

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»
В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: «Web-додаток формування рейтингу відеоконтенту з відкритих джерел»

Здобувача (ки) групи ІТ-92 Литвиненка Антона Сергійовича
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

(підпис)

Антон ЛИТВИНЕНКО
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник старший викладач кафедри ІТ, к.т.н., Едуард КУЗНЄЦОВ _____
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ) (підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. зав. кафедри ІТ

_____ С. М. Ващенко

«__» _____ 2023р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Литвиненку Антону Сергійовичу

1 Тема роботи Web-додаток формування рейтингу відеоконтенту
з відкритих джерел

керівник роботи Кузнєцов Едуард Геннадійович, к.т.н., старший викладач,

затверджені наказом по університету від «29» 05 2023 р. №0588-VI

2 Строк подання студентом роботи «7» червня 2023 р.

3 Вхідні дані до роботи Дані про фільми

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) вступ, аналіз предметної області, постановка задачі та методи дослідження, практична реалізація web-додатка, висновки, список використаних джерел, технічне завдання, планування робіт

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) вступ, актуальність, постановка задачі, аналіз web-додатків підтримки, порівняння сайтів-аналогів, моделювання процесу формування рейтингу відеоконтенту з відкритих джерел, контекстна діаграма, діаграма декомпозиції, діаграма варіантів використання, практична реалізація проекту, демонстрація web-додатку (користувач, рецензент), висновки

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 8.02.2023

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	До 11.02.23	
2	Аналіз існуючих програмних продуктів для вирішення поставленої задачі	До 21.02.23	
3	Формування переліку вимог до сайту	До 08.03.23	
4	Вибір засобів реалізації	До 14.03.2023	
5	Моделювання програмного продукту в нотації IDEF0	До 22.03.2023	
6	Розробка моделі аналізу	До 01.04.2023	
7	Створення моделі проектування	До 04.04.2023	
8	Розробка моделі реалізації	До 11.04.2023	
9	Верстка сторінок	До 18.04.2023	
10	Робота з бек-енд	До 21.04.2023	
11	Розробка і впровадження бази даних	До 02.05.2023	
12	Beta-тестування	До 9.05.2023	
13	Alpha-тестування	До 18.05.2023	
14	Публікація на хостинг		
15	Написання програмної документації	До 03.06.2023	

Студент _____
(підпис)

Антон ЛИТВИНЕНКО

Керівник роботи _____
(підпис)

к.т.н., ст.викл., Едуард КУЗНЄЦОВ

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток формування рейтингу відеоконтенту з відкритих джерел».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, додатків. Загальний обсяг роботи – 53 сторінки, у тому числі 31 сторінка основного тексту, 2 сторінки списку використаних джерел, 20 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку формування рейтингу відеоконтенту з відкритих джерел.

У першому розділі проведений аналіз продуктів аналогів, які є в інтернет-доступі, також проведений аналіз останніх досліджень і публікацій з подальшою постановкою задачі.

У другому розділі проведено модулювання web-додатку за допомогою діаграми IDEF0 та її декомпозиції, діаграми варіантів використання, що допомогло з'ясувати функціонал web-додатку, його взаємодію з користувачем.

У третьому розділі проведена програмна реалізація web-додатка, визначена структура, показано, як працює додаток зі сторони анонімного чи авторизованого користувачів.

Результатом проведеної роботи є створений web-додаток формування рейтингу відеоконтенту з відкритих джерел. Практичне значення роботи полягає в тому, що можна оцінювати фільми за рейтингом, передивлятися та додавати коментарі до кожного фільма.

Ключові слова: фільми, відеоконтент, оцінки, рейтинг, коментарі.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень і публікацій.....	8
1.2 Аналіз програмних продуктів - аналогів	9
1.3 Постановка задачі.....	13
2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ	15
2.1 Структурно-функціональне моделювання	15
2.2 Моделювання варіантів використання web додатку	16
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ	18
3.1 Прототипування	18
3.2 Структура web-додатку	20
3.3 Програмна реалізація	25
3.4 Використання web-додатку зі сторони користувача	26
3.5 Використання web-додатку зі сторони рецензента	29
ВИСНОВКИ.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	32
ДОДАТОК А	34
ДОДАТОК Б ПЛАНУВАННЯ РОБІТ	41

ВСТУП

Велика кількість фільмів, доступних для перегляду на різних онлайн-платформах, призвела до значного збільшення кількості користувачів, які шукають рекомендації щодо вибору найбільш підходящого фільму на свій смак. Web-програми, які надають послуги оцінювання та рецензування на основі відкритих джерел інформації, таких як IMDb, Rotten Tomatoes і Metacritic, набули величезної популярності в останні роки. Однак якість таких рейтингових систем все ще є предметом дискусій через різні обмеження, такі як упереджена або неповна інформація, маніпулювання даними та недостатні механізми фільтрації.

Розробка web-додатку для рейтингування фільмів із відкритих джерел є актуальною проблемою, яка вимагає попереднього аналізу та вивчення існуючих рішень.

Початкові дані для розробки цієї теми можна отримати з різних онлайн-баз даних, які надають інформацію про фільми, включаючи рейтинги, огляди та метадані. Дослідження спрямоване на розробку web-додатку, який зможе агрегувати та фільтрувати ці дані, щоб надавати користувачам надійні та точні рейтинги та рецензії на фільми.

Метою даної роботи є розробка та реалізація web-додатку, який надасть користувачам можливість творити спільний рейтинг фільмів на основі даних з відкритих джерел інформації та власного враження від перегляду. Завдання, які необхідно вирішити для досягнення цієї мети, включають збір і попередню обробку даних про фільми, реалізацію алгоритму розрахунку рейтингів, розробку зручного інтерфейсу для навігації по масиву фільмів та надання супутньої інформації про фільми.

Практичне застосування отриманих результатів полягає в тому, щоб надати користувачам зручну та зрозумілу систему оцінювання, яка може

допомогти їм прийняти обґрунтовані рішення про те, які фільми дивитися. Крім того, web-додаток може служити цінним інструментом для агентів кінопрокату для вивчення вподобань аудиторії, тенденції в їхніх змінах.

Для досягнення поставленої задачі необхідно:

- Здійснити детальний аналіз предметної області з метою досягнення максимального розуміння всіх процесів.
- Порівняти існуючі програмні продукти, які реалізують подібну задачу, з розроблюваним web-додатком.
- Сформулювати мету, цілі.
- Промодельовати функціонал розробки, її взаємодію з користувачем.
- Обрати програмний інструментарій та середовище для розробки.
- Провести повну програмну реалізацію сформульованого завдання.
- Протестувати розроблений web-додаток.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Підходи на основі машинного навчання для рекомендацій фільмів: багато дослідників досліджують використання алгоритмів машинного навчання для систем рекомендацій фільмів. Ці системи аналізують поведінку та вподобання користувачів, щоб запропонувати фільми, які їм, ймовірно, сподобаються [1].

Етичні міркування в програмах для рейтингу фільмів: серед користувачів зростає занепокоєння щодо етичних наслідків збору та використання особистих даних у програмах рейтингування фільмів [2].

Контент, створений користувачами, і культурне розмаїття: багато дослідників вивчають як створений користувачами контент можна використовувати для сприяння культурному розмаїттю та репрезентації в кіноіндустрії. Наприклад, у дослідженні «Різноманітність та інклюзивність у кіно: аналіз створеного користувачами контенту на IMDb» [3] проаналізовано відгуки користувачів на IMDb, щоб виявити закономірності в представленні різних культурних груп у фільмах.

Відкриті джерела даних і спільне фільтрування: спільне фільтрування є популярним методом, який використовується в системах рекомендацій фільмів. Нещодавні дослідження досліджували використання для спільної фільтрації відкритих джерел даних, таких як платформи соціальних медіа та кінофоруми [4].

Іншою нещодавньою тенденцією в дослідженнях, пов'язаних із web-додатками для оцінювання фільмів, є використання методів аналізу настроїв для розуміння відгуків і оцінок користувачів. Алгоритми аналізу настроїв можуть ідентифікувати та класифікувати емоції, виражені у відгуках користувачів, надаючи цінну інформацію про те, як користувачі ставляться до певних фільмів. Дослідники використовують цю інформацію, щоб розробити

точніші рейтинги фільмів і рекомендації, а також покращити роботу web-додатків для оцінювання фільмів [5].

Загалом, дослідження та публікації, пов'язані з web-додатками для оцінювання фільмів із відкритих джерел, є активною та зростаючою областю досліджень. Із зростанням популярності транслюючих служб і великою кількістю даних про фільми, доступних в Інтернеті, існує значна потреба в точних і персоналізованих рекомендаціях щодо фільмів, і дослідники плідно працюють над розробкою нових алгоритмів і моделей для задоволення цієї потреби.

1.2 Аналіз програмних продуктів - аналогів

IMDb: IMDb є одним із найпопулярніших web-додатків для оцінювання фільмів. Це повна база даних фільмів, телешоу та відеоігор. IMDb дозволяє користувачам оцінювати фільми за шкалою від 1 до 10, а також надає огляди та оцінки від професійних критиків (рис. 1.1).

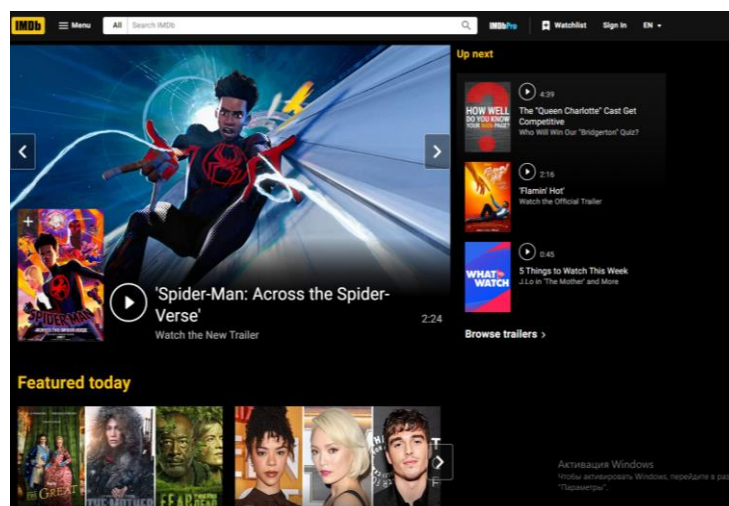


Рисунок 1.1 – Головна сторінка IMDb

Rotten Tomatoes: Rotten Tomatoes — ще один популярний web-додаток для рейтингів фільмів. Він об'єднує відгуки професійних критиків і обчислює

оцінку на основі відсотка позитивних відгуків. Користувачі також можуть оцінювати фільми та залишати власні відгуки (рис. 1.2).

Metacritic: Metacritic — це web-програма, яка збирає відгуки професійних критиків і обчислює оцінку на основі середньозваженого значення. Користувачі також можуть оцінювати фільми та залишати власні відгуки (рис. 1.3).

