

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
(повна назва)

Кафедра _____ програмної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

_____ Дослідження рекомендаційних систем _____
_____ вибору локацій для візиту _____
(тема)

Виконав:
здобувач _____ 2 _____ року навчання
групи _____ ІПЗм-23-1 _____

_____ Владислав ЛАПІН _____
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність _____ 121 – Інженерія програмного _____
забезпечення _____
(код і повна назва спеціальності)

Тип програми _____ освітньо-наукова _____

Керівник _____ доц. Анастасія ЧУПРИНА _____
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри

_____ Кирило СМЕЛЯКОВ _____
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук
 Кафедра _____ програмної інженерії
 Рівень вищої освіти _____ другий (магістерський)
 Спеціальність _____ 121 – Інженерія програмного забезпечення
 Тип програми _____ освітньо-наукова програма
 Освітня програма _____ Інженерія програмного забезпечення
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Лапіну Владиславу Ігоровичу
 (прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження рекомендаційних систем вибору локацій для візиту»

Затверджена наказом по університету від _____ 15.04.2025 р. № 290 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 13.06.2025

3. Вихідні дані до роботи в програмному застосунку передбачити: виконання та забезпечення взаємодії клієнтської та серверної частини програмного забезпечення для надання користувачеві можливості отримання релевантних рекомендацій локацій для візиту на основі обраних фільтрів та поточних умов.

4. Перелік питань, що потрібно опрацювати в роботі вступ, аналіз предметної галузі, огляд й аналіз літературних, наукових джерел, постановка задачі, теоретичне дослідження, практичне дослідження, опис розробленої програмної системи висновки, перелік джерел посилань, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	16.04.2025	<i>виконано</i>
2	Аналіз предметної галузі і постановка задачі	18.04.2025	<i>виконано</i>
3	Архітектура та проектування ПЗ	23.04.2025	<i>виконано</i>
4	Проектування структури зберігання даних	28.04.2025	<i>виконано</i>
5	Проектування схеми роботи системи рекомендацій	30.04.2025	<i>виконано</i>
6	UI/UX дизайн системи	05.05.2025	<i>виконано</i>
7	Опис проведеного дослідження	08.05.2025	<i>виконано</i>
8	Аналіз отриманих результатів дослідження	15.05.2025	<i>виконано</i>
9	Підготовка до апробації результатів дослідження. Публікація матеріалів	20.05.2025	<i>виконано</i>
10	Підготовка пояснювальної записки	24.05.2025	<i>виконано</i>
11	Підготовка презентації та доповіді	28.05.2025	<i>виконано</i>
12	Перевірка на плагіат	01.06.2025	<i>виконано</i>
13	Нормоконтроль	02.06.2025	<i>виконано</i>
14	Рецензування	03.06.2025	<i>виконано</i>
15	Попередній захист	10.06.2025	<i>виконано</i>
16	Занесення диплома в електронний архів	11.06.2025	<i>виконано</i>
17	Допуск до захисту у зав. кафедри	12.06.2025	<i>виконано</i>

Дата видачі завдання 15 квітня 2025 р.

Студент _____
(підпис)

_____ **Владислав ЛАПІН** _____

Керівник роботи _____
(підпис)

_____ **доц. Анастасія ЧУПРИНА** _____
(посада, Власне ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 87 с., 38 рис., 5 додатків, 16 джерел.

ВЕБ, ЛОКАЦІЇ, СИСТЕМА РЕКОМЕНДАЦІЙ, ФІЛЬТРАЦІЯ НА ОСНОВІ КОНТЕНТУ, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, ГІБРИДНИЙ ПІДХІД, PYTHON, DJANGO, TYPESCRIPT, REACT, POSTGRESQL.

Об'єкт дослідження – процес формування індивідуальних рекомендацій для вибору локацій, що базується на аналізі даних користувачів та характеристик локацій.

Мета роботи – розробка ефективного підходу до створення рекомендаційної системи, яка враховувала б індивідуальні потреби користувачів для вибору оптимальних локацій для відвідування.

Методи розробки та проектування – середовище розробки Visual Studio Code, мови програмування Python та TypeScript, фреймворк Django, бібліотека React, СКБД PostgreSQL.

У результаті роботи було проведено дослідження рекомендаційної системи, що надасть можливість користувачам отримувати релевантні пропозиції щодо відвідування локацій на основі обраних фільтрів та поточних умов.

WEB, LOCATIONS, RECOMMENDATION SYSTEM, CONTENT-BASED FILTERING, COLLABORATIVE FILTERING, HYBRID APPROACH, PYTHON, DJANGO, TYPESCRIPT, REACT, POSTGRESQL.

The object of the study is the process of forming individual recommendations for choosing locations, based on the analysis of user data and location characteristics.

The purpose of the work is to develop an effective approach to creating a recommendation system that would consider the individual needs of users to choose the optimal locations to visit.

Development and design methods are the Visual Studio Code development environment, Python and TypeScript programming languages, Django framework, React library, PostgreSQL database.

As a result of the work, a study of the recommendation system was conducted, which will allow users to receive relevant suggestions for visiting locations based on selected filters and current conditions.

Заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу EIArKhNURE.

Я, Лапін Владислав Ігорович, студент гр. ПЗм-23-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження рекомендаційних систем вибору локацій для візиту», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови до допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата

Підпис

ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі.....	10
1.1 Специфіка досліджуваного напрямку	10
1.2 Огляд існуючих підходів та обмежень.....	11
1.3 Тенденції та перспективи розвитку	13
2 Огляд й аналіз літературних, наукових джерел.....	15
2.1 Огляд основних джерел	15
2.2 Аналіз літератури.....	17
2.3 Оцінка актуальності на новизни	19
3 Постановка задачі	21
4 Теоретичне дослідження	24
4.1 Архітектура та проектування ПЗ	24
4.2 Проектування структури зберігання даних	26
4.3 Проектування схеми роботи системи рекомендацій	30
4.4 UI/UX дизайн системи	31
5 Практичне дослідження.....	34
5.1 Опис проведеного дослідження	34
5.2 Аналіз отриманих результатів дослідження.....	42
6 Опис розробленої програмної системи.....	46
Висновки	55
Перелік джерел посилання	57
Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії	59
Додаток А Апробація результатів роботи	60
Додаток Б Код рекомендаційної системи.....	66
Додаток В Слайди презентації.....	77
Додаток Г Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ.....	86
Додаток Д Експертний висновок результатів перевірки кваліфікаційної роботи .	87

ВСТУП

Рекомендаційні системи відіграють все більшу роль у сучасному світі, сприяючи оптимізації вибору та прийняття рішень у різних сферах людської діяльності. У контексті туризму, де існує велике різноманіття можливих локацій для відвідування, такі системи дозволяють індивідуалізувати пропозиції для користувачів, враховуючи їхні уподобання, бюджет, часові рамки та багато інших факторів. Проблема вибору ідеального місця для відпочинку чи культурного дозвілля стає все більш актуальною через зростання обсягів інформації та швидкий розвиток туристичної індустрії. Використання рекомендаційних систем може не лише підвищити задоволення користувачів, але й сприяти сталому розвитку туризму, оптимізуючи потоки туристів та знижуючи навантаження на популярні локації.

Основною метою роботи є розробка ефективного підходу до створення рекомендаційної системи, яка враховувала б індивідуальні потреби користувачів для вибору оптимальних локацій для відвідування. Для досягнення цієї мети поставлено такі завдання:

- аналіз існуючих підходів і алгоритмів у рекомендаційних системах;
- вивчення специфіки туристичної галузі як предметної області;
- розробка алгоритмів, здатних адаптувати рекомендації до особистих уподобань користувачів;
- проведення тестування та оцінки запропонованої системи на реальних даних.

Об'єктом дослідження є процес формування індивідуальних рекомендацій для вибору локацій, що базується на аналізі даних користувачів та характеристик локацій.

Предметом дослідження є методи і алгоритми рекомендаційних систем, орієнтовані на індивідуалізацію вибору локацій для відвідування.

У роботі використано системний підхід для вивчення рекомендаційних систем. Було застосовано методи теоретичного аналізу для огляду існуючих підходів і визначення ключових характеристик систем, методи математичного

моделювання для створення та налаштування алгоритмів рекомендацій, а також емпіричні методи для тестування системи на реальних даних. Комплексний аналіз включав використання статистичних методів для оцінки точності рекомендацій та аналізу результатів.

У ході роботи було досліджено рекомендаційну систему, яка забезпечує адаптивність і персоналізацію пропозицій для користувачів. Використання системи сприятиме підвищенню якості послуг із застосуванням рекомендацій для відвідування локацій та задоволеності клієнтів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Специфіка досліджуваного напрямку

Рекомендаційні системи є важливим інструментом у сфері інформаційних технологій, які активно використовуються у багатьох галузях, включаючи електронну комерцію, медицину, освіту та туризм. У контексті туристичної галузі рекомендаційні системи спрямовані на допомогу користувачам у виборі локацій для відвідування на основі аналізу їхніх уподобань, поведінкових даних та характеристик об'єктів. Аналог подібної системи можна побачити на прикладі зображеному на рис. 1.1. Специфіка туристичної галузі полягає у великій різноманітності пропозицій, сезонності, регіональних відмінностях та індивідуальних потребах користувачів. Це створює виклики для розробників рекомендаційних систем, оскільки необхідно забезпечити персоналізацію рекомендацій і врахувати фактори, які впливають на вибір туристів, або просто людей, що шукають цікаві місця.



Рисунок 1.1 – Приклад системи для пошуку та вибору локацій (за даними [1])

Особливість туристичної галузі полягає в її динамічності та багатогранності. Туристичний ринок характеризується великою кількістю пропозицій, які постійно оновлюються і залежать від таких факторів, як сезонність, популярність напрямків, культурні та природні особливості регіонів. Крім того, існують значні регіональні

відмінності: туристичні уподобання варіюються залежно від культурного контексту, кліматичних умов та економічних можливостей. Наприклад, популярність морських курортів може змінюватися залежно від пори року, тоді як історичні пам'ятки залишаються затребуваними незалежно від сезону.

Ключовою специфікою є також необхідність персоналізації. Кожен турист має унікальні потреби, які залежать від його віку, інтересів, бюджету, наявного часу для подорожі та навіть фізичних можливостей. Це вимагає від рекомендаційних систем високого рівня адаптивності для формування релевантних пропозицій. Наприклад, для сім'ї з дітьми важливими будуть доступність дитячих розваг та зручність транспорту, тоді як для молодих людей пріоритетними можуть бути активний відпочинок чи нічне життя.

Також значну роль відіграє інформаційна перевантаженість. Користувачі часто стикаються з проблемою вибору через величезну кількість доступних даних про туристичні локації, відгуки інших мандрівників, рейтинги і рекомендації. У таких умовах рекомендаційні системи стають необхідним інструментом для полегшення процесу прийняття рішень. Вони допомагають фільтрувати інформацію та пропонують користувачу найбільш підходящі варіанти.

Отже, специфіка рекомендаційних систем у туристичній галузі полягає у їхній здатності враховувати різноманітні фактори, такі як динамічність ринку, регіональні відмінності, індивідуальні потреби туристів та інформаційна перевантаженість. Це визначає складність завдань, які мають вирішуватися під час розробки таких систем.

1.2 Огляд існуючих підходів та обмежень

У сучасних рекомендаційних системах найчастіше застосовуються три основні підходи: контекстно-орієнтовані, колаборативні фільтри та гібридні моделі [2].

Контекстно-орієнтовані підходи (див. рис. 1.2) базуються на аналізі властивостей об'єктів, таких як категорії, ключові слова або метайнформація, і співставленні їх із вподобаннями користувачів. Наприклад, система може

рекомендувати історичні пам'ятки користувачу, який цікавиться культурою та мистецтвом. Головною перевагою такого підходу є його незалежність від наявності даних про інших користувачів. Проте цей метод часто має обмеження у вигляді недостатньої гнучкості, оскільки не враховує нові або несподівані інтереси користувачів.

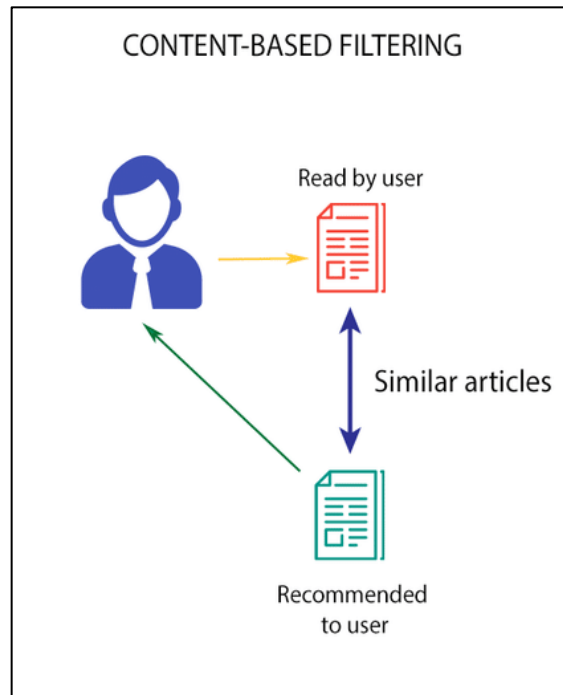


Рисунок 1.2 – Схема роботи рекомендацій на основі контенту (за даними [3])

Колаборативні підходи (див. рис. 1.3) орієнтуються на аналіз взаємодій між користувачами. Вони рекомендують об'єкти на основі уподобань схожих користувачів. Наприклад, якщо декілька людей з подібними інтересами оцінили певну туристичну локацію високо, система може запропонувати її новому користувачу. Однак цей підхід стикається з проблемою «холодного старту», коли відсутні дані про нових користувачів або об'єкти.

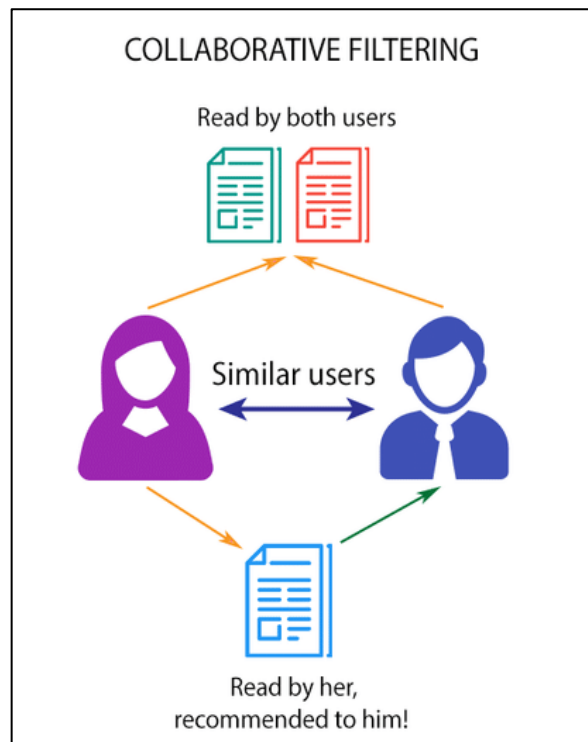


Рисунок 1.3 – Схема роботи колаборативних рекомендацій (за даними [3])

Гібридні моделі поєднують переваги обох підходів. Вони можуть використовувати одночасно дані про властивості об'єктів та уподобання інших користувачів, що забезпечує високу точність рекомендацій. Незважаючи на це, розробка таких моделей є складнішою та потребує значних обчислювальних ресурсів.

Крім того, існують і специфічні обмеження, пов'язані з туристичною галуззю: необхідність обробки динамічних даних, таких як ціни, розклад транспорту чи погодні умови. Це вимагає від систем гнучкості та адаптивності для роботи в реальному часі.

1.3 Тенденції та перспективи розвитку

Сучасні тенденції розвитку рекомендаційних систем включають активне впровадження новітніх технологій, таких як глибоке навчання, контекстно-орієнтовані підходи та інтеграцію із зовнішніми джерелами даних, такими як соціальні мережі чи геолокаційна інформація.

Глибоке навчання дозволяє створювати моделі, які можуть виявляти складні патерни в даних і забезпечувати високу якість рекомендацій [4]. Наприклад, рекурентні нейронні мережі (RNN) можуть аналізувати послідовності дій користувачів [5], тоді як згорткові нейронні мережі (CNN) ефективно працюють із візуальними даними, такими як фотографії локацій [6].

Контекстно-орієнтовані підходи, які враховують зовнішні фактори, такі як час, погода, місце перебування користувача, стають усе більш популярними. Наприклад, рекомендація може змінюватися залежно від того, чи подорожує користувач улітку або взимку, чи перебуває він у мегаполісі або гірському регіоні.

Інтеграція із зовнішніми джерелами даних, такими як соціальні мережі, огляди та рейтинги на онлайн-платформах, також є перспективним напрямком. Це дозволяє враховувати не лише прямі вподобання користувачів, але й їхню взаємодію з іншими людьми та контентом. Наприклад, аналіз геолокаційних даних може допомогти визначити популярні туристичні маршрути або улюблені місця місцевих жителів [7].

У перспективі розвиток рекомендаційних систем у туристичній галузі може зосереджуватися на створенні багатофункціональних платформ, які одночасно враховуватимуть уподобання користувачів, логістичні обмеження, бюджет та інші фактори. Це дозволить формувати інтегровані рекомендації, які будуть максимально персоналізованими та корисними для користувачів.

2 ОГЛЯД Й АНАЛІЗ ЛІТЕРАТУРНИХ, НАУКОВИХ ДЖЕРЕЛ

2.1 Огляд основних джерел

Основою для дослідження стали наукові праці та статті, присвячені рекомендаційним системам, вибору туристичних локацій і пов'язаним алгоритмам. Джерела обиралися за авторитетністю – науковий журнал IEEE, актуальністю – роботи за останні 7 років для врахування новітніх досягнень, об'єктивністю та достовірністю – публікації із відкритими результатами досліджень.

