

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

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Світлана ВАЩЕНКО

\_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня магістр**

зі спеціальності 122 «Комп'ютерні науки» \_\_\_\_\_,

освітньо-наукової програми «Інформаційні технології проектування» \_\_\_\_\_

на тему: Інформаційна технологія надання рекомендацій профілів за спільними інтересами для соціальної мережі \_\_\_\_\_

Здобувачки групи ІТ.м-21н Розгон Маргарита Олегівна \_\_\_\_\_  
(шифр групи) (прізвище, ім'я, по батькові)

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

\_\_\_\_\_  
(підпис)

Маргарита РОЗГОН  
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник к. т. н., доцент Юлія ПАРФЕНЕНКО \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ)

\_\_\_\_\_  
(підпис)

**Суми – 2024**

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

**ЗАТВЕРДЖУЮ**

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«\_\_\_\_\_» \_\_\_\_\_ 2024 р.

## **ЗАВДАННЯ**

**на кваліфікаційну роботу магістра студентіві**

Розгон Маргарита Олегівна

(прізвище, ім'я, по батькові)

**1 Тема кваліфікаційної роботи** Інформаційна технологія надання рекомендацій профілів за спільними інтересами для соціальної мережі  
затверджена наказом по університету від «01» лютого 2024 р. № 0096-VI

**2 Термін здачі студентом кваліфікаційної роботи** «10\_» \_\_\_\_\_ травня \_\_\_\_\_ 2024 р.

**3 Вхідні дані до кваліфікаційної роботи** Літературні джерела з питань побудови рекомендаційних систем, коефіцієнти подібності

**4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)** Аналіз предметної області; Постановка задачі та аналіз методів дослідження; Розробка інформаційної технології надання рекомендацій профілів за спільними інтересами для соціальної мережі.

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації)** Обґрунтування актуальності, Постановка задачі, Функціональні вимоги, Структурно-функціональне моделювання, Моделювання варіантів використання, Моделювання бази даних, Тестування, Висновки, Апробація результатів роботи

## 6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання \_\_\_\_\_.

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

### КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	1.11.23-29.12.23	
2	Аналіз аналогів	10.12.23-29.12.23	
3	Визначення мети системи	22.12.23-29.21.23	
4	Визначення вимог	03.01.24-10.01.24	
5	Вибір методів та інструментів реалізації	11.01.24-24.01.24	
6	Реалізація рекомендаційної системи	01.02.24-31.03.24	
7	Впровадження рекомендаційної системи у соціальну мережу	29.02.24-24.03.24	
8	Тестування	25.03.24-30.04.24	
9	Оформлення документації	15.04.24-10.05.24	

Магістрант \_\_\_\_\_

Маргарита РОЗГОН

Керівник роботи \_\_\_\_\_

к.т.н., доц. Юлія ПАРФЕНЕНКО

## АНОТАЦІЯ

Тема кваліфікаційної роботи магістра «Інформаційна технологія надання рекомендацій профілів за спільними інтересами для соціальної мережі».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 43 найменувань, додатків. Загальний обсяг роботи – 82 сторінки, у тому числі 55 сторінок основного тексту, 6 сторінок списку використаних джерел, 15 сторінок додатків.

Перший розділ присвячено аналізу предметної області. Розглянуто поняття рекомендаційна система, наведено їх основні типи та проблеми. Наведено аналіз систем рекомендації профілів у популярних соцмережах.

У другому розділі представлено постановку задачі та аналіз методів її вирішення. Задача полягає у реалізації рекомендаційної системи на основі розрахунку коефіцієнтів подібності та гібридної фільтрації, яка включає контентну, колаборативну та топологічну фільтрації. Далі наведено декілька різних коефіцієнтів подібності та способи їх застосування у рекомендаційних системах. У результаті аналізу обирається коефіцієнт Жаккара для реалізації кожного типу фільтрації.

Третій розділ містить результати проектування рекомендаційної системи у вигляді діаграм IDEF0 та варіантів використання. Також наведено частину ERD бази даних соцмережі, з таблицями, необхідними для роботи рекомендаційної системи.

У четвертому розділі представлено структуру реалізованої рекомендаційної системи. Наведено опис заздалегідь розробленої соціальної мережі, а також приклад використання рекомендаційної системи у даній соцмережі.

Результатом роботи є рекомендаційна система, розроблена у вигляді програмного модуля мовою JavaScript, інтегрована у заздалегідь розроблену соціальну мережу.

Ключові слова: інформаційна технологія, рекомендаційна система, рекомендації профілів, рекомендаційні алгоритми, соціальна мережа, коефіцієнт Жаккара.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	6
ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	9
1.1 Поняття соціальної мережі.....	9
1.2 Рекомендаційні системи .....	11
1.3 Аналіз методів та підходів до рекомендації профілів у популярних соцмережах .....	18
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	24
2.1 Постановка задачі.....	24
2.2 Методи рекомендації профілів.....	26
2.3 Коефіцієнти подібності що використовуються для рекомендацій профілів .....	31
3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ.....	38
3.1 Структурно-функціональне моделювання системи рекомендації профілів.....	38
3.2 Моделювання варіантів використання системи рекомендації профілів .....	41
3.3 Проектування моделі послідовності для соціальної мережі та рекомендаційної системи .....	42
3.4 Проектування моделі бази даних соціальної мережі.....	47
4 ПРОГРАМНА РЕАЛІЗАЦІЯ .....	50
4.1 Прототип соціальної мережі .....	50
4.2 Реалізація рекомендаційної системи .....	54
4.3 Впровадження рекомендаційної системи у прототип соціальної мережі .....	55
4.4 Тестування рекомендаційної системи .....	58
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	62
ДОДАТОК А.....	68
ДОДАТОК Б .....	75
ДОДАТОК В.....	81

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

РС – Рекомендаційна система

БД – База даних

IDEF0 – Методологія моделювання функцій (Integrated Definition For Function Modeling)

ERD – Модель «зв'язок-сутність» (Entity-Relationship Diagram)

ANN – Штучна нейронна мережа (Artificial Neural Network)

GNN – Графова нейронна мережа (Graph Neural Network)

GCN – Графова згорткова мережа (Graph Convolutional Network)

MViT – Багатомасштабний перетворювач візуальних даних (Multiscale Vision Transformer)

XLM-R – Міжмовна модель мови для пошуку (Cross-lingual Language Model for Retrieval)

XLM-V – Міжмовна модель мови для розуміння (Cross-lingual Language Model for Understanding)

FLAVA – Групування аудіо та візуальних даних на функціональному рівні (Feature-Level Audio-Visual Alignment)

NSFW – Not Safe for Work

API – Інтерфейс прикладного програмування (Application Programming Interface)

СУБД – Система управління базою даних

## ВСТУП

**Актуальність.** Важко уявити сучасне життя без соціальних мереж. Вони використовуються у найрізноманітніших сферах життя. Хтось їх використовує просто для розваг, хтось для просування та ведення бізнесу, хтось для встановлення професійних контактів або для навчання, а хтось для волонтерства чи благодійної діяльності. Під час війни соціальні мережі стали однією з основних платформ розповсюдження інформації про різноманітні збори на допомогу армії та людям, постраждалим від війни. Рекомендації профілів допомагають однодумцям знаходити один одного та об'єднуватись проти спільного ворога.

У роботах авторів N. B. Silva [1], та S. V. Reddy [2] представлено алгоритми, які базуються на різних критеріях відбору профілів у рекомендовану вибірку. У роботі автора S. Cheng [3] наведено алгоритм, який одночасно враховує багато різних критеріїв, але чим більше критеріїв враховується, тим більш складний алгоритм для реалізації, та тим більше він вимагає ресурсів. У роботах авторів M. Li [4] та X. Song [5] представлено алгоритми основані на моделі машинного навчання. Однак такі алгоритми мають високу складність реалізації. До того ж модель необхідно тренувати на реальних даних. Крім того модель машинного навчання знаходить приховані шаблони та закономірності, в результаті чого складно зрозуміти чи пояснити, на яких підставах рекомендаційна система рекомендує той чи інший профіль. Тож роботу присвячено розробці інформаційної технології рекомендації профілів на основі евристичних методів, яка б не була вимогливою до обчислювальних ресурсів та надавала якісні рекомендації.

**Об'єкт дослідження.** Процес рекомендацій профілів за спільними інтересами для соціальної мережі.

**Предмет дослідження.** Інформаційна технологія рекомендації профілів користувачів на основі аналізу даних користувача.

**Наукова новизна.** На відміну від існуючих моделей, при виборі оптимального алгоритму було враховано не тільки точність рекомендацій, а і складність та

ресурозатратність алгоритму. Розроблена інформаційна технологія рекомендації профілів використовує поєднання трьох фільтрацій: контентна, колаборативна, топологічна. Кожна з фільтрацій застосовує розрахунок коефіцієнта подібності Жаккара. Для поєднання різних фільтрацій використано стратегію гібридизації основу на зваженій сумі, де вага кожної фільтрації дорівнює одиниці.

**Мета** – розробити інформаційну технологію надання рекомендацій профілів за спільними інтересами для соціальної мережі.

**Задачі дослідження:**

- проаналізувати існуючі методи та рекомендаційні алгоритми, наведені у науковій літературі та виконати порівняльний аналіз;
- змінити соціальну мережу, реалізувати збір даних, необхідних для роботи рекомендаційного алгоритму;
- реалізувати різні методи надання рекомендацій для соціальної мережі та обрати найбільш оптимальний, враховуючи складність реалізації та вимоги до ресурсів;
- розробити інформаційну технологію надання рекомендацій профілів користувачів;
- реалізувати рекомендаційну підсистему на основі розробленої інформаційної технології та інтегрувати її в соціальну мережу;
- провести аналіз достовірності надання рекомендацій із застосуванням розробленої інформаційної технології.

**Практичне значення.** Інформаційна технологія надання рекомендацій профілів є важливою складовою соціальної мережі так, як вона вносить свої корективи у еволюцію мережевих з'єднань. А також вона може вплинути на життя людей, адже вона може порекомендувати знайомство людям зі спільними інтересами, які могли б ніколи не познайомитись у реальному житті.

**Апробація результатів.** По результатам роботи було опубліковано тези та здійснено виступ із доповіддю на конференції «Інформатика, Математика, Автоматика 2024». Копію публікації тез наведено у додатку В.



# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Поняття соціальної мережі

Соціальна мережа – це вебдодаток, у якому люди можуть спілкуватись та ділитись інформацією. Основні можливості соціальних мереж: встановлювати зв'язок з іншими користувачами (дати у друзі чи контакти), створювати дописи, додавати реакції на дописи, коментувати дописи, надсилати повідомлення іншим користувачам.

За останні десятиліття кількість користувачів соціальних мереж значно зростає. Соціальні мережі заповнюють повсякденне життя людей, ставши найголовнішою платформою для нових знайомств та розповсюдження різноманітної інформації. Соцмережі дозволяють ділитись думками, інтересами та творчістю, взаємодіяти з іншими людьми [6, 7]. Серед інформації яка шириться соцмережами можна виділити наступні категорії:

- Особисті думки та світлини;
- Результати творчої діяльності;
- Новини;
- Оголошення;
- Реклама;
- Особисті досягнення;
- Благодійні збори;
- Контент для навчання.

Існують різні типи соціальних мереж. Одні, як наприклад Facebook чи Instagram, спрямовані на розповсюдження різноманітного контенту. Інші використовуються для поширення контенту на конкретну тематику. Наприклад, LinkedIn та Xing – це соцмережі, спрямовані встановлення бізнес зв'язків, де користувачі діляться своїми професійними досягненнями. ResearchGate – соціальна мережа призначена для взаємодії між науковцями та розповсюдження наукового контенту. Flixster та Youtube дозволяють розповсюджувати фільмів та відео. Telegram та Facebook Messaging

створені для обміну повідомленнями. На рисунку 1.1 зображено розвиток соціальних мереж за їхніми категоріями.

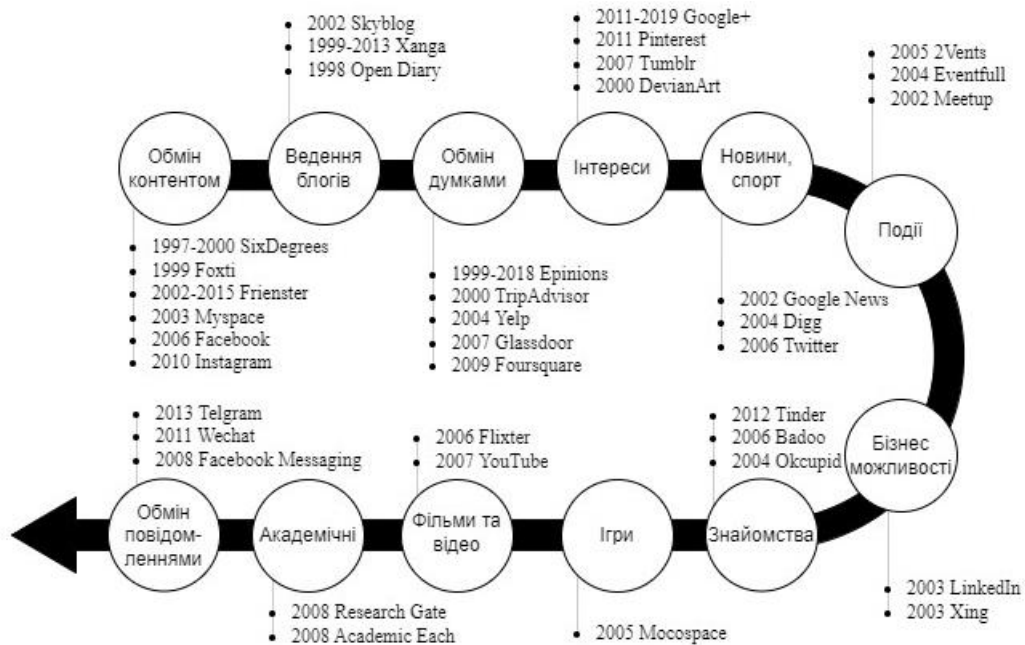


Рисунок 1.1 – розвиток соціальних мереж

*Джерело: [7]*

Коли соцмережі тільки з'явилися, їх користувачів було небагато. Щоб знайти підходящу для знайомства людину, можна було переглянути та проаналізувати профілі інших користувачів. Але, зі збільшенням користувачів соцмереж, цей підхід став майже неможливим. Тепер, щоб знайти підходящу людину таким способом, потрібно переглянути дуже багато профілів, що є досить виснажливим процесом для людини. Це називають проблемою перевантаження інформацією. Серед великої кількості інформації складно знайти те, що потрібно. Ця проблема характерна не тільки для соціальних мереж, а і для інтернету в цілому. Існують два основні підходи до вирішення проблеми перевантаження інформацією: пошукові та рекомендаційні системи. Пошукові системи надають однаковий результат для всіх користувачів. Рекомендаційні системи аналізують дані користувача, щоб надати йому персоналізовані результати [6, 8, 9].

## 1.2 Рекомендаційні системи

Рекомендаційна система (РС) – це система, яка фільтрує інформацію, вибираючи необхідний фрагмент з великої кількості інформації. Критеріями фільтрації є уподобання користувача, його дані або історія його дій. РС допомагає користувачам подолати проблему перевантаження інформацією, надаючи їм персоналізовані та ексклюзивні рекомендації [4]. РС застосовуються у різних сферах: для рекомендацій товарів, послуг, музики, відео, дописів та друзів у соцмережах, та іншого контенту.

Серед систем рекомендації профілів, можна виділити наступні типи фільтрації:

- Контентна фільтрація (або фільтрація на основі контенту, фільтрація на основі вмісту);
- Колаборативна фільтрація (або спільна фільтрація);
- Фільтрація на основі топології мережі;
- Гібридна фільтрація.

### Контентна фільтрація

При рекомендації предметів за контентною фільтрацією користувачу рекомендуються ті предмети, які мають певну схожість із предметами, які користувач уподобав у минулому [4]. Для рекомендації друзів контентна фільтрація полягає у пошуку користувачів зі схожими профільними даними. Тобто, якщо користувачі живуть в одному місті, навчались в одному університеті, належать до однієї вікової категорії, то імовірно вони стануть друзями. До переваг цієї фільтрації відноситься відсутність проблеми холодного старту та проблеми розрідженості даних. Але даний тип фільтрації має недоліки: залежність від контенту, відсутність використання оцінки користувачів [10]. На рисунку 1.2 наведено класифікацію методів контентної фільтрації.



Рисунок 1.2 – класифікація методів контентної фільтрації  
*Джерело:* побудовано автором на основі даних зі статей [11-13]

### Колаборативна фільтрація

При рекомендації предметів за колаборативною фільтрацією спочатку користувачів об'єднують у групи на основі їхніх спільних уподобань. Потім користувачу рекомендують предмети, які отримали позитивні оцінки від інших користувачів з його групи [4]. Для рекомендації профілів використовують підходи колаборативної фільтрації для пошуку схожих користувачів за їх оцінками. Тобто, якщо користувачі соцмережі додають уподобайки на одні і ті ж самі дописи, то імовірно вони стануть друзями. Цей тип фільтрації має наступні переваги: незалежність рекомендацій від контенту сервісу, використання оцінок користувачів. До недоліків відноситься проблема холодного старту та проблема розрідженості матриці оцінок [10]. На рисунку 1.3 зображено класифікацію методів колаборативної фільтрації.

Методи, засновані на пам'яті, [4, 11] також називають методами на основі сусідства [13] або евристичними [14] методами. Вони полягають у розрахунку подібності між користувачами або предметами, та у виборі найбільш подібних користувачів або предметів. Їх недолік полягає у необхідності перераховувати коефіцієнти подоби кожного разу, що при великому розмірі мережі спричинить надто велику вимогливість до обчислювальних ресурсів.



Рисунок 1.3 – класифікація методів колаборативної фільтрації  
*Джерело:* побудовано автором на основі даних зі статей [4, 11-14]

Методи, засновані на пам'яті та методи кластеризації застосовують коефіцієнти подоби для визначення подібних об'єктів. Прикладами коефіцієнтів подоби є Евклідова відстань, Манхетенська відстань, коефіцієнт Жаккара, кореляція Пірсона, косинусна подібність [4, 12, 13].

### Фільтрація на основі топології мережі

Соціальну мережу можна представити як граф, у якому користувачі є вузлами, а дружба або інші взаємозв'язки між користувачами є ребрами. Тоді рекомендацію дружби можна розглядати як передбачення появи зв'язків між вузлами у складній мережі. Для вирішення цієї задачі існує ряд алгоритмів теорії складних мереж з різними підходами [1, 3, 6, 15-16]. На рисунку 1.4 наведено класифікацію методів основаних на топології мережі.

Методи, засновані на сусідстві, полягають у розрахунку коефіцієнта подоби між двома вузлами на основі списків їх сусідів. Чим більше спільних сусідів мають вузли, тим більшим буде коефіцієнт подоби. Тобто, чим більше спільних друзів мають користувачі тим імовірніше, що вони стануть друзями.

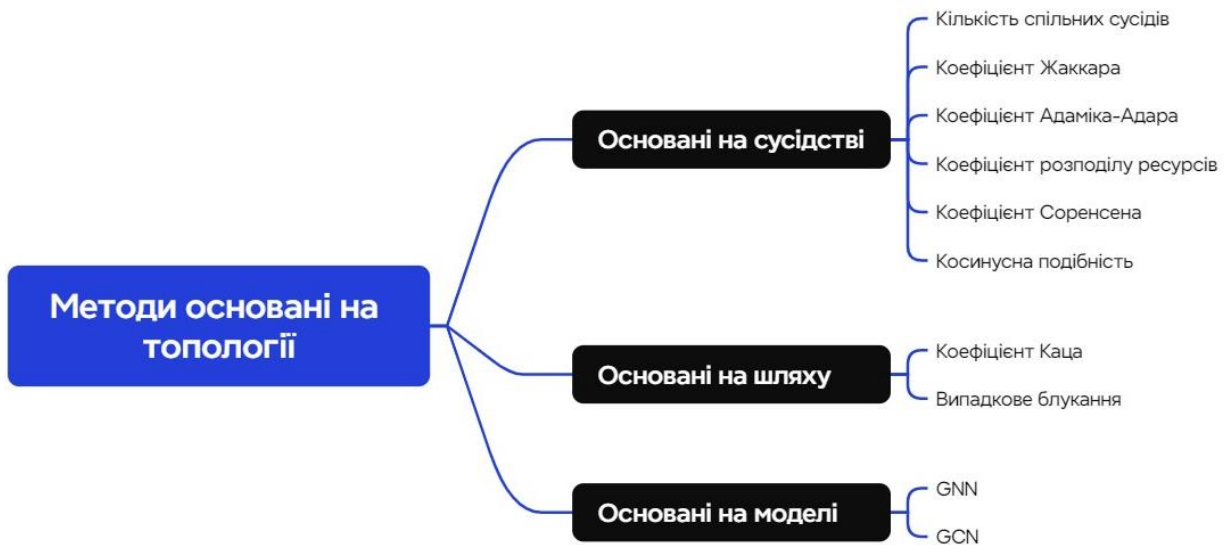


Рисунок 1.4 – методи основані на топології мережі  
*Джерело:* побудовано автором на основі даних зі статей [1, 15-18]

### Гібридна фільтрація

Гібридна фільтрація полягає у застосуванні декількох типів фільтрації одночасно. Це дозволяє надавати більш точні та ефективні рекомендації, адже використання комбінації алгоритмів допомагає перекрити слабкі сторони окремого алгоритму, що є перевагою цієї фільтрації [4]. До її недоліків можна віднести складність розробки та підтримки [10].

На рисунку 1.5 наведено різні підходи до гібридної фільтрації.

