

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова

Херсонський навчально-науковий інститут

Кафедра інформаційних технологій та фізико-математичних дисциплін

“Допущений до захисту”



Завідувач кафедри
д.т.н., доцент Гучек П.Й.

“24” грудня 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

на здобуття ступеня вищої освіти “магістр”

на тему: “Дослідження технологій та проектування інформаційної
системи онлайн кінотеатр”

Виконав: студент VI курсу, групи 6157
спеціальності

122 - “Комп’ютерні науки”, ОПП -

(шифр і назва спеціальності та освітньо-професійної програми)

“Інформаційні управляючі системи та технології”

Булатов О.С.

(прізвище та ініціали)

Керівник О.С. Булатов д.т.н., доцент, Гайдаєнко О.В.

(прізвище та ініціали)

Рецензент Михелєв І.Л.

(прізвище та ініціали)

м. Миколаїв-Херсон - 2024 р

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова
ХЕРСОНСЬКИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ

Кафедра інформаційних технологій та фізико-математичних дисциплінОсвітній ступень магістрГалузь знань 12 - "Інформаційні технології"

(шифр і назва)

Спеціальність 122 - "Комп'ютерні науки"

(шифр і назва)

Освітньо-професійна програма "Інформаційні управляючі системи та технології"**ЗАТВЕРДЖУЮ**

Гарант освітньої програми

Гучек П.Й."23" жовтня 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «магістр»

Студенту

Булатову Олександрові Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікацій роботи: "Дослідження технологій та проектування інформаційної системи онлайн кінотеатр"

Керівник роботи

Гайдаєнко Оксана Володимирівна, к.т.н., доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Тема затверджена розпорядженням по інституту від "17" жовтня 2024 року № 102. Термін подання студентом роботи 13.12.2024 р.

3. Вихідні дані до роботи:

4. Зміст та структура розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1 Титульний аркуш, завдання на магістерську роботу, анотація (українською, англійською), зміст, перелік умовних означень, символів, одиниць та термінів (при необхідності).

4.2 Вступ.

4.3 Дослідження технологій проектування інформаційних систем та аналіз їх особливостей.

4.4 Формування вимог до інформаційної системи, розробка її концепції і постановка задачі.

4.5 Розробка проектних рішень по інформаційній системі

4.6 Реалізація проекту інформаційної системи (технічна документація по проекту).

4.7 Розділ з економіки

4.8 Розділ з охорони праці та безпеки в надзвичайних ситуаціях (охорони оточуючого середовища).

4.9 Висновки.

4.10 Список використаної літератури.

4.11 Додатки (технічне завдання, загальний опис ІС, інструкція користувача, текст програми).

5. Перелік презентаційних матеріалів

5.1 Схема організаційної структури підприємства.

5.2 Архітектура автоматизованої ІС.

5.3 Організаційно-функціональна структура автоматизованої ІС.

5.4 Інформаційна модель програмного забезпечення автоматизованої ІС.

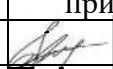
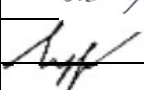
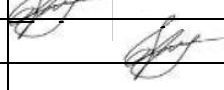
5.5 Модель переходів станів програмного забезпечення автоматизованої ІС.

5.6 Модель технологічного забезпечення автоматизованої ІС.

5.7 Інтерфейс програми

5.8

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4.7	Ломоносов Дмитро Анатолійович		
4.8	Щедролосєв Олександр Вікторович		

7. Дата видачі завдання 28 жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1.	Вивчення і аналіз предметної галузі	28.10.2024	
2.	Розробка концепції ІС	30.10.2024	
3.	Розробка проектних рішень по ІС	30.10.2024	
4.	Реалізація проекту, розробка робочої документації	02.11.2024	
5.	Вирішення питань з економіки	06.11.2024	
6.	Вирішення питань охорони праці та безпеки в надзвичайних ситуаціях (охорони оточуючого середовища)	12.11.2024	
7.	Виконання графічної частини роботи	23.11.2024	
8.	Оформлення пояснювальної записки	06.12.2024	
9.	Подання роботи на попередній захист	13.12.2024	
10.	Подання роботи рецензенту	20.12.2024	
11.	Подання роботи на державний захист	27.12.2024	

Студент

Керівник роботи


Підпис

Булатов О.С.

Прізвище та ініціали

Гайдаєнко О.В.

Прізвище та ініціали

Анотація

Кваліфікаційна робота присвячена дослідженню стримінгових сервісів і розробці веб-додатку онлайн-кінотеатру, який орієнтований на надання зручного пошуку та перегляду відеоконтенту в Інтернеті.

Вибрана тема є актуальною у зв'язку з популярністю технології передачі даних та кіно як форми проведення дозвілля.

Метою роботи є підвищення ефективності роботи веб-додатку онлайн-кінотеатру. Який буде надавати користувачам зручний доступ до перегляду медіаконтенту в Інтернеті, включаючи фільми, серіали та мультфільми.

В процесі розробки було проаналізовано сучасні засоби веб-розробки, технологію стримінгу та історію кінематографа. Також було проаналізовано статистичні дані щодо онлайн-кінотеатрів на сьогодні, в результаті чого була розроблена інформаційна система, яка надає можливість:

- знайти потрібну картину в максимально доступній якості;
- ознайомитись з інформацією про вибрану картину;
- переглянути вибрану картину;
- залишити оцінку та відгук після перегляду, а також ознайомитись з уже наявними відгуками інших глядачів.

Робота викладена українською мовою і виконана на 103 аркушах, містить 43 рисунків, 14 таблиць, 3 додатків. Список використаних джерел містить 18 найменувань.

Ключові слова: стримінг, медіаконтент, відеоконтент, онлайн-кінотеатр.

Abstract

The qualification work is devoted to the study of streaming services and the development of an online cinema web application, which is focused on providing convenient search and viewing of video content on the Internet.

The chosen topic is relevant due to the popularity of data transmission technology and cinema as a form of leisure.

The purpose of the work is to increase the efficiency of the online cinema web application. Which will provide users with convenient access to viewing media content on the Internet, including films, TV series and cartoons.

During the development process, modern web development tools, streaming technology and the history of cinema were analyzed. Statistical data on online cinemas today were also analyzed, as a result of which an information system was developed that provides the opportunity to:

- find a particular picture in the highest available quality;
- get acquainted with information about the selected picture;
- watch the selected title;
- rate and leave review after viewing, as well as read the existing reviews of other viewers.

The work is presented in Ukrainian and performed on 103 items, contains 43 figures, 14 tables, 3 appendices. The list of used sources contains 18 names.

Keywords: streaming, media content, video content, online cinema.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-КІНОТЕАТР. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1 Загальні відомості	10
1.2 Дослідження веб-додатків.....	12
1.3 Архітектура веб-сайту, як інформаційної системи	14
1.3.1 Загальні відомості щодо архітектури.....	14
1.3.2 Основні компоненти мови	15
1.3.3 Популярні frontend технології. Бібліотеки та фреймворки	16
1.4 Дослідження кінематографу, як форми розваги	19
1.4.1 Ключові віхи в історії кіно.....	19
1.4.2 Різниця між цифровою та аналоговою зйомкою	23
1.5 Аналіз стрімінгових сервісів.....	27
1.5.1 Найпопулярніші представники сфери	27
1.5.2 Аналіз готових рішень.....	29
1.6 Вимоги до інформаційної системи.....	31
1.7 Висновки до розділу 1	34
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН КІНОТЕАТР.....	35
2.1 Побудова моделі варіантів використання	35
2.2 Рішення по інформаційному забезпеченню	44
2.2.1 Розробка концептуальної моделі даних	45
2.2.2 Розробка логічної модель даних.....	45
2.2.3 Розробка фізичної модель даних	46
2.3 Рішення з математичного забезпечення	50
2.4 Рішення з програмного забезпечення	52
2.5 Рішення з технічного забезпечення	55
2.6 Висновки до розділу 2	56

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН КІНОТЕАТР	57
3.1 Головна сторінка	57
3.2 Сторінка картини	61
3.3 Список найкращих картин	65
3.4 Пошук за категоріями	67
3.5 Сторінка учасника	69
3.6 Діаграма розгортання системи	70
3.7 Висновки до розділу 3	71
РОЗДІЛ 4. РОЗРАХУНОК І ОЦІНКА ЕКОНОМІЧНОГО ЕФЕКТУ ВІД РОЗРОБКИ	73
4.1 Розрахунок витрат на створення програми	73
4.2 Розрахунок економічної ефективності	74
4.3 Висновки до розділу 4	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	77
ДОДАТОК А — Інструкція користувача	79
ДОДАТОК Б — Лістинг програми	90
ДОДАТОК В — Технічне завдання	99

ВСТУП

Темою кваліфікаційної роботи дослідження технологій та проектування інформаційної системи онлайн-кінотеатр. Досягнення поставленої мети починається з проведення дослідження, потім розробляється концепт продукту, і в результаті відбувається процес розробки. Метою роботи є підвищення ефективності роботи веб-додатку онлайн-кінотеатру. Який буде надавати користувачам зручний доступ до перегляду медіаконтенту в Інтернеті, включаючи фільми, серіали та мультфільми. Додаток має забезпечити простоту навігації, інтуїтивно зрозумілий інтерфейс та функціональність, що дозволяє користувачам легко знаходити відеоконтент і отримувати персоналізовані рекомендації на основі їхніх уподобань.

У першому розділі проведено аналіз предметної галузі, що включає:

- дослідження стрімінгових сервісів як таких;
- дослідження кіно, включаючи його історію та процес зйомки;
- розробка концепції інформаційної системи;
- розробка вимог до системи.

В другому розділі розроблено технічний проект інформаційної системи. Наведено загальносистемні рішення щодо використання веб-додатку. В інформаційному забезпеченні представлені концептуальна, логічна та фізична моделі бази даних. У математичному забезпеченні представлено блок-схему алгоритму роботи проекту. Описано засоби та технології розробки проекту, а також характеристики пристроїв користувача та адміністратора для використання інформаційної системи.

У третьому розділі представлено робочу документацію інформаційної системи, а також її реалізацію. Кожну розглянуту сторінку веб-додатку або окремий елемент підкріплено зображеннями для кращого розуміння документації.

У четвертому розділі наведено техніко-економічне обґрунтування розробки інформаційної системи.

У п'ятому розділі представлені правила охорони праці під час роботи за комп'ютером.

У шостому розділі представлені правила охорони навколишнього середовища.

Актуальність та новизна проекту полягають як у зручності пошуку відеоконтенту в Інтернеті, так і в навігації по сайту, що стало можливим завдяки інтуїтивно зрозумілому та зручному дизайну. Оскільки програма використовує адаптивний дизайн, перегляд фільмів та серіалів не обмежений певним місцем, що є однією з ключових функцій проекту. Легкий та зручний пошук, а також добірки, створювані додатком на основі дій користувача, роблять його досвід більш приємним та унікальним.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-КІНОТЕАТР. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальні відомості

Понад тридцять років відколи Інтернет став широкодоступним, люди активно його використовують, і з кожним роком кількість підключених до глобальної мережі пристроїв приватного користування незмінно зростає. Хто б міг подумати, що Інтернет сьогодні стане одним із найважливіших інструментів у світі? Використання Інтернету практично безмежне: від надання людям можливості залишатися на зв'язку з сім'єю та друзями до надання доступу до освітніх ресурсів та можливостей для розваг [1]. Хоча на початку 1990-х років Інтернет все ще був інструментом вузького кола, в тому числі вчених і військових, незабаром він став доступнішим ширшій аудиторії. За даними Statista [2], на сьогоднішній день до мережі підключено понад 18 мільярдів пристроїв, і, згідно з прогнозами, до 2030 року це число може перевищити 32 мільярди. Сьогодні глобальна мережа є невід'ємною частиною життя людей, надаючи різні можливості для свого використання.

На рисунку 1.1 зображено діаграму зміни кількості підключених пристроїв до Інтернету.

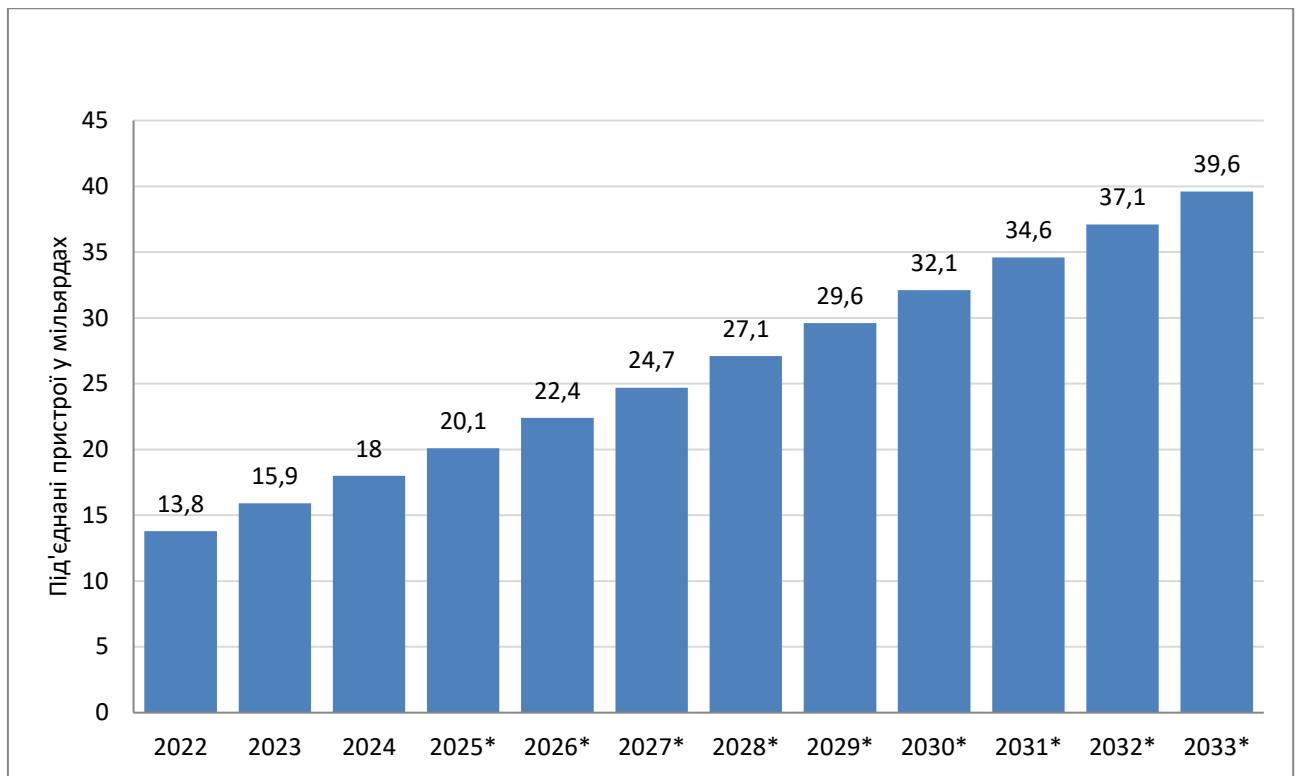


Рисунок 1.1 — Діаграма кількості підключених пристроїв до Інтернету

Одним із найзначніших проявів Інтернету, що й зробило його популярним серед широких мас, є онлайн-платформи для спілкування та обміну медіа-контентом. Позиціонуючи спочатку себе як майданчики для обміну повідомленнями, такі ресурси дуже швидко почали розвиватися та еволюціонувати у складніші послуги з великими можливостями. Спочатку, коли швидкість Інтернету становила лише кілька кілобайт, все обмежувалося лише текстовим спілкуванням та зображеннями низького дозволу, проте з розвитком технологій усе змінилося. Швидкість з'єднань зростала, сервіси набули все більшої популярності, а на сайтах вже можна було послухати аудіо запис або завантажити відео. Згодом платформи для спілкування стали провісниками одного з найвагоміших досягнень медіа — стрімінгових сервісів, які кардинально змінили спосіб споживання контенту.

Початком нової ери в галузі передачі медіа можна вважати появу компанії RealNetworks в середині 1994 року, яка 5 вересня 1995 успішно провела одну з перших аудіо трансляцій по мережі Інтернет [3]. Будучи

піонером у цьому починанні, компанія дала старт майбутньому інтернет-мовленню, а доступ до аудіо і незабаром до відео мовлення відкрив нові способи споживання контенту. Згодом стрімінгові послуги стали невід'ємною частиною сучасного Інтернету.

Важливе місце серед контенту, що транслюється, займають фільми та серіали. Кіно завжди було популярним видом проведення дозвілля, а коли в 1997 році, завдяки розвитку технологій та RealNetworks, стало можливим транслювати через Інтернет відео, цей вид розваг набув нового ажіотажу. Перегляд фільмів та серіалів перестав бути прив'язаний до конкретного місця та часу. Більше не потрібно було стояти в черзі до кінотеатру чи поспішати додому до нового випуску шоу — все стало доступно будь-якої миті та в будь-якому місці. Це дозволило глядачеві більше не бути залежним від традиційних графіків відвідування та мовлення та надало нові можливості для вибору контенту.

Таким чином, Інтернет змінив не лише спосіб спілкування, а й індустрію розваг. Стрімінгові послуги, що з'явилися на основі цих технологій, відіграли ключову роль у створенні нової ери споживання медіа-контенту. Сьогодні такі сервіси як Netflix, YouTube та інші вже давно є невід'ємною частиною онлайн-культури.

1.2 Дослідження веб-додатків

Веб-застосунки — це сучасне програмне забезпечення, яке надають деякі інформаційні послуги в Інтернеті через браузер. Основною відмінністю веб-додатка від звичайного настільного є спілкування між вузлами в глобальній та локальній мережі за допомогою HTTP(S) — протокол передачі гіпертексту. За невеликим винятком веб-програми використовують архітектуру клієнт-сервер, в якій клієнт (браузер користувача) взаємодіє з сервером, на якому розміщено програму, і набагато рідше використовується архітектура клієнт-клієнт.

Веб-додатки сильно різняться за складністю, типом та функціональністю. Вони можуть бути простими та одноцільовими, наприклад, калькулятор або список справ, або складними та багатограними, наприклад, інструменти онлайн-співробітництва або платформи електронної комерції. Типові приклади включають Google Docs, Gmail, Trello [4].

Оскільки існує безліч визначень для того чи іншого інтернет-ресурсу, добре дати визначення для поширених термінів, які так чи інакше ставляться до того самого «сайту».

Веб-сторінка — простий документ, який відображається на екрані комп'ютера за допомогою браузера. Такий документ написано мовою. Веб-сторінка може містити безліч різних матеріалів, таких як:

- Стильова інформація — контроль сторінки по сприйняттю та відчуттю;
- Скрипти — невеликі програми, які роблять сторінку динамічнішою та зручнішою у використанні для користувачів;
- Медіа — зображення, музика та відео.

Веб-сайт — це колекція сторінок, пов'язаних між собою будь-якими способами, доступними під єдиним доменним ім'ям. Кожна сторінка сайту містить прямі посилання, що дозволяє користувачеві швидко переходити від однієї сторінки веб-сайту до іншої.

Веб-сервер — це комп'ютер, що надає в мережу один або багато веб-сайтів (хостинг). Поняття «хостинг» означає, що всі сторінки та прикріплені до них файли містяться на цьому комп'ютері. Тобто. Веб-сервер буде надсилати будь-яку сторінку з сайту на запит будь-якого користувача, що і буде хостингом для браузера користувача [5].

1.3 Архітектура веб-сайту, як інформаційної системи

1.3.1 Загальні відомості щодо архітектури

Якщо розглядати будь-яку інформаційну систему, незалежно від її призначення, рівня складності та інших факторів, то в більшості випадків вона складатиметься із двох основних частин. Перша частина — це внутрішня реалізація, яка прихована від кінцевого користувача та іноді від інших програм і називається *backend*. Друга частина — це інтерфейс, за допомогою якого користувач взаємодіє з системою, яка називається *frontend*. У контексті web технологій для даних термінів можна надати такі визначення:

1) *Back-end* або «серверна частина» — це, по суті, те, як працює, оновлюється та змінюється сайт. Це стосується всього, що користувач не може бачити в браузері, наприклад, бази даних, сервери, що виконують логіку сайту, керують контентом або піклуються про безпеку даних і самого сервісу [6]. *Back-end*-розробка потрібні для створення динамічного-змінного сайту, як, наприклад, Facebook або Google Maps, де потрібно враховувати внесені зміни та надавати актуальну інформацію. Як мови програмування для написання можна використовувати різні, але в залежності від складності системи, її призначення до популярних можна віднести наступні PHP, Python, NodeJS, Java, C/C++. Така різноманітність мов визначається тим, що неважливо як, головне, щоб програма, що знаходиться на сервері, змогла отримати і відновити дані.

2) *Front-end* або розробка на стороні клієнта — це область розробки, яка включає створення та реалізацію візуальних та інтерактивних елементів веб-сайтів і веб-додатків, з якими користувачі безпосередньо взаємодіють. Він включає використання таких технологій, як HTML, CSS і JavaScript, для створення компонентів інтерфейсу користувача. Розробка зовнішнього інтерфейсу відіграє вирішальну роль у формуванні загального досвіду

користувача веб-сайту або веб-додатку, оскільки вона визначає, як користувачі взаємодіють з цифровим продуктом і сприймають його [7].

1.3.2 Основні компоненти мови

HTML (HyperText Markup Language) — це код, який використовується для розмітки та структурування веб-сторінки та її вмісту. HTML складається з ряду елементів або тегів, які використовуються для включення або обгортання різних частин контенту, щоб додати чи стилізувати належним чином. Кожен елемент розмітки сторінки поміщається в граф — DOM дерево (Document Object Model) у вигляді вузла, і служить для правильного створення веб-сторінки та її відображення [8].

CSS (Cascading Style Sheet) — це мова таблиць стилів, що використовується для форматування вмісту веб-сторінок HTML. Таблиці стилів CSS можуть визначати зовнішній вигляд і форматування тексту, таблиць та інших елементів окремо від контенту. Стили допомагають веб-розробникам створити унікальний вигляд всього веб-сайту [9].

JavaScript — це легковагова, кросплатформна мова програмування, яка виконується в браузері. Мова не є багатопоточною, але натомість має асинхронність і функції зворотного виклику для простоти використання. Також він відомий як мова сценаріїв для веб-сторінок, але при цьому його також використовують багато небраузерних середовищ [10].

JavaScript можна використовувати як для розробки на клієнтській стороні, так і для розробки на стороні сервера. У випадку розробки на стороні клієнта мові надаються об'єкти для маніпулювання з DOM-деревом, що включає візуальний та структурний дизайн елементів, API (Application Programming Interface) для роботи з самим браузером, а з недавнього часу і інструменти для безпосередньої роботи з відеокартою. У разі застосування мови на стороні сервера мається на увазі його синтаксис та особливості, які вже використовуються для зовсім іншого інструменту — Node.js, який включає

взаємодію з базами даних, маніпулювання з файлами, генерацію відповідей та інше.

Одним з розширень мови, або його надбудовою, є TypeScript. Ця мова надає статичну типізацію даних для написання чистішого коду, який простіше читати та підтримувати іншим фахівцям. Так само мова дозволяє швидше знаходити помилки типів, через що швидкість розробки значно зростає. Хоча ця мова не підтримується для виконання в браузері, проте компілятор мови перетворює її на чистий JavaScript, який зрештою і використовується.

1.3.3 Популярні frontend технології. Бібліотеки та фреймворки

ІТ індустрія завжди була великим гравцем на ринку і, залежно від області послуг, це завжди різний досвід і стиль роботи. Оскільки web сфера на тлі інших є однією з найпростіших для освоєння, вона стала однією з найбільш конкурентоспроможних і популярних областей у світі інформаційних технологій. Сучасний підхід до створення веб-додатків переважно передбачає короткострокове створення продукту, з мінімальними витратами для бізнесу. У прагненні спростити створення продукту було розроблено безліч інструментів, які є невід'ємною частиною процесу розробки. Одним з таких інструментів є фреймворки та бібліотеки націлені на створення динамічних та інтерактивних програм з мінімальними зусиллями.

React

React або ReactJS, розроблений у Facebook, є бібліотекою, призначеною для полегшення створення інтерактивних, що зберігають стан і компонентів користувацького інтерфейсу, що перевикористовуються. Бібліотека найкраще підходить для рендерингу складних інтерфейсів користувача з високою продуктивністю. Основний принцип бібліотеки — це концепція віртуального DOM, що створюється та зберігається у вигляді кеша, а також є абстрактною копією оригінального DOM, яка змінюється швидше за реальну

структуру. React виконує найменшу кількість можливих маніпуляцій із DOM, щоб підтримувати компоненти у актуальному стані [11].

Також бібліотека хоч і не була першою, але зробила компонентний підхід набагато популярнішим. Такий спосіб означає те, що засобами, що надаються, можна з базових HTML структур створити власні більш складні, а надалі використовувати їх, як звичайний елемент, що є ще одним способом спрощення розробки. Важливим є і розширення мови JavaScript — JSX, який дозволяє в рамках react додатку використовувати HTML елементи безпосередньо в коді.

Як приклади використання можна навести такі сервіси:

- Facebook,
- Instagram,
- Netflix.

Vue.js

Фреймворк для побудови інтерфейсів користувача Vue.js, розроблений колишнім співробітником Google Еваном Ю, є прогресивним продуктом, що володіє легковагістю, гнучкістю і інтуїтивно зрозумілою структурою, що дозволяє легко інтегруватися в вже існуючі проекти. Однією з унікальних технологій є система реактивності, яка дозволяє автоматично відстежувати залежності між компонентами та даними, миттєво застосовуючи зміни, роблячи це непомітно для користувача [12].

Таким чином, Vue є гнучким та адаптивним інструментом розробки, який можна застосовувати різними способами залежно від варіанта використання [13]:

- Покращення статичного HTML без етапу збирання, що дозволяє використовувати інструменти фреймворку для нескладних архітектурних рішень, не прив'язуючись до збирачів, а одразу підключивши скрипт.
- Використання веб-компонентів на будь-яку сторінку, що дозволяє інкапсулювати створений елемент і використовувати його повторно для інших елементів web програми.

- Односторінкова програма (SPA): Фреймворк підходить для веб-додатків, де для оновлення вмісту не обов'язково оновлювати саму сторінку.

- Fullstack/Рендеринг на стороні сервера (SSR): Vue надає набір інструментів, що дозволяють інтерпретувати vue код у реальному часі на стороні сервера та віддавати клієнту вже готову сторінку зі звичною структурою та логікою.

Як приклади використання можна навести такі сервіси:

- Adobe Portfolio,
- Upwork,
- GitLab.

