Розробка інтелектуальної системи підтримки користувача на основі Retrieval-Augmented Generation

Роботу виконав: Дяченко Максим Євгенійович
Науковий керівник: старший викладач, кандидит технічних наук Горборуков Вячеслав Вікторович

Актуальність теми, мета дослідження

З розвитком великих мовних моделей, все більше компаній та організацій впроваджують системи підтримки користувачів на їх основі, які забезпечують швидкий доступ до потрібної інформації і покращують взаємодію з користувачами. Одним із найперспективніших підходів є Retrieval-Augmented Generation (RAG), який дозволяє доповнювати запити пошуковими результатами.

Мета дослідження

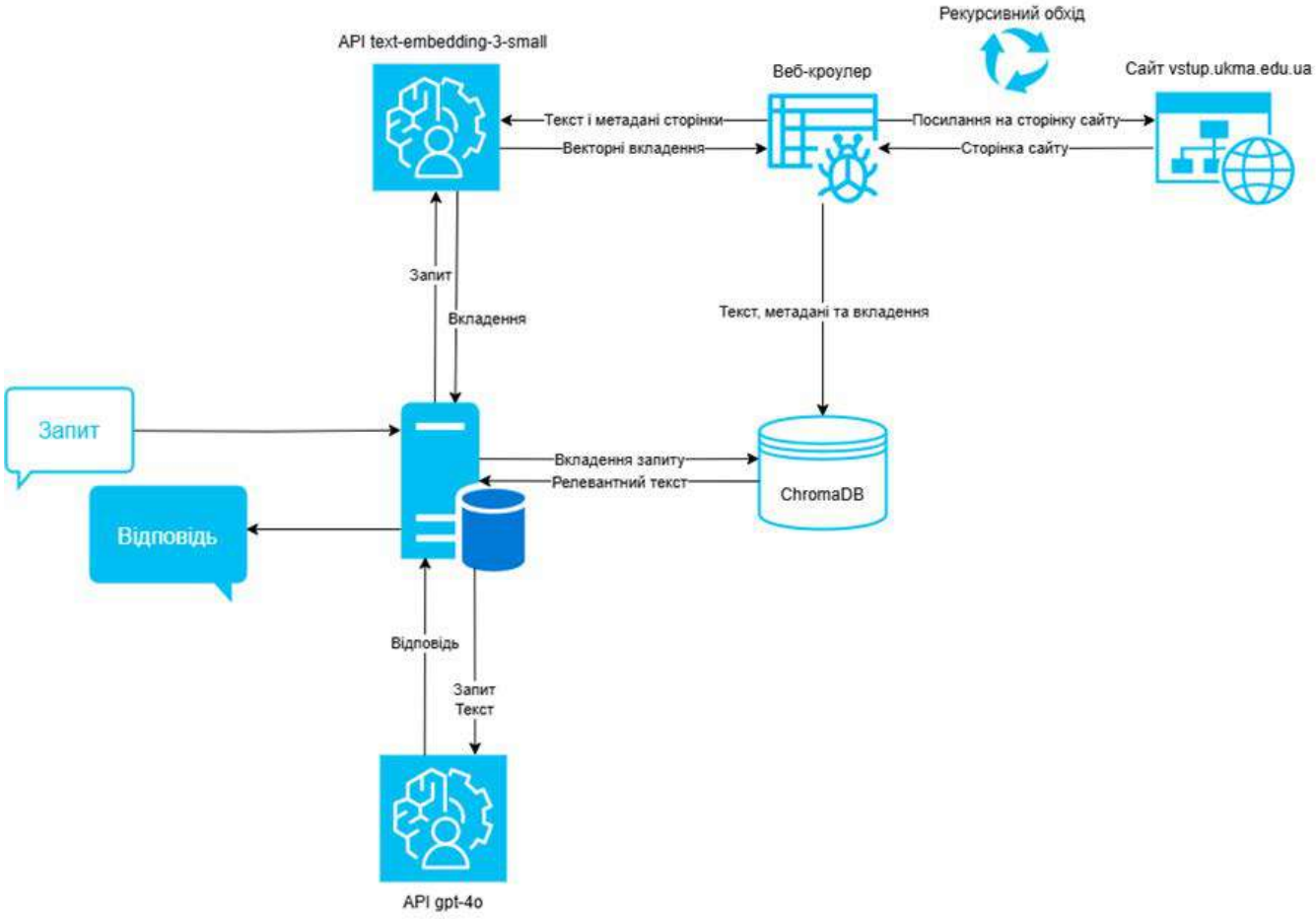
- Дослідити можливі практичні підходи в рамках RAG-системи, порівняти їх ефективність та обрати оптимальні для реального застосування.
- Створити застосунок, який дозволяє будувати векторні індекси веб-ресурсів та генерувати відповіді на запити користувачів.
- Розробити систему підтримки на основі сайту https://vstup.ukma.edu.ua/.

