	10.04.1957	Тривалість (хвилини)
Бюджет (\$)	397751	96
Середній Рейтинг	8,5	Дохід (\$)
Оригінальна Мова	en	4360000
Країни Виробництва	United States of America	Кількість Голосів
Розділіть кілька країн комами		9163
Шлях до Постера	/7bj9HoeWihfLJ5lc5wv01ALqedJ.jpg	PG Рейтинг
		G
		Виробничі Компанії
		United Artists, Orion-Nova Productions
		Розділіть кілька компаній комами
		Шлях до Бекдропу
		/bxgTSUenZDHNFeRQ1whRKplrMKF.jpg
		

Рисунок 3.29 - Меню редагування фільму, перша половина

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

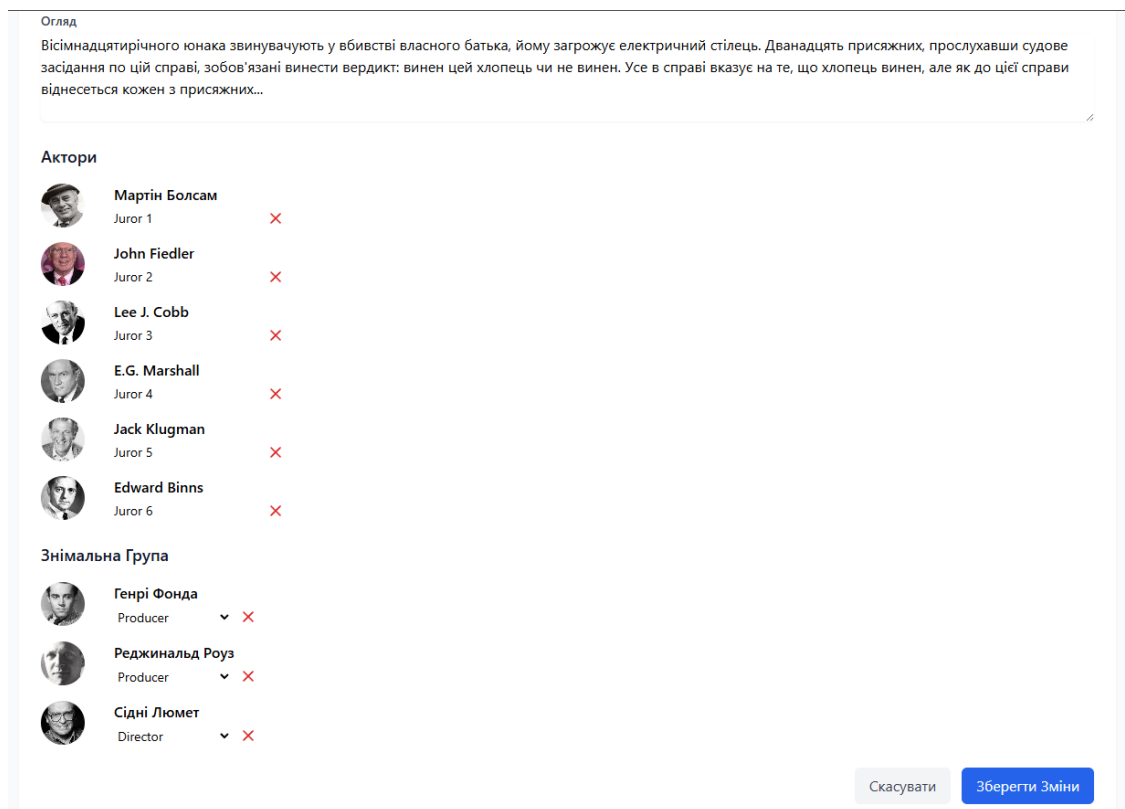


Рисунок 3.30 - Друга частина редагування фільма

На рис. 3.31 проілюстровано останній функціонал адміна - репорт система. На цій сторінці наявні і фільтри, і можливості видалити або змінити статус репорта, і переглянути детальніше репорт. Якщо було натиснено Подивитися деталі репорту, то буде відображено результат (рис. 3.32). Тут буде уся інформація про репорт, із можливістю видалити або змінити коментар, відгук або перейти на сторінку юзера. Також є кнопки зміни статусу та видалення репорту.

Менеджмент Репортів						
Всі Статуси		Всі Типи				
ПОВІДОМЛЕНИЙ ОБ'ЄКТ	ПОВІДОМЛЕНО КИМ	ПРИЧИНА		СТАТУС	ДАТА	ДІЇ
Відгук Залізний кіготь1	11	Spam	"312312312312"	Pending	3 hours ago	Переглянути Деталі Переглянуто Відрілено Видалити
Користувач 11	9 2	Inappropriate	"тарантіто???"	Pending	9 hours ago	Переглянути Деталі Переглянуто Відрілено Видалити
Spam		"спам"				

Рисунок 3.31 - Сторінка репортів

Деталі Репорту

← Назад до Репортів

Повідомлений Об'єкт

User

11

Переглянути профіль →

Пов'язані Репорти

Повідомлено: 2

Jun 12, 2025

Spam

"спам"

Інформація про Репорт

Повідомлено: 9

Дата: Jun 13, 2025 13:18

Статус: Pending

Причина: Inappropriate

Додаткові Деталі:

"тарантіто???"

Дії

Позначити як Переглянутий

Скасувати Репорт

Видалити Всі Репорти

Рисунок 3.32 - Деталі репорту

3.3 Тестування роботи платформи

Після завершення етапів розробки програмного забезпечення необхідно, щоб було проведено тестування з метою перевірки працездатності реалізованих функцій, стабільності системи та відповідності вимогам, визначеним на початкових етапах. Особлива увагу приділити критично важливим модулям, зокрема реєстрації та автентифікації користувачів, пошуку фільмів, публікації відгуків, роботі з рейтингами та системі репортів.

Перевірка реєстрації нового користувача та входу в акаунт є одними з ключових етапів тестування платформи. Ці процеси критично важливі, адже без успішної автентифікації користувач не зможе отримати доступ до особистого профілю, залишати відгуки, ставити оцінки чи брати участь у спілкуванні.

На рис. 3.33 представлено сторінку реєстрації, де були введені дані нового користувача - зокрема ім'я "Перевірка" та електронна пошта "check@gmail.com". Система успішно обробила запит, створивши обліковий запис і автоматично перенаправивши користувача на головну сторінку платформи вже залогіненим у новостворений акаунт.

Створити обліковий запис

Приєднуйтеся до нашої спільноти кіноманів

Ім'я

перевірка

Email адреса

check@gmail.com

Пароль

••••••

Підтвердження пароля

••••••

Зареєструватися

Вже маєте обліковий запис? [Увійти](#)

Рисунок 3.33 - Тест на створення акаунта

На рис. 3.34 та на рис. 3.35 видно, що і на сайті, і в базі даних було успішно створено новий профіль, отже тестування по реєстрації і логіну було успішно протестовано. Наступний тест буде пов'язаний із редагуванням профілю користувача.

