

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему: «Розроблення інформаційної системи для кінотеатру з використанням технології NextJS»

здобувача Хилі Олександра Олександровича
(ПІБ)

(Підпис)

Науковий керівник:

к.е.н., доцент

_____ Лозовик Ю. М.

**Робота допущена до захисту перед
екзаменаційною комісією з атестації
здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент

_____ Тішков Б.О.

Київ 2025

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проектної групи(гарант)
освітньо-професійної програми

_____ Помазун О.М.

“ _____ ” _____ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____ Тішков Б.О.

“ _____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувачу вищої освіти **Хилі Олександра Олександровича**

очної (денної) форми навчання

на підготовку кваліфікаційної бакалаврської роботи
на тему: «Розроблення інформаційної системи для кінотеатру з використанням
технології NextJS»

Тему затверджено наказом ректора Університету від « 7 » березня 2025 р. № 466- ст.

Кваліфікаційна бакалаврська робота виконується на матеріалах
отриманих під час навчання, здійснення студентських наукових досліджень, а також
зібраного фактичного матеріалу з обраної теми дослідження під час практики

План кваліфікаційної бакалаврської роботи

Розділ I Характеристика та аналіз предметної галузі.

Розділ II Розробка вимог і моделювання інформаційної системи кінотеатру.

Розділ III Проєктування та реалізація компонентів веб-системи з використанням NextJS.

Об'єкт дослідження Процес взаємодії користувачів із веб-платформою для перегляду
фільмів і шоу, включаючи оформлення підписки, вибір медіаконтенту, перегляд опису
фільмів, звернення до техпідтримки.

Предмет дослідження Методи, підходи та технології розробки адаптивної
інформаційної системи для кінотеатру на базі Next.js, з урахуванням інтеграції з
клієнтською частиною, базою даних та підтримкою мультимедійного контенту.

Мета кваліфікаційної бакалаврської роботи Розробити інформаційну систему
кінотеатру, яка дозволяє зручно переглядати фільми та шоу, оформлювати підписку,
звертатися до служби підтримки, забезпечуючи при цьому сучасний рівень
користувацького досвіду завдяки технологіям NextJS і адаптивному дизайну.

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I у п. 1.1 охарактеризувати галузь онлайн-медіа та стримінгових платформ, описати ключові процеси, такі як вибір контенту, оформлення підписки, підтримка користувачів; визначити основних користувачів системи: глядачів, адміністраторів, технічну підтримку. У п. 1.2 зробити огляд існуючих рішень у сфері медіаплатформ (Netflix, Megogo, Sweet.tv тощо); проаналізувати переваги та недоліки, функціональність та реалізацію інтерфейсів.

У розділі II у п. 2.1 сформулювати функціональні та нефункціональні вимоги до інформаційної системи кінотеатру, серед яких: наявність сторінок (головна, фільми та шоу, опис фільму, підписка, підтримка), адаптивність, швидкодія, зручність інтерфейсу. У п. 2.2 описати ключові задачі, які повинна вирішувати система, зокрема: перегляд і сортування фільмів, фільтрація за жанрами, оформлення підписки, звернення до служби підтримки; сформулювати алгоритм взаємодії користувача з платформою. У п. 2.3 побудувати та навести відповідні діаграми (Use Case, Activity, Component, Deployment) для візуалізації структури інформаційної системи, її процесів і взаємодій між компонентами та користувачами.

У розділі III у п. 3.1 визначити й описати типи даних, що використовуються у системі, зокрема: інформація про фільми, акторів, підписки, користувачів і звернення до підтримки; охарактеризувати джерела даних та способи їх зберігання й обробки. У п. 3.2 охарактеризувати апаратні вимоги для розгортання системи та середовище її роботи (сервери, хостинг, адаптивність під різні пристрої). У п. 3.3 описати обраний технологічний стек (Next.js, TailwindCSS, Firebase або Supabase, Firestore, Stripe для підписок), інтерфейси взаємодії між компонентами системи та механізми автентифікації. У п. 3.4 продемонструвати приклади реалізованої функціональності: перегляд та фільтрація контенту, система підписок, інтеграція служби підтримки, адаптивність для мобільних пристроїв.

**Завдання підготував
науковий керівник**

_____ **Лозовик Юрій Миколайович**
« 10 » березня 2025 р.

**Завдання одержав
здобувач**

_____ **Хиля Олександр Олександрович**
« 10 » березня 2025 р..

Відгук
на кваліфікаційну бакалаврську роботу
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

Хилі Олександра Олександровича

на тему

**«Розроблення інформаційної системи для кінотеатру з використанням
технології NextJS»**

1. Актуальність теми. У сучасних умовах стрімкого розвитку цифрових технологій питання автоматизації обслуговування клієнтів у сфері кіноіндустрії є вкрай актуальним. Онлайн-сервіси для перегляду фільмів, оформлення підписок та взаємодії з контентом набувають все більшого поширення. Запропонована тема роботи відповідає актуальним запитам ринку та реальним потребам користувачів.

2. Позитивні риси кваліфікаційної бакалаврської роботи: Проєкт реалізовано згідно з класичною структурою, що включає теоретичну частину, аналіз функціональних вимог, прототипування (макет у Figma), практичну реалізацію та підсумкові висновки. Особливої уваги заслуговує глибоке опрацювання інтерфейсів користувача, створених з урахуванням адаптивності для різних пристроїв. Розробник впевнено оперує сучасними технологіями фронтенду (Next.js, Tailwind CSS) у поєднанні з сервісами Firebase (Realtime Database, Authentication), що дозволило реалізувати повноцінну інформаційну систему без потреби у власному сервері.

3. Наявність самостійних розробок автора: У процесі реалізації автор самостійно розробив функціонал авторизації користувачів, розмежування прав доступу (користувач/адміністратор), модулі додавання та редагування фільмів, перегляд контенту з TMDB API, відгуки, адаптивні сторінки з тарифами та підтримкою. Це свідчить про високий рівень самостійності, технічної компетентності та здатність приймати архітектурні рішення у межах обраної технологічної бази.

4. Цінність теоретичних висновків та практичних рекомендацій: Робота містить логічні та обґрунтовані висновки, демонструє практичне розуміння процесу розробки SPA-застосунків, взаємодії з базами даних у режимі реального часу, побудови структурованого UI/UX на основі макету. Проєкт може бути основою для

подальшого розвитку системи, зокрема – інтеграції платіжних сервісів, розширення функцій адміністратора чи масштабування під більшу кількість користувачів.

5. Наявність недоліків: У роботі трапляються окремі незначні недоліки мовного та стилістичного характеру, а також бракує детального опису тестування (наприклад, прикладів тест-кейсів або оцінки продуктивності системи). Було б доцільно оцінити можливості масштабування проєкту при збільшенні навантаження на базу даних.

6. Загальна оцінка кваліфікаційної бакалаврської роботи та її допущення до захисту перед ЕК: Робота виконана на високому технічному та методичному рівні, має завершений вигляд і підтверджує фахову підготовку здобувача. Проєкт рекомендовано до захисту перед екзаменаційною комісією. Рекомендована оцінка «відмінно».

Науковий керівник

(підпис)

доцент, к.е.н. Лозовик Ю.М.

“9” червня 2025 р.

Рецензія
на кваліфікаційну бакалаврську роботу
здобувача вищої освіти
Хилі Олександра Олександровича

Тема «Розроблення інформаційної системи для кінотеатру з використанням технології NextJS»

Актуальність теми кваліфікаційної бакалаврської роботи і доцільність її розроблення. Тема проєкту є надзвичайно актуальною, оскільки цифровізація процесів у сфері розваг стає необхідністю для сучасного бізнесу. Розробка вебсайту для кінотеатру дозволяє суттєво покращити взаємодію з глядачами, спростити перегляд фільмів, ознайомлення з описом і відгуками, а також створює потенціал для впровадження платних підписок. Це сприяє підвищенню якості сервісу, зниженню витрат на підтримку та створює конкурентну перевагу на ринку.

Якість проведеного дослідження. Автор продемонстрував високий рівень підготовки та аналітичного мислення. У проєкті проведено детальний аналіз функціональних і нефункціональних вимог, складено прототип у Figma, обґрунтовано вибір інструментів.

Позитивні риси кваліфікаційної бакалаврської роботи. Проєкт реалізовано із використанням сучасного технологічного стеку, що відповідає актуальним вимогам до веброзробки. Автор вдало поєднав теоретичні знання з практичними навичками, продемонструвавши глибоке розуміння побудови адаптивних інтерфейсів, взаємодії з зовнішніми API та налаштування базової безпеки.

Зауваження. Деякі аспекти могли б бути розкриті детальніше — зокрема, питання масштабованості рішення, навантаження на базу даних та потенційна інтеграція з платіжними сервісами у разі впровадження системи підписок. Також бажано додати приклади тест-кейсів для оцінки функціональної цілісності.

Практична значимість висновків і рекомендацій. Проєкт має високу практичну цінність і може бути реалізований як реальний вебсервіс для кінотеатру або платформи з перегляду контенту. Запропоновані рішення є гнучкими, масштабованими та придатними до подальшого розширення — зокрема, впровадження оплати, аналітики чи рекомендаційних систем. Отримані результати можуть слугувати основою для реального впровадження в галузі.

Місце роботи та посада рецензента

Науковий ступінь, учене звання (за наявності) _____
(підпис, ПІБ)

Підпис засвідчую: _____
(посада, підпис)

АНОТАЦІЯ

кваліфікаційної бакалаврської роботи
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,
виконаної на тему: «Розроблення інформаційної системи для кінотеатру з
використанням технології NextJS»

Київ: кафедра інформаційних систем в економіці, 2025 р.

Хиля О.О. «Розроблення інформаційної системи для кінотеатру з використанням технології NextJS».

Кваліфікаційний бакалаврський проєкт спрямований на розробку інформаційної системи для онлайн-кінотеатру, що забезпечує зручний доступ до перегляду фільмів і шоу, управління підписками, а також звернення до служби підтримки. Розробка реалізована з використанням сучасного JavaScript-фреймворку Next.js з акцентом на продуктивність, адаптивність та зручність користувацького інтерфейсу.

Основні функціональні можливості інформаційної системи включають:

1. Головна сторінка: відображає добірки фільмів і шоу, інформацію про переваги сервісу, часті запитання та тарифні плани.
2. Сторінка “Фільми та шоу”: дозволяє переглядати фільми за жанрами, популярністю, категоріями.
3. Опис фільму або шоу: містить детальну інформацію про вибраний контент — опис, актори, відгуки користувачів.
4. Сторінка з підпискою: представлено порівняльну таблицю тарифних планів, доступ до оформлення підписки.
5. Сторінка підтримки: реалізовано форму звернення до технічної підтримки, розділ із поширеними питаннями.

Інформаційна система використовує базу даних для зберігання інформації про контент, користувачів, підписки та звернення до підтримки. Реалізовано адаптивну верстку для всіх типів пристроїв, можливість перегляду фільмів та популярних категорій, отримання тех. підтримки та оформлення підписки. Компоненти інтерфейсу реалізовані з урахуванням сучасних підходів до UI/UX.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 60 сторінок, 4 таблиці, 31 рисунок, список використаних джерел з 15 найменувань, 2 додатка.

«Розроблення інформаційної системи для кінотеатру з використанням технології NextJS»

Об'єкт дослідження: Процес взаємодії користувачів із вебсайтом кінотеатру, що включає вибір фільмів, перегляд розкладу, оформлення підписки, звернення до технічної підтримки та управління персональним кабінетом.

Предмет дослідження: Сукупність підходів, методів і технологій розробки сучасних вебзастосунків з використанням фреймворку Next.js, зокрема на прикладі створення інформаційної системи для кінотеатру.

Мета і завдання дослідження: Розробити та реалізувати інформаційну систему для кінотеатру у вигляді адаптивного вебзастосунку, яка забезпечує користувачам зручну навігацію, перегляд фільмів, оформлення підписки та доступ до додаткових сервісів.

Відповідно до поставленої мети визначені такі *завдання*:

- Дослідження предметної області: аналіз функціонування сучасних кінотеатрів та їх цифрових платформ.
- Аналіз існуючих вебсайтів кінотеатрів і стримінгових сервісів.
- Проєктування архітектури системи на базі Next.js.
- Вибір методології розробки вебзастосунку (наприклад, Agile).
- Формулювання функціональних і нефункціональних вимог до вебзастосунку.
- Аналіз і структурування вихідних даних (каталоги фільмів, розклади, підписки тощо).
- Визначення вхідних даних, які вводять користувачі (вибір фільмів, реєстрація, оплата).
- Створення діаграм взаємодії користувачів із системою.
- Визначення інформаційного забезпечення вебплатформи.
- Обґрунтування технічного забезпечення для хостингу та розгортання системи.
- Розробка модулів для головної сторінки, каталогу фільмів, сторінки підтримки, підписки, особистого кабінету.
- Демонстрація реалізації вебсайту та його ключових функцій у реальних умовах використання.

Рік виконання кваліфікаційної бакалаврської роботи – 2025.

Рік захисту роботи – 2025.

Ключові слова: Система, код, користувач, адміністратор, сторінка

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	4
ВСТУП.....	5
РОЗДІЛ 1	8
ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	8
1.1 Характеристика предметної галузі та об'єкта дослідження.....	8
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі	10
РОЗДІЛ 2	14
РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КІНОТЕАТРУ	14
2.1 Аналіз і специфікація вимог до інформаційної системи	14
2.2 Постановка та алгоритм розв'язання задачі	22
2.2.1 Постановка задачі.....	23
2.2.2 Алгоритм розв'язання задачі.....	25
2.3 Моделювання інформаційної системи	26
2.3.1 Моделювання поведінки системи.....	26
2.3.2 Моделювання структури системи.....	28
2.3.3 Розподіл вимог за компонентами системи.....	30
РОЗДІЛ 3	33
ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ З ВИКОРИСТАННЯМ NEXTJS.....	33
3.1 Інформаційне забезпечення.....	33
3.1.1 Загальна характеристика інформаційного забезпечення.....	33
3.1.2 Організація збору і передання первинної інформації.....	34
3.1.3 Побудова системи класифікації та кодування.....	35
3.1.4 Проектування форм первинних документів та відеокадрів	35
3.1.5 Структура інформаційних масивів	36
3.1.6 Вибір СКБД.....	36
3.1.7 Інфологічна модель бази (сховища) даних	37
3.1.8 Даталогічна модель бази (сховища) даних	38
3.2 Технічне забезпечення	39
3.2.1 Загальні положення та схема автоматизації	40
3.2.2 Структура комплексу технічних засобів.....	41

3.2.3 Опис автоматизованого робочого місця	41
3.2.4 Схема мережі передачі даних.....	42
3.3 Програмне забезпечення.....	42
3.3.1 Структура програмного забезпечення.....	43
3.3.2 Системне програмне забезпечення.....	43
3.3.3 Прикладне програмне забезпечення.....	44
3.3.4 Програмна документація	44
3.4 Результати реалізації інформаційної системи	45
ВИСНОВКИ	50
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ	52

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

SSR – server side rendering

SSG – static site generation

UX – user experience

UI – user interface

ВСТУП

Актуальність тематики. За останні роки спостерігається зростання популярності та використання онлайн-кінотеатрів, що дозволяють користувачам зручно переглядати фільми та серіали в режимі реального часу. Актуальність розробки інформаційної системи для кінотеатру полягає у високій потребі в адаптивних, надійних та ефективних цифрових платформах, які відповідають сучасним вимогам користувачів до якості, швидкодії та зручності використання. Використання сучасних фреймворків, таких як NextJS, дозволяє реалізувати високопродуктивні системи з оптимізованим рендерингом і широкими можливостями масштабування. Розробка такої системи має велике значення для вдосконалення цифрових сервісів у сфері медіарозваг, популяризації легального перегляду контенту та задоволення потреб користувачів у високоякісному мультимедійному досвіді.

