

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут прикладного системного аналізу

Кафедра штучного інтелекту

До захисту допущено:

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«__» _____ 20__ р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Системи і методи штучного
інтелекту»**

спеціальності 122 «Комп'ютерні науки»

**на тему: «Трансформація тексту з урахуванням контексту з
використанням чат-боту ChatGPT»**

Виконав (-ла):

студент (-ка) IV курсу, групи КІ-03

Осовська Анна Едуардівна _____

Керівник:

доцент, кандидат технічних наук,

Тимошенко Юрій Олександрович _____

Консультант з економічного розділу:

доцент, доктор філософії з економіки,

доцент кафедри економічної кібернетики ФММ,

Мажара Гліб Анатолійович _____

Консультант з нормоконтролю:

фахівець першої категорії кафедри ШІ,

Кравець Павло Володимирович _____

Рецензент:

декан ФПМ, професор, доктор технічних наук,

професор кафедри програмного забезпечення комп'ютерних систем ФПМ,

Дичка Іван Андрійович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент (-ка) _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«31» січня 2024 р.

ЗАВДАННЯ

на дипломну роботу студенту

Осовській Анні Едуардівні

1. Тема роботи «Трансформація тексту з урахуванням контексту з використанням чат-боту ChatGPT», керівник роботи Тимошенко Юрій Олександрович, доцент, кандидат технічних наук, затверджені наказом по університету від «31» травня 2024 р. № 2240-С
2. Термін подання студентом роботи «10» червня 2024 року.
3. Вихідні дані до роботи: чат-бот ChatGPT, велика мовна модель OpenAI, вихідні тексти (без графічного матеріалу), сценарії трансформації тексту.
4. Зміст роботи: Вибір технології ChatGPT для трансформації тексту, Концепція функціональності, Детальний опис функціональності, Робота функціональності, Функціонально-вартісний аналіз функціоналу трансформації тексту з урахуванням контексту з використанням чат-боту ChatGPT.
5. Перелік ілюстративного матеріалу: електронна презентація Microsoft PowerPoint

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Мажара Гліб Анатолійович, доцент, доктор філософії з економіки, доцент кафедри економічної кібернетики ФММ		

7. Дата видачі завдання «05» лютого 2024 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз можливих напрямків роботи	15.04.2024	Виконано
2	Вивчення теми	21.04.2024	Виконано
3	Підбір літератури з теми	21.04.2024	Виконано
4	Розділ 1. Визначення та актуальність трансформації тексту. Вибір чат-боту ChatGPT як технології для трансформації.	28.04.2024	Виконано
5	Розділ 2. Концепція функціональності	12.05.2024	Виконано
6	Розділ 3. Детальний опис функціональності	26.05.2024	Виконано
7	Розділ 4. Результати запитів	02.05.2024	Виконано
8	Розділ 5. Функціонально-вартісний аналіз функціоналу трансформації тексту з урахуванням контексту з використанням чат-боту ChatGPT	10.06.2024	Виконано

Студент

А. Е. ОСОВСЬКА

Керівник

Ю. О. ТИМОШЕНКО

РЕФЕРАТ

Дипломна робота: 109 с., 22 рис., 16 табл., 19 посилань, 1 додаток.

Ключові слова: Текст, трансформація, контекст, інтеграція, чат-бот, Chat-GPT, універсальність.

Об'єкт дослідження – використання чат-боту ChatGPT для трансформації тексту.

Предмет дослідження – трансформація тексту з урахуванням контексту.

Мета роботи – розробка концепції комплексного та універсального функціоналу для трансформації тексту з урахуванням контексту в рамках веб-додатків, інтегрованого з чат-ботом ChatGPT від OpenAI.

Проведено комплексне дослідження визначення, прикладів та мети трансформації тексту, пояснено актуальність трансформації тексту в сучасному світі. Проведено ретельний аналіз інструменту чат-боту ChatGPT як механізму реалізації трансформації тексту з урахуванням контексту, обґрунтовано вибір чат-боту ChatGPT. Розроблено концепцію універсальної функціональності для перетворення тексту за допомогою чат-бота ChatGPT у веб-додатках, яка відповідатиме поточним і майбутнім потребам перетворення тексту, таким чином гарантуючи, що функціонал можна буде надійно використовувати протягом тривалого часу у багатьох випадках використання. Проведено аналіз та формулювання високорівневих функціональних та нефункціональних вимог для реалізації функціональності трансформації тексту, ретельне дослідження аналогічних функціональностей на ринку та найкращих практик створення запитів в ChatGPT для трансформації тексту. Уточнено та задокументовано вимоги до функціональності, створено прототипи користувацького інтерфейсу та UML діаграми роботи функціональності. Складено можливі запити в ChatGPT для трансформації тексту на основі проведених досліджень. Продемонстровано самі трансформації тексту, які показують тексти "до" і "після" трансформацій. Обґрунтовано успішність реалізованої концепції.

ABSTRACT

Master's thesis: 109 p., 22 figures, 16 tables, 19 references, 1 appendix.

Keywords: Text, transformation, context, integration, chatbot, Chat-GPT, generic.

The object of the study is using the ChatGPT chatbot for text transformation.

The subject of research is context-aware text transformation.

The goal of the work is to develop the concept of a comprehensive and universal functionality for transforming text within web applications, integrated with the ChatGPT chatbot from OpenAI.

A comprehensive study of the definition, examples and purpose of text transformation was conducted, the relevance of text transformation in the modern world was explained. A thorough analysis of the ChatGPT chatbot tool as a mechanism for implementing text transformation taking into account the context was carried out, the choice of the ChatGPT chatbot was justified. A universal functionality for text conversion using the ChatGPT chatbot in web applications has been conceptualized to meet current and future text conversion needs, thus ensuring that the functionality can be used reliably for a long time in many use cases. The analysis and formulation of high-level functional and non-functional requirements for the implementation of the text transformation functionality, a thorough study of similar functionalities on the market and the best practices for creating requests in ChatGPT for text transformation were carried out. Functionality requirements were clarified and documented, user interface prototypes and UML functional diagrams were created. Created possible requests in ChatGPT for text transformation based on conducted research. The text transformations themselves are demonstrated, showing the texts "before" and "after" the transformations. The success of the implemented concept is substantiated.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 ВИБІР ТЕХНОЛОГІЇ CHATGPT ДЛЯ ТРАНСФОРМАЦІЇ ТЕКСТУ	11
1.1 Визначення та приклади трансформації тексту	11
1.1.1 Визначення трансформації тексту	11
1.1.2 Приклади трансформації тексту	11
1.1.3 Математична основа для трансформації тексту	14
1.2 Цілі трансформації тексту	15
1.3 Цінність та актуальність трансформації тексту	18
1.4 Вибір та застосування чат-боту ChatGPT як технології для трансформації тексту	21
1.4.1 Важливість використання інструментів на основі штучного інтелекту для трансформації текстів	21
1.4.2 Інструменти на основі штучного інтелекту на сучасному ринку	22
1.4.3 Вибір ChatGPT як найкращого інструменту для трансформації тексту	23
1.4.4 Математична основа ефективності ChatGPT	24
1.5 Формулювання завдання на роботу	25
Висновки до розділу 1	26
РОЗДІЛ 2 КОНЦЕПЦІЯ ФУНКЦІОНАЛЬНОСТІ	28
2.1 Високорівневий опис вимог до функціональності	28
2.1.1 Функціональні вимоги для ролі Адміністратор	28
2.1.2 Функціональність для ролі Користувач	31
2.1.3 Нефункціональні вимоги	33
2.2 Дослідження побудови запиту в чат-бот ChatGPT	36
2.2.1 Дослідження найкращих практик побудови запиту для трансформації тексту	36

	7
2.2.2 Дослідження аналогічних функціональностей ринку	40
Висновки до розділу 2	44
РОЗДІЛ 3 ДЕТАЛЬНИЙ ОПИС ФУНКЦІОНАЛЬНОСТІ	45
3.1 Детальний опис користувацької функціональності	45
3.1.1 Функціональність ролі Адміністратор	45
3.1.2 Функціональність ролі Користувач	52
3.2 Приклади запитів в чат-бот ChatGPT	67
Висновки до розділу 3	71
РОЗДІЛ 4 РОБОТА ФУНКЦІОНАЛЬНОСТІ	72
4.1 Опис роботи функціональності	72
4.2 Результати запитів з трансформації тексту в чат-бот ChatGPT	74
Висновки до розділу 4	78
РОЗДІЛ 5. ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ФУНКЦІОНАЛУ ТРАНСФОРМАЦІЇ ТЕКСТУ З УРАХУВАННЯМ КОНТЕКСТУ З ВИКОРИСТАННЯМ ЧАТ-БОТУ CHATGPT	80
5.1. Формування завдання проектування	80
5.2. Обґрунтування функцій програмного продукту	81
5.3. Обґрунтування системи параметрів програмного продукту	85
5.4. Аналіз експертного оцінювання параметрів	86
5.5. Аналіз рівня якості варіантів реалізації функцій	89
5.6. Економічний аналіз варіантів розробки ПП	90
5.7. Вибір кращого варіанта техніко-економічного рівня	93
Висновки до 5 розділу	94
ВИСНОВКИ	95
ПЕРЕЛІК ПОСИЛАНЬ	98
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ	100

ВСТУП

Сьогодні у світі цифрових технологій на перший план виходить здатність трансформувати текст з урахуванням контексту, використовуючи найсучасніші інструменти штучного інтелекту, такі як чат-бот ChatGPT. Цей проект присвячений використанню чат-бота ChatGPT для трансформації тексту, з акцентом на контекстно-орієнтованих методологіях, які ефективно адаптуються до текстового контенту для будь-яких ситуативних потреб. Такі перетворення є актуальними та важливими, оскільки вони дозволяють більш точну і змістовну комунікацію в різних сферах, покращуючи як особисту, так і професійну взаємодію.

Об'єктом дослідження є застосування чат-бота ChatGPT для трансформації тексту, а предметом дослідження – вивчення стратегій трансформації тексту з урахуванням контексту.

Основним завданням і метою цієї бакалаврської роботи є розробка концепції універсальної функціональності для перетворення тексту у веб-додатках. Ця функціональність повинна бути надійною, гнучкою та універсальною, щоб знайти застосування в широкому спектрі галузей. Ця функціональність зможе бути як самостійним додатком, так і бути легко інтегрованою в будь-який існуючий веб-додаток, тим самим значно підвищуючи його корисність і масштабованість. Мета цієї роботи - запропонувати функціональну концепцію, яка відповідатиме поточним потребам перетворення тексту і намагатиметься передбачити майбутні потреби, таким чином гарантуючи, що функціонал можна буде надійно використовувати протягом тривалого часу у багатьох випадках використання.

У цій роботі проводиться комплексне дослідження та розробка концепції функціональності трансформації тексту за допомогою чат-бота ChatGPT. У роботі визначається, що таке трансформація тексту, надаючи також приклади для кращого розуміння. Далі визначено мету трансформації тексту та

проаналізовано, наскільки актуальним є її застосування в наш час. Також проводиться аналіз інструменту ChatGPT як механізму реалізації контекстної трансформації тексту. Отриманий результат комплексного аналізу свідчить про те, що ChatGPT є найбільш оптимальним рішенням з огляду на розширені можливості в сенсі ефективного та результативного вирішення численних і різноманітних складних завдань з трансформації тексту.

Далі в цій роботі аналізуються та формулюються високорівневі функціональні та нефункціональні вимоги необхідні для реалізації надійної системи перетворення тексту. Досліджено та задокументовано найкращі практики створення запитів для перетворення тексту в ChatGPT, щоб дотримуватися рекомендацій щодо формування ефективного запиту, який підвищить якість та релевантність вихідного тексту. Також проводиться дослідження ринку, щоб оцінити, який подібний функціонал є в найпопулярніших додатках, що дає уявлення про конкурентне середовище і тенденції на поточний момент.

На основі цього дослідження вимоги деталізуються і створюються прототипи користувацького інтерфейсу для візуалізації функціонування додатку. Також пропонуються деякі можливі запити, які були отримані в результаті вивчення найкращих практик та задокументованих результатів.

Далі, наводиться опис роботи функціональності, розробленої на основі описаних вимог, також підкріплений UML діаграмою послідовностей. І окрім детального опису функціональних можливостей, у цій роботі представлені самі трансформації тексту, які показують тексти "до" і "після". Результати цих трансформацій є прямим наслідком ретельно розроблених запитів, описаних у розділі 3.2.

Кожна трансформація демонструє значне покращення якості тексту. З цими реальними досягненнями, проект підтверджує, що розроблений функціонал, як з точки зору технічних вимог, так і з точки зору побудови запитів, був теоретично правильним, практично можливим і корисним в багатьох різних сферах і додатках. Це також ще раз підтверджує тезу про те, що

вивчена і описана концепція функціональності дійсно застосовна і актуальна в умовах зростання вимог до сучасних веб-додатків, що є переконливим аргументом для її інтеграції в різні галузі.

РОЗДІЛ 1 ВИБІР ТЕХНОЛОГІЇ СНАТГРТ ДЛЯ ТРАНСФОРМАЦІЇ ТЕКСТУ

1.1 Визначення та приклади трансформації тексту

1.1.1 Визначення трансформації тексту

У сучасну цифрову епоху текст пронизує кожен аспект нашого життя, слугуючи основним засобом комунікації в різних сферах. Від академічних публікацій і бізнес-звітів до онлайн-контенту та технічних інструкцій - повсюдність тексту не викликає сумнівів. Незважаючи на таке широке використання, не всі тексти оптимально відповідають своєму призначенню. Вони можуть страждати від поганої структури, бракувати необхідної термінології або не відповідати вимогам, необхідним для ефективного розуміння людиною чи машинного опрацювання.

Трансформація тексту – це процес модифікації тексту з метою підвищення його зрозумілості, ефективності та придатності для виконання певної функції. Це передбачає реструктуризацію речень, добір точної лексики та приведення тексту у відповідність до контексту, щоб він відповідав поставленій меті.

1.1.2 Приклади трансформації тексту

У якості прикладів трансформації тексту можна навести такі.

1. Оптимізація резюме.

Резюме можна трансформувати, включивши в нього певні ключові слова з опису вакансії, щоб воно відповідало вимогам роботодавця. Сюди входить не лише включення галузевого жаргону, але й стратегічне представлення

інформації, наприклад, про досягнення та кваліфікацію, у спосіб, що максимізує вплив на роботодавця.

2. Маркетингові матеріали.

Маркетингові тексти можна трансформувати, щоб вони резонували з різними цільовими демографічними групами. Це включає в себе адаптацію тону, стилю та змісту, щоб привабити різні вікові групи, культурне походження або конкретні потреби споживачів. Наприклад, опис продукту може бути скоригований, щоб підкреслити розкіш та ексклюзивність для елітного сегменту ринку або доступність та практичність для споживачів з обмеженим бюджетом.

3. Контент для соціальних мереж.

Адаптація тексту для соціальних мереж передбачає, що він має бути цікавим, лаконічним і придатним для швидкого споживання. Це часто включає використання неформальної мови, смайликів та хештегів, щоб збільшити охоплення та видимість публікації.

4. Машинна обробка - SEO-оптимізація.

Трансформація тексту для пошукової оптимізації (SEO – Search Engine Optimization) передбачає модифікацію контенту веб-сайту для покращення видимості в пошукових системах. Це включає стратегічну інтеграцію ключових слів, оптимізацію мета-тегів та описів, а також забезпечення якості контенту відповідно до алгоритмів пошукових систем.

