

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод пошуку інформації на вебсайті університету з доповнюючим інформаційним пошуком

Галузь знань \_\_\_\_\_ 12 – Інформаційні технології \_\_\_\_\_  
Шифр і назва галузі знань  
Спеціальність \_\_\_\_\_ 122 – Комп'ютерні науки \_\_\_\_\_  
Шифр і назва спеціальності  
Освітня програма \_\_\_\_\_ Комп'ютерні науки \_\_\_\_\_  
Назва освітньої програми

Виконав: \_\_\_\_\_ студент групи КН-21-1 \_\_\_\_\_ Данило ТВЕРДОХЛІБОВ \_\_\_\_\_  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ  
Керівник: \_\_\_\_\_ к.т.н., доц. каф. КН \_\_\_\_\_ Руслан БАГРІЙ \_\_\_\_\_  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ  
Нормоконтроль: \_\_\_\_\_ к.т.н., доц. каф. КН \_\_\_\_\_ Руслан БАГРІЙ \_\_\_\_\_  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:  
Зав. кафедри КН, д.т.н., професор \_\_\_\_\_ Олександр БАРМАК \_\_\_\_\_  
Підпис Ім'я, ПРІЗВИЩЕ

09 06 2025 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор Олександр БАРМАК

« 10 » 02 2025 року

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод пошуку інформації на вебсайті університету з доповнюючим інформаційним пошуком»

2. Завдання видано студенту Данилу ТВЕРДОХЛІБОВУ  
(Ім'я, прізвище)

3. Керівник роботи доцент кафедри КН Руслан БАГРІЙ  
(посада, ім'я, прізвище)

4. Затверджено наказом університету від « 07 » 02 2025 р. № 23

5. Дата видачі завдання студенту: « 10 » 02 2025 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета кваліфікаційної роботи бакалавра – підвищення ефективності пошуку інформації на вебсайті університету шляхом застосування генеративного штучного інтелекту. У межах дослідження планується провести аналіз предметної області, здійснити огляд сучасних методів та технологій інтерактивного пошуку інформації, а також вивчити наявні програмні рішення, що застосовуються в аналогічних задачах. На основі результатів аналізу буде сформульовано постановку задачі та запропоновано метод інтерактивного пошуку, який поєднує можливості генеративних мовних моделей зі структурованими веб-даними.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи	січень 2025	Виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	лютий 2025	Виконано
3	Проектування та розроблення методу вирішення завдання, загальної архітектури програмного забезпечення, інтерфейсу користувача, вибір засобів реалізації програмного забезпечення	березень 2025	Виконано
4	Створення та тестування програмного забезпечення, дослідження ефективності, висновки з виконаної роботи	квітень 2025	Виконано
5	Написання пояснювальної записки, урахування зауважень керівника, оформлення згідно з вимогами	травень 2025	Виконано
6	Розробка презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2025	Виконано
7	Отримання відгуку керівника, рецензії, перевірка на плагіат, нормоконтроль	червень 2025	Виконано
8	Підготовка до захисту та захист кваліфікаційної роботи	червень 2025	Виконано

Виконавець: студента групи КН-21-1  
Група виконавця

  
Підпис

Данило ТВЕРДОХЛІБОВ  
Ім'я, ПРІЗВИЩЕ

Керівник: к.т.н., доц. каф. КН  
Науковий ступінь, посада

  
Підпис

Руслан БАГРІЙ  
Ім'я, ПРІЗВИЩЕ

## Анотація

Тема кваліфікаційної роботи бакалавра: «Метод пошуку інформації на вебсайті університету з доповнюючим інформаційним пошуком»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-21-1  
Данило ТВЕРДОХЛІБОВ

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН  
Руслан БАГРІЙ

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
52	24	8	42	2

Мета кваліфікаційної роботи бакалавра – підвищення ефективності пошуку інформації на вебсайті університету шляхом застосування генеративного штучного інтелекту. Розроблено метод інтерактивного пошуку інформації на вебсайті університету, що базується на генеративному штучному інтелекті. У основі методу лежить технологія RAG для пошуку та обробки даних із векторних представлень нормативних PDF-документів університету. Для генерації відповідей застосовано модель DeepSeek, а для пошуку та зберігання даних – векторну базу FAISS. Реалізацію здійснено на Python із фреймворком FastAPI

Результатом виконання кваліфікаційної роботи бакалавра є реалізований вебзастосунок, який використовує штучний інтелект для надання релевантних відповідей про навчальний процес, вступну кампанію та студентське життя в ХНУ.

Ключові слова: RAG, LLM, генеративний штучний інтелект, вебзастосунок, векторний пошук, ембеддинг, векторні представлення, векторна база даних, інтерактивний пошук, FAISS, DeepSeek.

Виконавець: студента групи КН-21-1

Група виконавця

  
Підпис

Данило Твердохлібов

Ім'я, ПРІЗВИЩЕ

## Зміст

Вступ.....	4
Розділ 1 Огляд технологій генеративного штучного інтелекту для інтерактивного пошуку інформації .....	5
1.1 Пошук та аналіз інформації в мережі Інтернет.....	5
1.2 Огляд технологій генеративного штучного інтелекту .....	6
1.3 Огляд існуючих засобів пошуку інформації на вебсайті на основі генеративного штучного інтелекту .....	9
1.4 Мета та завдання кваліфікаційної роботи .....	13
Розділ 2 Проектування методу інтерактивного пошуку інформації для вебсайту університету.....	14
2.1 Загальна ідея інтерактивного пошуку інформації на основі генеративного штучного інтелекту.....	14
2.2 Архітектура LLM моделі з пошуково-доповненою генерацією .....	15
2.3 Векторна база даних .....	20
2.4 Метод інтерактивного пошуку інформації на основі генеративного штучного інтелекту.....	22
2.5 Критерії та метрики оцінки якості методу пошуку інформації .....	27
2.6 Висновки до розділу 2 .....	30
Розділ 3. Особливості реалізації та результати тестування методу інтерактивного пошуку інформації .....	31
3.1 Особливості розробки вебзастосунку для імплементації запропонованого методу.....	31
3.1.1. Серверна частина та клієнтський інтерфейс.....	31
3.1.2 Структура вебзастосунку .....	35
3.2 Результати тестування методу пошуку інформації .....	39
3.3 Висновки до розділу 3 .....	46
Загальні висновки.....	47
Перелік посилань.....	49
Додатки	

### Перелік скорочень

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
RNN	Recurrent Neural Network
LLM	Large Language Model
AI	Artificial Intelligence
RAG	Retrieval-Augmented Generation
CRUD	Create, Read, Update, Delete
FAISS	Facebook AI Similarity Search
PQ	Product Quantization
IVF	Inverted File Index
ХНУ	Хмельницький Національний Університет
API	Application Programming Interface
LCS	Longest Common Subsequence
UI	User Interface
NLP	Natural Language Processing
CPU	Central Processing Unit
GPU	Graphics Processing Unit
DOM	Document Object Model
JSX	JavaScript XML
MoE	Mixture of Experts

## Вступ

Кваліфікаційна робота бакалавра присвячена розробці методу інтерактивного пошуку інформації на вебсайті університету на основі генеративного штучного інтелекту.

Сучасна діяльність університетів регламентується положеннями, розміщеними на вебсайтах у форматі PDF, що ускладнює швидкий пошук інформації для студентів, абітурієнтів та викладачів. Метод інтерактивного пошуку на основі генеративного штучного інтелекту покращить доступ до даних і взаємодію з вебсайтом університету. Об'єкт дослідження – процес інтерактивного пошуку інформації на вебсайті університету.

Предмет дослідження – методи генеративного штучного інтелекту та обробки природної мови для інтерактивного пошуку інформації на вебсайті університету.

Мета кваліфікаційної роботи бакалавра – підвищення ефективності пошуку інформації на вебсайті університету шляхом застосування генеративного штучного інтелекту.

Завдання кваліфікаційної роботи бакалавра:

– Дослідити існуючі методи та технології інтерактивного пошуку інформації.

– Розробити метод інтерактивного пошуку інформації на вебсайті університету з використанням генеративного штучного інтелекту.

– Реалізувати вебкомпоненти для забезпечення ефективного інтерактивного пошуку інформації.

– Оцінити ефективність методу інтерактивного пошуку інформації за допомогою тестування та метрик якості.

Результатом виконання роботи вважатиметься створення вебзастосунку, який зможе формулювати релевантні відповіді на запити користувачів.

## **Розділ 1 Огляд технологій генеративного штучного інтелекту для інтерактивного пошуку інформації**

### **1.1 Пошук та аналіз інформації в мережі Інтернет**

Процес пошуку інформації в мережі Інтернет – це багатоаспектне явище, яке поєднує в собі використання спеціалізованих інструментів та когнітивні стратегії користувача. Основу цього процесу складає застосування пошукових систем[1].

Системи пошуку на вебсайтах – це ключовий компонент інформаційних ресурсів, що забезпечують швидкий доступ користувачів до необхідних даних. Вони працюють на основі алгоритмів індексації, ранжування та обробки запитів, що дозволяє знаходити релевантну інформацію серед великого обсягу контенту.

Основним етапом функціонування пошукових систем є створення індексу вебсторінок. На основі цього працює гугл-пошук та більшість інших популярних вебсервісів. Для створення індексу застосовуються краулери, які аналізують структуру сайту, збирають текстовий вміст сторінок і зберігають його у вигляді структурованої бази даних. Під час індексації враховуються такі параметри: унікальні ключові слова, заголовки та підзаголовки, метатеги, внутрішні та зовнішні посилання [2].

Коли користувач вводить запит у пошуковий рядок, система проводить аналіз введеного тексту, застосовуючи методи розбору та нормалізації [3]-[4]. Щоб визначити, які сторінки є найбільш релевантними запиту, пошукова система використовує алгоритми ранжування, які перевіряють відповідність ключовим словам. Сторінки, що містять запитувані терміни в заголовках, тексті або метаданих після такого процесу отримують вищий пріоритет. Також пошукові системи враховують популярність сторінки та поведінкові фактори.

Пошук освітньої інформації важливий для викладачів та студентів. Основна складність такого пошуку полягає в необхідності аналізу великих обсягів інформації, а також виборі актуальних і науково підтверджених даних, що зробити без додаткових засобів стає проблематично[5].

Багато вебсайтів використовують додаткові технології для покращення роботи пошуку, зокрема кешування запитів, що зменшує час обробки популярних пошуків; автодоповнення та підказки, які спрощують введення запиту; семантичний пошук, що аналізує не лише ключові слова, але й контекст запиту

Попри всі перераховані оптимізаційні технології пошуку сучасні пошукові системи часто обмежені традиційними методами, які вимагають від користувачів точних ключових слів або ручного пошуку шляхом навігації через меню на сторінках [6].

Застосування штучного інтелекту у вебпошуку значно покращує точність та швидкість знаходження релевантної інформації [7]. Методи обробки природної мови дозволяють системам краще розуміти зміст пошукових запитів. Глибокі нейронні мережі аналізують запити та результати пошуку. Штучний інтелект аналізує попередні пошукові запити, поведінку користувача та його інтереси, щоб формувати персоналізовані результати [8].

Використання застосунків для пошуку інформації на вебсайтах значно спростило б роботу з освітньою інформацією. Застосування генеративного штучного інтелекту в таких програмах зробить їх універсальними та актуальними для студентів та викладачів.

## **1.2 Огляд технологій генеративного штучного інтелекту**

Генеративний штучний інтелект – це тип штучного інтелекту, який може створювати новий контент, наприклад, текст, зображення, аудіо та відео, використовуючи породжувальні моделі. Він відрізняється від традиційних представників, які в основному використовуються для аналізу та класифікації даних. Генеративний ШІ використовує моделі машинного навчання для вивчення закономірностей у великих наборах даних [9].

Породжувальні моделі вивчають зразки та структури своїх вхідних тренувальних даних і потім породжують нові дані, що мають подібні характеристики [10]. Генеративний ШІ може допомогти дизайнерам створювати

нові продукти, викладачам – складати навчальні плани, розробникам ігор – створювати нові рівні, персонажів та історії для них.

LLM, або велика мовна модель – це тип штучного інтелекту, який навчений на величезних обсягах даних [11]. Такі моделі навчаються на терабайтах текстової інформації, включаючи книги, статті, вебсторінки та код, а також встановлюють статистичні зв'язки між словами та фразами (рис. 1.1). Вони мають мільярди або навіть трильйони параметрів, які дозволяють встановити складні мовні шаблони. Це дозволяє розуміти та генерувати людську мову[12].



Рисунок 1.1 – Процес навчання великої мовної моделі

Більшість сучасних LLM використовують трансформерну архітектуру, яка ефективно обробляє послідовні дані. Завдяки цьому моделі генерують граматично правильний, логічний і часто дуже схожий на людський текст. Вони можуть виконувати широкий спектр завдань обробки природної мови, включаючи генерацію текстів, переклад мов, відповіді на запитання, підсумовування інформації, класифікацію тексту а також встановлювати ймовірність появи наступного слова в реченні [13].

Трансформерна архітектура – це революційна архітектура нейронних мереж, яка змінила спосіб обробки послідовностей даних [14].

Основна інновація трансформерів – це механізм уваги, завдяки якому модель зосереджується на різних частинах вхідної послідовності при обробці

кожного елемента. Така архітектура дозволяє враховувати контекст і залежності між словами в реченні, що є критично важливим для розуміння мови [15]. Система-трансформер зазвичай складається з двох основних частин: кодувальника і декодувальника (рис. 1.2).

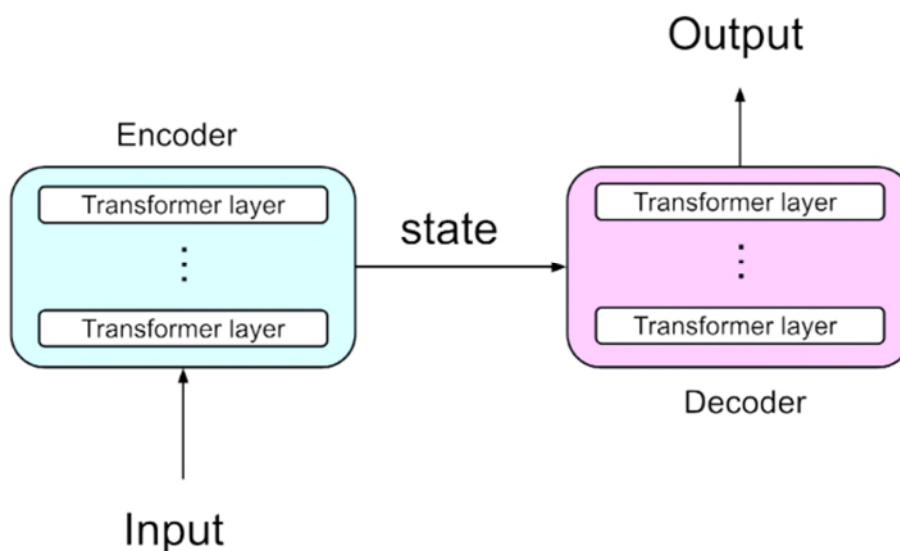


Рисунок 1.2 – Принцип роботи трансформерної архітектури

Кодувальник обробляє вхідну послідовність і перетворює її в внутрішнє представлення. Декодувальник використовує це представлення для генерації вихідної послідовності. На відміну від RNN, які обробляють інформацію послідовно, трансформери можуть обробляти всі елементи послідовності паралельно. Це значно прискорює навчання і робить трансформери більш ефективними для обробки довгих послідовностей. Трансформерна архітектура лежить в основі багатьох сучасних LLM, таких як GPT-3 і BERT, а також використовується в інших областях, таких як комп'ютерний зір і розпізнавання мови [16].

### 1.3 Огляд існуючих засобів пошуку інформації на вебсайті на основі генеративного штучного інтелекту

Засоби пошуку інформації на основі генеративного штучного інтелекту протягом останніх років набули популярності через великий вибір та зручність у користуванні.

Google Custom Search JSON API – це інтерфейс прикладного програмування, що надає можливість здійснювати пошук інформації через Google за допомогою програмного коду. Він дозволяє інтегрувати функціональність пошукового механізму Google у сторонні вебзастосунки, мобільні додатки або інші інформаційні системи (рис. 1.3). Для автентифікації запитів до Google інтерфейс використовує унікальний ідентифікатор – API-ключ. Користувачі отримують API-ключ через Google Cloud Console, де можна також встановлювати обмеження для запитів, що забезпечує контроль над доступом та запобігає несанкціонованому використанню. Окрім API-ключа, для роботи з Google Custom Search JSON API необхідно мати ідентифікатор пошукової системи Search Engine ID [17]. Цей ідентифікатор визначає, які вебресурси або домени використовуватимуться для пошуку, що дозволяє здійснювати як глобальний пошук через Google, так і обмежений пошук у межах конкретного сайту або групи сайтів.