Аналіз останніх досліджень і публікацій. В останні роки було проведено низку досліджень та опубліковано чимало праць, присвячених тематиці розробки вебзастосунків у сфері стримінгових сервісів. У ряді робіт розглянуто переваги використання таких технологій, як SSR, SSG та JAMstack, що лежать в основі сучасних систем доставки мультимедійного контенту. Дослідження також акцентують увагу на необхідності підвищення продуктивності сайтів, поліпшення UX/UI і забезпечення надійного доступу до даних користувача. Водночас, актуальними залишаються питання побудови адаптивної архітектури інформаційної системи з урахуванням потреб глядачів, адміністраторів та служби підтримки.

Мета і завдання дослідження. Метою даного дослідження є розробка та реалізація інформаційної системи для онлайн-кінотеатру з використанням технології NextJS, що дозволить підвищити якість обслуговування користувачів, розширити доступ до мультимедійного контенту, зменшити навантаження на сервер і забезпечити безперебійну роботу системи. У рамках реалізації зазначеної мети, було сформульовано наступні завдання:

- дослідження предметної області, вивчення функціоналу сучасних онлайн-кінотеатрів;
- аналіз існуючих систем у сфері стримінгових сервісів; визначення основних функцій та вимог до інформаційної системи;
- проектування архітектури інформаційної системи;
- вибір методології моделювання архітектури; формулювання функціональних та нефункціональних вимог до системи;
- огляд вихідних даних, необхідних для належної роботи онлайн-кінотеатру;
- визначення вхідних даних, які надають користувачі при реєстрації, оформленні підписки та зверненні до служби підтримки; моделювання діаграм структури та поведінки системи за обраною методологією;
- визначення інформаційного забезпечення; вибір та обґрунтування технічного забезпечення; виокремлення програмного забезпечення та його компонентів; опис результатів реалізації інформаційної системи для кінотеатру на базі NextJS.

Об'єкт та предмет дослідження. Об'єктом даного дослідження виступає процес взаємодії користувача з цифровою платформою онлайн-кінотеатру, що охоплює перегляд контенту, оформлення підписки та звернення до технічної підтримки. Предметом дослідження виступає сукупність підходів, методів та технологій розробки інформаційної системи з використанням фреймворку NextJS.

Методи дослідження та відповідний інструментарій. Завдання полягає в розробці та впровадженні ефективної, масштабованої та адаптивної інформаційної системи онлайн-кінотеатру. У дослідженні використовувалися:

- Методи аналізу потреб користувачів:
 - аналіз конкурентних платформ: для визначення ключових функцій та очікувань користувачів було проведено вивчення провідних стримінгових сервісів;

- аналіз вимог: на основі зібраної інформації були сформульовані вимоги до функціоналу системи.
- Методи проектування програмного забезпечення:
 - реалізація продукту: на основі аналізу ринку та визначених вимог була розроблена концепція інформаційної системи;
 - обґрунтування методології проектування: була обрана методологія розробки з використанням UML-діаграм, яка найбільш ефективно відображає структуру майбутньої системи.
- Методи розробки та тестування інформаційної системи:
 - проектування архітектури та інтерфейсу користувача: на основі зібраних вимог розроблено загальну архітектуру системи з адаптивним інтерфейсом;
 - реалізація компонентів: створено окремі функціональні модулі (головна сторінка, каталог фільмів, система підписки, підтримка);
 - тестування: проведено перевірку працездатності системи на різних пристроях і сценаріях використання.

РОЗДІЛ 1

ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

Дослідження даного напрямку буде спрямоване перш за все на розробку інформаційної системи для онлайн-кінотеатру. В собі він буде орієнтоватися на взаємодію між користувачами, або ж називатимемо їх глядачами та платформою, яка в свою чергу надаватиме доступ до перегляду фільмів або шоу. Система також в собі передбачає функціонал адміністрування контенту, керування підписками та підтримку клієнтів. В глобальному розумінні необхідність впровадження такої системи є зумовленою зростанням попиту на цифрові сервіси розваг та потребою у зручному, адаптивному та досить швидкому й стабільному доступі до мультимедійного контенту з будь-якого пристрою.

Предметна галузь буде охоплювати сучасні інтернет-платформи, що забезпечують легальний стримінг відеоконтенту. Ключовими аспектами цієї галузі є:

- Відеоконтент – це будуть фільми, серіали, шоу та інші розважальні або інформаційні відеоматеріали, які в свою чергу будуть доступні для перегляду через мережу Інтернет;
- Система підписки – це є модель монетизації, яка передбачає оплату користувачем за доступ до певного переліку функцій або контенту (на певний період);
- Цифрова дистрибуція – спосіб поширення медіаконтенту в основу якого закладена доставка до користувача виключно в електронному вигляді.

Учасниками процесу взаємодії із системою визначаються:

- Користувач (глядач) – особа, яка переглядає фільми, оформлює підписку,

звертається до служби підтримки.

- Адміністратор системи – особа, яка додає/редагує контент, керує тарифами та переглядає звернення користувачів.

Процес користування інформаційною системою кінотеатру складається з таких етапів:

- Огляд каталогу – користувач знайомиться з доступним контентом, переглядає категорії, добірки, популярні жанри, фільми які наразі рахуються найпереглядаючими.
- Вибір фільму або шоу – користувач відкриває сторінку конкретного фільму з описом, акторами, рейтингом, відгуками.
- Оформлення підписки – для доступу до повного функціоналу користувач обирає тарифний план і оплачує його, це дозволить йому переглядати відеоконтент в кращій якості, завантажувати його і т.п.
- Перегляд контенту – користувач дивиться відео у відповідному плеєрі.
- Використання підтримки – за потреби звертається до служби підтримки через спеціальну форму на сайті.

Основні характеристики предметної галузі:

- Економічний зміст – галузь цифрової трансляції контенту передбачає надання доступу до фільмів, серіалів та шоу за моделлю підписки або безкоштовного перегляду з рекламою. Вона охоплює як правовласників контенту, так і платформу, що забезпечує його дистрибуцію кінцевим користувачам.
- Проблемні ситуації та цілі розв'язання – серед актуальних проблем – складність організації адаптивного і зручного інтерфейсу, потім на черзі йде стабільна трансляція контенту на різних пристроях в реальному часі, оптимізація продуктивності вебплатформи. Мета дослідження – розробити рішення, яке дозволяє швидко, інтуїтивно і безперебійно взаємодіяти з мультимедійною системою, скорочуючи витрати на обслуговування і збільшуючи задоволеність користувача.
- Рішення та учасники процесу – ключові рішення в системі ухвалюють

користувачі (вибір тарифу, контенту) та адміністратори (наприклад такі як додавання фільмів, їх редагування, обробка звернень, надання певних відповідей на відгуки глядачів). Вони взаємодіють через вебінтерфейс, що має бути швидким, безпечним та адаптивним.

- Інформаційні аспекти – для ефективної взаємодії з платформою користувач повинен мати доступ до структурованої інформації про фільми (жанр, опис, актори, відгуки), тарифні плани, терміни дії підписки та способи звернення до підтримки. Вся інформація зберігається у базі даних та відображається динамічно, відповідно до запитів користувача.

1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

Впродовж усього останнього часу активно зростає цікавість до впровадження сучасних цифрових рішень у сферу стримінгових сервісів, зокрема — у формі вебзастосунків для перегляду відеоконтенту. У публікаціях, присвячених даній тематиці, часто розглядаються питання організації структури онлайн-кінотеатрів, удосконалення користувацького досвіду, забезпечення високої продуктивності та адаптивності, а також підвищення ефективності доставки мультимедіа. Проблематика оптимізації процесів перегляду відео, підключення платіжних систем, керування контентом та комунікації з користувачами посідає важливе місце в багатьох дослідженнях.

Фокусування та вивчення саме українського сегменту ринку цифрових сервісів свідчить про те, що подібні платформи в більшості реалізовані у вигляді повнофункціональних вебсайтів або мобільних застосунків. Більшість із них дозволяє оформити підписку, переглядати відео у високій якості, отримувати рекомендації та звертатися до технічної підтримки. Щоб краще зрозуміти актуальні тенденції, було проаналізовано декілька платформ, які вже тривалий час

представлені на ринку:

- Megogo — одна з найпопулярніших українських платформ з перегляду фільмів і телебачення онлайн. Вона пропонує широкий вибір контенту, різні тарифні плани, можливість підключення до кількох пристроїв одночасно (рис. 1.1).

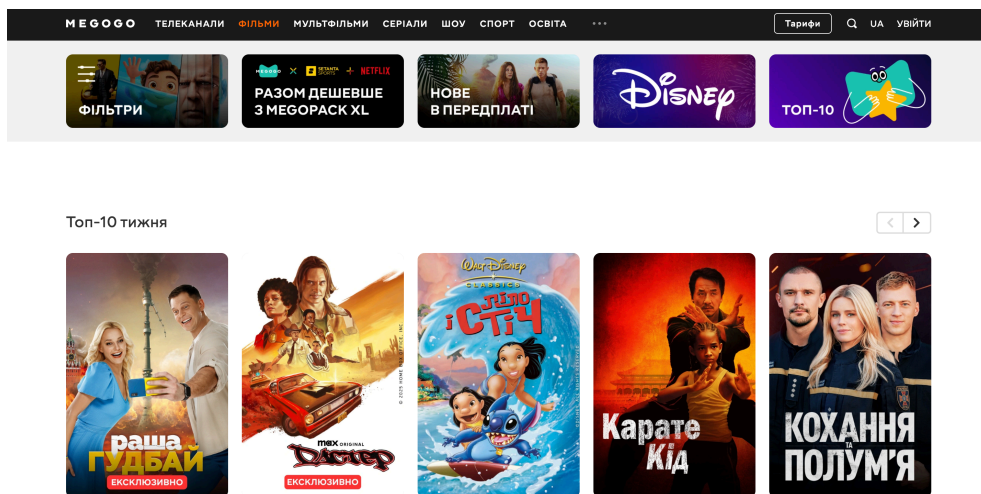


Рисунок 1.1 – Megogo головна сторінка

Джерело: знімок екрану створено автором

- Sweet.tv — сервіс, орієнтований на легальний перегляд відео, з інтерфейсом, адаптованим для української аудиторії. Підтримує перегляд як на телевізорах, так і на смартфонах, має вбудовану систему добірок і рекомендацій (рис. 1.2).

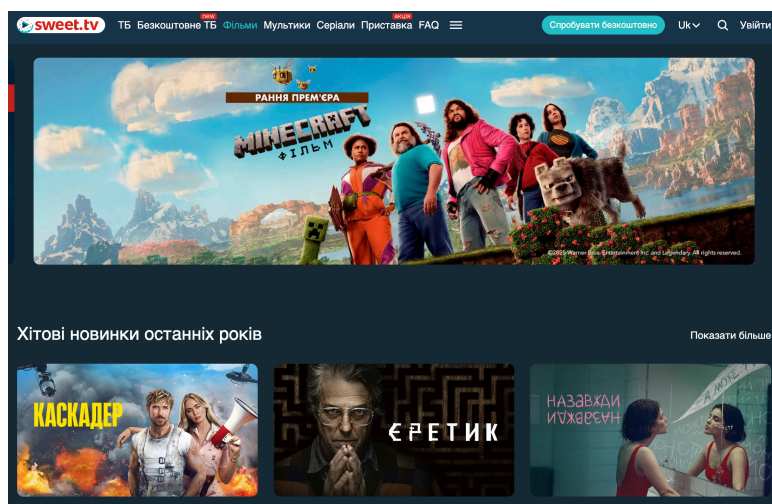


Рисунок 1.2 – Sweet.tv головна сторінка

Джерело: знімок екрану створено автором

- OmegaTV — онлайн-платформа, яка поєднує можливість перегляду телеканалів та фільмів. В основі її функціоналу — простота використання та мінімалістичний інтерфейс (рис. 1.3).

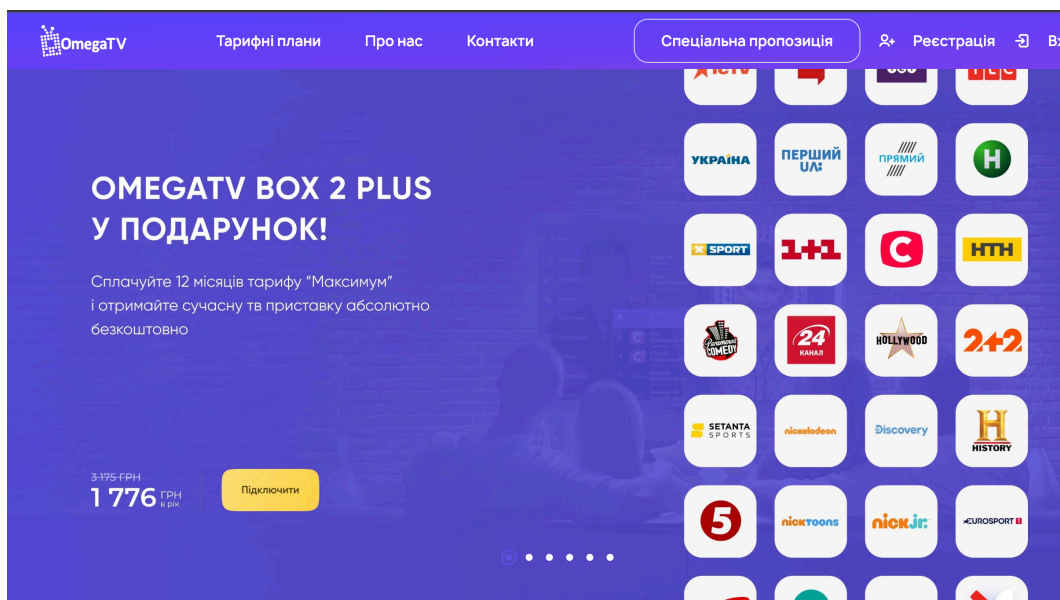


Рисунок 1.3 – OmegaTV головна сторінка

Джерело: знімок екрану створено автором

- Kyivstar TV — сервіс від одного з найбільших мобільних операторів, що поєднує можливості перегляду телеканалів і фільмів. Має вигідні пакети для абонентів мобільного зв'язку (рис. 1.4).