5. Академічне письмо.

Трансформація наукової статті може передбачати коригування мови відповідно до публікаційних стандартів, забезпечення використання відповідних технічних термінів і структурування контенту відповідно до стилістичних настанов академічної спільноти.

6. Посібники користувача.

Технічна документація, така як посібники користувача, потребує трансформації, щоб бути зрозумілою та дієвою для кінцевого

користувача. Разом з використанням візуальних засобів, таких як діаграми, це може включати спрощення складних технічних описів і структурування контенту в покроковому форматі для зручності використання.

7. Юридичні документи.

Трансформація юридичних текстів передбачає уточнення мови без зміни юридичної точності чи намірів. Це може означати переклад складного юридичного жаргону більш простою мовою для контрактів, які мають бути зрозумілими споживачам, або реструктуризацію термінів для відповідності місцевим законам і правилам.

8. Новинні статті.

У журналістиці трансформація тексту може передбачати зміну фокусу статті залежно від платформи або аудиторії. Новина для місцевої газети може підкреслювати місцевий вплив, тоді як та сама історія для національного видання вимагатиме ширшої перспективи. Це включає зміни в заголовку, першому абзаці та деталях, на яких акцентується увага в статті.

9. Навчальні тексти.

Навчальний контент часто потребує трансформації для різних рівнів навчання. Для учнів початкових класів складні наукові поняття можуть бути спрощені та проілюстровані прикладами і візуальними образами. Для студентів університетів ті ж самі поняття можуть бути представлені з поглибленим аналізом і специфічною термінологією.

10. Наука про дані, обробка природної мови.

Трансформація тексту має вирішальне значення для попередньої обробки текстових даних, щоб покращити продуктивність моделі машинного навчання. Наприклад, в аналізі настроїв текст трансформується за допомогою таких кроків, як токенізація, нормалізація та векторизація. Ці перетворення передбачають розбиття тексту на керовані частини, стандартизацію формату, видалення непотрібних слів і перетворення

тексту на числові вектори. Цей оброблений текст потім використовується для навчання моделей, які класифікують настрої у відгуках клієнтів як позитивні, негативні або нейтральні, що дозволяє компаніям ефективно отримувати інформацію з відгуків клієнтів.

Ці приклади ілюструють, що трансформація тексту має вирішальне значення для адаптації повідомлень та інформації відповідно до конкретних вимог та ефективного донесення до цільової аудиторії. Кожен контекст вимагає унікального підходу до модифікації тексту, що підкреслює універсальність і необхідність навичок трансформації тексту в різних професійних сферах.

1.1.3 Математична основа для трансформації тексту

Для кількісної оцінки ефективності трансформації тексту можна використовувати такі метрики, як частота терміна до частоти документа – TF-IDF (TF – term frequency, IDF – inverse document frequency) – яка вимірює важливість слова в документі в колекції [1].

Це може бути корисно для оптимізації резюме, оскільки показує, наскільки рідко (а отже, більш значуще) певні терміни зустрічаються в подібних документах, але часто - в описі вакансії, що свідчить про їх стратегічне включення.

Ще одним прикладом використання метрики TF-IDF є Пошукова оптимізація (SEO). Автори веб-контенту можуть використовувати TF-IDF для оптимізації своїх статей або веб-сторінок, щоб покращити їхній рейтинг у пошукових системах. Аналізуючи ключові слова, які є важливими в конкурентному наборі документів, але недостатньо представлені в їхньому контенті, автори можуть стратегічно додавати ці терміни, щоб підвищити релевантність і видимість.

Загалом, у моделях машинного навчання на основі тексту вибір правильних ознак (слів) має вирішальне значення. TF-IDF допомагає в цьому, зважуючи терміни в документах, таким чином надаючи пріоритет словам, важливим для завдань класифікації або прогнозування, і зменшуючи шум загальних і менш інформативних слів.

Формула TF-IDF визначається наступним чином:

$$\text{TF-IDF}(t,d,D)=\text{TF}(t,d)\times\text{IDF}(t,D),$$

де t – термін;

d – документ;

D – колекція документів;

$\text{TF}(t,d)$ – Term Frequency, частота терміну t в документі d ;

$\text{IDF}(t,d)$ – Inverse document frequency, обчислюється як $\log\left(\frac{N}{|\{d \in D; t \in d\}|}\right)$;

N – загальна кількість документів у колекції.

Застосовуючи цю метрику, можна доказово вдосконалити текст, щоб він краще відповідав своєму призначенню, таким чином гарантуючи, що трансформація є не лише суб'єктивною, але й емпірично підтвердженою.

1.2 Цілі трансформації тексту

Отже, як було описано в пункті 1.1, трансформація тексту передбачає переписування тексту, щоб забезпечити його відповідну термінологію, структуру та стиль, таким чином оптимізуючи його для конкретних цілей і підвищуючи його придатність для виконання певних функцій. Основними цілями трансформації тексту є покращення читабельності, забезпечення точності переданих значень та оптимізація ефективності обробки як людиною,

так і машиною. Конкретні цілі можуть суттєво відрізнятись залежно від сфери, в якій використовується текст.

Наприклад, повертаючись до прикладу трансформації резюме з підрозділу 1.1, мета полягає в тому, щоб відформатувати і адаптувати резюме таким чином, щоб воно ефективно проходило через системи відстеження кандидатів (ATS – Applicant Tracking Systems) і привертало увагу рекрутерів. Це передбачає стратегічне включення ключових слів, реструктуризацію інформації, щоб підкреслити релевантний досвід, і забезпечення сумісності макета зі стандартними алгоритмами систем відстеження кандидатів .

У маркетингу трансформація тексту має на меті узгодити контент із голосом бренду та стратегіями залучення клієнтів. Це може включати коригування тону, стилю та форматування маркетингових матеріалів, щоб вони резонували з цільовою демографічною групою, оптимізацію розміщення закликів до дії та використання переконливої мови, яка стимулює конверсію.

Також можна трансформувати описи продуктів, маркетингові повідомлення і навіть мультимедійний контент для відображення місцевих мов, культурних особливостей, сленгу та суспільних норм. Це матиме на меті адаптувати контент до різних культурних контекстів, гарантуючи, що він резонує з місцевою аудиторією та відповідає культурним нормам. Ця мета має вирішальне значення для глобальних брендів і сервісів, яким необхідно підтримувати єдиний голос бренду, водночас локалізуючи свої повідомлення.

Трансформація контенту для соціальних мереж допомагатиме створювати публікації, які не лише привертають увагу користувачів, але й заохочують до взаємодії.

В машинній обробці (SEO) мета трансформації тексту – підвищити видимість веб-сайту та його рейтинг на сторінках пошукової видачі (SERP – Search engine results page). Це досягається за рахунок того, що контент стає більш релевантним і легко інтерпретується алгоритмами пошукових систем.

В академічному письмі, трансформація наукових робіт може бути націлена на підвищення видимості наукової роботи в академічних базах даних і пошукових системах, що полегшить іншим дослідникам її пошук і цитування.

Трансформування текстів технічної документації здійснюється з метою роблення технічної документації чіткою, зрозумілою та дієвою для кінцевого користувача.

Метою трансформації тексту для юридичних документів є уточнення використовуваної мови, щоб зробити її зрозумілою без шкоди для юридичної точності та змісту документа. Це гарантує, що юридичні тексти будуть доступними та відповідатимуть чинним законам і правилам.

Для новинних статей, метою є адаптація контенту для різних платформ і аудиторій, щоб він був актуальним і цікавим. Це передбачає коригування фокусу, мови та деталей історії відповідно до інтересів та очікувань конкретної читацької аудиторії.

Трансформація навчальних текстів має на меті адаптацію та представлення інформації у спосіб, який є доречним і зрозумілим для учнів різних освітніх рівнів, що сприяє кращому розумінню та залученню їх до матеріалу.

У галузі науки про дані, зокрема в обробці природної мови, трансформація тексту має важливе значення для підвищення ефективності інформаційно-пошукових систем. Такі методи, як стемінг, лематизація та використання синонімів, можуть суттєво допомогти розширити діапазон успішних відповідей на запити, зробити системи більш надійними та зручними для користувача.

Трансформація тексту також може мати за мету регулювання тону і настрою тексту, щоб він краще відповідав бажаному емоційному впливу або тону. Це важливо для будь-якої комунікації з роботодавцями, клієнтами, взаємодії в соціальних мережах, тощо та може бути застосовано в будь-якій галузі, де використовуються тексти.

1.3 Цінність та актуальність трансформації тексту

Отже, трансформація тексту – це потужний інструмент, який використовується в безлічі сфер, підвищуючи функціональність і доступність текстового контенту для досягнення конкретних цілей. Його корисність очевидна в різних сферах – від управління персоналом до цифрового маркетингу, що ілюструє його широке застосування та вирішальну роль у сучасній цифровій взаємодії.

У сфері управління персоналом трансформація тексту відіграє життєво важливу роль в оптимізації резюме та описів вакансій. Завдяки включенню конкретних ключових слів і структуруванню контенту відповідно до систем відстеження кандидатів (ATS), резюме з більшою ймовірністю пройдуть первинний відбір, а описи вакансій стануть більш ефективними в залученні відповідних кандидатів. Наприклад, трансформація резюме з метою висвітлення відповідних навичок і досвіду в чіткій, лаконічній формі підвищує ймовірність привернути увагу рекрутерів і HR-фахівців. Отже, в сучасному світі, де ринок праці є висококонкурентним, а цифрові платформи домінують у процесі рекрутингу, трансформація тексту в кадровому менеджменті має вирішальне значення для підвищення видимості та ефективності резюме та описів вакансій. Узгоджуючи контент із системами відстеження кандидатів і адаптуючи його до конкретних критеріїв вакансії, трансформація тексту значно підвищує шанси кандидатів бути поміченими рекрутерами і відповідати потрібним профілям вакансій, спрощуючи процес найму на роботу.

У сфері пошукової оптимізації (SEO) трансформація тексту гарантує, що цифровий контент буде знайдений цільовою аудиторією. Завдяки стратегічному включенню високоцінних ключових слів та оптимізації мета-тегів і описів, веб-контент з більшою ймовірністю з'являється у верхніх рядках результатів пошукової видачі, тим самим збільшуючи відвідуваність веб-сайту і потенційні потоки доходів. У сучасному цифровому ландшафті, де видимість в Інтернеті

безпосередньо пов'язана з успіхом бізнесу, трансформація тексту за допомогою пошукової оптимізації (SEO) має важливе значення. Оптимізуючи веб-контент за допомогою високоцінних ключових слів та ефективних мета-тегів, компанії можуть значно покращити свою видимість у пошукових системах, залучаючи більше цільового трафіку. Таке підвищення видимості не лише стимулює відвідуваність веб-сайту, але й збільшує потенційні потоки доходів, що робить SEO критично важливим компонентом сучасних стратегій цифрового маркетингу.

У маркетингу та рекламі трансформація тексту відповідно до споживчої психології та ринкових тенденцій може значно підвищити рівень залученості та конверсії. Створення переконливих закликів до дії, використання переконливої мови та адаптація контенту до конкретних демографічних сегментів роблять рекламу ефективнішою, а кампанії успішнішими. У швидкоплинному світі маркетингу та реклами трансформація тексту має вирішальне значення для приведення контенту у відповідність до швидкозмінної споживчої психології та ринкових тенденцій. Стратегічно розробляючи повідомлення з переконливими закликами до дії та переконливою мовою, адаптованою до конкретних демографічних сегментів, компанії можуть підвищити рівень залученості та конверсії. Такий цілеспрямований підхід гарантує, що маркетингові зусилля ефективніше резонують з аудиторією, максимізуючи вплив та успіх рекламних кампаній у висококонкурентному цифровому середовищі.

В освіті трансформація тексту допомагає адаптувати навчальні матеріали до різних стилів і рівнів навчання. Спрощення складних понять для молодших школярів або доповнення матеріалу поглибленими науковими посиланнями для студентів вищих навчальних закладів робить навчальний контент більш доступним і цікавим. У сучасному різноманітному освітньому середовищі трансформація тексту має важливе значення для адаптації матеріалів до різних стилів навчання та рівнів підготовки студентів. Спрощуючи складні поняття для молодших школярів і збагачуючи контент поглибленими науковими посиланнями для студентів університетів, викладачі можуть зробити навчання

більш доступним і цікавим. Така кастомізація покращує розуміння та запам'ятовування матеріалу, надаючи всім студентам можливість досягти успіху в навчанні.

Отже, можна зробити висновок, що актуальність трансформації тексту в сучасному світі важко переоцінити. Оскільки цифрові платформи продовжують поширюватися та інформаційне перевантаження домінує, обсяг даних, що генеруються і споживаються щодня в інтернеті, в бізнесі та в особистому спілкуванні, є величезним. Трансформація тексту стає ключовим інструментом в управлінні цим потоком інформації, роблячи дані не лише більш доступними, але й більш дієвими. Здатність швидко доопрацьовувати та адаптувати текст для конкретних цілей допомагає окремим особам та організаціям вирізнятися серед шуму та ефективно доносити цільові повідомлення до аудиторії. Оскільки цифрові комунікації продовжують розвиватися, потреба в оптимізованому контенті, який легко шукати, який є зрозумілим і цікавим, стає все більш критичною. Трансформація тексту не лише підвищує зручність використання інформації, але й гарантує, що взаємодія – між людьми чи між людьми і машинами – буде максимально ефективною та результативною. Використовуючи можливості трансформації тексту, зацікавлені сторони в різних секторах можуть ефективніше досягати своїх комунікаційних та операційних цілей, що робить цю практику наріжним каменем сучасних цифрових стратегій. Таке широке застосування підкреслює її цінність у формуванні взаємодії, рішень і результатів у світі, що стрімко оцифровується.

1.4 Вибір та застосування чат-боту ChatGPT як технології для трансформації тексту

1.4.1 Важливість використання інструментів на основі штучного інтелекту для трансформації текстів

Трансформація тексту вручну за своєю суттю є трудомістким та неефективним завданням. Більш того, традиційний підхід до модифікації тексту вручну не лише забирає значну кількість часу та ресурсів, але й підвищує ймовірність невідповідностей та помилок. Вони можуть виникати через втому людини, суб'єктивну інтерпретацію і людські обмеження в управлінні великими обсягами даних з високою точністю.

Окрім того, люди схильні до помилок, особливо коли трансформований текст призначений для взаємодії з системами штучного інтелекту. Люди природно упереджені у використанні мови і часто не здатні обробляти інформацію з об'єктивністю і швидкістю, необхідною для ефективної взаємодії зі штучним інтелектом. Ця невідповідність стає особливо очевидною, коли йдеться про завдання, що вимагають розуміння та генерування мовних шаблонів у масштабі - наприклад, навчання моделей ШІ або обробки запитів клієнтів у різних демографічних групах і регіонах. Люди, як правило, не можуть імітувати моделі "мислення" систем штучного інтелекту, які призначені для роботи зі складними алгоритмами та обробки даних неупереджено і швидко.

Цей розрив між людськими можливостями та вимогами передових програм зі штучним інтелектом значно підкреслює зростаючу потребу в автоматизованих інструментах у сфері трансформації тексту.

