

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Красюкову Микиті Сергійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення вебзастосунку формування списку рекомендованих фільмів на основі оброблення опитувальника вподобань користувача

затверджена наказом університету від 26 травня 2026 року № 435Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 20 червня 2026 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, офіційна документація технологій та програмних платформ.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз існуючих вебзастосунків формування списку рекомендованих фільмів на основі оброблення опитувальників вподобань користувачів.2. Проектування моделей гібридної системи рекомендації фільмів.3. Розроблення вебзастосунку формування списку рекомендованих фільмів на основі оброблення опитувальника вподобань користувача.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми персоналізації контенту шляхом генерацій рекомендацій на основі опитувальника вподобань, постановка задачі, вихідні дані для вирішення задачі, етапи виконання роботи, тестування створеного рішення.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	13.04.2026	
2	Аналіз завдання, підбір літератури	14.04.26-24.04.26	
3	Аналіз літератури з проблеми, що вирішується	25.04.26-28.04.26	
4	Аналіз існуючих методів розпізнавання	29.04.26-01.05.26	
5	Формування кроків вирішення проблеми	02.05.26-05.05.26	
6	Програмна реалізація	06.05.26-22.05.26	
7	Аналіз отриманих результатів	23.05.26-28.05.26	
8	Оформлення пояснювальної записки	28.05.26-09.06.26	
9	Перевірка на нормоконтроль	10.06.26-19.06.26	
10	Перевірка на плагіат	11.06.26-30.06.26	
11	Рецензування	20.06.26-30.06.26	
12	Підготовка презентації та доповіді	20.06.26-30.06.26	
13	Занесення роботи в електронний архів	30.06.26-09.07.26	
14	Попередній захист кваліфікаційної роботи	30.06.26-09.07.26	

Дата видачі завдання 13 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Творошенко І. С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 76 с., 10 табл., 32 рис., 31 джерело.

ВЕБЗАСТОСУНОК, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, КОНТЕНТНА ФІЛЬТРАЦІЯ, РЕКОМЕНДАЦІЇ, C#, POSTGRESQL.

Об'єктом роботи є процес розроблення вебзастосунку для формування списку рекомендованих фільмів на основі оброблення опитувальника вподобань користувача.

Метою роботи є розроблення вебзастосунку для формування списку рекомендованих фільмів шляхом оброблення вподобань користувача.

Під час роботи використано: методи машинного навчання (контентна та колаборативна фільтрація), методи веброблення.

Практична цінність вебзастосунку полягає в автоматизації формування персоналізованих рекомендацій на основі вподобань і поведінкової активності користувача.

Розроблено вебзастосунок, що забезпечує проходження опитувальника, ведення списку перегляду, оцінювання фільмів і генерацію рекомендацій на основі гібридної рекомендаційної системи.

Область застосування: персоналізовані рекомендаційні сервіси, системи керування медіаконтентом.

C#, COLLABORATIVE FILTERING, CONTENT-BASED FILTERING, POSTGRESQL, RECOMMENDATIONS, WEB APPLICATION.

The objective of this work is the process of developing a web application for generating a list of recommended films based on processing a user preference questionnaire.

The goal of this work is to develop a web application for generating a list of recommended films by processing user preferences.

The following were used during the work: machine learning methods (content-based and collaborative filtering), web development methods.

The practical value of the web application lies in the automation of generating personalized film recommendations based on user preferences and behavioral activity on the platform.

A web application has been developed that enables passing a preference questionnaire, maintaining a watch list, rating films, and generating personalized recommendations based on a hybrid recommendation system.

Applications: personalized recommendation services, media content management systems.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	9
1 Аналіз існуючих вебзастосунків формування списку рекомендованих фільмів на основі оброблення опитувальників вподобань користувачів....	10
1.1 Аналіз сучасних вебзастосунків формування списку рекомендованих фільмів в Україні та за кордоном.....	10
1.2 Аналіз методів побудови рекомендаційних систем та підходів до вирішення проблем персоналізації	18
1.3 Постановка задачі	21
2 Проєктування моделей гібридної системи рекомендації фільмів.....	22
2.1 Проєктування структури бази даних вебзастосунку.....	22
2.2 Модель формування профілю користувача на основі опитувальника вподобань	27
2.3 Модель побудови векторного подання фільмів.....	29
2.4 Модель рекомендації на основі поєднання контентної та колаборативної фільтрації.....	31
2.4.1 Опис методу побудови контентної фільтрації.....	32
2.4.2 Опис методу побудови колаборативної фільтрації.....	33
2.5 Модель оцінювання якості рекомендацій	36
3 Розроблення вебзастосунку формування списку рекомендованих фільмів на основі оброблення опитувальника вподобань користувача	40
3.1 Вибір інструментальних засобів реалізації вебзастосунку формування списку рекомендованих фільмів на основі оброблення опитувальника вподобань користувача.....	40
3.2 Етапи програмної реалізації формування списку рекомендованих фільмів на основі оброблення опитувальника вподобань користувача.....	41
3.3 Інструкція користувача	47

	6
3.4 Тестування розробленого вебзастосунку та аналіз результатів.....	56
3.5 Перспективи подальшої роботи	69
Висновки	71
Перелік джерел посилення	73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

% – символ відсотка, що використовується для вираження статистичних показників

API – Application Programming Interface (інтерфейс програмного застосунку, використовується для обміну даними між клієнтом і сервером)

CF – Collaborative Filtering (колаборативна фільтрація, метод побудови рекомендацій на основі поведінки користувачів)

CTR – Click-Through Rate (показник клікабельності рекомендацій, що визначає частку рекомендацій, за якими користувач здійснив перехід)

Endpoints – кінцеві точки доступу API, конкретні URL-адреси, через які клієнтський застосунок звертається до сервера для взаємодії з ресурсами та обміну даними

ERD – Entity-Relationship Diagram (діаграма «сутність–зв’язок», використовується для моделювання структури бази даних та взаємозв’язків між сутностями)

JSONB – JSON Binary (бінарний формат зберігання JSON-даних у PostgreSQL)

JWT – JSON Web Token (токен автентифікації та авторизації користувачів, що використовується для безпечної передачі даних між клієнтом і сервером)

K – Kilo (загальноприйняте скорочення для позначення тисячі)

MVC – Model-View-Controller (архітектурний шаблон проєктування програмного забезпечення, що розділяє модель даних, представлення та логіку керування)

ORM – Object-Relational Mapping (технологія відображення об’єктів програмного коду на структури реляційної бази даних)

RMSE – Root Mean Square Error (середньоквадратична помилка)

TMDB – The Movie Database (онлайн-база даних фільмів та серіалів, що використовується як джерело додаткової інформації про кіно контент)

TF-IDF – Term Frequency – Inverse Document Frequency (метод статистичного аналізу тексту для визначення важливості слів у документах)

UML – Unified Modeling Language (уніфікована мова моделювання для опису архітектури програмних систем)

ВСТУП

Сучасний розвиток інформаційних технологій супроводжується стрімким зростанням обсягів цифрового медіаконтенту та ускладненням процесів його споживання. Користувачі щоденно стикаються з великою кількістю фільмів, серіалів та інших відеоматеріалів, що ускладнює процес вибору релевантного контенту. У результаті виникає проблема інформаційного перевантаження, коли традиційні підходи до пошуку перестають бути ефективними.

У зв'язку з цим особливого значення набувають рекомендаційні системи, які дозволяють автоматизувати процес підбору контенту та підвищити якість користувацького досвіду. Такі системи базуються на методах аналізу даних, статистичного моделювання та елементах машинного навчання, що дає змогу враховувати як явні, так і приховані вподобання користувача.

Актуальність даної роботи зумовлена необхідністю розроблення ефективних підходів до персоналізації рекомендацій у вебзастосунках для перегляду фільмів. Особливої уваги потребує поєднання різних джерел інформації про користувача, таких як опитувальники вподобань та поведінкові дані, що дозволяє підвищити точність формування рекомендацій на початкових етапах використання системи.

Практична цінність роботи полягає у створенні вебзастосунку, який забезпечує автоматизоване формування персоналізованих рекомендацій фільмів та може бути використаний у медіасервісах і платформах потокового відео.

Результати роботи можуть бути використані у сфері розробки рекомендаційних систем, потокових сервісів та інших інформаційних систем, орієнтованих на персоналізацію контенту.

1 АНАЛІЗ ІСНУЮЧИХ ВЕБЗАСТОСУНКІВ ФОРМУВАННЯ СПИСКУ РЕКОМЕНДОВАНИХ ФІЛЬМІВ НА ОСНОВІ ОБРОБЛЕННЯ ОПИТУВАЛЬНИКІВ ВПОДОБАНЬ КОРИСТУВАЧІВ

1.1 Аналіз сучасних вебзастосунків формування списку рекомендованих фільмів в Україні та за кордоном

У сучасних умовах розвитку суспільства, а саме в період активного впровадження нових інформаційних технологій, суттєво змінився процес споживання людиною медіаконтенту, зокрема фільмів і серіалів. Якщо раніше відеоконтент переглядався переважно за допомогою телебачення або кінотеатрів, то зараз більша частина аудиторії використовує цифрові платформи [1], такі як Netflix, Disney+, HBO Max та Amazon Prime Video, що забезпечують швидкий доступ до великої кількості матеріалів у будь-який час. Поширення мережі Інтернет, розвиток мобільних пристроїв та вдосконалення вебтехнологій створили умови, за яких користувач отримав можливість самостійно формувати власний інформаційний простір і обирати контент відповідно до особистих інтересів.

Глобальна зміна підходів до перегляду фільмів стала невід'ємною частиною сучасного цифрового суспільства. Користувач більше не обмежений у перегляді телевізійною програмою чи розкладом кінотеатрів, зараз він має набагато більшу свободу у виборі. Глядач має доступ до неймовірно великої кількості фільмів та іншого медіаконтенту, який стає доступним тільки завдяки появі нових технологій. Перед користувачем постають фільми з різними жанрами, акторами, режисерами, темами та ідеями, через що процес вибору стає надзвичайно складним. Така доступність створює труднощі, пов'язані з пошуком саме того контенту, який найбільше відповідає індивідуальним уподобанням, що призводить до втоми користувача та втрату бажання витрачати час на вирішення цієї проблеми [2].

При наявності десятків тисяч фільмів, серіалів та інших мультимедійних матеріалів процес вибору перестає бути простим, швидким та може супроводжуватися невпевненістю у власному виборі. Така ситуація негативно впливає на загальний користувацький досвід та знижує рівень задоволення від використання цифрових платформ.

Одним з найпопулярніших інструментів боротьби з інформаційним перевантаженням під час пошуку контенту є система його фільтрації. Але навіть за наявності зручних фільтрів за жанрами, темами, датою виходу, режисерами, акторами чи рейтингом користувачеві часто складно швидко знайти саме той варіант, який буде відповідати його настрою, інтересам та попередньому досвіду перегляду.

Тому в наш час набувають популярності застосунки, які мають вбудовану систему рекомендацій. Їх головним призначенням є спрощення процесу пошуку та перегляду контенту шляхом аналізу вподобань користувача. Такі застосунки дозволяють не лише скоротити час на вибір фільму, а й підвищити ймовірність того, що запропонований варіант буде дійсно цікавим для конкретної людини. Це особливо важливо в умовах високої конкуренції між цифровими платформами, де якість персоналізації безпосередньо впливає на рівень залучення користувачів.

Важливим елементом у роботі рекомендаційних систем є визначення смакових уподобань користувача. Для цього у вебзастосунках часто використовуються опитувальники, які дозволяють зібрати базову інформацію про жанрові інтереси, улюблені теми, бажані формати контенту та інші важливі характеристики. Такий підхід дозволяє сформувати більш точний початковий профіль користувача ще до накопичення історії переглядів. Це є важливим елементом для побудови якісної системи рекомендацій, особливо в нових або спеціалізованих вебзастосунках.

Сучасні вебзастосунки для пошуку та перегляду фільмів суттєво відрізняються від звичайних каталогів минулих років. Якщо раніше основною функцією таких сервісів було лише зберігання інформації про

фільми та надання короткого опису, то сьогодні вони виконують значно ширший спектр завдань. Такі платформи аналізують поведінку користувача, враховують його оцінки, історію переглядів, пошукові запити, обрані жанри та час активності. На основі цього формується персоналізований список пропозицій, який дозволяє значно спростити процес вибору контенту.

Розглянемо популярні вебзастосунки, які дозволяють отримати рекомендації щодо перегляду фільмів.

Netflix є однією з найбільших світових потокових платформ, яка надає користувачам доступ до великої бібліотеки фільмів, серіалів і документального контенту (рис. 1.1) [3].

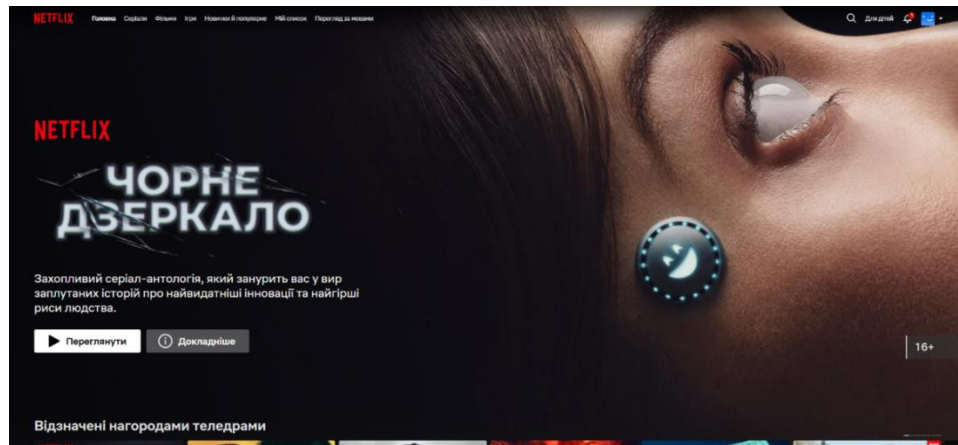


Рисунок 1.1 – Інтерфейс головної сторінки вебзастосунку Netflix

Основною особливістю сервісу є глибока персоналізація інтерфейсу та контенту, яка формується на основі взаємодії користувача з платформою. При першій реєстрації система пропонує користувачеві обрати кілька фільмів, які йому подобаються, щоб сформувати початковий профіль вподобань. Надалі рекомендації уточнюються на основі накопиченої активності – часу перегляду, частоти повернення до певного контенту та поведінкових шаблонів.

Переваги потокової платформи Netflix:

- висока точність рекомендацій завдяки великому обсягу даних про поведінку користувача;

- адаптивний інтерфейс, який змінюється залежно від інтересів;
- швидкий доступ до релевантного контенту без необхідності ручного пошуку;

- постійне оновлення бібліотеки та додавання нового контенту.

Недоліки потокової платформи Netflix:

- нові користувачі отримують менш точні рекомендації через відсутність історії переглядів;

- існує ризик повторюваності контенту у рекомендаціях;

- обмежена видимість менш популярних фільмів через алгоритмічну пріоритизацію.

IMDb (Internet Movie Database) є однією з найбільших у світі інформаційних баз даних про фільми та кінематограф, що належить компанії Amazon (рис. 1.2) [4].

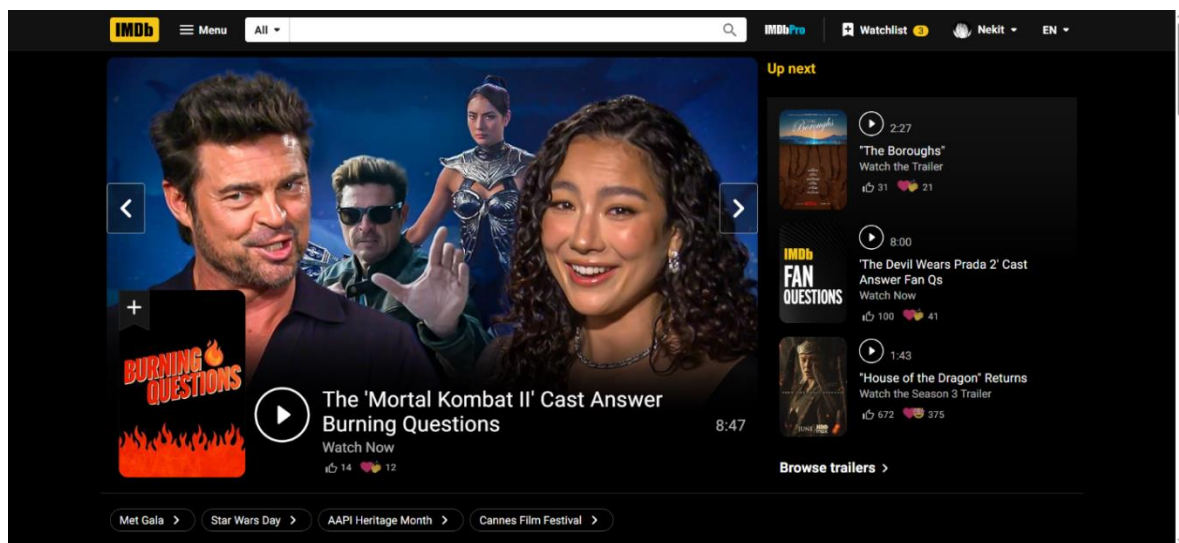


Рисунок 1.2 – Інтерфейс головної сторінки вебзастосунку IMDb

Окрім довідкової інформації, платформа дозволяє користувачам оцінювати фільми, залишати рецензії та створювати власні списки перегляду. Персоналізовані рекомендації формуються на основі виставлених оцінок – система порівнює дані користувача з оцінками людей зі схожими вподобаннями.

Для кожної рекомендації відображається перелік фільмів, на основі яких вона була сформована.

Переваги інформаційної бази даних IMDb:

- найбільша структурована база даних про фільми і серіали у світі;
- наявність детальної рейтингової системи та користувацьких оцінок;
- можливість створення персональних списків перегляду;
- пояснення до кожної рекомендації з переліком фільмів.

Недоліки інформаційної бази даних IMDb:

- персоналізація недоступна без реєстрації та виставлення оцінок;
- перевантажений інтерфейс ускладнює навігацію новим користувачам;
- рекомендації доступні лише у спеціальному розділі, а не інтегровані в основний перегляд.

Letterboxd є соціальною платформою для ведення обліку переглянутих фільмів, написання рецензій та створення персональних списків (рис. 1.3) [5].

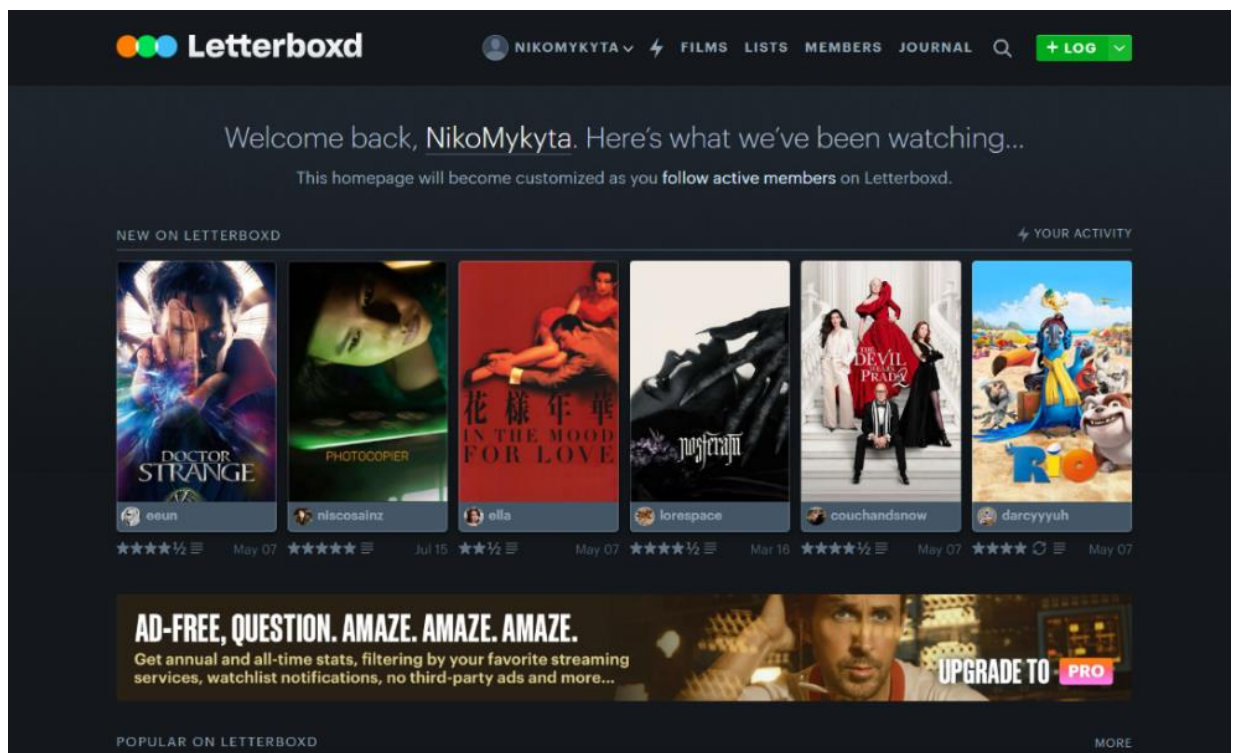


Рисунок 1.3 – Інтерфейс головної сторінки вебзастосунку Letterboxd

Окрім довідкової інформації, основою роботи є фіксація переглядів та оцінок користувача, а також взаємодія з іншими учасниками спільноти. Рекомендації на платформі формуються на основі схожості оцінок між користувачами.