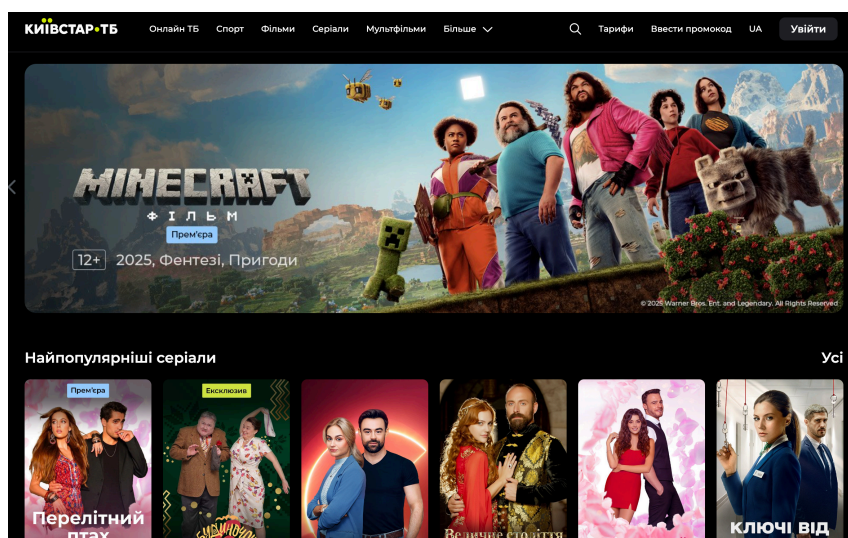


Рисунок 1.4 – Kyivstar TV головна сторінка

Джерело: знімок екрану створено автором

Звісно варто відокремити сильні та слабкі сторони кожного з рішень. Для цього було складено узагальнену порівняльну табл. 1.1, у якій відображено основні технічні та функціональні характеристики платформ.

Таблиця 1.1 – Порівняльна характеристика цифрових платформ у сфері онлайн-кінотеатрів

Назва	Megogo[1]	Sweet.tv[2]	OmegaTV[3]	Kyivstar TV[4]
Основне призначення	Стрімінг фільмів та ТВ	Онлайн-кінотеатр	Телебачення + VOD	Комбінований сервіс
Каталог контенту	Широкий вибір відео	Актуальні фільми	Переважно ТБ та архів	Великий вибір контенту
Авторизація	За бажанням	За бажанням	Не обов'язкова	Обов'язкова
Підписка	Так	Так	Так	Так
Інтерфейс	Інтуїтивний, адаптивний	Зрозумілий, сучасний	Мінімалістичний	З брендованим оформленням
Повідомлення користувача	Email, push	Сповіщення, рекомендації	Обмежено	Повідомлення в застосунку
Додаткові можливості	Персоналізовані добірки	Контент для дітей	Запис ефірів	Програма ТБ, система лояльності
Зручність	Висока	Висока	Середня	Висока

Джерело: розроблено автором на основі [1-4]

У результаті аналізу було визначено кілька ключових функцій, які варто реалізувати в рамках власної інформаційної системи:

- Простий та зрозумілий інтерфейс із швидким доступом до каталогу фільмів і шоу;
- Підтримка різних варіантів підписок з інтеграцією платіжних систем в майбутньому;
- Можливість отримувати повідомлення про нові відомості про світ фільмів, акції або завершення дії підписки;
- Система рекомендацій на основі історії переглядів;
- Доступ до служби підтримки через спеціальну форму або розділ FAQ.

Саме так, платформи демонструють успішне впровадження цифрових технологій у сфері стрімінгових сервісів і можуть слугувати джерелом ідей для створення власного рішення, адаптованого під вимоги та очікування користувачів.

РОЗДІЛ 2

РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КІНОТЕАТРУ

2.1 Аналіз і специфікація вимог до інформаційної системи

Бізнес-вимоги визначають основні очікування, цілі та обмеження платформи для досягнення її функціональної цінності, задоволення потреб користувачів і забезпечення технічної керованості (рис. 2.1)[5].

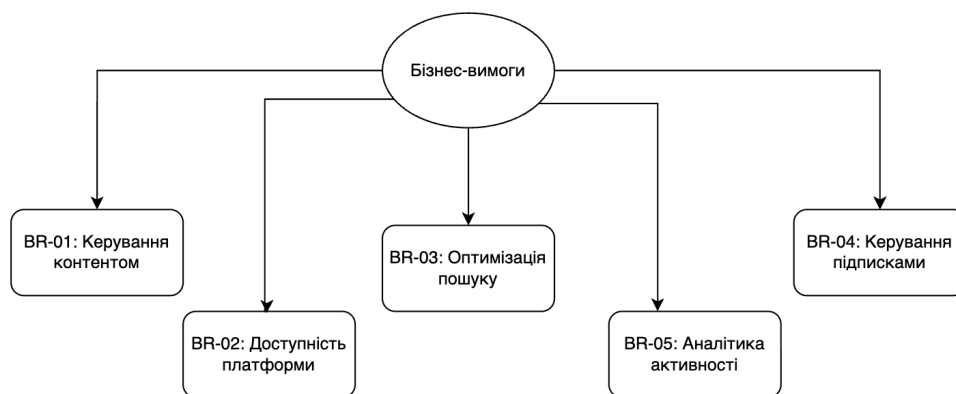


Рисунок 2.1 - Діаграма бізнес-вимог

Джерело: знімок екрану створено автором

Порівняльна характеристика цифрових платформ у сфері онлайн-кінотеатрів звізуалізована у табл 2.1.

Таблиця 2.1 – Порівняльна характеристика цифрових платформ у сфері онлайн-кінотеатрів.

ID	Стисла назва	Повна назва	Тип	Статус	Складність	Пріоритетність	Ризик	Примітки
----	--------------	-------------	-----	--------	------------	----------------	-------	----------

BR-01	Керування контентом	Автоматизація додавання, редагування та видалення фільмів	Бізнес - вимога	Затверджено	Середня	Висока	Середній	Виконується адміністратором
BR-02	Доступність платформи	Цілодобовий доступ до фільмів через авторизований вебінтерфейс	Бізнес - вимога	Затверджено	Низька	Висока	Низький	Потрібен SSR/CSR рендеринг
BR-03	Оптимізація пошуку	Зручний пошук фільмів за жанром, роком, рейтингом	Бізнес - вимога	Затверджено	Середня	Висока	Середній	Інтеграція з динамічними фільтрами
BR-04	Керування підписками	Реалізація платного/безкоштовного доступу до фільмів через підписку	Бізнес - вимога	Затверджено	Середня	Середня	Високий	Потрібна авторизація та зберігання статусу
BR-05	Аналітика активності	Надання адміністраторам звітів про перегляди фільмів і популярність категорій	Бізнес - вимога	Затверджено	Висока	Середня	Низький	Використовується Chart.js

Опис бізнес-вимог:

- BR-01: Керування контентом дозволяє адміністратору керувати каталогом фільмів без участі розробника, через зручний інтерфейс.
- BR-02: Доступність платформи забезпечується безперервною роботою фронтенду та серверної частини.

- BR-03: Оптимізація пошуку дозволяє користувачам фільтрувати контент за потрібними параметрами — жанр, рік, рейтинг, популярність.
- BR-04: Керування підписками дає змогу диференціювати доступ до фільмів залежно від активної/неактивної підписки користувача.
- BR-05: Аналітика активності дає адміністраторам змогу аналізувати дані про популярність фільмів, частоту переглядів і поведінку користувачів.

Функціональні вимоги визначають конкретні функції, які повинна виконувати інформаційна система для перегляду фільмів, щоб задовольнити потреби користувачів. Вони сформульовані відповідно до стандарту представлені у вигляді ієрархії та зв'язані з бізнес-вимогами (рис. 2.2) [6].

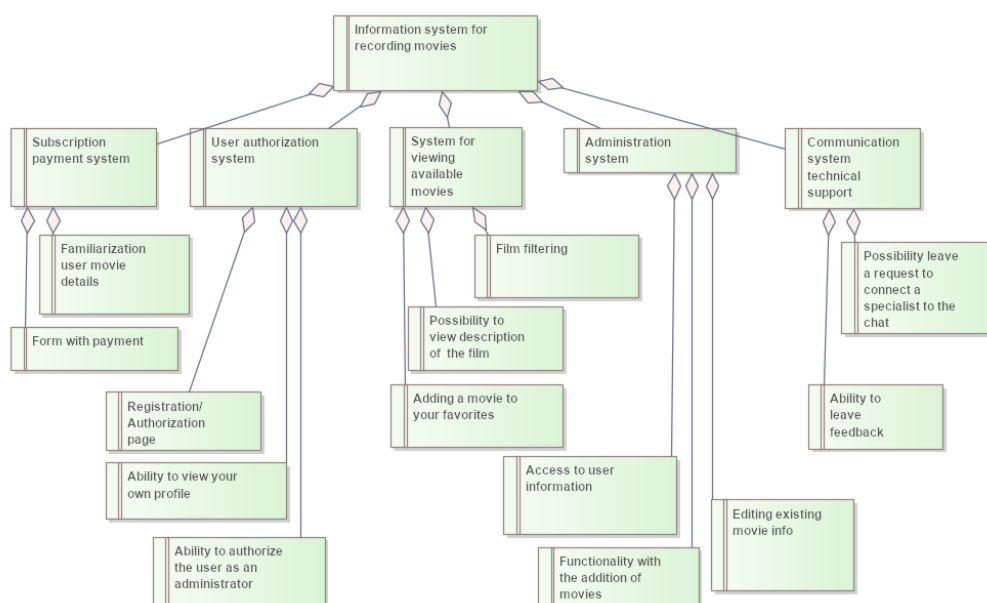


Рисунок 2.2 - Діаграма функціональних вимог

Джерело: знімок екрану створено автором

Специфікація функціональних вимог була викладена у табл. 2.2.

Таблиця 2.2 – Специфікація функціональних вимог

ID	Стисла назва	Повна назва	Тип	Статус	Складність	Пріоритет	Ризик	Примітки
FR-01	Авторизація	Реєстрація, вхід та	Функціональна	Затверджено	Середня	Високий	Середній	Здійснюється

ID	Стисла назва	Повна назва	Тип	Статус	Складність	Пріоритет	Ризик	Примітки
		вихід користувача, збереження сесії						через Firebase/Auth з підтримкою ролей
FR-02	Перегляд фільмів	Доступ до каталогу фільмів з можливістю сортування та пошуку	Функціональна	Затверджено	Середня	Високий	Середній	SSR/CSR, можливість фільтрації за жанром, роком, рейтингом
FR-03	Управління фільмами	Додавання, редагування та видалення фільмів адміністратором	Функціональна	Затверджено	Середня	Високий	Середній	Доступне лише адміну; потребує підтвердження при видаленні
FR-04	Оформлення підписки	Вибір тарифу, оплата та активація доступу до контенту	Функціональна	Затверджено	Середня	Високий	Середній	Інтегрується з платіжною системою; зберігає статус користувача
FR-05	Статус підписки	Перевірка активності підписки при зверненні до обмеженого контенту	Функціональна	Затверджено	Низька	Середній	Низький	Реалізується через middleware або on-demand перевірку
FR-06	Пошук	Пошук фільмів	Функціональна	Затверджено	Низька	Середній	Низький	Інтеграція з

ID	Стисла назва	Повна назва	Тип	Статус	Складність	Пріоритет	Ризик	Примітки
		за ключовими словами						локальним індексом або зовнішнім пошуковим API
FR-07	Аналітика	Перегляд статистики переглядів та категорій фільмів	Функціональна	Затверджено	Середня	Середній	Низький	Реалізовано через Chart.js або власні компоненти
FR-08	Генерація звітів	Формування звітів про користувачів, перегляди, категорії	Функціональна	Затверджено	Середня	Середній	Низький	Для адміністраторів; табличний і графічний вигляд
FR-09	Експорт звітів	Експорт звітів у форматі CSV для офлайн-аналізу	Функціональна	Затверджено	Низька	Низький	Низький	Інтегрується з FR-08
FR-10	Техпідтримка	Зв'язок користувача з адміністрацією платформи	Функціональна	Затверджено	Середня	Середній	Середній	Включає форму звернення, email/чат, обробку скарг

Опис функціональних вимог:

1. FR-01: Авторизація дозволяє користувачам створювати облікові записи, входити в систему та зберігати сесію.
2. FR-02: Перегляд фільмів включає можливість бачити перелік, фільтрувати та переглядати детальну інформацію про фільм.

3. FR-03: Управління фільмами включає CRUD-операції з контентом (тільки для адміністратора).
4. FR-04: Оформлення підписки забезпечує оплату та активацію обраного плану.
5. FR-05: Статус підписки визначає доступ до обмеженого контенту згідно з активною підпискою.
6. FR-06: Пошук забезпечує швидке знаходження фільмів за назвою або ключовими словами.
7. FR-07: Аналітика дає змогу бачити статистику взаємодії користувачів із платформою.
8. FR-08: Генерація звітів дозволяє створювати адміністративні звіти у зручному форматі.
9. FR-09: Експорт звітів дозволяє зберігати інформацію для зовнішньої обробки.
10. FR-10: Техпідтримка дає змогу користувачу звернутися до адміністрації через форму або чат.

Нефункціональні вимоги визначають якісні характеристики інформаційної системи електронної бібліотеки, такі як продуктивність, безпека, зручність використання та масштабованість. Вони сформульовані відповідно до стандарту SysML (ISO/IEC 19514:2017) і представлені у вигляді ієрархії вимог із зв'язками до бізнес-вимог та інших елементів моделі. Для організації використано діаграми пакетів, а специфікацію вимог підготовлено за допомогою менеджера специфікацій у форматі, сумісному з Enterprise Architect [7]. Вимоги є недвозначними, контрольованими, ясними, виконуваними, елементарними та необхідними. Результат у табл. 2.3.