Особливо корисними є інструменти на основі штучного інтелекту. Інструменти зі штучним інтелектом можуть ефективно виконувати повторювані завдання, підтримувати узгодженість у великих наборах даних, виконувати завдання з обробки мови, динамічно адаптуючись до тонкощів людської мови. Вони значно підвищують точність і значно скорочують час і зусилля, пов'язані з

ручною обробкою. Отже, автоматизація за допомогою штучного інтелекту не тільки впорядковує нудні, повторювані завдання, але й забезпечує рівень послідовності та точності, якого важко досягти вручну. Використання передових алгоритмів штучного інтелекту дає змогу трансформувати текст так, щоб він оптимально відповідав вимогам подальшої обробки.

Ці можливості є не просто покращенням, а необхідністю в галузях, де точність і швидкість обробки тексту безпосередньо впливають на успіх роботи та якість обслуговування. Таким чином, перехід до автоматизації трансформації тексту є не лише питанням ефективності, але й стратегічним імперативом для використання повного потенціалу технологічних досягнень у галузі штучного інтелекту.

1.4.2 Інструменти на основі штучного інтелекту на сучасному ринку

На сучасному ринку існує вже безліч інструментів на основі штучного інтелекту для створення або трансформації тексту.

Одними з найбільш популярних є ChatGPT, Quillbot, Jasper, Wordtune, PrepostSEO, та багато інших.

Більш того, велика кількість популярних додатків з високим рейтингом, які спочатку не були розроблені з можливостями штучного інтелекту, тим чи іншим чином інтегрували штучний інтелект у свої основні функціональні можливості. Таке вдосконалення значно покращило користувацький досвід, простоту використання і дало чудові результати, підвищивши загальну продуктивність додатків [2]. Прикладами таких є Copilot (Bing), Grammarly, Notion, Socratic.

1.4.3 Вибір ChatGPT як найкращого інструменту для трансформації тексту

Серед безлічі інструментів, доступних для перетворення тексту, ChatGPT від OpenAI виділяється своїми складними можливостями та передовою технологічною базою [3]. Побудований на основі найсучаснішої архітектури перетворювача, ChatGPT особливо добре генерує зв'язні та контекстно-релевантні текстові відповіді на основі заданого запиту[4]. Здатність розуміти і створювати контекстно-залежний текст робить ChatGPT безцінним інструментом для широкого спектру застосувань.

Майстерність ChatGPT у створенні тексту полягає не лише у створенні граматичних речень, а й у створенні контенту, який є лінгвістично точним, стилістично адаптованим і глибоко узгодженим з нюансами людської мови. Це робить його надзвичайно корисним для завдань, які вимагають високого ступеня лінгвістичної точності та гнучкості. Наприклад, у сфері обслуговування клієнтів ChatGPT може автономно генерувати відповіді, які не тільки правильні, але й адаптовані для відображення голосу бренду компанії та емоційного тону клієнта. У створенні контенту він допомагає авторам, пропонуючи правки, розширюючи ідеї чернеток або навіть створюючи повноцінні статті на основі короткого конспекту.

Крім того, нейронна мережа, що лежить в основі ChatGPT, навчена на різноманітних текстах з Інтернету, що дозволяє їй працювати з різними темами з розумінням, яке часто можна порівняти з розумінням людини-експерта. Таке широке навчання дозволяє ChatGPT добре працювати в багатьох галузях, включаючи маркетинг, де він може створювати переконливі рекламні тексти, і в технічних галузях, де він може перекладати складну інформацію доступною мовою, зрозумілою для неспеціалістів [5].

Універсальність і адаптивність ChatGPT роблять його ідеальним вибором для тих, хто прагне підвищити ефективність і якість своїх текстових операцій.

Незалежно від галузі застосування, здатність ChatGPT легко інтегруватися в існуючі робочі процеси та вдосконалювати їх за допомогою потужних можливостей штучного інтелекту - це те, що вирізняє його з-поміж інших інструментів для перетворення текстів[6].

1.4.4 Математична основа ефективності ChatGPT

Ефективність ChatGPT ґрунтується на математичних основах, що лежать в його основі, насамперед на механізмах уваги та рівнях нейронної мережі [7], які дозволяють йому розуміти та генерувати текст, подібний до людського. Здатність моделі контекстуалізувати слова в реченні математично представлена вагами уваги, які обчислюються таким чином [8]:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V ,$$

де Q, K, V – це запити, ключі та матриці значень, отримані з вхідного тексту, а d_k – це розмірність ключів. Ця формула допомагає моделі визначити, які частини тексту є найбільш релевантними, що дозволяє їй генерувати відповіді, які відповідають контексту.

Отже, розширені можливості ChatGPT роблять його оптимальним вибором для автоматизації завдань перетворення тексту в різних галузях. Його використання не тільки підвищує ефективність і якість обробки тексту, але й звільняє людські ресурси, щоб зосередитися на більш стратегічних, творчих завданнях, тим самим підвищуючи продуктивність та інновації на підприємствах.

1.5 Формулювання завдання на роботу

У підрозділах 1.1-1.4 проведено ретельний аналіз концепції трансформації тексту, включаючи її цілі та значення в сучасному цифровому ландшафті, а також огляд сучасних технологій, доступних для трансформації тексту; було прийнято рішення обрати чат-бот ChatGPT від OpenAI як технологію для цієї роботи.

Отже, завданням на цю бакалаврську дипломну роботу є розробка концепції комплексного та універсального функціоналу для перетворення тексту в рамках веб-додатків, яка буде не лише надійною та гнучкою, але й достатньо універсальною, щоб її можна було використовувати для різних застосунків у різних галузях. Ця універсальність гарантуватиме, що функціонал зможе легко інтегруватися в будь-який існуючий веб-додаток, підвищуючи його корисність і масштабованість.

В основі функціоналу лежатимуть потужні можливості API-інтерфейсів ChatGPT від OpenAI для виконання складних перетворень тексту. Інтегруючи ці передові інструменти штучного інтелекту, функціональність використовуватиме найсучасніші мовні моделі, які підвищують ефективність і точність завдань з обробки та трансформації тексту. Інтеграція буде розроблена таким чином, щоб бути надійною і гнучкою, забезпечуючи механізм, який може легко та швидко впоратися з різноманітними потребами в перетворенні тексту.

Висновки до розділу 1

У сучасному цифровому світі тексти всюди, вони пронизують кожен аспект особистого та професійного життя. Трансформація тексту вдосконалює

ці тексти, щоб забезпечити їхню оптимізацію для конкретних цілей, підвищуючи зрозумілість, залученість і функціональність у різних сферах.

Основна мета трансформації тексту - адаптувати текст таким чином, щоб він ефективно передавав задумане повідомлення, резонував з цільовою аудиторією і виконував конкретні операційні завдання, наприклад, відповідав практикам SEO або вписався в діалогову систему, керовану штучним інтелектом.

Цінність трансформації тексту полягає в тому, що вона робить інформацію доступнішою та зручнішою для використання, сприяючи тим самим кращій комунікації, кращому розумінню та підвищенню ефективності бізнес-процесів. Це особливо актуально в сучасному світі через експоненціальне зростання виробництва та споживання даних; ефективне перетворення тексту допомагає керувати цим потоком інформації, гарантуючи, що ключові повідомлення будуть виділені та зрозумілі різним аудиторіям.

Серед інструментів, доступних для перетворення тексту, ChatGPT виділяється як оптимальний вибір завдяки своїм розширеним можливостям генерування зв'язного, контекстно-орієнтованого тексту. Створений на основі складної моделі штучного інтелекту, ChatGPT може автоматизувати та покращити процес перетворення тексту, надаючи високоякісний контент, який відповідає конкретним потребам своїх користувачів. Здатність безперешкодно інтегруватися в різні робочі процеси робить його не лише потужним інструментом для підвищення продуктивності, а й важливим активом для використання всього потенціалу текстового спілкування в цифрову епоху.

Метою та завданням цієї роботи є розробка концепції функціоналу для перетворення тексту у веб-додатку, розробленого таким чином, щоб бути достатньо універсальним для використання в різних сферах. Функціонал використовуватиме можливості API-інтерфейсів ChatGPT від OpenAI для виконання перетворень тексту. Інтеграція цього передового інструменту штучного інтелекту підвищуватиме ефективність і точність обробки тексту, забезпечить задоволення різноманітні потреб в перетворенні тексту.

РОЗДІЛ 2 КОНЦЕПЦІЯ ФУНКЦІОНАЛЬНОСТІ

2.1 Високорівневий опис вимог до функціональності

Щоб задовольнити вимоги до функціональності та різні потреби різних типів користувачів, програма матиме різні функціональні можливості, пристосовані для двох ролей користувачів: Адміністратор і Користувач.

2.1.1 Функціональні вимоги для ролі Адміністратор

Роль Адміністратора наділена функціональними можливостями, призначеними для організації ефективної взаємодії між користувацьким додатком і API-інтерфейсами ChatGPT. Основний фокус обов'язків адміністратора полягає у створенні та управлінні запитом для трансформації тексту, який надсилається до ChatGPT. Ця функціональність є критично важливою, оскільки вона створює основу для того, як програма обробляє і перетворює текст на основі даних, введених користувачем, і заздалегідь визначених шаблонів.

Отже, основною функціональністю Адміністратора є складання запиту. Адміністратор матиме можливість написання та налаштування запитів, які надсилаються до ChatGPT. Це включає в себе не тільки формулювання тексту запитів, але й включення змінних, які пізніше будуть замінені реальними даними користувача під час виконання запиту. Така динамічна вставка даних гарантує, що запити будуть контекстуально відповідними, персоналізованими, і релевантними для кожної взаємодії з користувачем, підвищуючи застосовність і точність перетворень тексту.

Адміністратор також матиме можливість переглядати інструкції та рекомендації щодо створення запитів. Це допоможе Адміністратору керувати процесом створення запитів, гарантуючи, що запити будуть не тільки ефективними, але й оптимізованими для роботи з ChatGPT. Це матиме вирішальне значення для підтримання якості та ефективності процесів перетворення тексту за допомогою програми.

Отже, при розробці концепції функціоналу з API ChatGPT було прийнято ключове рішення забезпечити повну конфігурацію запитів, а не їх прописування в програмному коді.

Це рішення ґрунтується на розумінні того, що якщо запити прописані в програмному коді, це буде значно обмежувати можливість багаторазового використання функціональності, гнучкість і потенціал для забезпечення чудового користувацького досвіду в різних доменах і додатках. Конфігурованість має важливе значення для підтримки універсальності та широкої застосовності програми, дозволяючи їй служити різним цілям і адаптуватися до різних потреб користувачів. Тобто, конфігуровані запити запобігають обмеженню додатку однією областю або функцією. Така гнучкість гарантує, що з мінімальними налаштуваннями додаток можна перепрофілювати або інтегрувати в різноманітні середовища, максимізуючи його корисність і подовжуючи життєвий цикл.

Прописування тексту запитів в програмному коді також вимагало б від команди розробників досліджень для визначення найбільш ефективних структур запитів. Цей процес може значно подовжити цикл розробки та затримати реалізацію функцій. Дозволяючи Адміністратору налаштовувати запити через інтерфейс додатку, ми зменшуємо початкове навантаження на розробників і забезпечуємо швидке оновлення та адаптацію запиту за потреби. Більш того, часто кінцеві користувачі розуміють специфіку контексту, в якому працює додаток, краще за команду розробників. Отже, якщо дозволити саме користувачам адаптувати запити до їхніх конкретних потреб, це призведе до

більш точної та релевантної взаємодії з ChatGPT, підвищуючи загальну ефективність відповідей.

Більш того, мовні моделі штучного інтелекту та бізнес-середовища динамічні, вони розвиваються і змінюються з часом. Прописаний в коді запит може стати менш ефективним або навіть застарілим у міру того, як відбуваються ці зміни. Конфігуровані Адміністратором запити дають змогу легко вносити зміни без додаткового програмування, що дозволяє додатку залишатися актуальним серед оновлень алгоритмів штучного інтелекту та зміни бізнес-вимог. Така адаптивність має вирішальне значення для підтримки довгострокової життєздатності та ефективності програми.

До того ж, покладатися на команду розробників для постійного оновлення та модифікації прописаних у коді запитів може бути ресурсомістким. Існують практичні проблеми, такі як наявність команди розробників та бюджетні обмеження, які можуть перешкоджати своєчасному оновленню. Конфігуровані запити дають можливість кінцевим користувачам самостійно вносити необхідні корективи, тим самим зменшуючи залежність від втручання розробників і уникаючи потенційних перебоїв у роботі сервісу та користувацького досвіду.

Наостанок, конфігурованість запитів означає, що функціональність може бути легко перепрофільована та додана в будь-який веб-додаток з мінімальними технічними налаштуваннями та функціональними покращеннями.

Отже, завдяки можливості Адміністратора конфігурувати запити, додаток не тільки підвищує свою адаптивність до мінливих вимог користувачів і технологічних ландшафтів, а й позиціонує себе як цінний інструмент, здатний надавати ефективні результати в різних галузях. Такий підхід зменшує потенційні ризики, пов'язані із застарілими або неефективними запитами, тим самим забезпечуючи актуальність та ефективність програми на конкурентному ринку.

2.1.2 Функціональність для ролі Користувач

Роль Користувача наділена функціональними можливостями, зосередженими навколо основної мети – перетворення тексту. Ця роль матиме інтерфейси, необхідні для введення тексту, отримання перетворених результатів та можливого вдосконалення результатів відповідно до конкретних вимог.

Отже, Користувач матиме можливість вводити текст, який потребує трансформації, безпосередньо в систему. Це може бути будь-який текст – від простого речення, яке потрібно перефразувати, до більш складних абзаців, що вимагають значного переписування.

Також Користувач зможе надати додатковий контекст щодо тексту, який потребує трансформації. Це може включати мету перетворення, довідкову інформацію про предметну область, а також конкретні параметри або критерії, яким має відповідати перетворений текст. Наприклад, користувач, який бажає трансформувати текст свого резюме до конкретної вакансії, може ввести опис вакансії, на яку він претендує. Потім система може використовувати цю інформацію, щоб виділити або змінити аспекти резюме, щоб більш точно відповідати опису вакансії, підкреслюючи навички та досвід, які особливо важливі для цієї посади. Надання контексту є дуже важливою особливістю цього функціонала, адже контекст значно підвищує релевантність і точність процесу перетворення, адаптуючи його до конкретних цілей або вимог (більш детально про це буде викладено в підрозділі 2.2).

Після введення тексту та, опціонально, додаткового контексту, система використовуватиме інтегровані API-інтерфейси ChatGPT для обробки і видачі перетвореної версії тексту, дотримуючись попередньо визначених або налаштованих Адміністратором запитів, які визначають характер трансформації.

Користувач не лише отримуватиме трансформований текст, але зможе й далі взаємодіяти з системою. Якщо початкове перетворення не повністю відповідає очікуванням Користувача або його конкретним потребам, Користувач зможе запросити систему зробити подальше перетворення. Ця функціональність має вирішальне значення для досягнення високого рівня задоволеності користувачів, оскільки дозволяє застосовувати більш індивідуальний підхід, коли користувачі можуть вказати, які аспекти трансформації вони хотіли б відкоригувати або покращити. Отже, Користувач також зможе безпосередньо впливати на логіку трансформації, гарантуючи, що результат більш точно відображає його вимоги та очікування.

