

Table 5

**Ranking of the risks of using generative AI services based on a survey
of KNEU and PUT students in 2025–2026**

Rank		KHEY	Rank		PUT
2026	2025	Risk	2026	2025	Risk
1	1	Decline in critical thinking	1	1	Decline in critical thinking
2	2	Violation of academic integrity	2	2	Dependence on technology від технологій
3	4	Insufficient interpretation of context and misinterpretation of data	3	3	Limitation of creativity
4	3	Dependence on technology	4	4	Insufficient interpretation of context and misinterpretation of data
5	6	Privacy and security issues	5	6	Privacy and security issues
6	5	Limitation of creativity	6	5	Violation of academic integrity
7	7	Excessive automation and interference in the educational process	7	7	Excessive automation and interference in the educational process
8	8	Unequal access to AI technologies	8	8	Unequal access to AI technologies

Therefore, the results of the study indicate an increase in the proportion of students who consider their use of AI to be responsible. This is a positive signal; however, it also highlights the need to further develop clear rules, AI literacy, critical thinking, and a culture of academic integrity in the context of the active integration of AI into the educational process.

References

1. Skitsko V., Pawłyszyn I. Responsible use of Artificial intelligence in Ukrainian and Polish higher education institutions. Цифрова економіка: зб. матеріалів III Міжнародної науковопрактичної конференції, 5-6 червня 2025 р., м. Київ. К.: КНЕУ, 2025. С.881-885. DOI: <https://doi.org/10.33111/978-966-926-559-3>

Безкоровайний В.С.,
к.е.н., доцент,

Київський національний економічний університет імені Вадима Гетьмана

ЛОКАЛЬНА RAG-СИСТЕМА ІНТЕЛЕКТУАЛЬНОГО ПОШУКУ ДЛЯ УКРАЇНОМОВНИХ УНІВЕРСИТЕТСЬКИХ ТЕКСТОВИХ МАСИВІВ

В умовах цифрової трансформації вищої освіти офіційні web-портали університетів перетворилися на складні інформаційні екосистеми. Вони акумулюють величезні масиви різномірних даних: від умов вступу, навчальних планів і розкладів до внутрішніх нормативних актів, наукових вісників та розпоряджень адміністрації. Студенти, викладачі та абітурієнти щодня стикаються із проблемою інформаційного перевантаження. Пошук конкретної відповіді серед сотень сторінок, вкладених PDF-документів та застарілих лінків перетворюється на тривалий і виснажливий квест, що суттєво знижує загальну ефективність комунікації всередині закладу вищої освіти.

Традиційна пошукова система, інтегрована в університетський сайт (Київський національний економічний університет імені Вадима Гетьмана [КНЕУ], б. д.), працює за принципом пошуку за ключовими словами, що суттєво обмежує її функціональність. Такі системи не здатні розуміти контекст запиту користувача, синоніми чи розмовну мову, видаючи або занадто великий список нерелевантних посилань або взагалі порожній результат. Сучасний користувач бажає миттєво отримати точну та сформульовану природною мовою відповідь на своє конкретне запитання. Створення інтелектуального помічника, здатного вести осмислений діалог та допомогти користувачу з пошуком потрібної інформації — необхідний крок модернізації освітнього простору.

Популярні загальнодоступні великі мовні моделі (LLM), такі як ChatGPT, Claude або Gemini, демонструють вражаючі здібності у генерації текстів та веденні діалогу, проте вони не пристосовані для вирішення специфічних завдань університету «з коробки». Головне обмеження публічних LLM полягає в тому, що вони навчені на історичних даних із відкритих джерел і не мають доступу до актуальної внутрішньої інформації нашого університету. Окрім того, базові версії LLM не вміють працювати з динамічними змінами, наприклад, оперативно оновлюваними новинами чи свіжими розпорядженнями або наказами ректора, які з'явилися на сайті лише кілька годин тому.

Розв'язання проблеми оновлення інформації за допомогою стандартних запитів до публічних LLM призводить до «галюцинацій», коли модель впевнено генерує вигадані або застарілі факти. Для університету така дезінформація є неприпустимою, оскільки помилка у правилах прийому чи іншій інформації може зірвати навчальний процес або зашкодити репутації. Додатковим фактором може бути конфіденційність — передача внутрішніх нормативних чи фінансових документів університету на сервери сторонніх компаній порушує внутрішні правила безпеки.