ПЕ

перевірка

Змінити профіль

Змінити фон

“ Поділіться своїм улюбленим цитатою з фільму або особистим девізом...”

🐦
📷
🗨️

Член з червень 2025 року

Улюблений актор

Не вибрано

Вибрати актора

Улюблений режисер

Не вибрано

Вибрати режисера

Фільм за роком народження

Не вибрано

Вибрати фільм

0

Відгуки

0

Коментарі

0

Список перегляду

0

Оцінені фільми

Рисунок 3.34 - Новий створений акаунт

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

Рисунок 3.35 - Поле в базі даних

На рис. 3.36 відбувається спроба зміни даних профілю, а саме зміна імені, пошти та аватару. На рис. 3.37 проілюстровано вже результат зміни і як видно, зміна була успішною.

Змінити Профіль

PE

Change Avatar

Вибрати файл

mod.png

Ім'я

НОВИЙ ПОЧАТОК

Пошта

newstart@gmail.com

Поточний Пароль

Новий Пароль

Підтвердити Новий Пароль

Скасувати

Зберегти Зміни

Рисунок 3.36 - Зміна даних профілю

Змінити Профіль



Change Avatar

Вибрати файл

Файл не вибрано

Ім'я

НОВИЙ ПОЧАТОК

Пошта

newstart@gmail.com

Поточний Пароль

Новий Пароль

Підтвердити Новий Пароль

Скасувати

Зберегти Зміни

Рисунок 3.37 - Результат зміни профілю

На рис. 3.38 проілюстровано спробу зміни фону профілю з дефолтної на іншу, а на рис. 3.39 вже зображено знову успішний результат тесту функціоналу.

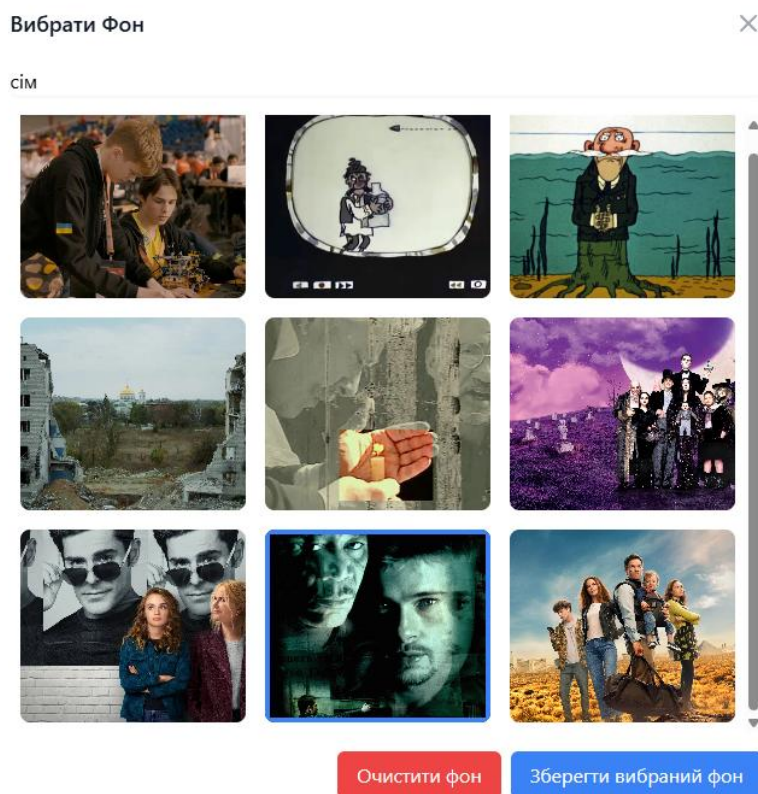


Рисунок 3.38 - Зміна фону

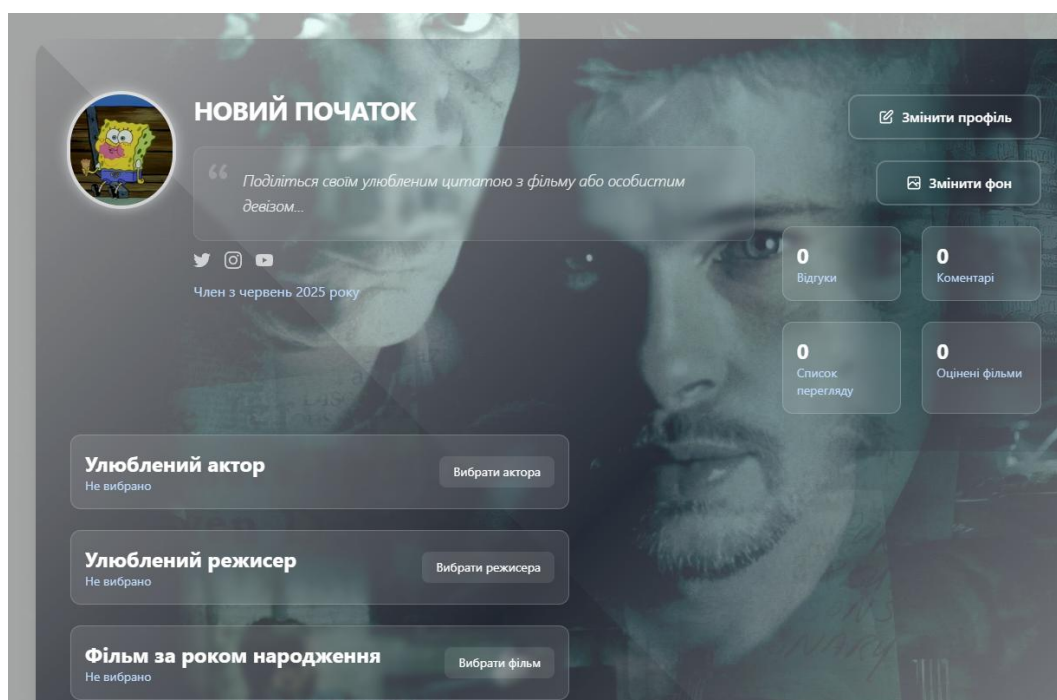


Рисунок 3.39 - Змінений фон

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Кушнір Н.О.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

На рис. 3.40 відображено функція вибору улюбленого актора з пошуком, після натиснення на кнопку збереження, вибраний актор має зберегтися в базу даних і відображатися. На рис.3.41 зображено, що тест був успішно виконаний та перевірений.

