

КВАЛІФІКАЦІЙНА РОБОТА

Група ІІЗ-20-1

Кізера В.В.

2024

ЗВО УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА

Факультет суспільних та прикладних наук

Кафедра інформаційних технологій

на правах рукопису

Кізера Віктор Вікторович

УДК 004.4

Розробка веб-каталогу кінематографічної продукції

Спеціальність 121 – «Інженерія програмного забезпечення»

Кваліфікаційна робота на здобуття кваліфікації бакалавр

Нормоконтроль

Студент

_____ Сτισло О.В.

(підпис, дата, розшифрування підпису)

_____ Кізера В.В.

(підпис, дата, розшифрування підпису)

Допускається до захисту

Керівник роботи

Завідувач кафедри

_____ к.т.н., доц. Ващишак С.П.

(підпис, дата, розшифрування підпису)

_____ к.т.н., доц. Слабінога М.О.

(підпис, дата, розшифрування підпису)

Івано-Франківськ – 2024

ЗВО УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА
Факультет суспільних та прикладних наук
Кафедра інформаційних технологій

Освітній ступінь: «бакалавр»

Спеціальність: 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

« ____ » _____ 2024 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Кізера Віктор Вікторович

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:

Розробка веб-каталогу кінематографічної продукції

керівник роботи:

Слабінога Мар'ян Остапович, к.т.н., доцент.

затверджена наказом вищого навчального закладу від «12 » березня 2024 року

№ 19/1

2. Термін подання студентом роботи 05.06.2024

3. Вихідні дані роботи: Формальні моделі, методи та алгоритми

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)

1. Огляд предметної області.

2. Розробка структури.

3. Реалізація та тестування системи.

5. Дата видачі завдання

14.03.2024

КОНСУЛЬТАНТИ РОЗДІЛІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Розділ	Консультант (прізвище, ініціали та посада)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження існуючих аналогів	10.01.2024	Виконано
2	Аналіз предмету області	25.01.2024	Виконано
3	Проектування архітектури та створення дизайну веб-сайту	10.02.2024	Виконано
4	Програмна реалізація веб-сайту	10.03.2024	Виконано
5	Тестування веб-сайту	19.04.2024	Виконано
6	Формування висновків	30.04.2024	Виконано
7	Доопрацювання всіх розділів, відповідно вимогам	15.05.2024	Виконано
8	Оформлення графічного елемента та підготовка до захисту кваліфікаційної роботи	24.05.2024	Виконано

Студент

(підпис)

Кізера В.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Слабінога М.О

(прізвище та ініціали)

Перелік графічного матеріалу (з точним зазначенням к обов'язкових креслень):

Сторінка	Опис граф. матеріалу	Сторінка	Опис граф. матеріалу
16	Головна сторінка веб сайту IMDb.com	17	Розділ подкастів на сайті IMDb.com
18	“Танотометр” на Rotten Tomatoes	19	Головна сторінка megogo.net

21	Вибір жанру на Fandom.com	28	Карта сайту
29	БД failed_jobs	30	БД migrations
30	БД personal_access_tokens	31	БД password_reset_tokens
31	БД users	32	Діаграма “Реєстрація користувача”
34	Діаграма “Пошук фільмів”	35	Діаграма “Додавання коментаря”
36	Діаграма послідовності “Реєстрація користувача”	37	Діаграма послідовностей “Пошук фільмів”
40	Діаграма класів проекту	48	Головна сторінка проекту
50	Сторінка пошуку фільмів	52	Сторінка фільму
53	Реєстрація користувача		

АНОТАЦІЯ

Кваліфікаційна робота присвячена вивченню, аналізу та розробці веб-сайту для перегляду фільмів, серіалів та мультфільмів.

В першому розділі проведено аналіз веб-сайтів для перегляду фільмів, надано загальний опис майбутнього веб-сайту “MagMovie” та проведено аналіз існуючих аналогів - IMDb та MEGOGO, Rotten Tomatoes та інші. Виявлено їхні переваги та недоліки.

У другому розділі розглянуто структуру проекту, засоби які обиралися для створення сайту, розроблено архітектуру для веб-сайту “MagMovie”, а також макет. Проведено вибір технологій реалізації та розгортання.

В третьому розділі описано програмну реалізацію веб-сайту перегляду кінопродукції, починаючи від створення базової аплікації та налаштування тестового середовища, і закінчуючи вводом аплікації в експлуатацію.

В результаті, було створено повноцінний веб-сайт, що забезпечує ефективний перегляд, організацію та відображення популярних фільмів для глядачів, під категорію кожного.

КЛЮЧОВІ СЛОВА: HTML, CSS, JS, LARAVEL, PHP, WEB-ДИЗАЙН.

SUMMARY

The thesis is devoted to the study, analysis and development of a website for watching movies, series and cartoons.

In the first chapter, an analysis of websites for viewing movies was carried out, a general description of the future website "MagMovie" was provided, and an analysis of existing analogues - IMDb and MEGOGO, Rotten Tomatoes and others - was carried out. Their advantages and disadvantages are revealed.

In the second section, the structure of the project, the means chosen to create the site, the architecture for the "MagMovie" website, and the layout were developed. Implementation and deployment technologies have been selected.

The third chapter describes the software implementation of the movie viewing website, starting from the creation of the basic application and setting up the test environment, and ending with the application's commissioning.

As a result, a full-fledged website was created, which provides effective viewing, organization and display of popular films for viewers, under each category.

KEYWORDS: HTML, CSS, JS, LARAVEL, PHP, WEB DESIGN.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1. Огляд предмету області.....	11
1.2. Огляд існуючих рішень.....	15
1.3. Постановка задачі.....	22
Висновки до розділу 1.....	22
РОЗДІЛ 2. РОЗРОБКА СТРУКТУРИ.....	24
2.1. Вибір засобів для реалізації.....	24
2.2. Загальна структура системи.....	27
2.3 Структура проєкту.....	38
Висновки до розділу 2.....	41
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ.....	43
3.1. Реалізація системи.....	43
3.2. Тестування роботи системи.....	48
Висновки до розділу 3.....	55
ВИСНОВОК.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

HTML – стандартизована мова розмітки документів для перегляду веб сторінок у браузері.

CSS – спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду.

UI – засіб зручної взаємодії користувача з інформаційною системою.

JS – динамічна, об'єктно-орієнтована прототипна мова програмування.

Preview (прев'ю) – попередній перегляд.

БД – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних

Laravel – відкритим кодом PHP-фреймворк.

API – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

UML – уніфікована мова моделювання (UML) є візуалом загального призначення мова моделювання який призначений для надання стандартного способу візуалізації дизайну системи.

PHP – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб сервера.

ВСТУП

Актуальність теми. Розробка веб-каталогу фільмів залишається надзвичайно актуальною задачею в інженерії програмного забезпечення з урахуванням постійного зростання інтересу до відео контенту та потреби у зручному способі його організації та доступу. Такий каталог надає можливість користувачам швидко знаходити інформацію про різноманітні фільми, від жанру та режисера до акторського складу та рейтингу. Крім того, з відкриттям нових платформ для стрімінгу та розширенням кінематографічного контенту, розробка ефективного інструменту для пошуку і відсіву цього контенту стає ще більш важливою для забезпечення задоволення потреб користувачів.

Технічно, створення веб-каталогу фільмів вимагає комплексного підходу до проектування та розробки, включаючи архітектуру бази даних, розробку користувацького інтерфейсу, реалізацію алгоритмів рекомендацій та обробку великих обсягів даних. Важливо також забезпечити безпеку даних користувачів, забезпечити швидкий та ефективний пошук та фільтрацію фільмів із зростаючою кількістю записів у базі даних. Такі розробки відображають не лише потреби сучасного ринку розваг, а й виклики та можливості, які зустрічають інженери програмного забезпечення у сфері інформаційних технологій.

Метою роботи є розробка каталогу кінематографічної продукції на базі технологій HTML5, CSS3, PHP та фреймворку Laravel.

Досягнення мети передбачає **виконання таких задач:**

- аналіз предметної області;
- огляд аналогічних рішень та постановка задачі;
- вибір засобів для реалізації;
- розробка структури системи;
- реалізація та тестування системи.

Об'єктом дослідження є процес систематизації кінематографічної продукції у вигляді веб-орієнтованої системи.

Предметом дослідження є системи систематизації та відображення інформації про кінематографічну продукцію.

Методи, які використовувалися в даній роботі:

- порівняльний аналіз;
- методи проектування баз даних;
- методи інженерії програмного забезпечення;
- методи контролю якості програмного забезпечення.

Практичне значення даної роботи полягає в тому, що було розроблено веб-орієнтований каталог кінематографічної продукції, що дає змогу систематизувати та переглядати інформацію про кінофільми в зручний для кінцевого користувача спосіб.

Апробація результатів дослідження. Матеріали кваліфікаційної роботи були представлені на XI Міжнародній науковій конференції «Студентські наукові дискусії поза форматом», яка відбулася 11 квітня 2024 року в університеті Короля Данила.

Структура. Кваліфікаційна робота складається з трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 59 сторінок.

РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Огляд предмету області

Розробка веб-каталогу кінематографічної продукції - це тема, яка здобуває все більше популярності в сучасному світі. Швидкий розвиток інтернет-технологій і зростаючий інтерес до кіноіндустрії роблять такий проект надзвичайно актуальним.

Зручний доступ до інформації: веб-каталоги кінематографічної продукції створюють цифрове середовище, де користувачі можуть легко та швидко отримати доступ до великої кількості інформації про фільми та серіали. На цих платформах інформація структурована та легко доступна завдяки різним категоріям, фільтрам та пошуковим опціям. Користувачі можуть знайти описи фільмів, інформацію про акторів, трейлери та рецензії, все це забезпечується безпосередньо на одному веб-ресурсі, що робить процес пошуку та ознайомлення з кінематографічним контентом простим та зручним.

Просування та маркетинг: веб-каталоги кінематографічної продукції виконують важливу роль у маркетингових стратегіях кіностудій та розповсюдженні фільмів. Ці платформи дозволяють кіностудіям презентувати свої нові творіння широкій аудиторії, залучати увагу глядачів за допомогою привабливих матеріалів, таких як відеоогляди та постери, та створювати цікаві заходи, наприклад, віртуальні тури за кулісами або інтерактивні вікторини. Ці маркетингові ініціативи допомагають збільшувати свідомість про нові фільми та залучати глядачів до кінематографічного простору.

Підвищення інтересу до кінематографу: створення веб-каталогів кінематографічної продукції сприяє популяризації кіносфери та стимулює інтерес глядачів до кінематографічної культури. Ці платформи надають можливість відкривати нові творчі таланти, вивчати історію кіно та розвивати естетику фільмів. Широкий спектр представленого контенту, від класичних

шедеврів до сучасних кіноінновацій, робить кінематограф доступним та захоплюючим для глядачів різного віку та смаку.

Глобальний ринок: інтернет-каталоги кінематографічної продукції надають можливість просування фільмів та серіалів на міжнародному ринку. Ці платформи дозволяють розширити аудиторію та залучити глядачів з різних країн світу, забезпечуючи їм доступ до різноманітного кіносвіту. Культурні різноманіття та унікальність кіноіндустрій кожної країни стають доступними для глобальної аудиторії, що сприяє культурній обміні та розширює кінематографічну карту світу.

Потреби користувачів: пошук фільмів: Користувачі, орієнтовані на різноманітність кінематографічних творів, вимагають зручного та ефективного способу пошуку фільмів. У веб-каталогах кінематографічної продукції реалізовані різні функції пошуку, які дозволяють користувачам знаходити фільми за різними критеріями, такими як жанр, рік випуску, рейтинг, акторський склад, режисер та інші параметри. Швидкий та точний пошук сприяє задоволенню потреб користувачів та дозволяє їм ефективно використовувати час для вибору фільмів.

Інформація про фільми: користувачі очікують наявності детальної та достовірної інформації про фільми в каталогах. Окрім загальної інформації, такої як назва, жанр і рік випуску, вони хочуть ознайомитися з описом сюжету, рецензіями, рейтингами, акторським складом, технічними деталями та іншими важливими аспектами. Відповідна та комплексна інформація допомагає користувачам зробити обдумані вибори та забезпечує їм приємний та насичений кінематографічний досвід.

Перегляд трейлерів: можливість перегляду трейлерів фільмів вважається важливим аспектом для багатьох користувачів веб-каталогів. Трейлери дозволяють глядачам отримати уявлення про зміст та якість фільмів, а також викликають інтерес до перегляду. У веб-каталогах надається можливість швидко знаходити та переглядати трейлери безпосередньо на сторінці фільму, що додає зручності та ефективності користування.

Відгуки та рецензії: відгуки та рецензії від інших глядачів вважаються важливими для багатьох користувачів під час вибору фільмів. Вони дозволяють зробити обдумані вибори, отримати додаткову інформацію та оцінку якості фільму від реальних глядачів. У веб-каталогах надається можливість залишати відгуки, оцінки та коментарі, а також читати відгуки інших користувачів, що сприяє взаємодії та обміну думками в онлайн-спільноті кіноглядачів.

Підписка та обслуговування: Для задоволення потреб користувачів у веб-каталогах можуть бути реалізовані різноманітні сервіси підписки та обслуговування. Користувачі можуть бажати підписатися на сервіс, щоб отримувати оновлення про нові фільми, ексклюзивний контент або спеціальні пропозиції. Підписка дозволяє створити постійний зв'язок з аудиторією та підтримувати її зацікавленість у контенті.

Зміст веб-каталогу: пошук та фільтрація: функціонал пошуку та фільтрації веб-каталогів кінематографічної продукції є ключовим для користувачів, оскільки він дозволяє їм швидко та ефективно знаходити фільми, які відповідають їхнім вподобанням та інтересам. Користувачі можуть здійснювати пошук за різними критеріями, такими як жанр, рік випуску, рейтинг, країна виробництва, режисер та інші параметри. Додаткові фільтри дозволяють здійснювати точнішу фільтрацію результатів пошуку, що полегшує користувачам знаходження відповідних фільмів.

Детальна інформація про фільми: сторінки з детальною інформацією про кожен фільм включають в себе важливі дані, які допомагають користувачам зробити обдуманий вибір. Окрім загальної інформації, такої як назва і жанр фільму, користувачі можуть знайти опис сюжету, рейтинги та рецензії від критиків та інших глядачів, трейлери для ознайомлення зі змістом фільму, а також інформацію про акторський склад та інших учасників зйомок.

Рекомендації та підбірки: спеціальні розділи з рекомендаціями та підбірками дозволяють користувачам отримувати персоналізовані рекомендації на основі їхніх уподобань та історії переглядів. Ці рекомендації можуть бути складені за різними критеріями, такими як жанр, рейтинг, рік випуску, або

можуть бути спеціально підібрані для конкретних категорій глядачів, наприклад, "Новинки", "Кращі фільми року", "Класика жанру" та інші.

Користувацькі відгуки: можливість залишати відгуки та рецензії на фільми дозволяє користувачам висловлювати свої думки та враження від переглянутих фільмів, а також ділитися ними з іншими користувачами. Читання відгуків інших глядачів допомагає користувачам отримати різні точки зору та зробити більш об'єктивний вибір перед переглядом фільму.

Особистий кабінет: функціонал особистого кабінету надає користувачам зручність та можливість персоналізації веб-каталогу. Користувачі можуть створювати свої власні списки улюблених фільмів, підписуватися на оновлення та отримувати персоналізовані рекомендації, що сприяє збереженню і підтримці інтересу до кінематографічної продукції.

Система рекомендацій: алгоритми рекомендаційних систем аналізують історію переглядів та вподобань користувачів для надання персоналізованих рекомендацій. Ці алгоритми враховують різноманітні чинники, такі як жанр, рейтинг, історія переглядів, взаємодія зі схожими фільмами та інші, для надання користувачам рекомендацій, які найбільш відповідають їхнім вподобанням та інтересам.

Веб-каталог кінематографічної продукції відповідає потребам сучасного глядача, надаючи зручний доступ до інформації про фільми, допомагаючи в їх виборі та сприяючи популяризації кіноіндустрії.