Найбільш ефективним технологічним розв'язанням окреслених проблем є архітектура RAG (Retrieval-Augmented Generation — генерація, доповнена пошуком). Цей підхід розділяє «мислення» та «знання». Замість того, щоб намагатися завантажити всі знання університету в пам'ять самої нейромережі шляхом регулярного та довготривалого донавчання (fine-tuning), RAG використовує LLM як інтелектуальний процесор. Модель отримує точні зразки тексту, знайдені у локальній базі даних спеціальним пошуковим алгоритмом, і на їх основі формує правильну, контекстуальну відповідь користувачу.

Процес проектування, налаштування та розгортання повноцінної автономної RAG-системи на локальних серверних потужностях університету складається з п'яти послідовних етапів.

Першим етапом побудови системи є збір, очищення та підготовка текстових даних університету. Для тестування працездатності технології використано виключно інформацію зі сторінок web-сайту університету. Це зумовлено тим, що для використання текстових або табличних файлів (.docx, .xlsx, .pptx тощо) та сканованих наказів у форматі PDF, вміст необхідно «витягувати» за допомогою спеціалізованих парсерів та інструментів. Всього сайт університету налічує 40 288 сторінок з контентом, при цьому свіжа інформація, яку було створено або оновлено за останній рік — лише 1 626 сторінок (посилань на документи на цих сторінках 6 233).

Другий етап — розмітка отриманого тексту у форматі Markdown (спрощена мова розмітки, яка дозволяє формувати звичайний текст за допомогою таких символів, як #, *, _ тощо). Така розмітка найчастіше використовується для структурування, візуалізації та логічного виділення відповідей, щоб користувачеві було максимально зручно читати великі обсяги згенерованого LLM тексту. Більшість HTML-тегів мають відповідні аналоги у Markdown, тому таку розмітку сторінок web-сайту можна реалізувати без ускладнень. Що стосується текстових файлів — можливе отримання форматування тексту (напівжирний, курсив), а більш складні структури (заголовки, таблиці, формули тощо) розбирати набагато важче.

Оскільки документи можуть бути занадто великими для сприйняття мовною моделлю, на третьому етапі, їх потрібно розбити на логічні фрагменти — так звані «чанки» (chunks)

розміром до 1024 токенів із перекриттям. Це дозволяє зберегти контекст речень і підготувати інформацію для наступного математичного аналізу. У розробленій системі, враховуючи, що більшість тексту написана кирилицею і на кожен символ використовується 2 байти, розмір чанку встановлено у 250 слів із перекриттям у 150 слів.

Четвертий етап передбачає перетворення текстових фрагментів у векторну форму за допомогою моделі ембедингів (використано BAAI/bge-m3, оптимізовану для української мови). Цей процес трансформує кожен фрагмент тексту у вектор — послідовність чисел, що відображає семантичний зміст інформації. Отримані вектори завантажуються у спеціалізовану векторну базу даних Qdrant. Це дає змогу порівнювати запити користувачів із документами не за збігом літер, а за схожістю їхнього внутрішнього змісту.

У підсумку 1 626 сторінок з сайту університету збережено у колекцію з 4 596 об'єктів (векторів з метаданими). Колекція містить вектори розмірністю (кількість вимірів) 1 024, а схожість між ними визначається за косинусною відстанню (Cosine). Розподілення векторів зі збереженим текстом, представлено у двовимірному просторі за допомогою алгоритму t-SNE на рис. 1. Цей алгоритм машинного навчання стискає багатовимірні вектори з бази даних у двовимірний простір для графічного відображення. Така візуалізація допомагає оцінити розподіл інформації в базі даних, помітити приховані закономірності, виявити аномалії, а також побачити, як саме дані групуються за семантичним змістом.