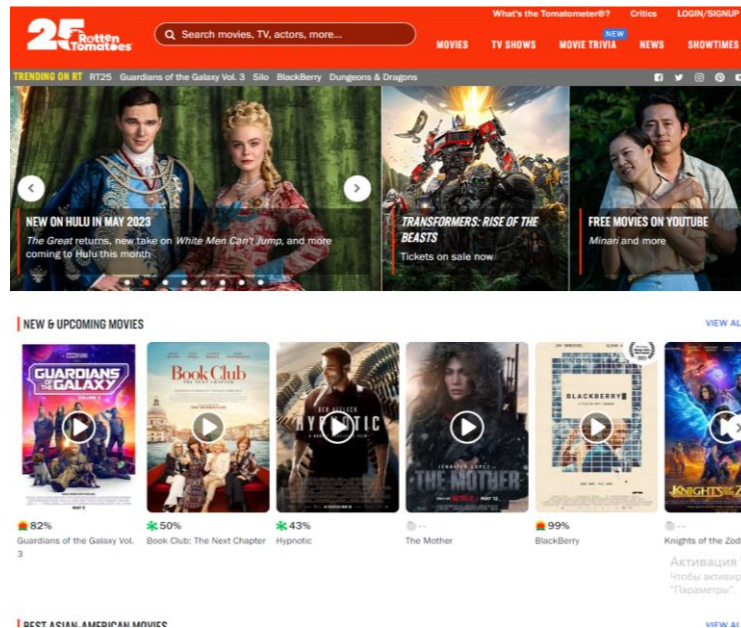


Рисунок 1.2 – Головна сторінка Rotten Tomatoes

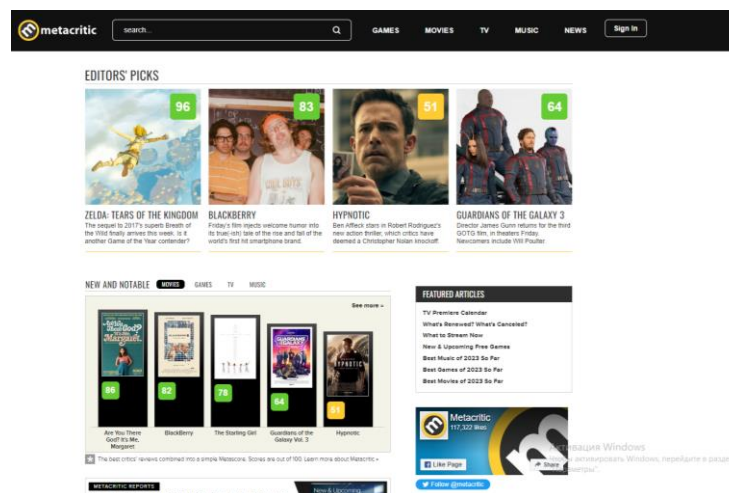


Рисунок 1.3 – Головна сторінка Metacritic

Letterboxd: Letterboxd — це служба соціальних мереж для кінолюбителів. Користувачі можуть оцінювати та переглядати фільми, створювати списки та

стежити за іншими користувачами. Letterboxd також надає персоналізовані рекомендації на основі оцінок і відгуків користувачів (рис. 1.4).

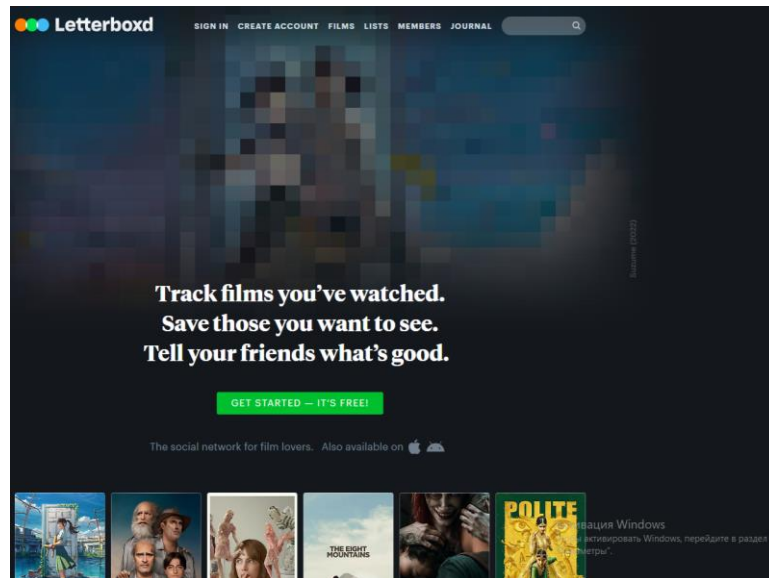


Рисунок 1.4 – Головна сторінка Letterboxd

TMDb: База даних фільмів (TMDb) — це web-додаток, керований спільнотою, для рейтингів фільмів та інформації. Користувачі можуть оцінювати фільми, створювати списки перегляду та робити внески в базу даних, додаючи інформацію та відгуки (рис. 1.5).

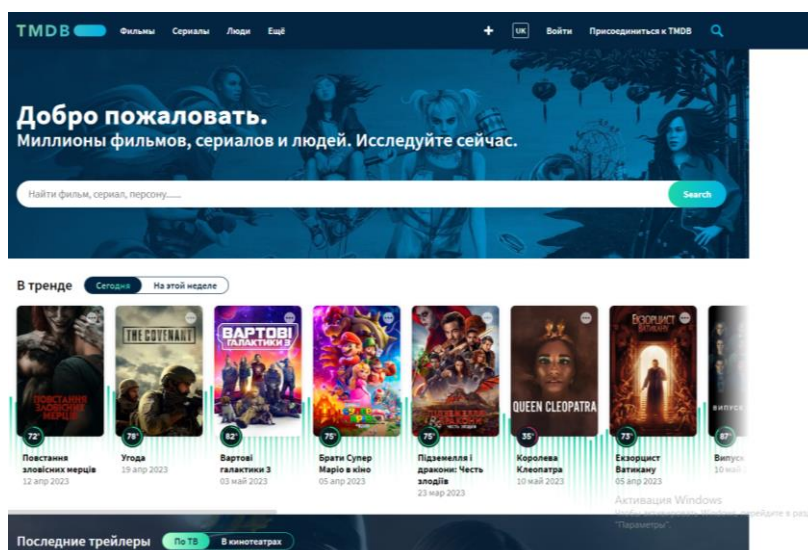


Рисунок 1.5 – Головна сторінка TMDb

Ці web-додатки дозволяють користувачам відкривати нові фільми, читати рецензії та ділитися своїми думками з іншими. Усі вони базуються на відкритих джерелах, що означає, що дані є у вільному доступі для використання та доповнення.

У результаті проведеного аналізу аналогів додатків можна зробити підсумки, які представлено в таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця характеристик аналогів web-додатків

Web-додаток	Система оцінювання	Система оглядів	Джерело оцінок	Соціальні мережі
IMDb	Шкала 1-10	Так	Користувачі та професійні критики	Ні
Rotten Tomatoes	Відсоткова оцінка	Так	Професійні критики	Ні
Metacritic	Взважена середня	Так	Професійні критики	Ні
Letterboxd	Шкала 1-5	Так	Користувачі	Так
TMDb	Шкала 1-10	Так	Користувачі та професійні критики	Ні

IMDb, Rotten Tomatoes та Metacritic базуються на оцінках професійних критиків, з додатковою можливістю оцінки від користувачів в IMDb. Rotten Tomatoes використовує унікальну систему оцінювання, що розраховує відсоток позитивних відзивів для фільму. Metacritic використовує систему зваженої середньої оцінки, яка враховує оцінки кількох критиків.

Letterboxd зосереджений на спільноті користувачів, з оцінками та оглядами переважно від користувачів. Крім того, він має можливості соціальних мереж, які дозволяють користувачам слідкувати за іншими

користувачами та створювати списки фільмів, які вони переглянули або хотіли б переглянути.

TMDb також заснований на оцінках користувачів, але включає також оцінки професійних критиків. Він фокусується на наданні повної інформації про фільми та телешоу, таких як актори та знімальна група, сюжетні описи та дати випуску.

Загалом, кожен із цих web-додатків має свої переваги та особливості. Вибір залежить від особистих уподобань користувача та потреб у пошуку та оцінці фільмів.

Аналізуючи дані з таблиці 1.1 сформульовано функціональні вимоги до розроблюваного web-додатку, які представлені в додатку А.

1.3 Постановка задачі

Метою даного дослідження є розробка web-додатка формування рейтингу відеоконтенту з відкритих джерел. Додаток може бути дооснащений можливістю зберігати фільми у власному списку вподобань, залишати коментарі для інших користувачів.

Для досягнення поставленої мети треба вирішити наступні задачі:

1. Проведення аналізу предметної області, дослідження існуючих інтернет-сервісів рейтингування кінопродукції, формулювання вимог до розроблюваного web-додатку;
2. Вибір програмних засобів та середовища розробки;
3. Моделювання роботи web-додатку;
4. Розроблення додатку, підключення API сторонніх серверів;
5. Тестування web-додатку.

Для реалізації даного web-додатку було обрано такі технології, як мову розмітки гіпертексту HTML для створення каркасу web-сторінок, каскадні таблиці стилів CSS для надання web-сторінкам візуальних ефектів та

адаптивності, скриптову мову програмування JavaScript з бібліотекою ReactJS для надання динамічності та асинхронної взаємодії з серверною частиною.

2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ

2.1 Структурно-функціональне моделювання

Для попереднього моделювання роботи була створена структурна та функціональна модель робочого процесу web-додатку за допомогою нотації IDEF0. Ця модель включає всі дані та вхідну інформацію, які використовуються web-додатком для створення рейтингу відеоконтенту з відкритих джерел. На рисунку 2.1 зображена діаграма нульового рівня, яка ілюструє цей процес.

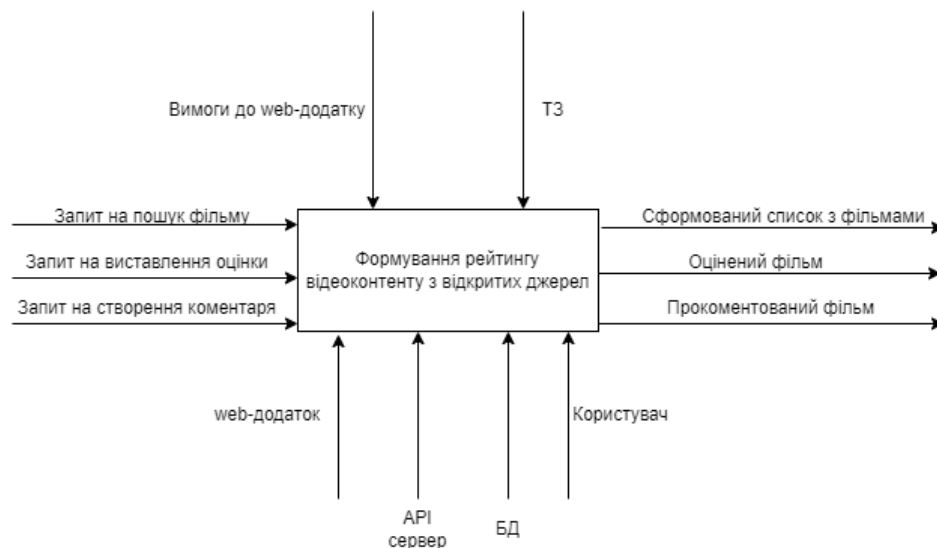


Рисунок 2.1 – Контекстна діаграма IDEF0

На рисунку 2.1 зображено, як до web-додатку підходять 3 стрілочки: запит на пошук фільму та запити на виставлення оцінки та створення коментаря. А на виході ми бачимо кінцевий результат, який виконує наш додаток – сформований список з фільмами, оцінений фільм, та прокоментований.

На рисунку 2.2 зображено варіант діаграми декомпозиції IDEF0, яка включає детальну розбивку кожної фази проекту. Ця діаграма ілюструє послідовність кроків або процесів, що виконуються в рамках проекту. Вона

сприяє кращому розумінню структури та взаємозв'язків між різними елементами проекту, а також докладно відображає кожну фазу.

