МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «**РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ОЦІНКИ ТА РЕЦЕНЗУВАННЯ ФІЛЬМІВ ІЗ ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ**»

Виконав: студент 4 курсу, групи 6.1212-пі-3
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)
освітньої програми програмна інженерія
(назва освітньої програми)
В.М. Барановський
(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії, доцент, к.ф.-м.н. Горбенко В.І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри комп'ютерних наук, доцент, к.т.н. Борю С.Ю.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя – 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення
(шифр і назва)

Освітня програма програмна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри програмної інженерії, к.ф.-м.н., доцент

_____ Лісняк А.О.
(підпис)

“ ____ ” _____ 2025 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Барановському Владиславу Миколайовичу
(прізвище, ім’я та по-батькові)

1. Тема роботи Розробка вебзастосунку для оцінки та рецензування фільмів із використанням штучного інтелекту

керівник роботи Горбенко Віталій Іванович, к.ф.-м.н., доцент
(прізвище, ім’я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 02 » грудня 2025 року № 1741-с

2. Строк подання студентом роботи 01.06.2026 р.

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Аналіз предметної області та сучасних технологій розробки вебзастосунків.
2. Проєктування вебзастосунку.
3. Розробка та тестування вебзастосунку.

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація за темою доповіді

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 09.12.2025 р.

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	19.12.2025	
2.	Збір вихідних даних.	09.02.2026	
3.	Обробка методичних та теоретичних	23.02.2026	
	джерел.		
4.	Розробка першого та другого розділу.	18.03.2026	
5.	Розробка третього розділу.	14.05.2026	
6.	Оформлення та нормоконтроль	25.05.2026	
	кваліфікаційної роботи бакалавра.		
7.	Захист кваліфікаційної роботи.	18.06.2026	

Студент _____ В.М. Барановський
(підпис) (ініціали та прізвище)

Керівник роботи _____ В.І. Горбенко
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ А.В. Столярова
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка вебзастосунку для оцінки та рецензування фільмів із використанням штучного інтелекту»: 57 с., 12 рис., 6 табл., 15 джерел.

ВЕБЗАСТОСУНОК, ОЦІНЮВАННЯ ФІЛЬМІВ, РЕКОМЕНДАЦІЙНА СИСТЕМА, РЕЦЕНЗІЇ, ШТУЧНИЙ ІНТЕЛЕКТ, AI-МОДУЛЬ, EXPRESS.JS, MSSQL, NODE.JS, REACT, REST API, TMDB API.

Об’єкт дослідження – процеси зберігання, обробки та аналізу інформації про фільми, а також формування персоналізованих рекомендацій у вебзастосунках кінематографічної тематики.

Мета роботи – розробка вебзастосунку для рецензування та оцінки фільмів із використанням технологій штучного інтелекту для формування персоналізованих рекомендацій користувачам.

Методи дослідження – аналіз предметної області, порівняльний аналіз існуючих платформ для оцінювання фільмів, методи об’єктно-орієнтованого проєктування, моделювання засобами UML, методи веброзробки, тестування програмного забезпечення та аналіз ефективності рекомендаційних систем.

У роботі проведено аналіз сучасних платформ для рецензування та оцінки фільмів, досліджено принципи функціонування рекомендаційних систем та можливості використання штучного інтелекту для персоналізації контенту. На основі проведеного аналізу спроєктовано архітектуру вебзастосунку, структуру бази даних та модуль формування рекомендацій. У процесі виконання роботи реалізовано вебзастосунок із використанням технологій React, Node.js, Express.js та Microsoft SQL Server. Забезпечено функціонал перегляду інформації про фільми, створення рецензій, виставлення оцінок та отримання персоналізованих рекомендацій.

SUMMARY

Bachelor's qualification paper "Design and Development of a Web Application for Movie Evaluation and Review Using Artificial Intelligence": 57 pages, 12 figures, 6 tables, 15 references.

AI MODULE, ARTIFICIAL INTELLIGENCE, EXPRESS.JS, MOVIE RATING, MOVIE REVIEWS, MSSQL, NODE.JS, REACT, RECOMMENDATION SYSTEM, REST API, TMDB API, WEB APPLICATION.

Object of research includes processes of storing, processing and analyzing movie-related information, as well as generating personalized recommendations in movie-oriented web applications.

Purpose of the thesis is the development of a web application for movie reviewing and rating using artificial intelligence technologies to provide personalized movie recommendations for users.

Research methods involve subject area analysis, comparative analysis of existing movie rating platforms, object-oriented design methods, UML modeling, web development technologies, software testing methods, and recommendation system efficiency analysis.

The thesis analyzes modern movie review and rating platforms, investigates recommendation systems and the possibilities of applying artificial intelligence technologies for content personalization. Based on the conducted analysis, the architecture of the web application, database structure, and recommendation module were designed. During the implementation stage, a web application based on React, Node.js, Express.js, and Microsoft SQL Server was developed. The system provides functionality for browsing movie information, creating reviews, assigning ratings, and receiving personalized recommendations. An AI module was developed to analyze user ratings and reviews in order to generate personalized recommendations.

ЗМІСТ

Завдання на кваліфікаційну роботу 2
Реферат 4
Summary 5
Вступ 7
1 Аналіз предметної області та сучасних технологій розробки вебзастосунків .. 10
1.1 Аналіз сучасних платформ для рецензування та оцінки фільмів 10
1.2 Дослідження рекомендаційних систем та можливостей використання штучного інтелекту 14
1.3 Аналіз технологій React, Node.js, Express.js, MSSQL та TMDB API..... 18
2 Проєктування вебзастосунку 23
2.1 Проєктування архітектури вебзастосунку 23
2.2 Проєктування структури бази даних та серверної частини..... 27
2.3 Проєктування AI-модуля формування рекомендацій 35
3 Розробка та тестування вебзастосунку 39
3.1 Реалізація функціоналу..... 39
3.2 Інтеграція штучного інтелекту для формування рекомендацій..... 42
3.3 Тестування вебзастосунку та аналіз ефективності AI-рекомендацій..... 48
Висновки 53
Перелік посилань..... 56

ВСТУП

Стрімкий розвиток інформаційних технологій та широке поширення цифрових сервісів суттєво змінили способи отримання, обробки та поширення інформації. Особливої популярності набули вебзастосунки, які забезпечують користувачам швидкий доступ до різноманітного контенту незалежно від місця перебування чи типу пристрою. Однією з найбільш затребуваних сфер використання вебтехнологій є індустрія розваг, зокрема сервіси для перегляду інформації про фільми, виставлення оцінок та публікації рецензій [1]. Такі платформи дозволяють користувачам ділитися власними враженнями, формувати рейтинги та знаходити новий контент відповідно до своїх уподобань. Зростання обсягів доступної інформації створює потребу в інструментах, які здатні ефективно допомагати користувачам у виборі фільмів. Саме тому особливого значення набувають рекомендаційні системи та технології персоналізації контенту [2]. Вони забезпечують підвищення зручності використання сервісів та покращують загальний користувацький досвід [4].

Сучасні платформи для рецензування та оцінки фільмів, такі як IMDb, Rotten Tomatoes та Letterboxd, надають широкий набір функціональних можливостей для взаємодії користувачів із кіноконтентом. Водночас більшість таких систем використовує традиційні механізми рекомендацій, які не завжди враховують індивідуальні особливості користувача [7]. У результаті рекомендований контент може бути недостатньо релевантним або не відповідати поточним інтересам аудиторії. Розвиток технологій штучного інтелекту відкрив нові можливості для підвищення якості рекомендаційних сервісів. Використання алгоритмів аналізу даних дозволяє враховувати історію оцінок, рецензії, жанрові вподобання та інші фактори поведінки користувача. Це забезпечує більш точне прогнозування інтересів та формування персоналізованих рекомендацій. Саме тому інтеграція штучного

інтелекту у вебзастосунки кінематографічної тематики є актуальним напрямом досліджень.

Актуальність теми бакалаврської роботи обумовлена необхідністю створення сучасних вебзастосунків, які поєднують функціональність інформаційних систем для оцінювання фільмів із можливостями інтелектуального аналізу даних. Використання штучного інтелекту дозволяє підвищити рівень персоналізації сервісу та забезпечити більш ефективну взаємодію користувача з інформаційною системою. Крім того, застосування сучасних вебтехнологій забезпечує високу швидкодію, масштабованість та зручність використання програмного забезпечення. В умовах постійного збільшення кількості цифрового контенту персоналізація стає одним із ключових чинників успішності інформаційних сервісів. Тому розробка вебзастосунку для рецензування та оцінки фільмів із використанням штучного інтелекту є актуальним та практично значущим завданням. Результати дослідження можуть бути використані під час створення аналогічних рекомендаційних систем у сфері цифрових сервісів.

Об’єктом дослідження є процеси зберігання, обробки та аналізу інформації про фільми, а також взаємодія користувачів із системами оцінювання та рецензування кіноконтенту.

Предметом дослідження є методи та засоби розробки вебзастосунків для рецензування та оцінки фільмів із використанням технологій штучного інтелекту для формування персоналізованих рекомендацій.

Метою роботи є розробка вебзастосунку для рецензування та оцінки фільмів, який забезпечує можливість перегляду інформації про фільми, публікації рецензій та оцінок, а також формування персоналізованих рекомендацій на основі технологій штучного інтелекту.

Для досягнення поставленої мети, необхідно вирішити такі **завдання**:

- провести аналіз сучасних платформ для рецензування та оцінки фільмів;
- дослідити принципи роботи рекомендаційних систем та можливості використання штучного інтелекту;

- виконати аналіз сучасних технологій веброзробки;
- спроєктувати архітектуру вебзастосунку;
- розробити структуру бази даних та серверну частину системи;
- реалізувати клієнтську частину вебзастосунку;
- інтегрувати модуль штучного інтелекту для формування рекомендацій;
- провести тестування програмного забезпечення та оцінити ефективність роботи рекомендаційної системи.

Практичне значення одержаних результатів полягає у створенні працездатного вебзастосунку, який може бути використаний для організації сервісу рецензування та оцінювання фільмів із функціями інтелектуальної підтримки користувачів. Розроблене програмне забезпечення дозволяє автоматизувати процес формування рекомендацій, підвищити якість взаємодії користувачів із контентом та забезпечити можливість подальшого розвитку системи шляхом інтеграції нових алгоритмів аналізу даних і технологій штучного інтелекту.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СУЧАСНИХ ТЕХНОЛОГІЙ РОЗРОБКИ ВЕБЗАСТОСУНКІВ

1.1 Аналіз сучасних платформ для рецензування та оцінки фільмів

Сучасний розвиток інформаційних технологій сприяв появі великої кількості вебплатформ, призначених для перегляду інформації про фільми, їх оцінювання та обговорення [1]. Такі системи стали важливим інструментом для користувачів, які прагнуть отримати додаткову інформацію про кінопродукцію перед переглядом. Завдяки розвитку мережі Інтернет та цифрових технологій користувачі отримали можливість швидко знаходити відомості про фільми, переглядати рейтинги, читати рецензії та формувати власну думку щодо конкретного контенту. Значна частина таких сервісів функціонує за принципом соціальних платформ, де користувачі можуть взаємодіяти між собою через коментарі, оцінки та рекомендації. З кожним роком кількість кіноконтенту стрімко збільшується, що ускладнює процес вибору фільмів для перегляду. Саме тому сервіси для рецензування та оцінювання відіграють важливу роль у процесі пошуку релевантного контенту. Вони дозволяють користувачам швидко отримувати узагальнену інформацію про популярність та якість фільмів. Крім того, такі системи формують інформаційне середовище для обговорення сучасної кіноіндустрії.

Однією з найбільш відомих платформ у сфері кінематографії є IMDb (Internet Movie Database). Даний сервіс містить одну з найбільших баз даних про фільми, серіали, акторів та режисерів. Користувачі можуть переглядати рейтинги, залишати власні оцінки та читати рецензії інших відвідувачів ресурсу. Перевагою IMDb є значний обсяг інформації та висока популярність серед користувачів у всьому світі. Крім цього, система використовує власний алгоритм формування рейтингів, який враховує велику кількість оцінок та забезпечує їх об’єктивність. Разом із тим сервіс має певні обмеження щодо персоналізації контенту. Рекомендації формуються переважно на основі

загальних показників популярності та схожості фільмів. Це може знижувати рівень індивідуальної відповідності рекомендацій конкретному користувачу.

Ще однією популярною платформою є Rotten Tomatoes, яка спеціалізується на зборі професійних та користувацьких рецензій. Особливістю даного сервісу є використання показника Tomatometer, що формується на основі оцінок кінокритиків. Такий підхід дозволяє отримувати більш професійне бачення якості кінопродукції. Користувачі також можуть переглядати рейтинги аудиторії та залишати власні відгуки. Платформа активно використовується як джерело інформації під час вибору фільмів для перегляду. Водночас механізми персоналізації на Rotten Tomatoes реалізовані досить обмежено. Основна увага приділяється загальним рейтингам та експертним оцінкам. Це створює потребу в більш гнучких рекомендаційних механізмах, орієнтованих на конкретного користувача.