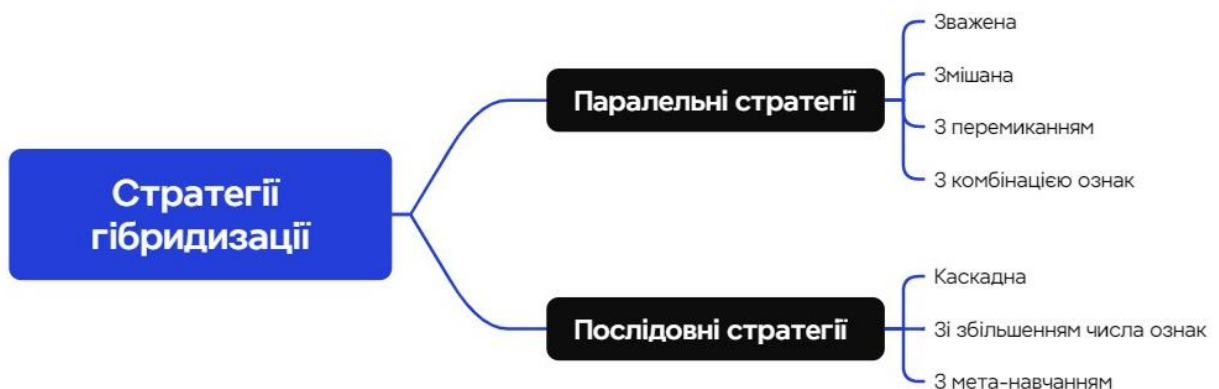


Рисунок 1.5 – стратегії гібридизації методів рекомендаційних систем  
*Джерело:* побудовано автором на основі даних зі статей [11, 13]

1. Зважена. Результати декількох фільтрацій об'єднуються за допомогою зваженої суми.
2. Змішана. Об'єднання декількох вибірок з результатами окремої роботи кількох фільтрацій.
3. З перемиканням. Відбувається вибір фільтрації на основі критерію для перемикання.
4. З комбінацією ознак. Коефіцієнти, отримані за допомогою однієї фільтрації, передаються як додаткові ознаки у іншу фільтрацію.
5. Каскадна. Результат роботи однієї фільтрації, уточнюється за допомогою іншої фільтрації.
6. Зі збільшенням числа ознак. Результати роботи кількох фільтрацій, передаються як ознаки у іншу фільтрацію.
7. З мета-навчанням. Побудова моделі машинного навчання, яка навчається на результатах роботи кількох фільтрацій [11, 13].

### Проблеми рекомендаційних систем

У рекомендаційних системах існують наступні проблеми: проблема холодного старту, проблема постійного холодного старту, проблема бульбашки фільтрів, проблема розрідженості даних, проблема білих ворон (або чорних овець) [19].

Проблема холодного старту виникає при появі нового об'єкту. Наприклад, у системі зареєструвався новий користувач, історія дій якого порожня. Тобто РС не має необхідних даних про користувача, на основі яких можна сформулювати рекомендації. Для вирішення цієї проблеми застосовують гібридну фільтрацію.

Проблема постійного холодного старту виникає, коли об'єкт змінює свої властивості. Наприклад, користувач з часом змінив свої інтереси. Не всі РС здатні адаптуватись до цього. Подолати цю проблему допомагає застосування машинного навчання.

Проблема бульбашки фільтрів полягає у тому, що РС надає користувачу інформацію на основі його уподобань, ізолюючи його від усієї іншої інформації. Тобто

РС надає користувачу лише обмежену кількість однотипних об'єктів. Для подолання цієї проблеми застосовують додаткові вимоги до формування списку рекомендацій.

Прикладом таких вимог є забезпечення наступних властивостей [19]:

- Різноманітність. Список рекомендацій повинен містити різнотипні об'єкти.
- Неочікуваність. Список рекомендацій повинен містити хоча б один об'єкт, не схожий на об'єкти з історії користувача.
- Новизна. Список рекомендацій повинен містити не тільки популярні об'єкти.

При застосуванні колаборативної фільтрації формується матриця оцінок, на основі якої формується список рекомендацій. Так як користувачі оцінюють лише незначну кількість елементів, то матриця оцінок має багато пропусків, що затрудняє формування рекомендацій. Це називають проблемою розрідженості даних. Для вирішення цієї проблеми використовують алгоритми матричної факторизації [10].

Проблема білих ворон полягає у тому, що існують користувачі, уподобання яких відрізняється від уподобань більшості. Рекомендаційні системи не здатні надати корисні рекомендації для таких користувачів.

Отже, для подолання проблем РС застосовують: гібридні алгоритми, додаткові вимоги до формування рекомендацій, алгоритми машинного навчання, алгоритми матричної факторизації.

### Використання машинного навчання в РС

Кожен тип фільтрації містить методи, засновані на моделі. Вони полягають у створенні та навчанні моделі засобами машинного навчання. Серед сучасних методів рекомендацій профілів є популярними методи, засновані на глибокому навчанні [20]. Глибоке навчання базується на моделі нейронних мереж. Глибоке навчання розглядається як клас техніки машинного навчання, який використовується для навчання представленням шляхом використання багатьох рівнів етапів обробки інформації в ієрархічних архітектурах [21]. У рекомендаційних системах часто



застосовується модель GNN (graph neural network) [20, 22-24], яка розширює нейронні мережі для обробки графових даних [20].

Методи машинного навчання мають свої переваги та недоліки. Вони стійкі до багатьох проблем. Аналізуючи дані, вони можуть знайти закономірності, які людині складно знайти чи зрозуміти. З одного боку це є перевагою, але з іншого боку складно зрозуміти чи пояснити на яких підставах РС рекомендує той чи інший предмет або профіль. Інші недоліки полягають у високій складності реалізації та необхідності тренувати модель на реальних даних [24].

#### Огляд моделей рекомендаційних систем для соціальних мереж

У роботі [3] наведено мультиджерельний алгоритм, який робить ряд вибірок, враховуючи різні критерії та робить їх злиття в фінальну вибірку, за допомогою теорії доказів.

У роботі [20] запропоновано алгоритм, який базується на моделі під назвою згорткова мережа з мультисоціальним графом (multi-social graph convolutional network). Даний алгоритм для рекомендації друзів враховує дані та соціальні зв'язки користувача з кількох соціальних мереж, що допомагає зменшити розрідженість даних та підвищити точність рекомендацій.

У роботі [2] представлено алгоритм, для виявлення спільнот (тобто груп користувачів з спільними інтересами). Наведений алгоритм оснований на гібридній фільтрації з використанням сингулярного розкладання.

У роботі [1] наведено систему рекомендацій друзів, яка базується на структурних властивостях соціальних мереж та генетичному алгоритмі. Дана система заснована на ідеї: «більш ймовірно, що ви знаєте друга свого друга, ніж будь-яку іншу випадкову людину». Тож дана система включає у рекомендаційну вибірку профілі людей, з якими у користувача є хоча б один спільний друг. Врахування структури соціальної мережі, покращує якість рекомендацій, але якщо використовувати її разом з іншими критеріями, як профільні дані чи історія дій користувача. Якщо під час рекомендацій враховується лише структура соціальної мережі, то рекомендації не будуть достатньо якісними [3].

Отже, було розглянуто різні рекомендаційні системи, засновані на машинному навчанні, матричній факторизації чи структурі соціальної мережі. Більшість з них мають високу складність реалізації. РС, наведена у роботі [3], враховує багато критеріїв відбору профілів, що ускладнює її реалізацію та підвищує вимоги до ресурсів. РС, представлена у роботі [20], має ще вищу складність та вимоги до ресурсів, адже потрібно отримати та проаналізувати дані з кількох соціальних мереж. У роботі [1] враховується лише структура мережі. Тож одні рекомендаційні системи враховують багато різних критеріїв, інші враховують лише один критерій.

### **1.3 Аналіз методів та підходів до рекомендації профілів у популярних соцмережах**

Компанія Meta є власником одразу кількох популярних соціальних мереж, серед яких Facebook та Instagram. Тому обидві соцмережі використовують у рекомендаційних системах моделі, основані на PyTorch. Крім того Meta займається дослідженням технологій MViT, XLM-R/XLM-V і FLAVA/Omnivore, щоб враховувати при рекомендаціях, семантичне значення різного контенту (включаючи текст, фото, відео та аудіо). Щоб розв'язати проблему холодного старту, застосовується система навчання за кількома випадками під назвою Meta Interest Learner, яка здійснює підбір нового контенту для потенційної аудиторії на основі її інтересів, навіть якщо залучень дуже мало. Щоб розпізнати зв'язок між різними інтересами, використовуються найсучасніші методи навчання за допомогою вбудовування та навчання на графіках, а також використовують моделювання невизначеності в поєднанні з навчанням з підкріпленням [25].

Facebook є найпопулярнішою соціальною мережею. Ця соцмережа налічує більше 3 мільярдів користувачів. У Facebook є блок з назвою «Люди, яких ви можете знати», де розміщується список рекомендованих профілів. Для вибору рекомендованих профілів застосовується колаборативна фільтрація. Крім того

Facebook використовує Apache Giraph, щоб аналізувати соціальний граф, який складається з користувачів та зв'язків між ними [26]. Рекомендовані профілі відбираються за наступними критеріями: наявність спільних друзів; схожі профільні дані (спільне місто проживання, школа чи робота); спільні групи, фото, публікації; список контактів з телефону [27].

Робота рекомендаційної системи включає наступні кроки [28]:

1. Збір профілів за вказаними вище, критеріями.
2. За допомогою моделі машинного навчання будується передбачення, з якими із відібраних профілів, користувач імовірніше буде взаємодіяти. Відбираються ті профілі, з якими імовірність взаємодії найвища.
3. До обраних профілів застосовується фільтр, щоб прибрати профілі, які порушують стандарти спільноти Facebook.

Передбачення моделі будуються на основі вхідних сигналів. При цьому модель застосовує різні вхідні сигнали в залежності від того, як часто користувач використовує Facebook та як багато у нього друзів [28].

На рисунку 1.6 наведено блок з рекомендованими профілями у Facebook.

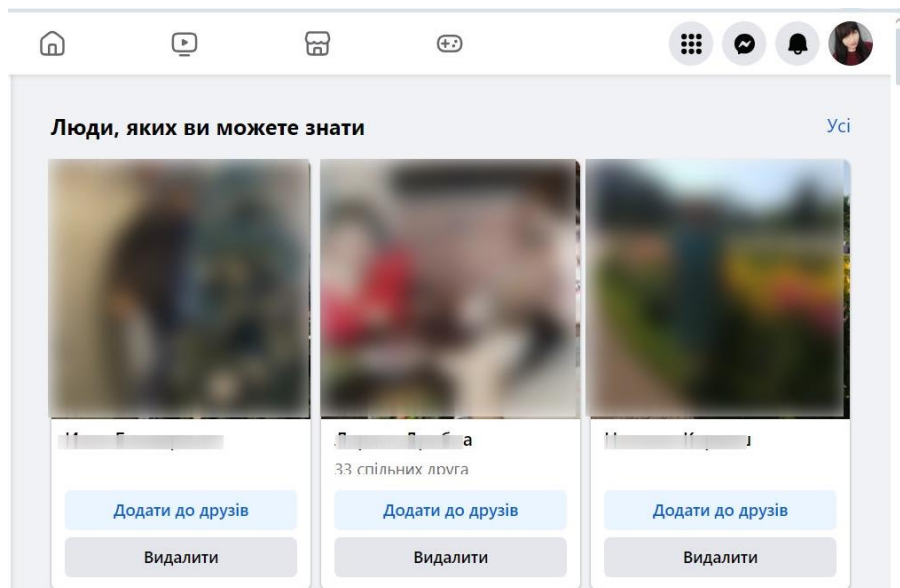


Рисунок 1.6 – блок «Люди, яких ви можете знати» у Facebook

*Джерело: побудовано автором (знімок з екрану)*

На рисунку 1.7 наведено блок зі списком рекомендованих профілів у Instagram, який відображається у профілі іншого користувача.

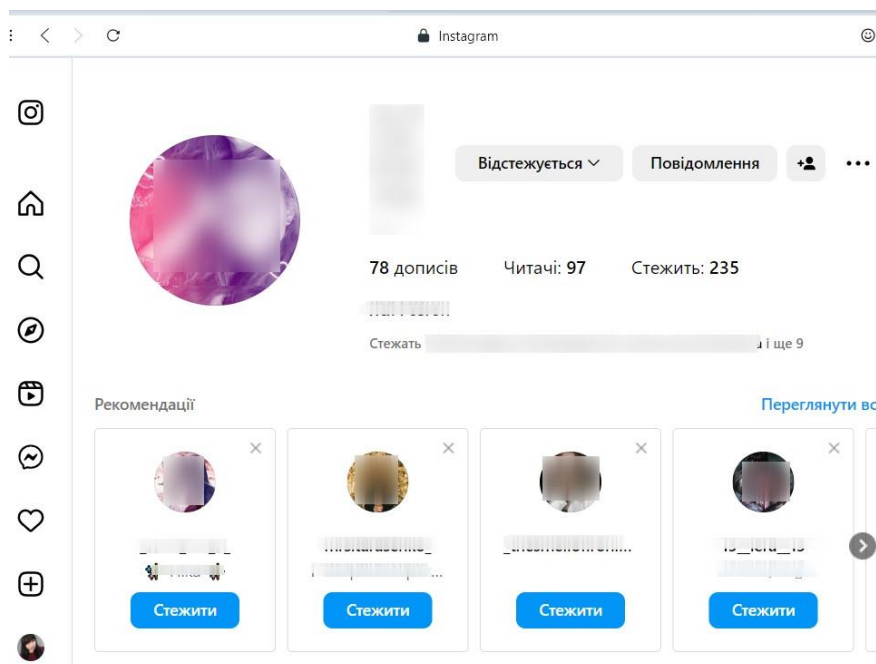


Рисунок 1.7 – блок зі списком рекомендованих профілів у Instagram  
*Джерело: побудовано автором (знімок з екрану)*

В Instagram є дві системи рекомендації профілів. Вони використовують різні моделі машинного навчання.

Перша РС: відображає список рекомендованих профілів у стрічці новин; враховує наступні фактори: подібність профілів, чи користувач є новим; сигнали, на основі яких модель приймає рішення, відрізняються в залежності від розташування користувача, та від того чи користувач є новим.

Друга РС: відображає список рекомендованих профілів, якщо користувач знаходиться на своєму або чужому профілі та натискає кнопку для показу рекомендацій або на кнопку стежити; враховує дані поточного профілю на якому знаходиться користувач; сигнали, на основі яких модель приймає рішення, відрізняються в залежності від кількості підписників профілю, який переглядає користувач та від того чи користувач є новим.

Робота цих рекомендаційних систем включає наступні кроки [29]:

1. Збір профілів потенційних кандидатів;

2. Розрахунок відстані між профілями поточного користувача та профілями потенційних кандидатів.
3. Фільтрація, яка видаляє заблоковані, неактивні профілі та профілі які порушують правила спільноти.
4. Система створює рейтинг профілів, з якими користувач імовірно буде взаємодіяти.

У 2023 році Twitter(X) опублікував вихідний код рекомендаційної системи. РС містить Follow Recommendations Service (FRS), який використовується для рекомендацій профілів. Для створення рейтингу кандидатів FRS використовує як машинне навчання, так і евристичні правила. Щоб передбачити ймовірність взаємодії між користувачами, РС використовує модель машинного навчання, засновану на класифікаторах градієнтного спуску. Більша частина рекомендаційної системи розроблена мовою Scala, також використано мови Java, Starlark, Python, Thrift, C++ та інші [30].

Система рекомендації профілів у Twitter робить рекомендації на основі враховуючи наступні пункти: список контактів з телефону (якщо користувач дав дозвіл на їх завантаження); розташування користувача; дії користувача (уподобайки, репости, підписки, дописи та профілі, які переглядає користувач) [31];

Робота системи складається з наступних кроків [31]:

1. Збір профілів на основі, вказаних вище, критеріїв;
2. Зважене зменшення вибірки;
3. Фільтрація дублікатів;
4. Складання рейтингу, відібраних профілів, на основі машинного навчання;
5. Фільтрація профілів, які користувач заблокував чи за якими перестав слідкувати а також профілів з високим рівнем NSFW.

Система рекомендації профілів у TikTok рекомендує профілі користувачів, які: є у списку контактів з телефону користувача; мають спільні підписки або спільних підписників з користувачем; дружать з користувачем на Facebook; є підписниками користувача; мають контактні дані користувача у списку контактів на телефоні [32].

На рисунку 1.8 зображено блок з рекомендованими акаунтами у TikTok.

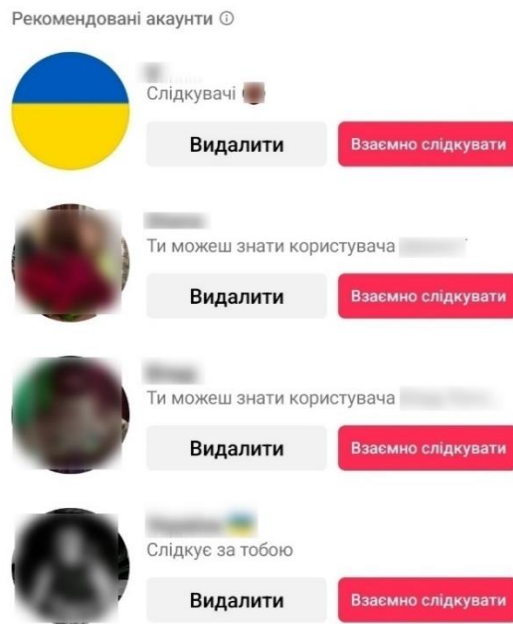


Рисунок 1.8 – Блок з рекомендованими акаунтами в TikTok

*Джерело: побудовано автором (знімок з екрану)*

У таблиці 1.1 наведено порівняння розглянутих систем рекомендації профілів та РС власної розробки.

Таблиця 1.1 – порівняння систем рекомендації профілів

Параметри	Facebook	Instagram	Twitter	TikTok	РС власної розробки
Надання рекомендованих профілів	+	+	+	+	+
Надання повідомлення про причину рекомендації	-	-	-	+	+
Відкритий вихідний код	-	-	+	-	+
Застосування гібридної фільтрації	+	+	+	+	+
Нескладна реалізація	-	-	-	-	+
Відсутність потреби в реальних даних для тренування системи	-	-	-	-	+

*Джерело: побудовано автором*

Отже, у даному підрозділі було розглянуто системи рекомендації профілів, які застосовуються у популярних соцмережах. Над розробленням розглянутих рекомендаційних систем працюють команди розробників, які постійно удосконалюють їхні алгоритми. Складність реалізації цих РС дуже висока. Вихідний код, більшості розглянутих РС, закритий. Робота розглянутих РС заснована на моделях машинного навчання, які потрібно натренувати на великій кількості реальних даних.

## 2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

### 2.1 Постановка задачі

Соціальну мережу можна розглядати як граф, який має кілька типів вузлів та багато зв'язків між ними (формула 2.1).

$$G = (V, E), \quad (2.1)$$

де  $V$  – множина вузлів усіх типів,

$E$  – ребра, які показують зв'язки між вузлами.

$V$  містить кілька типів вузлів (формула 2.2) включаючи користувачів, дописи чи коментарі.  $E$  містить кілька типів зв'язку (формула 2.3). Наприклад, дружба між користувачами чи уподобання допису користувачем.

$$V = \{V_j^{(i)} \mid 1 \leq i \leq K_V, 1 \leq j \leq k_i\}, \quad (2.2)$$

$$E = \{E_j^{(i)} \mid 1 \leq i \leq K_E, 1 \leq j \leq k_i\}, \quad (2.3)$$

де  $i$  – позначає тип вузла чи зв'язку,

$j$  – позначає порядковий номер вузла чи зв'язку в середині  $V^i$  чи  $E^i$ .

Тож завдання для системи рекомендації профілів можна виразити наступним чином: дано вузол користувача  $v$  всередині  $V^1$ , потрібно знайти ранжований список деяких користувацьких вузлів, які можуть з'єднатися з  $v$ , ранжувати за релевантністю у порядку спадання. При чому вузли, які вже з'єднані з  $v$ , не повинні входити у цей список [3].



У заздалегідь розробленій соціальній мережі додати до профілю користувача дані про день народження, місто проживання та інтереси, які будуть застосовуватись рекомендаційною системою.

Розробити систему рекомендації профілів користувачів соціальної мережі, за їхніми інтересами на основі коефіцієнтів подібності та гібридної фільтрації. Гібридна фільтрація має включати у себе контентну, колаборативну та топологічну фільтрації.

Рекомендаційну систему реалізувати у вигляді програмного модуля на мові JavaScript. Система має приймати на вхід дані поточного (авторизованого) користувача та дані користувачів-кандидатів у рекомендації, у вигляді векторів або множин ознак. Далі система повинна проаналізувати вхідні дані та на основі цього аналізу сформувати масив з ід рекомендованих профілів для поточного користувача. РС повинна здійснювати аналіз на основі наступних даних:

- Профільні дані:
  - Рік народження;
  - Місце проживання:
    - Країна;
    - Область;
    - Місто;
  - Університет;
  - Інтереси;
- Уподобані дописи;
- Список друзів.

Інтегрувати розроблену рекомендаційну систему з соціальною мережею. Додати до API соціальної мережі кінцеву точку яка читає необхідні дані з бази даних, перетворює їх на вектори ознак, викликає рекомендаційну систему, передаючи у неї дані, обробляє результати роботи РС, повертає список рекомендованих профілів. Забезпечити виведення списку рекомендованих профілів у профілі користувача.

## 2.2 Методи рекомендації профілів

Одним із підходів до отримання ранжованого списку користувацьких вузлів є пошук користувачів, які схожі на поточного користувача. Щоб оцінити схожість між користувачами, потрібно спочатку перетворити дані користувачів на вектори або множини ознак. Подібність між користувачами в рамках різних фільтрацій визначається на основі різних даних. Далі у даному розділі розглядається, які саме дані використовуються у рамках контентної, колаборативної, топологічної та гібридної фільтраціях.

### Контентна фільтрація

У рамках контентної фільтрації подібність між користувачами визначають на основі профільних даних. Чим більше спільних даних у профілях користувачів тим вони подібніші, як показано на рисунку 2.1.

На рисунку 2.1 А та В мають спільну країну та місто проживання, навчалися (-ються) в одному університеті, мають два спільних інтереси. В та С користувачі мають спільну країну проживання та один спільний інтерес. Отже А та В більш подібні ніж В та С. Тому користувачу В імовірніше буде рекомендовано користувача А ніж користувача С.

Перед розрахунком подібності між користувачами їх профільні дані потрібно перетворити на вектори ознак. Наприклад, дано користувача з наступним профільними даними: `country_id: 230, city_id: 16254, university_id: 123`. Вектором ознак для даного користувача буде вектор: `[230, 16254, 123]`. У векторах усіх користувачів, дані повинні бути в однаковому порядку. Довжини векторів мають бути однаковими. Якщо у профілі користувача є невказані дані, замість них можна підставити 0 або інше спеціальне значення. Щоб уникнути випадків, коли обидва вектори мають 0 на одному і тому ж місці, потрібно формувати вектори на основі лише тих даних які є вказаними у профілі поточного користувача, для якого формується рекомендований список.