Цільова аудиторія веб-каталогів кінематографічної продукції може бути досить різноманітною, оскільки фільми та серіали цікавлять різні групи людей різного віку, інтересів та культурних уподобань. Ось кілька ключових аспектів цільової аудиторії, які варто розглянути:

1. Молоді глядачі: студенти та молодь можуть бути активними користувачами веб-каталогів для пошуку нових фільмів та серіалів для відпочинку, навчання та розваг. Любителі кіно мають великий інтерес до різноманітних жанрів та класичних фільмів. Вони можуть активно

використовувати веб-каталоги для пошуку інформації про нові та старі твори мистецтва кіно.

2. Професійні кінокритики та ентузіасти: фахівці з кіно та серіалів, журналісти та ентузіасти можуть використовувати веб-каталоги для дослідження та обговорення нових творів кіно.

3. Сімейна аудиторія: батьки шукають фільми та серіали для всієї родини, які будуть цікавими та відповідатимуть віковим особливостям дітей.

4. Фахівці та студенти кінематографії: студенти кінематографії та фахівці з кінематографії можуть використовувати веб-каталоги для дослідження та аналізу фільмів, режисерів та кіноіндустрії.

5. Аудиторія з різними смаками та інтересами: веб-каталоги можуть надати вміст для різних смаків та інтересів, включаючи жанри від екшну та комедії до драми та фентезі.

1.2. Огляд існуючих рішень

Сучасний світ рясніє веб-каталогами, що пропонують доступ до інформації про кінематографічну продукцію. Завдяки доступності інтернету та безлічі гаджетів, глядачі отримали безпрецедентний доступ до кіно контенту. На тлі цих змін зростає й конкуренція на ринку веб-каталогів. З одного боку, це свідчить про наявність значного ринку та високий попит на подібні послуги. З іншого боку, це робить ретельний аналіз конкурентного середовища не лише актуальним, але й необхідним. У цій динамічній сфері успіх веб-каталогу залежить від його здатності задовольнити потреби користувачів та запропонувати їм унікальні переваги.

Метою цього розділу є проведення порівняльного аналізу 4 популярних веб-каталогів з кінематографічною продукцією. Ми дослідимо їх функціонал, дизайн, контент, цільову аудиторію, а також визначимо їхні сильні та слабкі сторони.

IMDb. Як зображено на рисунку 1.1 головну сторінку веб-каталогу з кінематографічною продукцією те що вона складається з каруселі з популярними фільмами та серіали, пропонує доступ до рейтингів фільмів, списків очікування, рецензій. Сайт надає розширений список за пошуком продукції за акторами, роками виходу, режисерами.

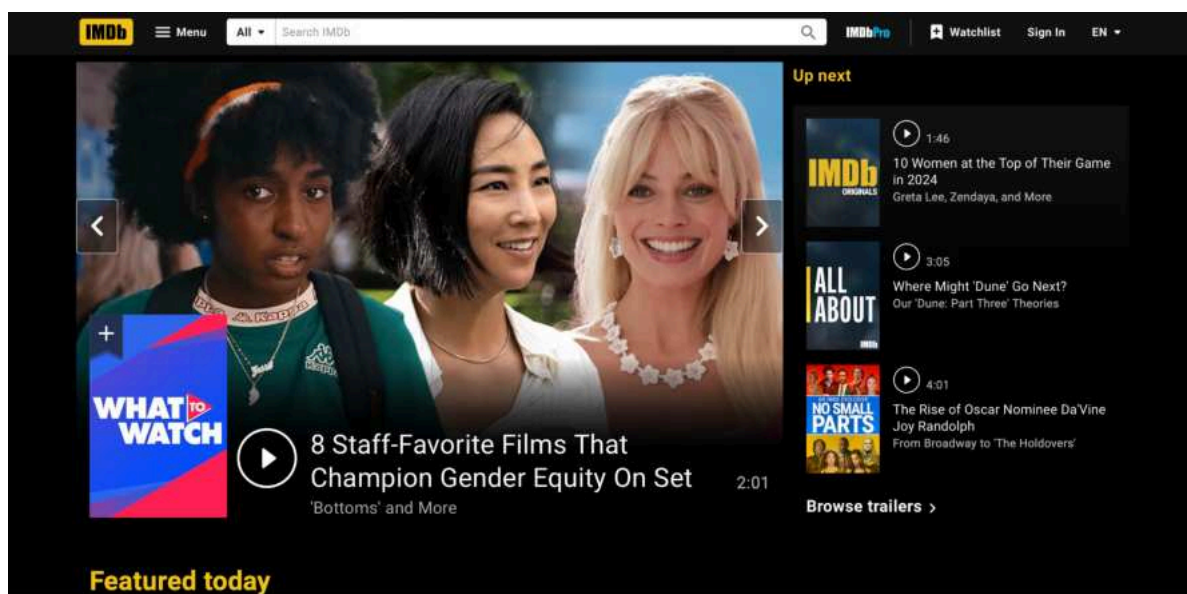


Рисунок 1.1 – Головна сторінка веб сайту IMDb.com

Дизайн сайту витриманий у жовтих тонах, які символізують довіру, авторитет та знання. Це відчуття підсилюється чіткими та зрозумілими шрифтами, які не втомлюють очі, а також інтуїтивно зрозумілою навігацією.

Навігація по сайту проста та зручна. Основні розділи – "Фільми", "Серіали", "Актори", "Новини" – розташовані у верхній панелі. Бічна панель пропонує фільтрацію по жанру, року та країні. А якщо ви знаєте, що шукаєте, скористайтесь пошуком, який доступний у будь-який момент.

IMDb.com володіє найобширнішою базою даних про кінематограф. Тут ви знайдете інформацію про мільйони фільмів, серіалів, акторів, режисерів. Описи сюжетів, трейлери, фото, постери, рейтинги та рецензії від критиків і глядачів, списки очікування, списки найкращих фільмів, фільмографії акторів, біографії та цікаві факти – все це на вас чекає на IMDb.com.

Сайт також пропонує можливість створення списків. Ви можете зберегти фільми, які хочете подивитися, створювати власні рейтинги та ділитися ними з друзями.

Однак, у сайту є й свої недоліки. Деяким користувачам інтерфейс може здатися складним, адже тут дійсно багато інформації (рис. 1.2). Реклама також може відволікати від контенту, проте реклама дуже гарно вписана в систему і вона не є нав'язливою. І, на жаль, сайт доступний лише декількома іноземними мовами, для користувачів з України це може стати значною проблемою, доведеться перекладати сайт, а це в свою чергу не завжди правильно перекладаєть.

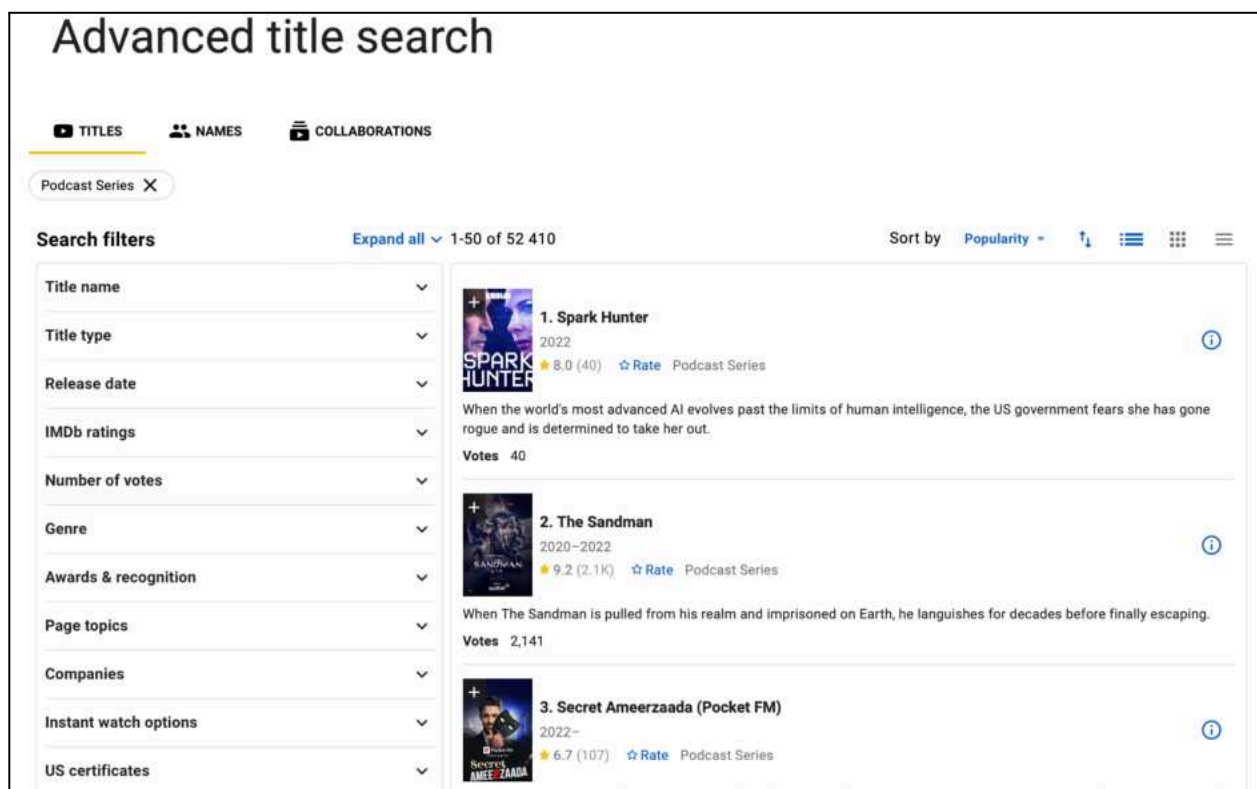


Рисунок 1.2 – Розділ подкастів на сайті IMDb.com

Rotten Tomatoes. Rotten Tomatoes - славиться революційною системою "Танометра", яка збирає рецензії критиків та глядачів, показуючи "рейтинг свіжості" для кожного фільму та серіалу. Цей рейтинг дає чітке уявлення про якість контенту, допомагаючи користувачам зробити свідомий вибір.

Сайт Rotten Tomatoes має інтуїтивно зрозумілий інтерфейс, що робить його доступним для користувачів з будь-яким рівнем досвіду. Зручні фільтри та пошук дозволяють швидко знайти потрібний фільм або серіал, а система навігації забезпечує легкий доступ до всієї інформації.

Пропонує широкий спектр інформації про кінематограф, окрім рейтингів та рецензій. Користувачі можуть ознайомитися з трейлерами, постерами, сюжетами, біографіями акторів та режисерів, а також з даними про касові збори та фестивалі.

З'являється унікальна можливість поділитися своїми враженнями, залишати рецензії та оцінювати фільми. Спілкуючись з іншими користувачами, ви можете знайти однодумців, отримати нові рекомендації та розширити свої знання про кінематограф.

The screenshot displays the Rotten Tomatoes website. At the top, there is a search bar and navigation links for MOVIES, TV SHOWS, SHOP, NEWS, and SHOWTIMES. Below the navigation is a 'TRENDING ON RT' section with links to 'Play Movie Trivia', 'Dune: Part Two', 'Kung Fu Panda 4', and 'Ricky Stanicky'. The main content area is divided into three columns: 'IN THEATERS', 'STREAMING MOVIES', and 'TV SHOWS'. The 'IN THEATERS' column lists movies with their respective Tomatometer scores, such as 'Labyrinth' (77%), 'Kung Fu Panda 4' (71%), 'Cabrin! (90%), 'Imaginary' (24%), 'Damsel' (57%), 'Glitter & Doom' (67%), 'Space: The Longest Goodbye' (81%), 'American Dreamer' (36%), 'Sibs of Pressure' (80%), 'Accidental Texan' (58%), 'Night Shift' (0%), and 'High & Low - John Galliano' (71%). Below this is a 'COMING SOON' section with release dates for 'Forty-Seven Days with Je...' (Mar 11), 'The Primevals' (Mar 11), 'Blackout' (Mar 13), 'Love Lies Bleeding' (Mar 15), and 'The American Society of ...' (Mar 15). The right side of the page features a large video player showing a trailer for 'SPACEMAN' with a play button. Below the video is a card for the movie 'SPACEMAN', rated R (2024, Sci-fi/Adventure, 1h 48m). The card displays a Tomatometer score of 52% (based on 116 reviews) and an Audience Score of 65% (based on 500+ ratings).

Рисунок 1.3 – “Танотометр” на Rotten Tomatoes

Rotten Tomatoes визнаний авторитетним джерелом інформації про кіно у світовій спільноті. Сайт використовується критиками, режисерами, акторами та іншими фахівцями кіноіндустрії.

Однак, у Rotten Tomatoes є й свої особливості. Деякі функції доступні лише після реєстрації. Сайт доступний лише англійською мовою, що може бути незручним для деяких користувачів. Не всі користувачі погоджуються з системою "Томатометра", вважаючи її не завжди об'єктивною.

Незважаючи на деякі недоліки, Rotten Tomatoes залишається цінним ресурсом для любителів кіно. Це джерело інформації, думок та спілкування, що допоможе вам зробити правильний вибір та отримати максимум задоволення від перегляду.

Загалом сайт можна назвати мережею кіноіндустрії, на ньому є все від перегляду фільмів до покупки сувенірів з фірмовим мерчем Rotten Tomatoes, та навіть використовувати як соціальні мережу.

MEGOGO. На відміну від двох попередніх порівнянь, цей сервіс що я зараз розглядаю являється українським, що пропонує широкий спектр контенту, включаючи фільми, серіали, мультфільми, телепередачі, спортивні трансляції та аудіоконтент.

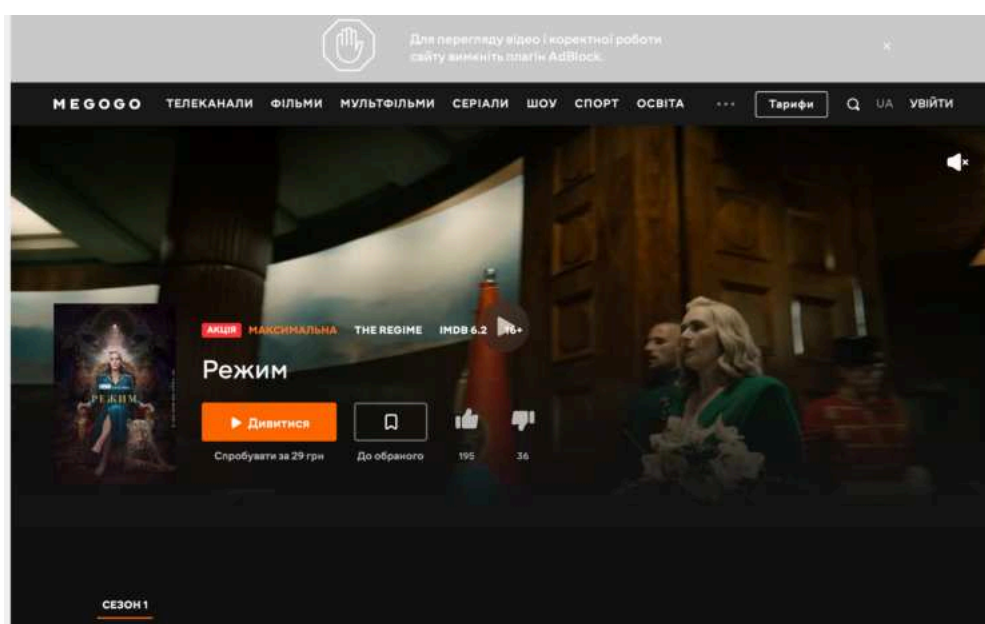


Рисунок 1.4 – Головна сторінка megogo.net

Для деяких користувачів це викликає обуреність, так як реклама заважає переглядати улюблений контент з насолодою. Звичайно що реклама потрібна так як вона допомагає авторам контенту отримувати винагороду, за проведену роботу, проте потрібно зробити так щоб реклама корентно працювала з сайтом та не привертала зайву увагу.