Серед платформ також варто виділити сервіс Letterboxd, який поєднує функціональність соціальної мережі та системи оцінювання фільмів. Користувачі можуть вести власні щоденники переглядів, створювати тематичні списки та публікувати детальні рецензії. Такий підхід сприяє формуванню активної спільноти кінолюбителів та забезпечує високий рівень залучення аудиторії. Однією з переваг Letterboxd є зручний інтерфейс та орієнтація на взаємодію між користувачами. Разом із тим функціональність сервісу щодо інтелектуального аналізу вподобань користувачів залишається обмеженою [1]. Рекомендації формуються переважно на основі базових алгоритмів та статистичних даних. Це не дозволяє повною мірою враховувати індивідуальні особливості поведінки користувачів. У результаті персоналізація контенту може бути недостатньо ефективною.

Аналіз розглянутих платформ свідчить про те, що більшість сучасних сервісів орієнтована на надання інформації про фільми, збір оцінок та публікацію рецензій [4]. При цьому основні механізми рекомендацій здебільшого базуються на популярності контенту, жанровій подібності або статистичних алгоритмах [7]. Такі підходи забезпечують прийнятний рівень рекомендацій, однак не враховують складні поведінкові характеристики

користувачів [5]. У сучасних умовах цього вже недостатньо для забезпечення максимальної релевантності рекомендацій. Саме тому активно розвиваються підходи, що передбачають використання технологій штучного інтелекту. Вони дозволяють аналізувати великі обсяги даних та виявляти приховані закономірності у вподобаннях користувачів. Це створює передумови для підвищення точності рекомендаційних систем. Крім того, штучний інтелект забезпечує можливість автоматичного адаптування рекомендацій до змін інтересів користувачів.

Для наочного порівняння функціональних можливостей розглянутих платформ доцільно виконати їх порівняльний аналіз за ключовими характеристиками. Основними критеріями оцінювання було обрано наявність системи оцінювання, можливість публікації рецензій, підтримку рекомендаційних механізмів, використання технологій штучного інтелекту та рівень персоналізації контенту. Результати проведеного аналізу наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика платформ для рецензування та оцінки фільмів

Характеристика	**IMDb**	**Rotten Tomatoes**	**Letterboxd**	**Розроблюваний вебзастосунок**
Перегляд інформації про фільми	Так	Так	Так	Так
Система оцінювання	Так	Так	Так	Так
Публікація рецензій	Так	Так	Так	Так
Персональні рекомендації	Частково	Частково	Частково	Так
Використання ШІ	Ні	Ні	Ні	Так
Аналіз вподобань користувача	Частково	Ні	Частково	Так
Інтеграція із зовнішніми API	Так	Так	Ні	Так
Можливість розширення	Середня	Середня	Середня	Висока

Отримані дані дозволяють визначити переваги та недоліки існуючих рішень. Крім того, вони є основою для формування вимог до розроблюваного вебзастосунку. Особливу увагу приділено можливостям інтеграції штучного інтелекту, які в сучасних платформах реалізовані недостатньо повно. Це

дозволяє обґрунтувати актуальність створення нового програмного продукту. Розроблювана система повинна усунути виявлені недоліки та забезпечити більш високий рівень персоналізації.

На основі проведеного аналізу можна зробити висновок, що існуючі платформи для рецензування та оцінки фільмів забезпечують широкий набір функцій для роботи з кіноконтентом. Вони дозволяють користувачам отримувати інформацію про фільми, оцінювати їх та обмінюватися думками з іншими учасниками спільноти. Проте більшість таких систем недостатньо використовує можливості сучасних технологій штучного інтелекту. У результаті рівень персоналізації рекомендацій часто не відповідає очікуванням користувачів. Це створює передумови для розробки нового вебзастосунку, який поєднуватиме функціональність платформ для рецензування з інтелектуальними механізмами формування рекомендацій.

Запропонований підхід дозволить підвищити релевантність запропонованого контенту та покращити якість користувацького досвіду. Таким чином, розробка вебзастосунку для рецензування та оцінки фільмів із використанням штучного інтелекту є актуальним та перспективним напрямом розвитку інформаційних систем.

Серед сучасних рішень, які використовують штучний інтелект для рекомендації кіноконтенту, особливий інтерес становлять Netflix, Amazon Prime Video та YouTube.

Netflix використовує гібридну рекомендаційну систему, що поєднує колаборативну фільтрацію, контентний аналіз та моделі машинного навчання. Алгоритми сервісу аналізують історію переглядів, оцінки користувачів, час перегляду та поведінкові характеристики.

Amazon Prime Video застосовує рекомендаційні моделі, які враховують історію переглядів, покупки та інтереси користувачів.

YouTube використовує багаторівневу нейронну мережу для прогнозування відео, які можуть зацікавити користувача. Усі зазначені платформи активно використовують технології машинного навчання для

підвищення рівня персоналізації контенту. Аналіз цих рішень показує, що сучасні рекомендаційні системи дедалі частіше базуються на використанні штучного інтелекту та аналізі великих обсягів даних.

Окремий напрям розвитку рекомендаційних систем пов’язаний з аналізом текстових рецензій користувачів. Для цього використовуються методи обробки природної мови (Natural Language Processing, NLP). Сучасні мовні моделі, зокрема BERT, RoBERTa та GPT, дозволяють визначати емоційне забарвлення тексту, виявляти ключові теми та аналізувати ставлення користувача до окремих характеристик фільму. Наприклад, модель BERT успішно використовується для задач класифікації тональності відгуків та демонструє високі показники точності.

Використання NLP-моделей дозволяє отримувати додаткову інформацію про вподобання користувачів, яка відсутня у звичайних числових оцінках. Саме тому в сучасних рекомендаційних системах дедалі частіше використовуються методи аналізу текстових рецензій як додаткове джерело даних для персоналізації рекомендацій.

1.2 Дослідження рекомендаційних систем та можливостей використання штучного інтелекту

У сучасних інформаційних системах рекомендаційні механізми відіграють важливу роль у забезпеченні ефективної взаємодії користувачів із цифровим контентом. Зі збільшенням обсягів інформації користувачам стає дедалі складніше самостійно знаходити матеріали, які відповідають їхнім інтересам та вподобанням [2]. Саме тому рекомендаційні системи стали невід’ємною складовою багатьох вебресурсів, онлайн-магазинів, соціальних мереж та мультимедійних платформ. Основним завданням таких систем є автоматичний підбір найбільш релевантного контенту для конкретного користувача.

Для досягнення цієї мети використовуються різноманітні алгоритми аналізу даних та прогнозування поведінки користувачів. Результатом їх роботи є формування індивідуальних рекомендацій, які враховують попередню активність користувача. Завдяки цьому підвищується рівень задоволеності аудиторії та покращується користувацький досвід. Особливо актуальним використання рекомендаційних систем є у сфері кінематографії, де кількість доступного контенту постійно зростає.

Рекомендаційні системи класифікуються за різними принципами та підходами до аналізу інформації. Одним із найпоширеніших методів є контентна фільтрація, яка базується на характеристиках об’єктів та історії взаємодії користувача з ними [7]. У межах цього підходу аналізуються жанри фільмів, ключові слова, акторський склад, режисери та інші атрибути контенту. На основі отриманих даних система формує перелік схожих об’єктів, які можуть зацікавити користувача. Перевагою такого підходу є можливість отримання рекомендацій навіть за відсутності великої кількості користувацьких оцінок. Водночас контентна фільтрація має певні обмеження, оскільки пропонує переважно схожий контент і не враховує інтереси інших користувачів. У результаті рекомендації можуть ставати одноманітними та менш різноманітними. Незважаючи на це, даний метод залишається важливою складовою багатьох сучасних рекомендаційних систем.

Іншим популярним підходом є колаборативна фільтрація, яка ґрунтується на аналізі поведінки великої кількості користувачів [7]. У цьому випадку система порівнює оцінки, перегляди та інші дії різних користувачів для виявлення схожих моделей поведінки. Якщо декілька користувачів мають подібні вподобання, система може рекомендувати їм однаковий контент. Головною перевагою колаборативної фільтрації є здатність знаходити неочевидні зв’язки між об’єктами та інтересами користувачів. Завдяки цьому рекомендації стають більш різноманітними та персоналізованими. Проте ефективність такого підходу значною мірою залежить від обсягу накопичених даних. На початкових етапах функціонування системи може виникати

проблема недостатньої кількості інформації для формування якісних рекомендацій. Даний недолік відомий як проблема «холодного старту».

У сучасних рекомендаційних системах дедалі частіше використовуються гібридні підходи, які поєднують переваги контентної та колаборативної фільтрації [7]. Такі рішення дозволяють компенсувати недоліки окремих алгоритмів та підвищити якість рекомендацій. Гібридні системи можуть одночасно враховувати характеристики контенту, історію взаємодії користувача та поведінкові особливості інших користувачів. Це забезпечує більш комплексний аналіз даних та покращує точність прогнозування інтересів. Використання декількох джерел інформації сприяє формуванню більш релевантних рекомендацій. Крім того, гібридні моделі дозволяють ефективніше вирішувати проблему холодного старту та забезпечують кращу адаптацію до змін уподобань користувачів. Саме такі підходи сьогодні використовуються у найбільших рекомендаційних сервісах світу. Їхня ефективність підтверджується успішним використанням у різних галузях цифрової економіки.

Подальший розвиток рекомендаційних систем став можливим завдяки впровадженню технологій штучного інтелекту та машинного навчання. На відміну від традиційних алгоритмів, сучасні AI-рішення здатні самостійно виявляти приховані закономірності у великих масивах даних [8]. Це дозволяє значно підвищити якість персоналізації контенту та адаптувати рекомендації до індивідуальних потреб користувачів. Алгоритми машинного навчання можуть аналізувати не лише оцінки користувачів, але й текстові рецензії, історію переглядів та поведінкові фактори. У результаті система отримує більш повне уявлення про вподобання аудиторії. Завдяки цьому рекомендації стають більш точними та корисними. Крім того, штучний інтелект забезпечує можливість автоматичного вдосконалення рекомендаційної моделі в процесі її використання. Це є важливою перевагою сучасних інтелектуальних систем.

Особливий інтерес становить використання великих мовних моделей для аналізу текстових рецензій користувачів [8]. Такі моделі здатні визначати

емоційне забарвлення тексту, виявляти основні теми відгуків та аналізувати ставлення користувача до певних характеристик фільму. Це дозволяє формувати рекомендації не лише на основі числових оцінок, але й з урахуванням змісту текстових коментарів. Наприклад, система може визначити, що користувач віддає перевагу психологічним трилерам із глибоким сюжетом та складними персонажами. Отримані результати використовуються для пошуку найбільш релевантного контенту. Такий підхід значно розширює можливості традиційних рекомендаційних систем. Він забезпечує більш глибоке розуміння інтересів користувачів та підвищує якість персоналізації. Саме тому технології обробки природної мови активно впроваджуються в сучасні інформаційні системи.

У межах даної роботи штучний інтелект використовується для формування персоналізованих рекомендацій фільмів на основі оцінок, рецензій та історії взаємодії користувача із системою. Додатковим джерелом інформації виступають дані, отримані за допомогою TMDB API, які містять відомості про жанри, рейтинги та пов’язані фільми [9].

Використання AI-модуля дозволяє поєднати переваги традиційних рекомендаційних алгоритмів із сучасними методами аналізу даних. У результаті користувач отримує більш точні рекомендації, які відповідають його індивідуальним інтересам. Такий підхід сприяє підвищенню ефективності роботи вебзастосунку та покращенню взаємодії з кіноконтентом. Крім того, система може адаптуватися до зміни вподобань користувача в процесі експлуатації. Це забезпечує актуальність рекомендацій протягом усього періоду використання програмного продукту. Отже, інтеграція штучного інтелекту є важливим елементом сучасних рекомендаційних сервісів.

Для наочного відображення процесу формування рекомендацій у розроблюваному вебзастосунку доцільно побудувати структурну схему взаємодії користувача, бази даних, зовнішнього API та AI-модуля (див. рис. 1.1).

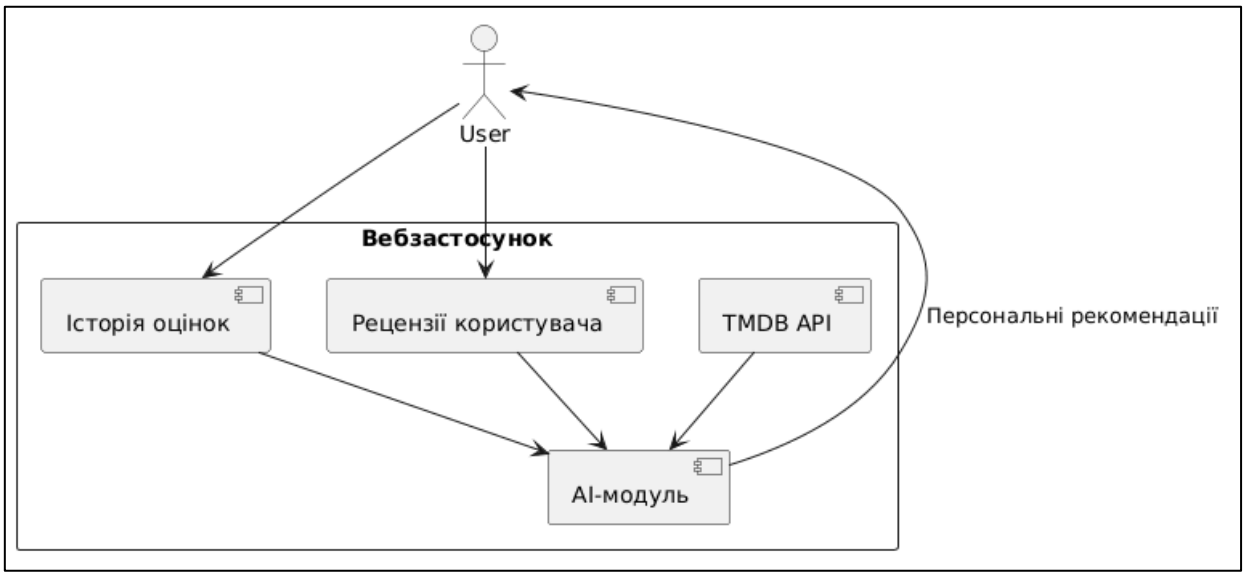


Рисунок 1.1 – Формування рекомендацій із використанням штучного інтелекту

Така схема демонструє основні етапи збору та обробки інформації, що використовуються для генерації рекомендацій. Вона дозволяє наочно показати місце штучного інтелекту в загальній архітектурі системи та його взаємозв’язок з іншими компонентами вебзастосунку.

