

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:

«Вебсайт для перегляду фільмів, серіалів тощо»

Виконав студент групи ІПЗ-20-4
ЛАКИЗЮК Ренат Валентинович

Керівник роботи:
ЛЕВЧЕНКО Антон Юрійович

Консультант:
ЄФРЕМОВ Юрій Миколайович

Рецензент:
ЛОБАНЧИКОВА Надія Миколаївна

Житомир – 2024

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

«ЗАТВЕРДЖУЮ»

Завідувач кафедри інженерії
програмного забезпечення

Тетяна ВАКАЛЮК

«26» лютого 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЛАКИЗИЮК Ренат Валентичнович**

Керівник роботи: **ЛЕВЧЕНКО Антон Юрійович**

Тема роботи: **«Вебсайт для перегляду фільмів, серіалів тощо»,**

затверджена Наказом закладу вищої освіти від **«23» лютого 2024 р., №74с**

Вихідні дані для роботи: створення та реалізація вебсайту для перегляду фільмів і серіалів, який використовує сучасні вебтехнології та API для забезпечення безперебійного доступу до великої кількості контенту, включаючи потокове відтворення відео.

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Левченко А.Ю., Єфремов Ю. М.	28.02.2024р	28.02.2024р
2	Левченко А.Ю., Єфремов Ю. М.	28.02.2024р	28.02.2024р
3	Левченко А.Ю., Єфремов Ю. М.	28.02.2024р	28.02.2024р

РЕФЕРАТ

Випускна робота бакалавра складається з реалізації вебсайта для перегляду фільмів, а також пояснювальної записки. Пояснювальна записка до дипломної роботи містить 55 сторінок, 21 ілюстрація, 16 таблиць, 3 додатки та 16 літературних джерел.

Основною метою цієї роботи є створення алгоритмів та реалізація програмного продукту, що використовує сучасні технології для забезпечення можливостей перегляду фільмів і серіалів, коментування, додавання нових фільмів та серіалів, а також створення та додавання студій озвучки.

У процесі роботи було визначено ключові завдання, пов'язані з розробкою вебсайту. Було проведено аналіз існуючих вебсайтів, які надають подібні послуги, для розуміння їх сильних та слабких сторін. Це допомогло визначити, які базові функції та можливості повинні бути реалізовані у нашому продукті.

Окремо було розроблено функціонал для створення та додавання студій озвучки. Це надасть можливість користувачам, які займаються дублюванням фільмів та серіалів, ділитися своєю роботою та створювати спільноти навколо своїх студій. Користувачі зможуть оцінювати озвучку, залишати відгуки та рекомендувати її іншим.

КЛЮЧОВІ СЛОВА: ВЕБСАЙТ, СТУДІЇ ОЗВУЧКИ, ФІЛЬМИ.

					ІПЗ.КР.Б - 121 - 24 - ПЗ					
					Вебсайт для перегляду фільмів, серіалів тощо. Пояснювальна записка					
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.		Лакизюк Р.В.								
Керівник		Левченко А.Ю.								
Рецензент		Лобанчикова Н.М.			Літ.		Арк.	Аркушів		
Зав. каф.		Вакалюк Т.А.					3	55		
					Житомирська політехніка, група ІПЗ-20-4					

ABSTRACT

The bachelor's thesis consists of the implementation of a movie viewing website as well as an explanatory note. The explanatory note to the thesis contains a 55 pages, 21 illustrations, 16 tables, 3 appendices and 16 references.

The main purpose of this work is to create algorithms and implement a software product that uses modern technologies to provide opportunities for watching movies and TV series, commenting, adding new movies and TV series, as well as creating and adding voice-over studios.

In the process, we identified key tasks related to the development of the website. We analyzed existing websites that provide similar services to understand their strengths and weaknesses. This helped us determine what basic functions and features should be implemented in our product.

Separately, we developed functionality for creating and adding voice over studios. This will allow users who dub movies and TV shows to share their work and create communities around their studios. Users will be able to rate voiceovers, leave feedback, and recommend them to others.

KEYWORDS: WEBSITE, VOICE OVER STUDIOS, MOVIES.

		Лакизиук Р.В.			ПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ	9
1.1. Постановка задачі	9
1.2. АНАЛІЗ АНАЛОГІВ ПРОГРАМНОГО ПРОДУКТУ	11
1.3. ОБґРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ТА ВИМОГИ ДО АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ	16
Висновки до першого розділу	22
РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБСАЙТУ ДЛЯ ПЕРЕГЛЯДУ ФІЛЬМІВ	23
2.1. ВИЗНАЧЕННЯ ВАРІАНТІВ ВИКОРИСТАННЯ ТА ОБ'ЄКТНО-ОРІЄНТОВАНОЇ СТРУКТУРИ СИСТЕМИ	23
2.2. Розробка бази даних	32
2.3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ АЛГОРИТМІВ РОБОТИ ВЕБСАЙТУ	36
2.4. РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ СИСТЕМИ КРАУДФАНДИНГОВОЇ ПЛАТФОРМИ	37
Висновки до другого розділу	39
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБСАЙТОМ	41
3.1. Порядок встановлення та налаштування	41
3.2 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ	42
3.3 ТЕСТУВАННЯ РОБОТИ ВЕБСАЙТУ	49
Висновки до третього розділу	50
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТКИ	56

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – Програмне забезпечення;

ОС – Операційна система;

UML – Unified Modeling Language (Уніфікована мова моделювання);

		Лакизюк Р.В.			ПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми полягає у зростаючій потребі користувачів у зручних, інтерактивних та функціональних платформах для перегляду фільмів та серіалів. Сучасний вебсайт для перегляду відеоконтенту повинен надавати широкий спектр можливостей, включаючи перегляд, коментування, додавання нових фільмів, а також підтримку студій озвучки. В умовах зростаючої конкуренції серед платформ, важливо створювати такі вебсайти, які забезпечують високу якість відео, швидкість завантаження та користувацький досвід. Це особливо актуально в епоху цифрової трансформації, де зручний доступ до мультимедійного контенту стає важливим аспектом повсякденного життя.

Метою випускної кваліфікаційної роботи бакалавра є розробка вебсайту для перегляду фільмів та серіалів, що використовує сучасні технології для забезпечення користувачам широких можливостей взаємодії з контентом та один з одним. Зокрема, вебсайт має надавати функції перегляду відео, коментування, додавання нових фільмів, а також створення та підтримки студій озвучки. Для досягнення цієї мети потрібно провести аналіз існуючих аналогічних платформ, розробити алгоритми та архітектуру системи, що забезпечать її ефективну роботу.

Встановлена мета обумовлює наступні завдання:

- аналіз задач, пов'язаних з користуванням вебсайтом для перегляду фільмів та серіалів;
- визначення узагальненої структури, ключових компонентів вебсайту;
- обґрунтування та вибір засобів реалізації продукту;
- проектування структурних складових та алгоритмів роботи вебсайту;
- провадження програмної реалізації вебсайту;
- тестування функціональності та продуктивності вебсайту;
- забезпечення зручного та інтуїтивно зрозумілого інтерфейсу для користувачів.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Об'єктом дослідження є процес надання відеоконтенту користувачам через вебсайт, що включає в себе механізми пошуку, перегляду, коментування та додавання фільмів і серіалів, а також інтеграцію студій озвучки.

Предметом дослідження є функціональність вебсайту для перегляду фільмів та серіалів, його можливості та переваги у порівнянні з існуючими платформами. Дослідження також охоплює аналіз архітектури вебсайту, алгоритмів взаємодії між компонентами системи, а також методів забезпечення якісного користувацького досвіду, таких як швидкість завантаження контенту, зручність інтерфейсу та можливості соціальної взаємодії користувачів.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ

1.1. Постановка задачі

Основною метою кваліфікаційної роботи бакалавра є створення вебсайту для перегляду фільмів, серіалів тощо, використовуючі сучасні інструменти, технології, що дозволить легко, просто користуватись і в подальшому масштабувати, підтримувати.

Вебсайт для перегляду фільмів та серіалів – це онлайн-платформа, яка дозволяє користувачам дивитися відеоконтент у будь-який час та з будь-якого місця. Цей тип сайтів стає все більш популярним через зручність доступу до великої кількості фільмів та серіалів, можливість вибору мов озвучення та субтитрів, а також інтерактивні функції, як-от коментування та оцінка контенту.

Одним із прикладів успішного проекту в цій галузі є сервіс Netflix, який розпочав свою діяльність як компанія з оренди DVD, але швидко перетворився на глобальну платформу для стрімінгу відеоконтенту. Завдяки використанню новітніх технологій, інтуїтивно зрозумілому інтерфейсу та великому асортименту контенту, Netflix став одним із найбільш впізнаваних брендів у світі. Його успіх значною мірою зумовлений можливістю перегляду відео на різних пристроях, від телевізорів до смартфонів, та індивідуальними рекомендаціями на основі уподобань користувачів.

Ключові модулі вебсайту. Вебсайт для перегляду фільмів та серіалів повинен включати кілька основних модулів, що забезпечують різноманітні функції та можливості для користувачів:

- Модуль перегляду відео. Цей модуль забезпечує користувачам можливість переглядати відеоконтент у високій якості, з функціями паузи, перемотування.
- Модуль коментування. Модуль дозволяє користувачам залишати коментарі до фільмів та серіалів, обговорювати їх та ділитися своїми враженнями з іншими користувачами.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

- Модуль додавання контенту. Цей модуль дає можливість користувачам додавати нові фільми та серіали на вебсайт, заповнюючи відповідні форми з інформацією про контент.
- Модуль студій озвучки. Модуль дозволяє користувачам створювати та підтримувати власні студії озвучки, додаючи свої озвучки до фільмів та серіалів.
- Модуль пошуку та навігації. Цей модуль забезпечує зручний пошук контенту за різними критеріями, такими як жанр, рік випуску, рейтинг тощо.

В результаті повинно бути створено вебсайт, на якому будь-який користувач може переглядати фільми та серіали, створювати студії озвучки, додавати новий контент та взаємодіяти з іншими користувачами. Важливою складовою є зручність використання та багатофункціональність, що забезпечить високий рівень задоволеності користувачів.

Основними етапами розробки платформи є:

1. Розробка дизайну вебсайту:

- створення загальної концепції дизайну додатку;
- створення UI компонентів;
- розробка дизайну сторінок.

2. Побудова архітектури клієнтської частини додатку:

- розробка клієнтської логіки додатку;
- інтеграція з API серверної частини;
- забезпечення адаптивності дизайну для різних пристроїв.

3. Побудова архітектури серверної частини додатку:

- створення серверної логіки для обробки запитів користувачів;
- реалізація функцій додавання, видалення та редагування контенту;
- забезпечення зберігання даних у базі даних.

4. Розробка системи авторизації та ролей користувачів:

- створення механізму реєстрації та входу користувачів;
- розподіл ролей (користувач, адміністратор, модератор);

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

- с) управління доступом до різних функцій залежно від ролі користувача.

5. Розробка основного функціоналу:

- а) створення логіки додавання та перегляду фільмів;
- б) розробка можливості створення та управління студіями озвучки;
- с) створення віддаленої бази даних для зберігання інформації;
- д) імплементація бекенду на клієнтській стороні;
- е) організація спілкування серверної частини з клієнтською;
- ф) інтеграція можливості коментування та оцінювання контенту.

6. Тестування додатку:

- а) функціональне тестування додатку;
- б) тестування графічного інтерфейсу;
- с) забезпечення стабільності роботи під високим навантаженням.

1.2. Аналіз аналогів програмного продукту

Розробка платформи для перегляду фільмів та серіалів потребує глибокого аналізу конкурентів на ринку. Оскільки ця галузь є висококонкурентною, важливо вивчити існуючі популярні платформи та їх функціональність.

Для аналізу ринку проведемо огляд наявних популярних платформ для перегляду відеоконтенту, таких як UaKino, UaSerials, Ua-Cinema. Розглянемо їхні ключові функції, переваги та недоліки, що дозволить визначити основні вимоги до розробки нашого вебсайту.

UaKino - великий онлайн-кінотеатр з широким асортиментом фільмів, серіалів та мультфільмів різних жанрів у високій якості. Пропонує безкоштовний перегляд, проте з сумнівною законністю контенту та можливими ризиками.

Одним з таких вебсайтів є Uakino (рис. 1.1.).