Таблиця 2.3 – Специфікація нефункціональних вимог

ID	Стисла назва	Повна назва	Тип	Статус	Складність	Пріоритет	Ризик	Примітки
NFR-01	Продуктивність	Забезпечення високої швидкості	Нефункціональна	Затверджено	Висока	Висока	Високий	Час обробки запитів до 2 секунд

ID	Стисла назва	Повна назва	Тип	Статус	Складність	Пріоритет	Ризик	Примітки
		роботи системи						
NF R-02	Час відповіді	Час відповіді на пошуковий або навігаційний запит не більше 1 секунди	Нефункціональна	Затверджено	Середня	Висока	Середній	Залежить від оптимізації SSR/CSR
NF R-03	Пропускна здатність	Підтримка до 100 одночасних активних користувачів та запитів	Нефункціональна	Затверджено	Середня	Середня	Середній	Вимагає налаштування інфраструктури та кешування
NF R-04	Безпека	Захист даних користувачів і обмеження доступу до адміністративних функцій	Нефункціональна	Затверджено	Висока	Висока	Високий	Підтримка HTTPS, JWT, валідації і ролей
NF R-05	Аутентифікація	Надійна авторизація та аутентифікація з розмежуванням прав доступу	Нефункціональна	Затверджено	Середня	Висока	Низький	Здійснюється через Firebase або власну систему
NF R-06	Захист даних	Шифрування чутливих даних у базі та при передачі	Нефункціональна	Затверджено	Висока	Висока	Високий	Використовується HTTPS, шифрування на стороні сервера
NF R-07	Зручність	Інтерфейс повинен бути зручним і приємним для користувача	Нефункціональна	Затверджено	Середня	Висока	Низький	Відповідність сучасним принципам UX/UI

ID	Стисла назва	Повна назва	Тип	Статус	Складність	Пріоритет	Ризик	Примітки
NFR-08	Інтуїтивність	Система повинна бути зрозумілою без потреби у навчанні	Нефункціональна	Затверджено	Низька	Середня	Низький	Перевіряється на етапі юзабіліті-тестування
NFR-09	Масштабованість	Система має адаптуватися до зростання кількості користувачів та контенту	Нефункціональна	Затверджено	Висока	Середня	Високим	Підтримується через хмарну інфраструктуру
NFR-10	Підтримка навантаження	Стабільна робота системи при наявності до 10,000 активних користувачів	Нефункціональна	Затверджено	Середня	Середня	Середній	Потребує горизонтального масштабування

Опис нефункціональних вимог:

1. NFR-01: Продуктивність забезпечує швидке завантаження сторінок і обробку запитів для комфортного користування.
2. NFR-02: Час відповіді гарантує, що фільми, категорії та результати пошуку відкриваються швидко.
3. NFR-03: Пропускна здатність дозволяє системі обслуговувати велику кількість запитів одночасно.
4. NFR-04: Безпека передбачає захист як даних користувачів, так і адміністративної панелі.
5. NFR-05: Аутентифікація обмежує доступ до захищених розділів сайту, залежно від ролі користувача.
6. NFR-06: Захист даних включає шифрування при збереженні та передачі даних.
7. NFR-07: Зручність передбачає логічне розміщення елементів інтерфейсу, адаптацію до різних пристроїв.

8. NFR-08: Інтуїтивність дозволяє користувачу самостійно розібратися в навігації без навчання.
9. NFR-09: Масштабованість гарантує стабільну роботу системи при зростанні обсягів контенту і відвідувачів.
10. NFR-10: Підтримка навантаження дозволяє системі витримувати велике навантаження без втрати стабільності.

2.2 Постановка та алгоритм розв'язання задачі

Комплекс завдань, що ставиться перед розробкою даної інформаційної системи, пов'язаний з автоматизацією процесів взаємодії між користувачем і онлайн-кінотеатром. Метою є зробити роботу з платформою простою та зрозумілою: від першого входу на сайт і перегляду фільмів до оформлення підписки та звернень до технічної підтримки.

Для ефективної реалізації функціоналу системи необхідно чітко визначити, яка саме інформація передається між користувачами та платформою. Зокрема, йдеться про дані, які відображаються користувачу: інформація про фільми, жанри, рейтинг, відгуки, статус підписки, а також підтвердження дій, як-от оплата або завершення перегляду. Важливо, щоб ці дані були структурованими, відображались у відповідному місці та надходили вчасно.

Крім того, велике значення має вхідна інформація — тобто те, що користувач вводить під час взаємодії з системою. Наприклад, це вибір фільму чи шоу, введення платіжних реквізитів, даних для реєстрації, оформлення підписки або надсилання звернення. Також до вхідних даних належать дії адміністратора: додавання або редагування контенту, оновлення тарифних планів, зміна статусу підписки, перегляд запитів від користувачів.

Окрему роль відіграє внутрішнє зберігання даних. Інформаційна система повинна підтримувати збереження таких об'єктів, як користувацькі профілі, активні та завершені підписки, історія перегляду, записи звернень до служби підтримки, а також повна база даних фільмів із описами, жанрами, акторами та

рейтингами. Завдяки цьому вдається забезпечити стабільну роботу системи, підтримувати цілісність даних, реалізувати персоналізацію контенту, а також аналізувати поведінку користувачів з метою подальшого розвитку платформи.

2.2.1 Постановка задачі

Моделювання поведінки системи демонструє, як система взаємодіє з ключовими акторами — користувачами та адміністраторами, і як реалізуються її функціональні можливості. На основі функціональних вимог було побудовано такі артефакти:

1. Діаграма прецедентів (Use Case diagram) – відображає основні сценарії взаємодії користувачів та адміністраторів із системою, включаючи вхід у систему, перегляд фільмів, оформлення підписки, звернення до техпідтримки, керування фільмами [8].
2. Текстові сценарії прецедентів – деталізовано описують послідовність дій у межах кожного варіанту використання (наприклад, «Перегляд фільму», «Оплата підписки» або «Додавання фільму адміністратором») [9].
3. Діаграма послідовності (Sequence diagram) – описує обмін повідомленнями між користувачем, інтерфейсом і логікою системи під час виконання типових операцій, таких як авторизація або перегляд відео [10].
4. Діаграма діяльності (Activity diagram) – демонструє логіку виконання дій користувача або адміністратора в межах одного процесу, наприклад, «Підписка на план», «Додавання нового фільму» або «Обробка звернення до техпідтримки» [11].

Застосування цих діаграм дозволяє краще зрозуміти потік дій у системі, роль кожного користувача, обробку подій та загальну логіку функціонування вебплатформи.

Діаграма прецедентів узагальнює функціональні вимоги системи,

представляючи їх як варіанти використання (use cases), що описують типову взаємодію акторів із системою (рис. 2.3).

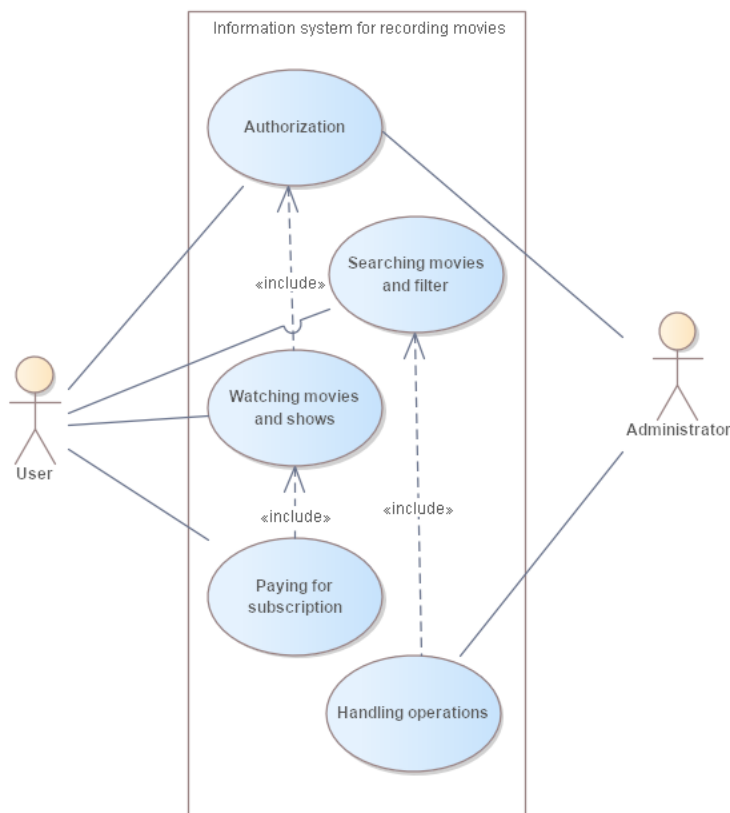


Рисунок 2.3 - Діаграма прецедентів

Джерело: знімок екрану створено автором

Опис діаграми:

- Актори:
 1. User (Користувач) — взаємодіє із системою для перегляду контенту, пошуку фільмів, оформлення підписки та авторизації.
 2. Administrator (Адміністратор) — відповідає за обслуговування платформи та керування її наповненням.
- Основні прецеденти:
 1. Authorization (Авторизація): Користувач проходить авторизацію (реєстрацію або вхід), щоб отримати персоналізований доступ до функцій платформи.

2. Searching movies and filter (Пошук і фільтрація фільмів): Доступний як для користувача, так і для адміністратора. Дозволяє знаходити фільми за жанром, роком, назвою тощо.
3. Watching movies and shows (Перегляд фільмів і серіалів): Основний функціонал користувача, що дає змогу переглядати доступний контент. Прецедент включає пошук фільмів та оплату підписки.
4. Paying for subscription (Оплата підписки): Дозволяє користувачу обрати план підписки, здійснити оплату та активувати доступ до преміум-контенту.
5. Handling operations (Адміністрування операцій): Реалізується адміністратором, включає керування контентом: моніторинг роботи системи. Прецедент включає функції пошуку фільмів.

2.2.2 Алгоритм розв'язання задачі

Діаграма послідовності деталізує взаємодію об'єктів системи для реалізації прецеденту "Пошук фільмів" (рис. 2.4).

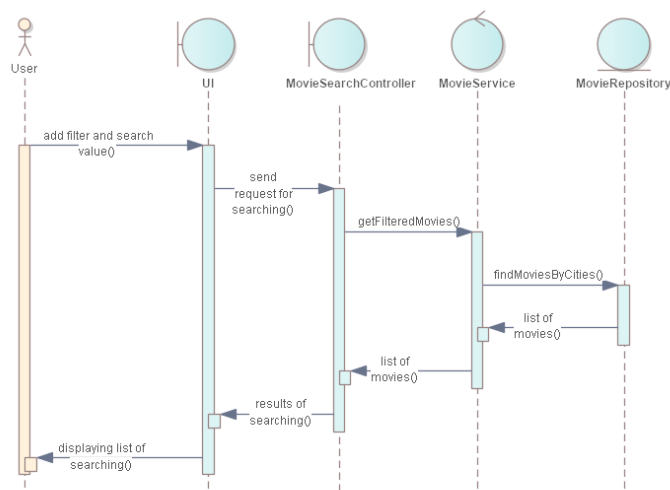


Рисунок 2.4 - Діаграма послідовності

Джерело: знімок екрану створено автором

На діаграмі зображено процес взаємодії між користувачем і внутрішніми компонентами вебплатформи під час виконання пошуку фільмів. Діаграма ілюструє, як запит, ініційований користувачем, проходить через усі рівні системи, включно з контролером, сервісом і репозиторієм, перш ніж повернути результат.

2.3 Моделювання інформаційної системи

Моделювання структури системи описує статичну організацію інформаційної системи електронної бібліотеки, включаючи її компоненти, їх взаємозв'язки та ієрархію. Відповідно до методичних вказівок, структура моделюється за допомогою діаграм SysML (стандарт ISO/IEC 19514:2017) та UML, зокрема діаграми визначення блоків, діаграми внутрішніх блоків і діаграми класів. Ці артефакти підготовлено з використанням Enterprise Architect, забезпечуючи прозорість, зрозумілість і відповідність вимогам стандартів.

2.3.1 Моделювання поведінки системи

Діаграма класів UML моделює статичну структуру інформаційної системи для перегляду фільмів, відображаючи сутності (класи), їх атрибути, методи та зв'язки між ними. Система логічно структурована навколо основних функцій: авторизація, підписка, перегляд фільмів, історія пошуку та облік оплат. Кожен клас відповідає окремій логічній сутності системи — користувач, фільм, підписка, платіж, відгук — та має чітко визначені зв'язки із суміжними об'єктами. Особливу увагу приділено правильному визначенню типів зв'язків: асоціації, агрегації та композиції, що дозволяє забезпечити гнучкість і масштабованість архітектури.

Такий підхід полегшує подальшу реалізацію бізнес-логіки та інтеграцію з базою даних. (рис. 2.5).

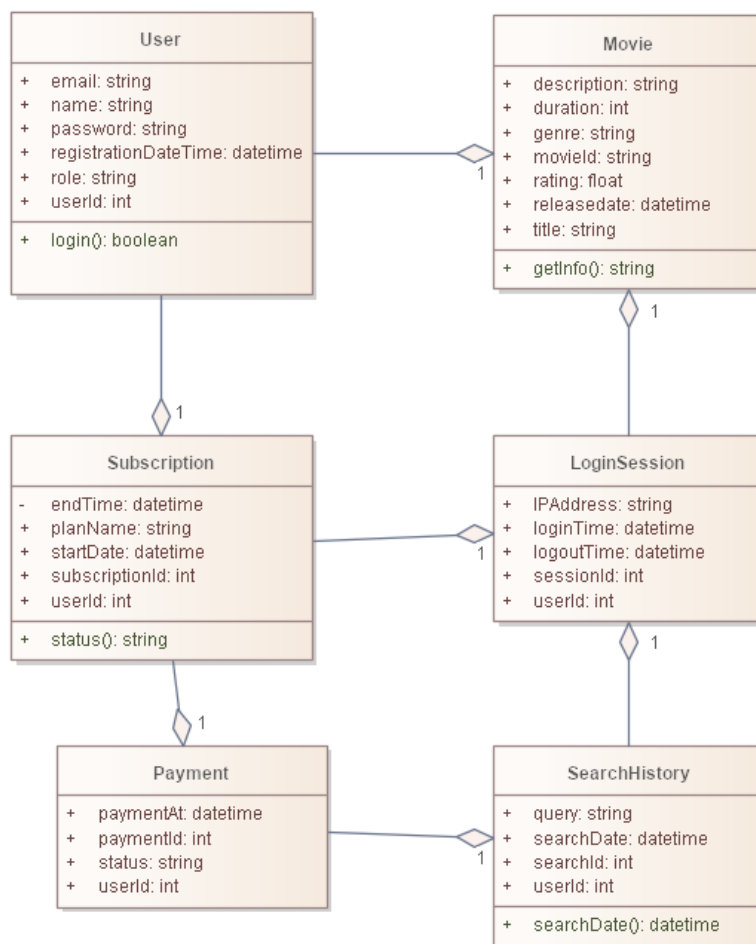


Рисунок 2.5 - Діаграма класів

Джерело: знімок екрану створено автором

1. Клас User. Містить дані про користувача: email, ім'я, пароль, дату реєстрації, роль та ідентифікатор. Має метод login(), що відповідає за авторизацію. Зв'язаний з усіма іншими класами як основний суб'єкт взаємодії.
2. Клас Movie. Моделює фільм як мультимедійний об'єкт: назва, опис, жанр, тривалість, рейтинг, дата релізу тощо. Має метод getInfo() для отримання метаданих.
3. Клас Subscription. Відповідає за керування активними підписками користувача. Містить дату початку/завершення, назву тарифу та метод

status() для перевірки активності. Зв'язаний із користувачем та історією оплат.