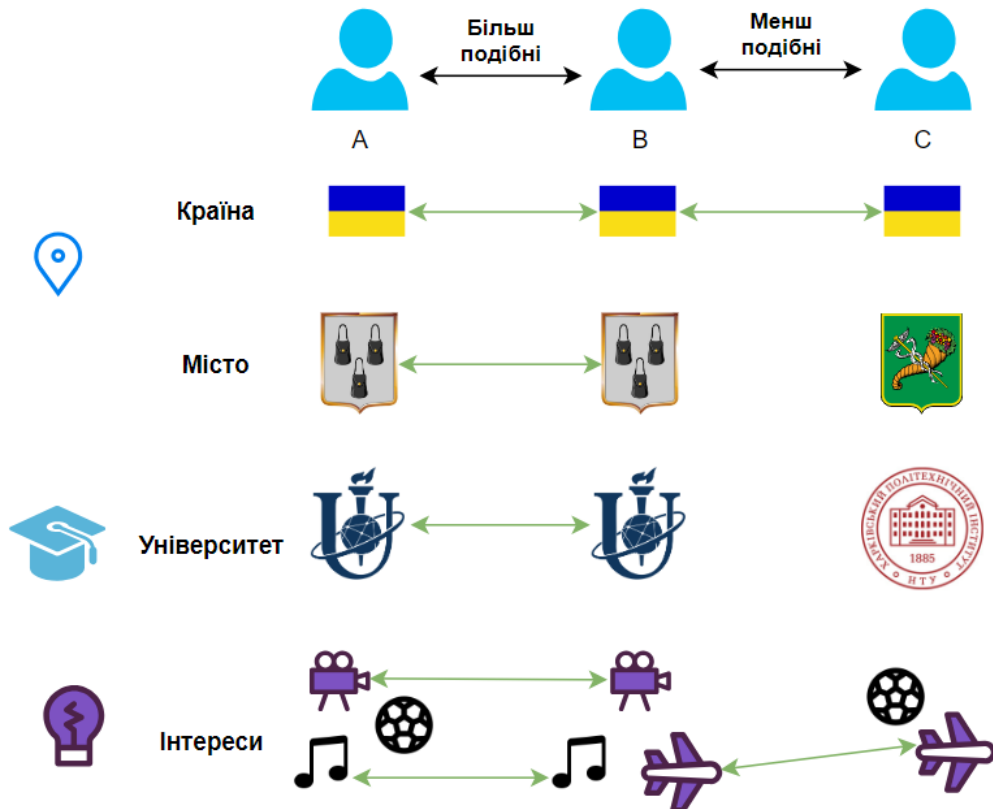


Рисунок 2.1 – порівняння користувачів на основі профільних даних

*Джерело: побудовано автором*

Для розрахунку подібності на основі інтересів векторний підхід не підходить, адже кожен користувач може мати різну кількість інтересів. Тому для цього підходить підхід на основі множин. Наприклад, дано користувача з наступними інтересами: `interest_id: 5, interest_id: 2, interest_id: 8`. Множиною ознак для даного користувача буде множина:  $\{5, 2, 8\}$ .

### Колаборативна фільтрація

За колаборативною фільтрацією, щоб знайти подібних користувачів, складають матрицю оцінок у вигляді Користувач-Предмет-Оцінка, де предметом може бути товар, послуга, фільм чи щось інше. Приклад матриці оцінок наведено у таблиці 2.1. Кожен рядок цієї матриці є вектором ознак користувача. Для зменшення кількості пропусків у матриці оцінок використовують різні підходи матричної факторизації.

Таблиця 2.1 – приклад матриці оцінок

	Предмет 1	Предмет 2	Предмет 3	Предмет 4	Предмет 5
Користувач 1	5	-	-	3	-
Користувач 2	4	3	-	5	-
Користувач 3	-	4	4	-	3

*Джерело:* побудовано автором

У соціальній мережі можна лише або уподобати допис, або ні. Тому для соціальної мережі матриця оцінок складається лише з одиниць, де одиниця означає уподобання допису користувачем. Приклад такої матриці наведено у таблиці 2.2.

Таблиця 2.2 – приклад матриці оцінок для соцмережі

	Допис 1	Допис 2	Допис 3	Допис 4	Допис 5
Користувач 1	1	-	1	-	-
Користувач 2	1	1	-	1	-
Користувач 3	-	1	-	-	1

*Джерело:* побудовано автором

Так як матриця оцінок у соцмережі складається лише з одиниць, більш оптимально застосувати множинний підхід до розрахунку подібності, ніж векторний. Тоді множиною ознак користувача, буде множина ід дописів, які подобаються користувачу.

Чим більше дописів подобаються обом користувачам, тим більша подібність між цими користувачами, як показано на рисунку 2.2.

На рисунку 2.2 показано, що користувачам А та В обом подобаються 3 дописи. Користувачам В та С обом подобається один допис. Тобто А та В більш подібні, ніж В та С. Отже, користувачу В імовірніше буде рекомендовано користувача А, ніж користувача С. При цьому користувачі А та С не мають жодного спільно уподобаного допису, тож вони не подібні. Тому їх не буде рекомендовано одне одному за колаборативною фільтрацією.

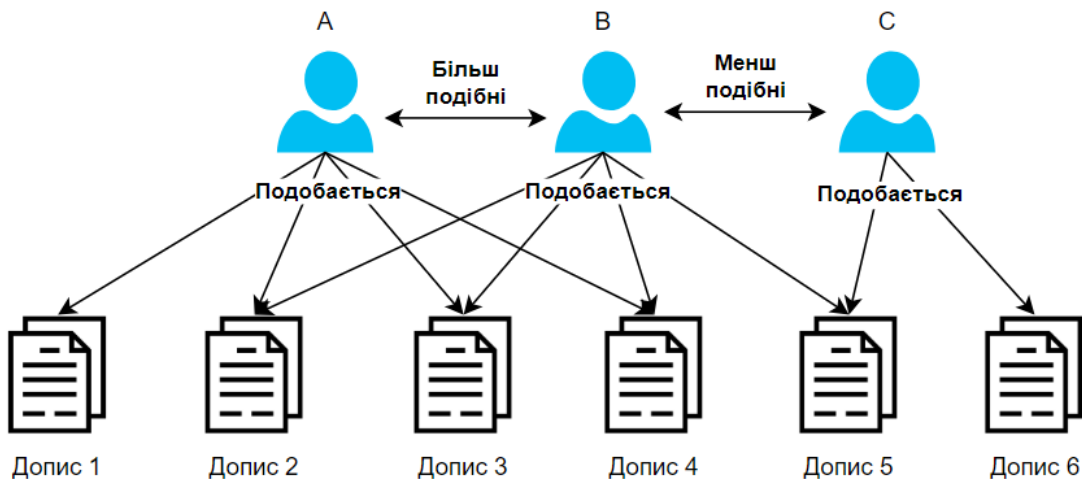


Рисунок 2.2 – порівняння користувачів на основі уподобань  
*Джерело: побудовано автором*

### Топологічна фільтрація

Існує кілька різних підходів до топологічної фільтрації. За підходом, заснованому на сусідстві, подібність між користувачами визначається на основі списку друзів. Чим більше спільних друзів, тим більша подібність між користувачами, як показано на рисунку 2.3.

На рисунку 2.3 показано, що користувачі А та В більш подібні ніж В та С. Адже А та В мають трьох спільних друзів, а В та С лише одного. Користувачі А та С не подібні, адже не мають жодного спільного друга.

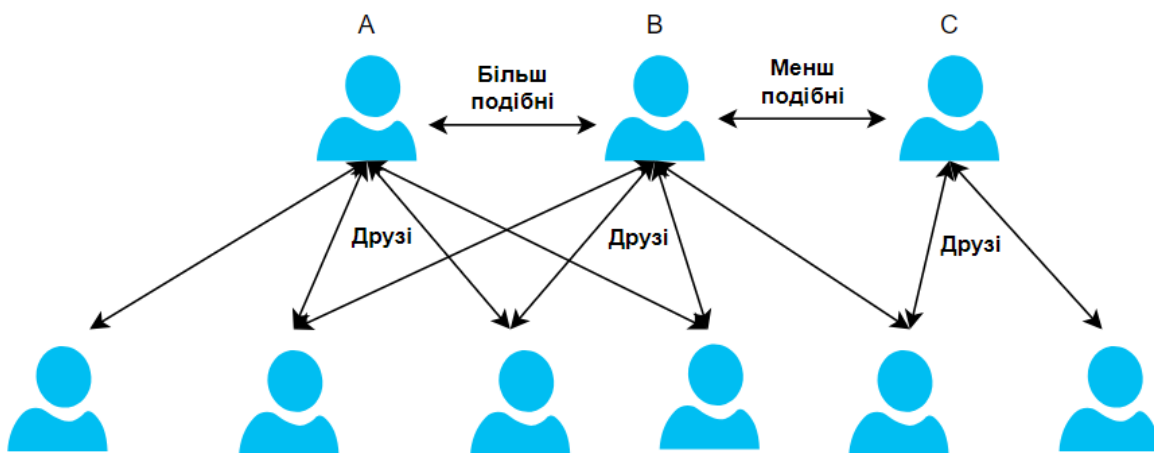


Рисунок 2.3 – визначення подібності на основі спільних друзів  
*Джерело: побудовано автором*

Для розрахунку подібності на основі списку друзів використовують підхід на основі множин, де множиною ознак користувача є множина id друзів користувача.

### Гібридна фільтрація

Гібридна фільтрація полягає поєднанні кількох, розглянутих вище фільтрацій. Однією з стратегій гібридизацію є стратегія на основі зваженої суми (формула 2.4). За цією стратегією декілька фільтрацій проводяться паралельно. Результатами фільтрацій є списки рекомендованих користувачів з чисельним значенням подібності. Далі ці списки об'єднуються за допомогою зваженої суми.

$$R = \sum_{i=1}^n w_i S_i \quad (2.4)$$

де  $i$  – номер фільтрації,

$w_i$  – вага поточної фільтрації,

$S_i$  – чисельне значення подібності, розраховане на основі поточної фільтрації.

Щоб кожна фільтрація мала рівний вплив на результат, можна застосувати вагу 1 для кожної фільтрації. Тобто розрахунок подібності повинен здійснюватися за формулою 2.5.

$$R = S_c + S_{cb} + S_t \quad (2.5)$$

де  $S_c$  – значення подібності, розраховане на основі контентної фільтрації,

$S_{cb}$  – значення подібності, розраховане на основі колаборативної фільтрації,

$S_t$  – значення подібності, розраховане на основі топологічної фільтрації.

На рисунку 2.4 наведено приклад об'єднання результатів контентної, колаборативної та топологічної фільтрацій, використовуючи формулу 2.5, де score це чисельне значення подібності.

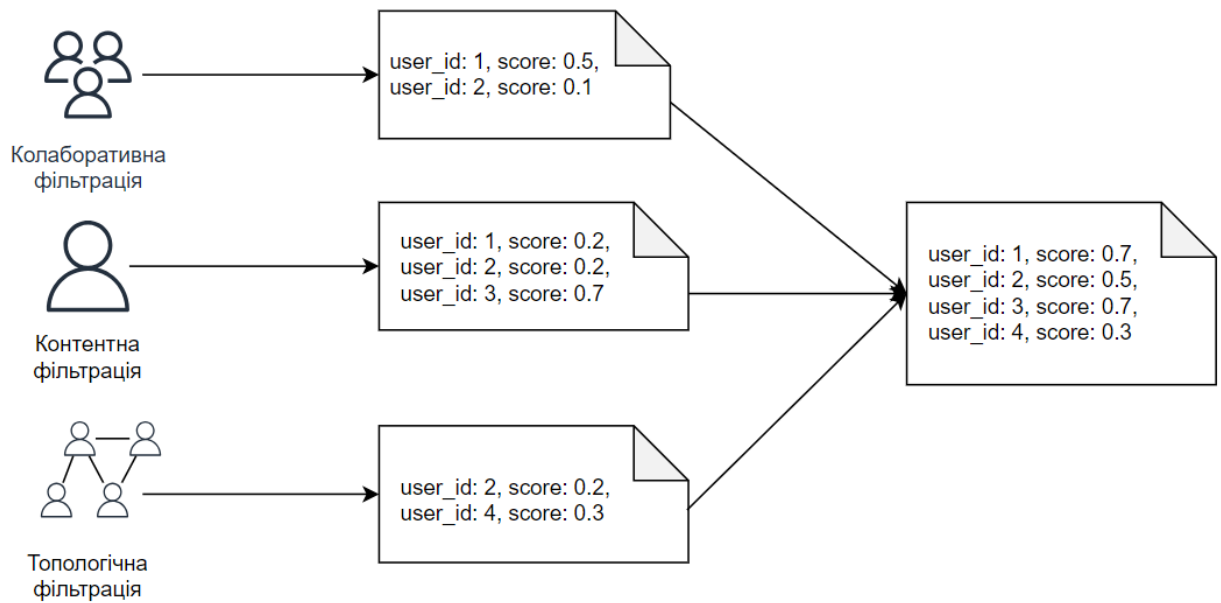


Рисунок 2.4 – об'єднання результатів кількох фільтрацій  
*Джерело: побудовано автором*

Після здійснення об'єднання результатів повинно відбутися сортування користувачів за спаданням значення подібності та формування рекомендованого списку з  $n$  найбільш подібних користувачів.

### 2.3 Коефіцієнти подібності що використовуються для рекомендацій профілів

Для чисельного розрахунку подібності між користувачами застосовуються коефіцієнти подібності. Прикладами таких коефіцієнтів є коефіцієнт Жаккара, Адаміка-Адара, Соренсена-Дайса, косинусна подібність та манхетенська відстань. Деякі з цих коефіцієнтів розглядаються у даному розділі.

Коефіцієнт Жаккара

Коефіцієнт Жаккара широко застосовується у таких напрямках, як інформатика, біологія, екологія, пошук подібних текстів, геоботаніка [33].

Коефіцієнт подібності Жаккара (Jaccard) розраховується як відношення кількості елементів перетину двох множин до кількості елементів їх об'єднання (формула 2.6) [17]. Область можливих значень коефіцієнта –  $[0, 1]$ . Якщо елементи множин повністю співпадають, то значення коефіцієнта дорівнює 1. Якщо елементи множин повністю не співпадають, то значення коефіцієнта дорівнює 0.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2.6)$$

Чим більше однакових елементів містять множини, тим більшим буде коефіцієнт подібності. При цьому не важливо який порядок елементів у множині, та чи однакова їх довжина (рисунок 2.5).

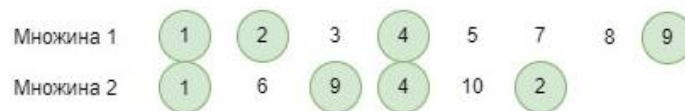


Рисунок 2.5 – порівняння двох множин  
Джерело: побудовано автором

У векторному вигляді коефіцієнт Жаккара розраховується за формулою 2.7.

$$Jaccard(u, v) = \frac{\sum_{i=1}^n u_i v_i}{\sum_{i=1}^n u_i + \sum_{i=1}^n v_i - \sum_{i=1}^n u_i v_i} \quad (2.7)$$

де  $u = (u_1 \dots u_n)$ ,  $v = (v_1 \dots v_n)$  – вектори ознак об'єктів [33].

При порівнянні векторів вектори повинні бути однакового розміру. При пошуку однакових елементів враховуються лише ті, які знаходяться на одній позиції (рисунок 2.6).



Вектор 1	1	2	3	4	5	7
Вектор 2	1	6	9	4	10	2

Рисунок 2.6 – порівняння векторів  
*Джерело: побудовано автором*

У рекомендаційній системі коефіцієнт Жаккара можна використати для різних фільтрацій:

- Контентна фільтрація;
- Колаборативна фільтрація [34, 36, 37];
- Фільтрації на основі топології [17, 35, 38].

Щоб оцінити подібність профілів на основі контентної фільтрації, використовуються профільні дані. Профільні дані двох користувачів перетворюються на вектори ознак, до яких застосовується формула 2.7. Вектори ознак формуються з таких даних, як ід країни, ід області, ід міста та ід університету. Більшість коефіцієнтів подібності враховують, наскільки сильно кожне значення у векторах відрізняється одне від одного. Але якщо вектори складаються лише з ід, то врахування різниці між ід може спотворити результати. Коефіцієнт Жаккара цього не враховує, тому він підходить для розрахунку подібності профілів краще ніж інші коефіцієнти.

Під час колаборативної фільтрації формується матриця оцінок Користувач-Предмет. Коефіцієнт Жаккара показує подібність між двома користувачами на основі кількості предметів, які оцінили обидва користувачі. При цьому не враховується, які саме оцінки поставили користувачі. Тому коефіцієнт Жаккара у колаборативній фільтрації не застосовують окремо, а разом з іншими коефіцієнтами подібності, які враховують оцінки. Однак у соціальній мережі матриця оцінок бінарна. Тобто користувач або уподобав допис, або ні. Тому коефіцієнт Жаккара підходить для застосування під час колаборативної фільтрації у соціальній мережі.

Під час фільтрації на основі топології коефіцієнт Жаккара розраховується на основі списків друзів користувачів. Список друзів кожного користувача розглядається

як множина іd користувачів. Щоб знайти подібність між двома користувачами використовуються списки їх друзів, до яких застосовується формула 2.6.

### Коефіцієнт Адаміка-Адара

Коефіцієнт Адаміка-Адара використовується для обчислення подібності між двома вузлами у графі, що застосовується для передбачення виникнення зв'язку. Подібність між вузлами  $x$  та  $y$  розраховується за формулою 2.8. Чим більше значення коефіцієнта, тим імовірніше виникнення зв'язку між вузлами [16, 39].

$$AA(x, y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|} \quad (2.8)$$

де  $u$  – спільний сусід для  $x$  та  $y$ ,

$|N(u)|$  – кількість сусідів вузла  $u$ .

Соціальну мережу можна представити у вигляді графа, де користувачі є вузлами, а дружба між ними є ребрами. Таким чином коефіцієнт Адаміка-Адара застосовують у рекомендаційних системах для передбачення виникнення дружби між користувачами [6, 16, 18, 36].

### Косинусна подібність

Косинусна міра подібності вимірює кут між двома векторами. Чим менший кут тим більша подібність [37]. Косинусна подібність розраховуються за формулою 2.9.

$$\text{cosine}(u, v) = \frac{\sum_{i=1}^n u_i v_i}{\sqrt{\sum_{i=1}^n u_i^2 \sum_{i=1}^n v_i^2}} \quad (2.9)$$

де  $u = (u_1 \dots u_n)$ ,  $v = (v_1 \dots v_n)$  – вектори ознак об'єктів [33].

Косинусна подібність для множин розраховується за формулою 2.10 [15].

$$\text{cosine}(A, B) = \frac{|A \cap B|}{\sqrt{|A| * |B|}} \quad (2.10)$$

Косинусна міра подібності широко застосовується у рекомендаційних системах при колаборативній фільтрації, для пошуку користувачів зі схожими уподобаннями. При цьому в якості ознак об'єктів, застосовуються оцінки користувачів. Зазвичай для цього використовують векторний підхід (формула 2.9) [4, 34, 37, 40]. Однак для обчислень на основі бінарної матриці оцінок, добре підходить формула 2.10 для множин. Множинами ознак для користувачів будуть множини з ід уподобаних дописів.

При контентній фільтрації також можна застосувати косинусну подібність. Для цього потрібно створити вектор ознак на основі профільних даних. Однак профільні дані складаються з ід країни, ід області, ід міста та ід університету, і коефіцієнт буде враховувати те, наскільки сильно ід різних користувачів відрізняються одне від одного.

Так само як і коефіцієнт Жаккара, косинусну подібність на основі множин (формула 2.10) можна застосувати при топологічній фільтрації.

#### Порівняння коефіцієнтів подібності

У таблиці 2.3 показано відповідність між типом фільтрації та коефіцієнтом подібності, який може застосовуватись під час цієї фільтрації.

Таблиця 2.3 – відповідність між типом фільтрації та коефіцієнтом подібності

Фільтрація	Контентна	Колаборативна	Топологічна
Коефіцієнт			
Жаккара	+	+	+
Адаміка-Адара	-	-	+
Косинусна подібність	+	+	+

*Джерело: побудовано автором*

У таблиці 2.4 наведено порівняльний аналіз векторного та множинного підходів коефіцієнта Жаккара та косинусної подібності для контентної фільтрації.

Таблиця 2.4 – порівняння коефіцієнтів для контентної фільтрації

Коефіцієнт Жаккара		Косинусна подібність	
для векторів	для множин	для векторів	для множин
для розрахунку подібності на основі інтересів потрібен додатковий крок перетворення даних в матрицю Користувач-Інтерес	для розрахунку подібності на основі інтересів не потрібен додатковий крок	для розрахунку подібності на основі інтересів потрібен додатковий крок перетворення даних в матрицю Користувач-Інтерес	для розрахунку подібності на основі інтересів не потрібен додатковий крок
не враховує відстань між елементами вектора ознак	не підходить для розрахунку подібності на основі інших профільних даних	враховує відстань між елементами вектора ознак	не підходить для розрахунку подібності на основі інших профільних даних
підходить краще ніж косинусна подібність	дає такі ж результати як і косинусна подібність для множин	підходить гірше ніж коефіцієнт Жаккара	дає такі ж результати як і коефіцієнт Жаккара для множин

*Джерело:* побудовано автором

В результаті для розроблюваної рекомендаційної системи обрано коефіцієнт Жаккара. Для розрахунку подібності на основі інтересів обрано підхід на основі множин. Для розрахунку подібності на основі інших профільних даних обрано підхід на основі векторів.

У таблиці 2.5 наведено порівняння векторного та множинного підходів коефіцієнта Жаккара та косинусної подібності для колаборативної фільтрації.

Таблиця 2.5 – порівняння коефіцієнтів для контентної фільтрації

Коефіцієнт Жаккара		Косинусна подібність	
для векторів	для множин	для векторів	для множин
потрібен додатковий крок перетворення даних в матрицю Користувач-Допис	не потрібен додатковий крок	потрібен додатковий крок перетворення даних в матрицю Користувач-Допис	не потрібен додатковий крок
	дає такі ж результати як і косинусна подібність для множин		дає такі ж результати як і коефіцієнт Жаккара для множин

*Джерело:* побудовано автором

Отже, для колаборативної фільтрації краще підходить підхід на основі множин. При чому коефіцієнт Жаккара та косинусна подібність для множин, однаково підходять для цієї фільтрації. Так як коефіцієнт Жаккара обрано для контентної фільтрації, то вирішено застосувати його і для колаборативної фільтрації.

У таблиці 2.6 наведено порівняння коефіцієнтів Жаккара та Адаміка-Адара для топологічної фільтрації.