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

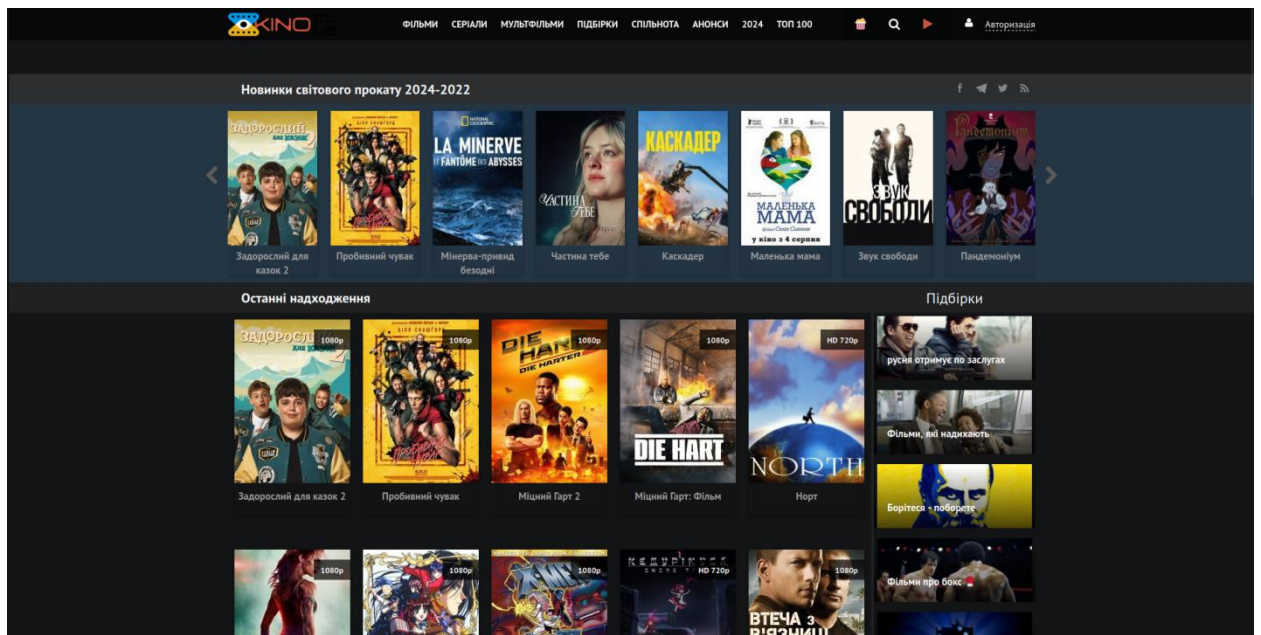


Рис. 1.1. Вебсайт UaKino.

Даний вебсайт чудовим сайтом для перегляду відеоконтенту, має зручний інтерфейс, є адаптивним під різні пристрої, різноманітність відеоконтенту, анонси.

Основними перевагами є:

- Широкий асортимент контенту: сайт пропонує різноманітні жанри, від комедій до бойовиків і мультфільмів;
- Зручний інтерфейс: фільми та серіали розділені на категорії, що полегшує пошук потрібного контенту;
- Високоякісні посилання: сайт пропонує посилання на потокове відео з високою роздільною здатністю;
- Безкоштовний перегляд: контент доступний для безкоштовного перегляду;
- адаптивність на різних пристроях;
- новини;
- підбірка.

Недоліками є:

- Можливість реклами: на сайті може бути небажана реклама.

Ua-Cinema - сайт з великою колекцією фільмів та серіалів, включаючи українські, з зручною навігацією та категоризацією. Безкоштовний перегляд,

		Лакизюк Р.В.			ПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

але існують питання з авторськими правами та безпекою (рис. 1.2.).

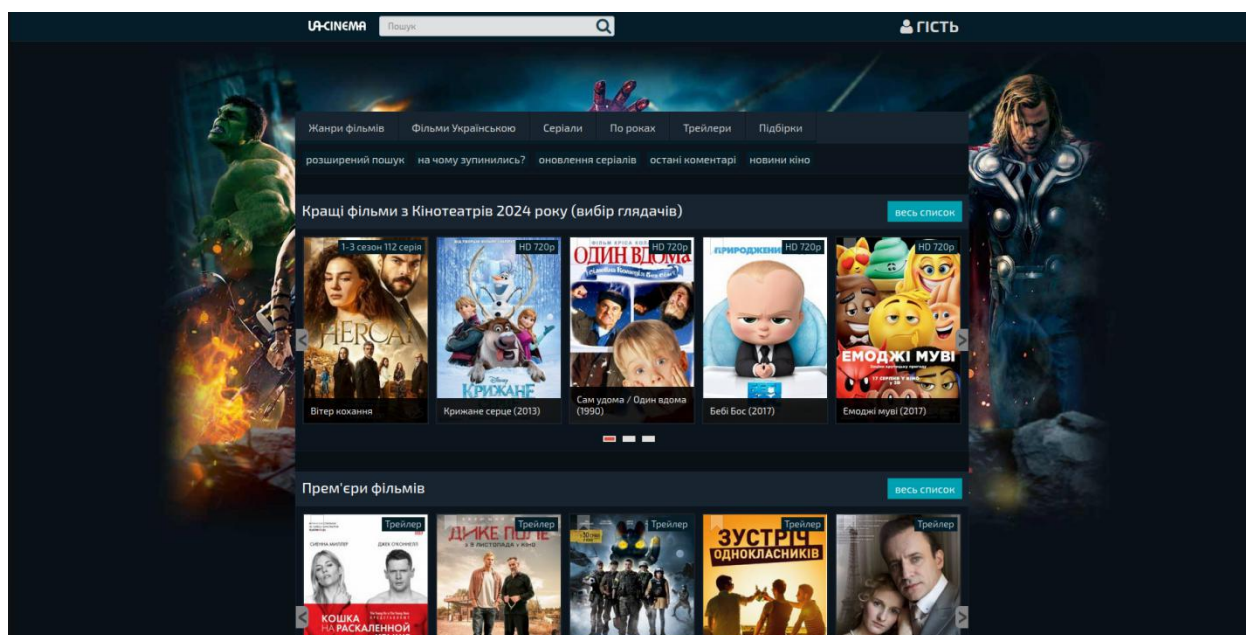


Рис. 1.2. Вебсайт Ua-Cinema.

Переваги:

- Великий вибір фільмів та серіалів різних жанрів;
- Окремий розділ для українських фільмів та серіалів;
- Наявність трейлерів для ознайомлення з фільмами;
- Категоризація контенту за роками, жанрами тощо для зручного пошуку;
- Безкоштовний перегляд більшості контенту.

Недоліки:

- Обмежена кількість якісного контенту високої роздільної здатності;
- Наявність реклами та можливість несподіваних переадресувань на сторонні сайти.

UaSerials - ресурс з вибором серіалів, фільмів та мультсеріалів різних жанрів. Має зручний інтерфейс та можливість створювати списки для відстеження переглядів. Безкоштовний, але з юридичними та технічними ризиками (рис. 1.3.).

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

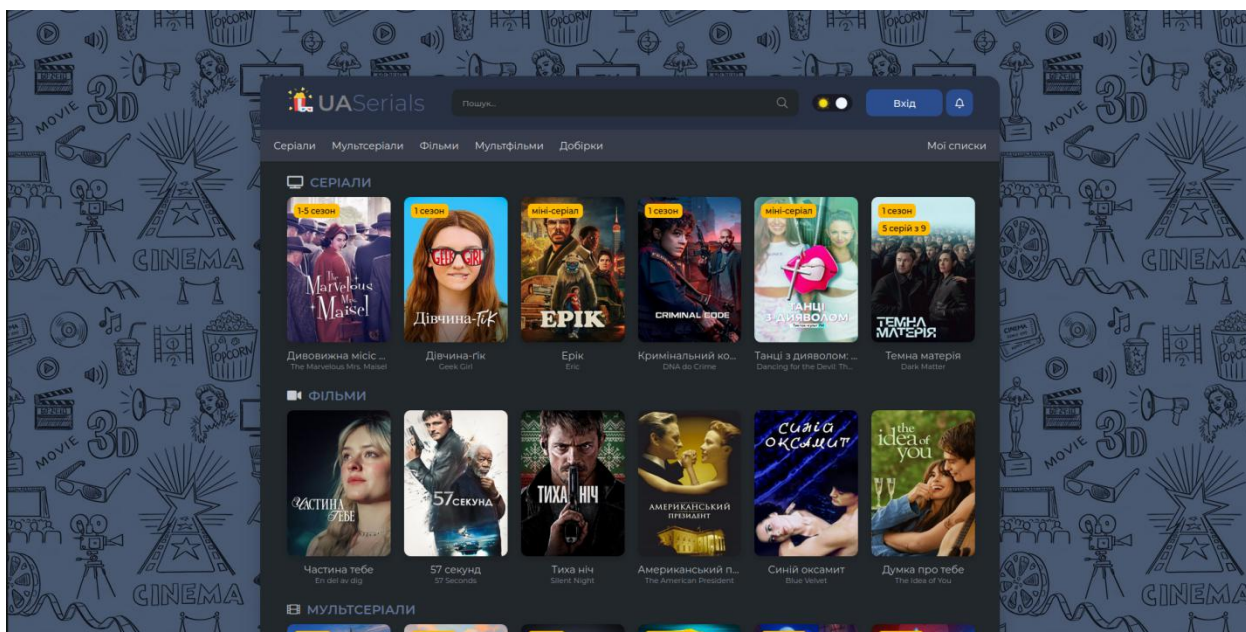


Рис. 1.3. Вебсайт UaSerials.

Переваги:

- Широкий вибір серіалів, фільмів та мультсеріалів;
- Зручний інтерфейс з категоризацією контенту;
- Підбірки новинок і популярного контенту;
- Можливість додавати серіали до списку "Мої списки" для відстеження переглядів;
- Безкоштовний доступ до більшості контенту.

Недоліки:

- Обмежена інформація про серіали/фільми - відсутні рецензії, описи тощо;
- Нестабільна якість контенту, переважно стандартна роздільна здатність.

Виходячи з наявних переваг та недоліків, складемо порівняльну таблицю вищезгаданих програмних аналогів (табл. 1.1).

Таблиця 1.1

Таблиця порівняння вебсайтів для перегляду відеоконтенту

Характеристики	UaKino	Ua-Cinema	UaSerials
Вибір контенту	Широкий	Широкий	Широкий

	(фільми, серіали, мультфільми)	(особливо українські)	(серіали, фільми, мультсеріали)
Якість контенту	Висока (1080p)	Здебільшого стандартна	Переважно стандартна
Безкоштовний перегляд	Так	Так	Так
Категоризація контенту	Присутня	Присутня	Присутня
Трейлери	Немає	Є	Немає
Списки для відстеження	Немає	Немає	Присутні
Реклама	Присутня	Присутня	Присутня

Переглянувши популярні платформи для перегляду фільмів та серіалів, можна виділити основні відмінності: деякі з них пропонують безкоштовний доступ з рекламними вставками. Вебсайти також відрізняються за підтримкою різних типів контенту: нові релізи, класичні фільми, документальні стрічки тощо. Дані вебсайти орієнтовані на локальні аудиторію.

Враховуючи інформацію, отриману при дослідженні, було прийнято рішення створити вебсайт для перегляду фільмів та серіалів, яка буде враховувати всі ці аспекти та запропонує користувачам новий рівень зручності та доступності. Платформа буде підтримувати функціонал для додавання та управління власними студіями озвучки, що дозволить користувачам створювати власні озвучення для контенту.

Таким чином, даний вебсайт має бути дуже простим у використанні, з інтуїтивним дизайном; надавати можливість взаємодії користувачів через коментарі та рейтинги; підтримувати будь-які категорії відеоконтенту. Платформа також буде інтегрована з соціальними мережами для полегшення спільного перегляду та обговорення фільмів та серіалів.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Вебдодаток буде побудований на клієнт-серверній архітектурі. Серверна частина буде представлена у вигляді API та буде написана з використанням фреймворку Express Js. Для клієнтської частини буде використано фреймворк Next Js, який побудований на основі бібліотеки ReactJS. Як сховище буде використовуватись база даних PostgreSQL, а для взаємодії з базою даною буде використовуватись Prisma ORM. Для збереження медіа файлів використано Google Cloude Storage.

Express.js — це мінімалістична та гнучка фреймворк для веб-додатків на Node.js, яка надає безліч потужних функцій для створення веб-додатків і API. Маючи в своєму розпорядженні безліч допоміжних HTTP-методів та проміжних обробників, створювати надійні API можна легко і швидко. Express забезпечує тонкий прошарок базової функціональності для веб-застосунків, що не спотворює звичну та зручну функціональність Node.js.

Переваги Express.js:

- *Простота у використанні.* Express.js має простий і інтуїтивно зрозумілий API, що дозволяє швидко розпочати роботу навіть новачкам у Node.js.
- *Гнучкість.* Завдяки мідлварам та модульній структурі, Express.js дозволяє налаштовувати додаток під конкретні вимоги та легко інтегрувати сторонні бібліотеки.
- *Висока продуктивність.* Express.js побудований на основі швидкого та асинхронного двигуна Node.js, що забезпечує високу продуктивність і можливість обробляти великий обсяг запитів.
- *Широке співтовариство.* Велике та активне співтовариство розробників забезпечує наявність безлічі навчальних матеріалів, прикладів, плагінів та розширень для Express.js.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

- *Сумісність з іншими технологіями.* Express.js легко інтегрується з іншими технологіями та базами даних, такими як MongoDB, PostgreSQL, Redis тощо.

Переваги Express.js для вебсайту з перегляду фільмів:

- *Швидка розробка.* Express.js дозволяє швидко створити прототип вебсайту та реалізувати основні функції, такі як реєстрація користувачів, авторизація, додавання та перегляд фільмів.
- *Масштабованість.* Завдяки асинхронній природі Node.js та гнучкості Express.js, можна легко масштабувати додаток для обробки великої кількості одночасних користувачів.
- *Обробка мультимедійних даних.* Express.js дозволяє легко інтегруватися з CDN та серверами для обслуговування мультимедійного контенту, що є важливим для платформи з перегляду фільмів.
- *Підтримка RESTful API.* Express.js спрощує створення RESTful API, що дозволяє легко інтегрувати фронтенд з бекендом та забезпечити взаємодію між компонентами системи.
- *Безпека.* За допомогою мідлварів можна легко додати різні рівні безпеки, такі як захист від CSRF, захист маршрутів та обробку автентифікації.
- *Можливість розширення.* Велика кількість доступних плагінів та модулів дозволяє розширювати функціональність платформи без необхідності писати багато коду з нуля.