Використані технології

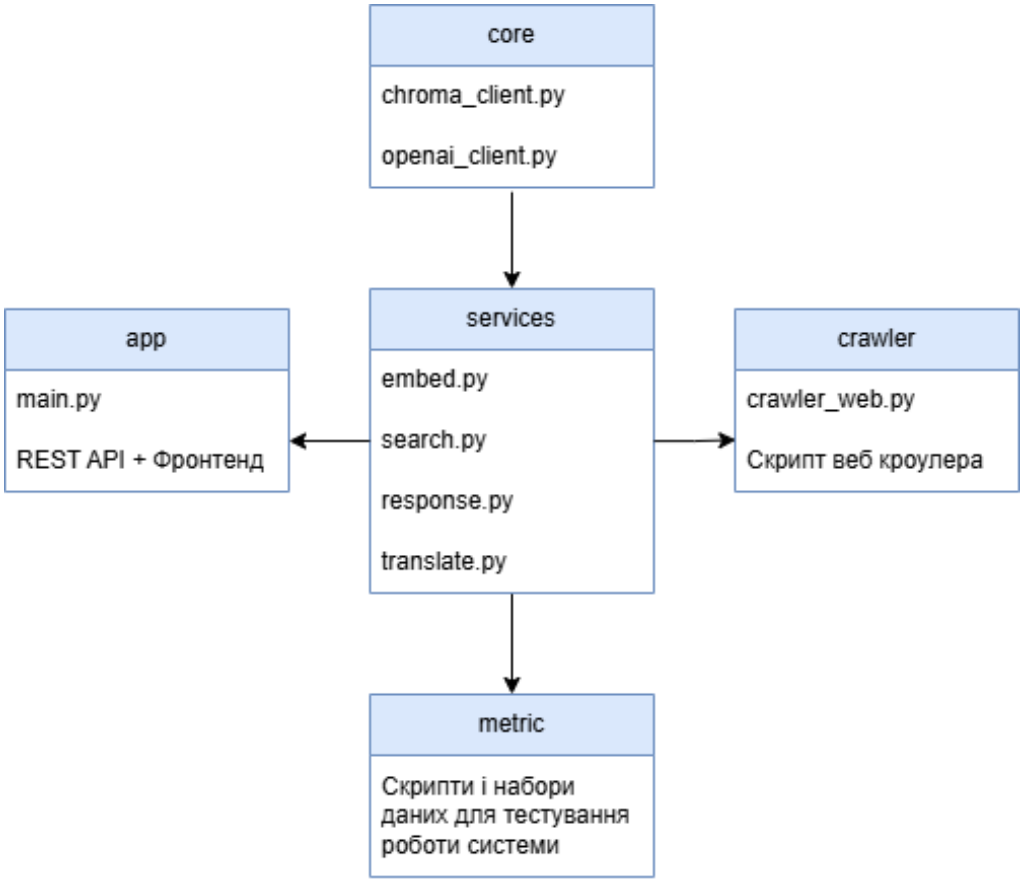
- Мова програмування: Python
- Векторна база даних: ChromaDB
- Моделі вкладень: text-embedding-3-small, text-embedding-3-large, bge-m3
- Генеративна модель: gpt-4o
- Веб-інтерфейс/API: Flask
- Збір даних з сайту: Selenium