Таблиця 2.6 – порівняння коефіцієнтів для топологічної фільтрації

Коефіцієнт Жаккара (для множин)	Коефіцієнт Адаміка-Адара
враховує кількість спільних сусідів	враховує кількість спільних сусідів та їх сусідів
потребує менше даних	потребує більше даних

*Джерело:* побудовано автором

Отже, обидва коефіцієнта подібності підходять для вирішення даної задачі, але коефіцієнт Жаккара потребує менше даних, тому його було обрано для реалізації.

## 3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

### 3.1 Структурно-функціональне моделювання системи рекомендації профілів

У рамках структурно-функціонального моделювання розроблено IDEF0 діаграма процесу надання рекомендованих профілів, який реалізується розробленою інформаційною технологією. Метою цієї моделі є відображення структурних представлень функцій, залучених до роботи системи. Модель IDEF0 складається з декількох ієрархічних та пов'язаних діаграм, які відображають декомпозицію досліджуваного процесу. IDEF0 діаграми складаються з блоків та стрілок (або дуг). Блоки відображають певний процес, функцію або систему. Входи та виходи, тобто вхідні та вихідні дані, відображаються як стрілки зліва та справа від блоків. Керування та механізми, відображаються як стрілки згори та знизу від блоків [41].

Розроблену IDEF0 діаграму зображено на рисунку 3.1. Об'єктом є інформаційна технологія рекомендації профілів за спільними інтересами. Вхідними даними є інформація про авторизованого користувача, для якого потрібно створити список рекомендацій, та інформація про користувачів, які розглядаються як кандидати для рекомендацій. Вихідними даними є список рекомендованих профілів. Елементами керування є рекомендаційні алгоритми та коефіцієнт Жаккара, за яким розраховується подібність між користувачами, а також модель даних, тобто те, у якому вигляді дані зберігаються у БД та передаються всередині застосунку. Механізмами є соціальна мережа та рекомендаційна система.

Щоб уникнути перевантаження системи, результати рекомендацій потрібно зберегти до бази даних та надавати користувачу збережений список. Один раз на добу слід запускати знову процес формування рекомендацій та оновлювати список рекомендацій у базі даних, щоб врахувати дії користувачів, які відбулись за останню добу.

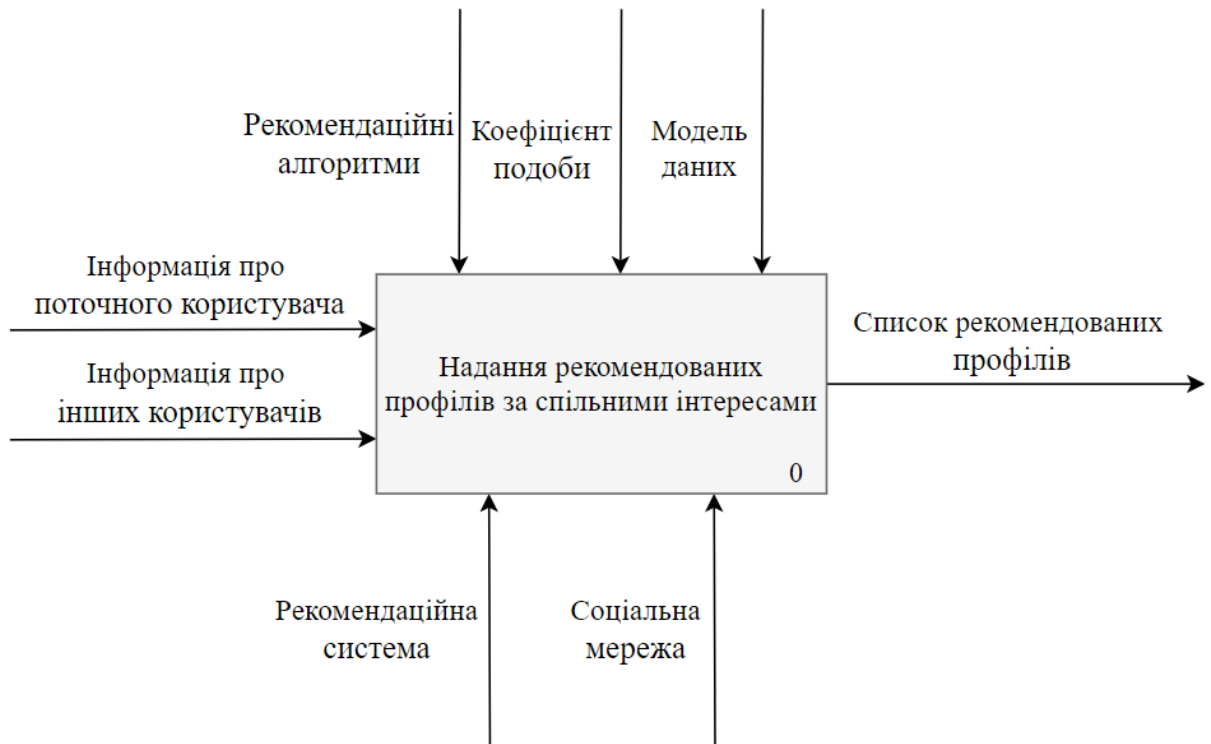


Рисунок 3.1 – IDEF0 діаграма процесу надання рекомендованих профілів  
*Джерело: побудовано автором*

На рисунку 3.2 зображено декомпозицію IDEF0 діаграми. Інформаційна технологія надання рекомендацій складається з етапів:

- На першому етапі відбувається перетворення вхідної інформації на вектори ознак для поточного користувача та користувачів-кандидатів у рекомендації.
- На другому етапі, на основі векторів ознак, відбуваються розрахунки коефіцієнтів подібності для користувачів-кандидатів.
- На третьому етапі, на основі коефіцієнтів подібності, відбувається відбір 10 рекомендованих користувачів.
- На четвертому етапі, соцмережа читає з бази даних інформацію про рекомендованих користувачів, необхідну для їх відображення.

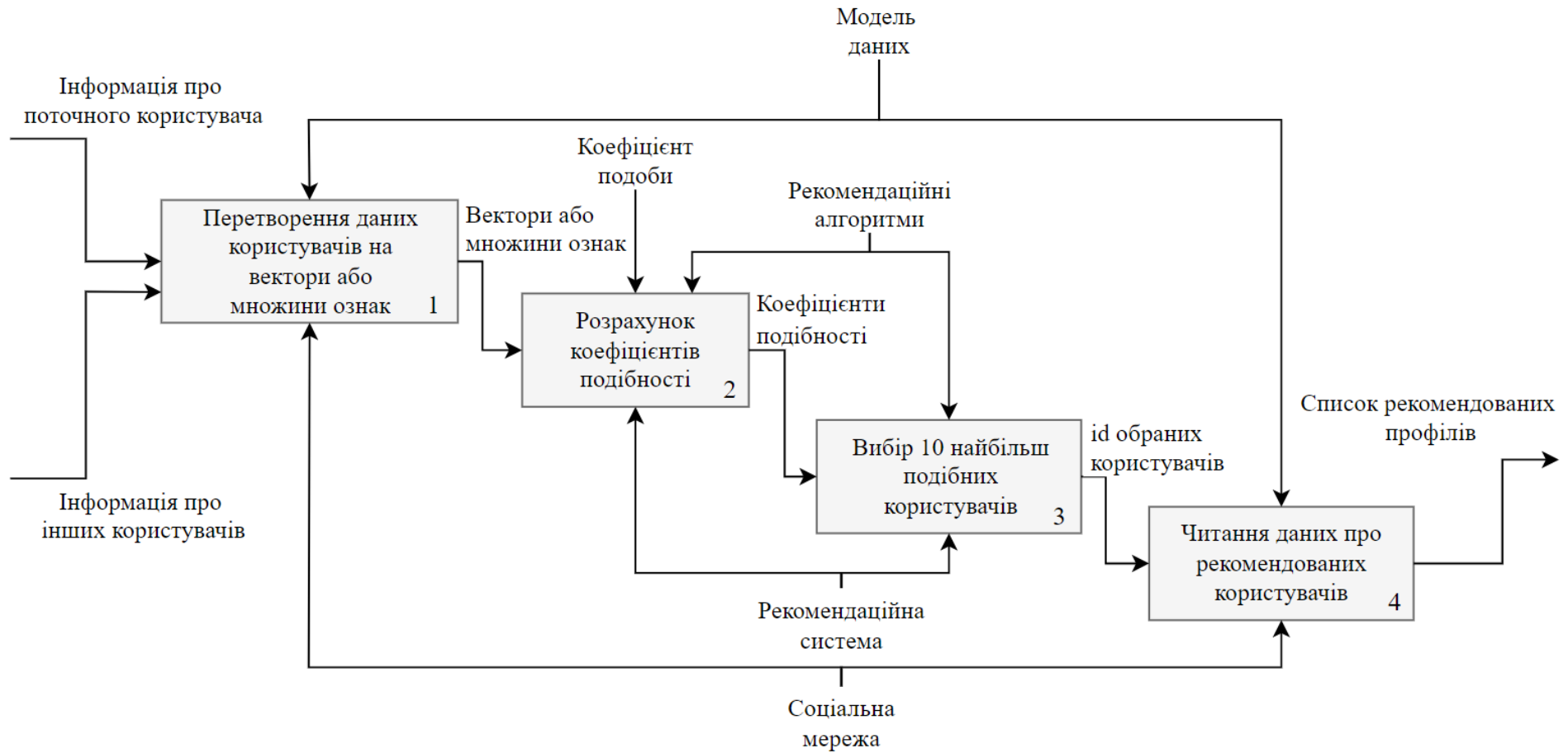


Рисунок 3.2 – Декомпозиція діаграми IDEF0

Джерело: побудовано автором



### 3.2 Моделювання варіантів використання системи рекомендації профілів

Модель варіантів використання використовують для відображення функціональних можливостей системи. Модель складається з акторів, варіантів використання та взаємозв'язків між ними. Діаграму варіантів використання для розробленої рекомендаційної системи зображено на рисунку 3.3. На діаграмі виділено 2 актори:

1. Користувач;
2. База даних соціальної мережі, яка містить дані про користувачів.

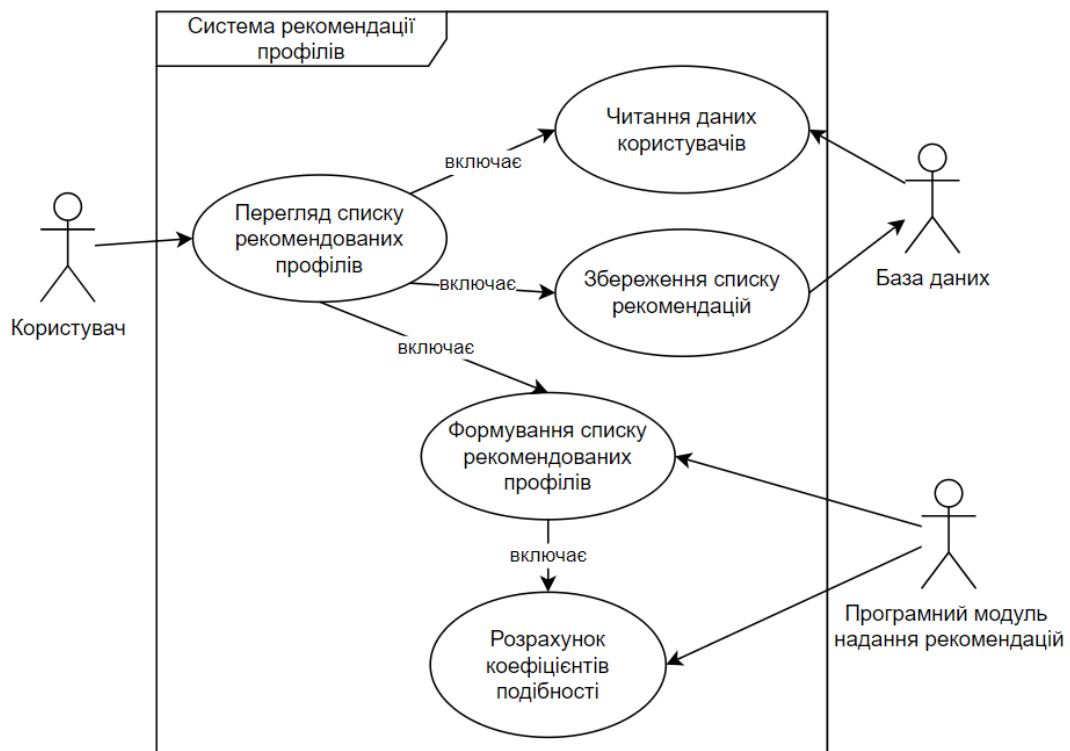


Рисунок 3.3 – діаграма варіантів використання системи рекомендації профілів

*Джерело: побудовано автором*

Розроблена рекомендаційна системи надає користувачу можливість перегляду списку рекомендованих профілів, що включає читання інформації про користувачів, з бази даних, та формування списку рекомендованих користувачів на основі

коефіцієнтів подібності, а також збереження сформованого списку рекомендацій до бази даних.

### 3.3 Проектування моделі послідовності для соціальної мережі та рекомендаційної системи

На рисунку 3.4 наведено діаграму послідовності для авторизації у соціальну мережу.

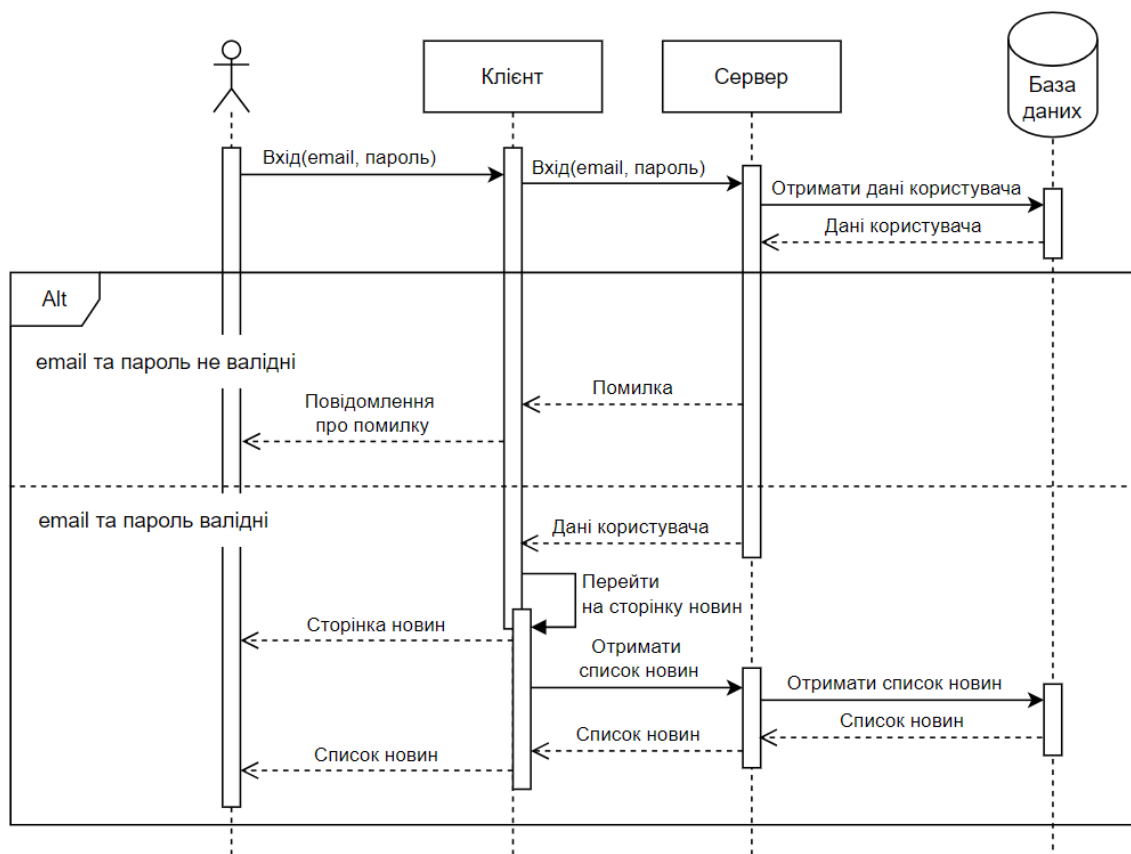


Рисунок 3.4 – діаграма послідовності авторизації у соцмережу  
Джерело: побудовано автором

Користувач вводить адресу електронної пошти (email) та пароль і натискає кнопку LogIn. Клієнт відправляє введений email та пароль на сервер. Сервер шукає у

базі даних користувача за вказаним email та перевіряє правильність введеного пароля. Якщо пароль невірний, сервер повертає помилку. Після чого клієнт показує користувачу повідомлення про помилку. Якщо пароль вказано вірно, сервер повертає дані користувача. Клієнт записує дані користувача у контекст авторизації та у localStorage браузера, після чого відбувається перенаправлення на сторінку новин. Спочатку користувач бачить сторінку з індикатором завантаження, потім на сторінку завантажуються список новин.

На рисунку 3.5 показано діаграму послідовності для завантаження сторінки профілю користувача у соціальній мережі.

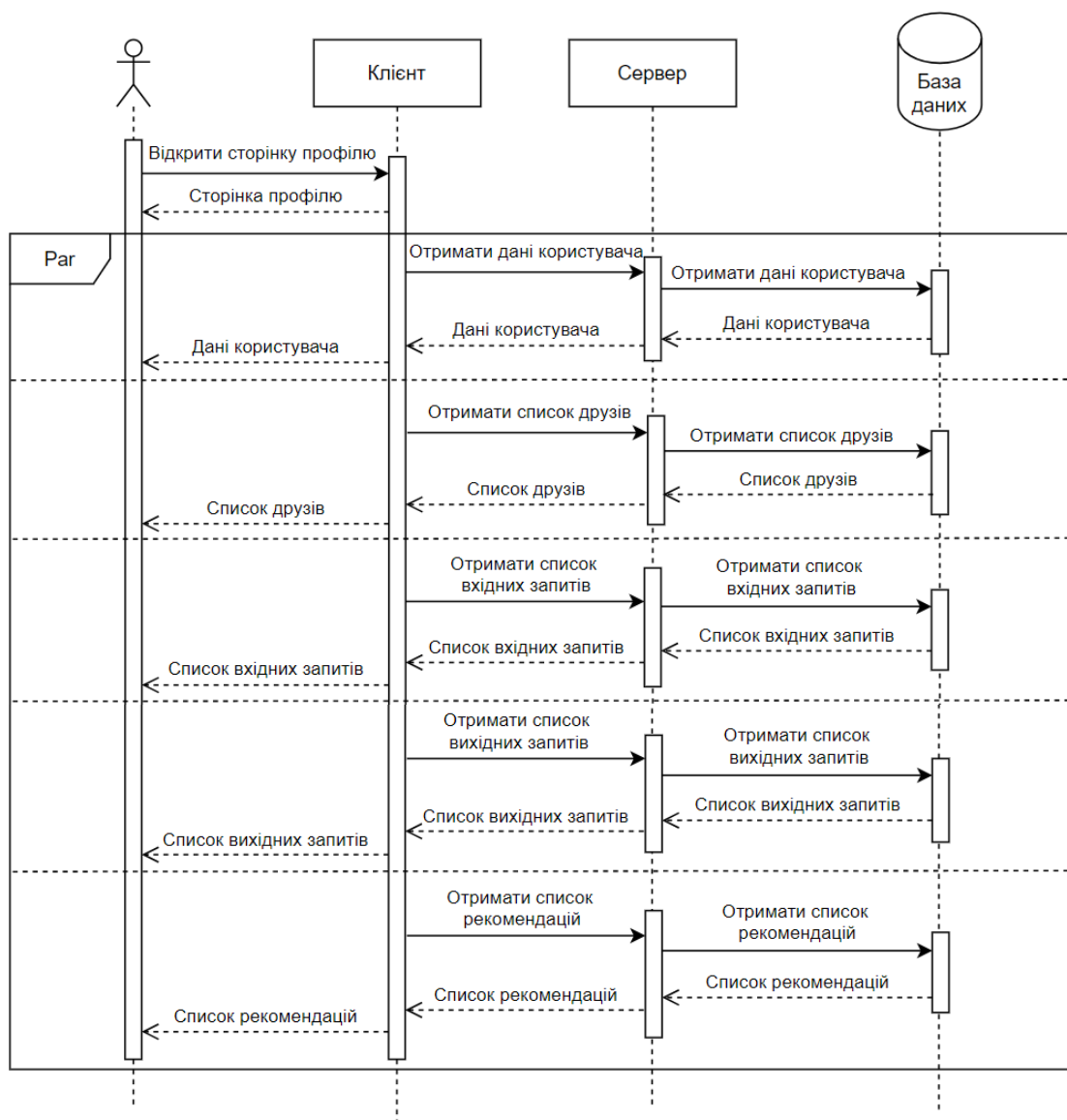


Рисунок 3.5 – діаграма послідовності завантаження сторінки профілю

*Джерело: побудовано автором*

Коли користувач переходить на сторінку профілю, відображається сторінка з індикаторами завантаження. Клієнт робить паралельні запити на сервер, щоб отримати дані користувача, список друзів, списки вхідних та вихідних запитів у друзі, та список рекомендованих профілів. Сервер робить запити до бази даних, щоб отримати ці дані, та повертає їх клієнту. Клієнт відображає отримані дані у відповідних блоках сторінки профілю.

На рисунку 3.6 наведено більш детальну діаграму послідовності завантаження списку рекомендованих профілів. Коли користувач відкриває сторінку профілю, клієнт робить запит на сервер на отримання списку рекомендованих профілів. Сервер читає з бази даних рекомендації для поточного користувача, де вказано також дату створення рекомендацій. Якщо рекомендації існують та з моменту їх створення ще не пройшла доба, сервер повертає клієнту список рекомендацій отриманий з бази даних. Якщо рекомендації існують та з моменту їх створення пройшла доба, сервер також повертає клієнту список рекомендацій, отриманий з бази даних та асинхронно запускає процес оновлення списку рекомендацій. В результаті відбувається перерахування подібності між користувачами та вибір найбільш подібних користувачів. Оновлений список рекомендацій записується у базу даних. Якщо у базі даних відсутні рекомендації для поточного користувача, сервер спочатку запускає процес створення списку рекомендацій та їх запису у базу даних, а потім повертає створений список, клієнту. Клієнт відображає у блоці рекомендацій список рекомендованих профілів, отриманий від сервера.

