МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук та технологій
Кафедра «Комп'ютерні системи та мережі»

ЗАХАРЧЕНКО КОСТЯНТИН РОМАНОВИЧ
Група КНТ-514м

КОНСУЛЬТАТИВНО-ДОВІДКОВА СИСТЕМА ПІДТРИМКИ СТУДЕНТІВ НА БАЗІ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

АВТОРЕФЕРАТ

дипломної роботи на здобуття освітньо-кваліфікаційного рівня
«магістр» 123 Комп'ютерна інженерія
освітньої програми «Комп'ютерні системи та мережі»

2025 р.

Дипломна робота є рукопис.

Робота виконана в національному університеті «Запорізька політехніка», на кафедрі комп’ютерних систем та мереж

Керівник	кандидат технічних наук, доцент **Тягунова Марія Юріївна,** Національний університет «Запорізька політехніка», декан фак. КНТ, доцент кафедри комп'ютерних систем та мереж
Офіційний рецензент:	**ЩЕРБАКОВ Володимир Іванович,** директор НВФ "Бізнес Комп'ютер Сервіс"

Захист відбудеться "22" січня 2026 р.

Секретар екзаменаційної комісії, доцент кафедри комп’ютерних систем та мереж Т.В. Голуб

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. Актуальність обраної теми зумовлена зростанням обсягів інформації в освітньому середовищі та потребою у швидкому, зручному й достовірному доступі студентів до довідкових матеріалів у зрозумілому форматі.

У межах роботи об’єктом дослідження визначено консультативно-довідкову систему підтримки студентів на базі великих мовних моделей із використанням RAG, а метою – підвищення зручності доступу до довідкової інформації університету шляхом створення прототипу чат-асистента, що формує відповіді на основі релевантних фрагментів бази знань і керованих правил генерації.

Аналіз предметної області показує, що традиційні рішення (сценарні чат-боти та FAQ-платформи), хоча й орієнтовані на оперативні відповіді, мають суттєві обмеження через нестабільність і залежність від вручну сформованої бази відповідей.

Водночас застосування LLM є перспективним завдяки роботі з природною мовою, проте встановлено, що використання лише генеративної моделі без доступу до актуальної доменної інформації є недостатнім для довідкових задач; натомість інтеграція семантичного пошуку з LLM підвищує точність і надійність відповідей та забезпечує доступ до поточних даних.

Для довідкових сервісів критичним є запобігання домислюванню фактів, тому в роботі акцентовано правило «джерела істини», за яким відповідь має формуватися виключно на основі retrieved context, а за відсутності релевантних фрагментів має повертатися стандартизоване повідомлення про нестачу даних.

Запропоноване рішення орієнтоване на масштабованість і супровід та може адаптуватися до змін інформаційного наповнення без перенавчання мовної моделі, що є важливим для динамічного університетського середовища.

Практична значущість теми підсилюється тим, що система може зменшувати навантаження на адміністративний і викладацький персонал, забезпечувати цілодобовий доступ до довідкової інформації та створювати умови для підвищення якості інформаційної підтримки студентів, із потенціалом подальшого

розширення та інтеграції з внутрішніми інформаційними системами.

Мета і завдання дослідження. Мета дипломної роботи полягає у підвищенні зручності доступу користувачів до довідкової інформації університету шляхом розроблення прототипу чат-асистента, який генерує відповіді на основі релевантних фрагментів бази знань і керованих правил генерації.

Для досягнення поставленої мети необхідно виконати такі основні завдання:

1. Провести аналіз предметної області та аналогів системи, а також визначити вихідні дані й підходи, зокрема використання RAG для пошуку релевантних фрагментів у базі знань.
2. Спроєктувати архітектуру консультативно-довідкової системи та реалізувати її програмні компоненти, включно з веб-інтерфейсом, серверним модулем та інтеграцією з LLM через OpenAI API, а також формуванням запиту.
3. Провести тестування системи, виконати аналіз результатів і сформувати рекомендації щодо подальшого масштабування та супроводу.