Retrieval-Augmented Generation

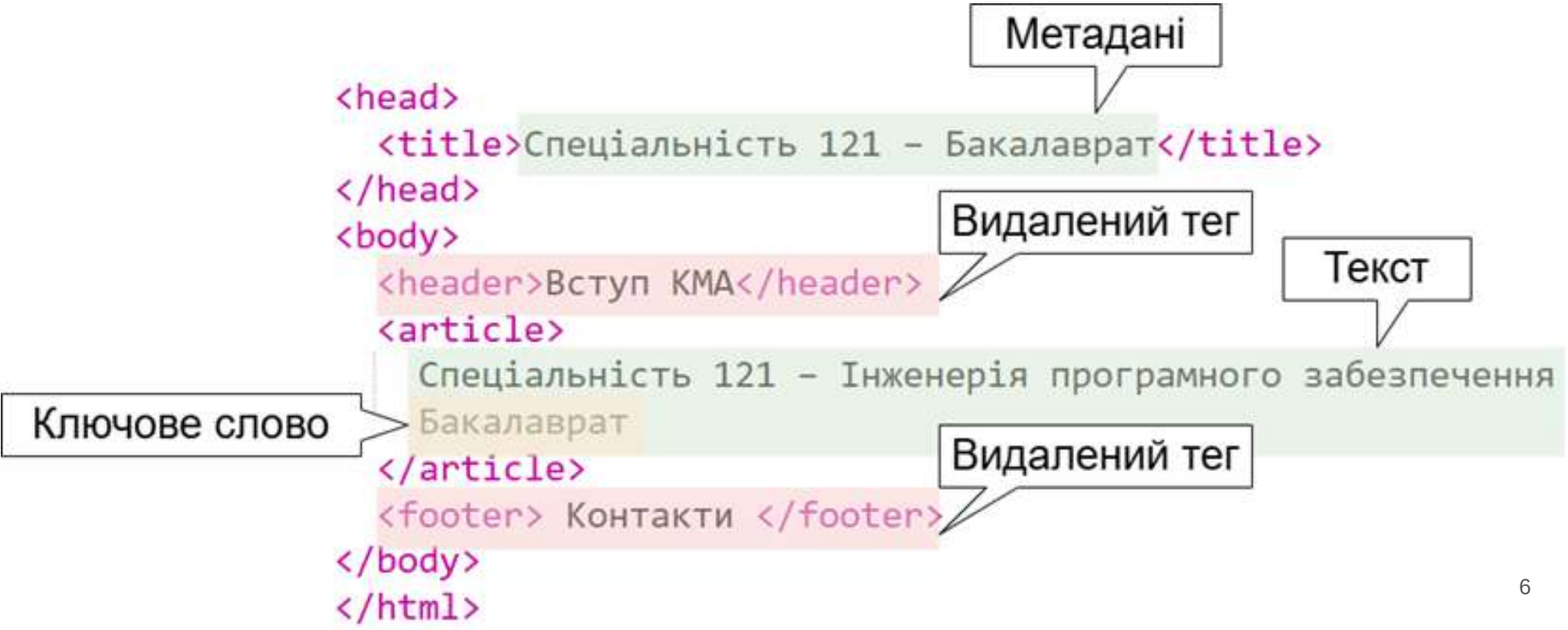


Структура проекту

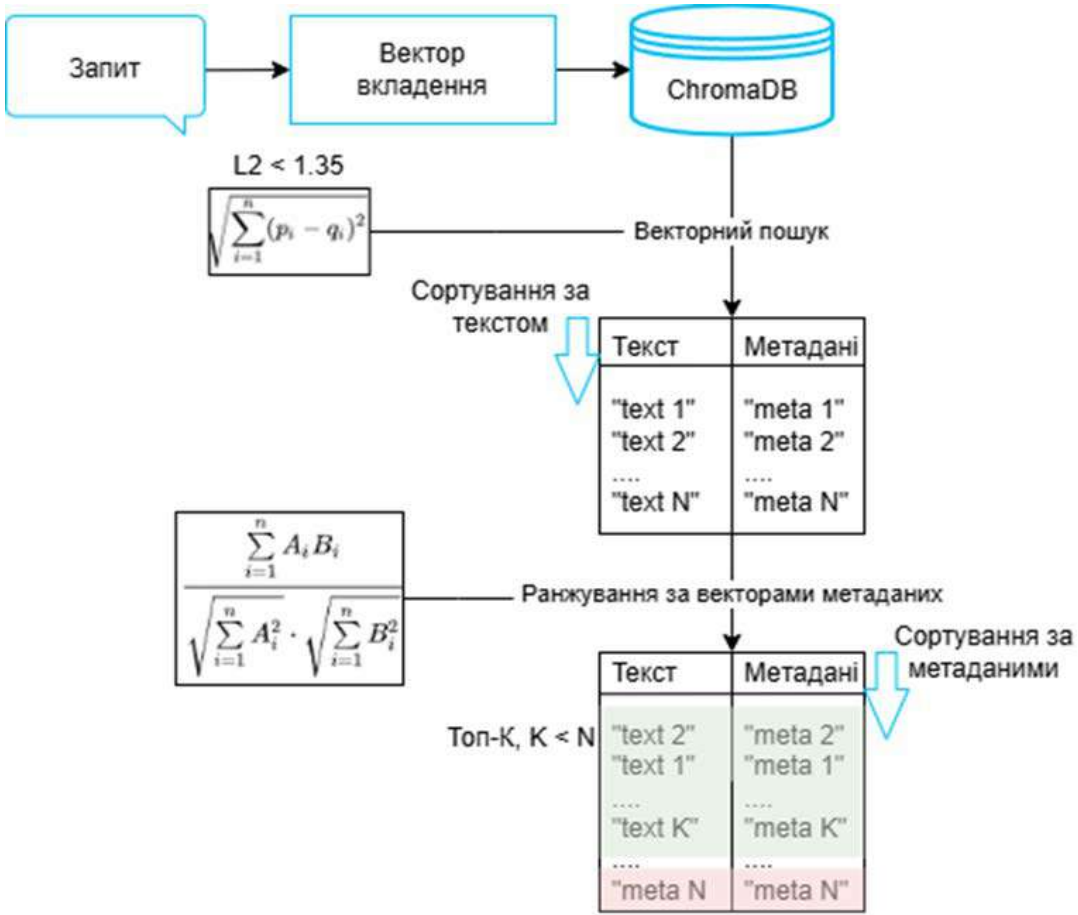


Врахована семантика сторінки

Під час обробки враховується семантика сторінки: нормалізується текст, видаляються повторювані теги, виділяються метадані, автоматично обираються ключові слова. Всього було збережено інформацію з більш ніж 300 сторінок.



Двокроковий алгоритм пошуку



Тестування пошуку

Використана метрика: Hits@K

$$\frac{|\{q \in Q : q < k\}|}{|Q|} \in [0, 1]$$

Моделі: text-embedding-3-small, text-embedding-3-large, bge-m3

Тестовий набір даних: 40 пар питань з різними категоріям та відповідний URL.
Приклад:
“Bachelor,Які загальні строки подачі документів?,https://vstup.ukma.edu.ua/bachelor/entry-steps”

Результат:
text-embedding-3-large – 0.6, text-embedding-3-small – 0.8 bge-m3 – 0.875

Результати

Hits@K (K=10/20, N=30)
Тестовий набір даних: розширено до 140 питань, використання семантично незначимих конструкцій: *“будь ласка”, “я шукаю інформацію про”* та інші.

text-embedding-3-small	K	metadata	plain
	10	0.7286	0.65
	20	0.8214	0.75
bge-m3	K	metadata	plain
	10	0.7714	0.8429
	20	0.8714	0.9357

Вплив ранжування варіюється в залежності від моделі.
Покращення результатів для моделі **text-embedding-3-small** свідчить про ефективність цього підходу в умовах, де є доцільним використання сторонньої моделей через API.