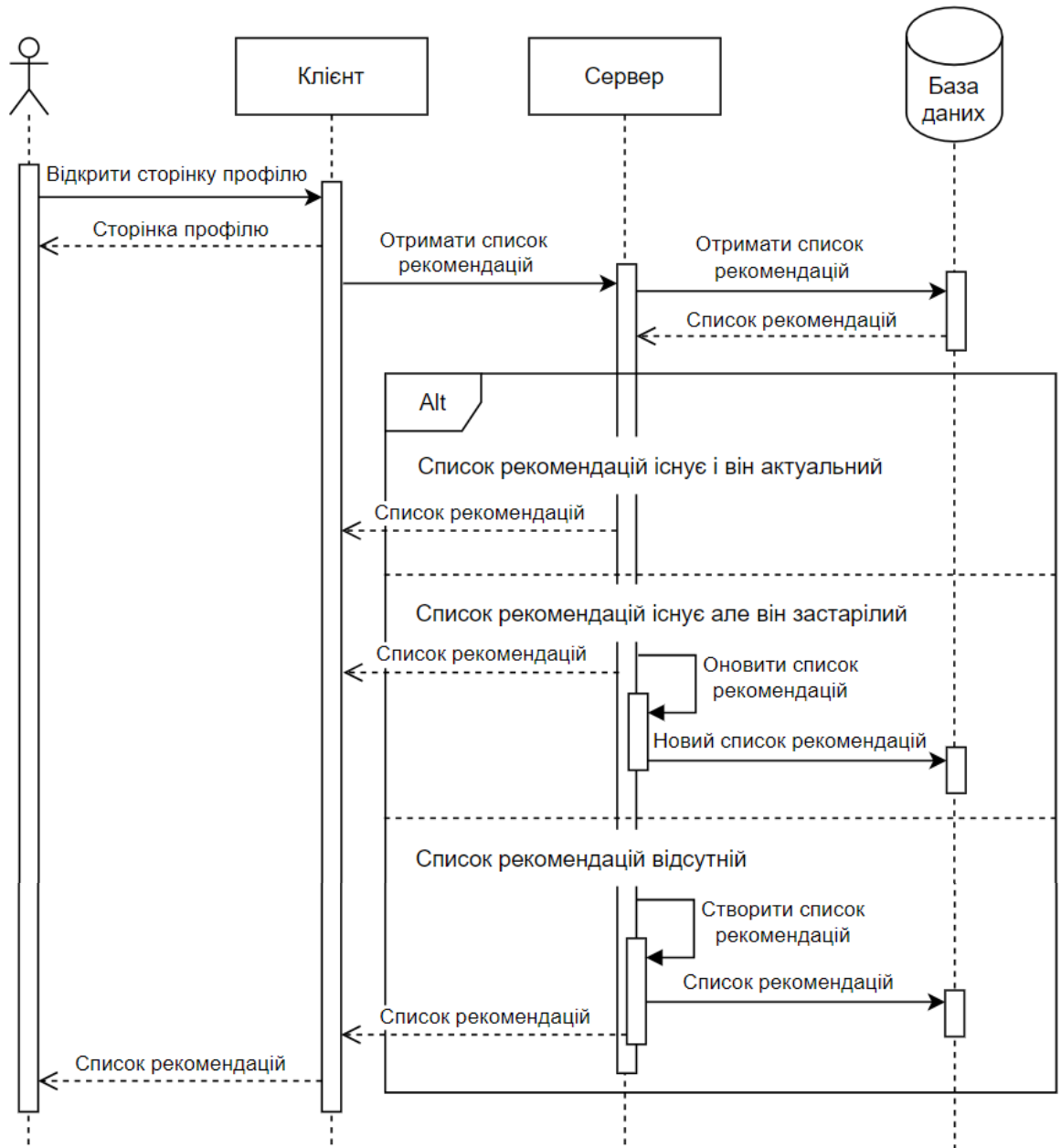


Рисунок 3.6 – діаграма послідовності завантаження списку рекомендацій  
*Джерело: побудовано автором*

На рисунку 3.7 зображено діаграму послідовності додавання користувача у друзі у соціальній мережі.

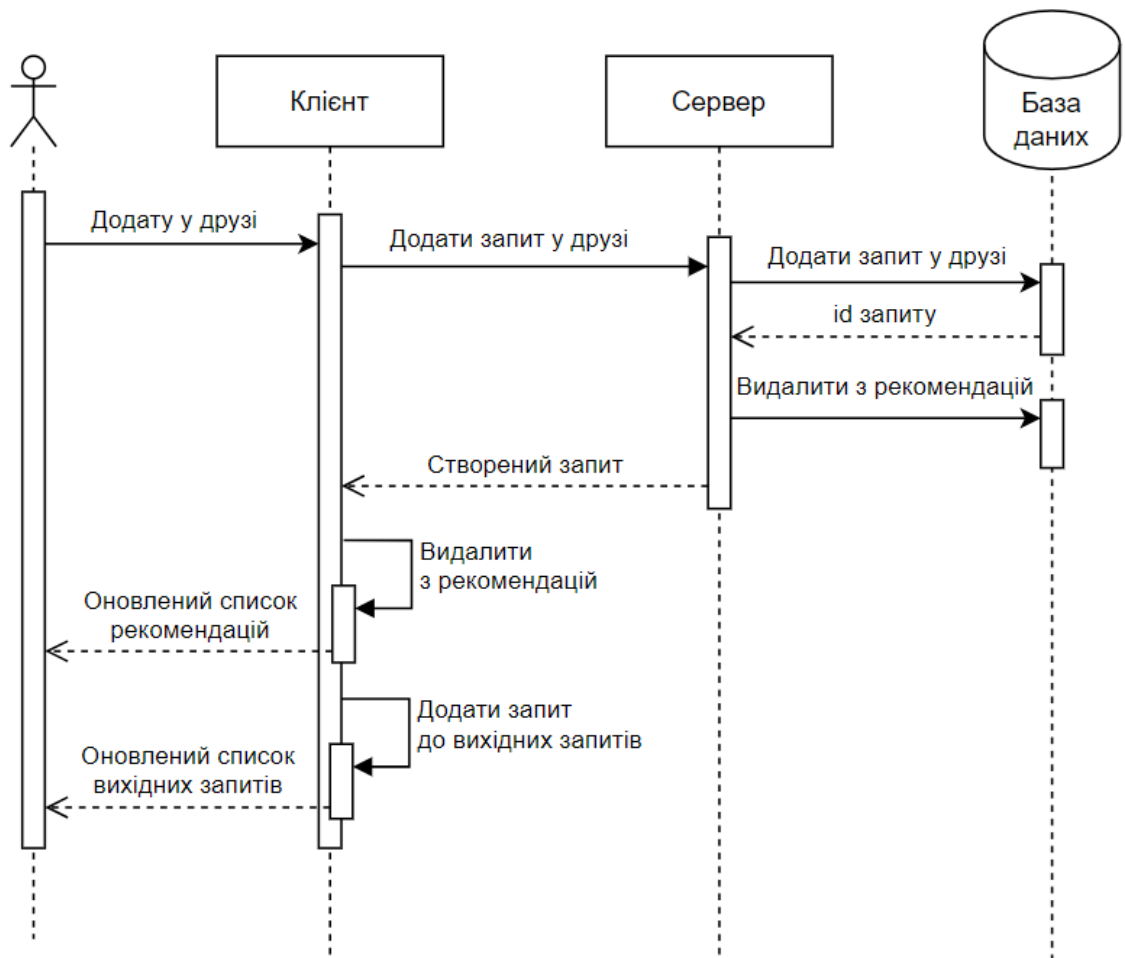


Рисунок 3.7 – діаграма послідовності додавання у друзі  
*Джерело: побудовано автором*

Коли користувач натискає додати у друзі, клієнт відправляє запит на сервер, передаючи id поточного користувача та id користувача, якому потрібно надіслати запит у друзі. Сервер створює новий запит у друзі у базі даних та видаляє користувача зі списку рекомендованих профілів у базі даних. Далі сервер повертає клієнту створений запит. Клієнт видаляє користувача, якому було надіслано запит, із списку рекомендованих профілів, та додає його до списку вихідних запитів. Користувач бачить оновлені списки рекомендацій та вихідних запитів у друзі.

### 3.4 Проектування моделі бази даних соціальної мережі

Технологія рекомендації профілів розроблюється для застосування у заздалегідь розробленій соціальній мережі. Для виконання рекомендацій використовуються дані про користувачів з бази даних соціальної мережі. На рисунку 3.8 зображено фрагмент фізичної моделі бази даних соцмережі, де наведено лише ті таблиці, які застосовуються при наданні рекомендацій.

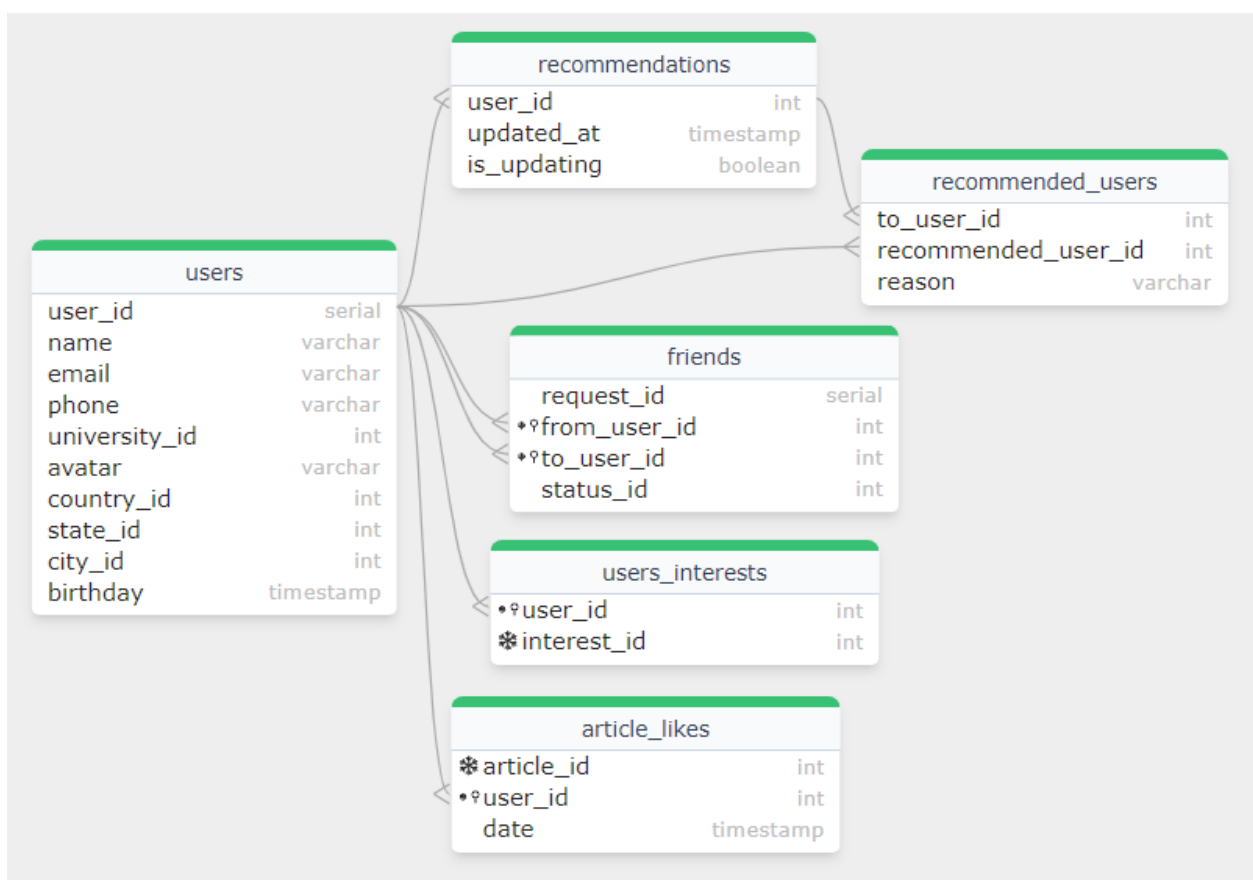


Рисунок 3.8 – фрагмент фізичної моделі бази даних соціальної мережі  
Джерело: побудовано автором

У таблиці 3.1 наведено опис таблиць, зображених на рисунку 3.8.

У таблиці 3.2 наведено опис полів таблиць бази даних соцмережі, наведених у таблиці 3.1 та на рисунку 3.8.

Таблиця 3.1 – Опис таблиць бази даних соціальної мережі

Назва таблиці	Опис таблиці
users	Містить дані користувачів.
user_interests	Реалізує відношення багато до багатьох між користувачами та їх інтересами.
friends	З'єднує користувачів, які є друзями, або між якими є запит у друзі.
article_likes	З'єднує користувачів та статті, які їм сподобались.
recommendations	Містить дані про час створення списку рекомендацій.
recommended_users	Містить список рекомендованих користувачів

*Джерело: побудовано автором*

Таблиця 3.2 – структура таблиць бази даних соціальної мережі

Таблиця	Поле	Опис
users	user_id	Ідентифікатор користувача
	name	Ім'я користувача
	email	Електронна адреса користувача
	phone	Номер телефону користувача
	university_id	Ідентифікатор університету у якому навчався(-ється) користувач
	avatar	Посилання на аватар користувача
	country_id	Ідентифікатор країни проживання користувача
	state_id	Ідентифікатор області проживання користувача
	city_id	Ідентифікатор міста проживання користувача
	birthday	Дата народження користувача
friends	request_id	Ідентифікатор запиту у друзі
	from_user_id	Ідентифікатор користувача, який надіслав запит
	to_user_id	Ідентифікатор користувача, якому надійшов запит
	status_id	Ідентифікатор статусу запиту
users_interests	user_id	Ідентифікатор користувача
	interest_id	Ідентифікатор інтересу



## Продовження таблиці 3.2

Таблиця	Поле	Опис
article_likes	article_id	Ідентифікатор допису
	user_id	Ідентифікатор користувача, якому сподобався допис
	date	Дата уподобання
recommendations	user_id	Ідентифікатор користувача, для якого створено список рекомендацій
	updated_at	Час останнього оновлення списку рекомендацій
	is_updating	Показує чи оновлюється список на даний момент
recommended_users	to_user_id	Ідентифікатор користувача для якого призначено дану рекомендацію
	recommended_user_id	Ідентифікатор користувача якого рекомендовано користувачу to_user_id
	reason	Повідомлення про причину рекомендації

*Джерело:* побудовано автором

Отже, у даному розділі розглянуто таблиці, що містять дані користувачів, які можна використати для розрахунку подібності між користувачами для формування списку рекомендованих профілів. Також розглянуто таблиці для збереження списку рекомендацій.

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 4.1 Прототип соціальної мережі

Рекомендаційну систему розроблено для заздалегідь розробленого прототипу соціальної мережі. Прототип створено у вигляді веб-додатку з використанням клієнт-серверної архітектури мовою JavaScript. Клієнтську частину додатку розроблено за допомогою бібліотеки React. Серверну частину додатку реалізовано у вигляді API з використанням фреймворку Express. Для зберігання даних використано СУБД PostgreSQL. Авторизацію реалізовано за допомогою JWT токenu. Для реалізації авторизації за допомогою Facebook та Google використано технологію OAuth 2.0. На рисунку 4.1 зображено архітектуру розробленого додатку.

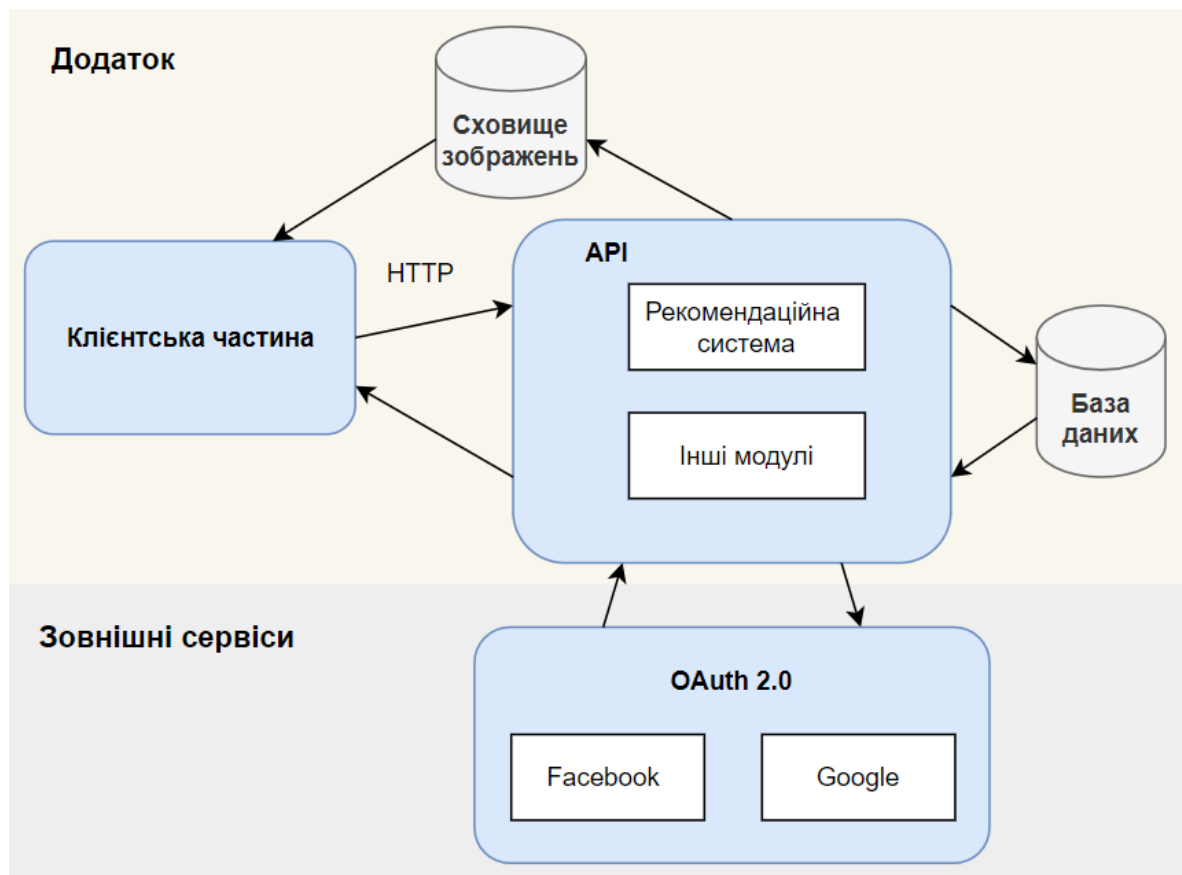


Рисунок 4.1 – архітектура соціальної мережі

*Джерело: побудовано автором*

Користувач соціальної мережі має можливість:

- Зареєструватись/авторизуватись (у тому числі за допомогою Google та Facebook);
- Створювати/редагувати/видаляти дописів;
- Ставити лайки на дописи;
- Створювати/редагувати/видаляти коментарів до дописів;
- Редагувати власний профіль;
- Надсилати запити на дружбу, видаляти їх, видаляти з друзів;
- Приймати або відхиляти вхідні запити на дружбу;
- Переглядати список рекомендованих користувачів.

Для того, щоб продемонструвати роботу системи, базу даних наповнено тестовими даним. Дані користувачів та дописів отримано за допомогою API для генерації тестових даних [42, 43]. Частину користувацьких даних а також уподобайки та зв'язки між користувачами згенеровано випадковим чином.

На рисунку 4.2 наведено сторінку авторизації розробленої соцмережі.

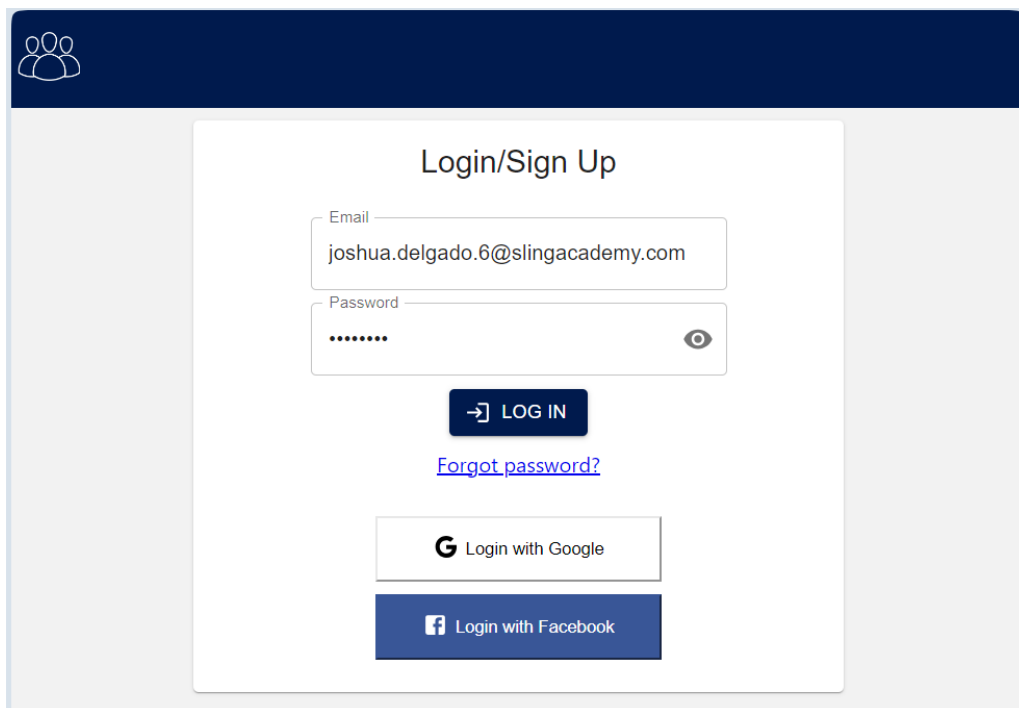


Рисунок 4.2 – сторінка авторизації у соцмережу  
*Джерело: побудовано автором*

Після авторизації користувач потрапляє на сторінку з новинами, де відображаються дописи його друзів та його власні дописи. Дану сторінку зображено на рисунку 4.3.



Рисунок 4.3 – сторінка з дописами  
*Джерело: побудовано автором (знімок з екрану)*

У рамках роботи до профілю користувача було додано наступні поля: країна, область, місто, день народження та інтереси. На рисунках 4.4-4.5 наведено форму для редагування профілю користувача. У профілі користувача є необов'язкові поля, тому на рисунках деякі з полів не вказані.