4. Клас Payment. Фіксує факт оплати підписки: дата, ідентифікатор платежу, статус та посилання на користувача.
5. Клас LoginSession. Моделює інформацію про сесію авторизованого користувача: IP-адреса, час входу й виходу, ідентифікатор сесії. Дозволяє вести облік активностей.
6. Клас SearchHistory. Фіксує історію пошукових запитів користувача: ключові слова, дата пошуку, зв'язок з користувачем. Містить метод searchDate() для отримання дати останнього запиту.

2.3.2 Моделювання структури системи

Діаграма внутрішніх блоків SysML (IBD) демонструє внутрішню структуру інформаційної системи для перегляду фільмів, включаючи основні підсистеми, модулі та їхню взаємодію [12]. Діаграма розкриває, з яких частин складається система, та як між ними здійснюється передача даних і функціональна взаємодія (рис. 2.6).

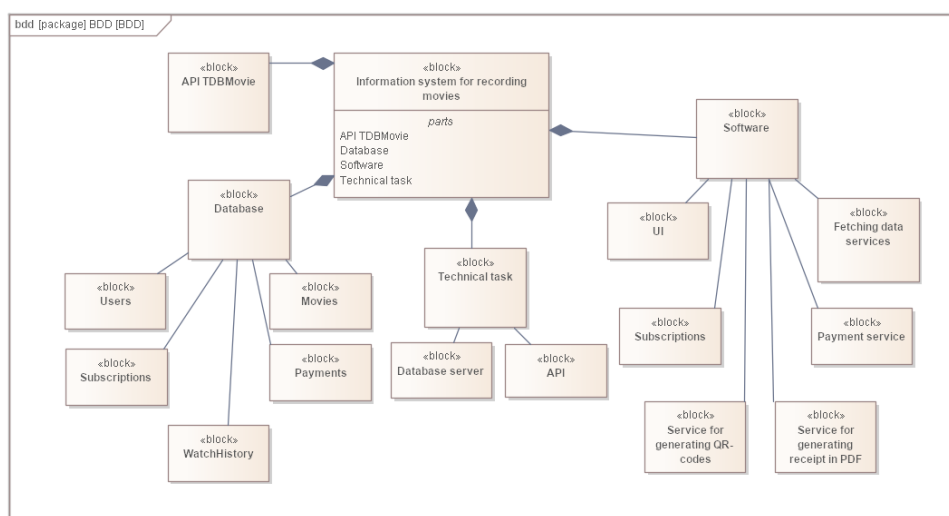


Рисунок 2.6 - Діаграма внутрішніх блоків

Джерело: знімок екрану створено автором

Опис діаграми:

Information system for recording movies — головна система, що складається з чотирьох основних частин:

1. API TDBMovie — зовнішній API для отримання інформації про фільми.
2. Database — підсистема зберігання та обробки даних.
3. Software — прикладна логіка, що відповідає за інтерфейс, сервіси оплати, генерацію чеків і інші функції.
4. Technical task — вимоги та інфраструктурна частина (сервер, API, документація).

Компоненти підсистем:

1. Database:

- Users — інформація про зареєстрованих користувачів.
- Subscriptions — записи про підписки.
- WatchHistory — історія переглядів користувачів.
- Movies — основна база фільмів.
- Payments — інформація про здійснені оплати.

2. Software:

- UI — користувацький інтерфейс платформи.
- Subscriptions — модуль для керування підписками.
- Fetching data services — служба отримання даних (у тому числі з API або бази).
- Payment service — інтеграція з платіжною системою.
- Service for generating QR-codes — генерація QR-кодів (наприклад, для чеків).
- Service for generating receipt in PDF — формування чеків/рахунків у форматі PDF.

3. Technical task:

- Database server — сервер бази даних.
- API — внутрішній API для взаємодії між модулями.

Також можна виділити наступні потоки та зв'язки:

1. Модулі пов'язані через внутрішні порти, що відповідають за передачу даних між компонентами (наприклад, між базою даних, інтерфейсом і сервісами).
2. Інтеграція з зовнішнім API TDBMovie використовується для отримання детальної інформації про фільми, яка може не зберігатись локально.
3. Система підтримує окремі сервіси для генерації чеків і візуалізації оплат, що дозволяє масштабувати функціональність незалежно від основного ядра.

2.3.3 Розподіл вимог за компонентами системи

Зважаючи на важливість інтуїтивно зрозумілого інтерфейсу в інформаційних системах, орієнтованих на широке коло користувачів. Було створено прототипи ключових екранів системи, кожен із яких виконує визначену роль у взаємодії користувача з функціональністю платформи. Основні екрани було змодельовано з урахуванням реальної логіки переходів між сторінками (рис. 2.7).

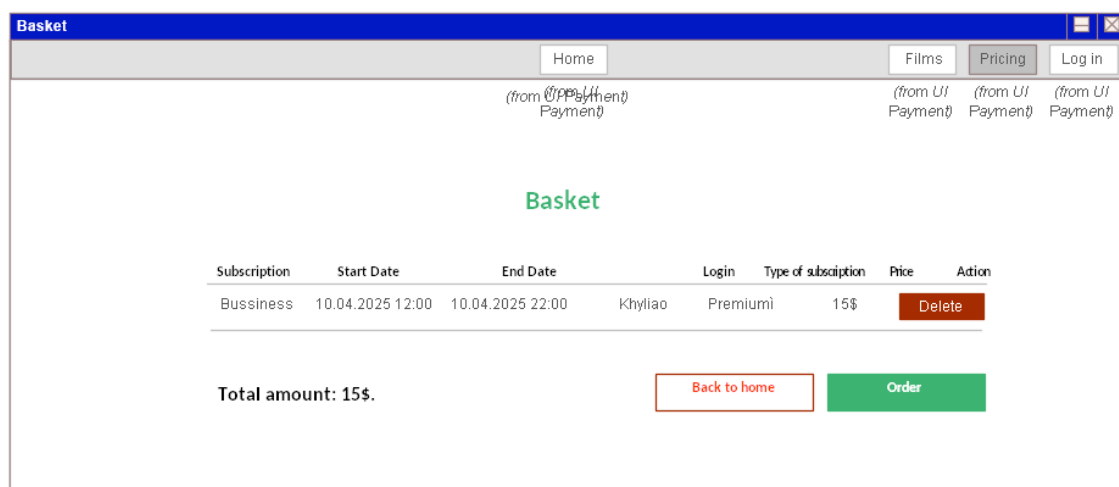
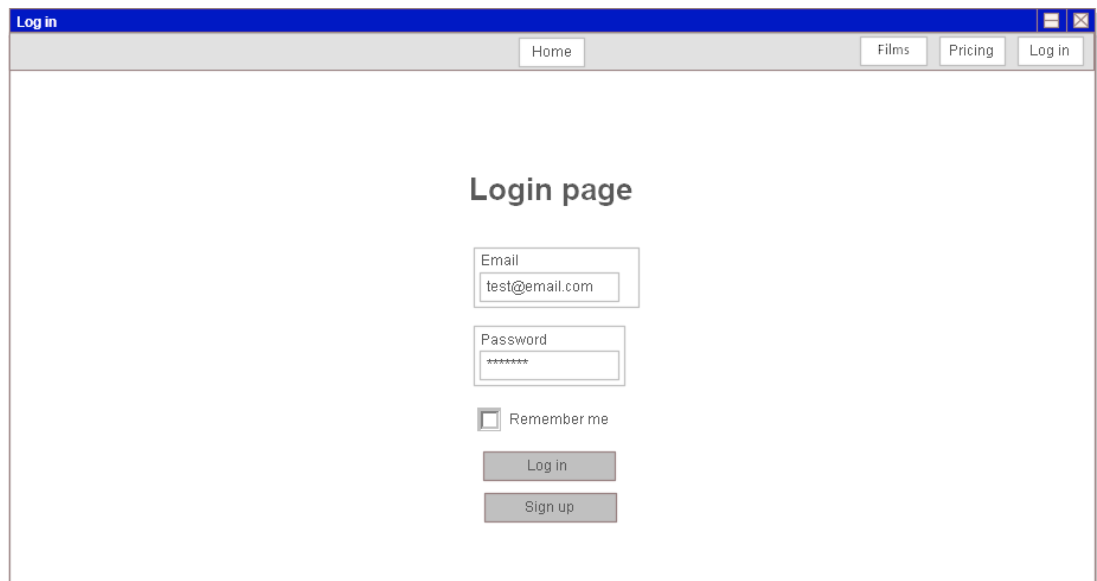


Рисунок 2.7 – Сторінка корзини

Джерело: знімок екрану створено автором

Сторінка авторизації (рис. 2. 8).

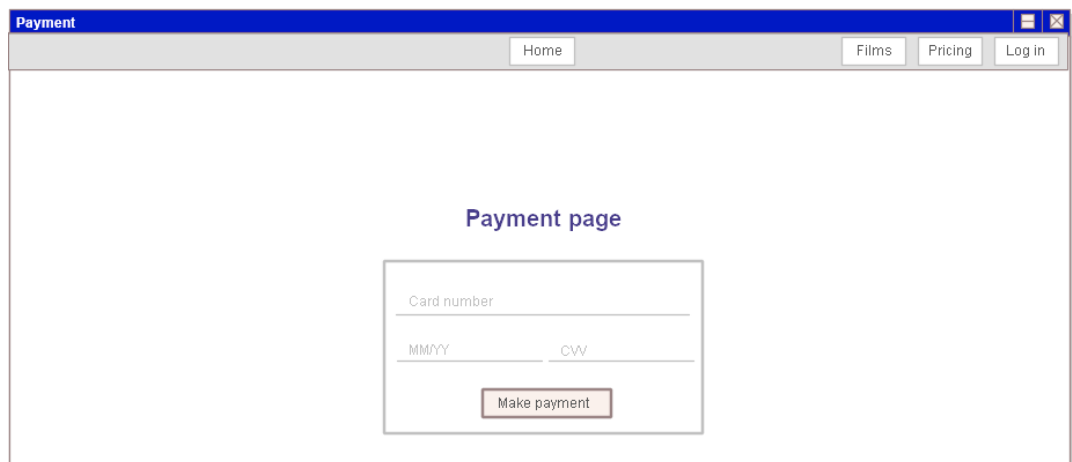


The screenshot shows a web browser window titled "Log in". The browser's address bar is empty. The page has a navigation bar with links: "Home", "Films", "Pricing", and "Log in". The main content area is titled "Login page". It contains a form with two input fields: "Email" (with the text "test@email.com") and "Password" (with masked characters "*****"). Below the password field is a checkbox labeled "Remember me". At the bottom of the form are two buttons: "Log in" and "Sign up".

Рисунок 2.8 – Сторінка авторизації

Джерело: знімок екрану створено автором

Сторінка оплати підписки (рис. 2. 9).



The screenshot shows a web browser window titled "Payment". The browser's address bar is empty. The page has a navigation bar with links: "Home", "Films", "Pricing", and "Log in". The main content area is titled "Payment page". It contains a form with three input fields: "Card number", "MMYY", and "CVV". Below these fields is a button labeled "Make payment".

Рисунок 2.9 – Сторінка оплати

Джерело: знімок екрану створено автором

Структура інтерфейсу включає:

1. Головне меню (присутнє на всіх сторінках): кнопки навігації до розділів Home, Films, Pricing, Log in.

2. Login page: дозволяє користувачу¹ авторизуватися або зареєструватися. Реалізована форма з полями для email та пароля, кнопкою “Remember me”, а також кнопками Log in і Sign up.
3. Basket: показує активну підписку користувача з зазначенням її типу, дати початку і завершення, вартості. Реалізовано кнопки Delete, Back to home та Order для подальших дій. Сторінка автоматично підраховує загальну суму до сплати.
4. Payment page: сторінка для введення платіжних даних. Має інтерфейс із полями введення номера картки, дати та CVV-коду. Кнопка Make payment ініціює обробку платежу.

Принципи, закладені в дизайн:

1. UX/UI-дружність: великі кнопки, очевидна навігація, відсутність перевантаження текстом.
2. Доступність: інтерфейс виглядає однаково на десктопах і планшетах, адаптований під мобільні пристрої.
3. Колірна ієрархія: зелений — підтвердження дії, червоний — видалення або скасування, сірий — неактивні елементи.
4. Модульність: кожна сторінка реалізована як окремий компонент, що спрощує подальше масштабування або редизайн.

РОЗДІЛ 3

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ З ВИКОРИСТАННЯМ NEXTJS

3.1 Інформаційне забезпечення

У рамках даного проєкту для організації та обробки інформації використовується вебфреймворк Next.js та хмарна платформа Firebase, які забезпечують ефективну інтеграцію клієнтської частини застосунку з серверними сервісами без потреби в окремому сервері чи локальній базі даних.

Firebase виконує роль основної системи керування базами даних (СКБД), а саме використовуються сервіси Realtime Database для зберігання структурованих даних, Firebase Authentication — для автентифікації користувачів, а також Vercel для розміщення вебзастосунку.

Realtime Database зберігає інформацію про користувачів та фільми, що дозволяє забезпечити централізоване збереження та доступ до даних у режимі реального часу. Завдяки вбудованим механізмам синхронізації, усі зміни в базі даних автоматично оновлюються на клієнтській стороні.

Взаємодія з користувачем реалізована через інтерфейс, створений з використанням Next.js, який забезпечує динамічну маршрутизацію, серверний рендеринг та високу швидкодію застосунку. Це дозволяє користувачам легко переглядати фільми.

3.1.1 Загальна характеристика інформаційного забезпечення

Інформаційне забезпечення вебзастосунку для онлайн-перегляду фільмів

охоплює сукупність даних, структур, методів їх обробки та збереження, необхідних для ефективного функціонування системи. Основними видами інформації, які обробляються системою, є облікові записи користувачів, повідомлення службі підтримки, а також метадані про фільми й серіали.

Дані зберігаються у структурованому форматі JSON у базі даних Firebase Realtime Database, що забезпечує оперативний обмін інформацією між клієнтською частиною застосунку і сервером. Оскільки архітектура сайту передбачає використання ролей (адміністратор і користувач), відповідні записи супроводжуються інформацією про доступ та права.