Отримані результати аналізу підтверджують доцільність використання технологій штучного інтелекту для підвищення ефективності рекомендаційних систем. Використання сучасних AI-підходів дозволяє покращити якість персоналізації та підвищити задоволеність користувачів. Це є одним із ключових напрямів розвитку інформаційних систем у сфері цифрових сервісів. Саме тому інтеграція штучного інтелекту є важливою складовою розроблюваного вебзастосунку для рецензування та оцінки фільмів.

1.3 Аналіз технологій React, Node.js, Express.js, MSSQL та TMDB API

Під час розробки сучасних вебзастосунків важливе значення має правильний вибір програмних засобів та технологій, які забезпечують

необхідний рівень продуктивності, масштабованості та зручності підтримки програмного забезпечення. Вибір технологічного стеку безпосередньо впливає на швидкість розробки, якість реалізації функціональних можливостей та можливість подальшого розвитку системи.

Для створення вебзастосунку для рецензування та оцінювання фільмів було обрано комплекс сучасних технологій, які широко використовуються у веброзробці та мають значну підтримку спільноти розробників. До складу технологічного стеку увійшли React для клієнтської частини, Node.js та Express.js для серверної логіки, Microsoft SQL Server для зберігання даних і TMDB API для отримання інформації про фільми.

Кожна з цих технологій виконує власну функцію та забезпечує реалізацію окремих компонентів програмного продукту. Разом вони формують єдину інформаційну систему, здатну забезпечити ефективну взаємодію користувачів із кіноконтентом. Використання популярних технологій також спрощує процес супроводу та модернізації програмного забезпечення. Саме тому їх вибір є обґрунтованим для реалізації поставлених завдань.

Для реалізації клієнтської частини вебзастосунку було обрано бібліотеку React. Дана технологія була розроблена компанією Meta та сьогодні є одним із найпопулярніших інструментів для створення користувацьких інтерфейсів. Основною перевагою React є компонентний підхід до розробки, який дозволяє розділяти інтерфейс на незалежні багаторазово використовувані елементи. Це значно спрощує процес розробки та підтримки програмного забезпечення [11]. React використовує механізм Virtual DOM, який забезпечує швидке оновлення сторінок та покращує продуктивність вебзастосунків. Завдяки цьому користувачі отримують швидкий та плавний інтерфейс без необхідності повного перезавантаження сторінки. Крім того, React має розвинену екосистему бібліотек та інструментів для створення складних вебсистем. У межах даної роботи React використовується для реалізації інтерфейсу перегляду фільмів, сторінок авторизації, рецензування та формування рекомендацій.

Для реалізації серверної частини було використано середовище виконання Node.js. Дана технологія дозволяє виконувати JavaScript-код на сервері та забезпечує ефективну обробку великої кількості одночасних запитів. Основною особливістю Node.js є використання неблокуючої асинхронної моделі введення-виведення. Завдяки цьому сервер здатний обслуговувати велику кількість користувачів із мінімальними витратами системних ресурсів. Node.js широко застосовується під час створення вебзастосунків реального часу, API-сервісів та високонавантажених інформаційних систем. Ще однією перевагою є використання однієї мови програмування як для клієнтської, так і для серверної частини системи. Це спрощує процес розробки та дозволяє зменшити витрати часу на підтримку програмного забезпечення. Саме тому Node.js було обрано як основу для реалізації серверної логіки вебзастосунку.

Для спрощення створення серверного API використовується фреймворк Express.js. Він є одним із найпоширеніших інструментів для розробки вебзастосунків на платформі Node.js. Express.js забезпечує набір механізмів для маршрутизації запитів, обробки даних та взаємодії між компонентами системи. Завдяки мінімалістичній архітектурі фреймворк дозволяє швидко створювати REST API та забезпечує високу гнучкість під час розробки. Express.js має простий синтаксис та підтримує інтеграцію великої кількості сторонніх модулів. Це дозволяє легко реалізовувати функціональність авторизації, обробки помилок та взаємодії з базами даних. У розроблюваному вебзастосунку Express.js використовується для реалізації серверних маршрутів, взаємодії з клієнтською частиною та забезпечення доступу до даних. Використання цього фреймворку дозволило створити масштабовану та зручну для підтримки серверну архітектуру.

Для зберігання інформації про користувачів, рецензії, оцінки та рекомендації було обрано систему керування базами даних Microsoft SQL Server. Дана СКБД є одним із провідних рішень для роботи з реляційними базами даних та широко використовується в корпоративних інформаційних

системах. Основною перевагою MSSQL є висока надійність, продуктивність та підтримка механізмів забезпечення цілісності даних. Система підтримує транзакції, резервне копіювання та засоби контролю доступу до інформації. Це дозволяє гарантувати безпечне зберігання даних користувачів та стабільну роботу програмного забезпечення. Крім того, MSSQL має зручні інструменти адміністрування та підтримує виконання складних SQL-запитів. У межах даного проєкту база даних використовується для зберігання інформації про користувачів, фільми, оцінки та текстові рецензії. Використання реляційної моделі даних забезпечує ефективну організацію структури інформації та швидкий доступ до неї.

Важливим компонентом розроблюваного вебзастосунку є інтеграція із зовнішнім сервісом TMDB API. The Movie Database є однією з найбільших відкритих баз даних кінематографічного контенту. Даний сервіс надає доступ до великого обсягу інформації про фільми, серіали, акторів, режисерів та інші пов’язані об’єкти. За допомогою API можна отримувати описи фільмів, рейтинги, жанри, зображення та інші метадані [9]. Це дозволяє уникнути необхідності самостійного наповнення бази даних великою кількістю контенту. Інтеграція TMDB API забезпечує актуальність інформації та автоматичне оновлення даних про нові кінорелізи. Крім того, сервіс підтримує механізми пошуку та рекомендацій, які можуть використовуватися під час формування персоналізованого контенту. Завдяки цьому TMDB API є важливим джерелом даних для реалізації функціональних можливостей вебзастосунку.

Для наочного представлення характеристик обраних технологій доцільно виконати їх порівняльний аналіз. Основними критеріями оцінювання виступають призначення технології, її переваги та роль у структурі програмного забезпечення. Результати аналізу наведено в таблиці 1.2.

Отримані дані дозволяють обґрунтувати доцільність використання кожного компонента технологічного стеку. Крім того, таблиця демонструє взаємозв’язок між окремими технологіями та їх внесок у реалізацію функціональних можливостей вебзастосунку.

Таблиця 1.2 – Технології, використані при розробці вебзастосунку

Технологія	**Призначення**	**Переваги**
React	Розробка клієнтської частини	Компонентна архітектура, швидкодія
Node.js	Серверне середовище виконання	Висока продуктивність
Express.js	Розробка REST API	Простота та масштабованість
MSSQL	Система керування БД	Надійність та безпечність
TMDB API	Отримання інформації про фільми	Велика база кіноконтенту
JWT	Автентифікація користувачів	Безпечна авторизація
AI API	Формування рекомендацій	Персоналізація контенту

Використання сучасних програмних засобів забезпечує можливість створення конкурентоспроможного програмного продукту. Вони дозволяють реалізувати ефективну архітектуру та забезпечити подальше масштабування системи. Саме тому вибір зазначених технологій є обґрунтованим для реалізації поставлених завдань.

На основі проведеного аналізу можна зробити висновок, що обраний набір технологій повністю відповідає вимогам до розроблюваного вебзастосунку. React забезпечує створення сучасного та зручного користувацького інтерфейсу, Node.js та Express.js реалізують серверну логіку та механізми обробки запитів, а MSSQL гарантує надійне зберігання інформації. Інтеграція із TMDB API дозволяє отримувати актуальні дані про фільми без необхідності створення власної великої бази кіноконтенту. Сукупність цих технологій забезпечує високу продуктивність, масштабованість та зручність подальшої підтримки програмного забезпечення. Крім того, обраний стек технологій добре підходить для інтеграції модулів штучного інтелекту та рекомендаційних систем. Це створює необхідні передумови для реалізації функціоналу персоналізованих рекомендацій у межах даного проєкту. Таким чином, використання React, Node.js, Express.js, MSSQL та TMDB API є оптимальним рішенням для створення вебзастосунку рецензування та оцінки фільмів із використанням штучного інтелекту.

2 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ

2.1 Проєктування архітектури вебзастосунку

Проєктування архітектури вебзастосунку є одним із ключових етапів розробки програмного забезпечення, оскільки саме на цьому етапі визначається загальна структура системи, її основні компоненти та способи їх взаємодії. Для розроблюваного вебзастосунку було обрано клієнт-серверну архітектуру, яка є поширеним підходом у сучасній веброзробці. Такий підхід передбачає розподіл функціональності між клієнтською частиною, серверною частиною та базою даних.

Клієнтська частина відповідає за відображення інтерфейсу користувача та взаємодію з ним. Серверна частина забезпечує обробку запитів, виконання бізнес-логіки та взаємодію з базою даних. База даних використовується для зберігання інформації про користувачів, рецензії, оцінки та інші сутності системи.

Додатковим компонентом архітектури є зовнішній сервіс TMDB API, який використовується для отримання актуальної інформації про фільми. У межах проєктування було визначено, що клієнтська частина вебзастосунку реалізується з використанням бібліотеки React. Вона забезпечує створення динамічного та зручного інтерфейсу користувача, який дозволяє переглядати фільми, відкривати детальну інформацію про них, залишати рецензії та виставляти оцінки. Користувач взаємодіє із системою через браузер, а всі необхідні дані отримуються шляхом надсилання HTTP-запитів до серверної частини. Такий підхід дозволяє відокремити логіку відображення від логіки обробки даних. Це спрощує подальше розширення системи та полегшує її підтримку. Крім того, компонентна структура React дозволяє повторно використовувати окремі елементи інтерфейсу. Це є важливим для створення масштабованого вебзастосунку.

Серверна частина вебзастосунку проєктується на основі Node.js та Express.js. Вона виконує роль проміжного рівня між клієнтською частиною, базою даних, зовнішнім API та AI-модулем. Сервер приймає запити від клієнта, перевіряє їх коректність, звертається до відповідних сервісів та повертає результат у форматі JSON. Такий підхід відповідає принципам REST API, які широко використовуються у сучасних вебзастосунках. Express.js дозволяє організувати маршрутизацію запитів і розділити функціональність системи на окремі контролери та сервіси. Це забезпечує логічну структуру програмного коду та полегшує його подальше тестування. У серверній частині також реалізується автентифікація користувачів та обробка операцій із рецензіями й оцінками. Саме сервер відповідає за інтеграцію AI-модуля у загальну логіку вебзастосунку.

База даних є важливою складовою архітектури, оскільки вона забезпечує зберігання основної інформації, необхідної для роботи вебзастосунку. Для зберігання даних було обрано Microsoft SQL Server, який підтримує реляційну модель даних та забезпечує цілісність інформації. У базі даних зберігаються відомості про користувачів, їх оцінки, рецензії та взаємодію з фільмами. Інформація про самі фільми може частково отримуватися з TMDB API, що дозволяє не дублювати великий обсяг кінематографічних даних у локальній базі. Такий підхід зменшує навантаження на локальне сховище та забезпечує актуальність інформації про фільми. Водночас локальна база необхідна для збереження користувацьких дій, які є основою для формування персональних рекомендацій. Саме ці дані використовуються AI-модулем для аналізу вподобань користувачів. Це дозволяє поєднати зовнішні джерела інформації з внутрішніми даними системи.

Окреме місце в архітектурі займає AI-модуль, призначений для формування рекомендацій фільмів. Він отримує дані про фільм, оцінки користувача, рецензії та додаткову інформацію з TMDB API. На основі цих даних модуль аналізує вподобання користувача та формує список рекомендованих фільмів. Такий підхід дозволяє реалізувати персоналізацію, яка є однією з головних переваг розроблюваного вебзастосунку. AI-модуль

може працювати як із зовнішнім API штучного інтелекту, так і у fallback-режимі, коли рекомендації формуються на основі внутрішнього алгоритму. Це підвищує надійність системи та дозволяє демонструвати функціональність навіть без підключення зовнішнього AI-сервісу. У загальній архітектурі AI-модуль взаємодіє із серверною частиною, а клієнт отримує вже готовий результат. Таким чином, складна логіка рекомендацій приховується від користувача та виконується на сервері.