Генерація відповідей

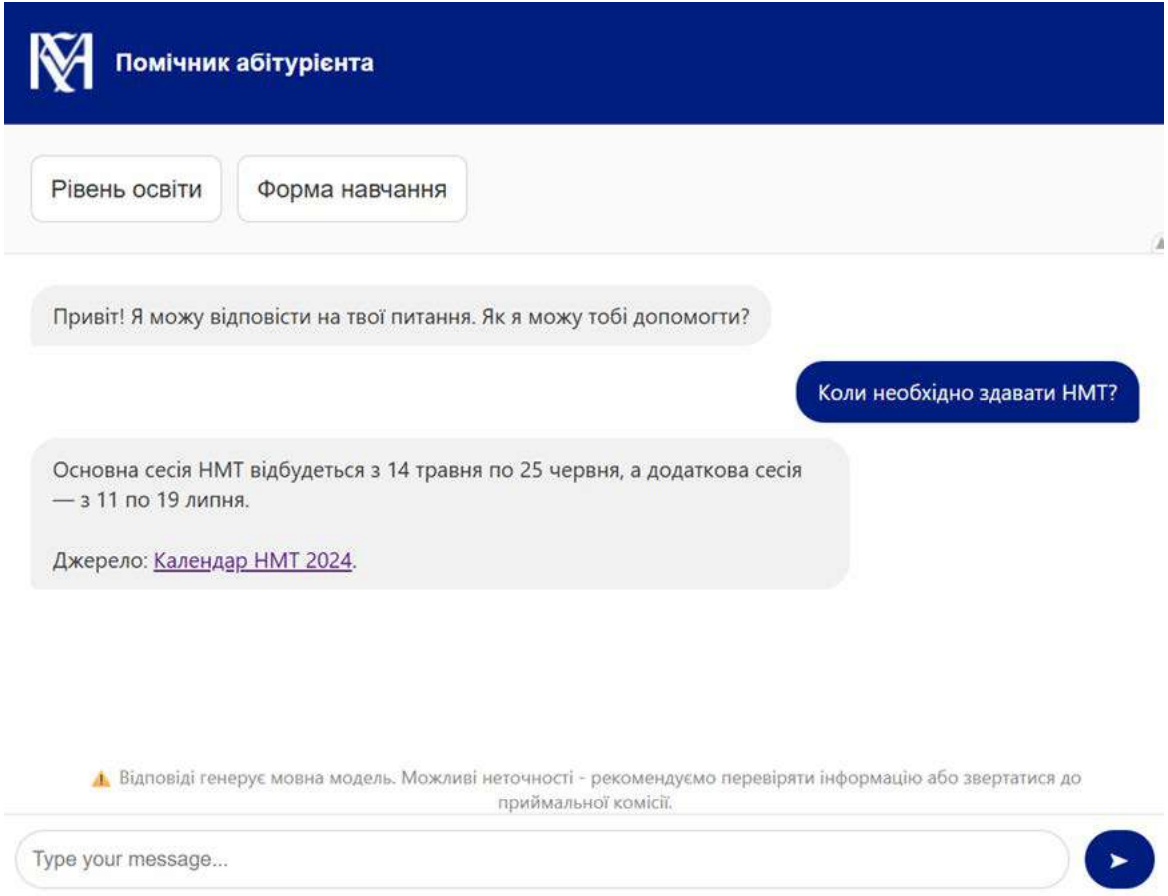
Для покращення якості відповідей застосована інженерія запитів - структура запиту містить інструкцію щодо ролі системи, мови відповіді, запит складається з питання користувача та впорядкованих джерел знань.

Ручне тестування якості відповідей підтвердило попередні результати - при використанні “bge-m3” частіше знаходила відповідь без додаткових уточнень.