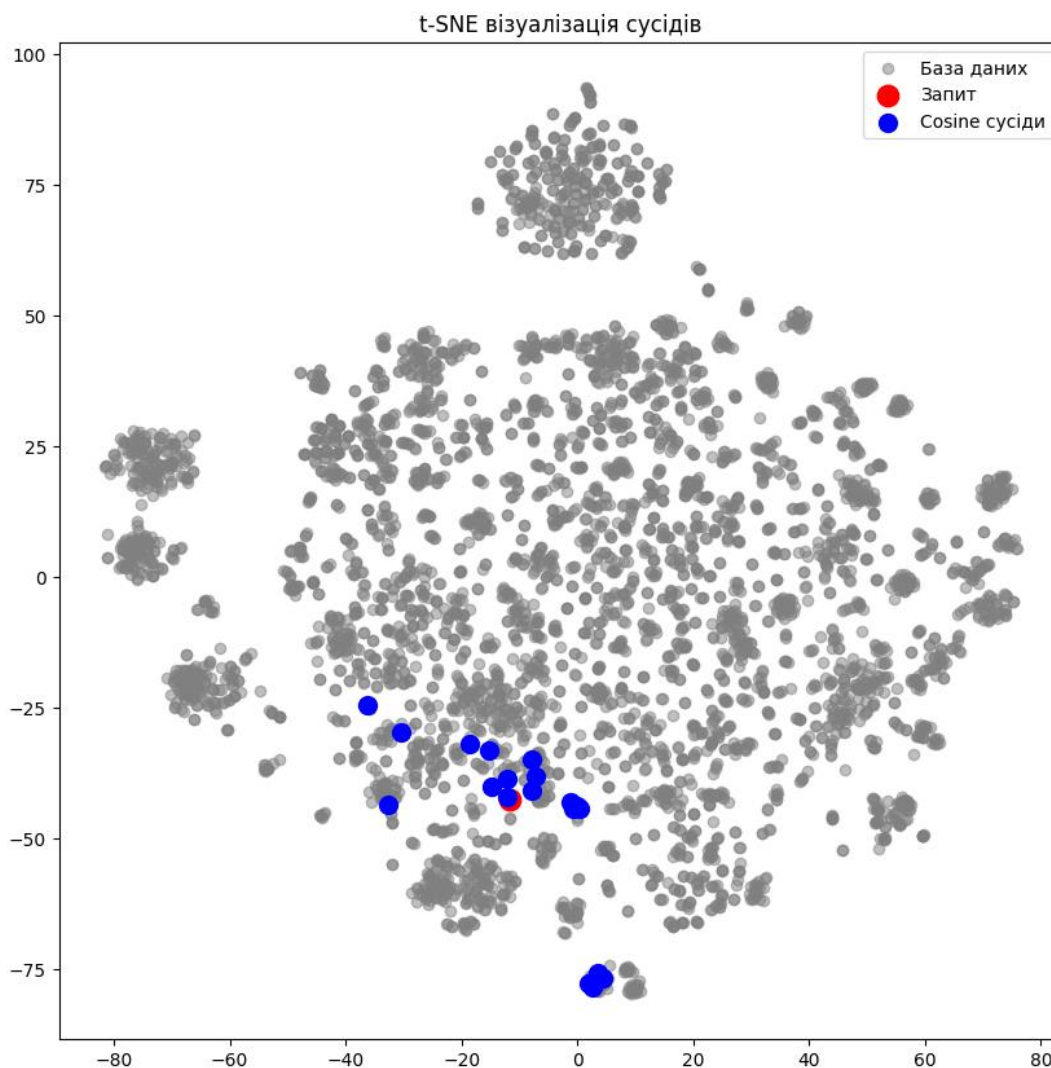


Рисунок 1. t-SNE візуалізація сусідів векторної бази даних

Джерело: побудовано автором на основі (КНЕУ, б. д.).

На рис. 1 також наведено приклад як векторна база даних шукатиме відповідь на запит користувача: «Хочу вступити до університету, розкажи про магістерські програми, які я можу обрати». Сам запит позначено червоною точкою, а 20 найближчих за змістом текстових блоків — синім кольором. Як можна бачити, не всі блоки отримані з бази даних розташовані поблизу запиту, вони можуть бути з інших кластерів, або як у нашому випадку, з різних розділів сайту. Так відповідь на цей запит може знаходитись у розділі Приймальної комісії, кафедр, факультетів, Навчально-наукового інституту бізнес-освіти тощо.