Загальна архітектура вебзастосунку передбачає послідовну взаємодію між користувачем, клієнтською частиною, сервером, базою даних, TMDB API та AI-модулем. Користувач надсилає запит через інтерфейс React, після чого frontend звертається до REST API серверної частини. Сервер обробляє запит, за необхідності звертається до бази даних або зовнішнього API та формує відповідь. Якщо запит пов’язаний із рекомендаціями, сервер додатково передає дані до AI-модуля.

Після обробки результат повертається клієнтській частині та відображається користувачу у зручному вигляді. Така архітектура забезпечує гнучкість, масштабованість та можливість подальшого розвитку системи. Вона також дозволяє окремо вдосконалювати frontend, backend та AI-компоненти. Це є важливим для підтримки програмного продукту після завершення основної розробки.

Для наочного представлення архітектури вебзастосунку доцільно використати UML-схему компонентів [12, 13]. Вона дозволяє показати основні частини системи та напрямки обміну даними між ними. На схемі користувач взаємодіє з клієнтською частиною, яка через HTTP-запити звертається до серверної частини. Серверна частина, у свою чергу, взаємодіє з базою даних, TMDB API та AI-модулем. Така схема демонструє, що система має модульну структуру та може бути розширена новими функціональними блоками. Наприклад, у майбутньому можна додати модуль аналізу тональності рецензій або модуль адміністративної аналітики. Це підтверджує доцільність обраної архітектури для реалізації вебзастосунку. Загальну схему архітектури наведено на рисунку 2.1.

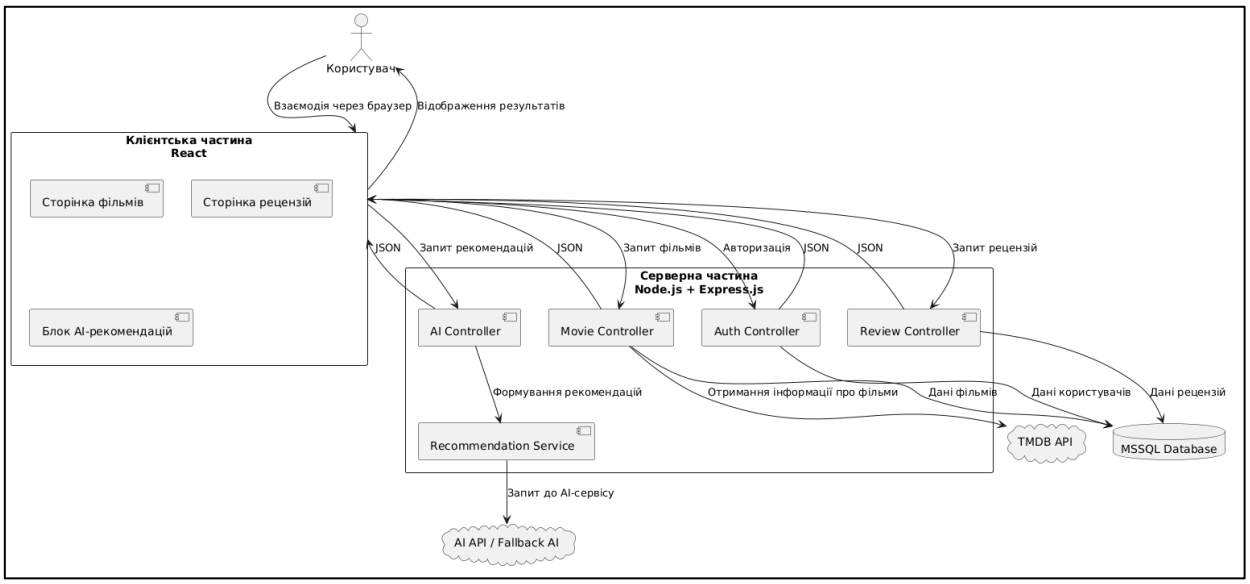


Рисунок 2.1 – Загальна архітектура вебзастосунку для рецензування та оцінки фільмів

Розроблена архітектура вебзастосунку забезпечує чіткий розподіл функціональності між клієнтською та серверною частинами. Використання контролерів і сервісів спрощує підтримку та подальше розширення системи. Інтеграція з TMDB API дозволяє отримувати актуальну інформацію про фільми, а AI-сервіс забезпечує формування персоналізованих рекомендацій для користувачів (див рис. 2.2).

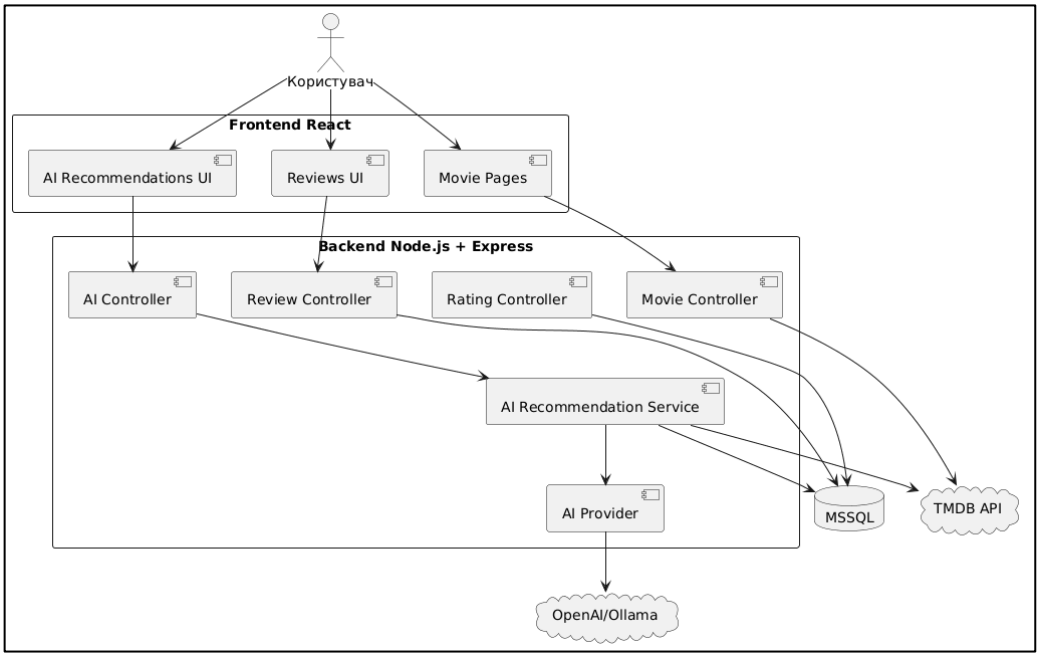


Рисунок 2.2 – Компонентна архітектура вебзастосунку

На рисунку 2.2 наведено компонентну архітектуру вебзастосунку. Діаграма відображає структуру взаємодії між клієнтською частиною, серверною частиною, AI-модулем, базою даних MSSQL та зовнішніми сервісами TMDB API і OpenAI/Ollama. Представлена архітектура демонструє розподіл відповідальності між окремими компонентами системи та забезпечує можливість масштабування програмного забезпечення.

Запропонована структура є масштабованою, надійною та відповідає сучасним вимогам до вебзастосунків.

2.2 Проєктування структури бази даних та серверної частини

Проєктування структури бази даних є важливим етапом створення вебзастосунку, оскільки саме база даних забезпечує зберігання основної інформації системи. У межах розроблюваного програмного продукту необхідно зберігати дані про користувачів, їх рецензії, оцінки та зв’язок із фільмами. Оскільки інформація про фільми значною мірою отримується з TMDB API, локальна база даних не повинна дублювати весь обсяг кінематографічних даних. Натомість у ній зберігаються ідентифікатори фільмів, які використовуються для зв’язку користувацьких дій із конкретними об’єктами із зовнішнього сервісу. Такий підхід дозволяє оптимізувати структуру бази та зменшити обсяг збережених даних. Крім того, це забезпечує актуальність інформації про фільми, оскільки основні метадані отримуються безпосередньо з TMDB. Локальна база виконує роль сховища користувацької активності. Саме ці дані є основою для роботи рекомендаційного AI-модуля.

Основними сутностями бази даних є користувачі, рецензії та оцінки. Сутність користувача містить дані, необхідні для автентифікації та ідентифікації особи в системі. До таких даних належать ім’я користувача, електронна пошта, пароль у захищеному вигляді та службова інформація. Сутність рецензії містить текстовий відгук користувача про конкретний фільм.

Кожна рецензія пов’язується з користувачем та ідентифікатором фільму з TMDB. Сутність оцінки використовується для збереження числової оцінки, яку користувач виставляє фільму. Наявність окремих таблиць для рецензій та оцінок дозволяє гнучко аналізувати поведінку користувача. Це також дає змогу використовувати ці дані для персоналізованих рекомендацій.

Серверна частина вебзастосунку реалізується відповідно до принципів багаторівневої архітектури. Вона складається з маршрутів, контролерів, сервісів, моделей та модуля підключення до бази даних. Маршрути відповідають за визначення адрес API, за якими клієнтська частина може надсилати запити. Контролери обробляють вхідні запити, перевіряють параметри та викликають відповідні сервіси. Сервіси містять основну бізнес-логіку системи, наприклад отримання фільмів, збереження рецензій або формування рекомендацій. Моделі описують структуру таблиць бази даних і використовуються ORM-бібліотекою Sequelize для взаємодії з MSSQL. Такий розподіл відповідальності полегшує підтримку коду та зменшує залежність між окремими компонентами. Крім того, він дозволяє зручно тестувати окремі частини серверної логіки.

Для взаємодії з базою даних використовується ORM Sequelize. Це дозволяє описувати таблиці у вигляді JavaScript-моделей та виконувати операції з даними без написання великої кількості SQL-запитів вручну. Sequelize забезпечує створення зв’язків між моделями, виконання запитів, синхронізацію структури таблиць і перевірку даних. Такий підхід значно спрощує розробку серверної частини та робить код більш зрозумілим. У проєкті модель користувача може бути пов’язана з моделлю рецензії зв’язком «один до багатьох». Це означає, що один користувач може залишити багато рецензій. Аналогічно користувач може виставити багато оцінок різним фільмам. Така структура відповідає логіці роботи системи рецензування та оцінювання фільмів.

Серверна частина також відповідає за інтеграцію з TMDB API. Коли користувач відкриває список фільмів або сторінку конкретного фільму,

frontend надсилає запит до backend. Сервер отримує дані з TMDB API та передає їх клієнтській частині у зручному форматі. Це дозволяє приховати токени доступу до зовнішнього API від клієнта та підвищити безпеку системи. Крім того, backend може додатково обробляти отримані дані перед передачею на frontend. Наприклад, можна залишати лише необхідні поля, об’єднувати дані з локальної бази або додавати інформацію про оцінки користувача. Такий підхід забезпечує централізовану обробку даних. Він також дозволяє в майбутньому змінити зовнішнє API без значних змін у клієнтській частині.

Для наочного представлення структури бази даних доцільно використати ER-схему. Вона демонструє основні сутності системи та зв’язки між ними. На схемі користувач пов’язаний із рецензіями та оцінками. Кожна рецензія та оцінка містить ідентифікатор фільму, який відповідає фільму в TMDB API. Такий підхід дозволяє поєднати локальні дані користувачів із зовнішньою базою фільмів. У результаті система зберігає тільки ті дані, які необхідні для персоналізації та аналізу взаємодії користувача. Це робить структуру бази даних простою, ефективною та придатною для подальшого розширення. Схему бази даних наведено на рисунку 2.3.

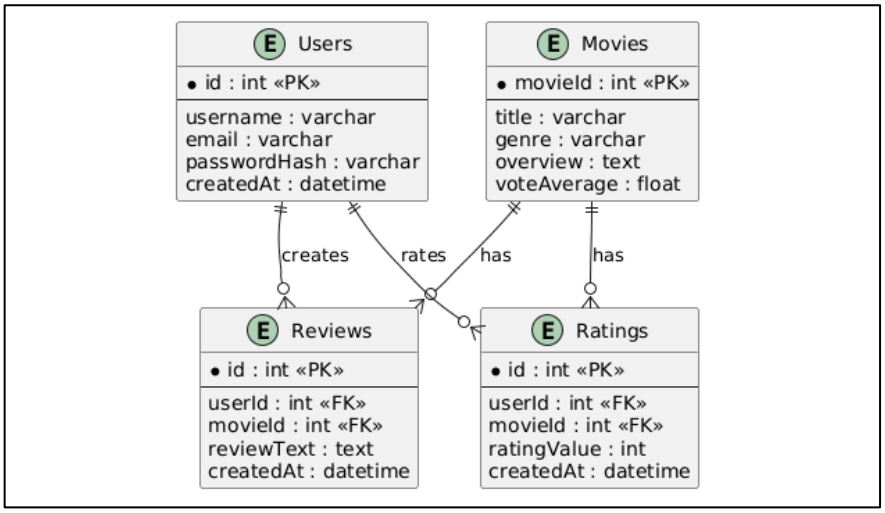


Рисунок 2.3 – Структура бази даних вебзастосунку

Спроєктована структура бази даних забезпечує зберігання інформації про користувачів, рецензії, оцінки та фільми. Визначені зв’язки між

сутностями дозволяють ефективно відстежувати взаємодію користувачів із фільмами через рецензування та оцінювання. Використання окремих таблиць для рецензій і рейтингів сприяє нормалізації даних та підвищує продуктивність роботи системи (див. рис. 2.4).