Next.js — це потужний фреймворк для створення веб-додатків на базі React, розроблений командою Vercel. Next.js надає широкий набір інструментів для побудови сучасних веб-додатків, включаючи серверний рендеринг, статичну генерацію сторінок, маршрутизацію та багато інших корисних функцій.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

Переваги Next.js:

- *SEO-оптимізація.* Завдяки серверному рендерингу та статичній генерації сторінок, Next.js покращує індексацію сайтів пошуковими системами.
- *Висока продуктивність.* Статична генерація та оптимізація зображень значно покращують швидкість завантаження сторінок, що забезпечує кращий досвід користувачів.
- *Простота розробки.* Автоматична маршрутизація, підтримка CSS та TypeScript, а також API Routes роблять процес розробки значно простішим та швидшим.
- *Масштабованість.* Next.js легко масштабується від простих статичних сайтів до складних динамічних додатків з великою кількістю користувачів.
- *Гнучкість.* Next.js підтримує різні методи рендерингу (SSR, SSG, ISR), що дозволяє вибрати оптимальний підхід для конкретних завдань.

Переваги Next.js для вебсайту з перегляду фільмів:

- *Поліпшена SEO-оптимізація.* Серверний рендеринг дозволяє індексувати сторінки фільмів та серіалів, покращуючи видимість у пошукових системах та залучаючи більше трафіку.
- *Швидке завантажуваність.* Статична генерація сторінок фільмів забезпечує швидке завантаження контенту для користувачів, що покращує загальний досвід перегляду.
- *Динамічне оновлення контенту.* Інкрементальна статична генерація дозволяє динамічно оновлювати сторінки фільмів та серіалів без необхідності повного перезавантаження сайту.
- *Легка інтеграція з базами даних та API.* API Routes в Next.js дозволяють легко створювати серверні функції для роботи з базами даних та зовнішніми API, що є важливим для зберігання та відображення інформації про фільми.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

- *Можливість використання TypeScript.* Підтримка TypeScript дозволяє створювати більш надійний та стабільний код, зменшуючи кількість потенційних помилок.
- *Оптимізація зображень та мультимедіа.* Вбудована оптимізація зображень покращує завантаження великих зображень фільмів, афіш та інших мультимедійних файлів, що є ключовим для вебсайту з перегляду фільмів.

PostgreSQL — це потужна, відкрита система керування реляційними базами даних (RDBMS), яка отримала широке визнання завдяки своїй надійності, гнучкості та можливостям. Вона підтримує SQL-стандарт (і більшість його розширень), що робить її універсальним інструментом для зберігання, обробки та аналізу даних. PostgreSQL, також відомий як "Postgres".

Переваги PostgreSQL:

- *Безкоштовність та відкритий код.* PostgreSQL є повністю безкоштовним і відкритим програмним забезпеченням з ліцензією PostgreSQL, яка дозволяє використовувати, модифікувати та поширювати її без будь-яких обмежень.
- *Надійність та стабільність.* PostgreSQL відомий своєю стабільністю та надійністю. Він успішно використовується у великих корпоративних середовищах, де надійність є критично важливою.
- *Гнучкість та розширюваність.* Можливість додавання нових функцій, типів даних та індексів робить PostgreSQL надзвичайно гнучким інструментом для вирішення різноманітних завдань.
- *Потужний механізм транзакцій.* Підтримка ACID-транзакцій забезпечує консистентність даних та їх цілісність навіть у складних сценаріях.
- *Широка підтримка спільноти.* Велика спільнота розробників та користувачів забезпечує постійну підтримку та розвиток PostgreSQL, а також доступ до великої кількості ресурсів та документації.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

Переваги PostgreSQL для вебсайту з перегляду фільмів:

- *Обробка великих обсягів даних.* PostgreSQL може ефективно обробляти та зберігати велику кількість даних про фільми, серіали, студії озвучки, користувацькі коментарі та інші метадані.
- *Розширені можливості пошуку.* Вбудований повнотекстовий пошук дозволяє швидко та ефективно знаходити інформацію про фільми та серіали за ключовими словами, описами та іншими текстовими атрибутами.
- *Складні типи даних.* Підтримка масивів, JSON та інших складних типів даних дозволяє гнучко зберігати та обробляти різноманітні атрибути фільмів, включаючи рейтинг, жанри, акторів та інші.
- *Транзакційна цілісність.* Завдяки підтримці ACID-транзакцій, система забезпечує цілісність даних навіть у випадках одночасного редагування чи додавання великої кількості інформації.
- *Висока доступність.* Механізми реплікації та резервного копіювання дозволяють забезпечити високу доступність даних та мінімізувати ризики втрати інформації.
- *Розширюваність.* Можливість створення власних функцій та процедур дозволяє адаптувати систему під конкретні вимоги проєкту, наприклад, автоматизацію завантаження нових даних про фільми.
- *Безпека.* PostgreSQL надає потужні засоби контролю доступу, шифрування та управління правами користувачів, що є критично важливим для захисту конфіденційної інформації.

Prisma ORM — це сучасний ORM (Object-Relational Mapping) інструмент, який суттєво полегшує взаємодію з базами даних у сучасних веб-додатках. Prisma допомагає розробникам з'єднуватися з базами даних, створювати запити, та керувати даними за допомогою зрозумілого та інтуїтивно зрозумілого API.

Переваги Prisma ORM:

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

- *Простота та швидкість розробки.* Інтуїтивно зрозумілий API та автоматизація рутинних задач значно прискорюють процес розробки.
- *Зменшення кількості помилок.* Типізовані клієнти для TypeScript дозволяють виявляти помилки на етапі компіляції, що значно підвищує надійність коду.
- *Автоматизація міграцій.* Автоматичне генерування міграцій на основі змін у схемі бази даних спрощує управління версіями структури даних.
- *Гнучкість.* Підтримка різних СУБД дозволяє використовувати Prisma у різних проектах незалежно від обраної бази даних.
- *Ефективність.* Можливість виконання складних реляційних запитів та автоматичне генерування SQL запитів підвищує продуктивність розробки.

Переваги Prisma ORM для вебсайту з перегляду фільмів:

- *Швидка та ефективна розробка.* Використання Prisma дозволяє швидко створювати та керувати базою даних для вебсайту, що значно прискорює процес розробки.
- *Типізовані дані.* Типізовані клієнти забезпечують додатковий рівень безпеки та зменшують ймовірність помилок у коді, що особливо важливо при роботі з великими обсягами даних про фільми, серіали та користувачів.
- *Зручне керування даними.* Prisma Studio дозволяє легко переглядати та редагувати дані в базі даних через зручний інтерфейс користувача, що полегшує адміністрування.
- *Автоматизація міграцій.* Зручні інструменти для міграцій дозволяють легко оновлювати структуру бази даних при додаванні нових функцій або зміні вимог.
- *Підтримка складних реляційних зв'язків.* Вебсайт з перегляду фільмів може мати складну структуру даних з багатьма реляційними зв'язками між фільмами, серіалами, студіями озвучки та користувачами. Prisma дозволяє ефективно керувати цими зв'язками.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

- *Гнучкість та масштабованість.* Prisma дозволяє легко масштабувати базу даних та адаптувати її під зростаючі потреби проекту, що важливо для вебсайту з великою кількістю користувачів та даних.

Висновки до першого розділу.

Проведено аналіз предметної області, в рамках якого були визначені основні вимоги до платформи для перегляду фільмів та серіалів. Основною задачею цього вебсайту є надання користувачам можливості перегляду, коментування, додавання фільмів та серіалів, а також додавання студій озвучення.

В процесі дослідження аналогічних і схожих за концепцією продуктів було проаналізовано та порівняно різні платформи для перегляду відеоконтенту. Основний акцент був зроблений на функціональність, зручність користування та інтерактивні можливості, які надаються користувачам.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБСАЙТУ ДЛЯ ПЕРЕГЛЯДУ ФІЛЬМІВ

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

У цьому розділі представлено аналіз вимог до вебсайту для перегляду фільмів та серіалів з акцентом на виявленні функціональних та нефункціональних вимог. Крім того, визначено вимоги до часових характеристик реакції додатку на дії користувача, а також окреслено різні варіанти використання. Для забезпечення чіткого розуміння структури додатково включено узагальнену діаграму елементів (сутностей системи), їх структури та методів з описом.

Аналіз вимог передбачає докладне вивчення потреб та очікувань користувачів щодо функціональності та характеристик вебсайту для перегляду фільмів та серіалів. Основною метою аналізу вимог є зрозуміти, що потрібно втілити в програмному забезпеченні для задоволення потреб користувачів. За допомогою аналізу, опитувань та зворотного зв'язку були визначені функціональні вимоги, які окреслюють специфічні особливості та можливості, якими повинен володіти програмний продукт. На додаток були зібрані нефункціональні вимоги, які стосуються таких аспектів, як продуктивність, стабільність роботи і зручність використання.

Функціональні вимоги

Пошук та фільтрація контенту:

- Користувач повинен мати можливість швидко знаходити фільми та серіали за різними критеріями, включаючи жанр, рік випуску, рейтинг та інші параметри.

Перегляд відео:

- Вебсайт повинен забезпечувати можливість стрімінгового перегляду фільмів та серіалів без затримок або перерв у відтворенні.

Коментування та відгуки:

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

- Користувачі повинні мати можливість залишати коментарі та відгуки під відео, що дозволяє спілкуватися та обмінюватися враженнями про переглянутий контент.

Інтеграція з соціальними мережами:

- Можливість авторизації та входу через соціальні мережі для спрощення процесу реєстрації та облікового запису користувача.

Мобільна сумісність:

- Вебсайт повинен бути оптимізований для перегляду на мобільних пристроях, забезпечуючи адаптивний дизайн та зручну навігацію через мобільні додатки або браузерери.

Нефункціональні вимоги

Швидка реакція системи:

- Система повинна швидко реагувати на дії користувача, забезпечуючи безперебійну та чутливу роботу. Це включає миттєве завантаження сторінок, швидке відтворення відео та плавну навігацію між різними розділами вебсайту.

Захист даних:

- Вебсайт повинен забезпечувати безпеку та конфіденційність збережених даних користувачів. Це включає захист персональних даних, історії переглядів та іншої чутливої інформації.

Інтуїтивний інтерфейс:

- Вебсайт повинен мати зрозумілий та легкий у використанні інтерфейс. Це включає чітку навігацію, зручні пошукові функції та інтуїтивно зрозумілі елементи управління для покращення користувацького досвіду.

Підтримка різних пристроїв та браузерів:

- Вебсайт повинен коректно працювати на різних пристроях, включаючи настільні комп'ютери, планшети та смартфони, а також підтримувати основні браузери, такі як Chrome, Firefox, Safari та Edge. Це забезпечить доступність вебсайту для широкої аудиторії.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Масштабованість:

- Вебсайт повинен бути масштабованим, щоб витримувати високі навантаження під час пікових періодів активності. Це важливо для забезпечення стабільної роботи системи при великій кількості одночасних користувачів.

Безперебійне оновлення:

- Вебсайт повинен підтримувати можливість безперебійного оновлення контенту та функціоналу без необхідності зупинки роботи сервісу. Це дозволить постійно вдосконалювати систему та забезпечувати актуальність контенту.

Часові характеристики реакції системи:

Час відгуку на дії користувача:

- Швидкий час відгуку забезпечує миттєву відповідь на дії користувача, що поліпшує загальний досвід перегляду. Вебсайт повинен швидко реагувати на кліки, переходи між сторінками та інші інтерактивні дії, забезпечуючи плавний та безперервний досвід користувача.

Час завантаження сторінок:

- Головна сторінка та інші ключові сторінки повинні завантажуватися не більше ніж за 5 секунди. Це включає швидке відображення списку фільмів та серіалів, категорій, а також персоналізованих рекомендацій для користувачів.

Час відтворення відео:

- Відтворення фільмів та серіалів повинно починатися протягом декількох секунд після натискання кнопки "Play". Це включає швидке завантаження та буферизацію відео для забезпечення безперервного та високоякісного відтворення без затримок та зупинок.

Час завантаження зображень і обкладинок:

- Зображення обкладинок та постерів фільмів і серіалів повинні завантажуватися швидко та відображатися миттєво. Оптимізація розміру зображень та використання кешування допомагають

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечити швидке завантаження та мінімізувати час очікування користувачів.

Час виконання пошукових запитів:

- Пошукові запити повинні виконуватися швидко, з результатами, що відображаються протягом 5 секунд після введення пошукового терміну. Це дозволяє користувачам легко знаходити потрібний контент без затримок.

Час додавання та збереження коментарів:

- Коментарі користувачів повинні додаватися та відображатися миттєво після натискання кнопки "Відправити". Це забезпечує активну взаємодію та обговорення контенту між користувачами.