Інформаційна структура була спроектована з урахуванням розширюваності: у майбутньому планується інтеграція з медіасховищем, де будуть зберігатись обкладинки, трейлери та інші медіаелементи, пов'язані з контентом.

3.1.2 Організація збору і передання первинної інформації

Первинна інформація вноситься до системи користувачем під час взаємодії з інтерфейсом. Важливими каналами збору є форма автентифікації (email/пароль) та форма звернення до служби підтримки (рис. 3.1).

Рисунок 3.1 – Форма звернення до тех. підтримки

Джерело: знімок екрану створено автором

Передача даних здійснюється у вигляді асинхронних HTTP-запитів із використанням вбудованих можливостей Firebase SDK. Завдяки цьому зміни миттєво синхронізуються між клієнтом та сервером у реальному часі, без необхідності ручного оновлення сторінки.

3.1.3 Побудова системи класифікації та кодування

Усі об'єкти в системі мають унікальні ідентифікатори, які присвоюються автоматично під час запису в базу даних. Для класифікації ролей користувачів використовується логічний прапорець `role`, який приймає значення `user` або `admin`.

Також реалізована структурована організація даних: кожна сутність зберігається в окремій гілці бази даних. Наприклад:

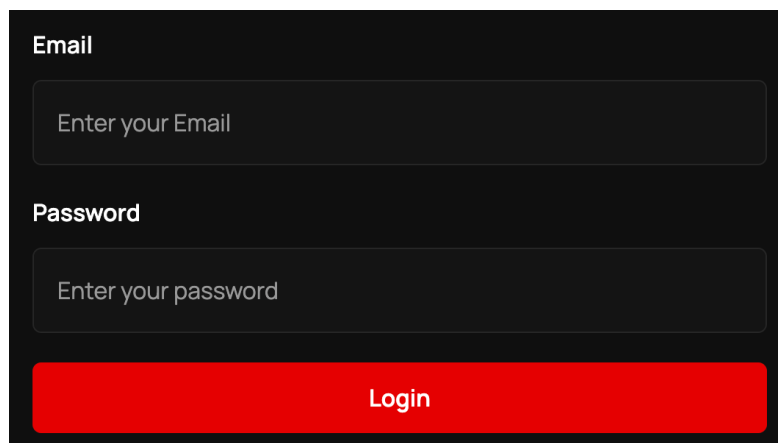
- `/users/{userId}` — облікові записи
- `/messages/{messageId}` — звернення користувачів
- `/movies/{movieId}` — фільми та серіали

3.1.4 Проектування форм первинних документів та відеокадрів

Формат введення інформації до системи має бути зручним, інтуїтивно зрозумілим та забезпечувати цілісність даних. Для цього форми реалізовані за допомогою бібліотеки `react-hook-form`, яка також дозволяє здійснювати валідацію полів без додаткових залежностей.

Користувач вводить `email`, `password` та після цього натискає на кнопку для входу. Всі поля обов'язкові, а помилки відображаються одразу після взаємодії з

полем (рис. 3.2).



The image shows a dark-themed login form. At the top, there is a label 'Email' above a text input field with the placeholder 'Enter your Email'. Below this is a label 'Password' above another text input field with the placeholder 'Enter your password'. At the bottom of the form is a large red button with the text 'Login' in white.

Рисунок 3.2 – Сторінка оплати

Джерело: знімок екрану створено автором

3.1.5 Структура інформаційних масивів

Інформаційні масиви зберігаються в ієрархічній структурі, притаманній Realtime Database. Кожна сутність має свій власний вузол у дереві даних, що забезпечує швидкий доступ до потрібної інформації (рис 3.3).

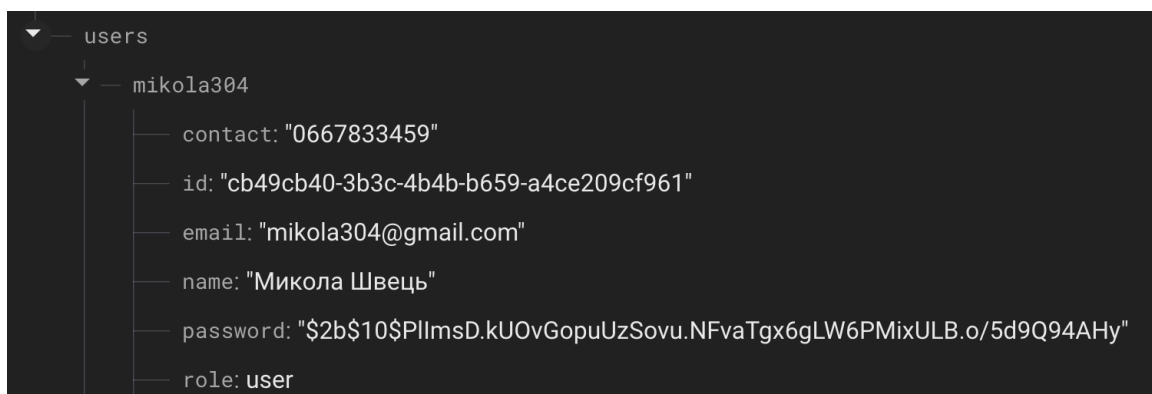


Рисунок 3.3 – Зображення схеми users в базі даних

Джерело: знімок екрану створено автором

3.1.6 Вибір СКБД

Для зберігання даних обрано Firebase Realtime Database, яка надає масштабовану хмарну інфраструктуру, що дозволяє легко розгортати застосунок без налаштування окремого сервера баз даних.

Серед переваг:

- Підтримка обміну в реальному часі
- Простий у використанні SDK
- Безкоштовний тариф для невеликих проєктів
- Висока інтеграція з іншими сервісами Firebase

Вибір обґрунтований саме потребами односторінкового застосунку (SPA), розробленого за допомогою Next.js.

3.1.7 Інфологічна модель бази (сховища) даних

Інфологічна модель описує сутності та їх зв'язки на логічному рівні. Основними сутностями є:

- Користувач (User)
- Звернення (SupportMessage)
- Фільм/Шоу (Movie)

Зв'язки між сутностями:

- Користувач → може надіслати → багато звернень
- Адміністратор → має доступ → до перегляду всіх звернень
- Користувач → може переглядати → фільми/шоу

Вся інфологічна модель разом дозволяє ефективно організувати дані та забезпечити їх логічну узгодженість у межах інформаційної системи. Вона дозволяє відокремити ролі користувачів та реалізувати доступ до функціоналу згідно з цими ролями. Структура даних підтримує масштабованість системи в майбутньому, зокрема для додавання нових функцій, таких як система збереження

обраного чи переглянутого контенту. Такий підхід спрощує супровід та розвиток системи, зберігаючи чіткість взаємозв'язків між ключовими об'єктами (рис. 3.4).

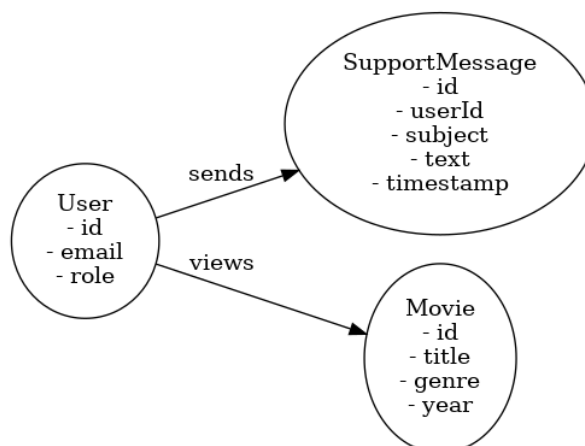


Рисунок 3.4 – Зображення схеми users в базі даних

Джерело: знімок екрану створено автором

3.1.8 Даталогічна модель бази (сховища)даних

Даталогічна модель деталізує, як саме інформація зберігається у структурі JSON в Firebase. У ній зазначаються ключі, типи даних та приклади вкладених об'єктів (рис 3.5).

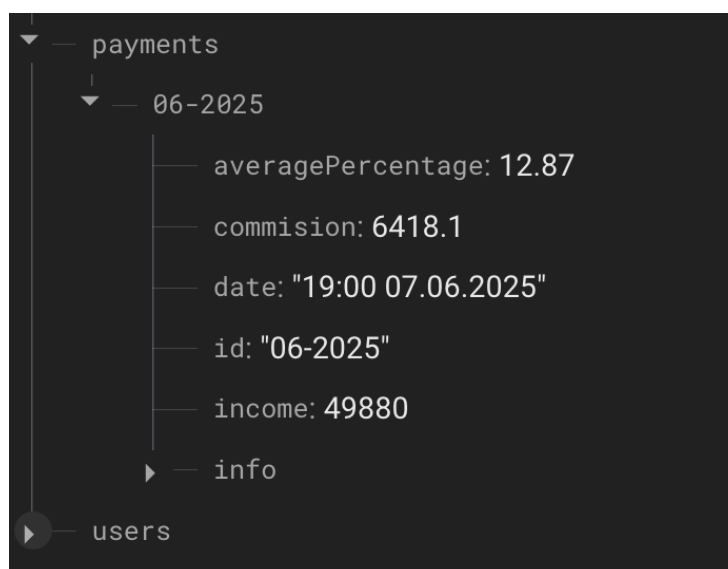


Рисунок 3.5 – Зображення даталогічної моделі в базі даних

Джерело: знімок екрану створено автором

3.2 Технічне забезпечення

Інформаційна система для кінотеатру реалізована на основі сучасних хмарних технологій без використання власного сервера. Застосунок розроблено з використанням Next.js, а хостинг і backend функціональність реалізовані за допомогою Firebase та Vercel.

Складові системи:

- Next.js – фреймворк для клієнтської та серверної частини (рендеринг сторінок, маршрутизація, взаємодія з API);
- Vercel – хмарна платформа для деплою Next.js проєкту;
- Firebase Authentication – для реєстрації, авторизації користувачів та розмежування доступу (роль "клієнт", "адміністратор");
- Firebase Realtime Database – для зберігання інформації про фільми, підписки, користувачів, перегляди тощо.

Система повністю працює у браузері та не потребує встановлення додаткового ПЗ.

Мінімальні вимоги до комп'ютера або ноутбука:

- Операційна система: Windows 7+, macOS, Linux;
- Браузер: Google Chrome, Firefox, Safari, Edge (актуальна або передостання версія);
- Процесор: від Intel Core i3 або аналогічний;
- Оперативна пам'ять (RAM): від 2 ГБ;
- Інтернет: стабільне з'єднання.

Сайт адаптований до мобільних пристроїв і доступний через браузер.

iOS:

- Версія iOS 9.0 або новіша;
- Safari або інший браузер, сумісний з WebKit;

- Мінімум 100 МБ вільного місця.

Android:

- Android 5.0 або новіший;
- Будь-який сучасний браузер (Chrome, Firefox);
- Від 512 МБ оперативної пам'яті;
- Мінімум 100 МБ вільного місця.

Особливості реалізації

- Безсерверна архітектура: не вимагає налаштування або адміністрування фізичного сервера;
- Масштабованість: автоматично забезпечується за рахунок Firebase та Vercel;
- Надійність: усі дані зберігаються у Firebase Realtime Database з автоматичним резервним копіюванням та захистом;
- Доступність: користувачі можуть входити з будь-якого пристрою за допомогою логіна/пароля або інших методів автентифікації (наприклад, Google Sign-In).

3.2.1 Загальні положення та схема автоматизації

Інформаційна система для кінотеатру призначена для автоматизації процесу перегляду фільмів і шоу в онлайн-форматі, з можливістю керування вмістом через адміністраторську панель. Вона надає користувачам зручний інтерфейс для ознайомлення з каталогом фільмів, перегляду описів, відгуків, а також для оформлення підписки на перегляд контенту.

Система реалізована як веб-застосунок на основі фреймворку Next.js, що забезпечує високу продуктивність, SEO-оптимізацію та адаптивну верстку. Сайт має дві основні ролі користувачів — адміністратор та звичайний користувач. Адміністратор має змогу додавати, редагувати або видаляти фільми, які

відображаються користувачам. Користувач має доступ до інформації про фільми, може переглядати їх описи, рейтинги та відгуки.

Контент фільмів автоматично імпортується з API TMDb (The Movie Database), що забезпечує актуальність та різноманітність даних без необхідності ручного введення кожного фільму.

3.2.2 Структура комплексу технічних засобів

Система функціонує в середовищі клієнт-серверної архітектури. Клієнтська частина — це веб-застосунок, доступний через сучасні браузеры на всіх пристроях (десктоп, планшет, смартфон). Серверна частина обробляє запити користувачів, відповідає за авторизацію, отримання та обробку даних з TMDb, керування ролями та контентом.

До складу технічного комплексу входять:

- Хостинг-сервер, де розміщено backend-логіку і базу даних;
- CDN або хостинг для frontend-частини на основі Next.js;
- Особисті пристрої користувачів з доступом до Інтернету;
- Браузер з підтримкою JavaScript

3.2.3 Опис автоматизованого робочого місця

Автоматизоване робоче місце адміністратора складається з комп'ютера з доступом до адміністративної панелі сайту. Через цю панель адміністратор може:

- Додавати нові фільми, використовуючи API TMDb або власноруч;
- Редагувати наявні записи про фільми (опис, жанри, актори);
- Видаляти фільми з каталогу;

- Переглядати запити користувачів до служби підтримки.

Для користувача автоматизоване робоче місце — це будь-який пристрій з браузером, через який він може:

- Переглядати доступні фільми й шоу;
- Ознайомлюватися з описом, рецензіями, акторами;
- Оформлювати підписку;
- Надсилати запити до техпідтримки.

3.2.4 Схема мережі передачі даних

Схема передачі даних передбачає взаємодію між користувачем, сервером застосунку та зовнішнім API TMDb. Користувач надсилає запити через інтерфейс браузера, які обробляються сервером. У разі потреби сервер звертається до TMDb API для отримання або оновлення інформації про фільми, і після обробки надсилає дані назад клієнту.

3.3 Програмне забезпечення

Система розроблена з використанням сучасних веб-технологій, які забезпечують швидкодію, масштабованість та зручність у використанні. Основу становить Next.js, який дозволяє реалізувати як серверний рендеринг (SSR), так і генерацію статичних сторінок (SSG) для оптимізації швидкості завантаження сайту.

Серед основних технологій також використовуються:

- Firebase для авторизації користувачів та зберігання інформації;
- Tailwind CSS для стилізації інтерфейсу;

- TypeScript для забезпечення стабільності коду;
- TMDb API як основне джерело інформації про фільми.