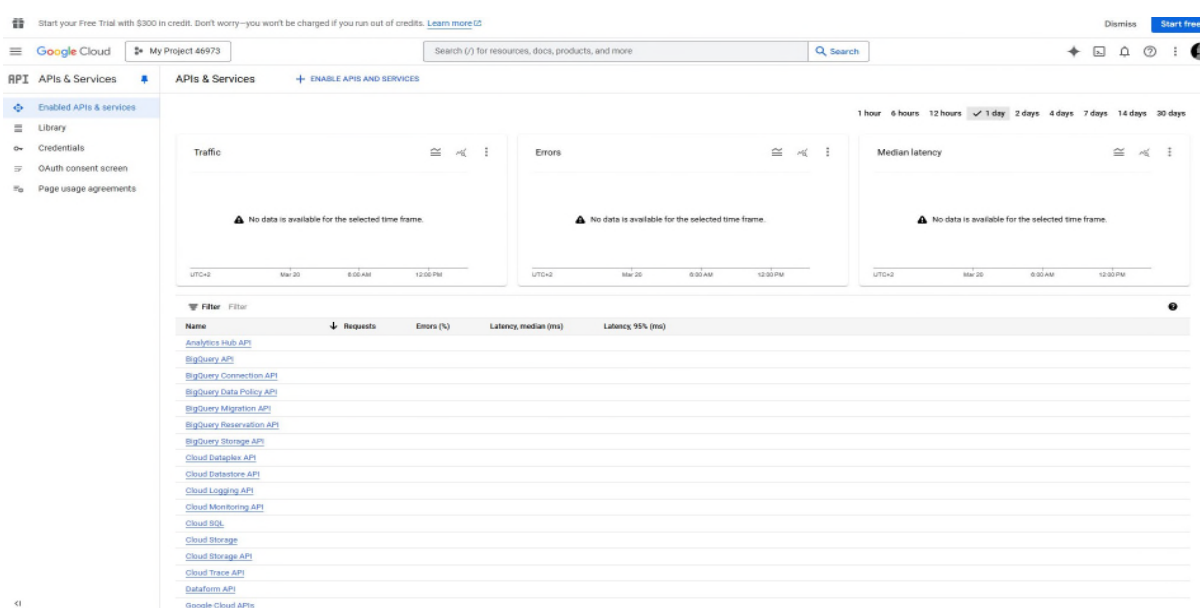


Рисунок 1.3 – Google Custom Search JSON API

Основні можливості Google Custom Search JSON API включають виконання пошукових запитів у програмному середовищі з використанням HTTP-запитів, отримання результатів у форматі JSON, фільтрацію та налаштування результатів пошуку, та використання додаткових параметрів, таких як SafeSearch, мова результатів та персоналізовані критерії пошуку [18].

Google Custom Search JSON API – ефективний інструмент для розробників, які прагнуть інтегрувати розширені пошукові можливості в свої застосунки, не витрачаючи ресурси на створення власного пошукового механізму. Однак важливо дотримуватися політики безпеки Google та не розголошувати API-ключ у відкритих джерелах, щоб запобігти його несанкціонованому використанню [19].

Perplexity AI – це пошукова система, яка використовує генеративний штучний інтелект на основі LLM для аналізу та узагальнення інформації з різних джерел в Інтернеті [20]. Вона працює подібно до традиційних пошукових систем, але замість простого списку посилань надає детальні, згенеровані AI відповіді з посиланнями на джерела (рис. 1.4).

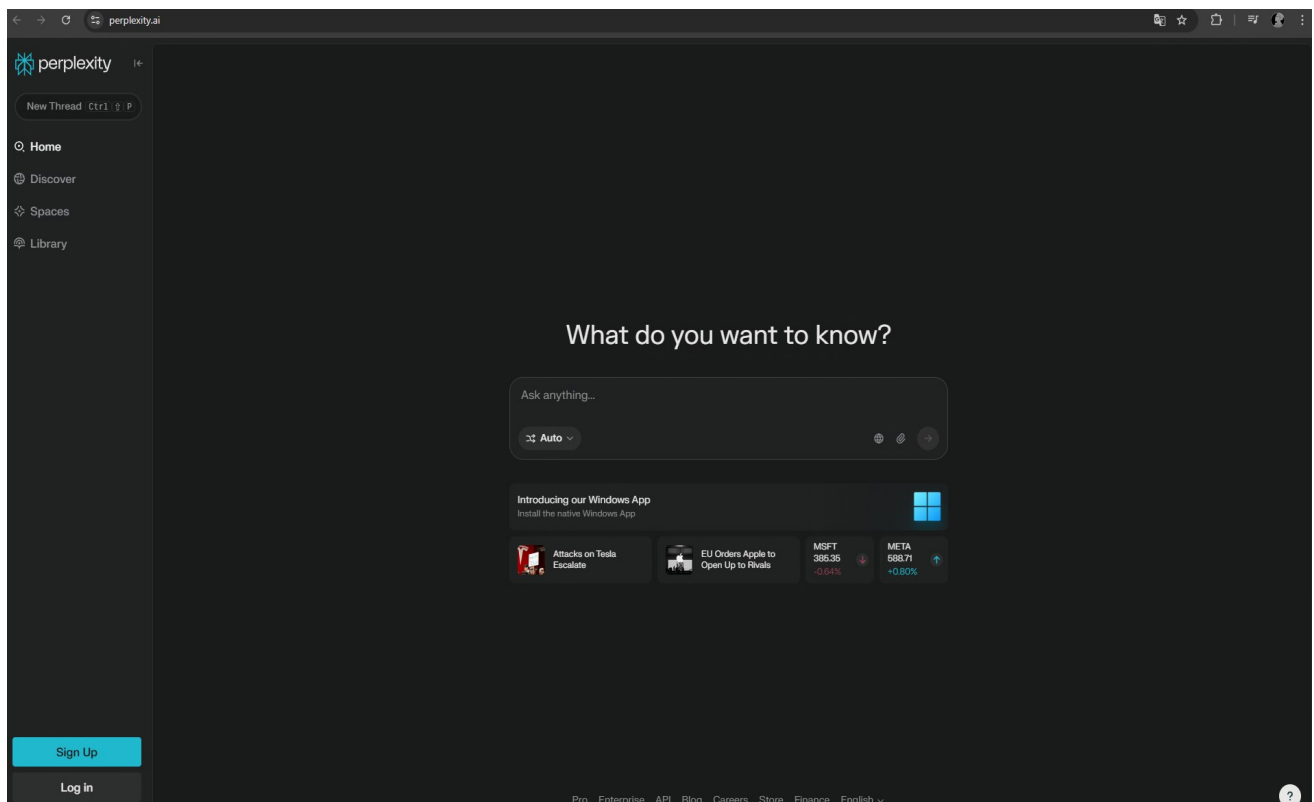


Рисунок 1.4 – Інтерфейс пошукової системи Perplexity AI

Система аналізує дані з різних сайтів, узагальнює ключову інформацію та генерує відповіді на основі кількох джерел. При генерації інформації Perplexity AI підтверджує джерела шляхом вставлення посилань на першоджерела у відповідних місцях. Також пошукова система підтримує діалоговий формат, контекстний пошук та голосовий пошук. Perplexity AI доступна для інтеграції в інші сервіси [21].

Glasp – це розширення для браузерів, яке дозволяє виділяти, зберігати та організовувати інформацію з вебсторінок [22]. Розширення використовує штучний інтелект для генерації підсумків та створення нотаток на основі збереженого контенту (рис. 1.5).

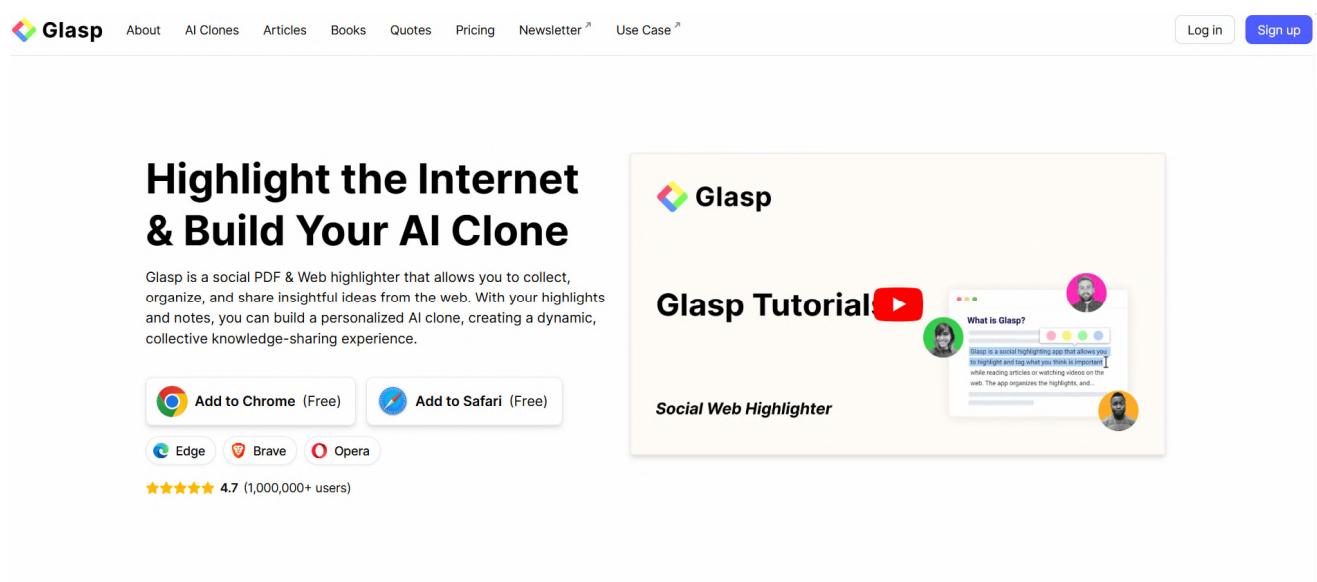


Рисунок 1.5 – Основна сторінка Glasp

За допомогою Glasp можна виділяти важливі фрагменти тексту на сайтах та зберігати їх, генерувати підсумки прочитаного матеріалу та організовувати збережені нотатки. Також присутня можливість ділитися збереженим контентом із іншими користувачами через соціальні мережі.

Glasp синхронізується з Notion та Obsidian для автоматично збереження виділеного тексту у нотатках. Додатково можна генерувати конспекти у вигляді короткого викладу статей [23].

Розширення для браузерів Glasp буде особливо зручним для дослідників і студентів, оскільки допомагатиме збирати матеріали для навчання; для блогерів і авторів – спрощуватиме збереження ідей для статей; для аналітиків та журналістів – дозволить швидко та зручно організувати дослідницькі дані [24].

You.com — це пошукова система на основі штучного інтелекту, яка відрізняється від традиційних систем типу Google чи Bing завдяки інтеграції генеративного ШІ та можливості персоналізації пошуку (рис.1.6).

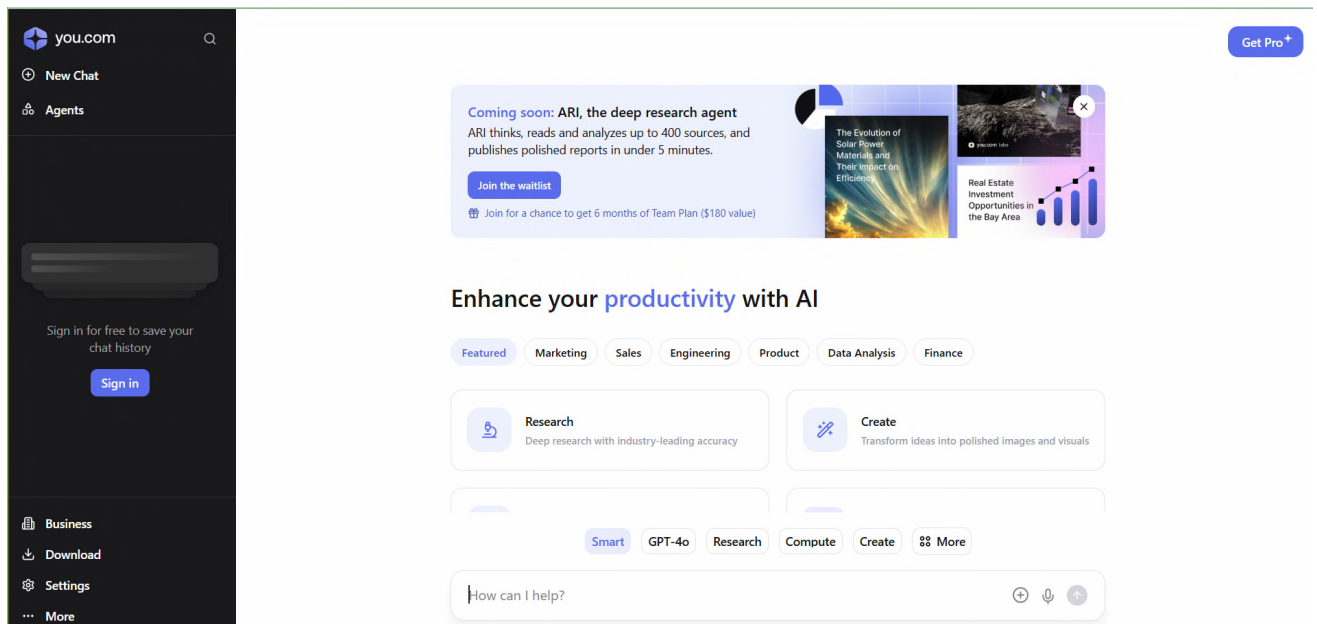


Рисунок 1.6 – Пошукова система You.com

You.com використовує великі мовні моделі для формування відповідей на запити у зрозумілій текстовій формі. You.com має YouChat – вбудований ШІ-асистент, який працює подібно до ChatGPT, даючи розгорнуті відповіді на складні питання [25]. Чат може використовувати мультимодальний пошук, тобто комбінувати текстові, візуальні, відео- та кодові результати в єдиній видачі.

Користувачі You.com можуть кастомізувати результати, тобто налаштувати джерела інформації для більш гнучкого пошуку[26].

You.com забезпечує конфіденційність – розробники заявили, що не відстежують персональні дані користувачів так активно, як Google [27].

Детальний огляд засобів пошуку інформації на основі генеративного штучного інтелекту допоможе врахувати всі аспекти та розробити застосунок, який буде якісно виконувати функції пошуку та аналізу інформації на вебресурсі [28].

#### **1.4 Мета та завдання кваліфікаційної роботи**

Мета кваліфікаційної роботи бакалавра – підвищення ефективності пошуку інформації на вебсайті університету шляхом застосування генеративного штучного інтелекту.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- Дослідити існуючі методи та технології інтерактивного пошуку інформації.
- Розробити метод інтерактивного пошуку інформації на вебсайті університету з використанням генеративного штучного інтелекту.
- Реалізувати вебзастосунок для забезпечення ефективного інтерактивного пошуку інформації.
- Оцінити ефективність методу інтерактивного пошуку інформації за допомогою тестування та метрик якості.

Інтерфейс користувача вебзастосунку для роботи з методом пошуку інформації на основі генеративного штучного інтелекту повинен бути зручним, функціональним та адаптивним, забезпечуючи ефективний і швидкий доступ до інформації.

Створений застосунок на основі методу інтерактивного пошуку дозволить значно спростити отримання інформації для студентів, викладачів та адміністрації університету. Генеративний штучний інтелект забезпечить швидку та релевантну видачу відповідей, адаптованих до природної мови запитів користувачів, що підвищить ефективність використання вебсайту університету.

## Розділ 2 Проектування методу інтерактивного пошуку інформації для вебсайту університету

### 2.1 Загальна ідея інтерактивного пошуку інформації на основі генеративного штучного інтелекту.

Метод інтерактивного пошуку інформації на вебсайті університету на основі генеративного штучного інтелекту отримує повідомлення від користувача та надсилає відповіді у форматі чату, а також здійснює пошук інформації на вебсайті. Використання генеративного штучного інтелекту дозволяє формувати відповіді користувачу, максимально наближені до природньої мови.

Схема методу роботи представлена на рисунку 2.1.

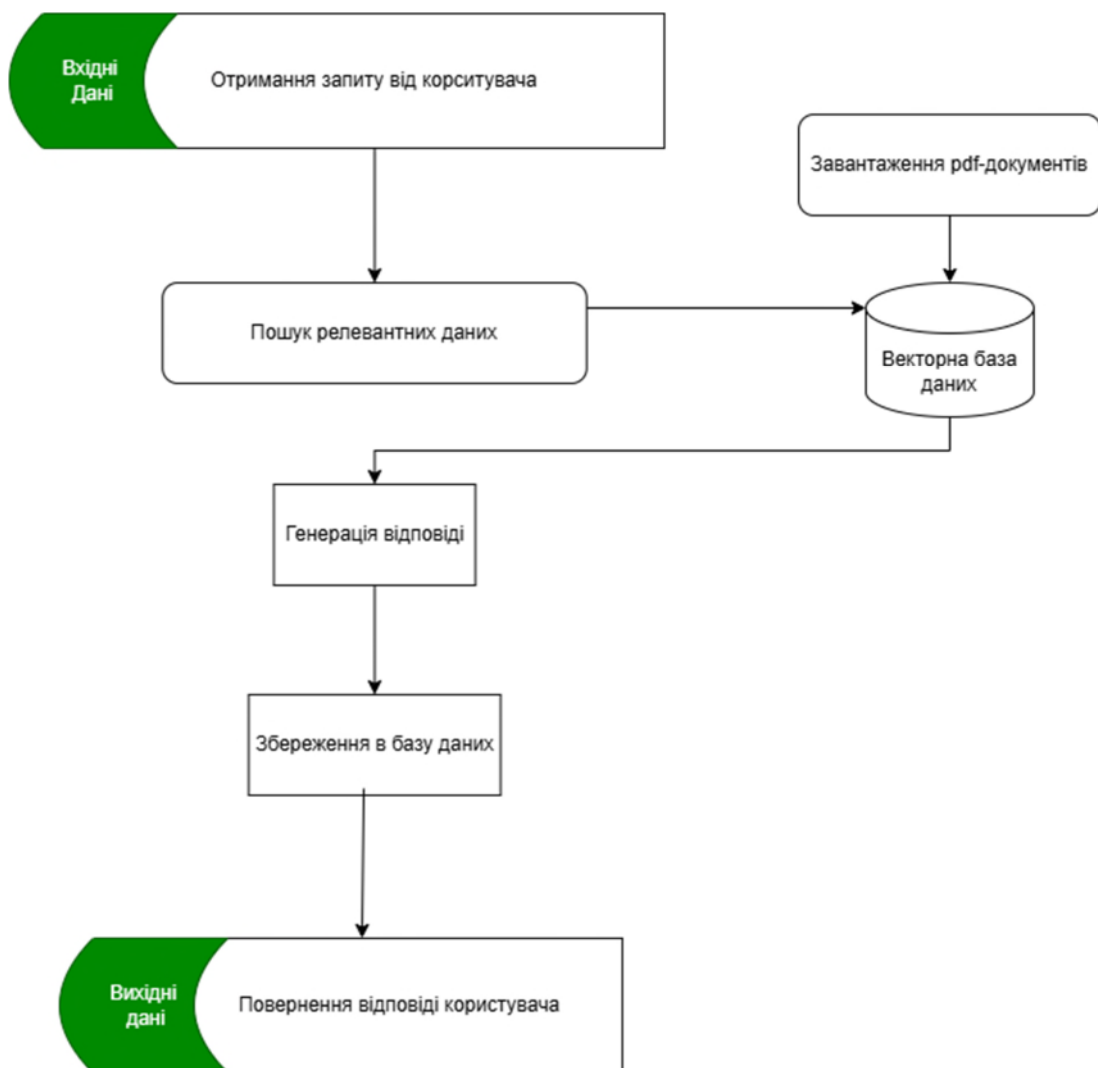


Рисунок 2.1 – Загальна схема пошуку інформації на основі генеративного штучного інтелекту.