Megogo.net надає великі функціональні можливості, так як бібліотека контенту, яка доступна як за підпискою так і безкоштовно. Про підписку можна сказати що вона підходить для кожного, так як пропонується декілька її пакетів, вона надає доступ до ексклюзивного контенту, а саме в HD-якості та має інші переваги. Проте саме одним з неділоком, є ціни на підписку, на мою думку їх можна було зробити меншими і отримати додаткову кількість людей, які матимуть змогу користуватися сервісом.

Загалом можна зробити ось такий висновок, що:

1. Сильною стороною даного сервісу є: різноманіття пакетів підписки, велика бібліотека контенту, доступність безкоштовного контенту (проте в маленькій кількості), мультиплатформність, та звичайно українська мова.
2. Слабкою стороною є: ексклюзивні фільми та серіали доступні лише за підпискою, дорогі ціни на підписки, не всі канали доступні у HD-якості навіть за підпискою, можуть виникати технічні проблеми час від часу, так як уже було продемонстровано (рис. 1.4), що сайт некоректно відображатиме роботу.

Для українських сервісів, які надають послуги з кінематографічною продукцією, зараз спостерігається велика проблема, викликана широкомасштабною війною, те що багато людей не готові витратити свої кошти на розваги. Тому важливо для таких сервісів працювати не тільки на свою аудиторію, а також на закордонну, та впроваджувати інші мови сервісу. Насправді зараз ринок переживає серйозні труднощі з цим, тому бізнес повинен використовувати всі можливі дії, щоб такі сервіси розвивалися, та ставали популярними, а піратські сайти навпаки ставали нікому не потрібними, так як це рахується злочином.

Fandom. Останнім сервісом для аналізу є Fandom - платформа що спеціалізується на кураторському доборі артхаусних, іноземних та незалежним фільмів. Даний сервіс недоступний в українському регіоні, тому я вирішив провести аналіз, якими відмінностями користуються люди за кордоном, на відміну від наших.

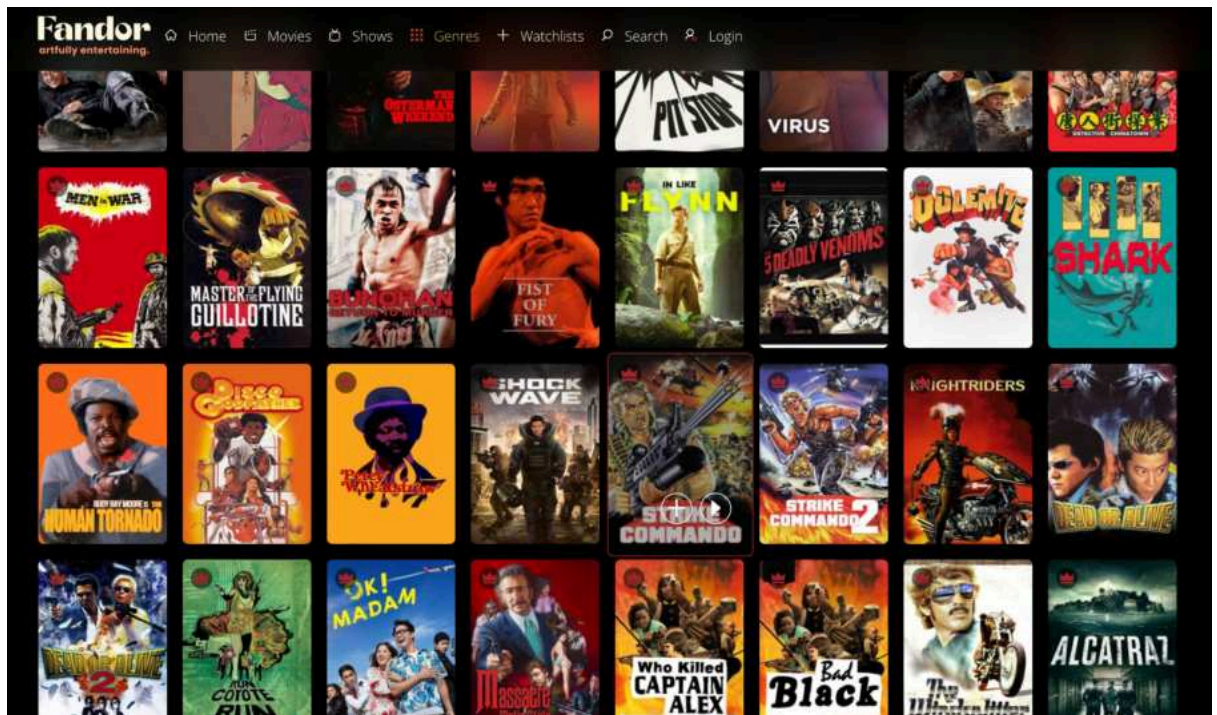


Рисунок 1.5 – Вибір жанру на Fandom.com

Найбільшим недоліком для мене стало, те що при виборі жанру фільму, коли користувач наводиться на свій вибір (рис. 1.5), є те що не показується короткий опис фільму або хоча б назва, так назва є на прев'ю фільму, проте для користувачів це є значною проблемою. Також слабкою стороною є невеликий вибір класичних та популярних фільмів, немає можливості скачати фільм та переглядати оффлайн.

На відміну від недоліків та слабких сторін, платформа має безліч переваг, які я буду використовувати також у своєму сервісі під час розробки проєктування та моделювання. Перевагами являються: висока якість трансляції, даний сервіс надає високу якість на кожний фільм чи серіал. Ціни на підписку

доступні для громадян, тому кожен має можливість скористатися нею, та отримати додаткові можливості. Безумовно важливо підкреслити, що цей сервіс є кросплатформовий та дає змогу користуватися на кожному пристрої не важливо чи це TV чи Android. Контент регулярно оновлюється та додаються нові озвучки (для англомовних людей, з різними акцентами).

Структура сайту дуже добре організована, для людей які перший раз заходять на сервіс, не буде проблемою зрозуміти що саме надає сайт та де воно знаходиться, найважливішим є те що зручно користуватися, а отже і користувачі будуть ставити хороші відгук про платформу.

1.3. Постановка задачі

Виходячи з проведеного аналізу було прийнято рішення розробляти каталог кінематографічної продукції з такими можливостями: головна сторінка сайту, пошук фільмів за фільтрами, вхід в обліковий запис, під'єднання API для витягування фільмів та залишок коментарів.

Сайт буде постійно оновлюватися новою інформацією про фільми, рецензіями та іншими матеріалами, щоб користувачі завжди були в курсі останніх новинок кіно світу.

Висновки до розділу 1

У першому розділі дипломної роботи було проведено глибоке дослідження обраного напрямку в кінематографічній продукції. Автор детально проаналізував існуючі рішення, що дозволило зрозуміти сучасний стан і тенденції розвитку галузі. Проведене дослідження стало основою для формування чіткої постановки задачі, що є важливим кроком у подальшій роботі над проектом.

Крім того, був здійснений огляд предметної області, що включає аналіз ключових концепцій та технологій, які використовуються в кінематографі. Цей

огляд допоміг визначити основні виклики та можливості, які слід враховувати при розробці власних рішень. Таким чином, перший розділ заклав міцну основу для подальшої роботи, забезпечивши необхідні теоретичні знання та практичні висновки для успішного виконання завдань проекту.

РОЗДІЛ 2. РОЗРОБКА СТРУКТУРИ

2.1. Вибір засобів для реалізації

Створення якісного та функціонального веб-каталогу з кінематографічною продукцією потребує ретельного підбору технологій, які забезпечують оптимальне поєднання зручності використання, візуальної привабливості та широких можливостей. Для реалізації даного проекту було обрано сучасний стек технологій, який дозволяє досягти поставлених цілей.

Фундаментальні інструменти:

1. HTML5: для побудови структури сайту та визначення його основних елементів, таких як заголовки, параграфи, списки, форми, зображення та інше, було обрано мову розмітки HTML5. Її семантична розмітка забезпечує чітке розуміння контенту як для пошукових систем, так і для користувачів з обмеженими можливостями. Використання тегів дозволяє структурувати сторінки відповідно до їх змісту та ролі, що покращує доступність та індексацію сайту.

2. CSS3: для надання сайту привабливого та унікального вигляду, управління стилями елементів HTML, такими як кольори, шрифти, розміщення, анімація та інше, було обрано мову каскадних таблиць стилів CSS3. Застосування CSS3 дозволило створити адаптивний дизайн, який коректно відображається на різних пристроях, від комп'ютерів до смартфонів та планшетів. Використання CSS-фреймворків, таких як Bootstrap або Foundation, значно спростило процес стилізації та забезпечило кросбраузерну сумісність.

Динаміка та інтерактивність:

1. JavaScript: для додавання інтерактивності та динамічного функціоналу сайту, обробки дій користувача, оновлення контенту без перезавантаження сторінки та реалізації складних ефектів, було обрано мову

програмування JavaScript. За допомогою JavaScript можна реалізувати такі функції, як:

- a. Валідація форм: перевірка введених даних користувачем на коректність перед їх відправкою на сервер.
- b. Асинхронні запити (AJAX): отримання даних з сервера без перезавантаження сторінки, що дозволяє оновлювати контент динамічно та створювати більш плавний користувацький досвід.
- c. Анімація: додавання візуальних ефектів, таких як плавні переходи, згасання та інші, що роблять сайт більш привабливим та інтерактивним.

2. jQuery: для спрощення роботи з JavaScript та забезпечення кроссбраузерної сумісності, було обрано бібліотеку jQuery. Вона надає зручний API для маніпулювання DOM-елементами, обробки подій, анімації та AJAX-запитів, що значно скорочує час розробки та спрощує код.

Ефективне представлення контенту:

1. Slick Slider: для реалізації адаптивних та функціональних слайдерів, які дозволяють ефективно представляти кінематографічну продукцію, було обрано бібліотеку Slick Slider. Цей інструмент надає широкий спектр налаштувань, включаючи:

- a. Кількість слайдів: можливість визначити кількість слайдів, які відображаються одночасно.
- b. Швидкість перемикавання: регулювання швидкості, з якою відбувається перехід між слайдами.
- c. Ефекти переходів: вибір з різноманітних ефектів переходів, таких як згасання, зсув, обертання та інші.
- d. Автопрокрутка: налаштування автоматичного перемикавання слайдів через заданий проміжок часу.
- e. Навігація: додавання кнопок або стрілок для ручного перемикавання слайдів.

Серверна частина та зберігання даних:

1. PHP: для обробки даних на сервері, взаємодії з базою даних, реалізації логіки роботи сайту та створення динамічних сторінок, було обрано мову програмування PHP. Її можливості включають:

- a. Обробка форм: отримання даних, введених користувачем в форми, та їх обробка на сервері.
- b. Аутентифікація користувачів: реалізація системи реєстрації та входу користувачів, що дозволяє їм зберігати свої дані та налаштування.
- c. Взаємодія з базою даних: виконання запитів до бази даних для отримання, додавання, зміни та видалення даних.

2. Laravel: для спрощення розробки бекенду та підвищення його ефективності було обрано PHP-фреймворк Laravel. Він надає такі переваги:

- a. MVC-архітектура: структурування коду за принципом "Модель-Вид-Контролер", що полегшує розробку та підтримку проекту.
- b. Eloquent ORM: зручний інтерфейс для роботи з базою даних, що спрощує виконання запитів та маніпулювання даними.
- c. Blade шаблонізатор: потужний шаблонізатор, який дозволяє легко створювати динамічні сторінки з повторно використовуваними компонентами.
- d. Безпека: вбудовані механізми для захисту від XSS, CSRF та інших типів атак.

3. База даних: для зберігання інформації про фільми, користувачів та інший контент було обрано реляційну базу даних, таку як MySQL. Її структура дозволяє:

- a. Структуроване зберігання даних: організація даних в таблиці з колонками та рядками, що полегшує пошук та обробку.
- b. Можливість пошуку та фільтрації: виконання складних запитів для пошуку та фільтрації даних за різними критеріями.
- c. Цілісність та безпека даних: забезпечення цілісності та безпеки даних шляхом використання механізму транзакцій та контролю доступу.

Для отримання інформації про фільми, мій проект використовує API The Movie Database (TMDb) - популярний сервіс, що надає доступ до великої бази

даних фільмів, серіалів, акторів та знімальних груп. API TMDb дозволяє отримувати різноманітні дані, такі як:

1. Інформація про фільми (назва, опис, дата виходу, жанри, актори, знімальна група, постери, трейлери), популярні та найрейтинговіші фільми.
2. Фільми, що йдуть зараз в кінотеатрах. Пошук фільмів за назвою, жанром, роком випуску та іншими критеріями. Інформація про акторів та знімальні групи.

Використання API TMDb дозволяє мені отримати доступ до актуальної та повної інформації про фільми, не потребуючи створення власної бази даних фільмів.

2.2. Загальна структура системи

Під час розробки проекту веб-каталогу з кінематографічною продукцією було приділено особливу увагу створенню ефективної та зручної структури сайту, яка забезпечує інтуїтивну навігацію та легкий доступ до необхідної інформації для користувачів.

Структура сайту була розроблена з урахуванням наступних ключових принципів:

1. Логічна організація контенту: інформація про фільми, акторів, режисерів та інші аспекти кінематографу організована в логічні розділи та підрозділи, що полегшує пошук та доступ до потрібного контенту.
2. Зручна навігація: система навігації сайту, включаючи меню, пошуковий рядок та інші елементи, дозволяє користувачам легко переміщатися між сторінками та розділами та швидко знаходити потрібну інформацію.
3. Адаптивний дизайн: структура сайту адаптується до різних пристроїв, від комп'ютерів до смартфонів та планшетів, забезпечуючи оптимальний користувацький досвід на будь-якому екрані.

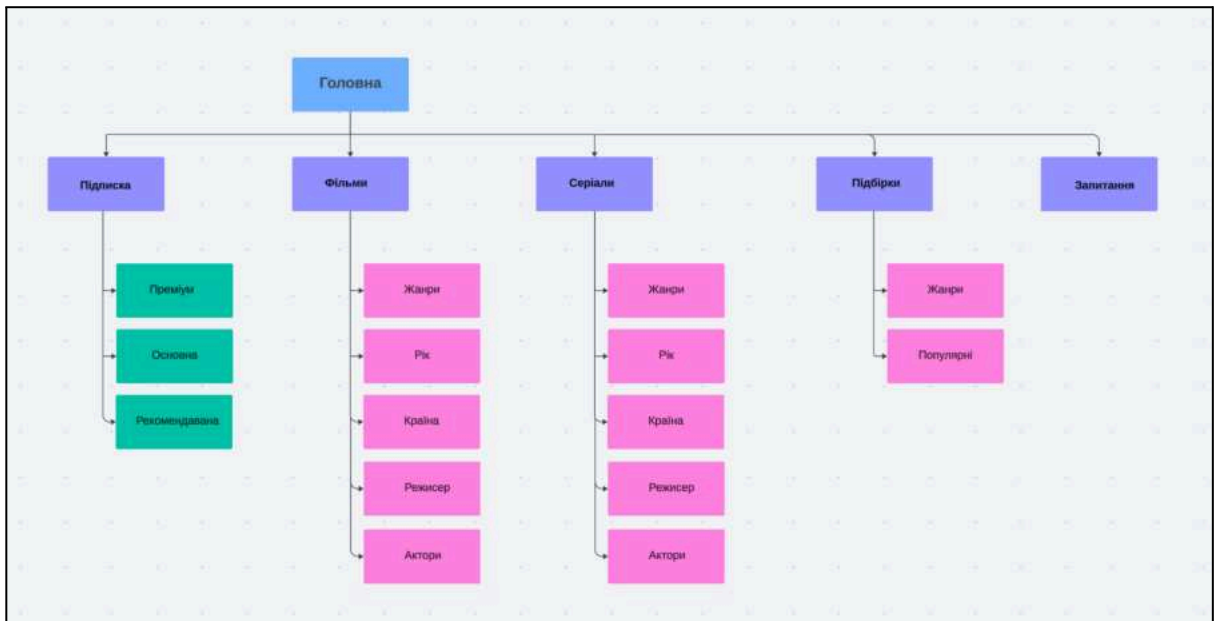


Рисунок 2.1 – Карта сайту

Центральним елементом сайту є головна сторінка, що слугує точкою входу та надає доступ до всіх основних розділів. Тут користувачі можуть знайти банери з популярними фільмами і серіалами, рекомендації, новинки та інший актуальний контент. Головна сторінка також забезпечує швидкий доступ до пошуку та основних категорій контенту.