Angular

Заснований на TypeScript і розроблений компанією Google у 2016 році, Angular швидко зайняв нішу і став конкурентоспроможним frontend фреймворком. На рівні з аналогами інструмент також має компонентний підхід до створення додатків, але головною його особливістю є функція двосторонньої прив'язки даних. Завдяки цьому розробник може бути впевнений у синхронізації змінної — моделі та елемента на сторінці — подання у реальному часі в Angular. Таким чином, коли в моделі відбувається якась зміна, вона миттєво відбивається на уявленні, і навпаки. Тому всі зміни у поданні відобразяться в моделі і навпаки при використанні Angular.

Як приклади використання можна навести такі сервіси:

- Xbox,
- Blender,
- Autodesk.

jQuery

Запущений у 2006 році і тривалий час, будучи однією з найпопулярніших бібліотек, jQuery досі залишається актуальним для багатьох веб-додатків. На відміну від сучасних фреймворків, jQuery не вимагає для використання сторонніх інструментів, як препроцесор або компілятор, а поширюється у

вигляді JavaScript коду, що підключається. Незважаючи на те, що jQuery має кардинальні відмінності від багатьох сучасних frontend бібліотек, надаючи зовсім іншу функціональність, він є зручним у мінімізуванні коду. Завдяки своїй легкості та простоті використання, jQuery значно прискорює виконання таких завдань, як маніпуляція з DOM, робота з CSS, обробка подій та створення анімацій.

Як приклади використання можна навести такі сервіси:

- WordPress,
- Twitter,
- Kickstarter.

1.4 Дослідження кінематографу, як форми розваги

З'явившись приблизно двісті років тому, мистецтво зображень, що рухаються, завоювало розуми безлічі винахідників, художників і просто ентузіастів. Подолавши безліч трансформацій та змін, сьогодні нам відомо колись небувале диво як кіно, яке посідає почесне місце серед найвпливовіших видів мистецтва. Від камер обскуру, до чарівних ліхтарів, від фотонегативів до сирих пікселів кінематограф пройшов тернистий шлях і розвинувся в те, де сьогодні використовуються найінноваційніші технології.

Про кіно можна дискутувати і просто говорити дуже довго, торкаючись багатьох аспектів. Втім, ніхто не буде сперечатися, що кінематограф має вагомий вплив на кожного незалежно від віку, характеру або місця в соціумі.

1.4.1 Ключові віхи в історії кіно

- 1887 рік. Перша кінокамера

Винахідник Луї Лепренс розробив однооб'єктивну кінокамеру, яку використовував для створення послідовності зображень, що рухаються. У 1888

році зняв кілька епізодів, які відомі сьогодні, як «Сцена в саду Раундхей» та «Рух транспорту мостом Лідс» і є одними з перших кінофільмів [13].

- 1891 рік. Кінетоскоп

Томас Едісон та його помічник У.К.Л. Діксон розробили пристрій для перегляду зображень, що рухаються — кінетоскоп. Зображення були кадрами позитиву плівки, а сам пристрій став мостом в епоху німого кіно.

- 1895 рік. Кінематограф

22 березня брати Люм'єр провели демонстрацію свого винаходу — камери-проектора під назвою кінематограф. У ході закритої демонстрації було проведено показ фільму «Робітники, що залишають фабрику Люм'єр». А у грудні того ж року відбувся перший публічний кінопоказ десяти фільмів, зокрема знаменитої картини «Прибуття поїзда на вокзал Ла-Сьота».

- 1902 рік. «Мандрівка на Місяць» Жорж Мельєс

Французький піонер кіно Жорж Мельєс зняв фільм «Мандрівка на Місяць», який став першим фільмом зі статусом культового шедевра. Крім цього, картина заслуговує на першість з демонстрації інноваційних спецефектів, вплинув тим самим значний вплив на кінематографістів та їх роботи.

- 1911 рік. Надсилання зображення

Німецький та російський інженер Микола Розінг провів успішний експеримент із передачі зображень на відстань. Використовуючи фотокатоди та електровакуумні трубки як основні елементи пристрою, він продемонстрував передачу зображення через електронні промені, що в майбутньому стало основою для телебачення.

- 1912 рік. Chronochrome

Французький кінопідприємець та винахідник Леон Гомон, ґрунтуючись на роботі пристрою Kinemascolor, представив пристрій Chronochrome — камеру з повнокольоровою адитивною системою. У камері використовувалися три об'єктиви з кольоровими фільтрами для одночасного фотографування кольорних

складових на чорно-білу плівку, яка у свою чергу проектувалась через відповідні лінзи для відображення кольорового достовірного зображення.

- 1927 рік. Звукове кіно

В 1927 кіноіндустрія зробила величезний крок з появою звукового кіно. 20 березня компанія Fox випустила перший звуковий фільм із використанням системи Movietone — «Вони йдуть за мною». А у лютому того ж року завдяки системі Vitaphone Warner Bros. представили світові «Співака джазу», перший повнометражний фільм із синхронним звуком, що ознаменувало кінець ери німого кіно.

- 1927 рік. «Метрополіс» Фріц Ланг

Вихід на екрани фільму Фріца Ланга «Метрополіс» залишив в історії кіно відчутний слід, ставши першим фільмом, спецефекти в якому стали справжнім проривом. Картина є одним із перших випадків застосування системи дзеркал для створення ілюзії реальності та масштабу складних декорацій, що колосально здешевлювало виробництво картини, дозволяючи створювати візуально вражаючі, але технічно складні сцени.

- 1933 рік. «Кінг Конг»

«Кінг Конг». став першим фільмом, який використав стоп-моушн анімацію в масовому кіно. Техніка покадрової зйомки моделей до створення складних і нереальних персонажів показала небувалий раніше результат. Крім того, картина славиться тим, що є однією з перших, де активно застосовувалася комбінована зйомка, коли персонажі знімалися окремо, а на етапі постпродакшну їх зображення збиралися в загальні кадри.

- 1950-1960-ті роки. Розвиток широкоформатного кіно

У зв'язку з популярністю телевізійного мовлення перед кіновиробниками постало питання про повернення глядачів у кінотеатри. Одним із значних рішень було запровадження широкоформатного зображення, всупереч традиційному, пропонуючи нові можливості та емоції. Найбільш впливовими форматами, які зробили значний внесок в історію кіно, стали:

1) Cinerama: Для створення панорамного зображення з кутом огляду 146 градусів, у 1952 році було розроблено систему Cinerama. Ґрунтуючись на паралельному використанні трьох кіноплівок, на кожен з яких знімалася своя частина панорамного кадру, технологія надавала глядачеві нові відчуття від перегляду, задіявши не тільки прямий зір, а й периферичний.

2) CinemaScope: Розроблена в 1953 році студією 20th Century Fox система CinemaScope, дозволила проектувати зображення зі стандартної 35-мм плівки на широкі екрани, вдавшись до використання анаморфотних об'єктивів. У 1950-х роках цей формат став стандартом для широкоформатних фільмів, надаючи унікальні просторові відчуття глядачеві.

3) VistaVision: У планах покращити якість та деталізацію зображення, у 1954 році Paramount Pictures розробляє свій кіноформат VistaVision. Відмовившись від стандартного вертикального компонування кадрів, формат переходить на горизонтальне розташування на плівці, тим самим збільшуючи площу кадру і відповідно підвищуючи якість проекції. Велику популярність формат набув фільмів, що вимагають високої чіткості та різкості зображення.

4) Todd-AO: У 1955 році Майкл Тодд запропонував використовувати 70-мм плівку для створення нової системи широкого кіно, яка отримала назву Todd-AO і стала першою у світі комерційно успішною, ставши для багатьох основою, на якій надалі будувалися інші. Створена система широкоформатного кіно забезпечила ще більш високу якість зображення та звуку, закріпивши на довгий час формат одного з головних гравців у кінематографі.

- 1982 рік. Поява комп'ютерної графіки

Справжнім проривом у кінематографі можна вважати вихід на широкі екрани фільму «Трон». Фільм має новаторські для того часу технології комп'ютерної графіки (CGI — Computer Generated Imagery), для створення небувалих раніше візуальних сцен. Подолавши колосальну, зокрема унікальну роботу, фільм став першим серйозним кроком до переходу до цифрового кіно.

- 1990-ті. Нова ера

У 1990-ті роки цифрова кіноіндустрія почала швидко розвиватися. Будучи знятим у 1988 році повністю у цифровому форматі, фільм «Дика кінь» став важливим етапом на новому ступені в кіноіндустрії. На щорічній виставці 1999 року в Лас-Вегасі було представлено перший цифровий кінопроектор Multiplex, що справило значне враження на майстрів кіномистецтва. А одним із ключових моментів у формуванні інтересу до цифрового кінематографу стало використання технологій у першому епізоді «Зоряних воєн» Джорджа Лукаса, якість та видовищність якого викликали величезний фурор на всю індустрію кіно [14].

1.4.2 Різниця між цифровою та аналоговою зйомкою

Як відомо, все своє існування кінематограф невпинно розвивається, надаючи нові несподівані рішення, які довго приживається в ньому. І так само, як колись кіноплівка потіснила свого попередника, так і цифрове кіно дуже швидко почало утискувати класичну плівку. Насправді цей перехід має дуже вагомні чинники, які зробили цифрове кіно якимось стандартом сьогодення, ставши новим кроком у цій індустрії розваг.

Поки ігрове кіно завжди було і є бізнесом, ставши таким ще на бульварі Капуцинок, а отже основним її мотивом є фінансовий зиск. Поява цифрових камер стала неймовірною полегшенням для босів кіно. Гнучкість, зручність і простота використання — одні з численних чинників якими наділена цифрова камера.

Говорячи про фінансову вигоду, природно маються на увазі витрати, що накладаються на творців під час зйомки. У такому разі можна провести порівняльний аналіз для детального розгляду поставленої проблеми.

Для початку варто виділити кілька етапів зйомки кіно:

- Препродакшн,
- Знімальний процес,

- Постпродакшн,
- Довгострокові витрати.

Предпродакшн — це етап підготовки до виробництва кіно. Одним із ключових завдань на даному етапі є вибір камер, які будуть задіяні у зйомці картини. Цей аспект є важливим, оскільки саме за допомогою камери режисер передає глядачеві задуманий стиль та настрій фільму. Для оцінки витрат на камери слід зазначити, що багато компаній не продають свої камери, а лише здають в оренду, тому порівняння варто проводити за цим параметром. Примітка: наведені нижче ціни є орієнтовними, оскільки компанії часто не розкривають реальної вартості.

Якщо розглядати як об'єкт оренди цифрову камеру виробництва RED, вартість оренди буде варіюватися від 300 до 1000 доларів за день, до цього додається страховка, яка становить від 10% до 20% від вартості оренди.

Якщо взяти як об'єкт аналогову камеру виробництва Panavision, яка є однією з еталонних, вартість оренди буде залежати від безлічі факторів — від типу фільму до погодних умов і місця зйомки. У середньому вартість оренди може становити від 1000 до 10,000 доларів за знімальний день. Страхування може досягти 500,000 доларів за весь знімальний процес

Також важливим чинником підготовки є підбір освітлення, що дозволяє робити потрібні акценти як у окремих деталях, і на роботі загалом. Але оскільки професійне освітлення є взаємозамінним як для зйомки на аналогову, так і для зйомки на цифрову камеру, то в даному випадку цим аспектом можна нівелювати.

Знімальний процес — це основний етап виробництва кіно, а носії інформації одними з найважливіших елементів цього процесу. Залежно від того, яка використовується камера, залежатиме і носій інформації — плівка або картка пам'яті (іноді жорсткий диск).

Якщо йдеться про цифрову зйомку, носій інформації можна використовувати повторно, перезаписуючи дані, а витрати на нього становлять незначну частину бюджету. Що стосується аналогової зйомкою плівка одна із

основних споживачів бюджету. Наприклад, на зйомки трилогії про «Володаря кілець» було витрачено понад 1800 кілометрів плівки.

Якщо розглядати як носій інформації цифрової камери RED карту пам'яті, то її вартість становить приблизно 300 доларів за 1 Тб пам'яті, що вистачає приблизно на 4 години відеоматеріалу. Таким чином, вартість носія на одну годину відео буде близько 75 доларів.

Якщо розглядати аналогову камеру Panavision, то стандартний рулон плівки 65 мм на 5 перфорацій, який підходить для цієї камери має довжину близько 180 метрів і коштує від 150 до 300 доларів. Різниця в ціні залежить від якості плівки, терміну придатності, а також того, чи вона була розкрита і як зберігалася. Для зйомки однієї години відео за стандартної частоти 24 кадри в секунду потрібно 86 400 метрів плівки. Таким чином, буде потрібно в середньому 480 котушок плівки, що обійдеться у суму від 72,000 до 144,000 доларів.

Крім витрат на витратні матеріали, такі як носії інформації, оплата роботи оператора залежить від виду зйомки. Оскільки така інформація не підлягає розголосу і лише рідко надаються приблизні суми, можна стверджувати, що вартість роботи оператора аналогової камери оцінюється вище вартості роботи оператора цифрової камери, навіть за умови, що це може бути одна і та ж людина.

Постпродакшн — це етап, що йде за знімальним процесом, коли матеріал, знятий на камеру, піддається обробці та підготовці до фінального релізу. Цей процес включає монтаж, корекцію кольору, додавання візуальних ефектів і створення фінальної копії фільму. Варто зауважити, що оцінки вартості етапів постпродакшену часто залишаються закритими через комерційні та контрактні угоди. Однак у деяких випадках, наприклад, для великих проектів, дані про вартість можуть бути частково доступні через податкову звітність або перед інвесторами. Винятком може стати етап додавання візуальних ефектів, який часто є найдорожчим у процесі. Також варто відзначити, що у випадку з цифровим кіно програмне забезпечення для роботи найчастіше купується один

раз за невелику вартість, тоді як у випадку із плівкою великі студії можуть дозволити собі мати спеціальні лабораторії або вдаються до послуг третіх осіб.

Першим етапом після того, як кіно знято, є монтаж. У разі цифрового кіно монтаж виконується на комп'ютері за допомогою спеціалізованого програмного забезпечення, такого як Avid Media Composer або Adobe Premiere. У разі плівкового кіно процес монтажу починається з прояву плівки. Цей етап є важливим перш за все, щоб не зашкодити негативу. Далі монтаж може бути як ручним, коли монтажер збирає кожен кадр вручну, так і оцифрованим, коли позитив плівки сканується спеціальним обладнанням та записується на цифровий носій, де вже після можна приступити до процесу монтажу, аналогічного цифровому.

Наступний етап, який, як правило, є найдорожчим, це додавання візуальних ефектів. Варто зауважити, що створення візуальних ефектів на плівці, не оцифровуючи її, можливе лише під час самої зйомки завдяки кольорофільтрам та освітленню. Загалом процес додавання візуальних ефектів однаковий як цифрового кіно, так аналогового. Дорожнеча даного етапу обумовлена кількома факторами:

- Вартість висококваліфікованих спеціалістів.
- Технічні складності процесу створення, які можуть включати як створення складних 3D-сцен та опрацювання найдрібніших деталей, так і анімацію та захоплення рухів.
- Час та обсяг роботи. Оскільки цей процес є дуже ретельним і скрупульозним, постпродакшн часто займає роки, тоді як сам фільм міг бути знятий лише за місяць.
- Технічні складнощі. Через те, що для достовірно реалістичного рендеру завжди використовується технологія трасування шляхів, де процес створення одного кадру може тривати від кількох хвилин до кількох днів. Саме тому студії орендують великі обчислювальні кластери або ферми.

Передостаннім етапом постпродакшену є корекція кольору. Так само як і в попередньому випадку із плівкою використовують її оцифровану версію. Цей

етап процесу є одним із найшвидших, де фахівець, а іноді й сам режисер, підбирає загальний тон картини для точної передачі потрібного настрою глядачеві.

Останній етап — це перенесення готової картини на носій. Цей етап може бути як простим чи складним, все залежить від типу фільму та його формату. Простим процесом є перенос цифрової копії на жорсткі диски для подальшої дистрибуції. Складним є процес, коли цифрову версію переносять на плівку для того, щоб досягти естетичної переваги при показі або коли студія хоче архівувати роботу, оскільки плівка є стабільнішим носієм.

1.5 Аналіз стрімінгових сервісів

Як уже відомо з першого розділу, родоначальником стрімінгових сервісів є компанія RealNetworks, яка й задала тон розвитку технології як такої. Підхопивши популярність RealNetworks, стали з'являтися аналогічні сервіси, які в гонці за першість, залучали все більше і більше нових користувачів.

1.5.1 Найпопулярніші представники сфери

Сьогодні стрімінг є невід'ємною частиною проведення дозвілля. Хоч стрімінг все ще розбивається, у цій сфері вже закріпилися свої гіганти, які не лише є прокатниками у сучасному розумінні, а й диктують свої правила. Ці компанії не лише надають контент, а й формують уподобання аудиторії, впливаючи на переваги глядачів, створюючи тренди та формати та популяризуючи певні жанри. У контексті цього варто розглянути найяскравіших представників цього напрямку.

Netflix.

Будучи заснованим двома підприємцями — Рідом Хастінгсом та Марком Рендольфом у 1997 році, сервіс став першим, який запропонував оренду DVD-фільмів поштою. Будь-який користувач міг, не виходячи надвір, замовити

фільм за своїм бажанням і незабаром отримати його додому. Це рішення на той момент було революційним, але справжню революцію у світі кіно компанія зробила за десять років, коли запустила стрімінг. Цей крок став ключовим моментом у розвитку не тільки компанії, а й усієї індустрії в цілому, вплинув на подальший розвиток цифрових розваг. Цілодобово доступна з величезною бібліотекою фільмів та серіалів компанія надавала користувачам необмежене використання контенту без плати за кожен перегляд, а вимагала лише оформити передплату за символічну суму. Наростивши аудиторію, яка полюбилася всім, компанія незабаром стала не лише прокатником, а й незалежним продюсером, що остаточно змінило світ кінематографу. Саме завдяки Netflix багато акторів можуть зберігати свою популярність, беручи участь в оригінальних проектах компанії, навіть якщо їхні роботи не стають світовими хітами, тоді як раніше кар'єрі таких акторів відразу настав кінець. Будучи першими, компанія виховала у своїх користувачах розуміння того, як має виглядати стрімінг, і саме тому сьогодні, коли аналогів незліченна кількість, глядачі все одно віддають перевагу Netflix.

Hulu

Hulu було засновано у 2007 році як спільне підприємство акціонерів в особі таких великих медіа-гігантів, як NBCUniversal, News Corporation, до якого пізніше приєднався і The Walt Disney Company. На відміну від Netflix, який розпочав свою діяльність з оренди дисків, Hulu спочатку орієнтувався на потокове відео. З самого початку метою компанії стало надання користувачам доступу до поточних телевізійних шоу та фільмів з можливістю перегляду в записі або в реальному часі. Незабаром цю ідею підхопило багато традиційних телевізійних каналів при виході в Інтернет. Будучи орієнтованим на надання нового контенту від телемовних компаній, Hulu відрізнявся від конкурентів на початковому етапі, що й дозволило сервісу зайняти свою нішу. Будучи все ще переважно дистриб'ютором телевізійної мережі, Hulu швидко набрав популярність, головним чином завдяки тому, що не транслював рекламу, як це роблять традиційні телеканали. За щомісячну передплату користувачі мають

широкий спектр послуг, чого не може запропонувати класичне телебачення. Надалі сервіс почав активно розвивати свій оригінальний контент, і на сьогоднішній день є автором таких популярних проєктів, як «Оповідання служниці» та «Касл-Рок». Сьогодні Hulu продовжує відрізнятися від Netflix своєю відданістю традиційним телемовним та кіновиробничим моделям, що робить його платформою з унікальною аудиторією, що поєднує старі та нові формати.

Disney+

Запущений у листопаді 2019 року, Disney+ став важливим кроком для корпорації The Walt Disney Company у спробі конкурувати із сервісом Netflix та ще одним кроком у спробі монополізації ринку. Сервіс швидко завоював аудиторію, орієнтуючись насамперед на дитячу та підліткову вікові групи. Основним продуктом дистрибуції сервісу стали мультфільми завдяки унікальній бібліотеці контенту, що містить усі фільми та серіали, створені під марками Disney, Pixar, Marvel, Star Wars, включаючи класичні анімаційні роботи. Незважаючи на те, що сервіс позиціонує себе як інструмент для сімейного перегляду, пропонуючи справді нові рішення серед конкурентів у даній сфері, згідно з відгуками Disney+ користується популярністю завдяки величезній бібліотеці старих проєктів, які вже не транслюються ні в кіно, ні на телебаченні.

1.5.2 Аналіз готових рішень

Незважаючи на те, що описані раніше послуги є свого роду зразками у своїй галузі, існує також безліч дрібніших рішень, орієнтованих на широку аудиторію. Як правило, такі сервіси надають різноманітний контент різного типу, формату та якості. Вивчаючи популярні сервіси, що використовуються в нашій країні, було проведено аналіз наступних онлайн-кінотеатрів:

- Filmix,
- HDrezka,

- KinoGo,
- Andyday.

Основними недоліками досліджуваних рішень є:

- Повна або часткова відсутність добірок картин для перегляду, у тому числі на основі вже переглянутих та оцінених користувачем;
- Незручний пошук, який орієнтований в основному на пошук за повною назвою, а не за ключовими словами, що є кращим;
- Неможливість виключати небажані параметри під час фільтрації;
- Тяжка структура, заснована на складних фреймворках та бібліотеках;
- Незручна система коментарів, яка робить коментарі важко прочитаними;
- Відсутність можливості створення групових облікових записів, що корисно при відстеженні активності дитини на сайті;
- Відсутність можливості спільного перегляду у реальному часі.

Крім основних недоліків послуги мають і ряд позитивних якостей, які варто відзначити, а саме:

- широка база фільмів, серіалів, мультфільмів, яка активно поповнюється;
- Можливість перегляду контенту в максимально доступній роздільній здатності;
- Активована адміністрація, що взаємодіє з користувачами;
- Швидка робота сервісів.

Таким чином, незважаючи на деякі недоліки, сервіси все одно є відмінними представниками доступу до медіа-контенту, орієнтованим на різні вікові та етнічні групи.

1.6 Вимоги до інформаційної системи

Інформаційна система онлайн-кінотеатру призначена для надання зручної можливості перегляду фільмів та серіалів. У ході аналізу раніше досліджуваних аналогів можна скласти низку вимог, які мають бути виконані для надання сервісу статусу онлайн кінотеатру, а саме:

- Нефункціональні вимоги:
 - Сервіс повинен базуватися на технології SPA;
 - Сервіс має бути доступним для користування в будь-який період часу;
 - Сервіс повинен мати тестову версію, яка буде використана для тестування нововведень та перевірки працездатності нового матеріалу;
 - Сервіс повинен надавати доступ до широкого вибору матеріалу
 - Сервіс має бути зручним та інтуїтивно зрозумілим у використанні користувачами;
 - Сервіс повинен залишатися працездатним та надавати базовий набір функціональності для не авторизованих користувачів.
- Функціональні вимоги
 - Додаток повинен складатися як мінімум з двох основних логічних сторінок — сторінка зі списком картин на сайті, сторінка з програвачем для перегляду;
 - Список картин для перегляду має бути впорядкований з можливістю сортування та розбиття на колекції за внутрішніми критеріями фільму або серіалу;
 - Сервіс повинен мати можливість пошуку картини як мінімум, за ключовими словами, чи назвою;
 - Кожна картина повинна мати набір мінімальної інформації, даючи користувачеві можливість для ознайомлення перед переглядом, а

саме: назва, опис, хронометраж, рік зйомки, віковий рейтинг, оцінка користувачів, список акторів, список авторів.

- Відтворення відеоматеріалу має здійснюватися в потоковому режимі, підвантажуючи фрагменти, а не завантажуючи відеофайл.
- Сервіс повинен мати можливість надавати користувачеві можливість оцінки картини та її коментування, обговорення з іншими користувачами.

Виходячи з низки функціональних вимог для сервісу, що позиціонує себе, як онлайн-кінотеатр можна скласти низку вимог виключно для поточного проекту:

- Весь матеріал на сервісі має бути розділений на три групи:
 - фільми,
 - серіали,
 - мультфільми.
- У сервісу повинен бути пошук за ключовими словами;
- На сервісі повинні бути такі розділи:
 - головна сторінка;
 - сторінка для кожної групи;
 - сторінка зі списком кращих картин за мірками сервісу;
 - сторінка для перегляду картини та з інформацією про неї;
 - сторінка про актора чи режисера;
 - сторінка реєстрації на сайт.
- Будь-який логічний елемент, який є контейнером для картин, повинен відображати їх у вигляді картки, на якій повинна бути розміщена основна інформація для ознайомлення, без безпосереднього переходу на сторінку картини.
- Будь-який логічний елемент, який є контейнером для колекції картин певного критерію, повинен бути представлений у вигляді слайдера.
- На першій сторінці першими повинні відображатися останні додані картини на сайт.

- Кожна картина повинна мати можливість вибору якості перегляду.
- Кожна картина повинна мати можливість включення та відключення показу субтитрів за наявності, що вони передбачені

Технічні вимоги до frontend:

- HTML 5.0,
- CSS 3.0,
- JS ES6.

Виходячи з того, що продукт, що розробляється, є веб-додатком і спираючись на описані вище вимоги можна зробити висновок, що на сайті не будуть проводитися складні обчислення та/або задіяна відеокарта, як окремий обчислювальний пристрій. У такому разі системними вимогами для пристрою користувача є мінімальні версії 80.0, які потрібні для запуску браузера Google Chrome.

Апаратні вимоги до персонального комп'ютера:

- Процесор intel Pentium 4 чи пізнішої версії;
- Об'єм оперативної пам'яті 1024 Мб та більше;
- Відеопам'ять щонайменше 64 Мб.

Вимоги до операційної системи:

- Windows: версія не нижче 7;
- Linux: Ubuntu 20.04 та вище, Debian 10 і вище;
- macOS: не нижче 10.10;
- Android: версія не нижче 4.4;
- IOS: версія не нижче 12.

Технічні вимоги до backend:

- PHP 7.2,
- MySQL 8.0.

Оскільки для обробки запитів клієнтів на сайті та виконання операцій із базою даних використовується віддалений комп'ютер (сервер), для коректної роботи висуваються такі системні вимоги:

- Об'єм жорсткого диска:
 - 10 Гб для бази даних та системних файлів;
 - 4.5 Гб + 15% з розрахунку на одну годину контенту (відеоматеріал та ресурси).
- Оперативна пам'ять:
 - 16 Гб оперативної пам'яті для роботи системи;
 - 120Мб з розрахунком на одного користувача.

Процесор з тактовою частотою не менше 3 ГГц та 4 ядрами з підтримкою Hyper-Threading (з розрахунком на обробку до 10 одночасних користувачів).