Отже, як і для концепції функціональностей Адміністратора, для концепції функціональностей ролі Користувача метою є максимізація універсальності, загальна застосовність і розширення можливостей користувачів. Отже, основна функціональність – перетворення тексту – має бути не тільки потужною та ефективною, але й надзвичайно адаптивною, здатною легко інтегруватися в будь-який існуючий додаток, та забезпечуючою чудовий користувацький досвід.

Інтерфейс трансформації, де користувач надає текст, який потребує трансформації, та додатковий контекст до нього, може бути як самодостатнім додатком, так і інтегрованою частиною майже будь-якого додатка в будь-якій галузі. Отже, це гарантуватиме, що функціонал трансформації тексту може бути частиною майже будь-якої програми.

До того ж, однією з ключових особливостей системи є можливість для Користувачів взаємодіяти з процесом перетворення тексту та динамічно керувати ним. Користувачі не обмежуються прийняттям початкового результату. Натомість вони можуть дати системі команду переглянути результат відповідно до конкретних інструкцій, які вони зможуть надати у своєму інтерфейсі. Це може включати в себе збільшення або зменшення довжини відповіді, підкреслення певних моментів або інтеграцію додаткових концепцій та ідей. Отже, система дає користувачам можливість запитувати зміни або повторювати

трансформації, забезпечуючи їм контроль над кінцевим результатом. Ця функція є життєво важливою для адаптації контенту до точних специфікацій, що особливо важливо в сценаріях, де потрібна нюансована або вузькоспецифічна інформація. Вона дозволяє користувачам ітеративно вдосконалювати текст, поки він не задовольнить їхні потреби, тим самим підвищуючи загальну корисність і зручність користування додатком.

2.1.3 Нефункціональні вимоги

При розробці концепції функціональності будь-якого додатку, нефункціональним вимогам часто не приділяється стільки уваги, скільки функціональним. Однак врахування цих нефункціональних аспектів є критично важливим для загальної цілісності та успіху програми. Ці вимоги, як правило, стосуються роботи системи, а не поведінки, підкреслюючи такі аспекти, як безпека, зручність використання та відповідність вимогам, які є невід'ємними для підтримки довіри користувачів та ефективності [9].

По-перше, забезпечення дотримання Загального регламенту захисту даних (GDPR – General Data Protection Regulation) має першорядне значення, особливо коли йдеться про функціональність, що інтегрується з такими сервісами, як ChatGPT, які обробляють значні обсяги потенційно конфіденційних даних [11]. Вкрай важливо впроваджувати суворі заходи для обробки персональних даних користувачів з максимальною обережністю, гарантуючи, що вони не будуть ненавмисно передані третім особам без явної згоди користувача. В рамках дотримання GDPR і для зміцнення довіри користувачів, функціональність повинна включати механізми для чіткого інформування користувачів про те, як використовуються їхні дані, а також про особливості будь-якого обміну даними, який може відбуватися з третіми сторонами, такими як ChatGPT. Це передбачає оновлення Умов та положень

(Terms and Conditions), щоб відобразити будь-які зміни або вдосконалення, пов'язані з новими функціональними можливостями. Користувачі повинні бути ознайомлені з цими оновленнями до того, як вони почнуть користуватися новими інтегрованими функціями. Щоб забезпечити належне інформування всіх користувачів про ці зміни, можна розіслати електронні листи з поясненням змін в обробці даних та вимог до згоди користувачів, використати функціонал сповіщень (“Notifications“) у додатку для сповіщення користувачів при вході або під час взаємодії з відповідними функціями, інтегрувати цю інформацію в процес онбордингу нових користувачів, або ж впровадити спливаючі повідомлення (pop-up), які вимагатимуть підтвердження користувача, перш ніж він зможе продовжити використання нових функцій.

Також критично важливою нефункціональною вимогою є безпечна передача даних. При передачі даних через Інтернет, вони можуть бути перехоплені. Отже, вирішальне значення для захисту цих даних має використання безпечних технологій передачі даних, таких як протоколи SSL (Secure Socket Layer)/TLS (Transport Layer Security), що забезпечують встановлення безпечного і зашифрованого зв'язку між сервером і клієнтом.

Ще однією важливою нефункціональною вимогою до функціональності трансформації тексту є продуктивність, зокрема швидкість відгуку системи. Вкрай важливо, щоб інтеграція з API ChatGPT працювала ефективно, щоб користувачі відчували мінімальну затримку. Очікується, що сервер повинен відповідати менш ніж за одну хвилину з моменту запиту. Така швидка реакція має вирішальне значення для підтримки чудового користувацького досвіду.

Також важливою нефункціональною вимогою до функціональності трансформації тексту є здатність ефективно обробляти тексти різної довжини та складності. Ця універсальність має вирішальне значення, оскільки користувачам може знадобитися трансформувати тексти від лаконічних речень до більш складних текстів, що складаються з кількох абзаців або списків. Система повинна вміти обробляти всі ці варіанти з однаковою продуктивністю,

гарантуючи, що незалежно від розміру і структури вхідного тексту, вихідні дані залишатимуться незмінно точними та якісними.

Ще однією ключовою нефункціональною вимогою до нашої функціональності є здатність ефективно трансформувати текст різними мовами. Враховуючи глобальний характер сучасних додатків і різноманітні мовні потреби користувачів, важливо, щоб функціонал для трансформації тексту міг безперешкодно працювати з різними мовами. Можливість обробки декількох мов розширює універсальність програми, роблячи її придатною для різних сфер та аудиторій. Це відкриває численні можливості для поширення цієї функціональності на ринку. Також можливість обробки декількох мов гарантує, що користувачі можуть взаємодіяти з додатком своєю рідною або принаймні бажаною мовою, що підвищує зручність користування та покращує користувацький досвід. ChatGPT, розроблений OpenAI, здатний розуміти і генерувати текст більш ніж 25 мовами [12], включаючи як основні світові мови, такі як англійська, іспанська, французька, німецька, китайська, японська, корейська, італійська, голландська та португальська, так і українську мову. Отже, вибір ChatGPT як основної технології, що забезпечує можливості трансформації тексту в нашому додатку, забезпечує задоволення цієї вимоги.

Також вкрай важливою нефункціональною вимогою є простота використання і зрозумілість функціоналу. Враховуючи, що перетворення тексту є нетиповою функцією, дуже важливо забезпечити, щоб користувачі з різними технологічними навичками могли ефективно взаємодіяти з цією функцією та використовувати її. Ця вимога є важливою для ролі Користувача, та є вкрай важливою для ролі Адміністратора. Користувачі повинні знайти функціонал перетворення тексту інтуїтивно зрозумілим і простим. Щоб досягти цього, додаток повинен надавати докладні інструкції про те, як вводити текст для перетворення, яких результатів очікувати і як ініціювати зміни або повторне надсилання. Це можуть бути покрокові інструкції, інтерактивні розділи довідки та наочні посібники, які допоможуть користувачам пройти весь процес. Для Адміністраторів основна увага має бути зосереджена на наданні їм знань та

інструментів, необхідних для ефективного налаштування та керування запитом на перетворення тексту. Вичерпні інструкції про те, як створювати ефективні запити до ChatGPT, динамічно використовувати змінні та дотримуватися найкращих практик у формулюванні запитів, є вкрай важливими. Це можуть бути розширені посібники користувача, навчальні модулі та регулярні оновлення найновіших стратегій оптимізації запитів. Оскільки загальна продуктивність і ефективність функції перетворення тексту значною мірою залежать від якості цих запитів, надання Адміністраторам якісної інструкції та підтримки має вирішальне значення.

2.2 Дослідження побудови запиту в чат-бот ChatGPT

2.2.1 Дослідження найкращих практик побудови запиту для трансформації тексту

За допомогою запитів користувачі повідомляють ChatGPT про свої потреби та завдання. Ефективність запитів безпосередньо впливає на якість відповіді штучного інтелекту. Коректна постановка запиту є необхідною для використання повного потенціалу чат-боту ChatGPT [13, 14].

Отже, критично важливим є приділення уваги до проектування запиту. Проектування запиту – стратегічний процес проектування, уточнення та оптимізації запитів до штучного інтелекту для точної передачі намірів користувача мовній моделі, такої як ChatGPT. Основна мета цієї практики – забезпечити отримання точних, релевантних і послідовних відповідей від штучного інтелекту. Ефективне проектування запиту значно підвищує якість та продуктивність взаємодії зі штучним інтелектом.

На вибір і створення ефективних запитів впливають кілька факторів: наміри користувача, які визначають бажаний результат взаємодії; можливості моделі, які охоплюють сильні та слабкі сторони ШІ; специфіка предметної

області, яка передбачає використання відповідної мови та контексту; ясність, яка гарантує, що запит буде однозначним та зрозумілим, щоб запобігти його неправильне тлумачення штучним інтелектом. Разом ці фактори допомагають створювати запити, які є не лише чіткими та конкретними, але й адаптованими для використання всього потенціалу ChatGPT у наданні корисних та контекстуально відповідних відповідей.

Існують різні методи, які покращують взаємодію між користувачами та ChatGPT, гарантуючи, що відповіді будуть не лише точними, але й конкретно відповідатимуть потребам користувача.

Однією з основних методик є надання чітких, однозначних, і конкретних інструкцій у запитах. Це мінімізує двозначності, які можуть призвести до нечітких або нецільових відповідей, та безпосередньо вказує, яка інформація або тип відповіді є бажаним. Наприклад, замість того, щоб запитувати "Покращити це резюме: {текст резюме}", більш ефективним буде запит "Трансформуй текст резюме, щоб він містив ключові слова з опису цільової вакансії. Текст резюме: {текст резюме}. Опис цільової вакансії: {текст резюме}". Таким чином, ChatGPT спрямований на конкретний аспект трансформації тексту – включення ключових слів. Це забезпечує ефективну трансформацію, завдяки якій користувач з більшою ймовірністю досягає своєї цілі – в даному випадку, працевлаштування.

Ще одним важливим принципом побудови запитів для трансформації тексту є чітке включення мети трансформації в сам запит. Чітко визначивши мету трансформації, ChatGPT може ефективніше сфокусувати свої можливості, щоб гарантувати, що отриманий текст точно відповідатиме бажаному результату. Цей метод гарантує, що процес трансформації є не просто загальною модифікацією тексту, а свідомим і цілеспрямованим покращенням. Він узгоджує зміст із конкретними вимогами ситуації – чи то заява на роботу, чи то маркетингове коригування, чи то будь-який інший контекст, де бажані конкретні результати від трансформації тексту. Чітке формулювання мети трансформації в підказці значно підвищує ймовірність досягнення успішного й

бажаного результату, роблячи процес трансформації тексту ефективнішим і таким, що відповідає очікуванням користувача. Прикладом чіткого включення мети трансформації в сам запит є таке формулювання: “Перероби наданий текст резюме, щоб підкреслити навички, які необхідні для успішного подання на цю вакансію.”.

Іншою ефективною методикою є надання явних обмежень у запитах. Це також відіграє вирішальну роль у формуванні характеристик відповідей ChatGPT. Встановивши в запиті такі обмеження, як довжина відповіді (в символах, словах, абзацах, тощо), формат, або ступінь деталізації, користувачі можуть контролювати вихідні дані відповідно до конкретних вимог та потреб, що робить результати роботи функціоналу більш практичними і зручними для користувача. Отже, до запиту можна додати уточнення на кшталт: “Трансформований текст повинен містити приблизно 500 символів та бути у форматі маркованого списку.”.

Включення контексту і прикладів в запитах ще більше підвищує релевантність і точність відповідей штучного інтелекту. Це передбачає включення довідкової інформації або сценаріїв, які створюють основу для того, щоб ChatGPT генерував більш контекстно-відповідні відповіді. Повертаючись до прикладу з резюме, можна надати ChatGPT більше контексту про цільову вакансію, компанію, сферу діяльності, або особисті дані користувача, такі як його біографія або інші дані з його профіля.

Також надзвичайно ефективним методом є надання інструкції щодо включення конкретних фраз або певних висновків, зроблених штучним інтелектом. Отже, до прикладу запиту на трансформацію резюме можна додати інструкцію на кшталт: “Напиши, що я мотивований та швидко навчаюсь. Також визнач та підкресли м'які навички, якими я володію, виходячи виду спорту, яким я займаюся – {спорт користувача}.”.

Крім того, важливо зберігати баланс між наданням чітких інструкцій і творчими можливостями моделі. Це означає, що в ідеалі запити мають бути не лише чіткими та цілеспрямованими, але й достатньо відкритими, щоб

дозволити ChatGPT творчо підходити до пошуку відповідей. Повертаючись до прикладу в абзаці вище (“Також визнач та підкресли м'які навички, якими я володію, виходячи виду спорту, яким я займаюся – {спорт користувача}.”), таке формулювання запиту дозволяє моделі включати в трансформований текст свої висновки, використовуючи її повний потенціал.

Наостанок, можна розглянути більш складні методи, які дозволяють користувачам тонко налаштувати свою взаємодію з ChatGPT, підвищуючи якість і специфічність відповідей ШІ. Однією з найважливіших стратегій є регулювання характеристик відповіді за допомогою таких налаштувань, як температура і контроль токенів. Контроль температури дозволяє користувачам диктувати випадковість або непередбачуваність відповідей ChatGPT; нижча температура призводить до більш передбачуваних і консервативних результатів, тоді як вища температура може генерувати більш креативні та різноманітні відповіді. З іншого боку, контроль токенів допомагає керувати довжиною вихідних даних, гарантуючи, що відповіді не будуть ані занадто короткими, ані надто багатослівними, таким чином узгоджуючи їх з конкретними вимогами поставленого завдання.

Загалом, гарною практикою для створення ефективних запитів є ітеративне тестування та вдосконалення запитів. Цей процес включає в себе створення початкових запитів, спостереження за відповідями, які вони викликають, а потім внесення необхідних коректив для покращення ясності та точності. Цей ітеративний цикл допомагає точно налаштувати структуру запитів, гарантуючи, що вони будуть оптимально розроблені, щоб ефективно донести до ШІ наміри користувача. Таке безперервне вдосконалення може значно підвищити якість результатів роботи штучного інтелекту, зробивши їх більш релевантними та корисними для конкретних потреб користувача.

Отже, добре продумані запити значно підвищують точність, релевантність і ефективність відповідей ChatGPT, долаючи розрив між намірами користувача і розумінням штучного інтелекту. Завдяки точному і продуманому проектуванню запитів користувачі можуть адаптувати широкі можливості ChatGPT до

конкретних завдань, гарантуючи, що результат не тільки відповідає очікуванням, але й часто перевершує їх. Це максимізує ефективність штучного інтелекту в багатьох сферах, включаючи сферу трансформації тексту. Дотримуючись описаних найкращих практик для побудови запитів, користувачі можуть підвищити ефективність їхньої взаємодії з ChatGPT, досягаючи більш точних, креативних, релевантних і контекстно-відповідних результатів, роблячи цю модель ще більш потужним активом у широкому діапазоні застосувань.

2.2.2 Дослідження аналогічних функціональностей ринку

Як було помічено в пункті 1.4.2, деякі провідні програми використовують штучний інтелект для покращення можливостей трансформації тексту. Ці програми не використовують власні моделі штучного інтелекту, натомість вони інтегруються з існуючими передовими мовними моделями штучного інтелекту, щоб розширити свої функціональні можливості. Наприклад, Copilot (Bing) використовує комбінацію потужних моделей, зокрема ChatGPT-4 від OpenAI, модель Prometheus від Microsoft і DALL-E 3 [15], щоб забезпечити надійний користувацький досвід, керований штучним інтелектом. Аналогічно, Notion розширює свою платформу великими мовними моделями (LLM – Large Language Models) від різних постачальників, таких як Anthropic, OpenAI, а також специфічною моделлю Notion, розробленою в партнерстві з Cohere [16]. Ще одним яскравим прикладом є Grammarly, який вирішив використовувати Azure OpenAI як базову технологію для своїх вдосконалених пропозицій з граматики, пунктуації та стилю [17].