Об’єктом дослідження є консультативно-довідкова система підтримки студентів закладу вищої освіти на базі великих мовних моделей.

Предмет дослідження є архітектурні та алгоритмічні рішення побудови прототипу чат-асистента для довідкових сценаріїв університету, зокрема формування відповіді на основі релевантних фрагментів бази знань і керованих правил генерації.

Методи дослідження базуються на аналізі і порівнянні підходів до побудови освітніх асистентів, проєктуванні архітектури системи з розмежуванням рівнів, експериментальному тестуванню працездатності й продуктивності і аналізі економічної доцільності експлуатації.

Наукова новизна отриманих результатів:

Обґрунтовано доцільність поєднання семантичного пошуку з генеративною моделлю в межах RAG для підвищення точності й надійності відповідей у довідкових задачах, а також запропоновано архітектурний підхід, що дає змогу масштабувати та актуалізувати

базу знань без перенавчання мовної моделі, використовуючи формалізовану схему побудови запиту.

Практичне значення отриманих результатів:

Розроблену систему можна використовувати як універсального цифрового асистента для студентів: вона забезпечує цілодобовий доступ до довідкової інформації та зменшує навантаження на адміністративний персонал; результати тестування підтверджують стабільну роботу при декількох одночасних діалогах.

Апробація результатів дипломної роботи. Основні ідеї застосування підходу RAG та порівняння векторних баз даних для цього наведено у тезах:

Тягунова М.Ю., Захарченко К.Р. Векторні бази даних для освітніх RAG-систем. Збірник тез IX Всеукраїнської науково-практичної конференції "Нові інформаційні технології управління бізнесом". Київ: Спілка автоматизаторів бізнесу, 2025. - у друці.

Структура та обсяг роботи. Дипломна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел, двох додатків. Основна частина містить 75 сторінок, 28 рисунків і 10 лістингів, список використаних джерел зі 15 найменувань.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У першому розділі проведено аналіз предметної області консультативно-довідкових систем для студентів та здійснено огляд наявних цифрових освітніх сервісів з метою обґрунтування технологічного підходу до створення чат-асистента. Показано еволюцію рішень від простих сценарних чатботів і FAQ-платформ, що надають наперед підготовлені відповіді, до систем, у яких ядром виступають великі мовні моделі. Наголошено, що шаблонні підходи забезпечують передбачуваність і контроль якості, проте є слабко масштабованими та залежать від ручного поповнення бази знань, через що погано працюють за широкої тематики запитів і не здатні адаптуватися до нестандартних формулювань.

Окрему увагу приділено можливостям LLM у контексті природномовної взаємодії, зокрема здатності формувати відповіді в реальному часі, пояснювати матеріал і надавати контекстуалізовані

рекомендації. Водночас зафіксовано ключові ризики прямого використання LLM у довідкових задачах закладу вищої освіти, насамперед проблему актуальності інформації та ймовірність генерації фактично некоректних тверджень у разі нестачі достовірних даних. Унаслідок цього обґрунтовано доцільність застосування Retrieval-Augmented Generation як механізму, що поєднує генеративні можливості LLM із доступом до актуальної бази знань. Узагальнений сценарій роботи такої системи та послідовність взаємодії між пошуком і генерацією наведено на рисунку 1.

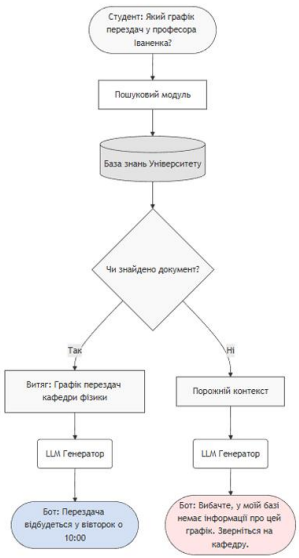

Рисунок 1 – Сценарій роботи чатбота на LLM з RAG

На завершення сформульовано практичні висновки щодо впровадження: необхідність оновлюваного корпусу даних, інструментів швидкого пошуку, контрольованих налаштувань генерації, дотримання вимог інформаційної безпеки та організації зворотного зв’язку для підтримання якості відповідей у реальних умовах експлуатації.