1.7 Висновки до розділу 1

У першому розділі було досліджено історію стрімінгових сервісів, історію кіно загалом та досліджено інструменти веб-розробки. На базі дослідженого матеріалу був проведений аналіз предметної області, що дозволило зробити постановку задачі та розробити як вимоги до системи, так і вимоги до апаратної частини. На основі цього дослідження буде спроектовано інформаційну систему онлайн-кінотеатру.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН КІНОТЕАТР

2.1 Побудова моделі варіантів використання

Діаграми варіантів використання в UML — це діаграми, які моделюють поведінку системи та допомагають фіксувати вимоги до неї. Діаграми варіантів використання описують високорівневі функції та область дії системи. Ці діаграми також визначають взаємодію між системою та її суб'єктами. Варіанти використання та суб'єкти в даних діаграмах описують, що робить система і як суб'єкти її використовують, але не те, як система працює усередині. Діаграми варіантів використання ілюструють та визначають контекст та вимоги як усієї системи, так і її важливих частин [15].

Компоненти діаграми:

- актор,
- варіант використання,
- кордон системи,
- включення,
- розширення,
- узагальнення.

На рисунках 2.1 та 2.2 зображено моделі варіантів використання для проекту, що розробляється. На діаграмі відображені основні відносини між Акторами та системами, що відбуваються завдяки прецедентам.

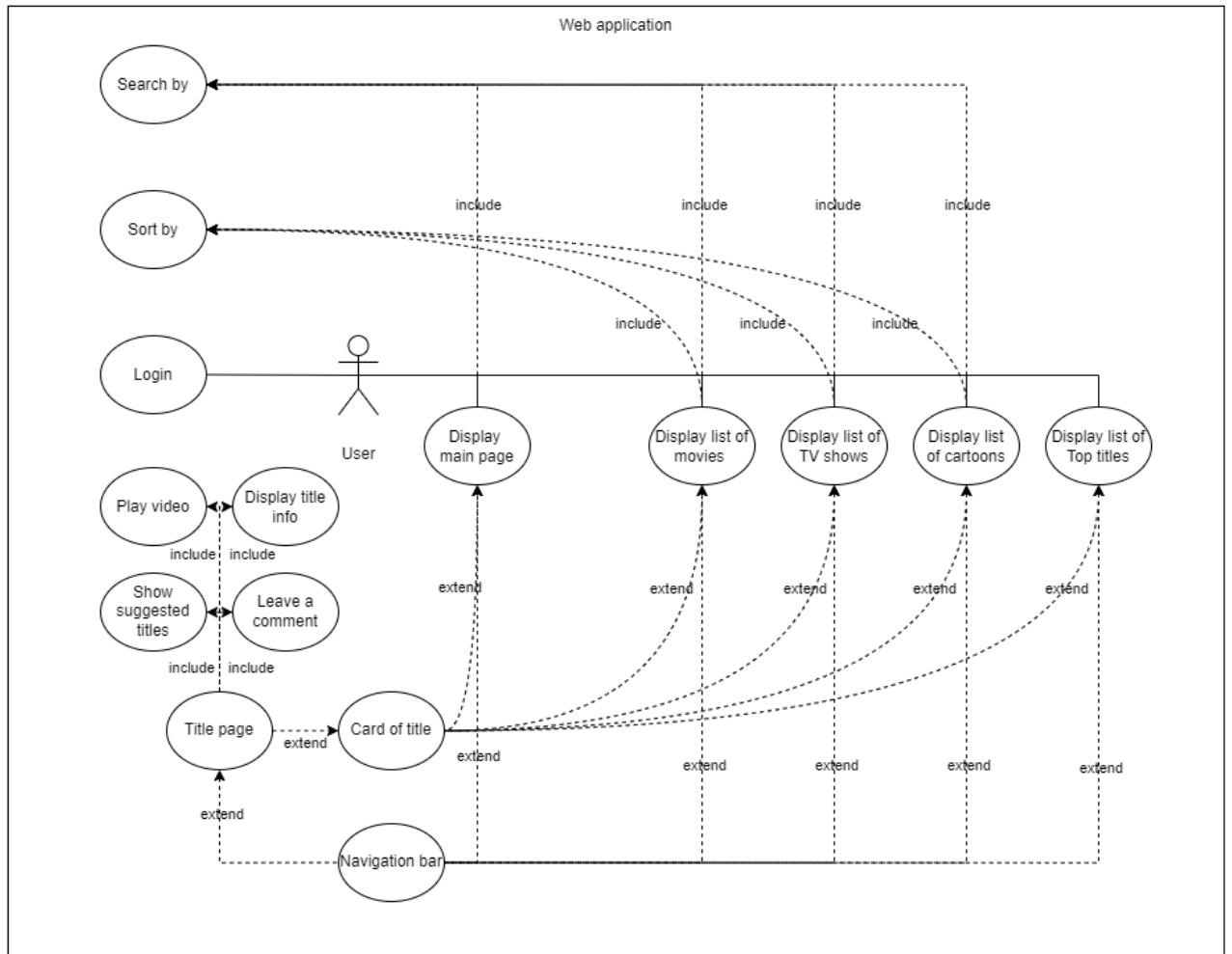


Рисунок 2.1 — діаграма варіантів використання. Взаємодія користувача з веб-додатком

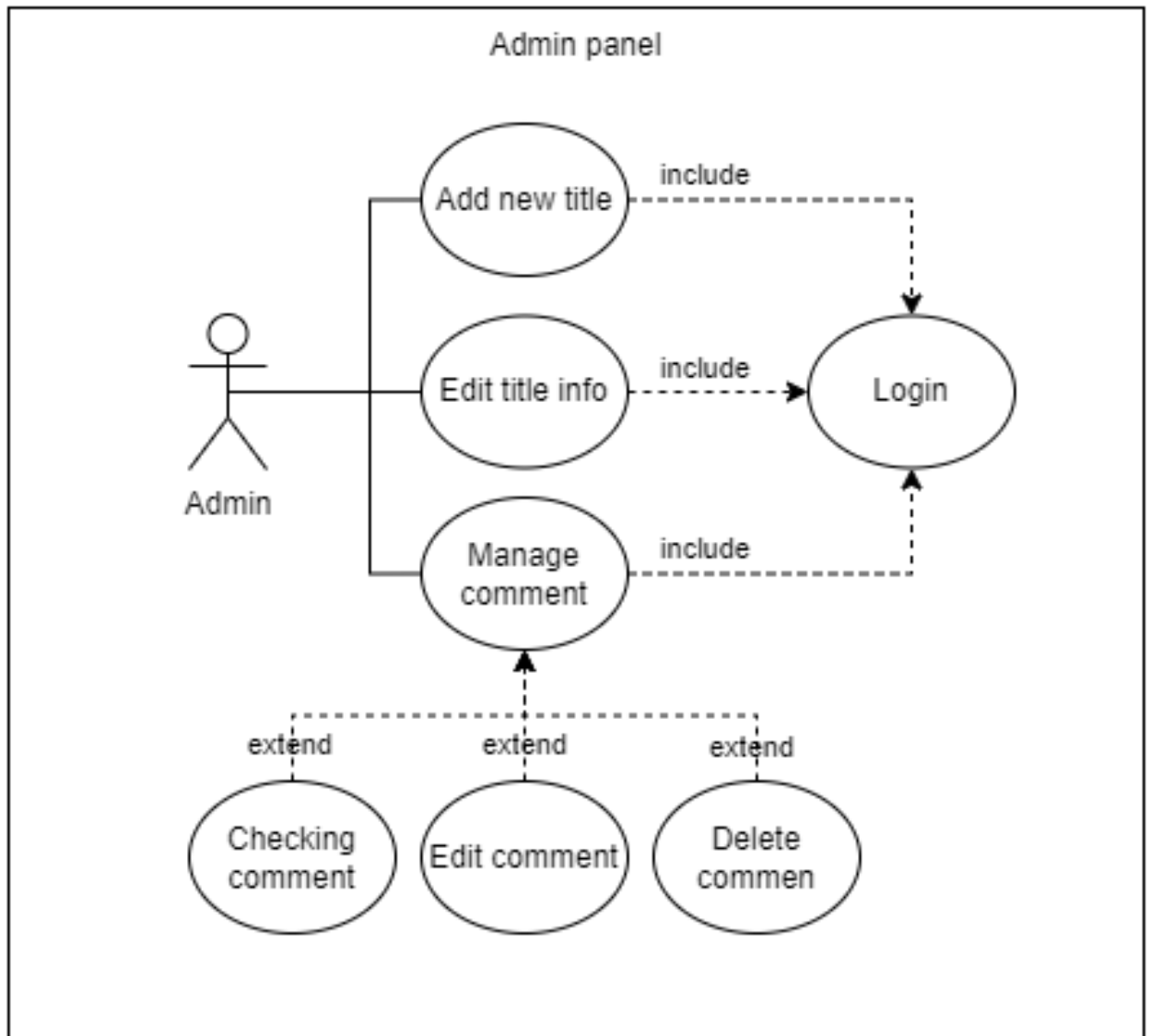


Рисунок 2.2 — діаграма варіантів використання. Взаємодія адміністратора з веб-додатком

Для кращого розуміння діаграм варіантів використання нижче надані специфікації до них.

Опис акторів

User — будь-який користувач розробленого веб-додатку.

Admin — адміністратор або менеджер веб-додатку.

Опис систем

Web application — середовище, що є розробленим веб-додатком. Основна роль цієї системи — надання доступу до перегляду фільмів, серіалів, мультфільмів користувачам.

Admin panel — середовище, яке є стороннім додатком для управління основним проектом.

Специфікації

Display main page:

- Головний актор: User.
- Інші учасники: відсутні.
- Опис: ця дія служить для переходу на головну сторінку.
- Передумова: відсутня.
- Основний потік: клієнт натискає певну кнопку.
- Постумова: сторінка оновлюється. Відобразиться головна сторінка.
- Альтернативний потік: у разі втрати з'єднання перехід не здійснюється.

Display list of movies:

- Головний актор: User.
- Інші учасники: відсутні.
- Опис: ця дія служить для переходу на сторінку зі списком усіх завантажених фільмів.

- Передумова: відсутня.
- Основний потік: клієнт натискає певну кнопку.
- Постумова: сторінка оновлюється. Відображається сторінка зі списком фільмів.

- Альтернативний потік: у разі втрати з'єднання перехід не здійснюється.

Display list of TV shows:

- Головний актор: User.

- Інші учасники: відсутні.
- Опис: ця дія служить для переходу на сторінку зі списком усіх завантажених серіалів.
- Передумова: відсутня.
- Основний потік: клієнт натискає певну кнопку.
- Постумова: сторінка оновлюється. Відобразиться сторінка зі списком серіалів.
- Альтернативний потік: у разі втрати з'єднання перехід не здійснюється.

Display list of cartoons:

- Головний актор: User.
- Інші учасники: відсутні.
- Опис: ця дія служить для переходу на сторінку зі списком усіх завантажених мультфільмів.
- Передумова: відсутня.
- Основний потік: клієнт натискає певну кнопку.
- Постумова: сторінка оновлюється. Відобразиться сторінка зі списком мультфільмів.
- Альтернативний потік: у разі втрати з'єднання перехід не здійснюється.

Display list of Top titles:

- Головний актор: User.
- Інші учасники: відсутні.
- Опис: ця дія служить для переходу на сторінку зі списком кращих картин за версією цього сервісу.
- Передумова: відсутня.
- Основний потік: клієнт натискає певну кнопку.
- Постумова: сторінка оновлюється. Відображається сторінка зі списком найкращих картин.

- Альтернативний потік: у разі втрати з'єднання перехід не здійснюється.

Play video:

- Головний актор: User.
- Інші учасники: відсутні.
- Опис: ця дія служить для запуску відеопрогравача, який знаходиться у певній секції сторінки тієї чи іншої картини.

- Передумова: сторінку картини завантажено.
- Основний потік: клієнт натискає певну кнопку.
- Постумова: на сторінці відображається вбудований відеоплеєр, який дозволяє програти відео, вибрати якість, увімкнути або вимкнути субтитри.

- Альтернативний потік: у разі втрати з'єднання відеоплеєр не програватиме відео. У разі погіршення з'єднання відеоплеєр автоматично перемкне якість на нижчу.

Display title info:

- Головний актор: User.
- Інші учасники: Admin.
- Опис: ця дія служить для відображення певної секції сторінки тієї чи іншої картини, на якій міститься інформація про неї. Деяка інформація, така як рейтинг користувачів, підвантажується і з інших джерел.

- Передумова: сторінку картини завантажено.
- Основний потік: відсутній.
- Постумова: на сторінці відображається актуальна інформація.
- Альтернативний потік: відсутній.

Show suggested titles:

- Головний актор: User.
- Інші учасники: відсутні.

- Опис: ця дія служить для відображення певної секції сторінки тієї чи іншої картини, на якій розміщується список зі схожими картинами.

- Передумова: сторінку картини завантажено.

- Основний потік: відсутній.

- Постумова: на сторінці відображається список зі схожими картинами.

- Альтернативний потік: відсутній.

Leave a comment:

- Головний актор: User.

- Інші учасники: Admin.

- Опис: ця дія служить для відображення певної секції сторінки тієї чи іншої картини, на якій розташовується область, що дозволяє залишити коментар, та область для відображення списку коментарів, залишених іншими користувачами.

- Передумова: сторінку картини завантажено, користувач авторизовано (опціонально).

- Основний потік:

- Якщо користувач авторизований, актор вводить коментар і натискає на кнопку «Відправити».

- Якщо користувач не авторизовано — відсутня.

- Постумова:

- Якщо користувач авторизовано, коментар надсилається на сервер для модерації адміністратором. Відображається список коментарів, що залишилися раніше.

- Якщо користувач не авторизований, то відображається список коментарів, що залишилися користувачами раніше.

- Альтернативний потік: у разі втрати з'єднання коментар не буде надіслано на сервер, а коментарі, залишені іншими користувачами, не будуть завантажені.

Login:

- Головний актор: User, Admin.
- Інші учасники: відсутні.
- Опис: ця дія служить для авторизації. Для користувачів сервісу є опціональною дією.
- Передумова: сторінку авторизації завантажено.
- Основний потік:
 - Актор запроваджує свої авторизаційні дані.
 - Актор натискає кнопку авторизації.
- Постумова: актор авторизований.
- Альтернативний потік: у разі втрати з'єднання актор не пройде авторизації.

Add new title:

- Головний актор: Admin.
- Інші учасники: відсутні.
- Опис: ця дія служить для завантаження нової картини.
- Передумова: адміністратор авторизовано.
- Основний потік:
 - Адміністратор вводить основні дані про картину.
 - Адміністратор завантажує ресурси.
 - Адміністратор натискає на відповідну кнопку.
- Постумова: на тестовому сервері з'являється нова картина, де перевіряється коректність інформації про неї та якість відеоматеріалу.
- Альтернативний потік: у разі втрати з'єднання завантаження не здійснюється.

Edit title info:

- Головний актор: Admin.
- Інші учасники: відсутні.
- Опис: ця дія служить для редагування інформації про картину.

- Передумова: адміністратор авторизовано.
- Основний потік: адміністратор редагує наявну інформацію.
- Постумова: відредагована інформація оновлюється.
- Альтернативний потік: у разі втрати сполуки зміни не застосовуються.

Checking comment:

- Головний актор: Admin.
- Інші учасники: User.
- Опис: дана дія служить для валідації коментаря, що залишився користувачем.

- Передумова: адміністратор авторизовано.
- Основний потік: адміністратор вирішує опублікувати або не публікувати коментар.

- Постумова: якщо коментар опубліковано, то його буде відображено у списку коментарів.

- Альтернативний потік: у разі втрати сполуки рішення про валідацію коментаря не застосовується, коментар залишається невалідованим.

Edit comment:

- Головний актор: Admin.
- Інші учасники: User.
- Опис: ця дія служить для редагування адміністратором залишеного користувачем коментаря.

- Передумова: адміністратор авторизовано.
- Основний потік: адміністратор застосовує зміни до коментарів.
- Постумова: зміни застосовуються.
- Альтернативний потік: у разі втрати сполуки зміни до коментаря не застосовуються.

Delete comment:

- Головний актор: Admin.

- Інші учасники: User.
- Опис: ця дія служить для видалення адміністратором залишеного користувачем коментаря.
- Передумова: адміністратор авторизовано.
- Основний потік: адміністратор видаляє коментар.
- Постумова: коментар і всі відповіді на нього, якщо такі були, видаляються.
- Альтернативний потік: у разі втрати з'єднання видалення коментаря не відбувається.

2.2 Рішення по інформаційному забезпеченню

Одним із ключових компонентів проекту є сховище даних. Сховище даних можна поділити на дві частини. Перша частина призначена для зберігання графічних та медіа-ресурсів, які розміщуються у відповідних директоріях на сервері. Друга частина — це текстові дані, що містять інформацію про основний матеріал, користувачів та інше.

Таким чином, сховище даних включає всі елементи, які служать для зберігання інформації, використовуваної проектом. Текстові дані зазвичай зберігаються в базі даних, а медіа-ресурси у файловій системі з метаінформацією, збереженою в базі даних. Для зручної взаємодії з даними у проекті використовується система управління базами даних (СКБД) MySQL v8.0.

MySQL — найпопулярніша у світі система керування базами даних з відкритим вихідним кодом. Її надійність, продуктивність, масштабованість та простота використання роблять MySQL популярним вибором для розробників. MySQL досить універсальна, щоб підтримувати широкий спектр додатків від невеликих особистих проектів до корпоративних критично важливих для бізнесу систем [16].

2.2.1 Розробка концептуальної моделі даних

Концептуальна модель — це абстрактна модель предметної області, що складається зі списку понять, які використовуються для опису предметної області, а також властивостей та характеристик, класифікації понять.

Нижче наведено концептуальну модель, представлену у вигляді діаграми «сутність-зв'язок» на рисунку 2.3. На діаграмі відображено основну частину даних для розуміння структури проекту та взаємодії його елементів між собою.

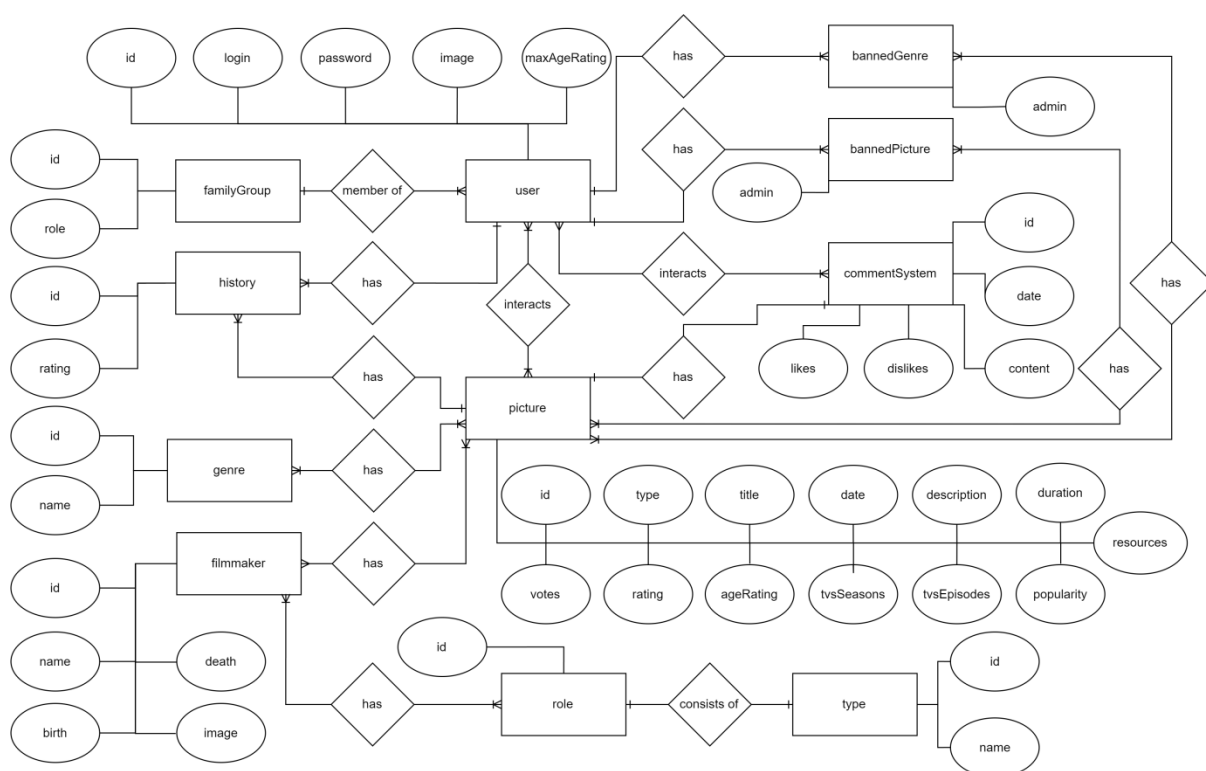


Рисунок 2.3 — Концептуальна модель

2.2.2 Розробка логічної модель даних

Логічна модель даних — це вдосконалена версія концептуальної моделі даних. Логічна модель даних конкретніша і враховує деталі реалізації. Вона включає всі сутності, атрибути, ключі і взаємозв'язки, які представляють бізнес-інформацію і визначають бізнес-правила.

Нижче наведено логічну модель проекту у вигляді діаграми «сутність-зв'язок» на рисунку 2.3.

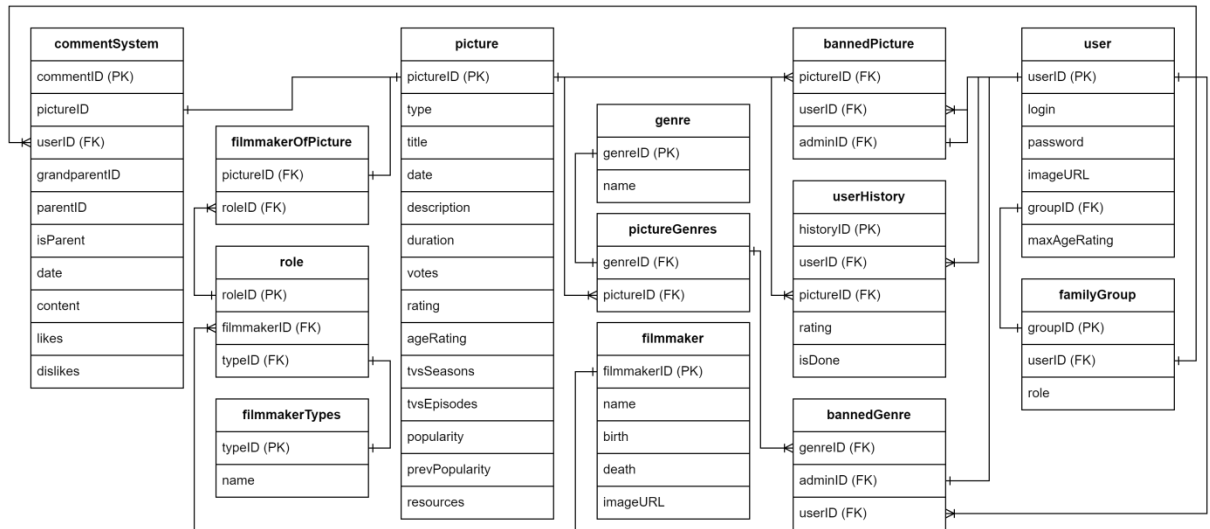


Рисунок 2.4 — Логічна модель

2.2.3 Розробка фізичної модель даних

Фізична модель даних — це наступний етап після логічної моделі або нижчий рівень абстракції. Фізична модель даних визначає, як модель буде побудовано в базі даних. У ній описуються всі структури таблиць, включаючи назву стовпця й полів, типи даних та обмеження по розміру.

На рисунку 2.5 представлена фізична модель проекту. У таблицях 2.1 — 2.13 наведено фізичну модель проекту у табличному форматі.

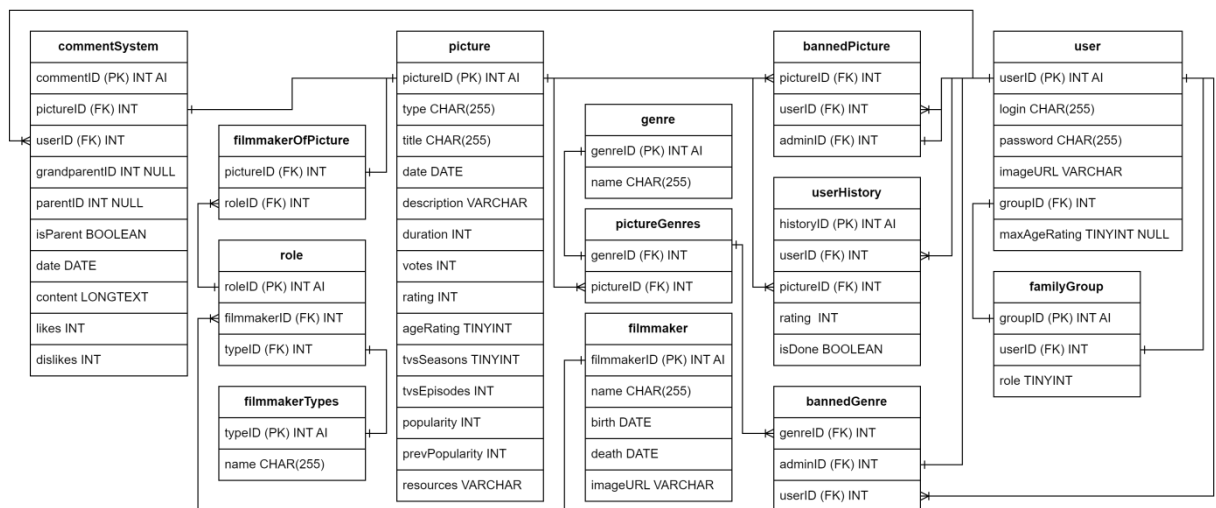


Рисунок 2.5 — Фізична модель

Таблиця 2.1 — Таблиця commentSystem

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	ККЛ юч
commentID	INT	-	+	PK
pictureID	INT	-	-	FK
userID	INT	-	-	FK
grandparentID	INT	+	-	
parentID	INT	+	-	
isParent	BOOLEAN	-	-	
date	DATE	-	-	
content	LONGTEXT	-	-	
likes	INT	-	-	
dislikes	INT	-	-	

Таблиця 2.2— Таблиця filmmakerOfPicture

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
pictureID	INT	-	-	FK
roleID	INT	-	-	FK

Таблиця 2.3— Таблиця role

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
roleID	INT	-	+	PK
filmmakerID	INT	-	-	FK
typeID	INT	-	-	FK

Таблиця 2.4 — Таблиця filmmakerTypes

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
typeID	INT	-	+	РК
name	CHAR(255)	-	-	

Таблиця 2.5 — Таблиця picture

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
pictureID	INT	-	+	РК
type	CHAR(255)	-	-	
title	CHAR(255)	-	-	
date	DATE	-	-	
description	VARCHAR	-	-	
duration	INT	-	-	
votes	INT	-	-	
rating	INT	-	-	
ageRating	TINYINT	-	-	
tvSeasons	TINYINT	-	-	
tvEpisodes	INT	-	-	
popularity	INT	-	-	
prevPopularity	INT	-	-	
resources	VARCHAR	-	-	

Таблиця 2.6 — Таблиця genre

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
genreID	INT	-	+	РК
name	CHAR(255)	-	-	

Таблиця 2.7 — Таблиця pictureGenres

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
genreID	INT	-	-	FK
pictureID	INT	-	-	FK

Таблиця 2.8 — Таблиця filmmaker

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
filmmakerID	INT	-	+	РК
name	CHAR(255)	-	-	
birth	DATE	-	-	
death	DATE	-	-	
imageUrl	VARCHAR	-	-	

Таблиця 2.9 — Таблиця bannedPicture

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
pictureID	INT	-	-	FK
userID	INT	-	-	FK
adminID	INT	-	-	FK

Таблиця 2.10 — Таблиця userHistory

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
historyID	INT	-	+	PK
userID	INT	-	-	FK
pictureID	INT	-	-	FK
rating	INT	-	-	
isDone	BOOLEAN	-	-	

Таблиця 2.11 — Таблиця bannedGenre

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
pictureID	INT	-	-	FK
adminID	INT	-	-	FK
userID	INT	-	-	FK

Таблиця 2.12 — Таблиця user

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
userID	INT	-	+	PK
login	CHAR(255)	-	-	
password	CHAR(255)	-	-	
imageURL	VARCHAR	-	-	
groupID	INT	-	-	FK
maxAgeRating	TINYINT	+	-	

Таблиця 2.13 — Таблиця familyGroup

Назва поля	Тип даних	NULL за замовчуванням	Автоматичне збільшення	Ключ
groupID	INT	-	+	PK
userID	INT	-	-	FK
role	TINYINT	-	-	

Фізична модель бази даних буде використана за допомогою СКБД MySQL v8.0.