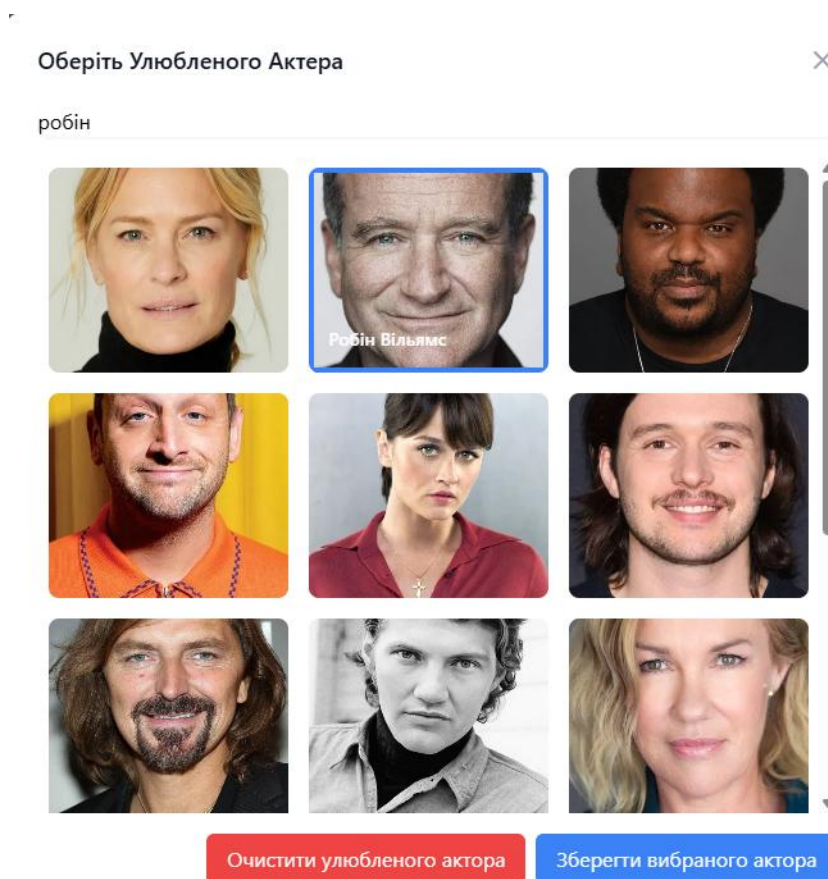


Рисунок 3.40 - Вибір улюбленого актора



Рисунок 3.41 - Результат вибору улюбленого актора

Наступний тест буде на перевірку імпорту фільму з деякими змінами. На рис. 3.42 і рис. 3.43 представлена сторінка імпорту фільму зі зміненими даними для тесту чи збережеться фільм зі змінами даних, деякі дані неприпустимі. Ні актори, ні знімальна команда не буде обрана через чекбокси. Валідація на рис. 3.44 проходить успішно.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		71



Пікселі

2015 5.7/10 (7,706 голосів) 105 хв

Огляд

Прибульці з космосу помилково трактують тематику комп'ютерних ігор як оголошення їм війни і атакують Землю, використовуючи відеоігри 80-х в якості моделей для різноманітних стратегій вторгнення. Президент країни звертається за допомогою до друга дитинства, чемпіона тих років по грі в автомати Сема Бреннера. Той збирає команду гравців, фахівців з відеоігор, щоб спільно здолати прибульців і врятувати планету.

Актори

Вибрати всіх

Скасувати вибір



Ім'я
Адам Сандлер
Персонаж
Brenner



Ім'я
Кевін Джеймс
Персонаж
Cooper

Рисунок 3.42 - Перша частина тесту на додання зміненого фільма

Назва Пікселі	Дата Виходу 16.07.2015
Тривалість (хвилини) 105	Бюджет (\$) -88000000
Дохід (\$) -888888	Середня Оцінка 5,7
Кількість Голосів 7706	Оригінальна Мова en
Віковий Рейтинг R	
Рейтинг визначається за змістом та аудиторією	
Країни Виробництва China, France, India, United States of America	
Розділіть кілька країн комами	
Кінокомпанії Columbia Pictures, Happy Madison Productions, 1492 Pictures, LStar Capital, China Film Group Corporation, Film Croppers Entertainment, Prime Focus	
Розділіть кілька компаній комами	
Категорії/Жанри	
☐ Драма	☐ Смієйний
☐ Фентезі	☐ Історія
☐ Жахи	☐ Музика
☐ Детектив	☐ Романтика
☒ Наукова фантастика	☐ Телефільм
☐ Трилер	☐ Військовий
☐ Вестерн	
Оригінальні жанри TMDB:	
☐ Бойовик ☐ Комедія ☐ Фантастика ☐ Фентезі	
Огляд	
111	

Рисунок 3.43 - Друга частина тесту на додання зміненого фільма

Бюджет (\$)
-88000000
Середня Оцінка
5,7
Оригінальна Мова

Значення має бути більшим або дорівнювати 0.

Рисунок 3.44 - Валідація текстового поля

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		72

На рис. 3.45 вже зображено результат тесту на зміну даних фільма при його імпорті: немає ні акторів, ні знімальної команди та інші елементи було успішно збережено.

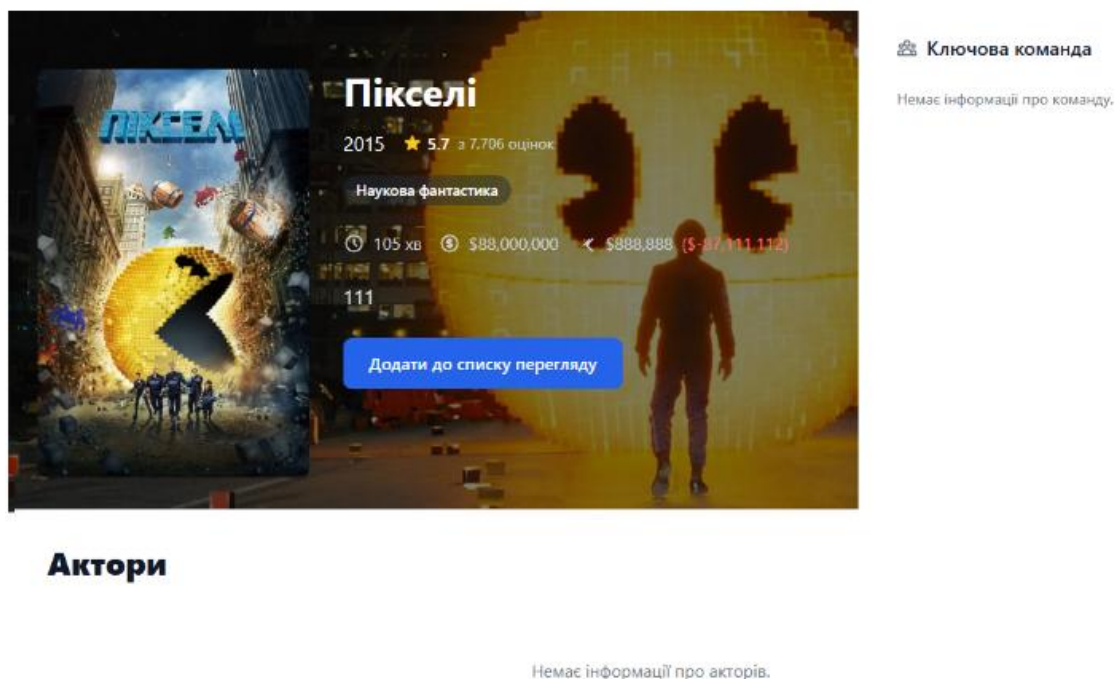


Рисунок 3.45 - Результат тесту на зміну фільма при імпорті

На рис. 3.46 вже можна побачити нещодавно доданий фільм вже в панелі управління фільмами, яке доступно тільки адміністратору. Наступний тест буде на видалення фільму з бази даних, а також перевірка чи видалиться зв'язок в таблиці жанрів із фільмами, який проілюстровано на рис. 3.47.