Схема зображує загальну ідею пошуку методу інтерактивного пошуку інформації на вебсайті університету на основі RAG із використанням векторної бази даних і великої мовної моделі для генерації відповідей на основі PDF-документів.

Робота методу пошуку інформації на вебсайті університету розпочинається з отримання запиту від користувача. Далі текст запиту передається до RAG.

Після отримання запиту від користувача, RAG створює ембеддинг запиту, тобто перетворює вхідний текст запиту у набір чисел та векторів для подальшої обробки. Паралельно до системи вже попередньо завантажено PDF-документи, які також були перетворені у вектори і збережені у векторну базу даних.

Після створення ембеддингу запиту відбувається пошук у векторній базі даних найбільш релевантних даних на основі схожості векторів. Найбільш відповідні векторні представлення витягуються, і з них формується текстовий контекст – основа для відповіді на запит.

Далі сформований контекст разом із отриманим від користувача запитом передається у велику мовну модель. Вона аналізує обидва компоненти і генерує відповідь, яка максимально точно відповідає запиту користувача з урахуванням знайденої інформації. Після генерації відповідь зберігається у базу даних для подальшого використання або аудиту. У завершальному етапі система повертає відповідь користувачеві як вихідні дані.

При отриманні від користувача нового запиту або уточнення попереднього весь описаний алгоритм повторюється.

## **2.2 Архітектура LLM моделі з пошуково-доповненою генерацією**

Метод інтерактивного пошуку інформації на вебсайті університету на основі генеративного штучного інтелекту використовуватиме LLM для генерації відповідей користувачам та RAG для розширення можливостей великих мовних моделей за допомогою зовнішньої бази знань.

Large language model – це велика модель нейронної мережі, яка передбачає наступний токен на основі попереднього прогнозованого. Популярність LLM пояснюється їх універсальністю та ефективністю. Вони чудово справляються з такими завданнями, як переклад, конспектування та аналіз даних [29].

Архітектура великої мовної моделі включає кілька рівнів та компонентів, кожен з яких робить внесок у здатність моделі до розуміння та створення текстів близьких до людської мови. Ключові компоненти архітектури LLM: вхідний рівень та токенізація, шар ембеддингу, блок трансформерів, вихідний шар декодування а також навчання та точне налаштування.

Після отримання даних на вхід велика мовна модель виконує токенізацію. Вхідний текст розбивається на менші одиниці, які називаються токенами, що можуть бути словами, підсловами або символами. Ці токени потім перетворюються на числові представлення, які модель може обробити, тобто виконується індексація. Індексація – це процес призначення числових ідентифікаторів (індексів) кожному слову, токену або символу, щоб модель могла обробляти текст у вигляді чисел, а не символів. Всі токени збираються у словник, кожному токену призначається унікальний індекс – ціле число. Текст перетворюється в послідовність чисел (індексів токенів).

На шарі ембеддингу модель будує словникові та позиційні ембеддинги.

Для створення словникових ембеддингів кожен токен зіставляється з вектором у багатовимірному просторі, який відображає його семантичне значення. Оскільки трансформери самі по собі не враховують порядок токенів, до словникових ембеддингів додається інформація про позицію токена в реченні – позиційні ембеддинги.

На етапі блоку трансформерів спрацьовує механізм самоуваги. Цей механізм виконує оцінки уваги та визначає скільки уваги кожен токен повинен приділити іншим токенам у послідовності. Вектор «запиту» порівнюється з «ключами», щоб обчислити ваги уваги через скалярний добуток і Softmax. Потім ці ваги застосовуються до «значень». Кілька «голів» обробляють увагу паралельно в різних підпросторах. Результати об'єднуються і перетворюються назад у

початковий простір. Після відпрацювання механізму уваги вихідні дані обробляються окремо для кожної позиції через щільну нейронну мережу з нелінійною активацією. Кожен підшар супроводжується нормалізацією шару та залишковим з'єднанням. Це стабілізує навчання та дозволяє будувати глибші мережі.

Зазвичай LLM складається з багатьох трансформер-блоків. Кожен блок включає багатоголову увагу та нейронну мережу. Для об'єднання блоків виконується глибоке стекування, яке дозволяє моделі будувати складні ієрархічні уявлення на основі тексту.

На вихідному шарі відбувається декодування даних та передача даних до автогресивних моделей або маскованих мовних моделей. Автогресивні моделі – передбачають наступний токен на основі попередніх. Масковані мовні моделі – передбачають приховані токени в середині речення.

Останній шар застосовує Softmax для перетворення векторів у ймовірнісні розподіли по словнику, щоб вибрати найімовірніший токен. Softmax – це математична функція, яка перетворює набір чисел у ймовірнісний розподіл.

Далі за необхідності відбувається навчання та донавчання.

Навчання великих мовних моделей – це багатоетапний процес який включає такі кроки: збір текстових даних, очищення даних, поділ даних, навчання моделі, перевірка та налаштування, використання моделі для генерування тексту на основі нових даних та покращення моделі (рис.2.2).

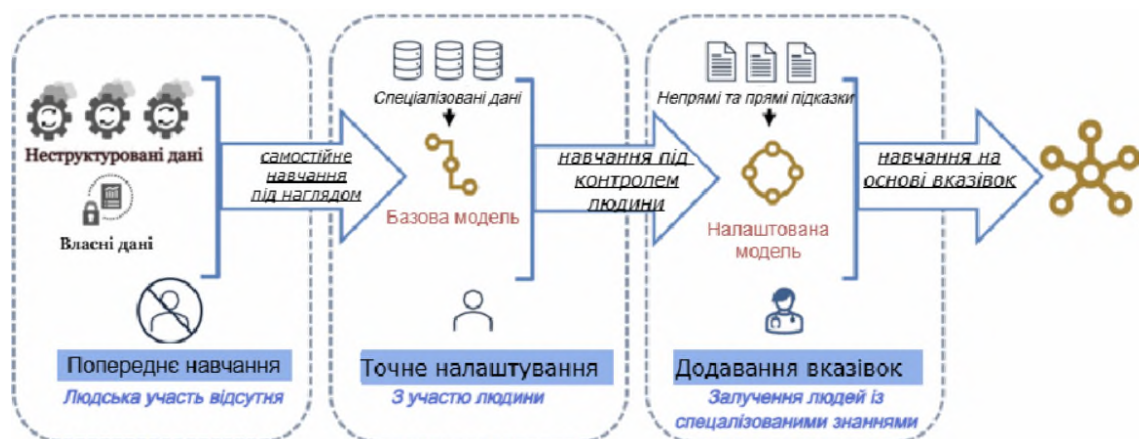


Рисунок 2.2 – Класичний підхід до налаштування LLM на основі даних, специфічних для предметної області [30]

Навчання LLM починається зі збору величезної кількості текстових даних. Ці дані можуть надходити з книг, вебсайтів, pdf-файлів, статей або платформ соціальних мереж. Мета – охопити багате розмаїття людської мови.

На етапі очищення даних необроблені текстові дані приводять у порядок у процесі, який називається попередньою обробкою. Це включає такі завдання, як видалення небажаних символів, розбиття тексту на менші частини, які називаються токенами, і переведення отриманих токенів у формат, з яким може працювати модель.

На наступному етапі очищені дані розділяються на два набори. Один набір – навчальні дані, буде використовуватися для навчання моделі. Інший набір – дані перевірки, буде використано пізніше для тестування продуктивності моделі.

Наступний етап передбачає саме навчання. Модель LLM навчається, переглядаючи навчальні дані, роблячи прогнози на основі того, що вона навчилася до цього часу, а потім регулює свої внутрішні параметри, щоб зменшити різницю між її прогнозами та фактичними даними.

Після навчання моделі обов'язково відбувається перевірка моделі. Навчання моделі LLM перевіряється за допомогою даних для перевірки. Це допомагає побачити, наскільки добре працює модель, і налаштувати параметри моделі для кращої продуктивності.

Після навчання та оцінки модель готова до використання. Тепер її можна інтегрувати в програми чи системи, де вона генеруватиме текст на основі нових введених даних. Модель LLM можна вдосконалювати з часом, використовуючи оновлені дані або коригуючи налаштування на основі відгуків і реального використання [31].

Незважаючи на велику популярність LLM, їх використання дуже часто супроводжується певними обмеженнями, серед яких актуальність інформації чи її глибина та специфічність. Програми LLM на основі RAG вирішують саме цю проблему та розширюють корисність LLM для конкретних джерел даних [32].

Retrieval-augmented generation – це технологія для підвищення точності та надійності генеративних моделей ШІ за допомогою фактів, отриманих із зовнішніх джерел. Практично будь-яка LLM може підключитися до будь-якого зовнішнього ресурсу за допомогою RAG. Технологія робить додаток більш надійним для користувачів, оскільки RAG надає моделям джерела, на які вони можуть посылатися, тож користувачі можуть перевірити будь-які твердження. А також retrieval-augmented generation зменшує ймовірність того, що модель зробить неправильне припущення, це явище іноді називають галюцинаціями. RAG робить метод швидшим і дешевшим, ніж перенавчання моделі з додатковими наборами даних, це дозволяє користувачам швидко замінювати нові джерела [33].

Щоб механізм RAG міг ефективно знаходити релевантну інформацію для генерації відповіді, йому потрібен спосіб порівнювати текстові запити з великою кількістю документів. Це досягається за допомогою ембеддингів – векторних подань тексту, які дозволяють вимірювати семантичну схожість між запитом і потенційними джерелами інформації.

Запит перетворюється на вектор за допомогою вбудованої моделі embedding [34]. Моделі ембеддингу фіксують семантичне значення об'єктів даних і представляють їх у вигляді масиву чисел (рис. 2.3).

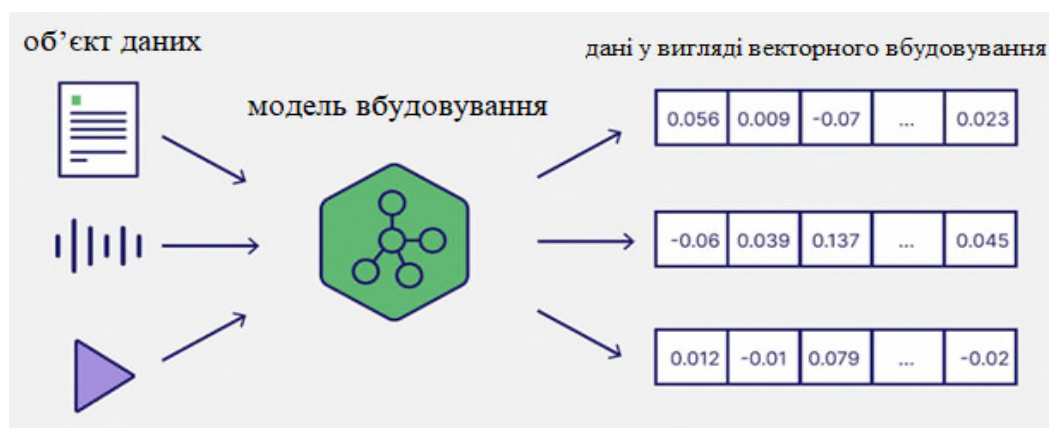


Рисунок 2.3 – Модель embedding [35]

Після створення ембеддингу метод інтерактивного пошуку інформації на вебсайті університету виконує пошук відповідного тексту у базі даних, використовуючи векторне представлення запити. Отриманий з БД текст буде

використано з оригінальним запитом для створення нової відповіді або тексту. Генератори об'єднуюватимуть інформацію з цих уривків тексту для створення зворотного зв'язку. Після цього створений текст буде повернуто користувачеві як повну відповідь.

### 2.3 Векторна база даних

Успіх RAG значною мірою залежить від його здатності ефективно отримувати доступ до величезних обсягів інформації та обробляти їх. Тут на допомогу приходять векторні бази даних, які є основою систем RAG. Ці бази даних зберігають і обробляють багатовимірні векторні дані, як правило, отримані зі складних структур даних, таких як текст, зображення або звуки, перетворені в математичні вектори. Ефективність моделі RAG нерозривно пов'язана з продуктивністю векторних баз даних [36].

Різниця між векторними базами даних і векторними бібліотеками в тому, що векторні бібліотеки здебільшого використовуються для статичних даних, де дані індексу незмінні. Бібліотеки зберігають лише вбудовані вектори, а не пов'язані об'єкти з яких вони були згенеровані, тому, на відміну від векторних баз даних, векторні бібліотеки не мають підтримки CRUD.

Пошук подібності для векторів – це техніка, яка використовується для пошуку векторів у просторі великої розмірності, подібних до заданого вектора запиту. Векторами зберігають дані будь-яких типів, які можна представити як вектор, включаючи текст, зображення та аудіо.

Пошук подібності – це популярна проблема в машинному навчанні, вирішення її з часом лише ускладнюється оскільки розміри даних збільшуються. Для великомасштабних наборів даних традиційні методи пошуку подібності, такі як лінійний пошук і k-дерева (рис. 2.4), стають неможливими. FAISS вирішує цю проблему, надаючи ефективні алгоритми для пошуку подібності та кластеризації, які здатні працювати з даними великої розмірності.

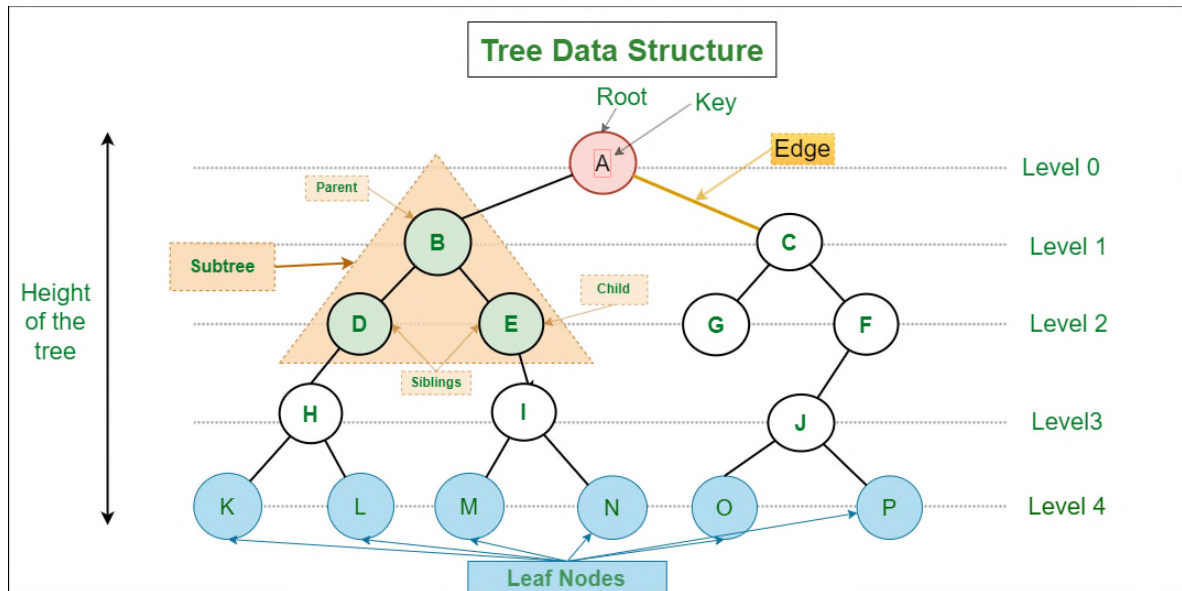


Рисунок 2.4 – K-дерево

FAISS також має можливість повернення не лише найближчого сусіда, а  $k$ -го найближчого сусіда. Це дає більше гнучкості та дозволяє приймати кращі рішення, враховуючи не просто найближчого, а загальну тенденцію серед схожих. Присутня можливість збереження даних на диску замість збереження в оперативній пам'яті.

Facebook AI Similarity Search виконує пошук кількох векторів одночасно, а не одного як це відбувається зазвичай. Для багатьох типів індексів це швидше, ніж пошук одного вектора за іншим (рис. 2.5). Замість мінімального евклідового пошуку має підтримку інших відстаней та може виконувати пошук за діапазоном – повернути всі елементи, які знаходяться в заданому радіусі точки запиту. За основу пошуку FAISS використовує індексні двійкові вектори, а не вектори з плаваючою комою [38].