Ці інтеграції ілюструють тенденцію в індустрії цифрових інструментів, коли додатки використовують сильні сторони існуючих технологій штучного інтелекту, щоб забезпечити складні та зручні для користувача функції. Таким чином, вони не тільки підвищують ефективність і результативність текстових

операцій, а й посилюють залучення користувачів завдяки більш розумним та інтуїтивно зрозумілим інтерфейсам. Покладання на такі інтеграції у сфері штучного інтелекту також підкреслює важливість вибору правильного постачальника штучного інтелекту, оскільки можливості базової моделі безпосередньо впливають на якість функцій додатку та загальний користувацький досвід. Ці приклади слугують орієнтирами в конкурентному середовищі, підкреслюючи, як інтеграція штучного інтелекту може мати вирішальне значення для диференціації продуктів на переповненому ринку.

Через міркування інтелектуальної власності отримати детальну та достовірну інформацію про точні запити, які надсилаються моделям штучного інтелекту різними програмами, досить складно. Такі запити часто вважаються інтелектуальною власністю, адже специфічний спосіб створення запитів – так звані фірмові префікси запитів – може суттєво відрізнити функціонал та результати роботи програм, які використовують однакові базові моделі штучного інтелекту. Наприклад, обидві програми трансформації тексту Jasper і CopyAI використовують GPT-3 як базову модель штучного інтелекту. Проте кожна з них розробляє унікальні запити, які адаптують відповіді штучного інтелекту відповідно до їхніх конкретних продуктових стратегій.

У світлі цих обмежень найдоцільніший метод зрозуміти, як ці додатки взаємодіють з моделями штучного інтелекту, - це обґрунтоване припущення, підтримане методами зворотного інжинірингу. Цей підхід, хоч і непрямий, може дати уявлення про потенційні конфігурації та структури запитів, що використовуються програмами.

Прикладом такої дослідницької роботи є детальний аналіз [18] функцій штучного інтелекту Notion, проведений за допомогою методів зворотної інженерії. Автор цього аналізу зміг визначити типи запитів, які, ймовірно, використовує Notion, на основі поведінки та результатів, що спостерігалися в додатку. В результаті цього ретельного аналізу було зроблено припущення, що Notion використовує спеціально розроблені запити для активації окремих

функцій штучного інтелекту, які призначені для оптимізації користувацького досвіду та підвищення ефективності інструменту.

Наприклад, для функції “Help Me Write” (укр. “Допоможи мені писати”), автор припускає такий запит: “You are an assistant helping a user write more content in a document based on a prompt. Output in markdown format. Do not use links. Do not include literal content from the original document. Use this format, replacing text in brackets with the result. Do not include the brackets in the output: Output in [Identified language of the document]: [Output based on the prompt, in markdown format.]” [19].

Для функції “Continue Writing” (укр. “Продовжуй писати”): “You are an assistant helping a user write a document. Output how the document continues, no more than 3 sentences. Output in markdown format. Do not use links. Use this format, replacing text in brackets with the result. Do not include the brackets in the output: Continuation in [Identified language of the document]: [Continuation of the document in markdown format, no more than 3 sentences.]” [19].

Для функції “Summarize” (укр. “Підсумувати”): “You are an assistant helping summarize a document. Use this format, replacing text in brackets with the result. Do not include the brackets in the output: Summary in [Identified language of the document]: [One-paragraph summary of the document using the identified language.]” [19].

Для функції “Outline” (укр. “Конспект”): “You are an assistant helping to draft an outline for a document. Use this format, replacing text in brackets with the result. Do not include the brackets in the output: Outline in [Identified language of the topic]: # [Title of document] [Bulleted list outline of document, in markdown format]” [19].

Для функції “Find action items” (укр. “Знайди пункти дій”): “You are an assistant helping find action items inside a document. An action item is an extracted task or to-do found inside of an unstructured document. Use this format, replacing text in brackets with the result. Do not include the brackets in the output: List of action items in [Identified language of the document]: [List of action items in the

identified language, in markdown format. Prefix each line with "- []" to make it a checkbox.]" [19].

Для функції “Recruiting Email” (укр. “Рекрутинговий лист”): “You are an assistant helping to draft a personalized recruiting email. Use this format, replacing text in brackets with the result. Do not include the brackets in the output: Recruiting email in [Identified language of the notes]: # [Recruiting email title] [Recruiting email subject] [Recruiting email body]” [19].

Отже, проаналізувавши можливі запити до штучного інтелекту в програмі Notion, стає очевидним, що вони включають в себе багато з кращих практик, про які йшлося раніше. Зокрема, вони включають чіткі та однозначні інструкції до запитів, гарантуючи, що штучний інтелект має чіткі вказівки, яких слід дотримуватися. Ця ясність має вирішальне значення для отримання релевантних і точних відповідей. Крім того, ці запити зазвичай містять конкретні обмеження щодо розміру та формату бажаного результату, що допомагає оптимізувати відповіді, щоб вони відповідали вимогам користувача. Більше того, хоча запити ретельно сфокусовані на отриманні конкретних типів інформації або відповідей, вони також створені таким чином, щоб бути достатньо відкритими. Такий баланс дозволяє моделям штучного інтелекту використовувати свої можливості для генерування креативних та інноваційних рішень. Такий стратегічний підхід до формулювання запитів не лише підвищує ефективність взаємодії зі штучним інтелектом, а й оптимізує якість результатів, забезпечуючи їхню корисність та креативність.

Висновки до розділу 2

У другому розділі спочатку були описані високорівневі вимоги та загальна концепція функціональності, був наданий концептуальний огляд дизайну

системи та її передбачуваної роботи. Вибрані рішення були проаналізовані та була підтверджена їх ефективність та відповідність потребам. Також були описані та детально проаналізовані нефункціональні вимоги до функціональності.

Також були ретельно досліджені та задокументувані найкращі практики створення запитів до штучного інтелекту. Це дослідження включало глибокий аналіз ефективних стратегій створення запитів, які підвищують функціональність і точність трансформації тексту, керованої штучним інтелектом. Крім того, було проведене дослідження ринку, щоб вивчити існуючі функціональності трансформації тексту на основі штучного інтелекту в провідних програмах. Це дослідження мало на меті вивчити поточні впровадження в галузі та підтвердити ефективність і актуальність найкращих практик, які були описані. Проаналізувавши, як популярні додатки використовують штучний інтелект у своїй діяльності, ми змогли провести паралелі та підкреслити практичну застосовність описаних стратегій побудови запитів.

РОЗДІЛ 3 ДЕТАЛЬНИЙ ОПИС ФУНКЦІОНАЛЬНОСТІ

3.1 Детальний опис користувацької функціональності

У цьому підрозділі, присвяченому детальному опису користувацької функціональності, буде використано історії користувачів (User Stories) та критерії прийнятності у форматі Gherkin (Acceptance Criteria) як основні методи специфікації. Там, де це доречно, історії користувачів та критерії прийнятності будуть доповнені діаграмами та прототипами інтерфейсів, створеними у Figma, щоб проілюструвати, як повинен виглядати і функціонувати користувацький інтерфейс.

3.1.1 Функціональність ролі Адміністратор

Адміністратор матиме наступні функціональності: Сторінка конфігурації запиту, Додавання змінних в поле запиту, Конфігурація запиту, Сторінка інструкції зі складання запиту. Історії користувача та критерії прийнятності викладені в табл. 3.1, 3.2, 3.3, 3.4 відповідно та підкріплені рисунками з прототипами, де це доцільно.

Таблиця 3.1 – Сторінка конфігурації запиту

Історія користувача	Як Адміністратор, я хочу переглянути сторінку конфігурації запиту, щоб переглянути поточно збережений запит та редагувати його.
Критерії прийнятності	Сценарій 1: Адміністратор успішно переглядає сторінку конфігурації запиту (рис. 3.1) <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Адміністратор ● When: Адміністратор знаходиться на сторінці конфігурації запиту ● Then: Адміністратор бачить заголовок: “Конфігурація запиту”

Закінчення табл. 3.1

Критерії прийнятності	<ul style="list-style-type: none"> ● And: Адміністратор бачить текст під заголовком сторінки: “Налаштування запиту до штучного інтелекту для трансформації тексту користувача. Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть тут.”. Слово “тут” також містить гіперпосилання на сторінку інструкції (див. табл. 3.4) ● And: Адміністратор може натиснути на гіперпосилання на сторінку інструкції в тексті під заголовком сторінки, щоб бути перенаправленим на сторінку інструкції (див. табл. 3.4) ● And: Адміністратор бачить текстове поле для запиту “Запит”, заповнене поточно збереженим запитом ● And: Адміністратор може редагувати поле “Запит” (див. табл. 3.3) ● And: Адміністратор бачить опцію “Додати змінну” (див. табл. 3.2) ● And: Адміністратор бачить кнопку “Відміна”, що є неактивною доки він не почне редагувати поле “Запит” ● And: Адміністратор бачить кнопку “Зберегти”, що є неактивною доки він не почне редагувати поле “Запит”
-----------------------	---

Конфігурація запиту

Налаштування запиту до штучного інтелекту для трансформації тексту користувача.
Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть [тут](#).

[Додати змінну](#)

Запит

Перероби текст публікації для соціальної мережі, щоб підвищити залученість користувачів. Використовуй енергійну та заманливу мову. Включи пряме запитання, щоб заохотити до коментарів. Встав переконливий заклик до цільової дії. Враховуй додатковий контекст, якщо він наданий. Трансформований текст повинен містити не більше ніж 150 символів. Наданий текст публікації для соціальної мережі: %%вміст поля "Початковий текст"%%. Наданий додатковий контекст: %%вміст поля "Додатковий контекст"%%.

Відмінити Зберегти

Рисунок 3.1 – Сторінка конфігурації запиту

Таблиця 3.2 – Додавання змінних в поле запиту

Історія користувача	Як Адміністратор, я хочу додати змінні до запити, щоб надіслані запити містили дані Користувача, а отже були контекстуально відповідними та персоналізованими.
Критерії прийнятності	<p>Передумова:</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Адміністратор <p>Сценарій 1: Адміністратор успішно відкриває спливаюче вікно “Додати змінну” (рис. 3.2)</p> <ul style="list-style-type: none"> ● Given: Адміністратор знаходиться на сторінці конфігурації запити ● When: Адміністратор натискає на кнопку “Додати змінну” ● Then: відкривається спливаюче вікно, яке містить наступні елементи: <ul style="list-style-type: none"> ○ Заголовок: “Додати змінну” ○ Текст: “Виберіть, які дані користувача будуть передані до штучного інтелекту час трансформації тексту” ○ Список усіх доступних змінних ○ Кнопка “закрити вікно” <p>Сценарій 2: Адміністратор успішно додає змінну до поля запити</p> <ul style="list-style-type: none"> ● Given: Адміністратор знаходиться на спливаючому вікні “Додати змінну” ● When: Адміністратор натискає на змінну ● Then: спливаюче вікно “Додати змінну” закривається ● And: заповнювач (placeholder) обраної додається до поля “Запит” в місце, де знаходиться курсор, та має такий формат: “%%змінна%%” <p>Сценарій 3: Адміністратор успішно відміняє додавання змінної до поля запити</p> <ul style="list-style-type: none"> ● Given: Адміністратор знаходиться на спливаючому вікні “Додати змінну” ● When: Адміністратор натискає на кнопку “закрити вікно” ● Then: спливаюче вікно “Додати змінну” закривається ● And: ніякого заповнювача (placeholder) не додано до поля “Запит”

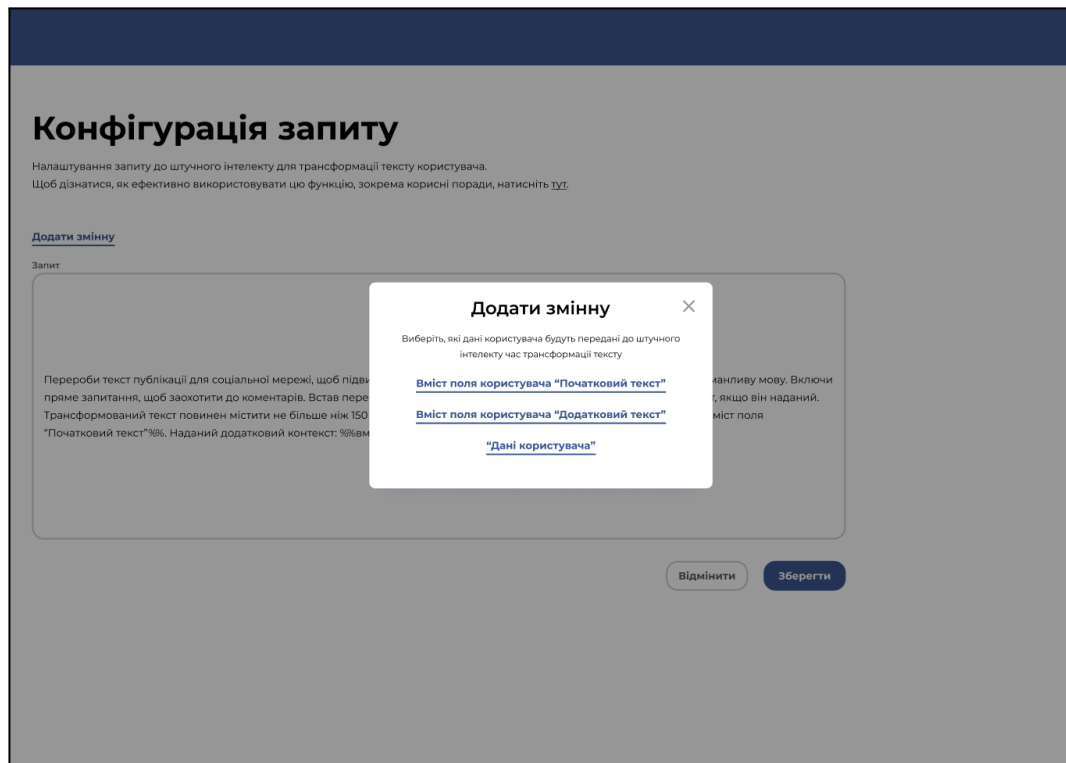


Рисунок 3.2 – Спливаюче вікно “Додати змінну”

Таблиця 3.3 – Конфігурація запиту

Історія користувача	Як Адміністратор, я хочу конфігурувати запит, що буде надсилатися в ChatGPT, щоб забезпечити ефективну трансформацію тексту Користувача.
Критерії прийнятності	<p>Передумова (рис. 3.3):</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Адміністратор ● And: Адміністратор знаходиться на сторінці Конфігурації запиту (див. табл. 3.1) ● And: Адміністратор редагує поле “Запит”, отже, кнопки “Відміна” та “Зберегти” є активними <p>Сценарій 1: Адміністратор успішно зберігає оновлений запит (рис. 3.4)</p> <ul style="list-style-type: none"> ● Given: поле “Запит” заповнено валідним значенням: <ul style="list-style-type: none"> ○ поле не порожнє, та ○ поле містить допустиму кількість символів: менше або дорівнює 5000 ● When: Адміністратор натискає на кнопку “Зберегти” ● Then: оновлений запит зберігається ● And: кнопки “Відміна” та “Зберегти” знов стають неактивними ● And: спливаюче повідомлення про успіх “Запит збережено” відображається протягом 5 секунд

Закінчення табл. 3.3

Критерії прийнятності	<p>Сценарій 2: Адміністратор не може зберегти оновлений запит через невалідне значення поля “Запит” (рис. 3.5)</p> <ul style="list-style-type: none"> ● Given: поле “Запит” заповнено невалідним значенням: <ul style="list-style-type: none"> ○ поле порожнє, або ○ поле містить недопустиму кількість символів: більше 5000 ● When: Адміністратор натискає на кнопку “Зберегти” ● Then: оновлений запит не зберігається ● And: відповідне повідомлення про помилку відображається під полем “Запит”: <ul style="list-style-type: none"> ○ якщо поле порожнє: “Будь ласка, заповніть поле” ○ якщо поле містить недопустиму кількість символів (більше 5000): “Поле має містити менше 5000 символів” ● And: кнопки “Відміна” та “Зберегти” все ще активні <p>Сценарій 3: Адміністратор успішно відмінює зміни до запиту</p> <ul style="list-style-type: none"> ● Given: поле “Запит” заповнено будь-яким значенням ● When: Адміністратор натискає на кнопку “Відміна” ● Then: зміни до значення поля “Запит” відмінюються, поле залишається заповненим останнім збереженим значенням ● And: кнопки “Відміна” та “Зберегти” знов стають неактивними
-----------------------	--

Конфігурація запиту

Налаштування запиту до штучного інтелекту для трансформації тексту користувача. Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть [тут](#).