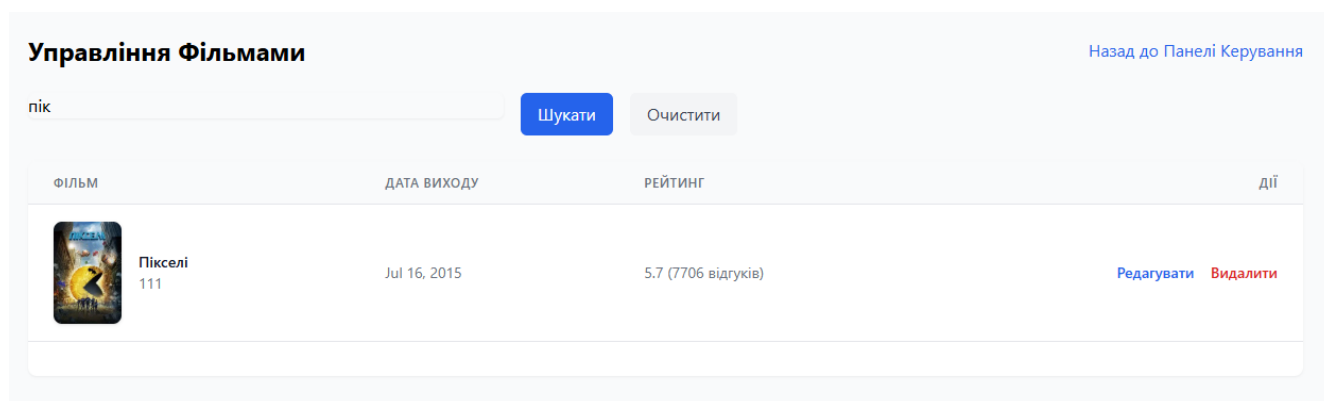


Рисунок 3.46 - Нещодавно доданий фільм в панелі

1

>

>>

☐ Показати все

Число рядків:

25

Фільтрувати рядки:

179

Екстра параметри

id

film_id

category_id

created_at

updated_at

☐

Редагувати

Копіювати

Видалити

296

117

9

NULL

NULL

Рисунок 3.47 - Поле в таблиці зв'язків з жанрами

Натиснувши на кнопку видалити, з'являється вікно, яке проілюстровано на рис. 3.48, що є правильним, оскільки адмін може випадково натиснути не на той фільм або не на ту кнопку. На рис. 3.49 зображено результат пошуку на сайті і в базі даних після видалення фільму. На сайті фільму немає, а в базі даних видається зв'язок іншого фільму.

Видалити Фільм?

Ви впевнені, що хочете видалити "Пікселі"? Це призведе до видалення всіх пов'язаних відгуків і рейтингів. Цю дію не можна скасувати.

Скасувати

Видалити

Рисунок 3.48 - Вікно підтвердження перед видаленням

Результати пошуку для "пікселі"

Немає результатів для "пікселі"

1	>	>>	☐ Показати все	Число рядків: 25	Фільтрувати рядки: 179						
Екстра параметри											
				id	film_id	category_id	created_at	updated_at			
☐		Редагувати		Копіювати		Видалити	296	117	9	NULL	NULL

☐ Позначити все
 ☐ Вибрані

Рисунок 3.49 - Пошук після видалення фільму

Далі було проведено тест шляхів, а саме була спроба перейти на сторінку, до якої доступ має тільки адмін. На рис. 3.50 відбувається спроба перейти на сторінку

		Рику А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

репортів, яка доступна тільки модератору і адміністратору, як незалогінований користувач.

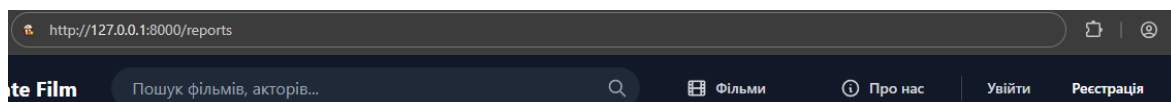


Рисунок 3.50 - Спроба перейти на сторінку без доступу

На рис. 3.51 показана сторінка логіну після спроби увійти на захищену сторінку, але навіть після заходу в створений акаунт і повторення спроби, звичайний користувач все одно не зможе потрапити на сторінку, яка зарезервована для адміністратора.

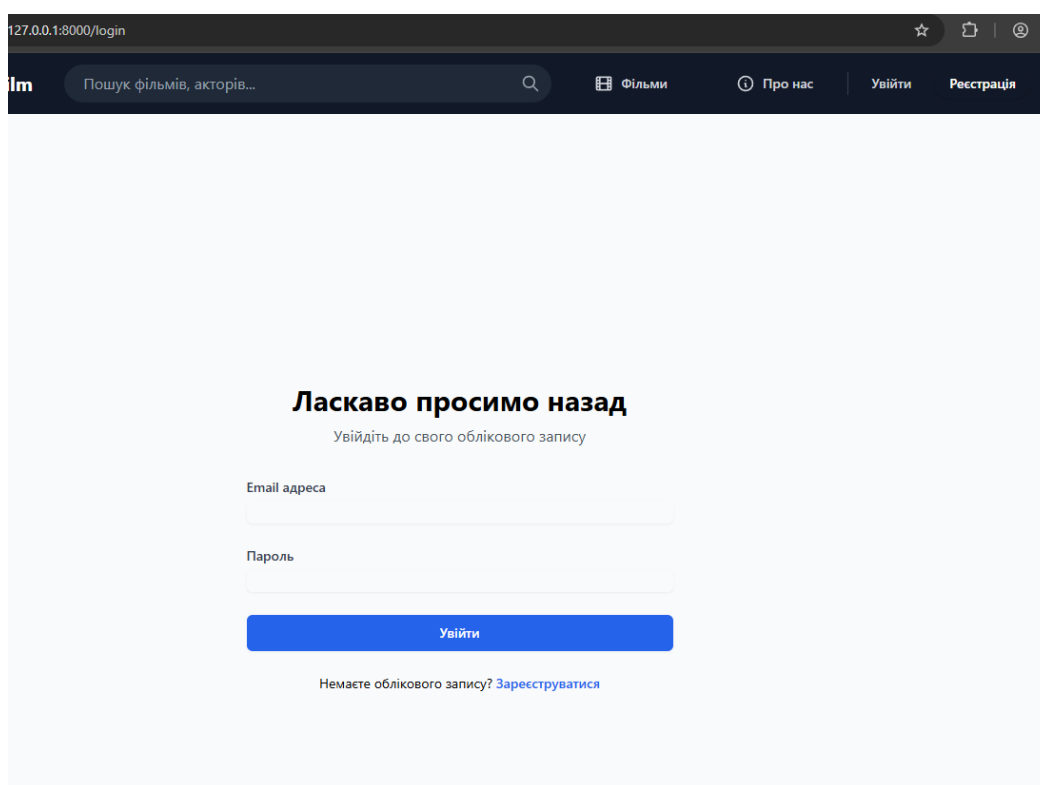


Рисунок 3.51 - Сторінка логіна після спроби зайти на захищену сторінку

Далі, на рис. 3.52, відображено сторінку чату, з відправленим непрочитаним повідомленням та обраною темою. За користувача, з яким відбувається бесіда - чат, буде надіслано повідомлення та змінено тему чата, а потім буде повернення до оригінального користувача, щоб подивитися чи прочитається повідомлення, чи отримає користувач нове повідомлення і чи зміниться тема.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		75