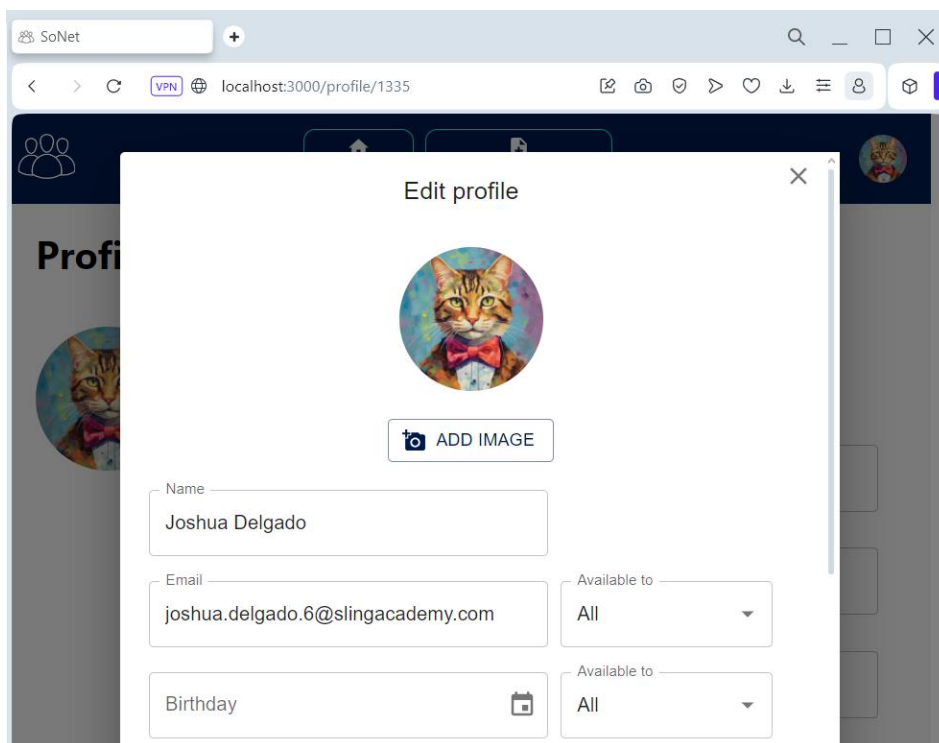


Рисунок 4.4 – форма редагування профілю 1 частина  
*Джерело: побудовано автором (знімок з екрану)*

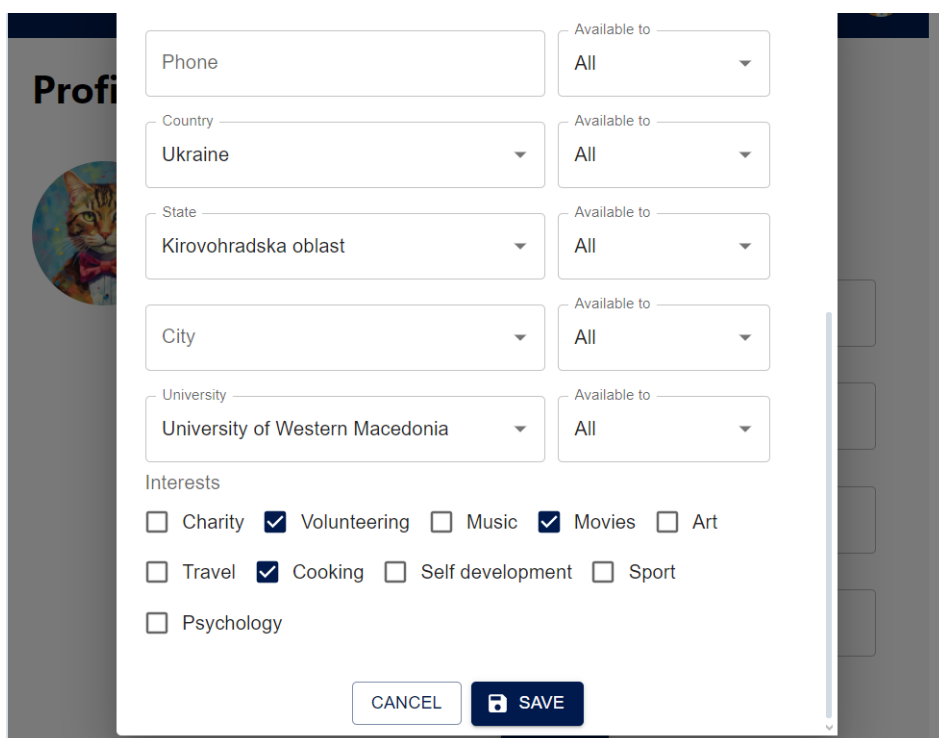


Рисунок 4.5 – форма редагування профілю 2 частина  
*Джерело: побудовано автором (знімок з екрану)*

## 4.2 Реалізація рекомендаційної системи

Рекомендаційну систему реалізовано на мові JavaScript у вигляді ряду функцій, які не залежать від соціальної мережі чи структури даних в БД. На рисунку 4.6 зображено файли, з яких складається рекомендаційна система.

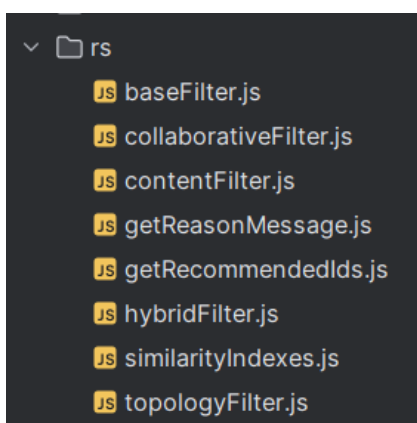


Рисунок 4.6 – файли рекомендаційної системи  
*Джерело:* побудовано автором (знімок з екрану)

`similarityIndexes.js` містить методи, які приймають на вхід два вектори або дві множини, розраховують на їх основі коефіцієнт подібності та повертають його значення.

`collaborativeFilter.js`, `contentFilter.js`, `topologyFilter.js` містять методи, які приймають на вхід вектор або множину ознак поточного користувача, та вектори або множини ознак користувачів кандидатів у рекомендації. Повертають масиви об'єктів, які містять `id` користувача та значення коефіцієнта подібності для цього користувача та назву поточної фільтрації.

`baseFilter.js` містить спільний код з попередніх фільтрацій.

`getReasonMessage.js` містить функцію, яка отримує на вхід список фільтрацій, на основі яких рекомендовано користувача, та формує повідомлення про причину рекомендації.

`getRecommendenIds.js` містить функцію, яка приймає на вхід масив об'єктів, який містить `id` користувачів, значення подібності та список фільтрацій. Масив сортується за спаданням значення подібності та обираються перші  $n$  користувачів зі списку. Для кожного з обраних користувачів формується повідомлення про причину рекомендації за допомогою `getReasonMessage.js`. У результаті метод повертає масив `id` обраних користувачів та масив з повідомленнями про причину рекомендації.

`hybridFilter.js` містить функцію, яка приймає на вхід масив, що містить кілька масивів об'єктів, які містять `id` користувачів, значення подібності та список фільтрацій. Для кожного користувача, з вхідних масивів, функція розраховує сумарне значення коефіцієнтів подоби.

Код описаних вище файлів наведено у додатку Б.

### 4.3 Впровадження рекомендаційної системи у прототип соціальної мережі

У рамках впровадження рекомендаційної системи у розроблений прототип соціальної мережі:

- На серверній частині додатку реалізовано:
  - кінцеву точку для отримання списку рекомендованих профілів;
  - методи для читання даних користувачів, необхідних для надання рекомендацій;
  - методи для перетворення даних користувачів на вектори ознак;
  - виклики методів рекомендаційної системи;
  - методи для читання даних рекомендованих користувачів, необхідних для їх відображення на клієнті.
- На клієнтській частині додатку реалізовано:
  - звернення до кінцевої точки, на сервері, для отримання списку рекомендованих профілів;

- відображення рекомендованих користувачів у вигляді карток з іменами користувачів, аватаром, повідомленням про причину рекомендації та можливістю додати у друзі або приховати.

Щоб оптимізувати роботу рекомендаційної системи та уникнути непотрібних обчислень коефіцієнтів подібності, на вхід до фільтрацій подаються не всі користувачі соцмережі, а лише ті, у яких є щось спільне з поточним користувачем. Перед контентною фільтрацією з БД вибираються лише користувачі, з якими у поточного користувача є спільні профільні дані або інтереси. Перед колаборативною фільтрацією з БД вибираються ті користувачі, з якими у поточного користувача є хоча б один спільно уподобаний допис. При чому враховуються уподобання лише за останні 30 днів, що не тільки зменшує кількість даних, які потрібно проаналізувати, а і допомагає підлаштовуватись під зміни в уподобаннях користувачів. Перед топологічною фільтрацією з бази даних вибираються лише ті користувачі, з якими у поточного користувача є хоча б один спільний друг.

Коли користувач заходить на сторінку профілю вперше, запускається процес створення списку рекомендованих профілів. Далі список записується у відповідну таблицю бази даних. Також у окрему таблицю записується дата створення/оновлення списку та прапорець, який показує чи відбувається оновлення у даний момент. Після чого список рекомендацій відображається у блоці рекомендацій. Якщо користувач заходить на сторінку профілю знову, то йому буде відображатися уже сформований список рекомендацій з бази даних. Якщо список рекомендацій не оновлювався більше доби і не оновлюється зараз, система запустить оновлення списку рекомендацій асинхронно.

У результаті впровадження рекомендаційної системи, у профілі користувача відображається блок з рекомендованими профілями, як наведено на рисунку 4.6.

У кожній картки у блоці, є контекстне меню з пунктами: додати у друзі та приховати, як показано на рисунку 4.7.



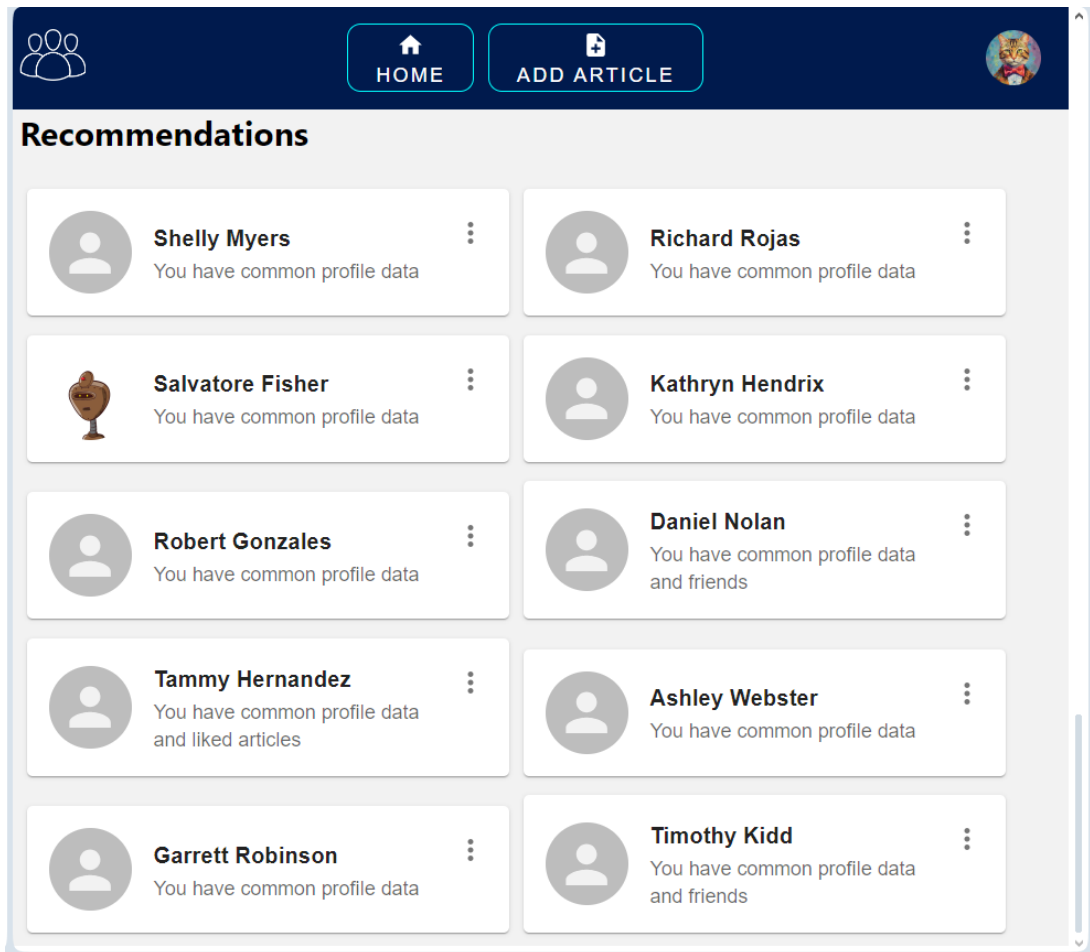


Рисунок 4.7 – блок з рекомендованими профілями  
*Джерело: побудовано автором (знімок з екрану)*

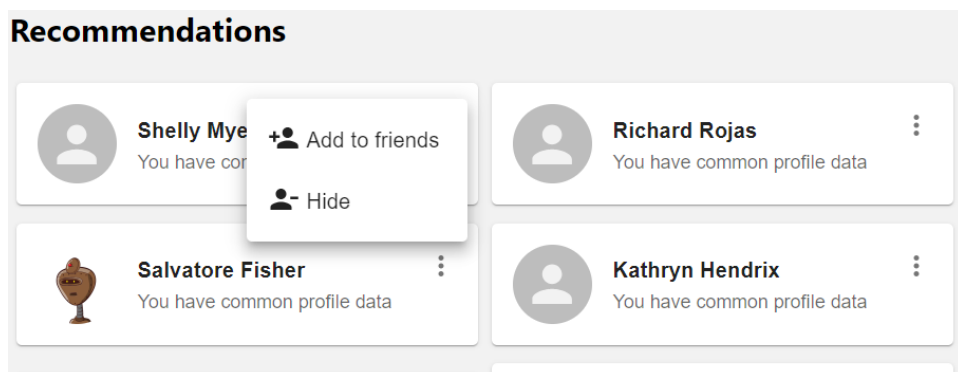


Рисунок 4.8 – контекстне меню  
*Джерело: побудовано автором (знімок з екрану)*

## 4.4 Тестування рекомендаційної системи

Щоб перевірити результати роботи рекомендаційної системи, потрібно порівняти профіль поточного авторизованого користувача з профілями користувачів, яких було рекомендовано. На рисунку 4.9 зображено профіль поточного користувача.

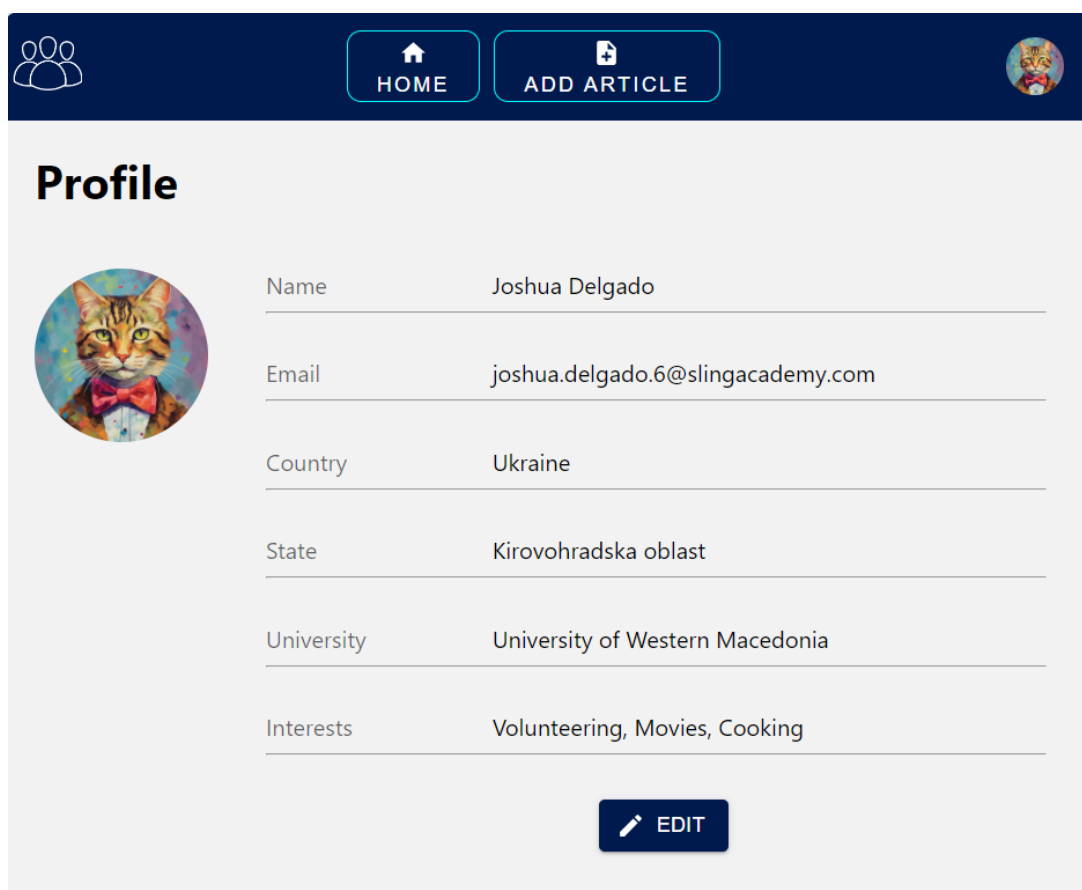


Рисунок 4.9 – профіль поточного користувача  
*Джерело:* побудовано автором (знімок з екрану)

На рисунку 4.10 наведено профіль користувача, якого було рекомендовано поточному користувачу (рисунок 4.9).

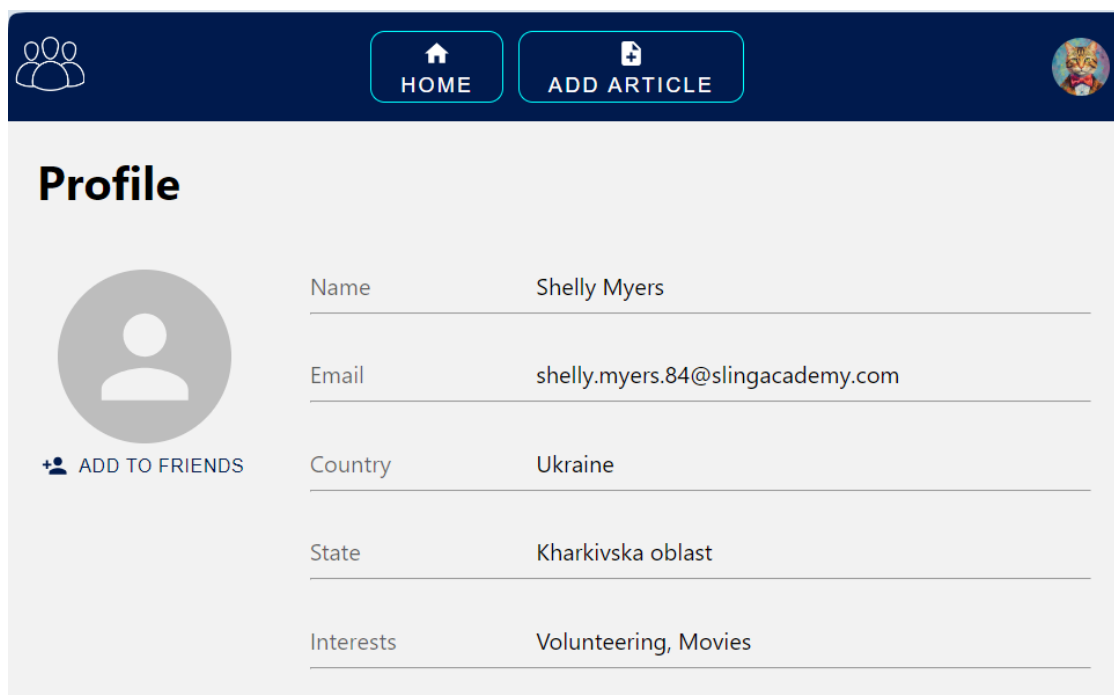


Рисунок 4.10 – профіль користувача з блоку рекомендацій  
*Джерело: побудовано автором (знімок з екрану)*

Отже, користувач, якого було рекомендовано має спільну країну проживання та два спільних інтереси з поточним користувачем.

На рисунку 4.11 зображено список друзів поточного користувача.

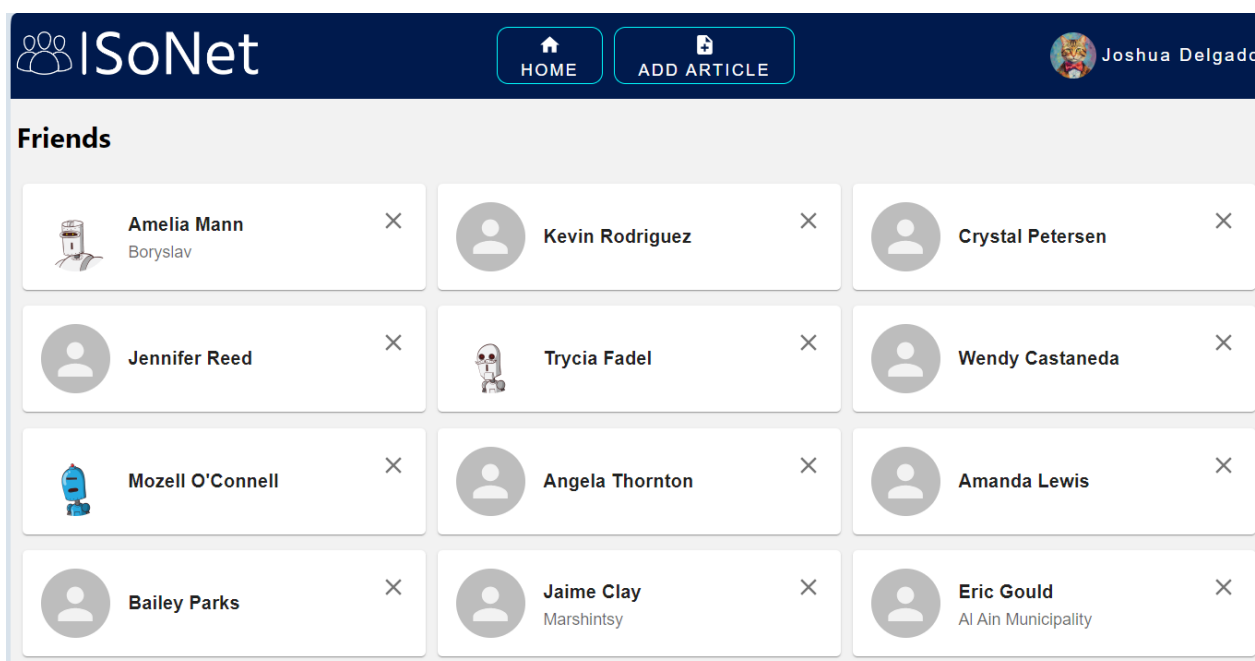


Рисунок 4.11 – список друзів поточного користувача  
*Джерело: побудовано автором (знімок з екрану)*

На рисунку 4.12 зображено профіль одного з рекомендованих користувачів.