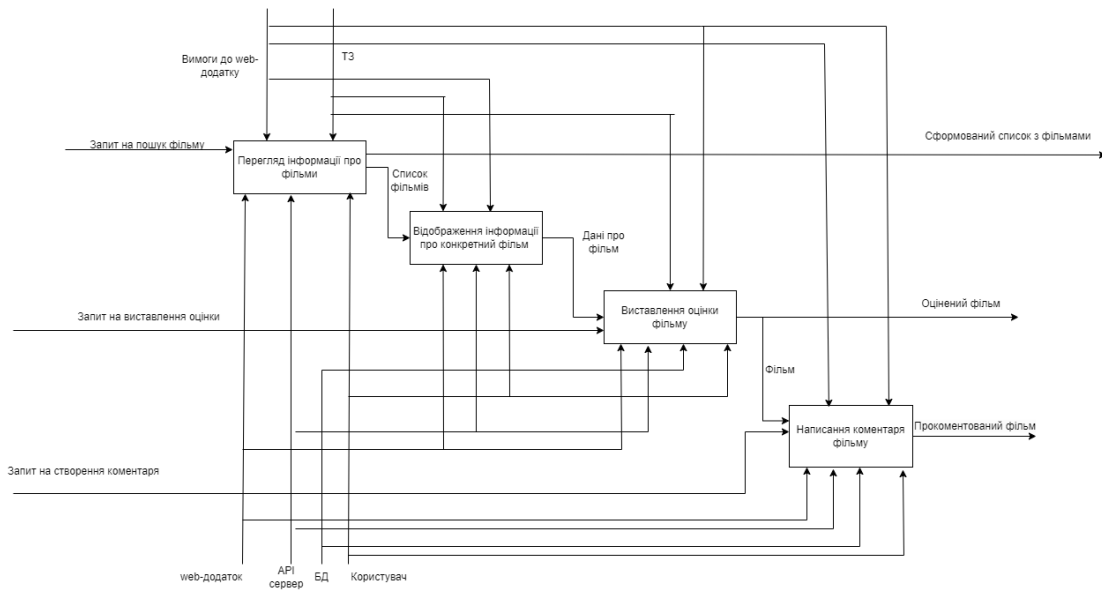


Рисунок 2.2 – Діаграма декомпозиції IDEF0

2.2 Моделювання варіантів використання web додатку

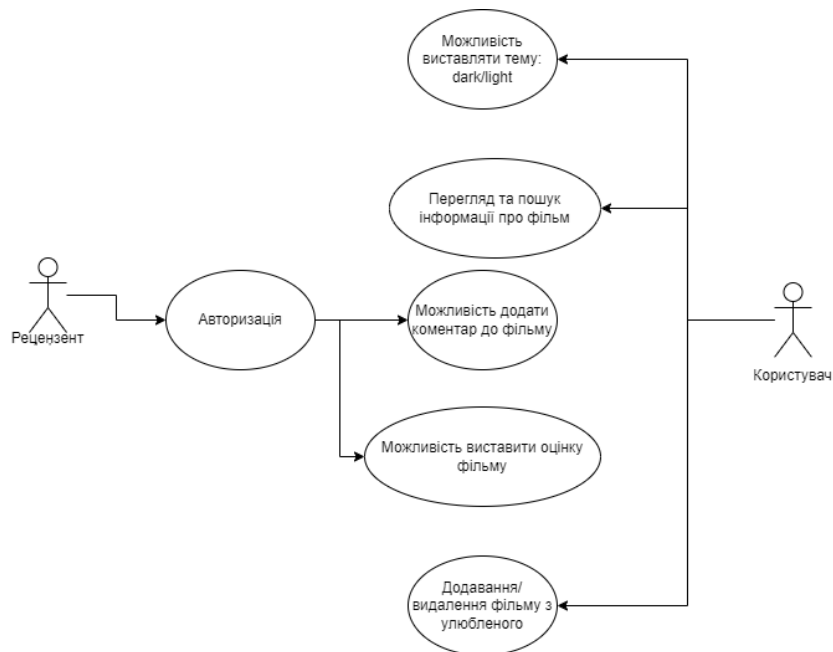


Рисунок 2.3 – Варіанти використання web-додатку

У таблицях 2.1-2.2 наведено опис акторів та варіантів використання

Табл. 2.1 – Актори-користувачі

Назва	Опис
Користувач	Анонімний користувач web-додатку
Рецензент	Авторизований користувач web-додатку

Табл. 2.2 – Опис варіантів використання

Назва	Опис
Перегляд та пошук інформації про фільм	Можливість знайти та переглянути детальну інформацію про кожен фільм
Можливість поставити оцінку фільму	Виставляння оцінки рейтингу фільму
Можливість додати коментар до фільму	Додавання коментарів для їх перегляду іншими користувачами
Додавання та видалення з улюбленого	Кожен фільм можна зберегти в списку вподобань, потім можна видалити зі списку
Можливість змінити тему	Light/dark

На діаграмі можна помітити, що двома суб'єктами у варіантах використання web-додатку підтримки є користувач та рецензент, кожен може здійснити обмежену кількість дій.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Прототипування

Для розробки web-додатка було використано набір з чотирьох основних компонентів: HTML, CSS, JavaScript, React. На рисунку 3.1 показано функціонал та зв'язок між цими файлами та актором. Крім того, при взаємодії актора з інтерфейсом, інформація оновлюється і знову виводиться за допомогою React. На рисунку 3.1 представлено приклад прототипування, який був використаний для web-додатку з метою скорочення часу розробки та поліпшення результату.

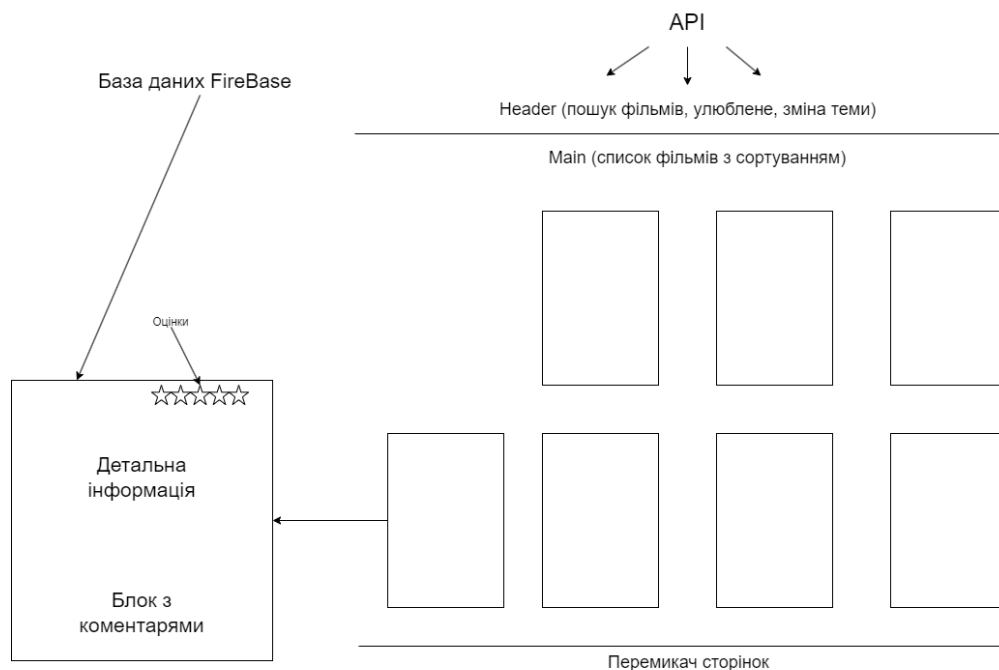


Рисунок 3.1 – Варіант розробки макету у паперову вигляді

Створення макету прототипу (наприклад, такого паперового) для додатку з відкритим кодом може бути корисним з таких причин:

– Низька вартість і швидка ітерація: паперові прототипи доступні, їх легко створити. Ви можете швидко накреслити екрани, потоки користувачів та інтерактивні елементи, що дозволить швидко повторювати та вдосконалювати свої концепції дизайну. Ця гнучкість може допомогти тестувати різні ідеї та збирати відгуки на ранніх стадіях процесу розробки.

– Зосередження на взаємодії з користувачем. Створивши паперовий прототип, можна визначити пріоритети взаємодії з користувачем, не відволікаючись на візуальну естетику чи технічну реалізацію. Це дозволяє зосередитися на основній функціональності та зручності використання сайту, гарантуючи, що користувачі зможуть орієнтуватися, оцінювати фільми та ефективно взаємодіяти з прототипом.

– Співпраця та спілкування: паперові прототипи сприяють співпраці між членами команди та зацікавленими сторонами. Оскільки вони відчутні та доступні для спільного використання, вони дозволяють обговорювати та мозкові штурми, гарантуючи, що кожен учасник має чітке розуміння особливостей і функціональності сайту.

– Рання перевірка концепцій: перш ніж інвестувати значний час і ресурси в розробку, паперовий прототип дозволить вам перевірити свої ідеї та припущення. Спостерігаючи за взаємодією користувачів і збираючи відгуки, ви можете визначити будь-які потенційні недоліки або втрачені можливості на ранній стадії, заощадивши час і зусилля в довгостроковій перспективі.

– Доступність та інклюзивність. Паперові прототипи інклюзивні за своєю природою, оскільки не вимагають жодних спеціальних технологій чи навичок цифрової грамотності. Ними можна легко ділитися та тестувати різноманітну групу користувачів, гарантуючи, що дизайн враховує широкий спектр точок зору та можливостей.

Однак важливо зазначити, що паперовий прототип — це симуляція. Це цінний інструмент для дослідження та перевірки дизайну на ранній стадії, але потребує цифрового впровадження для повноцінної інтерактивної та функціональної онлайн-платформи.

3.2 Структура web-додатку

Структура web-додатку формування рейтингу фільмів з відкритих джерел реалізована наступним чином:

– Навігація між сторінками. Тут використовується BrowserRouter, який при потраплянні на певну сторінку рендерять окремий js код (компонент)

```
{isModalAuth && <LogikaAuth />}
<Routes>
  <Route
    path="/"
    element={<Layout isKorzinaClicked={isKorzinaAnia} />}
  >
    <Route
      index
      element={
        <MovieContainer
          addedItemsLS={addedItemsLS}
          korzinaHandler={korzinaHandler}
        />
      }
    />
    <Route
      path="page/:id"
      element={
        <MovieContainer
          addedItemsLS={addedItemsLS}
          korzinaHandler={korzinaHandler}
        />
      }
    />
    <Route
      path="movie-details/:title"
      element={
        <AuthContextMovieDetailsProvider
          addedItemsLS={addedItemsLS}
          korzinaHandler={korzinaHandler}
        />
      }
    />
    <Route
      path="/:kategory/page/:id"
      element={
        <MovieContainer
          addedItemsLS={addedItemsLS}
          korzinaHandler={korzinaHandler}
        />
      }
    />
    <Route
      path="my-movies"
      element={
        <MyMovies
          korzinaHandler={korzinaHandler}
          addedItemsLS={addedItemsLS}
        />
      }
    />
    <Route
      path="search/:title"
      element=
    >
      <SearchPage
    </>
  </Route>
</Routes>
```

Рисунок 3.2 – Код навігаційного меню

– Головна сторінка: тут можна буде побачити список фільмів з певною інформацією про них. З контексту (AuthContext) реакту беруться дані з відфільтрованим списком фільмів згідно запиту до серверу API TMDb та виводиться все на екран

```

import { MovieContainer, scss } from './MovieContainer.scss'
import MovieItem from './MovieItem'
import { useContext, useEffect, useState } from 'react'
import { useContext, useEffect, useState } from 'react'
import AuthContext from '../AuthContext'
import OverViewMovies from './OverViewMovies'
import Categories from './Categories'
import AuthContextPagination from '../AuthContextPagination'
import { useLocation } from 'react-router-dom'

const MovieContainer = (props) => {
  const context = useContext(AuthContext)
  const contextPagination = useContext(AuthContextPagination)
  const location = useLocation()
  const [loader, setLoader] = useState(true)

  useEffect(() => {
    setTimeout(() => {
      setLoader(false)
    }, 900)
  }, [])

  const katalog = context.katalog.map(movie => (
    <MovieItem
      darkTema={context.darkTema}
      key={movie.id}
      id={movie.id}
      poster_path={movie.poster_path}
      backdrop_path={movie.backdrop_path}
      title={movie.title}
      overview={movie.overview}
      vote_average={movie.vote_average}
      genre_ids={movie.genre_ids}
      korzinaHandler={props.korzinaHandler}
      isAdd={
        props.addedItemsLS.find(item => item.id === movie.id) ? true
      }
    />
  ))

  return (
    <div
      {context.isLoading ? loader : <Categories />}
      <OverViewMovies
        items={katalog}
        page={
          location.pathname.includes('page')
            ? 1
            : parseInt(context.location.pathnameArr)
        }
        totalPages={context.totalPages}
        changePage={change, pageId =>
          contextPagination.changePageHandler(change, pageId, 'main')}
        choosePage={id =>
          contextPagination.choosePageHandler(id, 'main')}
      />
    />
  )
}