Час обробки даних користувачів:

- Персоналізовані рекомендації та інші дані користувачів повинні оновлюватися в реальному часі без значних затримок. Це включає оновлення списків переглянутого вподобань та коментування.

Час відкриття та закриття вебсайту:

- Швидке відкриття вебсайту дозволяє користувачам швидко почати перегляд контенту, а швидке закриття забезпечує збереження часу та енергії користувача. Вебсайт повинен бути оптимізованим для швидкого запуску та закриття на різних пристроях та платформах.

Важливо враховувати що на продуктивність, швидкість, стабільність впливає якість інтернет-з'єднання.

Варіанти використання. Користувачі можуть шукати фільми та серіали за різними критеріями, такими як жанр, рік випуску, актори, режисери тощо. Вибір контенту може бути розширений за допомогою рекомендаційних систем або популярних запитів та можливість оцінювати контент, залишати відгуки та коментарі, ділитися враженнями з іншими користувачами.

Діаграма використання відображає основні функціональні можливості вебсайту та взаємодію користувачів з системою через різні випадки

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

використання. Основними акторами на діаграмі є користувачі, які виконують такі дії, як пошук, перегляд, оцінювання, коментування. Оцінювання і коментування можуть виконувати тільки автентифіковані/авторизовані користувачі (рис. 2.1.).



Рис. 2.1. Діаграма варіантів використання.

Ці характеристики та функціональні можливості спрямовані на створення зручного та ефективного інтернет-ресурсу для перегляду фільмів та серіалів, який задовольняє потреби та очікування користувачів у сфері розваг.

Опис діаграми варіантів використання наведено в таблицях 2.1 – 2.6.

Таблиця 2.1

Опис прецеденту «Вхід/Реєстрація»

Назва варіанту використання		Вхід/реєстрація	
Основні актори		Адміністратор, модератор, студія озвучки, користувач	
Короткий опис		Користувачі можуть зареєструватись,	

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

	щоб переглядати, оцінювати, коментувати відеоконтент, також є авторизація для адміністраторів, модераторів і студій озвучки
--	---

Таблиця 2.2

Опис прецеденту «Створення фільму, додавання відео до фільмів»

Назва варіанту використання	Створення фільму
Основні актори	Адміністратор, студії озвучки
Короткий опис	Адміністратор створює фільм (заповнює всі дані про фільм), студії озвучки додають озвученні файли до існуючого фільму

Таблиця 2.3

Опис прецеденту «Пошук і фільтрування»

Назва варіанту використання	Пошук і фільтрування
Основні актори	Користувачі
Короткий опис	Користувачі можуть шукати відеоконтент за назвою, а також фільтрувати відеоконтент за критеріями

Таблиця 2.4

Опис прецеденту «Модерація»

Назва варіанту використання	Модерація
Основні актори	Модератори
Короткий опис	Модератори стежать за коментаріями які написали користувачі, якщо в результаті є несумісні коментарі модератори їх скривають

Таблиця 2.5

Опис прецеденту «Додавання банерів, модераторів»

Назва варіанту використання	Додавання банерів, модераторів
Основні актори	Адміністратор

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

Короткий опис	Адміністратор має можливість додавати банери і надавати користувачам права модератора
---------------	---

Таблиця 2.6

Опис прецеденту «Оцінювання, коментування»

Назва варіанту використання	Оцінювання, коментування
Основні актори	Користувачі
Короткий опис	Користувачам є можливість оцінювати фільм, також написати коментарій

Визначені функціональні можливості для користувача створюють комплексне рішення, яке задовольняє різноманітні графічні потреби.

Загалом, діаграма варіантів використання надає всебічний і наочний огляд функціональних можливостей. Визначення та документування цих варіантів робить діаграму важливим інструментом для аналізу системи, збору вимог та проектування.

Аналіз функціональних вимог дозволив виділити наступні сутності, що забезпечать програмну реалізацію додатку. На рисунку 2.2. наведено Діаграму класів, яка використовується для його роботи.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

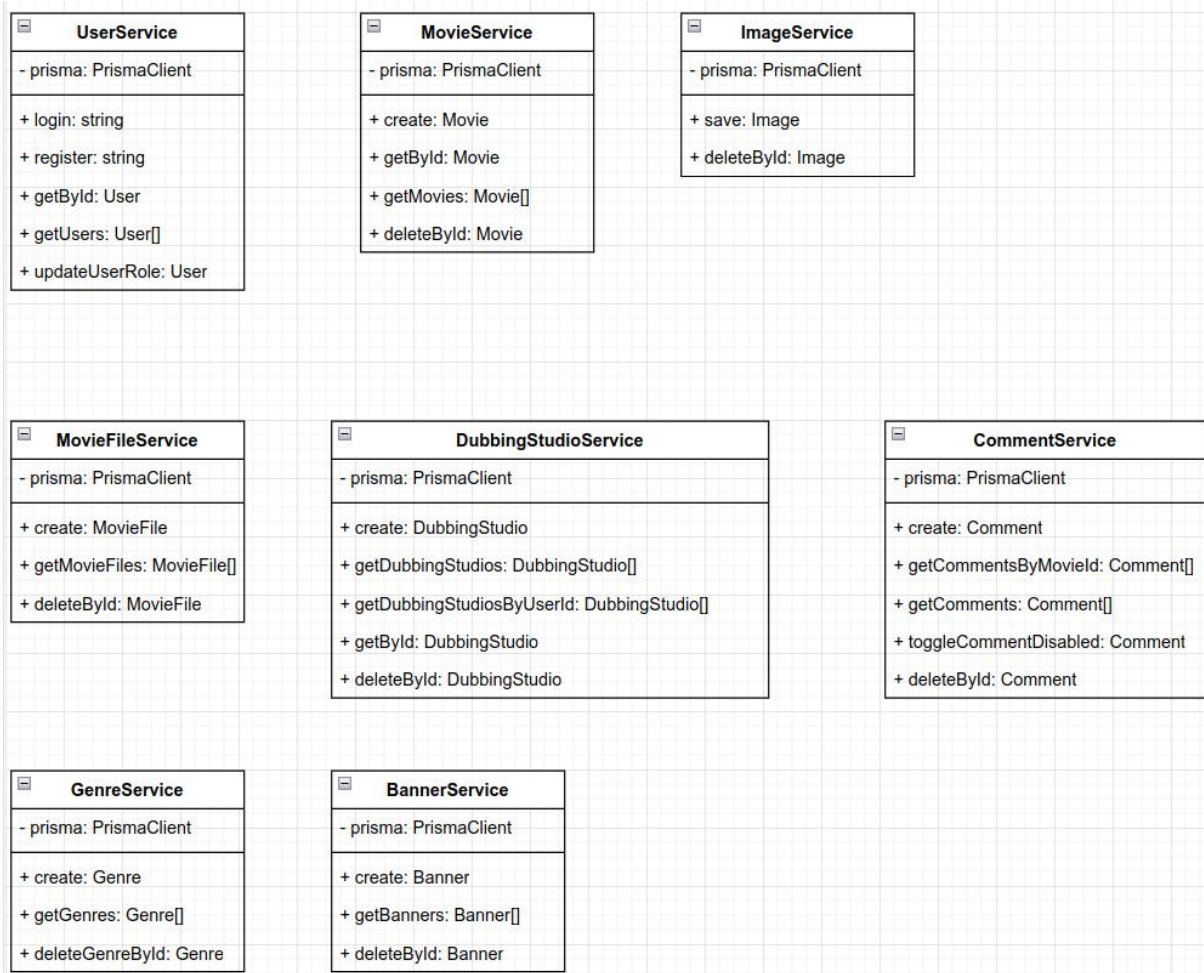


Рис. 2.2. Діаграма основних класів.

UserService – клас який відповідає за користувача:

- login - відповідає за авторизацію/автентифікацію;
- register - відповідає за реєстрацію;
- getById - повертає користувача за ідифікатором
- getUsers - повертає користувачі;
- updateUserRole - оновлює роль користувача.

MovieService – клас який відповідає за фільми:

- create - створює фільм;
- getById - повертає фільм за id ;
- getMovies - повертає фільми;
- deleteById - видаляє фільм за id.

ImageService – клас який відповідає за картинки:

- save - зберігає, створює;
- deleteById - видаляє за id.

MovieFileService – клас який відповідає за відео файли фільмів:

- create - створює файл до фільм;
- getMovieFiles - повертає відеофайли конкретного фільма;
- deleteById - видаляє за id.

DubbingStudioService – клас який відповідає за студії озвучки:

- create - створює студію озвучки;
- getDubbingStudios - повертає всі студії озвучки;
- getDubbingStudiosByUserId - повертає студії озвучки за id користувача;
- getById - повертає студію озвучки за id;
- deleteById - видаляє студію озвучки за id.

CommentService – клас який відповідає за коментарі:

- create - створює коментра;
- getCommentsByMovieId - повертає коментарі за id фільму;
- getComments - повертає коментарі;
- toggleComentDisabled - переключає стан is_disabled;
- deleteById - видалає за id.

GenreService – клас який відповідає за жанри фільму:

- create - створює жанр;
- getGanres - повертає жанри фільму;
- deleteGenreById - видаляє за id жанру.

BannerService – клас який відповідає за банери:

- create - створює банер;
- getBanners - повертає банери;
- deleteById - видаляє банер за його id.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2. Розробка бази даних

Для збереження файлів та інформації у додатку використовуються PostgreSQL, це система управління базами даних, яка базується на реляційній моделі.

База даних вебсайту складається з 11 таблиць:

- banner - таблиця, яка зберігає дані про банери;
- comment - таблиця, яка зберігає коментарі;
- dubbing_studio - таблиця, яка зберігає дані про студії озвучки;
- dubbing_studio_user - таблиця, яка формує зв'язки багато до багатьох таблиць dubbing_studio і user;
- genre - таблиця, яка зберігає жанри фільмів;
- image - таблиця, яка зберігає дані про картинки;
- movie - таблиця, яка зберігає дані про фільми;
- movie_file - таблиця, яка зберігає дані про файли фільмів;
- movie_genre - таблиця, яка формує зв'язки багато до багатьох таблиць movie і genre;
- movie_reaction - таблиця, яка зберігає реакції фільму (лайки, дизлайки);
- user - таблиця, яка зберігає інформацію про користувачів.

Опис кожної з таблиць бази даних наведений у таблицях 2.7 – 2.15

Таблиця 2.7

Опис таблиці user

Поле	Тип	Призначення
id	int	
email	varchar	
password	varchar	
nickname	varchar	
avatar	text	
register_date	timestamp	
date_birth	timestamp	

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

role	varchar	
------	---------	--

Таблиця 2.8

Опис таблиці movie

Поле	Тип	Призначення
movie_id	int	
movie_name	text	
short_description	text	
description	text	
release_date	timestamp	
age_category	int	
producers	_text	
actors	_text	
countries	_text	

Таблиця 2.9

Опис таблиці genre

Поле	Тип	Призначення
genre_id	int	
genre_name	varchar	

Таблиця 2.10

Опис таблиці dubbing_studio

Поле	Тип	Призначення
dubbing_studio_id	int	
dubbing_studio_name	varchar	
email	text	
phone	varchar	
contact_person	varchar	
address	text	
languages	_text	

Таблиця 2.11

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

Опис таблиці comment

Поле	Тип	Призначення
comment_id	int	
movie_id	int	
user_id	int	
comment	text	
is_disabled	bool	
creation_date	timestamp	

Таблиця 2.12

Опис таблиці banner

Поле	Тип	Призначення
banner_id	int	
file_id	varchar	
file_url	text	
redirect_url	text	

Таблиця 2.13

Опис таблиці image

Поле	Тип	Призначення
image_id	int	
movie_id	int	
file_id	varchar	
file_url	text	

Таблиця 2.14

Опис таблиці movie_file

Поле	Тип	Призначення
movie_file_id	int	
movie_id	int	
dubbing_studio_id	int	
file_id	varchar	

file_url	text	
language	varchar	
quantity	int	

Таблиця 2.15

Опис таблиці movie_reaction

Поле	Тип	Призначення
reaction_id	int	
movie_id	int	
user_id	int	
type	varchar	

У контексті вебсайту для перегляду фільмів і серіалів, дані після їх завантаження зберігаються в базі даних та файловій системі сервера. Система зберігання даних складається з кількох компонентів, кожен з яких виконує свою унікальну функцію для забезпечення надійності, швидкості доступу та безпеки.

Коли новий фільм або серіал додається до платформи, дані, такі як метадані (назва, опис, актори, жанр, дата виходу), а також медіафайли (відео, зображення постера) завантажуються до бази даних та файлової системи. Ці дані стають доступними для користувачів через вебінтерфейс.

Для звернення до даних використовується REST API, який надає доступ до різних ресурсів вебсайту. API забезпечує функції для отримання, додавання, оновлення та видалення даних про фільми та серіали. Наприклад, можна використовувати наступні кінцеві точки:

GET /api/movies – отримання списку всіх фільмів

GET /api/movies/{id} – отримання інформації про конкретний фільм

POST /api/movies – додавання нового фільму

PATCH /api/movies/{id} – оновлення інформації про фільм

DELETE /api/movies/{id} – видалення фільму

2.3. Проектування та реалізація алгоритмів роботи вебсайту

Вебсайт є простим у використанні, має всі основні функції платформи, що дозволяє користувачам легко взаємодіяти з розробленим вебсайтом. Алгоритм роботи з вебсайтом зображено на рисунку 2.3, також детально описано його пункти.