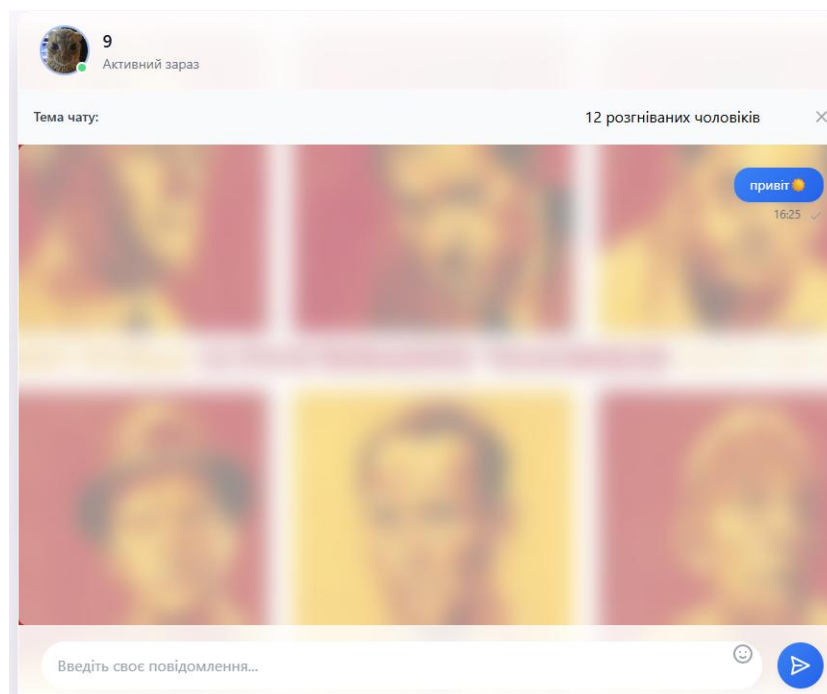


Рисунок 3.52 - Тест сторінки чату

На рис. 3.53 відображено, що тест був успішний, оскільки всі зміни було збережено, користувач прочитав повідомлення та тема була змінена і справа зверху можна помітити нову назву з зміненим фоном. Повідомлення зберіглося правильно із емодзі і відображається також коректно. Одна галочка змінилася на дві, що сигналізує про те, що повідомлення було прочитано, в той час як наступне повідомлення не було, що також є коректним.

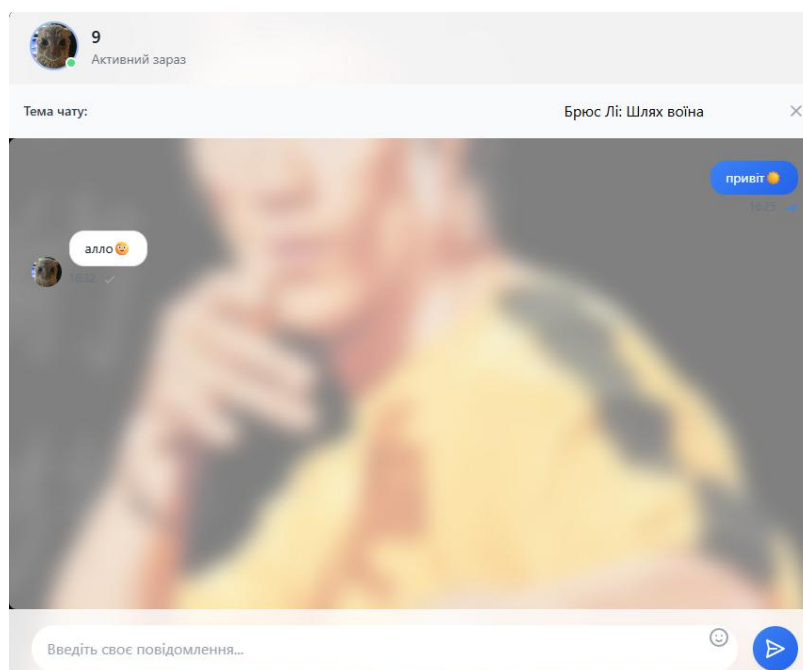


Рисунок 3.53 - Результат тесту сторінки чату

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Кушнір Н.О.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

Підведемо підсумок: тестування охоплювало процеси реєстрації, входу, редагування профілю, редагування акторів та фільмів як адміністратор, імпорту фільмів, перевірку доступу до захищених сторінок, а також функціональність системи чату.

Нижче в табл. 3.3 представлено результати тестування із вхідними даними.

Таблиця 3.3

Результати тестування

№	Тестована функціональність	Вхідні дані / Дії	Очікуваний результат	Результат тесту
1	Реєстрація нового користувача	Ім'я: "перевірка", Email: check@gmail.com	Створення нового облікового запису	Успішно
2	Вхід користувача	Авторизація через створений акаунт	Перехід у профіль	Успішно
3	Зміна профілю користувача	Зміна імені, email, аватару	Дані змінено, збережено	Успішно
4	Зміна фону профілю	Вибір іншого фону	Новий фон відображається у профілі	Успішно
5	Вибір улюбленого актора	Пошук і збереження актора	Актор збережений і відображається	Успішно
6	Імпорт фільму з частковими даними	Дані фільму без акторів та знімальної групи	Фільм імпортовано, дані збережено	Успішно
7	Відображення нового фільму в адмін-панелі	Перехід у панель після додавання фільму	Новий фільм видно у списку	Успішно
8	Видалення фільму	Кнопка «Видалити» + підтвердження	Фільм і зв'язки видалено, не відображається на сайті	Успішно
9	Перехід до сторінки без доступу	Спроба зайти на сторінку «Репорти» без авторизації або з роллю користувача	Переадресація на сторінку логіну або обмеження доступу	Успішно
10	Тестування чату	Відправка повідомлення, зміна теми, перевірка відображення у співрозмовника	Нове повідомлення та тема відображаються, статус прочитання оновлено	Успішно

Висновки до третього розділу

У ході тестування платформи для обговорення та оцінювання фільмів було перевірено ключові функціональні можливості системи, які безпосередньо впливають на взаємодію користувача з застосунком. Деякі функції являються базованими та обов'язковими, без яких не може бути проєкта, тому вони були ретельно протестовані, в той час як інші функції є доповненням до звичайного стандартного досвіду, який користувач отримуємо при перебуванні на платформі. Результати тестування є задовільними, оскільки все, що було протестовано було успішним і не було виявлено ніяких додаткових проблем.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Кушнір Н.О.				78
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Метою даної випускної кваліфікаційної роботи бакалавра була реалізація платформи для обговорення й оцінювання фільмів, яка забезпечувала б інтерактивність, персоналізацію контенту та зручність користування для кінолюбителів. У процесі виконання роботи ця мета була досягнута шляхом аналізу аналогічних сервісів, таких як IMDB, Letterboxd та TMDb, визначення їх сильних і слабких сторін, а також на основі сформованих вимог до функціоналу майбутньої системи.