Одним з основних джерел є «A Hybrid Recommendation System: A Review» [8]. Стаття опублікована в авторитетному журналі IEEE, що гарантує високу якість дослідження. Автори мають значний досвід у галузі розробки гібридних рекомендаційних систем. Робота охоплює сучасні тенденції у створенні рекомендаційних систем, особливо в контексті об'єднання кількох методів для покращення точності рекомендацій. Огляд ґрунтується на останніх дослідженнях та враховує актуальні виклики галузі. Автори висвітлюють різні методології та порівнюють їх ефективність без явного акценту на перевагах певного підходу, що підтверджує об'єктивність аналізу. Результати статті базуються на систематичному огляді великої кількості джерел, що додає достовірності викладеній інформації.

Іншим основним джерелом є «Hybrid Recommender System for Tourism Based on Big Data and AI: A Conceptual Framework» [9]. Ця робота створена авторами, які глибоко аналізують сучасні тенденції в галузі туризму та великих даних, що підтверджується якісним рівнем структурування матеріалу, використанням актуальних даних та методів, а також публікацією у провідному науковому журналі, що спеціалізується на тематиці інформаційних технологій. Стаття стосується впровадження сучасних технологій великих даних та штучного інтелекту в рекомендаційні системи для туризму, що робить її вкрай актуальною в умовах швидкого розвитку туристичної галузі. Автори пропонують концептуальну основу, враховуючи можливі переваги та недоліки підходу, що дозволяє зберігати баланс між позитивними та негативними аспектами дослідження. Викладені висновки підкріплені посиланнями на попередні перевірені дослідження, що свідчить про високий рівень достовірності.

До основних джерел можна також віднести «An Efficient Hybrid Recommendation Model With Deep Neural Networks» [10]. Стаття належить групі дослідників, які спеціалізуються на розробці рекомендаційних моделей із застосуванням глибоких нейронних мереж. У статті розглянуто гібридну модель, яка поєднує традиційні алгоритми рекомендацій із сучасними підходами на базі глибокого навчання. Це робить її релевантною для вирішення сучасних завдань. Автори наводять детальний аналіз ефективності моделі на основі емпіричних даних, що демонструє їх прагнення до об'єктивного висвітлення переваг та обмежень підходу. Експериментальні результати представлені з чіткою методологією, що підтверджує надійність та достовірність викладеної інформації.

Заслуговує уваги наступна робота «Linked Open Data in Location-Based Recommendation System on Tourism Domain: A Survey» [11]. Дослідження виконане з дотриманням високих стандартів наукового аналізу, що підтверджується ретельністю методології, використанням загальноприйнятих міжнародних стандартів обробки даних та публікацією у спеціалізованому виданні, яке має високу репутацію в академічних колах. Стаття аналізує використання відкритих даних у рекомендаційних системах для туризму, що відповідає сучасним вимогам прозорості та доступності даних. Автори надають всебічний огляд можливостей і викликів використання відкритих даних, уникнувши однобічного трактування результатів. Використані джерела та стандарти, зокрема «Linked Open Data», є загально визнаними у науковому середовищі, що підтверджує високу достовірність роботи.

Також, розглядалась робота «Recommendation System Issues, Approaches and Challenges Based on User Reviews» [12]. Робота належить авторському колективу, який досліджує рекомендаційні системи з акцентом на аналізі відгуків користувачів. У статті розглядаються сучасні виклики та проблеми, що виникають при впровадженні рекомендаційних систем, особливо з урахуванням зворотного зв'язку від користувачів. Аналіз різних підходів та обговорення їх переваг і недоліків свідчать про нейтральність і прагнення до об'єктивності. Достовірність

забезпечується використанням широкого спектру емпіричних даних і різних джерел для обґрунтування висновків.

Огляд літератури демонструє, що сучасні підходи забезпечують високу якість рекомендацій, але мають обмеження, пов'язані з розрідженістю даних, адаптацією до контексту та ефективністю обчислень. Прогалини включають недостатню увагу до обробки географічної інформації та аналізу контекстуального оточення. Це обґрунтовує необхідність подальших досліджень та інтеграції гібридних алгоритмів із застосуванням геолокаційних даних.

2.2 Аналіз літератури

Рекомендаційні системи є одним із ключових напрямів розвитку штучного інтелекту та машинного навчання. Аналіз літератури дозволяє визначити основні підходи, теорії та моделі, що використовуються для розробки рекомендаційних систем у контексті туристичних локацій. Розглянуто ключові роботи, які охоплюють сучасні методи, включаючи гібридні системи, використання великих даних та штучного інтелекту, нейронні мережі, а також інтеграцію відкритих даних.

Гібридні рекомендаційні системи об'єднують кілька підходів для покращення якості рекомендацій, уникаючи недоліків окремих методів. Згідно зі статтею «A Hybrid Recommendation System: A Review», ці системи поєднують колаборативну фільтрацію та контентний підхід для підвищення точності результатів. Головна теоретична база ґрунтується на комбінуванні методів для адаптації до різноманітних потреб користувачів. У статті «Hybrid Recommender System for Tourism Based on Big Data and AI: A Conceptual Framework» запропоновано концептуальну основу, яка використовує великі дані та методи машинного навчання для аналізу поведінки користувачів. Ця концепція є перспективною завдяки здатності обробляти великі обсяги даних із різних джерел, таких як соціальні мережі, геолокаційні сервіси та відгуки. Таким чином, гібридні системи мають значний потенціал у подоланні обмежень традиційних підходів, особливо в умовах обмеженості даних про користувача.

Застосування нейронних мереж значно розширило можливості гібридних моделей, як зазначено в «An Efficient Hybrid Recommendation Model With Deep Neural Networks». Ця робота демонструє ефективність глибокого навчання в аналізі складних патернів поведінки користувачів. Головна ідея полягає у використанні багаторівневих нейронних мереж для створення персоналізованих рекомендацій. Згідно з цією статтею, основні переваги підходу полягають у здатності враховувати як явні, так і неявні характеристики користувачів. Однак, модель потребує значних обчислювальних ресурсів і великих обсягів навчальних даних, що є викликом для її впровадження. Отже, можна сказати, що глибокі нейронні мережі демонструють високу ефективність у створенні персоналізованих рекомендацій, проте потребують оптимізації ресурсів.

Робота «Linked Open Data in Location-Based Recommendation System on Tourism Domain: A Survey» аналізує використання відкритих даних у рекомендаційних системах для туризму. Автори підкреслюють значення «Linked Open Data» для забезпечення прозорості та доступності інформації. Цей підхід дозволяє інтегрувати дані з різних джерел, таких як бази даних туристичних об'єктів і відгуки користувачів, для створення комплексних рекомендацій. Основною проблемою є неоднорідність даних і необхідність стандартизації. Таким чином, можна дійти висновку, що інтеграція відкритих даних відкриває нові можливості для вдосконалення рекомендаційних систем, однак вимагає подолання технічних викликів.

У роботі «Recommendation System Issues, Approaches and Challenges Based on User Reviews» розглядається важливість аналізу відгуків користувачів для побудови ефективних рекомендаційних систем. Автори відзначають, що аналіз текстових і числових даних дозволяє отримати цінну інформацію про вподобання користувачів. Методика обробки текстових даних, включаючи аналіз емоцій, дозволяє точніше визначати інтереси користувачів, однак вимагає складних алгоритмів обробки природної мови (NLP). Отже, аналіз відгуків користувачів є перспективним напрямом, проте його ефективність залежить від якості алгоритмів обробки тексту.

Таким чином, всі розглянуті джерела підтверджують важливість експериментального підходу для оцінки ефективності моделей. Використання великих даних, глибоких нейронних мереж та відкритих даних демонструє позитивні результати, проте кожен метод має свої обмеження, які необхідно враховувати під час розробки нової системи. Попередні дослідження заклали міцну теоретичну та практичну основу для подальших розробок у сфері рекомендаційних систем.

2.3 Оцінка актуальності на новизни

Оцінка актуальності та новизни розглянутих джерел дозволяє визначити, наскільки представлені підходи, методи та результати досліджень відповідають сучасному стану розробки рекомендаційних систем для вибору туристичних локацій. Проаналізовані статті є прикладами передових наукових робіт, які пропонують інноваційні рішення для вирішення існуючих викликів у галузі.

Актуальність інформації у джерелах базується на сучасних тенденціях використання великих даних, штучного інтелекту та гібридних підходів у рекомендаційних системах. Наприклад, у статті «A Hybrid Recommendation System: A Review» зроблено огляд методів, які об'єднують різні підходи для подолання обмежень традиційних систем. Це відповідає сучасній потребі в універсальних і точних системах, здатних адаптуватися до змін у поведінці користувачів. У роботі «Hybrid Recommender System for Tourism Based on Big Data and AI: A Conceptual Framework» акцент зроблено на використанні великих даних, що є особливо актуальним у сфері туризму, де рішення часто ґрунтуються на різноманітних джерелах даних. Робота «An Efficient Hybrid Recommendation Model With Deep Neural Networks» демонструє використання передових нейронних мереж, що відповідає сучасній тенденції до автоматизації процесів обробки даних та прогнозування.

Кожне з джерел робить внесок у вирішення поточних проблем рекомендаційних систем: роботи, що стосуються гібридних систем, знижують ризики неточності рекомендацій та забезпечують гнучкість у роботі системи; статті

про використання відкритих даних акцентують увагу на інтеграції прозорих джерел, що є важливим для підвищення довіри користувачів до системи; дослідження відгуків користувачів сприяє створенню більш персоналізованих рішень.

Наукова новизна представлених джерел полягає у впровадженні інноваційних підходів до рекомендаційних систем: «A Hybrid Recommendation System: A Review» пропонує глибокий аналіз комбінованих методів, створюючи основу для універсальних систем; «Hybrid Recommender System for Tourism Based on Big Data and AI: A Conceptual Framework» пропонує концептуальний підхід, який розширює можливості систем за рахунок інтеграції великих даних та алгоритмів AI; «An Efficient Hybrid Recommendation Model With Deep Neural Networks» демонструє інноваційність у застосуванні глибоких нейронних мереж для оптимізації рекомендацій; «Linked Open Data in Location-Based Recommendation System on Tourism Domain: A Survey» вперше систематично аналізує переваги «Linked Open Data» у туристичній сфері; «Recommendation System Issues, Approaches and Challenges Based on User Reviews» акцентує увагу на аналізі текстових даних для створення високоточних рекомендацій, що є важливим для розвитку нових методів обробки природної мови.

Проаналізовані роботи задають нові напрями для подальших досліджень: гібридні підходи відкривають перспективи створення універсальних систем, які враховують різноманіття даних і запитів користувачів; використання великих даних та AI формує базу для масштабованих рішень, які адаптуються до потреб різних категорій користувачів; інтеграція відкритих даних сприяє створенню екосистеми прозорих та доступних інформаційних джерел; розробка моделей на основі нейронних мереж та аналізу відгуків користувачів підвищує точність і релевантність рекомендацій, що має вирішальне значення для ефективності систем.

Таким чином, актуальність і новизна розглянутих джерел підкреслює їхній значний вплив на розвиток рекомендаційних систем. Вони слугують міцною основою для побудови сучасних систем, здатних ефективно вирішувати виклики в сфері вибору туристичних локацій.

3 ПОСТАНОВКА ЗАДАЧІ

Розробка рекомендаційної системи вибору туристичних локацій є складним завданням, що потребує врахування різноманітних аспектів, включаючи потреби користувачів, особливості туристичної галузі та технологічні вимоги. Основною метою є створення системи, яка надаватиме релевантні рекомендації на основі аналізу вподобань користувачів, зовнішніх даних та поточних умов.

Для досягнення цієї мети передбачено виконання наступних завдань:

- визначення структури даних, які будуть оброблятися системою. Це включає дані про локації, профілі користувачів, а також дані про поточні умови;
- обґрунтування вибору методів та алгоритмів для побудови рекомендацій. Планується використання гібридного підходу, що об'єднує контентно-орієнтовані та колаборативні методи, з можливим застосуванням сучасних технологій машинного навчання та глибокого навчання;
- проектування архітектури системи. Система складатиметься з клієнтської частини, реалізованої за допомогою TypeScript / React, серверної частини на основі Python / Django, бази даних PostgreSQL для зберігання інформації про користувачів та локації;
- розробка та інтеграція компонентів системи. Це включає створення користувацького інтерфейсу, налаштування API для взаємодії між клієнтською та серверною частинами, а також підключення бази даних;
- тестування системи з метою оцінки її ефективності та зручності використання. Особливу увагу буде приділено якості рекомендацій, швидкодії та адаптивності до змінних умов.

Для розробки клієнтської частини використовується TypeScript / React, оскільки ця технологія забезпечує високу продуктивність, гнучкість та легкість створення сучасних інтерфейсів. Серверна частина на основі Python/Django обрана завдяки її надійності, широкому спектру можливостей та зручності роботи з API. Використання бази даних PostgreSQL буде оптимальним вибором, оскільки вона підтримує складні запити, масштабованість та інтеграцію з Django. Як середовище

розробки планується застосування Visual Studio Code через його зручність, розширюваність та підтримку обраних технологій.

У процесі розробки можуть виникати обмеження, пов'язані з:

- обсягом даних: на початковому етапі система працюватиме з обмеженою кількістю локацій та користувачів;
- часовими рамками: розробка має бути завершена в межах, визначених академічним календарем;
- обчислювальними ресурсами: тестування алгоритмів машинного навчання може потребувати значних ресурсів, що вплине на їх вибір.

Для виконання проєкту необхідні наступні ресурси: робоче середовище Visual Studio Code; програмні засоби Python, Django, TypeScript, React, PostgreSQL; дані для тестування – відкриті набори даних про локації або створені вручну тестові дані; доступ до зовнішніх джерел інформації (наприклад, API популярних платформ, таких як Google Maps та OpenStreetMap).

Задачі, які має вирішити програмна система рекомендацій для вибору локацій, включають:

- розробка зручного та інтуїтивно зрозумілого дизайну й інтерфейсу. Забезпечення користувачів інтуїтивним доступом до функцій системи. Інтерфейс повинен бути адаптивним для різних пристроїв і відповідати сучасним стандартам UI/UX-дизайну. Користувачі повинні легко отримувати рекомендації, переглядати деталі локацій та фільтрувати інформацію відповідно до їхніх потреб;
- розробка системи керування даними про локації та профілі користувачів. Створення інфраструктури для зберігання та обробки даних про локації та профілі користувачів. Забезпечення безпеки обміну інформацією між клієнтською та серверною частинами через використання протоколів шифрування;
- розробка алгоритмів пошуку, фільтрації та відображення контенту. Реалізація ефективної системи пошуку, що дозволяє користувачам знаходити релевантні локації за різними параметрами. Забезпечення

можливості гнучкої фільтрації та персоналізованого сортування результатів;

- розробка інтеграції з зовнішніми платформами. Підключення API сервісу, такого як Google Maps, для отримання додаткової інформації та карти. Забезпечення можливості отримувати актуальні дані для покращення рекомендацій;
- розробка системи рекомендацій на основі машинного навчання. Впровадження алгоритмів, що поєднують контентно-орієнтований та колаборативний підходи. Система має аналізувати історію взаємодії користувача з платформою, його вподобання та дані про поточні умови для надання релевантних рекомендацій;
- реалізація адаптивної системи для збору зворотного зв'язку. Система, яка дозволить користувачам оцінювати якість рекомендацій та залишати відгуки про локації. Цей функціонал сприятиме постійному вдосконаленню алгоритму рекомендацій.

Ці функції спрямовані на створення потужного інструменту для персоналізованого пошуку та рекомендацій локацій, який буде адаптований до потреб кінцевих користувачів.

4 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

4.1 Архітектура та проєктування ПЗ

Для розуміння можливостей користувачів була створена UML діаграма прецедентів користувачів (див. рис. 4.1).