Розділ "Підписка" присвячений вибору та оформленню підписки на сервіс. Тут користувачі можуть ознайомитися з різними типами підписок (преміум, основна, рекомендована), їх вартістю та перевагами, а також оформити підписку, що відповідає їхнім потребам.

Розділи "Фільми" та "Серіали" мають схожу структуру, що забезпечує послідовність та зручність для користувача. В кожному розділі користувачі можуть знайти каталог фільмів або серіалів з можливістю фільтрації пошуку за різними критеріями, такими як жанр, рік випуску, країна виробництва, режисер та актори. Функціональність дозволяє швидко знайти потрібний контент та ознайомитися з детальною інформацією про кожен фільм або серіал.

Розділ "Підбірки" містить тематичні добірки фільмів та серіалів, створені редакторами або користувачами. Користувачі можуть переглядати підбірки за

жанрами та популярністю, що дозволяє знаходити цікавий контент відповідно до своїх уподобань.

Розділ "Запитання" надає інформацію про сайт, правила користування, контакти, а також відповіді на часті запитання. Тут користувачі можуть знайти необхідну інформацію про роботу сервісу та звернутися за допомогою до служби підтримки.

Для забезпечення функціональності веб-каталогу з кінематографічною продукцією, включаючи аутентифікацію користувачів, зберігання коментарів та іншої інформації, використовується реляційна база даних MySQL. База Даних дозволяє структуровано зберігати дані, забезпечувати їх цілісність та безпеку, а також ефективно отримувати та обробляти інформацію.

В рамках проекту була розроблена база даних "movies_laravel", яка складається з наступних таблиць:

Field	Type	Null	Key	Default	Extra
id	bigint unsigned	NO	PRI	NULL	auto_increment
uuid	varchar(255)	NO	UNI	NULL	
connection	text	NO		NULL	
queue	text	NO		NULL	
payload	longtext	NO		NULL	
exception	longtext	NO		NULL	
failed_at	timestamp	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED

Рисунок 2.2 – БД failed_jobs

Ця таблиця зберігає інформацію про невдалі завдання (jobs):

1. id: унікальний ідентифікатор завдання (первинний ключ).
2. uuid: універсальний унікальний ідентифікатор завдання.
3. connection: назва з'єднання з чергою завдань.
4. queue: назва черги завдань.
5. payload: дані завдання (серіалізовані).
6. exception: інформація про помилку, що виникла під час виконання завдання.
7. failed_at: дата та час, коли завдання було визнано невдалим.

Field	Type	Null	Key	Default	Extra
id	int unsigned	NO	PRI	NULL	auto_increment
migration	varchar(255)	NO		NULL	
batch	int	NO		NULL	

Рисунок 2.3 – БД migrations

Ця таблиця зберігає інформацію про виконані міграції бази даних:

1. id: унікальний ідентифікатор міграції (первинний ключ).
2. migration: назва файлу міграції.
3. batch: номер пакету міграцій.

Field	Type	Null	Key	Default	Extra
id	bigint unsigned	NO	PRI	NULL	auto_increment
tokenable_type	varchar(255)	NO	MUL	NULL	
tokenable_id	bigint unsigned	NO		NULL	
name	varchar(255)	NO		NULL	
token	varchar(64)	NO	UNI	NULL	
abilities	text	YES		NULL	
last_used_at	timestamp	YES		NULL	
expires_at	timestamp	YES		NULL	
created_at	timestamp	YES		NULL	
updated_at	timestamp	YES		NULL	

Рисунок 2.4 – БД personal_access_tokens

Ця таблиця зберігає персональні токени доступу для авторизації API:

1. id: унікальний ідентифікатор токена (первинний ключ).
2. tokenable_type: тип об'єкта, якому належить токен (наприклад, "App\Models\User").
3. tokenable_id: ідентифікатор об'єкта, якому належить токен.
4. name: назва токена.
5. token: унікальний токен доступу (хешований).
6. abilities: список дозволів, наданих токену (опціонально).
7. last_used_at: дата та час останнього використання токена (опціонально).

8. expires_at: дата та час закінчення терміну дії токена (опціонально).
9. created_at: дата та час створення токена.
10. updated_at: дата та час останнього оновлення токена.

Field	Type	Null	Key	Default	Extra
email	varchar(255)	NO	PRI	NULL	
token	varchar(255)	NO		NULL	
created_at	timestamp	YES		NULL	

Рисунок 2.5 – БД password_reset_tokens

Ця таблиця зберігає токени для скидання пароля користувачами:

1. email: електронна адреса користувача (первинний ключ).
2. token: унікальний токен для скидання пароля.
3. created_at: дата та час створення токена.

Field	Type	Null	Key	Default	Extra
id	bigint unsigned	NO	PRI	NULL	auto_increment
name	varchar(255)	NO		NULL	
email	varchar(255)	NO	UNI	NULL	
email_verified_at	timestamp	YES		NULL	
password	varchar(255)	NO		NULL	
remember_token	varchar(100)	YES		NULL	
created_at	timestamp	YES		NULL	
updated_at	timestamp	YES		NULL	
avatar	varchar(255)	YES		NULL	

Рисунок 2.6 – БД users

Ця таблиця зберігає інформацію про користувачів системи, включаючи:

1. id: унікальний ідентифікатор користувача (первинний ключ).
2. name: ім'я користувача.
3. email: електронна адреса користувача (унікальна).
4. email_verified_at: дата та час підтвердження електронної адреси.
5. password: захешований пароль користувача.

6. remember_token: токен для запам'ятовування сесії користувача.
7. created_at: дата та час створення запису.
8. updated_at: дата та час останнього оновлення запису.
9. avatar: посилання на аватар користувача (опціонально).

Взаємозв'язки між таблицями: таблиця users має зв'язок "один-до-багатьох" з таблицями password_reset_tokens та personal_access_tokens за полем email та tokenable_id відповідно.

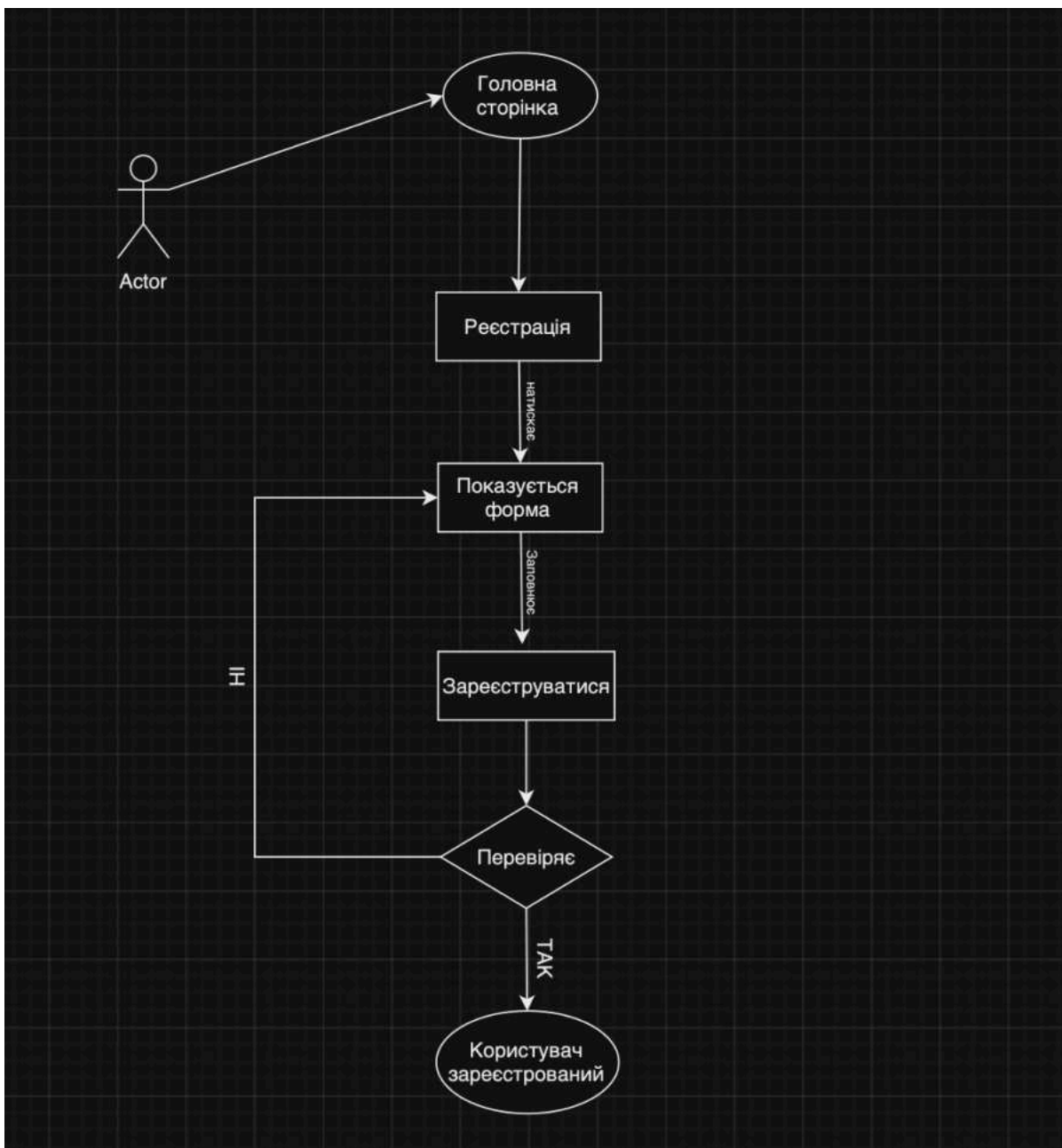


Рисунок 2.7 – Діаграма “Реєстрація користувача”

Діаграма сценарію "Реєстрація користувача" візуалізує процес створення нового облікового запису на сайті. Процес розпочинається з того, що користувач заходить на головну сторінку веб-сайту.

1. Головна сторінка: користувач потрапляє на головну сторінку сайту, де йому доступні різні опції та функції.
2. Реєстрація: користувач обирає опцію "Реєстрація", яка зазвичай представлена у вигляді кнопки або посилання.
3. Натискає: користувач активує опцію "Реєстрація" шляхом натискання на кнопку або посилання.
4. Показується форма: система реагує на дію користувача та відображає форму реєстрації. Ця форма містить поля для введення необхідної інформації, такої як ім'я користувача, електронна адреса та пароль.
5. Заповнює: користувач вводить свої дані у відповідні поля форми реєстрації.
6. Зареєструватися: після заповнення всіх необхідних полів, користувач підтверджує свою реєстрацію, натиснувши кнопку "Зареєструватися" або аналогічну.
7. Перевірка: система здійснює перевірку введених даних. Це може включати перевірку формату електронної адреси, складності пароля та унікальності імені користувача.
8. Рішення: на основі результатів перевірки система приймає рішення.
 - a. Так: якщо всі дані введені коректно та відповідають вимогам системи, реєстрація вважається успішною.
 - b. Ні: якщо виявлені помилки або невідповідності, користувачу повідомляється про проблему, і він має можливість виправити дані та повторити спробу реєстрації.
9. Користувач зареєстрований: після успішної перевірки система створює новий обліковий запис для користувача. Тепер користувач має доступ до всіх функцій та можливостей сайту, які вимагають авторизації.

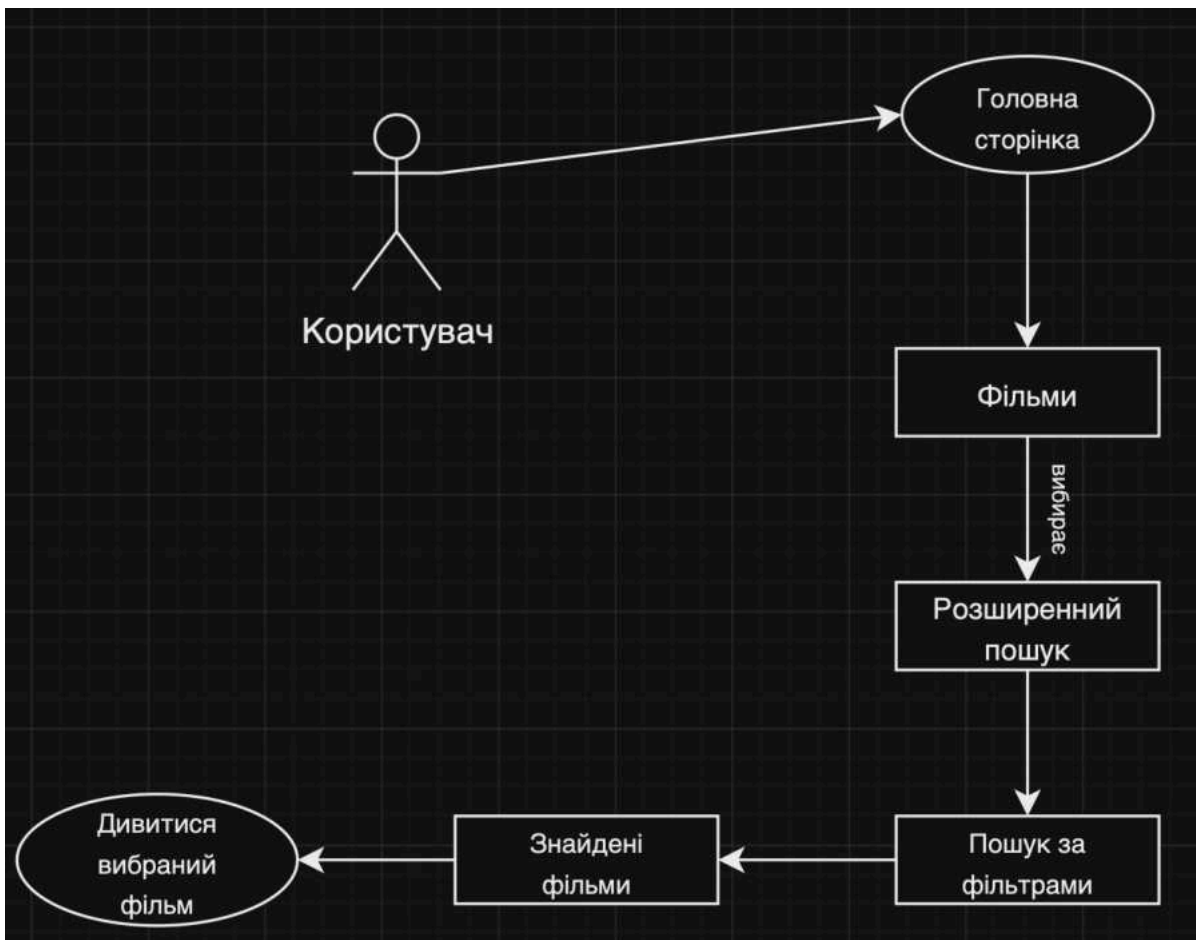


Рисунок 2.8 – Діаграма “Пошук фільмів”

Діаграма сценарію "Пошук фільмів" візуалізує процес пошуку та вибору фільмів користувачем на веб-сайті.

1. Головна сторінка: подорож користувача розпочинається з головної сторінки веб-сайту, де він має доступ до різних розділів та функціоналу.

2. Фільми: користувач виявляє та обирає розділ "Фільми", що містить каталог доступних фільмів для перегляду. Цей розділ може бути представлений у вигляді окремої сторінки, категорії в меню або виділеного блоку на головній сторінці.