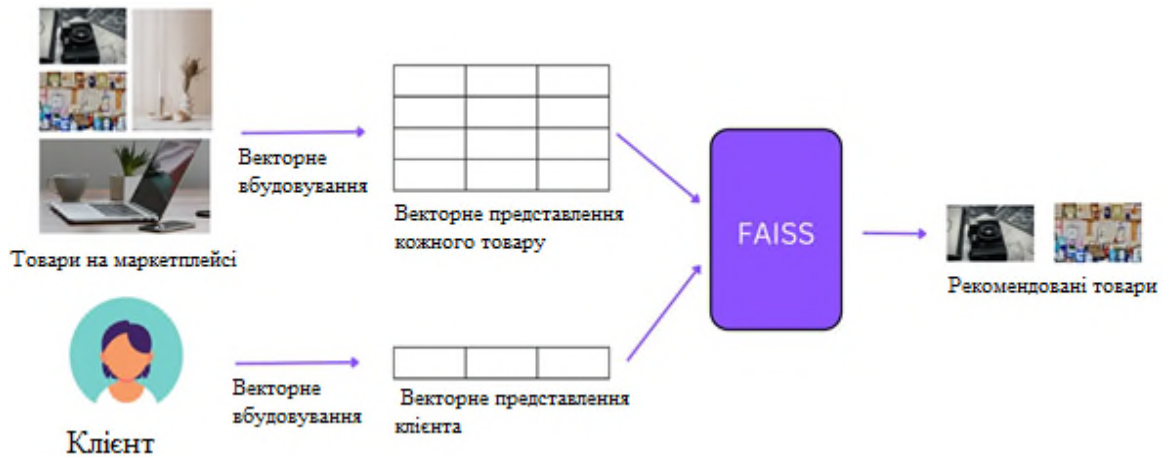


Рисунок 2.5 – Схема роботи векторного пошуку [39]

FAISS побудовано на концепції індексування, що є методом попередньої обробки набору даних для покращення його пошуку. Групуючи порівнянні компоненти разом, індексування зменшує кількість елементів, які необхідно порівнювати протягом пошуку. Квантування продукту (PQ) та інвертовані структури індексування файлів – це два основні типи структур індексування, що використовуються у FAISS (IVF) [39].

#### 2.4 Метод інтерактивного пошуку інформації на основі генеративного штучного інтелекту

На основі загальної ідеї інтерактивного пошуку, описаної в підрозділі 2.1, метод інтерактивного пошуку деталізує процес доступу до інформації на вебсайті університету з використанням генеративного штучного інтелекту. Метод інтегрує концепції DeepSeek API для обробки запитів і генерації відповідей, а також FAISS для роботи з векторною базою даних. Концепція використання DeepSeek API виступає абстрактним підходом до інтеграції генеративного штучного інтелекту в метод, забезпечуючи обробку текстових запитів. Процес роботи методу включає:

- аналіз запитів природної мови із перетворенням їх на векторні представлення для подальшого пошуку;
- генерацію відповідей на основі релевантних даних, отриманих із векторної бази;

– оптимізацію процесу синтезу інформації для підвищення ефективності та точності.

DeepSeek API концептуально пов'язаний із пошуково-доповненою генерацією RAG та дозволяє методу адаптуватися до потреб університету. DeepSeek Chat представляє підхід до розмовного штучного інтелекту, який забезпечує інтерактивність у методі та дозволяє зберігати контексти запитів для обробки уточнень, що підвищує зручність взаємодії та генерацію відповідей, адаптованих до термінології університету. Також DeepSeek Chat надає підтримку послідовного діалогу для вдосконалення результатів пошуку.

Векторна база даних FAISS використовується у методі інтерактивного пошуку інформації для оптимізації пошуку релевантних даних. FAISS виконує задачі індексації векторних представлень документів для швидкого доступу а також пошук найближчих до запиту векторів.

Застосунок пошуку інформації на вебсайті університету отримуватиме запит від користувача у форматі чату. Після генерації відповіді користувач отримає сформований текст у полі чату. Якщо відповідь, яку отримав користувач не повна або не відповідає на питання користувача у повному обсязі, він може надати додаткові або уточнюючі запитання і застосунок згенерує нову відповідь, яка відповідатиме потребам користувача. Детальна схема роботи застосунку зображена на рисунку 2.6.

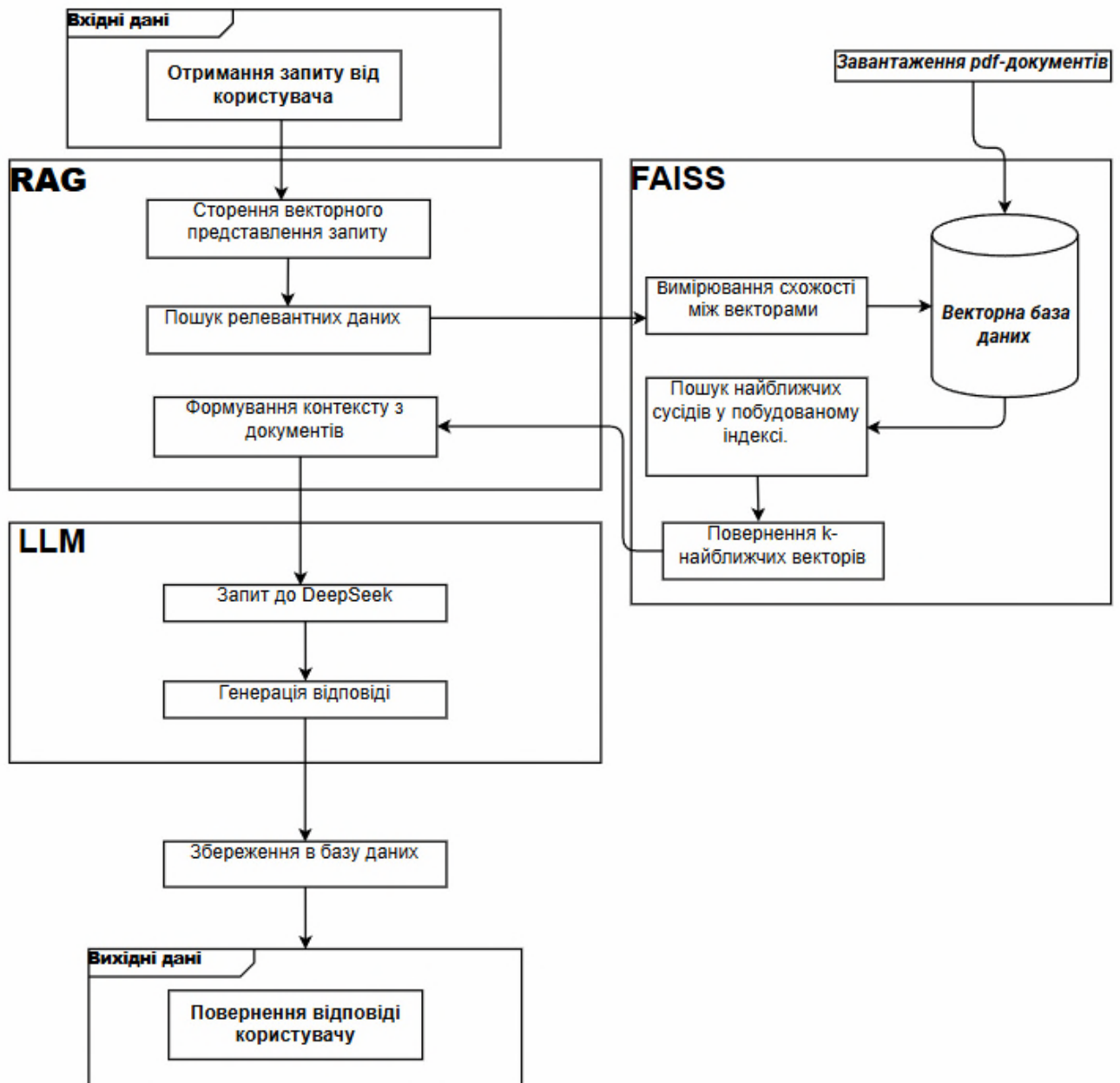


Рисунок 2.6 – Детальна схема роботи методу пошуку інформації на вебсайті

Для формування відповідей на запити користувачів застосунок використовує велику мовну модель DeepSeek Арі.

Схема роботи DeepSeek Арі на прикладі реального запиту від користувача зображена на рисунку 2.7. Запит: Скільки всього факультетів є у ХНУ? Отримана відповідь: 8 факультетів.

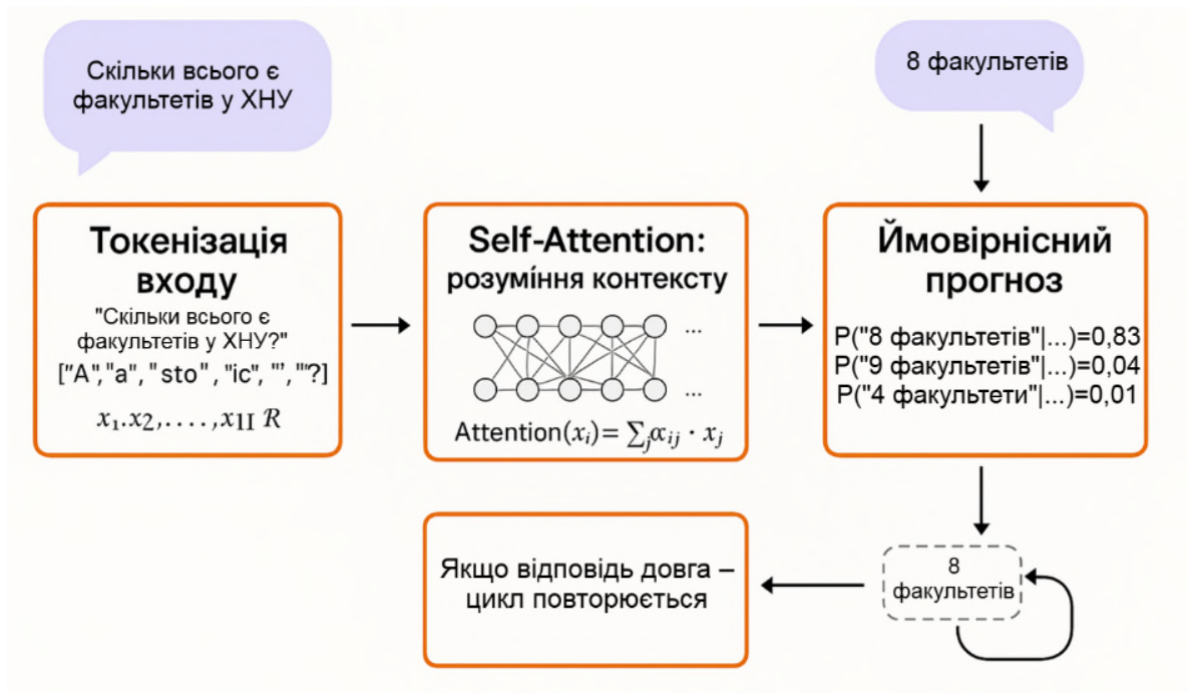


Рисунок 2.7 – Принцип роботи DeepSeek Chat на прикладі реального запиту від користувача

Донавчання відбувається на основі PDF-файлів, які містять інформацію із сайту університету (рис 2.8).

Весь процес роботи RAG можна розділити на 2 фази:

1. Індексція – підготовка та векторизація даних.
2. Генерації відповіді – пошук релевантних фрагментів.



Рисунок 2.8 – Двофазний процес Retrieval-augmented generation [40]

Перша фаза роботи RAG передбачає завантаження даних із зовнішніх джерел у текст. Цей процес здійснюється за допомогою фреймворку Langchain та бібліотекою PyPDFLoader. Після завантаження тексту бібліотекою PyPDFLoader, він трансформується, тобто перетворюється на фрагменти за допомогою рекурсивного роздільника символів тексту – відбувається поділ тексту на невеликі частини, для оптимізації обробки, зберігання та пошуку інформації, а також покращення якості і точності системи штучного інтелекту. Далі отриманий текст перетворюється на вектори чисел та вбудовується у векторну базу даних FAISS (рис. 2.9).

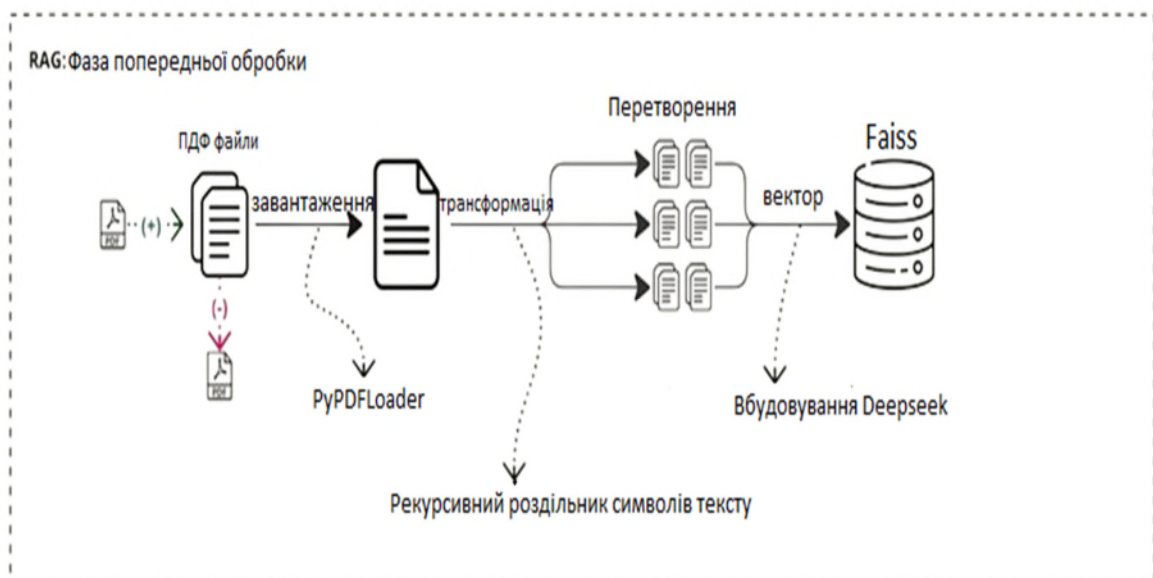


Рисунок 2.9 – Фаза попередньої обробки системи RAG [40]

Друга фаза роботи Retrieval-augmented generation передбачає обробку запиту користувача. Після отримання запиту, він перетворюється на вектор. Система шукатиме відповідні запиту фрагменти тексту у векторній базі даних та на основі них формуватиме запит. Схему другої фази роботи RAG зображено на рисунку 2.10.

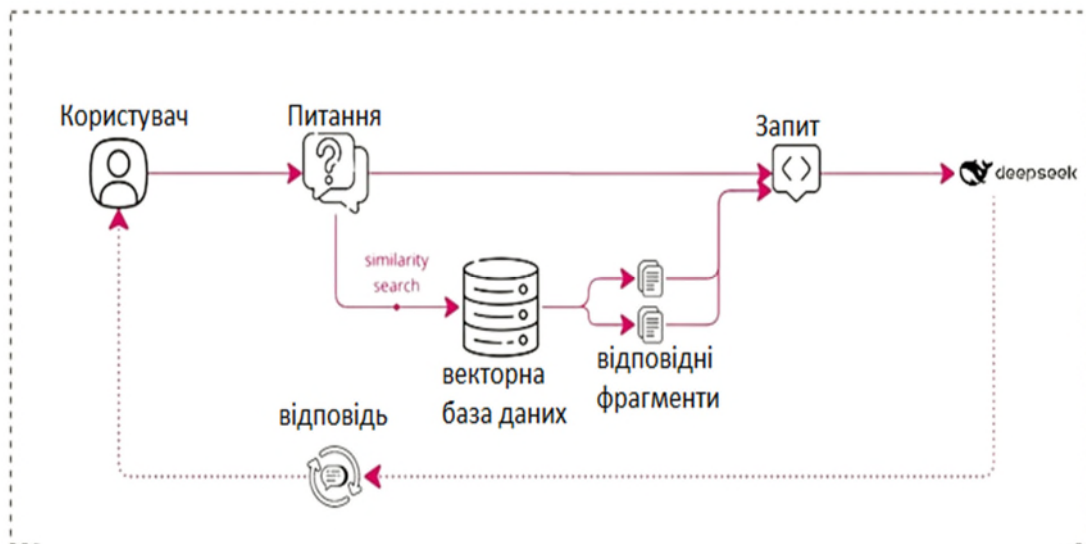


Рисунок 2.10 – Фаза формування відповіді користувачу [40]

Метод інтерактивного пошуку інформації на вебсайті університету на основі генеративного штучного інтелекту використовуватиме векторну бібліотеку Facebook AI Similarity Search.

Facebook AI Similarity Search або FAISS – це бібліотека для ефективного пошуку подібності та кластеризації щільних векторів. Вона містить алгоритми, які здійснюють пошук у наборах векторів будь-якого розміру, аж до тих, які, можливо, не поміщаються в оперативній пам'яті, а також допоміжний код для оцінки та налаштування параметрів [41].

## 2.5 Критерії та метрики оцінки якості методу пошуку інформації

Оцінка якості роботи методу інтерактивного пошуку інформації на основі генеративного ШІ проводитиметься двома способами – шляхом експертної оцінки, де користувач виступатиме у ролі експерта та за метрикою ROUGE.

Експертна оцінка релевантності роботи методу пошуку інформації – це порівняння і перевірки чи здатен метод знайти коректну й точну інформацію з заданого набору джерел.

Критерії експертного оцінювання якості роботи методу пошуку інформації на вебсайті університету:

1. Чи знайдено інформацію у відповідному джерелі.
2. Чи є відповідь релевантною запиту.
3. Чи збережено повноту та точність змісту.
4. Чи витягнуто вміст з pdf-фала без викривлення

За кожен з критеріїв експерт може виставити оцінку у балах мінімум – 0 балів, максимум – 5 балів.

ROUGE – це стандартна метрика для автоматичної оцінки якості згенерованого тексту шляхом порівняння його з еталонними відповідями. Вона вимірює збіги за словами, фразами або послідовностями використовуючи метрики ROUGE-N та ROUGE-L. ROUGE особливо часто застосовується для оцінки систем автоматичного реферування, чат-ботів і генерації природної мови.

Для тестування на основі метрики ROUGE потрібно підготувати дві множини текстів: *reference* (еталонні тексти) – тексти, які вважаються «правильними», написані людиною та *generated* (згенеровані тексти) – результати моделі, які потрібно оцінити.