Якщо авторизувався адміністратор:

- після авторизація адміністратор потрапляє на головну сторінку сайту, де вже може перейти на адмін панель;
- в адмін панелі адміністратор може переглядати фільми, баннери, жанри фільмів, фільми, студії озвучки;
- створення та видалення фільмів, банерів, жанрів фільму, студії озвучки;
- адміністратор має можливість призначати нових модераторів.

Якщо авторизувався модератор:

- після авторизація модератор потрапляє на головну сторінку сайту, де може перейти на панель модератора;
- модератор може переглядати коментарі;
- також має можливість коментарі скривати, якщо вони є недопустимими.

Якщо авторизувалась студія озвучки:

- після авторизації потрапляє на головну сторінку сайту, де може переходити на панель студії озвучки;
- може додавати, переглядати, видаляти відеофайли фільму своє студії.

Якщо авторизувався користувач:

- після авторизації потрапляє на головну сторінку сайту;
- може обирати відеоконтент і переглядати його;
- має можливість пошуку фільму за назвою;
- користувач може користуватись фільтром щоб відфільтрувати відеоконтент;
- може залишати коментарі під фільмом;
- також може ставити лайки і дизлайки.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

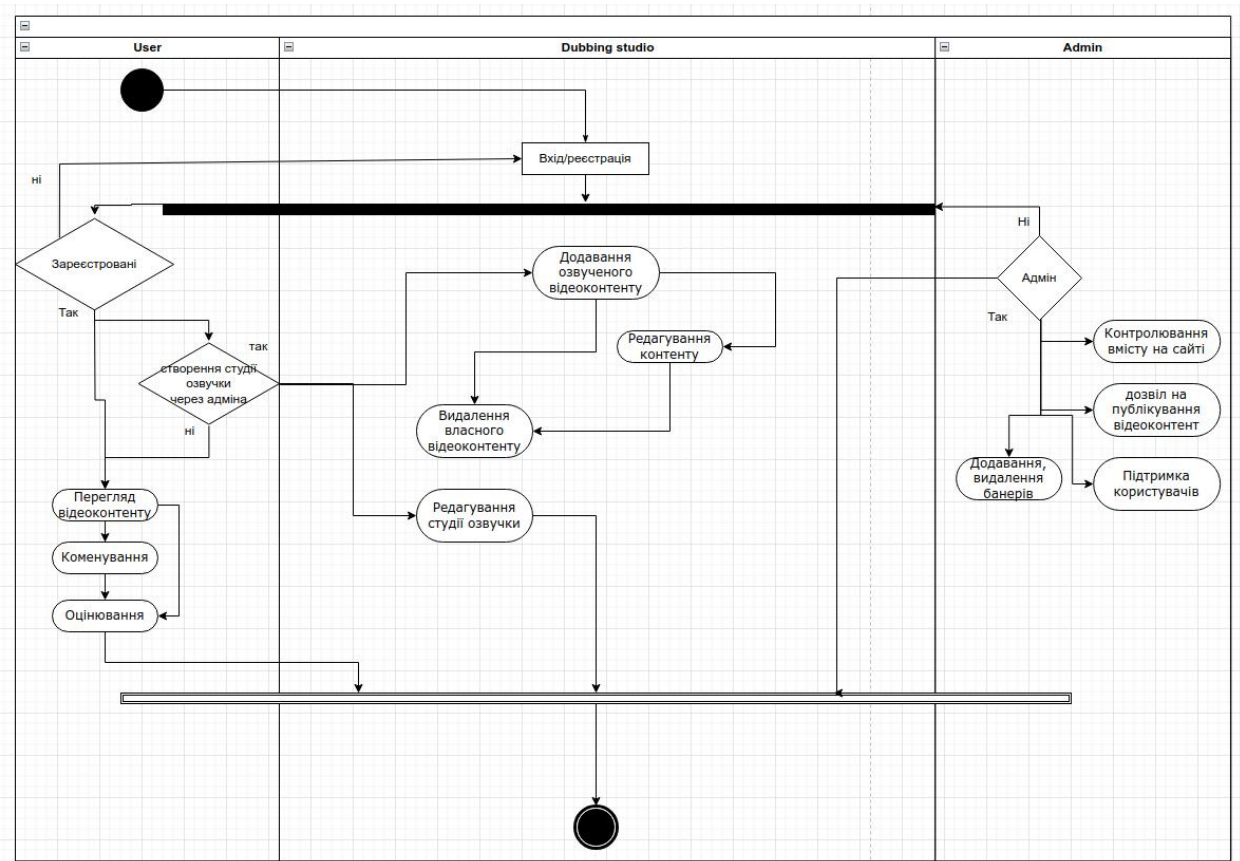


Рис. 2.3. Діаграма активності.

2.4. Реалізація функціоналу системи краудфандингової платформи

У цьому розділі ми обговоримо етапи процесу впровадження та висвітимо ключові аспекти розробки вебсайту. Впровадження такого проекту передбачає розробку клієнтської та серверної частин, а також використання API для забезпечення комунікації між ними.

Розглянемо основні етапи створення фільму на вебсайті. Для цього адміністратор повинен зайти в адмін панель та заповнити інформацію про фільм. Метод створення фільму зображення на рисунку 2.4.

Адміністратор заповнює поля, що містять інформацію про назву фільму, опис, режисера, акторський склад, жанри, дату випуску та інші важливі дані. Після збереження цих даних, фільм стає доступним на вебсайті для подальшої обробки.

```
async create(movieData: IMovieCreate, files: FileArray) {
  const isValidMovieDate = validateCreateMovie(movieData);
```

```

if ( !isValidMovieDate ) throw ApiError.BadRequest();

const movie = await this.prisma.movie.create({
  data: { ...movieData }
});

if ( !movie ) throw ApiError.InternalServerError();

await this.prisma.movie_genre.createMany({
  data: movieData.genres.map(genreId => ({
    movie_id: movie.movie_id,
    genre_id: genreId
  }))
});

const _files = await fileService.uploadFilesToGoogleCloudStorage(files);
const fileId = Array.isArray(_files) ? _files[0].fileId : _files.fileId,
fileUrl = Array.isArray(_files) ? _files[0].url : _files?.url;

if ( !fileUrl ) throw ApiError.InternalServerError();

const image = await imageService.save({
  file_id: fileId,
  file_url: fileUrl,
  movie_id: movie.movie_id
});

return { movie, image };
}

```

Після створення адміном фільму, студії озвучки можуть додавати до створеного фільму озвучені ними відеофайли. Метод на рисунку 2.5.

```

async create(
  { quantity, language, movie_id, dubbing_studio_id }: IMovieFileCreate,
  files: any
) {
  const _files = await fileService.uploadFilesToGoogleCloudStorage(files);

  const fileUrl = Array.isArray(_files) ? _files[0].url : _files.url;
  const fileMessage = Array.isArray(_files)

```

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ІЗ	Арк.
		Левченко А.Ю.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        ? _files[0].message
        : _files.message;

    if (!fileUrl) return { message: fileMessage };

    const movieFile = await this.prisma.movie_file.create({
      data: {
        movie_id,
        quantity,
        language,
        dubbing_studio_id,
        file_id: Array.isArray(_files) ? _files[0].fileId : _files.fileId,
        file_url: fileUrl
      }
    });