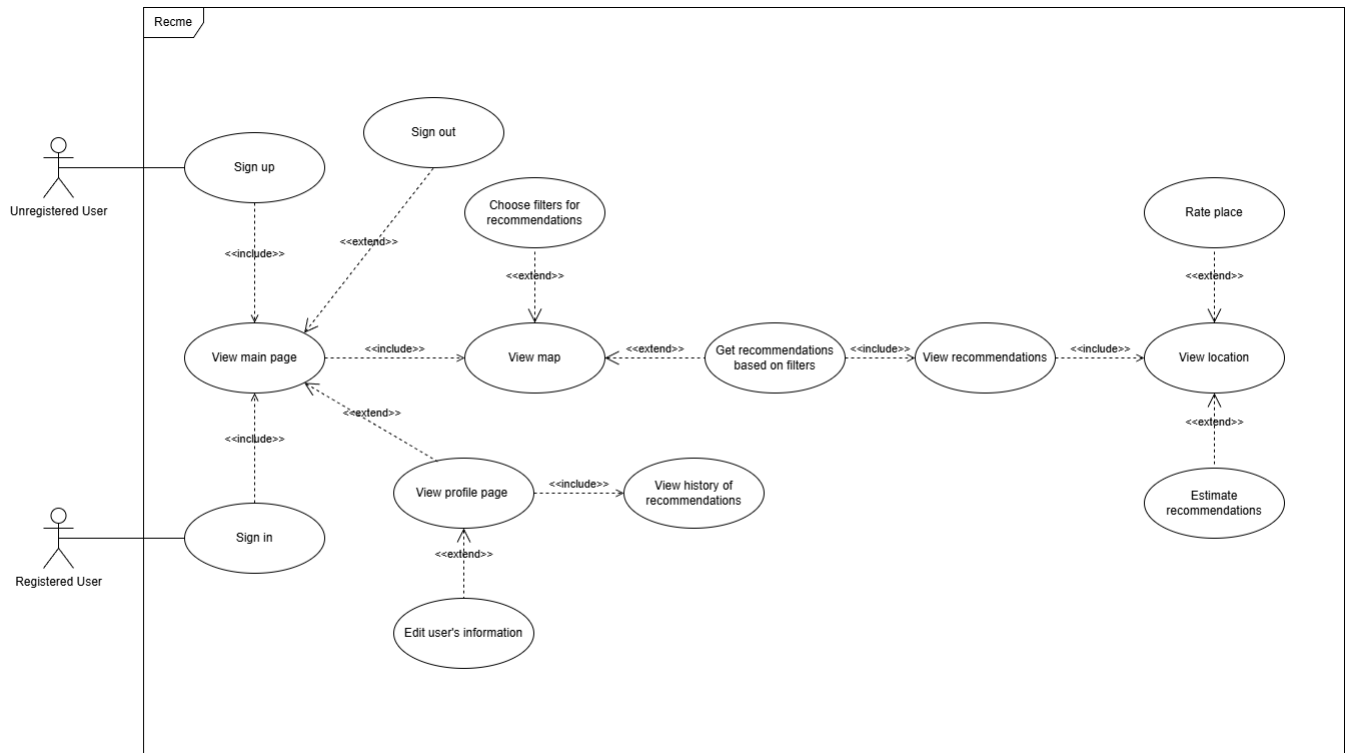


Рисунок 4.1 – UML діаграма прецедентів користувачів (рисунок виконано самостійно)

Ця діаграма прецедентів демонструє функціональні можливості системи рекомендацій та взаємодію різних категорій користувачів із системою. Система обслуговує дві основні категорії користувачів: незареєстрованих та зареєстрованих.

Незареєстровані користувачі мають змогу створити обліковий запис через процес реєстрації, після чого вони отримують доступ до основного функціоналу системи. Їх перший крок після реєстрації – це доступ до головної сторінки, яка є центральною точкою для подальшої навігації.

Зареєстровані користувачі можуть авторизуватися. Після цього вони також отримують доступ до головної сторінки, звідки можуть виконувати різні дії. Вони

мають змогу переглядати свою профільну сторінку, яка надає доступ до особистої інформації. Зареєстровані користувачі можуть змінювати свої персональні дані або переглядати історію попередніх рекомендацій. Основний функціонал системи зосереджений навколо отримання рекомендацій. Користувачі можуть перейти на карту, яка є основним інтерфейсом для взаємодії із системою. Перед отриманням рекомендацій вони можуть обирати різноманітні фільтри, що допомагає уточнити їхні запити. Використовуючи ці фільтри, система генерує рекомендації на основі наданих даних. Користувачі мають змогу переглядати ці рекомендації на карті у вигляді карток локацій. На цьому етапі вони можуть оцінити запропоновані рекомендації або поставити рейтинг окремій локації, що сприяє покращенню майбутніх результатів роботи системи. Система також надає можливість завершення сесії через вихід із системи, що доступно зареєстрованим користувачам.

Діаграма забезпечує картину основних функціональних можливостей системи та демонструє, як різні користувачі можуть взаємодіяти з її елементами для досягнення своїх цілей.

Програмна система спроектована відповідно до клієнт-серверної архітектури. UML діаграма розгортання зображена на рис. 4.2.

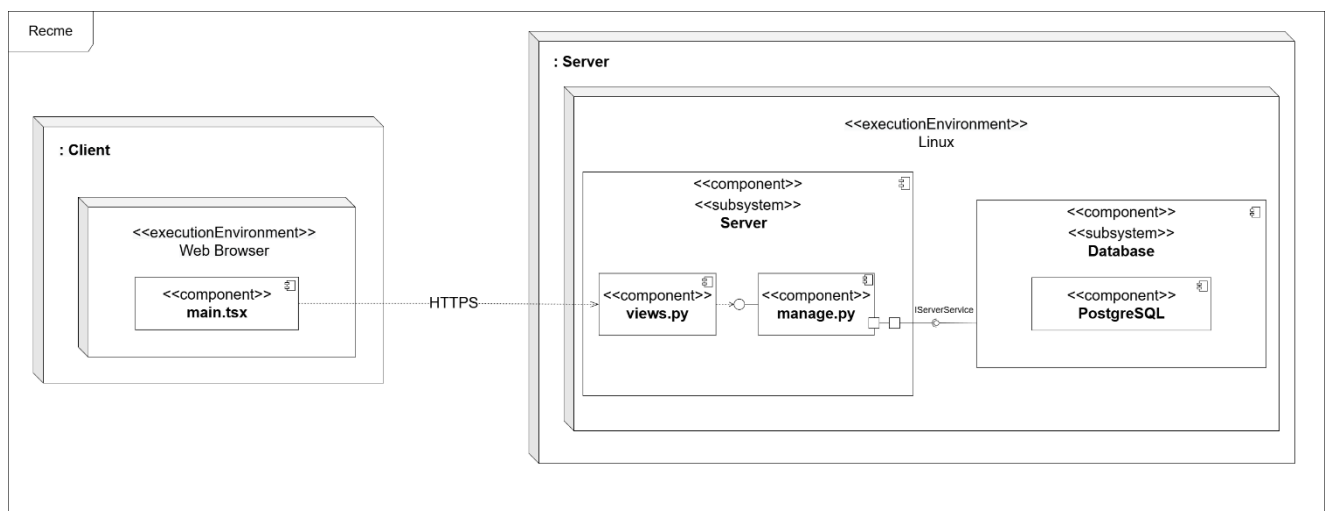


Рисунок 4.2 – UML діаграма розгортання (рисунок виконано самостійно)

Ця діаграма відображає клієнт-серверну архітектуру проєкту, де компонентами є клієнт, сервер та база даних, які взаємодіють один з одним через HTTPS-з'єднання.

Клієнтська частина представлена у вигляді браузера. Цей компонент відповідає за відображення інтерфейсу користувача, а також за збір і надсилання даних користувачів на сервер. Взаємодія клієнта із сервером забезпечується через протокол HTTPS, який гарантує захищеність передачі даних.

Серверна частина розгорнута у середовищі Linux і складається з двох основних компонентів: `views.py` та `manage.py`. Компонент `views.py` обробляє HTTP-запити від клієнта, виконує необхідну бізнес-логіку та формує відповіді. Він тісно взаємодіє з компонентом `manage.py`, який виступає керуючим модулем серверної частини, відповідає за конфігурацію, маршрутизацію, запуск серверного застосунку та управління взаємодією із базою даних.

База даних PostgreSQL також розгорнута у серверному середовищі Linux. Цей компонент використовується для зберігання всіх даних системи. Взаємодія між сервером і базою даних оптимізована для забезпечення надійності та ефективності.

Загалом діаграма демонструє чітко структуровану архітектуру, де клієнт, сервер та база даних розділені на окремі підсистеми, що забезпечує масштабованість і модульність проєкту.

4.2 Проєктування структури зберігання даних

При проєктуванні структури зберігання даних була створена ER-діаграма (див. рис. 4.3), яка відображає основні сутності програмної системи, відношення та зв'язки між ними. На діаграмі зображено 9 сутностей: «User», «Location», «Category», «Review», «Recommendation», «Preference», «Filter», «Condition», «Context».

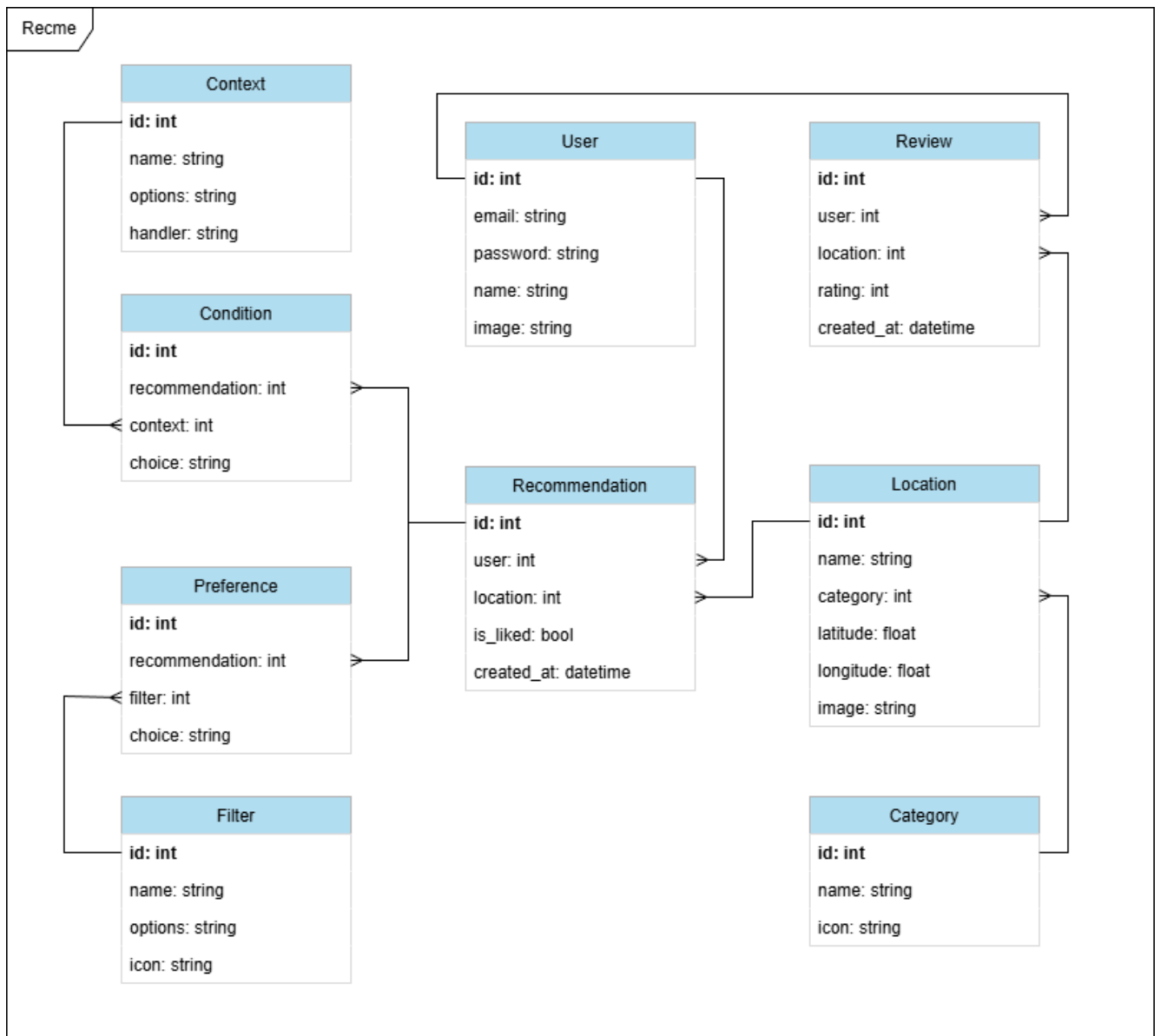


Рисунок 4.3 – ER-діаграма (рисунок виконано самостійно)

Сутність «User» несе в собі інформацію про користувача програмної системи.

Ця сутність включає в себе такі атрибути:

- id (унікальний ідентифікатор та первинний ключ);
- email (пошта);
- password (хешований пароль);
- name (ім'я користувача);
- image (зображення користувача).

Сутність «Location» несе в собі інформацію про локації програмної системи.

Ця сутність включає в себе такі атрибути:

- id (унікальний ідентифікатор та первинний ключ);
- name (назва локації);
- category (тип локації);
- latitude (широта локації);
- longitude (довгота локації);
- image (зображення локації).

Сутність «Category» несе в собі інформацію про тип локації програмної системи. Ця сутність включає в себе такі атрибути:

- id (унікальний ідентифікатор та первинний ключ);
- name (назва категорії);
- icon (зображення категорії).

Сутність «Review» несе в собі інформацію про поставлені рейтинги локаціям користувачами програмної системи. Ця сутність включає в себе такі атрибути:

- id (унікальний ідентифікатор та первинний ключ);
- user (користувач, що поставив рейтинг);
- location (локація, якій було надано рейтинг);
- rating (значення рейтингу);
- created_at (час надання рейтингу).

Сутність «Recommendation» несе в собі інформацію про надані рекомендації користувачами програмної системи. Ця сутність включає в себе такі атрибути:

- id (унікальний ідентифікатор та первинний ключ);
- user (користувач, що отримав рекомендацію);
- location (локація, яку було рекомендовано);
- is_liked (вподобання користувачем наданої рекомендації);
- created_at (час надання рекомендації).

Сутність «Preference» несе в собі інформацію про обрані фільтри програмної системи. Ця сутність включає в себе такі атрибути:

- id (унікальний ідентифікатор та первинний ключ);
- recommendation (рекомендація з обраним фільтром);
- filter (обраний фільтр);

- choice (зроблений вибір).

Сутність «Filter» несе в собі інформацію про всі доступні фільтри програмної системи. Ця сутність включає в себе такі атрибути:

- id (унікальний ідентифікатор та первинний ключ);
- name (назва фільтру);
- options (можливі варіанти вибору);
- icon (зображення фільтру).

Сутність «Condition» несе в собі інформацію про обрані умови програмної системи. Ця сутність включає в себе такі атрибути:

- id (унікальний ідентифікатор та первинний ключ);
- recommendation (рекомендація з обраною умовою);
- context (обрана умова);
- choice (зроблений вибір).

Сутність «Context» несе в собі інформацію про всі доступні контекстні умови програмної системи. Ця сутність включає в себе такі атрибути:

- id (унікальний ідентифікатор та первинний ключ);
- name (назва умови);
- options (можливі варіанти вибору);
- handler (код обробки для отримання контекстної умови).

PostgreSQL була обрана для реалізації цього проєкту завдяки її надійності, масштабованості та широкому набору функціональних можливостей, що ідеально підходять для роботи з даними рекомендаційної системи. Вона забезпечує високу продуктивність і здатна обробляти великі обсяги даних, які генеруються користувачами під час взаємодії із системою. Крім того, PostgreSQL підтримує складні запити, реляційні та нереляційні дані, що дозволяє зберігати інформацію, необхідну для функціонування системи. Її розширюваність і активна спільнота розробників роблять її ідеальним вибором для систем, які потребують гнучкості та адаптивності.

4.3 Проектування схеми роботи системи рекомендацій

Для розуміння системи рекомендацій, було створено її схему роботи (див. рис. 4.4).

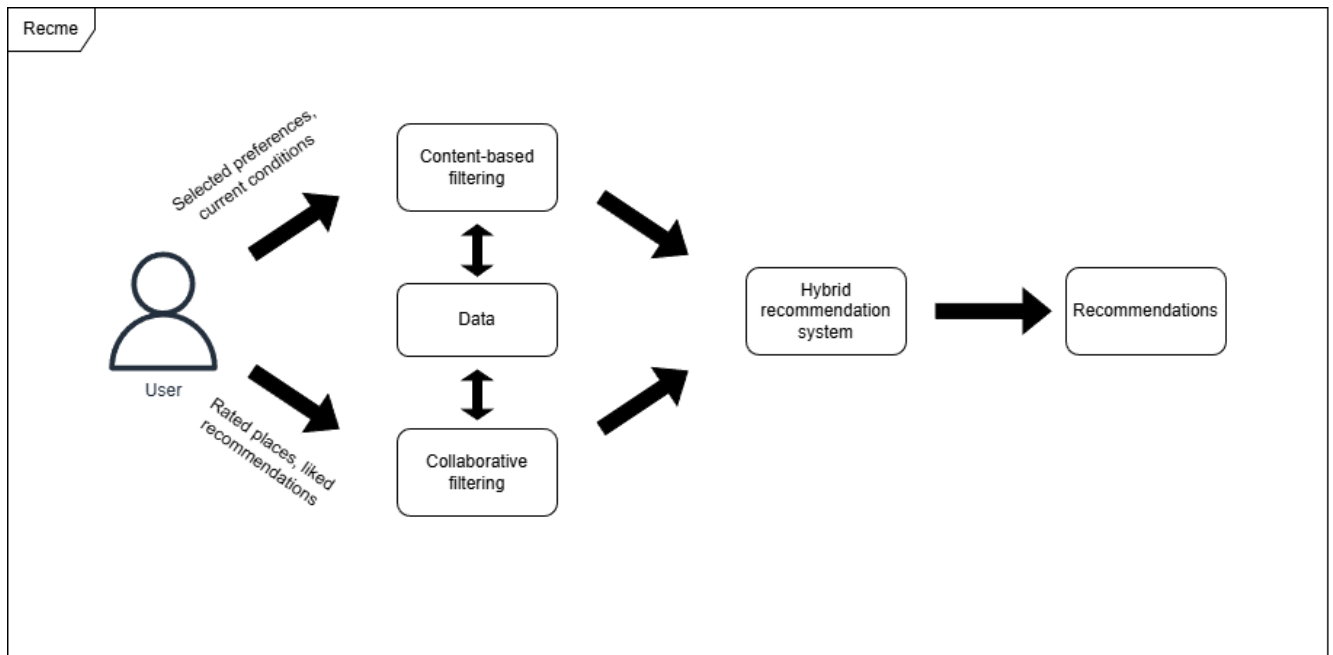


Рисунок 4.4 – Схема роботи гібридної рекомендаційної системи (рисунок виконано самостійно)

Рекомендаційна система працює за гібридним підходом, поєднуючи контентно-орієнтоване та колаборативне фільтрування для створення персоналізованих рекомендацій [13]. Система отримує на вхід дані, які включають обрані користувачем інтереси, поточні умови (час доби, погодні умови, тощо), а також оцінки, які користувач надавав місцям та рекомендації, що сподобалися.