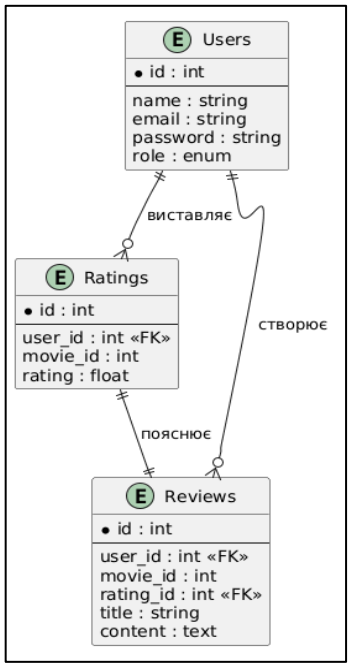


Рисунок 2.4 – ER-діаграма бази даних вебзастосунку

Запропонована модель є гнучкою та може бути розширена додатковими функціями без суттєвих змін у структурі бази даних (див. рис. 2.5).

Наведений фрагмент коду демонструє приклад опису моделі рецензії засобами Sequelize. У моделі визначено унікальний ідентифікатор запису, ідентифікатор користувача, ідентифікатор фільму та текст рецензії.

Поле userId використовується для зв’язку рецензії з конкретним користувачем системи.

Поле movieId зберігає ідентифікатор фільму з TMDB API, що дозволяє пов’язати локальний відгук із зовнішніми даними про фільм.

Поле reviewText містить текстову рецензію, яку користувач залишає після перегляду фільму.

Параметр timestamps дозволяє автоматично зберігати дату створення та оновлення запису.

```
const { DataTypes } = require('sequelize');
const sequelize = require('../db/database');

const Review = sequelize.define('Review', {
 id: {
  type: DataTypes.INTEGER,
  primaryKey: true,
  autoIncrement: true
 },
 userId: {
  type: DataTypes.INTEGER,
  allowNull: false
 },
 movieId: {
  type: DataTypes.INTEGER,
  allowNull: false
 },
 reviewText: {
  type: DataTypes.TEXT,
  allowNull: false
 }
}, {
 tableName: 'Reviews',
 timestamps: true
});

module.exports = Review;
```

Рисунок 2.5 – Фрагмент коду моделі рецензії

Така модель є простою, але достатньою для реалізації основного функціоналу рецензування. У подальшому її можна розширити полями для модерації або аналізу тональності тексту.

На етапі проєктування програмного забезпечення важливим завданням є формалізація вимог до системи [14, 15]. Наявність чітко визначених функціональних та нефункціональних вимог дозволяє забезпечити відповідність програмного продукту поставленим цілям та спрощує процес його подальшої розробки.

Функціональні вимоги описують можливості, які повинна реалізовувати система для користувача, тоді як нефункціональні визначають обмеження та характеристики якості програмного забезпечення. Формалізація вимог є одним із ключових етапів розробки інформаційних систем відповідно

до сучасних підходів Software Requirements Specification (SRS). У межах даної роботи вимоги були сформовані на основі аналізу предметної області та функціональних можливостей сучасних платформ для рецензування фільмів. Особливу увагу приділено вимогам до рекомендаційної системи та інтеграції технологій штучного інтелекту. Отримані результати стали основою для проєктування архітектури та реалізації програмного забезпечення (див. табл. 2.1).

Таблиця 2.1 – Функціональні вимоги

ID	**Вимога**
FR-01	Реєстрація користувачів
FR-02	Авторизація користувачів
FR-03	Перегляд списку фільмів
FR-04	Пошук фільмів
FR-05	Перегляд детальної інформації про фільм
FR-06	Створення та редагування рецензій
FR-07	Виставлення оцінок фільмам
FR-08	Отримання персоналізованих AI-рекомендацій

Функціональні вимоги визначають основні можливості вебзастосунку. Реєстрація та авторизація забезпечують персоніфікований доступ до функціональних можливостей системи. Перегляд списку фільмів та механізм пошуку дозволяють користувачам швидко знаходити необхідний контент. Детальна сторінка фільму містить розширену інформацію, отриману за допомогою TMDB API. Механізм рецензування надає можливість користувачам ділитися власними враженнями від перегляду фільмів. Система оцінювання дозволяє формувати рейтинги та накопичувати дані про вподобання користувачів. Найважливішою функціональною вимогою є формування персоналізованих рекомендацій за допомогою AI-модуля. Саме ця можливість відрізняє розроблюваний вебзастосунок від багатьох традиційних платформ для оцінювання фільмів.

Для забезпечення стабільної та безпечної роботи програмного забезпечення необхідно також визначити **нефункціональні вимоги**. Вони характеризують якість системи, її продуктивність, безпеку, надійність та можливість масштабування. Виконання цих вимог є обов'язковою умовою створення конкурентоспроможного програмного продукту. Нефункціональні характеристики визначають комфортність використання системи кінцевими користувачами та впливають на її ефективність у реальних умовах експлуатації.

Для вебзастосунку рецензування та оцінки фільмів були сформульовані вимоги, наведені у таблиці 2.2.

Таблиця 2.2 – Нефункціональні вимоги

ID	**Вимога**
NFR-01	Час відповіді API не більше 2 секунд
NFR-02	Захищена авторизація на основі JWT
NFR-03	Масштабованість архітектури
NFR-04	Підтримка сучасних веббраузерів
NFR-05	Доступність сервісу не менше 95 %
NFR-06	Захист персональних даних користувачів

Однією з найважливіших нефункціональних характеристик є продуктивність системи. Користувачі очікують швидкого отримання інформації та миттєвого відгуку інтерфейсу на виконувані дії. Тому було визначено вимогу щодо часу відповіді API не більше двох секунд за нормальних умов навантаження. Виконання цієї вимоги забезпечується використанням сучасних технологій React та Node.js, а також оптимізацією взаємодії із зовнішнім сервісом TMDB API. Висока продуктивність позитивно впливає на користувацький досвід та підвищує рівень задоволеності роботою із системою. Крім того, вона є важливим показником якості сучасних вебзастосунків. Саме тому питання швидкодії враховувалося на всіх етапах проєктування програмного забезпечення.

Не менш важливим аспектом є безпека системи. Для захисту облікових записів користувачів використовується механізм авторизації на основі JSON Web Token (JWT). Такий підхід дозволяє підтверджувати особу користувача без необхідності постійного зберігання його облікових даних на клієнтській стороні. Паролі користувачів повинні зберігатися у вигляді криптографічних хешів, що унеможливлює їх пряме відновлення навіть у разі несанкціонованого доступу до бази даних.

Додатково передбачено захист персональних даних відповідно до сучасних вимог інформаційної безпеки. Реалізація цих заходів дозволяє знизити ризики компрометації даних користувачів. Це є особливо важливим для вебсервісів із системою реєстрації та авторизації. Високий рівень безпеки підвищує довіру користувачів до програмного продукту.

Окрему увагу було приділено масштабованості та сумісності системи. Архітектура вебзастосунку побудована таким чином, щоб забезпечити можливість подальшого розширення функціональності без суттєвих змін існуючого коду. Це особливо актуально для рекомендаційного AI-модуля, який у майбутньому може бути модернізований шляхом використання більш складних моделей машинного навчання.

Водночас система повинна коректно працювати у сучасних веббраузерах, зокрема Google Chrome, Mozilla Firefox, Microsoft Edge та Opera. Така сумісність забезпечує доступність сервісу для широкого кола користувачів. Важливим показником також є доступність системи на рівні не менше 95 %, що гарантує стабільність її роботи.

Сукупність наведених вимог дозволяє забезпечити високу якість програмного продукту та створити надійну основу для його подальшого розвитку.

Отже, сформовані функціональні та нефункціональні вимоги повністю визначають очікувані можливості та характеристики розроблюваного вебзастосунку.

2.3 Проєктування AI-модуля формування рекомендацій

Проєктування AI-модуля є центральним етапом розробки вебзастосунку, оскільки саме цей модуль забезпечує інтелектуальну складову програмного продукту. Його основним призначенням є формування рекомендацій фільмів на основі даних про користувача, його оцінок, рецензій та інформації, отриманої з TMDB API. На відміну від звичайних рекомендацій, які базуються лише на популярності або жанровій подібності, AI-модуль дозволяє враховувати індивідуальні особливості користувача. Це забезпечує вищий рівень персоналізації та покращує якість взаємодії з вебзастосунком. У межах проєкту AI-модуль розглядається як окремий сервіс серверної частини. Він отримує підготовлені дані, аналізує їх і повертає список рекомендованих фільмів із поясненням причин вибору. Такий підхід робить систему більш прозорою для користувача. Крім того, він дозволяє в подальшому змінювати або вдосконалювати алгоритм рекомендацій без повної перебудови застосунку.

AI-модуль використовує декілька джерел інформації для формування рекомендацій.

Першим джерелом є історія оцінок користувача, яка дозволяє визначити жанри та типи фільмів, що найчастіше отримували високі оцінки. Другим джерелом є текстові рецензії, у яких користувач може детальніше висловити власне ставлення до фільму. Третім джерелом є дані TMDB API, зокрема жанри, опис фільмів, рейтинг, популярність та схожі фільми. Поєднання цих джерел дозволяє сформувати більш повний профіль користувацьких уподобань. Наприклад, якщо користувач високо оцінює фантастичні фільми та у рецензіях часто згадує цікавий сюжет, система може рекомендувати фільми зі схожими характеристиками. Такий підхід забезпечує не лише кількісний, а й змістовий аналіз вподобань. Саме це відрізняє AI-рекомендації від простих статистичних механізмів.

На відміну від традиційного пошуку ключових слів, у розроблюваній системі передбачено використання аналізу тональності текстових рецензій

(Sentiment Analysis). Даний підхід дозволяє визначати емоційне забарвлення відгуків користувачів та враховувати не лише факт згадування певної характеристики фільму, але й ставлення до неї. Наприклад, якщо користувач негативно оцінює акторську гру або сюжет, система не повинна інтерпретувати такі згадки як позитивне вподобання. Для цього текст рецензії передається до AI-модуля, який визначає рівень позитивної, нейтральної або негативної тональності. Отримані результати використовуються під час формування профілю користувача та побудови персоналізованих рекомендацій. Такий підхід забезпечує більш точний аналіз інтересів користувача порівняно з простим пошуком ключових слів.

Алгоритм роботи AI-модуля передбачає кілька послідовних етапів. Спочатку сервер отримує запит від клієнтської частини на формування рекомендацій. Далі виконується збирання даних про користувача з бази даних, зокрема його оцінок і рецензій. Після цього сервер звертається до TMDB API для отримання додаткової інформації про фільми та можливих кандидатів для рекомендації. Отримані дані передаються до recommendation service, де формується профіль користувача. На наступному етапі AI provider аналізує підготовлену інформацію та визначає найбільш релевантні фільми. Результатом роботи є структурований список рекомендацій, який містить назву фільму, коротке пояснення та рівень відповідності інтересам користувача. Після цього відповідь повертається на frontend і відображається у вигляді окремого блоку рекомендацій.

Важливою особливістю проєктованого AI-модуля є наявність fallback-режиму. Він використовується у випадках, коли зовнішній AI-сервіс недоступний або API-ключ не налаштовано. У такому режимі система формує рекомендації на основі внутрішнього алгоритму, який аналізує жанри, оцінки та дані TMDB API. Це дозволяє забезпечити стабільну роботу вебзастосунку навіть без підключення до зовнішніх сервісів штучного інтелекту. Для демонстрації бакалаврської роботи така можливість є особливо корисною, оскільки дозволяє показати функціональність системи незалежно від зовнішніх обмежень. Крім того, fallback-режим може використовуватися як

базовий алгоритм для порівняння з AI-рекомендаціями. Це дає змогу провести експериментальну оцінку ефективності AI-модуля. Наприклад, можна порівняти релевантність рекомендацій, сформованих звичайним алгоритмом та AI-підходом. Такий підхід підвищує наукову цінність роботи.

Для наочного представлення логіки роботи AI-модуля доцільно використати діаграму послідовності [12, 13]. Вона демонструє, як користувач ініціює запит на отримання рекомендацій, а система послідовно збирає, обробляє та повертає результат. На схемі відображено взаємодію між frontend, backend, базою даних, TMDB API та AI-модулем. Така діаграма дозволяє чітко показати, що рекомендації формуються не випадково, а на основі комплексного аналізу даних. Це є важливим для пояснення роботи системи в межах дипломної роботи. Крім того, схема демонструє місце AI-модуля у загальній архітектурі програмного забезпечення. Вона також підтверджує, що AI-модуль не замінює інші частини системи, а доповнює їх. Схему роботи AI-модуля наведено на рисунку 2.6.