Під час розробки було обрано архітектуру MVC з використанням фреймворку Laravel, що забезпечило структурованість, гнучкість та можливість подальшого масштабування. Візуальна частина була реалізована засобами Blade-шаблонів, CSS-фреймворку Tailwind та JavaScript з додатковим використанням бібліотеки Alpine.js, що дозволило створити сучасний та адаптивний інтерфейс з елементами взаємодії без перезавантаження сторінки. Для роботи з фільмами та акторами інтегровано TMDb API, що дало змогу зручно отримувати та редагувати зовнішні дані. Крім того, реалізовано модуль аутентифікації з підтримкою ролей користувачів, систему написання та оцінювання відгуків, функціонал репортів на неприйнятний контент такий як відгуки, коментарі або самі користувачі, обмін повідомленнями між користувачами через чат, списки рекомендованих фільмів, а також основні засоби безпеки й перевірки доступу до адміністративних сторінок.

Усі поставлені завдання були реалізовані відповідно до технічних і функціональних вимог, що дозволило провести повноцінне тестування системи. Перевірка сценаріїв взаємодії користувача з інтерфейсом, обробки відгуків, імпорту фільмів і керування ними, а також захищеності доступу до адміністративних модулів підтвердила правильність реалізації. Подальше вдосконалення системи може включати адаптації під мобільні пристрої, вдосконалення рекомендацій, використовуючи алгоритми машинного навчання, підтримку кількох мов, а також додаткову інтеграцію з іншими зовнішніми платформами.

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		79

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рикун А.О., Кушнір Н.О. Платформа для обговорення та оцінювання фільмів. Тези доповідей Всеукраїнської науково-практичної on-line конференції здобувачів вищої освіти і молодих учених, присвяченої Дню науки, 12-17 травня 2025 року. Житомир: «Житомирська політехніка», 2025. С. 63.
2. Adeyomola Dev: Деплой Laravel через Apache. URL: <https://adeyomoladev.medium.com/how-to-deploy-a-laravel-app-using-apache-and-mysql-4910a07f9a0c>
3. Alpine.js Documentation. URL: <https://alpinejs.dev>
4. Alpine.js Start Guide. URL: <https://alpinejs.dev/start-here>
5. Blade Template Engine Documentation. URL: <https://laravel.com/docs/9.x/blade>
6. Clearcode: Composer-залежності. URL: <https://clearcode.cc/blog/manage-application-setup-composer>
7. Composer Documentation. URL: <https://getcomposer.org/doc/01-basic-usage.md>
8. Daily Dev: Tailwind CSS for Beginners. URL: <https://daily.dev/blog/tailwind-css-basics-for-beginners>
9. Dev.to: Laravel + XAMPP (Windows). URL: <https://dev.to/patricklusaya/how-to-setup-laravel-on-windows-with-xampp-quick-guide-16ng>
- 10.DevOps Blog: TMDb API y Laravel 9. URL: <https://blog.devops.dev/working-with-tmdb-api-in-laravel-9-4b4c578b75e4>
- 11.GeeksForGeeks: MySQL Tutorial. URL: <https://www.geeksforgeeks.org/mysql/mysql-tutorial>
- 12.IMDb - база даних фільмів та серіалів. URL: <https://www.imdb.com>
- JavaScript Documentation - Mozilla Developer Network. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- 13.JavaScript for Designers - Medium. RL: <https://nicklawrencedesign.medium.com/javascript-for-designers-dabc8871d4c1>
- 14.Laravel Documentation. URL: <https://laravel.com/docs>

		Рикун А.О.			ІПЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Кушнір Н.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		80

15. LaunchSchool: TMDb API Overview. URL: https://launchschool.com/books/working_with_apis/read/tmdb_api
16. Letterboxd - соціальна платформа для кіноманів. URL: <https://letterboxd.com>
17. MageComp: Tailwind CSS у Laravel. URL: <https://magecomp.com/blog/mastering-tailwind-css-in-laravel>
18. Medium: UI/UX Design Roadmap. URL: <https://medium.com/design-bootcamp/the-ultimate-ui-ux-design-roadmap-a-detailed-and-structured-guide-to-self-learning-11f29f82f867>
19. MySQL Documentation. URL: <https://dev.mysql.com/doc>
20. PHP Manual. URL: <https://www.php.net/manual>
21. phpMyAdmin Official Documentation. URL: <https://docs.phpmyadmin.net/en/latest/intro.html>
22. Skillforge: Створення бази в phpMyAdmin через XAMPP. URL: <https://skillforge.com/how-to-create-a-database-using-phpmyadmin-xampp>
23. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs>
24. Tailwind CSS Theme Variables. URL: <https://tailwindcss.com/docs/theme>
25. The Movie Database (TMDb). URL: <https://www.themoviedb.org>
26. TMDb API: Getting Started. URL: <https://developer.themoviedb.org/docs/getting-started>
27. TMDb API Docs - Search & Images. URL: <https://developer.themoviedb.org>
28. TMDb API Search Movies. URL: <https://developer.themoviedb.org/reference/search-movie>
29. TMDb Image Basics. URL: <https://developer.themoviedb.org/docs/image-basics>
30. W3Schools - основи веб-технологій. URL: <https://www.w3schools.com>

ДОДАТКИ

		<i>Рикун А.О.</i>			ІПЗ.КР.Б – 121 – 25 – ІЗ	<i>Арк.</i>
		<i>Кушнір Н.О.</i>				
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		82

Клас,що відповідає за імпорт з API

```

<?php

class TmdbService
{
    private static ?TmdbService $instance = null;
    protected string $apiKey;
    protected string $baseUrl = 'https://api.themoviedb.org/3';

    // Make constructor private
    private function __construct()
    {
        $this->apiKey = config('services.tmdb.api_key');
    }

    // Singleton getInstance method
    public static function getInstance(): TmdbService
    {
        if (self::$instance === null) {
            self::$instance = new self();
        }
        return self::$instance;
    }

    // Prevent cloning
    private function __clone() {}

    // Prevent unserialization
    public function __wakeup()
    {
        throw new \Exception("Cannot unserialize singleton");
    }

    /** Generic GET with error-handling */
    public function get(string $uri, array $qs = [])
    {
        return Http::get("{ $this->baseUrl}/{ $uri}", $qs + ['api_key' => $this->apiKey])
            ->throw()
            ->json();
    }

    /* ----- public helpers ----- */

    public function movie(int $id): array
    {
        // cache each movie for 1 hour
    }
}