3. Вибирає: користувач активно обирає розділ "Фільми", висловлюючи бажання переглядати фільми та досліджувати доступний контент.

4. Розширений пошук: у розділі "Фільми" користувачу надається можливість скористатися розширеним пошуком, щоб знайти фільми, які відповідають його інтересам та вподобанням.

5. Пошук за фільтрами: користувач має можливість застосувати різні фільтри, такі як жанр, рік випуску, режисер, актори або ключові слова, щоб звузити результати пошуку та знайти релевантні фільми.

6. Знайдені фільми: після застосування фільтрів система відображає список фільмів, які відповідають заданим критеріям пошуку. Цей список може включати назви фільмів, постери, короткі описи та іншу відповідну інформацію.

7. Дивитися вибраний фільм: користувач переглядав список знайдених фільмів та обирає той, який він бажає переглянути. Після вибору фільму система перенаправляє користувача на сторінку фільму, де він може розпочати перегляд.

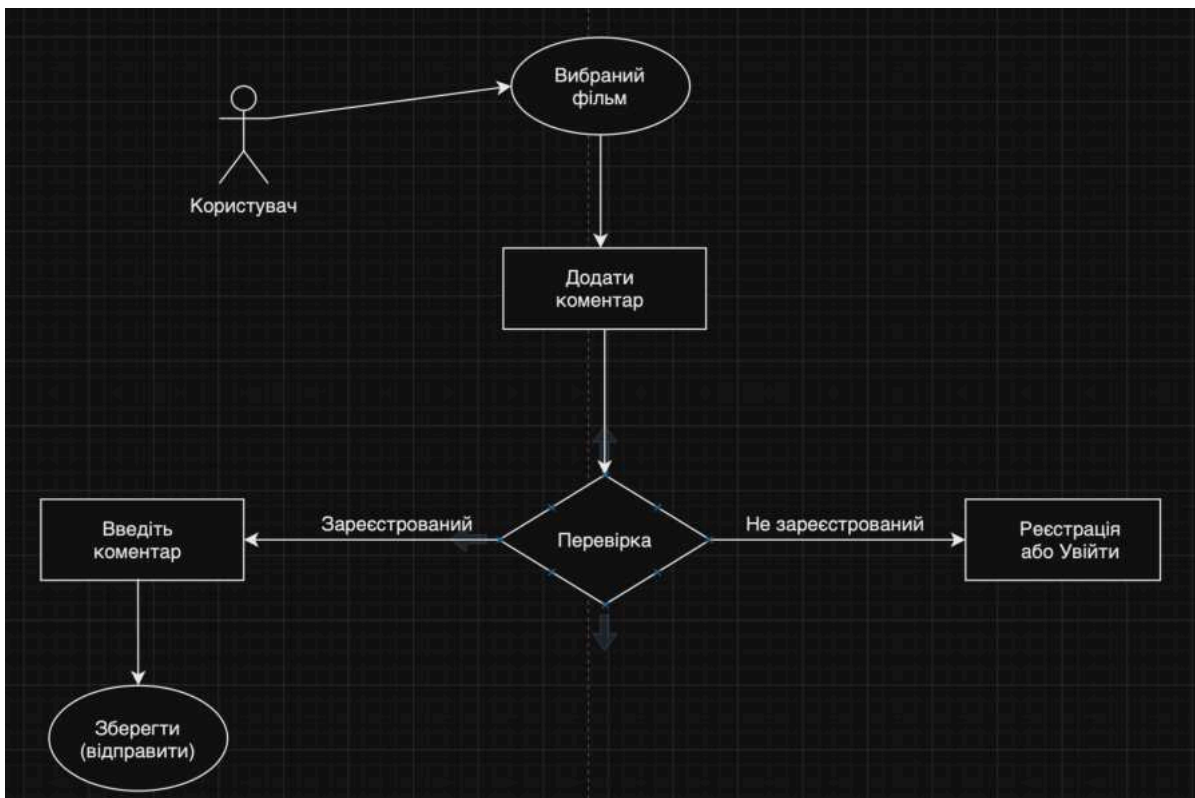


Рисунок 2.9 – Діаграма “Додавання коментаря”

Діаграма реалізує сценарій "Додавання коментаря до фільму" та демонструє два можливі шляхи залежно від того, чи зареєстрований користувач на сайті.

1. Вибраний фільм: користувач знаходиться на сторінці вибраного фільму, до якого він хоче додати коментар.
2. Додати коментар: користувач виявляє та активує функцію "Додати коментар", висловлюючи бажання поділитися своєю думкою про фільм.
3. Перевірка: система здійснює перевірку статусу реєстрації користувача, щоб визначити подальший шлях.
4. Зареєстрований: якщо користувач вже зареєстрований та авторизований на сайті, він може одразу перейти до додавання коментаря.
5. Ввести коментар: користувач вводить текст свого коментаря у відповідне поле.
6. Зберегти (відправити): користувач підтверджує додавання коментаря, натиснувши кнопку "Зберегти" або "Відправити". Коментар публікується на сторінці фільму та стає видимим для інших користувачів.
7. Не зареєстрований: якщо користувач не зареєстрований, система пропонує йому пройти реєстрацію або авторизуватися.
8. Реєстрація або Увійти: користувач має можливість створити новий обліковий запис або увійти в існуючий, щоб отримати доступ до функції додавання коментарів.

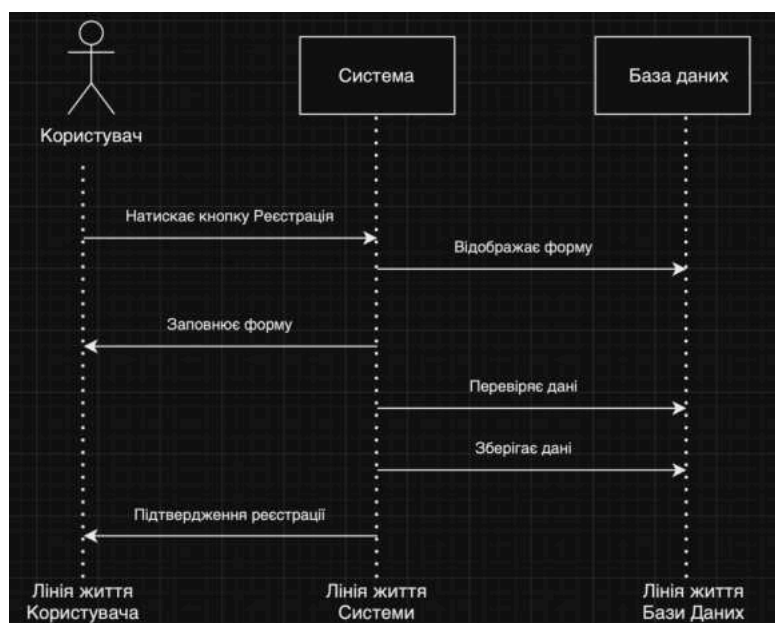


Рисунок 2.10 – Діаграма послідовності “Реєстрація користувача”

1. Користувач натискає кнопку "Реєстрація": користувач ініціює процес реєстрації, натиснувши кнопку або посилання "Реєстрація" на сайті.
2. Система відображає форму: у відповідь на дію користувача система відображає форму реєстрації, якимістить поля для введення необхідних даних, таких як ім'я користувача, електронна адреса та пароль.
3. Користувач заповнює форму: користувач вводить свої дані у відповідні поля форми реєстрації.
4. Система перевіряє дані: після того як користувач заповнив форму та натиснув кнопку "Зареєструватися", система здійснює перевірку введених даних на коректність та відповідність вимогам.
5. Система зберігає дані: якщо дані успішно пройшли перевірку, система зберігає їх у базі даних.
6. Система надсилає підтвердження реєстрації: після збереження даних система надсилає користувачу повідомлення про успішну реєстрацію. Це може бути повідомлення на екрані, email або інший спосіб підтвердження.

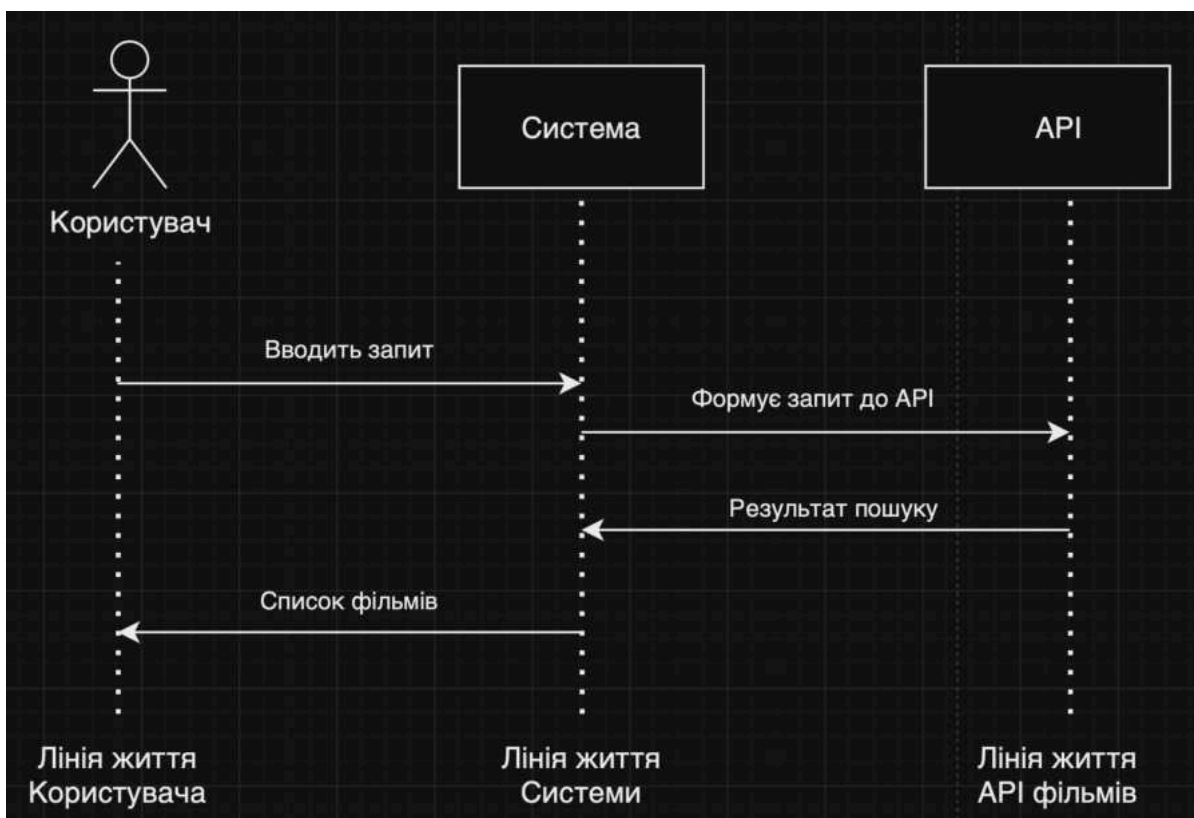


Рисунок 2.11 – Діаграма послідовностей "Пошук фільмів"

1. Користувач вводить запит: користувач починає взаємодію, вводячи ключові слова або назву фільму в поле пошуку на сайті.
2. Система формує запит до API: система отримує запит від користувача і перетворює його у формат, зрозумілий для API фільмів. Це може включати додавання параметрів пошуку, таких як жанр, рік випуску або інші критерії.
3. Система надсилає запит до API: сформований запит відправляється до API фільмів для обробки.
4. API фільмів повертає результат пошуку: API фільмів обробляє запит та здійснює пошук фільмів у своїй базі даних, повертаючи результати системі.
5. Система отримує результат пошуку: система отримує відповідь від API фільмів, яка містить список фільмів, що відповідають критеріям пошуку, або повідомлення про відсутність результатів.
6. Система відображає список фільмів: система обробляє отримані дані та відображає користувачу список знайдених фільмів, зазвичай у вигляді списку з назвами, постерами та іншою релевантною інформацією.

2.3 Структура проєкту

Файлова система проєкту організована згідно з типовою структурою Laravel-проєкту, що забезпечує модульність, чіткий поділ відповідальності та зручність в управлінні кодом. Логічна організація файлів та каталогів спрощує навігацію по проєкту, пошук необхідних компонентів та розуміння їх взаємозв'язків.

Основний код програми розміщено в каталозі `app/`. Він поділений на підкаталоги, кожен з яких відповідає за певний аспект роботи програми. Наприклад, каталог `app/Http/Controllers/` містить контролери, що обробляють HTTP-запити та формують відповіді, каталог `app/Models/` містить моделі даних, що представляють сутності в базі даних, а каталог `app/Services/` містить класи

сервісів, що реалізують бізнес-логіку програми. Каталог `bootstrap/` відповідає за ініціалізацію програми та автозавантаження класів. Файли в цьому каталозі налаштовують основні компоненти програми, такі як обробка винятків, ведення журналу та маршрутизація. Конфігураційні файли розміщено в каталозі `config/`. Вони дозволяють налаштовувати різні аспекти програми, такі як база даних, email, кешування, аутентифікація, безпека тощо. Наприклад, файл `config/database.php` містить налаштування підключення до бази даних, а файл `config/auth.php` налаштовує параметри аутентифікації користувачів.

Каталог `database/` містить файли, пов'язані з базою даних. Міграції, що описують зміни в структурі бази даних, розміщені в каталозі `database/migrations/`. Фабрики для генерації тестових даних знаходяться в каталозі `database/factories/`, а сідери для початкового заповнення бази даних - в каталозі `database/seeders/`. Публічно доступні файли, такі як CSS-стилі, JavaScript-скрипти та зображення, розміщено в каталозі `public/`. У папці `public` містяться файли JavaScript, що відповідають за інтерактивність та динамічну поведінку інтерфейсу. Зокрема, тут можуть бути розміщені скрипти, що використовують фреймворк Slick для створення слайдерів та каруселей зображень

Файли стилів CSS визначають візуальне оформлення інтерфейсу програми, а файли JavaScript додають інтерактивність та динамічну поведінку.

Каталог `resources/` містить ресурси проекту, що не є кодом програми. Шаблони Blade для відображення даних користувачу знаходяться в каталозі `resources/views/`, а файли для локалізації інтерфейсу на різні мови - в каталозі `resources/lang/`. Маршрути програми, що визначають зв'язок між URL-адресами та контролерами, розміщені в каталозі `routes/`. Файл `routes/web.php` містить маршрути для веб-інтерфейсу програми, а файл `routes/api.php` - для API програми.

Каталог `storage/` використовується для зберігання файлів, генерованих програмою під час роботи, таких як кеш, сесії, журнали, завантажені користувачами файли тощо. Каталог `tests/` містить тести для перевірки

коректності роботи програми. Тести дозволяють автоматизувати процес перевірки функціональності та виявляти потенційні помилки. Каталог vendor/ містить сторонні бібліотеки та залежності, встановлені через Composer. Composer - це інструмент для управління залежностями в PHP-проектах, що дозволяє легко встановлювати та оновлювати сторонні бібліотеки.

Додаткові файли, такі як .env (файл змінних оточення), artisan (файл для запуску команд Artisan), composer.json (файл конфігурації Composer) та readme.md (файл з описом проекту), знаходяться в кореновому каталозі проекту.

Ця файлова структура забезпечує логічну організацію коду та ресурсів, що сприяє зручності розробки, підтримки та масштабування проекту. Чіткий поділ відповідальності між каталогами та файлами спрощує навігацію по проекту, розуміння його структури та полегшує внесення змін.

Для візуалізації взаємозв'язків між основними класами мого проекту, я розробив діаграму класів UML (Unified Modeling Language). Ця діаграма демонструє структуру класів, їх атрибути, методи та залежності між ними.