Переваги соціальної платформи Letterboxd:

- детальне структурування історії переглядів та особистих рецензій;
- рекомендації схожих фільмів на сторінці кожної стрічки;
- можливість створення великої кількості персональних списків;
- зручність для відстеження власної кінематографічної історії.

Недоліки соціальної платформи Letterboxd:

- відсутній повноцінний персоналізований розділ рекомендацій у межах самої платформи;
- менш ефективна робота при низькій активності користувача;
- відсутність опитувальника або іншого механізму для нових користувачів без історії переглядів.

Megogo є українською мультимедійною платформою, яка поєднує функції онлайн-кінотеатру, телебачення та сервісу персоналізованих рекомендацій (рис. 1.4) [6].

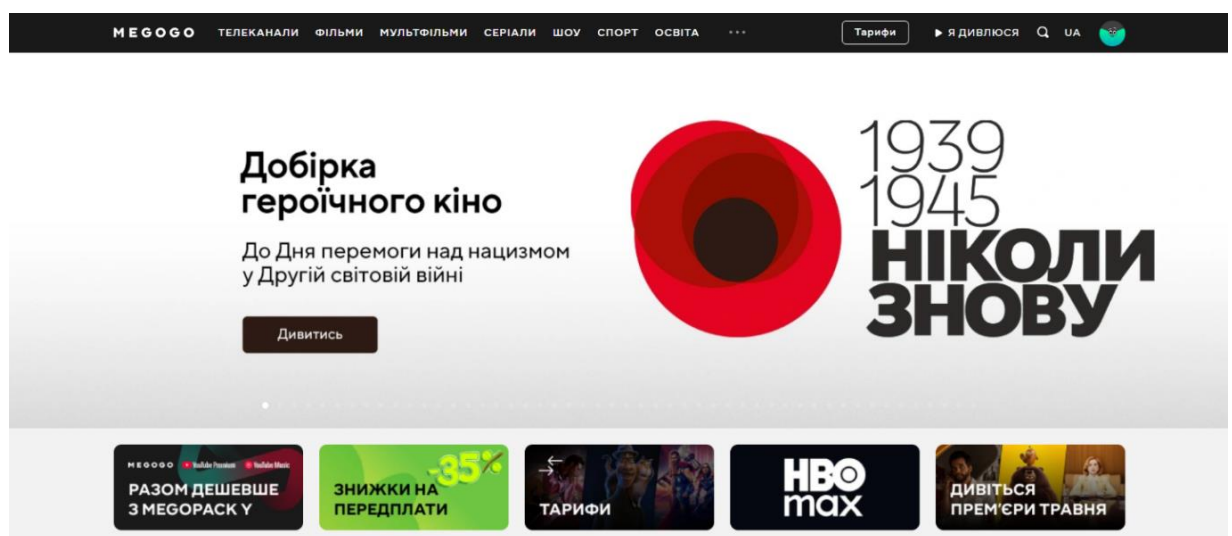


Рисунок 1.4 – Інтерфейс головної сторінки вебзастосунку Megogo

Платформа адаптована під локальний ринок і враховує мовні та регіональні особливості користувачів, пропонуючи як ліцензований зарубіжний контент, так і вітчизняне виробництво. Рекомендації формуються на основі історії переглядів та жанрових вподобань користувача.

Переваги мультимедійної платформи Megogo:

- адаптація до українського користувача та локального контенту;
- врахування мовних налаштувань та регіональної популярності;
- зручний інтерфейс та доступність різних типів контенту;
- поєднання телевізійного та потокового контенту на одній платформі.

Недоліки мультимедійної платформи Megogo:

- менший обсяг бібліотеки порівняно з глобальними платформами;
- обмежена кількість ексклюзивного контенту;
- менш розвинена система персоналізації порівняно з лідерами ринку;
- відсутність механізму збору початкових вподобань при реєстрації.

Taste.io є спеціалізованим застосунком для персоналізованих рекомендацій фільмів та серіалів (рис. 1.5) [7], розробленим компанією Taste Labs.

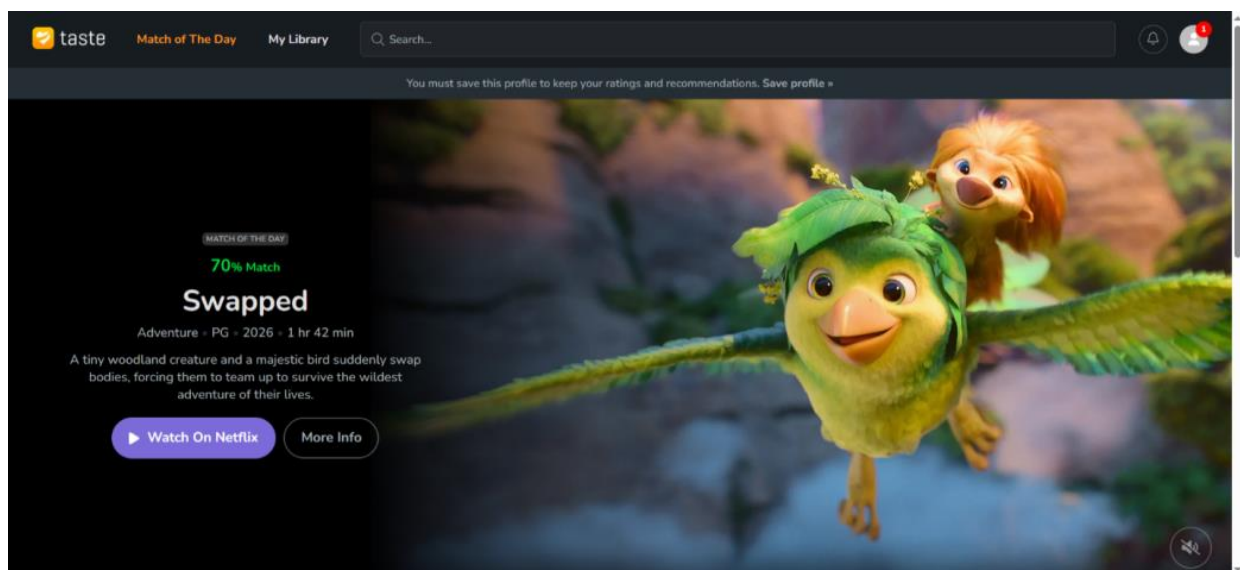


Рисунок 1.5 – Інтерфейс головної сторінки вебзастосунку Taste.io

Сервіс позиціонує себе як незалежну платформу без комерційної упередженості.

Механізм роботи базується на пошуку користувачів зі схожими вподобаннями – після оцінювання певної кількості фільмів система знаходить людей з подібним смаком і формує рекомендації на їх основі.

Переваги спеціалізованого застосунку Taste.io:

- отримання персональних рекомендацій після проходження опитувальника;
- простий та зрозумілий механізм взаємодії.

Недоліки спеціалізованого застосунку Taste.io:

- для отримання точних рекомендацій необхідно оцінити щонайменше 20 фільмів;
- менша бібліотека порівняно з великими платформами.

Результати проведеного аналізу занесено до таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика платформ, що формують рекомендації на основі користувацьких уподобань

Критерій	Netflix	IMDb	Letterboxd	Megogo	Taste.io
Опитувальник при реєстрації	Так	Ні	Ні	Ні	Так
Персоналізовані рекомендації	Так	Так	Ні	Так	Так
Робота без історії переглядів	Так	Ні	Ні	Ні	Так
Пояснення до рекомендацій	Так	Так	Ні	Ні	Так
Локалізація для України	Часткова	Часткова	Ні	Так	Так

Проведений аналіз показує, що більшість сучасних вебзастосунків для формування списку рекомендованих фільмів покладаються переважно на накопичену історію переглядів та оцінок користувача. При цьому проблема нового користувача, який ще не має достатньої кількості даних для побудови точного профілю, вирішується по-різному або не вирішується взагалі.

Тільки декілька з розглянутих платформ використовують розгорнутий структурований опитувальник як основний інструмент формування початкового профілю вподобань.

1.2 Аналіз методів побудови рекомендаційних систем та підходів до вирішення проблем персоналізації

Через розглянуту проблему інформаційного перевантаження виникла необхідність у створенні ефективних підходів, які дозволяють автоматизувати процес пошуку та підбору релевантного контенту для користувача. Вибір методу побудови таких сервісів безпосередньо впливає на точність рекомендацій, швидкість адаптації до нового користувача та загальну ефективність роботи вебзастосунку [8]. Виділяють кілька основних підходів до побудови рекомендаційних систем, кожен із яких має власний принцип роботи, сферу застосування та характерні обмеження.

Одним із найбільш поширених підходів є контентна фільтрація (Content-Based filtering) [8–11]. Її суть полягає в аналізі характеристик об'єктів, які користувач уже оцінив позитивно, та пошуку схожого контенту на основі цих ознак. У випадку рекомендації фільмів такими характеристиками виступають: жанр, режисер, акторський склад, рік випуску, країна виробництва, тематика та інші атрибути. Система формує числовий профіль користувача на основі векторних подань переглянутих фільмів і порівнює його з векторами фільмів-кандидатів за допомогою математичних мір подібності.

Перевагою такого підходу є незалежність від даних інших користувачів, що особливо важливо на початковому етапі роботи. Водночас надмірна орієнтація лише на схожий контент може звужувати коло рекомендацій і обмежувати можливість відкриття нових жанрів.

Іншим важливим методом є колаборативна фільтрація (Collaborative filtering), яка базується на аналізі поведінки користувачів, а не характеристик контенту [8–11]. Система визначає схожість між користувачами або між самими об'єктами на основі оцінок, переглядів та інших дій. Якщо користувачі зі схожими вподобаннями високо оцінили певний фільм, він може бути рекомендований іншим користувачам із подібним профілем.

Розрізняють два різновиди підходу:

- колаборативна фільтрація на основі користувачів (User-Based CF), який шукає користувачів зі схожими вподобаннями;
- колаборативна фільтрація на основі елементів (Item-Based CF), який знаходить об'єкти з подібними шаблонами оцінювання.

Такий метод дозволяє знаходити нетипові рекомендації, однак його ефективність значно залежить від кількості накопичених оцінок і знижується при недостатній активності користувачів.

Окремим підходом є фільтрація на основі знань (Knowledge-Based filtering) [9]. Вона не використовує історію оцінок, а спирається на явно сформульовані вимоги і обмеження, задані користувачем. Наприклад, якщо користувач шукає фільм певного жанру, тривалості та з конкретним рейтингом, система формує рекомендації відповідно до цих параметрів. Такий підхід є прозорим і зрозумілим для користувача, однак потребує ручного введення критеріїв при кожному пошуку, що знижує рівень автоматизації.

Ще одним допоміжним методом є демографічна фільтрація (Demographic filtering) [10], яка використовує загальні характеристики користувача, зокрема вік, стать, географічне розташування та мовні налаштування.

На основі вподобань певної групи людей система формує початкові рекомендації для нового користувача. Цей підхід не потребує активної взаємодії з платформою, проте його точність є нижчою, оскільки користувачі однієї демографічної групи можуть мати різні інтереси. Саме тому демографічна фільтрація використовується як допоміжний інструмент.

У практиці сучасних платформ найчастіше застосовуються гібридні рекомендаційні системи (Hybrid recommender systems), які поєднують кілька методів одночасно [8]. Їх основна мета полягає у компенсації недоліків окремих алгоритмів та підвищенні точності персоналізації. Наприклад, на початковому етапі може використовуватися контентна фільтрація, а після накопичення достатньої кількості оцінок – колаборативний підхід.

Також можливе зважене поєднання результатів кількох алгоритмів або каскадний підхід, коли один метод формує список кандидатів, а інший уточнює їх ранжування.

Незважаючи на різноманіття підходів, розроблення рекомендаційних систем пов'язана з низкою типових проблем:

- «Проблема холодного старту» (Cold start problem) є однією з найбільш поширених [8] – на момент реєстрації система не має жодних даних про вподобання людини, що унеможливорює застосування персоналізованих методів. Для її вирішення використовуються опитувальники вподобань, що дозволяють зібрати базову інформацію про інтереси користувача ще до накопичення активності, а також контентна або демографічна фільтрація як перехідні механізми;

- «Бульбашка фільтрів» (filter bubble) виникає переважно при використанні контентної фільтрації – система, орієнтована виключно на схожість з уже переглянутим контентом, поступово звужує діапазон рекомендацій до вузького кола тематик та жанрів [11]. Вирішенням є впровадження елементів різноманітності у список рекомендацій або поєднання з колаборативною фільтрацією, здатною виходити за межі очевидних вподобань.

1.3 Постановка задачі

Персоналізований підбір фільмів відповідно до індивідуальних вподобань користувача є актуальним завданням у контексті розвитку рекомендаційних систем та зростаючого обсягу доступного відеоконтенту. Тому ставиться завдання створити вебзастосунок, що дозволить формувати персоналізований список рекомендованих фільмів на основі оброблення опитувальника вподобань користувача із застосуванням методів контентної та колаборативної фільтрації.

Об'єктом роботи є процес розроблення вебзастосунку для формування списку рекомендованих фільмів на основі оброблення опитувальника вподобань користувача.

Метою роботи є розроблення вебзастосунку для формування списку рекомендованих фільмів шляхом оброблення вподобань користувача.

Для досягнення мети необхідно вирішити такі завдання:

- проаналізувати сучасні вебзастосунки формування списку рекомендованих фільмів та існуючі підходи до побудови рекомендаційних систем;
- розробити структуру опитувальника вподобань користувача, що забезпечує формування початкового профілю вподобань для вирішення проблеми «холодного старту»;
- спроектувати реляційну базу даних для зберігання інформації про фільми, користувачів, їх оцінки та результати рекомендацій;
- реалізувати метод контентної фільтрації для формування рекомендацій на основі характеристик контенту;
- реалізувати метод колаборативної фільтрації для формування рекомендацій на основі поведінки схожих користувачів;
- розробити клієнтську та серверну частини застосунку;
- протестувати працездатність системи та оцінити якість рекомендацій за допомогою відповідних метрик.

2 ПРОЄКТУВАННЯ МОДЕЛЕЙ ГІБРИДНОЇ СИСТЕМИ РЕКОМЕНДАЦІЇ ФІЛЬМІВ

2.1 Проєктування структури бази даних вебзастосунку

Розроблюваний вебзастосунок є комплексною системою що поєднує функціонал каталогу фільмів, платформи для ведення обліку переглянутого контенту та рекомендаційної системи. Для визначення функціональних вимог до бази даних насамперед необхідно розглянути взаємодію користувача з системою в цілому. Діаграма прецедентів що відображає основні сценарії використання вебзастосунку наведена на рисунку 2.1.

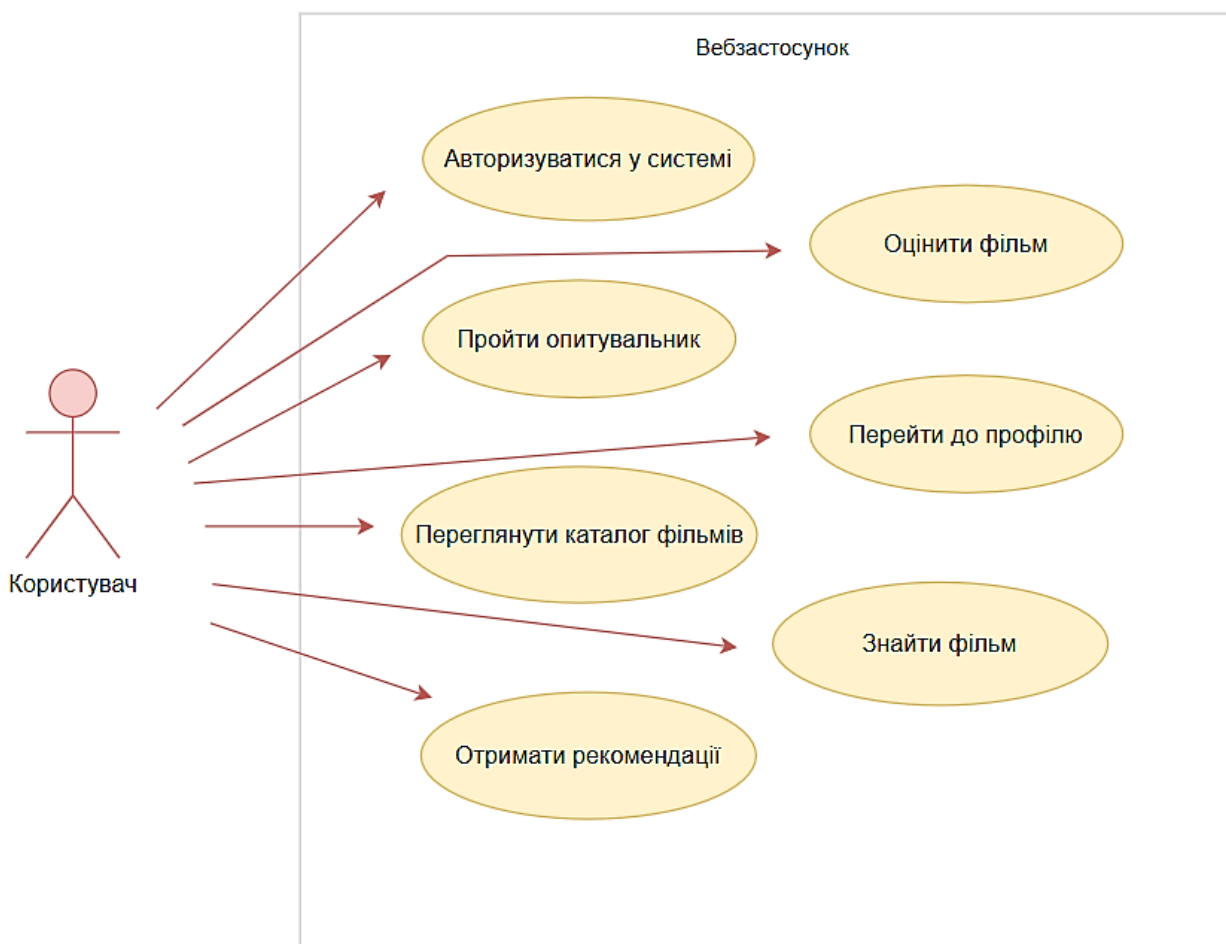


Рисунок 2.1 – Головна діаграма прецедентів вебзастосунку у нотації UML

Відповідно до діаграми прецедентів (рис. 2.1) користувач взаємодіє з системою через такі основні сценарії використання:

- авторизуватися у системі, користувач може пройти реєстрацію нового облікового запису або виконати вхід до вже існуючого профілю для отримання доступу до повного функціоналу вебзастосунку;
- пройти опитувальник уподобань, користувач надає інформацію про власні жанрові, мовні та тематичні вподобання для формування початкового профілю рекомендаційної системи;
- знайти фільм, користувач може здійснювати пошук та фільтрацію фільмів за назвою, жанром, роком випуску та рейтингом;
- оцінити фільм, користувач має можливість виставляти оцінки та залишати рецензії, які використовуються системою для уточнення персональних рекомендацій;
- перейти до профілю, користувач може переглядати власну історію оцінок та список перегляду;
- отримати рекомендації, система формує персоналізований список фільмів на основі опитувальника, історії оцінок та алгоритмів рекомендаційної системи.