2.3 Рішення з математичного забезпечення

Діаграма діяльності — це графічне представлення робочих процесів, алгоритмів із підтримкою вибору, ітерації та паралельності. Вона дозволяє моделювати послідовність дій, потік управління, паралельні процеси.

Для демонстрації основної логіки проекту на рисунку 2.6 надано блок-схему.

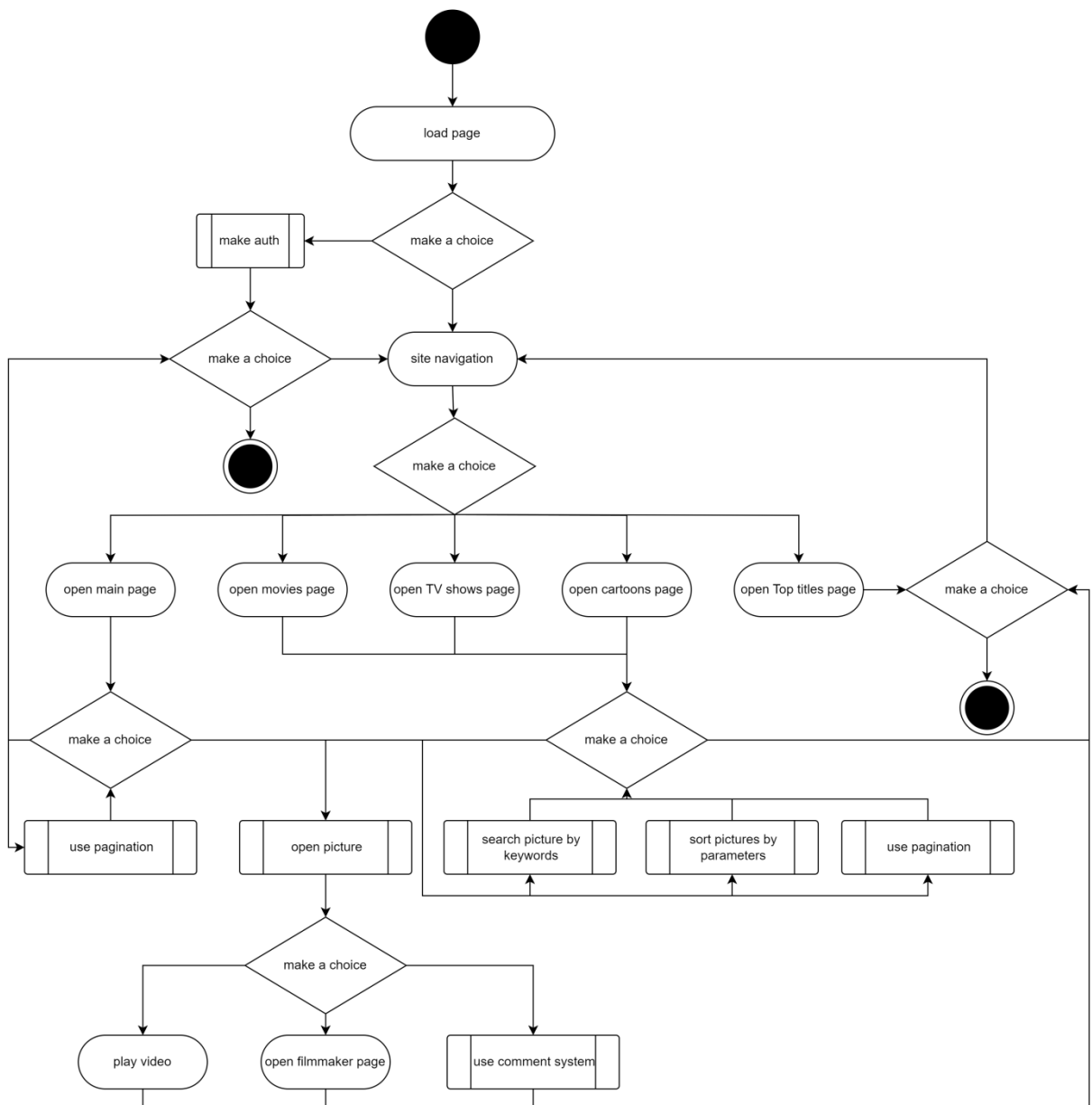


Рисунок 2.6 — Блок-схема основної логіки проекту

Проаналізувавши блок-схему, можна побачити, що сайт надає вільну навігацію користувачеві та не змушує робити якісь дії обов'язково.

Спочатку, після переходу на сайт, незалежно від зазначеного шляху, перед користувачем встає вибір, пройти авторизацію, якщо його ще не авторизовано, або відразу почати використання сайту. Оскільки авторизація є опціональним варіантом, не зробивши її, жодна головна функціональність додатка не буде зламана і залишається доступною користувачеві.

Використання сайту передбачає навігацію по ньому. Для навігації є чотири основні розділи та один додатковий.

«Main page» — розділ або сторінка сайту, яка є головною і відкривається при переході за посиланням без вказівки шляху. На цій сторінці знаходиться список останніх випущених фільмів, незалежно від типу та жанру. Також на сторінці є згруповані списки за однією з категорій. Ці категорії можуть включати як список улюблених серіалів користувача, так і список фільмів, об'єднаних певною темою. В самому низу головної сторінки є блок з усім медіа-контентом сайту, розбитий на сторінки для зручної навігації.

«Movies page» — сторінка сайту, яка надає список усіх фільмів, завантажених на сайт. На сторінці є пошук фільмів та їх сортування.

«TV Shows page» — сторінка сайту, яка надає список усіх серіалів, завантажених на сайт. На сторінці є пошук серіалів та їх сортування. Пошук дозволяє знаходити як за назвою серіалу, так і за назвою окремого епізоду.

«Cartoons page» — сторінка сайту, яка надає список усієї анімації, завантаженої на сайт. На сторінці також є пошук та сортування. Цей розділ має опцію для роботи з повнометражними або серійними картинками.

«Top title page» — сторінка сайту, яка надає список відсортованих картин кожного з типів за рейтингом сайту. Рейтинг автоматично складається на основі оцінок користувачів.

Окрім основних розділів, одним з основних способів навігації є пошук, який завжди знаходиться на навігаційній панелі. Він дозволяє знайти потрібну

картину або її актора. Пошук здійснюється за ключовими словами, а не за пов'язаними за змістом словосполученнями.

Кожна картина представлена у вигляді картки на сайті. При натисканні на картку користувач потрапляє на сторінку картини. На сторінці кожної картини доступна найголовніша функція — функція відтворення відео. При натисканні на кнопку «Play» на сторінці сайту з'являється відеоплеєр з можливістю вибору якості відео та вмикання/вимикання субтитрів. Оскільки відео відтворюється в міру перегляду, користувач може змінювати якість без переривання самого потоку.

Окрім перегляду, користувач має можливість залишати коментарі під переглянутим фільмом. Усі коментарі, якщо вони не є основними або «кореневими», можна згорнути або розгорнути для зручності користування та навігації по сторінці.

Також у кожного актора, режисера та іншого учасника зйомки картини є особиста сторінка, на яку можна потрапити, натиснувши на його ім'я.

2.4 Рішення з програмного забезпечення

Виходячи з опису програми, що розробляється, інформаційна система використовує архітектуру клієнт-сервер. Як клієнт виступає браузер користувача. Як сервер виступає спеціалізоване програмне забезпечення, встановлене на віддаленому комп'ютері.

Для розробки web-програми було обрано наступні інструменти:

- HTML для верстки сайту.
- Css для стилізації верстки.
- JavaScript для написання коду браузерної частини.
- PHP для написання коду серверної частини.
- MySQL як СКБД.
- OpenServer для запуску локального сервера.

- Notepad++ в якості середовища для написання коду.
- Figma в якості сервісу для розробки дизайну.

Насамперед треба зазначити, що для реалізації проєкту не використовувалися ніякі бібліотеки, фреймворки чи спеціалізовані двигуни, котрі дозволяють спростити процес розробки.

Для реалізації клієнтської частини було вирішено не вдаватися до використання бібліотек, що надають готові компоненти для прискорення процесу розробки, інструментів таких як React або систем управління контентом типу WordPress. Якщо випадок з двигуном WordPress та його аналогами не розглядається з природних причин, то стосовно перших двох випадків можна виділити дві причини:

- Відмова від використання компонентних бібліотек обумовлений отриманням нового досвіду під час створення інтерактивних елементів сторінки.
- Застосування класичних інструментів (HTML, CSS, JS) обумовлено тим, що бібліотеки або фреймворки завжди будуть надавати гіршу швидкість роботи кінцевого продукту. Ці інструменти роблять процес розробки простішим, прискорюють його і не змушують розробників глибше продумувати логіку, ніж у випадку з «ванільним» JavaScript, але тим самим внутрішня логіка даних технологій також потребує обчислювальних ресурсів.

Як серверну мову програмування було обрано PHP. Вибір саме цієї мови було зроблено з кількох причин:

- Мова є простою в освоєнні і надає мінімальний поріг входження, котрий нижче, ніж у того ж Python.
- PHP тісно пов'язаний з HTML і використовується на сервері для створення динамічних веб-сторінок. Окрім цього, мова має низку вбудованих інструментів для роботи із запитами від веб-сторінок та маршрутизацією.
- Мова підтримує безліч баз даних та їх СКБД, маючи вже вбудовані інструменти для роботи з ними.

- Якщо продукт, що розробляється, не передбачає складних обчислень, а лише орієнтується на базові завдання на кшталт роботи з базами даних, обробка запитів, генерація HTML розмітки, то в такому випадку мова надає цілком високу швидкість роботи.

MySQL — реляційна система управління базами даних, що є однією з найпопулярніших у світі і використовується від простих сервісів, як Wikipedia і закінчуючи високонавантаженими платформами на зразок Facebook. На підтвердження популярності можна надати статистику, зібрану сервісом DB-Engines [17].

На рисунку 2.7 зображено діаграму з 10 найпопулярніших баз даних на грудень 2024 року. Система балів популярності, або рейтинг підраховується сервісом DB-Engines з урахуванням різних метрик.

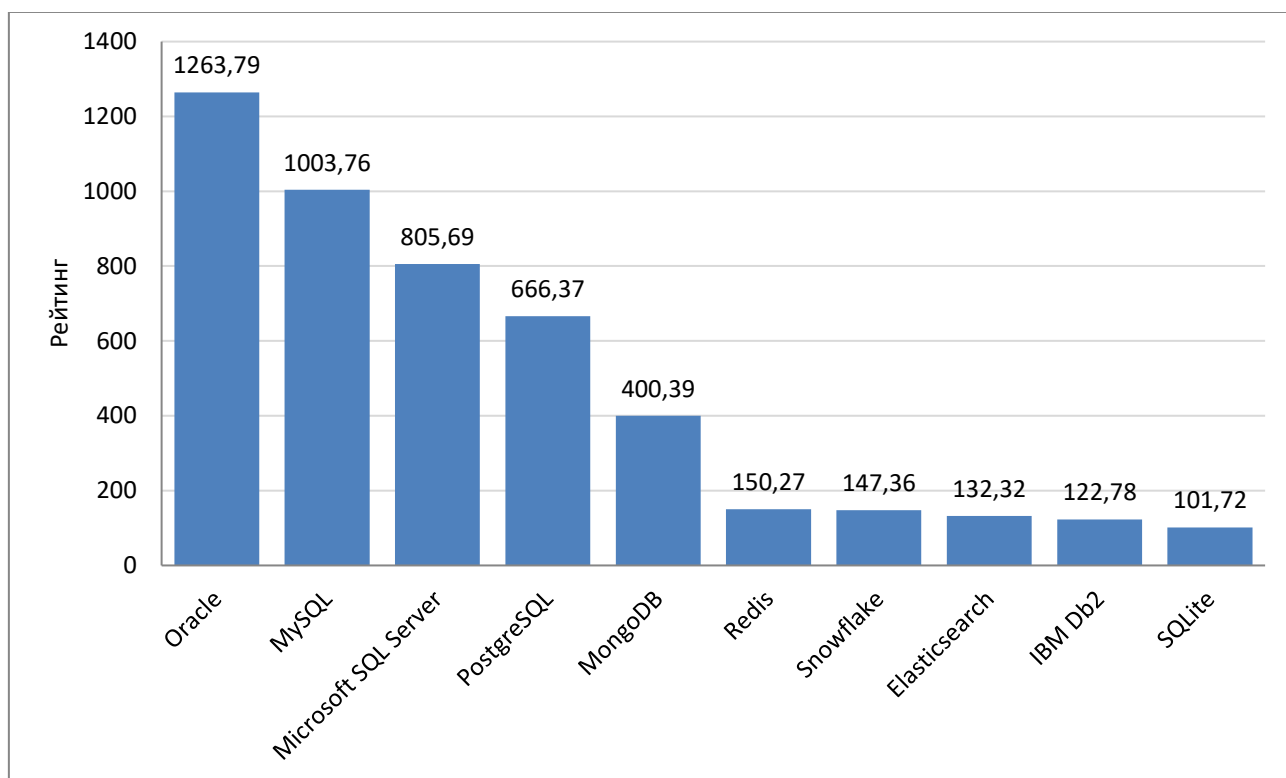


Рисунок 2.7 — Діаграма найпопулярніших баз даних на грудень 2024 року

Важливим фактором популярності MySQL є її надійність та хороша оптимізація, що дозволяє впевнено працювати з великими обсягами даних у рамках малих та середніх за складністю проектах. Ця СКБД є сумісною з усіма

відомими операційними системами. Будучи безкоштовною, простою в освоєнні та легко інтегрованою з більшістю мов програмування, даний інструмент є явним претендентом на використання.

OpenServer — це портативне програмне середовище, створене спеціально для веб-розробників. Є гнучкішим аналогом Denwer або Xampp з більшою функціональністю.

Notepad++ — вільний текстовий редактор, що має підсвічування синтаксису, згортання блоків коду, менеджер проектів і потужний пошук, який дозволяє використовувати регулярні вирази.

Figma — це графічний інструмент для дизайну та прототипування. Головним плюсом даного продукту є відображення стилів з точними розмірами, координатами та відступами елементів, що дозволяє легко перенести дизайн у верстку.

2.5 Рішення з технічного забезпечення

На основі вимог до програмного продукту та рішень щодо використання програмного забезпечення можна сформулювати вимоги до технічних характеристик пристроїв для роботи проекту.

Оскільки продукт, що розробляється, є веб-додатком і, спираючись на описані вище вимоги, можна зробити висновок, що на сайті не будуть проводитися складні обчислення та/або не буде задіяна відеокарта як окремий обчислювальний пристрій. для запуску браузера Google Chrome 80.0.

Апаратні вимоги до персонального комп'ютера:

- Процесор intel Pentium 4 чи пізнішої версії;
- Об'єм оперативної пам'яті 1024 Мб та більше;
- Відеопам'ять щонайменше 64 Мб.

Вимоги до операційної системи:

- Windows: версія не нижче 7;
- Linux: Ubuntu 20.04 та вище, Debian 10 і вище;

- macOS: не нижче 10.10;
- Android: версія не нижче 4.4;
- IOS: версія не нижче 12.

Оскільки для обробки запитів клієнтів на сайті та виконання операцій із базою даних використовується віддалений комп'ютер (сервер), для коректної роботи висуваються такі системні вимоги:

- Об'єм жорсткого диска:
 - 10 Гб для бази даних та системних файлів;
 - 4.5 Гб + 15% з розрахунку на одну годину контенту (відеоматеріал та ресурси).
- Оперативна пам'ять:
 - 16 Гб оперативної пам'яті для роботи системи;
 - 120Мб з розрахунком на одного користувача.
- Процесор з тактовою частотою не менше 3 ГГц та 4 ядрами з підтримкою Hyper-Threading (з розрахунком на обробку до 10 одночасних користувачів).

2.6 Висновки до розділу 2

В результаті розробки технічного проекту інформаційної системи онлайн-кінотеатру були розроблені наступні рішення, а саме:

- Спроековано діаграму варіантів використання;
- Спроековано концептуальну модель бази даних;
- Спроековано логічну модель бази даних;
- Спроековано фізичну модель бази даних;
- Спроековано блок-схему взаємодії користувача з сайтом.

Також на основі наявних даних були розраховані технічні вимоги для підтримки системи та її використання.

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН КІНОТЕАТР

У ході виконання роботи було розроблено веб-додаток «Онлайн кінотеатр». Dodatok складається з кількох веб-сторінок. До основних сторінок можна віднести:

- Головну сторінку;
- Сторінку з відеоплеєром, що відповідає за перегляд відеоматеріалу;
- Сторінки зі списком картин.

До додаткових сторінок можна віднести ті, котрі розширюють функціональність сайту й не є обов'язковими для отримання кінцевого досвіду. До таких сторінок можна віднести:

- Сторінку зі списком кращих картин;
- Сторінки акторів, режисерів та інших осіб, причетних до тієї чи іншої картини.

Примітка: оскільки деякі сторінки мають висоту, більшу за висоту екрану, у такому випадку буде надано макет сторінки, а далі кожен елемент окремо.

3.1 Головна сторінка

Головна сторінка складається з кількох логічних блоків:

- Навігаційна панель;
- Список останніх картин, що надійшли на сайт (це можуть бути як повнометражні фільми, сезони серіалів, так і зовсім епізод серіалу);
- Декілька групованих списків картин. Картини можуть бути груповані за жанром, можуть надавати список останніх переглянутих користувачем картин або це можуть бути кращі картини за версією сайту;
- Блок, що представляє всі картини на сайті.

На рисунку 3.1 надано макет головної сторінки сайту.

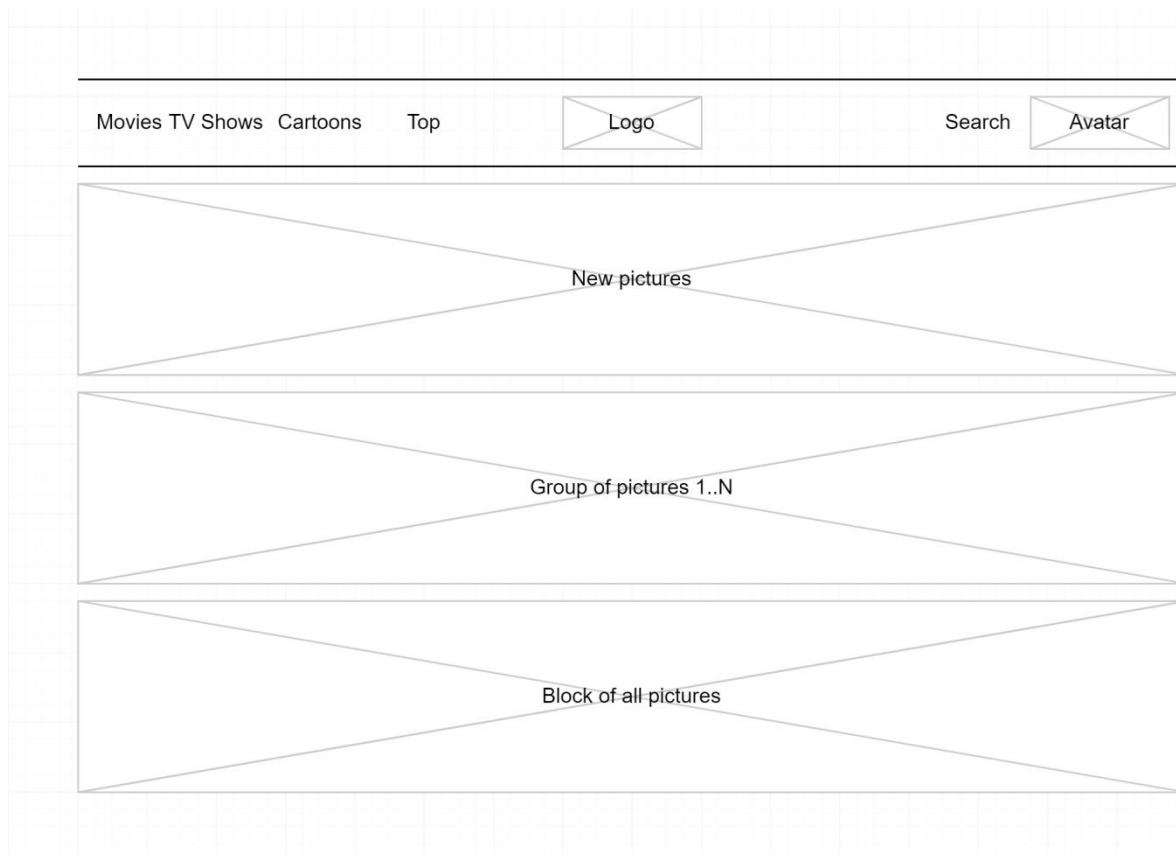


Рисунок 3.1 — Макет головної сторінки сайту

На рисунку 3.2 представлено зображення навігаційної панелі. Навігаційна панель складається із семи кнопок:

- Movies: відкриття сторінки для пошуку фільмів;
- TV Shows: відкриття сторінки для пошуку серіалів;
- Cartoons: відкриття сторінки для пошуку мультиплікації;
- Top: відкриття сторінки зі списком найкращих картин за версією сайту;
- Логотип CINEMA: відкриття головної сторінки сайту;
- Зображення лупи: відкриття загального пошуку сайту;
- Зображення аватара: відкриття кабінету користувача.



Рисунок 3.2 —Зображення навігаційної панелі

На рисунку 3.3 представлений один із елементів списку останніх завантажених на сайт картин. Список є слайдером — контейнером, що зберігає в собі кілька подібних елементів. Цей слайдер можна перегортати, що зображено рисунку 3.4.

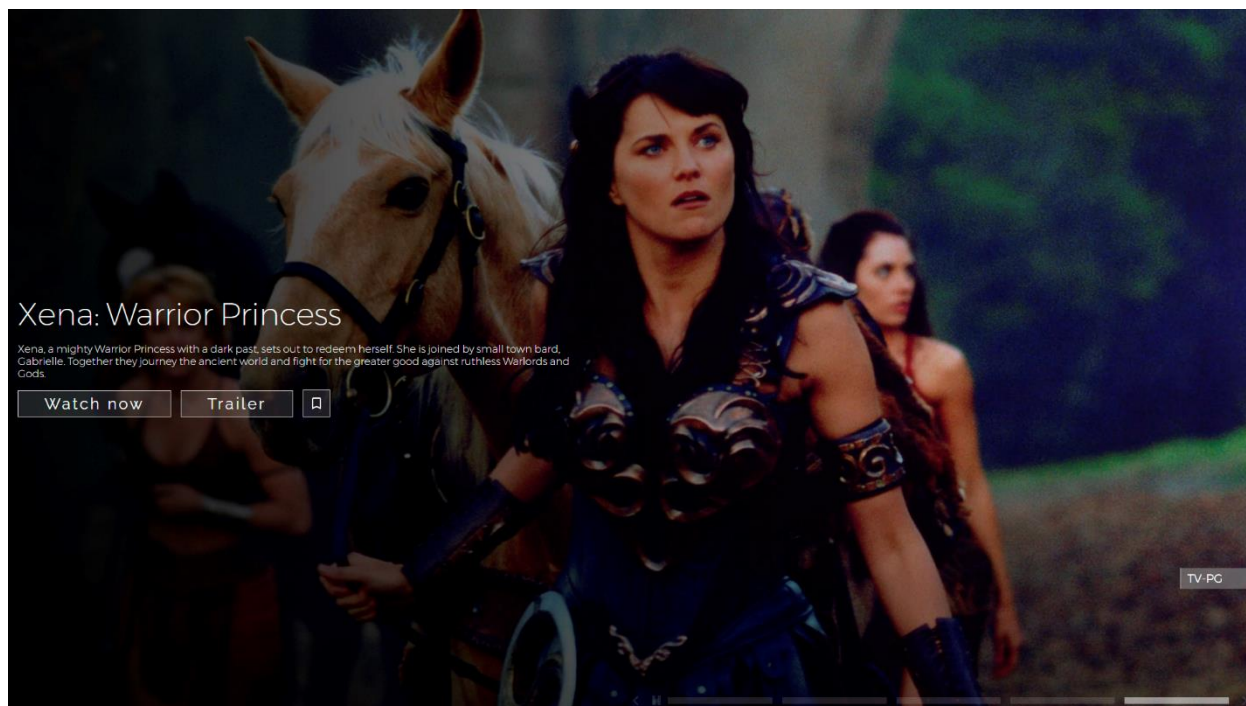


Рисунок 3.3 — Елемент списку останніх завантажених картин

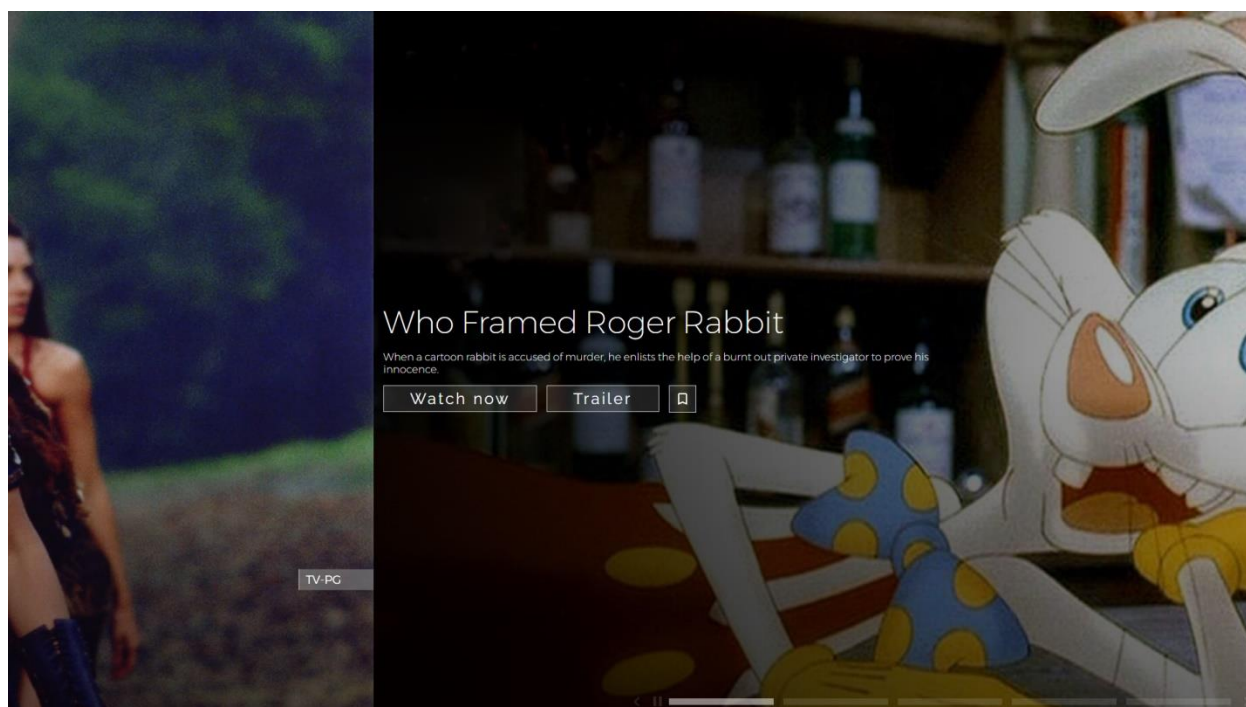


Рисунок 3.4 — Перегортання слайдера

На рисунку 3.5 представлено зображення списку картин, який групований за якоюсь ознакою. В даному випадку картини становлять список останніх переглянутих користувачем. Слід зазначити, що кількість списків та їх типи можуть змінюватися під час кожного оновлення сторінки.