```

Рисунок 3.3 – Код головної сторінки

– Сторінка зі списком вподобань «Улюблене»: можна додавати та видаляти фільм зі списку. Фільми з улюбленим зберігаються в LocalStorage браузера і тут беруться всі фільми звідти і відрисовуються методом масива map компоненти MovieItem

```

import { useContext } from 'react'
import AuthContext from '../AuthContext'
import MovieItem from './Movies/MovieItem'

const MyMovies = (props) => {
  const context = useContext(AuthContext)
  const addedItemsKatalog = props.addedItemsLS.map(movie => (
    <MovieItem
      darkTema={context.darkTema}
      key={movie.id}
      id={movie.id}
      poster_path={movie.poster_path}
      backdrop_path={movie.backdrop_path}
      title={movie.title}
      overview={movie.overview}
      vote_average={movie.vote_average}
      genre_ids={movie.genre_ids}
      korzinaHandler={props.korzinaHandler}
      isAdd={movie.isAdd}
    />
  ))

  return (
    <div className={movie-container${context.darkTema ? ' dark' : ''}}
      <div className="container">
        <JSON.parse(localStorage.getItem("addedItems")).length !== 0 ? (
          addedItemsKatalog
        ) : (
          <div className="movie_null">
            поки що у вас немає улюбленого
          </div>
        )
      </div>
    </div>
  )
}

export default MyMovies

```

Рисунок 3.4 – Код сторінки зі списком вподобань

– Сторінка з конкретним фільмом: рецензент після авторизації може поставити оцінку певному фільму та прокоментувати його, а також з

користувачем може продивитись детальну інформацію про фільм. Робиться RestApi запит до API TMDb і отримується детальна інформація про окремий фільм, а також робюється запит до бази даних FireBase, звідки отримуються дані про оцінки до цього фільму та коментарі.

```

import React from "react";
import { useState, useEffect, useContext } from "react";
import { useRouter } from "react-router-dom";
import AuthContext from "../AuthContext";
import MovieDetails from "../components/MovieDetails";
import { serverZapros } from "../components/ServerZapros";

const AuthContextMovieDetails = React.createContext({});

export const AuthContextMovieDetailsProvider = (props) => {
  const [movieDetails, setMovieDetails] = useState("");
  const [location, setLocation] = useState("");
  const [isLoading, setIsLoading] = useState(false);
  const [searching, setSearch] = useState("");
  const [recommendations, setRecommendations] = useState("");
  const [movie, setMovie] = useState("");
  const context = useContext(AuthContext);
  const id = location.pathname.split("/").pop().split("-")[0];
  const [isLoading, setIsLoading] = useState(false);
  const [movie, setMovie] = useState("");
  const [movieDetails, setMovieDetails] = useState("");

  useEffect(() => {
    setMovieDetails(serverZapros.getMovie(id) ? true : false);
    }, [location.pathname]);

    useEffect(() => {
      const API_KEY = "a6737266d7a2768667683212d7";
      const id = location.pathname.split("/").pop().split("-")[0];
      const setIsLoading = true;
      document.body.style.overflow = "hidden";
      setIsLoading(() => {
        window.scrollTo(0, 0);
        document.body.style.overflow = "auto";
      }, 500);

      fetch(
        `https://api.themoviedb.org/3/movie/${id}?api_key=${API_KEY}&language=en-US`
      )
      .then((response) => response.json())
      .then((data) => {
        fetch(
          `https://www.googleapis.com/default-rdb.firebaseio.com/movie/${id}.json`
        )
        .then((response) => response.json())
        .then((data) => {
          const vote_average = [];
          for (let rate of Object.values(data)) {
            if (rate.id === movie.id) {
              vote_average.push(rate.rate);
            }
          }
        });
      });
    });
  }
}

```

Рисунок 3.5 – Код сторінки з конкретним фільмом

– Сторінка пошуку: всі користувачі web-додатку можуть по назві знайти фільм. Те, що ввів користувач зберігається в state та передається query параметром при відправленні GET запиту до API TMDb. Якщо запит нічого не віддав, то користувачу виводиться повідомлення, що нічого не знайдено по тому, що він ввів в пошуку

```

import React from "react";
import { useState, useEffect, useContext } from "react";
import { serverZapros } from "../components/ServerZapros";
import AuthContext from "../AuthContext";
import { useRouter } from "react-router-dom";

const AuthContextSearchPage = React.createContext({});

export const AuthContextSearchPageProvider = (props) => {
  const [location, setLocation] = useState("");
  const [locationPathname, setLocationPathname] = useState("");
  const [isLoading, setIsLoading] = useState(false);
  const [searching, setSearch] = useState("");
  const [movie, setMovie] = useState("");
  const [movieDetails, setMovieDetails] = useState("");
  const [movieDetails, setMovieDetails] = useState("");

  useEffect(() => {
    if (location.pathname.includes("search")) {
      const path = location.pathname.split("/")[2];
      const searchQuery = encodeURIComponent(path);
      const API_KEY = "a6737266d7a2768667683212d7";
      const locationPathname = location.pathname.split("/");
      const locationPathname = locationPathname[2];

      document.body.style.overflow = "hidden";
      document.body.style.overflow = "auto";
      window.scrollTo(0, 0);
    }
  }, [location]);

  serverZapros(
    `https://api.themoviedb.org/3/search/movie?api_key=${API_KEY}&language=en-US&query=${searchQuery}&page=1`
  )
  .then((response) => {
    fetch(
      `https://www.googleapis.com/default-rdb.firebaseio.com/movie/${id}.json`
    )
    .then((response) => response.json())
    .then((data) => {
      for (let movie of Object.values(data)) {
        if (movie.id === movie.id) {
          movie.vote_average.push(rate.rate);
        }
      }
    });
  });
}

```

Рисунок 3.6 – Код сторінки пошуку

```

1 import {OberthkMovies} from "../OberthkMovies"
2 import { useContext } from "react"
3 import MovieItem from "../Movies/MovieItem"
4 import AuthContextSearchPage from "../../AuthContextSearchPage"
5 import AuthContextPagination from "../../AuthContextPagination"
6 import AuthContext from "../../AuthContext"
7 import { useLocation } from "react-router-dom"
8
9 const SearchPage = (props) => {
10   const location = useLocation()
11   const context = useContext(AuthContext)
12   const contextSearchPage = useContext(AuthContextSearchPage)
13   const contextPagination = useContext(AuthContextPagination)
14
15   const katalogSearchPage = contextSearchPage.katalogSearchPage.map(
16     (movie) => {
17       <MovieItem
18         darkTema={context.darkTema}
19         key={movie.id}
20         id={movie.id}
21         poster_path={movie.poster_path}
22         backdrop_path={movie.backdrop_path}
23         title={movie.title}
24         overview={movie.overview}
25         vote_average={movie.vote_average}
26         genre_id={movie.genre_ids}
27         korzinahandler={props.korzinahandler}
28         isLoading={
29           props.addedItems5.find(item => item.id === movie.id) &&
30             true
31         }
32       />
33     }
34   )
35   return (
36     <OberthkMovies
37       items={katalogSearchPage}
38       page={
39         location.pathname.includes("page")
40           ? 1
41           : parseInt(contextSearchPage.locationPathnameArr)
42       }
43       totalPages={contextSearchPage.totalPageSearchPage}
44       changePage={changePageId} =>
45         contextPagination.changePageHandler(changePageId, "search")
46     )
47     <choosePage={id} =>
48       contextPagination.choosePageHandler(id, "search")
49     />
50   )
51 }
52
53 export default SearchPage

```

Рисунок 3.7 – Ключовий код сторінки пошуку

– Категорії: допомагають перемикається між сторінками по категоріям (популярні, зараз дивляться та очікуються). Категорії також зберігаються в state реакту, а також в localStorage, щоб при відновленні сторінки категорія залишилась та сама. В кодї нижче відрисовуються категорії зі своїми атрибутами

```

import "../kategorie.scss"
import AuthContext from "../AuthContext"
import { useContext } from "react"
import { useLocation } from "react-router-dom"

const Kategorie = () => {
  const location = useLocation()
  const context = useContext(AuthContext)
  const category = location.pathname.split("/") [1]
  return (
    <div className={kategorie${context.darkTema ? " dark" : ""}} >
      <input
        className="kategorie_input"
        type="radio"
        name="kat"
        id="popular"
        onChange={event} =>
          context.changeCategoryHandler(event.target.id)
      />
      <label className="kategorie_label" htmlFor="popular">
        <span>popular</span>
      </label>
      <input
        className="kategorie_input"
        type="radio"
        name="kat"
        id="now_playing"
        onChange={event} =>
          context.changeCategoryHandler(event.target.id)
      />
      <label className="kategorie_label" htmlFor="now_playing">
        <span>now_playing</span>
      </label>
      <input
        className="kategorie_input"
        type="radio"
        name="kat"
        id="upcoming"
        onChange={event} =>
          context.changeCategoryHandler(event.target.id)
      />
      <label className="kategorie_label" htmlFor="upcoming">
        <span>upcoming</span>
      </label>
    </div>
  )
}

```

Рисунок 3.8 – Код сторінки категорій

– Компонент для виставлення оцінки: рецензент може поставити оцінку фільму від 1 до 10. Коли користувач ставить оцінку, то відправляєється HTTP запит POST до FireBase, та зберігається там оцінка відносно цього фільму. Потім виконується відновлення стану.

```
import { useDispatch } from "react-redux"
import { setWholeAuth } from "../reducers/userReducer"
import { Comment } from "react"

const Rate = (props) => {
  const context = useContext(AutContextProviderDetails)
  const { isAuthenticated } = useContext(context)
  const dispatch = useDispatch()

  const rateFilmHandler = () => {
    if (!isAuthenticated || !isAuthorized) {
      localStorage.setItem("@@@@.id", "@@@" + rate)
    }

    fetch(
      "https://www.firebaseio.com/rateFilm.json",
      {
        method: "POST",
        body: JSON.stringify({
          id: "@@@" + rate,
          rate: "@@@" + rate,
        })
      }
    )

    setTimeout(() => {
      context.setIsAuthenticated(true)
    }, 200)

    if (context.isAuthenticated && !isAuthorized) {
      dispatch(setWholeAuth(true))
    }
  }

  const onWholeAuthHandler = (event) => {
    const all = event.target.parentElement.children
    let className = [...all].findIndex((item) => item === event.target)

    for (let i = 0; i <= className; i++) {
      all[i].classList.remove("active")
    }
  }

  const onWholeAuthHandler = (event) => {
    const all = event.target.parentElement.children
    for (let element of all) {
      element.classList.remove("active")
    }
  }

  return (
    <div>
      <button onClick={rateFilmHandler}>Rate</button>
      <button onClick={onWholeAuthHandler}>WholeAuth</button>
      <button onClick={onWholeAuthHandler}>WholeAuth</button>
      <input type="text" value={rate} />
      <input type="button" value={rate} />
    </div>
  )
}

export default Rate
```

Рисунок 3.9 – Основна частина коду компоненту для виставлення оцінки

– Компонент для додавання коментаря: рецензент може додати коментар до будь якого фільму. Тут так як і з оцінками, коментарі зберігаються в базі даних, котрі потім виводяться та зберігаються в state

```
import { useDispatch } from "react-redux"
import { useState } from "react"
import { Comment } from "react"
import { useEffect } from "react"
import { setWholeAuth } from "../reducers/userReducer"
import Comment from "react-router-dom"
import { useLocation } from "react-router-dom"

const Comments = (props) => {
  const { isAuthenticated, email } = useContext(context)
  const dispatch = useDispatch()

  const [comments, setComments] = useState([])
  const [comment, setComment] = useState("")
  const [isCorrect, setIsCorrect] = useState(true)
  const [show, setShow] = useState(true)

  useEffect(() => {
    fetch(
      "https://www.firebaseio.com/comments.json",
      {
        method: "GET",
      }
    )
      .then((response) => response.json())
      .then((data) => {
        setComments(data)
        dispatch(values(data).filter((item) => item.id === props.id))
      })
  }, [show, getHome])

  const commentFormHandler = () => {
    if (!isAuthenticated || !isAuthorized || !comments.length > 0) {
      fetch(
        "https://www.firebaseio.com/comments.json",
        {
          method: "POST",
          body: JSON.stringify({
            id: "@@@" + comment,
            comment: "@@@" + comment,
            time: new Date().toLocaleDateString(),
          })
        }
      )
      .then(() => {
        setComments([...comments, { id: "@@@" + comment, comment: "@@@" + comment, time: new Date().toLocaleDateString() }])
        setShow(true)
      })
    } else if (!isAuthenticated) {
      dispatch(setWholeAuth(true))
    } else if (!isAuthorized && !isAuthenticated) {
      dispatch(setWholeAuth(true))
    }
  }

  return (
    <div className="form-comment">
      <input type="text" value={comment} />
      <button onClick={commentFormHandler}>Add Comment</button>
    </div>
  )
}

export default Comments
```

Рисунок 3.10 – Головна частина коду компоненту для додавання коментарів

3.3 Програмна реалізація

Для розробки web-додатку було використано середовище розробки Visual Studio Code та для хостингу сайту було використано сервіс Vercel. Всі файли були збережені на github.