Тестування методу інтерактивного пошуку інформації на вебсайті університету проводитиметься на основі метрик ROUGE-1 та ROUGE-L.

ROUGE-1 – це метрика з набору ROUGE, яка оцінює якість згенерованого тексту, порівнюючи його з еталонним за допомогою окремих слів. Вона визначає, наскільки слова з відповіді моделі збігаються зі словами в еталонному тексті.

Метрика розраховує: *recall* – частка слів з еталонного тексту, які є у згенерованому, *precision* – частка слів згенерованого тексту, які є в еталонному, *F1-score* – гармонійне середнє між *precision* та *recall*. ROUGE-1 добре підходить для оцінки загального тематичного збігу, але не враховує порядок слів чи структуру речення. ROUGE-1 видає результат у форматі відсоткового відношення для кожного з трьох критеріїв. Наприклад *precision* 0.8, означає, що 80% слів у відповіді моделі збігаються з еталонним; *recall* 0.60 тобто 60% слів еталону покриті відповіддю моделі; *f1* 0.69 – загальна узгодженість слів 69%. Високий показник ROUGE-1 означає, що модель добре вгадує ключові слова та тематику; низький ROUGE-1 – текст відрізняється лексично, навіть якщо має схожий зміст.

ROUGE-L – це одна з підметрик ROUGE, яка оцінює довжину найдовшої спільної підпоследовності між згенерованим текстом і еталонною відповіддю. Вона враховує порядок слів, що робить її чутливою до структури речення, на відміну від простих збігів слів.

Longest Common Subsequence – це найдовша спільна підпоследовність між двома рядками, яка зберігає порядок символів, але не обов’язково є безперервною. LCS використовується для порівняння згенерованого тексту з еталонним, щоб визначити, наскільки добре модель зберегла структуру та порядок слів.

Метрика ROUGE-L вимірює precision, recall та F1-міру на основі LCS. Recall – частка LCS від довжини еталонного тексту, precision – це частка LCS від довжини згенерованого тексту, F1-score – збалансований показник якості відповідності. ROUGE-L корисна для оцінки граматично й логічно правильних текстів, особливо в задачах генерації відповідей і реферування. Наприклад precision 0.7 означає, що 75% последовності згенерованого тексту узгоджуються з еталоном; recall 0.65 тобто 65% еталонної последовності відтворено у відповіді; f1 0.6 – середній рівень узгодженості структури 60%. Високий показник ROUGE-L дає розуміння, що модель відтворила фрази у правильному порядку, не просто окремі слова, Низький ROUGE-L – текст може містити ті самі слова, але в іншому порядку або без логічної структури.

У результаті тестування метриками метрик ROUGE-1 та ROUGE-L можливі ситуації описані у таблиці 2.1.

Таблиця 2.1 – Можливі ситуації при тестуванні метриками ROUGE-1 та ROUGE-L

Ситуація	ROUGE-1	ROUGE-L	Інтерпретація
Всі ключові слова, без порядку	Високий	Низький	Зміст правильний, але структура порушена
Слова та порядок збігаються	Високий	Високий	Відповідь якісна, схожа на людську

Мало збігів взагалі	Низький	Низький	Відповідь слабка, не відповідає еталону
Близька структура, інші слова	Низький	Високий	Можливо, перефразовано або синоніми

Для тестування на основі метрик ROUGE можна використати бібліотеку `rouge_score`.

## 2.6 Висновки до розділу 2

Для інтерактивного пошуку інформації на вебсайті університету запропоновано використати велику мовну модель для генерації відповідей користувачам та RAG для розширення можливостей великих мовних моделей за допомогою зовнішньої бази знань.

Описано технологію донавчання великих мовних моделей на основі pdf-файлів, які містять інформацію про діяльність університету. Файли перетворюються у векторні представлення та зберігаються у векторній базі даних за допомогою вбудованої технології ембеддингу.

Запропоновано використати методи бібліотеки FAISS для швидкого та ефективного пошуку інформації у векторній базі даних, шляхом порівняння її векторних представлень та знаходження одного та декількох найближчих векторів.

Розроблено метод інтерактивного пошуку інформації, що передбачає обробку користувацького запиту системою RAG, яка з використанням великої мовної моделі DeepSeek виконує пошук релевантних даних і формує контекст на основі знайдених документів для генерації змістовних відповідей.

Описано критерії та метрики оцінювання ефективності методу пошуку інформації двома способами: шляхом експертної оцінки релевантності результатів та за допомогою метрики ROUGE, що дозволяє порівнювати згенеровані відповіді з еталонними.

## Розділ 3. Особливості реалізації та результати тестування методу інтерактивного пошуку інформації

### 3.1 Особливості розробки вебзастосунку для імплементації запропонованого методу

#### 3.1.1. Серверна частина та клієнтський інтерфейс

Для розробки серверної частини методу пошуку інформації на вебсайті університету використано мову програмування python та асинхронний вебфреймворк для створення REST API – FastAPI. Функції зберігання документів та історії запитів виконує вбудована реляційна база даних SQLite. За обробку PDF-документів відповідає бібліотека PyMuPDF (fitz). Transformers (Hugging Face) – для роботи з трансформерними моделями методу пошуку інформації на вебсайті. FAISS – бібліотека для ефективного пошуку та збереження векторних ембедингів. Генерує відповіді на основі контенту LLM – DeepSeek API.

Розробка клієнтського інтерфейсу здійснена за допомогою бібліотеки React.js та типізованої надбудова над JavaScript – TypeScript. Для створення сучасного інтерфейсу використано компонент React.js – Material-UI. Для HTTP-запитів бібліотека Axios (рис. 3.1).

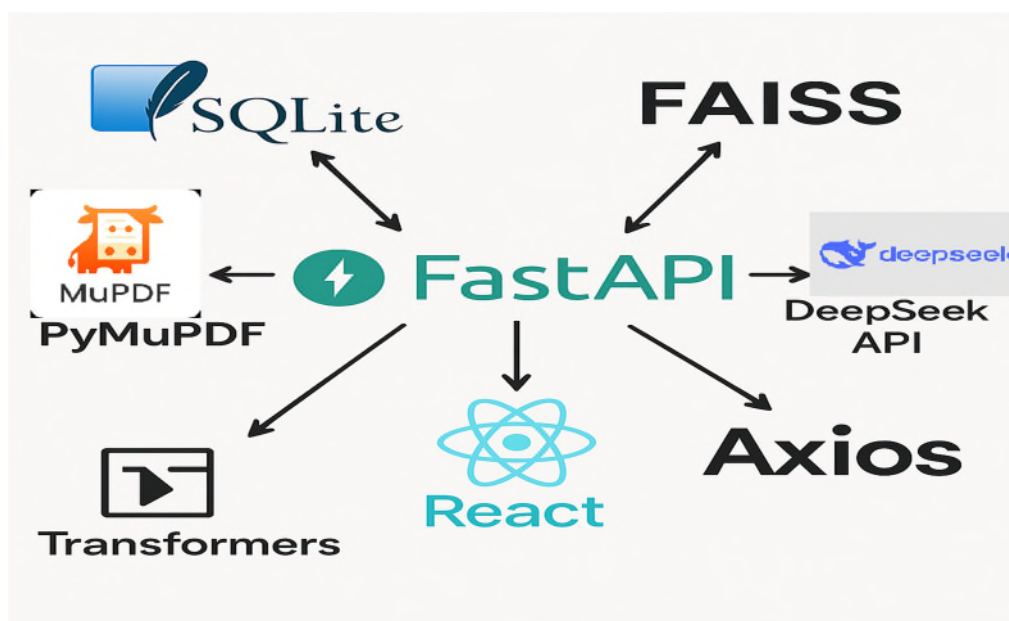


Рисунок 3.1– Інструменти для розробки методу пошуку інформації на вебсайті університету з доповнюючим інформаційним пошуком

FastAPI – це вебфреймворк, орієнтований на створення високопродуктивних RESTful API. Він був розроблений із фокусом на швидкість виконання, розробку без помилок та автоматичну документацію.

FastAPI повністю заснований на сучасному Python (3.7+), включаючи type hints, dataclasses, async/await та підтримує автоматичне документування коду. Документація OpenAPI (Swagger UI та ReDoc) створюється автоматично на основі написаного коду, що значно спрощує тестування та роботу з frontend-розробниками. Фреймворк працює на базі асинхронного вебфреймворку – Starlette та валідації даних – Pydantic. Завдяки використанню type hints, FastAPI виявляє помилки ще до запуску коду.

Фреймворк повністю підтримує async/await, що критично для високонавантажених додатків і мікросервісів та легко інтегрується з іншими Python-бібліотеками, такими як SQLAlchemy, Tortoise ORM, ML/AI моделями (через transformers, PyTorch, TensorFlow), а також інструментами для роботи з PDF або пошуком (PyMuPDF чи FAISS).

FastAPI підходить для бекенд-розробників, які хочуть швидко створювати API, для команд, які розробляють ML-сервіси чи AI-боти а також для стартапів, які цінують швидку розробку без втрати якості.

SQLite – це легка вбудована реляційна база даних, яка зберігає всі дані у звичайному файлі на диску. Вона не потребує встановлення або запуску окремого серверного процесу, тому її часто використовують у мобільних додатках, десктоп-програмах та прототипах. Завдяки підтримці стандартного SQL, вона дозволяє працювати з таблицями, запитами та транзакціями, як у повноцінних СКБД. Водночас SQLite не підходить для великих багатокористувацьких систем через обмеження в паралельності й масштабованості.

PyMuPDF – це Python-бібліотека для роботи з PDF, XPS, EPUB на основі MuPDF. Вона дозволяє читати, аналізувати, редагувати та конвертувати документи без потреби в зовнішніх інструментах.

Ця бібліотека відома своєю високою швидкістю, низьким споживанням пам'яті та широким функціоналом. З її допомогою можна витягувати текст,

зображення, шукати фрагменти, обрізати сторінки, додавати анотації або створювати нові документи з нуля. PyMuPDF активно використовується в автоматизації обробки документів, юридичних системах та аналітиці тексту.

Transformers – це фреймворк від компанії Hugging Face, який надає доступ до тисяч готових моделей машинного навчання для обробки природної мови, зображень, аудіо та інших типів даних. Він підтримує передові архітектури, як-от BERT, GPT, T5, RoBERTa, DistilBERT і дозволяє легко інтегрувати їх у власні проекти. Завдяки простому інтерфейсу, Transformers значно спрощує завантаження моделей, виконання задач (наприклад, класифікація тексту, генерація, переклад, відповіді на запитання) і тонке налаштування моделей під свої дані. Бібліотека активно використовується у наукових дослідженнях, продуктах зі штучним інтелектом і в системах, що потребують глибокого розуміння мови.

FAISS – це бібліотека з відкритим кодом, яка дозволяє швидко здійснювати пошук схожих елементів у великих масивах векторних даних. Її основна мета – ефективно знаходити найближчі сусіди у векторному просторі, що є критично важливим у задачах семантичного пошуку, рекомендаційних систем, кластеризації, обробки зображень та NLP. FAISS особливо цінна тим, що може працювати як на CPU, так і на GPU, масштабуючись на мільйони векторів. Вона підтримує як точні, так і приблизні методи пошуку, що дозволяє досягати балансу між швидкістю й точністю. У поєднанні з трансформерними моделями (для зберігання ембедінгів текстів) FAISS дозволяє створювати потужні системи семантичного пошуку.

React.js – це бібліотека для створення інтерфейсів користувача, яка дозволяє розробляти динамічні вебдодатки. Вона базується на компонентах, що приймають дані через пропси і можуть мати внутрішній стан. React використовує віртуальний DOM для ефективного оновлення лише тих частин інтерфейсу, які змінилися, що покращує продуктивність. Завдяки таким функціям, як JSX і хуки, розробка стає зручнішою та гнучкішою.

TypeScript – це надмножина JavaScript, яка додає статичну типізацію до мови. Вона дозволяє визначати типи змінних, функцій і об'єктів, що допомагає виявляти помилки ще на етапі компіляції. TypeScript сумісний з JavaScript, що дозволяє поступово інтегрувати його в існуючі проекти. Також він підтримує сучасні можливості JavaScript, такі як класи, модулі та асинхронні функції, і додає додаткові можливості, що підвищує надійність коду.

Material-UI – це бібліотека компонентів для React, яка реалізує дизайн від Google. Вона надає набір готових до використання компонентів, таких як кнопки, текстові поля, таблиці та модальні вікна, що допомагає створювати естетично приємні та функціональні інтерфейси. MUI підтримує теми, що дозволяє налаштовувати вигляд компонентів, а також інтегрується з іншими бібліотеками для побудови складних інтерфейсів.

Axios – це бібліотека для виконання HTTP-запитів у JavaScript. Вона дозволяє легко здійснювати запити до серверів для отримання або відправлення даних. Axios підтримує всі типи HTTP-запитів (GET, POST, PUT, DELETE), автоматично перетворює JSON-відповіді в об'єкти JavaScript, а також забезпечує обробку помилок. Крім того, Axios підтримує асинхронні операції, що робить його зручним для роботи з промісами.

DeepSeek Chat – це інтерфейс програмування, який дозволяє інтегрувати LLM від DeepSeek у власні застосунки та має безкоштовний тариф із обмеженням токенів на місяць, а також платні плани з підвищеним лімітом. Він працює за стандартною схемою REST API. Запити надсилають через http метод post, отримуючи відповіді у форматі json [40].

DeepSeek приймає текст, аналізує його за допомогою своєї мовної моделі ШІ, генерує відповідь та надсилає її назад до користувачького застосунку. Він підтримує розмовний контекст – тобто модель пам'ятає попередні запити й відповіді, якщо користувач надсилає їх у вигляді діалогу [41]. Зв'язок відбувається онлайн через хмару на стороні DeepSeek.

### 3.1.2 Структура вебзастосунку

Метод інтерактивного пошуку інформації на веб-сайті університету на основі генеративного штучного інтелекту реалізований у вигляді вебзастосунку, який складається із серверної та користувацької частин, а також шести логічних компонентів.

Серверна частина складається із трьох модулів: Agent, DataBase та Api (рис 3.2).

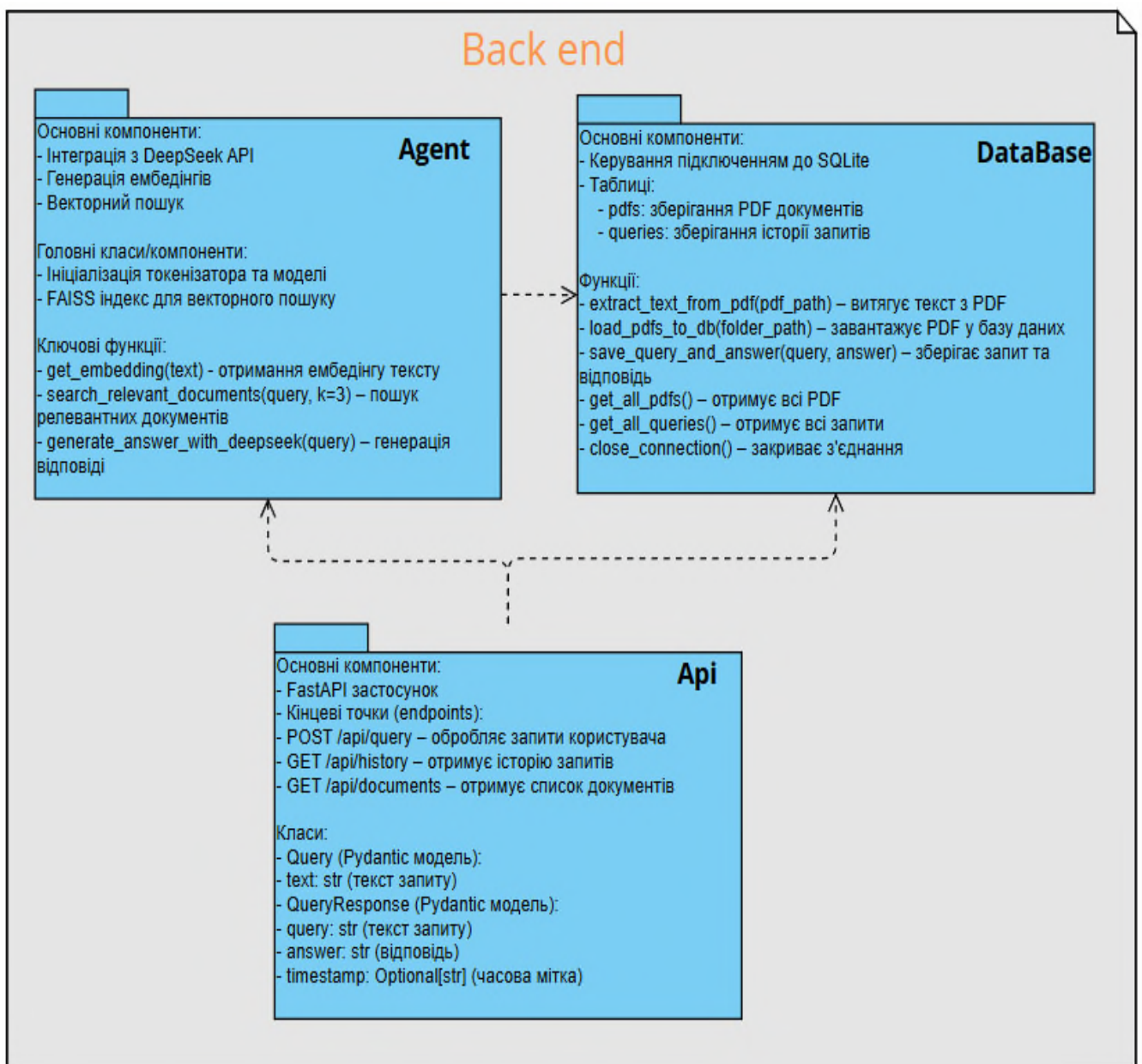


Рисунок 3.2 – Структура модулів серверної частини вебзастосунку

Модуль Agent відповідає за інтеграцію з DeepSeek API, створення ембеддингів та векторний пошук. У модулі використано клас Tokenizer з бібліотеки Transformers та бібліотеку FAISS. Основну логіку застосунку виконують три функції: `get_embedding()` – створює ембеддинг запиту, `search_relevant_documents()` – шукає релевантні документи, `generate_answer_with_deepseek()` – генерує відповідь користувачу.