Контентно-орієнтоване фільтрування аналізує дані про поточні умови разом із вподобаннями користувача для виявлення схожих характеристик у доступних місцях чи подіях. Це дозволяє визначити локації, які можуть відповідати особистим інтересам користувача на основі обраних фільтрів.

Колаборативне фільтрування враховує досвід інших користувачів, які взаємодіяли з системою. Воно використовує історію оцінок, схожість між вподобаннями користувачів та їхніми діями, щоб запропонувати місця, які були позитивно оцінені людьми з подібними вподобаннями.

Обидва підходи об'єднуються в гібридній рекомендаційній системі, яка зважає результати контентного та колаборативного фільтрування. Виходом системи є перелік рекомендацій, що максимально враховує персональні інтереси користувача, контекст взаємодії та досвід інших користувачів. Такий підхід забезпечує збалансованість між індивідуальними вподобаннями та колективним досвідом.

4.4 UI/UX дизайн системи

Дизайн користувацького інтерфейсу (UI) та досвід користувача (UX) є невід'ємною складовою будь-якої сучасної програмної системи, особливо такої, що пов'язана з рекомендаційними сервісами. Основна мета UI/UX дизайну полягає у створенні інтуїтивно зрозумілого, естетично привабливого та функціонального інтерфейсу, який забезпечує комфортну взаємодію користувача з системою.

Ключовим аспектом розробки дизайну є фокус на зручності використання. Інтерфейс має бути інтуїтивно зрозумілим навіть для нових користувачів, які не мають попереднього досвіду роботи з подібними системами. Для досягнення цього застосовуються принципи мінімалізму, структурованості інформації та інтерактивності. Дизайн базується на концепції Material UI – наборі компонентів і принципів, розроблених Google для забезпечення єдиного візуального стилю та функціональності. Material UI забезпечує адаптивність, що дозволяє інтерфейсу коректно відображатися на різних пристроях (мобільних, планшетах, ПК). Особливу увагу приділено адаптивності, щоб кожна складова системи автоматично підлаштовувалася під розмір екрану. Це дозволяє користувачам отримувати однаковий досвід незалежно від пристрою.

Основними елементами дизайну є панель навігації, карта, форми вводу даних та отримані рекомендації у вигляді карток локацій. Панель навігації виконує роль орієнтира для користувача, дозволяючи швидко перемикатися між основними функціональними блоками: пошуком / фільтрацією, рекомендаціями, налаштуваннями профілю тощо. Карта займає центральне місце в інтерфейсі, оскільки вона є ключовим елементом у візуалізації даних про локації.

Використовуються інтерактивні компоненти, такі як маркери на карті, які дозволяють користувачеві взаємодіяти з системою та переглядати інформацію про локації. Форми вводу даних забезпечують простоту налаштувань, таких як вибір фільтрів або зазначення параметрів пошуку. Для вводу використовуються текстові поля, випадаючі списки та перемикачі, які відповідають стилю Material UI.

Важливим елементом UX є швидкість доступу до інформації та зручність взаємодії. Для цього всі взаємодії користувача з системою організовані таким чином, щоб виконання дій займало мінімум часу. Основні кнопки (наприклад, «Отримати рекомендації») розташовані в найбільш помітних місцях, а результати рекомендацій надаються у зрозумілому вигляді з акцентом на найбільш релевантну інформацію.

Інтерактивність є невід'ємною частиною дизайну. Наприклад, при натисканні на маркер на карті анімація фокусує увагу користувача на деталях вибраної локації, а при завантаженні результатів рекомендацій застосовуються плавні переходи, що створюють відчуття безперервного процесу. Анімації використовуються мінімально, щоб не перевантажувати систему та не відволікати користувача. Вони застосовуються лише там, де це доречно, наприклад, для індикації завантаження або підтвердження виконаної дії.

Дизайн розробляється таким чином, щоб забезпечити його гнучкість для майбутніх оновлень та інтеграцій нових функцій. Використання компонентів Material UI дозволяє швидко адаптувати інтерфейс до нових вимог, а також впроваджувати модифікації без необхідності суттєвого перероблення базового коду. Завдяки цьому система залишатиметься актуальною, сучасною та зручною для користувачів навіть у разі зміни їхніх потреб або впровадження нових технологій.

Для розуміння дизайну інтерфейсу майбутньої системи було створено прототип (див. рис. 4.5).



Рисунок 4.5 – Прототип дизайну програмної системи (рисунок виконано самостійно)

Таким чином, застосування Material UI у поєднанні з чіткими принципами UI/UX дозволить створити гнучкий, сучасний та функціональний дизайн, який можна адаптувати під будь-які вимоги та особливості майбутньої системи. Цей підхід забезпечить високу якість взаємодії користувачів із системою та підвищить її конкурентоспроможність.

5 ПРАКТИЧНЕ ДОСЛІДЖЕННЯ

5.1 Опис проведеного дослідження

Розробка рекомендаційної системи починається з підготовки даних, що використовуються для навчання та тестування моделей. У даному дослідженні використано комбінований підхід до розробки системи персоналізованих рекомендацій для вибору місць для відвідування [14].

Дані, які застосовуються в роботі, включають різні джерела інформації про місця, такі як геолокаційні координати, відгуки користувачів, рейтинг популярності, категорії закладів та інші релевантні характеристики. Основним джерелом є публічні та відкриті дані Google Local Data (2021) [15]. Окрім того, враховуються індивідуальні переваги користувачів, які можуть бути отримані з історії їхніх відвідувань, оцінок місць та взаємодії з системою. Це дає змогу адаптувати рекомендації відповідно до інтересів кожного користувача.

На початковому етапі дані проходять процедуру очищення, що включає видалення неповних або некоректних записів. Видаляються дублікатні записи, усуваються помилки у категоріях, а також нормалізуються текстові дані (наприклад, знижується регістр, видаляються спеціальні символи та стоп-слова).

Після цього виконується відбір релевантних характеристик. Оскільки набір даних містить велику кількість атрибутів, деякі з них можуть бути малозначущими або корелювати між собою, що може призвести до надмірного ускладнення моделі. Тому проводиться аналіз важливості ознак, і вибираються лише ті, які найбільше впливають на результати рекомендацій. В якості найважливіших атрибутів обрано: категорія локації, рейтинг, кількість відгуків, контекстна інформація (погода, час і т.д.) та сумарна інформація (опис локації, відгуки користувачів і т.д.). Надалі, ці ознаки будуть трансформовані в числове представлення (див. додаток А).

Для ефективного навчання моделі було використано великий набір даних, що складався зі 128 000 зразків для тренувальної вибірки та 32 000 для валідаційної. Такий обсяг даних вимагав реалізації спеціального підходу до їхньої обробки, оскільки стандартні методи завантаження та трансформації могли призвести до надмірного споживання пам'яті та зниження продуктивності.

Щоб вирішити цю проблему, у системі були реалізовані нові класи для трансформування даних, які дозволяють обробляти інформацію порціонно. Це означає, що замість завантаження всього датасету в оперативну пам'ять одночасно, дані подаються в модель малими частинами, що значно оптимізує процес навчання.

Один із ключових аспектів такого підходу – збереження важливої інформації під час обробки. Оскільки система працює з різнорідними характеристиками місць, трансформаційні класи були розроблені таким чином, щоб кожен тип даних оброблявся відповідно до своєї специфіки. Наприклад, категоріальні ознаки могли бути перетворені на числові вектори, текстові дані піддавалися токенизації, а числові параметри нормалізувалися для поліпшення навчання моделі.

Завдяки цьому підходу система змогла:

- зменшити витрати пам'яті, обробляючи лише необхідні фрагменти даних у поточний момент;
- підвищити швидкість обробки, дозволяючи моделі ефективно працювати навіть із великими наборами даних;
- зберегти повноту інформації, необхідну для якісного навчання алгоритму.

Клас для трансформування категоріальних даних наведено на рис. 5.1.

```
class CategoricalEncoder:
    def __init__(self, max_categories: int, weight: int = 1) -> None:
        self._max_categories = max_categories
        self._weight = weight
        self._counter = Counter()

        self._encoder = OneHotEncoder(
            handle_unknown="ignore",
            max_categories=self._max_categories,
            sparse_output=False,
        )

    def fit(self, data: pd.DataFrame) -> None:
        self._counter.update(data.to_numpy().reshape(-1, 1).flatten())
        most_common_categories = [category[0] for category in self._counter.most_common(self._max_categories)]

        self._encoder = OneHotEncoder(
            categories=[most_common_categories],
            handle_unknown="ignore",
            max_categories=self._max_categories,
            sparse_output=False,
        )
        self._encoder.fit(data)

    def transform(self, data: pd.DataFrame) -> np.ndarray | spmatrix:
        return self._encoder.transform(data) * self._weight

    def fit_transform(self, data: pd.DataFrame) -> np.ndarray | spmatrix:
        self.fit(data)
        return self.transform(data)
```

Рисунок 5.1 – Клас для трансформування категоріальних даних (рисунок виконано самостійно)

Клас `CategoricalEncoder` призначений для трансформування категоріальних даних у формат `one-hot-encoding`. Він визначає найбільш популярні категорії (до `max_categories`) на основі частоти їх появи у вибірці та використовує `OneHotEncoder` для їх кодування. Клас підтримує ігнорування невідомих значень, дозволяє зважувати перетворені дані (`weight`) і забезпечує ефективну обробку категоріальних ознак.

Клас для трансформування числових даних наведено на рис. 5.2.

```
class NumericalScaler:
    def __init__(self, weight: int = 1) -> None:
        self._weight = weight

        self._scaler = StandardScaler()

    def fit(self, data: pd.DataFrame) -> None:
        self._scaler.partial_fit(data)

    def transform(self, data: pd.DataFrame) -> np.ndarray:
        return self._scaler.transform(data).reshape(-1, 1) * self._weight

    def fit_transform(self, data: pd.DataFrame) -> np.ndarray:
        self.fit(data)
        return self.transform(data)
```

Рисунок 5.2 – Клас для трансформування числових даних (рисунок виконано самостійно)

Клас `NumericalScaler` призначений для масштабування числових даних за допомогою `StandardScaler`. Він виконує нормалізацію ознак, приводячи їх до стандартного розподілу із середнім 0 і стандартним відхиленням 1. Клас підтримує порційне навчання (`partial_fit`), що дозволяє ефективно працювати з великими наборами даних, і може змінювати вагу значень (`weight`) для подальшого використання в моделі.

Клас для трансформування текстових даних наведено на рис. 5.3.

```

class TextVectorizer:

    def __init__(self, max_features: int, weight: int = 1) -> None:
        self._max_features = max_features
        self._weight = weight
        self._counter = Counter()

        self._vectorizer = TfidfVectorizer(
            strip_accents="unicode",
            max_features=self._max_features,
        )

    def fit(self, data: pd.DataFrame) -> None:
        self._counter.update(data.iloc[:, 0].str.cat(sep=" ").split())
        most_common_vocabulary = [word[0] for word in self._counter.most_common(self._max_features)]

        self._vectorizer = TfidfVectorizer(
            vocabulary=most_common_vocabulary,
            strip_accents="unicode",
            max_features=self._max_features,
        )
        self._vectorizer.fit(data.iloc[:, 0])

    def transform(self, data: pd.DataFrame) -> np.ndarray:
        return np.asarray(self._vectorizer.transform(data.iloc[:, 0]).todense()) * self._weight

    def fit_transform(self, data: pd.DataFrame) -> np.ndarray:
        self.fit(data)
        return self.transform(data)

```

Рисунок 5.3 – Клас для трансформування текстових даних (рисунок виконано самостійно)

Клас `TextVectorizer` призначений для перетворення текстових даних у числовий формат за допомогою TF-IDF векторизації. Він визначає найбільш частотні слова (`max_features`) у вибірці, створюючи обмежений словник, а потім перетворює текст у матрицю TF-IDF. Клас підтримує нормалізацію символів (`strip_accents="unicode"`) і дозволяє змінювати вагу (`weight`) отриманих векторів. Це забезпечує ефективну обробку текстових ознак для навчання моделі.

Для того, щоб забезпечити подачу фіксованого розміру вхідних даних до глибокої нейронної мережі, було застосовано метод головних компонент (PCA), який враховує всі ознаки одночасно. Це включає числові атрибути, кодування категоріальних змінних та текстові дані, перетворені за допомогою TF-IDF. Така трансформація дозволяє створити компактне, але інформативне представлення кожної локації, оптимізуючи простір ознак для подальшої роботи глибокої нейронної мережі. Клас для зменшення розмірності даних наведено на рис. 5.4.

```

class Compressor:

    def __init__(self, n_components: int) -> None:
        self._n_components = n_components
        self._compressor = IncrementalPCA(n_components=self._n_components)

    def fit(self, data: np.ndarray) -> None:
        self._compressor.partial_fit(data)

    def transform(self, data: np.ndarray) -> np.ndarray:
        return self._compressor.transform(data)

    def fit_transform(self, data: np.ndarray) -> np.ndarray:
        self.fit(data)
        return self.transform(data)

```

Рисунок 5.4 – Клас для зменшення розмірності даних (рисунок виконано самотійно)

Клас Compressor використовує IncrementalPCA для зменшення розмірності даних, що дозволяє зберегти найважливішу інформацію при скороченні кількості ознак до `n_components`. Він підтримує порційне навчання (`partial_fit`), що робить його ефективним для роботи з великими наборами даних. Компресія допомагає зменшити обчислювальні витрати та покращити продуктивність моделі.

Була реалізована багатошарова перцептронна нейронна мережа (FeedForward Neural Network) за допомогою PyTorch для обробки перетворених ознак та створення рекомендацій. Архітектура була розроблена з урахуванням балансу між ефективністю роботи та обчислювальною витратою, варіюючи кількість нейронів на вході, кількість прихованих шарів та швидкість навчання.

Кількість нейронів на вході змінювалась залежно від вимірності оброблених даних. Перший прихований шар містив удвічі більше нейронів, ніж вхідний шар, а кожен наступний шар мав у два рази менше нейронів, поки не досягався вихідний шар. Кожен шар супроводжувався функцією активації LeakyReLU та Dropout для запобігання перенавчанню. Для генерації ймовірності рекомендації використовувався один нейрон з функцією активації Sigmoid. Клас нейронної мережі зображено на рис. 5.5.

```

class NeuralNetwork(nn.Module):

    def __init__(self, in_dim: int, hidden_dims: list[int], out_dim: int) -> None:
        super().__init__()

        self.network = nn.Sequential(
            nn.Linear(in_dim, hidden_dims[0]),
            nn.LeakyReLU(),
            nn.Dropout(),
            *[
                layer for i in range(len(hidden_dims) - 1)
                for layer in (
                    nn.Linear(hidden_dims[i], hidden_dims[i + 1]),
                    nn.LeakyReLU(),
                    nn.Dropout()
                )
            ],
            nn.Linear(hidden_dims[-1], out_dim),
            nn.Sigmoid()
        )

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        return self.network(x)

```

Рисунок 5.5 – Клас нейронної мережі (рисунок виконано самостійно)

Клас `NeuralNetwork` є динамічною реалізацією багатошарової нейронної мережі, що використовує бібліотеку `PyTorch`.

Для дослідження та розуміння найкращої архітектури моделі, було навчено 27 моделей з використанням різних конфігурацій: `input layer` – 256, 512, 1024; `hidden layers` – 3, 5, 7; `learning rates` – 0.01, 0.001, 0.0001. Код повноцінної рекомендаційної системи наведено у додатку Б.

Початкове навчання проводилося протягом 10 епох для всіх моделей. Моделі оцінювали на основі `accuracy` та `F1-score`, а результати було візуалізовано на графіках.

`Accuracy` 27 різних моделей, зображено на рис. 5.6.

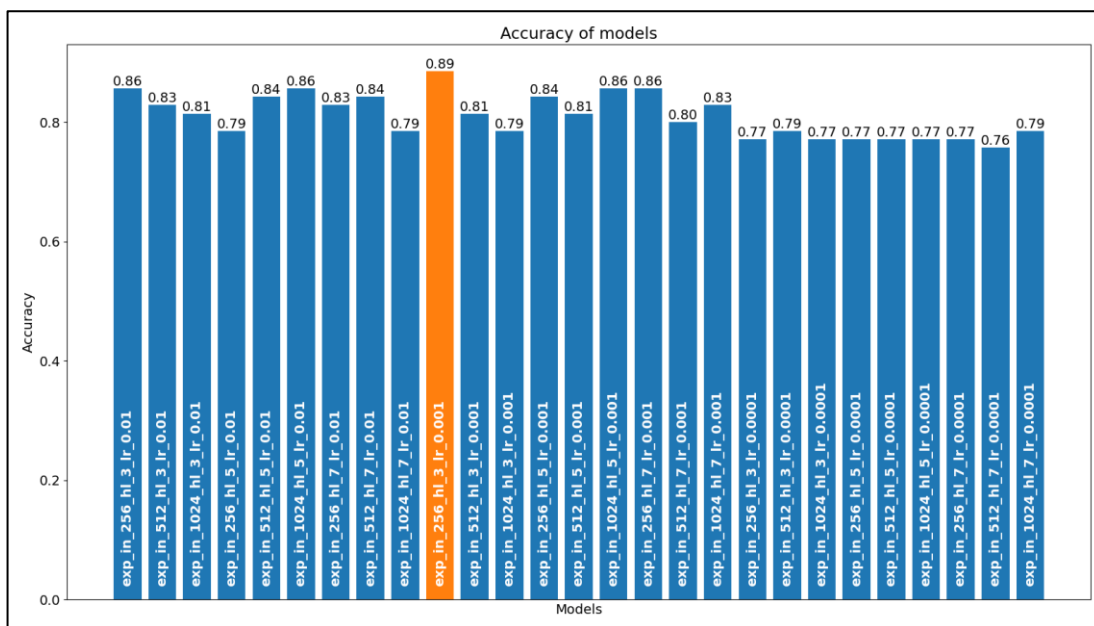


Рисунок 5.6 – Accuracy 27 різних моделей (рисунок виконано самостійно)

F1-score 27 різних моделей, зображено на рис. 5.7.