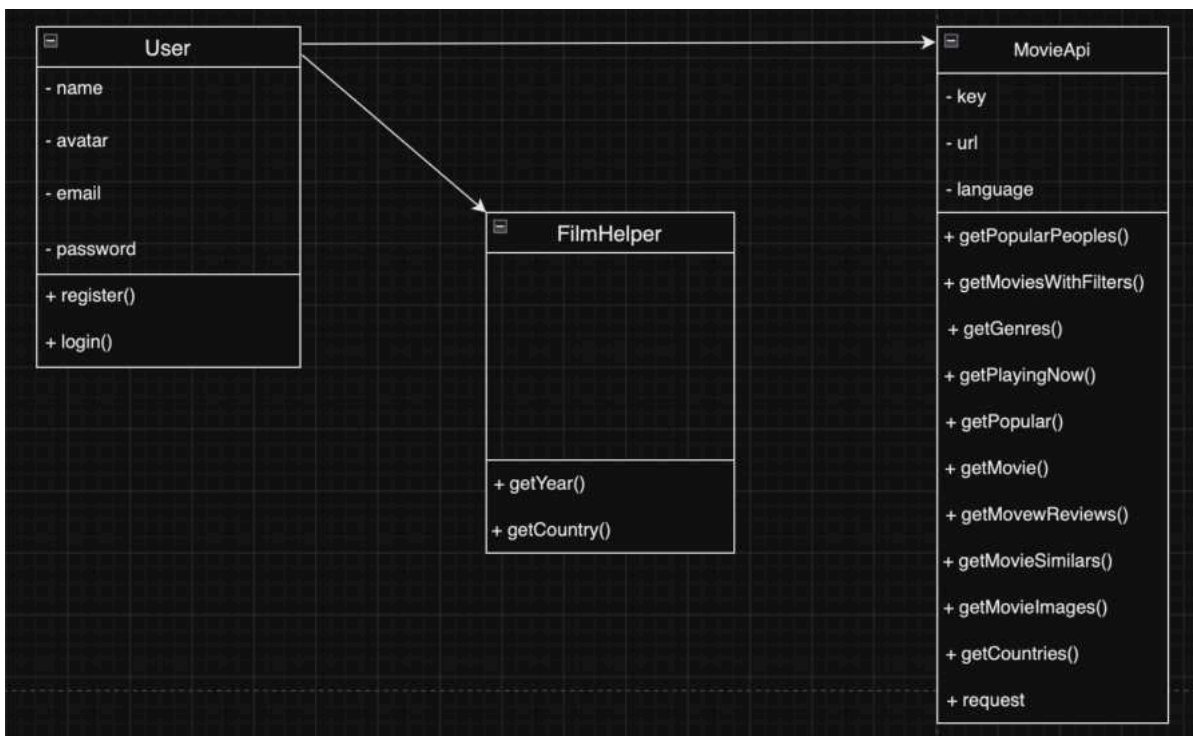


Рисунок 2.12 – Діаграма класів проекту

Основні класи проекту:

1. **User:** цей клас представляє користувача системи та містить такі атрибути, як ім'я (name), аватар (avatar), електронна адреса (email) та пароль (password). Клас User має методи для реєстрації (register()) та входу в систему (login()).
2. **FilmHelper:** цей клас надає допоміжні функції для роботи з даними про фільми. Він містить методи getYear() для вилучення року з дати виходу фільму та getCountry() для отримання коду країни виробництва фільму.
3. **MovieApi:** цей клас реалізує взаємодію з зовнішнім API для отримання інформації про фільми. Він має атрибути, такі як ключ API (key), URL API (url) та мова запитів (language). Клас MovieApi також містить різноманітні методи для отримання даних про фільми, наприклад, getPopularPeoples() для отримання список популярних акторів та знімальної групи, getMoviesWithFilters() для отримання списку фільмів з фільтрами, getGenres() для отримання списку жанрів та інші.

Зв'язки між класами:

1. **User → FilmHelper (Залежність):** клас User залежить від класу FilmHelper для виконання допоміжних функцій, пов'язаних з фільмами.
2. **User → MovieApi (Залежність):** клас User залежить від класу MovieApi для отримання інформації про фільми з зовнішнього API.

Висновки до розділу 2

У другому розділі дипломної роботи було детально описано засоби, які будуть використані для реалізації системи. Проведено ретельну оцінку різних технологій і інструментів, зваживши їхні переваги та недоліки. Це дозволило обрати оптимальні рішення, які забезпечують ефективну і стабільну роботу системи. Обґрунтування вибору показує виважений і зважений підхід до планування проекту.

Було розроблено і представлено структуру проекту, яка включає детальний опис архітектури системи з усіма ключовими компонентами та їхніми взаємозв'язками. Це допомагає чітко зрозуміти загальну концепцію та логіку функціонування системи. Завдяки детальному опису структури проекту, розробка відбуватиметься більш послідовно та організовано, що є важливим для успішного досягнення цілей проекту.

Особливу увагу приділено загальній структурі системи, включаючи базу даних, яка буде використовуватись у проекті. Було описано структуру бази даних, її основні таблиці та взаємозв'язки між ними. Це забезпечує ефективне управління даними і дозволяє системі працювати на високому рівні продуктивності та надійності. Таким чином, другий розділ заклав міцні технічні основи для подальшої реалізації проекту, забезпечуючи необхідну базу для його успішного впровадження і експлуатації.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1. Реалізація системи

Одним з ключових аспектів функціональності мого сайту з кінематографічною продукцією є можливість отримання актуальної інформації про фільми з зовнішніх джерел. Для цього я вирішив використовувати API (Application Programming Interface) популярного сервісу The Movie Database (TMDb), який надає доступ до великої бази даних фільмів, серіалів та іншого контенту.

Інтеграція з API TMDb. Для взаємодії з API TMDb я створив клас `MovieApi` на мові PHP з використанням фреймворку Laravel. Цей клас інкапсулює логіку виконання запитів до API, обробки відповідей та витягнення необхідних даних:

```
<?php
namespace App\Services;
use Illuminate\Support\Facades\Http;
use Exception;
class MovieApi
{
    private string $key;
    private string $uri;
    private string $language = 'uk-UA';

    public function __construct ()
    {
        $this->key = config('movie.key');
        $this->uri = config('movie.uri');
    }
}
```

Ключові методи класу **MovieApi**:

1. `getPopularPeoples()`: цей метод отримує список популярних акторів та актрис з API TMDb. Він приймає опціональний параметр `$page` для вказання номера сторінки результатів.

```
public function getPopularPeoples (int $page = 1): array
{
    return $this->request('/person/popular', 'GET', [
        'language' => $this->language,
        'page' => $page
    ]); }

```

2. `getMoviesWithFilters()`: цей метод дозволяє отримати список фільмів з фільтрацією та сортуванням за різними критеріями. Він приймає опціональні параметри `$page` для вказання номера сторінки результатів та `$filters` для вказання фільтрів (наприклад, жанр, рік випуску).

```
public function getMoviesWithFilters (int $page = 1, array $filters
= [])
{
    return $this->request('/discover/movie', 'GET', [
        'language' => $this->language,
        'page' => $page,
        ...$filters,
    ]); }

```

3. `getGenres()`: цей метод отримує список жанрів фільмів з API TMDb.

```
public function getGenres (): array
{
    return $this->request('/genre/movie/list', 'GET', [
        'language' => $this->language
    ]); }

```

4. `getPlayingNow()`: цей метод отримує список фільмів, які зараз йдуть в кінотеатрах.

```
public function getPlayingNow (): array
{
    return $this->request('/movie/now_playing', 'GET', [
        'language' => $this->language
    ]); }

```

5. `getPopular()`: цей метод отримує список популярних фільмів з API TMDb.

```
public function getPopular (): array
{
    return $this->request('/movie/popular', 'GET', [
        'language' => $this->language,
        'page' => 1
    ]); }

```

6. `getMovie()`: цей метод отримує детальну інформацію про конкретний фільм за його ID.

```
public function getMovie (int $movieId): ?array
{
    try {
        return $this->request('/movie/' . $movieId, 'GET', [
            'language' => $this->language,
            'page' => 1
        ]);
    } catch (\Throwable $throwable) {
        return null;
    } }

```

7. `getMovieReviews()`: цей метод отримує відгуки про конкретний фільм.

```
public function getMovieReviews (int $movieId): ?array
{
    try {
return $this->request('/movie/' . $movieId . '/reviews', 'GET', [
        'page' => 1
    ]);
    } catch (\Throwable $throwable) {
        return null;
    } }

```

8. `getMovieSimilar()`: цей метод отримує список схожих фільмів.

```
public function getMovieSimilar (int $movieId): ?array
{
    return $this->request('/movie/' . $movieId . '/similar', 'GET', [
        'page' => 1,
        'language' => $this->language
    ]); }

```

9. `getMovieImages()`: цей метод отримує зображення для конкретного фільму.

```
public function getMovieImages (int $movieId): array
{
    return $this->request('/movie/' . $movieId . '/images', 'GET', [
        'page' => 1,
    ]); }

```

10. `getCountries()`: цей метод отримує список країн.

```
public function getCountries (): array

```

```
{
    return $this->request('/configuration/countries', 'GET', [
        'language' => $this->language,
    ]); }
}
```

Перевірка на помилки:

```
$movieApi = new MovieApi();
try {
    $popularMovies = $movieApi->getPopular();

    // Обробка даних про популярні фільми
    foreach ($popularMovies['results'] as $movie) {
        // ... (обробка даних про кожен фільм)
    }
} catch (Exception $e) {
}
}
```

Переваги використання API TMDb:

1. **Актуальність даних:** API TMDb надає доступ до актуальної інформації про фільми, що дозволяє моєму сайту завжди бути в курсі останніх новинок.
2. **Широкий функціонал:** API TMDb пропонує різноманітні методи для отримання різних типів даних, що дозволяє мені реалізувати різноманітні функції на сайті, такі як пошук, фільтрація, сортування, відображення детальної інформації про фільми, відгуків, схожих фільмів тощо.
3. **Зручність використання:** Laravel Http Client спрощує взаємодію з API, роблячи код більш читабельним та легким в обслуговуванні.

Після отримання даних з API TMDb, я використовую їх для заповнення бази даних мого сайту та відображення інформації про фільми на сторінках. Наприклад, я можу відображати список популярних фільмів на головній сторінці, дозволити користувачам шукати фільми за назвою або жанром,

переглядати детальну інформацію про фільми, читати відгуки та переглядати схожі фільми.

Використання API TMDb дозволяє мені створити функціональний та інформативний сайт з кінематографічною продукцією, який буде цікавий для користувачів та надасть їм зручний доступ до інформації про фільми.

3.2. Тестування роботи системи

Для забезпечення коректної та стабільної роботи сайту з кінематографічною продукцією, я провів ретельне тестування всіх його функцій.

Головна сторінка. На наданому скріншоті зображена головна сторінка мого сайту з кінематографічною продукцією. Вона має зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко знаходити цікаві фільми.

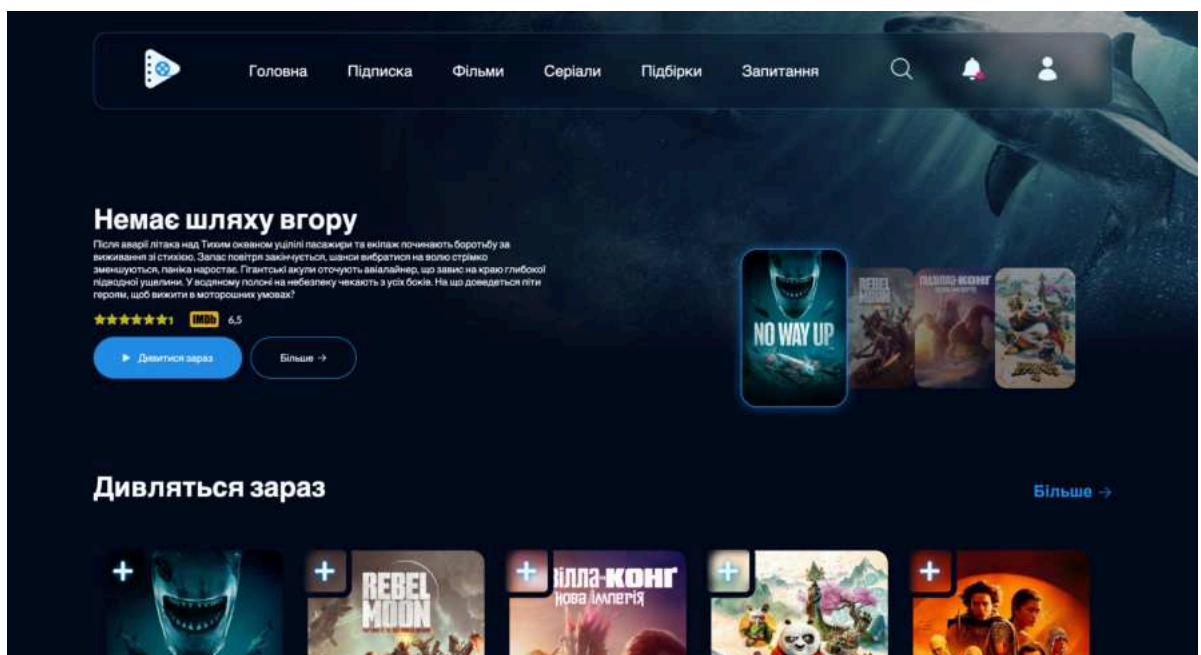


Рисунок 3.1 – Головна сторінка проєкту

Елементи інтерфейсу:

1. Верхнє меню: у верхній частині сторінки розташоване меню з наступними розділами:
 - a. Логотип: логотип сайту, який також служить посиланням на головну сторінку.
 - b. Головна: посилання на головну сторінку.
 - c. Підписка: розділ з інформацією про підписку на сайт та її переваги.
 - d. Фільми: розділ з каталогом фільмів.
 - e. Серіали: розділ з каталогом серіалів.
 - f. Підбірки: розділ з тематичними підбірками фільмів та серіалів.
 - g. Запитання: розділ з відповідями на поширені запитання.
 - h. Сповіщення: іконка для перегляду сповіщень.
 - i. Профіль користувача: іконка для доступу до профілю користувача та налаштувань.

2. Фільм дня: у центральній частині сторінки відображається фільм дня з постером, назвою, коротким описом, рейтингом та кнопками "Дивитися зараз" та "Більше".

3. Рекомендації: нижче фільму дня розташований розділ "Дивитися зараз" з каруселлю постерів фільмів та серіалів, які рекомендуються до перегляду. Користувач може прокручувати карусель та натискати на постери, щоб перейти на сторінки з детальною інформацією про фільми.

Переваги сайту:

1. Зручна навігація: інтуїтивно зрозуміле меню та структура сайту дозволяють користувачам легко знаходити потрібну інформацію.
2. Різноманітність контенту: сайт пропонує широкий вибір фільмів та серіалів різних жанрів та років випуску.
3. Персоналізовані рекомендації: система рекомендацій допомагає користувачам знаходити фільми та серіали, які відповідають їхнім інтересам.
4. Актуальна інформація: сайт регулярно оновлюється новою інформацією про фільми та серіали.

5. Зручний перегляд: користувачі можуть дивитися фільми та серіали онлайн в високій якості.

6. Підписка: сайт пропонує платну підписку, яка надає доступ до додаткових функцій та контенту.

Загалом, головна сторінка мого сайту надає користувачам зручний та інтуїтивно зрозумілий інтерфейс для пошуку та перегляду фільмів та серіалів, а також пропонує різноманітні переваги, які роблять його привабливим для любителів кіно.