Модуль DataBase керує підключенням до бази даних SQLite, а саме працює із таблицями `pdfs` – зберігання PDF документів та `queries` – зберігання історії запитів. Модуль містить функції `extract_text_from_pdf()` – витягує текст з PDF, `load_pdfs_to_db()` – завантажує PDF у базу даних, `save_query_and_answer()` – зберігає запит та відповідь, `get_all_pdfs()` – отримує всі PDF, `get_all_queries()` – отримує всі запити та `close_connection()` – закриває з'єднання.

Третій модуль серверної частини – API, використовуючи фреймворк FastAPI він опрацьовує запити користувача, повертає історію запитів та список PDF файлів. Модуль містить два класи `Query` – запит користувача та `QueryResponse` – відповідь на запит.

Клієнтська частина вебзастосунку пошуку інформації на вебсайті університету складається із трьох модулів: Chat, Type, App (рис. 3.3).

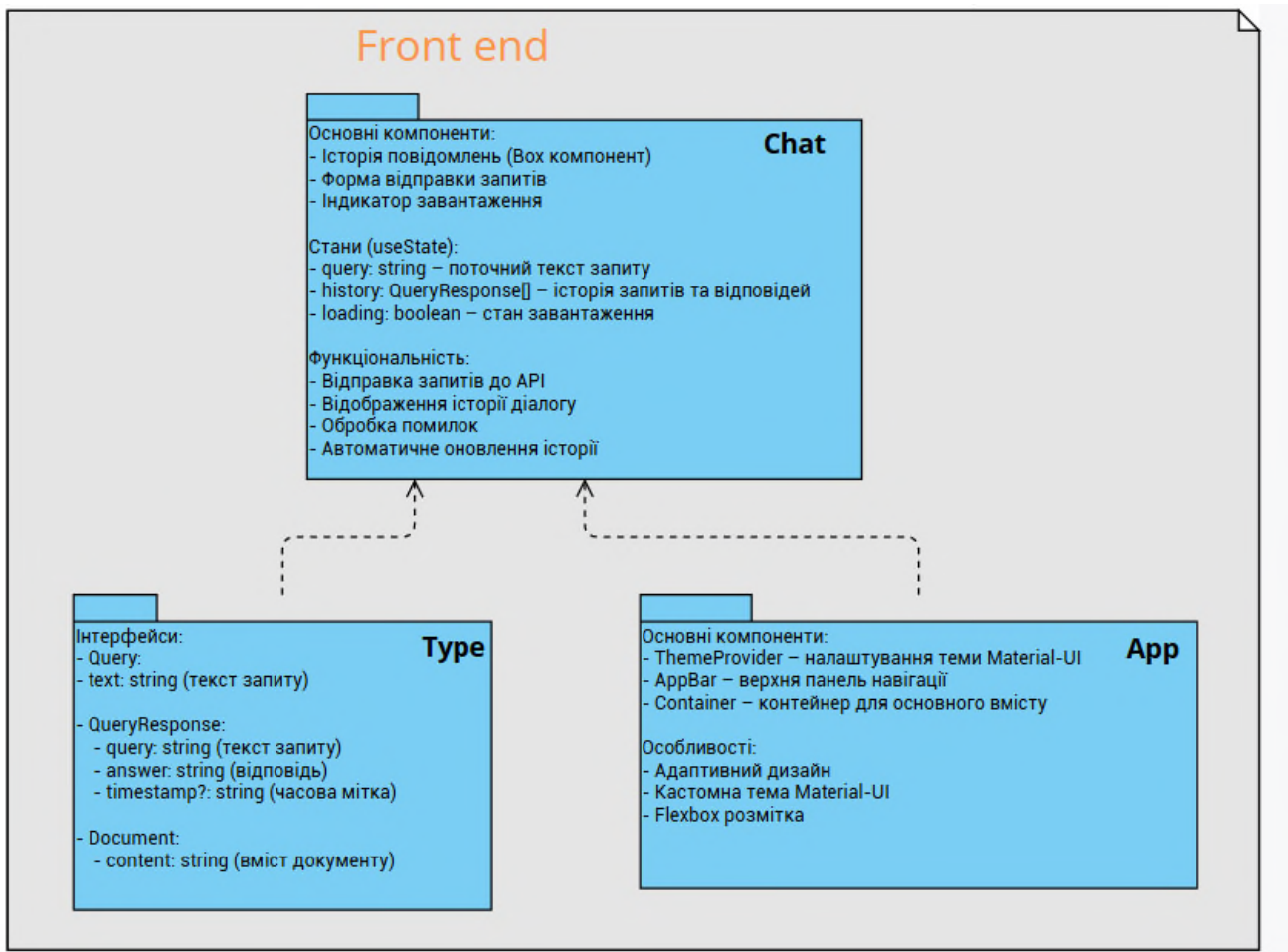


Рисунок 3.3 – Структура модулів клієнтської частини вебзастосунку

Модуль Chat відповідає за відправку запитів до модуля API з інтерфейсу вебзастосунку методу, відображення історії діалогу, обробку помилок та автоматичне оновлення історії. Основні компоненти модуля: історія повідомлень (QueryResponse[]), форма відправки запитів, індикатор завантаження (loading: boolean) та текст повідомлення (query: string).

Модуль Chat використовує DeepSeek Api – це реалізація нейронної мережі трансформерного типу, яка перетворює вхідний текст на ймовірнісний розподіл наступних токенів. Весь текст розбивається на токени – фрагменти тексту, тобто слова, частини слів чи символи. Кожен токен перетворюється у вектор – векторне представлення у великому багатовимірному просторі (рис 3.4).

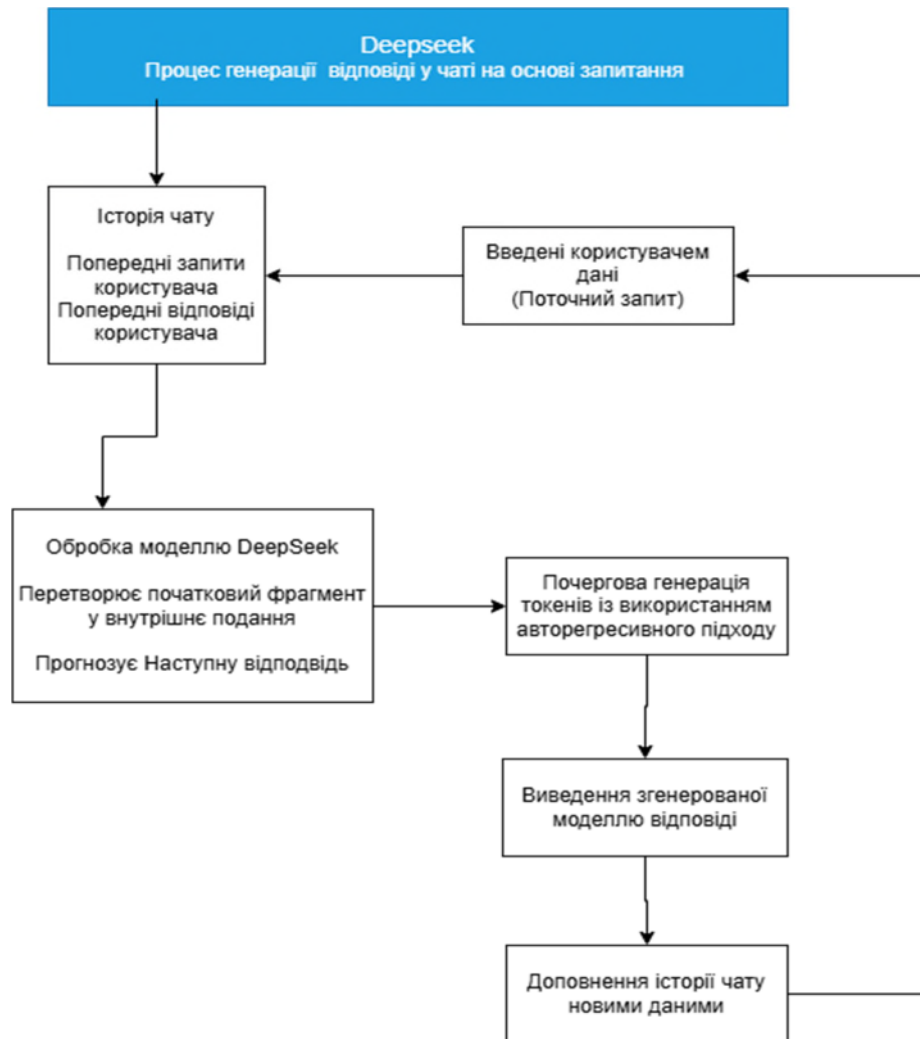


Рисунок 3.4 – Складові роботи DeepSeek Api[42]

На наступному кроці DeepSeek використовує архітектуру трансформера, яка складається з багатьох шарів із механізмом самоуваги і повнозв'язними шарами. На кожному шарі обчислюється – скільки уваги приділяти кожному попередньому токenu. Всі токени оновлюють свої векторні представлення відповідно до контексту.

Після проходження через трансформери, модель отримує розподіл ймовірностей та вибирає наступний токен (одним із методів : семплінгу, top-k, top-r або іншими) і додає його до тексту. Лише після цього модель навчається за допомогою максимізації правдоподібності. На етапі навчання модель визначає, наскільки її передбачення відрізняються від справжніх наступних токенів у тренувальних даних.

DeepSeek Api, на відміну від базових трансформерів, використовує MoE – для кожного токена активується лише частина «експертів» (модулів), що економить обчислювальні ресурси. З точки зору математичної моделі – це величезна функція яка трансформує вхідні токени у відповідь, знаходячи наймовірніші наступні слова з урахуванням контексту та статистичних закономірностей з даних, на яких вона навчена.

Модуль Туре складається із трьох інтерфейсів: Query – дані про запит користувача, QueryResponse – відповідь на запит користувача та Document – вміст документу.

Модуль App відповідає за налаштування інтерфейсу користувача, а саме адаптивний дизайн, кастомну тему Material-UI та Flexbox розмітку. Основні компоненти модуля: ThemeProvider – налаштування теми Material-UI, AppBar – верхня панель навігації та Container – контейнер для основного вмісту.

### **3.2 Результати тестування методу пошуку інформації**

Для тестування роботи методу пошуку інформації на вебсайті університету обрано запити, які охоплюють типові питання від користувачів.

Процес тестування відбувається наступним чином:

1. Експерт вводить запитання.
2. Метод пошуку генерує відповідь на поставлене запитання.
3. Експерт порівнює отриману відповідь із еталонною відповіддю та виставляє оцінку у межах від 0 до 5 для кожного критерію.

Також для всіх запитів автоматично проводиться тестування метрикою ROUGE.

Всього для оцінки методу інтерактивного пошуку інформації на вебсайті університету відібрано 20 запитань. З них розглянемо 3 запитання:

1. Скільки коштує навчання студента 3 курсу спеціальності «Комп'ютерні науки?»
2. Скільки всього є факультетів у ХНУ?

### 3. Які умови вступу на спеціальність Комп'ютерна інженерія в ХНУ?

Для написання тестових методів мовою програмування python використано бібліотеку rouge\_score та функцію generate\_answer\_with\_deepseek. Функція generate\_answer\_with\_deepseek використовується для генерації відповіді з використання DeepSeek Api.

Тестовий запит 1. Користувач вводить запит: Скільки коштує навчання студента 3 курсу спеціальності «Комп'ютерні науки?»

Надсилання запиту користувачем та отримання відповіді від застосунку продемонстровано на рисунку 3.5.

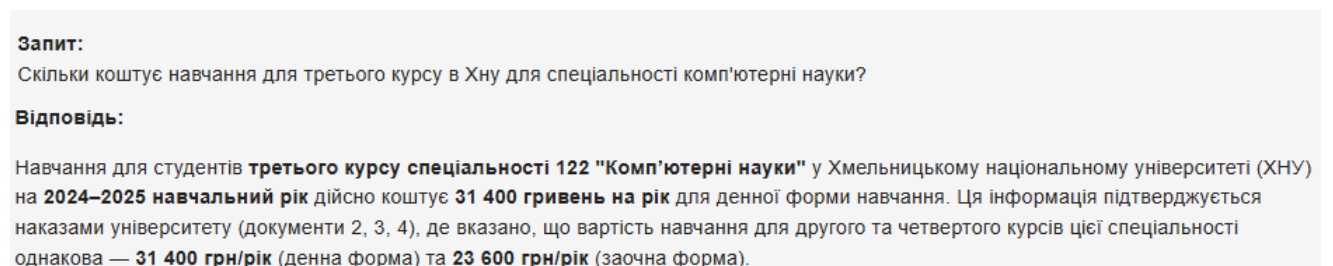


Рисунок 3.5 – Згенерована методом пошуку інформації відповідь на запит про вартість навчання

Експертна оцінка роботи методу на прикладі запиту про вартість навчання зібрана у таблиці 3.1.

Таблиця 3.1 – Оцінювання роботи методу на прикладі запиту про вартість навчання

Критерії	Коментар	Оцінка експерта
Чи знайдено інформація у відповідному джерелі.	Так, інформація знайдена у відповідному джерелі	5 балів
Чи є відповідь релевантною запиту.	Згадка про назву та код спеціальності а також рік та ціна навчання якого цікавила.	4 бали

Чи збережено повноту та точність змісту.	Повноту змісту збережено а також додано уточнення про можливу зміну вартості навчання	5 балів
Чи витягнуто вміст з pdf-файла без викривлення	Вартість навчання, код та назва спеціальності точно відповідає описаній в з pdf-файлі	4 бали

Метод інтерактивного пошуку інформації на вебсайті університету сформулював коректну відповідь на поставлене запитання.

Результати тестування метрикою ROUGE для тестового запиту 1 зібрано у таблиці 3.2.

Таблиця 3.2 – Оцінка роботи методу пошуку метрикою ROUGE для тестового запиту 1

ROUGE-1 F1	ROUGE-1 Precision	ROUGE-1 Recall	ROUGE-L F1	ROUGE-L Precision	ROUGE-L Recall
0,807692308	0,818181818	1	0,707692308	0,881818182	1

Середнє значення тестування метриками ROUGE F1 для першого тестового запиту –0,76, середнє значення експертної оцінки – 0,9.

Тестовий запит 2. Користувач вводить запит: Скільки всього є факультетів у ХНУ?

Надсилання запиту методу та отримання відповіді зображено на рисунку 3.6.

**Запит:**  
скільки всього є факультетів в ХНУ?

**Відповідь:**  
У Хмельницькому національному університеті (ХНУ) є **8 факультетів**. Ось їхній перелік:

1. Гуманітарно-педагогічний факультет
2. Факультет здоров'я, психології, фізичної культури та спорту
3. Факультет міжнародних відносин і права
4. Факультет економіки і управління
5. Факультет управління, адміністрування та туризму
6. Факультет інформаційних технологій
7. Факультет інженерії, транспорту та архітектури
8. Факультет технологій і дизайну

Ця інформація підтверджується документами університету, зокрема наказами та освітніми програмами.

26.05.2025, 22:34:58

Ask a question... SEND >

Рисунок 3.6 – Згенерована методом пошуку відповідь про кількість факультетів  
Експертна оцінка роботи методу на прикладі запиту про кількість факультетів у Хмельницькому національному університеті зібрана у таблиці 3.3.

Таблиця 3.3 – Оцінювання роботи методу на прикладі запиту про кількість факультетів

Критерії	Коментар	Оцінка експерта
Чи знайдено інформація у відповідному джерелі.	Для запиту знайдено три релевантних документи, які відповідають критеріям	5 балів
Чи є відповідь релевантною запиту.	У відповіді перераховано всі факультети та їх повні назви.	5 балів
Чи збережену повноту та точність змісту.	Точність та повноту змісту збережено.	5 балів
Чи витягнуто вміст з pdf-файла без викривлення	Перелік факультетів отримано з pdf-файлів без викривлень.	5 балів

Метод інтерактивного пошуку інформації на вебсайті університету сформулював коректну відповідь на запитання про перелік та кількість факультетів в Хмельницькому національному університеті.

Результати тестування метрикою ROUGE для тестового запиту 2 зібрано у таблиці 3.4.

Таблиця 3.4 – Оцінка роботи методу пошуку метрикою ROUGE для тестового запиту 2

ROUGE-1 F1	ROUGE-1 Precision	ROUGE-1 Recall	ROUGE-L F1	ROUGE-L Precision	ROUGE-L Recall
0,818182	0,798768	1	0,818182	0,798679	1

Середнє значення тестування метриками ROUGE F1 для другого тестового запиту – 0,82, середнє значення експертної оцінки – 1.

Тестовий запит 3. Користувач вводить запит: Які умови вступу на спеціальність Комп'ютерна інженерія в ХНУ?

Фрагмент роботи методу пошуку інформації зображено на рисунку 3.7.

**Запит:**  
Які умови вступу на спеціальність Комп'ютерна інженерія в ХНУ?

**Відповідь:**

**Умови вступу на спеціальність "Комп'ютерна інженерія" (123) у Хмельницькому національному університеті (ХНУ) у 2025 році:**

**1. Освітній рівень:**

- Після 11 класів (ПЗСО) — на базі **бакалаврату**.
- Результати **НМТ** (2022–2025 роки) з обов'язкових предметів:
  - Українська мова.
  - Математика.
  - Історія України.
  - Додатковий предмет на вибір (фізика, іноземна мова, хімія тощо).