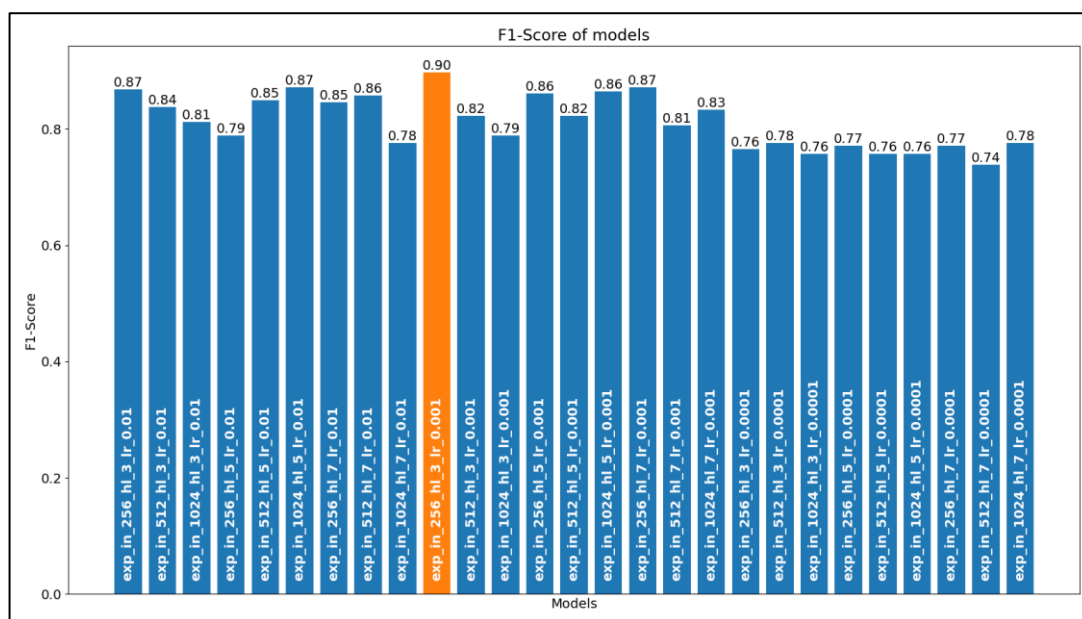


Рисунок 5.7 – F1-score 27 різних моделей (рисунок виконано самостійно)

Найефективніша модель досягла accuracy 0.89 та F1-score 0.90, демонструючи ефективність обраного гібридного рекомендаційного підходу.

Після цього, було проведено тренування 5 найкращих моделей протягом 100 epoch, щоб удосконалити продуктивність.

Accuracy 5 найкращих моделей, зображено на рис. 5.8.

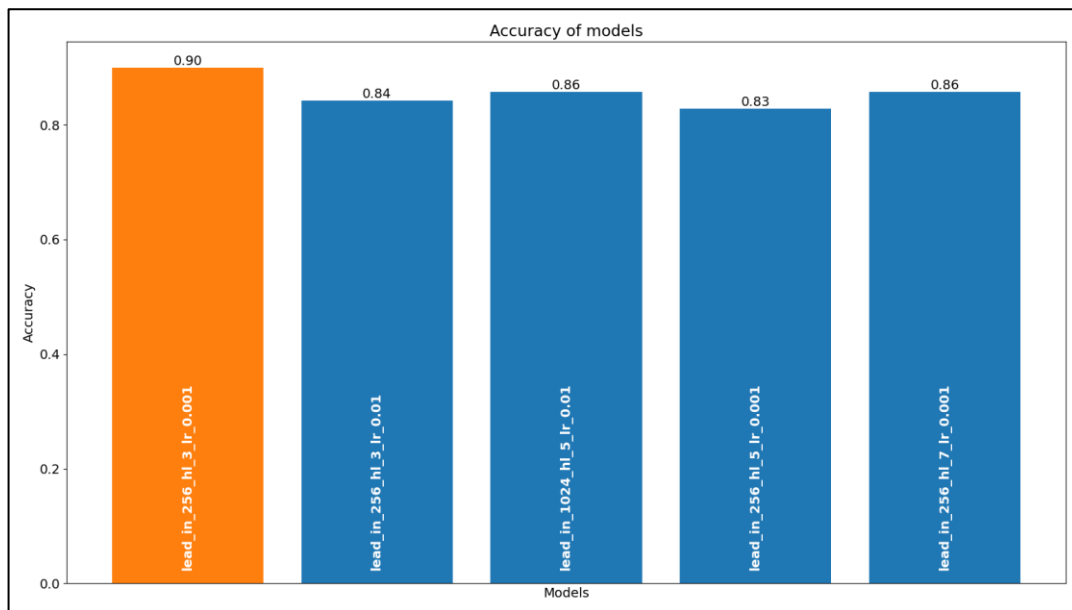


Рисунок 5.8 – Accuracy 5 найкращих моделей (рисунок виконано самостійно)

F1-score 5 найкращих моделей, зображено на рис. 5.9.

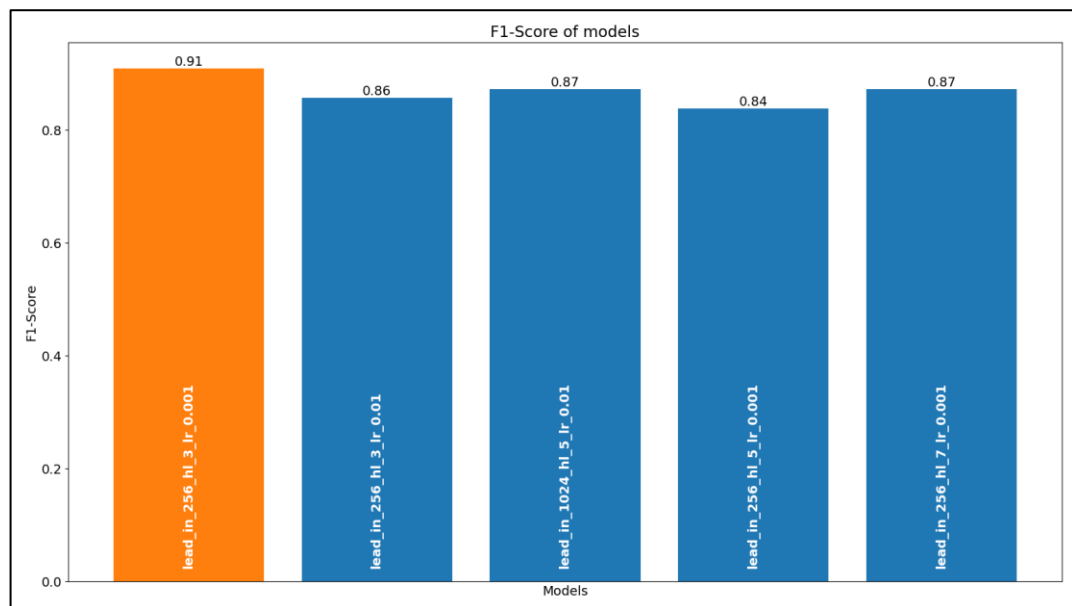


Рисунок 5.9 – F1-score 5 найкращих моделей (рисунок виконано самостійно)

Виходячи з порівняння accuracy та F1-score, найкраща модель досягла найвищої точності 0.90 та найвищої оцінки F1 0.91, що вказує на те, що вона забезпечує найкращу результативність.

5.2 Аналіз отриманих результатів дослідження

Для початку, проаналізуємо процес тренування найкращої моделі. Для цього розглянемо графік втрат (див. рис. 5.10).

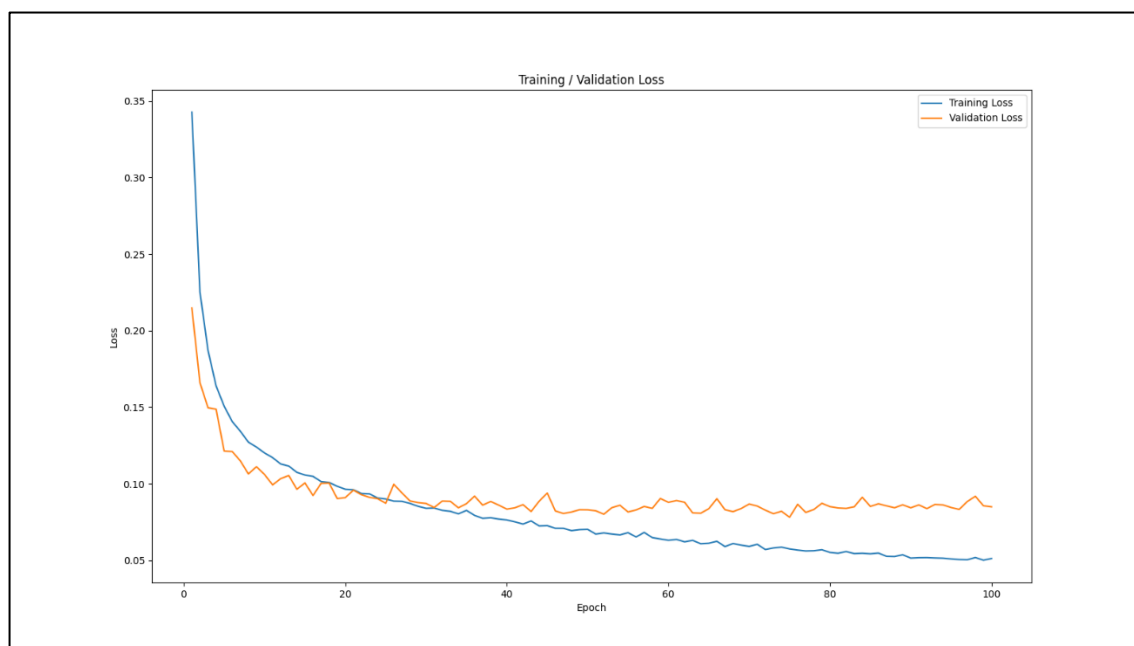


Рисунок 5.10 – Графік втрат (рисунок виконано самостійно)

Як можна побачити, на початкових епохах значення втрат зменшується дуже швидко, що є очікуваним, оскільки модель швидко адаптується до вхідних даних. Згодом крива виходить на більш стабільний рівень. Важливо зазначити, що різниця між тренувальними та валідаційними втратами невелика, що підтверджує гарне узагальнення моделі.

Щоб забезпечити найкращі результати, під час навчання моделі використовується механізм збереження найкращої версії моделі. Система відстежує мінімальне значення функції втрат на валідаційній вибірці та зберігає відповідну модель. Це дозволяє уникнути ситуації, коли модель продовжує навчання, але не покращується, а лише запам'ятовує дані навчальної вибірки.

Для навчання використовувалася Binary Cross Entropy Loss (BCELoss) з бібліотеки PyTorch. Вона добре підходить для задач бінарної класифікації, оскільки оцінює розбіжність між передбаченими ймовірностями та фактичними мітками класів.

Тепер розглянемо графік точності (див. рис. 5.11).

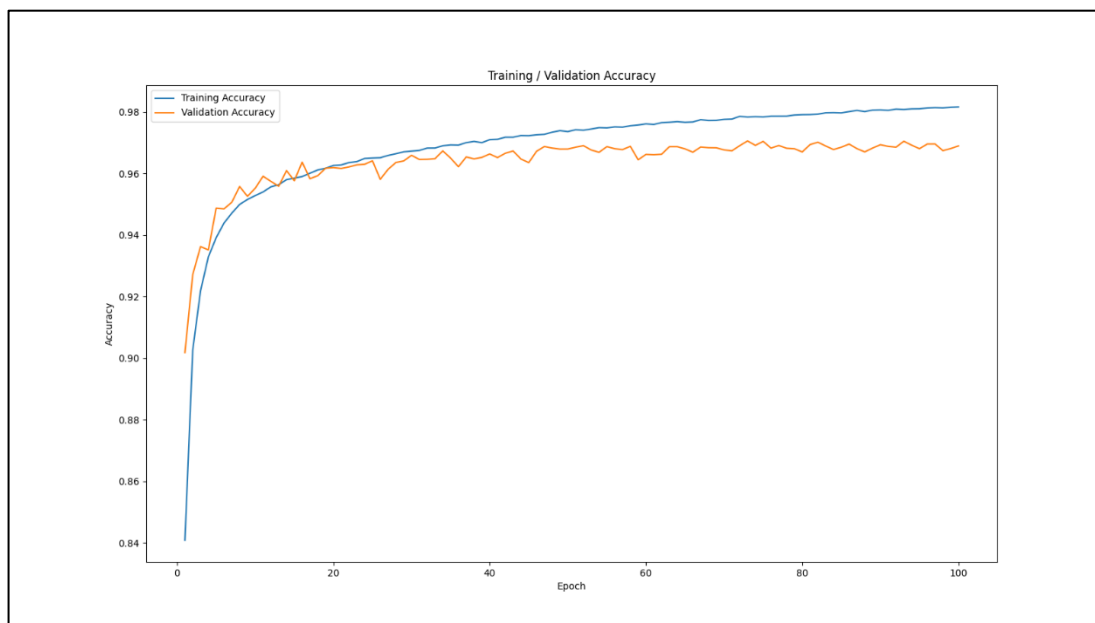


Рисунок 5.11 – Графік точності (рисунок виконано самостійно)

Як можна побачити, вже на початкових епохах точність швидко зростає, що свідчить про ефективне навчання моделі. Згодом зростання уповільнюється, але навіть після 100 епох спостерігається незначне покращення точності. Важливо зазначити, що валідаційна точність залишається близькою до тренувальної, що говорить про відсутність значного перенавчання.

Загалом – модель демонструє високу точність, що перевищує 96% на валідаційній вибірці, втрати на тренувальних та валідаційних даних залишаються низькими, відсутність значного розриву між тренувальними та валідаційними кривими свідчить про відсутність перенавчання, а завдяки використанню BCELoss та збереженню найкращої моделі, система забезпечує стабільне та надійне навчання.

Щоб проаналізувати та оцінити ефективність розробленої рекомендаційної моделі, було проведено серію експериментів (інформація про експерименти наведена в додатку А.) та проаналізовано результати з використанням декількох метрик, включаючи accuracy, F1-score, ROC-AUC curve та confusion matrix.

AUC-ROC curve, зображена на рис. 5.12, демонструє здатність моделі розрізняти класи. Площа під кривою становить 0.91, що вказує на відмінну продуктивність.

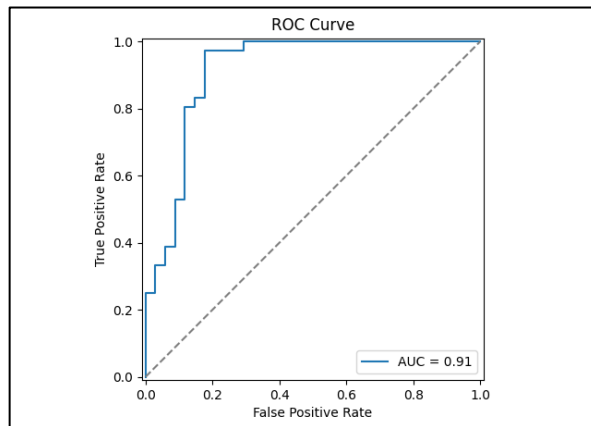


Рисунок 5.12 – AUC-ROC curve (рисунок виконано самостійно)

Confusion matrix, що зображена на рис. 5.13, показує, що модель правильно передбачила 28 істинно негативних та 35 істинно позитивних результатів, лише з 6 помилково позитивними результатами та 1 помилково негативним. Це говорить про те, що модель є надійною для рекомендаційних завдань.

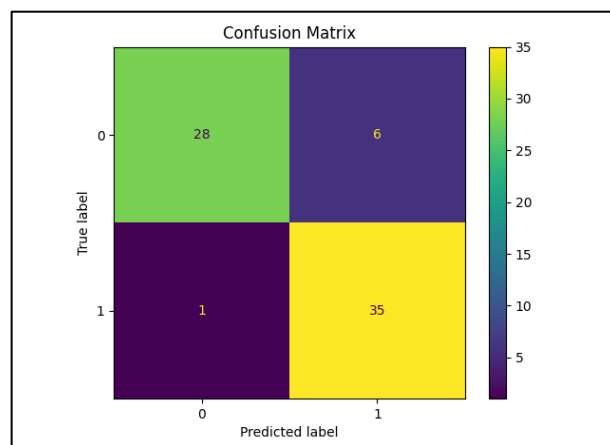


Рисунок 5.13 – Confusion matrix (рисунок виконано самостійно)

Однією з головних переваг моделі є її здатність враховувати контекстуальні фактори. Наприклад, під час тестування модель правильно уникала рекомендацій парків і майданчиків у дощову погоду, навіть якщо інші фактори підходили. Це свідчить про те, що модель враховує погодні умови та не робить наївних рекомендацій, ґрунтуючись тільки на інших характеристиках.