Останній, п'ятий етап — поєднання запиту користувача з контекстом, який отримано з векторної бази даних, і передача до LLM. Для тестування розробленої системи розгорнуто web-сервер у середовищі Python на одному з серверів університету, інтерфейс системи представлено на рис. 2. Локальною LLM обрано багатомовну модель Mistral-7B-Instruct-v0.2, яка вміщується в оперативну пам'ять (займає 16 Гб) та може працювати на CPU (AMD EPYC 7302). Результат запиту містить Markdown елементи, відповідає поданому контексту та наявній інформації на сайті університету.

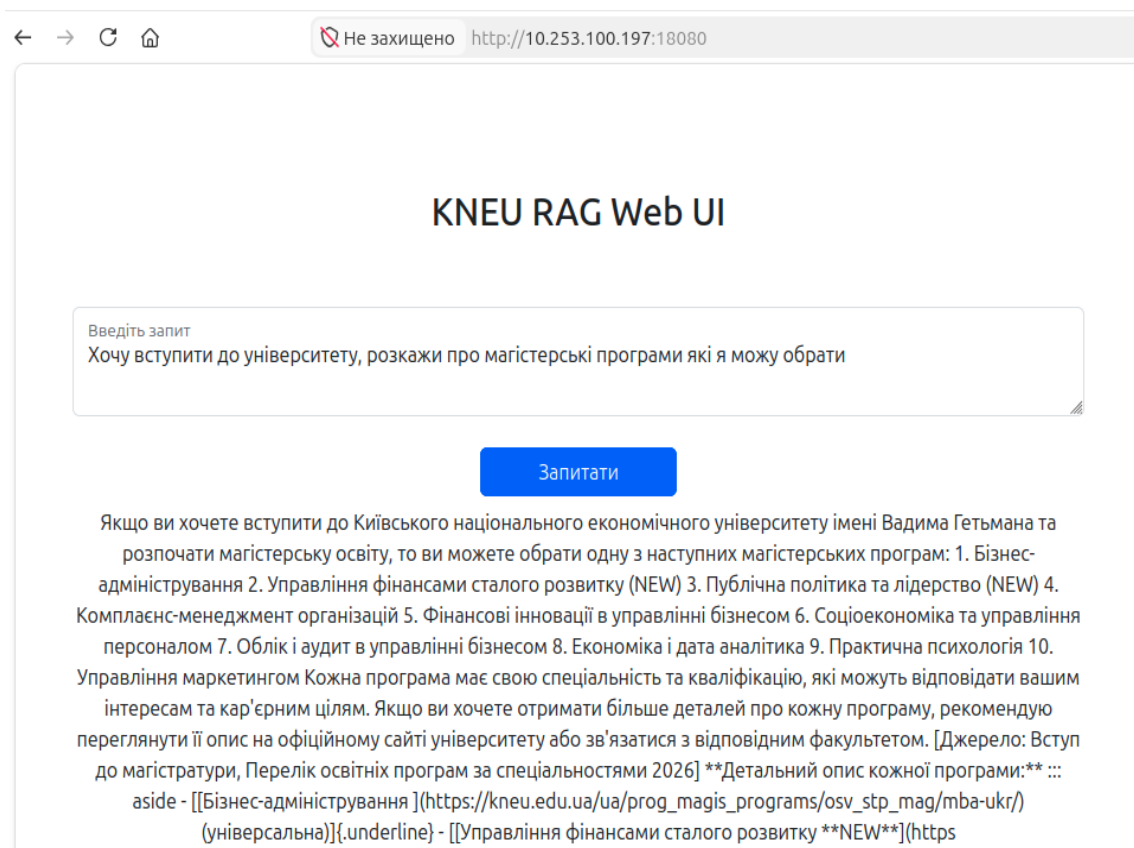


Рисунок 2. Інтерфейс RAG-системи з прикладом виконання користувацького запиту

Джерело: створено автором.

Створення суверенного чат-бота дозволить повністю ліквідувати ризик витоку інформації, забезпечить стовідсоткову точність відповідей без галюцинацій та автоматизує обробку типових запитів користувачів. Локальна інфраструктура гарантує незалежність університету від зовнішніх платних API, лімітів іноземних провайдерів та забезпечить повний контроль над цифровими знаннями.

Список використаних джерел

1. Київський національний економічний університет імені Вадима Гетьмана. (б. д.). Головна сторінка. <https://kneu.edu.ua/>