Кожен із цих сценаріїв породжує певні вимоги до структури та складу даних що зберігаються у базі даних. Для забезпечення їх повноцінної роботи необхідна надійна та добре структурована база даних яка виступає єдиним централізованим сховищем усієї інформації що циркулює у системі. З точки зору функціональних потреб вебзастосунку база даних виконує кілька принципово різних завдань:

- зберігання інформації про фільми, включаючи назву, опис, жанри, акторський склад, режисерів, країни виробництва та тематичні теги, ці характеристики будуть необхідні для роботи рекомендаційної системи;
- забезпечення управління обліковими записами користувачів, зберігання даних для автентифікації та підтримка персоналізованої взаємодії з платформою;

- фіксація поведінки користувачів у системі, зокрема оцінок, рецензій, та списків перегляду, що використовуються для формування рекомендацій;
- підтримка механізму початкового опитувальника вподобань, формування базового профілю інтересів та зберігання результатів роботи рекомендаційних моделей.

Для наочного відображення послідовності взаємодії між компонентами системи при отриманні персоналізованих рекомендацій – наведено діаграму послідовності на рисунку 2.2.

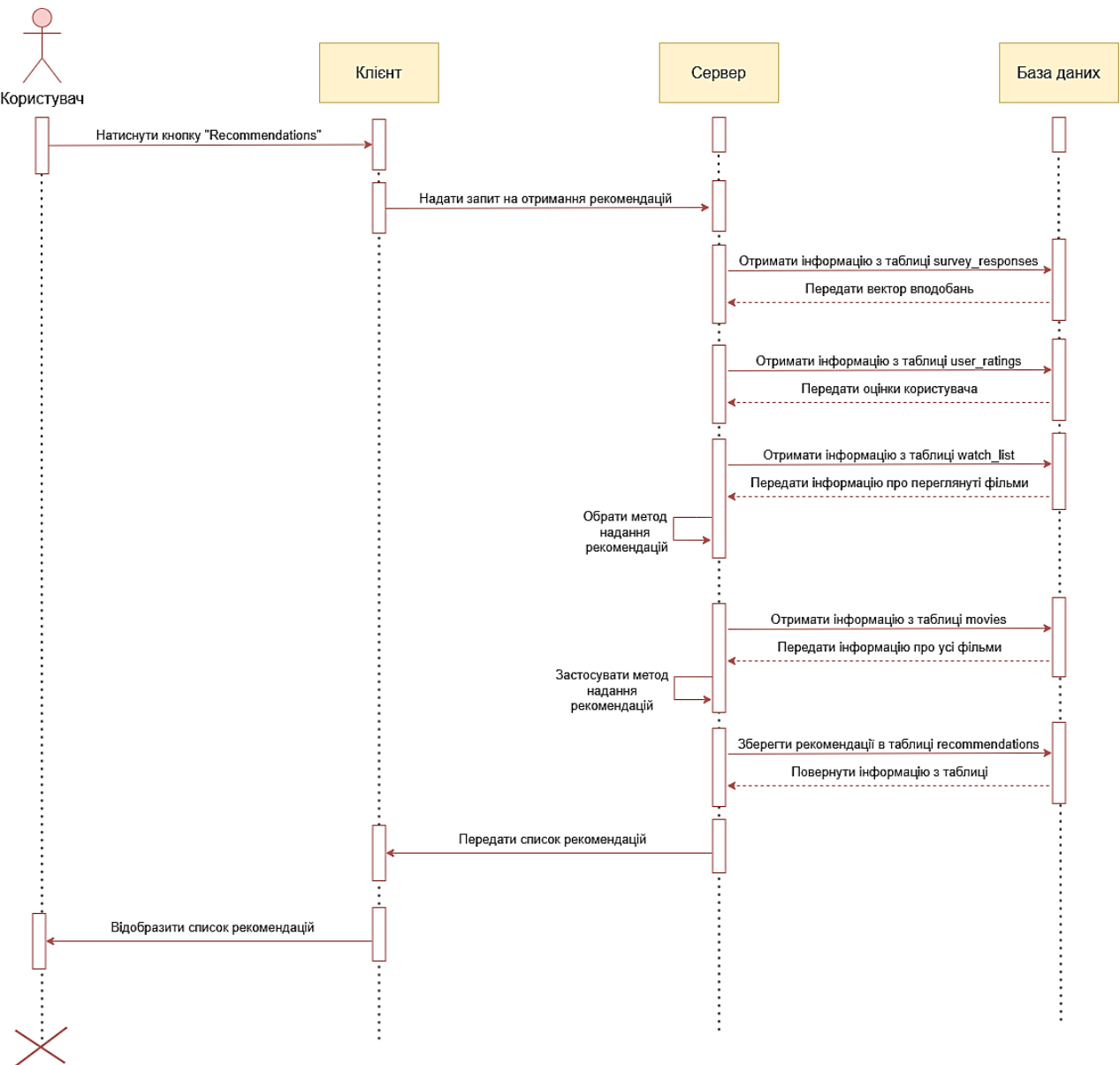


Рисунок 2.2 – Діаграма послідовності процесу «Отримати рекомендації»

Діаграма послідовності (рис. 2.2) відображає взаємодію між чотирма основними учасниками процесу: користувачем, клієнтською частиною застосунку, серверною частиною та базою даних. Користувач ініціює запит на отримання рекомендацій через інтерфейс. Клієнтська частина передає запит до серверного API. Серверна частина звертається до бази даних для отримання профілю користувача з таблиці `survey_responses`, його оцінок з таблиці `user_ratings` та переліку вже переглянутих фільмів з таблиці `watch_list`. На основі отриманих даних, залежно від кількості накопичених оцінок, обирається відповідна модель рекомендації, а саме контентна або колаборативна фільтрація. Програма звертається до бази даних для отримання векторів ознак фільмів-кандидатів та виконує обчислення схожості. Результати зберігаються у таблиці `recommendations` та повертаються клієнтській частині для відображення користувачеві.

З урахуванням перелічених функціональних вимог та визначених сценаріїв використання для вебзастосунку обрано реляційну модель бази даних з використанням системи управління PostgreSQL. Реляційний підхід забезпечує структурованість даних та підтримку зв'язків між сутностями через механізм зовнішніх ключів. Дані рекомендаційної системи мають численні зв'язки, зокрема фільми пов'язані з жанрами, акторами та режисерами, а користувачі пов'язані з оцінками та рекомендаціями, що природно представляється у реляційній моделі. Крім того реляційна база даних підтримує складні аналітичні запити з об'єднанням кількох таблиць що є необхідною умовою для роботи рекомендаційних алгоритмів.

Спроектвана база даних складається із сімнадцяти таблиць. Для відображення структури бази даних, сутностей та зв'язків між ними використано ER-діаграму в нотації Crow's Foot, яку наведено на рисунку 2.3.

Центральною таблицею системи є таблиця `movies` що містить основну інформацію про кінострічки, а саме назву, опис, рік випуску, тривалість, рейтинг, індекс популярності та мову оригіналу.

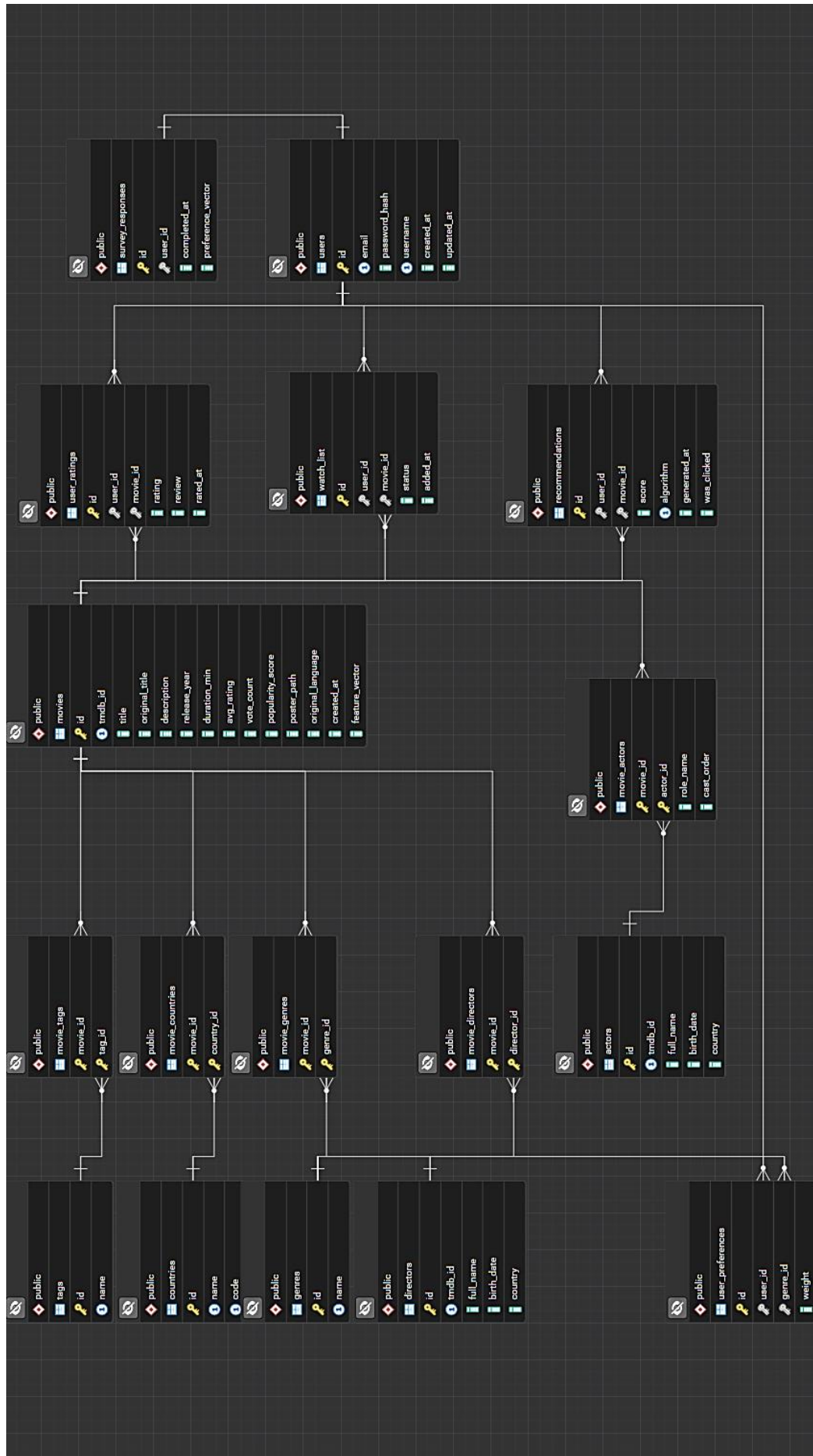


Рисунок 2.3 – ER-діаграма бази даних вебзастосунку в нотатції Crow's Foot

Передбачено поле `feature_vector` типу JSONB для зберігання числового вектору ознак фільму що є ключовим елементом контентної фільтрації.

Характеристики фільмів що можуть належати до певних категорій, зокрема жанри, країни виробництва, теги, актори та режисери, винесено у окремі таблиці `genres`, `countries`, `tags`, `actors` та `directors` відповідно, що усуває надлишковість даних. Зв'язки типу «багато-до-багатьох» між фільмами та цими сутностями реалізовано через п'ять проміжних таблиць, а саме `movie_genres`, `movie_directors`, `movie_actors`, `movie_countries` та `movie_tags`, кожна з яких має складений первинний ключ з відповідних зовнішніх ключів.

Взаємодія користувачів з контентом фіксується через три окремі таблиці що відповідають різним типам поведінкових даних. Таблиця `user_ratings` зберігає оцінки фільмів що є основним джерелом для алгоритму колаборативної фільтрації. Таблиця `watch_list` реалізує список перегляду з трьома статусами та додатково слугує джерелом даних для виключення вже переглянутих фільмів зі списку кандидатів при генерації рекомендацій. Таблиця `user_preferences` зберігає жанрові ваги користувача у числовому вигляді для швидкого доступу.

Для підтримки механізму опитувальника передбачено таблицю `survey_responses` що зберігає сформований вектор вподобань у форматі JSONB. Результати роботи рекомендаційних алгоритмів зберігаються у таблиці `recommendations` де для кожної рекомендації фіксується `score` релевантності, назва алгоритму та факт переходу користувача на сторінку фільму.

2.2 Модель формування профілю користувача на основі опитувальника вподобань

Одним із ключових викликів при розробці персоналізованих рекомендаційних систем є проблема «холодного старту», яка виникає у

момент реєстрації нового користувача. На цьому етапі система не має жодних даних про його вподобання, що унеможливорює застосування персоналізованих методів фільтрації та суттєво знижує якість перших рекомендацій [8]. Для вирішення цієї проблеми у розроблюваному вебзастосунку запроваджено структурований опитувальник вподобань, який дозволяє сформувати початковий профіль користувача ще до накопичення будь-якої активності на платформі.

Профіль користувача представлений у вигляді числового вектору ознак що відповідає структурі вектору фільму. Така відповідність є принциповою вимогою для забезпечення коректного математичного порівняння профілю користувача з характеристиками фільмів-кандидатів. Вектор профілю зберігається у полі `preference_vector` таблиці `survey_responses` у форматі JSONB та оновлюється динамічно у міру накопичення нових оцінок.

Модель формування профілю базується на двох джерелах даних, що використовуються з різними ваговими коефіцієнтами. Першим є результати опитувальника, що формують базовий вектор із коефіцієнтом 0,4. Другим є оцінки фільмів, виставлені користувачем у процесі використання платформи, які уточнюють профіль із коефіцієнтом 0,6. Модель формування профілю користувача описується таким виразом:

$$\overrightarrow{P_{user}} = 0,4 \cdot \vec{A} + 0,6 \cdot \vec{B}, \quad (2.1)$$

де $\overrightarrow{P_{user}}$ – підсумковий вектор профілю користувача;

$\vec{A}$ – вектор, сформований на основі результатів опитувальника;

$\vec{B}$ – вектор, сформований на основі оцінок та поведінкової активності користувача.

Розподіл вагових коефіцієнтів обґрунтовується принциповою різницею між природою цих двох джерел. Відповіді опитувальника не повністю відражають реальні вподобання користувача, він висловлює вподобання абстрактно без прив'язки до конкретного досвіду перегляду.

Дослідження у галузі поведінкової психології свідчать про те що люди часто неточно оцінюють власні вподобання в абстрактному контексті [12]. Тому оцінки переглянутих фільмів точніше відображають реальні вподобання і традиційно вважаються більш надійним джерелом для побудови профілю [13]. На початковому етапі коли оцінок ще немає профіль повністю визначається опитувальником. Після появи перших оцінок вплив анкети поступово зменшується у міру накопичення активності.

Опитувальник побудований як послідовний збір інформації про вподобання користувача, де кожен етап відповідає окремій групі характеристик профілю. Основу складають жанрові вподобання – обрані користувачем жанри отримують вищі значення у векторі профілю, що безпосередньо впливає на ранжування рекомендованих фільмів.

Додатково опитувальник враховує мовні та часові характеристики контенту. На основі відповідей користувача система визначає переваги щодо мови фільмів, періоду їх випуску та окремих тематичних особливостей. Відповідні параметри також відображаються у векторі профілю та використовуються при подальшому порівнянні з характеристиками фільмів.

Результатом проходження опитувальника є сформований вектор вподобань, що використовується як початкова основа для алгоритму контентної фільтрації та дозволяє системі формувати персоналізовані рекомендації вже після завершення опитування.

2.3 Модель побудови векторного подання фільмів

Контентна фільтрація як метод формування рекомендацій ґрунтується на математичному порівнянні характеристик фільмів з профілем користувача. Для забезпечення такого порівняння кожен фільм у системі повинен бути представлений у вигляді числового вектору ознак що відображає його змістовні характеристики у формі придатній для математичних обчислень.

Побудова такого векторного подання є підготовчим етапом що передуює безпосередньому застосуванню алгоритму контентної фільтрації та виконується одноразово після початкового наповнення бази даних фільмами.

Векторне подання кожного фільму формується на основі даних що зберігаються у реляційній базі даних вебзастосунку. Основним джерелом інформації є поля таблиці `movies` та пов'язані з нею довідникові таблиці. Сформований вектор зберігається безпосередньо у полі `feature_vector` таблиці `movies` у форматі JSONB що дозволяє уникнути повторного обчислення при кожному запиті та забезпечує швидкий доступ до векторного подання під час генерації рекомендацій.

Формування вектору починається з жанрових характеристик фільму оскільки саме жанрова приналежність є найбільш інформативною ознакою для визначення змістовної схожості між фільмами. Оскільки між фільмами та жанрами у базі даних реалізовано зв'язок типу «багато-до-багатьох» через таблицю `movie_genres` один фільм може належати до кількох жанрів одночасно. Для кожного жанру у векторі формується окрема числова характеристика. У випадку належності фільму до відповідного жанру ознака отримує значення 1, в іншому випадку значення встановлюється як 0. Такий спосіб кодування відомий як бінарне кодування забезпечує однозначне і сумірне представлення жанрової приналежності у числовій формі.

Поряд із жанровою приналежністю вектор містить середній рейтинг із поля `avg_rating` та індекс популярності із поля `popularity_score` таблиці `movies`. На відміну від бінарних жанрових ознак ці характеристики є неперервними числами що можуть суттєво відрізнятись за величиною між різними фільмами. Їх безпосереднє використання у векторі призвело б до домінування числово більших значень над рештою ознак та спотворення результатів порівняння. Тому обидві характеристики нормалізуються відносно максимального значення відповідного показника серед усіх фільмів бази даних що приводить їх до діапазону від 0 до 1 і забезпечує сумірність із бінарними ознаками.

На основі поля `release_year` таблиці `movies` визначається десятиріччя випуску кінострічки. Для кожного десятиліття що охоплює основний масив фільмів у базі даних формується окрема ознака що набуває значення 1 або 0 залежно від належності фільму до цього періоду. Мовна характеристика фільму, що формується на основі поля `original_language` таблиці `movies`, також кодується у бінарний спосіб для кожної з мов що найбільш широко представлені у бібліотеці системи. Фільм знятий певною мовою отримує відповідне значення 1 для цієї мовної ознаки та 0 для решти.

Описана процедура векторизації виконується для всіх фільмів бази даних у рамках одноразового процесу ініціалізації системи. Отримані вектори зберігаються у полі `feature_vector` таблиці `movies` та залишаються незмінними до оновлення характеристик відповідного фільму.

2.4 Модель рекомендації на основі поєднання контентної та колаборативної фільтрації

Гібридна рекомендаційна система розроблюваного вебзастосунку поєднує два принципово різні підходи до формування персоналізованих рекомендацій – контентну та колаборативну фільтрацію. Вибір між ними здійснюється автоматично залежно від кількості оцінок накопичених користувачем на платформі.

На початковому етапі коли поведінкових даних недостатньо для застосування колаборативного підходу система використовує контентну фільтрацію що спирається на профіль сформований через опитувальник.

У міру накопичення активності стає доступною колаборативна фільтрація що дозволяє виявляти нетривіальні зв'язки між вподобаннями користувачів.

2.4.1 Опис методу побудови контентної фільтрації

Контентна фільтрація є основним методом формування рекомендацій на початковому етапі роботи системи та залишається доступною протягом усього часу використання платформи. Принцип її роботи полягає у визначенні ступеня змістовної схожості між числовим профілем користувача та векторними поданнями фільмів-кандидатів. Фільми що мають найвищий ступінь схожості з профілем формують список рекомендацій.

Вхідними даними для алгоритму є нормалізований вектор профілю користувача та нормалізовані вектори ознак фільмів. Перед початком обчислень із множини фільмів-кандидатів виключаються ті що користувач вже переглянув або оцінив, це забезпечує що система не буде пропонувати вже знайомий контент. Міра схожості між профілем користувача та кожним фільмом-кандидатом обчислюється за допомогою косинусної подібності. Косинусна подібність вимірює кут між двома векторами у багатовимірному просторі ознак і формально описується таким виразом:

$$\text{sim}(A, B) = \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n A_i^2 \cdot \sum_{i=1}^n B_i^2}}, \quad (2.2)$$

де A та B – вектори ознак двох об'єктів.

Результат обчислення є числом у діапазоні від 0 до 1 де значення 1 відповідає повному збігу напрямків векторів у просторі ознак а значення 0 свідчить про відсутність спільних характеристик.

На рисунку 2.4 наведено приклад геометричної інтерпретації косинусної подібності. Вектор A відповідає профілю користувача, вектор B – векторному поданню фільму-кандидата. Значення міри схожості визначається косинусом кута θ між двома векторами: чим менший кут, тим вища схожість і тим вірогідніше що фільм буде включений до списку рекомендацій.