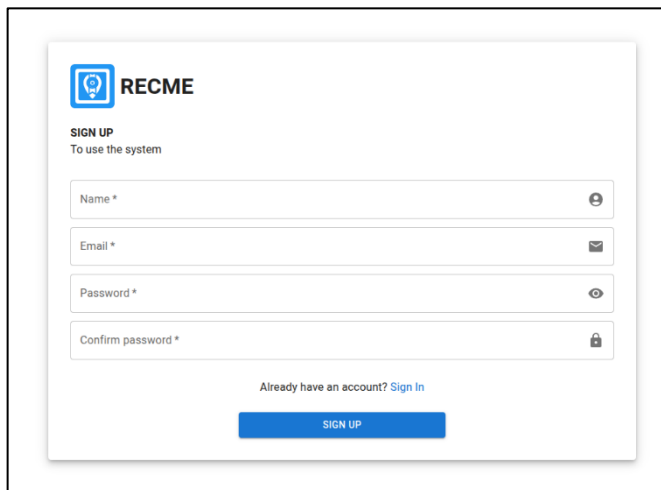
Модель також продемонструвала розуміння надійності оцінок. Вона не переоцінює високі рейтинги, коли кількість відгуків дуже низька, що дозволяє уникнути впливу потенційних упереджень або штучного завищення оцінок. З іншого боку, модель правильно знижує рейтинг місць з низькими відгуками, навіть якщо вони мають велику кількість відгуків, гарантуючи, що місця з поганими відгуками не потрапляють у рекомендації.

Хоча модель працює добре в цілому, інколи вона стикається з труднощами при фільтрації. Це може бути пов'язано з подібністю термінів між різними категоріями, що призводить до незначних неточностей. Однак загалом, модель показує хороші результати, і її здатність постійно вдосконалюватися за рахунок відгуків користувачів дозволить підвищити точність та надійність у майбутньому.

6 ОПИС РОЗРОБЛЕНОЇ ПРОГРАМНОЇ СИСТЕМИ

Розроблена програмна система є веб-додатком з адаптивним інтерфейсом під всі види пристроїв [16].

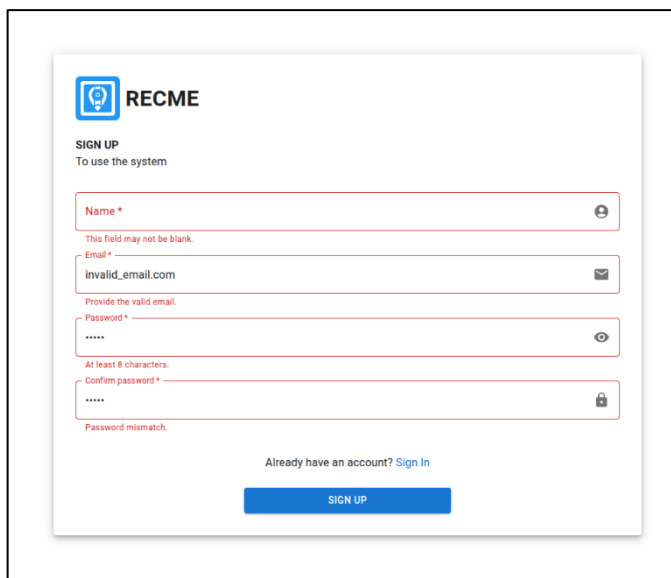
Для того, щоб розпочати використання програмної системи, користувачеві необхідно пройти реєстрації (див. рис. 6.1).



The image shows a registration form for a system named 'RECME'. The form is titled 'SIGN UP' and includes the instruction 'To use the system'. It contains four input fields: 'Name *', 'Email *', 'Password *', and 'Confirm password *'. Each field has a small icon to its right: a person for Name, an envelope for Email, an eye for Password, and a lock for Confirm password. Below the fields is a link that says 'Already have an account? Sign In' and a blue 'SIGN UP' button.

Рисунок 6.1 – Форма реєстрації (рисунок виконано самостійно)

Для того, щоб система оперувала тільки вірними даними, була вбудована валідація даних. При спробі ввести невалідні дані, користувач отримує інформацію про те, які дані були у невірному форматі (див. рис. 6.2).



The image shows the same registration form as in Figure 6.1, but with validation errors. The 'Name' field has a red border and the message 'This field may not be blank.' below it. The 'Email' field has a red border and the message 'invalid_email.com' below it. The 'Password' field has a red border and the message 'Provide the valid email.' below it. The 'Confirm password' field has a red border and the message 'At least 8 characters.' below it. At the bottom of the form, there is a red message 'Password mismatch.' and a blue 'SIGN UP' button.

Рисунок 6.2 – Валідація форми реєстрації (рисунок виконано самостійно)

Маючи профіль, користувач може пройти авторизацію для того, щоб потрапити до системи (див. рис. 6.3).

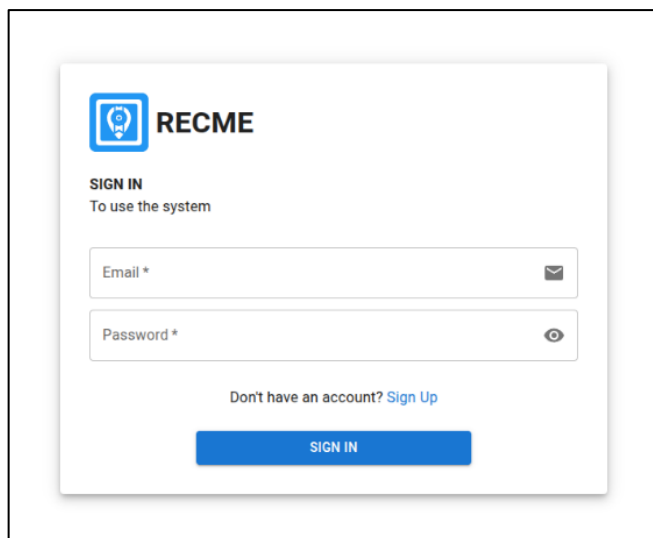
The image shows a login form for the RECME system. At the top left is the RECME logo, which consists of a blue square with a white lightbulb icon and the word "RECME" in bold black text. Below the logo, the text "SIGN IN" is displayed in bold, followed by "To use the system" in a smaller font. There are two input fields: "Email *" and "Password *". The "Email *" field contains a small envelope icon on the right. The "Password *" field contains a small eye icon on the right. Below the input fields, there is a link that says "Don't have an account? Sign Up". At the bottom of the form is a blue button with the text "SIGN IN" in white.

Рисунок 6.3 – Форма авторизації (рисунок виконано самостійно)

При спробі ввести невірні дані для авторизації, користувач отримує інформацію про те, що такого профілю не було знайдено (див. рис. 6.4).

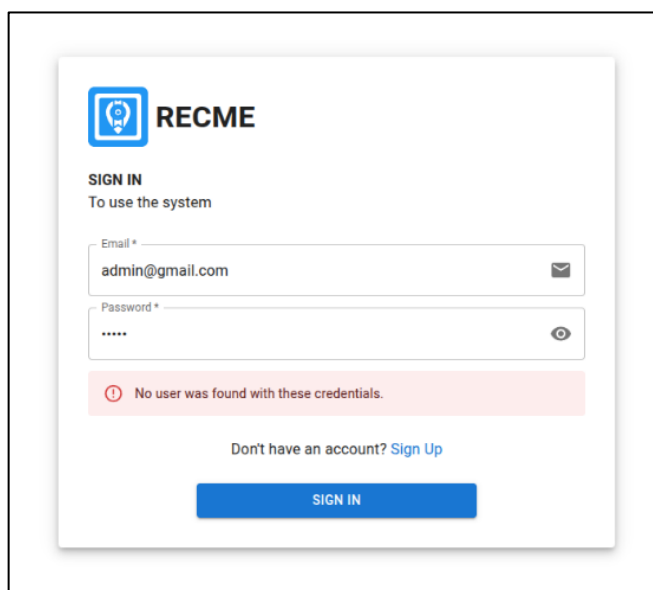
This image shows the same login form as in Figure 6.3, but with an error message. The "Email *" field now contains the text "admin@gmail.com" and the "Password *" field contains six dots. Below the input fields, a pink error message box is displayed with a red warning icon and the text "No user was found with these credentials." Below the error message, there is a link that says "Don't have an account? Sign Up". At the bottom of the form is a blue button with the text "SIGN IN" in white.

Рисунок 6.4 – Невірні дані авторизації (рисунок виконано самостійно)

В разі успішної авторизації користувач потрапляє на головну сторінку (див. рис. 6.5). На головній сторінці користувач побачить інтерактивну карту, з якою зможе взаємодіяти.

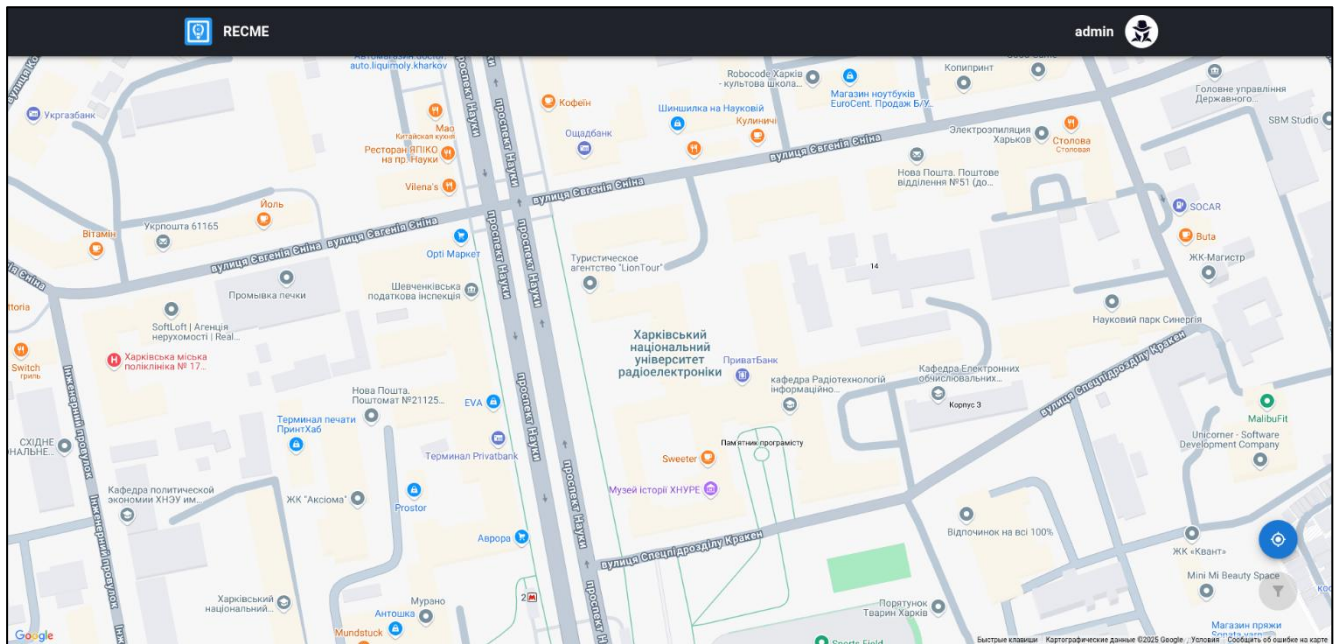


Рисунок 6.5 – Головна сторінка (рисунок виконано самостійно)

Користувач може надати можливість системи визначити свою поточну геолокацію, натиснувши відповідну кнопку (див. рис. 6.6).

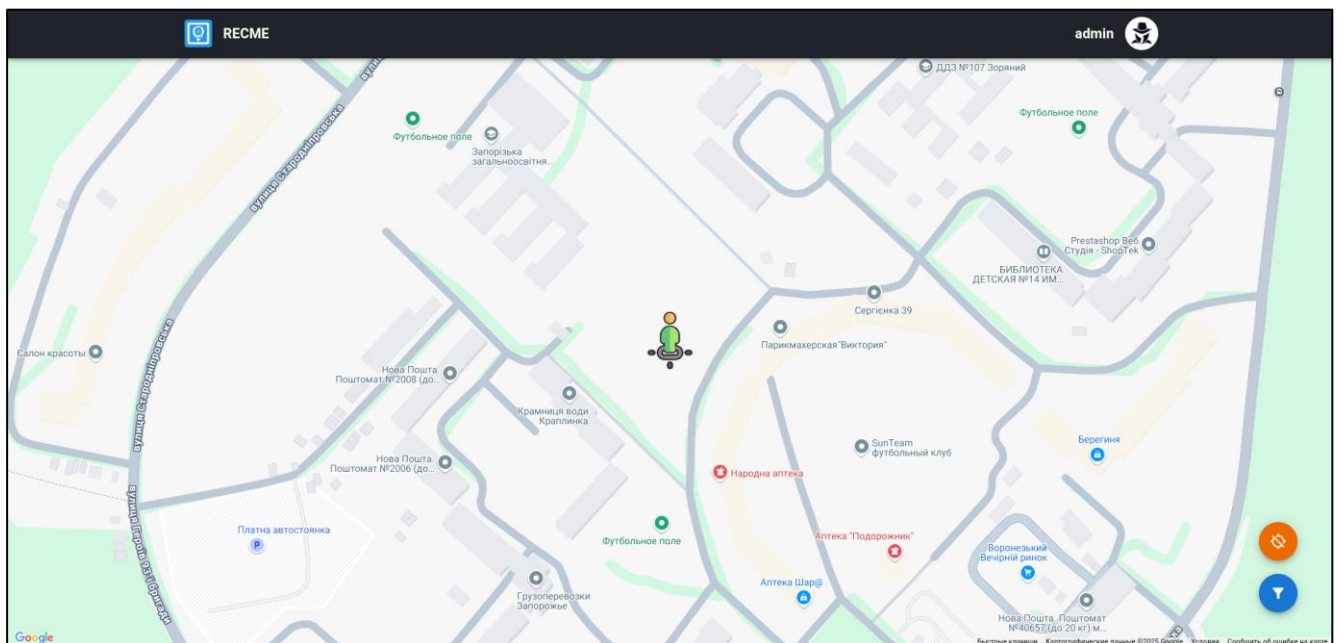


Рисунок 6.6 – Автоматична геолокація користувача (рисунок виконано самостійно)

Також, користувач може власноруч відмітити геолокацію, натиснувши на відповідне місце на карті (див. рис. 6.7).

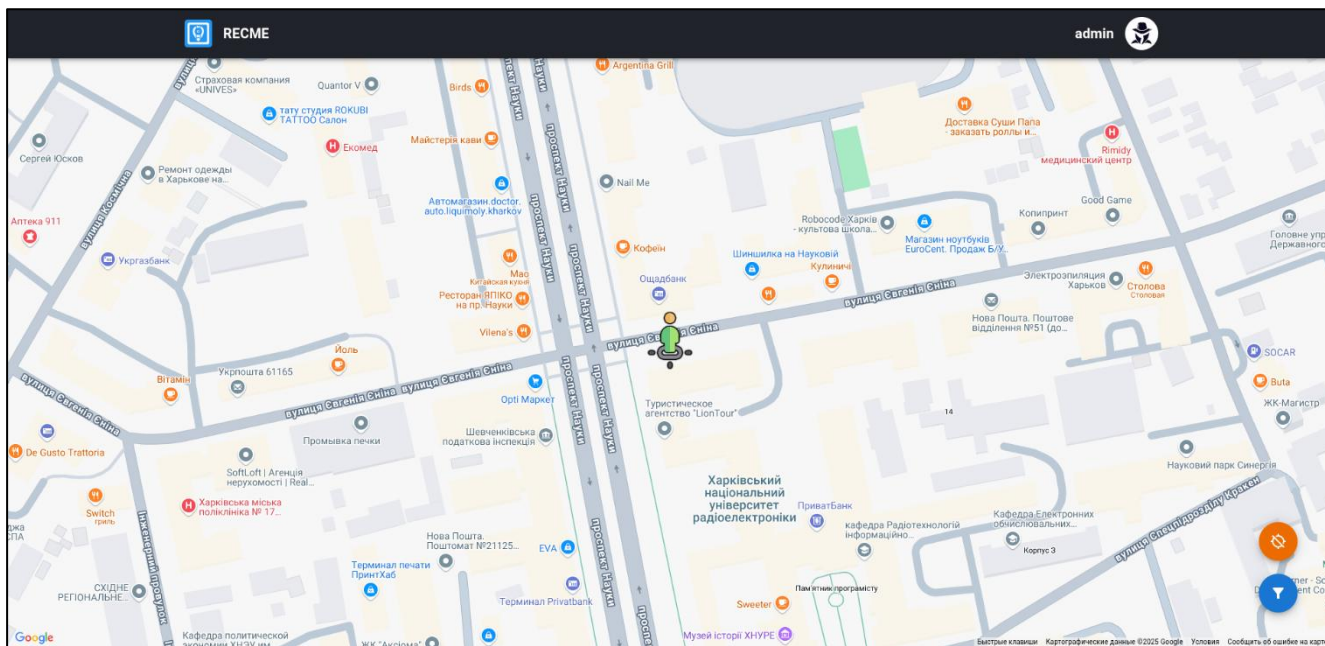


Рисунок 6.7 – Обрана геолокація користувача (рисунок виконано самостійно)

Натиснувши відповідну кнопку, користувач отримує можливість обрати необхідні фільтри пошуку (див. рис. 6.8).