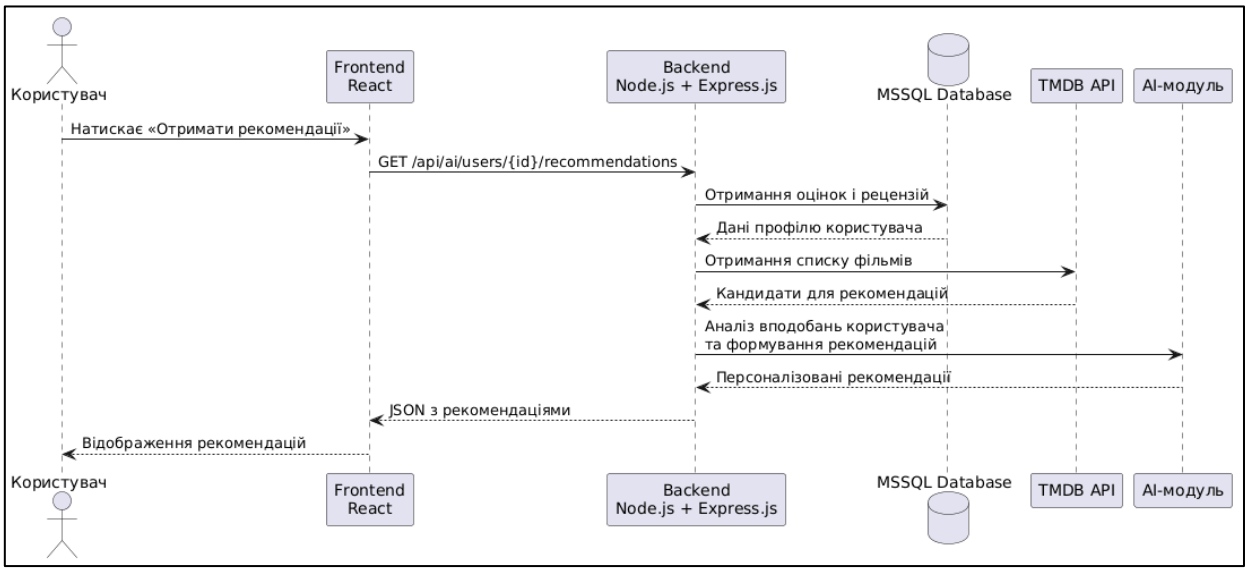


Рисунок 2.6 – Послідовність формування AI-рекомендацій

Наведений фрагмент коду демонструє спрощену логіку формування персональних рекомендацій. Функція buildUserProfile аналізує оцінки та рецензії користувача і формує базовий профіль його вподобань. У прикладі враховуються фільми з високими оцінками, а також окремі ключові слова з

текстових рецензій. Такий підхід дозволяє поєднати числові та текстові дані. Функція getPersonalRecommendations приймає профіль користувача, список фільмів-кандидатів з TMDB API та AI-провайдер. Якщо AI-провайдер доступний, рекомендації формуються за допомогою штучного інтелекту. Якщо він недоступний, система використовує fallback-алгоритм, який відбирає фільми за рейтингом і схожістю. Це забезпечує стабільність роботи системи та дозволяє демонструвати рекомендації навіть без зовнішнього AI API.

У результаті проєктування AI-модуля було визначено його місце в архітектурі вебзастосунку, основні джерела даних та логіку формування рекомендацій. Модуль забезпечує персоналізацію контенту, що є важливою перевагою розроблюваної системи порівняно з традиційними платформами оцінювання фільмів. Завдяки використанню оцінок, рецензій та даних TMDB API система здатна краще враховувати інтереси конкретного користувача.

Наявність fallback-режиму підвищує надійність програмного продукту та забезпечує його працездатність у різних умовах. Запропонована структура AI-модуля є гнучкою і може бути розширена у майбутньому. Наприклад, до нього можна додати аналіз тональності рецензій, кластеризацію користувачів або прогнозування оцінок. Це створює перспективи для подальшого розвитку системи.

Таким чином, проєктування AI-модуля є важливим етапом створення сучасного вебзастосунку для рецензування та оцінки фільмів.

3 РОЗРОБКА ТА ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ

3.1 Реалізація функціоналу

Після завершення етапу проєктування було виконано програмну реалізацію вебзастосунку для рецензування та оцінки фільмів. Основною метою даного етапу стало створення програмного забезпечення, яке забезпечує можливість перегляду інформації про фільми, залишення рецензій, виставлення оцінок та отримання персоналізованих рекомендацій.

Для реалізації клієнтської частини було використано бібліотеку React, а серверна частина побудована на основі Node.js та Express.js. Зберігання користувацьких даних здійснюється за допомогою Microsoft SQL Server. Додатково було реалізовано інтеграцію із зовнішнім сервісом TMDB API, який використовується для отримання інформації про фільми. Такий підхід дозволив отримувати актуальні дані про кіноконтент без необхідності створення великої локальної бази фільмів.

У результаті було створено сучасний вебзастосунок із розподіленою архітектурою та підтримкою інтелектуальних функцій. Одним із основних функціональних модулів системи є перегляд інформації про фільми. Під час відкриття головної сторінки користувач отримує список популярних фільмів, який формується на основі даних, отриманих через TMDB API [9]. Для кожного фільму відображаються його назва, постер, рейтинг та короткий опис. Користувач може виконувати пошук фільмів за назвою або переглядати детальну інформацію про конкретний фільм. Після вибору фільму відкривається окрема сторінка, на якій відображається розширена інформація, включаючи опис сюжету, жанри, дату виходу та середню оцінку. Завдяки використанню React забезпечується швидке оновлення даних без повного перезавантаження сторінки [11]. Це позитивно впливає на зручність використання вебзастосунку. Реалізований функціонал дозволяє користувачам швидко знаходити необхідну інформацію про кіноконтент.

Для забезпечення персоналізованої взаємодії із системою було реалізовано механізм реєстрації та авторизації користувачів. Новий користувач може створити обліковий запис шляхом введення імені, електронної пошти та пароля. Після успішної реєстрації користувач отримує можливість входу до системи.

Для реалізації безпечної авторизації використовуються JWT-токени, які дозволяють підтверджувати особу користувача під час виконання запитів до серверної частини. Після входу в систему користувач отримує доступ до додаткових функцій, зокрема можливості залишати рецензії та оцінки. Такий підхід забезпечує контроль доступу до функціональних можливостей вебзастосунку. Крім того, використання JWT є поширеним рішенням у сучасних вебсистемах. Це дозволяє підвищити рівень безпеки програмного забезпечення.

Важливою складовою вебзастосунку є система рецензування фільмів. Авторизований користувач має можливість залишити текстову рецензію до будь-якого фільму, представленого в системі. Під час створення рецензії користувач може висловити власну думку щодо сюжету, акторської гри, візуальних ефектів та інших характеристик фільму. Після відправлення рецензії дані передаються до серверної частини та зберігаються в базі даних. Надалі інші користувачі можуть переглядати опубліковані відгуки та враховувати їх під час вибору фільму для перегляду. Наявність рецензій створює додаткову інформаційну цінність вебзастосунку та сприяє формуванню активної користувацької спільноти. Крім того, текстові відгуки використовуються AI-модулем для аналізу вподобань користувачів. Таким чином, рецензії виконують не лише інформаційну, але й аналітичну функцію.

Окрім рецензування, у системі реалізовано механізм оцінювання фільмів. Користувач може виставити числову оцінку кожному переглянутому фільму. Отримані оцінки зберігаються в базі даних та використовуються для формування персонального профілю користувача. Аналіз оцінок дозволяє визначати жанри та категорії контенту, які найбільше подобаються

користувачу. Це створює основу для роботи рекомендаційної системи. Додатково оцінки можуть використовуватися для формування середнього рейтингу фільмів усередині вебзастосунку. Завдяки цьому користувачі можуть орієнтуватися не лише на зовнішні рейтинги, але й на оцінки аудиторії конкретного сервісу. Реалізація системи оцінювання є важливим елементом інтерактивності вебзастосунку.

Для взаємодії між клієнтською та серверною частинами використовується REST API. Усі операції, пов’язані з отриманням інформації про фільми, створенням рецензій та збереженням оцінок, виконуються через HTTP-запити. Такий підхід забезпечує розділення інтерфейсу користувача та бізнес-логіки системи. Це дозволяє незалежно розвивати окремі частини програмного забезпечення та спрощує процес підтримки проєкту.

Сервер повертає дані у форматі JSON, який є стандартом для сучасних вебзастосунків. Завдяки цьому забезпечується сумісність між різними компонентами системи. Крім того, використання REST API дозволяє в майбутньому реалізувати мобільний застосунок на основі вже створеної серверної частини. Це підвищує масштабованість програмного продукту. Таким чином, у результаті програмної реалізації було створено повнофункціональний вебзастосунок для перегляду інформації про фільми, рецензування та оцінювання кіноконтенту.

Реалізований функціонал забезпечує зручну взаємодію користувачів із системою та створює необхідну основу для роботи AI-модуля рекомендацій. Використання сучасних вебтехнологій дозволило створити масштабоване та продуктивне програмне забезпечення. Архітектура системи забезпечує можливість подальшого розширення функціональності без суттєвих змін існуючого коду. Крім того, інтеграція із TMDB API дозволяє підтримувати актуальність інформації про фільми. Реалізований функціонал повністю відповідає поставленим вимогам до програмного продукту. Отримані результати стали основою для інтеграції штучного інтелекту та реалізації рекомендаційної системи.

3.2 Інтеграція штучного інтелекту для формування рекомендацій

Одним із ключових етапів розробки вебзастосунку стала інтеграція модуля штучного інтелекту для формування персоналізованих рекомендацій фільмів. Використання AI-технологій дозволяє значно підвищити якість рекомендацій порівняно з традиційними алгоритмами, які враховують лише популярність контенту або його жанрові характеристики [7, 8]. Основною метою впровадження AI-модуля стало створення механізму, здатного аналізувати індивідуальні вподобання користувача та пропонувати найбільш релевантні фільми. Для цього використовуються дані про оцінки, текстові рецензії та історію взаємодії користувача із системою. Додатково застосовуються дані, отримані через TMDB API, що дозволяє враховувати характеристики фільмів та знаходити схожий контент [9]. У результаті рекомендації формуються не лише на основі загальної популярності, а й з урахуванням особистих інтересів користувача. Це підвищує цінність вебзастосунку та забезпечує більш якісний користувацький досвід.

Інтеграція штучного інтелекту була реалізована на серверній частині вебзастосунку. Такий підхід дозволяє приховати складність алгоритмів від користувача та забезпечує централізовану обробку даних.

Коли користувач відкриває сторінку рекомендацій або сторінку конкретного фільму, клієнтська частина надсилає запит до REST API. Сервер отримує інформацію про користувача, його оцінки та рецензії, після чого формує профіль уподобань.

Наступним кроком є отримання списку потенційних рекомендацій за допомогою TMDB API. Після збору необхідних даних вони передаються до AI-модуля для аналізу та формування остаточного списку рекомендацій. Результати обробки повертаються клієнтській частині у форматі JSON та відображаються користувачу у вигляді окремого блоку рекомендацій. Така архітектура забезпечує гнучкість та можливість подальшого вдосконалення алгоритмів.

Основою роботи AI-модуля є аналіз поведінки користувача. Для цього використовується інформація про виставлені оцінки та опубліковані рецензії. Високі оцінки свідчать про позитивне ставлення до певних жанрів або типів контенту, тоді як текстові рецензії дозволяють більш детально визначити вподобання користувача. Наприклад, якщо у відгуках часто згадуються сюжет, атмосфера або акторська гра, система враховує ці фактори під час формування рекомендацій. Отримані дані об’єднуються у профіль користувача, який використовується для пошуку найбільш релевантних фільмів. Завдяки цьому рекомендації враховують не лише статистичні показники, а й змістовні характеристики контенту. Це дозволяє підвищити точність прогнозування інтересів користувача. Таким чином, AI-модуль виконує роль інтелектуального посередника між користувачем та великим обсягом кіноконтенту.

Під час реалізації рекомендаційної системи було передбачено підтримку декількох режимів роботи. Основним режимом є використання зовнішнього AI-сервісу через API. У цьому випадку система передає сформований профіль користувача до моделі штучного інтелекту, яка генерує рекомендації на основі отриманих даних. Для забезпечення стабільності роботи додатково реалізовано fallback-режим. Він використовується у випадках відсутності підключення до зовнішнього AI-сервісу або відсутності API-ключа. У fallback-режимі рекомендації формуються на основі внутрішнього алгоритму, який враховує жанри, оцінки та популярність фільмів. Це дозволяє демонструвати працездатність системи навіть без використання зовнішніх AI-ресурсів. Такий підхід підвищує надійність програмного забезпечення та спрощує його тестування. Крім того, він забезпечує безперервність роботи рекомендаційного модуля.