[Додати зміну](#)

Запит

Перероби текст публікації для соціальної мережі, щоб підвищити залученість користувачів. Використовуй енергійну та заманливу мову. Включи пряме запитання, щоб заохотити до коментарів. Встав переконливий заклик до цільової дії. Враховуй додатковий контекст, якщо він наданий. Трансформований текст повинен містити не більше ніж 150 символів. Наданий текст публікації для соціальної мережі: %вміст поля "Початковий текст"% . Наданий додатковий контекст: %вміст поля "Додатковий контекст"%.

Відмінити
Зберегти

Рисунок 3.3 – Адміністратор редагує поле “Запит”

Конфігурація запиту

Налаштування запиту до штучного інтелекту для трансформації тексту користувача.
Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть [тут](#).

[Додати зміну](#)

Запит

Перероби текст публікації для соціальної мережі, щоб підвищити залученість користувачів. Використовуй енергійну та заманливу мову. Включи пряме запитання, щоб заохотити до коментарів. Встав переконливий заклик до цільової дії. Враховуй додатковий контекст, якщо він наданий. Трансформований текст повинен містити не більше ніж 150 символів. Наданий текст публікації для соціальної мережі: %%вміст поля "Початковий текст"%%. Наданий додатковий контекст: %%вміст поля "Додатковий контекст"%%.

Рисунок 3.4 – Запит збережено

Конфігурація запиту

Налаштування запиту до штучного інтелекту для трансформації тексту користувача.
Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть [тут](#).

[Додати зміну](#)

Запит

Будь ласка, заповніть поле

Рисунок 3.5 – Помилка через невалідне значення поля “Запит”

Таблиця 3.4 – Сторінка інструкції зі складання запиту

Історія користувача	Як Адміністратор, я хочу переглянути інструкцію зі складання запиту, щоб скласти ефективний запит, що забезпечує ефективну трансформацію тексту Користувача.
Критерії прийнятності	<p>Сценарій 1: Сторінка з інструкцією успішно відкривається</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Адміністратор ● And: Адміністратор знаходиться на сторінці конфігурації запиту (див. табл. 3.1) ● When: Адміністратор натискає на гіперпосилання на сторінку інструкції в тексті під заголовком сторінки ● Then: сторінка інструкції відкривається в тій же вкладці браузера ● And: Адміністратор може переглянути її контент (див. Сценарій 2) <p>Сценарій 2: Адміністратор успішно переглядає контент сторінки з інструкцією (рис. 3.6)</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Адміністратор ● When: Адміністратор знаходиться на сторінці інструкції ● Then: Адміністратор бачить заголовок сторінки: “Інструкція з конфігурації запиту” ● And: Адміністратор бачить текст сторінки з інструкцією, що також містить посилання на сторінку конфігурації запиту (див. табл. 3.1)



Рисунок 3.6 – Сторінка інструкції з конфігурації запиту

3.1.2 Функціональність ролі Користувач

Користувач матиме наступні функціональності: Сторінка трансформації тексту, Трансформація тексту, Копіювання трансформованого тексту, Запит на покращення результату трансформації та перегляд варіантів трансформації, Видалення початкового тексту, додаткового контексту і трансформованого тексту, Сторінка інструкції з трансформації тексту. Історії користувача та критерії прийнятності викладені в таблицях 3.5, 3.6, 3.7, 3.8, 3.9, 3.10 відповідно та підкріплені рисунками з прототипами, де це доцільно.

Таблиця 3.5 – Сторінка трансформації тексту

Історія користувача	Як Користувач, я хочу переглянути сторінку трансформації тексту, щоб трансформувати текст та переглядати результати трансформації.
Критерії прийнятності	<p>Сценарій 1: Користувач успішно переглядає сторінку трансформації тексту (рис. 3.7)</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Користувач ● When: Користувач знаходиться на сторінці трансформації тексту ● Then: Користувач бачить заголовок сторінки: “Трансформація тексту” ● And: Користувач бачить текст під заголовком сторінки: “Трансформація тексту відповідно до конкретного контексту за допомогою штучного інтелекту. Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть тут.” Слово “тут” також містить гіперпосилання на сторінку інструкції (див. табл. 3.10) ● And: Користувач може натиснути на гіперпосилання на сторінку інструкції в тексті під заголовком сторінки, щоб бути перенаправленим на сторінку інструкції (див. табл. 3.10) ● And: Користувач бачить текстове поле для тексту “Початковий текст” ● And: Користувач може редагувати поле “Початковий текст” ● And: Користувач бачить текстове поле для “Додатковий контекст” ● And: Користувач може редагувати поле “Додатковий контекст” ● And: Користувач бачить кнопку “Трансформувати текст” ● And: Користувач бачить кнопку “Видалити” ● And: Користувач бачить порожнє поле для результату трансформації “Трансформований текст” ● And: Користувач не може редагувати поле “Трансформований текст”

Рисунок 3.7 – Сторінка конфігурації запиту

Таблиця 3.6 – Трансформація тексту

Історія користувача	Як Користувач, я хочу трансформувати свій текст, щоб він був більш відповідним згідно моїх цілей.
Критерії прийнятності	<p>Передумова (рис. 3.8):</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Користувач ● And: Користувач знаходиться на сторінці трансформації тексту (див. табл. 3.5) ● And: Користувач редагує поле “Початковий текст” ● And: опціонально, Користувач редагує поле “Додатковий контекст” <p>Сценарій 1: Користувач успішно надсилає запит трансформації тексту (рис. 3.9)</p> <ul style="list-style-type: none"> ● Given: поле “Початковий текст” заповнено валідним значенням: <ul style="list-style-type: none"> ○ поле не порожнє, та ○ поле містить допустиму кількість символів: менше або дорівнює 5000 ● And: поле “Додатковий контекст” заповнено валідним значенням: <ul style="list-style-type: none"> ○ поле містить допустиму кількість символів: менше або дорівнює 5000 ● When: Користувач натискає на кнопку “Трансформувати текст”

Продовження табл. 3.6

Критерії прийнятності	<ul style="list-style-type: none"> ● Then: поля “Початковий текст” та “Додатковий контекст” стають нередагованими ● And: на кнопці “Трансформувати текст” Користувач бачить лоадер (loader) ● And: в полі “Трансформований текст” Користувач бачить лоадер ● And: Користувач бачить кнопку “Видалити” (див. табл. 3.9) ● And: надсилається запит до ChatGPT, сконфігурований Адміністратором (див. табл. 3.3) <p>Сценарій 2: Користувач успішно отримує результат трансформації тексту (рис. 3.10)</p> <ul style="list-style-type: none"> ● Given: шага Сценарію 1 ● When: система отримала результат трансформації від ChatGPT ● Then: в полі “Трансформований текст” Користувач бачить трансформований текст ● And: Користувач бачить кнопку “Скопіювати трансформований текст” (див. табл. 3.7) ● And: замість кнопки “Трансформувати текст” Користувач бачить кнопку “Покращити далі” (див. табл. 3.8) <p>Сценарій 3: Користувач не може отримати результат трансформації тексту через технічну помилку (рис. 3.11)</p> <ul style="list-style-type: none"> ● Given: шага Сценарію 1 ● When: стається будь-яка технічна помилка (сервер не відповідає протягом однієї хвилини, тощо) ● Then: Користувач бачить спливаюче повідомлення протягом 5 секунд: “Сталась помилка. Будь ласка, спробуйте ще раз.” ● And: поля “Початковий текст” та “Додатковий контекст” стають редагованими ● And: Користувач бачить кнопку “Трансформувати текст” без лоадера ● And: Користувач бачить поле “Трансформований текст” без лоадера <p>Сценарій 4: Користувач не може надіслати запит на трансформацію тексту через невалідне значення полей “Початковий текст” або “Додатковий контекст” (рис. 3.12)</p> <ul style="list-style-type: none"> ● Given: поле “Початковий текст” заповнено невалідним значенням: <ul style="list-style-type: none"> ○ поле порожнє, ○ поле містить недопустиму кількість символів: більше 5000 ● And/Or: поле “Додатковий контекст” заповнено невалідним значенням: <ul style="list-style-type: none"> ○ поле містить недопустиму кількість символів: більше 5000 ● When: Користувач натискає на кнопку “Трансформувати текст” ● Then: відповідне/відповідні повідомлення про помилку/помилки відображаються під відповідним полем:
-----------------------	---

Закінчення табл. 3.6

Критерії прийнятності	<ul style="list-style-type: none"> ○ якщо поле “Початковий текст” порожнє: “Будь ласка, заповніть поле” ○ якщо поле “Початковий текст” або “Трансформувати текст” містить недопустиму кількість символів (більше 5000): “Поле має містити менше 5000 символів” ● And: поля “Початковий текст” та “Додатковий контекст” залишаються редагованими ● And: Користувач все ще бачить кнопку “Трансформувати текст” ● And: запит до ChatGPT не надсилається
-----------------------	--

Трансформація тексту

Трансформація тексту відповідно до конкретного контексту за допомогою штучного інтелекту.
Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть [тут](#).

Початковий текст

Ми щойно випустили новий продукт. Він доступний вже зараз, перевірте його на нашому сайті.

Додатковий контекст

Блог про технології. Потрібен текст, що залучить користувачів.

Трансформований текст

Видалити Трансформувати текст

Рисунок 3.8 – Користувач редагує поля “Початковий текст” та “Додатковий контекст”

Трансформація тексту

Трансформація тексту відповідно до конкретного контексту за допомогою штучного інтелекту.
Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть [тут](#).

Початковий текст

Ми щойно випустили новий продукт. Він доступний вже зараз; перевірте його на нашому сайті.

Додатковий контекст

Блог про технології. Потрібен текст: що залучить користувачів.

Трансформований текст

Видалити

Рисунок 3.9 – Користувач успішно надіслав запит трансформації тексту

Трансформація тексту

Трансформація тексту відповідно до конкретного контексту за допомогою штучного інтелекту.
Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть [тут](#).

Початковий текст

Ми щойно випустили новий продукт. Він доступний вже зараз; перевірте його на нашому сайті.

Додатковий контекст

Блог про технології. Потрібен текст: що залучить користувачів.

Трансформований текст

Чудові новини, любителі технологій! Ви вже бачили наш новітній гаджет, який змінює уявлення про зв'язок? Пориньте у майбутнє з нашим найінноваційнішим продуктом - ідеальним для вашого мобільного способу життя! Навіщо чекати? Вивчайте функції та будьте першими, хто відчуж інновації. Переходьте за посиланням у нашій біографії прямо зараз! Як ви будете використовувати його, щоб залишатися на зв'язку? Напишіть в коментарях нижче!

Видалити

Покращити далі

Рисунок 3.10 – Користувач успішно отримав результат трансформації тексту

Трансформація тексту
Трансформація тексту відповідно до конкретного контексту за допомогою штучного інтелекту.
Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть [тут](#).

Початковий текст

Ми щойно випустили новий продукт. Він доступний вже зараз; перевірте його на нашому сайті.

Трансформований текст

Додатковий контекст

Блог про технології. Потрібен текст, що залучить користувачів.

Видалити Трансформувати текст

Сталась помилка. Будь ласка, спробуйте ще раз ✕

Рисунок 3.11 – Користувач не може отримати результат трансформації тексту через технічну помилку

Трансформація тексту
Трансформація тексту відповідно до конкретного контексту за допомогою штучного інтелекту.
Щоб дізнатися, як ефективно використовувати цю функцію, зокрема корисні поради, натисніть [тут](#).