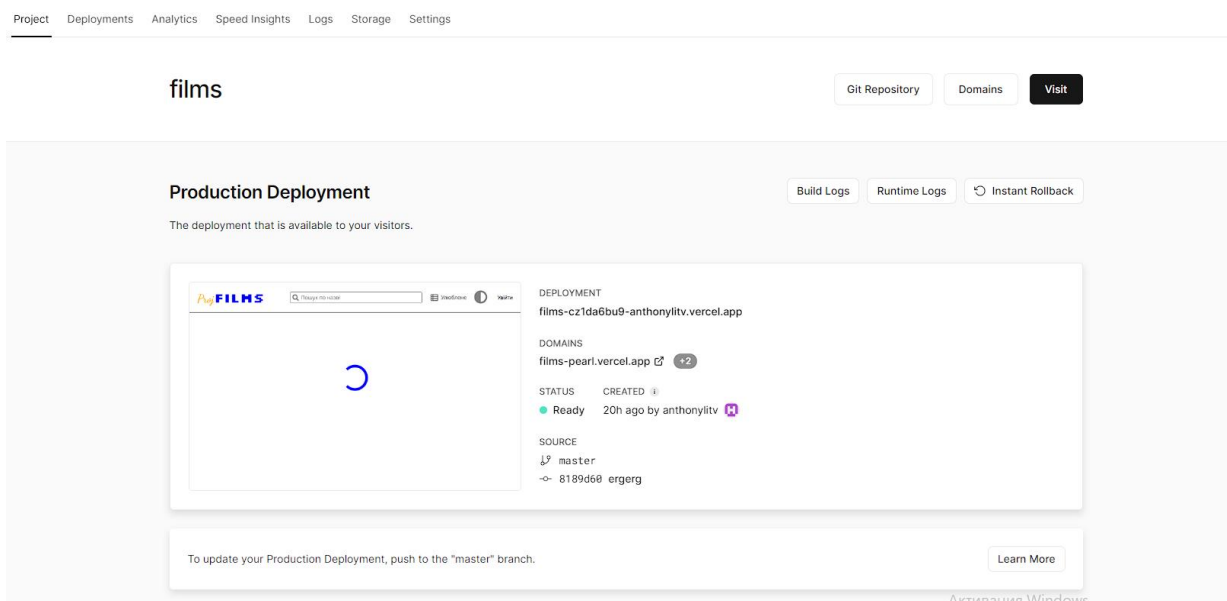


Рисунок 3.11 – Web-додаток на Vercel

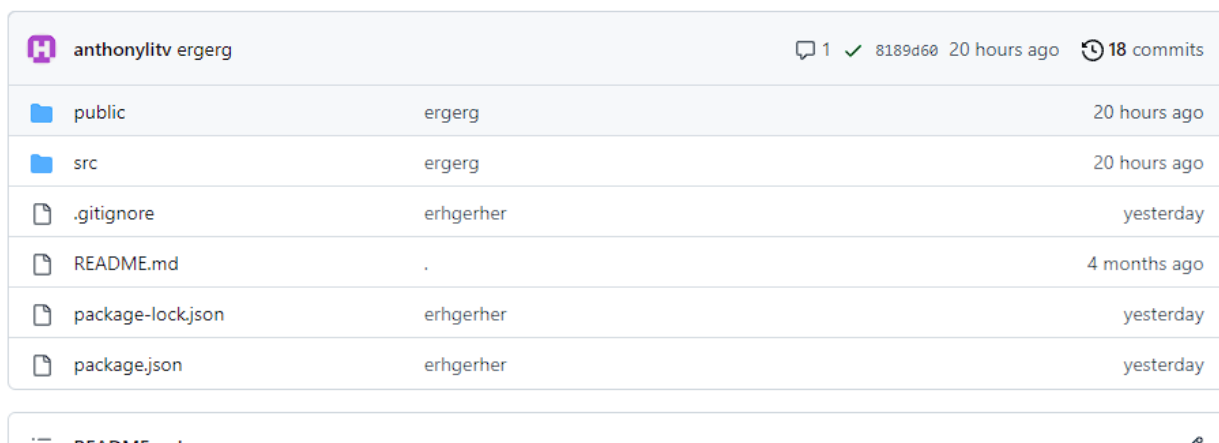


Рисунок 3.12 – Збережені файли web-додатку на github

3.4 Використання web-додатку зі сторони користувача

В момент заходу на сайт – користувач потрапляє на головну сторінку:

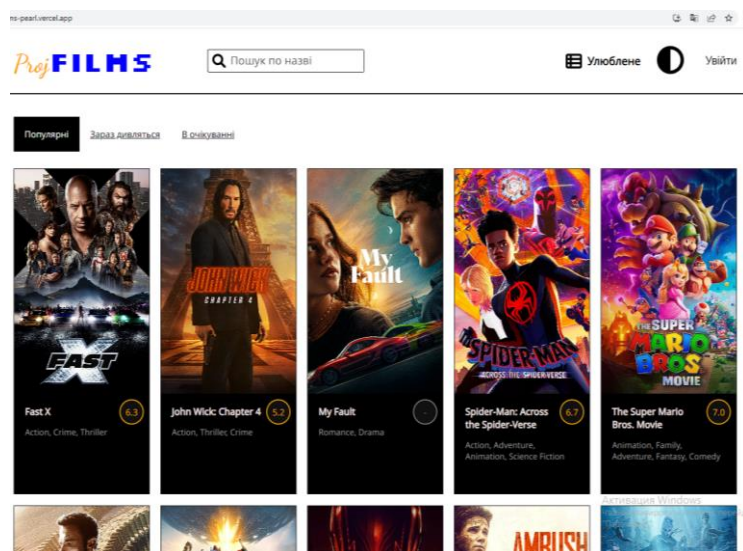


Рисунок 3.13 – Головна сторінка

Користувач може з панелі верхньої здійснити пошук фільму по назві:

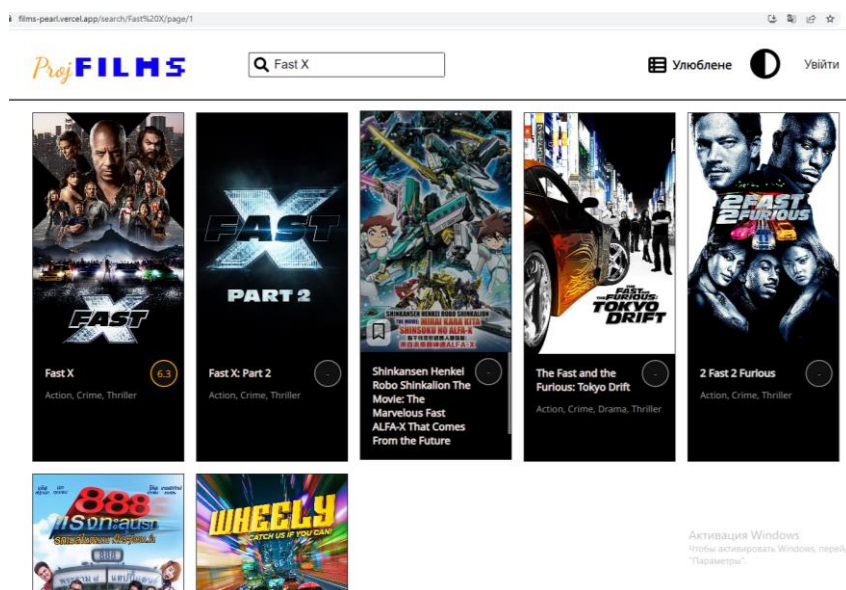


Рисунок 3.14 – Сторінка пошуку

Також користувач може перемикатись між категоріями:



Рисунок 3.15 – Категорії

Також користувач може додати фільм до улюбленого натиснувши на кнопку , яка з'являється, коли наводиться курсор миші на фільм



Рисунок 3.16 – Кнопка для додавання (видалення) фільму до списку вподобань

Сторінка з улюбленими фільмами:

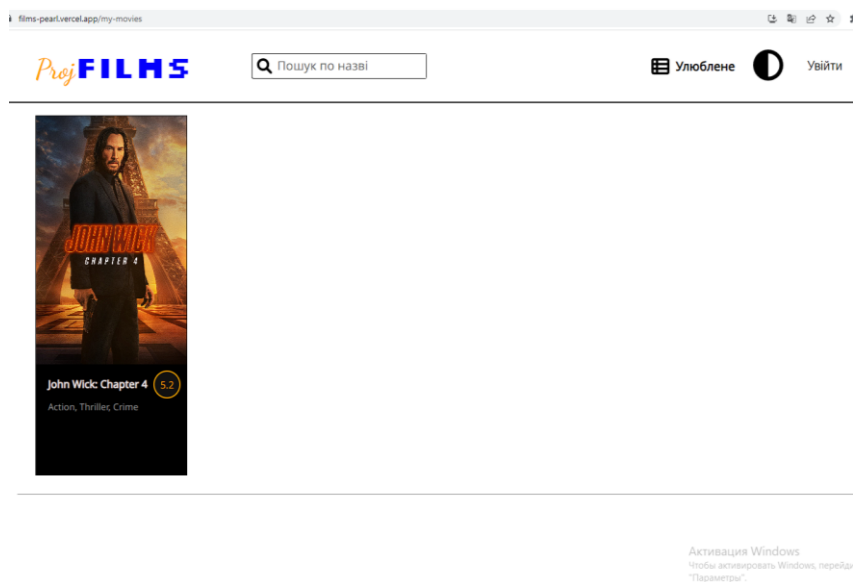


Рисунок 3.17 – Сторінка зі списком вподобань

Користувач може натиснути на будь який фільм і продивитись детальну інформацію про нього