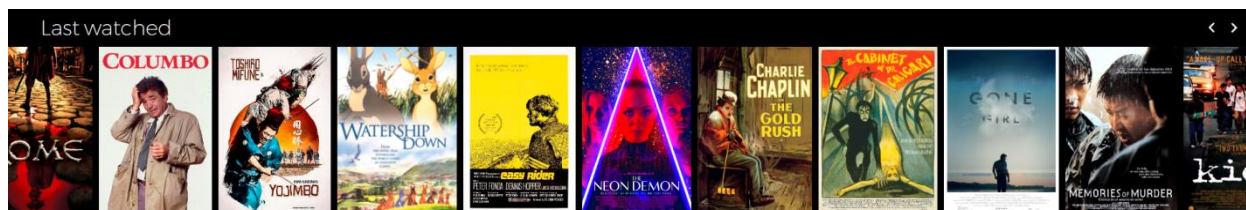


Рисунок 3.5 — Групований список

Слід зауважити, що кожна картина представлена у вигляді картки, на лицьовій стороні якої є постер, але при наведенні на неї можна побачити більш детальну інформацію. При натисканні на картку можна перейти на сторінку картини.

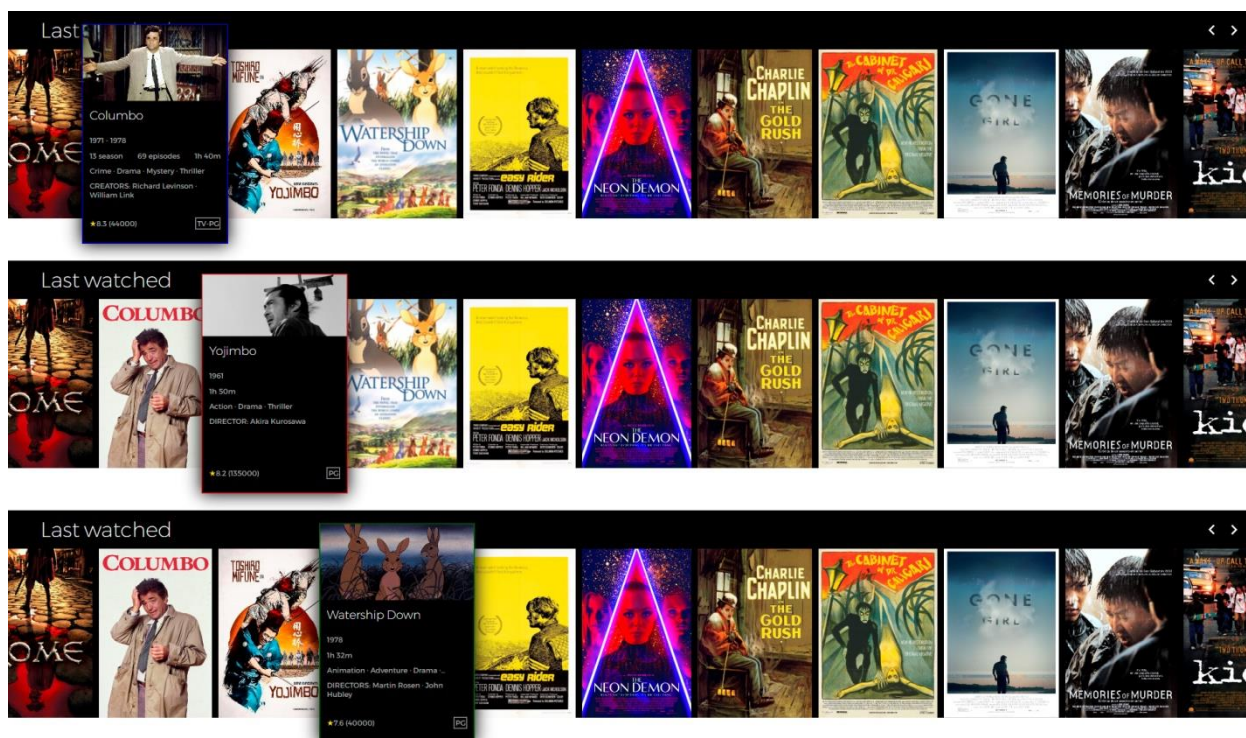


Рисунок 3.6 — Відображення деякої інформації про картину, наводячи на її картку

На рисунку 3.7 представлено зображення блоку з усіма картинами на сайті. Усі картини розбиті на сторінки.

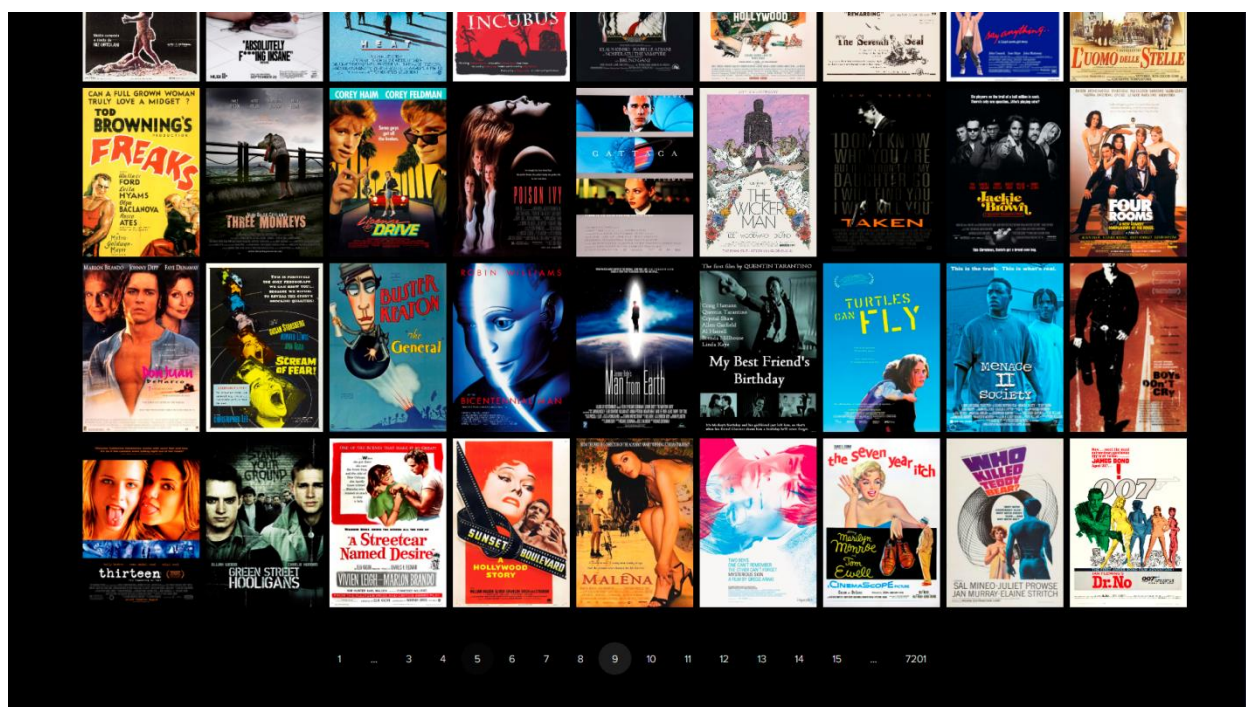


Рисунок 3.7— Блок, що відображає всі картини, наявні на сайті

3.2 Сторінка картини

Сторінка кожної картини складається з кількох логічних блоків:

- Навігаційна панель;
- Блок із основною інформацією про картину;
- Блок зі списком акторів;
- Блок, який представляє систему коментарів до картини;
- Блок із додатковою інформацією про картину.

На рисунках 3.8 та 3.9 представлені макети сторінки картини з відкритим із запущеним відеоплеєром і без нього.

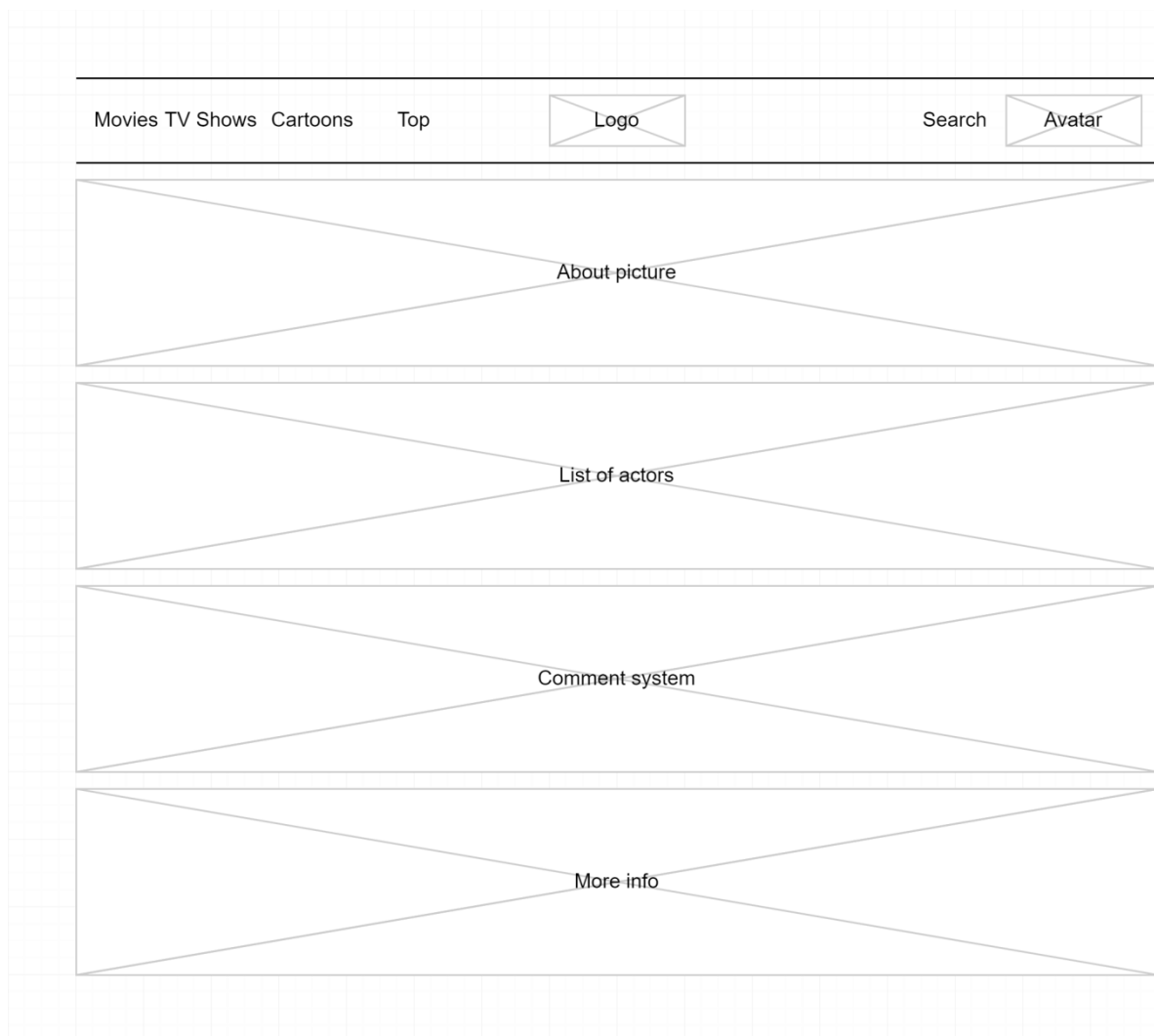


Рисунок 3.8 — Макет сторінки картини без запущеного відеоплеєра

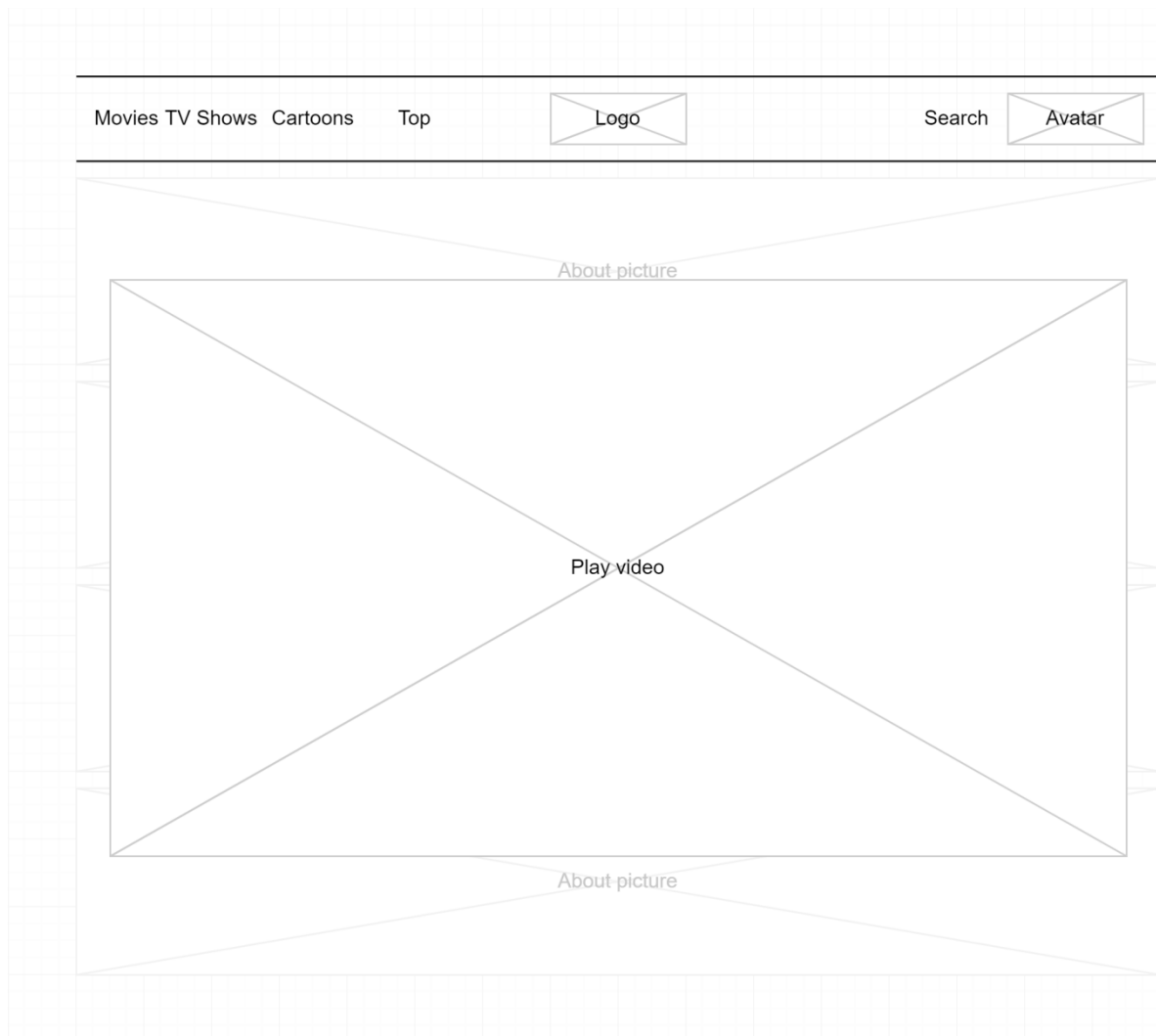


Рисунок 3.9 — Макет сторінки картини із запущеним відеоплеєром

На рисунку 3.10 подано зображення блоку з відображенням основної інформації про картину. Крім відображення інформації, у блоці є кнопка «Watch now», при натисканні на яку відкривається відеоплеєр.

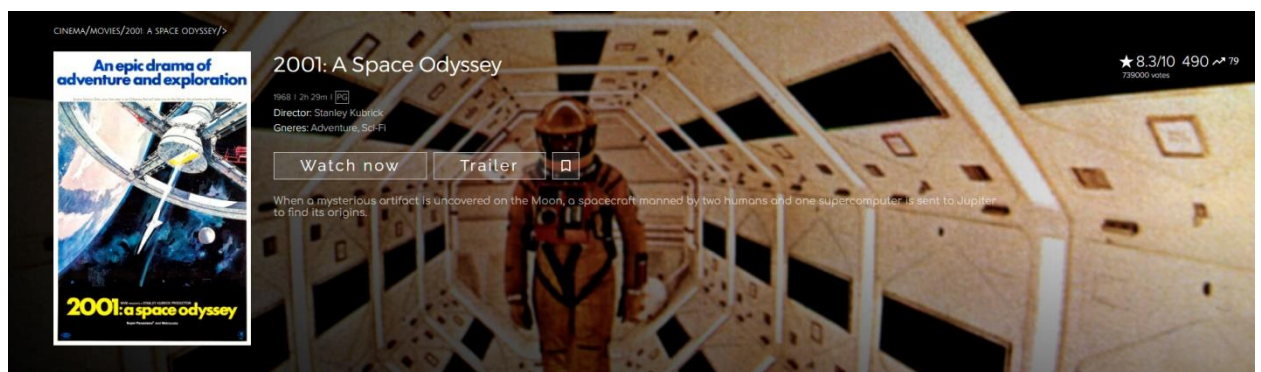


Рисунок 3.10 — Блок із основною інформацією про фільм

На рисунку 3.11 надано відображення блоку зі списком акторів. Список подано у вигляді слайдера.

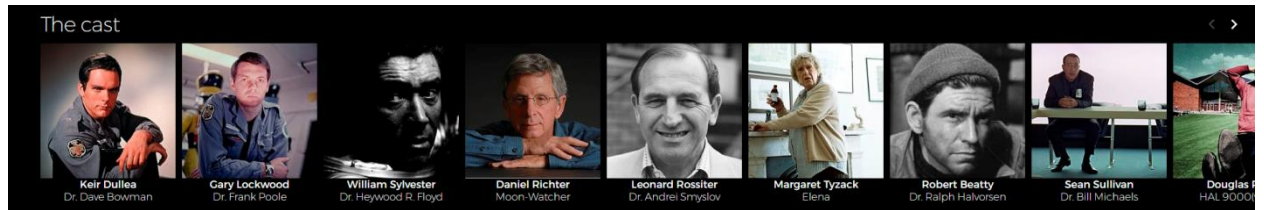


Рисунок 3.11 — Блок зі списком акторів

На рисунку 3.12 представлено відображення системи коментарів. Система коментарів складається із трьох логічних блоків:

1. Область для введення та відправлення коментарів;
2. Область із відображенням дерева коментарів;
3. Пагінація коментарів.

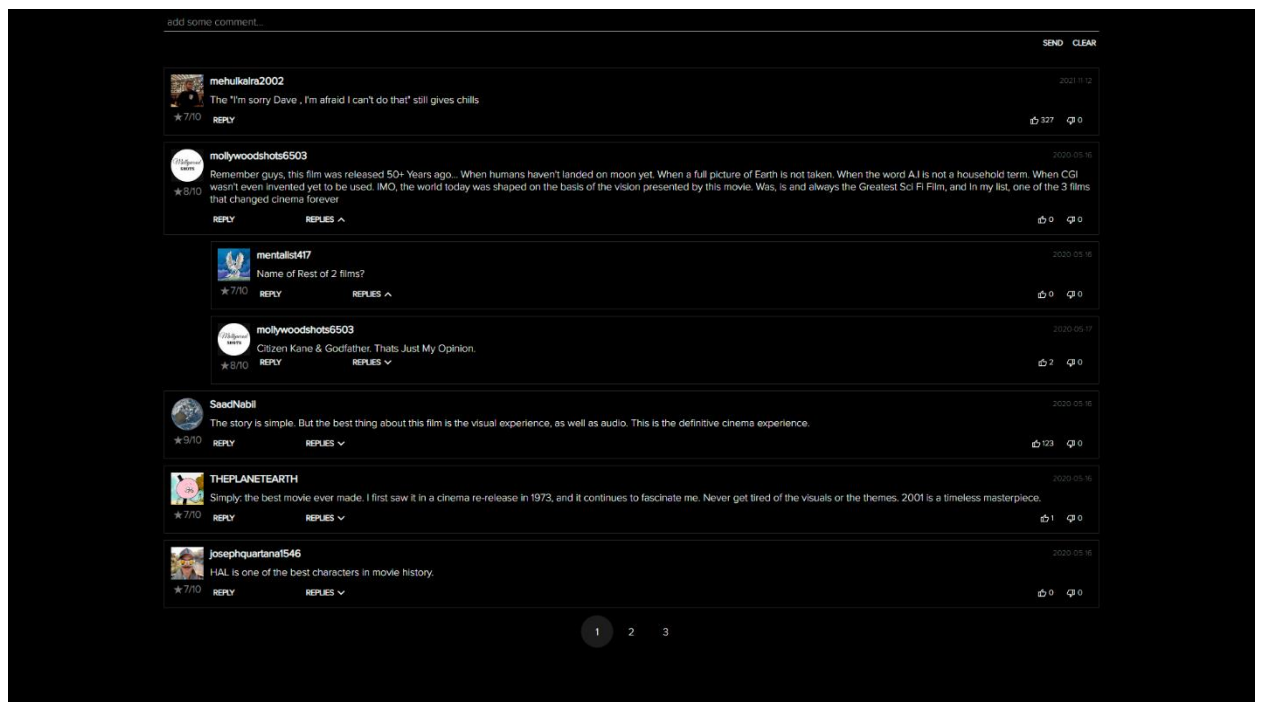


Рисунок 3.12 — Система коментарів

На рисунку 3.13 наведено відображення блоку з додатковою інформацією про картину.



Рисунок 3.13— Додаткова інформація

На рисунку 3.14 представлено відображення блоку з відеоплеєром, що відкривається поверх сторінки.

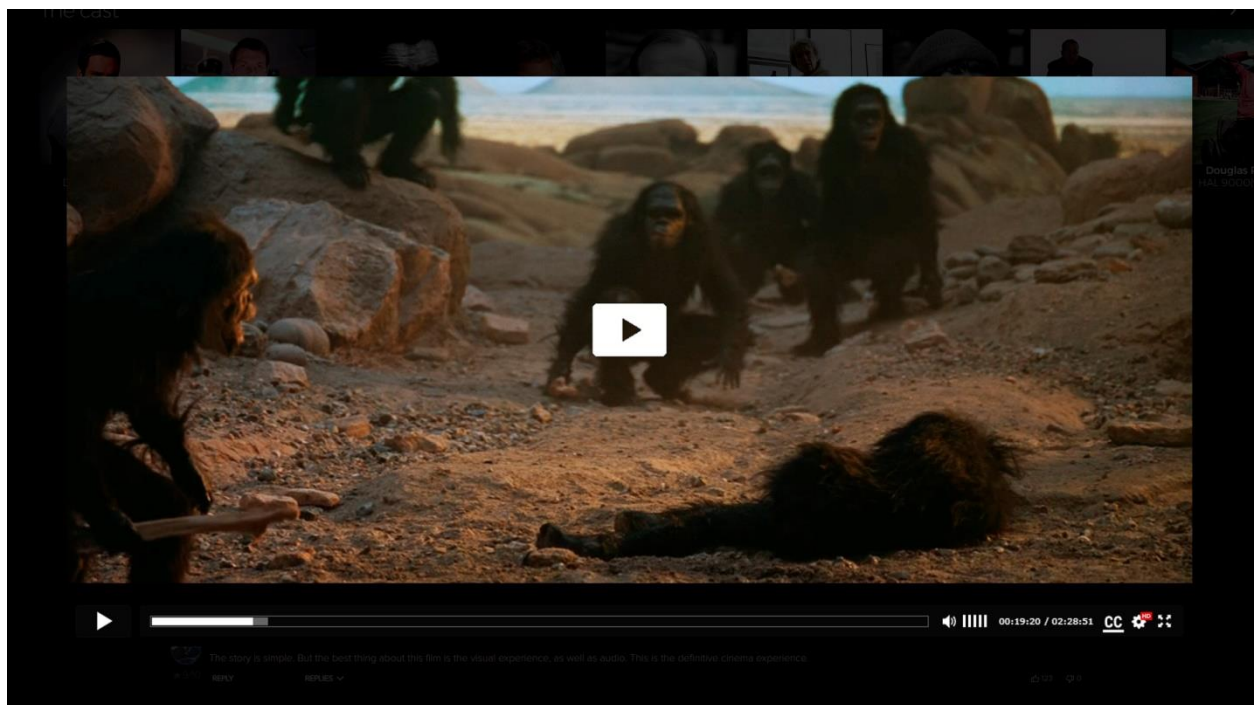


Рисунок 3.14 — Відеоплеєр

3.3 Список найкращих картин

Для відкриття сторінки з відображенням списку кращих картин за версією сайту слід натиснути кнопку «Топ» на навігаційній панелі. На рисунку 3.15 представлений макет сторінки з кращими картинами.