У другому розділі оформлено вимоги до консультативно-довідкової системи та виконано її проєктування як вебзастосунку з серверним API-шлюзом і інтеграцією LLM із застосуванням RAG. Визначено призначення системи як сервісу діалогового доступу до відкритих інформаційних матеріалів університету, де відповіді формуються на основі релевантних фрагментів внутрішньої бази знань. Сформульовано функціональні вимоги з розподілом ролей на користувача та адміністратора, при цьому підкреслено відсутність механізмів автентифікації й персоналізації через роботу виключно з публічними даними, а також зафіксовано, що історія діалогу не зберігається на сервері й кожен запит обробляється незалежно. Окремо визначено вимоги до клієнтського інтерфейсу як до засобу введення запиту природною мовою та відображення відповіді у структурованому вигляді з коректною обробкою типових помилок введення і мережевих збоїв, а серверну частину описано як компонент, що уніфікує взаємодію між фронтендом і зовнішнім LLM-провайдером та забезпечує стандартизований контракт обміну даними.

У межах нефункціональних вимог окреслено очікування щодо стилю й мови відповідей, а також визначено цільові показники продуктивності та надійності для інтерактивного режиму використання, включно з вимогою коректної роботи за паралельних звернень і повернення стандартизованих повідомлень у разі збоїв провайдера. Для безпеки зафіксовано винесення секретів у змінні середовища, застосування CORS для контролю джерел запитів і базового обмеження частоти звернень. Паралельно обґрунтовано необхідність RAG як обов’язкової властивості системи, що забезпечує розширення запиту релевантним контекстом з внутрішніх джерел, зменшуючи ризик некоректних тверджень і підвищуючи відтворюваність відповідей за рахунок опори на актуальні дані.

Далі в розділі описано вибір технологій реалізації прототипу та їх відповідність поставленим вимогам. Клієнтську частину визначено як React-застосунок на TypeScript, що забезпечує компонентний підхід і керування станом діалогу, а серверний рівень реалізовано на Express із TypeScript як компактний API-шлюз для централізованого керування параметрами генерації, форматом відповіді та правилами поведінки асистента. Інтеграцію

з LLM організовано через OpenAI API, при цьому архітектурно передбачено уніфікований інтерфейс провайдера та конфігураційне перемикання між реалізованими інтеграціями без зміни клієнтської логіки. RAG реалізовано серверним модулем, який перед викликом моделі відбирає релевантні фрагменти з бази знань і додає їх до промпту, що дозволяє актуалізувати інформаційне наповнення шляхом редагування даних без модифікації коду.

Завершальна частина розділу деталізує динаміку роботи системи через моделювання повного циклу обробки запиту у вигляді діаграми послідовності. Підкреслено, що такий вид діаграм фіксує часовий порядок обміну повідомленнями між учасниками, активацію компонентів і повернення результатів, завдяки чому стає можливим виявлення критичних точок інтеграції та потенційних джерел затримок ще на етапі проєктування. Розроблену діаграму послідовності обробки користувацького запиту наведено на рисунку 2.

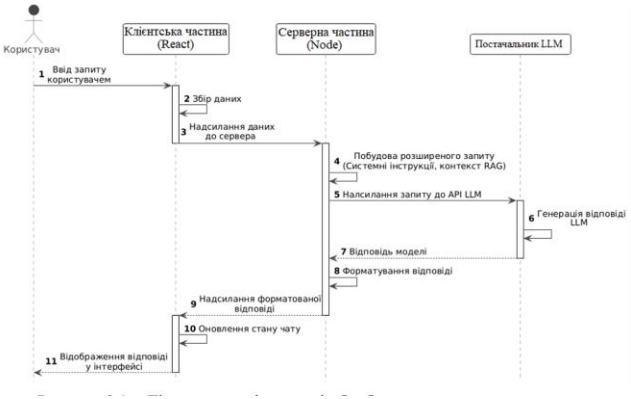

Рисунок 2 – Діаграма послідовності обробки користувацького запиту

У третьому розділі описано створення системи як сукупності клієнтського вебінтерфейсу та серверного компонента, що забезпечує інтеграцію з великою мовною моделлю та реалізацію RAG для підвищення точності відповідей. Розкрито логіку взаємодії складових, де клієнтська частина відповідає за діалогову взаємодію з користувачем, а серверна – за приймання запитів,

формування контексту з бази знань, звернення до LLM і повернення уніфікованої відповіді.