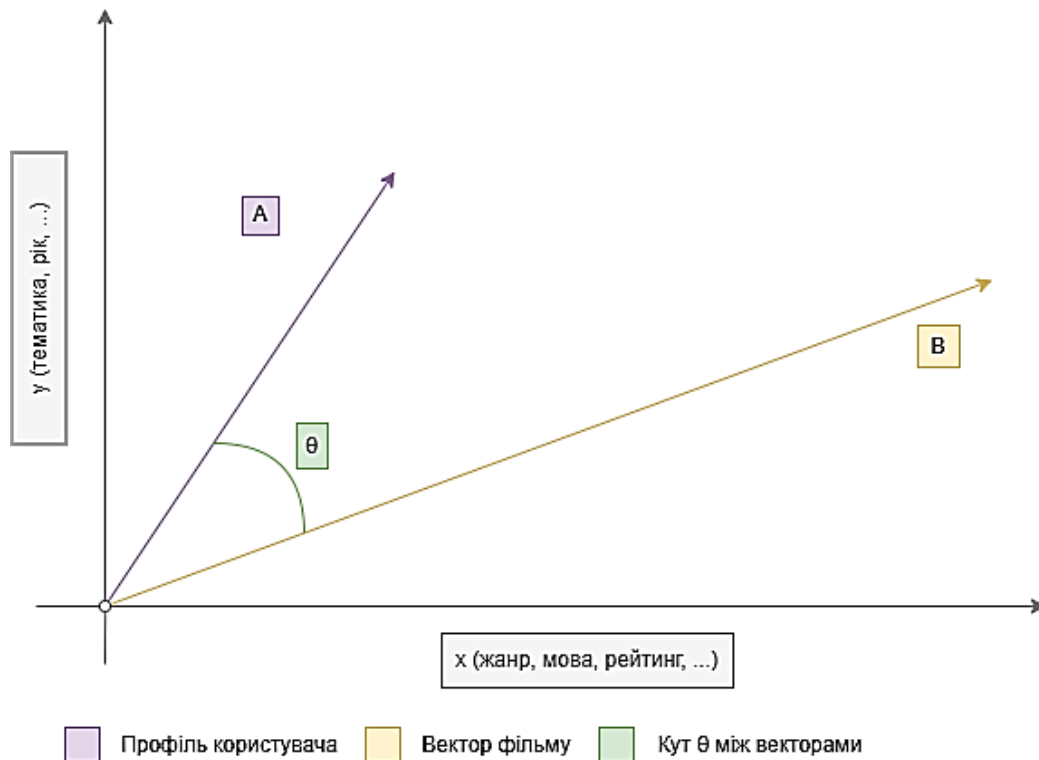


Рисунок 2.4 – Геометрична інтерпретація косинусної подібності між вектором профілю користувача та вектором ознак фільму у просторі характеристик

Після обчислення косинусної подібності між профілем користувача та кожним фільмом-кандидатом отримані значення використовуються для ранжування результатів. Фільми впорядковуються за спаданням значення подібності і до списку рекомендацій включаються ті що отримали найвищі показники, потім він зберігається у таблиці recommendations бази даних із зазначенням числового значення подібності у полі score та ідентифікатора алгоритму у полі algorithm.

2.4.2 Опис методу побудови колаборативної фільтрації

Колаборативна фільтрація є другим методом формування рекомендацій у розроблюваній системі та стає доступною після накопичення користувачем достатньої кількості оцінок фільмів.

На відміну від контентної фільтрації, що базується на характеристиках самого контенту, колаборативний підхід аналізує поведінкові дані користувачів та виявляє схожість між ними на основі виставлених оцінок.

Фільми, які отримали високі оцінки від користувачів зі схожими вподобаннями, рекомендуються поточному користувачеві навіть у випадках, коли вони не відповідають його явно вираженим жанровим перевагам. Саме здатність виявляти приховані закономірності у поведінці користувачів є ключовою перевагою колаборативного підходу.

У розроблюваній системі реалізовано підхід колаборативної фільтрації на основі користувачів (User-Based CF). Його сутність полягає у знаходженні користувачів із найбільш схожими профілями оцінок та формуванні рекомендацій на основі їхніх уподобань. Для визначення ступеня схожості між користувачами використовується коефіцієнт кореляції Пірсона. Його використання дозволяє коректно порівнювати користувачів, які можуть систематично ставити різні за абсолютним значенням оцінки, але демонструють подібні відносні вподобання.

Вхідними даними для алгоритму є матриця оцінок, сформована на основі даних таблиці `user_ratings`. Матриця має вигляд, у якому рядки відповідають користувачам, а стовпці – фільмам. Кожен елемент містить оцінку, виставлену користувачем відповідному фільму, або є порожнім у разі відсутності оцінки. Формально така матриця позначається як:

$$R = \{r_{ui}\}, \quad (2.3)$$

де r_{ui} – оцінка користувача u для об'єкта i .

Для обчислення схожості між поточним користувачем та іншими користувачами системи застосовується коефіцієнт кореляції Пірсона, який розраховується лише за тими фільмами, що були оцінені обома користувачами. Формально коефіцієнт кореляції Пірсона визначається таким чином:

$$sim(u, v) = \frac{\sum_i (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_i (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_i (r_{vi} - \bar{r}_v)^2}}, \quad (2.4)$$

де $\bar{r}_u$ та $\bar{r}_v$ – середні оцінки користувачів u та v відповідно.

Отримане значення коефіцієнта належить діапазону від -1 до 1. Значення, близьке до 1, свідчить про високу позитивну кореляцію між користувачами та подібність їхніх вподобань. Значення, близьке до -1, вказує на протилежні вподобання, тоді як значення, близьке до 0, означає відсутність суттєвого статистичного зв'язку між оцінками. До подальших обчислень допускаються лише ті користувачі, для яких значення кореляції з поточним користувачем перевищує заданий пороговий рівень, що дозволяє зменшити вплив випадкових збігів. Додатково враховується мінімальна кількість спільно оцінених фільмів, що забезпечує статистичну надійність отриманих результатів.

Після обчислення схожості формується множина найбільш подібних користувачів, які виступають як «сусіди» поточного користувача. На основі їхніх оцінок виконується прогнозування оцінки для кожного фільму, який ще не був переглянутий користувачем. Передбачувана оцінка обчислюється як зважене середнє оцінок сусідів:

$$pred(u, i) = \frac{\sum r(u, v) \cdot r_{vi}}{\sum |r(u, v)|}, \quad (2.5)$$

де $pred(u, i)$ – прогнозована оцінка користувача u для фільму i ;

$r(u, v)$ – коефіцієнт кореляції між користувачами u та v ;

r_{vi} – коефіцієнт кореляції між користувачами u та v .

Прогнозовані значення використовуються для ранжування фільмів-кандидатів. Фільми впорядковуються за спаданням передбачуваної оцінки, і найбільш релевантні з них включаються до списку рекомендацій. Сформовані результати зберігаються у таблиці recommendations.

2.5 Модель оцінювання якості рекомендацій

Метрики оцінювання є невід'ємною складовою процесу проєктування та розроблення рекомендаційних систем, оскільки вони дозволяють об'єктивно визначити ефективність роботи алгоритмів та якість сформованих рекомендацій. Застосування кількісних показників надає можливість оцінити точність прогнозування, визначити оптимальні параметри моделей і встановити доцільність використання певних методів у межах системи.

У науковій літературі традиційно виділяють два основні підходи до оцінювання рекомендаційних систем:

- офлайн-оцінювання, що використовується для тестування алгоритмів у контрольованих умовах;
- онлайн-оцінювання, що застосовується для аналізу ефективності системи під час реальної взаємодії з користувачами.

Офлайн-оцінювання базується на аналізі даних без безпосередньої участі реальних користувачів у процесі тестування. Такий підхід дозволяє швидко перевіряти ефективність алгоритмів, експериментувати з параметрами моделей та порівнювати різні методи між собою в контрольованих умовах. Онлайн-оцінювання, у свою чергу, ґрунтується на аналізі поведінки користувачів під час реальної взаємодії із системою та дозволяє оцінити практичну корисність сформованих рекомендацій у реальному середовищі використання [14].

У межах розроблюваного вебзастосунку реалізовано обидва підходи до оцінювання, це допомагає отримати більш повну та об'єктивну характеристику ефективності роботи системи. При цьому для кожного методу формування рекомендацій обирається метрика, що найбільш повно відображає специфіку отриманих результатів. Оскільки колаборативна фільтрація здійснює прогнозування числової оцінки для фільмів, які користувач ще не оцінював, результатом його роботи є конкретне числове значення, що може бути порівняне з реальною оцінкою користувача.

Натомість контентна фільтрація не виконує прямого прогнозування оцінок, а формує впорядкований список рекомендацій на основі змістовної схожості між характеристиками фільмів та профілем користувача. Таким чином, різниця у природі вихідних даних моделей обумовлює необхідність використання різних підходів до оцінювання їх ефективності.

Для офлайн-оцінювання алгоритмів у системі використовується підхід відкладеної вибірки (hold-out підхід). Сутність даного підходу полягає у поділі наявних даних на дві частини:

- навчальна вибірка, що використовується для побудови моделі;
- тестова вибірка, використовується для перевірки якості рекомендацій.

Для кожного користувача частина вже оцінених фільмів включається до тестової вибірки та тимчасово не використовується під час формування рекомендацій, тоді як решта даних формує навчальну вибірку. Після формування рекомендацій на основі навчальної вибірки отримані результати порівнюються з оцінками з тестової вибірки. Використання hold-out підходу дозволяє моделювати реальні умови роботи рекомендаційної системи, за яких алгоритм повинен рекомендувати фільми, щодо яких відсутня попередня інформація про оцінки користувача [15].

Для оцінювання алгоритму колаборативної фільтрації використовується метрика середньоквадратичної похибки (*RMSE*). Доцільність використання цієї метрики пояснюється тим, що алгоритм формує числові передбачення, які можна безпосередньо порівняти з фактичними оцінками користувачів. Саме значення *RMSE* обчислюється за формулою:

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{r}_i - r_i)^2}{n}}, \quad (2.6)$$

де $\hat{r}_i$ – оцінка, передбачена алгоритмом для i -го фільму;

r_i – реальна оцінка користувача;

n – кількість прогнозованих значень, що беруть участь у порівнянні.

Отримане значення характеризує середню величину похибки прогнозування у межах використовуваної шкали оцінювання. У розроблюваній системі шкала оцінок становить від 1 до 10 балів, наприклад, отримане значення $RMSE = 1$ буде означати, що середня помилка алгоритму складає приблизно 1 бал. Однією з ключових особливостей даної метрики є її висока чутливість до значних помилок прогнозування.

Для оцінювання роботи моделі контентної фільтрації використовується метрика – влучність на K ($Precision@K$), оскільки даний алгоритм працює не з числовим прогнозуванням оцінок, а з ранжуванням об'єктів за ступенем їх релевантності для користувача [8]. Дана метрика визначає частку релевантних елементів серед перших K рекомендованих об'єктів та формально задається виразом:

$$Precision@K = \frac{\text{Кількість релевантних результатів серед перших } K}{K}, \quad (2.7)$$

де K – кількість елементів, яка розглядається.

У розроблюваній системі визначено що фільм може вважатися релевантним у випадку, якщо його оцінка користувачем становить не менше 8 балів, що відповідає позитивному сприйняттю контенту. Для оцінювання використовується hold-out підхід, за якого частина позитивно оцінених фільмів приховується та використовується для перевірки якості сформованих рекомендацій. Значення параметра K у системі відповідає кількості рекомендацій, які одночасно відображаються користувачу в інтерфейсі вебзастосунку.

Наприклад, якщо отримано значення $Precision@10 = 0,2$ то це буде означати, що в середньому 2 із 10 рекомендованих фільмів є релевантними для користувача, що буде характеризувати високий рівень якості рекомендацій. Значення нижче 0,1 будуть говорити про невелику ефективність алгоритму та недостатню якість побудови профілю користувача або векторного подання фільмів.

Окрім офлайн-оцінювання, у системі реалізовано механізм онлайн-аналізу ефективності рекомендацій, який базується на відстеженні поведінки користувачів під час взаємодії із вебзастосунком. Для оцінювання використовується метрика CTR (Click-Through Rate), що характеризує частку рекомендацій, за якими користувач здійснив перехід. З цією метою у таблиці recommendations передбачено спеціальне поле was_clicked, яке набуває значення true у випадку переходу користувача до сторінки детального перегляду рекомендованого фільму. Даний показник дозволяє оцінити рівень зацікавленості користувача сформованими рекомендаціями та може використовуватись для подальшого аналізу ефективності системи, дослідження поведінкових закономірностей і вдосконалення рекомендаційних моделей у майбутньому [14].

3 РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ ФОРМУВАННЯ СПИСКУ РЕКОМЕНДОВАНИХ ФІЛЬМІВ НА ОСНОВІ ОБРОБЛЕННЯ ОПИТУВАЛЬНИКА ВПОДОБАНЬ КОРИСТУВАЧА

3.1 Вибір інструментальних засобів реалізації вебзастосунку формування списку рекомендованих фільмів на основі оброблення опитувальника вподобань користувача

Вибір інструментів для розробки програмної реалізації створеної моделі здійснювався окремо для кожної структурної частини вебзастосунку.

Для реалізації серверної частини було обрано платформу .NET з використанням підходу Minimal API. Платформа .NET забезпечує високу продуктивність виконання коду та широкі можливості для побудови вебзастосунків. Підхід Minimal API на відміну від класичного Model-View-Controller (MVC) дозволяє описувати кінцеві точки (endpoints) безпосередньо без надлишкової інфраструктури, що облегшує створення невеликої системи.

Мова програмування C# є однією з провідних мов сучасної розробки програмного забезпечення та широко застосовується для створення вебзастосунків, корпоративних інформаційних систем і хмарних сервісів.

Для організації доступу до даних використовується реляційна система керування базами даних PostgreSQL. Реляційна модель є оптимальним вибором для рекомендаційної системи завдяки необхідності виконання складних аналітичних запитів з об'єднанням таблиць фільмів, оцінок і жанрів. PostgreSQL має тип даних JSONB, що використовується для зберігання числових векторів ознак фільмів і векторів вподобань користувачів.

Взаємодія серверної частини з базою даних здійснюється за допомогою SQL-запитів із використанням мікро-ORM (Object-Relational Mapping) бібліотеки Dapper для відображення результатів запитів на об'єкти мови C#.

Автентифікація користувачів реалізована на основі технології JSON Web Token (JWT), а для захисту паролів використовується бібліотека хешування BCrypt.

Для клієнтської частини обрано бібліотеку React у поєднанні з мовою TypeScript та збірником Vite. React є однією з найпоширеніших технологій для розроблення односторінкових вебзастосунків, а використання TypeScript забезпечує статичну типізацію, що дозволяє виявляти помилки на етапі компіляції. Vite забезпечує швидкий запуск проєкту.

3.2 Етапи програмної реалізації формування списку рекомендованих фільмів на основі оброблення опитувальника вподобань користувача

Реалізація створеного вебзастосунку охоплює кілька взаємопов'язаних підсистем що забезпечують повний цикл роботи системи надання рекомендацій.

На рисунку 3.1 наведено структуру серверної частини у середовищі розробки Microsoft Visual Studio 2022.

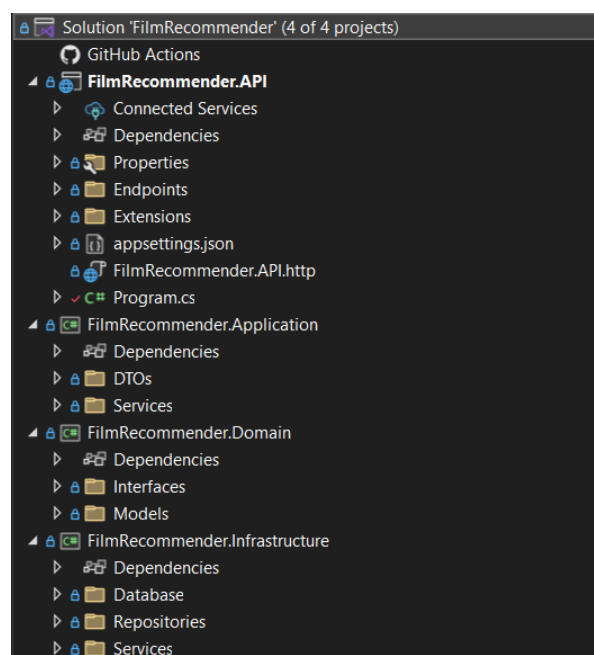


Рисунок 3.1 – Структура рішення у Microsoft Visual Studio

Реалізуються такі групи кінцевих точок (endpoints):

- автентифікація (/api/auth);
- управління фільмами (/api/movies);
- жанри (/api/genres);
- список перегляду (/api/watchlist);
- опитувальник (/api/survey);
- рекомендації (/api/recommendations);
- оцінювання якості (/api/evaluation).

Документація всіх endpoints автоматично генерується бібліотекою Swashbuckle та доступна через інтерфейс Swagger UI за адресою /swagger (рис. 3.2).

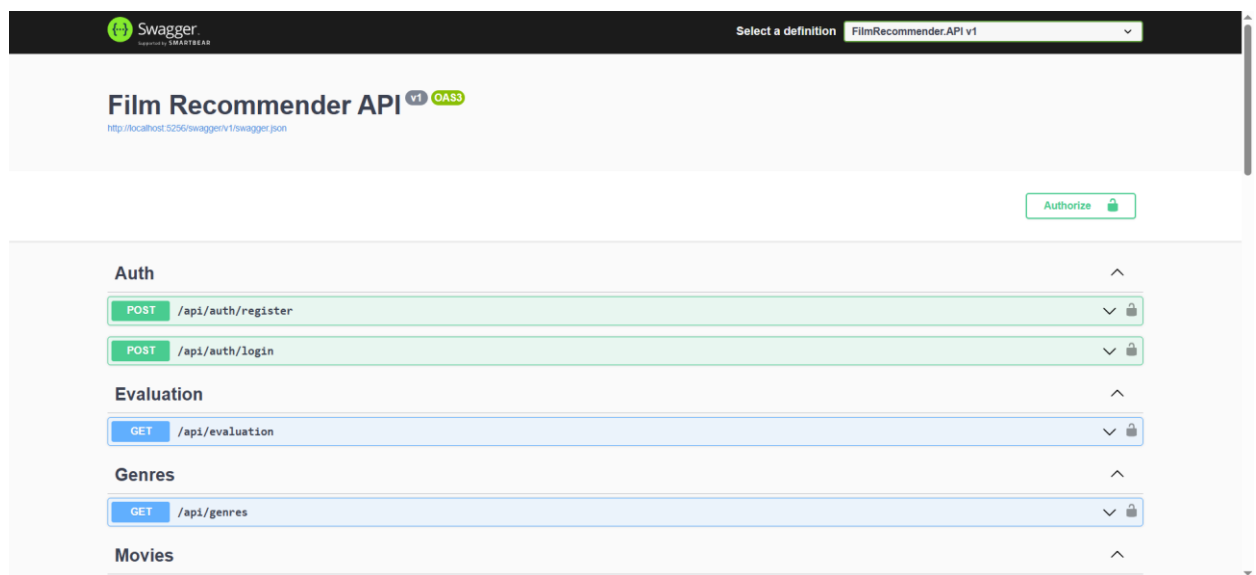


Рисунок 3.2 – Інтерфейс Swagger UI із переліком кінцевих точок (endpoints)

Наповнення бази даних здійснюється у два послідовні етапи. Під час першого етапу використовується набір даних MovieLens 100K, який містить 100000 оцінок, 943 користувачів та 1682 фільмів. Цей набір даних слугує основним джерелом інформації про взаємодії користувачів із фільмами.

На другому етапі виконується збагачення даних шляхом інтеграції зовнішнього API The Movie Database (TMDb) (рис. 3.3).