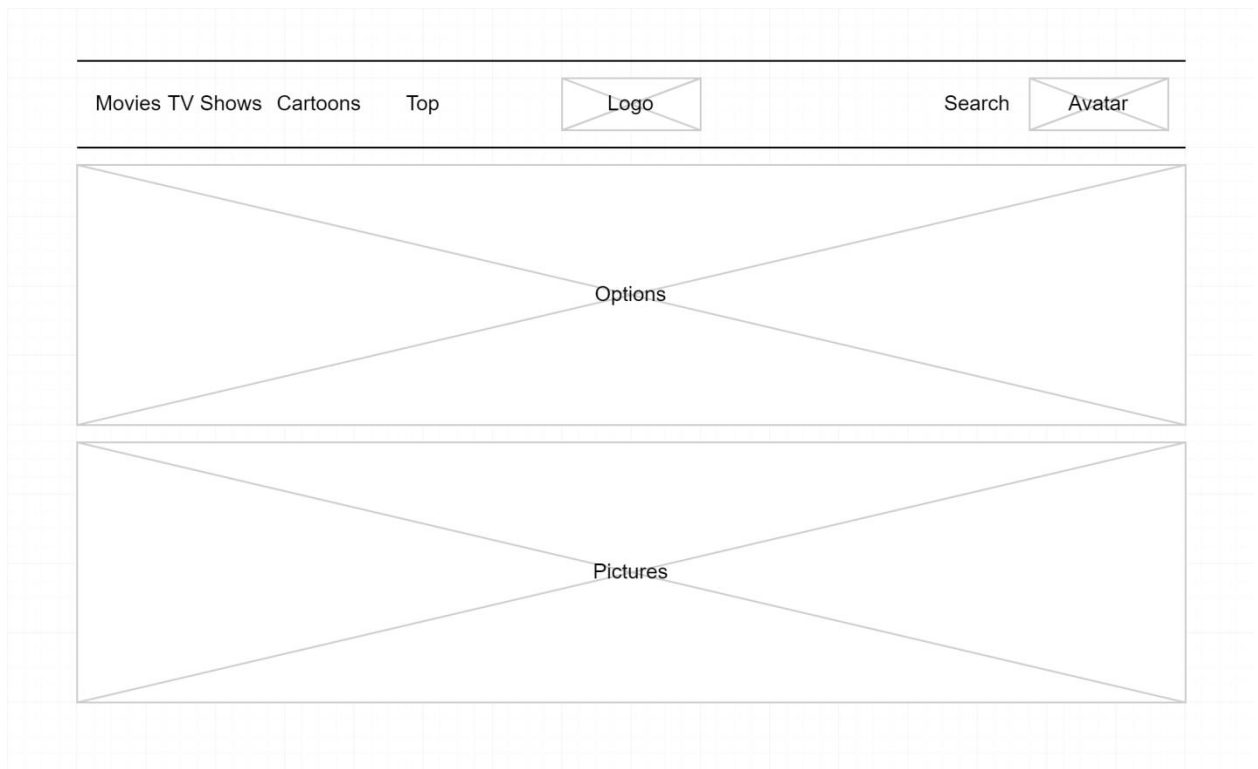


Рисунок 3.15 — Макет сторінки зі списком найкращих картин

На рисунку 3.16 представлено зображення сторінки зі списком найкращих картин. У даному випадку відображено список кращих фільмів, але така ж функціональність надана і для серіалів та мультиплікації.

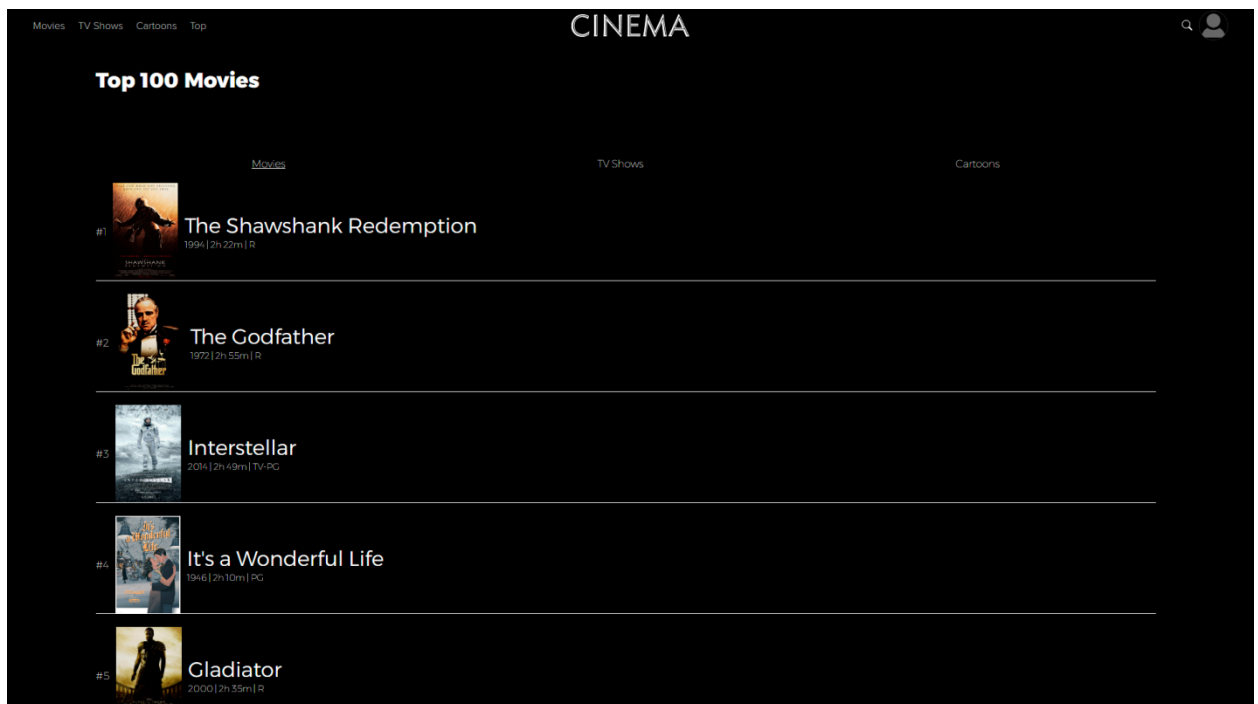


Рисунок 3.16 — Сторінка зі списком кращих картин

3.4 Пошук за категоріями

При натисканні на одну з трьох кнопок на навігаційній панелі — «Movies», «TV Shows», «Cartoons» можна перейти на сторінку пошуку картини того чи іншого типу. На рисунку 3.17 представлено макет сторінки пошуку.

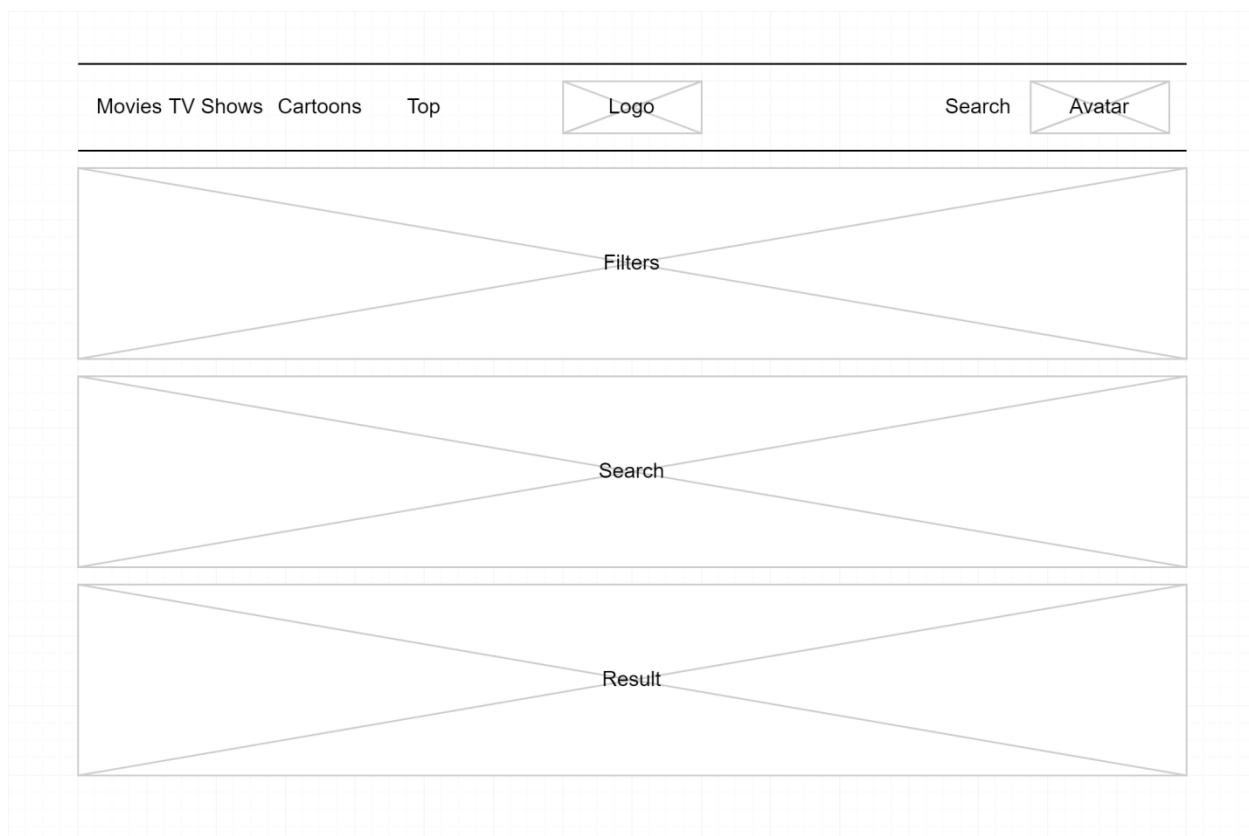


Рисунок 3.17 — Макет сторінки пошуку

На рисунку 3.18 показано відображення блоку з фільтрами, які можна застосувати до пошуку.

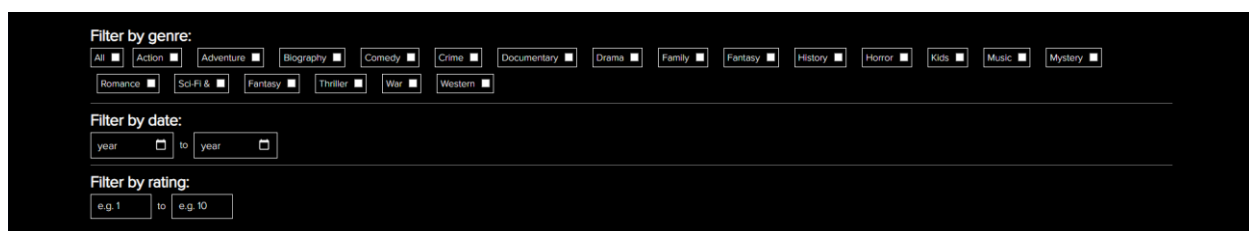


Рисунок 3.18 — Фільтри пошуку

На рисунку 3.19 подано відображення блоку з пошуком. Пошук здійснюється за ключовими словами.



Рисунок 3.19 — Пошук

На рисунку 3.20 подано зображення сторінки з пошуком. На цьому зображенні відображається пошук для типу «Movies», але він ідентичний і для інших категорій.

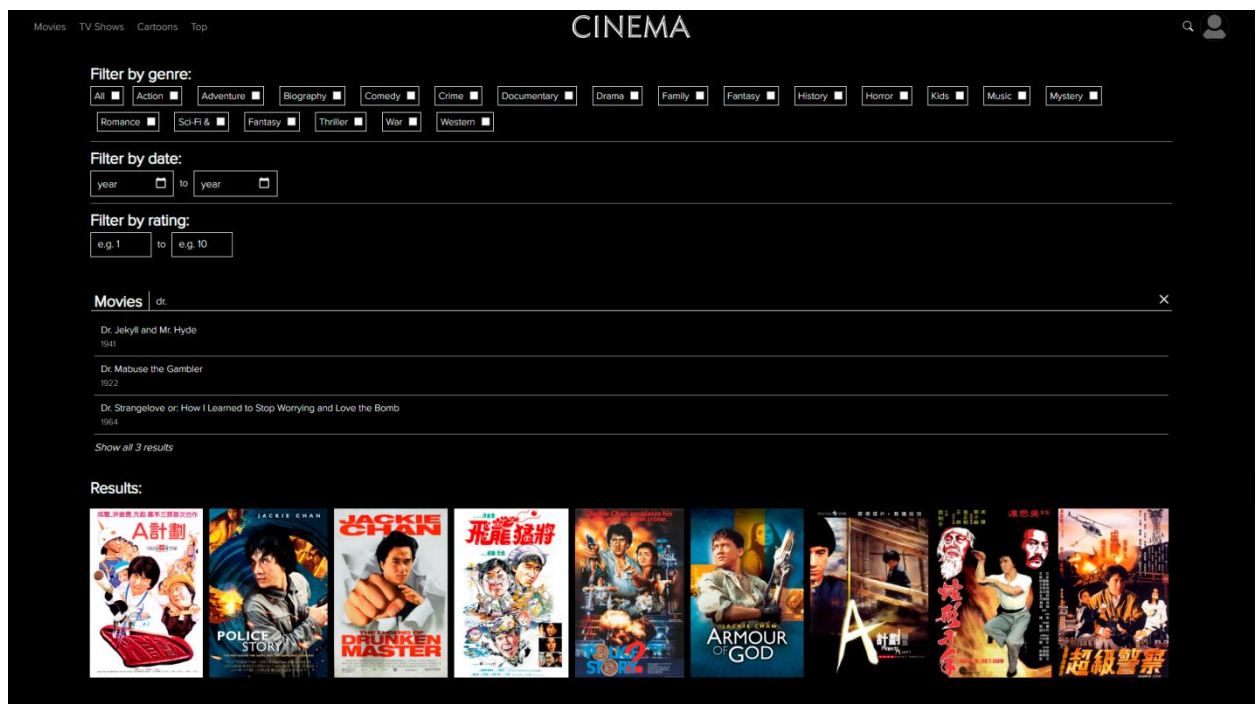


Рисунок 3.20 — Сторінка пошуку для категорії «Movies»

3.5 Сторінка учасника

На рисунку 3.21 представлено макет сторінки учасника зйомки картини.

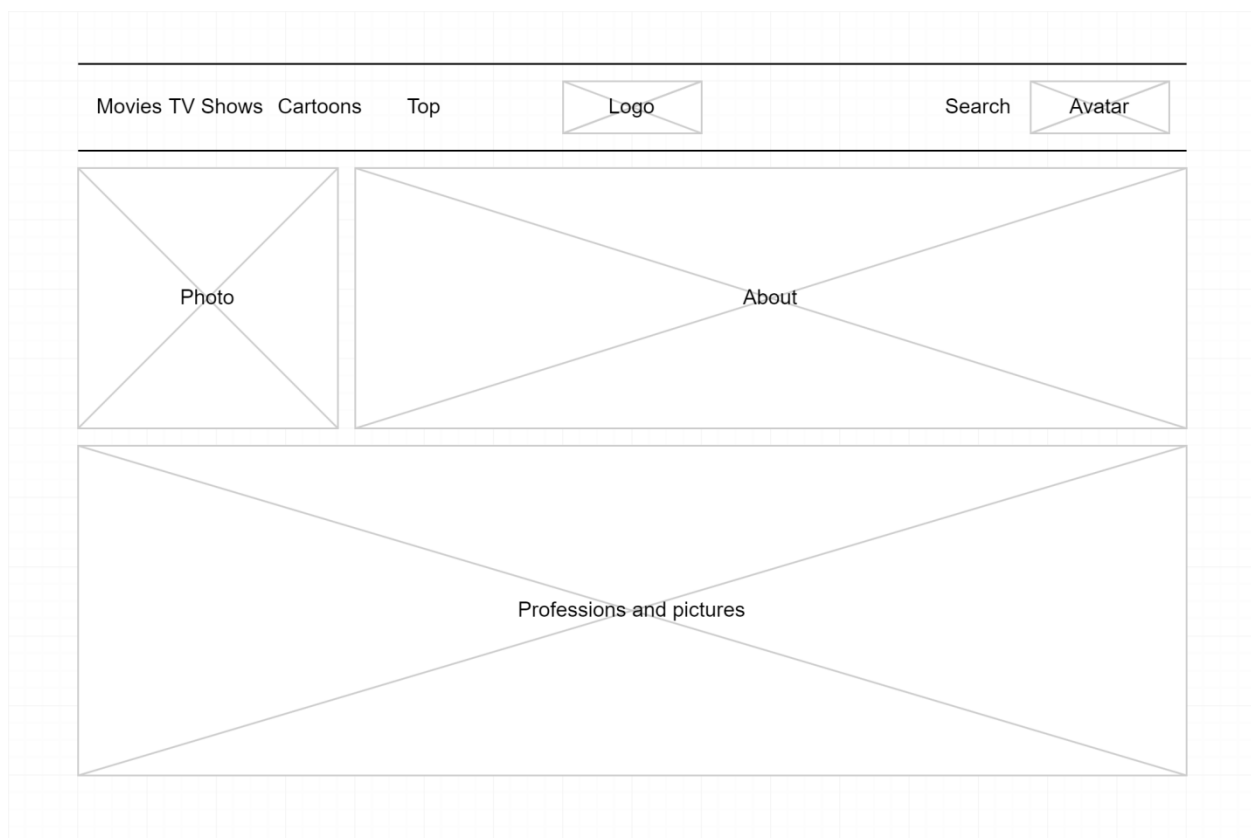


Рисунок 3.21 — Макет сторінки учасника

На рисунку 3.22 представлено зображення сторінки учасника, де розміщуються зображення, деяка особиста інформація та список робіт, які доступні на сайті.

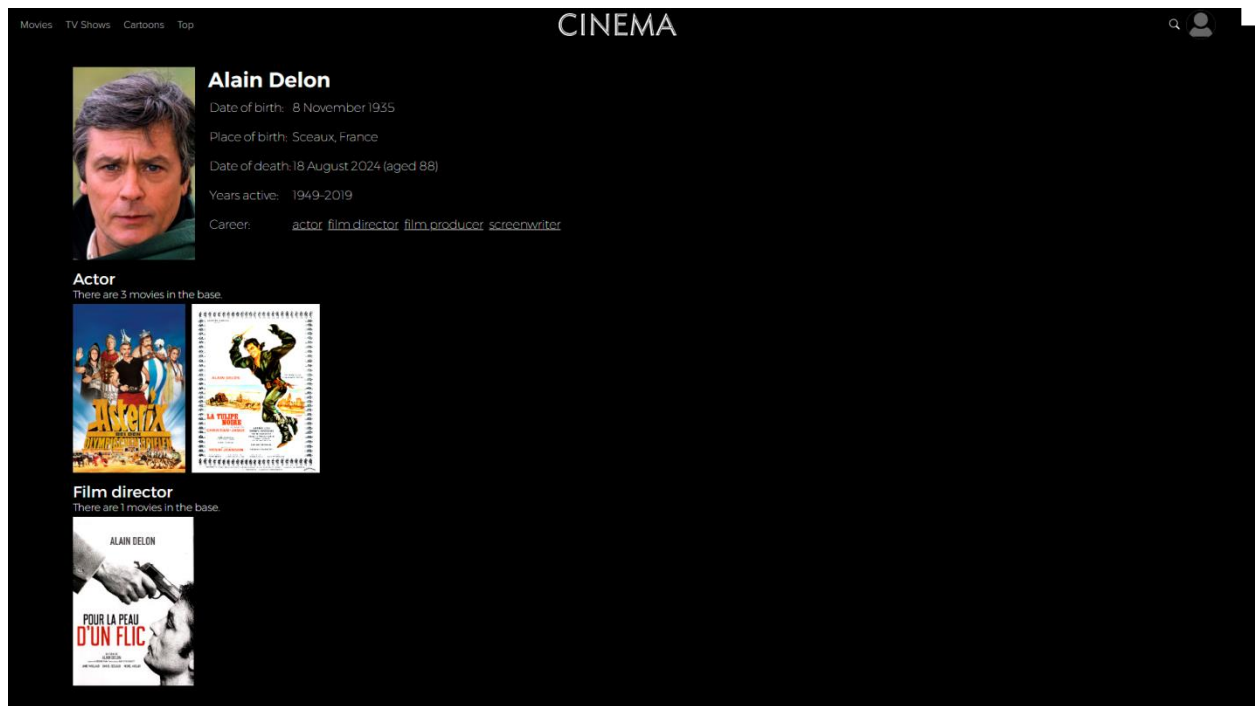


Рисунок 3.22 — Сторінка учасника

3.6 Діаграма розгортання системи

Діаграма розгортання — це тип UML-діаграми, яка показує архітектуру виконання системи, включаючи такі вузли, як апаратні або програмні середовища виконання, а також проміжне програмне забезпечення, яке їх з'єднує. На рисунку 3.23 представлено діаграму розгортання системи онлайн-кінотеатр.

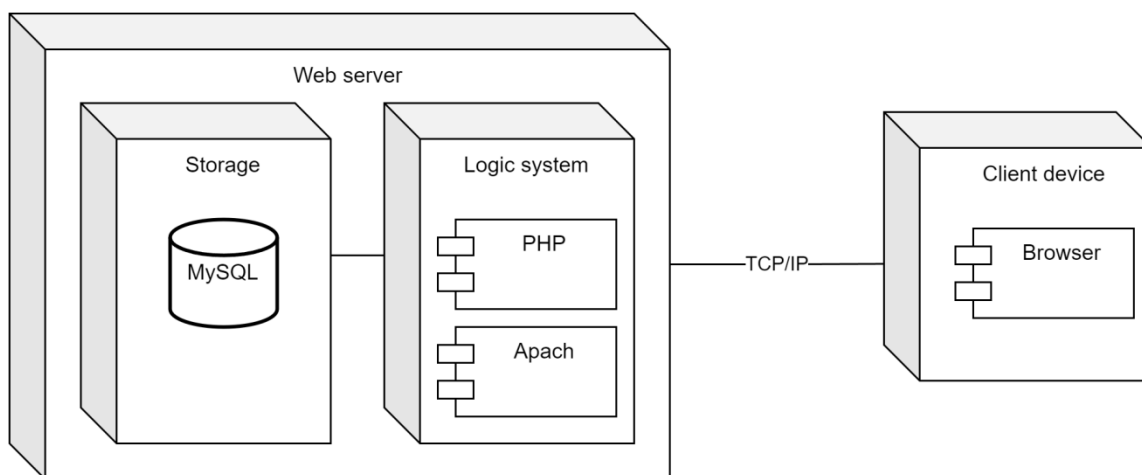


Рисунок 3.23 — Діаграма розгортання

3.7 Висновки до розділу 3

В результаті виконання кваліфікаційної роботи було розроблено веб-додаток онлайн-кінотеатр. Завдання проекту — створення інформаційної системи для автоматизації бізнес-процесів стрімінгового сервісу з метою підвищення ефективності роботи та надання позитивного користувацького досвіду.

З точки зору технічної вигоди, розробку можна обґрунтувати через:

- Автоматизація бізнес-процесів. Розробка інформаційної системи дозволить автоматизувати такі аспекти як управління медіа контентом з боку адміністрації, пошук потрібного контенту, доступ до перегляду, ознайомлення з базовою інформацією з боку користувача.
- Управління даними. Засоби керування базою даних MySQL та SQL дозволять ефективно зберігати, оновлювати та отримувати дані з максимальною продуктивністю.
- Інтерфейс користувача. Інтуїтивно зрозумілий і орієнтований на користувача дизайн додатка дозволить легко і невимушено виконувати ті чи інші дії, тим самим підвищуючи досвід користувача.

- Масштабованість системи. Інформаційна система спроектована з урахуванням можливості збільшення обсягу користувачів та розширення функціональності. Цей аспект дозволить підтримувати стабільність роботи під час зростання навантаження.

З точки зору економічної вигоди, розробку можна обґрунтувати через:

- Конкурентоспроможність. На сьогоднішній день, коли у світі кіно стрімінг є лідером з надання медіа-контенту, стратегічно важливе для галузі бізнесу входження в цей сегмент. Інформаційна система дозволить сервісу залишатися конкурентоспроможним над ринком.

- Зменшення витрат на підтримку. Автоматизація процесів, таких як керування медіаконтентом, скоротить необхідність у ручній обробці даних, що знизить витрати на технічну підтримку.

- Аналіз даних та прийняття рішень. Інформаційна система дозволить збирати та аналізувати дані про користувальницьку активність та переваги. Це забезпечить можливість адаптуватися під користувача та розробляти нові стратегії.

- Відсутність фізичних обмежень. На відміну від фізичних кінотеатрів, стрімінговий сервіс не обмежений місткістю залу чи географічним розташуванням. Це дозволяє обслуговувати необмежену кількість користувачів одночасно, забезпечуючи високий рівень доступу до контенту.

- Оптимізація трудовитрат. Для функціонування стрімінгового сервісу потрібно набагато менше працівників, ніж для фізичного кінотеатру. Більшість операцій автоматизована, що дозволяє знизити витрати на персонал та оптимізувати бізнес-процеси.

РОЗДІЛ 4. РОЗРАХУНОК І ОЦІНКА ЕКОНОМІЧНОГО ЕФЕКТУ ВІД РОЗРОБКИ

Під час виконання дипломної роботи необхідно було розробити веб-додаток онлайн-кінотеатр. Розрахунок економічної ефективності дозволить обґрунтувати доцільність розробки.

4.1 Розрахунок витрат на створення програми

Керуючись тим, що, згідно з сервісом work.ua [18], середня заробітна плата Full Stack розробників програмного забезпечення складає 53500 грн/міс, вартість кіловат-години електроенергії складає 4,32 грн, а вартість Інтернету 150грн, розрахуємо вартість розробки програми.