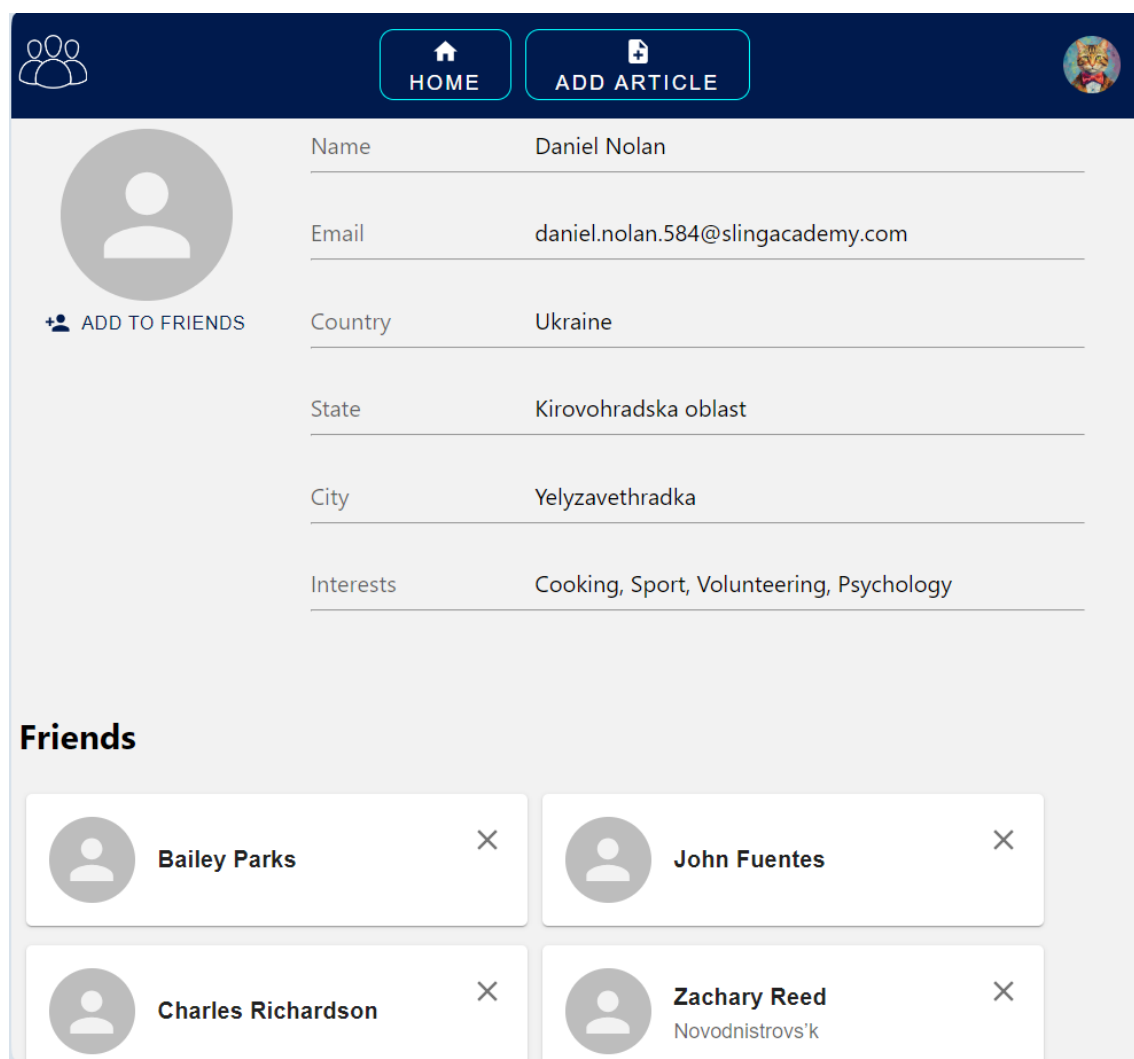


Рисунок 4.12 – профіль та список друзів одного з рекомендованих користувачів

*Джерело: побудовано автором (знімок з екрану)*

Отже, рекомендований користувач, з поточним користувачем, має спільні країну та область проживання, два спільні інтереси та одного спільного друга.

Було проаналізовано дані, всіх інших, рекомендованих користувачів. В результаті виявлено, що у всіх наведених користувачів у профілі вказано ту ж саму країну проживання що і у поточного користувача. А також у кожного з рекомендованих користувачів є б один спільний інтерес із поточним користувачем. Деякі з рекомендованих користувачів мають спільну область проживання, університет, спільного друга чи спільно уподобаний допис.

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи магістра було розроблено інформаційну технологію надання рекомендацій профілів за спільними інтересами для соціальної мережі за допомогою мови JavaScript. Для цього було проведено дослідження предметної області та проаналізовано основні типи рекомендаційних систем. Також було здійснено аналіз рекомендаційних систем, які застосовуються у популярних соціальних мережах.

Для розроблюваної рекомендаційної системи було обрано гібридний тип фільтрації, який включає три типи фільтрації: контентну, колаборативну, топологічну. Для гібридизації обрано стратегію зваженої суми, де для кожної фільтрації вага дорівнює одиниці. Далі було проведено порівняльний аналіз коефіцієнтів подоби, які використовуються для розрахунку подібності профілів. В результаті аналізу, для реалізації кожного типу фільтрації у розроблюваній рекомендаційній системі, було обрано коефіцієнт Жаккара.

На наступному етапі виконання роботи було здійснено проектування системи. У рамках проектування було розроблено контекстну IDEF0 діаграму та діаграму варіантів використання. Також наведено частину фізичної моделі бази даних соцмережі, з таблицями, необхідними для роботи рекомендаційної системи.

До заздалегідь розробленої соціальної мережі було додано до профілю користувача додаткові поля, необхідні для роботи рекомендаційної системи. Далі було розроблено рекомендаційну систему у вигляді програмного модуля. Програмний модуль впроваджено у соціальну мережу, та реалізовано виведення списку рекомендованих профілів у профілі користувача.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A graph-based friend recommendation system using Genetic Algorithm [Електронний ресурс] / N. B. Silva, I. Tsang, G. D.C. Cavalcanti, I. Tsang. // IEEE. – 2010. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/5586144>
2. Reddy V. S. An Efficient and Improved Algorithm for a Recommender System to Detect & Recognize Communities in Social Networks [Електронний ресурс] / V. S. Reddy, K. K. Kumar // International Journal on Recent and Innovation Trends in Computing and Communication. – 2023. – №11 (9s). С. 675-679. – Режим доступу до ресурсу: <https://doi.org/10.17762/ijritcc.v11i9s.7481>
3. Friend recommendation in social networks based on multi-source information fusion [Електронний ресурс] / [Cheng, S., Zhang, B., Zou, G. та ін.]. // International Journal of Machine Learning and Cybernetics. – 2019. – №10. – С. 1003-1024. – Режим доступу до ресурсу: <https://doi.org/10.1007/s13042-017-0778-1>
4. Isinkaye F. O. Recommendation systems: Principles, methods and evaluation [Електронний ресурс] / F. O. Isinkaye, Y. O. Folajimi, B. A. Ojokoh. // Egyptian Informatics Journal. – 2015. – №16. – С. 261-273. – Режим доступу до ресурсу: [\(PDF\) Recommendation systems: Principles, methods and evaluation \(researchgate.net\)](#)
5. Friend Recommendations with Self-Rescaling Graph Neural Networks [Електронний ресурс] / [X. Song, J. Lian, H. Huang та ін.]. // ACM SIGKDD Conference on Knowledge Discovery and Data Mining. – 2022. – С. 3909–3919. – Режим доступу до ресурсу: <https://dl.acm.org/doi/10.1145/3534678.3539192>
6. A Friend Recommendation System Using Users' Information of Total Attributes [Електронний ресурс] / Z.Zhang, Y. Liu, W. Ding, W. Huang. // Proceedings of the Second International Conference on Data Science. – 2015. – №9208. – С. 34–41. – Режим доступу: <https://www.cs.umb.edu/~ding/papers/ICDS2015.pdf>

7. Applications of Link Prediction in Social Networks: A Review [Електронний ресурс] / [N. N. Daud, S. H. A. Hamid, M. Saadoon та ін.]. // Journal of Network and Computer Applications. – 2020. – Режим доступу до ресурсу: <https://doi.org/10.1016/j.jnca.2020.102716>
8. Walter F. A model of a trust-based recommendation system on a social network / F. Walter, S. Battiston, F. Schweitzer. // Autonomous Agents and Multi-Agent Systems. – 2008. – №16. – С. 57–74.
9. Li M. An Efficient Adaptive Attention Neural Network for Social Recommendation [Електронний ресурс] / M. Li, K. Tei, Y. Fukazawa. // IEEE Access. – 2020. – №8. – С. 63595-63606. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/9050750>
10. Парфененко Ю.В. Рекомендаційна інформаційна система для пошуку відеоматеріалів [Електронний ресурс] / Ю. В. Парфененко, А. А. Ковтун, А. А. Вербицька // Вісник КрНУ імені Михайла Остроградського. – 2019. – №5 (118). – С. 97-102. – Режим доступу до ресурсу: [http://visnikkrnu.kdu.edu.ua/statti/2019\\_5\\_2019-5-97-102.pdf](http://visnikkrnu.kdu.edu.ua/statti/2019_5_2019-5-97-102.pdf)
11. Abbaschian B. A REVIEW OF HYBRID RECOMMENDER SYSTEMS [Електронний ресурс] / B. Abbaschian, S. Khorshidi. // Journal of Interdisciplinary Research. – 2017. – С. 259–263. – Режим доступу до ресурсу: [https://www.magnanimitas.cz/ADALTA/0702/papers/I\\_khorshidi.pdf](https://www.magnanimitas.cz/ADALTA/0702/papers/I_khorshidi.pdf)
12. Мелешко Є. В. Дослідження методів побудови рекомендаційних систем в мережі інтернет [Електронний ресурс] / Є. В. Мелешко, С. Г. Семенов, В. Д. Хох // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2018. – Т. 1 (47). – С. 131-136. – Режим доступу до ресурсу: <https://doi.org/10.26906/SUNZ.2018.1.131>
13. Мелешко Є. В. Методологія забезпечення стійкості рекомендаційних систем до дестабілізуючих факторів у комп'ютерних мережах : дис. докт. техн. наук [Електронний ресурс] / Мелешко Є. В. – Черкаси, 2021. – 323 с. – Режим доступу до ресурсу: <https://er.chdtu.edu.ua/handle/ChSTU/1781>

14. Chen S. Social Network Based Recommendation Systems: A Short Survey [Електронний ресурс] / S. Chen, S. Owusu, L. Zhou. // International Conference on Social Computing. – 2013. – С. 882–885. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/abstract/document/6693431>
15. Samad A. Structural Importance-based Link Prediction Techniques in Social Network [Електронний ресурс] / A. Samad, M. Azam, M. Qadir // EAI Endorsed Transactions on Industrial Networks and Intelligent. – 2021. – С. 1-13. – Режим доступу до ресурсу: <https://eudl.eu/doi/10.4108/eai.7-1-2021.167840>
16. Kadam S. Predicting future link in social network: A review [Електронний ресурс] / S. Kadam, R. Bhattacharya // International Journal of Health Sciences. – 2022. – 6(S2). – С. 8232–8240. – Режим доступу до ресурсу: <https://doi.org/10.1016/j.jnca.2020.102716>
17. Sathre P. Edge-Connected Jaccard Similarity for Graph Link Prediction on FPGA [Електронний ресурс] / P. Sathre, A. Gondhalekar, W. Feng // High Performance Extreme Computing Conference. – 2022. – Режим доступу до ресурсу: <https://doi.org/10.1109/HPEC55821.2022.9926326>
18. BayDNN: Friend Recommendation with Bayesian Personalized Ranking Deep Neural Network [Електронний ресурс] / [D. Ding, M. Zhang, S. Li та ін.]. // Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. – 2017. – С. 1479–1488. – Режим доступу до ресурсу: <https://dl.acm.org/doi/10.1145/3132847.3132941>
19. Мелешко Є. В. Проблеми сучасних рекомендаційних систем та методи їх рішення [Електронний ресурс] / Є. В. Мелешко // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2018. – Т. 4 (50). – С. 120-124. – Режим доступу до ресурсу: <https://doi.org/10.26906/SUNZ.2018.4.120>
20. Friend Recommendation Based on Multi-Social Graph Convolutional Network [Електронний ресурс] / [L. Chen, Y. Xie, Z. Zheng та ін.]. // IEEE Access. – 2020. – №8. – С. 43618-43629. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/abstract/document/9019700>



21. Da'u A. Recommendation system based on deep learning methods: a systematic review and new directions [Електронний ресурс] / A. Da'u, N. Salim. // Artificial Intelligence Review. – 2019. – №53. – С. 2709–2748. – Режим доступу до ресурсу: <https://link.springer.com/article/10.1007/s10462-019-09744-1>
22. Personalized Point-of-Interest Recommendation Using Improved Graph Convolutional Network in Location-Based Social Network [Електронний ресурс] / J. Liu, H. Yi, Y. Gao, R. Jing. // Electronics. – 2023. – №12(16). – С. 3495 – Режим доступу до ресурсу: <https://www.mdpi.com/2079-9292/12/16/3495>
23. Xu X. Social Recommendation Algorithm Based on Self-Supervised Hypergraph Attention [Електронний ресурс] / X. Xu, K. Przystupa, O. Kochan. // Electronics. – 2023. – №12(4). – С. 906. – Режим доступу до ресурсу: <https://www.mdpi.com/2079-9292/12/4/906>
24. Машинне навчання простими словами. Частина 1 [Електронний ресурс] – Режим доступу до ресурсу: <http://www.mmf.lnu.edu.ua/ar/1739>
25. Мета пояснила, як працює її алгоритм Instagram та рекомендації у 2023 [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.proidei.com/instagram-algorithms-3006/>
26. Pradhan A. How Facebook Suggests A Friend? [Електронний ресурс] / Arkopravo Pradhan. – 2022. – Режим доступу до ресурсу: <https://www.linkedin.com/pulse/how-facebook-suggests-friend-arkopravo-pradhan/>
27. Люди, яких ви можете знати [Електронний ресурс] – Режим доступу до ресурсу: <https://www.facebook.com/help/336320879782850>
28. Facebook People You May Know AI system [Електронний ресурс] – Режим доступу до ресурсу: <https://transparency.fb.com/features/explaining-ranking/fb-people-you-may-know/>
29. Instagram Suggested Accounts AI system [Електронний ресурс] – Режим доступу до ресурсу: <https://transparency.fb.com/features/explaining-ranking/ig-suggested-accounts/>

- 30.the-algorithm [Електронний ресурс] – Режим доступу до ресурсу:  
<https://github.com/twitter/the-algorithm>
- 31.Account Recommendations [Електронний ресурс] – Режим доступу до ресурсу:  
<https://help.twitter.com/en/resources/recommender-systems/account-recommendations>
- 32.TikTok Suggested Accounts: TikTok’s Account Recommendation Explained [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.highsocial.com/resources/tiktok-suggested-accounts-tiktoks-account-recommendation-explained/>
- 33.Альперт С. І. Основні міри подібності та нові підходи до їх застосування при класифікуванні гіперспектральних космічних зображень [Електронний ресурс] / С. І. Альперт. // Математичні машини і системи. – 2019. – №1. – С. 143–151. – Режим доступу до ресурсу:  
[http://www.immsp.kiev.ua/publications/articles/2019/2019\\_1/01\\_2019\\_Alper.pdf](http://www.immsp.kiev.ua/publications/articles/2019/2019_1/01_2019_Alper.pdf)
- 34.Amer A. A. Combinations of Jaccard with Numerical Measures for Collaborative Filtering Enhancement: Current Work and Future Proposal [Електронний ресурс] / A. A. Amer, L. Nguyen. – 2021. – Режим доступу до ресурсу:  
<https://doi.org/10.48550/arXiv.2111.12202>
- 35.Lee S. Applying Different Similarity Measures based on Jaccard Index in Collaborative Filtering / S. Lee // Journal of The Korea Society of Computer and Information. – 2021. – № 26(5). – С. 47-53.
- 36.Taleb A. JKRW Link Prediction – A New Ensemble Technique Based on Merging Other Known Techniques in The Social Network Analysis [Електронний ресурс] / A. Taleb, R. M. H. Al-Sayyed, H. S. Al-Bdour // International Journal of Interactive Mobile Technologies. – 2021. – №15(12). – С. 125-139. – Режим доступу до ресурсу: <https://doi.org/10.3991/ijim.v15i12.22831>
- 37.Bag S. An efficient recommendation generation using relevant Jaccard similarity / S. Bag, S. K. Kumar, M. K. Tiwari // Information Sciences. – 2019. – №483. – С. 53-64. – Режим доступу до ресурсу: <https://doi.org/10.1016/j.ins.2019.01.023>

38. Hybrid Collaborative Filtering Using Social and Semantic Suggestions for Friend Recommendation [Электронный ресурс] / [M. T. Ramakrishna, V. K. Venkatesan, R. Bhardwaj та ін.]. // Electronics. – 2023. – №12(6) – Режим доступу до ресурсу: <https://www.mdpi.com/2079-9292/12/6/1365>
39. Adamic L. A. Friends and neighbors on the Web [Электронный ресурс] / L. A. Adamic, E. Adar. – 2003. – №25(3). – С. 211-230 – Режим доступу до ресурсу: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ec6c9d160ffd83e4beb016608e7a8bfbac4c321a>
40. A Personalized Explainable Learner Implicit Friend Recommendation Method [Электронный ресурс] / [C. Li, B. Zhou, W. Lin та ін.]. // Data Science and Engineering. – 2023. – №8 – С. 23-35. – Режим доступу до ресурсу: <https://link.springer.com/article/10.1007/s41019-023-00204-z>
41. Functional Modelling and IDEF0 to Enhance and Support Process Tailoring in Systems Engineering [Электронный ресурс] / G. Manenti, M. Ebrahimi-arjestan, L. Yang, M. Yu. // IEEE International Systems Engineering Symposium (ISSE). – 2019. – Режим доступу до ресурсу: <http://dx.doi.org/10.1109/ISSE46696.2019.8984539>
42. DummyJSON [Электронный ресурс] – Режим доступу до ресурсу: <https://dummyjson.com>
43. Free Mock API for Testing and Learning [Электронный ресурс] – Режим доступу до ресурсу: <https://www.slingacademy.com/series/mock-api-for-testing-and-learning/>

## ДОДАТОК А

### ПЛАНУВАННЯ РОБІТ

**Деталізація мети проекту методом SMART.** Продуктом дипломного проекту є інформаційна технологія надання рекомендацій профілів за спільними інтересами для соціальної мережі.

Результати деталізації методом SMART розміщені у табл. А.1.

Таблиця А.1 – Деталізація мети методом SMART

Specific (конкретна)	Розробити систему рекомендації профілів для соціальної мережі, яка надає рекомендації на основі профільних даних користувача, його уподобань та списку друзів.
Measurable (вимірювана)	Розробити рекомендаційну систему, яка рекомендує користувачу 10 профілів, які є найбільш подібними до нього (тобто з найбільшим значенням коефіцієнту подібності).
Achievable (досяжна)	Розробити, мовою JavaScript, систему рекомендації профілів для соціальної мережі, яка розраховує подібність між користувачами на основі коефіцієнта подібності.
Relevant (реалістична)	Розробити рекомендаційну систему, яка допоможе користувачам подолати проблему перевантаження інформацією.
Time-framed (обмежена у часі)	Розробити систему рекомендації профілів для соціальної мережі, дотримуючись розробленого календарного графіку.

**Планування змісту структури робіт IT-проекту (WBS).** Структура декомпозиції робіт (work breakdown structure, WBS) це інструмент, який застосовується у проектному менеджменті, що допомагає розкласти проект на окремі

фрагменти (роботи), щоб легше керувати ними і визначити їх послідовність виконання. WBS дозволяє зрозуміти структуру проекту, та визначити обсяг робіт, які необхідно виконати для досягнення мети проекту. Окремі роботи, виділені за допомогою WBS розподіляють між членами команди, що допомагає ефективно організувати командну роботу над проектом. WBS є основою для створення календарного графіку проекту.

На рисунку А.1 наведено WBS діаграму даного проекту.