3.3.1 Структура програмного забезпечення

Програмне забезпечення складається з наступних логічних модулів:

- Frontend (Next.js):
 - Головна сторінка з підбілками фільмів;
 - Каталог фільмів і шоу;
 - Сторінка з детальною інформацією про фільм;
 - Форма підписки;
 - Форма зворотного зв'язку.
- Backend (API-обробники Next.js + Firebase):
 - Авторизація користувачів;
 - Обробка форм (підтримка, підписка);
 - Управління фільмами (адміністратор).
- Інтеграція з TMDb API: імпорт та оновлення інформації про фільми.

3.3.2 Системне програмне забезпечення

Серед системного ПЗ, необхідного для функціонування системи:

- Операційна система серверу: Ubuntu / Node.js середовище виконання;
- Веб-сервер: Vercel або інший хостинг для Next.js застосунку;
- Firebase Authentication для реєстрації та збереження користувачів;
- СУБД (як частина Firebase або інша): для зберігання інформації про підписки, ролі та відгуки.

3.3.3 Прикладне програмне забезпечення

До прикладного ПЗ належить сам веб-застосунок, реалізований за допомогою:

- Next.js — фреймворк для React;
- Tailwind CSS — утилітарна бібліотека стилів;
- Axios / fetch API — для звернення до TMDB API;
- Firebase SDK — для авторизації та зберігання даних користувачів.

Функціональні можливості обмежені базовим набором:

- Реєстрація / вхід;
- Перегляд каталогу фільмів;
- Перегляд опису фільму;
- Сторінка підтримки;
- Адмін-панель для керування контентом.

3.3.4 Програмна документація

У межах розробки було створено наступні документи:

- Файл з вимогами до системи (функціональні, нефункціональні);
- Use-case діаграми та специфікації, реалізовані в Enterprise Architect;
- Інструкція користувача (для користувача та адміністратора);
- README-файл із описом розгортання системи;
- Коментарі до коду згідно зі стандартами TypeScript та React.

3.4 Результати реалізації інформаційної системи

Реалізація головної сторінки з хедером, логотипом, та секцією-героем (рис. 3.6).

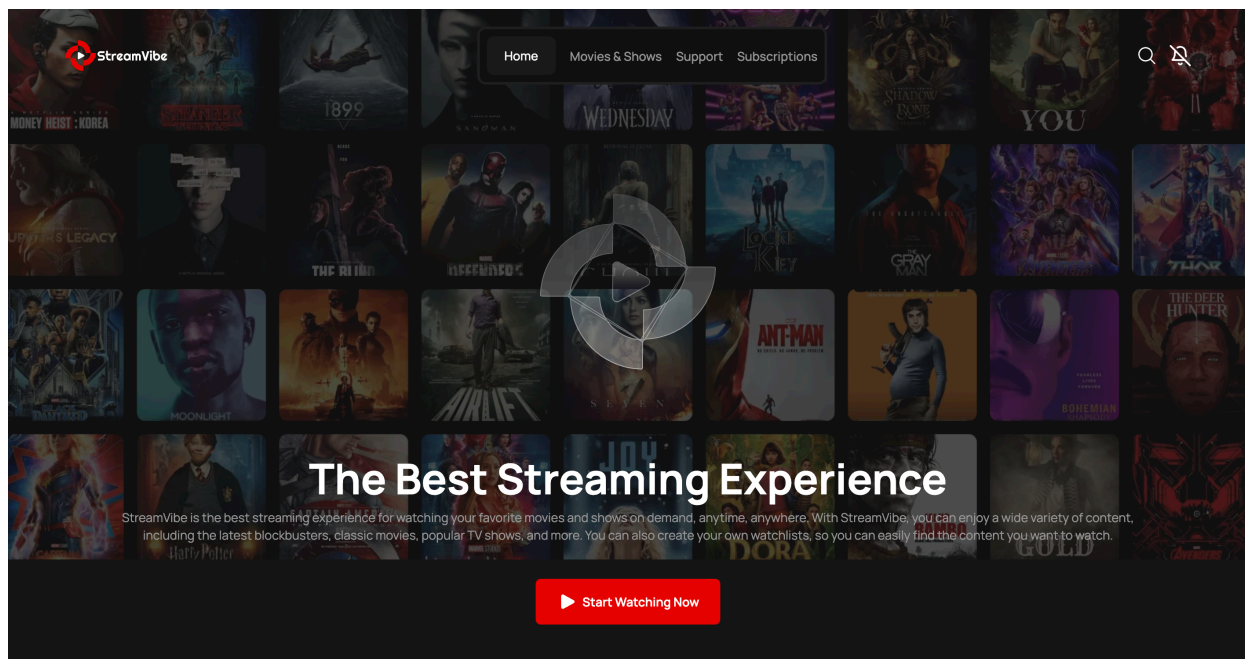


Рисунок 3.6 – Зображення головної сторінки

Джерело: знімок екрану створено автором

Реалізація категорії фільмів, переваг платформи (рис. 3.7).

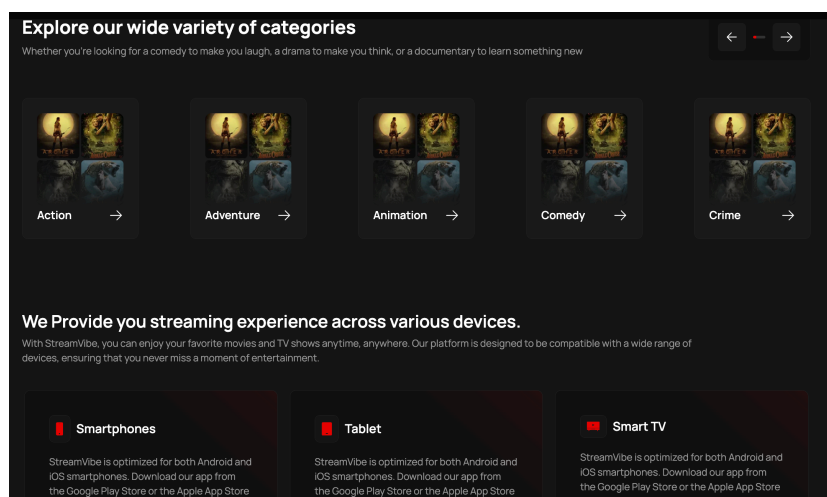


Рисунок 3.7 – Зображення категорій фільмів та переваг платформи

Джерело: знімок екрану створено автором

Реалізація компоненту часто задаваних питань (рис. 3.8).

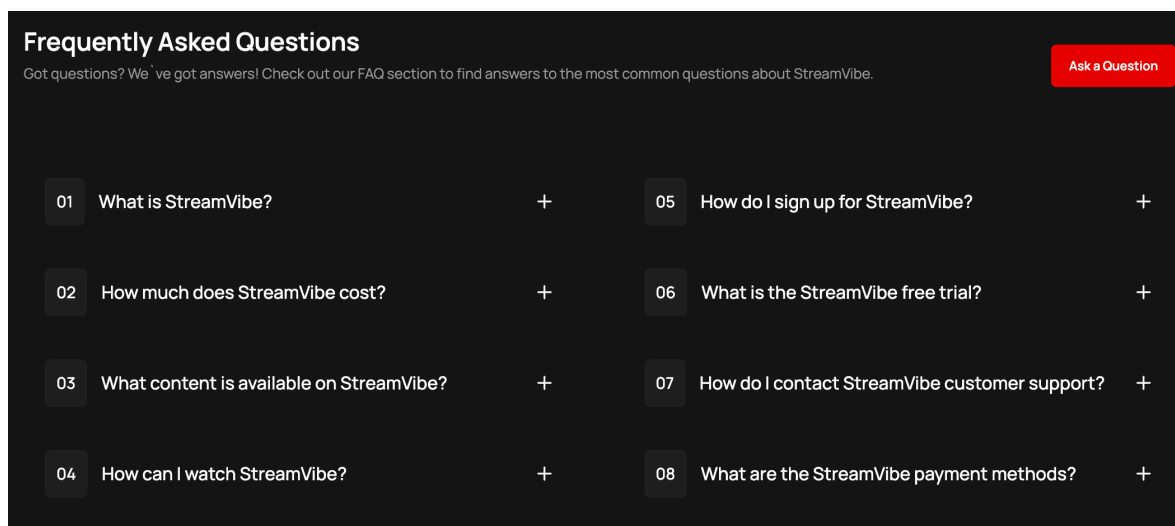


Рисунок 3.8 – Зображення часто задаваних питань

Джерело: знімок екрану створено автором

Реалізація компонента з доступними тарифами для користувача (рис. 3.9).

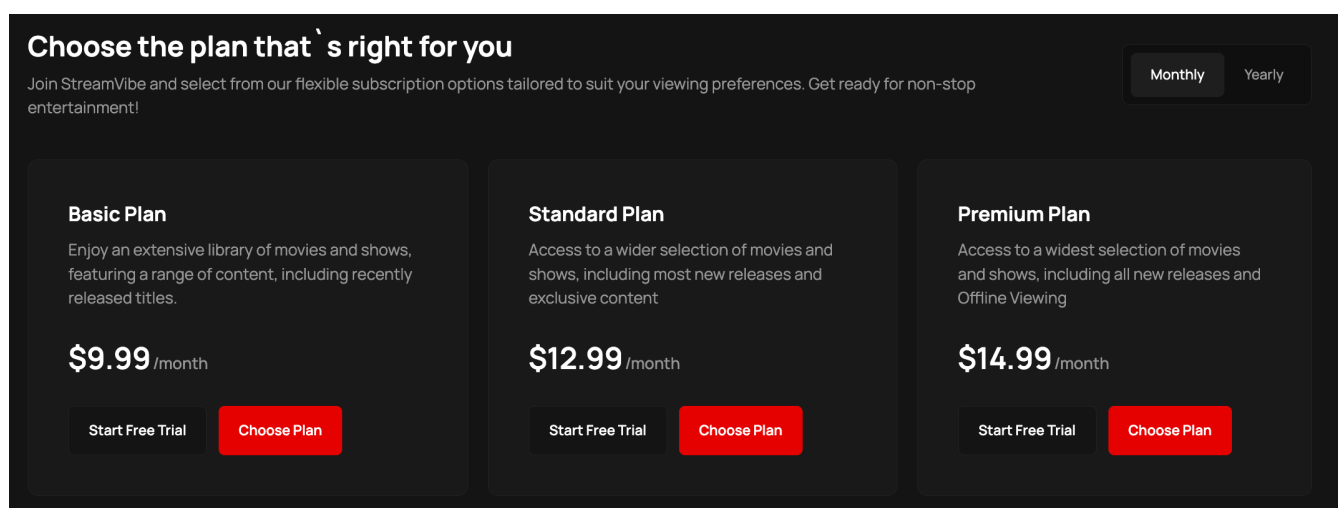


Рисунок 3.9 – Компонент з доступними тарифами

Джерело: знімок екрану створено автором

Реалізація компонента з рекламним банером (рис. 3.10).

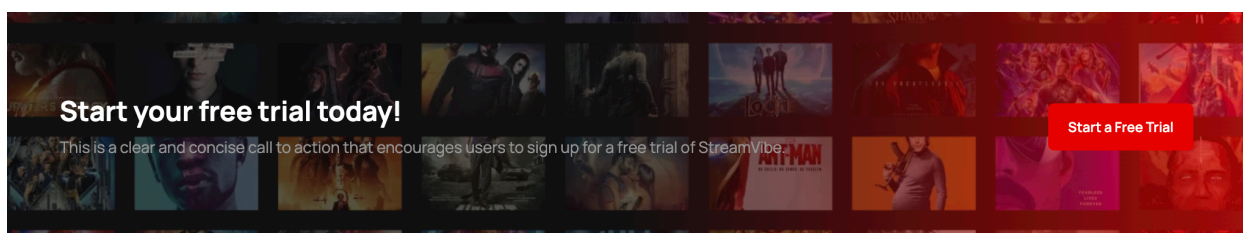


Рисунок 3.10 – Компонент з рекламним банером

Джерело: знімок екрану створено автором

Реалізація футера вебсайту (рис. 3.11).

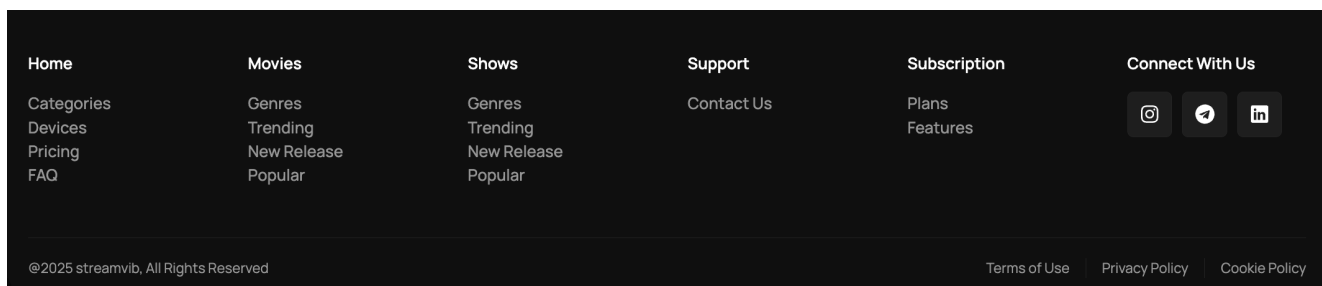


Рисунок 3.11 – Футер вебсайту

Джерело: знімок екрану створено автором

Реалізація головної сторінки Movies & Shows (рис. 3.12).

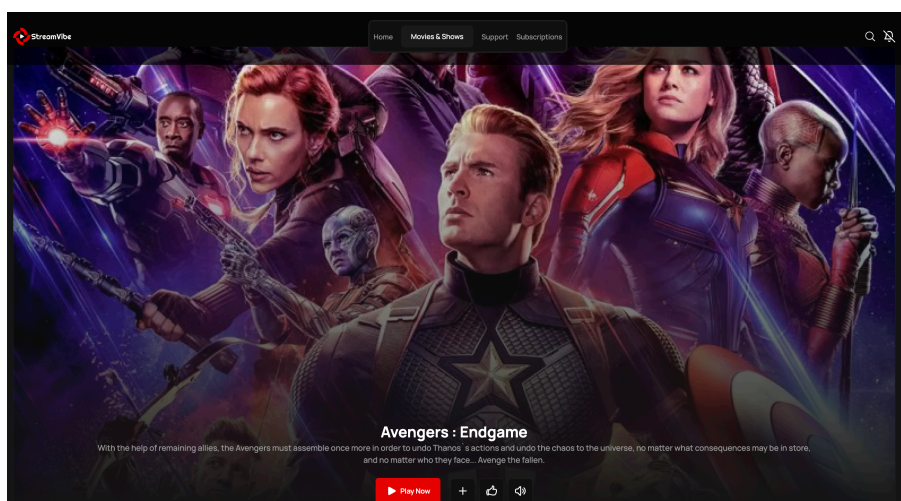


Рисунок 3.12 – Головна сторінка Movies & Shows

Джерело: знімок екрану створено автором

Реалізація каруселі з топ 10 популярними фільмами (рис. 3.13).