Обґрунтовано вибір OpenAI як провайдера LLM з огляду на практичну придатність API та керованість параметрів генерації. Взаємодію з моделлю інкапсульовано в окремому провайдерному модулі з уніфікованим інтерфейсом, що зменшує залежність від конкретного постачальника й дозволяє змінювати реалізації конфігураційно. Ключ доступу та налаштування зосереджено на серверному рівні, що підвищує безпеку та забезпечує відтворюваність поведінки асистента.

Описано принцип побудови запиту до моделі в межах RAG через розділення інструкцій, контексту та користувацького запиту. Наголошено, що модель повинна формувати відповідь на основі відібраних фрагментів бази знань, а в разі їх недостатності застосовується керована відмова, що знижує ризик генерації непідтверджених тверджень. Для підвищення якості рекомендується обмежувати обсяг контексту та мінімізувати шум і дублікати.

Клієнтську частину реалізовано на React і TypeScript як діалоговий інтерфейс із керуванням станами під час асинхронного обміну даними. Передбачено збереження історії повідомлень у межах клієнтського стану, індикацію обробки та уніфіковане відображення помилок. Відповіді відображаються з підтримкою форматування, що підвищує читабельність і практичну придатність довідкових повідомлень.

Серверна частина реалізує API-шлюз, виконує базову валідацію запитів, формує контекст із локальної бази знань та оркеструє звернення до LLM. RAG реалізовано як відбір релевантних фрагментів, їх структурування та включення до запиту моделі. Для гнучкості передбачено абстракцію провайдерів і фабричний підхід, а також механізми контролю доступу та уніфікованої обробки збоїв, що забезпечує передбачувану роботу системи в реальних умовах експлуатації.

У четвертому розділі було проведено тестування та аналіз роботи системи з акцентом на двох взаємодоповнювальних аспектах: швидкодії серверного API та якості сформованих відповідей у наближених до реальної експлуатації сценаріях. Для оцінювання продуктивності застосовано навантажувальне

тестування маршруту POST /api/chat із фіксованою тривалістю прогонів і контрольованими режимами одночасних підключень, що дало змогу порівняти поведінку системи за послідовної взаємодії одного користувача та за помірного паралельного навантаження. Умови експерименту були відтворюваними, а результати зберігалися у форматі, придатному для подальшої обробки та візуалізації.

Окремо зафіксовано апаратне середовище тестування як локальну платформу споживчого класу, що відповідає типовому розгортанню прототипу поза спеціалізованою серверною інфраструктурою. Зазначено, що розмір бази знань у прототипі був невеликим, тому внесок модуля добору контексту у загальну затримку був мінімальним, а основна частка часу відповіді формувалася на етапі виклику зовнішнього сервісу LLM і генерації тексту.

За результатами вимірювання часу відповіді встановлено, що система демонструє прийнятну інтерактивність для діалогового довідкового сервісу, а ключові перцентилі не погіршуються при переході до паралельного режиму, що свідчить про відсутність тенденції до накопичення черги запитів за невеликого зростання одночасних звернень. Аналіз пропускної здатності показав очікуване масштабування інтенсивності обробки при паралельних підключеннях і підтвердив здатність підтримувати кілька одночасних діалогів без деградації коректності роботи, з урахуванням того, що інтегральні показники залежать не лише від локального сервера, а й від часових характеристик провайдера LLM.