Вартість програми розраховується за формулою:

$$C_{\text{пр}} = Z_e * T_z + (Z_{\text{кп}} * Z_{\text{п}} + B_{\text{І}}) * T_{\text{м}}, \text{ де} \quad 4.1.1$$

$T_{\text{м}}$ — тривалість розробки, міс.;

T_z — тривалість розробки, змін.;

$Z_{\text{кп}}$ — загальна кількість персоналу (розробників)

$Z_{\text{п}}$ — заробітна плата персоналу, грн.;

Z_e — витрати на електроенергію за зміну;

$B_{\text{І}}$ — вартість Інтернету

Витрати на електроенергію розраховуються за формулою:

$$Z_e = P_{\text{бж}} * B_e * \Gamma_z, \text{ де} \quad 4.1.2$$

$P_{\text{бж}}$ — потужність блоку живлення робочої машини

B_e — вартість електроенергії

Γ_z — кількість годин витрачених на розробку за зміну

Тривалість годин роботи за зміну складає 8 годин. Вартість кіловат-години електроенергії 4,32 грн. Потужність блоку живлення становить 0,4КВт.

$$З_е = 0,4 * 4,32 * 8 = 13,824\text{грн} \quad 4.1.3$$

Кількість календарних днів ($K_{\text{кл}}$) 30, вихідних (B) 8, у такому разі формула розрахунку кількості днів витрачених на розробку за місяць становить:

$$K_{\text{др}} = (K_{\text{кл}} - B) * T_{\text{м}}. \quad 4.1.4$$

Оскільки на розробку було витрачено 1 місяць у такому разі $K_{\text{др}}$ становить:

$$K_{\text{др}} (30 - 8) * 1 = 22 \text{ дні} \quad 4.1.5$$

Маючи всі змінні, загальна вартість розробки коштує:

$$C_{\text{пр}} = 13,824 * 22 + (1 * 53500 + 150) * 1 = 53954.128\text{грн} \quad 4.1.6$$

4.2 Розрахунок економічної ефективності

Оскільки програмний продукт не розроблявся з метою отримання економічної вигоди і необхідні функції у додаток не закладені, то в такому випадку визначати економічну ефективність не доцільно. Але якщо, як приклад, провести аналогію з класичним кінотеатром, де 50% від прибутку від продажу квитків отримує кіностудія, від 20 до 30% йде на податкові виплати, а зазвичай 10-15% — на оплату праці робітникам і утримання обладнання, то, якщо кіно виявилось успішним, то мінімум 5% чистого доходу отримує сам кінотеатр. У такому випадку для того, щоб кіно могло дозволити придбати програмне забезпечення і при цьому не вийти за рамки бюджету, з продажу квитків кінотеатр повинен заробити приблизно 2,160,000 грн. Варто зауважити, що наведені вище відсоткові значення є приблизно середніми та можуть відрізнятися від регіону, сезону і самого кінотеатру.

4.3 Висновки до розділу 4

Внаслідок роботи над розділом було проведено розрахунки собівартості розробленого продукту, що склало 53954.128грн за результатом розробки

одного місяця. Також, ґрунтуючись на приблизній інформації, котра доступна, кінотеатр, який вирішить купити програмний продукт, при цьому не вийшовши за рамки бюджету, повинен буде продати квитків на суму приблизно 2,160,000 грн.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи була виконана поставлена мета та спроектовано веб-додаток — онлайн-кінотеатр. Поставлена мета полягає у підвищенні ефективності роботи веб-додатку онлайн-кінотеатр.

Для досягнення мети кваліфікаційної роботи був проведений аналіз предметної області та досліджено технології розробки, так і вже готові аналоги проекту. Були виділені основні концептуальні класи й на базі цього створено блок-схеми й діаграми:

- діаграма варіантів використання;
- концептуальна модель;
- логічна модель даних;
- фізична модель даних.

Проаналізувавши предметну область і визначившись з функціональністю системи, було розроблено логіку та інтуїтивно зрозумілий інтерфейс користувача. Також було розроблено документацію до проекту, де описані основні аспекти проекту.

У ході виконання проекту було покращено навички у frontend, навички роботи з мовами JavaScript, PHP, а також навички у проектуванні баз даних.

Актуальність та новизна проекту полягають як у зручності пошуку відеоконтенту в рамках Інтернету, так і в навігації по сайту, що стало можливим завдяки інтуїтивно зрозумілому та зручному дизайну. Оскільки програма використовує адаптивний дизайн, перегляд фільмів та серіалів не обмежений певним місцем, що є однією з ключових функцій проекту. Легкий і зручний пошук, а також добірки, створювані додатком на основі дій користувача, роблять його досвід більш приємним і унікальним.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Statista. (2024). *Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2033*. Взято з <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>
- [2] Wikipedia. (2024). *RealNetworks*. Взято з <https://en.wikipedia.org/wiki/RealNetworks>
- [3] Coreteka. (2023). *Web Application Development Fundamentals*. Взято з <https://coreteka.com/blog/fundamentals-of-web-application-development>
- [4] MDN. (б.д.). *What is the difference between web page, website, web server, and search engine?*. Взято з https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/Pages_sites_servers_and_search_engines
- [5] Dr. Rafah M. Almuttairi. (б.д.). *What's the Difference Between the FrontEnd and Back-End?* Взято з https://www.uobabylon.edu.iq/eprints/publication_4_27425_1402.pdf
- [6] Slideshare. (2024). *Introduction to Frontend Web Development*. Взято з <https://www.slideshare.net/slideshow/introduction-to-frontend-web-development/267200649>
- [7] MDN. (б.д.). *HTML basics*. Взято з https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics
- [8] TechTerms. (б.д.). *CSS*. Взято з <https://techterms.com/definition/css>
- [9] GeeksforGeeks. (2024). *Introduction to JavaScript*. Взято з <https://www.geeksforgeeks.org/introduction-to-javascript/>
- [10] Anurag Kumar, Ravi Kumar Singh. (б.д.). *COMPARATIVE ANALYSIS OF ANGULARJS AND REACTJS*. Взято з <https://www.ijltet.org/journal/148051944230.1245.pdf>
- [11] Lemon School. (2023). *Що таке Vue.js?* Взято з <https://lemon.school/blog/shho-take-vue-js>
- [12] Vue.js. (б.д.). *Introduction*. Взято з <https://vuejs.org/guide/introduction.html>

- [13] David L. Quinlan. (б.д.). *Louis Aimé Augustin (Edmée Auguste) Le Prince : 1842 - 1890?*. Взято з https://webarchive.nla.gov.au/awa/20040408015126/http://www.acmi.net.au/AIC/LE_PRINCE_BIO.html
- [14] Timetoast. (б.д.). *Основні етапи розвитку кінематографу*. Взято з <https://www.timetoast.com/timelines/-4bc99724-2579-45c2-9c28-5425e1b46498>
- [15] IBM. (2023). *Use-case diagrams*. Взято з <https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=diagrams-use-case>
- [16] Oracle. (2024). *MySQL: Understanding What It Is and How It's Used*. Взято з <https://www.oracle.com/mysql/what-is-mysql>
- [17] DB-engines. (2024). *DB-Engines Ranking*. Взято з <https://db-engines.com/en/ranking>
- [18] WORK.ua (2024). Взято з <https://www.work.ua/ru/jobs-full+stack+developer/>

ДОДАТОК А

Інструкція користувача

Інструкція буде присвячена неосновним аспектам сайту, котрі є допоміжними функціями і без яких користувач може повноцінно взаємодіяти з сайтом.

В інструкції будуть описані функції, які не були згадані в основній частині роботи.

Авторизація

Авторизація користувача може бути виконана у будь-який момент перебування на сайті. Для виконання авторизації користувач повинен перейти на відповідну сторінку, натиснувши кнопку-аватар. На сторінці авторизації необхідно вибрати потрібну форму, заповнити її та натиснути на кнопку.

Слід зазначити, що наявність електронної пошти не є обов'язковою для реєстрації. За потреби її можна додати пізніше.

На рисунку А.1 показано кнопку переходу на сторінку авторизації. На рисунку А.2 представлено сторінку авторизації.

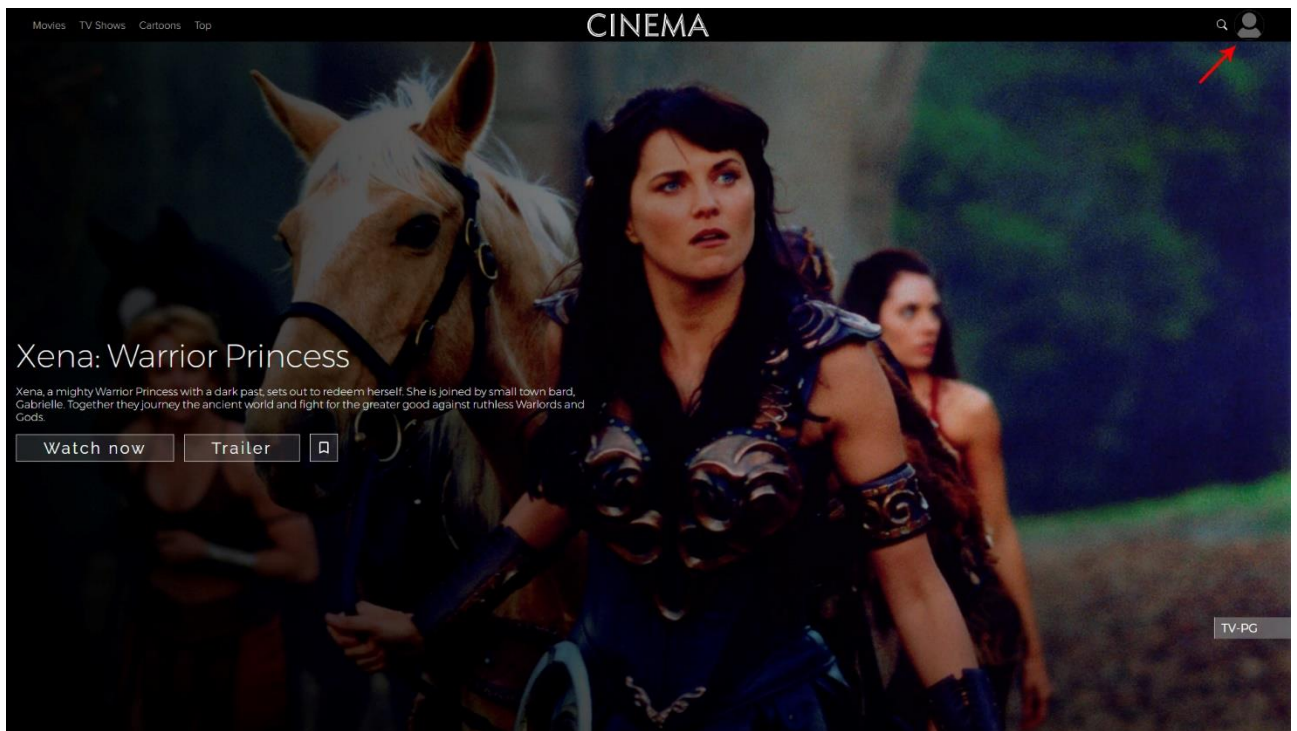


Рисунок А.1 — Кнопка для авторизації користувача

Log In

login or email

password

Sign Up

* login

* password

email

Рисунок А.2 — Сторінка авторизації

Сімейні групи

Однією з функцій, доступних користувачеві після авторизації є можливість перебувати в групах. Ці групи є способом об'єднання облікових

записів користувачів, що належать до одного кола осіб. Основна ідея цієї функції полягає у введенні батьківського контролю, який призначений для обмеження доступу дитячих облікових записів до певних типів контенту або до конкретних картин.

Важливо зазначити, що один користувач може перебувати лише в одній групі.

Користувач, який ще не є членом групи, побачить на відповідній сторінці форму, представлену на рисунку А.3. У верхній частині сторінки відображається його особистий ідентифікатор, а нижче форма для введення ідентифікатора нового члена групи.

Слід враховувати, що користувач, який запросив хоча б одного члена, автоматично стає адміністратором групи.

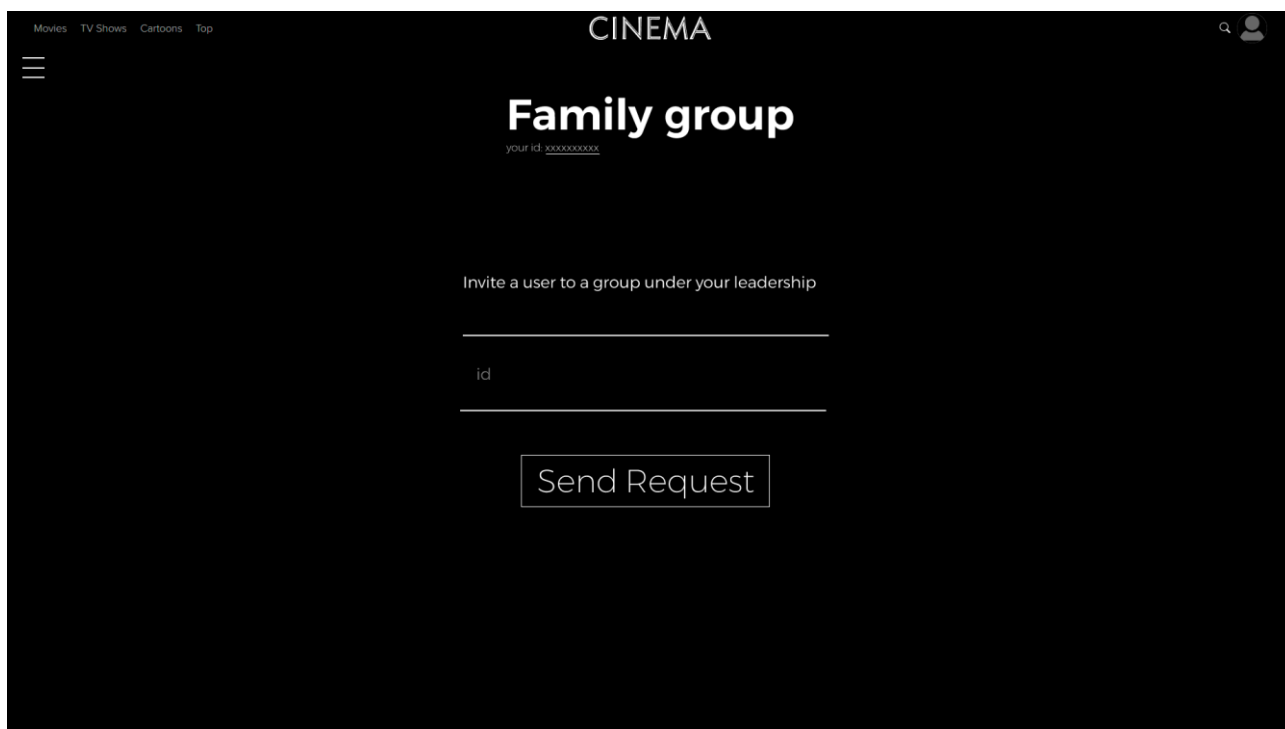


Рисунок А.3 — Сторінка із формою створення групи

Після того, як умовний користувач А запросив умовного користувача Б приєднатися до своєї групи, запрошений користувач Б на тій самій сторінці побачить повідомлення, представлене на зображенні А.4, з можливістю вибору: приєднатися до групи або відмовитися.

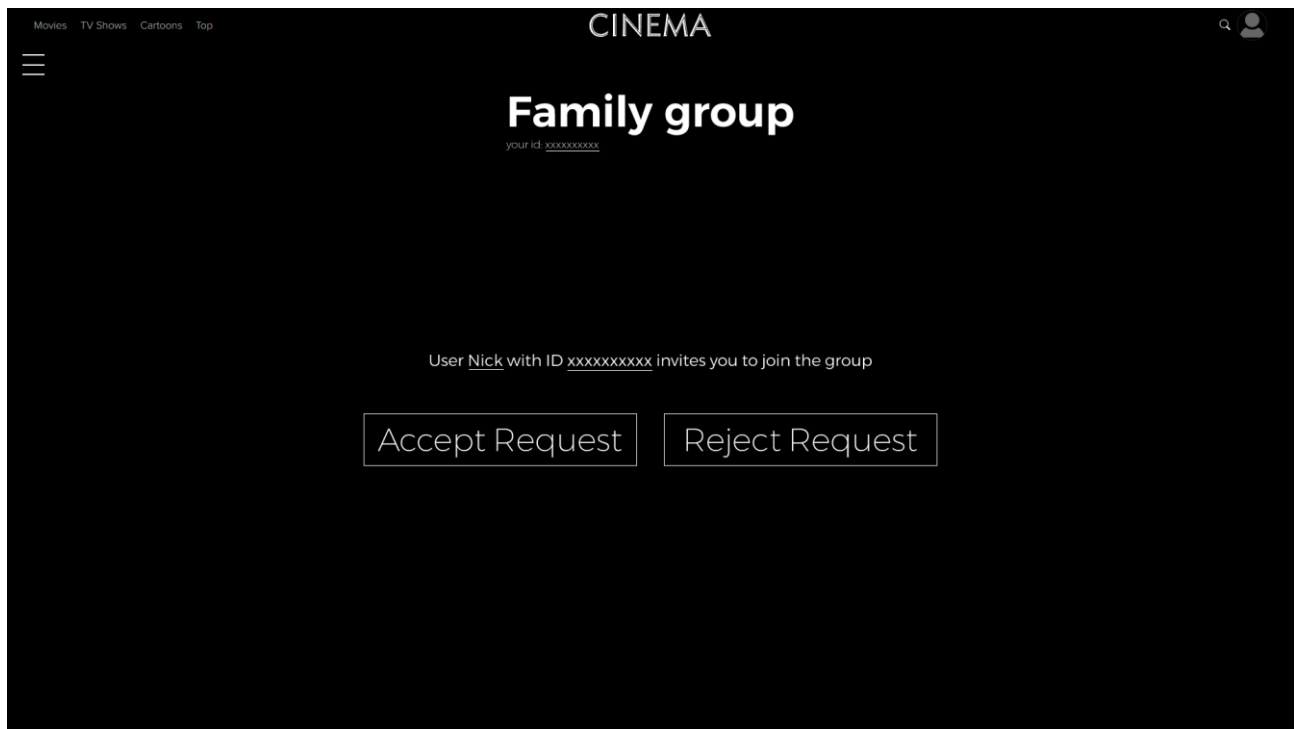


Рисунок А.4 — Запит на приєднання до групи

Додати нового члена групи можна не тільки через надсилання запиту, але й через створення облікового запису для користувача. Ця функція призначена для користувачів, доступ яких потрібно обмежити.

Щоб створити такий обліковий запис, необхідно натиснути на відповідну кнопку. Після цього буде згенеровано ідентифікатор, а також тимчасові логін та пароль нового члена групи.

На рисунку А.5 показано сторінку з кнопкою реєстрації. На рисунку А.6 представлені ідентифікатор та авторизаційні дані нового користувача.

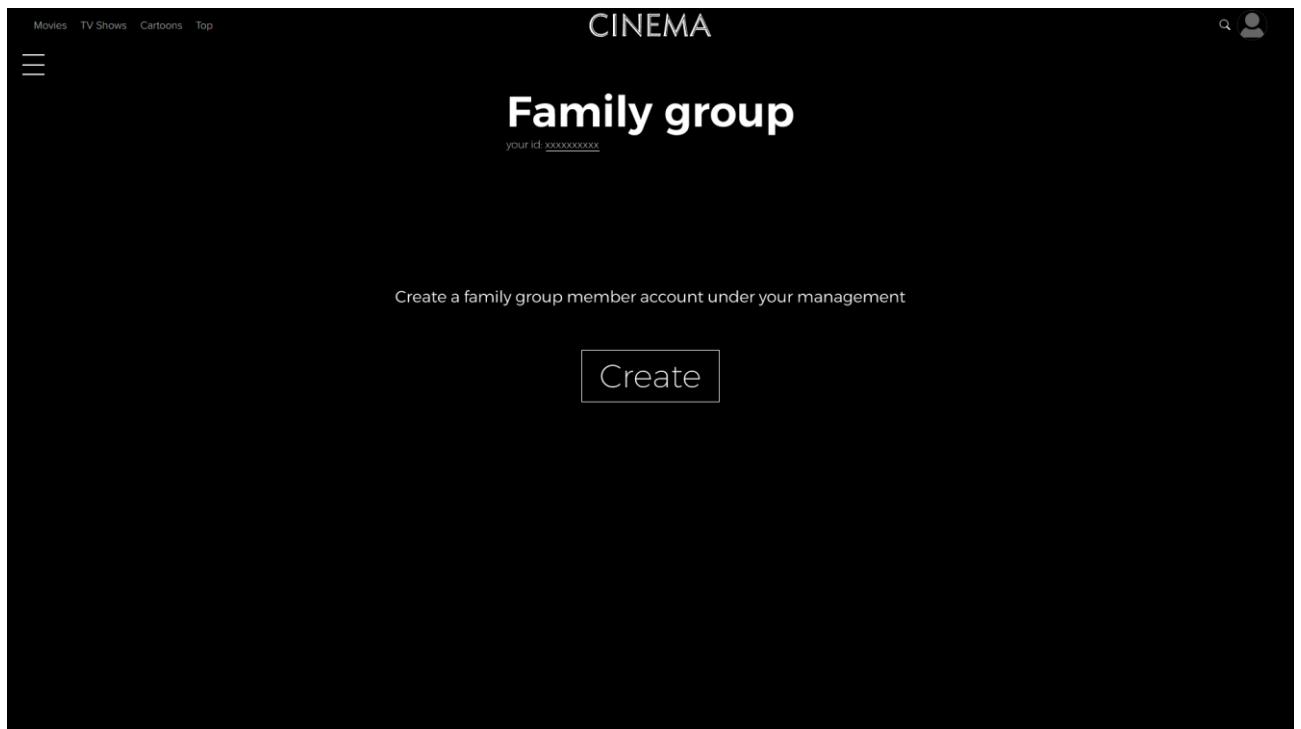


Рисунок А.5 — Кнопка реєстрації облікового запису для члена групи

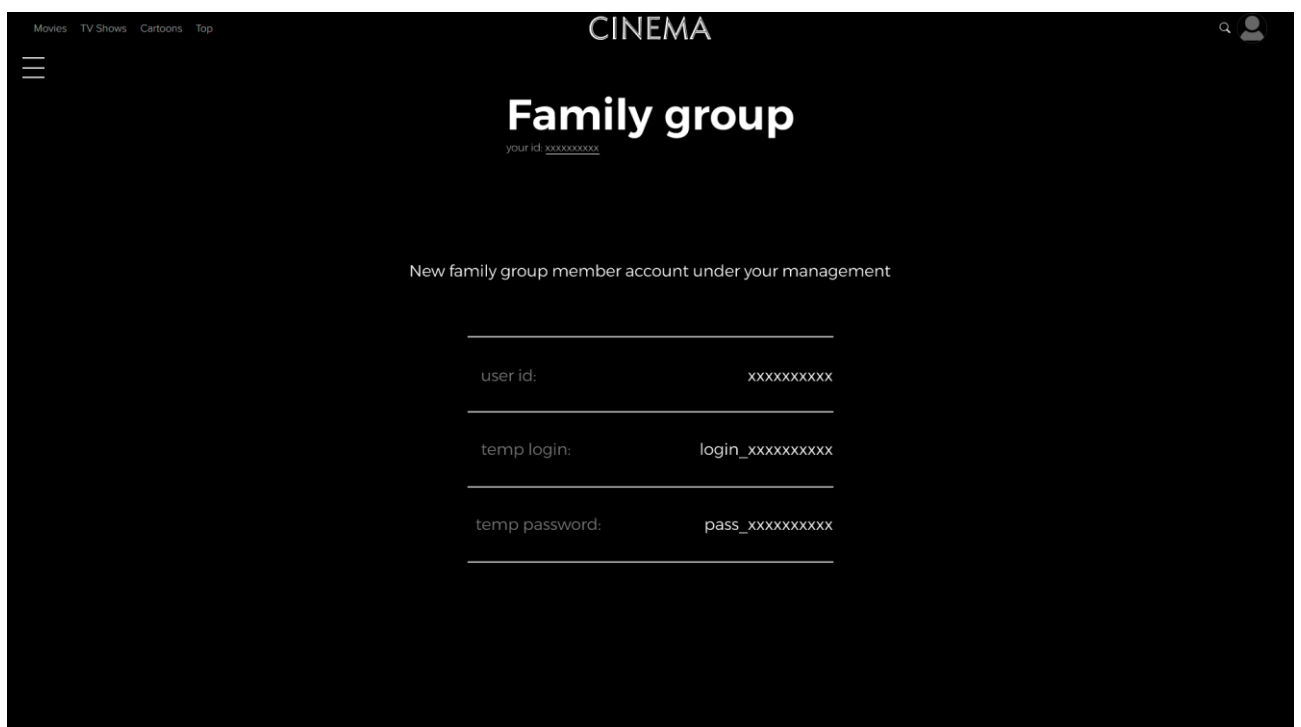


Рисунок А.6 — Дані нового користувача та члена групи

Після того, як група була створена, кожен її член може переглядати список учасників. На рисунку А.7 показано сторінку адміністратора після того,

як він запросив хоча б одного учасника. На цій сторінці є форма для подальшого запрошення нових учасників та список поточних учасників.

Адміністратор групи має можливість видаляти користувачів або налаштовувати для них доступ до контенту.

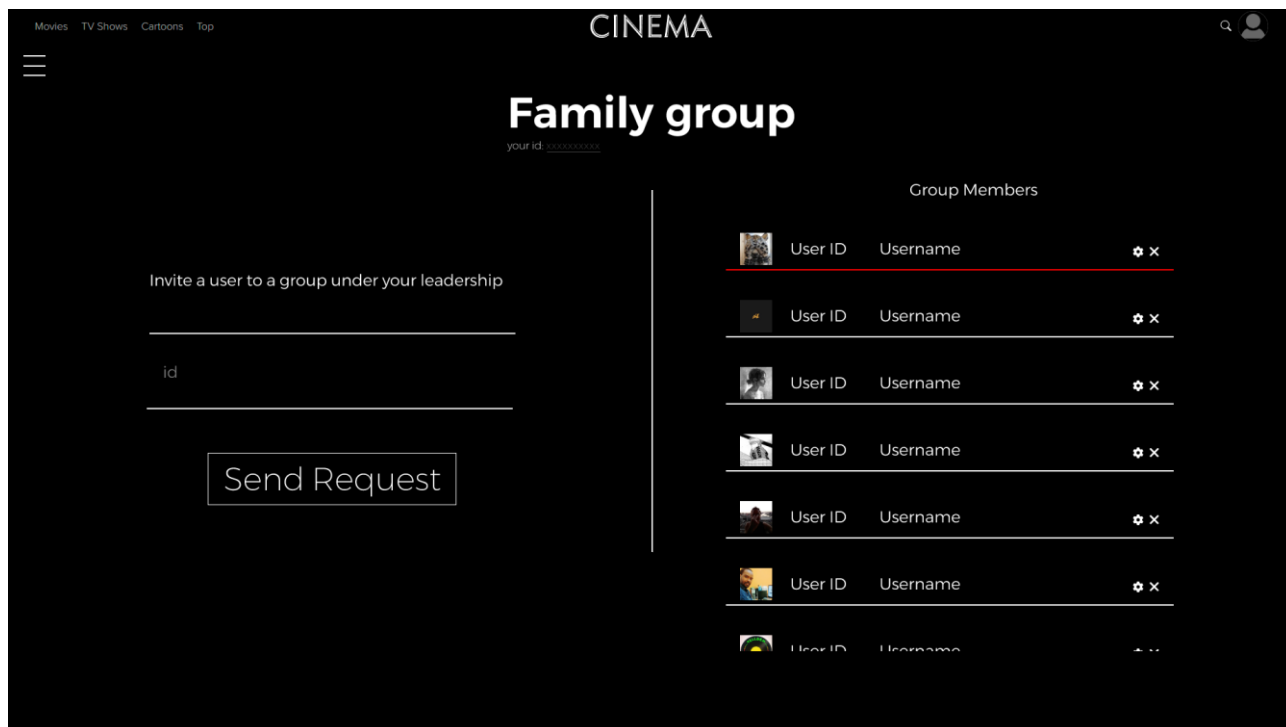


Рисунок А.7 — Сторінка адміністратора зі списком учасників групи

Кожен член групи також має доступ до списку учасників, але без прав адміністратора. Такому користувачеві надано можливість лише покинути групу.