**2. Конкурсний бал:**

- **Держзамовлення:** ≥130 балів.
- **Контракт:** ≥100 балів.

**3. Документи:**

- Заява (через [електронний кабінет](#)).
- Копія атестата та додатку.
- Результати НМТ.
- Паспорт, ІПН, військовий облік (для чоловіків).
- 6 фото 3×4 см.
- Мотиваційний лист (1–2 сторінки).

**4. Вартість навчання (2024–2025):**

- **Денна форма:** 26 700 грн/рік.
- **Заочна форма:** 17 300 грн/рік.

26.05.2025, 22:50:03

Рисунок 3.7 – Робота методу на прикладі питання про умови вступу

Оцінка роботи методу на прикладі запиту про умови вступу на спеціальність Комп'ютерна інженерія в ХНУ зібрана у таблиці 3.5.

Таблиця 3.5 – Оцінка роботи методу на прикладі запиту про умови вступу

Критерії	Коментар	Оцінка експерта
Чи знайдено інформація у відповідному джерелі.	Для запиту знайдено два релевантних документи, які відповідають критеріям	5 балів
Чи є відповідь релевантною запиту.	Відповідь охоплює опис алгоритму вступу, мінімальні прохідні бали, вартості навчання на різні форми навчання.	4 бали
Чи збережену повноту та точність змісту.	Точність та повноту змісту збережено, інформація подана у надлишковому обсязі.	4 бали
Чи витягнуто вміст з pdf-файла без викривлення	Інформацію про вступ отримано з pdf-файлів без викривлень.	5 балів

Метод інтерактивного пошуку інформації на вебсайті університету сформулював коректну відповідь на запитання про умови вступу на факультет Комп'ютерна інженерія, а також надав надлишкову інформацію.

Результати тестування метрикою ROUGE для тестового запиту 3 зібрано у таблиці 3.6.

Таблиця 3.6 – Оцінка роботи методу пошуку метрикою ROUGE для тестового запиту 3

ROUGE-1 F1	ROUGE-1 Precision	ROUGE-1 Recall	ROUGE-L F1	ROUGE-L Precision	ROUGE-L Recall
0,942857	0,766038	1	0,807143	0,778038	1

Середнє значення тестування метриками ROUGE F1 для третього тестового запиту –0,87, середнє значення експертної оцінки – 0,9.

Аналогічним чином проведено тестування для всіх 20 тестових запитів. Дані про середню експертну оцінку та середню оцінку метрикою ROUGE для 20 запитів подані на рисунку 3.8.

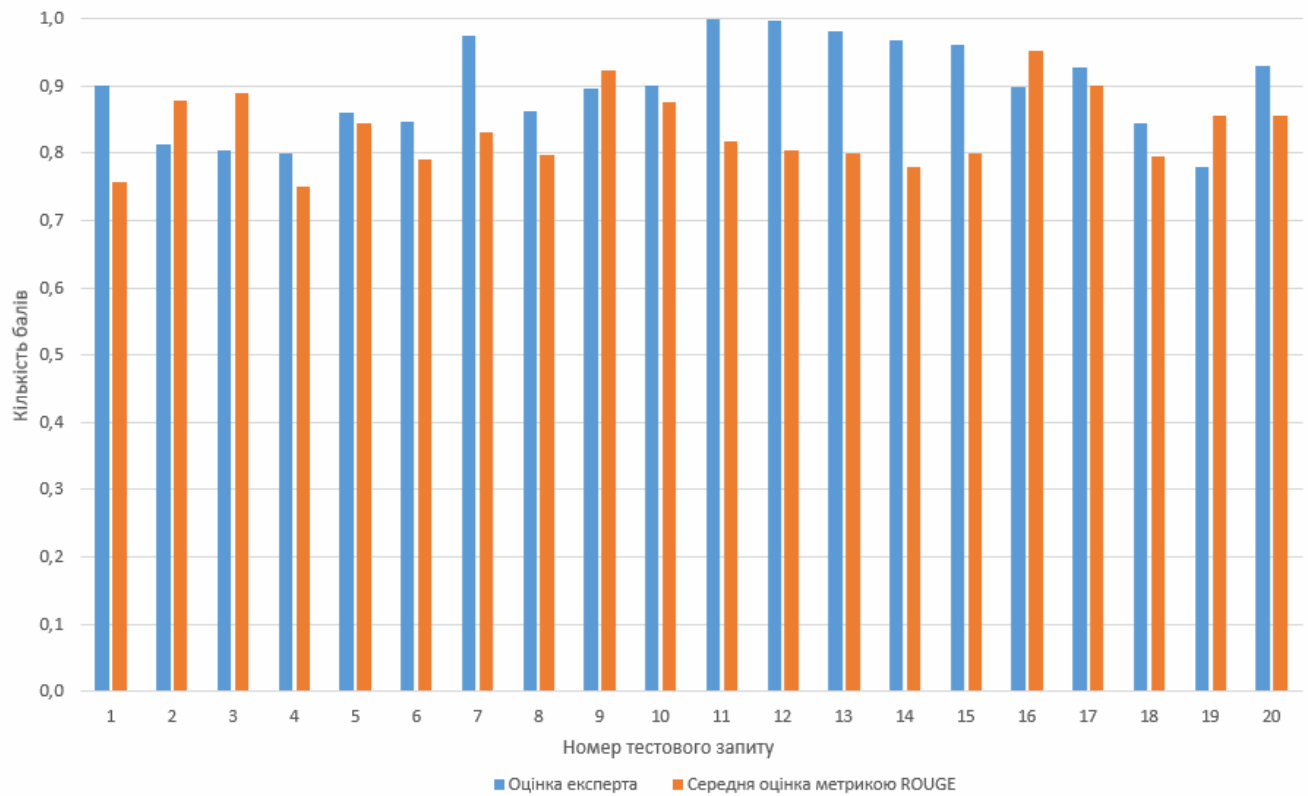


Рисунок 3.8 – Зведена оцінка якості роботи методу пошуку інформації на основі 20 тестових запитів

Середні оцінки метрикою ROUGE для 20 питань зібрано у таблиці 3.7.

Таблиця 3.7 – Середні оцінки метрикою ROUGE для 20 запитань

ROUGE-1 F1	ROUGE-1 Precision	ROUGE-1 Recall	ROUGE-L F1	ROUGE-L Precision	ROUGE-L Recall
0,8530827	0,7988539	0,99788182	0,82086173	0,82724909	0,97272727

Проведено тестування методу пошуку інформації двома способами – експертним оцінюванням та за допомогою метрики ROUGE. Проведені тести показали, що метод швидко та ефективно повертає релевантні відповіді на

поставлені запитання, середня оцінка метрикою ROUGE – 0,83, середня оцінка експерта – 0,9. Отримані результати свідчать про те, що метод пошуку інформації на вебсайті університету демонструє хорошу продуктивність у генерації контенту, а також вказують на ефективність системи в цілому.

### 3.3 Висновки до розділу 3

Метод інтерактивного пошуку інформації на вебсайті університету на основі генеративного штучного інтелекту складається із серверної та клієнтської частини. Для розробки серверної частини використано мову програмування python та фреймворк FastAPI. Вбудована реляційна база даних SQLite виконує функції зберігання документів та історії запитів. Для роботи з PDF-документами використано бібліотеку PyMuPDF (fitz), а для роботи із з трансформерними моделями – Transformers (Hugging Face). Роль векторної бази даних для пошуку та збереження векторних ембеддингів виконує бібліотека FAISS. Генерацією відповіді на основі контенту займається DeepSeek API.

Клієнтську частину розроблено на основі бібліотек React.js та TypeScript. Для створення інтерфейсу використано компонент React.js – Material-UI. Для HTTP-запитів бібліотека Axios.

Оцінка ефективності методу інтерактивного пошуку інформації на вебсайті університету проведена двома способами: експериментальною перевіркою релевантності згенерованих відповідей та за допомогою метрик ROUGE, зокрема ROUGE-1 і ROUGE-L, шляхом порівняння автоматично згенерованого тексту з еталонним.

Для більшості запитань вебзастосунок надав користувачам повні та релевантні відповіді, іноді надлишкові. Інформація отримується з pdf-файлів без викривлень. Для всіх тестів метрикою ROUGE отримані відповіді говорять про помірне співпадіння між отриманими та очікуваними результатами. Середня оцінка метрикою ROUGE F1 – 0,83, середня оцінка експерта – 0,9.

Розроблений вебзастосунок повністю відповідає поставленим завданням.

## Загальні висновки

Метою кваліфікаційної роботи бакалавра є підвищення ефективності пошуку інформації на вебсайті університету шляхом застосування генеративного штучного інтелекту.

Створений вебзастосунок на основі методу інтерактивного пошуку дозволяє значно спростити отримання інформації для студентів, викладачів та адміністрації університету. Генеративний штучний інтелект забезпечує швидку та релевантну видачу відповідей, адаптованих до природної мови запитів користувачів, що підвищує ефективність використання вебсайту Хмельницького національного університету.

Для досягнення поставленої мети виконано наступні завдання:

- Досліджено існуючі методи та технології інтерактивного пошуку інформації.
- Розроблено метод інтерактивного пошуку інформації на вебсайті університету з використанням генеративного штучного інтелекту.
- Реалізовано вебзастосунок для забезпечення ефективного інтерактивного пошуку інформації.
- Проведено тестування ефективності методу інтерактивного пошуку інформації за допомогою експертної оцінки та метрик ROUGE.

На основі поставлених завдань розроблено вебзастосунок, який дозволяє здійснювати пошук на вебсайті Хмельницького національного університету. Інтерфейс користувача зручний, функціональний, адаптивний, та забезпечує ефективний і швидкий доступ до інформації

Для генерації відповідей використано велику мовну модель та технологію RAG для розширення можливостей великих мовних моделей за допомогою зовнішньої бази знань. Донавчання LLM відбувалося на основі pdf-файлів, які містять інформацію про діяльність університету. Файли перетворюються у векторні представлення та зберігаються у векторній базі даних за допомогою вбудованої технології ембеддингу.

Векторна бібліотека FAISS забезпечує швидкий та ефективний пошуку інформації у векторній базі даних, шляхом порівняння її векторних представлень та знаходження одного та декількох найближчих векторів.

Розроблений вебзастосунок складається із серверної та клієнтської частини. Серверну частину написано мовою Python на основі фреймворку FastAPI. Вбудована база даних SQLite зберігає документи та історії запитів. Для роботи з pdf -документами використано бібліотеку PyMuPDF, а для роботи із з трансформерними моделями – бібліотеку Transformers. Роль векторної бази даних для пошуку та збереження векторних ембеддингів виконує бібліотека FAISS. Генерацією відповідей на основі контенту займається DeepSeek API.

Оцінка методу інтерактивного пошуку інформації на веб-сайті університету проводилася двома способами: експериментальною перевіркою релевантності згенерованих відповідей та за допомогою метрик ROUGE, зокрема ROUGE-1 і ROUGE-L, шляхом порівняння автоматично згенерованого тексту з еталонним. Для більшості запитань вебзастосунок надав користувачам повні відповіді, іноді надлишкові. Інформація отримується з pdf-файлів без викривлень. Проведені тести показали, що метод швидко та ефективно повертає релевантні відповіді на поставлені запитання, середня оцінка метрикою ROUGE F1 – 0,83, середня оцінка експерта – 0,9. Отримані результати свідчать про те, що метод пошуку інформації на вебсайті Хмельницького національного університету демонструє хорошу продуктивність у генерації контенту, а також вказують на ефективність системи в цілому.

У процесі подальшого використання базу даних розробленого вебзастосунку можна розширювати шляхом додавання нових та актуальних pdf-файлів. Ще одним вектором подальшого розвитку розробленого вебзастосунку є розміщення його на постійному хостингу.

## Перелік посилань

1. Хрiстiян О. Інформація як чинник розвитку суспільних відносин: Кат. вист. Харків : держ. наук. б-ка ім. В. Г. Короленка, 2015. 51 с.
2. Browsing the web. Mdn web docs. URL: [https://developer.mozilla.org/en-US/docs/Learn\\_web\\_development/Getting\\_started/Environment\\_setup/Browsing\\_the\\_web](https://developer.mozilla.org/en-US/docs/Learn_web_development/Getting_started/Environment_setup/Browsing_the_web).
3. Searching the internet. <https://uq.pressbooks.pub/>. URL: <https://uq.pressbooks.pub/digital-essentials-information-essentials/chapter/searching-the-internet/>.
4. Меннінг, К. Д., Рагде, П., Шютце, Г. Вступ до інформаційного пошуку. Кембридж: Cambridge University Press, 2008. 496 с.
5. Павленко, Ю. С. Пошукова оптимізація, технології та сервіси вебаналітики: Конспект лекцій. Луцьк: Волинський національний університет імені Лесі Українки, 2022. 51 с.
6. Advancements Changing the World of Search. <https://builtin.com/>. URL: <https://builtin.com/artificial-intelligence/search-technology-advancements>.
7. The AI Web Search Landscape. <https://sophiamyang.medium.com/>. URL: <https://sophiamyang.medium.com/the-ai-web-search-landscape-5b4ad0e67505>.
8. Gemini can now personalize its answers based on your search history. Theverge. URL: <https://www.theverge.com/news/629022/gemini-google-search-history-personalization>.
9. What is GenAI? Generative AI explained. Techtarget. URL: <https://www.techtarget.com/searchenterpriseai/definition/generative-AI>.
10. How AI and Machine Learning Are Revolutionizing Information Retrieval. Coveo. URL: <https://www.coveo.com/blog/ai-information-retrieval/>.
11. Mapp B. Generative AI in Practice. Wiley, 2024. 320 с.
12. Allamap Dj., Grootendorts M. Hands-On Large Language Models: Language Understanding and Generation. O'Reilly Media, 2024. 320 с.

13. What is a large language model (LLM)?. Cloudflare. URL: <https://www.cloudflare.com/learning/ai/what-is-large-language-model/>.
14. What are transformers in artificial intelligence?. AWS. URL: <https://aws.amazon.com/what-is/transformers-in-artificial-intelligence/>.
15. What Is a Transformer Model?. Nvidia. URL: <https://blogs.nvidia.com/blog/what-is-a-transformer-model/>.
16. What is a transformer model?. IBM. URL: <https://www.ibm.com/think/topics/transformer-model>.
17. Programmable Search Engine. Programmable Search Engine. URL: <https://developers.google.com/custom-search/v1/overview>
18. Implementing Google Custom Search Engine's JSON API in the JAMStack. Raymond Camden. URL: <https://www.raymondcamden.com/2020/03/22/implementing-google-custom-search-engines-json-api-in-the-jamstack>.
19. How to Use Google Custom Search Engine API in Python. The python code. URL: <https://thepythoncode.com/article/use-google-custom-search-engine-api-in-python>.
20. Introducing Perplexity Deep Research. Perplexity. URL: <https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research>.
21. What is Perplexity AI?. Techtarget. URL: <https://www.techtarget.com/searchenterpriseai/definition/Perplexity-AI>.
22. Overview of Glasp - The Social Web Highlighter for Collecting and Sharing Knowledge. Deepgram. URL: <https://deepgram.com/ai-apps/glasp>.
23. Highlight the Internet & Build Your AI Clone. Glasp. URL: <https://glasps.co/>.
24. A Beginner's Guide to Using Glasp. Dev. URL: <https://dev.to/steveororo/a-beginners-guide-to-using-glasp-2cgp.v>
25. You. Agents. URL: <https://you.com/>
26. You.com: AI-Powered Personalised Search Experience. Medium. URL: <https://medium.com/@MeteCevik/you-com-ai-powered-personalised-search-experience-efabd13add02>.

27. You.com: An AI-Powered Productivity Engine Enhancing Knowledge Work with Pinecone. Pinecone. URL: <https://www.pinecone.io/customers/you-com/>.

28. 10 безкоштовних інструментів на основі штучного інтелекту для пошуку інформації. Бібліотека БДПУ. URL: <https://library.bdpu.org.ua/ai-for-education-and-research/artificial-intelligence-for-information-retrieval/>.

29. All you need to know to Develop using Large Language Models. Towards Data Science. URL: <https://towardsdatascience.com/all-you-need-to-know-to-develop-using-large-language-models-5c45708156bc/>.

30. Large language models in medicine: the potentials and pitfalls. *Researchgate*. URL: [https://www.researchgate.net/figure/Overview-of-LLM-training-process-LLMs-learn-from-more-focused-inputs-at-each-stage-of\\_fig1\\_373642018](https://www.researchgate.net/figure/Overview-of-LLM-training-process-LLMs-learn-from-more-focused-inputs-at-each-stage-of_fig1_373642018).

31. Великі мовні моделі (LLM): Повний посібник у 2025 році. *Shaip*. URL: <https://uk.shaip.com/blog/a-guide-large-language-model-llm>

32. APA Style James, A., Trovati, M., & Bolton, S. (2025). Retrieval-Augmented Generation to Generate Knowledge Assets and Creation of Action Drivers. *Applied Sciences*, 15(11), 6247.

33. DeepSeek Chat. *Spring*. URL: <https://docs.spring.io/spring-ai/reference/api/chat/deepseek-chat.html>.

34. What are embeddings in machine learning?. *Cloudflare*. URL: <https://www.cloudflare.com/learning/ai/what-are-embeddings/>.

35. Step-by-Step Guide to Choosing the Best Embedding Model for Your Application. *Weaviate*. URL: <https://weaviate.io/blog/how-to-choose-an-embedding-model>.

36. Optimizing RAG: A Guide to Choosing the Right Vector Database. *Medium*. URL: <https://medium.com/@mutahar789/optimizing-rag-a-guide-to-choosing-the-right-vector-database-480f71a33139>.

37. Which Industries Use DeepSeek. *Deepseek*. URL: <https://deepseek.seesaa.net/article/510366403.html>.

38. End-to-End Neural Embedding Pipeline for Large-Scale PDF Document Retrieval Using Distributed FAISS and Sentence Transformer Models / B. Chandra та ін. *ResearChgate*. 2024. c. 12.