Якість відповідей оцінювалася на підготовленому наборі запитів, який охоплював коректні звернення, неоднозначні формулювання, некоректні або нереалістичні припущення та питання поза предметною областю. Методологія передбачала перевірку того, чи спирається система на контекст RAG, чи зберігає керовану поведінку за нестачі даних, а також чи формує практично придатні відповіді у довідковому форматі. Експертне оцінювання за трирівневою шкалою показало переважно узгоджену поведінку, однак виявило окремі випадки, де відповідь потребує посилення правил, що обмежують необґрунтовану конкретизацію або помилкові припущення.

На основі отриманих результатів сформульовано рекомендації щодо подальшого використання та вдосконалення системи. Для користувачів підкреслено доцільність формулювати запити з мінімально необхідними уточненнями, що зменшує частку неоднозначних сценаріїв. Для адміністратора визначено пріоритетність підтримання актуальності та структурованості бази знань, бажаність регулярного оновлення, версіонування і розширення покриття типових тем. Рекомендовано уніфікувати стратегії поведінки у двох проблемних класах ситуацій: неоднозначності (з уточненням конкретних параметрів) та відсутності даних або виходу за межі предметної області (з керованою відмовою без здогадок). Також окреслено напрями технічних покращень, зокрема оптимізацію добору контексту, контроль його обсягу та можливе ускладнення механізму пошуку для підвищення стійкості до перефразувань у разі масштабування бази знань.

Додатково наведено економічну оцінку експлуатації: під час тестування було виконано майже 1000 звернень до провайдера LLM на невеликих і середніх за складністю запитаннях, а сукупна вартість склала менше 1 долара США. Для вимірювань використовувалась модель gpt-4.1-mini, і отримані значення свідчать про низьку змінну собівартість обробки типового навантаження та можливість масштабування кількості користувачів, а також про потенційну можливість переходу на потужнішу, але дорожчу модель без зміни загальної архітектури.

ВИСНОВКИ

У дипломній роботі розв’язано задачу розроблення консультативно-довідкової системи підтримки студентів на базі великих мовних моделей із використанням підходу RAG.

Отримано такі теоретичні та практичні результати:

Проведено аналіз сучасних цифрових рішень у сфері освітніх асистентів та виконано порівняльний аналіз підходів до побудови консультативних систем. Підтверджено, що для довідкових задач використання лише генеративної моделі без доступу до актуальної доменної інформації є недостатнім, тоді як поєднання

семантичного пошуку з LLM підвищує точність і надійність відповідей.

Розроблено архітектуру системи з розмежуванням клієнтського, серверного, інтелектуального рівнів та рівня даних, що забезпечує масштабованість, спрощує супровід та дозволяє адаптувати наповнення без перенавчання мовної моделі.

Реалізовано прототип вебсистеми, у межах якого сформовано клієнтський інтерфейс взаємодії з користувачем і серверну логіку обробки запитів з уніфікованим контрактом взаємодії.

Реалізовано механізм керованої генерації відповіді на основі RAG: відповідь формується лише з урахуванням релевантних фрагментів бази знань, а структура промпта підтримує розподіл інструкцій і контексту.

Також передбачено можливість перемикання LLM-провайдера шляхом зміни конфігурації без модифікації коду, з єдиним форматом API та уніфікованою обробкою помилок.

Проведено експериментальну перевірку працездатності та продуктивності системи. За результатами навантажувального тестування середня затримка становила 0.977 с, а 99% відповідей – до 1.386 с; у режимі з п’ятьма паралельними підключеннями середня затримка – 0.872 с, а 99% відповідей – до 1.375 с.

Пропускна здатність у режимі одиночного підключення становила в середньому 1 запит/с, а для п’яти паралельних підключень – 5.57 запитів/с у середньому.

Оцінка якості відповідей підтвердила здатність системи формувати змістовні та релевантні відповіді на типові студентські запити за умови достатньої визначеності запиту та наявності релевантного контексту; у випадках нечіткості доцільним є уточнення формулювання.

Практична цінність полягає у можливості використання рішення як допоміжного каналу довідкової підтримки студентів у режимі 24/7 з перспективою подальшого масштабування та адаптації (зокрема під інші навчальні заклади або інтеграцію з внутрішніми інформаційними системами).

Таким чином, усі поставлені задачі виконано, а мети дипломної роботи досягнуто.