Запит

Будь ласка, заповніть поле

Трансформований текст

Додатковий контекст

Видалити Трансформувати текст

Рисунок 3.12 – Користувач не може надіслати запит на трансформацію тексту через невалідне значення поля

Таблиця 3.7 – Копіювання трансформованого тексту

Історія користувача	Як Користувач, я хочу швидко скопіювати весь трансформований текст, щоб зручно продовжити працювати з ним.
Критерії прийнятності	<p>Сценарій 1: Користувач успішно копіює трансформований текст (рис. 3.13)</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Користувач ● And: Користувач знаходиться на сторінці трансформації тексту (див. табл. 3.5) ● And: Користувач успішно отримав результат трансформації тексту (див. табл. 3.6, Сценарій 2) ● When: Користувач натискає кнопку “Скопіювати трансформований текст” ● Then: трансформований текст копіюється в буфер обміну ● And: спливаюче повідомлення про успіх “Скопійовано” відображається протягом 5 секунд

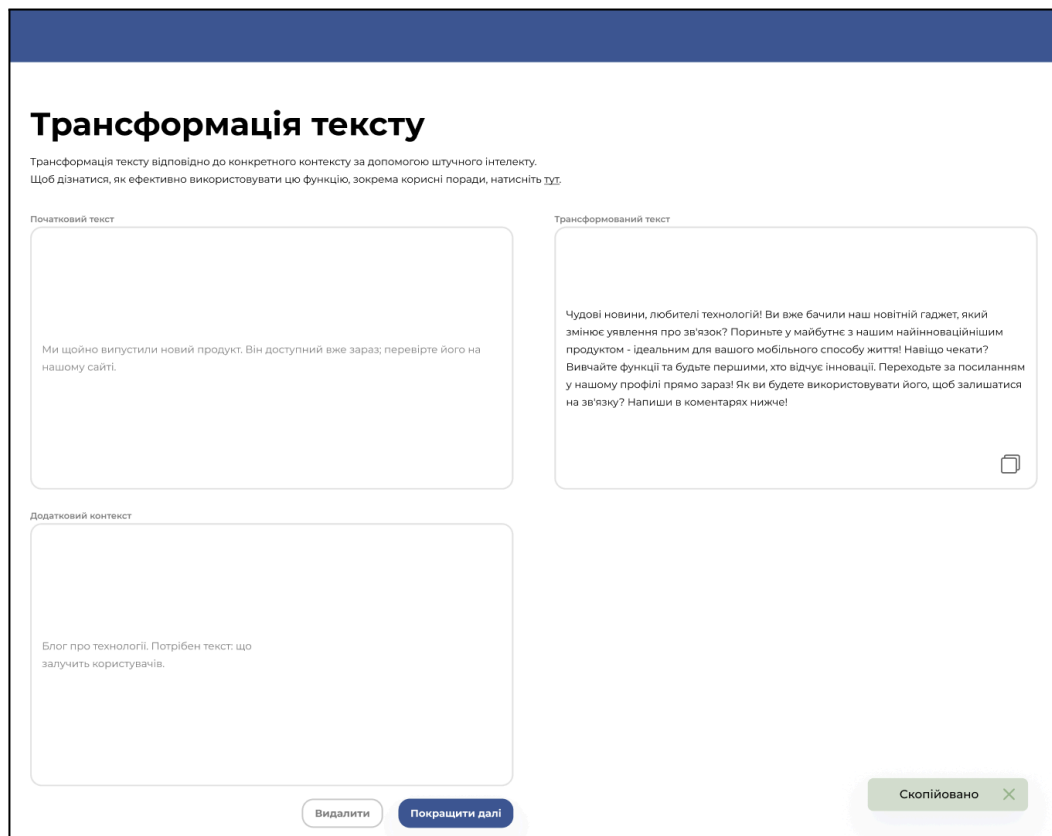


Рисунок 3.13 – Користувач успішно копіює трансформований текст

Таблиця 3.8 – Запит на покращення результату трансформації та перегляд варіантів трансформації

Історія користувача	Як Користувач, я хочу зробити запит на покращення в трансформованому тексті, щоб отримати кращий результат трансформації.
Критерії прийнятності	<p>Передумова:</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Користувач ● And: Користувач знаходиться на сторінці трансформації тексту (див. табл. 3.5) ● And: Користувач успішно отримав результат трансформації тексту (див. табл. 3.6, Сценарій 2) <p>Сценарій 1: Користувач успішно натискає на кнопку “Покращити далі” (рис. 3.14)</p> <ul style="list-style-type: none"> ● When: Користувач натискає кнопку “Покращити далі” ● Then: Відкривається спливаюче вікно з наступними елементами: <ul style="list-style-type: none"> ○ Заголовок: “Покращити далі” ○ Текст: “Введіть запит для подальшого покращення трансформованого тексту” ○ Текстове поле “Запит покращення” ○ Кнопка “Відміна” ○ Кнопка “Застосувати” <p>Сценарій 2: Користувач успішно надсилає запит на подальше покращення (рис. 3.15)</p> <ul style="list-style-type: none"> ● Given: Користувач знаходиться на спливаючому вікні “Покращити далі” ● And: текстове поле “Запит покращення” заповнено валідним значенням: <ul style="list-style-type: none"> ○ поле не порожнє, та ○ поле містить допустиму кількість символів: менше або дорівнює 5000 ● When: Користувач натискає кнопку “Застосувати” ● Then: поле “Запит покращення” стає нередагованим ● And: на кнопці “Застосувати” Користувач бачить лоадер (loader) ● And: надсилається запит до ChatGPT зі змістом поля “Запит покращення” <p>Сценарій 3: Користувач успішно отримує результат подальшого покращення (рис. 3.16)</p> <ul style="list-style-type: none"> ● Given: шага Сценарію 2 ● When: система отримала результат подальшого покращення від ChatGPT ● Then: спливаюче вікно “Покращити далі” закривається ● And: в полі “Трансформований текст” Користувач бачить покращений текст ● And: в полі “Трансформований текст” Користувач може передивитись минулий або наступний результат трансформації через кнопки “назад” і “далі”

Закінчення табл. 3.8

Критерії прийнятності	<p>Сценарій 4: Користувач не може отримати результат подальшого покращення через технічну помилку (рис. 3.17)</p> <ul style="list-style-type: none"> ● Given: шага Сценарію 2 ● When: стається будь-яка технічна помилка (сервер не відповідає протягом однієї хвилини, тощо), ● Then: Користувач бачить спливаюче повідомлення протягом 5 секунд: “Сталась помилка. Будь ласка, спробуйте ще раз.” ● And: поле “Запит покращення” стає редагованим ● And: Користувач бачить кнопку “Застосувати” без лоадера <p>Сценарій 5: Користувач не може надіслати запит на подальше покращення через невалідне значення поля “Запит покращення” (рис. 3.18)</p> <ul style="list-style-type: none"> ● Given: Користувач знаходиться на спливаючому вікні “Покращити далі” ● And: текстове поле “Запит покращення” заповнено невалідним значенням: <ul style="list-style-type: none"> ○ поле порожнє, або ○ поле містить недопустиму кількість символів: більше 5000 ● When: Користувач натискає кнопку “Застосувати” ● Then: відображається відповідне/відповідні повідомлення про помилку/помилки: <ul style="list-style-type: none"> ○ якщо поле порожнє: “Будь ласка, заповніть поле” ○ якщо поле містить недопустиму кількість символів (більше 5000): “Поле має містити менше 5000 символів” ● And: поле “Запит покращення” залишається редагованим ● And: Користувач бачить кнопку “Застосувати” без лоадера ● And: запит до ChatGPT не надсилається <p>Сценарій 6: Користувач відмінює подальше покращення</p> <ul style="list-style-type: none"> ● Given: Користувач знаходиться на спливаючому вікні “Покращити далі” ● When: Користувач натискає кнопку “Відміна” ● Then: спливаюче вікно “Покращити далі” закривається без подальших дій
-----------------------	--

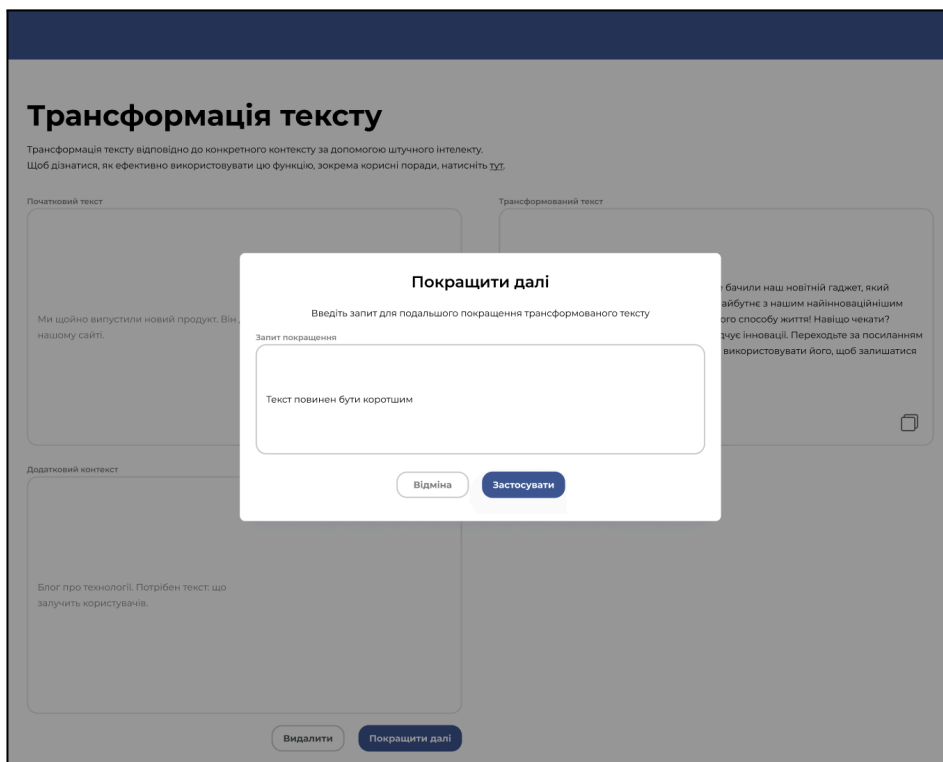


Рисунок 3.14 – Користувач успішно натиснув на кнопку “Покращити далі”

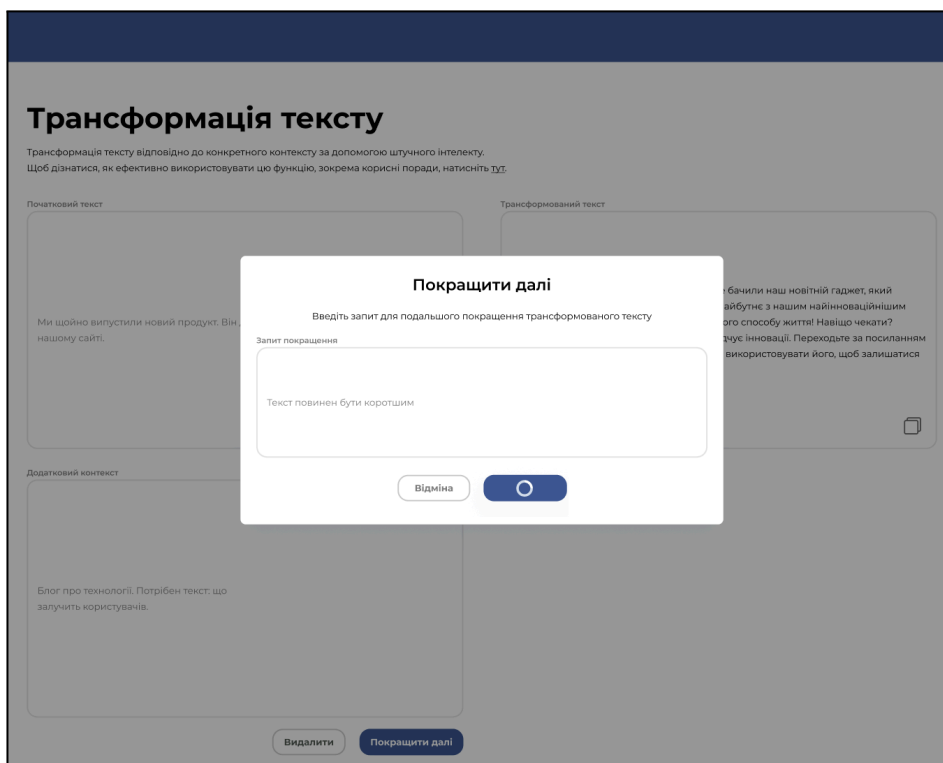


Рисунок 3.15 – Користувач успішно надсилає запит на подальше покращення

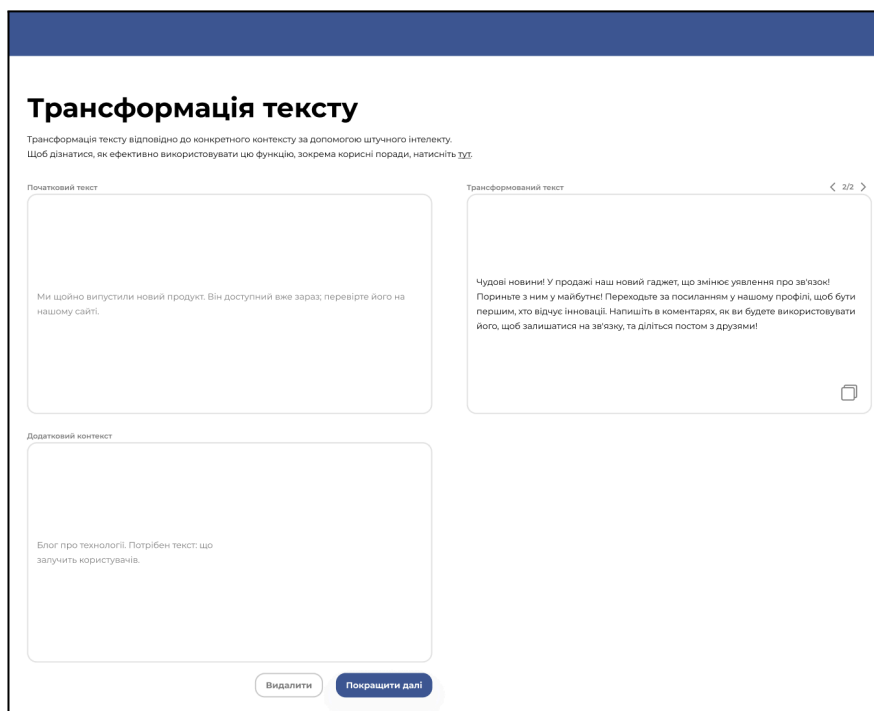


Рисунок 3.16 – Користувач успішно отримав результат подальшого покращення

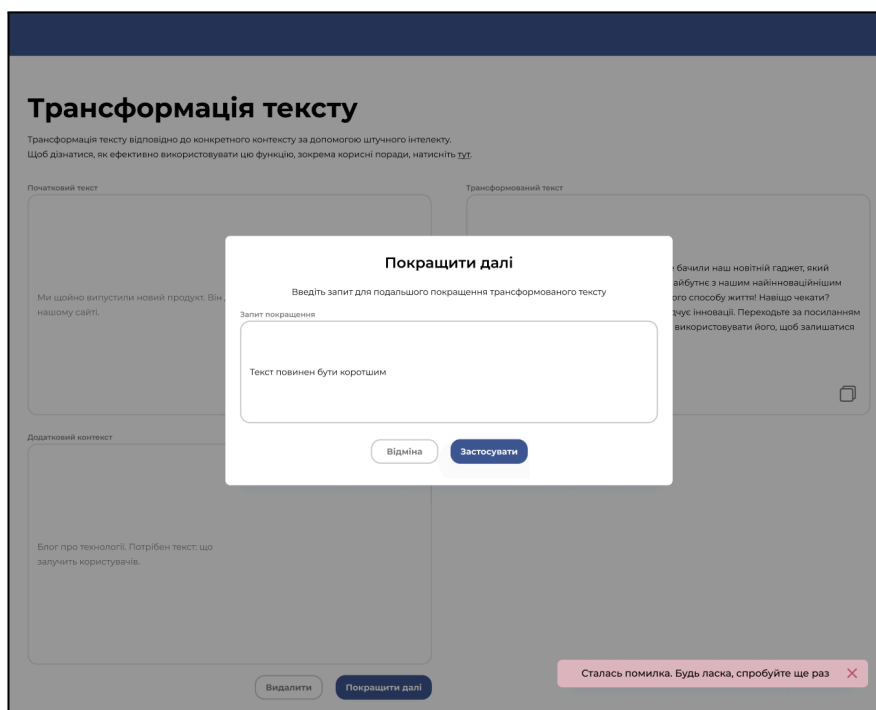


Рисунок 3.17 – Користувач не може отримати результат подальшого покращення через технічну помилку