39. How to perform High-Performance Search using FAISS. *Medium*. URL: <https://kharepratyush.medium.com/how-to-perform-high-performance-search-using-faiss-da2ab12f606c>.

40. Implementing RAG to build a “Chat with Multiple PDFs” app. *Medium*. URL: <https://blog.gopenai.com/lab-3-implementing-rag-to-build-a-chat-with-multiple-pdfs-app-88c1d7cd5d19>

41. Faiss. *Langchain*. URL: <https://python.langchain.com/docs/integrations/vectorstores/faiss/>.

42. Neha F., Bhati D. A Survey of DeepSeek Models. *TechRxiv*. 2025.

# ДОДАТКИ

## Додаток А

### Програмний Код

Посилання на репозиторій з програмним кодом розробленого методу та застосування: <https://github.com/Sansmmr/KRB/tree/master>

Sansmmr Update README.md		5c0deaf · 36 minutes ago	3 Commits
PDFS	first commit		last week
__pycache__	first commit		last week
data	first commit		last week
frontend	first commit		last week
README.md	Update README.md		36 minutes ago
agent.py	first commit		last week
api.py	first commit		last week
asd.txt	first commit		last week
classes.png	first commit		last week
database.py	first commit		last week
packages.png	first commit		last week
query.py	first commit		last week
requirements.txt	first commit		last week
rouge_results.xlsx	first commit		last week
test_data.json	first commit		last week
test_pdf_content.py	first commit		last week
test_pdf_loading.py	first commit		last week
test_rouge.py	first commit		last week

Репозиторій містить 3 папки: PDFS, data, frontend. Директорія PDFS містить pdf-файли, необхідні для роботи системи RAG. Директорія data – бази даних, frontend – файли для роботи серверної частини.

Файл agent.py – серверна частина методу пошуку інформації, відповідає за пошук релевантних даних, створення контексту, використовує DeepSeek Api для генерації відповідей.

Файл api.py на основі фреймворку FastAPI виконує створення RestAPI. Файл api.py приймає текст запиту та повертає відповідь, а також повертає повну історію запитів та містить методи для тестування метрикою ROUGE.

Файл database.py містить функції для роботи з базою даних sqlite, а саме створення таблиць, збереження, отримання та очищення всіх історій запитів.

У файлі requirements.txt записані версії компонент методу пошуку інформації, які необхідні для його коректної роботи.

Проект включає тестові файли: test\_pdf\_loading.py – тести завантаження PDF-документів, test\_pdf\_content.py – тести вилучення контенту, test\_rouge.py – тести метрик якості відповідей.

## Додаток Б

### Презентаційний матеріал

# КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

## МЕТОД ПОШУКУ ІНФОРМАЦІЇ НА ВЕБСАЙТІ УНІВЕРСИТЕТУ З ДОПОВНЮЮЧИМ ІНФОРМАЦІЙНИМ ПОШУКОМ

**Виконав:** студент групи КН-21-1 Данило  
Твердохлібов

**Керівник:** к.т.н., доц. каф. КН Руслан БАГРІЙ

2

## Актуальність

Сучасна діяльність університетів регламентується численними положеннями, які зазвичай розміщені на офіційних вебсайтах у форматі PDF-документів. У таких умовах студентам, абітурієнтам та викладачам важко швидко знайти необхідну інформацію серед великої кількості файлів. Це ускладнює доступ до важливих даних, особливо коли потрібна оперативна відповідь. Тому реалізація методу пошуку, що генеруватиме точні відповіді природною мовою, матиме значний попит та покращить досвід взаємодії з університетським вебсайтом.

3

## Мета

Підвищення ефективності пошуку інформації на вебсайті університету шляхом застосування генеративного штучного інтелекту.

## Завдання

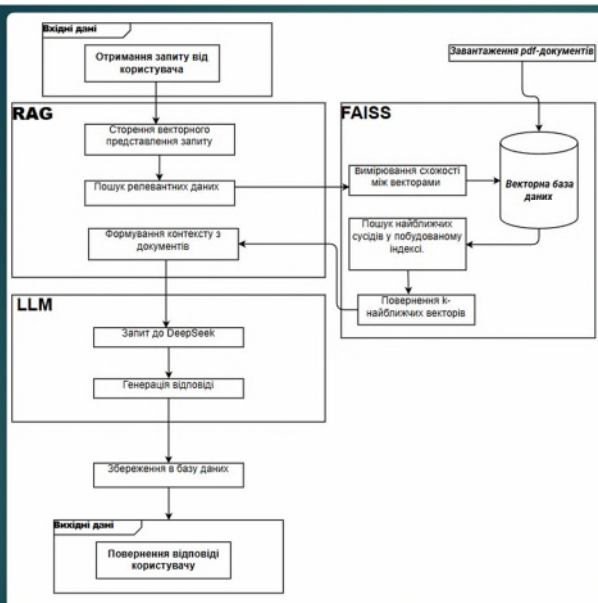
Дослідити існуючі методи та технології інтерактивного пошуку інформації.

Розробити метод інтерактивного пошуку інформації на вебсайті університету з використанням генеративного штучного інтелекту.

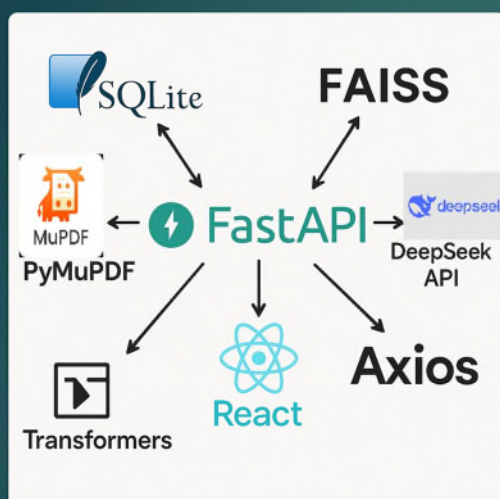
Реалізувати вебкомпоненти для забезпечення ефективного інтерактивного пошуку інформації.

Оцінити ефективність методу інтерактивного пошуку інформації за допомогою тестування та метрик якості.

4



Детальна схема роботи методу пошуку інформації на вебсайті



Інструменти для розробки методу пошуку інформації на вебсайті університету

## Інтерфейс вебзастосунку

RAG Chat Assistant CHAT HOW TO RESULTS

**Заява:**  
Які умови вступу на спеціальність Комп'ютерна інженерія в ХНУ?

**Відповідь:**  
Умови вступу на спеціальність "Комп'ютерна інженерія" (122) у Хмельницький національний університет (ХНУ) у 2025 році:

- Освітні вимоги:**
  - Освіта: повна загальна середня освіта (11 класів) або диплом молодшого бакалавра.
  - Вступні випробування:
    - НМТ (національний мультипредметний тест) з обов'язковими предметами:
      - Українська мова
      - Математика
      - Історія України
    - Додатковий предмет на вибір: фізика (рекомендовано для комп'ютерної інженерії)
    - Мінімальний бал:
      - 130 — для державного вступу
      - 100 — для контракту
- Документи:**
  - Заява (заповнюється онлайн)
  - Копія атестаційного та додатку
  - Результати НМТ
  - Копія паспорта та БТН
  - 4 фото 3x4 см
  - Мотиваційний лист (1-2 сторінки)
- Важливі дати:**
  - Реєстрація на НМТ: 06.03 – 03.04.2025
  - Подання заяв: 10.07 – 01.08.2025 через [вступний портал](#)
- Вартість навчання (2024-2025):**
  - Денна форма: 31 400 грн/рік
  - Заочна форма: 23 600 грн/рік
- Контакти:**
  - Адреса: м. Хмельницький, вул. Інститутська, 11
  - Телефон: +38 (067) 122-66-60
  - Емейл: [rk@hntu.edu.ua](mailto:rk@hntu.edu.ua)

**Деталі:** [сайт ХНУ](#)  
04.08.2025, 14:18:20

Ask a question SEND

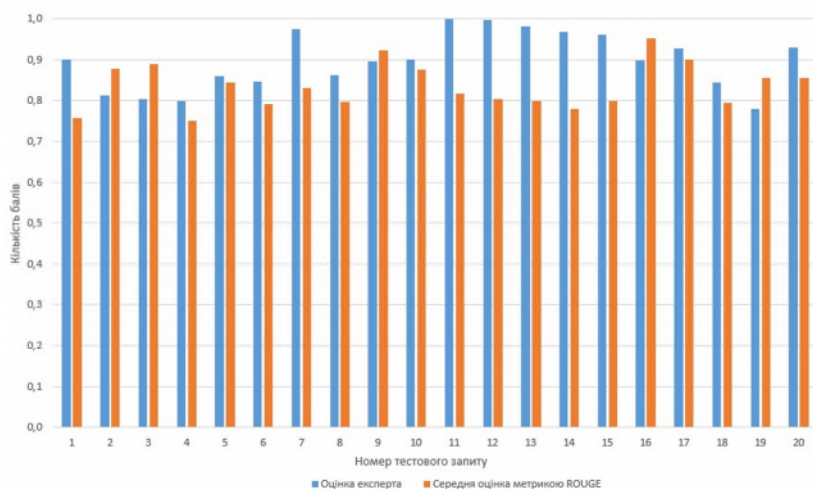
## Оцінювання якості роботи вебзастосунку

Процес тестування відбувався наступним чином:

1. Експерт вводить запитання.
2. Метод пошуку генерує відповідь на поставлене запитання.
3. Експерт порівнює отриману відповідь із еталонною відповіддю та виставляє оцінку у межах від 0 до 5 для кожного критерію.

Також для всіх запитів автоматично проводиться тестування метрикою ROUGE. Для кожного запиту визначалися показники ROUGE-1 F1, ROUGE-1 Precision, ROUGE-1 Recall, ROUGE-L F1, ROUGE-L Precision, ROUGE-L Recall.

Всього для оцінки методу інтерактивного пошуку інформації на вебсайті університету відібрано 20 запитань



Зведена оцінка якості роботи методу пошуку інформації на основі 20 тестових запитів

## Результати тестування

Проведено тестування методу пошуку інформації двома способами – експертним оцінюванням та за допомогою метрики ROUGE. Проведені тести показали, що метод швидко та ефективно повертає релевантні відповіді на поставлені запитання, середня оцінка метрикою ROUGE – 0,83, середня оцінка експерта – 0,9.

## Висновок

На основі поставлених завдань розроблено вебзастосунок, який дозволяє здійснювати пошук на вебсайті Хмельницького національного університету. Інтерфейс користувача вебзастосунку для роботи з методом пошуку інформації на основі генеративного штучного інтелекту зручний, функціональний та адаптивний, а також забезпечує ефективний і швидкий доступ до інформації.

У процесі подальшого використання бази даних розробленого вебзастосунку можна розширювати шляхом додавання нових та актуальних pdf-файлів. Ще одним вектором подальшого розвитку розробленого вебзастосунку є розміщення його на постійному хостингу.



Дякую за увагу

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Данило ТВЕРДОХЛБОВ

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод пошуку інформації на вебсайті університету з доповнюючим інформаційним пошуком

Науковий керівник: Руслан БАГРІЙ, к.т.н., доцент

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1:4.7%

Коефіцієнт подібності 2:2%

Мікропробіли: 1

Заміна букв: 2

Інтервали: 0

Білі знаки: 93

Дата створення звіту: 2025-06-06 02:45:53.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

Дата 06.06.2025

експерт

Ів. Резовський Р.С.

05.06.25, 17:13

Thu Jun 05 17:13:35 EEST 2025, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

# Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 3.0%

Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 13%

ID: 243746 Title: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод пошуку інформації на вебсайті університету з доповнюючим інформаційним пошуком Added in a DB: 2025-06-05 Authors: Данило ТВЕРДОХЛІБОВ Heads: Руслан БАГРІЙ Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	58955	865	4253 (7%)	67 (8%)

## Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК**

**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Назва кваліфікаційної роботи Метод пошуку інформації на вебсайті університету з деповіюючим інформаційним пошуком  
 Автор студент групи КН-21-1 Данило Твердохлібов  
 Освітня програма Комп'ютерні науки  
 Рівень вищої освіти перший (бакалаврський)  
 Спеціальність 122 – Комп'ютерні науки  
 Науковий керівник: к.т.н., доц. каф. комп'ютерних наук Руслан Багрій

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	<b>відповідає</b>
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	<b>відсутні</b>

**Підтвердження:**

*Запозичення, виявлені в роботі Данила Твердохлібова, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту.*

*Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:*

- за системою *Anti-Plagiarism*: 3%;

- за системою *StrikePlagiarism* КП1: 4.7%, КП2: 2%.

07.06.2025

Завідувач кафедри



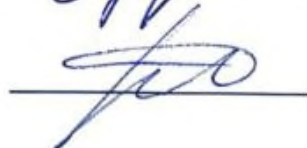
Олександр БАРМАК

Гарант освітньої програми



Олександр МАЗУРЕЦЬ

Керівник кваліфікаційної роботи



Руслан БАГРІЙ



**ВІДГУК НАУКОВОГО КЕРІВНИКА**  
**на кваліфікаційну роботу бакалавра**

студента гр. КН-21-1 Твердохлібова Данила Ігоровича  
за темою Метод пошуку інформації на веб-сайті університету з доповнюючим інформаційним пошуком

**1. Актуальність теми**

Актуальність теми обґрунтована потребою підвищення ефективності доступу до інформації на веб-сайті університету для студентів, абітурієнтів та викладачів. Особливістю теми є застосування технологій генеративного штучного інтелекту та пошуково-доповненої генерації для покращення пошукових можливостей.

**2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки**

Тема кваліфікаційної роботи "Метод пошуку інформації на веб-сайті університету з доповнюючим інформаційним пошуком" відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи бакалавра. Результатом роботи є розробка методу, що базується на генеративному ШІ, векторних базах даних і пошуково-доповненій генерації, що підвищує ефективність пошуку інформації на веб-сайті. При вирішенні поставлених завдань використано методи збору та аналізу даних, великі мовні моделі, технології векторного пошуку та методи розробки інформаційних систем.

**3. Професійні та особистісні якості бакалавра**

Твердохлібов Д. І. під час роботи над кваліфікаційною роботою продемонстрував розуміння теоретичних і практичних аспектів використання генеративного штучного інтелекту та векторних баз даних для створення ефективних пошукових систем.

**4. Ступінь самостійності під час виконання кваліфікаційної роботи**

Робота виконана самостійно, академічного плагіату не виявлено, усі запозичення оформлено з відповідними посиланнями на джерела.

**5. Ступінь оволодіння методами дослідження**

При реалізації кваліфікаційної роботи студент проявив високий рівень компетентності та володіння сучасними інструментами, методами й технологіями

комп'ютерних наук, зокрема з генеративного штучного інтелекту та пошуково-доповідної генерації.

**6. Повнота та якість розкриття теми роботи**

Тема роботи повністю розкрита: проведено аналіз актуальності, огляд існуючих методів, виконано всі поставлені задачі, а також розроблено концептуальну основу методу з можливістю подальшої реалізації для підтвердження його ефективності.

**7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу**

Викладення матеріалу логічне, послідовне та аргументоване. Мова і стиль роботи відповідають стандартам наукових текстів, забезпечуючи доступність сприйняття та відповідність вимогам кваліфікаційних робіт.

**8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин**

Запропонований метод пошуку інформації може бути застосований у веб-сайтах інших навчальних закладів, корпоративних порталах чи інформаційних системах, забезпечуючи швидкий і релевантний доступ до даних для користувачів.

**9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота**

Враховуючи високий рівень виконання, повноту розкриття теми та дотримання всіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка – **«відмінно»**.

Керівник \_\_\_\_\_



к.т.н., доц. Руслан Багрій



## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

студента *гр. КН-21-1 Твердохлібова Данила Ігоровича*  
за темою: *Метод пошуку інформації на вебсайті університету з доповнюючим інформаційним пошуком*

1. Актуальність обраної теми

*Сучасна діяльність університетів регламентується положеннями, розміщеними на вебсайтах у форматі PDF, що ускладнює швидкий пошук інформації для студентів, абітурієнтів та викладачів. Метод інтерактивного пошуку на основі генеративного штучного інтелекту покращить доступ до даних і взаємодію з вебсайтом університету.*

2. Повнота розкриття мети та завдань роботи

*Під час виконання кваліфікаційної роботи бакалавра був реалізований метод пошуку інформації на вебсайті університету з доповнюючим інформаційним пошуком, що відповідає меті та завданням кваліфікаційної роботи і розкриває їх повною мірою.*

3. Зміст кожного розділу роботи

*Записка кваліфікаційної роботи складається з трьох розділів. Перший розділ присвячено огляду технологій генеративного штучного інтелекту та аналізу сучасних рішень для пошуку інформації в Інтернеті, а також формулює постановку задачі. Другий розділ містить опис проєктування методу інтерактивного пошуку інформації для вебсайту університету. Третій розділ розглядає особливості реалізації вебзастосунку та результати тестування запропонованого методу пошуку.*

4. Оцінка розробленого методу та його практична цінність

*Розроблений метод, що використовує генеративний штучний інтелект здатен ефективно виконувати свою основну функцію – забезпечувати користувачам вебсайту університету швидкий і точний доступ до релевантної інформації у зручній, інтерактивній формі, що значно підвищує якість взаємодії з цифровим контентом.*

5. Якість оформлення кваліфікаційної роботи бакалавра

*Записка якісно оформлена відповідно до встановлених вимог, чітко і зрозуміло написана, зі структурованою побудовою розділів та логічною послідовністю викладення матеріалу.*

6. Недоліки кваліфікаційної роботи бакалавра

*Рекомендовано розглянути можливість покращення процесу оновлення векторної бази даних для забезпечення актуальності інформації.*

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

*Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка*

Рецензент \_\_\_\_\_

*Високо*  
*Твердохлібова Д.І.*