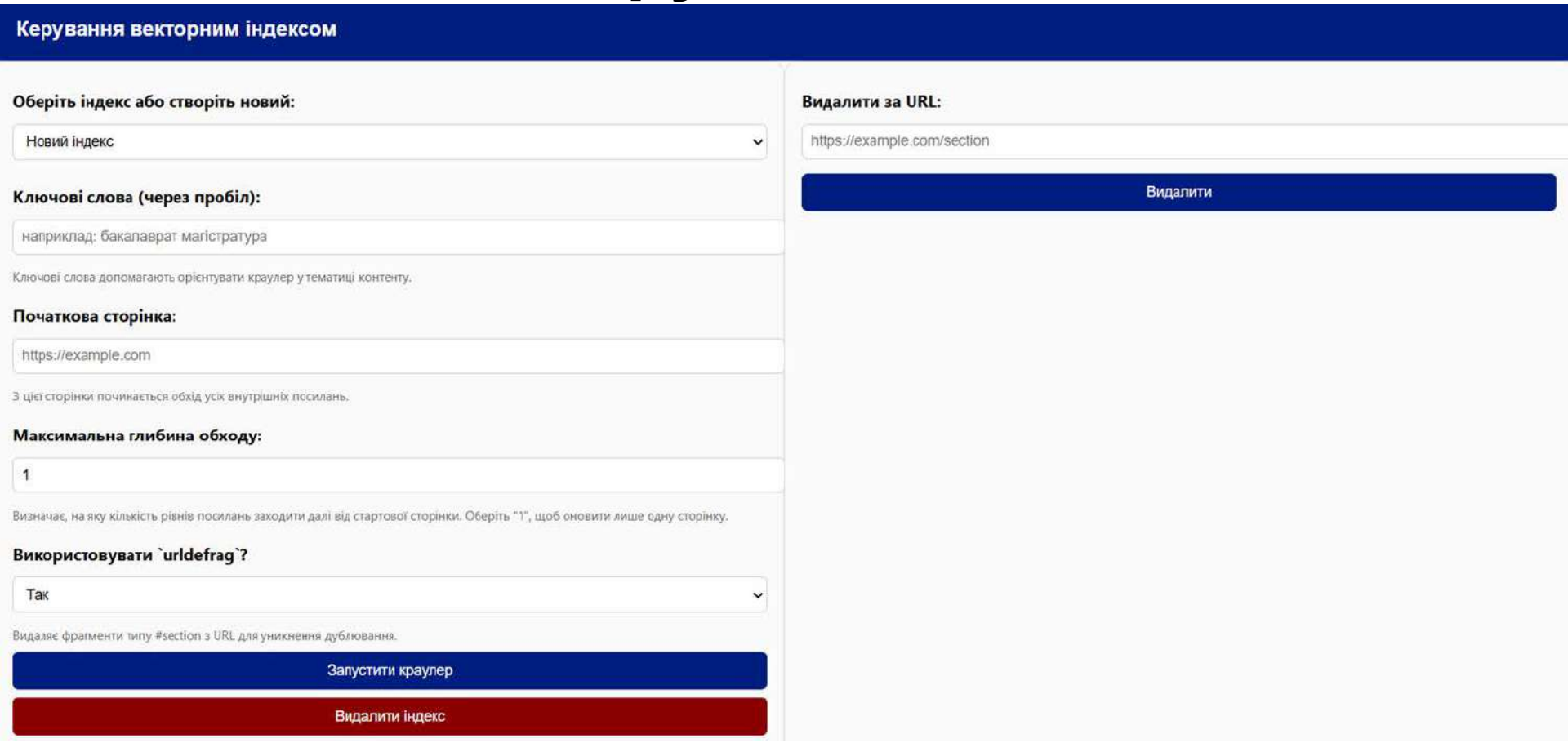
```
INSTRUCTION = "Ти - помічник абітурієнта або студента НаУКМА. " \
"Використовуючи надані фрагменти з офіційного сайту, надай об'ємну відповідь на запит користувача. " \
"Якщо інформація не міститься в наданих фрагментах, однак стосується вступної кампанії, надай її на основі власних знань." \
"З наданих фрагментів, обери найбільш релевантний та додай це посилання <url> в кінці відповіді в такому форматі: 'Джерело: <url>'." \
"Якщо відповідь не можна надати впевнено, вкажи, що інформація відсутня або потребує уточнення." \
"Якщо питання не стосується НаУКМА, вступної кампанії чи освітніх планів, скажи що надаєш відповіді тільки на ці питання." \
"Ввід буде містити інсрукцію щодо мови, якою має бути відповідь. " \
"Приклад цієї інструкції: 'Відповідь має бути написана такою мовою: uk.'" \
```

```python
def generate_response(prompt, response_id=None):
    response = client.responses.create(
        model="gpt-4o",
        instructions=INSTRUCTION,
        temperature=0.9,
        input=prompt,
    )
```

Користувацький інтерфейс



Панель керування індексами



Результати

Реалізований **прототип системи підтримки для абітурієнтів** на основі Retrieval-Augmented Generation: було пройдено етапи збору даних, пошуку релевантної інформації та генерації відповідей на її основі.

Запропоновано та реалізовано **двокроковий алгоритм пошуку** із повторним ранжуванням результатів за допомогою вкладень метаданих.

Було створено **систему побудови та адміністрування векторних індексів веб-сторінок**, яка дозволяє індексувати сайт. Розроблено веб-інтерфейс, що спрощує керування конфігурацією індексу.

Висновки

Застосування підходу **Retrieval-Augmented Generation** на основі існуючих веб-ресурсів є ефективним та гнучким методом для побудови систем підтримки користувачів.

Двокроковий алгоритм пошуку з ранжуванням за метаданими продемонстрував покращення якості пошуку для моделі «text-embedding-3-small».