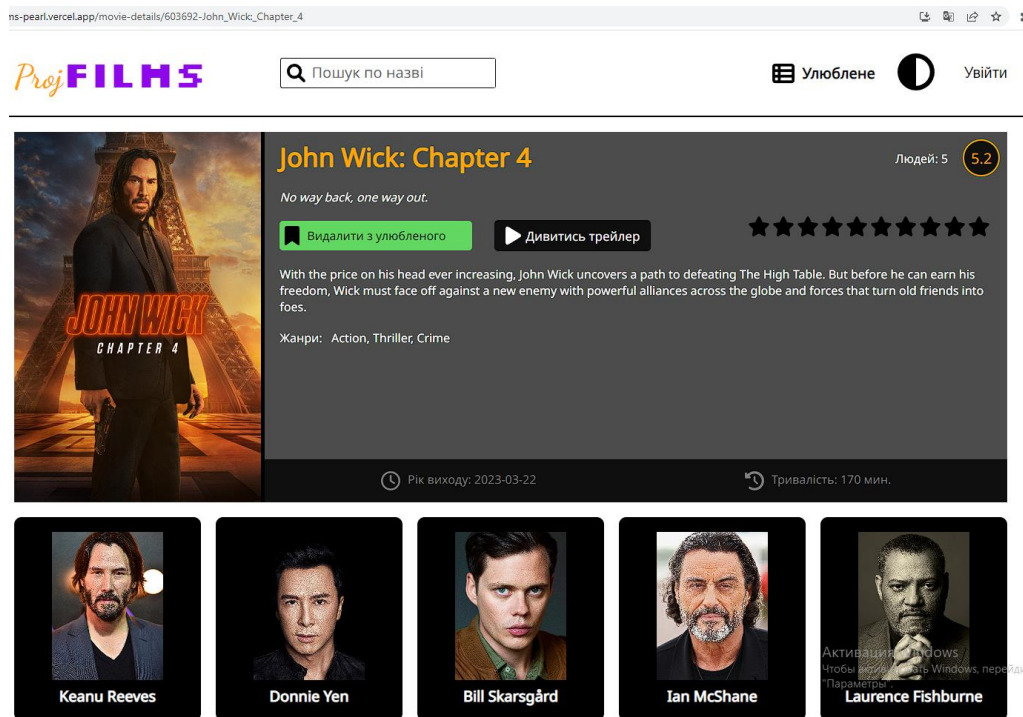


Рисунок 3.18 – Сторінка з детальною інформацією про фільм

На цій сторінці знаходяться також слайдер та рекомендації до фільму

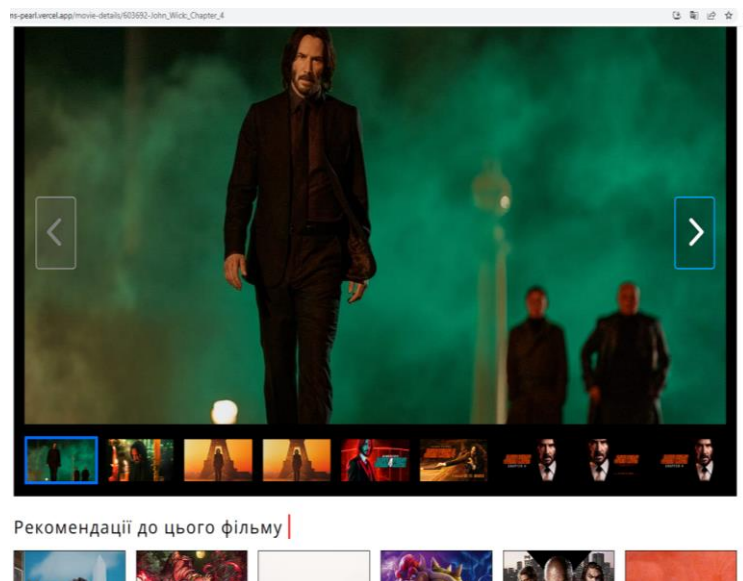


Рисунок 3.19 – Продовження сторінки з детальною інформацією про фільм

Після натискання на кнопку реєстрації в верхній панелі – користувач потрапляє на сторінку з реєстрацією

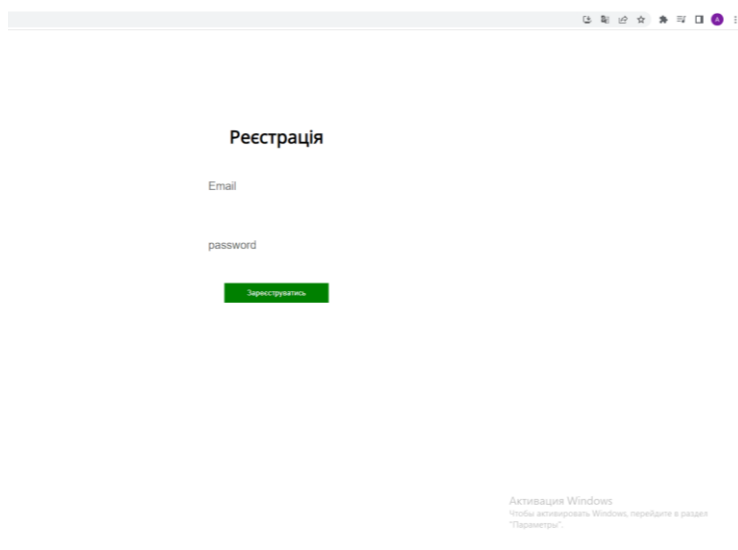


Рисунок 3.22 – Сторінка реєстрації

Потім автоматично сайт перенаправляє на головну сторінку.

Також користувач може зробити вхід натиснувши на відповідну кнопку в верхній панелі

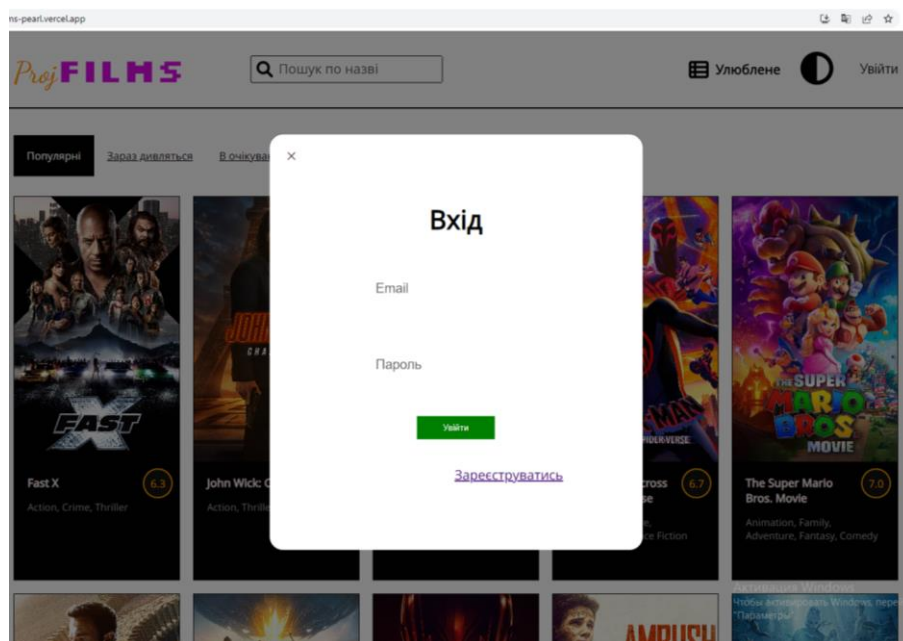


Рисунок 3.23 – Модальне вікно зі входом

3.5 Використання web-додатку зі сторони рецензента

Рецензент може все, що і звичайний користувач. Після входу в акаунт він може поставити оцінку будь якому фільму та залишити коментар. Вони зберігаються в базі даних Firebase. Така можливість є на сторінці з детальною інформацією про фільм.

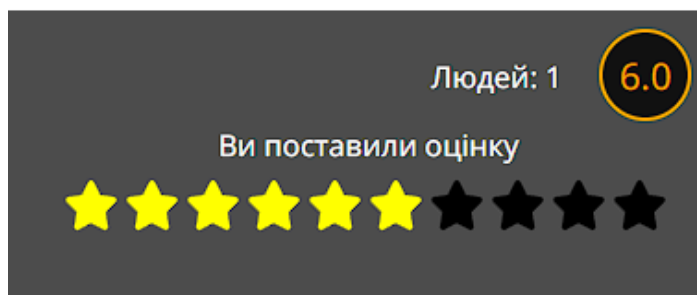


Рисунок 3.24 – Поставлена оцінка

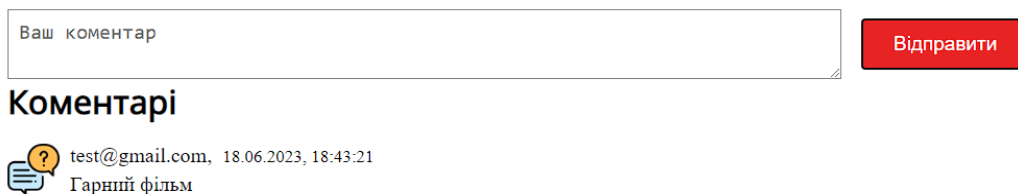


Рисунок 3.25 – Коментарі

ВИСНОВКИ

Web-додаток формування рейтингу фільмів з відкритих джерел створено, щоб надати користувачам платформу для обміну думками та враженнями про доступні для перегляду фільми – оцінювати їх та рецензувати у вільній формі. Основна мета додатку — запропонувати ентузіастам кіно можливість передати свій досвід та почерпнути чужий – повністю досягається за допомогою реалізованого функціоналу додатка.

У процесі розробки були вирішені наступні завдання:

Проведено аналіз тенденцій кіноіндустрії та показано значущість web-додатків для оцінювання фільмів, вплив їх на прокат.

Проведено моделювання роботи web-додатку, взаємодії його і користувачів. Це допомогло з'ясувати концепцію web-додатку.

Реалізовано web-додаток на основі розробленої моделі та структури сайту.

Основна функція програми полягає в тому, щоб слугувати комплексним інструментом для користувачів, щоб досліджувати та оцінювати широкий спектр фільмів. Це дозволить користувачам упереджено переглядати колекції фільмів, отримувати доступ до постерів, трейлерів, описів та іншої важливої інформації для кожного фільму.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нова система рекомендацій фільмів із використанням методів спільної фільтрації та глибокого навчання [Електронний ресурс] – Доступ до ресурсу:
https://www.researchgate.net/publication/331790084_A_Novel_Movie_Recommendation_System_Based_on_Collaborative_Filtering_and_Neural_Networks (Дата звернення: 09.05.23)
2. Етичні міркування в системах рекомендацій: опитування [Електронний ресурс] – Доступ до ресурсу:
<https://link.springer.com/article/10.1007/s00146-020-00950-y> (Дата звернення: 09.05.23)
3. Різноманітність та інклюзивність у кіно: аналіз створеного користувачами контенту на IMDb [Електронний ресурс] – Доступ до ресурсу:
<https://www.imdb.com/title/tt14799404/> (Дата звернення: 09.05.23)
4. Відкриті джерела даних для спільної фільтрації в системах рекомендацій фільмів [Електронний ресурс] – Доступ до ресурсу:
https://www.researchgate.net/publication/343435043_Movie_Recommender_System_Using_Collaborative_Filtering (Дата звернення: 09.05.23)
5. Аналіз настроїв [Електронний ресурс] – Доступ до ресурсу:
<https://techukraine.net/%D0%BF%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9-%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA-%D0%B7-%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7%D1%83-%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D1%97%D0%B2/>
 (Дата звернення: 09.05.23)
6. Відомості про типи архітектури web-додатку [Електронний ресурс] – Доступ до ресурсу: <https://www.netflix.com/ua-ru/browse/genre/839338> (Дата звернення: 09.05.23)

7. Порівняння [Електронний ресурс] – Доступ до ресурсу:
<https://mind.ua/publications/20254602-shcho-vibrati-dlya-domashnogo-pereglyadu-v-ukrayini-porivnyuemo-netflix-amazon-prime-video-apple-tv-me/> (Дата звернення: 09.05.23)
8. Архітектура web додатків [Електронний ресурс] – Доступ до ресурсу: <https://blog.ithillel.ua/articles/web-application-architecture/> (Дата звернення: 09.05.23)
9. Нотація IDEF0 [Електронний ресурс] – Доступ до ресурсу:
https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%9A%D0%BE%D0%BD%D0%B4%D1%96%D1%83%D1%81%20%20%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%B2%D0%B0/page9.html (Дата звернення: 09.05.23)
10. Діаграма варіантів використання [Електронний ресурс] – Доступ до ресурсу: <http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/laboratorna-robota-5-diahramy-variantiv-vykorystannja.pdf> (Дата звернення: 09.05.23)
11. Документація реакт [Електронний ресурс] – Доступ до ресурсу:
<https://uk.legacy.reactjs.org/docs/getting-started.html> (Дата звернення: 09.05.23)
12. Redux Toolkit [Електронний ресурс] – Доступ до ресурсу:
<https://redux-toolkit.js.org/introduction/getting-started> (Дата звернення: 09.05.23)
13. RTK Query [Електронний ресурс] – Доступ до ресурсу:
<https://redux-toolkit.js.org/rtk-query/overview> (Дата звернення: 09.05.23)
14. Developer Mozilla Javascript [Електронний ресурс] – Доступ до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (Дата звернення: 09.05.23)
15. Css in ua [Електронний ресурс] – Доступ до ресурсу:
<https://css.in.ua/> (Дата звернення: 09.05.23)

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

«Web-додаток формування рейтингу відеоконтенту з відкритих джерел»

ПОГОДЖЕНО:

Старший викладач кафедри ІТ

_____ Кузнецов Е. Г.

Студент групи ІТ-92/1

_____ Литвиненко А. С.

Суми 2023

1 Призначення й мета створення web-додатку

1.1 Призначення web-додатку

Web додаток має надавати повноцінну інформацію по фільмам з можливістю поставити оцінку та коментар.