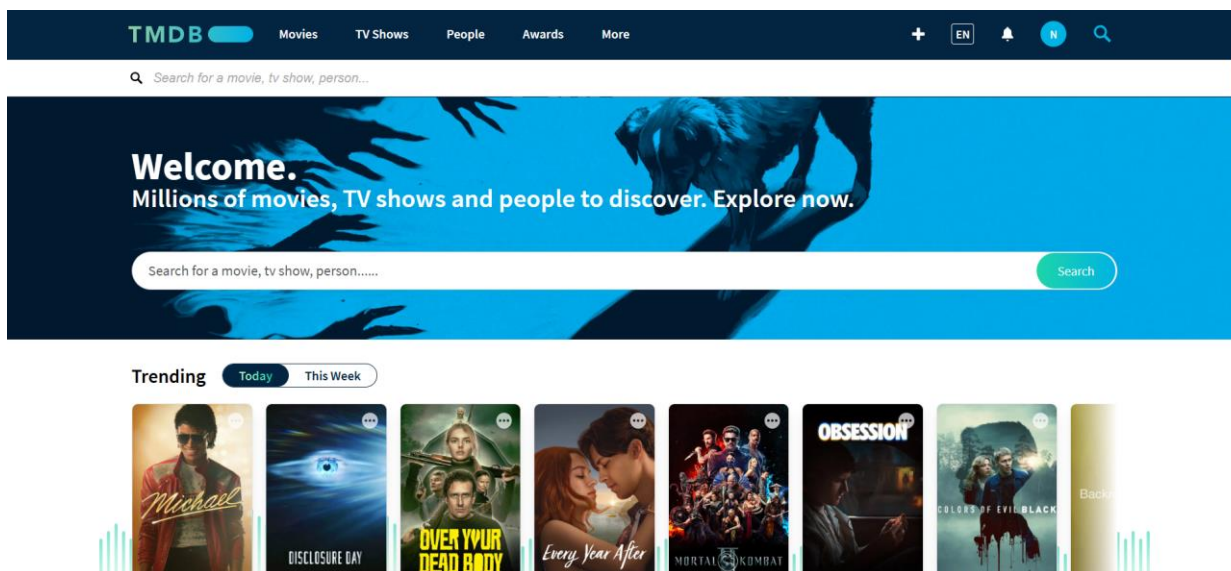


Рисунок 3.3 – Інтерфейс головної сторінки вебзастосунку TMDb

Набір даних MovieLens 100K містить тільки інформацію про оцінки користувачів, назви та жанри фільмів, але не містить деталей, необхідних для реалізації створеної моделі. Тому для кожного фільму з набору MovieLens виконується пошук відповідного запису у TMDb за назвою та роком випуску. У разі успішного знаходження відповідності до бази даних завантажуються та зберігаються додаткові атрибути:

- опис сюжету фільму;
- посилання на зображення;
- дата виходу у прокаті;
- мова оригіналу;
- рейтинг та популярність фільму;
- список жанрів;
- інформація про режисера;
- інформація про акторський склад.

Отримані дані використовуються для відображення детальної інформації про фільми в інтерфейсі вебзастосунку та для побудови векторів ознак у моделі контентної фільтрації. Саме на основі жанрів, мови оригіналу, року випуску, популярності та рейтингу формується набір ознак, який використовується під час обчислення схожості між профілем користувача та фільмами.

Якщо запис із набору даних MovieLens не знайдено у TMDb, фільм зберігається лише з базовими даними MovieLens 100K, задля збереження цілісності набору даних та уникання втрат інформації про оцінки користувачів.

Реєстрація та автентифікація користувачів реалізована через клас AuthService. При реєстрації пароль хешується за допомогою функцій бібліотеки BCrypt. Для підтвердження особи користувача використовується механізм JWT (JSON Web Token), що дозволяє безпечно передавати інформацію про автентифікованого користувача між клієнтом та сервером.

На рисунку 3.4 наведено приклад успішної автентифікації через Swagger UI.

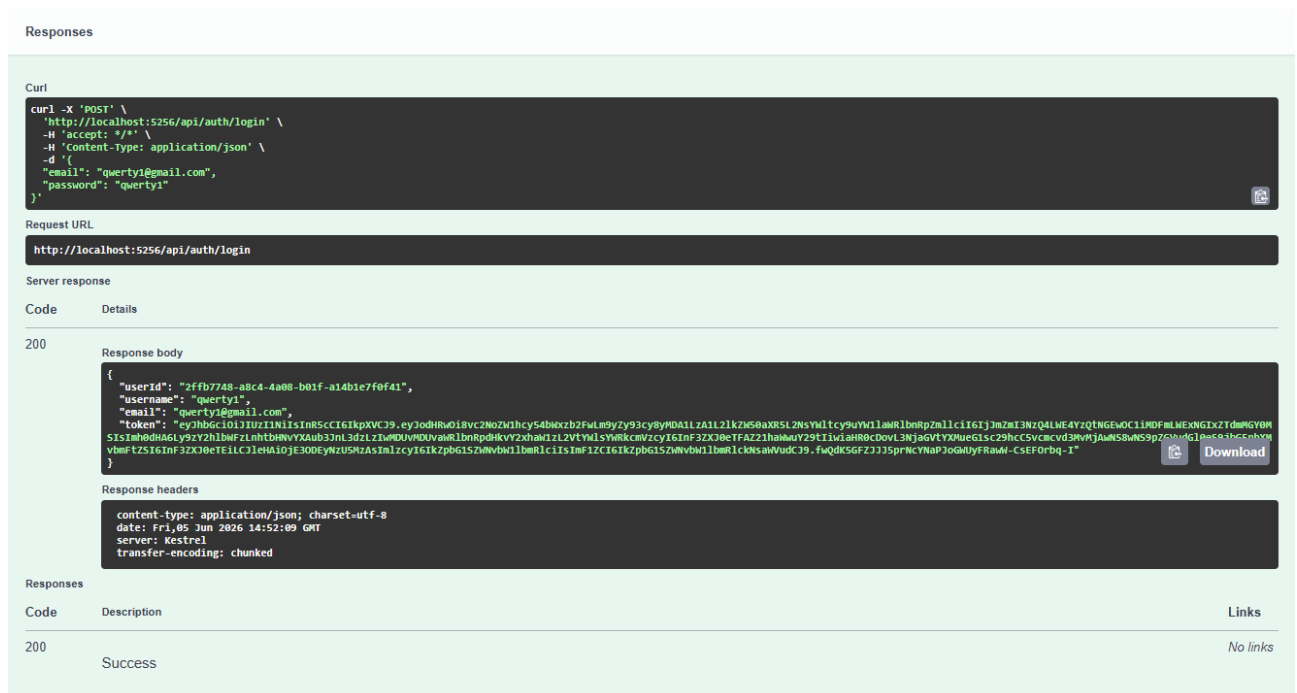


Рисунок 3.4 – Успішна автентифікація через Swagger UI

Реалізація контентної фільтрації зосереджена у класі ContentBasedService. Робота даного сервісу базується на побудові вектору вподобань користувача та подальшому порівнянні його з векторами характеристик фільмів. Формування профілю користувача здійснюється на основі двох джерел інформації:

- результат проходження опитувальника вподобань;
- історія оцінювання фільмів.

Дані опитувальника враховуються з вагою 40%, тоді як оцінки користувача формують 60% підсумкового профілю.

Для визначення ступеня подібності між векторним поданням профілю користувача та вектором характеристик фільму використовується косинусна міра подібності. Після обчислення коефіцієнтів схожості всі кандидати сортуються за спаданням отриманого значення, після чого формується список рекомендацій. Фрагмент реалізації обчислення подібності наведено на лістингу 3.1.

Лістинг 3.1 Реалізація косинусної міри подібності:

```
private static double CosineSimilarity(
Dictionary<string, double> a, Dictionary<string, double> b){
    var commonKeys = a.Keys.Intersect(b.Keys).ToList();
    if (!commonKeys.Any()) return 0;
    var dot = commonKeys.Sum(k => a[k] * b[k]);
    var magA = Math.Sqrt(a.Values.Sum(v => v * v));
    var magB = Math.Sqrt(b.Values.Sum(v => v * v));
    return (magA == 0 || magB == 0) ? 0 : dot / (magA * magB);
}
```

Другий метод рекомендацій реалізовано у класі CollaborativeService. Його робота відрізняється тим, що пошук схожих фільмів виконується не лише на основі їхніх атрибутів, а й на основі поведінкових даних користувачів. Для визначення ступеня схожості між користувачами використовується коефіцієнт кореляції Пірсона. Розрахунок виконується лише для користувачів, які мають достатню кількість спільно оцінених фільмів.

Після визначення множини найбільш схожих користувачів система прогнозує оцінки для фільмів, які користувач ще не переглядав. Прогнозована оцінка формується як зважене середнє відхилень оцінок схожих користувачів від їхніх власних середніх значень. Фрагмент реалізації обчислення схожості користувачів наведено на лістингу 3.2.

Лістинг 3.2 Реалізація обчислення коефіцієнта кореляції Пірсона:

```
private static double PearsonCorrelation(
Dictionary<Guid, double> a, Dictionary<Guid, double> b){
    var common = a.Keys.Intersect(b.Keys).ToList();
    if (common.Count < MinCommonMovies) return 0;
    var avgA = common.Average(k => a[k]);
    var avgB = common.Average(k => b[k]);
    var num = common.Sum(k => (a[k] - avgA) * (b[k] - avgB));
    var denA = Math.Sqrt(common.Sum(k => Math.Pow(a[k] - avgA, 2)));
    var denB = Math.Sqrt(common.Sum(k => Math.Pow(b[k] - avgB, 2)));
    return (denA == 0 || denB == 0) ? 0 : num / (denA * denB);
}
```

Клієнтська частина реалізована як односторінковий застосунок з маршрутизацією за допомогою технології React Router. Структура проекту поділена на окремі директорії задля логічної організації коду (рис. 3.5).

Архітектура клієнтської частини побудована за допомогою створення компонентів, що передбачає повторне використання елементів інтерфейсу в різних частинах застосунку. Взаємодія із серверною частиною реалізована в окремому наборі модулів. Логіка роботи з даними відокремлена від логіки відображення інтерфейсу користувача.

Для опису структур даних використовуються типи TypeScript. Поділ функціональності на окремі модулі допомагає облегшити розширення програмного забезпечення та підтримку коду.

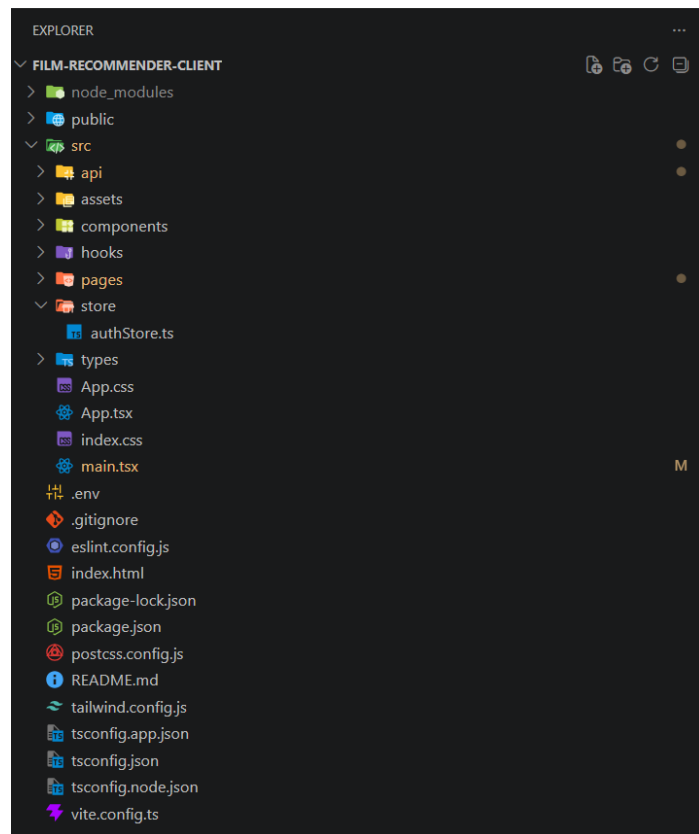


Рисунок 3.5 – Структура директорії src у Visual Studio Code

Як приклад, директорія «pages» містить компоненти окремих сторінок вебзастосунку, тоді як у «components» розміщені елементи інтерфейсу, які використовуються багато разів.

3.3 Інструкція користувача

Після запуску вебзастосунку на головній сторінці відображається каталог фільмів, який доступний для перегляду без проходження авторизації (рис. 3.6). Це дозволяє новим користувачам ознайомитися з доступним набором фільмів одразу, але для отримання персоналізованих рекомендацій та доступу до повного набору функцій необхідно пройти процедуру реєстрації. Кнопка «Sign In» що розташована у верхньому правому куті навігаційної панелі, відкриває сторінку входу до системи, де можна увійти до системи або пройти реєстрацію нового облікового запису (рис. 3.7).

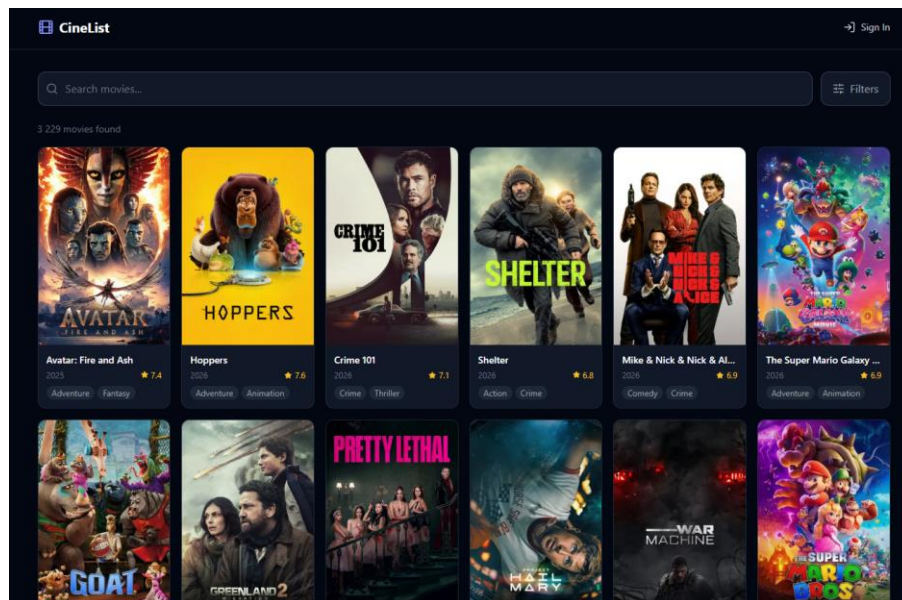


Рисунок 3.6 – Головна сторінка вебзастосунку

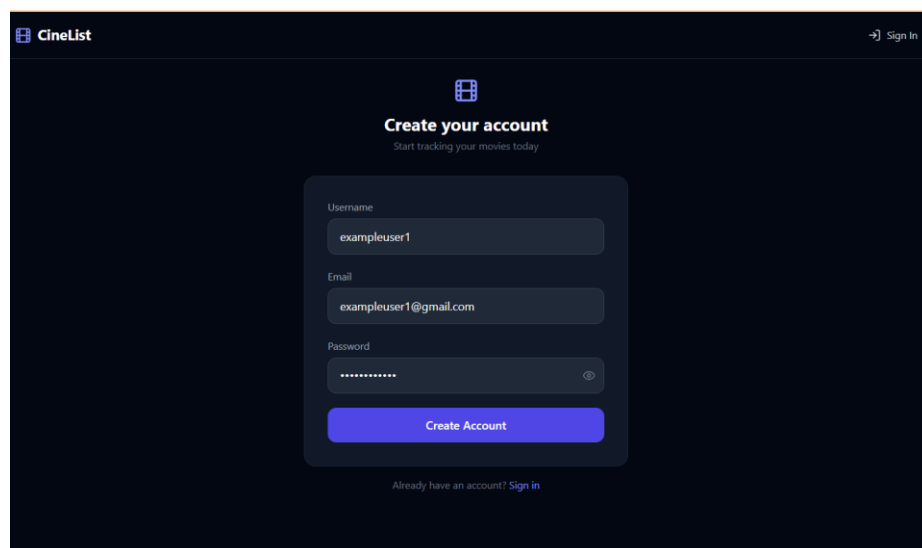


Рисунок 3.7 – Сторінка реєстрації нового користувача

Форма реєстрації містить поля для введення імені користувача, адреси електронної пошти та пароля.

Перед збереженням даних система виконує перевірку коректності введених значень. Контролюється заповнення всіх обов'язкових полів, відповідність електронної пошти встановленому формату, мінімальна довжина імені користувача та пароля. У разі порушення зазначених вимог користувач отримує відповідне повідомлення про помилку та не може завершити процедуру реєстрації до їх усунення.

Після успішного завершення реєстрації користувач автоматично авторизується та перенаправляється на сторінку проходження опитувальника вподобань, який використовується для формування початкового профілю користувача.

Опитувальник складається з чотирьох послідовних кроків. Проходження всіх етапів дозволяє системі отримати інформацію про інтереси користувача та сформувати його профіль вподобань. У верхній частині сторінки розташовано індикатор прогресу, який відображає поточний крок проходження опитування та їх загальна кількість.

На першому етапі користувач обирає жанри що відповідають його вподобанням. Жанри представлені у вигляді інтерактивних карточок, при натисканні на які жанр активується і з'являється повзунок для налаштування інтенсивності вподобання від 10% до 100% (рис. 3.8).

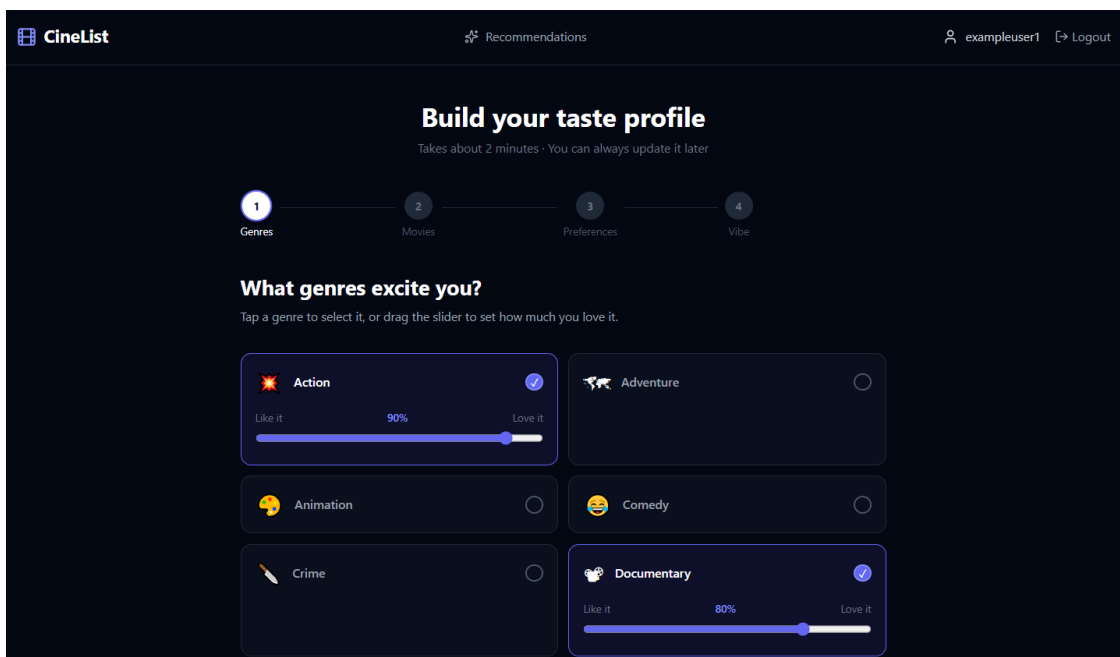


Рисунок 3.8 – Перший крок опитувальника («Жанри»)

На другому кроці відображається набір із десяти популярних фільмів, які представляють різні жанрові категорії. Для кожного фільму користувачу доступна позитивна або негативна реакція.

Позитивна реакція збільшує ваги жанрів, пов'язаних із відповідним фільмом, на 0,3, тоді як негативна реакція зменшує їх на 0,2 (рис. 3.9).

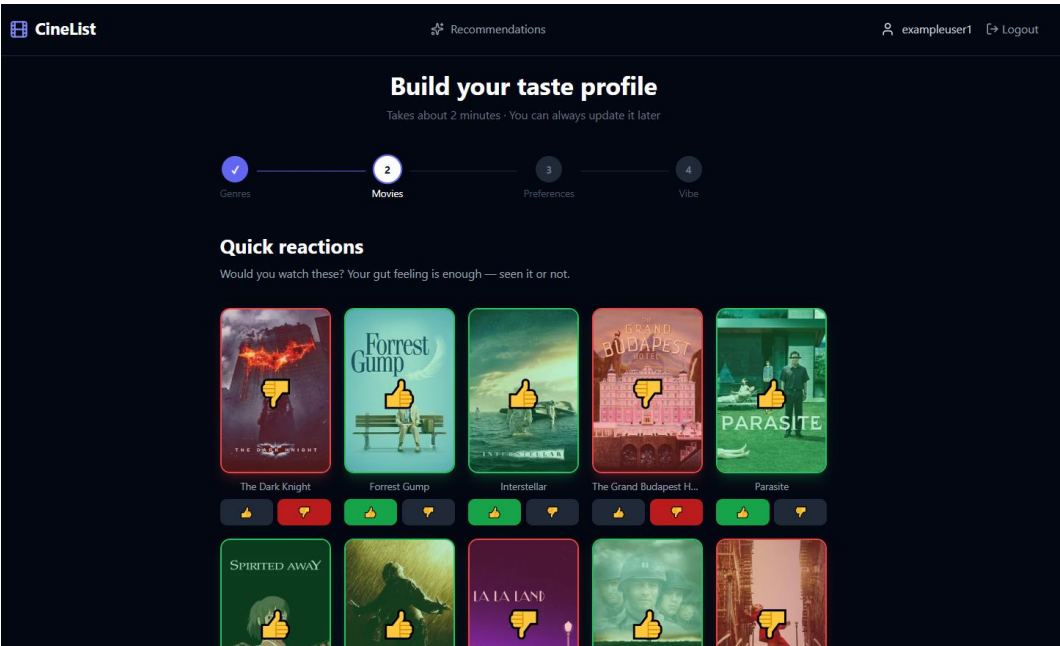


Рисунок 3.9 – Другий крок опитувальника («Реакція на популярні фільми»)

На третьому кроці користувач може вказати додаткові параметри бажаного контенту (рис. 3.10).