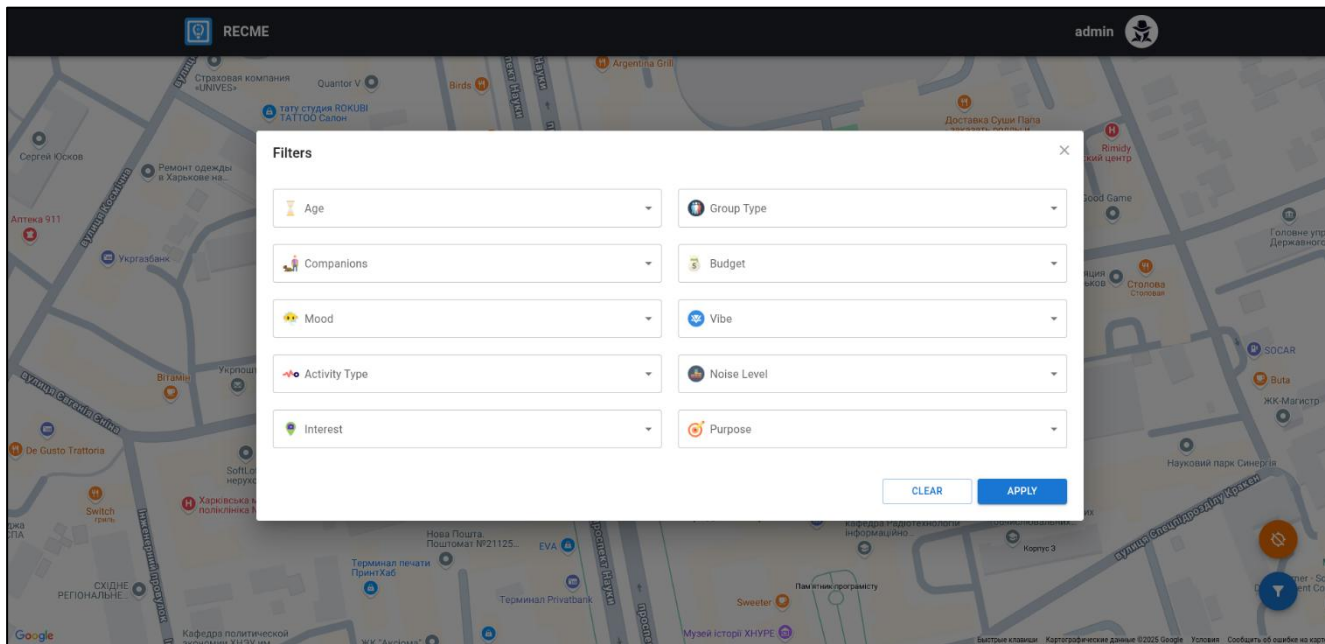


Рисунок 6.8 – Форма для обрання фільтрів пошуку (рисунок виконано самостійно)

Форма з фільтрами має просту валідацію – хоча б один фільтр має бути обраний для пошуку (див. рис. 6.9).

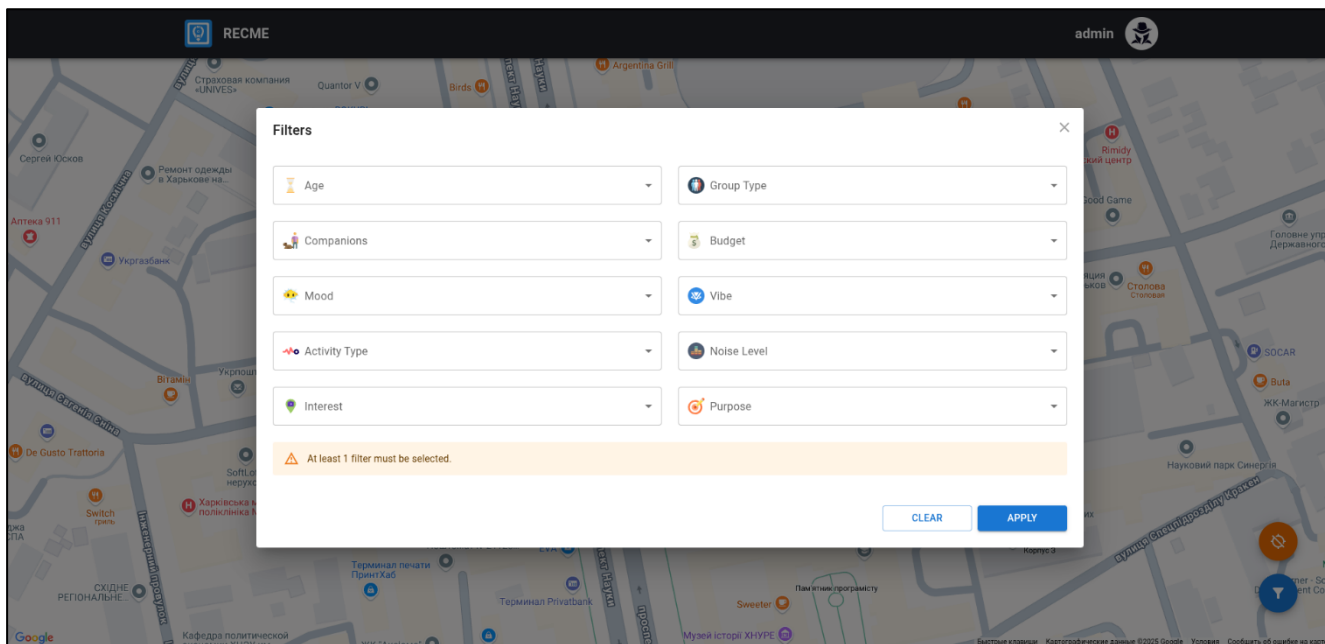


Рисунок 6.9 – Валідація форми з фільтрами (рисунок виконано самостійно)

Обравши необхідні фільтри, користувач має можливість запустити процес підбору рекомендацій натиснувши відповідну кнопку (див. рис. 6.10).

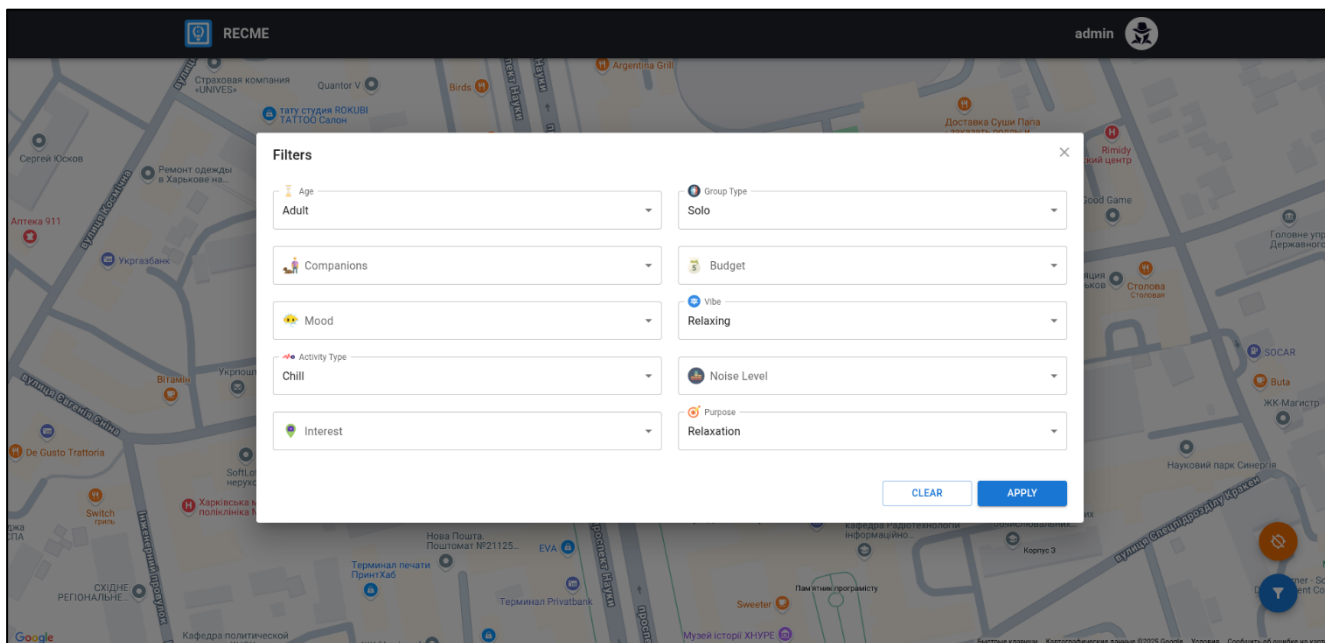


Рисунок 6.10 – Валідна форма для підбору рекомендацій (рисунок виконано самостійно)

Запустивши підбір рекомендацій, користувач отримує можливість побачити радіус пошуку та сам процес (див. рис. 6.11).

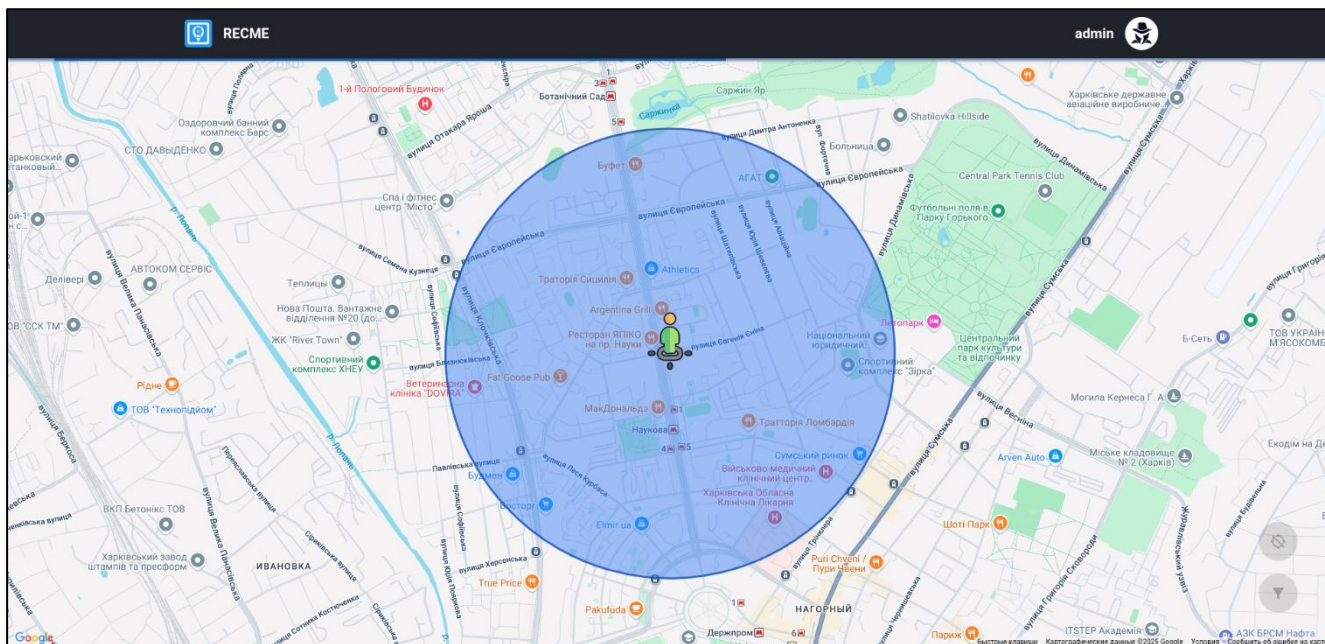


Рисунок 6.11 – Візуальне представлення підбору рекомендацій (рисунок виконано самостійно)

Після успішного підбору рекомендацій, користувач зможе побачити їх на карті та буде мати можливість взаємодіяти з ними (див. рис. 6.12).

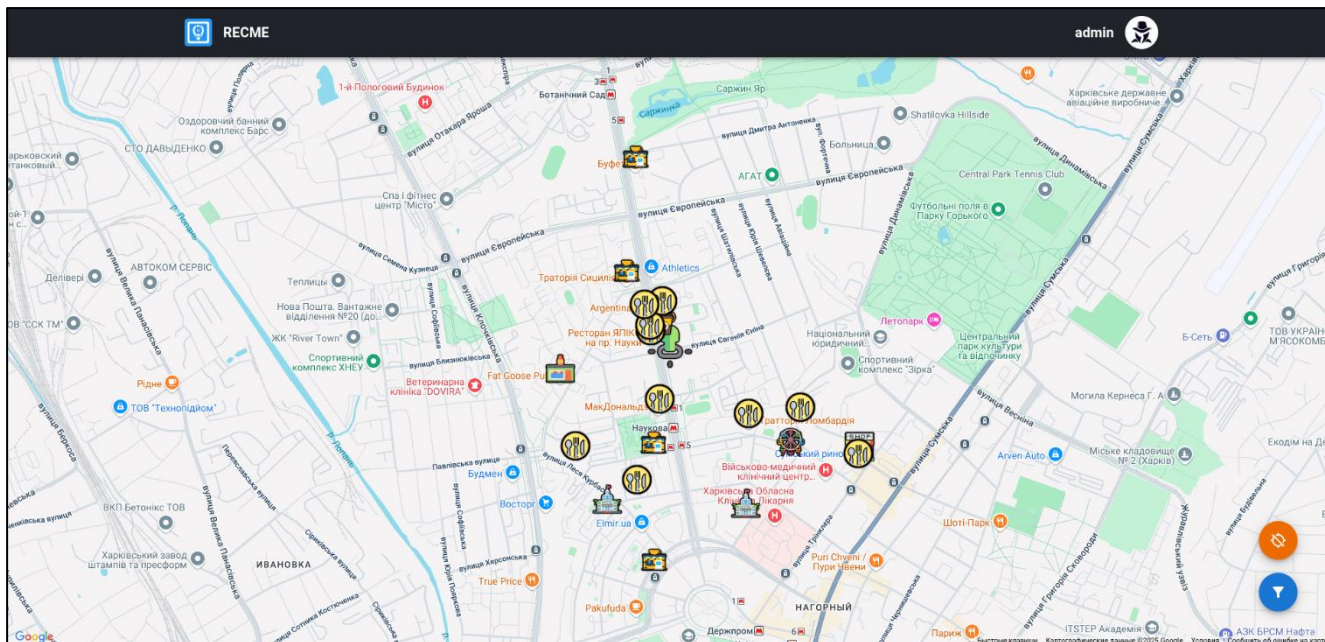


Рисунок 6.12 – Візуальне представлення підібраних рекомендацій (рисунок виконано самостійно)

Натиснувши на будь-яку рекомендацію, користувач побачить її модальне вікно (див. рис. 6.13). Синім виділені обрані користувачем фільтри, а помаранчевим контекст.

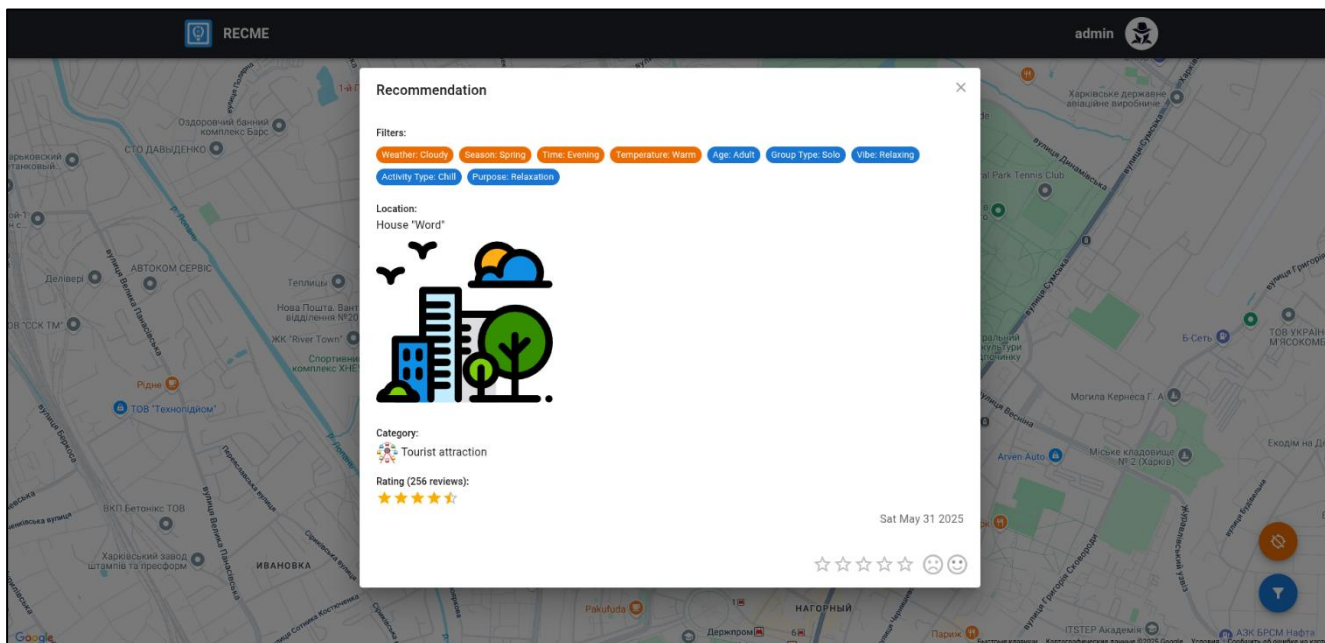


Рисунок 6.13 – Модальне вікно рекомендації (рисунок виконано самостійно)

В модальному вікні користувач має можливість проставити рейтинг локації (див. рис. 6.14), що вплине на майбутні рекомендації цього місця.