1.2 Мета створення web-додатку

Дати користувачам інтернета можливість шукати будь-яку інформацію про фільм, ставити оцінки та коментарі.

1.3 Цільова аудиторія

До цільової аудиторії web-додатку можна віднести практично всіх людей, що зацікавлені у фільмах.

2 Вимоги до web-додатку

2.1 Вимоги до web додатку в цілому

2.1.1 Вимоги до структури й функціонування web додатку

Web-додаток має бути доступним в мережі Інтернет за адресою <https://films-pearl.vercel.app>. Web-додаток повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2.1.2 Вимоги до персоналу

Від персоналу не має вимагатися особливих технічних навичок для підтримки й експлуатації web-додатку.

2.1.3 Вимоги до збереження інформації

Оцінки та коментарі по фільмам у web-додатку будуть зберігатися у базі даних реалізованій засобами системи управління базами даних Firebase.

2.1.4 Вимоги до розмежування доступу

Розроблюваний web-додаток має бути загальнодоступним.

Відповідно до прав доступу до інформації у web-додатку, усіх користувачів можна поділити тільки на відвідувачів та рецензентів.

Відвідувачі можуть переглядати усі сторінки web-додатку, але не можуть ставити оцінки та додавати коментарі в порівнянні з рецензентами

2.2 Структура web додатку

2.2.1 Загальна інформація про структуру web додатку

Структура web-додатку являє собою набір сторінок, які також є пунктами головного меню.

Такими розділами є:

Головна – на сторінці зображені меню сортування та всі фільми

Пошук – сторінка зі знайденими користувачем фільмами

Окремий фільм – сторінка з детальною інформацією про окремий фільм та можливістю поставити оцінку та додати коментар

2.2.2 Навігація

Для навігації, у шапці буде створена система контент меню. Меню необхідне для швидкого переміщення користувача по усім доступним сторінкам. Меню буде відображатися на всіх сторінках, щоб відвідувач міг в будь-який момент часу перейти на будь-яку сторінку web-додатку.

2.2.3 Наповнення web-додатку (контент)

Вся інформацію для наповнення web-додатку буде братись із API сервера

2.2.4 Дизайн та структура додатку

Стиль web-додатку має бути сучасним, приємним для сприйняття, у якості основних кольорів було запропоновано використати червоні та білі відтінки, так як це кольори, які переважають в стилістиці студії.

Основою мають бути фотографії гарної якості, виконані фотографом студії, web-додаток має бути інтуїтивно зрозумілим для використання.

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Перегляд детальної інформації про фільм	Клієнт
UN-02	Пошук окремого фільма по його назві	Клієнт
UN-03	Можливість поставити оцінку фільму	Рецензент
UN-04	Можливість додати коментар до фільму	Рецензент
UN-05	Можливість додавати фільм до улюбленого, а також видаляти	Клієнт

2.3.2 Функціональні вимоги

На основі потреб користувача були визначені такі функціональні вимоги:

- можливість поставити оцінку фільму та додати коментар

- відображення результатів пошуку на web-сторінці;
- можливість фільтрування та сортування фільмів
- можливість переглянути повну детальну інформацію про кожен окремий фільм
- можливість додавати фільм до улюбленого, а також видаляти

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

ІД	Системні вимоги	Пріоритет	Опис
SR-01	Пошук фільма	М	Надає можливість клієнту знайти фільм по його назві
SR-02	Сортування та фільтрування фільмів	М	Надає можливість клієнту відфільтрувати фільми та відсортувати
SR-03	улюблені фільмами	М	Формує можливість додавати та видаляти фільми з улюбленого
SR_04	Можливість поставити оцінку та додати коментар	М	

Умовні позначення в таблиці А.2:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація web-додатку відбувається з використанням:

- HTML, CSS
- Javascript
- React
- FireBase
- GIT, GitHub

2.4.2 Вимоги до лінгвістичного забезпечення

Web-додаток буде брати інформацію з відкритого джерела, де інформація про фільми на англійській мові. Все інше буде на українській мові.

2.4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Web-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище.

3 Склад і зміст робіт зі створення web-додатку

Докладний опис етапів роботи зі створення web-додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи створення web-додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Постановка цілей необхідних для досягнення певного результату	1 день
2	Складання технічного завдання	3 дні
3	Підготовка прототипу	2 дні
4	Створення макету дизайну web-додатку	2 дні
5	Верстка	3 дні
6	Робота над модулями для web-додатку	2 дні
7	Робота з контентом	1 день
8	Розміщення контенту та каталогів з фото у web-додатку	1 день
9	Перевірка працездатності web-додатку	1 день
10	Завершення роботи	1 день
	Загальна тривалість робіт	18 днів

4 Вимоги до складу й змісту робіт із введення web-додатку в експлуатацію

Для того, щоб web-додатком могли користуватися відвідувачі необхідно розмістити його у мережі Інтернет, тому необхідно скористатись хостингом. На хостинг переноситься web-додаток. Буде використаний хостинг Vercel.

ДОДАТОК Б ПЛАНУВАННЯ РОБІТ

1 Деталізація мети проекту методом SMART.

Продуктом дипломного проекту є web-додаток формування рейтингу фільмів з відкритих джерел.

Деталізацію мети проекту виконують за допомогою SMART-методу. Це дозволяє більш конкретно представити призначення розроблюваного продукту.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific (конкретна)	Створити web-додаток формування рейтингу фільмів з відкритих джерел
Measurable (вимірювана)	Результатом роботи проекту є оцінка замовника
Achievable (досяжна)	Реалізації системи здійснюється за допомогою середовища розробки Visual Studio Code, з використанням бібліотеки React
Relevant (реалістична)	У наявності є всі необхідні технічні та програмні засоби.
Time-framed (обмежена у часі)	Ціль має часове обмеження. Робота повинна бути виконана у терміни, що були оговорені замовником проекту. Проект повинен бути виконаний згідно з календарним планом.

2 Планування змісту робіт

Основним інструментом для планування змісту структури робіт служить WBS діаграма – графічне подання згрупованих елементів проекту у вигляді

пакета робіт, які ієрархічно пов'язані з продуктом проекту. Побудуємо структуру WBS, у якій детально опишемо роботи, які потрібно виконати на кожному етапі створення проекту. Виконаємо декомпозицію робіт для даного проекту. Діаграма WBS зображена на рис. Б.1.

3 Планування структури організації, для впровадження готового проекту (OBS)

Після побудови WBS розробимо організаційну структуру виконавців OBS. Організаційна структура проекту стосується тільки внутрішньої організаційної структури проекту і не стосується відносин проектних груп чи учасників з батьківськими організаціями. Діаграма OBS зображена на рис. Б.2. Список виконавців, що функціонують в проекті знаходиться в табл. Б.2

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Литвиненко А.С.	Виконує розробку основного функціоналу проекту та інтерфейс користувача
Проектувальник	Литвиненко А.С.	Розробляє дизайн програми та архітектуру.
Тестувальник	Литвиненко А.С.	Відповідає за тестування функціоналу та дизайну додатку, перевірку моделі на адекватність.
Косультант проекту	Кузнєцов Е. Г.	Формує завдання на розробку проекту.
Менеджер проекту	Литвиненко А.С.	Відповідає за виконання термінів, розподіл ресурсів та завдань між учасниками. Виконує збір та аналіз даних.