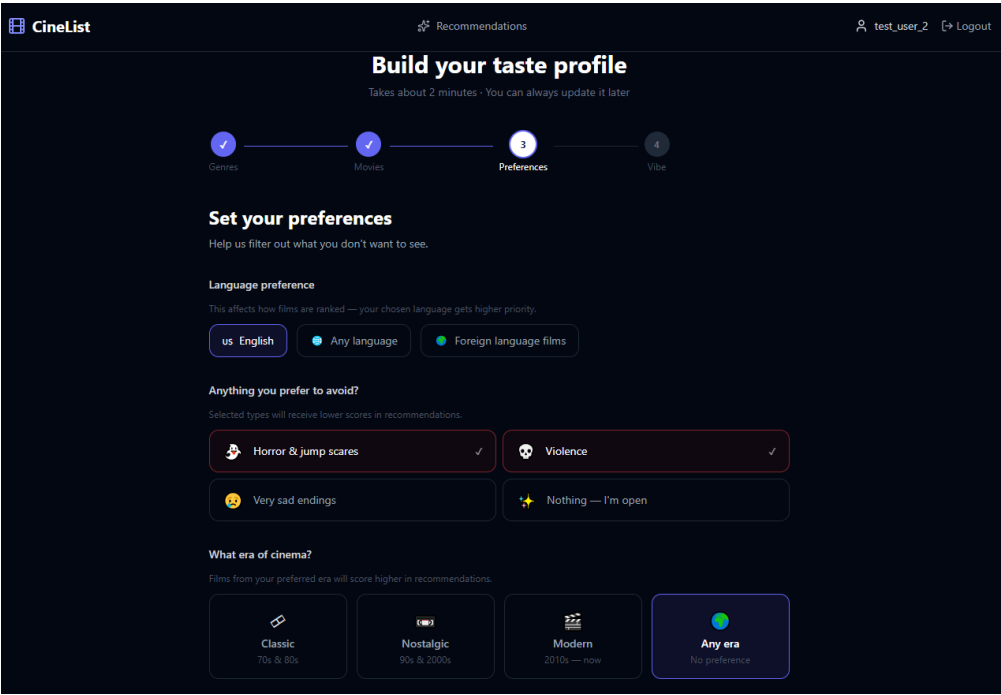


Рисунок 3.10 – Третій крок опитувальника («Налаштування вподобань»)

Цей етап дозволяє:

- налаштувати мовні уподобання;
- вибрати категорії контенту, які користувач бажає уникати у своїх рекомендаціях;
- визначити бажаний період кінематографу.

Врахування цих факторів дозволяє системі сформувати більш точний вектор вподобань та ефективно відсіяти контент, який не відповідає індивідуальним смакам користувача.

На четвертому кроці користувач обирає загальний емоційний тон бажаних фільмів (рис. 3.11).

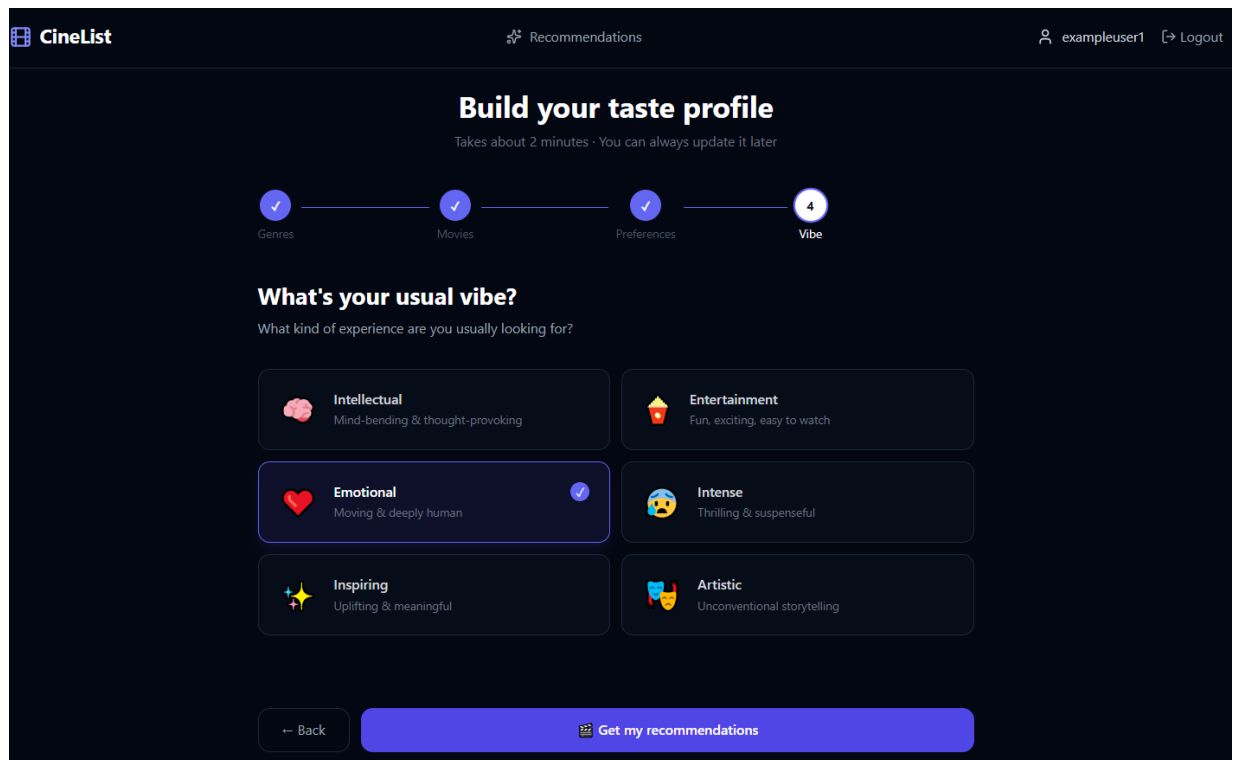


Рисунок 3.11 – Четвертий крок опитувальника («Емоційний тон»)

Цей вибір автоматично підвищує на 0,2 ваги тих жанрів у профілі користувача, які відповідають обраному настрою:

- інтелектуальний – збільшує ваги жанрів Science Fiction, Mystery та Documentary;
- розважальний – збільшує ваги жанрів Action, Comedy та Adventure;

- емоційний – збільшує ваги жанрів Drama та Romance;
- інтенсивний – збільшує ваги жанрів Thriller та Horror;
- надихаючий – збільшує ваги жанрів Drama та Family;
- художній – збільшує ваги жанрів Documentary та Animation.

Після збереження відповідей система формує векторне подання профілю користувача та перенаправляє на сторінку рекомендацій. Пройти опитувальник повторно або оновити вподобання завжди можна через кнопку «Update preferences» на сторінці профілю.

Щоб знайти необхідний користувачу фільм у системі необхідно перейти на головну сторінку застосунку, де відображається каталог фільмів у форматі таблиці у шість колонок. Для зручності пошуку реалізовано ряд інструментів фільтрації та сортування (рис. 3.12).

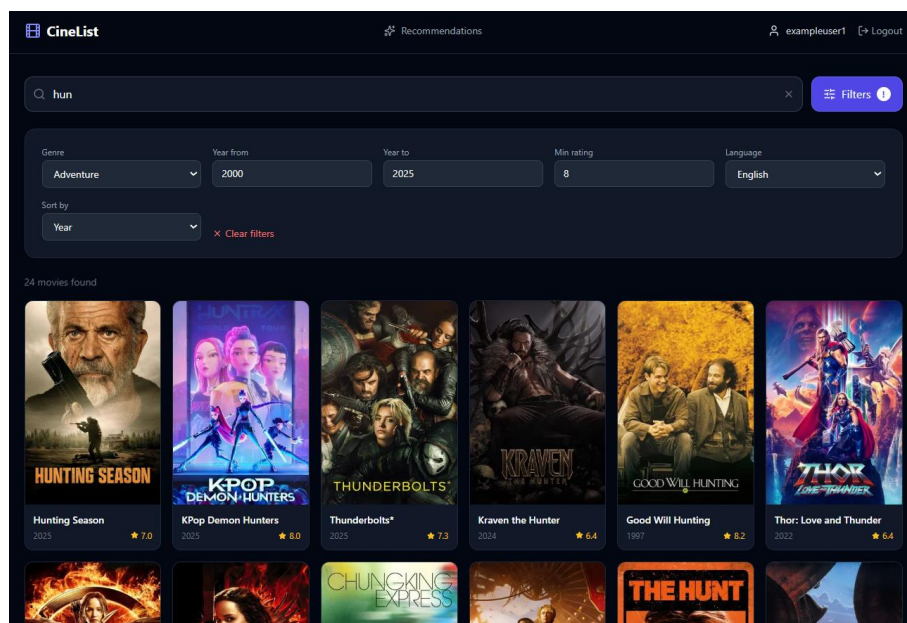


Рисунок 3.12 – Результат пошуку фільмів у каталозі за заданими параметрами

Функція пошуку дозволяє знаходити фільми за назвою, а кнопка «Filters» відкриває панель додаткових параметрів. Користувач може виконувати підбір за наступними фільтрами:

- жанр фільму;

- рік випуску;
- рейтинг фільму;
- мова оригіналу.

Додатково доступне сортування результатів:

- за популярністю;
- за рейтингом;
- за роком виходу;
- за алфавітним порядком назв.

При натисканні на картку фільму відкривається сторінка що містить детальну інформацію про фільм та функції додання до списків користувача (рис. 3.13).

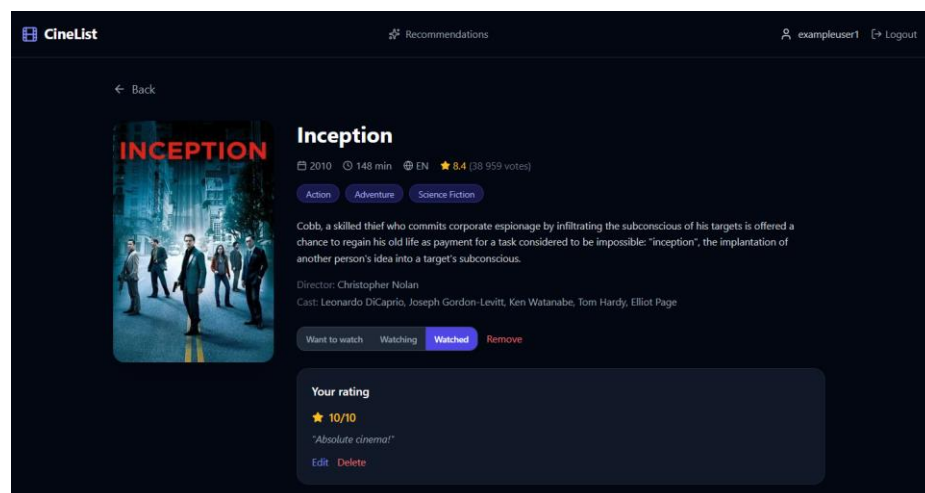


Рисунок 3.13 – Сторінка фільму із поставленою оцінкою та доданим коментарем

На сторінці фільму авторизований користувач може додати фільм до списку перегляду натиснувши кнопку «Add to Watch List». Також на сторінці розміщений блок оцінювання з інтерактивною шкалою з 10 зірок та полем для текстової рецензії. Сторінка профілю користувача доступна через навігаційне меню та містить основну інформацію про обліковий запис. Крім того, профіль надає доступ до додаткових функцій керування власними даними та перегляду історії взаємодії із системою.

Вкладка «Watch List» містить список фільмів, які користувач додав для подальшого перегляду, що дозволяє зберігати цікаві користувачу фільми на власній сторінці (рис. 3.14).

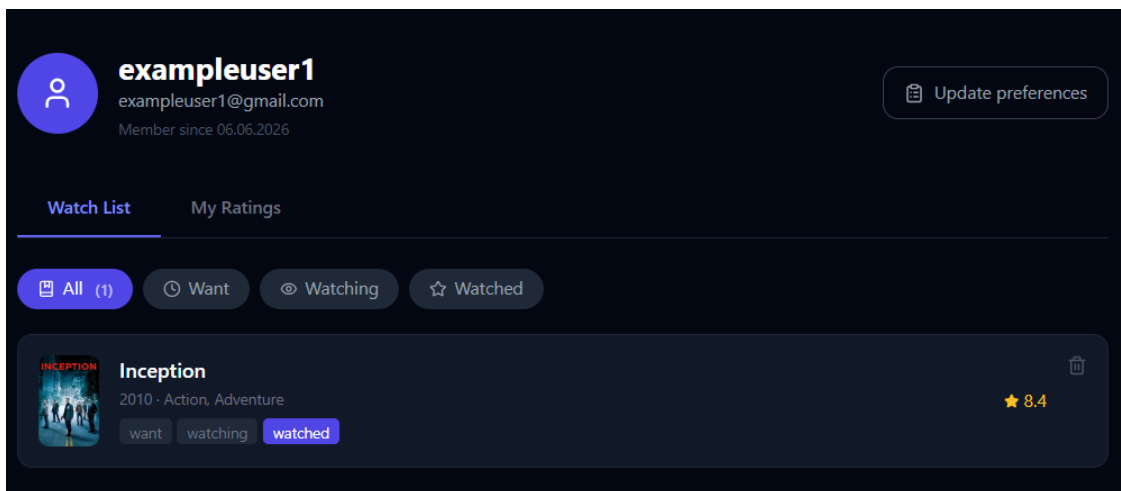


Рисунок 3.14 – Сторінка профілю користувача з доданим фільмом до списку

Сторінка рекомендацій доступна через посилання «Recommendations» у навігаційній панелі (рис. 3.15). У верхній частині відображаються дві карточки алгоритмів – «Content-based» та «Collaborative». Карточка «Content-based» завжди доступна та позначена зеленим індикатором. Карточка «Collaborative» відображає прогрес-бар з кількістю оцінок та кількістю що залишилась до порогу в 100 оцінок поки модель не розблокована.

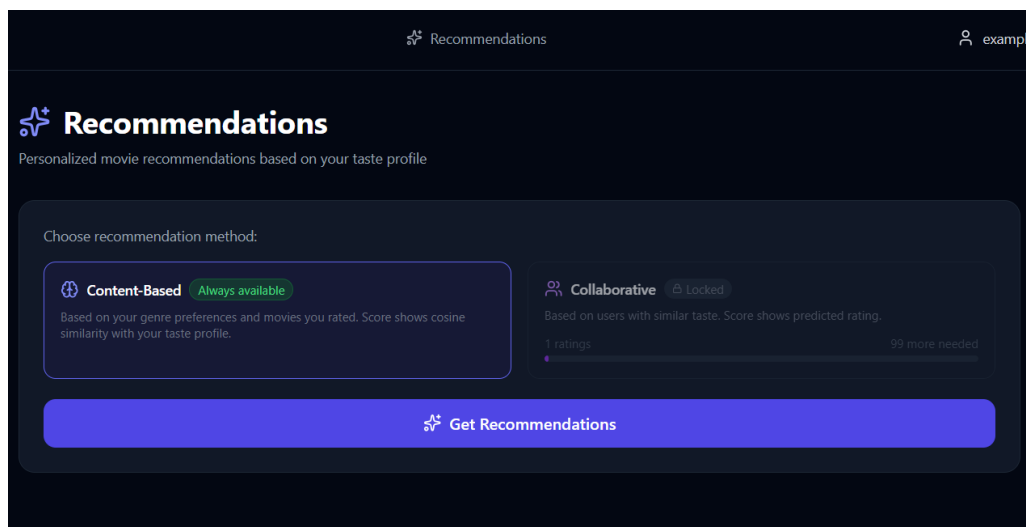


Рисунок 3.15 – Сторінка надання рекомендацій із вибором моделі

Після натискання кнопки «Get Recommendations» система виконує запуск відповідного алгоритму та формує список рекомендованих фільмів (рис. 3.16). Для кожної рекомендації відображається зображення, основна інформація про фільм та спеціальний індикатор. У випадку Content-Based filtering індикатор показує ступінь схожості фільму із профілем користувача у відсотках. Для колаборативної моделі індикатор відображає прогнозовану оцінку, яку користувач потенційно може поставити фільму.

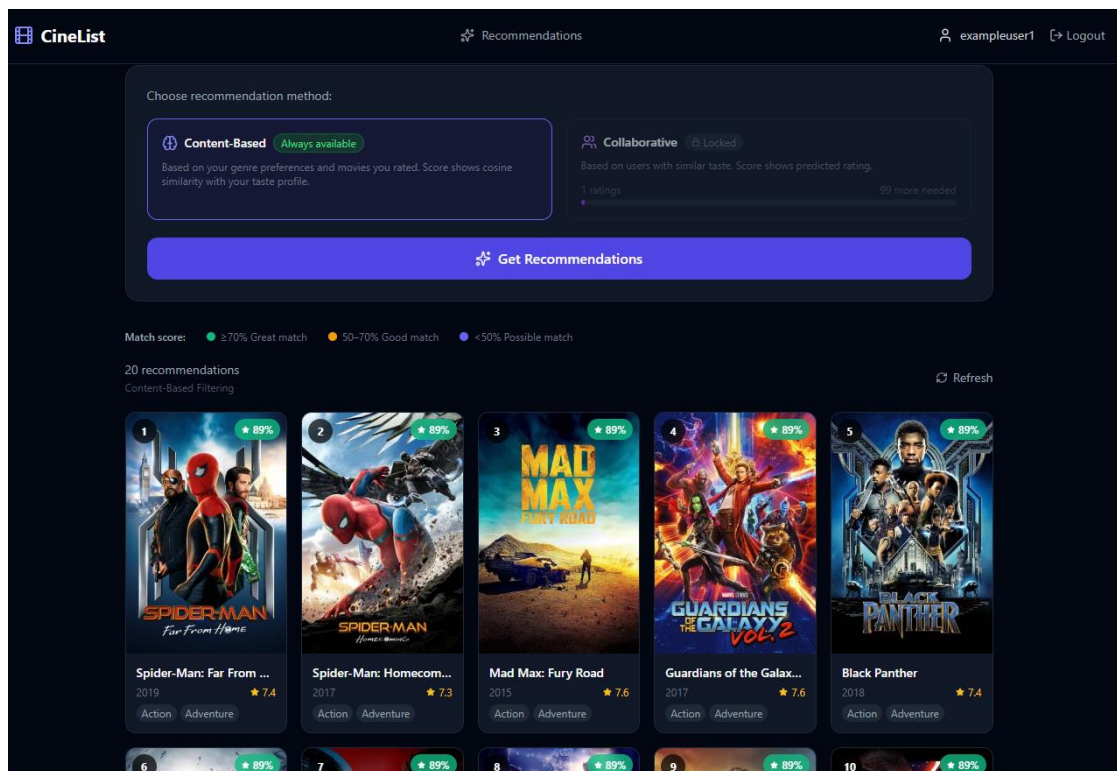


Рисунок 3.16 – Отриманий список рекомендацій
за допомогою Content-based методу

Для використання методу Collaborative Filtering користувач повинен оцінити не менше 100 фільмів. Така вимога обумовлена необхідністю накопичення достатнього обсягу даних для більш точного списку рекомендацій. Для цього користувач самостійно обирає фільми з каталогу та виставляє їм оцінки. У процесі накопичення оцінок формується історія взаємодії користувача із системою, яка використовується під час пошуку схожих користувачів.

Після досягнення необхідного порогу оцінок модель колаборативної фільтрації стає доступним для використання. Система аналізує оцінки інших користувачів, визначає найбільш схожі профілі та формує персоналізований список рекомендацій (рис. 3.17).