    return { ...movieFile, message: fileMessage };
  }
}

```

Студії озвучки мають власні акаунти, за допомогою яких вони можуть завантажувати відеофайли з озвученням. Вони заповнюють інформацію про мову озвучення, якість відео та інші характеристики файлів. Після цього відеофайли стають доступними для користувачів вебсайту.

Після додавання озвучень користувачі можуть переглядати фільми, залишати коментарі та оцінювати їх. Інтерфейс перегляду фільму містить плеєр для відтворення відео, а також секції для коментарів та оцінок. Користувачі можуть також залишати власні відгуки.

Після виконання всіх маніпуляцій, у кабінетах адміністратора та студій озвучки відображаються відповідні дані. Адміністратор може переглядати список усіх фільмів, керувати користувачами. Студії озвучки можуть переглядати свої відеофайли в кабінетах.

Висновки до другого розділу.

У другому розділі проведено детальний аналіз вимог до вебсайту для перегляду фільмів та серіалів, розглянуто основні функціональні та нефункціональні вимоги, а також часові характеристики реакції системи на дії користувачів. Зокрема, висвітлено ключові компоненти системи, їх

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

структуру та взаємодію, що є критично важливим для забезпечення надійності, швидкості та безпеки платформи.

Основні висновки розділу включають:

- *Функціональні вимоги:* Вебсайт повинен забезпечувати зручний пошук і фільтрацію контенту за різними критеріями, стрімінговий перегляд відео без затримок, можливість коментування та оцінювання контенту, а також інтеграцію з соціальними мережами і оптимізацію для мобільних пристроїв
- *Нефункціональні вимоги:* Важливість швидкої реакції системи на дії користувачів, захисту даних, інтуїтивно зрозумілого інтерфейсу, підтримки різних пристроїв та браузерів, масштабованості та безперебійного оновлення контенту і функціоналу.
- *Класи та їх функції:* Визначено основні класи, такі як UserService, MovieService, ImageService, MovieFileService, DubbingStudioService, CommentService, GenreService, BannerService, кожен з яких відповідає за певну функціональність вебсайту.
- *Структура бази даних:* Описано структуру бази даних, яка складається з 11 таблиць, включаючи таблиці для збереження даних про банери, коментарі, студії озвучки, жанри фільмів, зображення, фільми, файли фільмів, реакції фільмів та користувачів.
- *Проектування та реалізація алгоритмів роботи вебсайту:* Проаналізовано алгоритми роботи вебсайту для різних типів користувачів, включаючи адміністраторів, модераторів, студії озвучки та звичайних користувачів. Визначено функціональні можливості кожної категорії користувачів, що забезпечує ефективну взаємодію з системою.

Ці висновки надають чітке розуміння структури, функціональності та вимог до вебсайту для перегляду фільмів та серіалів. Зібрані дані та виявлені вимоги є основою для подальшого проектування та реалізації вебсайту, який відповідатиме потребам користувачів та забезпечуватиме високоякісний і зручний доступ до відеоконтенту.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБСАЙТОМ

3.1. Порядок встановлення та налаштування

Встановлення та налаштування платформи вимагає розгортання всіх його компонентів та налаштування як кожного елементу окремо, так і загальної системи в цілому.

Для опису архітектури платформи було створено діаграму розгортання (рис. 3.1).

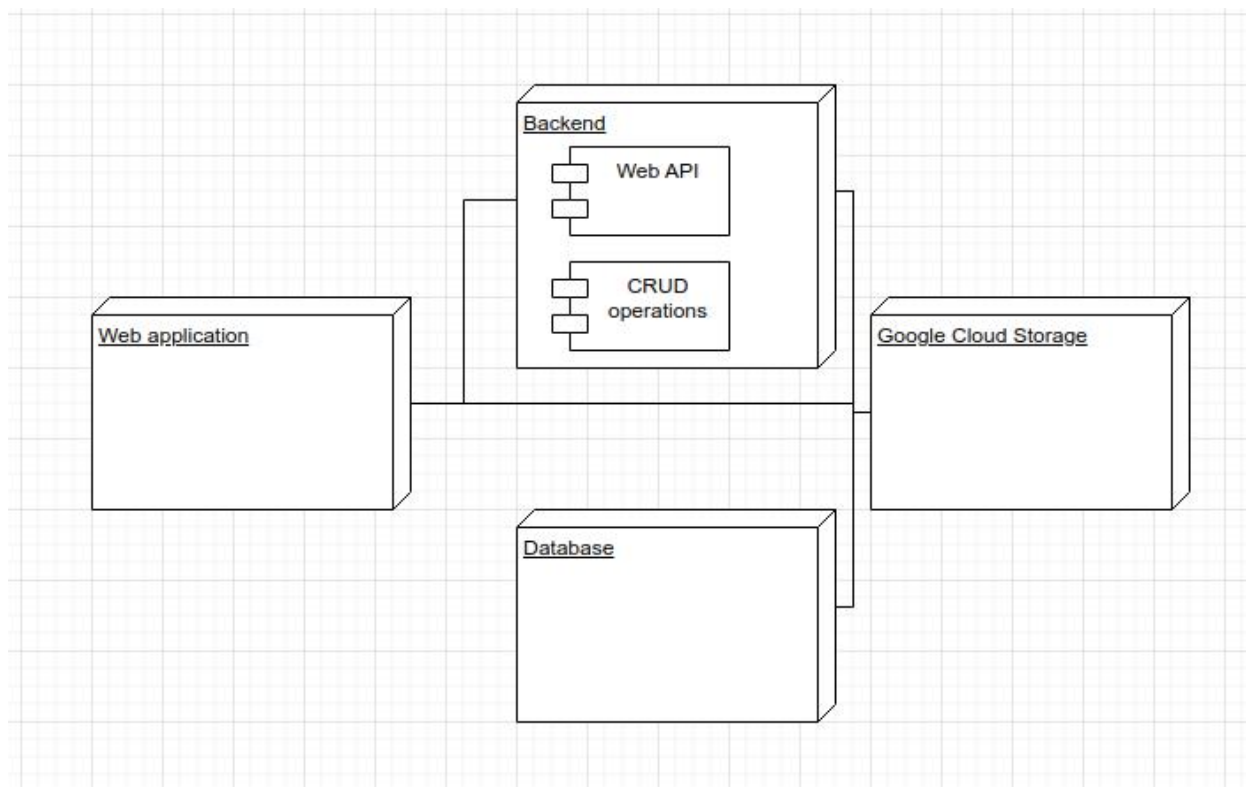


Рис. 3.1. Діаграма розгортання вебсайту

Опис процес розгортання.

WEB APP BACKEND – розгортання серверної частини вебсайту, яка забезпечує зберігання проміжних даних і проведення CRUD операцій з даними, пов'язаними з фільмами та серіалами. Це включає керування інформацією про користувачів, фільми, серіали, відгуки.

WEB APP FRONTEND – розгортання клієнтської частини вебсайту, яка об'єднує дані з серверної частини, а також відповідає за візуалізацію цих даних. Клієнтська частина забезпечує зручний інтерфейс для користувачів, дозволяючи їм переглядати фільми та серіали, залишати відгуки, оцінки.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

DATABASE – використання бази даних для зберігання структурованої інформації, пов'язаної з фільмами, серіалами та користувачами. База даних дозволяє ефективно зберігати, управляти та запитувати дані, необхідні для роботи вебсайту. Це включає інформацію про фільми та серіали (назва, опис, жанр, актори, режисери тощо), дані про користувачів (профілі).

GOOGLE CLOUD STORAGE – використання Google Cloud Storage для зберігання великих обсягів медіафайлів, таких як відео, постери та інші зображення, пов'язані з фільмами та серіалами. Google Cloud Storage забезпечує надійне, масштабоване та безпечне зберігання цих файлів, що дозволяє швидко і ефективно надавати контент користувачам вебсайту. Крім того, інтеграція з Google Cloud Storage полегшує керування файлами та їх доступність з різних пристроїв та локацій.

3.2 Інтерфейс та порядок роботи

Перейдемо до огляду інтерфейсу та робочого процесу вебсайту для перегляду фільмів і серіалів. Наведені різні скріншоти і короткі описи до кожного з них, підкреслюючи ключові особливості та функціональні можливості програмного продукту. Ці скріншоти демонструють елементи користувацького інтерфейсу, з якими користувачі взаємодіятимуть під час використання вебсайту.

Відкриття вебсайту: користувач відкриває вебсайт у своєму браузері. (рис. 3.2). При завантаженні головної сторінки користувач бачить інтерфейс, який дозволяє легко орієнтуватися і знаходити необхідний контент. На головній сторінці представлені, фільми.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

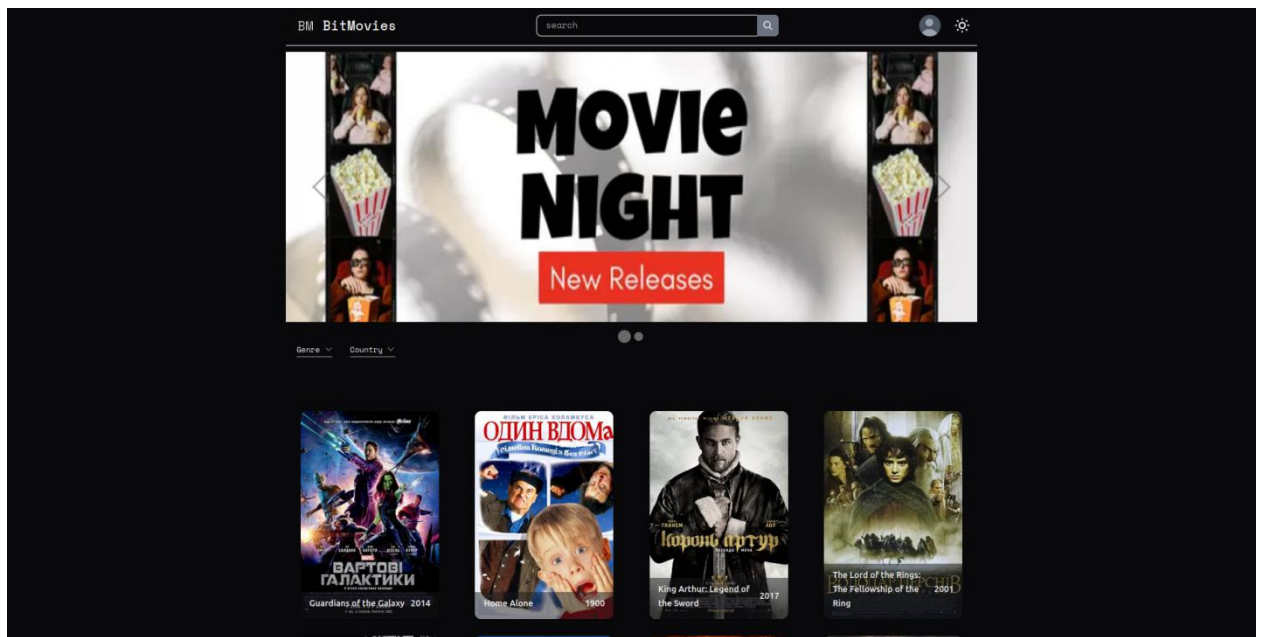


Рис. 3.2. Головна сторінка вебсайту

Реєстрація та вхід до системи: користувач має можливість зареєструвати новий обліковий запис або увійти в систему за допомогою вже існуючого облікового запису. (рис. 3.3). Процес реєстрації включає введення основної інформації, такої як електронна адреса, пароль. Після успішного входу користувач отримує можливості оцінювати контент, писати коментарі.

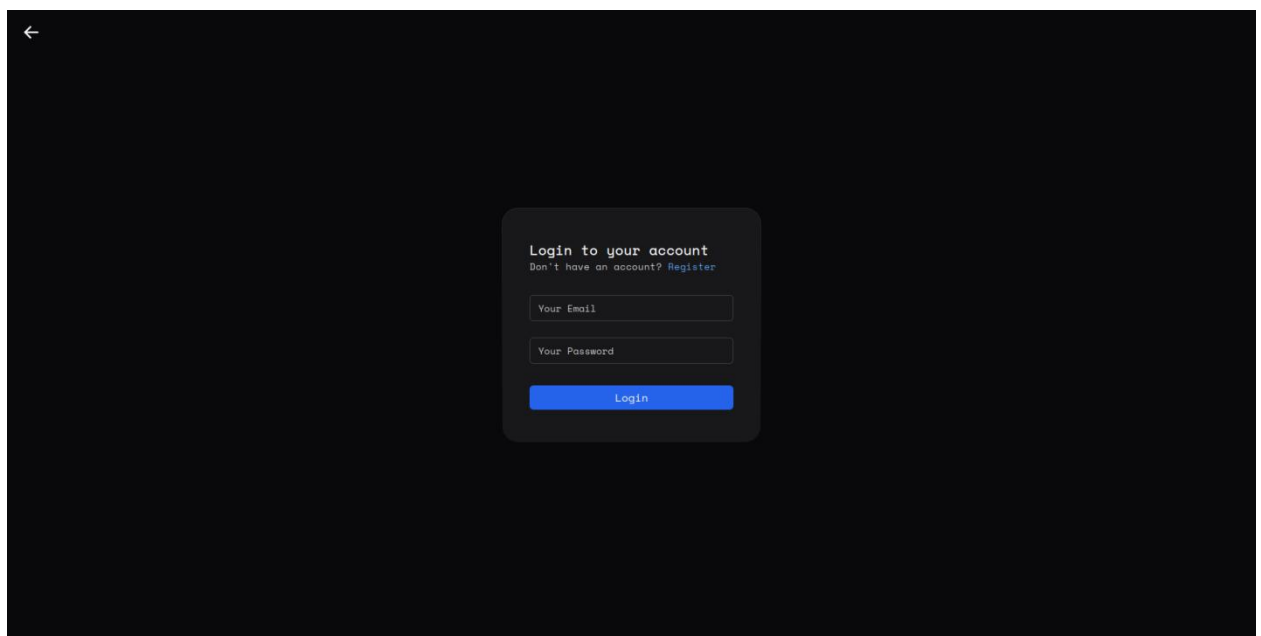


Рис. 3.3. Форма входу.

Перегляд каталогу: користувач може переглядати каталог фільмів та серіалів, використовуючи зручні фільтри та категорії. (рис. 3.4). Інтерфейс каталогу дозволяє швидко знаходити контент за жанрами, країнами.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

Користувач також може скористатися пошуковою функцією для швидкого знаходження конкретного фільму або серіалу.

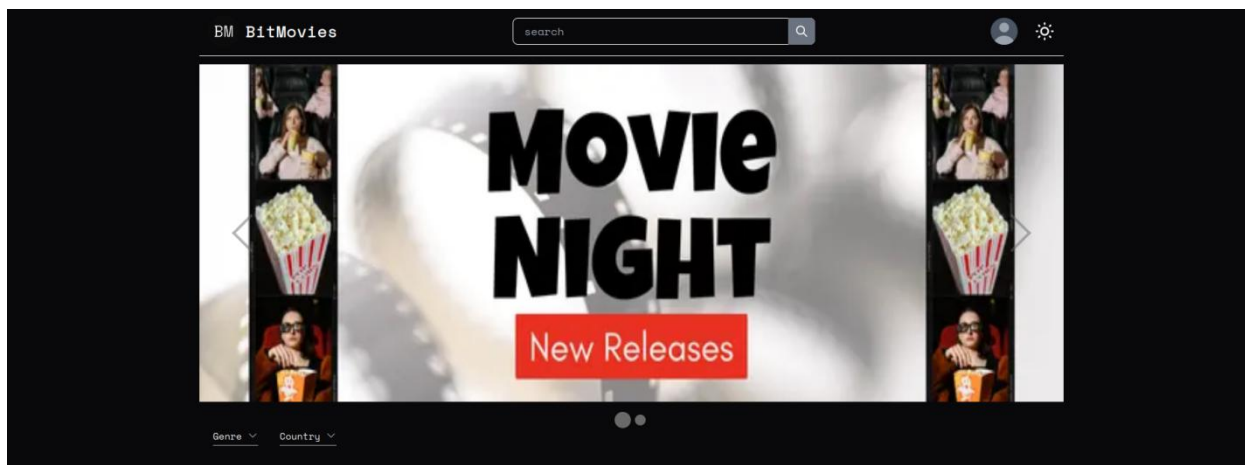


Рис. 3.4. Пошук і фільтр.

Перегляд детальної інформації про фільм: користувач може відкрити сторінку з детальною інформацією про обраний фільм . (рис. 3.5). На цій сторінці відображається картинка фільму, опис, країни, інформація про акторів та знімальну групу. Користувач може негайно розпочати перегляд, а після перегляду оцінити фільм і залишити коментар.

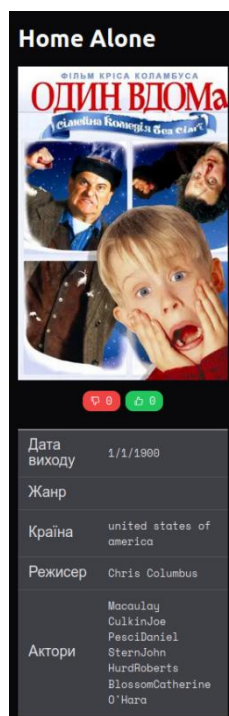


Рис. 3.5. Детальна інформація про фільм.

Перегляд відеоконтенту: користувач може розпочати перегляд обраного фільму або серіалу безпосередньо на вебсайті. (рис. 3.6).

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

Відеоплеєр підтримує функції, такі як перемотування, регулювання гучності, швидкість відтворення. Інтерфейс плеєра інтуїтивно зрозумілий і забезпечує комфортний перегляд контенту.

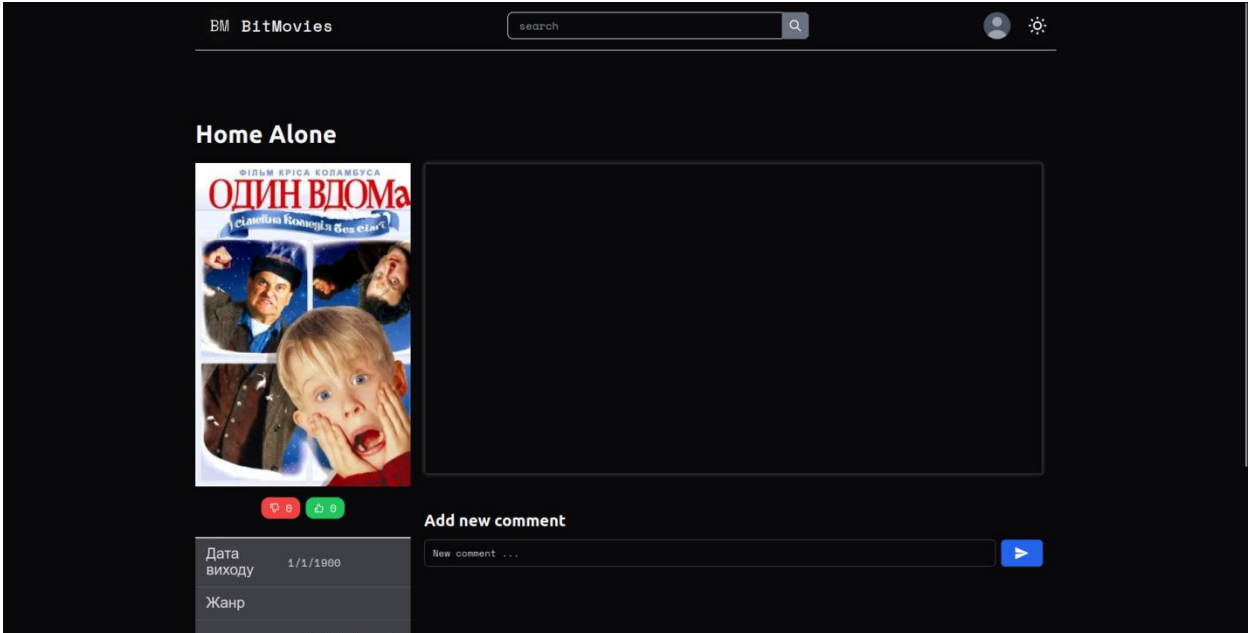


Рис. 3.6. Сторінка перегляду фільму.

Якщо користувач має права адміністратор, тоді при натисканні на іконку користувача буде з’являться меню де буде пункт “Admin”.

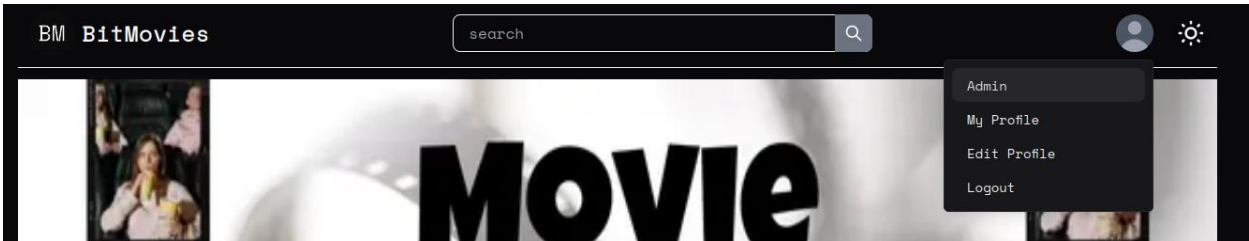


Рис. 3.7. Меню після натискання на іконку користувача.

Після його натискання користувача буде переведено на адмін панель. Користувач може використовувати зліва меню для навігації. Меню містить такі пункти “Banners”, “Dubbing studio”, “Genres”, “Movies”, “Users”, кожен з цих пунктів має свої підпункти, наприклад, “Create”, “List”. Ось так виглядає панель адміністратора на рисунку 3.8.

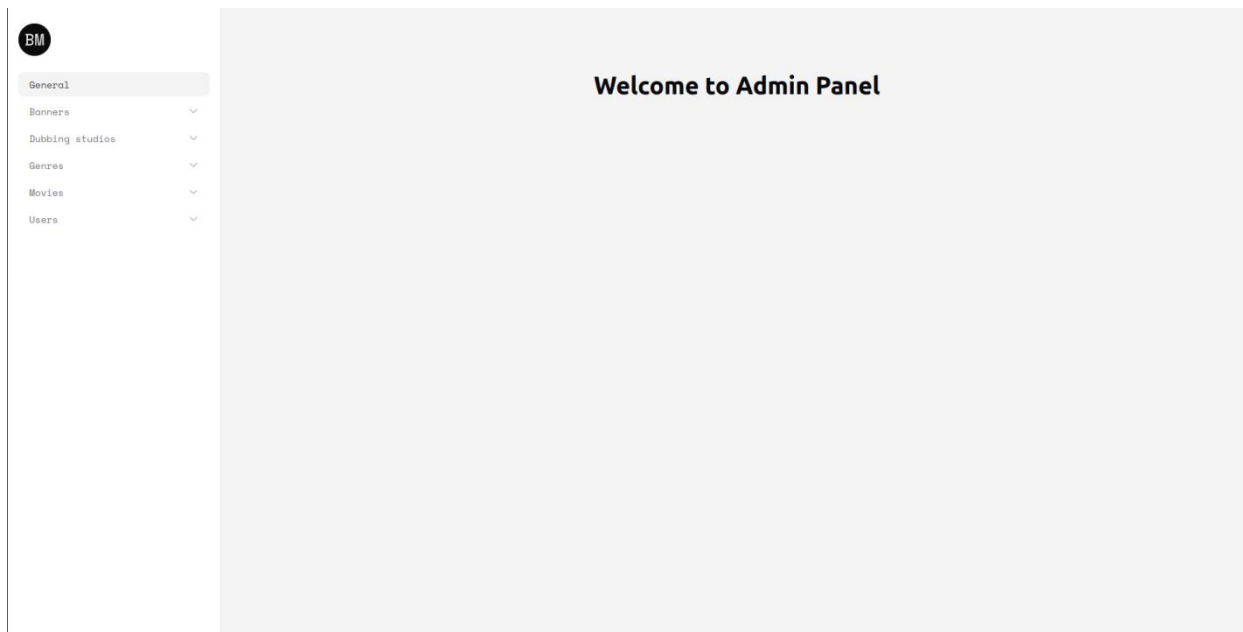


Рис. 3.8. Адмін панель.

Форма для додавання банерів. Адміністратор повинен ввести дані файл і можливо url по якому буде переадресовано користувача коли він натисне на цей банер. Форма для додавання банерів на рисунку 3.9.

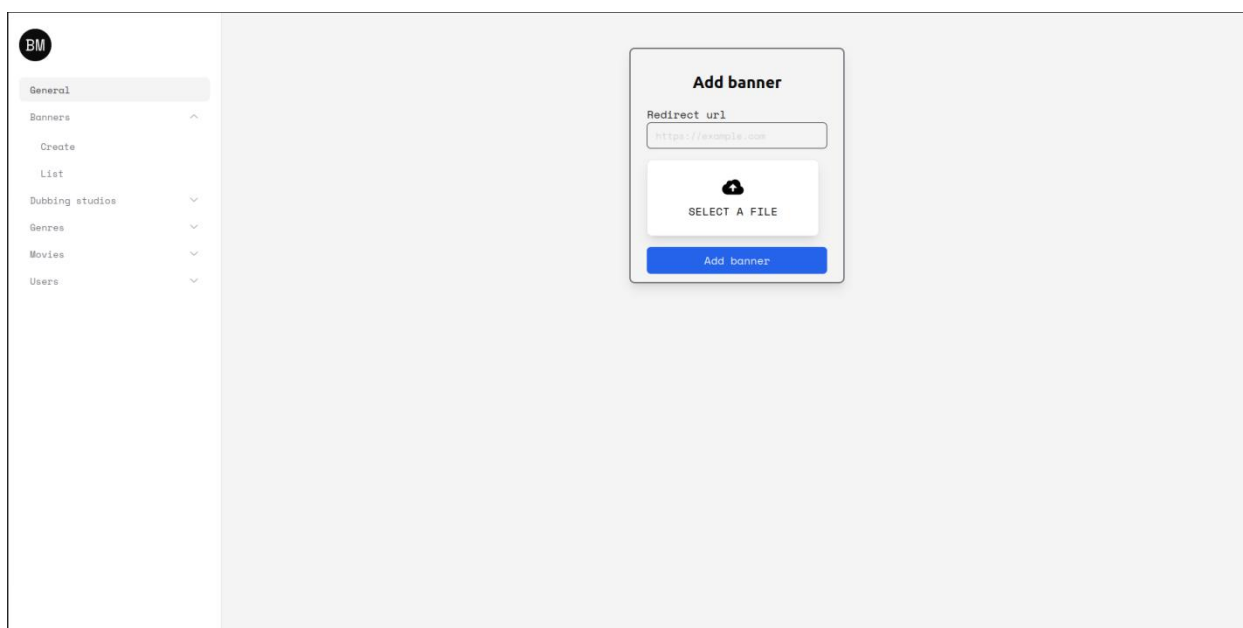


Рис. 3.9. Форма для додавання банерів.

Наступна форма це створення студії озвучки. Реєстрація студії озвучки буде відбуватись так: користувач надає студії озвучки адміністратору, а адміністратор заповнює дані, щоб створити студію озвучки. Це зроблено для того щоб кожний користувач не міг створити студію озвучки і завантажувати піратський контент. Форма створення студії озвучки на рисунку 3.10.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.10. Форма для створення студії озвучки.

Також адміністратор створює жанри фільмів. Форма створення жанру фільму на рисунку 3.11.

Рис. 3.11. Форма створення жанру фільму.

Наступна форма це форма створення фільму на рисунку 3.12. Перед цим етапом адміністратор повинен створити жанри фільму. Після створення фільму, студії озвучки можуть додавати до створеного фільму озвучені ними фільми, тобто, студії озвучки можуть додавати відеофайли до фільму. Дана форма буде наведена далі.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

Create movie

Movie name

Countries

Short description

Description

Release date

Age category

Genres

Producers

Actors

Image

Рис. 3.12. Форма для створення фільму.