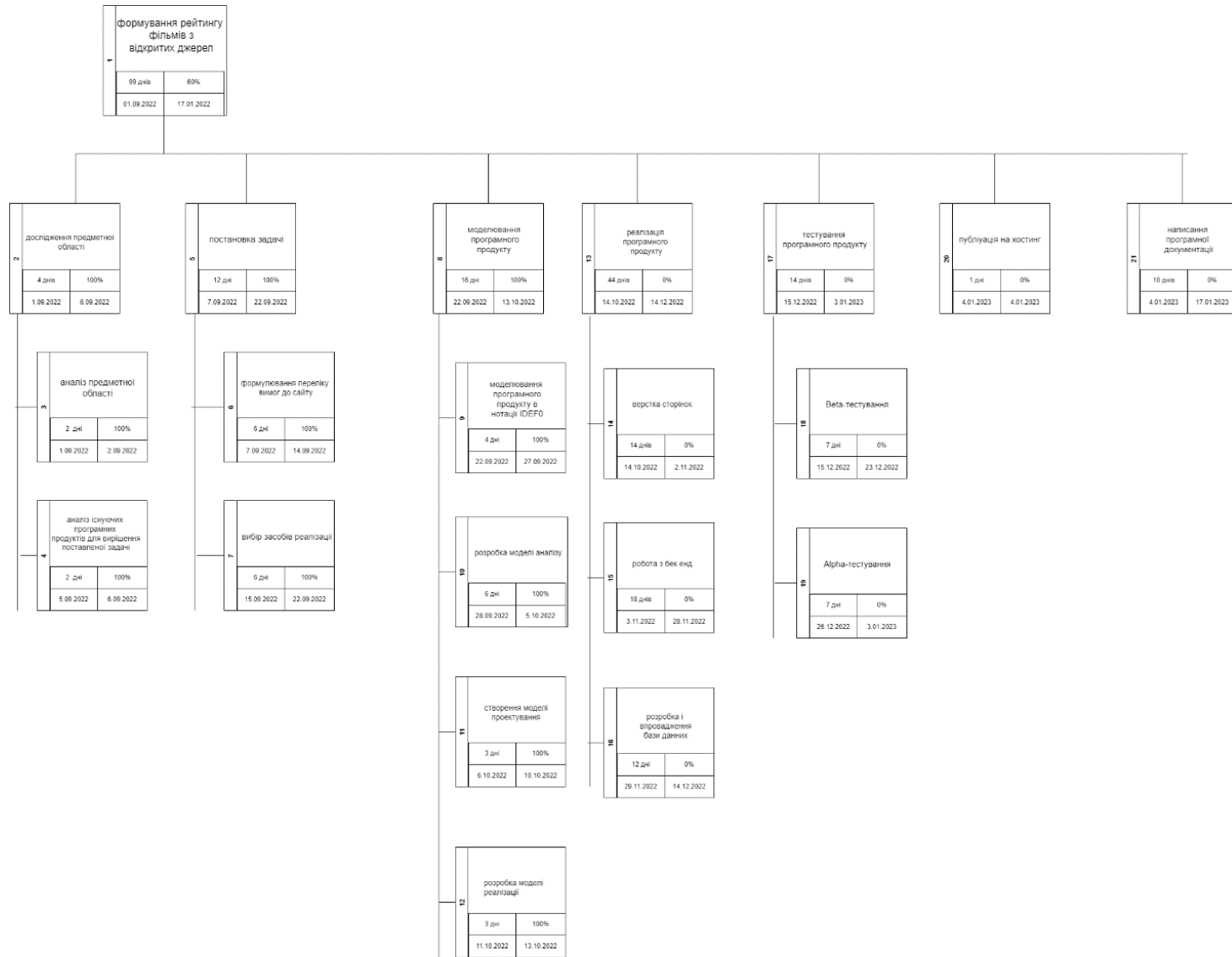


Рисунок Б.1 – WBS-структура робіт проекту

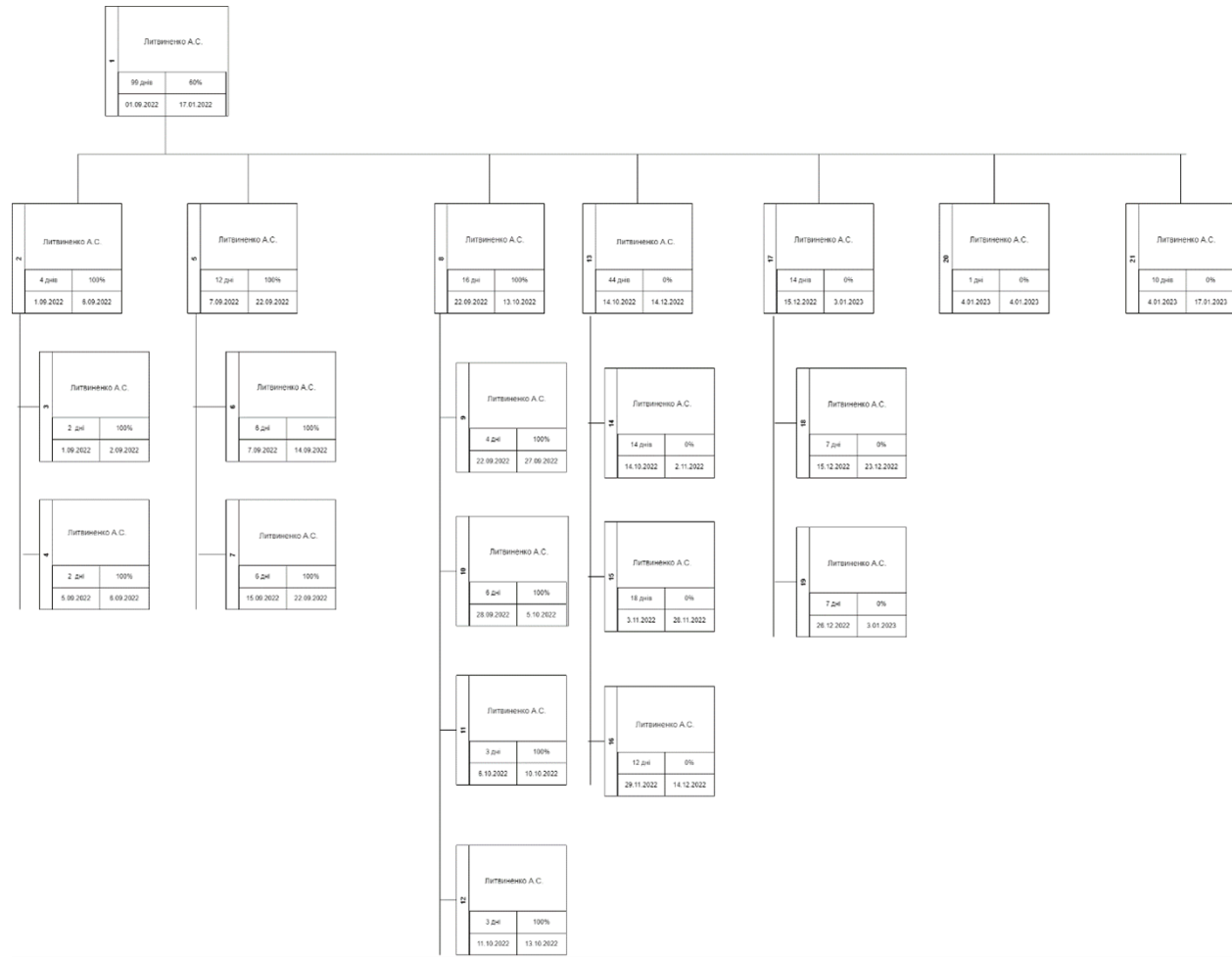


Рисунок Б.2 – OBS-структура робіт проекту

4 Діаграма Ганта

Далі побудуємо календарний план виконання дипломного проекту. Найпоширеніший формат графіка в будь-якій галузі — діаграма Ганта. Цей графік дозволяє менеджерам проекту і всій команді розробників візуалізувати графіки часу і взаємозв'язок між окремими завданнями та етапами роботи над проектом. Тривалість виконання робіт зазначена в днях, але фактична тривалість виконання робіт приблизно дорівнює 2-3 години на день. Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, з урахуванням вихідних та святкових днів, побудовано календарний графік. Діаграма Ганта зображена на рис. Б.3.

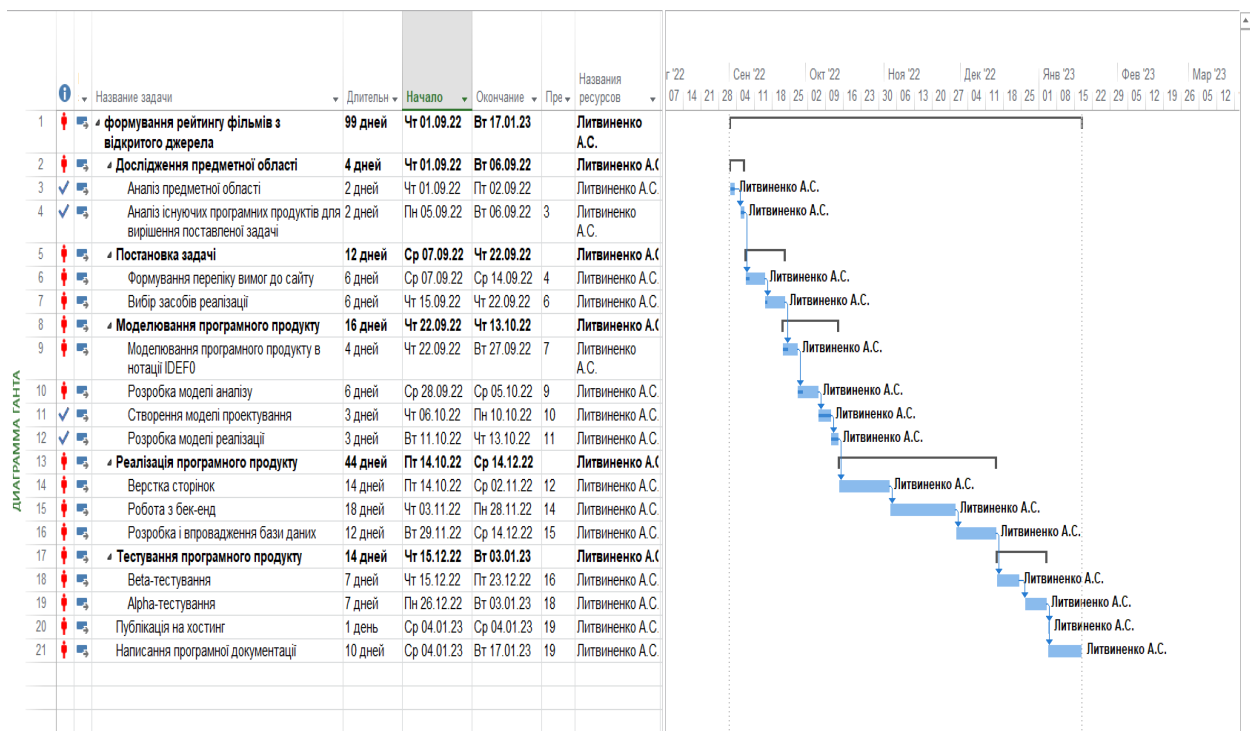


Рисунок Б.3 – Діаграма Ганта

5 Аналіз ризиків

Виконаємо якісну і кількісну оцінку ризиків роботи. При якісній оцінці визначимо ризики, що потребують швидкого реагування. Така оцінка визначить ступінь важливості ризику і дозволить вибрати спосіб реагування. Кількісна оцінка ризиків буде виконана для більш повної ідентифікації ризиків та ступеня їхнього впливу на виконання проекту. Кількісна і якісна оцінка ризиків можуть використовуватися окремо або разом, залежно від наявного часу і бюджету, необхідності в кількісній або якісній оцінці ризиків.

У табл. Б.5 показана класифікація ризиків за показниками ймовірності виникнення ризику та величині втрат. Далі виконаємо планування реагування на ризики — це розробка методів і технологій зниження негативного впливу ризиків на проект. Визначимо ефективність розробки реагування на проект, визначимо чи будуть наслідки впливу ризику на проект позитивними або негативним.

Оцінюємо ризики за показниками, що знаходяться в табл. Б.3. На основі оцінки будуємо матрицю ймовірності виникнення ризиків та впливу ризику, що зображена в таблиці Б.4.

Таблиця Б.3 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

Таблиця Б.4 – Матриця ймовірності виникнення ризиків та впливу ризику

3		RS_1,	RS_3, RS_5
2		RS_2	RS_8
1	RS_7	RS_4, RS_6	
	1	2	3

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики.

На підставі отриманого значення індексу ризику класифікують: за рівнем ризику, що знаходиться в табл. Б.5.

Таблиця Б.5 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять (номера)
1	Прийнятні	$1 \leq R \leq 2$	4,6,12
2	Виправдані	$3 < R \leq 4$	2,3
3	Недопустимі	$6 < R \leq 9$	1,5,8

Таблиця Б.6 – Оцінка ймовірності виникнення, величини витрат та індексу ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Нечітке завдання на розробку	Середня	Високий	6	<p>1. Ясно і однозначно обговорити із керівником усі види вимог.</p> <p>2. Скласти глосарій для запобігання розбіжностей у розумінні слів та термінів.</p> <p>3. Періодичний контроль керівником етапів роботи.</p>	Попередження	<p>При виявленні невідповідностей деяких характеристик продукту заявленим вимовам потрібно уважно та чітко окреслити те, що було виконано невірно та зробити правки.</p>

Продовження таблиці Б.6

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг	План А	Тип стратегії реагування	План Б
RS_2	Відкритий	Низька кваліфікація розробників проекту	Середня	Середній	4	1. Підвищити кваліфікацію розробнику. 2. Переглянути онлайн-ресурси для підвищення рівня знань.	Пом'якшення	Врахувати час на підготовку розробника.

Продовження таблиці Б.6

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг	План А	Тип стратегії реагування	План Б
RS_3	Відкритий	Неоптимальний розподіл часу	Висока	Високий	9	Провести аналіз актуальності найважливіших процесів та робіт. Звернути особливу увагу на правильність розподілу часу. Правильно визначити пріоритети виконання робіт. Чітко дотримуватися календарного плану	Пом'якшення	Змінити порядок пріоритетів робіт. Знайти способи оптимізації роботи із вже існуючою розстановкою. Обговорити варіанти внесення правок до термінів реалізації із керівником.

Продовження таблиці Б.6

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг	План А	Тип стратегії реагування	План Б
RS_4	Відкритий	Не вірна оцінка масштабів проекту	Низький	Середній	2	Провести детальний аналіз проекту. Визначити основні етапу проекту, розподілити час на їх виконання. Проаналізувати масштаби проекту на основі додаткових джерел.	Пом'якшення	Переоцінка масштабів проекту. Перебудова стратегії реалізації проекту.

Продовження таблиці Б.6

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг	План А	Тип стратегії реагування	План Б
RS_5	Відкритий	Помилки проектування	Висока	Високий	9	На етапі проектування тісно співпрацювати із керівником та на певних етапах демонструвати поточні результати.	Пом'якшення	Здійснювати проміжний контроль результатів в ході виконання проекту.
RS_6	Відкритий	Відсутність резервних копій даних	Низька	Середній	2	1. Налаштувати автоматичне збереження даних. 2. Зберігати дані на різних носіях інформації.	Попередження	Робити копію даних після кожного виконаного етапу.

Продовження таблиці Б.6

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг	План А	Тип стратегії реагування	План Б
RS_7	Відкритий	Реалізація непотрібної функціональності	Низька	Низький	1	Попередити керівника про можливість додаткового функціоналу.	Використання	Обговорити вигоди і збитки від можливих змін проекту.
RS_8	Відкритий	Вибір неефективної технології розробки	Середня	Високий	6	1. Проаналізувати методи та засоби, для виконання проекту. 2. Обрати зрозумілу та легку в використанні технологію розробки.	Пом'якшення	Виділити час та ресурси на пошуки покращення обраної технології. Застосувати допоміжні ресурси.