```

```

        return Cache::remember("tmdb.movie.$id", 3600, fn () => $this->get("movie/{$id}"));
    }

    public function genres(): array
    {
        // cache genre list for 24 h
        return Cache::remember('tmdb.genres', 86400, fn () => $this->get('genre/movie/list')['genres']);
    }

    public function importPopularMovies(int $pages = 1): array
    {
        $allMovies = [];

        for ($page = 1; $page <= $pages; $page++) {
            $response = Http::get("{ $this->baseUrl }/movie/popular", [
                'api_key' => $this->apiKey,
                'page' => $page
            ]);

            if (!$response->successful()) {
                Log::error('TMDB API error', [
                    'status' => $response->status(),
                    'body' => $response->body()
                ]);
                continue;
            }

            $data = $response->json();

            if (!isset($data['results']) || !is_array($data['results'])) {
                Log::error('Invalid TMDB API response format');
                continue;
            }

            $allMovies = array_merge($allMovies, $data['results']);
        }

        return $allMovies;
    }

    public function importPopularActors($pages = 1)
    {
        $actors = collect();

        for ($page = 1; $page <= $pages; $page++) {
            try {
                $data = $this->get('person/popular', ['page' => $page]);
            }
        }
    }

```

```

        if (isset($data['results']) && is_array($data['results'])) {
            $actors = $actors->concat($data['results']);
        }
    } catch (\Exception $e) {
        Log::error('Failed to fetch popular actors', [
            'page' => $page,
            'error' => $e->getMessage()
        ]);
    }
}

return $actors;
}

public function getActorDetails($actorId)
{
    try {
        return $this->get("person/{$actorId}");
    } catch (\Exception $e) {
        Log::error('Failed to fetch actor details', [
            'actor_id' => $actorId,
            'error' => $e->getMessage()
        ]);
        return null;
    }
}

public function getMovieDetails($movieId)
{
    return Cache::remember("tmdb.movie.{ $movieId}", 3600, function () use
($movieId) {
        $response = Http::get("{ $this->baseUrl}/movie/{ $movieId}", [
            'api_key' => $this->apiKey,
            'language' => 'uk-UA', // Set Ukrainian language
            'append_to_response' => 'videos,images'
        ]);

        if ($response->successful()) {
            return $response->json();
        }

        return null;
    });
}

```

Контроллер, що відповідає за рекомендації

```
class RecommendationController extends Controller
{
    public function index()
    {
        $user = auth()->user();

        // 1. Улюблені жанри (за рецензіями з оцінкою >= 7)
        $favoriteGenres = $user->reviews()
            ->where('rating', '>=', 7)
            ->with('film.categories')
            ->get()
            ->pluck('film.categories')
            ->flatten()
            ->pluck('id')
            ->countBy()
            ->sortDesc()
            ->keys(); // ID жанрів за популярністю

        // 2. Рекомендації на основі жанрів
        $genreBased = Film::whereHas('categories', function ($query) use
($favoriteGenres) {
            $query->whereIn('categories.id', $favoriteGenres);
        })
            ->whereDoesntHave('reviews', function ($query) use ($user) {
                $query->where('user_id', $user->id);
            })
            ->with('categories')
            ->orderBy('vote_average', 'desc')
            ->take(10)
            ->get();

        // 3. Популярні серед усіх (за TMDb або голосами)
        $topRated = Film::where('vote_count', '>=', 100)
            ->orderBy('vote_average', 'desc')
            ->take(10)
            ->get();

        // 4. Новинки за останні 6 місяців
        $recentReleases = Film::where('release_date', '>=', now()->subMonths(6))
            ->orderBy('vote_average', 'desc')
            ->take(10)
            ->get();

        // 5. Схожі фільми на ті, які оцінив користувач (розширення далі)
        // Поки що виведемо випадкові популярні фільми
```

```

$randomPopular = Film::inRandomOrder()
    ->where('vote_average', '>=', 7)
    ->take(10)
    ->get();

$favoriteDirectors = $user->reviews()
    ->where('rating', '>=', 7)
    ->with('film.crew')
    ->get()
    ->pluck('film.crew')
    ->flatten()
    ->filter(function ($crewMember) {
        return $crewMember->pivot->job === 'Director' || $crewMember->pivot-
>job === 'Режисер' || $crewMember->pivot->job === 'Директор';
    })
    ->pluck('id')
    ->countBy()
    ->sortDesc()
    ->keys();

$directorBased = Film::whereHas('crew', function ($query) use
($favoriteDirectors) {
    $query->whereIn('people.id', $favoriteDirectors)
        ->where(function ($q) {
            $q->where('crew_members.job', 'Director')
                ->orWhere('crew_members.job', 'Режисер')
                ->orWhere('crew_members.job', 'Директор');
        });
    });
->whereDoesntHave('reviews', fn($q) => $q->where('user_id', $user->id))
->orderBy('vote_average', 'desc')
->take(10)
->get();

$underrated = Film::where('vote_average', '>=', 7.5)
    ->where('vote_count', '<', 50)
    ->orderByDesc('vote_average')
    ->take(10)
    ->get();

$shortFilms = Film::where('runtime', '<=', 90)
    ->orderByDesc('vote_average')
    ->take(10)
    ->get();

$favoriteCountries = $user->reviews()
    ->where('rating', '>=', 7)
    ->with('film')
    ->get()
    ->pluck('film.production_countries')

```

```

->flatten()
->filter()
->countBy()
->sortDesc()
->keys()
->take(5); // Limit to top 5 favorite countries

$countryBased = Film::where(function ($query) use ($favoriteCountries) {
    foreach ($favoriteCountries as $country) {
        $query->orWhereJsonContains('production_countries', $country);
    }
})
->whereDoesntHave('reviews', function ($query) use ($user) {
    $query->where('user_id', $user->id);
})
->orderByDesc('vote_average')
->take(10)
->get();

return view('recommendations.index', compact(
    'genreBased',
    'topRated',
    'recentReleases',
    'randomPopular',
    'directorBased',
    'underrated',
    'shortFilms',
    'countryBased'
));
}

}