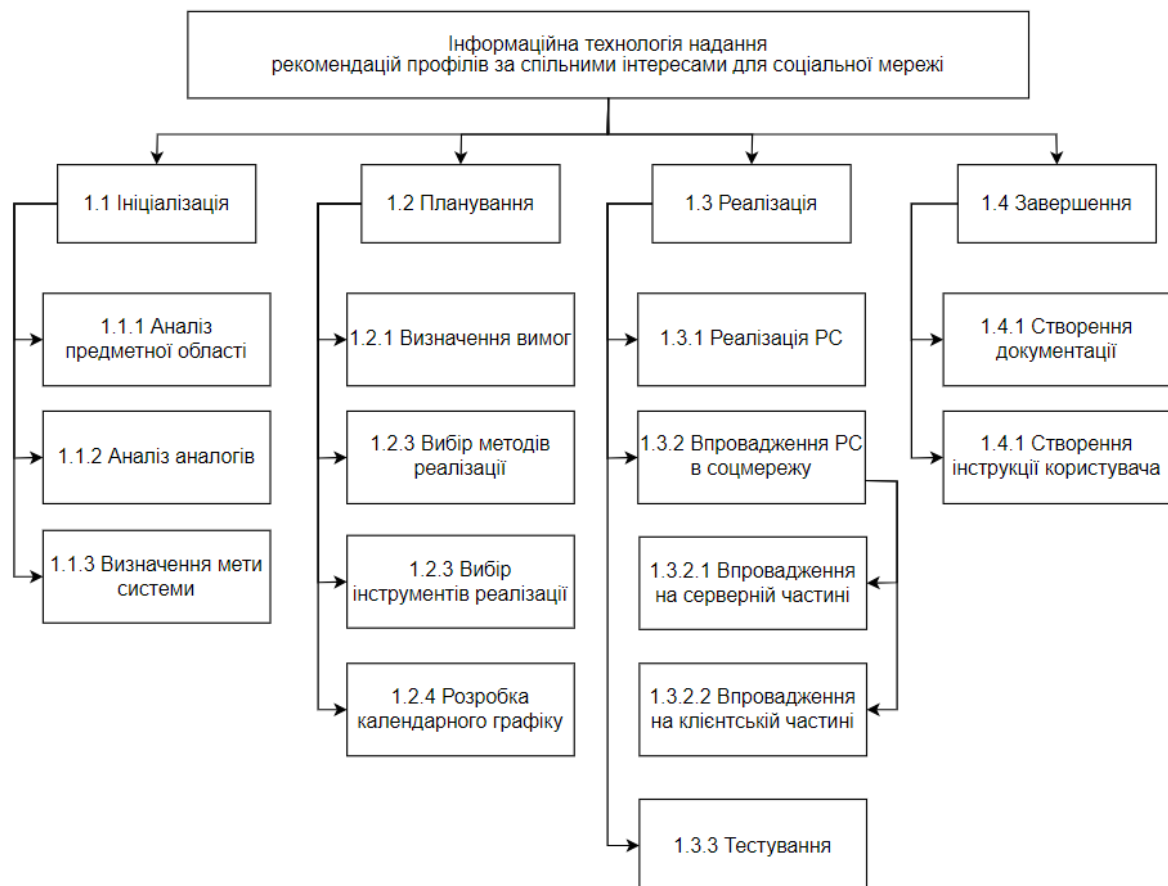


Рисунок А.1 – таблиця WBS (work breakdown structure)

**Організаційна структура проекту (OBS).** OBS (Organizational Breakdown Structure) – це інструмент, який використовується при управлінні проектом, для відображення ієрархічної структури організації, в якій виконується проект. OBS використовується для розподілу робіт, виділених за допомогою WBS, між членами команди. OBS має таку саму структуру як і WBS, але замість назв робіт, показує імена,

відповідальних за роботу, осіб. OBS важлива для чіткого розподілу ролей та відповідальності, покращення комунікації та координації між учасниками проекту або внутрішніми підрозділами організації.

На рисунку А.2 наведено OBS діаграму даного проекту.

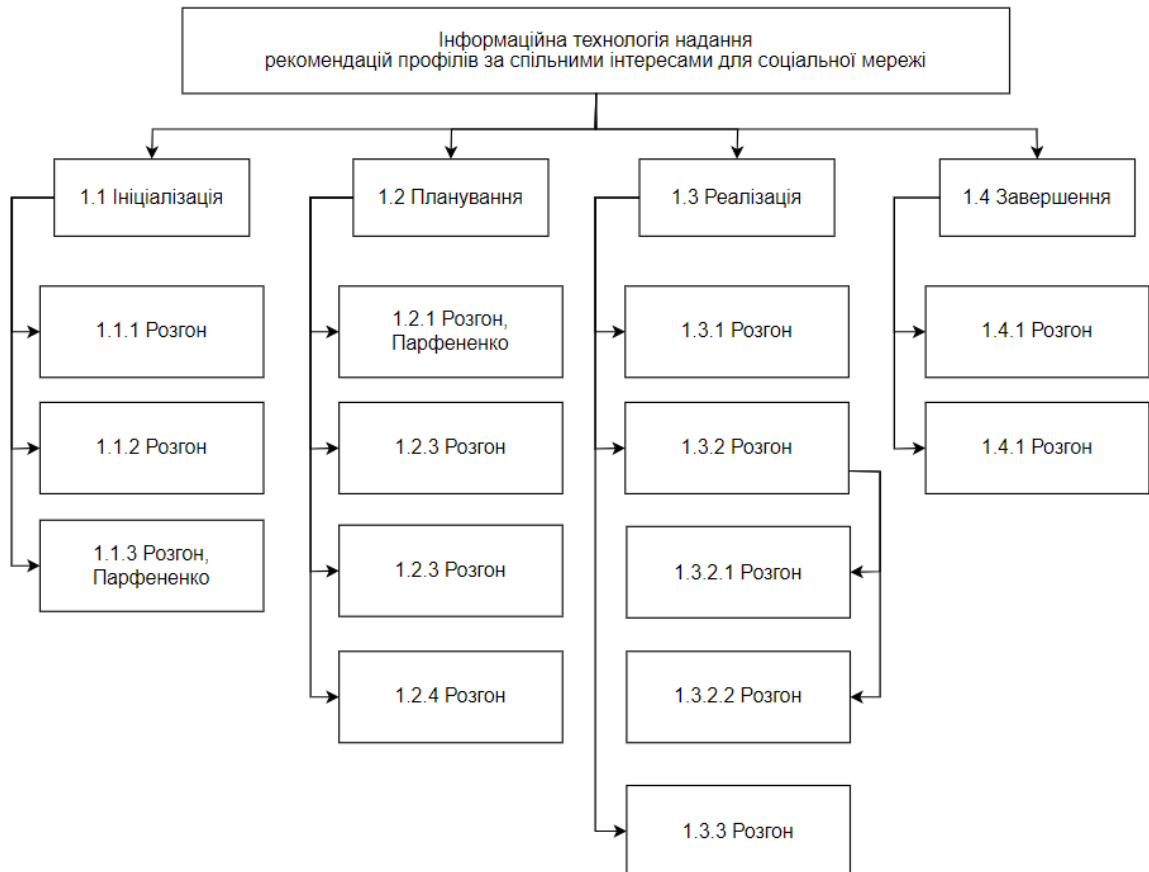


Рисунок А.2 – таблиця OBS (Organization Breakdown Structure)

**Побудова календарного графіка виконання ІТ-проекту.** Календарний графік складають для того, щоб окреслити терміни виконання робіт, для реалізації проекту, враховуючи можливості, ресурси та обмеження. Календарний графік створюють на основі робіт, виділених за допомогою WBS. Для того, щоб створити та візуалізувати календарний графік використовують діаграму Ганта. Вона складається з горизонтальних смуг, кожна з яких відображає тривалість однієї з робіт проекту. Тобто діаграма Ганта допомагає відстежувати порядок виконання робіт та стан проекту вцілому. За допомогою сервісу ClickUp Gantt побудовано діаграму Ганта (рис. А.3).

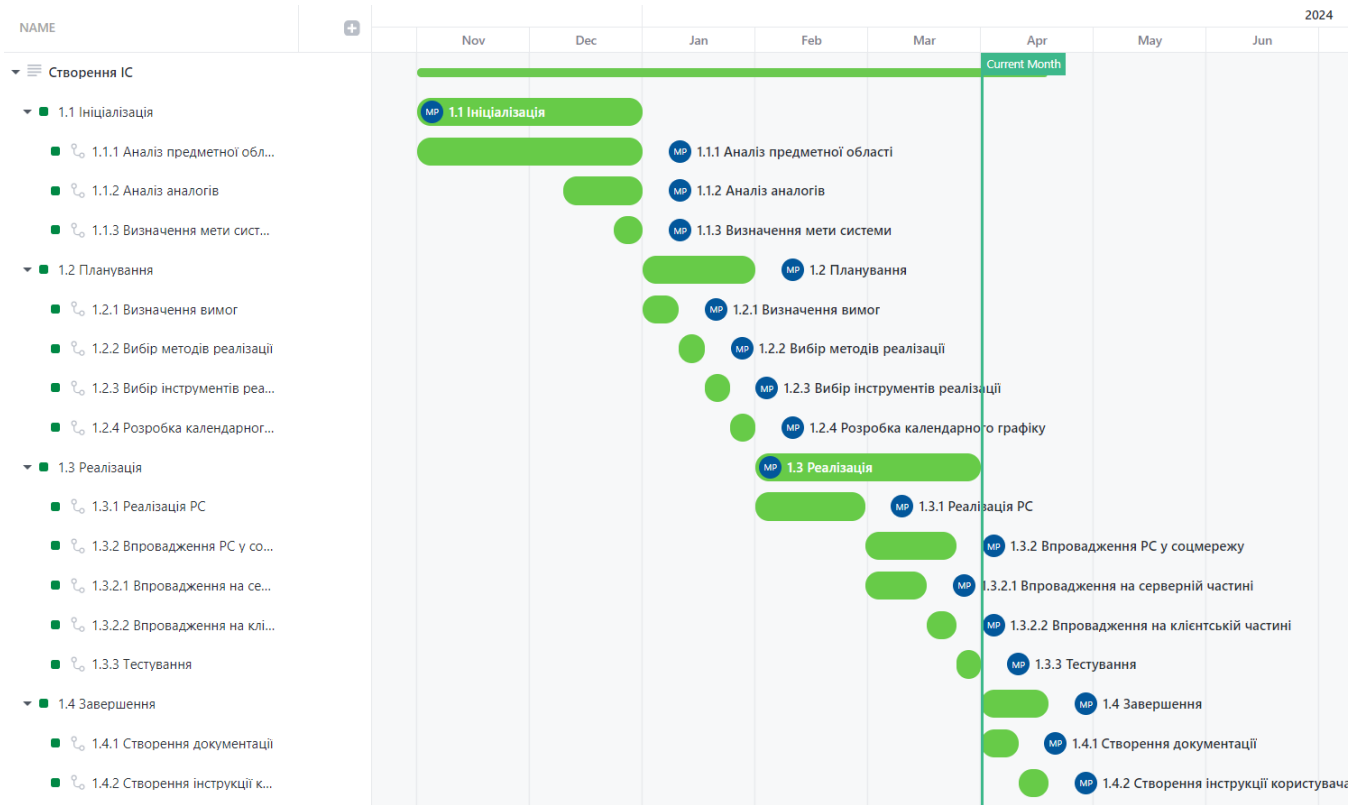


Рисунок А.3 – Діаграма Ганта

**Управління ризиками.** До основних ризиків розробки системи рекомендації профілів для соціальної мережі є:

- небажання користувача заповнювати профіль;
- не активність користувача у соціальній мережі;
- недовіра користувача до рекомендацій;
- порушення приватності користувача;
- відмова обладнання;
- перебої електропостачання;
- зміна строків виконання роботи;
- зміна цілей у ході реалізації проекту;
- зростання вимог до проекту.

Таблиця А.2. Ймовірність виникнення і величина ризику

№	Ризики	Виникнення	Втрати
1	Небажання користувача заповнювати профіль	3	3
2	Не активність користувача у соціальній мережі	3	3
3	Недовіра користувача до рекомендацій	2	5
4	Порушення приватності користувача	3	5
5	Відмова обладнання	2	4
6	Перебої електропостачання	2	4
7	Зміна строків виконання роботи	2	4
8	Зміна цілей у ході реалізації проекту	4	4
9	Зростання вимог до проекту	3	3



## Матриця «Ймовірність – Втрати»

Ймовірність	-	<b>5</b>	<b>5</b>	<b>10</b>	<b>15</b>	<b>20</b>	<b>25</b>
	Зміна цілей у ході реалізації проекту	<b>4</b>	<b>4</b>	<b>8</b>	<b>12</b>	<b>16</b>	<b>20</b>
	Зростання вимог до проекту Небажання користувача заповнювати профіль Не активність користувача у соціальній мережі Порушення приватності користувача	<b>3</b>	<b>3</b>	<b>6</b>	<b>9</b>	<b>12</b>	<b>15</b>
	Недовіра користувача до рекомендацій Відмова обладнання Перебої електропостачання Зміна строків виконання роботи	<b>2</b>	<b>2</b>	<b>4</b>	<b>6</b>	<b>8</b>	<b>10</b>
	-	<b>1</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
			<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>

Небажання користувача заповнювати профіль  
Не активність користувача у соціальній мережі  
Зростання вимог до проекту

Відмова обладнання  
Перебої електропостачання  
Зміна строків виконання роботи  
Зміна цілей у ході реалізації проекту

Недовіра користувача до рекомендацій  
Порушення приватності користувача

## Втрати

Аналізуючи ризики за ймовірністю їх виникнення, можемо їх розділити на:

- Ігноровані
  - Відсутні
- Незначні
  - Недовіра користувача до рекомендацій
  - Відмова обладнання
  - Перебої електропостачання
  - Зміна строків виконання роботи
- Помірні

- Небажання користувача заповнювати профіль
- Не активність користувача у соціальній мережі
- Порухення приватності користувача
- Зростання вимог до проекту
- Істотні
  - Зміна цілей у ході реалізації проекту
- Критичні
  - Відсутні

Класифікація ризиків за рівнем впливу:

- Прийнятні
  - Відсутні
- Виправдані
  - Небажання користувача заповнювати профіль
  - Не активність користувача у соціальній мережі
  - Недовіра користувача до рекомендацій
  - Відмова обладнання
  - Перебої електропостачання
  - Зміна строків виконання роботи
  - Зростання вимог до проекту
- Неприпустимі
  - Зміна цілей у ході реалізації проекту
  - Порухення приватності користувача.

## ДОДАТОК Б

Фрагменти коду реалізації інформаційної технології.

`similarityIndexes.js`:

```

module.exports = {
  adamicAdar: (graph, node1, node2) => {
    const neighbors1 = new Set(graph[node1]);
    const neighbors2 = new Set(graph[node2]);

    const commonNeighbors = new Set([...neighbors1].filter(x =>
neighbors2.has(x)));

    let adamicAdar = 0;
    for (const commonNeighbor of commonNeighbors) {
      const degree = graph[commonNeighbor].length;
      adamicAdar += 1 / Math.log(degree);
    }
    return adamicAdar;
  },

  cosine: (v1, v2) => {
    const dotProduct = v1.reduce((acc, el, index) => acc + el *
v2[index], 0);

    const getNorm = (v) => Math.sqrt(v.reduce((acc, el) => acc + el
* el, 0));

    const norm1 = getNorm(v1);
    const norm2 = getNorm(v2);

    return dotProduct / (norm1 * norm2);
  },

  cosineSalton: (s1, s2) => {
    const intersection = new Set([...s1].filter(x => s2.has(x)));
    const denominator = Math.sqrt(s1.size * s2.size);

    return denominator === 0 ? 0 : intersection.size / denominator;
  },

  jaccardVector: (v1, v2, specialValue = 0) => {
    if (v1.length !== v2.length) {
      throw new Error("Vectors length mismatch");
    }
  }
}

```

```

    }
    let equals = 0;
    let notEquals = 0;
    for (let i = 0; i < v1.length; i++) {
        if (v1[i] !== specialValue && v2[i] !== specialValue && v1[i]
=== v2[i]) {
            equals++;
        } else {
            notEquals++;
        }
    }

    return equals / (v1.length + notEquals);
},

jaccardSet: (s1, s2) => {
    if (s1.size === 0 || s2.size === 0) {
        return 0;
    }
    const intersection = new Set([...s1].filter(x => s2.has(x)));
    const union = new Set([...s1, ...s2]);

    return intersection.size / union.size;
}
};

```

### baseFilter.js:

```

module.exports = (userFeatures, usersFeatures, similarityFn,
source) => {

    return usersFeatures.map(u => {
        const score = similarityFn(userFeatures, u.features);
        return {user_id: u.user_id, score, source};
    });

};

```

### collaborativeFilter.js:

```

const baseFilter = require("../baseFilter");
const similarityIndexes = require("../similarityIndexes");

const source = "collaborative";

module.exports = {

```

```

    cosine: (userFeatures, usersFeatures) => {
      return      baseFilter(userFeatures,      usersFeatures,
similarityIndexes.cosineSalton, source);
    },
    jaccard: (userFeatures, usersFeatures) => {
      return      baseFilter(userFeatures,      usersFeatures,
similarityIndexes.jaccardSet, source);
    }
  }
}

```

### **contentFilter.js:**

```

const similarityIndexes = require("../similarityIndexes");

const source = "content";

module.exports = {
  jaccard: (userFeatures,      userInterests,
usersFeaturesAndInterests) => {

    return usersFeaturesAndInterests.map(ufi => {
      const s1 = similarityIndexes.jaccardVector(userFeatures,
ufi.features);
      const s2 = similarityIndexes.jaccardSet(userInterests,
ufi.interests);
      const score = (s1 + s2)/2;
      return {user_id: ufi.user_id, score, source};
    });

  },

  cosine: (userFeatures, userInterests, usersFeaturesAndInterests)
=> {

    return usersFeaturesAndInterests.map(ufi => {
      const s1 = similarityIndexes.cosine(userFeatures,
ufi.features);
      const s2 = similarityIndexes.cosineSalton(userInterests,
ufi.interests);
      const score = (s1 + s2)/2;
      return {user_id: ufi.user_id, score, source};
    });

  },
};

```

**topologyFilter.js:**

```

const baseFilter = require("../baseFilter");
const similarityIndexes = require("../similarityIndexes");

const source = "topology";

module.exports = {
  jaccard: (userFeatures, usersFeatures) => {
    return baseFilter(userFeatures, usersFeatures,
similarityIndexes.jaccardSet, source);
  },
  adamicAdar: (graph, users, id) => {
    return users.map(u => {
      const score = similarityIndexes.adamicAdar(graph, id, u);
      return {user_id: u, score, source}
    })
  }
};

```

**hybridFilter.js:**

```

module.exports = (results, weights) => {
  const userScores = {};
  const userSources = {};

  results.forEach((result, index) => {
    const weight = weights[index];
    result.forEach((obj) => {
      const {user_id, score, source} = obj;
      userScores[user_id] = (userScores[user_id] || 0) + weight
* score;
      const prevSources = (userSources[user_id] || []);
      userSources[user_id] = Array.isArray(source) ?
[...prevSources, ...source] : [...prevSources, source];
    });
  });

  return Object.keys(userScores).map(user_id => ({
    user_id,
    score: userScores[user_id],
    source: userSources[user_id]
  }));
};

```

**getReasonMessage.js:**

```

const defaultReasons = {
  content: "profile data",
  collaborative: "liked articles",
  topology: "friends"
}

module.exports = (source, reasons = defaultReasons) => {
  const startOfMessage = "You have common";
  if (Array.isArray(source)) {
    const sources = Array.from(new Set(source));
    if (sources.length === 1) {
      return `${startOfMessage} ${reasons[sources[0]]}`;
    } else if (sources.length === 2) {
      return `${startOfMessage} ${reasons[sources[0]]} and
${reasons[sources[1]]}`;
    } else if (sources.length > 2) {
      const common = sources.slice(0, sources.length - 1).map(s =>
reasons[s]).join(', ');
      const last = reasons[sources[sources.length - 1]];
      return `${startOfMessage} ${common}, and ${last}`;
    }
  } else {
    return `${startOfMessage} ${reasons[source]}`;
  }
}

```

**getRecommendedIds.js:**

```

const getReasonMessage = require("./getReasonMessage");

module.exports = (similarities, recommendationsLength) => {
  if (similarities.length === 0) {
    return [];
  }
  similarities.sort((a, b) => b.score - a.score);
  const recommendedIds = [];
  const sources = {};
  const end = similarities.length < recommendationsLength ?
similarities.length : recommendationsLength;
  for (let i = 0; i < end; i++) {
    recommendedIds.push(similarities[i].user_id);
    sources[similarities[i].user_id] = similarities[i].source;
  }
  const reasons = {};
  for (const user_id in sources) {

```

```

    reasons[user_id] = getReasonMessage(sources[user_id]);
  }
  return {recommendedIds, reasons};
}

```

### performCollaborativeFilter.js:

```

const likesService = require("../likes");
const usersService = require("../users");
const collaborativeFilter = require("../rs/collaborativeFilter");

const runFilter = async (id, filterFn) => {
  const userLikes = await likesService.getByUserId(id);
  if (userLikes.length === 0) {
    return [];
  }

  const usersIds = await usersService.getForCollaborativeFiltering(id);
  if (usersIds.length === 0) {
    return [];
  }

  const userLikesSet = new Set(userLikes.map(l => l.article_id));

  const usersLikes = await Promise.all(
    usersIds.map(async (user) => {
      const likes = await likesService.getByUserId(user.user_id);
      const likesSet = new Set(likes.map(l => l.article_id));
      return {...user, features: likesSet};
    })
  );

  return filterFn(userLikesSet, usersLikes);
}

module.exports = {
  jaccard: async (id) => {
    return runFilter(id, collaborativeFilter.jaccard)
  },
  cosine: async (id) => {
    return runFilter(id, collaborativeFilter.cosine);
  }
};

```



## ДОДАТОК В

Копія публікації тез.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
МІЖНАРОДНИЙ УНІВЕРСИТЕТ «АСТАНА»

ІНФОРМАТИКА, МАТЕМАТИКА,  
АВТОМАТИКА

**ІМА :: 2024**

**МАТЕРІАЛИ  
та програма**

МІЖНАРОДНОЇ НАУКОВОЇ КОНФЕРЕНЦІЇ  
молодих учених

(Суми – Астана, 22–26 квітня 2024 року)

Суми  
Сумський державний університет  
2024

### **Інформаційна система надання рекомендацій профілів за спільними інтересами для соціальної мережі**

Розгон М. О., студент ІТ.м-21н; Парфененко Ю. В., доцент

Сумський державний університет, м. Суми, Україна

Соціальні мережі стали основною платформою для розповсюдження різномірної інформації. Кількість користувачів соцмереж постійно зростає. Щоб знайти підходящу для знайомства людину, потрібно переглянути та проаналізувати безліч профілів, що є дуже виснажливим. Це називають проблемою перевантаження інформацією. Системи рекомендації профілів допомагають подолати цю проблему. Вони аналізують дані користувача та його дії, щоб надати персоналізований список профілів, рекомендованих для знайомства чи дружби. Крім того системи рекомендації профілів сприяють розвитку соцмережі.

Метою розробленої рекомендаційної системи, є надання користувачу соціальної мережі персоналізованого списку рекомендованих профілів, на основі його інтересів.

Розроблена рекомендаційна система виконує пошук користувачів, які подібні до поточного користувача, і рекомендує профілі знайдених користувачів. Подібність між користувачами визначається на основі їх профільних даних, списку друзів та дописів які їм подобаються. Для розрахунку подібності використовується коефіцієнт Жаккара. Спочатку система розраховує коефіцієнти подібності окремо для профільних даних, списку друзів, та списку уподобаних дописів. Потім система розраховує суму коефіцієнтів для кожного кандидата. В кінці система повертає 10 профілів з найбільшою сумою коефіцієнтів.

Рекомендаційну систему розроблено у вигляді програмного модуля мовою JavaScript та впроваджено у заздалегідь розроблений прототип соціальної мережі. Прототип соцмережі розроблено мовою JavaScript з використанням фреймворків React та Express та бази даних PostgreSQL. В результаті впровадження у профілі користувача соцмережі відображається список рекомендованих профілів. Рекомендовані профілі відображаються у вигляді карток, які містять ім'я користувача, аватар та повідомлення про те, на яких підставах рекомендовано даний профіль.