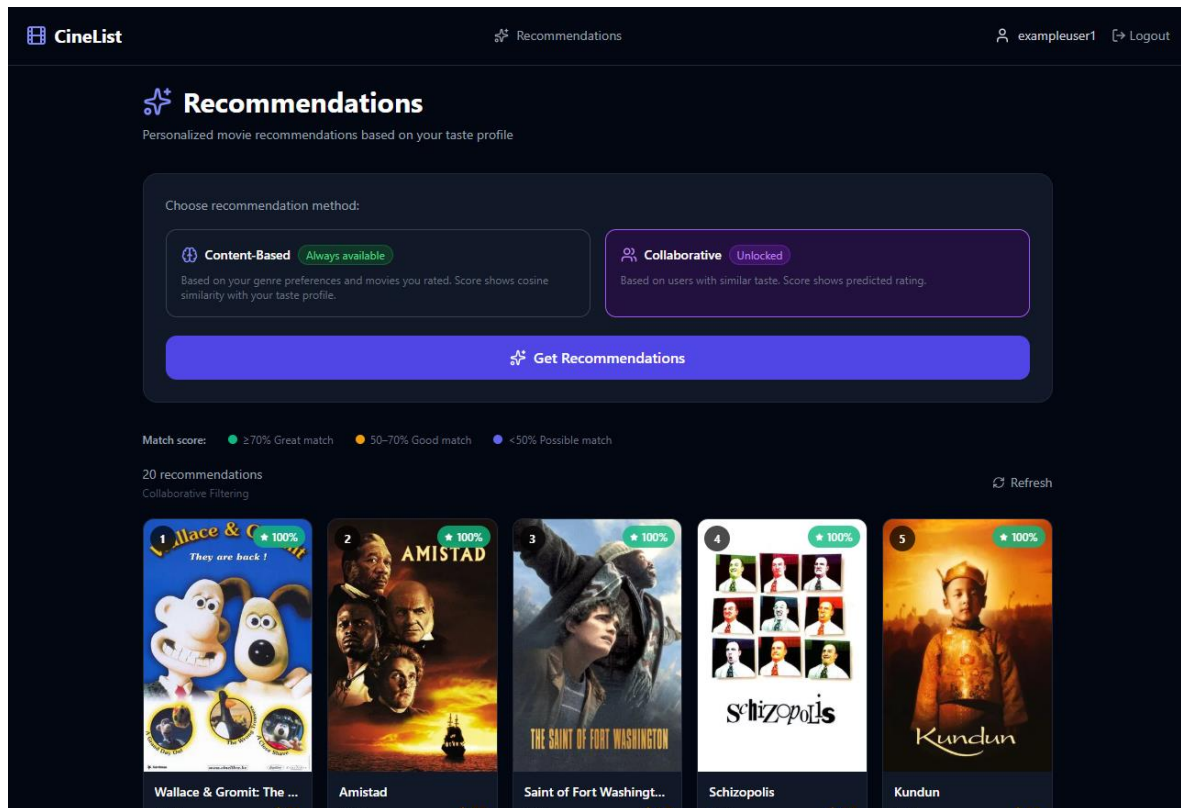


Рисунок 3.17 – Отриманий список рекомендацій за допомогою CF

3.4 Тестування розробленого вебзастосунку та аналіз результатів

Тестування розробленої рекомендаційної системи проводилось у два етапи:

- функціональне тестування;
- тестування за допомогою метрик.

Перший етап перевіряє коректність роботи всіх підсистем, тоді як другий перевірює якість роботи моделей надання рекомендацій за описаними метриками.

Функціональне тестування виконувалось вручну через інтерфейс Swagger UI для кінцевих точок (endpoints) серверної частини та через браузер для клієнтської частини.

Тестування підсистеми автентифікації підтвердило коректну роботу основних механізмів безпеки вебзастосунку, зокрема реєстрації користувачів із хешуванням пароля, генерації JWT-токенів та перевірки прав доступу до захищених endpoints. Під час реєстрації пароль користувача не зберігається у відкритому вигляді, а попередньо проходить процес хешування, що забезпечує додатковий рівень захисту даних.

Після успішної автентифікації користувач отримує JWT-токен, який використовується для подальших запитів до захищених ресурсів. Сервер перевіряє наявність і валідність токена перед наданням доступу до відповідних endpoints, що дозволяє обмежити доступ для неавторизованих користувачів.

Спроба звернення до захищеного endpoint без надання токена завершується поверненням статусу «401 Unauthorized», що підтверджує правильну роботу механізму контролю доступу (рис. 3.18).

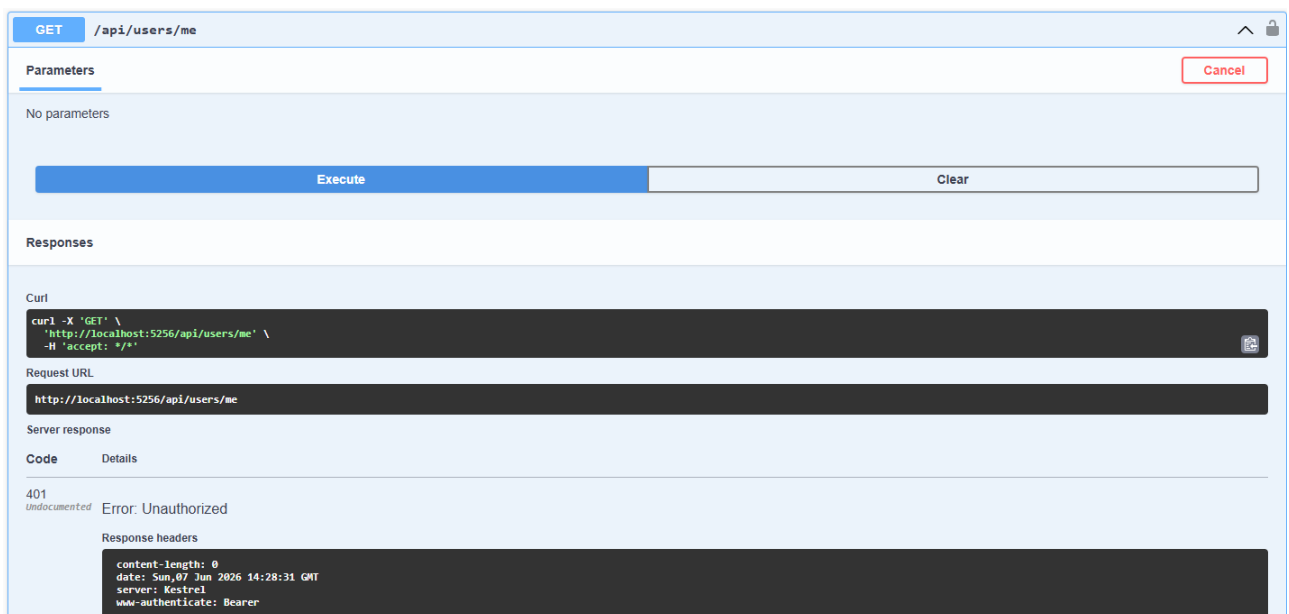


Рисунок 3.18 – Перевірка доступу до захищеного endpoint без авторизації

У разі виконання запиту з дійсним JWT-токеном система успішно надає доступ до ресурсу та повертає очікувані дані (рис. 3.19).

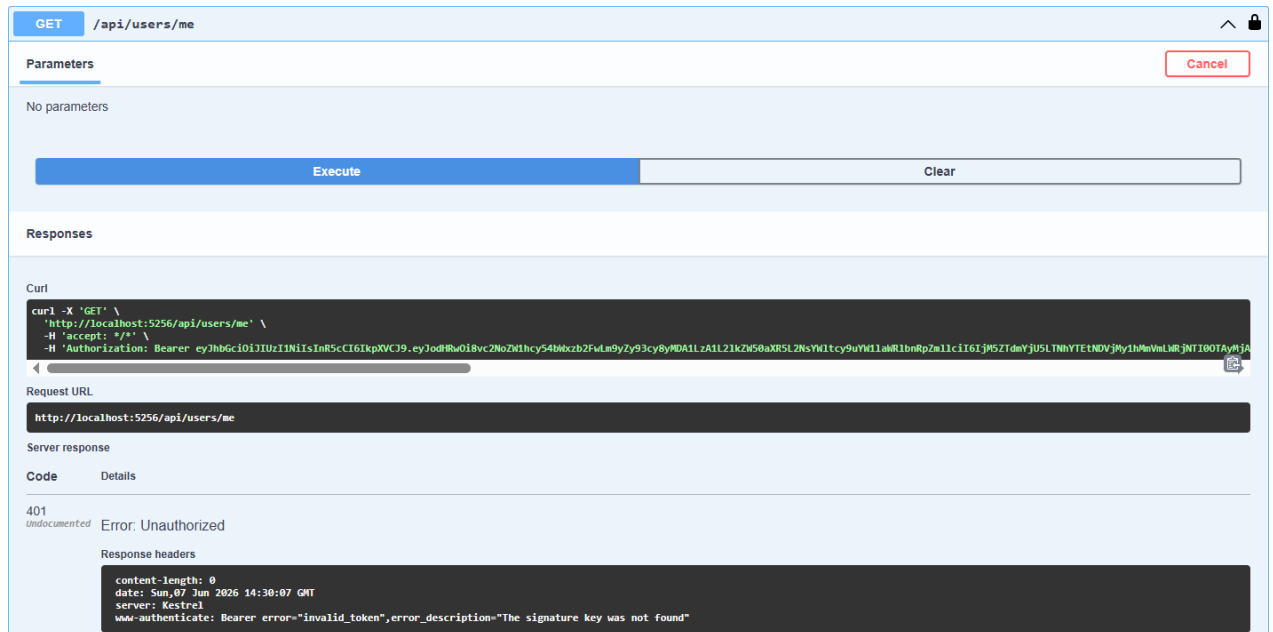


Рисунок 3.19 – Перевірка доступу до захищеного endpoint із успішною авторизацією

Протестовано систему пошуку та фільтрації фільмів, яка забезпечує швидкий пошук контенту за назвою та застосування розширених параметрів відбору.

Основною метою тестування було перевірити коректність роботи пошукових механізмів, взаємодію фільтрів між собою, а також відповідність отриманих результатів заданим критеріям користувача.

Тестування проводилось вручну через інтерфейс вебзастосунку для різних сценаріїв використання. Перевірялась робота текстового пошуку за введенням назви фільму, а також комбіноване використання фільтрів. Для більш детальної перевірки коректності роботи пошукової підсистеми було сформовано набір типових сценаріїв використання, що охоплюють як прості запити, так і складні комбіновані фільтрації. Результати тестування пошукової підсистеми наведені у таблиці 3.1.

Таблиця 3.1 – Сценарії та результати функціонального тестування модуля пошуку

Сценарій тестування	Вхідні параметри	Очікуваний результат	Статус
Пошук за частковою назвою	«The»	Відображення фільму назва якого починається на «The»	Успішно
Пошук за повною назвою	«Inception»	Відображення фільму «Inception»	Успішно
Фільтрація за жанром	Genre = Action	Відображення лише бойовиків	Успішно
Фільтрація за роком випуску	Year = 2010	Відображення фільмів 2010 року	Успішно
Фільтрація за рейтингом	Rating $\geq$ 8,0	Відображення фільмів із рейтингом не нижче 8,0	Успішно
Фільтрація за мовою	Language = English	Відображення англomовних фільмів	Успішно
Комбінована фільтрація	Genre = Action, Rating $\geq$ 7,5	Відображення бойовиків із високим рейтингом	Успішно
Сортування за рейтингом	Sort = Rating	Фільми впорядковані від найвищого рейтингу до найнижчого	Успішно

Підтверджено, що система коректно обробляє текстові запити, як за повною назвою фільму так і за частковою. Фільтри можуть застосовуватись одночасно, що надає можливість формувати складні запити з урахуванням декількох критеріїв. Після операцій додання чи видалення фільтру результати автоматично оновлюються відповідно до обраних параметрів.

Це підвищує зручність використання системи у цілому, тому що зменшує необхідну кількість дій для знаходження користувачем необхідної йому інформації.

На рисунку 3.20 наведено приклад роботи пошуку за комбінованим запитом.

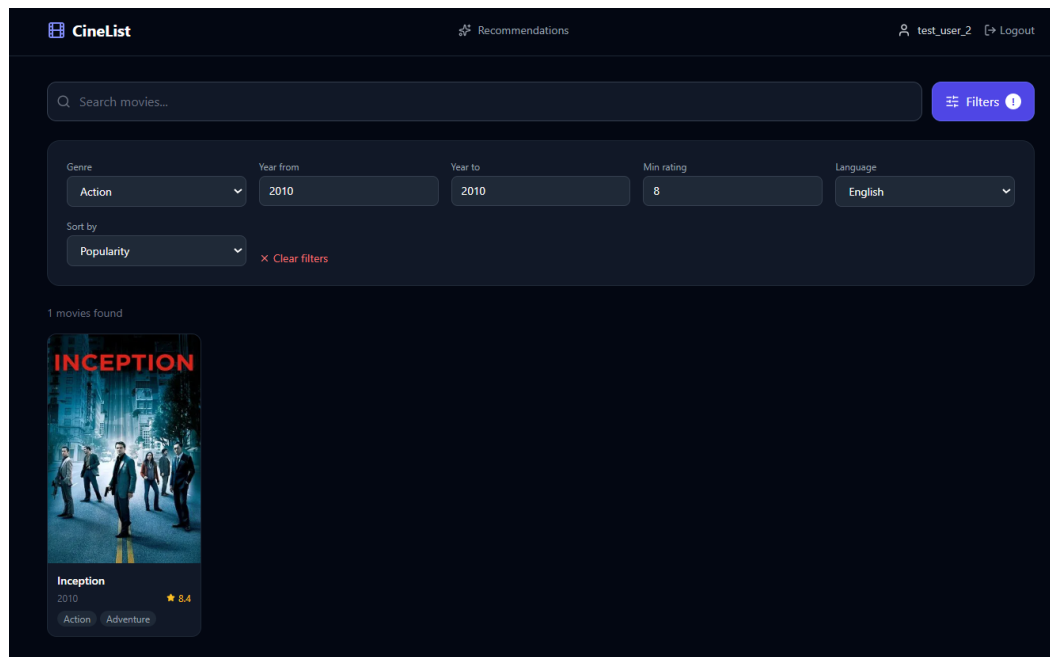


Рисунок 3.20 – Результат роботи пошуку фільму за комбінованим запитом

Тестування роботи опитувальника із подальшими рекомендаціями виконувалося з метою перевірки коректності формування персоналізованих списків фільмів для користувачів із різними вподобань. Для цього було створено два тестові облікові записи, які проходили опитувальник, обираючи різні один від одного категорії фільмів. Після завершення опитування було сформовано векторні подання смаків для кожного користувача, які надалі використовувались методом контентної фільтрації під час генерації рекомендацій.

Перший тестовий користувач обирав переважно науково-фантастичні бойовики, тоді як другий користувач надавав перевагу сімейним мультфільмам. Отримана системою інформація про користувачів наведена у таблицях 3.2 та 3.3.

Таблиця 3.2 – Вподобання користувача test_user_1, зазначені під час анкетування

Характеристика	Значення
Улюблені жанри	Action, Adventure, Science Fiction
Мова перегляду	Англійська
Небажані жанри	Horror
Ера кінематографу	Modern (2010s – now)
Емоційний тон	Entertainment (Action, Comedy, Adventure)

Таблиця 3.3 – Вподобання користувача test_user_2, зазначені під час анкетування

Характеристика	Значення
Улюблені жанри	Animation, Family
Мова перегляду	Foreign language films
Небажані жанри	War, Crime, Horror
Ера кінематографу	Any era
Емоційний тон	Inspiring (Drama, Family)

Після проходження опитувальника кожному користувачеві було сформовано список рекомендацій за допомогою методу контентної фільтрації. Аналіз отриманих результатів показав, що система враховує сформований профіль та пропонує різні набори фільмів залежно від індивідуальних вподобань користувача.

Для користувача test_user_1 були обрані жанри «Action», «Adventure» та «Science Fiction», англійська мова перегляду, сучасна епоха кінематографу, а також емоційний тон «Entertainment».

Також, користувач зазначив що не бажає отримувати рекомендації фільмів жанру «Horror».

У результаті сформований список рекомендацій містив переважно сучасні фільми, які повністю відповідали заданим жанрам, тоді як фільми жанру Horror були повністю відсутні (рис. 3.21).

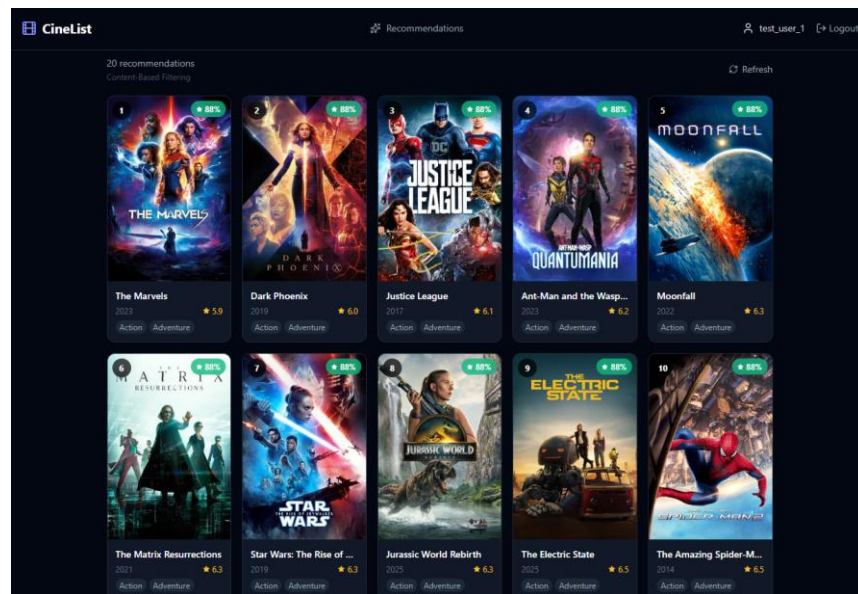


Рисунок 3.21 – Список рекомендацій для користувача test_user_1

Для користувача test_user_2 були обрані жанри «Animation» та «Family», перевага іноземномовного контенту, відсутність обмежень щодо епохи кінематографу та емоційний тон «Inspiring». Додатково користувач виключив з рекомендацій контент, пов'язаний із жанрами «War», «Crime» та «Horror». За результатами генерації рекомендацій система сформувала список, у якому переважали анімаційні та сімейні фільми, а також драматичні стрічки. Водночас фільми, що належать до небажаних категорій, не потрапляли до верхніх позицій рейтингу (рис. 3.22).

Візуальне порівняння отриманих рекомендацій показало суттєві відмінності між сформованими списками. Більшість рекомендованих фільмів для test_user_1 не збігалася з рекомендаціями для test_user_2, що підтверджує врахування індивідуальних характеристик профілю під час ранжування контенту.