```

Контроллер, що відповідає за маніпуляції з фільмами

```

class AdminController extends Controller
{
    public function manageFilms(Request $request)
    {
        $query = $request->get('query');
        $page = (int) $request->get('page', 1);
        $searchResults = collect();
        $pagination = null;

        if ($query) {
            try {
                $response = $this->tmdbService->searchMovies($query, $page);

                if (isset($response['results'])) {
                    $searchResults = collect($response['results'])
                        ->filter(function ($movie) {
                            return ($movie['adult'] ?? false) === false;
                        })
                        ->map(function ($movie) use ($query) {
                            $title = strtolower($movie['title']);
                            $searchQuery = strtolower($query);
                            similar_text($title, $searchQuery, $percent);
                            $movie['relevance'] = $percent;
                            return $movie;
                        })
                        ->sortByDesc(function ($movie) {
                            return ($movie['relevance'] * 0.7) +
($movie['popularity'] * 0.3);
                        })
                        ->values();

                    $pagination = [
                        'current_page' => $response['page'],
                        'total_pages' => min($response['total_pages'], 500), // TMDB
limits to 500 pages
                        'total_results' => $response['total_results'],
                        'has_more' => $response['page'] <
min($response['total_pages'], 500)
                    ];
                }
            } catch (\Exception $e) {
                Log::error('TMDB search failed', [
                    'query' => $query,
                    'page' => $page,
                    'error' => $e->getMessage()
                ]);
            }
        }
    }
}

```

```

        ]);
    }
}

return view('admin.films.manage', [
    'searchResults' => $searchResults,
    'pagination' => $pagination,
    'query' => $query
]);
}

public function previewFilm($tmdb_id)
{
    try {
        $movieDetails = $this->tmdbService->getMovieDetails($tmdb_id);
        $credits = $this->tmdbService->getMovieCredits($tmdb_id);
        $categories = Category::all(); // Add this line

        // Get detailed information for each cast member
        if (isset($credits['cast'])) {
            $credits['cast'] = collect(array_slice($credits['cast'], 0, 6))-
>map(function ($actor) {
                $actorDetails = $this->tmdbService-
>getPersonDetails($actor['id']);
                return array_merge($actor, [
                    'birthday' => $actorDetails['birthday'] ?? null,
                    'deathday' => $actorDetails['deathday'] ?? null,
                    'biography' => $actorDetails['biography'] ?? null,
                    'place_of_birth' => $actorDetails['place_of_birth'] ?? null,
                    'popularity' => $actorDetails['popularity'] ?? 0
                ]);
            }->toArray();
        }

        return view('admin.films.preview', [
            'movie' => $movieDetails,
            'credits' => $credits,
            'categories' => $categories,
            'isAdmin' => auth()->user()->role === 'admin',
            'isModerator' => auth()->user()->role === 'moderator'
        ]);
    } catch (\Exception $e) {
        Log::error('Failed to fetch movie preview', [
            'tmdb_id' => $tmdb_id,
            'error' => $e->getMessage()
        ]);

        return redirect()->route('admin.films.manage')
            ->with('error', 'Failed to load movie details.');
```

```

}

public function importFilm(Request $request)
{
    try {
        DB::beginTransaction();

        // Create the film with new fields
        $film = Film::create([
            'tmdb_id' => $request->tmdb_id,
            'title' => $request->title,
            'overview' => $request->overview,
            'runtime' => $request->runtime,
            'budget' => $request->budget,
            'revenue' => $request->revenue,
            'poster_path' => $request->poster_path,
            'backdrop_path' => $request->backdrop_path,
            'release_date' => $request->release_date,
            'vote_average' => $request->vote_average,
            'vote_count' => $request->vote_count,
            'original_language' => $request->original_language,
            // Add new fields here
            'production_countries' => $request->production_countries ?
                array_map('trim', explode(',', $request->production_countries)) :
                [],

            'production_companies' => $request->production_companies ?
                array_map('trim', explode(',', $request->production_companies)) :
                [],

            'pg_rating' => $request->pg_rating
        ]);
        // Ensure budget and revenue are not negative
        if ($film->budget < 0) {
            $film->budget = 0;
        }
        if ($film->revenue < 0) {
            $film->revenue = 0;
        }
        $film->save();
        // Sync categories if any selected
        if ($request->has('categories')) {
            $film->categories()->sync($request->categories);
        }

        // Process selected actors
        if ($request->has('selected_actors') && $request->has('actors')) {
            foreach ($request->selected_actors as $index) {
                $actorData = $request->actors[$index];

                $actor = Actor::updateOrCreate(
                    ['tmdb_id' => $actorData['tmdb_id']],

```

```

        [
            'name' => $actorData['name'],
            'birth_date' => $actorData['birth_date'] ?: null,
            'death_date' => $actorData['death_date'] ?: null,
            'biography' => $actorData['biography'],
            'place_of_birth' => $actorData['place_of_birth'],
            'photo' => $actorData['photo'],
            'popularity' => $actorData['popularity']
        ]
    );

    $film->actors()->attach($actor->id, [
        'character' => $actorData['character']
    ]);
}

// Process selected crew members
if ($request->has('selected_crew') && $request->has('crew')) {
    foreach ($request->selected_crew as $index) {
        $crewData = $request->crew[$index];

        $person = \App\Models\Person::updateOrCreate(
            ['tmdb_id' => $crewData['tmdb_id']],
            [
                'name' => $crewData['name'],
                'biography' => $crewData['biography'] ?? null,
                'profile_path' => $crewData['profile_path'] ?? null,
            ]
        );

        \App\Models\CrewMember::create([
            'film_id' => $film->id,
            'person_id' => $person->id,
            'job' => $crewData['job']
        ]);
    }
}

DB::commit();

return redirect()
    ->route('admin.films.manage')
    ->with('success', 'Film imported successfully');
} catch (\Exception $e) {
    DB::rollback();

    Log::error('Film import failed', [
        'error' => $e->getMessage(),
    ]);
}

```

```
        'trace' => $e->getTraceAsString(),  
        'request' => $request->all()  
    ]);  
  
    return back()  
        ->withInput()  
        ->with('error', 'Failed to import film: ' . $e->getMessage());  
    }  
}
```

```
class RatingController extends Controller
{
    protected $perPage = 10;

    public function index(Request $request, RatingFilter $filter)
    {
        $perPage = $request->input('per_page', $this->perPage);

        $ratings = Rating::query()
            ->with(['user', 'film'])
            ->filter($filter)
            ->paginate($perPage);

        return response()->json($ratings);
    }

    public function store(Request $request, Film $film)
    {
        $validated = $request->validate([
            'rating' => 'required|integer|between:1,10'
        ]);

        try {
            DB::beginTransaction();

            // Update or create rating
            $rating = Rating::updateOrCreate(
                [
                    'user_id' => auth()->id(),
                    'film_id' => $film->id
                ],
                ['rating' => $validated['rating']]
            );

            // Update review rating if exists
            $review = Review::where('user_id', auth()->id())
                ->where('film_id', $film->id)
                ->first();

            if ($review) {
                $review->update(['rating' => $validated['rating']]);
            }

            // Update film's average rating
            $averageRating = $film->ratings()->avg('rating');
            $film->update([
                'vote_average' => $averageRating,
                'vote_count' => $film->ratings()->count()
            ]);
        } catch (Exception $e) {
            DB::rollBack();
            return response()->json(['error' => $e->getMessage()]);
        }
    }
}
```

```
]);

DB::commit();
return back()->with('success', 'Rating updated successfully!');

} catch (\Exception $e) {
    DB::rollBack();
    return back()->with('error', 'Failed to update rating.');
}
}
```