На рисунку А.8 показано зображення зі списком користувачів для члена групи, який не є адміністратором.

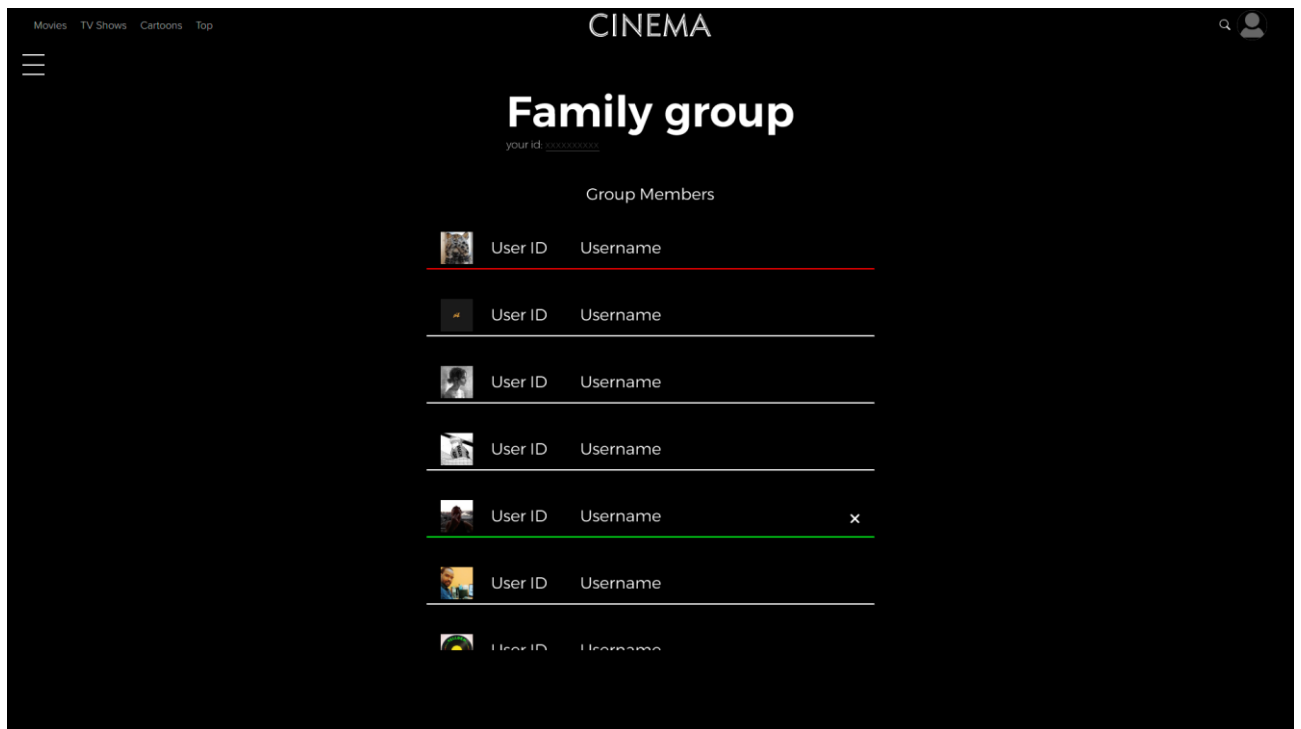


Рисунок А.8 — Сторінка члена групи зі списком користувачів

Слід зазначити, що можливість залишити групу є лише у користувачів, які прийняли запрошення на вступ. Користувачі, чий обліковий запис були створені адміністратором, не мають самотійної можливості покинути групу. Вони можуть залишити групу лише у разі виключення адміністратором.

Після виключення адміністратором такого користувача він більше не перебуває під контролем цього адміністратора.

На рисунку А.9 показано зображення зі списком користувачів для члена групи, чий обліковий запис було створено адміністратором.

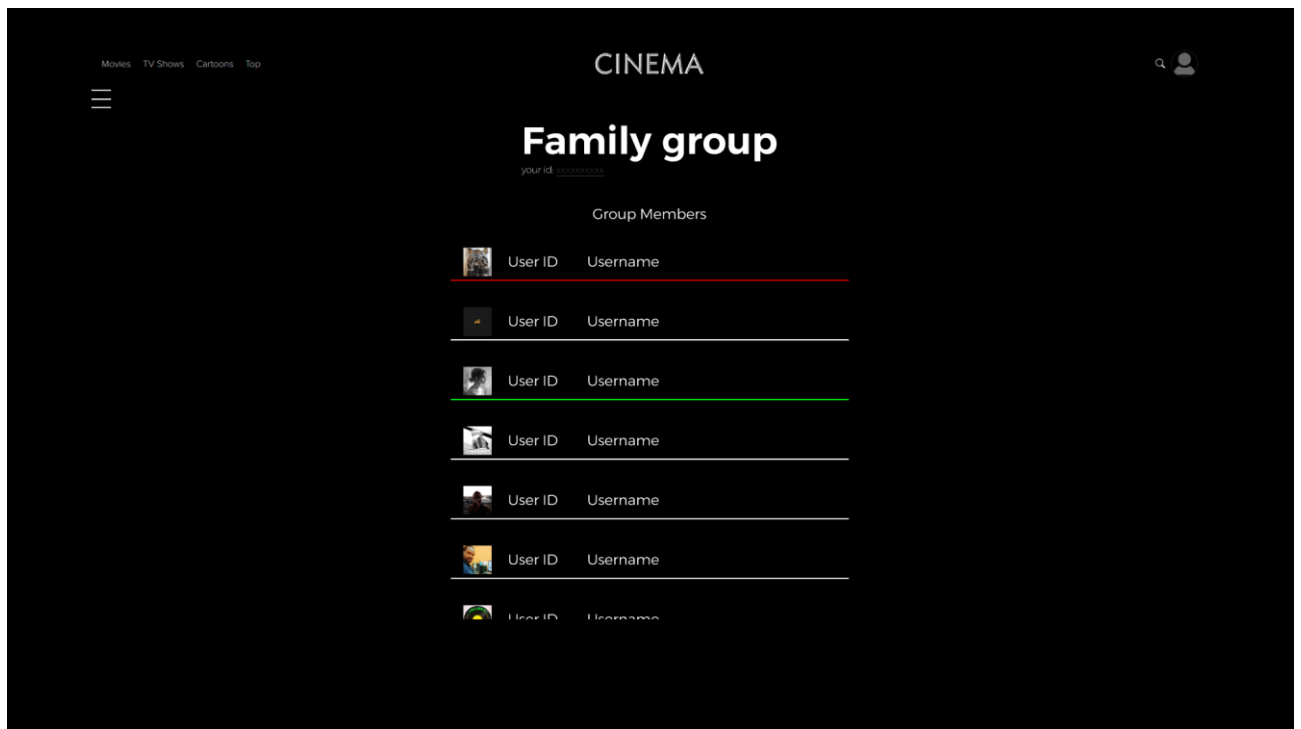


Рисунок А.9 — Сторінка члена групи зі списком користувачів

Спільний перегляд

Ще однією доступною для авторизованого користувача функцією є спільний перегляд картини із членами групи. На рисунку А.10 представлено зображення сторінки спільного перегляду, що складається із трьох логічних секцій:

- Плеєр,
- Метаінформація,
- Чат.

Варто зазначити, що почати спільний перегляд може будь-який користувач, на котрого під час перегляду будуть покладені функції адміністратора. Також варто додати, що під час спільного перегляду користувачі можуть спілкуватися як у голосовому, так і текстовому режимі.

На рисунку А.10 показано сторінку спільного перегляду.

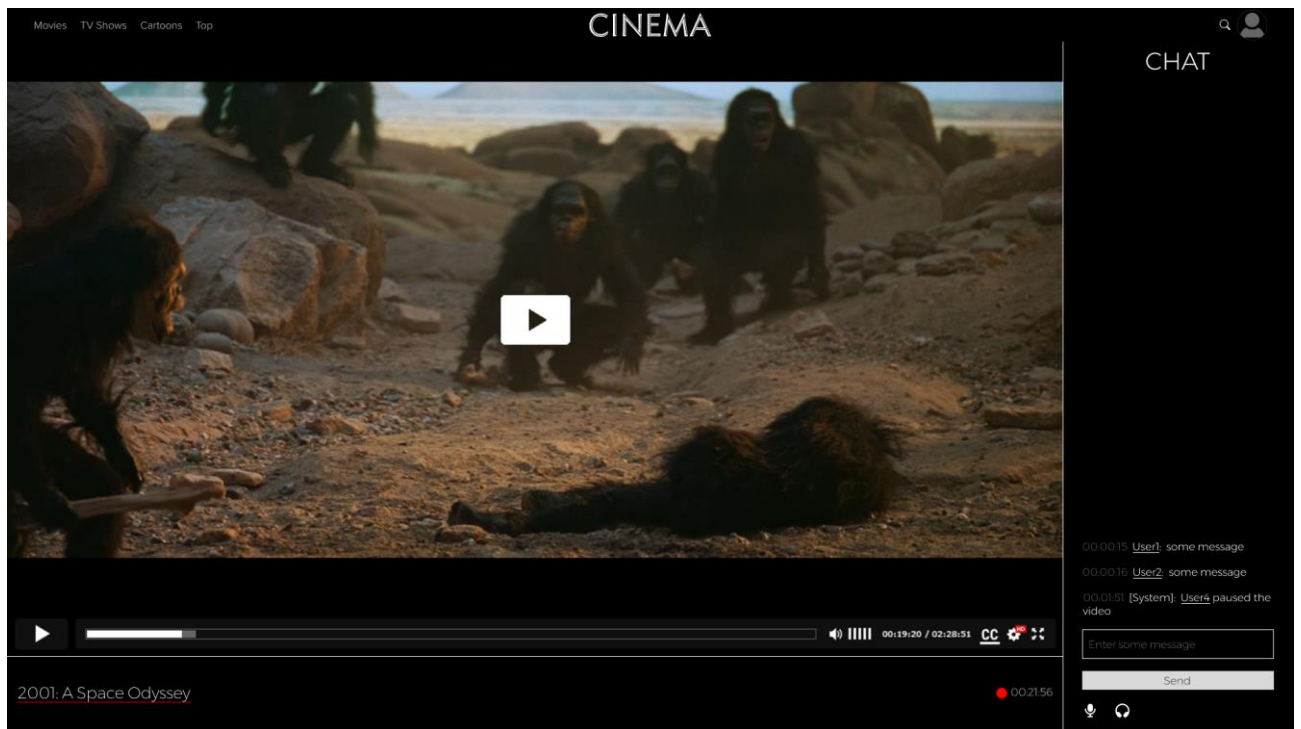


Рисунок А.10 — Сторінка спільного перегляду

Під час перегляду кожен користувач може управляти іншими. Для звичайного користувача це можливість зміни рівня гучності інших учасників, що на зображенні А.11. Адміністратор перегляду має ширший спектр можливостей, що також представлено на зображенні А.11, а саме:

- Зміна рівня гучності;
- Відключення мікрофона для конкретного користувача;
- Вимкнення можливості керування переглядом;
- Вимкнення відображення повідомлень для інших користувачів;
- Вимкнути можливість надсилати повідомлення.



User1



Рисунок А.11 — Спливаюче вікно взаємодії користувача



User1



— volume level

- 61 +



— mute



— block control of video



— do not display messages



— block sending messages

Рисунок А.12 — Спливаюче вікно взаємодії для адміністратора

ДОДАТОК Б

Лістинг програми

Слід зазначити, що нижче надано не весь вихідний код проекту, лише окремі функції і частини HTML розмітки.

Слід зазначити, що frontend частина проекту складається з модулів або деяких компонентів, що дозволяє досягти гнучкості і тієї самої модульності проекту.

Структура всього проекту має такий вигляд:

```
root
├── main
│   ├── index.html
│   ├── styles
│   │   ├── style1.css
│   │   └── style2.css
│   └──
│       └── scripts
│           ├── script1.js
│           └── script2.js
├──
├── module1
│   ├── template.html
│   ├── style.css
│   └── script.js
├──
├── module2
├──
└── assets
```

comment-input.template.html

```
<template>
  <div class="reply-form">
    <div class="input"
      contenteditable="true"
      role="textbox"
      dir="auto"
      tabindex="0"></div>
    <div class="reply-actions">
      <div class="btn-send">Send</div>
      <div class="btn-clear">Clear</div>
      <div class="btn-cancel">Cancel</div>
    </div>
  </div>
</template>
```

utils.js

```
/*ajax wrapper*/
function ajax(url, method, data, cbSuccess, cbError = null)
{
  let xhr = new XMLHttpRequest();

  xhr.open(method, url, true);

  xhr.setRequestHeader('Content-Type', 'application/json');

  xhr.onreadystatechange = function()
  {
    if(xhr.readyState === 4 && xhr.status === 200)
    {
      cbSuccess(xhr.responseText);
    }
  };

  if(cbError != null)
  {
    xhr.onerror = function()
    {
      cbError();
    };
  }
  if(data) xhr.send(JSON.stringify(data));
  else xhr.send();
}

function formatTime(minutes)
{
  let time = '';

  let date = new Date(minutes * 1000 * 60).toISOString();

  let hh = date.substring(11, 13);
  let mm = date.substring(14, 16);

  let _hh = '';
  let _mm = '';
}
```

```

        if(hh[0] != '0') _hh = `${hh}h`;
        else if(hh[1] != '0') _hh = `${hh[1]}h`;

        if(mm[0] != '0') _mm = `${mm}m`;
        else if(mm[1] != '0') _mm = `${mm[1]}m`;

        if(_hh && _mm) time = `${_hh} ${_mm}`;
        else time = `${_hh}${_mm}`;

        return time;
    }

function createElement(tagName, {classList = [], children = [], text = ''} = {})
{
    if(!tagName) false;
    let tag = document.createElement(tagName);

    if(classList.length > 0)
    {
        for(let className of classList)
            tag.classList.add(className);
    }

    if(children.length > 0)
    {
        for(let child of children)
            tag.appendChild(child);
    }

    if(text)
        tag.appendChild(document.createTextNode(text));

    return tag;
}

class Base
{
    ...
    loadTemplate()
    {
        ajax(this.templateURL, 'GET', null, rawTemplate =>
        {
            let dom = new DOMParser().parseFromString(rawTemplate,
'text/html');
            this.template = dom.querySelector('template').content;
            this.cbCheckCompletion();
        });
    }

    cbCheckCompletion() {}
}

class someModuleClass extends
{
    constructor()
    {
        super();
        ...
    }
}

```

```

    }

    ...

    cbCheckCompletion()
    {
        if(this.template && /*special class conditions*/)
        {
            this.node = this.template.firstElementChild;
            this.createDOM();
        }
    }
}

```

CommentInput.js

```

class CommentInput extends CommentBase
{
    constructor(containerID, templateUrl, backendURL)
    {
        super();

        this.containerID = containerID;
        this.templateURL = templateUrl;
        this.backendURL = backendURL;

        this.init();
    }

    static _cbBtnCancel(node, element = null)
    {
        node.querySelector('.btn-cancel').addEventListener('click',
            function()
            {
                node.parentElement.classList.remove('active');
            });
    }

    static _cbBtnSend(node, data)
    {
        node.querySelector('.btn-send').addEventListener('click', function()
        {
            let content = '';
            content = node.querySelector('.input').innerHTML;
            let url = data.url;
            ajax(url, 'POST', {type: 'send comment', ...data.content},
                function(){});
        });

        CommentInput._cbBtnClear(node);
    }

    static _cbBtnClear(node)
    {
        node.querySelector('.btn-clear').addEventListener('click',
            function()
            {
                node.querySelector('.input').innerHTML = '';
            });
    }
}

```

```

static _cbTabulation(node)
{
    node.querySelector('.input').addEventListener('keydown', function(e)
    {
        if (e.key === 'Tab')
        {
            e.preventDefault();

            let spaceNode =
                document.createTextNode('\xa0\xa0\xa0\xa0');

            let selection = window.getSelection();
            let range = selection.getRangeAt(0);

            range.insertNode(spaceNode);
            range.setStartAfter(spaceNode);
            range.setEndAfter(spaceNode);

            selection.removeAllRanges();
            selection.addRange(range);
        }
    });
}

cbCheckCompletion()
{
    if(this.template)
    {
        this.node = this.template.firstElementChild;
        this.createDOM();
    }
}

init()
{
    this.container = document.getElementById(this.containerID);
    this.loadTemplate();
}

fillReplyForm(form)
{
    form.querySelector('.input').setAttribute('placeholder', 'add some
    comment...');
}

createDOM()
{
    let replyForm = this.node.cloneNode(true);

    CommentInput._cbBtnCancel(replyForm);
    CommentInput._cbBtnClear(replyForm);
    CommentInput._cbBtnSend(replyForm);
    CommentInput._cbTabulation(replyForm);

    replyForm.querySelector('.btn-cancel').remove();

    this.fillReplyForm(replyForm);
    this.container.appendChild(replyForm);
}

getNodeInstance()
{
    if(this.node)

```

```

        return this.node.cloneNode(true);
    } else return null;
}
}

```

CommentsSection.js

```

class CommentsSection extends CommentBase
{
    constructor(containerID, templateUrl, backendURL)
    {
        super();
    }
    ...

    unselect()
    {
        if(this.selectedNodes_.selected)
            this.selectedNodes_.selected.classList.remove('selected');

        for(let i = 0; i < this.selectedNodes_.children.length; i++)
            this.selectedNodes_.children[i].classList.remove('selected');

        for(let i = 0; i < this.selectedNodes_.ancestors.length; i++)
            this.selectedNodes_.ancestors[i].classList.remove('selected');

        this.selectedNodes_.selected = null;
        this.selectedNodes_.children = [];
        this.selectedNodes_.ancestors = [];
    }

    #selectNodes(node)
    {
        this.selectedNodes_.selected = node;

        let children = node.parentElement.querySelector('.replies-
            section').children;
        for(let child of children)
            this.selectedNodes_.children.push(child.querySelector
                ('.content-section'));

        let parent = node.parentElement.parentElement.parentElement;
        while(parent && parent.classList.contains('comment'))
        {
            this.selectedNodes_.ancestors.push(parent.querySelector
                ('.content-section'));

            if(parent.classList.contains('root')) break;

            parent = parent.parentElement.parentElement;
        }
    }

    createDOM(data_)
    {
        const parents = {};

        let dataKeys = Object.keys(data_);
    }
}

```

```

let children = {};
for(let key of dataKeys)
{
    let data = data_[key];
    if(data.parentID != null)
        if(!children[data.parentID]) children[data.parentID] =
            true;
}

for(let key of dataKeys)
{
    let data = data_[key];

    let comment = this.node.cloneNode(true);
    this.fillComment(comment, data);
    this.setEvents(comment);

    if(data.parentID == null) comments.appendChild(comment);
    else
    {
        if(parents[data.parentID] != undefined)
            parents[data.parentID].appendChild(comment);
        else
        {
            let commentParent = this.container.querySelector
                (`[commentID="${data.parentID}"]`);
            section`.replies-
            commentParent.appendChild(comment);

            parents[data.parentID] = commentParent;
        }

        if(data.isParent && !children[data.commentID])
            this.observer.observe(comment);
    }
}
}
}

```

MinorSlider.js

```

class MinorSlider extends Base
{
    constructor(options = {})
    {
        this.options = options;
        super();
        ...
    }
    ...

    init()
    {
        ...

        this.elements.style.gap = `${this.gap}px`;

        for(let i = 0; i < data.length; i++)
        {
            let card = createCard(data[i]);
            this.elements.appendChild(card);
        }
    }
}

```

```

        card.querySelector('.bg-card').onload = function()
        {
            this.endX += card.querySelector('.bg-card').width + this.gap
            - (this.gap * (1 / data.length));
        };
    }

    this.endX -= screen.width - this.containerPadding * 2;

    this._btnPrev.onclick = () => this.prev();
    this._btnNext.onclick = () => this.next();
}

update()
{
    if(this.offset == 0)
        this._btnPrev.classList.add('navigation-btn-inactive');
    else this._btnPrev.classList.remove('navigation-btn-inactive');

    if(this.offset == -Math.abs(this.endX))
        this._btnNext.classList.add('navigation-btn-inactive');
    else this._btnNext.classList.remove('navigation-btn-inactive');

    this.elements.style.transition = 'left 1s ease';
    this.elements.style.left = `${this.offset}px`;
}

next()
{
    this.offset -= this.screenW / this.elementsScroll;
    if(this.offset < -Math.abs(this.endX))
        this.offset = -Math.abs(this.endX);
    this.update();
}

function prev()
{
    this.offset += this.screenW / this.elementsScroll;
    if(this.offset > 0) this.offset = 0;
    this.update();
}
}

```

connection.php

```

/*mysql connect wrapper*/
<?php
class Connection
{
    function __construct($db)
    {
        $servername = "localhost";
        $username = "root";
        $password = "";
        $this->dbname = $db;

        $this->conn = mysqli_connect($servername, $username,
                                    $password, $this->dbname);

        if(!$this->conn)
            die("Connection failed: " . mysqli_connect_error());
    }
}

```

```

    }

    function __destruct()
    {
        if ($this->conn) mysqli_close($this->conn);
    }
}

?>

```

utils.php

```

<?php
function isValidJSON($json)
{
    json_decode($json);
    return (json_last_error() === JSON_ERROR_NONE);
}

?>

```

frontunit.php

```

<?php
...

$POST_JSON = $json = file_get_contents('php://input');
$POST_DATA = null;

if(!isValidJSON($POST_JSON))
    return;

$POST_DATA = $json = json_decode($POST_JSON);

...

if($POST_DATA->type == /*some type*/)
{
    ...

    return;
}

?>

```

ДОДАТОК В

Технічне завдання

Вступ

Назва проекту, що розробляється: «Дослідження технологій та проектування інформаційної системи онлайн-кінотеатр».

1. Підстави для розробки

Підставою для розробки програмного забезпечення є виконання кваліфікаційної роботи за спеціальністю «Комп'ютерні науки».

Замовник повинен отримати:

- Веб-додаток — онлайн-кінотеатр.
- Документацію, яка включає:
 - опис системи,
 - інструкцію користувача,
 - вихідний код.

2. Призначення розробки

2.1 Функціональне призначення

Розроблювана система повинна надавати користувачам:

- Можливість зручного пошуку та фільтрації контенту.
- Потокowe відтворення відео.
- Систему оцінок та коментування картин.

2.2 Експлуатаційне призначення

Розроблений програмний продукт має слугувати для приємного проведення часу.

3. Вимоги до системи

3.1 Функціональні вимоги

Основні функції системи:

- Пошук фільмів, серіалів та мультфільмів за ключовими словами та фільтрами.
- Перегляд основної інформації про картину.
- Потокове відтворення контенту.
- Залишення відгуків та перегляд існуючих.

Вхідна інформація:

- Запити на пошук.
- Необхідні фільтри.

Вихідна інформація:

- Результати пошуку.
- Інформація про фільм.
- Відтворення відео.

3.2 Нефункціональні вимоги

Нефункціональними вимогами до системи є:

- Система повинна забезпечувати постійну та безперебійну роботу.
- Забезпечення високої продуктивності.
- Використання інтуїтивно зрозумілого інтерфейсу.

3.3 Умови експлуатації

Система не потребує спеціальних навичок від користувача для її використання. Інтерфейс інтуїтивно зрозумілий і зручний. Система призначена для користування людьми різного віку та статі.

Система повинна працювати в браузері, який підтримує сучасні стандарти (HTML, CSS, JavaScript).

3.4 Вимоги до транспортування та зберігання

Вихідний код проекту, ресурси та завантажені дані можуть зберігатися та транспортуватися як всі компакт-диски, флеш-накопичувачі, твердотілі носії або передаватися через Інтернет. З накопичувачами слід поводитись як з крихким матеріалом.

Ця система встановлюється на жорсткий диск, бажано серверного призначення. Для коректної роботи температура приміщення повинна бути в діапазоні від +15 до +35°C. Відносна вологість не більше 70%.

4. Вимоги до документації

Документація повинна включати:

- Технічне завдання.
- Опис архітектури системи.
- Інструкцію користувача.
- Вихідний код додатку.

5 Стадії та етапи розробки

Стадії та етапи розробки гри та терміни їх виконання повинні відповідати затвердженому графіку проектування кваліфікаційної роботи та зводитись до таких, що наведені в таблиці А.1.

Таблиця А.1– Стадії, етапи розробки системи, та терміни їх виконання

Стадії розробки	Етапи робіт	Зміст робіт	Терміни виконання
Технічне завдання	Розробка	Обґрунтування необхідності розробки системи.	11.09.24 – 01.10.24
	Затвердження	Розробка пояснювальної записки. Узгодження і затвердження технічного завдання.	01.10.24 – 12.10.24
Ескізний проект	Розробка	Попередня розробка структури вхідних та вихідних даних, уточнення методів рішення завдань, загальний опис.	12.10.24 – 15.10.24
	Затвердження	Розробка пояснювальної записки. Узгодження і затвердження ескізного проекту.	15.10.24 – 19.10.24
Технічний проект	Розробка	Уточнення структури вхідних та вихідних даних, розробка логіки, розробка архітектури системи.	19.10.24 – 23.10.24
	Затвердження	Розробка пояснювальної записки. Узгодження і затвердження технічного проекту.	23.10.24 – 26.10.24
Робочий проект	Написання коду системи	Програмування та налагодження системи.	26.10.24 – 26.11.24
	Випробування	Розробити, узгодити та затвердити програму і методику випробувань, коригування коду за результатами випробувань	26.11.24 – 01.12.24
	Розробка документації	Розробка програмної документації	09.12.24 – 16.12.24

6. Порядок прийому та контролю

Для контролю й прийому необхідно надати інструкцію користувача та опис програми, а також програму та методику випробування.

Порядок контролю і прийому даної розробки здійснюється представником замовника у присутності представників розробника згідно з програмою та методикою випробувань.

За результатами прийому складається акт, який підписується представником замовника та представником розробника та утверджується керівниками організації-замовника і організації-розробника.

У випадку виявлення помилок під час прийому програмного виробу складається акт про виявлені помилки, який підписується представниками замовника та розробника й стверджується керівниками організації-замовника й організації-розробника. Розробник повинен впродовж не більш як 1-го місяця видалити вказані зауваження, й повідомити замовника про повторне проведення перевірки, не пізніше як за 2 тижні.