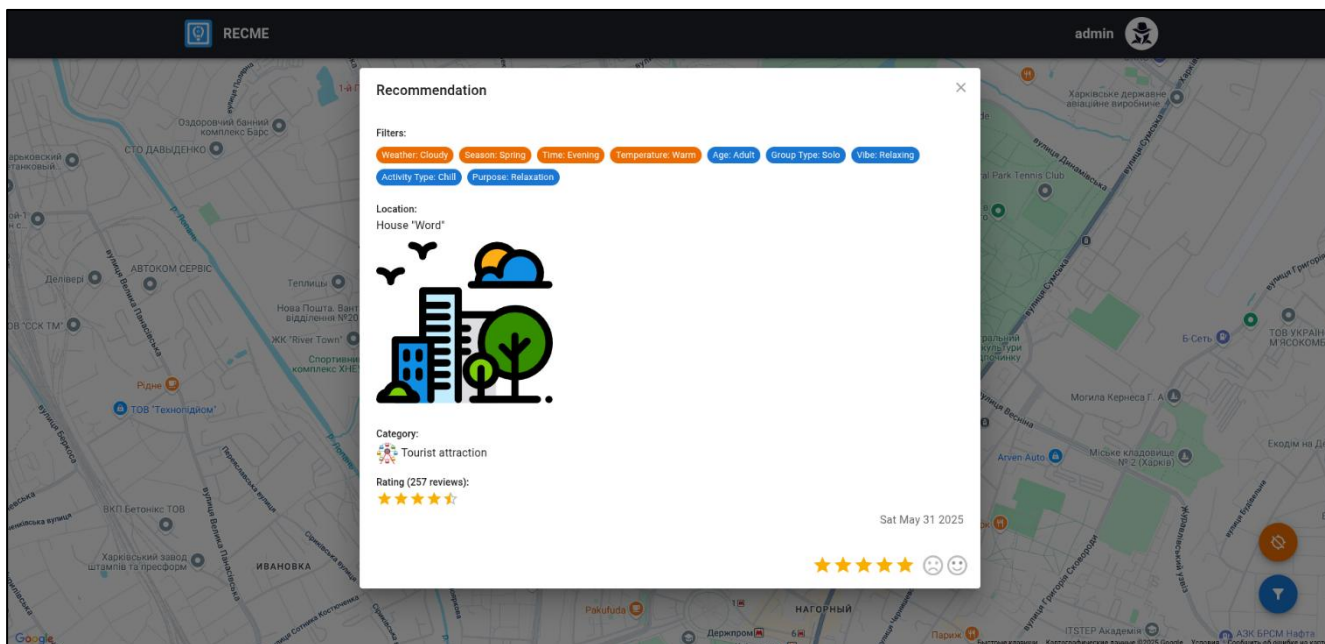


Рисунок 6.14 – Форма оцінки локації (рисунок виконано самостійно)

Також, в модальному вікні користувач має можливість оцінити рекомендацію (див. рис. 6.15), що вплине на майбутні рекомендації з подібними фільтрами.

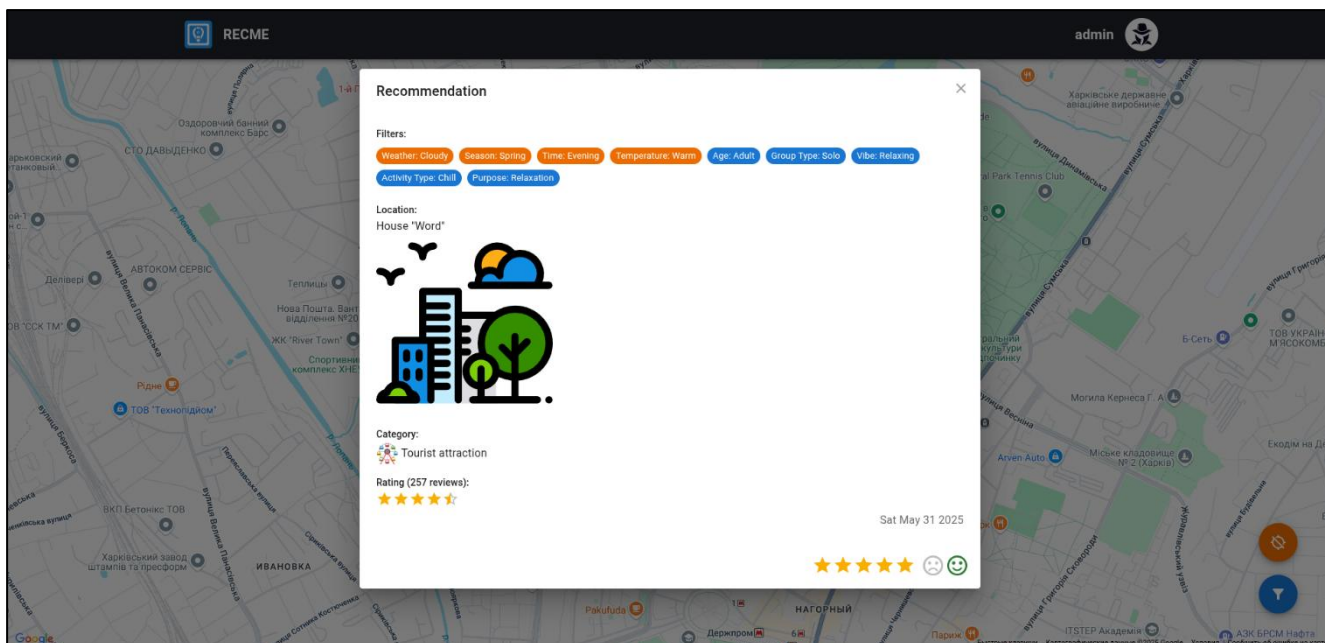


Рисунок 6.15 – Форма оцінки рекомендації (рисунок виконано самостійно)

На сторінці профілю, користувач може змінити своє ім'я, пошту, зображення та пароль. Також, він має можливість продивитися історію своїх рекомендацій (див. рис. 6.16). Присутня можливість сортування та фільтрації.

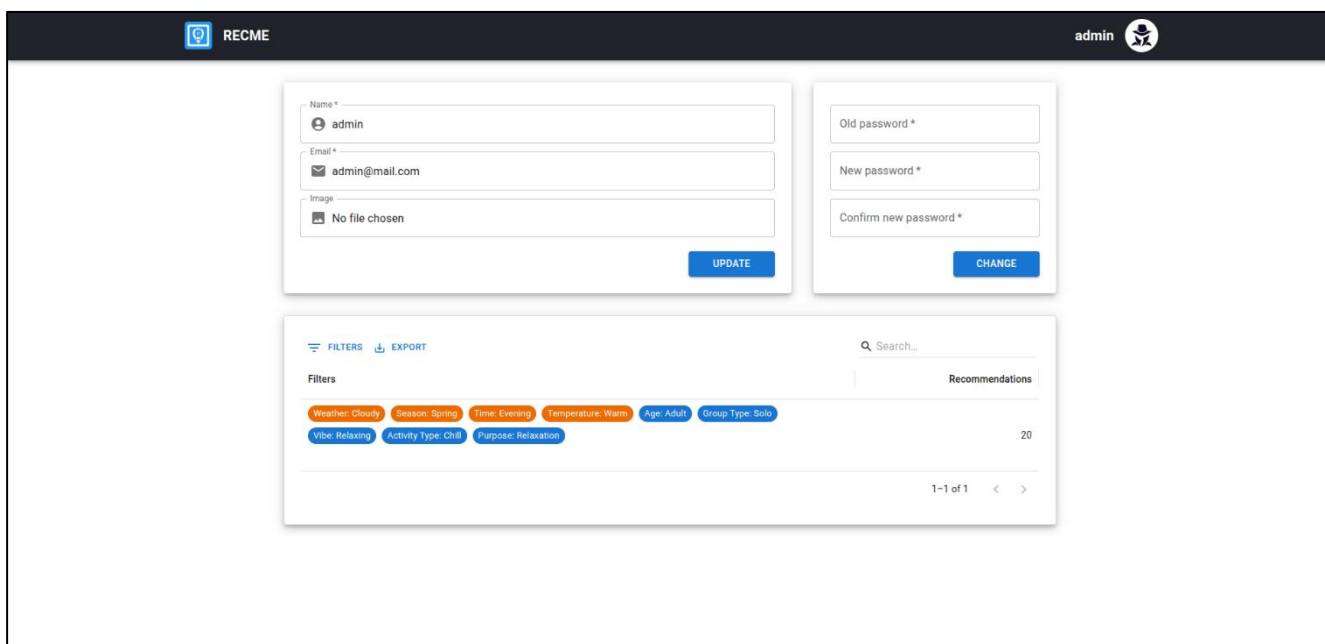


Рисунок 6.16 – Сторінка профілю користувача (рисунок виконано самостійно)

Обравши рекомендації з необхідними фільтрами та контекстом, користувач зможе продивитися усі отримані рекомендації та оцінити їх за необхідністю (див. рис. 6.17).

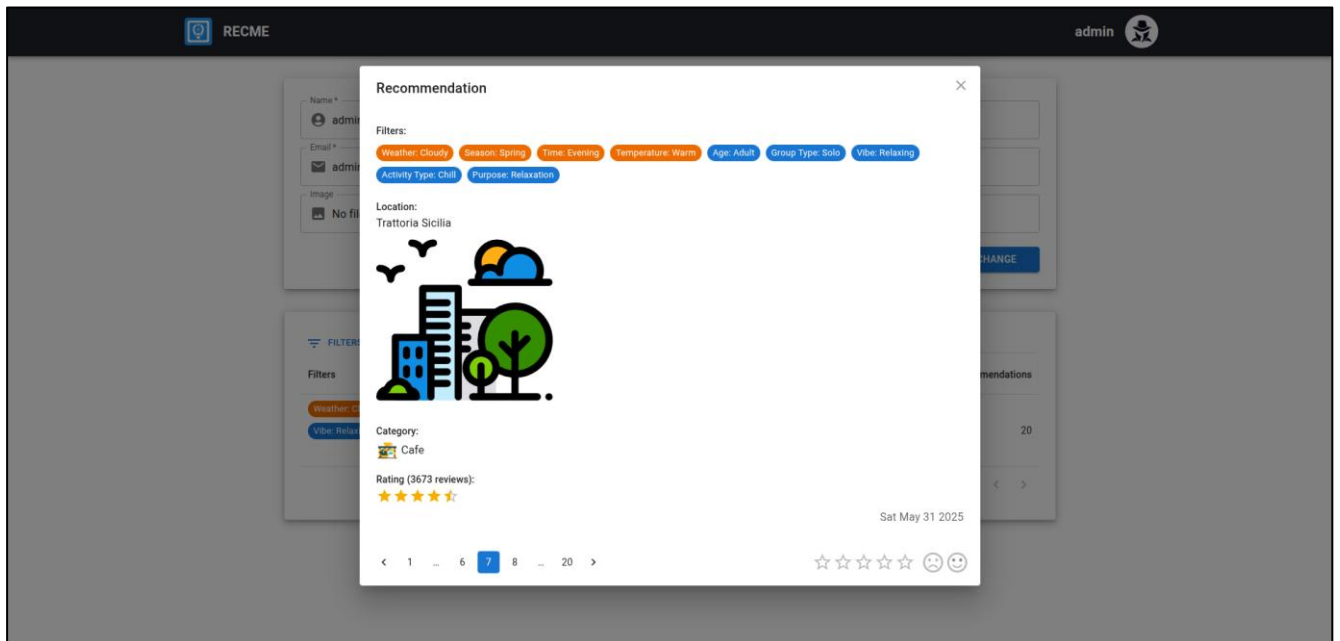


Рисунок 6.17 – Сторінка профілю користувача (рисунок виконано самостійно)

Оскільки передбачається, що найчастіше використання такої системи буде через телефон, інтерфейс всіх представлених функцій адаптований під усі види пристроїв для максимальної зручності використання.

ВИСНОВКИ

У ході дослідження були проведені аналіз предметної галузі, огляд і аналіз літературних та наукових джерел, постановка задачі розробки програмної системи, а також теоретичне обґрунтування і проєктування системи. У процесі роботи, було проведено дослідження специфіки напрямку, включаючи основні виклики та обмеження сучасних рекомендаційних систем. Було також розглянуто існуючі підходи та тенденції розвитку галузі, що дало змогу визначити перспективи подальшого вдосконалення.

На основі проведеного огляду літератури та аналізу наукових джерел сформовано основу для подальшого проєктування системи. Було оцінено актуальність і новизну сучасних підходів, що дозволило обрати найбільш ефективні методи для вирішення поставлених задач. У результаті розроблено концепцію програмної системи, яка враховує сучасні вимоги до рекомендаційних алгоритмів та забезпечує ефективну взаємодію з користувачами.

Під час постановки задачі було сформульовано основну мету роботи та окреслено задачі, які необхідно виконати для її досягнення. Основна мета полягала у розробці програмної системи для надання персоналізованих рекомендацій користувачам на основі їхніх уподобань, даних про поточні умови та взаємодії з платформою. Для реалізації цієї мети визначено такі ключові задачі, як створення ефективної архітектури програмної системи, розробка зручного користувацького інтерфейсу, організація зберігання і обробки даних, а також впровадження сучасних алгоритмів рекомендацій.

У теоретичному розділі описано архітектуру системи, яка базується на клієнт-серверній моделі, що забезпечує високу продуктивність та масштабованість. Розроблено структуру зберігання даних із використанням PostgreSQL, що гарантує надійність і ефективність роботи з великими обсягами інформації. Особливу увагу приділено проєктуванню системи рекомендацій, де реалізовано гібридний підхід, що поєднує контентний та колаборативний фільтри. Цей підхід дозволяє забезпечити точність і персоналізацію рекомендацій для користувачів залежно від їхніх уподобань та даних про поточні умови. Додатково розроблено концепцію

UI/UX дизайну, яка відповідає сучасним вимогам до зручності та естетики інтерфейсу. Дизайн базується на принципах доступності та інтуїтивності, що забезпечує позитивний користувацький досвід.

У розділі практичного дослідження описано розроблену рекомендаційну систему та проаналізовано результати проведених експериментів моделі.

Розділ з описом розробленої програмної системи надає опис розробленого програмного забезпечення з наглядною візуалізацією усіх можливостей системи.

Таким чином, можна дійти висновку, що система відповідає сучасним тенденціям і є перспективною для подальшого вдосконалення, зокрема у напрямках розширення функціоналу, підвищення точності алгоритмів та інтеграції з іншими сервісами. Проведене дослідження робить внесок у розвиток галузі рекомендаційних систем і відкриває нові можливості для реалізації нових систем на основі отриманих результатів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Darapisut, Sumet and Amphawan, Komate and Rimcharoen, Sunisa, "N-Most Interesting Location-based Recommender System," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 16, pp. 84-99, 2022.
2. I. Saifudin and T. Widiyaningtyas, "Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets," in *IEEE Access*, vol. 12, pp. 19827-19847, 2024.
3. Kalra, Mamta and Sangwan, Suman, "Design Engineering A Literature Review on Ontology-based and Deep Learning-based Recommendation System," *Design Engineering*, pp. 13898-13906, 2021.
4. Kirill S., Pribylnov D., Martovytskyi V., Chupryna A., "Investigation of network infrastructure control parameters for effective intellectual analysis", 14th International Conference on Advanced Trends in Radioelectronics, pp. 983-986, 2018.
5. N. M. Rezk, M. Purnaprajna, T. Nordström and Z. Ul-Abdin, "Recurrent Neural Networks: An Embedded Computing Perspective," in *IEEE Access*, vol. 8, pp. 57967-57996, 2020.
6. Arsenov A., Ruban I., Smelyakov K., Chupryna A., "Evolution of convolutional neural network architecture in image classification problems", *CEUR Workshop Proceedings*, pp. 35-45, 2018.
7. K. Yang and J. Zhu, "Next POI Recommendation via Graph Embedding Representation From H-Deepwalk on Hybrid Network," in *IEEE Access*, vol. 7, pp. 171105-171113, 2019.
8. A. Chaudhari, A. A. Hitham Seddig, A. Sarlan and R. Raut, "A Hybrid Recommendation System: A Review," in *IEEE Access*, vol. 12, pp. 157107-157126, 2024.
9. K. A. Fararni, F. Nafis, B. Aghoutane, A. Yahyaouy, J. Riffi and A. Sabri, "Hybrid recommender system for tourism based on big data and AI: A conceptual framework," in *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 47-55, March 2021.

10. Z. Huang, C. Yu, J. Ni, H. Liu, C. Zeng and Y. Tang, "An Efficient Hybrid Recommendation Model With Deep Neural Networks," in *IEEE Access*, vol. 7, pp. 137900-137912, 2019.
11. P. Yochum, L. Chang, T. Gu and M. Zhu, "Linked Open Data in Location-Based Recommendation System on Tourism Domain: A Survey," in *IEEE Access*, vol. 8, pp. 16409-16439, 2020.
12. K. Benabbes, K. Housni, A. El Mezouary and A. Zellou, "Recommendation System Issues, Approaches and Challenges Based on User Reviews," in *Journal of Web Engineering*, vol. 21, no. 4, pp. 1017-1054, 2022.
13. C. Huda, Y. Heryadi, Lukas and W. Budiharto, "Smart Tourism Recommender System Modeling Based on Hybrid Technique and Content Boosted Collaborative Filtering," in *IEEE Access*, vol. 12, pp. 131794-131808, 2024.
14. V. Lapin, K. Smelyakov, A. Chupryna and Z. Dudar, "A Hybrid Approach in Developing a Recommendation System for Personalized Selection of Locations for a Visit," 2025 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, pp. 1-6, 2025.
15. J. McAuley, "Google Local Data (2021)," URL: https://mcauleylab.ucsd.edu/public_datasets/gdrive/googlelocal/, (дата звернення: 01.06.2025).
16. GitHub, "MagisterFelix/Recme," URL: <https://github.com/MagisterFelix/Recme/>, (дата звернення: 03.06.2025).

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ
КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ**

4. Kirill S., Pribylnov D., Martovytskyi V., Chupryna A., "Investigation of network infrastructure control parameters for effective intellectual analysis", 14th International Conference on Advanced Trends in Radioelectronics, pp. 983-986, 2018.
6. Arsenov A., Ruban I., Smelyakov K., Chupryna A., "Evolution of convolutional neural network architecture in image classification problems", CEUR Workshop Proceedings, pp. 35-45, 2018.
14. V. Lapin, K. Smelyakov, A. Chupryna and Z. Dudar, "A Hybrid Approach in Developing a Recommendation System for Personalized Selection of Locations for a Visit," 2025 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, pp. 1-6, 2025.