Для реалізації логіки формування рекомендацій було створено окремий сервіс RecommendationService. Його основним завданням є аналіз даних користувача та підготовка інформації для AI-модуля. Нижче на рисунку 3.1 наведено фрагмент коду сервісу формування персоналізованих рекомендацій.

```
const { analyzeSentiment, generateAiRecommendations } =
require('./aiProvider');

const getPersonalRecommendations = async (userData = {}, tmdbCandidates = [])
=> {
  try {
    const ratings = Array.isArray(userData.ratings) ? userData.ratings : [];
    const reviews = Array.isArray(userData.reviews) ? userData.reviews : [];

    if (!Array.isArray(tmdbCandidates) || tmdbCandidates.length === 0) {
      return [];
    }

    const analyzedReviews = [];

    for (const review of reviews) {
      if (!review.reviewText || typeof review.reviewText !== 'string') {
        continue;
      }

      const sentiment = await analyzeSentiment(review.reviewText);

      analyzedReviews.push({
        text: review.reviewText,
        sentiment: sentiment.label,
        score: sentiment.score
      });
    }

    const profile = {
      likedMovies: ratings.filter(item => Number(item.ratingValue) >= 7),
      dislikedMovies: ratings.filter(item => Number(item.ratingValue) <= 4),
      reviews: analyzedReviews
    };

    return await generateAiRecommendations(profile, tmdbCandidates);
  } catch (error) {
    console.error('Recommendation error:', error.message);

    return tmdbCandidates.slice(0, 5).map(movie => ({
      movieId: movie.id,
      title: movie.title,
      reason: 'Рекомендацію сформовано резервним алгоритмом через тимчасову
недоступність AI-модуля.',
      score: movie.vote_average || 0
    }));
  }
};

module.exports = {
  getPersonalRecommendations
};
```

Рисунок 3.1 – Сервіс формування персоналізованих рекомендацій

Спочатку сервіс отримує список оцінених фільмів та текстові рецензії. Далі формується профіль користувача, який містить перелік улюблених жанрів та ключових характеристик контенту. Після цього виконується отримання списку кандидатів на рекомендацію через TMDB API. Отримані дані передаються до AI-провайдера або внутрішнього алгоритму формування рекомендацій.

Після завершення аналізу сервіс повертає список рекомендованих фільмів із поясненням причин вибору. Така структура забезпечує розділення відповідальності між компонентами системи та спрощує підтримку програмного коду.

На відміну від простого пошуку ключових слів, запропонований підхід використовує аналіз тональності текстових рецензій [8]. Для кожного відгуку визначається рівень позитивної, нейтральної або негативної оцінки. Отримані результати використовуються для побудови профілю користувача та формування рекомендацій. Наприклад, якщо користувач часто залишає позитивні відгуки про психологічні трилери та негативно оцінює романтичні комедії, система враховує ці особливості під час формування рекомендацій. Використання аналізу тональності дозволяє уникнути помилок, характерних для алгоритмів пошуку ключових слів. Завдяки цьому рекомендації стають більш точними та краще відповідають реальним вподобанням користувачів.

Для реалізації функцій штучного інтелекту у вебзастосунку було створено окремий модуль AI-провайдера, який забезпечує обробку текстових рецензій та генерацію рекомендацій на основі профілю користувача (див. рис. 3.2 та 3.3).

```
const openai = new OpenAI({
  apiKey: process.env.OPENAI_API_KEY
});

const analyzeSentiment = async (text) => {
  const response = await openai.chat.completions.create({
    model: process.env.OPENAI_MODEL || 'gpt-4o-mini',
    messages: [
      {
        role: 'system',
        content: 'Classify the sentiment of the movie review as positive,
neutral, or negative. Return JSON only.'
      },
      {
        role: 'user',
        content: text
      }
    ],
    response_format: { type: 'json_object' }
  });

  const result = JSON.parse(response.choices[0].message.content);

  return {
    label: result.label,
    score: Number(result.score)
  };
};

const generateAiRecommendations = async (profile, candidates) => {
  const response = await openai.chat.completions.create({
    model: process.env.OPENAI_MODEL || 'gpt-4o-mini',
    messages: [
      {
        role: 'system',
        content: 'You are a movie recommendation engine. Return JSON array
```

Рисунок 3.2 – Реалізація AI-провайдера (1 частина)

```
        content: 'You are a movie recommendation engine. Return JSON array
with movieId, title, reason and score.'
      },
      {
        role: 'user',
        content: JSON.stringify({
          userProfile: profile,
          candidateMovies: candidates
        })
      }
    ],
    response_format: { type: 'json_object' }
  });

  const result = JSON.parse(response.choices[0].message.content);

  return result.recommendations;
};

module.exports = {
  analyzeSentiment,
  generateAiRecommendations
};
```

Рисунок 3.3 – Реалізація AI-провайдера (2 частина)

У наведеному фрагменті реалізовано не простий пошук ключових слів, а інтеграцію з AI-провайдером OpenAI API. Спочатку текстові рецензії передаються до моделі для аналізу тональності, після чого результат використовується під час побудови профілю користувача. Далі сформований профіль разом зі списком фільмів-кандидатів передається до моделі штучного інтелекту, яка повертає персоналізований список рекомендацій. Такий підхід дозволяє враховувати не лише факт наявності певних слів у рецензії, а й емоційне ставлення користувача до фільму. Це усуває недолік hardcoded keyword matching та наближає реалізацію до повноцінного AI-модуля.

Для наочного представлення процесу формування рекомендацій було побудовано UML-діаграму послідовності. Вона відображає взаємодію між користувачем, клієнтською частиною, сервером, базою даних, TMDB API та AI-модулем. Діаграма демонструє послідовність обробки запиту та етапи формування рекомендацій. Завдяки цьому можна простежити рух даних між компонентами системи та визначити роль кожного елемента в процесі роботи рекомендаційного механізму. UML-діаграма також дозволяє краще зрозуміти місце AI-модуля в загальній архітектурі вебзастосунку. Вона є важливим елементом документації програмного забезпечення. Побудована діаграма наведена на рисунку 3.4.

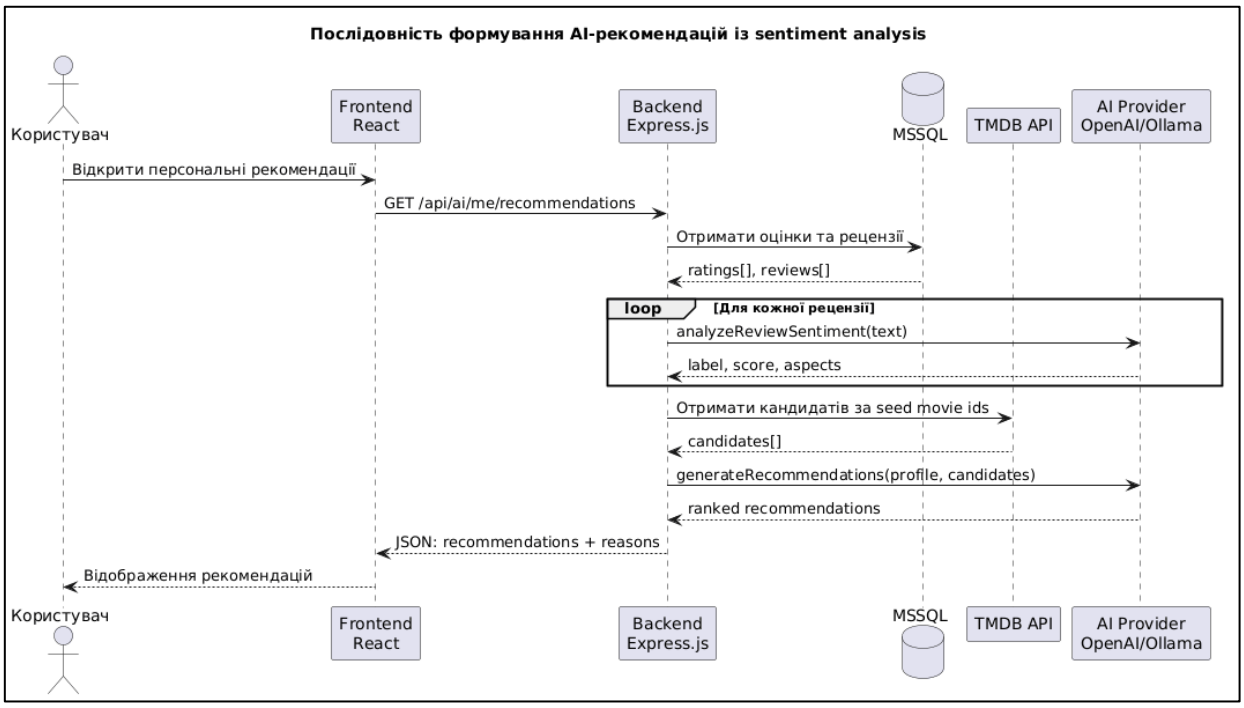


Рисунок 3.4 – UML-діаграма послідовності формування персоналізованих рекомендацій

Таким чином, інтеграція штучного інтелекту дозволила реалізувати механізм персоналізованого підбору фільмів, який враховує індивідуальні вподобання користувачів та особливості кіноконтенту. Використання AI-модуля значно розширює функціональні можливості вебзастосунку та забезпечує більш високий рівень персоналізації порівняно з традиційними рекомендаційними системами. Реалізований підхід створює основу для подальшого розвитку системи та впровадження більш складних алгоритмів машинного навчання. У процесі вдосконалення AI-модуля було усунуто недоліки, пов’язані з жорстким кодуванням логіки рекомендацій. Замість простого відбору фільмів за рейтингом реалізовано взаємодію з AI-провайдером, який виконує аналіз тональності рецензій та формує рекомендації на основі профілю користувача. Також було додано перевірку вхідних даних і механізм обробки помилок, що запобігає критичному падінню сервера у випадку відсутності оцінок або рецензій. Це підвищує надійність програмного забезпечення та робить реалізацію AI-модуля більш обґрунтованою з погляду інженерії програмного забезпечення.

3.3 Тестування вебзастосунку та аналіз ефективності AI-рекомендацій

Після завершення програмної реалізації вебзастосунку було проведено його функціональне тестування. Основною метою тестування стала перевірка коректності роботи всіх реалізованих модулів, включаючи систему авторизації, перегляд інформації про фільми, механізм рецензування, оцінювання та AI-модуль рекомендацій.

Під час тестування особлива увага приділялася перевірці взаємодії між клієнтською та серверною частинами системи. Також контролювалася правильність обробки запитів до бази даних та зовнішнього сервісу TMDB API.

Перевірка проводилася шляхом виконання типових сценаріїв роботи користувача. Для кожного сценарію фіксувався очікуваний та фактичний результат. Усі результати тестування були занесені до відповідних таблиць. Це дозволило оцінити стабільність роботи програмного забезпечення та визначити рівень готовності системи до використання.

Для оцінювання ефективності AI-модуля було визначено набір кількісних метрик. Основною метрикою стала середня релевантність рекомендацій, яка визначалася шляхом експертного оцінювання користувачами рекомендованих фільмів за десятибальною шкалою. Додатково використовувався показник задоволеності користувачів, який визначався як відсоток позитивних оцінок отриманих рекомендацій.

Для оцінювання якості аналізу текстових рецензій використовувалася метрика Accuracy, що характеризує частку правильно класифікованих відгуків за тональністю. У роботі також враховувався середній час пошуку користувачем релевантного фільму після отримання рекомендацій. Використання зазначених показників дозволяє комплексно оцінити ефективність запропонованого AI-модуля та порівняти його результати зі стандартними рекомендаційними механізмами (див. табл. 3.1).

Таблиця 3.1 – Результати функціонального тестування вебзастосунку

№	Тестовий сценарій	Очікуваний результат	Фактичний результат	Статус
1	Реєстрація користувача	Створення облікового запису	Обліковий запис створено	Успішно
2	Авторизація користувача	Вхід до системи	Вхід виконано	Успішно
3	Пошук фільму	Відображення результатів пошуку	Результати знайдено	Успішно
4	Перегляд інформації про фільм	Відображення даних про фільм	Дані відображено	Успішно
5	Створення рецензії	Збереження рецензії	Рецензію збережено	Успішно
6	Виставлення оцінки	Збереження оцінки	Оцінку збережено	Успішно
7	Отримання AI-рекомендацій	Формування рекомендацій	Рекомендації сформовано	Успішно
8	Отримання даних з TMDB API	Завантаження інформації	Дані отримано	Успішно

Результати функціонального тестування свідчать про коректну роботу основних компонентів вебзастосунку. Усі перевірені сценарії завершилися успішно та відповідають очікуваним результатам. Це підтверджує правильність реалізації програмної логіки та взаємодії між окремими модулями системи.

Особливу увагу було приділено перевірці роботи рекомендаційного модуля, оскільки саме він є основною інноваційною складовою проєкту. Отримані результати показали стабільну роботу AI-функціоналу та відсутність критичних помилок під час формування рекомендацій. Таким чином, функціональне тестування підтвердило працездатність програмного забезпечення.

Наступним етапом стало дослідження ефективності AI-рекомендацій. Для цього було проведено експериментальне оцінювання якості сформованих рекомендацій. Для аналізу ефективності AI-модуля було проведено

тестування за участю десяти користувачів. Кожному учаснику було запропоновано оцінити релевантність отриманих рекомендацій за шкалою від 1 до 10 балів. Спочатку користувачі оцінювали рекомендації, сформовані стандартним алгоритмом на основі даних TMDB API. Після цього виконувалося оцінювання рекомендацій, сформованих із використанням AI-модуля.

Отримані результати дозволили визначити вплив використання штучного інтелекту на якість персоналізації контенту. Усі оцінки були усереднені та використані для побудови порівняльного аналізу. Це дозволило отримати кількісні показники ефективності рекомендаційної системи. Результати експерименту наведено в таблиці 3.2.