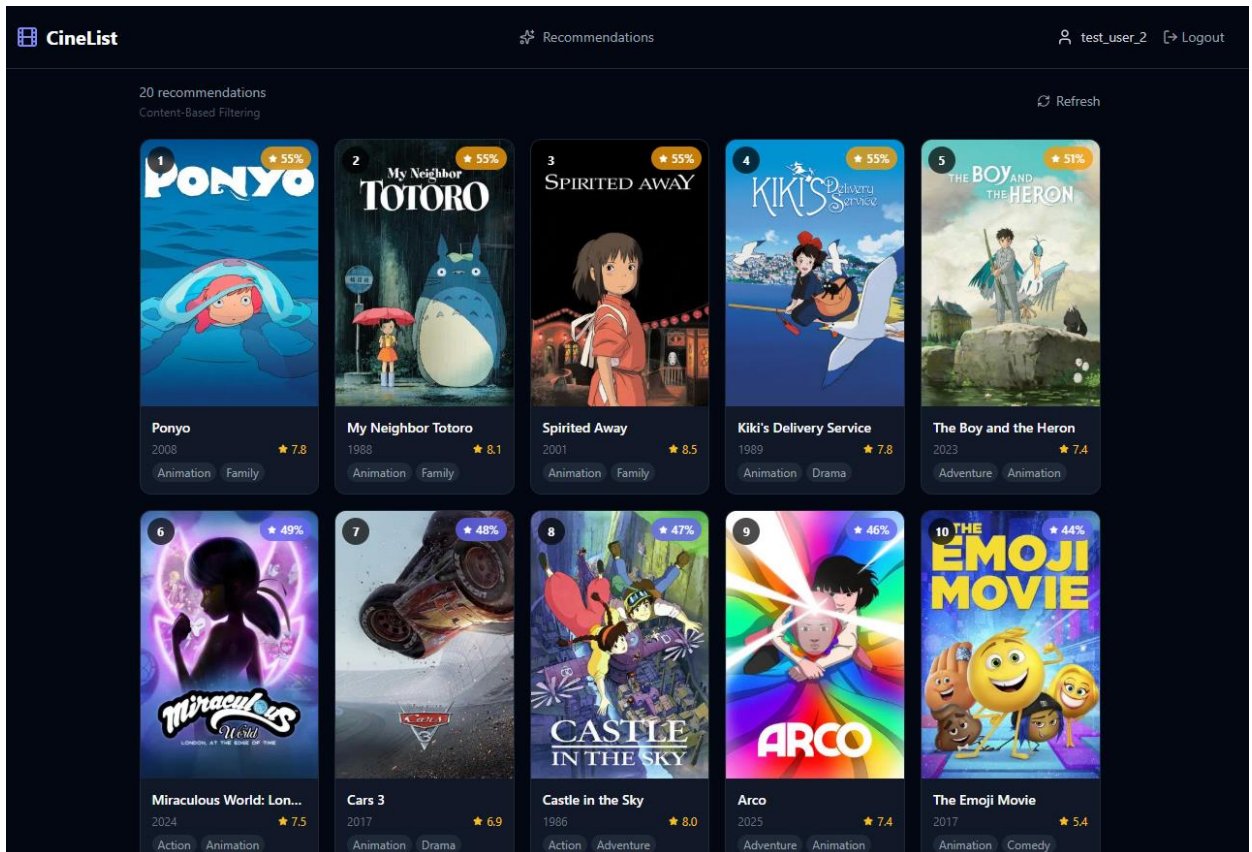


Рисунок 3.22 – Список рекомендацій для користувача test_user_2

Додатково було виконано порівняння числових значень векторного подання вподобань, сформованих на основі результатів анкетування користувачів. Завдяки цьому оцінимо не лише фінальні рекомендації, але й внутрішню структуру профілів користувачів, яка впливає на процес ранжування контенту.

Аналіз отриманих значень підтвердив вплив жанрів Action та Adventure на профіль користувача test_user_1, тоді як профіль test_user_2 має виражену схильність до Animation та Family. Також спостерігаються суттєві відмінності у мовних вподобаннях.

Отримані результати підтверджують, що система коректно відображає індивідуальні уподобання у вигляді числових ваг.

Найбільші ваги векторних подань профілів користувачів наведені у таблицях 3.4 та 3.5.

Таблиця 3.4 – Найбільші за впливом ваги векторних подань профілю користувача test_user_1

Вага вектору	Значення
Genre: Action	0,4531
Genre: Adventure	0,4531
Decade: 2010s	0,3625
Language: English	0,4078
Decade: 2020s	0,3625
Genre: Science fiction	0,3625

Таблиця 3.5 – Найбільші за впливом ваги векторних подань профілю користувача test_user_2

Вага вектору	Значення
Genre: Family	0,4851
Genre: Animation	0,4851
Language: Japanese	0,3395
Language: Korean	0,3395
Language: Spain	0,2910
Language: French	0,2910
Language: Deutsch	0,2425
Language: Italian	0,2425

Тепер, для перевірки оновлення рекомендацій після самостійного оцінювання перших фільмів, test_user_2 виставить оцінки фільмам, зазначеним у таблиці 3.6. Після внесення оцінок буде здійснено повторну перевірку результатів формування рекомендацій (рис. 3.23).

Таблиця 3.6 – Фільми оцінені користувачем test_user_2

Назва	Жанри	Мова	Рік	Оцінка
The Godfather	Action, Crime, Drama	Англійська мова	1972	10
The Godfather Part II	Action, Crime, Drama	Англійська мова	1974	10
Bonnie and Clyde	Crime, Drama	Англійська мова	1967	9
Taxi Driver	Crime, Drama	Англійська мова	1976	9
Dirty Harry	Action, Crime, Thriller	Англійська мова	1971	9

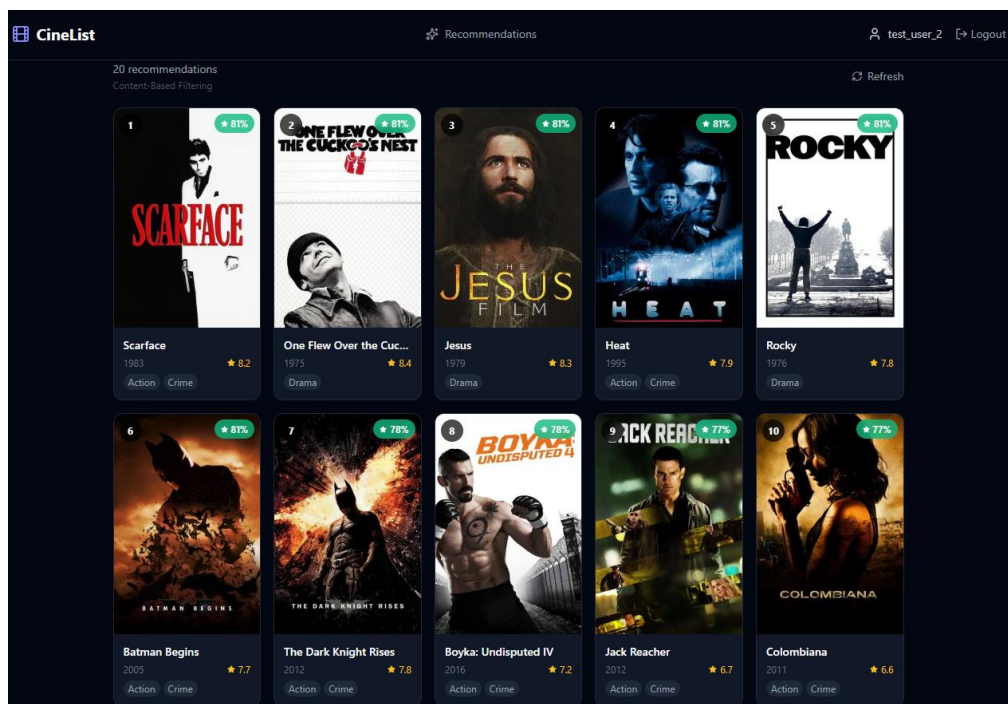


Рисунок 3.23 – Оновлені рекомендації для користувача test_user_2

Аналіз нових рекомендацій показав, що після внесення оцінок користувачем test_user_2 система адаптувала результуючий список відповідно до змінених уподобань. Порівняння ваг старого та нового векторів уподобань зазначено у таблиці 3.7.

Таблиця 3.7 – Порівняння ваг старого та нового вектору вподобань користувача test_user_2

Вага вектору	Старе значення	Нове значення
Language: English	0,0485	0,4732
Genre: Crime	0,0000	0,4732
Genre: Drama	0,0970	0,3960
Normalized Rating	0,0000	0,3826
Decade: 1970s	0,0000	0,3826
Genre: Action	0,0000	0,2920
Genre: Thriller	0,0000	0,0906
Genre: Family	0,4851	0,0671
Genre: Animation	0,4851	0,0671
Language: Japanese	0,3395	0,0470
Language: Korean	0,3395	0,0470
Language: Spain	0,2910	0,0403
Language: French	0,2910	0,0403
Language: Deutsch	0,2425	0,0336
Language: Italian	0,2425	0,0336
Normalized Popularity	0,0000	0,0216

Після оновлення вектору вподобань спостерігається суттєва зміна розподілу ваг між його компонентами. Якщо до оцінювання найбільший вплив на формування рекомендацій мали жанри «Family» та «Animation», а також ознаки, пов'язані з переглядом іноземномовного контенту, то після додавання оцінок домінуючими стали характеристики напряму пов'язані з характеристиками доданих до профілю фільмів.

Через це спостерігається зміщення рейтингу в бік фільмів із жанрами «Crime» та «Action», що відповідає оціненню користувачем стрічкам.

Методологія тестування якості розроблених моделей базується на використанні статистичних моделей аналізу даних для визначення релевантності результатів [16–30].

Оцінювання якості реалізованих моделей надання рекомендацій виконувалось за допомогою реалізації hold-out методу оцінювання.

Для кожного користувача набору даних MovieLens 100K, що мають щонайменше 20 оцінок, 40% оцінок відкладалось як тестова вибірка, а решта використовувалась для генерації рекомендацій.

Результати оцінювання моделі контентної фільтрації за метрикою Precision@10 наведені у таблиці 3.8.

Таблиця 3.8 – Результати оцінювання моделі контентної фільтрації

Метрика	Значення
Precision@10	0,1015
Кількість протестованих користувачів	546
Загальна кількість знайдених релевантних фільмів	554
Загальна кількість рекомендованих фільмів	5413

Отримане значення Precision@10 = 0,1015 свідчить про наявність релевантних рекомендацій у сформованих списках та підтверджує здатність моделі враховувати вподобання користувачів під час формування рекомендацій.

Для базової реалізації контентної фільтрації отриманий результат можна вважати задовільним. Найбільш перспективним напрямом для підвищення точності є розширення вектору ознак фільму шляхом включення текстових характеристик з поля description таблиці movies. Поточна реалізація використовує тільки ознаки що охоплюють жанри, десятиріччя, мову та якісні показники.

При цьому текстовий опис фільму що містить інформацію про сюжет, тематику та атмосферу залишається невикористаним.

Модель колаборативної фільтрації оцінювався за метрикою RMSE, результати наведені у таблиці 3.9.

Таблиця 3.9 – Результати оцінювання моделі колаборативної фільтрації

Метрика	Значення
RMSE	1,9315
Кількість протестованих користувачів	567
Кількість спрогнозованих пар	33889

Отримане значення RMSE становить 1,9315. Оскільки всі оцінки у системі були перетворені з п'ятибальної шкали набору даних MovieLens у десятибальну шляхом множення на два, значення RMSE також збільшилося приблизно вдвічі. У перерахунку на початкову шкалу оцінювання отримане значення відповідає $RMSE \approx 0,97$. Це свідчить про те, що середня похибка прогнозування становить менше однієї одиниці рейтингу по п'ятибальній шкалі оцінювання та підтверджує здатність моделі прогнозувати оцінки користувачів з прийнятною точністю.

Таким чином, за результатами проведеного оцінювання обидві реалізовані моделі продемонстрували задовільні результати та підтвердили можливість формування релевантних персоналізованих рекомендацій.

Водночас модель колаборативної фільтрації має суттєве обмеження, пов'язане з проблемою «холодного старту», коли для нових користувачів відсутня достатня кількість оцінок. Саме тому у розробленій системі використано гібридний підхід, що поєднує переваги обох моделей.

3.5 Перспективи подальшої роботи

Одним із перспективних напрямів розвитку є підвищення точності рекомендацій, наданих моделлю контентної фільтрації, для цього необхідно впровадити використання методу TF-IDF (Term Frequency – Inverse Document Frequency) для аналізу текстових описів сюжетів фільмів. У поточній реалізації система використовує переважно структуровані характеристики фільмів. Водночас значний обсяг корисної інформації міститься безпосередньо в текстових описах сюжетів, які наразі не беруть участі у формуванні рекомендацій. Метод TF-IDF дозволяє перетворити текстовий опис фільму на числовий вектор ознак, де кожне слово отримує вагу залежно від частоти його використання в конкретному описі та рідкості серед усіх фільмів бази даних. Завдяки цьому загальні слова, які зустрічаються майже в усіх описах, отримують низьку вагу, тоді як слова, що характеризують специфічну тематику фільму, отримують значно вищі значення.

Ще одним напрямом розвитку системи є використання показника CTR для аналізу взаємодії користувачів із рекомендаціями. Наразі розроблена система лише зберігає інформацію про сформовані рекомендації та фіксує переходи на сторінки рекомендованих фільмів. У майбутньому зібрані дані можуть використовуватись як додатковий сигнал під час формування профілю вподобань. Наприклад, фільми, за рекомендаціями яких користувач часто переходить, можуть отримувати більший вплив на подальше ранжування контенту, що сприятиме підвищенню рівня персоналізації системи.

Наступним покращенням може бути розширення інтеграції із зовнішнім сервісом TMDb шляхом отримання та оброблення додаткових метаданих про контент, зокрема:

- відеоматеріалів;
- текстових рецензій та відгуків критиків;
- актуального статусу виходу;

- ключових слів;
- інформації щодо належності до франшиз;
- фотографій акторського та режисерського складу;
- кадрів із фільмів.

Це дозволить покращити інформативність вебзастосунку та розширити набір характеристик, що використовуються моделями під час побудови рекомендацій.

Реалізація зазначених напрямів розвитку системи забезпечить підвищення релевантності рекомендацій та розширить функціональні можливості вебзастосунку.

ВИСНОВКИ

У рамках кваліфікаційної роботи розроблено та реалізовано вебзастосунок для формування списку рекомендованих фільмів на основі оброблення опитувальника вподобань користувача. Робота охоплює повний цикл створення застосунку – від аналізу існуючих рішень та методів побудови рекомендаційних систем до практичної реалізації гібридного алгоритму та оцінювання його якості, і вирішено такі завдання:

- проведено аналіз сучасних вебзастосунків формування списку рекомендованих фільмів в Україні та за кордоном, що дало можливість виявити їх переваги та недоліки, зокрема встановлено що не всі з розглянутих платформ використовують розгорнутий структурований опитувальник як основний інструмент формування початкового профілю вподобань користувача;

- проведено аналіз методів побудови рекомендаційних систем та підходів до вирішення проблем персоналізації що дало можливість обґрунтувати вибір гібридного підходу як найбільш відповідного для вирішення проблеми «холодного старту» при одночасному забезпеченні точності рекомендацій для активних користувачів;

- спроектовано реляційну базу даних що складається з сімнадцяти таблиць та забезпечує зберігання каталогу фільмів, профілів користувачів, результатів опитувальника, оцінок та рекомендацій;

- реалізовано алгоритм контентної фільтрації що дозволяє формувати персоналізовані рекомендації починаючи з першого входу в систему;

- реалізовано алгоритм колаборативної фільтрації що визначає схожість між користувачами та формує рекомендації спираючись на вподобання найбільш схожих користувачів.

Для реалізації серверної частини застосунку використано платформу .NET з архітектурним підходом Minimal API та систему управління базами даних PostgreSQL.

Клієнтську частину реалізовано з використанням бібліотеки React та мови TypeScript. Результати оцінювання якості розробленої системи підтвердили працездатність обох реалізованих моделей надання рекомендацій.

Проведено тестування створених моделей за допомогою hold-out методу на наборі даних MovieLens 100K. Модель контентної фільтрації досягла значення $\text{Precision}@10 = 0,1015$, що підтверджує її здатність формувати релевантні рекомендації. Модель колаборативної фільтрації отримала значення $\text{RMSE} \approx 0,97$ у перерахунку на п'ятибальну шкалу, що свідчить про прийнятну точність прогнозування оцінок користувачів.

Розглянуто перспективи подальшого розвитку системи, зокрема впровадження методу TF-IDF для аналізу текстових описів сюжетів фільмів, використання показника CTR як додаткового сигналу під час формування профілю вподобань, а також розширення інтеграції із зовнішнім сервісом TMDB для збагачення метаданих контенту. Реалізація зазначених напрямів сприятиме підвищенню точності та релевантності рекомендацій і розширенню функціональних можливостей вебзастосунку.

Результати роботи апробовано у вигляді тези доповіді під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» [31].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. JENEI, S., Róbert, N. A. G. Y., MORENO-GARCÍA, E., DÁVID, L. D., PALLÁS, E. I., & Klaudia, V. É. G. H. (2025). The Rise of Streaming Services: A Challenge to Traditional Television? *Journal of Cultural Analysis and Social Change*, 2547-2559.
2. Sheng, N., Yang, C., Han, L., & Jou, M. (2023). Too much overload and concerns: Antecedents of social media fatigue and the mediating role of emotional exhaustion. *Computers in Human Behavior*, 139, 107500.
3. Netflix – Watch TV Shows Online, Watch Movies Online. URL: <https://www.netflix.com> (дата звернення 01.05.2026).
4. IMDb: Ratings, Reviews, and Where to Watch the Best Movies & TV Shows. URL: <https://www.imdb.com> (дата звернення 01.05.2026).
5. Letterboxd • Social film discovery. URL: <https://letterboxd.com> (дата звернення 01.05.2026).
6. MEGOGO.NET Дивитися фільми та телеканали онлайн. URL: <https://megogo.net/ua> (дата звернення 01.05.2026).
7. Taste – Movie and TV show recommendations from likeminded people. URL: <https://www.taste.io> (дата звернення 01.05.2026).
8. Widayanti, R., Chakim, M. H. R., Lukita, C., Rahardja, U., & Lutfiani, N. (2023). Improving recommender systems using hybrid techniques of collaborative filtering and content-based filtering. *Journal of Applied Data Sciences*, 4(3), 289-302.
9. Uta, M., Felfernig, A., Le, V. M., Tran, T. N. T., Garber, D., Lubos, S., & Burgstaller, T. (2024). Knowledge-based recommender systems: overview and research directions. *Frontiers in big Data*, 7, 1304439.
10. Liao, M., Sundar, S. S., & B. Walther, J. (2022, April). User trust in recommendation systems: A comparison of content-based, collaborative and demographic filtering. In *Proceedings of the 2022 CHI conference on human factors in computing systems* (pp. 1-14).

11. Wang, W., Feng, F., Nie, L., & Chua, T. S. (2022, July). User-controllable recommendation against filter bubbles. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 1251-1261).
12. Hsee, C. K. (1999). Value seeking and prediction-decision inconsistency: Why don't people take what they predict they'll like the most? *Psychonomic Bulletin & Review*, 6(4), 555-561.
13. The Mind Behind the Stars: Decoding User Psychology in Recommendation Systems. URL: <https://medium.com/beyond-the-build/exploring-the-intricacies-of-psychology-in-user-preference-and-ratings-within-recommender-systems-bde83c9bb80a> (дата звернення 01.05.2026).
14. Elahi, A., & Zirak, A. (2024). Online and offline evaluations of collaborative filtering and content based recommender systems. *arXiv preprint arXiv:2411.01354*.
15. Understanding Hold-Out Methods for Training Machine Learning Models. URL: <https://www.comet.com/site/blog/understanding-hold-out-methods-for-training-machine-learning-models> (дата звернення 01.05.2026).
16. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Tools for fast metric data search in structural methods for image classification, *IEEE Access*, 10, pp. 124738-124746.
17. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.
18. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 5-12.

19. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.
20. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.
21. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125.
22. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.
23. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine*, pp. 25-27.
24. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, *Computers, Materials & Continua*, vol. 80, no. 2, pp. 3085-3106.
25. Gorokhovatskyi V., Chmutov Y., Tvoroshenko I., and Kobylin O. (2025) Reducing computational costs by compressing the structural description in image classification methods, *Advanced Information Systems*, vol. 9, no. 1, pp. 5–12.

26. Bohdan N., Tvoroshenko I., Gorokhovatskyi V., and Kobylin O. (2025) Development of a hybrid method to enhance context memory for a chatbot application based on large language models, *International Journal of Academic Information Systems Research*, 9(10), pp. 7-18.

27. Suprun A., Tvoroshenko I., Gorokhovatskyi V., and Yakovleva O. (2025) Development and research of a method for the combined use of large language models for text generation, *International Journal of Academic and Applied Research*, 9(10), pp. 249-263.

28. Гороховатський В.О., Творошенко І.С. (2025) Оцінювання значущості ознак для підвищення продуктивності структурних методів класифікації зображень. *Проблеми інформатики та моделювання (ПІМ-2025). Тези двадцять п'ятої міжнародної науково-технічної конференції (25 – 28 вересня 2025 року)*. Харків: НТУ «ХПІ», С. 38-43.

29. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2026) Feature space quantization and application of metrics in structural methods of image recognition, *IEEE Access*, vol. 14, pp. 17812-17824.

30. Гороховатський В., Творошенко І. (2026) Продуктивні моделі аналізу даних у методах розпізнавання зображень: монографія. Харків: ХНУРЕ. 153 с.

31. Красюков М.С. (2026) Аналіз взаємодії алгоритмів Content-Based та Collaborative Filtering у рекомендаційних комп'ютерних системах. *Матеріали 30-го Міжнародного молодіжного форуму «Радіoeлектроніка і молодь у XXI столітті». Конференція «Комп'ютерний зір, системний аналіз та математичне моделювання» (22 – 24 квітня 2026 року)*. Харків: ХНУРЕ, С. 89–91.