На даному рисунку 3.13. Тут адміністратор може призначати користувачам права такі як: модератор, користувач.

Users

Email	Nickname	Role
admin@gmail.com	admin	admin
adm@gmail.com	adm	user
moderator@gmail.com	moderator	moderator
ds@gmail.com	ds	user
user@gmail.com	user	user

Рис. 3.13. Таблиця Users.

Далі буде опис про панель модератора. Панель модератора зображена на рисунку 3.14. Модератор буде використовувати стежити за коментарями, якщо коментар є недопустимим модератор може змінити йому стан відображення, тобто, коментар не буде відображатись в інтерфейсі, але в базі даних зберігатиметься.

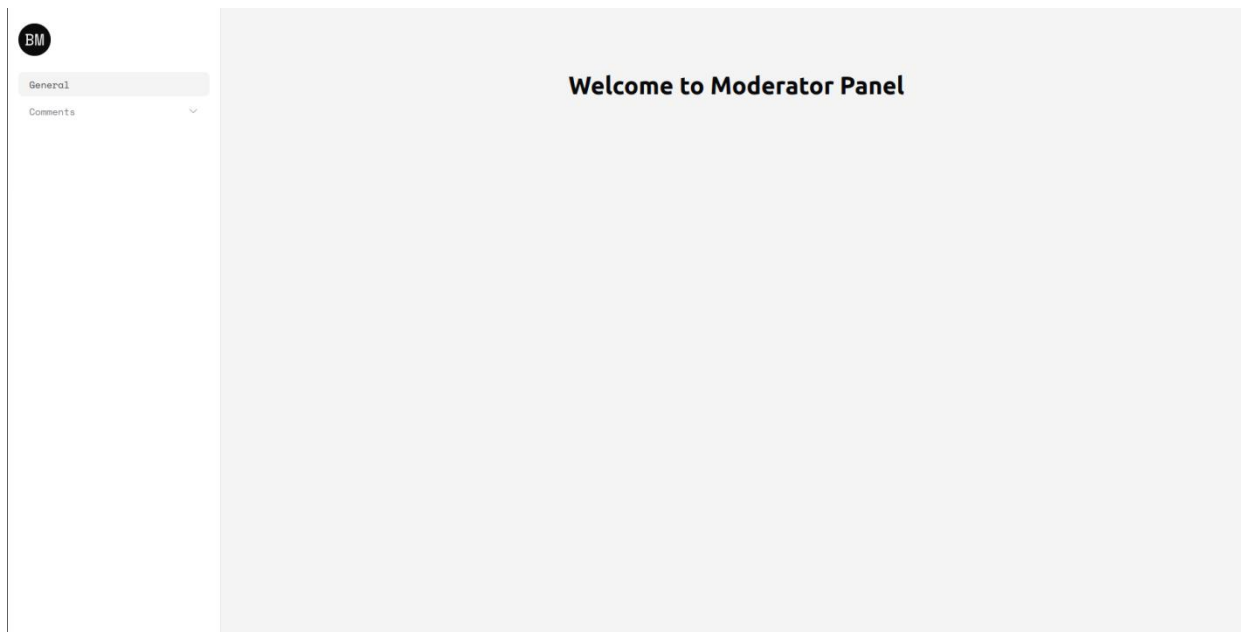


Рис. 3.14. Панель модератора.

Таблиця де модератор може змінювати відображення коментаря.

BM		Comments			
General					
Comments					
User	Comment	Creation date/time	Disabled		
user user@gmail.com	very interesting cartoon, I recommend it to everyone)))	6/3/2024 7:28:18 PM	Off	☐	On
user user@gmail.com	great movie :)	6/3/2024 7:27:31 PM	Off	☐	On

Рис. 3.15. Таблиця де відображаються коментарі.

3.3 Тестування роботи вебсайту

Тестування роботи вебсайту для перегляду фільмів і серіалів може включати декілька етапів, щоб перевірити різні аспекти його функціональності та продуктивності. Важливі аспекти, які потрібно врахувати під час тестування:

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

- *Функціональність основних компонентів:* тестування реєстрації та авторизації користувачів; можливість переглядати каталоги фільмів і серіалів; перегляд детальної інформації про контент; можливість відтворення відео.
- *Підтримка різних форматів відео:* перевірка можливості відтворення відео у різних форматах, таких як MP4, MKV; маніпуляція відео (пауза, перемотування, зміна якості).
- *Редагування профілю користувача:* тестування можливості зміни інформації профілю, додавання фільмів, залишення відгуків та оцінок.
- *Продуктивність та стабільність:* оцінка швидкості завантаження сторінок, відтворення відео, навантаження на сервер; перевірка оптимізації коду та зменшення затримок.
- *Інтерфейс користувача:* перевірка зручності навігації по сайту, відповідність дизайну сучасним стандартам, адаптивність до різних пристроїв (комп'ютери, планшети, смартфони).
- *Контроль якості відтворення:* перевірка якості зображення і звуку під час відтворення відео, підтримка високої роздільної здатності (HD); відсутність затримок і збоїв.

Проводячи тестування роботи вебсайту, ми можемо переконатися, що розроблюваний продукт відповідає необхідним функціональним можливостям і оптимально працює в різних умовах. Ці тести допомагають виявити та усунути будь-які потенційні проблеми або помилки, забезпечуючи стабільну роботу платформи для користувачів.

Висновки до третього розділу.

У третьому розділі було детально розглянуто процес встановлення та налаштування платформи для перегляду фільмів і серіалів, а також описано інтерфейс і порядок роботи вебсайту, завершуючи його тестуванням.

Основні висновки розділу включають:

- *Процес встановлення та налаштування:* Встановлення платформи включає розгортання серверної частини (WEB APP BACKEND) для

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

зберігання та обробки даних, клієнтської частини (WEB APP FRONTEND) для взаємодії з користувачами, бази даних (DATABASE) для зберігання структурованої інформації та Google Cloud Storage для зберігання великих обсягів медіафайлів. Кожен компонент виконує свою унікальну функцію, забезпечуючи надійну та ефективну роботу всієї системи.

- *Інтерфейс та порядок роботи:* Описано користувацький інтерфейс вебсайту, зокрема головну сторінку, процес реєстрації та входу в систему, перегляд каталогу фільмів і серіалів, перегляд детальної інформації про фільми, а також перегляд відеоконтенту. Вказано, як різні типи користувачів (адміністратори, модератори, звичайні користувачі) можуть взаємодіяти з вебсайтом. Наведені скріншоти і короткі описи до кожного з них допомагають зрозуміти ключові особливості та функціональні можливості платформи.
- *Тестування роботи вебсайту:* Проведення тестування було розділене на кілька етапів для перевірки різних аспектів функціональності та продуктивності вебсайту. Тестування включало перевірку функціональності основних компонентів, підтримку різних форматів відео, редагування профілю користувача, продуктивність та стабільність системи, зручність інтерфейсу користувача, а також контроль якості відтворення відео. Це дозволило виявити та усунути потенційні проблеми або помилки, забезпечуючи стабільну та якісну роботу платформи для кінцевих користувачів.

У цілому, третій розділ демонструє комплексний підхід до розгортання, налаштування, використання та тестування вебсайту для перегляду фільмів і серіалів, що є необхідним для створення надійного та зручного продукту. Ретельне виконання всіх описаних кроків сприяє досягненню високої якості роботи платформи, що відповідає потребам користувачів та забезпечує їм позитивний досвід використання.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано вебсайт для перегляду фільмів і серіалів, проведено аналіз предметної області та порівняно різні платформи, що надають подібні послуги. На основі цього аналізу були визначені основні вимоги до вебсайту для перегляду фільмів і серіалів. У процесі дослідження була розроблена архітектура, підходи та основні технології реалізації вебдодатку.

Проектування вебсайту включало аналіз варіантів використання програмного продукту та проектування структури, включаючи моделювання взаємодії між складовими частинами. Також було спроектовано базу даних для зберігання інформації про користувачів, фільми, а також для даних, що зберігаються на серверній частині додатку.

Описано алгоритм роботи з додатком та послідовні пункти варіантів його використання. Розглянуто основний функціонал, включаючи реєстрацію та авторизацію користувачів, перегляд каталогу фільмів і серіалів, перегляд детальної інформації про контент, а також можливість перегляду відео безпосередньо на вебсайті.

Також була надана детальна схема розгортання вебсайту для перегляду фільмів і серіалів, описано взаємодію структурних елементів системи та наведено опис інтерфейсу розробленого програмного продукту для різних типів користувачів. Також показано роботу пошуку контенту, що надає користувачам можливість швидко знаходити та переглядати улюблений контент.

Архітектура вебсайту побудована на базі сучасних технологій, що гарантують швидкодію та надійність сервісу. Для зберігання та обробки великих обсягів даних використовуються ефективні рішення, які забезпечують стабільну роботу вебдодатку.

Вебсайт має інтуїтивний інтерфейс, який дозволяє користувачам легко знаходити та переглядати фільми і серіали. Користувачі можуть, залишати

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

відгуки, оцінювати фільми, а також ділитися своїми рекомендаціями з іншими користувачами.

Розроблений вебсайт для перегляду фільмів і серіалів надає зручний та ефективний інструмент для пошуку та перегляду відеоконтенту, забезпечуючи приємний користувацький досвід та відповідаючи сучасним вимогам та тенденціям у сфері онлайн-розваг.

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Next.js [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs>
2. Next.js Server Component [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs/app/building-your-application/rendering/server-components>
3. Next.js Component Image [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs/app/api-reference/components/image>
4. next.config.js [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs/app/api-reference/next-config-js>
5. useSearchParams [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs/app/api-reference/functions/use-search-params>
6. Pages and Layouts [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs/app/building-your-application/routing/pages-and-layouts>
7. Express.js [Електронний ресурс] - Режим доступу до ресурсу:
8. Redux Toolkit [Електронний ресурс] - Режим доступу до ресурсу: <https://redux-toolkit.js.org/tutorials/typescript>
9. RTK Query [Електронний ресурс] - Режим доступу до ресурсу: <https://redux-toolkit.js.org/tutorials/rtk-query>
10. Google Cloud Storage:Node.js Clie [Електронний ресурс] - Режим доступу до ресурсу: <https://cloud.google.com/nodejs/docs/reference/storage/latest>
11. Upload file to Google Cloud Storage [Електронний ресурс] - Режим доступу до ресурсу: <https://www.bezkoder.com/google-cloud-storage-nodejs-upload-file/>
12. Upload file to Google cloud storage from Node.js server [Електронний ресурс] - Режим доступу до ресурсу: <https://dev.to/kamalhossain/upload-file-to-google-cloud-storage-from-nodejs-server-5cdg>
13. Next js with Axios Data Fetching [Електронний ресурс] - Режим доступу до ресурсу: <https://dev.to/syedmuhammadaliraza/next-js-with-axios-data-fetching-435a>

		Лакизюк Р.В.			ІІЗ.КР.Б - 121 - 24 - ІІЗ	Арк.
		Левченко А.Ю.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

14. Prisma ORM Docs [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.prisma.io/docs/getting-started>
15. Tailwind Css Docs [Електронний ресурс] - Режим доступу до ресурсу:
<https://tailwindcss.com/docs/installation>
16. PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.postgresql.org/>

		Лакизюк Р.В.			ІПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Лакизіюк Р.В.			ПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

Реалізація класу, що відповідає за завантаження файлів на google cloud storage.

Лістинг google-cloud-storage.service.ts:

```
import { Storage } from '@google-cloud/storage';
import { format } from 'util';
import { GOOGLE_CLOUD_STORAGE } from '../constants/index.js';
import { Id } from '../types/base.types.js';

const storage = new Storage({ keyFilename: 'google-cloud-key.json' });
const bucket = storage.bucket(GOOGLE_CLOUD_STORAGE.BUCKET);

export default new (class GoogleCloudStorageService {
  async upload(
    fileName: string,
    file: File
  ): Promise<{ message: string; url?: string }| {
    return new Promise((resolve, reject) => {
      // Create a new blob in the bucket and upload the file data.
      const blob = bucket.file(fileName);
      const blobStream = blob.createWriteStream({
        resumable: false
      });

      blobStream.on('error', err => {
        reject({ message: err.message });
      });

      blobStream.on('finish', async () => {
        // Create URL for directly file access via HTTP.
        const publicUrl = format(
          `https://storage.googleapis.com/ {bucket.name}/ {blob.name}`
        );

        console.log({ publicUrl });

        try {
          // Make the file public
          await bucket.file(fileName).makePublic();
        } catch {
          resolve({
```

		Лакизюк Р.В.			ІІЗ.КР.Б - 121 - 24 - ІІЗ	Арк.
		Левченко А.Ю.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        message: `Uploaded the file successfully:  {fileName}, but public access is
denied!`,
        url: publicUrl
    });
}

resolve({
    message: `Uploaded the file successfully: ` + fileName,
    url: publicUrl
});
});

blobStream.end(file);
});
}

async deleteFileById(fileId: IId) {
    const [files] = await bucket.GetFiles({ prefix: `  {fileId}` });
    if (files.length === 0) {
        return { message: `No files found with ID:  {fileId}` };
    }

    const deletePromises = files.map(file => file.delete());
    await Promise.all(deletePromises);

    return { message: `Successfully deleted files with ID:  {fileId}` };
}
}());

```

		Лакизиук Р.В.			ПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

Реалізація класу, що відповідає за роботу коментарів.

Лістинг comment.service.ts:

```
import { PrismaClient } from '@prisma/client';
import { TCommentCreate } from '../types/comment.types.js';
import ApiError from '../exceptions/api.error.js';

export default new (class CommentService {
  private prisma = new PrismaClient();

  async create({ movie_id, user_id, comment }: TCommentCreate) {
    const _comment = await this.prisma.comment.create({
      data: {
        movie_id: movie_id,
        user_id: user_id,
        comment: comment
      }
    });

    return _comment;
  }

  async getCommentsByMovieId(movieId: number) {
    const comments = await this.prisma.comment.findMany({
      where: {
        movie_id: movieId,
        is_disabled: false
      },
      include: {
        user: {
          select: {
            email: true,
            nickname: true,
            id: true,
            avatar: true,
            date_birth: true,
            register_date: true
          }
        }
      }
    });
  }
});
```

		Лакизюк Р.В.			ІІЗ.КР.Б - 121 - 24 - ІІЗ	Арк.
		Левченко А.Ю.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        orderBy: {
            creation_date: 'desc'
        }
    });

    if (!comments) throw ApiError.NotFoundError();

    return comments;
}

async getComments(page = 1, pageSize = 10) {
    const comments = this.prisma.comment.findMany({
        include: {
            user: true
        },
        orderBy: {
            creation_date: 'desc'
        },
        skip: (page - 1) * pageSize,
        take: pageSize
    });

    return comments;
}

async deleteById(commentId: number) {
    const comment = await this.prisma.comment.delete({
        where: {
            comment_id: commentId
        }
    });

    if (!comment) throw ApiError.NotFoundError();

    return comment;
}

async toggleCommentDisabled(commentId: number, isDisabled: boolean) {
    const comment = await this.prisma.comment.update({

```

		Лакизюк Р.В.			ІІЗ.КР.Б - 121 - 24 - ІІЗ	Арк.
		Левченко А.Ю.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

```
        data: { is_disabled: isDisabled },
        where: { comment_id: commentId }
    });

    return comment;
}
})();
```

		Лакизиук Р.В.			ПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

Реалізація класу, що відповідає за реакції фільму, тобто, лайки і дизлайки.

Лістинг movie-reaction.service.ts:

```
import { PrismaClient, ReactionType } from '@prisma/client';

export default new (class MovieReactionService {
  private prisma = new PrismaClient();

  async likeMovie(movieId: number, userId: number) {
    const reaction = await this.prisma.movie_reaction.upsert({
      where: { movie_id_user_id: { movie_id: movieId, user_id: userId } },
      update: { type: ReactionType.LIKE },
      create: { movie_id: movieId, user_id: userId, type: ReactionType.LIKE }
    });

    return reaction;
  }

  async dislikeMovie(movieId: number, userId: number) {
    const reaction = await this.prisma.movie_reaction.upsert({
      where: { movie_id_user_id: { movie_id: movieId, user_id: userId } },

      update: { type: ReactionType.DISLIKE },
      create: { movie_id: movieId, user_id: userId, type: ReactionType.DISLIKE }
    });

    return reaction;
  }

  async reactionStatusMovie(movieId: number, userId: number | undefined) {
    let reaction = null;

    if (typeof userId === 'number') {
      reaction = await this.prisma.movie_reaction.findFirst({
        where: { movie_id: movieId, user_id: userId }
      });
    }
  }
});
```

		Лакизюк Р.В.			ІІЗ.КР.Б - 121 - 24 - ІІЗ	Арк.
		Левченко А.Ю.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    const likeCount = await this.prisma.movie_reaction.count({
      where: { movie_id: movieId, type: ReactionType.LIKE }
    });

    const dislikeCount = await this.prisma.movie_reaction.count({
      where: { movie_id: movieId, type: ReactionType.DISLIKE }
    });

    return { reaction, likeCount, dislikeCount };
  }
}());

```

		Лакизіюк Р.В.			ПЗ.КР.Б - 121 - 24 - ПЗ	Арк.
		Левченко А.Ю.				63
Змн.	Арк.	№ докум.	Підпис	Дата		