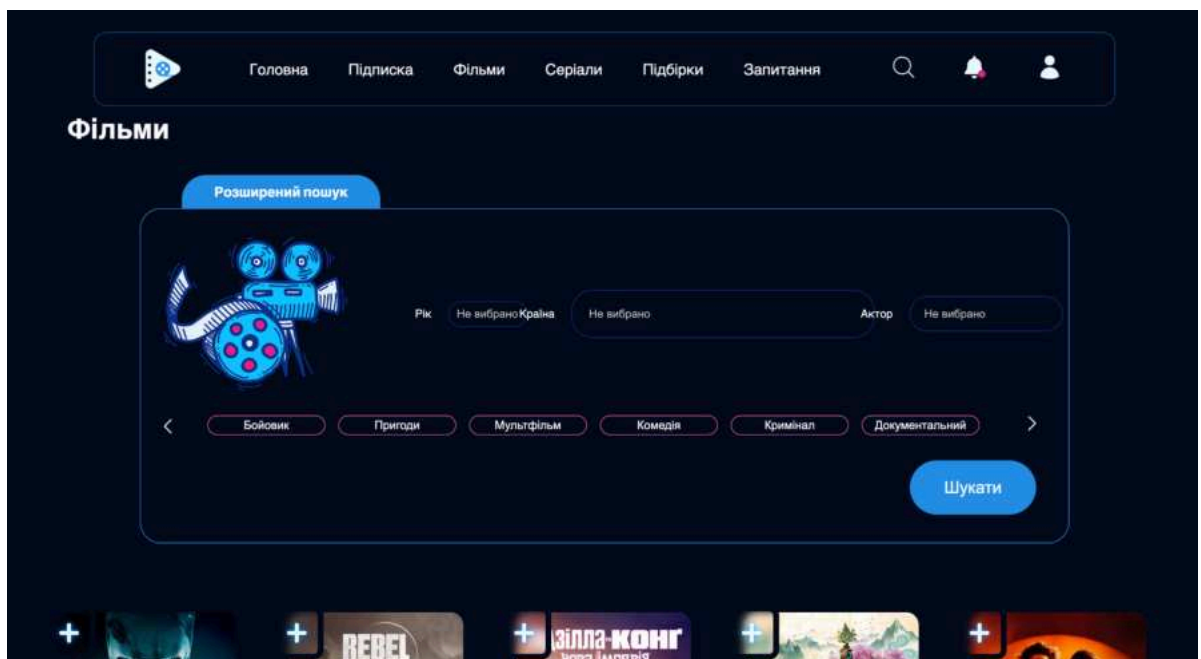


Рисунок 3.2 – Сторінка пошуку фільмів

Сторінка пошуку фільмів. На скріншоті рис. 3.2 зображена сторінка розширеного пошуку фільмів на моєму сайті. Вона надає користувачам зручний інструмент для знаходження фільмів за різними критеріями.

Елементи інтерфейсу:

1. Верхнє меню: аналогічне меню, як і на головній сторінці, що дозволяє легко переміщатися між розділами сайту.

2. Заголовок "Фільми": вказує на те, що користувач знаходиться в розділі пошуку фільмів.

3. Розширений пошук: заголовок блоку з фільтрами пошуку.
4. Фільтри:
 - a. Рік: випадаючий список для вибору року випуску фільму.
 - b. Країна: випадаючий список для вибору країни виробництва фільму.
 - c. Жанри: горизонтальний список кнопок з різними жанрами фільмів (бойовик, пригоди, мультфільм, комедія, кримінал, документальний).Користувач може вибрати один або декілька жанрів для фільтрації результатів пошуку.
- d. Актор: поле для введення імені актора.
5. Кнопка "Шукати": кнопка для запуску пошуку з вибраними фільтрами.

Функціональність:

1. Фільтрація: користувач може фільтрувати фільми за роком випуску, країною виробництва, жанром та актором.
2. Пошук: після вибору фільтрів користувач може натиснути кнопку "Шукати", щоб знайти фільми, які відповідають заданим критеріям.
3. Зручність: інтерфейс пошуку простий та інтуїтивно зрозумілий, що дозволяє користувачам легко знаходити потрібні фільми.

Переваги:

1. Точність пошуку: розширений пошук дозволяє користувачам знаходити фільми, які точно відповідають їхнім інтересам.
2. Зручність: інтерфейс пошуку простий та інтуїтивно зрозумілий.
3. Гнучкість: користувачі можуть комбінувати різні фільтри для отримання більш точних результатів пошуку.

Загалом, сторінка розширеного пошуку фільмів надає користувачам потужний інструмент для знаходження фільмів.

Сторінка фільму. На наданому скріншоті зображена сторінка з детальною інформацією про фільм "Мадам Павутина" на моєму сайті.

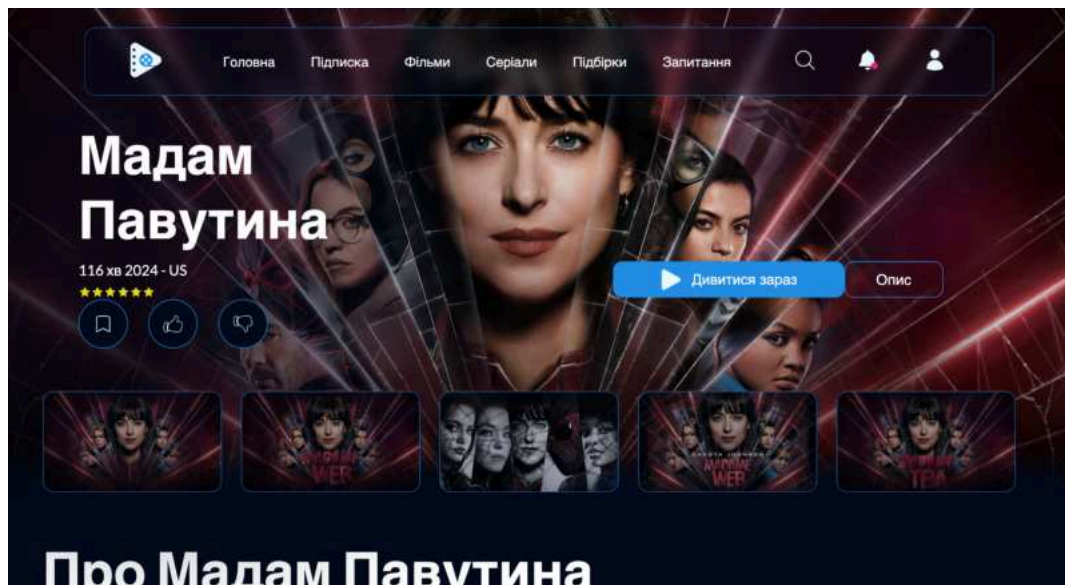


Рисунок 3.3 – Сторінка фільму

Елементи інтерфейсу:

1. Верхнє меню: аналогічне меню, як і на головній сторінці, що дозволяє легко переміщатися між розділами сайту.
2. Назва фільму: великим шрифтом відображається назва фільму "Мадам Павутина".
3. Інформація про фільм: під назвою фільму відображається його тривалість, рік випуску та країна виробництва.
4. Рейтинг: зірки відображають рейтинг фільму.
5. Кнопки дій:
 - a. "Дивитися зараз": кнопка для початку перегляду фільму.
 - b. "Опис": кнопка для переходу до опису фільму.
6. Постер фільму: великий постер фільму займає центральну частину сторінки.
7. Галерея зображень: нижче постера розташована галерея зображень з фільму, яку користувач може прокручувати.

Додаткова інформація (відображена на скріншоті):

1. Опис фільму: детальний опис сюжету фільму.
2. Жанри: список жанрів, до яких належить фільм.
3. Акторський склад: список акторів та їх ролей у фільмі.

4. Режисер: інформація про режисера фільму.
5. Трейлер: відео-трейлер фільму.
6. Схожі фільми: список фільмів, які схожі на "Мадам Павутина" за жанром, сюжетом або акторським складом.

Функціональність:

1. Перегляд інформації: користувач може переглядати детальну інформацію про фільм, включаючи його опис, жанри, акторський склад, трейлер та схожі фільми.
2. Перегляд фільму: користувач може почати перегляд фільму, натиснувши кнопку "Дивитися зараз".
3. Оцінка фільму: користувач може оцінити фільм, використовуючи систему рейтингу.
4. Додавання до списку обраних: користувач може додати фільм до свого списку обраних для подальшого перегляду.

Сторінка фільму, має всю необхідну інформацію про фільм, та дозволяє їм легко почати його переглядати.

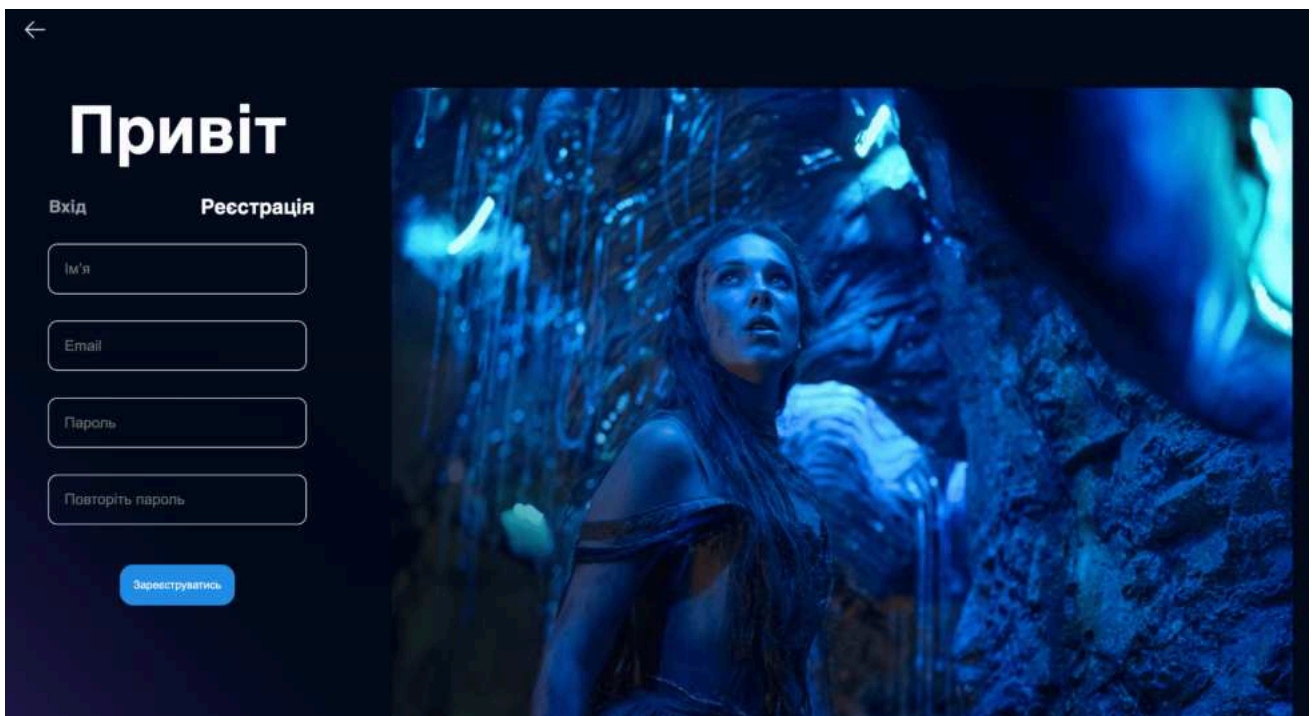


Рисунок 3.4 – Регістрація користувача

Авторизація користувача. На рисунку 3.4 зображена сторінка авторизації користувача на моєму сайті. Вона дозволяє новим користувачам зареєструватися, а існуючим - увійти до системи, використовуючи облікові дані.

Елементи інтерфейсу:

1. Заголовок "Привіт": вітає користувача на сторінці авторизації.
2. Вкладки "Вхід" та "Реєстрація": дозволяють користувачеві перемикатися між формами входу та реєстрації.
3. Форма реєстрації:
 - a. Поля вводу:
 - i. Ім'я: поле для введення імені користувача.
 - ii. Email: поле для введення адреси електронної пошти.
 - iii. Пароль: поле для введення пароля.
 - iv. Повторіть пароль: поле для повторного введення пароля для підтвердження.
 - b. Кнопка "Зареєструватися": кнопка для відправки форми реєстрації.
4. Фонове зображення: зображення з фільму створює атмосферу та візуально приваблює користувача.

Функціональність:

1. Реєстрація: нові користувачі можуть зареєструватися на сайті, вказавши своє ім'я, email та пароль.
2. Вхід: існуючі користувачі можуть увійти до системи, вказавши свій email та пароль.

Переваги:

1. Зручність: простий та інтуїтивно зрозумілий інтерфейс дозволяє користувачам легко зареєструватися або увійти до системи.
2. Безпека: паролі користувачів зберігаються в зашифрованому вигляді для забезпечення безпеки.
3. Персоналізація: авторизація дозволяє користувачам отримати доступ до персоналізованих функцій сайту, таких як список обраних фільмів та історія переглядів.

Загалом, сторінка авторизації надає користувачам зручний та безпечний спосіб доступу до всіх функцій мого сайту з кінематографічною продукцією

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи я реалізував систему, використовуючи засоби та структури, описані в попередніх розділах. Процес розробки включав написання коду, інтеграцію компонентів та створення інтерфейсу користувача. Кожен етап реалізації був ретельно спланований і виконаний відповідно до визначених технічних вимог і архітектурних рішень. Це дозволило створити функціональну і ефективну систему, яка відповідає поставленим завданням.

Після завершення розробки проведено всебічне тестування системи, щоб переконатися в її надійності та коректності роботи. Тестування включало перевірку всіх основних функцій, а також навантажувальні і стрес-тести для оцінки продуктивності системи під різними умовами. Виявлені помилки були оперативно виправлені, що забезпечило високу стабільність і ефективність роботи системи.

Таким чином, третій розділ підтвердив життєздатність розробленої системи і її готовність до впровадження. Реалізація і тестування довели, що система відповідає всім технічним вимогам і готова до експлуатації. Завдяки цьому, досягнуто поставлених цілей і забезпечити успішне завершення проекту.

ВИСНОВОК

Дана кваліфікаційна робота була присвячена актуальній темі розробки веб-каталогу кінематографічної продукції. В умовах постійного зростання кількості фільмів та серіалів, а також розширення платформ для стрімінгу, виникає гостра потреба в зручних та ефективних інструментах для організації, пошуку та доступу до відеоконтенту. Розробка такого каталогу є складним завданням, що вимагає комплексного підходу з урахуванням різноманітних аспектів, таких як архітектура бази даних, дизайн користувацького інтерфейсу, реалізація алгоритмів пошуку та фільтрації, а також забезпечення безпеки даних користувачів.

У рамках дослідження було проведено детальний аналіз предметної області, що включав огляд існуючих аналогів веб-каталогів кінематографічної продукції. Виявлення сильних та слабких сторін цих рішень дозволило сформулювати конкретну задачу розробки, спрямовану на створення каталогу, який би відповідав сучасним вимогам користувачів та враховував особливості розвитку індустрії кіно та телебачення.

На основі проведеного аналізу було обрано оптимальні інструменти для реалізації поставленої задачі. Вибір було зроблено на користь сучасних веб-технологій, таких як HTML5, CSS3 та PHP, а також популярного фреймворку Laravel, що забезпечує ефективну розробку веб-застосунків. Цей вибір обґрунтований широкими можливостями обраних інструментів, їх гнучкістю та відповідністю сучасним стандартам веб-розробки.

Наступним етапом стало проектування структури веб-каталогу. Було розроблено детальну схему бази даних, здатну зберігати та обробляти великі обсяги інформації про кінематографічну продукцію. Структура бази даних враховувала різноманітні атрибути фільмів та серіалів, такі як назва, жанр, режисер, акторський склад, рік випуску, рейтинг та інші, що забезпечує можливість гнучкого пошуку та фільтрації контенту. Також було визначено

основні модулі та компоненти веб-застосунку, що формують його архітектуру та забезпечують взаємодію з базою даних та користувачами.

Реалізація веб-каталогу здійснювалася поетапно, згідно з розробленою структурою. Кожен модулів та компонент застосунку ретельно розроблявся, з особливою увагою до якості коду та відповідності вимогам. Було реалізовано функціонал пошуку, фільтрації та відображення інформації про фільми, забезпечуючи зручний та інтуїтивно зрозумілий інтерфейс для користувачів.

Після завершення розробки було проведено комплексне тестування системи, метою якого було виявлення та усунення можливих помилок та недоліків. Тестування охоплювало різні аспекти роботи веб-каталогу, включаючи перевірку коректності роботи функцій пошуку та фільтрації, відображення інформації, а також перевірку навантаження та безпеки системи.