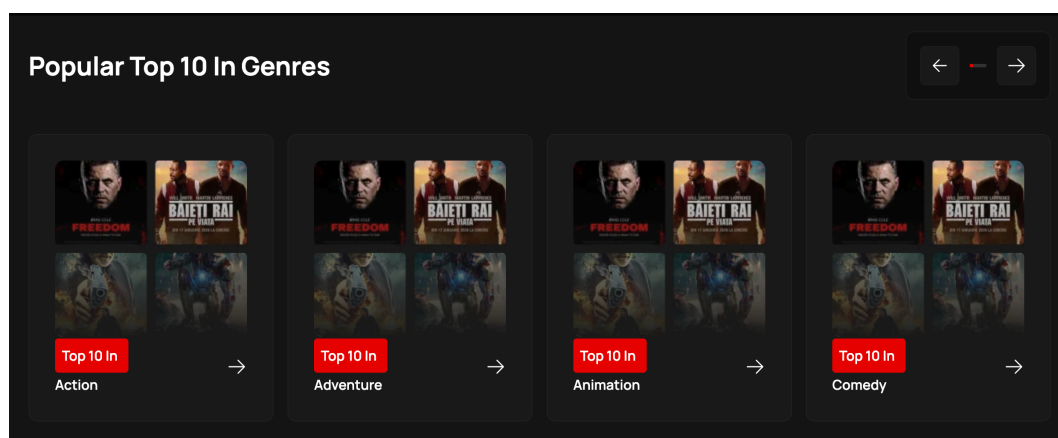


Рисунок 3.13 – Футер вебсайту

Джерело: знімок екрану створено автором

Реалізація каруселі з фільмами в тренді (рис. 3.14).

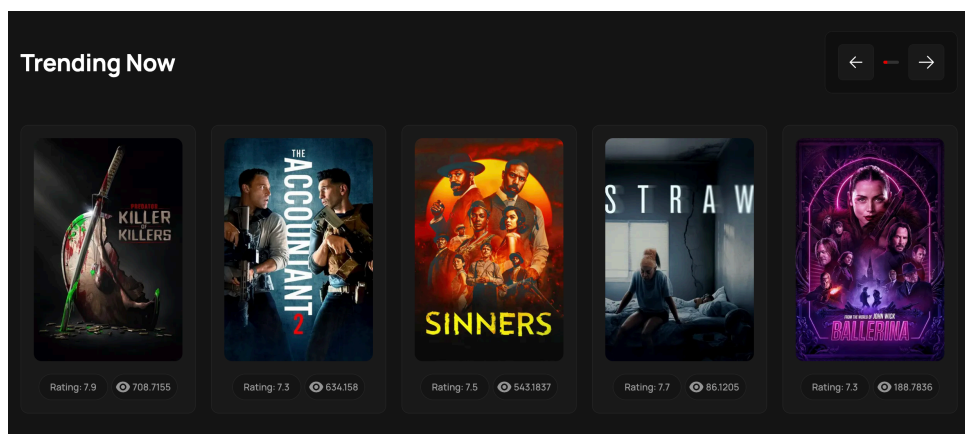


Рисунок 3.14 – Карусель з фільмами в тренді

Джерело: знімок екрану створено автором

Реалізація каруселі з новими фільмами (рис. 3.15).

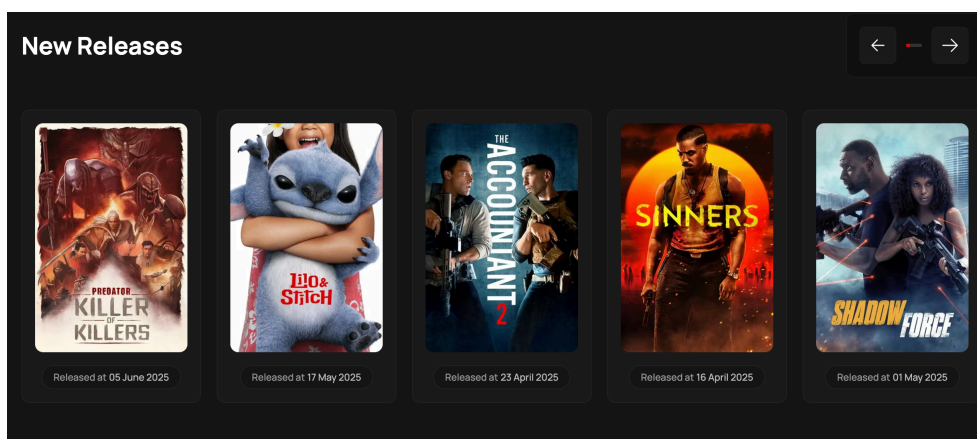


Рисунок 3.15 – Карусель з новими фільмами

Джерело: знімок екрану створено автором

Реалізація каруселі з фільмами обов'язковими до перегляду (рис. 3.16).

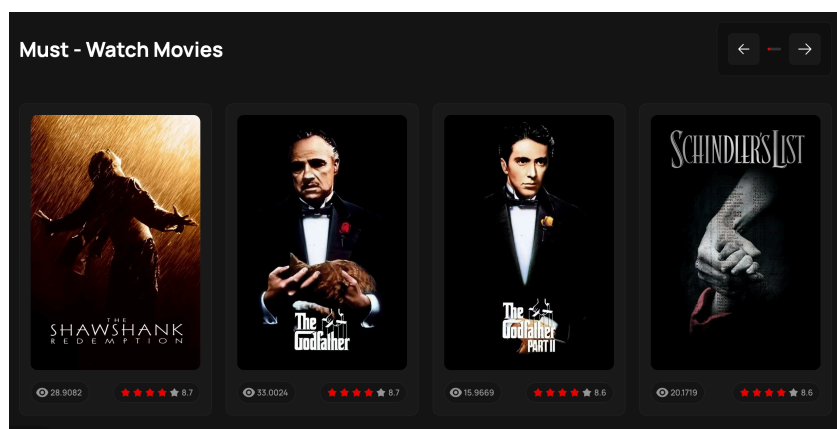


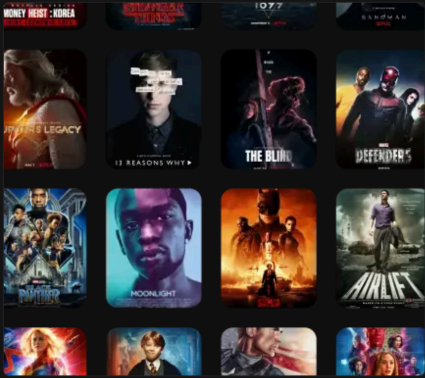
Рисунок 3.16 – Карусель з фільмами обов'язковими до перегляду

Джерело: знімок екрану створено автором

Реалізація каруселі форми тех. підтримки (рис. 3.17).

Welcome to our support page!

We're here to help you with any problems you may be having with our product.



First Name

Enter First Name

Last Name

Enter Last Name

Email

Enter your Email

Phone Number

Enter Phone Number

Message

Enter your Message

☐ I agree with Terms of Use and Privacy Policy

Send

Рисунок 3.17 – Форма звернення до тех.підтримки

Джерело: знімок екрану створено автором

Порівняльна таблиця різних планів (рис. 3.18).

Compare our plans and find the right one for you

StreamVibe offers three different plans to fit your needs: Basic, Standard, and Premium. Compare the features of each plan and choose the one that's right for you.

Features	Basic	Standard Popular	Premium
Price	\$9.99	\$12.99	\$14.99
Free Trail	7 Days	7 Days	7 Days
Content	Access to a wide selection of movies and shows, including some new releases.	Access to a wider selection of movies and shows, including most new releases and exclusive content	Access to a widest selection of movies and shows, including all new releases and Offline Viewing
Devices	Watch on one device simultaneously	Watch on Two device simultaneously	Watch on Four device simultaneously
Cancel Anytime	Yes	Yes	Yes
HDR	No	Yes	Yes
Dolby Atmos	No	Yes	Yes
Dolby Atmos	No	Yes	Yes

Рисунок 3.18 – Форма звернення до тех.підтримки

Джерело: знімок екрану створено автором

ВИСНОВКИ

У результаті виконання кваліфікаційного бакалаврського проєкту було реалізовано вебзастосунок для онлайн-перегляду фільмів та шоу, розроблений з використанням сучасного фреймворку Next.js та бази даних Firebase Realtime Database.

Проведений аналіз існуючих аналогів продемонстрував актуальність створення подібної системи, особливо з урахуванням можливості впровадження ролей користувача (клієнт, адміністратор), адаптивного інтерфейсу, та зручного управління контентом через вебінтерфейс. В ході розробки було визначено функціональні та нефункціональні вимоги до системи, створено моделі даних, діаграми взаємодії, спроектовано логіку аутентифікації, систему підтримки користувачів, та реалізовано рольову модель доступу.

У межах проєкту створено інформаційну структуру, яка забезпечує взаємозв'язок між основними об'єктами системи: користувачами, контентом (фільмами/шоу), а також повідомленнями до технічної підтримки. Користувачі мають змогу переглядати доступний контент, а адміністратори – здійснювати моніторинг звернень. Фронтенд частина була реалізована з використанням React, TypeScript, Tailwind CSS, а для керування формами застосовано React Hook Form із валідацією.

Інформаційна система підтримує масштабування функціоналу, передбачає розширення можливостей, таких як збереження історії перегляду або впровадження платних підписок.

Таким чином, розроблений програмний продукт є працездатним, адаптивним до різних пристроїв і готовим до використання в реальних умовах. У подальшому він може бути доповнений інтеграцією медіа-сховища, системою коментарів та впровадженням аналітики користувацької активності для покращення якості надання послуг кіноконтенту.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Megogo. URL: <https://megogo.net/ua>. (дата звернення: 01.06.2025)
2. Sweet.tv. URL: <https://sweet.tv/uk>. (дата звернення: 01.06.2025)
3. OmegaTV. URL: <https://omegatv.ua/>. (дата звернення: 01.06.2025)
4. Kyivstar TV. URL: <https://tv.kyivstar.ua/ua/>. (дата звернення: 01.06.2025)
5. Основи вимог до бізнесу. URL: <https://visuresolutions.com/uk/alm-guide/основи-вимог-до-бізнесу/> (дата звернення: 01.06.2025)
6. Функціональні вимоги. URL: https://uk.wikipedia.org/wiki/Функціональні_вимоги (дата звернення: 01.06.2025)
7. Нефункціональні вимоги. URL: https://uk.wikipedia.org/wiki/Нефункціональні_вимоги (дата звернення: 01.06.2025)
8. Діаграма прецедентів. URL: https://uk.wikipedia.org/wiki/Діаграма_прецедентів (дата звернення: 01.06.2025)
9. Сценарій використання. URL: https://uk.wikipedia.org/wiki/Сценарій_використання (дата звернення: 01.06.2025)
10. Zosym Махум. Діаграма послідовності (Sequence Diagrams) URL: <https://www.maxzosim.com/sequence-diagrams/> (дата звернення: 01.06.2025)
11. Діаграма діяльності в UML: символ, компоненти та приклад. URL: <https://www.guru99.com/uk/uml-activity-diagram.html> (дата звернення: 01.06.2025)
12. SysML. URL: <https://ru.wikipedia.org/wiki/SysML> (дата звернення: 01.06.2025)
13. Next.js. URL: <https://nextjs.org/> (дата звернення: 01.06.2025)
14. TailwindCSS. URL: <https://tailwindcss.com/> (дата звернення: 01.06.2025)
15. Firebase. URL: <https://tailwindcss.com/> (дата звернення: 01.06.2025)

ДОДАТКИ

Додаток А

Посилання на живу сторінку: <https://online-cinema-sepia.vercel.app/>

Звіт подібності



Дата звіту6/9/2025
Дата редагування---



Звіт не був оцінений

Звіт подібності

метадані

Назва організації
Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок
Розроблення інформаційної системи для кінотеатру з використанням технології NextJS

Автор
Науковий керівник / Експерт
Хиля Олександр ОлександровичЛозовик Ю.М.

підрозділ
кафедра інформаційних систем в економіці

Обсяг знайдених подібностей

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



Тривога

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

Заміна букв		0
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)	a	21

Подібності за списком джерел

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

10 найдовших фраз		Копію тексту	
ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	https://ir.kneu.edu.ua/bitstreams/22acc4f6-0461-4d18-a9a4-ff2fbd8c10ff/download	50	0.77 %
2	https://ir.kneu.edu.ua/bitstreams/22acc4f6-0461-4d18-a9a4-ff2fbd8c10ff/download	43	0.66 %
3	https://ir.kneu.edu.ua/bitstreams/850a2498-073e-45fe-874b-28be792ab58d/download	43	0.66 %
4	https://ir.kneu.edu.ua/bitstreams/22acc4f6-0461-4d18-a9a4-ff2fbd8c10ff/download	21	0.32 %
5	https://ir.kneu.edu.ua/bitstreams/850a2498-073e-45fe-874b-28be792ab58d/download	16	0.25 %

6	https://ir.kneu.edu.ua/bitstreams/850a2498-073e-45fe-874b-28be792ab58d/download	15 0.23 %
7	https://ir.kneu.edu.ua/bitstreams/850a2498-073e-45fe-874b-28be792ab58d/download	14 0.22 %
8	https://ir.kneu.edu.ua/bitstreams/850a2498-073e-45fe-874b-28be792ab58d/download	9 0.14 %
9	https://ir.kneu.edu.ua/bitstreams/850a2498-073e-45fe-874b-28be792ab58d/download	8 0.12 %
10	https://ir.kneu.edu.ua/server/api/core/bitstreams/1b35cb3a-b293-4677-9674-83983838c914/content	8 0.12 %

з бази даних RefBooks (0.00 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-----------	--

з домашньої бази даних (0.08 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Автоматизована система з обліку роботи автосалону 6/7/2019 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра інформаційних систем в економіці)	5 (1) 0.08 %

з програми обміну базами даних (0.00 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-----------	--

з Інтернету (3.89 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://ir.kneu.edu.ua/bitstreams/22acc4f6-0461-4d18-a9a4-ff2fbd8c10ff/download	128 (5) 1.97 %
2	https://ir.kneu.edu.ua/bitstreams/850a2498-073e-45fe-874b-28be792ab58d/download	105 (6) 1.62 %
3	https://ir.kneu.edu.ua/bitstreams/f8522cc1-4cb6-48ce-b298-126a95bd4768/download	11 (2) 0.17 %
4	https://ir.kneu.edu.ua/server/api/core/bitstreams/1b35cb3a-b293-4677-9674-83983838c914/content	8 (1) 0.12 %

Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-------	---------------------------------------