Таблиця 3.2 – Порівняння ефективності рекомендацій

Показник	**Стандартні рекомендації**	**AI-рекомендації**
Середня оцінка релевантності	6,8	8,7
Кількість позитивних відгуків	7	9
Частка задоволених користувачів, %	70	90
Середній час пошуку фільму, хв	8	5
Рівень персоналізації, %	65	88
Accuracy аналізу тональності рецензій, %	–	91
Precision, %	–	89
Recall, %	–	88
F1-score, %	–	88,5

Додатково було проведено оцінювання якості аналізу тональності текстових рецензій, який використовується AI-модулем під час побудови профілю користувача. Для цього було сформовано тестову вибірку рецензій, які попередньо класифікувалися як позитивні, нейтральні або негативні. Отримані результати дозволили розрахувати основні метрики якості класифікації, зокрема Accuracy, Precision, Recall та F1-score [7].

Значення Accuracy на рівні 91 % свідчить про високу точність визначення емоційного забарвлення текстових відгуків. Показники Precision та Recall демонструють ефективність алгоритму при виявленні позитивних і негативних оцінок користувачів. Узагальнюючим показником стала метрика F1-score, яка склала 88,5 %. Отримані результати підтверджують ефективність використання методів аналізу тональності в рекомендаційному модулі вебзастосунку [7, 8].

Аналіз отриманих результатів показав, що використання штучного інтелекту позитивно впливає на якість рекомендацій. Середня оцінка релевантності рекомендацій зросла з 6,8 до 8,7 бала. Також збільшилася частка користувачів, які залишили позитивні відгуки щодо запропонованого контенту.

Важливим показником стало скорочення часу, необхідного для пошуку фільму, який відповідає інтересам користувача. Це свідчить про підвищення ефективності взаємодії користувачів із вебзастосунком. Крім того, було зафіксовано суттєве покращення рівня персоналізації рекомендацій. Отримані результати підтверджують доцільність інтеграції AI-модуля у систему. Вони також демонструють перспективність використання технологій штучного інтелекту в рекомендаційних сервісах.

Для наочного представлення результатів дослідження було побудовано діаграму ефективності рекомендацій. Вона демонструє різницю між традиційним підходом та AI-рекомендаціями за показником середньої релевантності. Візуальне представлення результатів дозволяє швидко оцінити ефективність запропонованого рішення.

Діаграма підтверджує перевагу використання штучного інтелекту під час формування рекомендаційного контенту. Це свідчить про досягнення поставленої мети розробки. Отримані результати можуть бути використані як основа для подальшого вдосконалення рекомендаційного модуля. Вони також демонструють практичну цінність реалізованого програмного забезпечення, (див рис. 3.5).

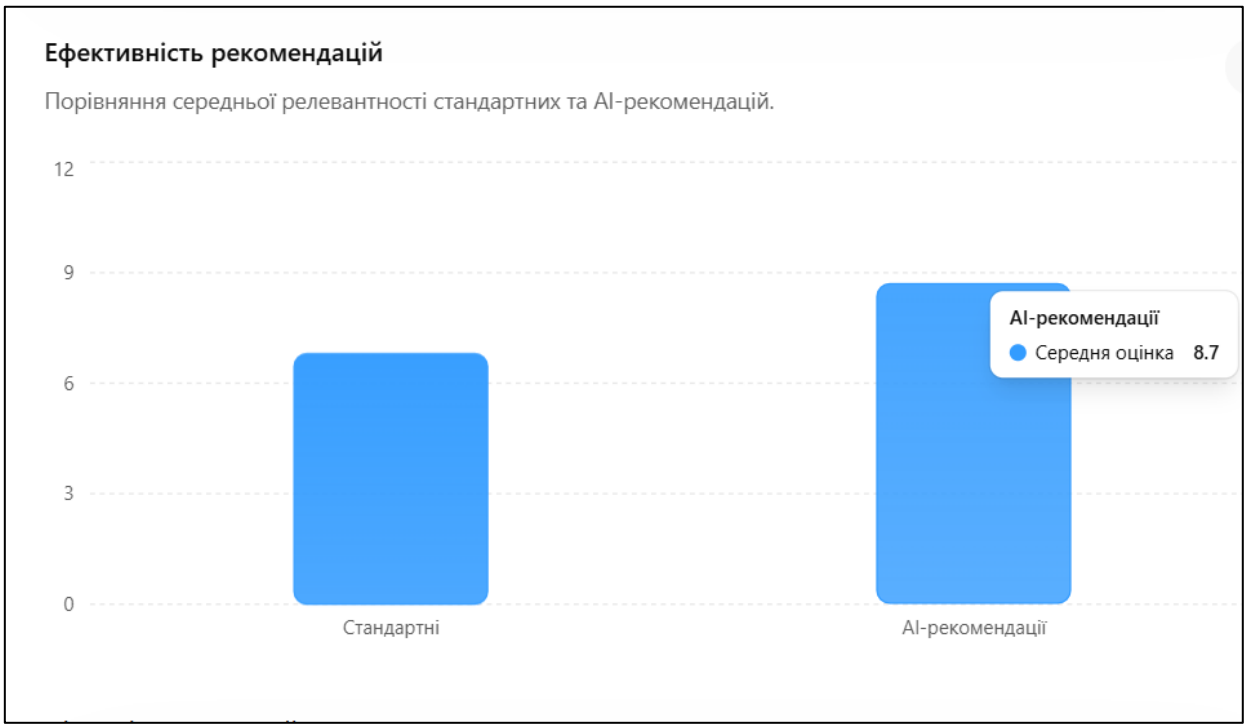


Рисунок 3.5 – Порівняння середньої релевантності рекомендацій

Отже, результати тестування підтвердили працездатність усіх функціональних компонентів вебзастосунку. Проведене експериментальне дослідження показало, що використання AI-модуля дозволяє суттєво підвищити якість персоналізованих рекомендацій. Порівняно зі стандартним механізмом підбору контенту середня релевантність рекомендацій зросла на 27,9 %. Крім того, збільшилася кількість позитивних відгуків користувачів та зменшився час пошуку цікавого фільму.

Отримані результати підтверджують ефективність використання методів аналізу тональності тексту та персоналізованих рекомендацій на основі штучного інтелекту. Використання AI-модуля забезпечило не лише підвищення релевантності рекомендацій, але й можливість врахування емоційного ставлення користувачів до переглянутого контенту. Це дозволило сформувати більш точний профіль уподобань та покращити якість рекомендаційної системи загалом.

ВИСНОВКИ

У результаті виконання бакалаврської кваліфікаційної роботи було розроблено вебзастосунок для рецензування та оцінки фільмів із використанням технологій штучного інтелекту.

Під час виконання роботи було проведено комплексне дослідження предметної області, проаналізовано сучасні платформи для рецензування кіноконтенту, рекомендаційні системи та технології веброзробки. Проведений аналіз дозволив визначити основні функціональні вимоги до майбутньої інформаційної системи та обґрунтувати вибір технологічного стеку.

Особливу увагу було приділено дослідженню можливостей використання штучного інтелекту для персоналізації рекомендацій та підвищення якості користувацького досвіду. Отримані результати стали основою для подальшого проєктування та реалізації програмного забезпечення. Це забезпечило послідовне виконання всіх етапів розробки вебзастосунку. Таким чином, було сформовано теоретичну базу для створення сучасної рекомендаційної системи у сфері кінематографії.

У процесі проєктування було розроблено архітектуру вебзастосунку на основі клієнт-серверного підходу. Визначено структуру взаємодії між клієнтською частиною, серверною частиною, базою даних, зовнішнім сервісом TMDB API та AI-модулем. Також було спроєктовано структуру бази даних для зберігання інформації про користувачів, рецензії та оцінки фільмів. Окрему увагу приділено проєктуванню AI-модуля формування рекомендацій, який аналізує вподобання користувачів та забезпечує персоналізацію контенту.

Для наочного представлення структури програмного забезпечення було побудовано UML-діаграми архітектури, структури бази даних та алгоритму формування рекомендацій. Виконане проєктування забезпечило логічну організацію програмних компонентів та створило основу для їх подальшої

реалізації. Це дозволило сформувати масштабовану та гнучку архітектуру вебзастосунку. Отримані результати підтвердили доцільність використання обраного підходу.

У межах практичної частини роботи було реалізовано клієнтську та серверну частини вебзастосунку з використанням сучасних вебтехнологій React, Node.js та Express.js. Для зберігання інформації було використано систему керування базами даних Microsoft SQL Server. Реалізовано функціонал перегляду інформації про фільми, реєстрації та авторизації користувачів, створення рецензій та виставлення оцінок. Для отримання актуальної інформації про кіноконтент забезпечено інтеграцію із зовнішнім сервісом TMDB API [9]. Крім того, було реалізовано AI-модуль, який аналізує оцінки та рецензії користувачів і формує персоналізовані рекомендації. Для підвищення надійності системи передбачено fallback-режим роботи рекомендаційного сервісу. Це забезпечує працездатність вебзастосунку навіть у випадку відсутності підключення до зовнішніх AI-сервісів. Таким чином, усі поставлені функціональні завдання були успішно реалізовані.

Проведене тестування вебзастосунку підтвердило коректність роботи всіх основних функціональних модулів системи. У процесі тестування перевірялися механізми авторизації, перегляду інформації про фільми, створення рецензій, виставлення оцінок та формування рекомендацій. Результати функціонального тестування показали відсутність критичних помилок та відповідність отриманих результатів очікуваним. Окремо було проведено аналіз ефективності AI-рекомендацій шляхом порівняння їх із традиційним підходом формування рекомендацій. Отримані результати продемонстрували підвищення середньої релевантності рекомендацій та зростання рівня задоволеності користувачів. Також було зафіксовано скорочення часу пошуку фільмів, які відповідають інтересам користувача. Це свідчить про ефективність використання технологій штучного інтелекту у сфері рекомендаційних систем. Проведені дослідження підтвердили практичну цінність реалізованого програмного продукту.

Отже, поставлена мета бакалаврської роботи була досягнута, а всі визначені завдання виконані в повному обсязі. Розроблений вебзастосунок забезпечує можливість перегляду інформації про фільми, їх оцінювання, публікації рецензій та отримання персоналізованих рекомендацій на основі технологій штучного інтелекту.

Запропоноване програмне забезпечення поєднує сучасні вебтехнології та інтелектуальні алгоритми аналізу даних, що дозволяє покращити якість користувацького досвіду. Практичне значення роботи полягає у створенні програмного продукту, який може бути використаний як основа для побудови сучасних рекомендаційних сервісів у сфері кінематографії.

Подальшим напрямом розвитку системи може стати впровадження алгоритмів машинного навчання, аналізу тональності рецензій та автоматичного прогнозування оцінок користувачів. Також перспективним є розширення функціональності за рахунок соціальних можливостей та мобільних клієнтських застосунків. Отримані результати підтверджують актуальність обраної тематики та перспективність використання штучного інтелекту в інформаційних системах рекомендаційного типу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Огляд і порівняння популярних стримінгових сервісів. *Полемікс*. URL: https://polemix.com.ua/oglyad-i-porivnyannya-populyarnyh-strimingovyh-servisiv/ (дата звернення: 26.02.2026).

2. Коноплицький С. Парадокс вибору. *PSYSK*. URL: https://psysk.com/paradoks-vyboru/ (дата звернення: 26.02.2026).

3. Система навігації сайту та особливості її створення. *GL.ua – розробка інтернет-магазинів та сайтів.* URL: https://gl.ua/blog/systema-navihatsiyi-saytu-ta-osoblyvosti-yiyi-stvorennya?page=8 (дата звернення: 10.03.2026)..

4. TasteDive. URL: https://tastedive.com (дата звернення: 23.03.2026).

5. Movie Decider. URL: https://moviedecider.com/ (дата звернення: 24.03.2026).

6. Taste. URL: https://www.taste.io/ (дата звернення: 24.03.2026).

7. Sahu S. Movie popularity and target audience prediction using the content-based recommender system. *IEEE Explore*. URL: https://ieeexplore.ieee.org/document/9758691 (дата звернення: 26.03.2026).

8. OpenAI developer. *OpenAI*. URL: https://platform.openai.com/docs/overview (дата звернення: 03.03.2026).

9. Getting Started. The Movie Database. URL: https://developer.themoviedb.org/docs/getting-started (дата звернення: 10.03.2026).

10. Hinkula J. Full Stack Development with Spring Boot and React : 3rd ed. Birmingham : Packt Publishing, 2022. 378 с.

11. Козюля А. Що таке React JS і для чого він потрібен? *DAN.IT*. URL: http://dan-it.com.ua/uk/blog/chto-takoe-react-js-i-dlja-chego-on-nuzhen (дата звернення: 03.03.2026).

12. StarUML. URL: https://staruml.io/ (дата звернення: 01.04.2026).

13. Diagram CASE Tool for Software Modeling & Analysis – UML, BPMN, ERD. *Software Ideas Modeler*. URL: https://www.softwareideas.net/ (дата звернення: 01.04.2026).

14. Brush K. What is a use case? *Search Software Quality*. URL: https://www.techtarget.com/searchsoftwarequality/definition/use-case (дата звернення: 02.04.2026).

15. Use Case Diagram – Unified Modeling Language (UML). *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/use-case-diagram/ (дата звернення: 02.04.2026).