Створений веб-каталог кінематографічної продукції має значне практичне значення. Він надає користувачам зручний інструмент для пошуку, систематизації та перегляду інформації про фільми та серіали, що є особливо актуальним в умовах постійного зростання обсягу кінематографічного контенту. Розроблений каталог може бути використаний як самостійний ресурс, так і інтегрований в інші веб-платформи, що працюють з кіно та телебаченням.

Результати кваліфікаційної роботи мають перспективи для подальшого розвитку та вдосконалення. Можливі напрямки розвитку включають розширення функціоналу каталогу, додавання нових можливостей для користувачів, інтеграцію з іншими сервісами, а також адаптацію каталогу під мобільні пристрої. Крім того, розроблені в рамках кваліфікаційної роботи рішення та підходи можуть слугувати основою для створення аналогічних систем, що оперують великими обсягами даних в інших предметних областях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація TMDb (API). веб-сайт. URL: <https://developer.themoviedb.org/reference/intro/getting-started> (дата звернення: 25.02.2024).
2. Адаптивний веб-дизайн. веб-сайт. URL: <https://www.yakaboo.ua/ua/adaptivniy-veb-dizayn.html> (дата звернення: 05.02.2024).
3. Create database connection. веб-сайт. URL: <https://www.geeksforgeeks.org/how-to-create-a-database-connection/> (дата звернення: 7.03.2024).
4. Create REST API Using Laravel. веб-сайт. URL: <https://blog.treblle.com/how-to-create-rest-api-using-laravel/> (дата звернення: 10.03.2024).
5. Figma. веб-сайт. URL: <https://www.figma.com/> (дата звернення: 30.01.2024).
6. HTML5. Документація W3C. веб-сайт. URL: <https://html.spec.whatwg.org/multipage/> (дата звернення: 10.03.2024).
7. IMDb. веб-сайт. URL: <https://www.imdb.com/> (дата звернення: 03.01.2024).
8. Інтерфейс користувача. Wikipedia.com. веб-сайт. URL: https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D1%81_%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%87%D0%B0 (дата звернення: 31.01.2024).
9. W3Schools. Поглиблені уроки з HTML5, CSS3, JavaScript. веб-сайт. URL: <https://www.w3schools.com/js/> (дата звернення: 01.03.2024).
10. Laravel. веб-сайт. URL: <https://laravel.com/> (дата звернення: 25.02.2024).

11. Мегого. веб-сайт. URL: <https://megogo.net/> (дата звернення: 03.01.2024).
12. Офіційна документація CSS3 на MDN Web Docs. веб-сайт. URL: <https://developer.mozilla.org/uk/docs/Web/CSS/CSS3> (дата звернення: 10.03.2024).
13. Офіційний сайт PHP. веб-сайт. URL: <https://dlf.ua/ua/aktualni-vimogi-do-veb-sajtiv-v-ukrayini-pravovi-aspekti/> (дата звернення: 01.03.2024).
14. Ринок онлайн кінотеатрів. веб-сайт. URL: <https://www.unian.ua/economics/telecom/prosto-pro-ott-funkcional-ta-perevagi-onlayn-kinoteatriv-novini-11473258.html> (дата звернення: 02.01.2024).
15. Рівність тестування системи. веб-сайт. URL: <https://www.techtarget.com/searchsoftwarequality/definition/system-testing> (дата звернення: 19.03.2024).
16. Система тестування. веб-сайт. URL: <https://www.techtarget.com/searchsoftwarequality/definition/system-testing> (дата звернення: 24.03.2024).
17. UML Class Diagram Tutorial. веб-сайт. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/> (дата звернення: 10.04.2024).
18. What is system testing? веб-сайт. URL: <https://www.techtarget.com/searchsoftwarequality/definition/system-testing> (дата звернення: 10.04.2024).
19. Що таке база даних? Wikipedia.com. веб-сайт. URL: [\[https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85\]](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)(<https://uk.wikipedia.org/wiki/%D0%91> (дата звернення: 05.03.2024).
20. Документація JavaScript. веб-сайт. URL: <https://javascript.info/> (дата звернення: 31.02.2024).

ДОДАТКИ

Додаток А Реалізація коду API

```
<?php
namespace App\Services;
use Illuminate\Support\Facades\Http;
use Exception;
class MovieApi
{
    /**
     * @var string $key
     */
    private string $key;

    /**
     * @var string $uri
     */
    private string $uri;

    /**
     * @var string $language
     */
    private string $language = 'uk-UA';

    public function __construct ()
    {
        $this->key = config('movie.key');
        $this->uri = config('movie.uri');
    }

    /**
     * @return array
     * @throws Exception
     */
}
```

```

    */
public function getPopularPeoples (int $page = 1): array
{
    return $this->request('/person/popular', 'GET', [
        'language' => $this->language,
        'page' => $page
    ]);
}

public function getMoviesWithFilters (int $page = 1, array
$filters = [])
{
    return $this->request('/discover/movie', 'GET', [
        'language' => $this->language,
        'page' => $page,
        ...$filters,
    ]);
}

/**
 * @return array
 * @throws Exception
 */
public function getGenres (): array
{
    return $this->request('/genre/movie/list', 'GET', [
        'language' => $this->language
    ]);
}

/**
 * @return array
 * @throws Exception
 */
public function getPlayingNow (): array
{
    return $this->request('/movie/now_playing', 'GET', [

```

```

        'language' => $this->language
    ]);
}

public function getPopular (): array
{
    return $this->request('/movie/popular', 'GET', [
        'language' => $this->language,
        'page' => 1
    ]);
}

/**
 * @param int $movieId
 * @return array|null
 */
public function getMovie (int $movieId): ?array
{
    try {
        return $this->request('/movie/' . $movieId, 'GET', [
            'language' => $this->language,
            'page' => 1
        ]);
    } catch (\Throwable $throwable) {
        return null;
    }
}

/**
 * @param int $movieId
 * @return array|null
 */
public function getMovieReviews (int $movieId): ?array
{
    try {
        return $this->request('/movie/' . $movieId . '/reviews',
'GET', [

```

```

        'page' => 1
    });
    } catch (\Throwable $throwable) {
        return null;
    }
}

/**
 * @param int $movieId
 * @return array|null
 */
public function getMovieSimilar (int $movieId): ?array
{
    return $this->request('/movie/' . $movieId . '/similar',
'GET', [
        'page' => 1,
        'language' => $this->language
    ]);
}

/**
 * @param int $movieId
 * @return array
 * @throws Exception
 */
public function getMovieImages (int $movieId): array
{
    return $this->request('/movie/' . $movieId . '/images',
'GET', [
        'page' => 1,
    ]);
}

/**
 * @return array
 * @throws Exception
 */

```



```
public function getCountries (): array
{
    return $this->request('/configuration/countries', 'GET', [
        'language' => $this->language,
    ]);
}

/**
 * @param string $uri
 * @param $method
 * @param array $data
 * @return array
 * @throws Exception
 */
private function request(string $uri = '/', $method = 'GET',
array $data = []): array
{
    $request =
Http::withToken($this->key)->{mb_strtolower($method)}($this->uri . $uri,
$data);

    if (!$request->successful()) {
        throw new Exception('Movie api is not available.');
```

Додаток В

Пошук фільмів

```

@extends('parts.layout')
@section('content')

    <section class="all__filml">
        <h2 class="bd-48">Фільми</h2>
        <div class="container">
            <div class="tittle__advanced__wrapper">
                <div class="title__advance">
                    <h4 class="bd-20">Розширений пошук</h4>
                </div>
            </div>
            <div class="advance__search">
                <form action="/movies" method="GET"
class="advance__content">
                    <div class="first__block">
                        <div class="left__block">
                            
                        </div>
                        <div class="right__block">
                            <div>
                                <div class="block__up">
                                    <div class="block__up-item">
                                        <span
class="mm-14">Пік</span>
                                        <select name="year">
                                            <option value=""
@if(!isset($_GET['year'])) selected @endif>Не вибрано</option>
                                            <option value="2024"
@if(isset($_GET['year']) && $_GET['year'] === '2024') selected
@endif>2024</option>
                                            <option value="2023"
@if(isset($_GET['year']) && $_GET['year'] === '2023') selected
@endif>2023</option>

```

```

                                <option value="2022"
@if(isset($_GET['year']) && $_GET['year'] === '2022') selected
@endif>2022</option>
                                <option value="2021"
@if(isset($_GET['year']) && $_GET['year'] === '2021') selected
@endif>2021</option>
                                <option value="2020"
@if(isset($_GET['year']) && $_GET['year'] === '2020') selected
@endif>2020</option>
                                <option value="2019"
@if(isset($_GET['year']) && $_GET['year'] === '2019') selected
@endif>2019</option>
                                <option value="2018"
@if(isset($_GET['year']) && $_GET['year'] === '2018') selected
@endif>2018</option>
                                <option value="2017"
@if(isset($_GET['year']) && $_GET['year'] === '2017') selected
@endif>2017</option>
                                <option value="2016"
@if(isset($_GET['year']) && $_GET['year'] === '2016') selected
@endif>2016</option>
                                <option value="2015"
@if(isset($_GET['year']) && $_GET['year'] === '2015') selected
@endif>2015</option>
                                <option value="2014"
@if(isset($_GET['year']) && $_GET['year'] === '2014') selected
@endif>2014</option>
                                <option value="2013"
@if(isset($_GET['year']) && $_GET['year'] === '2013') selected
@endif>2013</option>
                                <option value="2012"
@if(isset($_GET['year']) && $_GET['year'] === '2012') selected
@endif>2012</option>
                                <option value="2011"
@if(isset($_GET['year']) && $_GET['year'] === '2011') selected
@endif>2011</option>
```

```

                <option value="2010"
@if(isset($_GET['year']) && $_GET['year'] === '2010') selected
@endif>2010</option>

            </select>
        </div>
        <div class="block__up-item">
            <span
class="mm-14">Країна</span>

            <select name="country"
style="padding: 20px;">
                <option value=""
@if(!isset($_GET['country'])) selected @endif>Не вибрано</option>
                @foreach ($countries as
$country)
                    <option value="{
$country['iso_3166_1'] }" @if(isset($_GET['country']) &&
$_GET['country'] == $country['iso_3166_1']) selected @endif>{{
$country['native_name'] }}</option>
                @endforeach
            </select>
        </div>
        <div class="block__up-item">
            <span
class="mm-14">Актор</span>

            <select name="actor">
                <option value=""
@if(!isset($_GET['actor'])) selected @endif>Не вибрано</option>
                @foreach ($actors as
$actor)
                    <option value="{
$actor['id'] }" @if(isset($_GET['actor']) && $_GET['actor'] ==
$actor['id']) selected @endif>{{ $actor['name'] }}</option>
                @endforeach
            </select>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
</div>
<div class="last__block">
    <div class="nav__block">
        <div class="nav__movie">
            <div class="prev__arrow3"
id="arrow-prev3">
                <svg width="10" height="19"
viewBox="0 0 10 19" fill="none" xmlns="http://www.w3.org/2000/svg">
                    <path fill-rule="evenodd"
clip-rule="evenodd" d="M0.228822 8.94757C-0.0762749 9.25267 -0.0762749
9.74733 0.228822 10.0524L8.04132 17.8649C8.34642 18.17 8.84108 18.17
9.14618 17.8649C9.45127 17.5598 9.45127 17.0652 9.14618 16.7601L1.8861
9.5L9.14618 2.23993C9.45127 1.93483 9.45127 1.44017 9.14618
1.13507C8.84108 0.829976 8.34642 0.829976 8.04132 1.13507L0.228822
8.94757Z" fill="#EBFAFF"/>
                </svg>
            </div>
            <ul class="nav3__slider__all">
                <?php $genresFilter =
$_GET['genres'] ?? []; ?>
                @foreach ($genres as $genre)
                <li>
                    <input
value="{{ $genre['id'] }}" @if(in_array($genre['id'], $genresFilter))
checked @endif name="genres[]" type="checkbox" style="display: none;"
id="cat{{ $genre['id'] }}">
                        <button type="button"
@if(in_array($genre['id'], $genresFilter)) class="selected" @endif>
                            <label for="cat{{
$genre['id'] }}">
                                {{
$genre['name'] }}
                            </label>
                        </button>
                    </li>

```

```

        @endforeach
    </ul>
    <div class="next__arrow3"
id="arrow-next3">
        <svg width="10" height="19"
viewBox="0 0 10 19" fill="none" xmlns="http://www.w3.org/2000/svg">
            <path fill-rule="evenodd"
clip-rule="evenodd" d="M0.228822 8.94757C-0.0762749 9.25267 -0.0762749
9.74733 0.228822 10.0524L8.04132 17.8649C8.34642 18.17 8.84108 18.17
9.14618 17.8649C9.45127 17.5598 9.45127 17.0652 9.14618 16.7601L1.8861
9.5L9.14618 2.23993C9.45127 1.93483 9.45127 1.44017 9.14618
1.13507C8.84108 0.829976 8.34642 0.829976 8.04132 1.13507L0.228822
8.94757Z" fill="#EBFAFF"/>
            </svg>
        </div>
    </div>
</div>
<div>
    <button type="submit"
class="search-btn">Искати</button>
</form>
</div>
<div class="block__all__movie-search">
    @foreach ($movies as $movie)
    <div class="item__movie">
        <div class="add__button-movie">
        </div>
        <a href="/movie/{{ $movie['id'] }}">
            
            </a>
        </div>
    @endforeach
</div>

```

```

<div class="pagination">
    @php
        $url = request()->fullUrl(); // Получаем текущий
URL
        $separator = (parse_url($url, PHP_URL_QUERY) ==
null) ? '?' : '&'; // Определяем разделитель
    @endphp

    @if($page > 1)
        @if(strpos($url, 'page=') !== false)
            <button onclick="window.location.href = '{{
$url }}&page={{ $page - 1 }}'"> Попередня сторінка</button>
        @else
            <button onclick="window.location.href = '{{
$url }}{{ $separator }}page={{ $page - 1 }}'"> Попередня
сторінка</button>
        @endif
    @endif
    @if($total_pages > $page)
        <button onclick="window.location.href = '{{ $url
}}{{ $separator }}page={{ $page + 1 }}'">Наступна сторінка ></button>
    @endif
</div>

</div>
</section>

@endsection

<style>
    .pagination {
        display: flex;
        grid-gap: 20px;
        margin-top: 60px;
        justify-content: center;
    }
    .pagination button {

```

```
padding: 15px 40px;
color: #fff;
background: #228EE5;
transition: 0.2s;
border-radius: 10px;
font-size: 18px;
cursor: pointer;
}
.pagination button:hover {
  opacity: 0.8;
}
</style>
```




метадані

Заголовок

Розробка веб-каталогу кінематографічної продукції

Автор

Кізера В. Науковий керівник / Експерт

підрозділ

King Danylo University

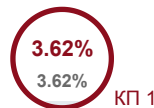
Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		3
Інтервали		0
Мікропробіли		0
Білі знаки		2
Парафрази (SmartMarks)		9

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

**25**

Довжина фрази для коефіцієнта подібності 2

**9845**

Кількість слів

78377

Кількість символів

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	Колір тексту
1	http://repository.ukd.edu.ua:8080/bitstream/handle/123456789/185/%D0%A2%D0%BE%D0%BC%D0%B5%D0%BD%D1%8E%D0%BA%20%D0%9D.%20%D0%92..pdf?sequence=1	89	0.90 %
2	http://ekhsuir.kspu.edu/bitstream/handle/123456789/14897/Bachynska_fuifzh_2021.pdf?sequence=1&isAllowed=y	24	0.24 %
3	http://repository.ukd.edu.ua:8080/bitstream/handle/123456789/185/%D0%A2%D0%BE%D0%BC%D0%B5%D0%BD%D1%8E%D0%BA%20%D0%9D.%20%D0%92..pdf?sequence=1	17	0.17 %
4	КРБ-Резвіна-НАІ196-14-06-2023.docx 6/18/2023 Odessa National Polytechnic University (Наукові журнали "BCIT" та "ПАІТ")	15	0.15 %