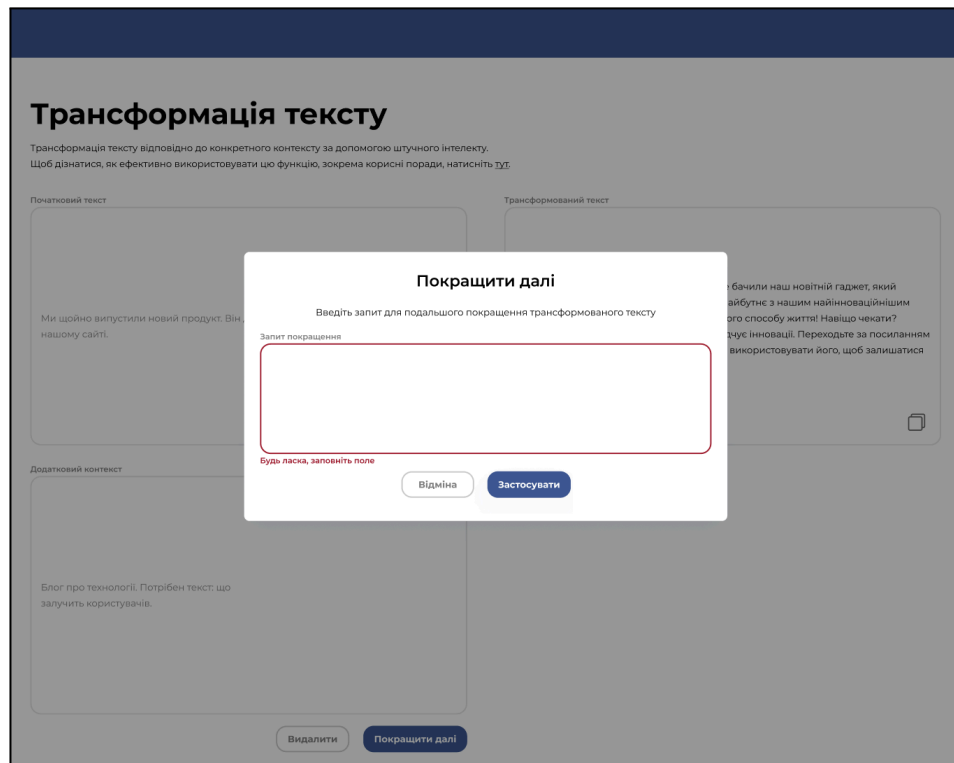


Рисунок 3.18 – Користувач не може надіслати запит на подальше покращення через невалідне значення поля “Запит покращення”

Таблиця 3.9 – Видалення початкового тексту, додаткового контексту і трансформованого тексту

Історія користувача	Як Користувач, я хочу видалити початковий текст, додатковий контекст і трансформований текст, щоб зробити нову текстову трансформацію.
Критерії прийнятності	<p>Передумова:</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Користувач ● And: Користувач знаходиться на сторінці трансформації тексту (див. табл. 3.5) <p>Сценарій 1: Користувач успішно натискає на кнопку “Видалити” (рис. 3.19)</p> <ul style="list-style-type: none"> ● When: Користувач натискає на кнопку “Видалити” ● Then: відкривається спливаюче вікно з наступними елементами: <ul style="list-style-type: none"> ○ Заголовок: “Видалити текст?” ○ Текст: “Початковий текст, додатковий контекст і трансформований текст буде видалено. Цю дію неможливо відмінити.” ○ Кнопки “Відміна”, “Видалити”

Закінчення табл. 3.9

Критерії прийнятності	<p>Сценарій 2: Користувач успішно видаляє початковий текст, додатковий контекст і трансформований текст</p> <ul style="list-style-type: none"> ● Given: Користувач знаходиться на спливаючому вікні “Видалити текст?” ● When: Користувач натискає на кнопку “Видалити” ● Then: поля “Початковий текст”, “Додатковий контекст” і “Трансформований текст” стають порожніми ● And: Користувач може редагувати поле “Початковий текст” ● And: Користувач може редагувати поле “Додатковий контекст” ● And: Користувач бачить кнопку “Трансформувати текст” ● And: Користувач бачить порожнє поле для результату трансформації “Трансформований текст” ● And: Користувач не може редагувати поле “Трансформований текст” <p>Сценарій 3: Користувач успішно відміняє видалення</p> <ul style="list-style-type: none"> ● Given: Користувач знаходиться на спливаючому вікні “Видалити текст?” ● When: Користувач натискає на кнопку “Відміна” ● Then: спливаюче вікно “Видалити текст?” закривається без подальших дій
-----------------------	--

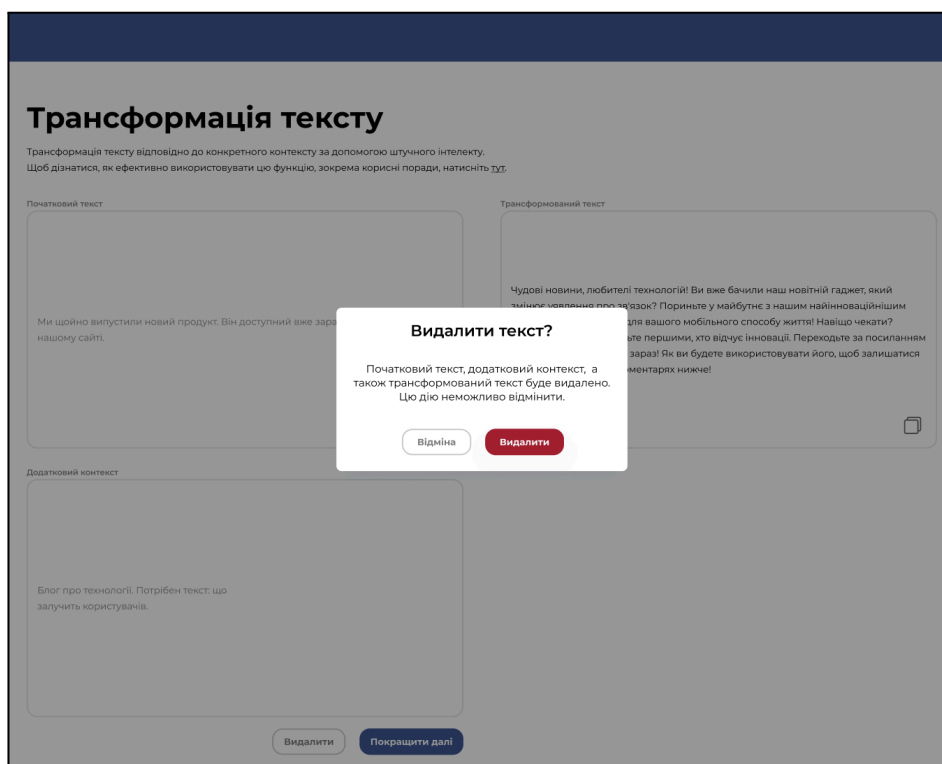


Рисунок 3.19 – Спливаюче вікно “Видалити текст?”

Таблиця 3.10 – Сторінка інструкції з трансформації тексту

Історія користувача	Як Користувач, я хочу переглянути інструкцію з користування функціональністю трансформації тексту та запити змін в трансформованому тексті, щоб дізнатися як ефективно користуватися цими функціональними.
Критерії прийнятності	<p>Сценарій 1: Сторінка з інструкцією успішно відкривається</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Користувач ● And: Користувач знаходиться на сторінці трансформації тексту (див. табл. 3.5) ● When: Користувач натискає на гіперпосилання на сторінку інструкції в тексті під заголовком сторінки ● Then: сторінка інструкції відкривається в тій же вкладці браузера ● And: Користувач може переглянути її контент (див. Сценарій 2) <p>Сценарій 2: Користувач успішно переглядає контент сторінки з інструкцією (рис. 3.20)</p> <ul style="list-style-type: none"> ● Given: користувач авторизований ● And: користувач має роль Користувач ● When: Користувач знаходиться на сторінці інструкції ● Then: Користувач бачить заголовок сторінки: “Інструкція з трансформації тексту” ● And: Користувач бачить текст, що також містить посилання на сторінку трансформації тексту (див. табл. 3.5)



Рисунок 3.20 – Сторінка інструкції з трансформації тексту

3.2 Приклади запитів в чат-бот ChatGPT

Як уже зазначалося раніше, запит є невід'ємною частиною цієї функціональності. Ефективність і релевантність результатів, які генерує штучний інтелект, значною мірою залежать від побудови запиту. Отже, характер і специфіка запиту безпосередньо впливають на якість і доречність відповідей штучного інтелекту.

У цьому підрозділі буде наведено приклади запитів, які можна застосувати в різних сценаріях трансформації тексту. Ці запити було створено відповідно до найкращих практик і на основі досліджень подібних функціональностей, доступних на ринку. Це гарантує, що наведені приклади не лише відповідають галузевим стандартам, але й включають перевірені стратегії для оптимізації їхньої ефективності в реальних додатках.

1. Запит для трансформації тексту резюме: “Перероби наданий текст резюме, щоб підкреслити навички, які необхідні для успішного подання на дану вакансію. Інтегруй ключові слова з доданого опису вакансії. Переконайся, що мова активна і орієнтована на результат. Підкресли вимірювані досягнення. Відформатуй трансформований текст як маркований список. Трансформований текст повинен містити приблизно 500 символів. Наданий текст резюме: %%вміст поля “Початковий текст”%%. Надана вакансія: %%вміст поля “Додатковий контекст”%%.”.
2. Запит для трансформації маркетингового тексту: “Перероби наданий маркетинговий текст так, щоб він апелював саме до його цільової аудиторії. Додай сильний заклик до дії. Враховуй додатковий контекст, якщо він наданий. Наданий маркетинговий текст: %%вміст поля “Початковий текст”%%. Наданий додатковий контекст: %%вміст поля “Додатковий контекст”%%.”.

3. Запит для трансформації тексту для публікації у соціальних мережах: “Перероби текст публікації для соціальної мережі, щоб підвищити залученість користувачів. Використовуй енергійну та заманливу мову. Включи пряме запитання, щоб заохотити до коментарів. Встав переконливий заклик до цільової дії. Враховуй додатковий контекст, якщо він наданий. Трансформований текст повинен містити не більше ніж 150 символів. Наданий текст публікації для соціальної мережі: %%вміст поля “Початковий текст”%%. Наданий додатковий контекст: %%вміст поля “Додатковий контекст”%%.”.
4. Запит для трансформації тексту для пошукової оптимізації (SEO): “Перероби наданий текст сайту для оптимізації для SEO, націливши його на ключові слова, що стосуються наших основних продуктів. Удоскональ текст так, щоб він природно включав ці ключові слова, збагати контент відповідними термінами, а також забезпеч привабливість та інформативність мови. Додай переконливі заклики до дії, які заохочують користувачів до взаємодії та вивчення сайту. Враховуй додатковий контекст, якщо він наданий. Структуруй контент за допомогою правильних заголовків і метаописів, щоб покращити його сканування та індексацію пошуковими системами. Наданий наданий текст сайту для оптимізації для SEO: %%вміст поля “Початковий текст”%%. Наданий додатковий контекст: %%вміст поля “Додатковий контекст”%%.”.
5. Запит для трансформації академічного тексту: “Доопрацюй наданий академічний текст, щоб оптимізувати його для більшої видимості в наукових базах даних і пошукових системах. Включи відповідні ключові слова. Переконайся, що в анотації та висновках ці терміни посідають чільне місце, а також, що вони природно вплетені в текст документа. Вдоскональ назву, щоб вона відображала ключові результати дослідження і була якнайбільш конкретною. Додай

структуровані абстрактні елементи, які чітко окреслюють цілі, методи, результати та висновки, щоб покращити можливість пошуку та читабельність. Враховуй додатковий контекст, якщо він наданий. Наданий академічний текст: %%вміст поля “Початковий текст”%%. Наданий додатковий контекст: %%вміст поля “Додатковий контекст”%%.”.

6. Запит для трансформації тексту посібнику користувача: “Перепиши текст наданої технічної документації, щоб зробити її зрозумілішою та зручнішою для користувача. Розбий складні інструкції на короткі та прості для виконання кроки у форматі нумерованого списку. Заміни будь-який жаргон на зрозумілу мову та визнач всі необхідні технічні терміни у глосарії. Якщо можливо, виділи типові помилки та поради щодо їхнього усунення в окремий розділ. Враховуй додатковий контекст, якщо він наданий. Наданий текст наданої технічної документації: %%вміст поля “Початковий текст”%%. Наданий додатковий контекст: %%вміст поля “Додатковий контекст”%%.”.
7. Запит для трансформації тексту юридичного документу: “Перепиши наданий текст юридичного документу так, щоб зробити його зрозумілішим для всіх зацікавлених сторін, але при цьому зберегти юридичну точність і зміст висловлювань. Спрости мову, вилучивши складний юридичний жаргон і замінивши її більш простою мовою. Юридична цілісність документа обов'язково має бути збережена. Переглянутий текст обов'язково має відповідати чинним правовим стандартам. Враховуй додатковий контекст, якщо він наданий. Наданий текст юридичного документу: %%вміст поля “Початковий текст”%%. Наданий додатковий контекст: %%вміст поля “Додатковий контекст”%%.”.
8. Запит для трансформації тексту новинних статей: “Трансформуй наданий текст новин так, щоб залучити цільову аудиторію. Зробіть

мову відповідною до цільової аудиторії. Наголоси на основній тезі новини. Враховуй додатковий контекст, якщо він наданий. Наданий текст юридичного документу: %%вміст поля “Початковий текст”%%. Наданий додатковий контекст: %%вміст поля “Додатковий контекст”%%.”.

9. Запит для трансформації навчального тексту: “Перепиши наданий навчальний текст так, щоб він підходив для цільової аудиторії учнів. Спрости або ускладни текст відповідно. В будь-якому випадку, розбий текст на прості для розуміння озаглавлені частини у форматі маркованого списку. Враховуй додатковий контекст, якщо він наданий. Наданий навчальний текст: %%вміст поля “Початковий текст”%%. Наданий додатковий контекст: %%вміст поля “Додатковий контекст”%%.”.

10. Запит для трансформації тексту для машинного навчання: “Застосуй методи НЛП-трансформації тексту, такі як стемінг, лематизація та інтеграція синонімів, щоб уточнити пошуковий запит для підвищення ефективності пошуку. Перетвори наданий текст, включивши до нього стемінги, лематизовані форми та синоніми. Переконай, що трансформований текст розширює можливості запиту знаходити релевантні документи. Враховуй додатковий контекст, якщо він наданий. Наданий навчальний текст: %%вміст поля “Початковий текст”%%. Наданий додатковий контекст: %%вміст поля “Додатковий контекст”%%.”.

Висновки до розділу 3

У третьому розділі кожен елемент функціоналу додатку був ретельно деталізований. Були описані історії користувачів (user story), щоб зафіксувати суть того, що кожна роль користувача очікує від додатку, гарантуючи, що наша розробка керується реальними потребами та сценаріями користувачів. Ці історії користувачів були доповнені критеріями прийнятності, сформульованими у форматі Gherkin – поведінково-орієнтованому методі розробки (BDD – Behavior Driven Development), який визначає умови, за яких функція вважається завершеною. Цей формат не тільки підвищує чіткість і точність завдань розробки, але й допомагає в автоматизованому тестуванні. Крім того, де це було можливо, функціональні описи візуально підтримувалися прототипами Figma. Ці прототипи забезпечували відчутне представлення користувацького інтерфейсу та взаємодії, сприяючи чіткішому розумінню кінцевого продукту.

Також було надано низку прикладів запитів до ChatGPT, спеціально розроблених для різних сценаріїв трансформації тексту. Ці запити були ретельно розроблені відповідно до найкращих практик створення запитів, які були детально вивчені та описані в розділі 2. Крім того, при розробці запитів були враховані результати ретельного аналізу ринку, також описаного в розділі 2. Такий підхід гарантує, що підказки будуть ефективними та релевантними, оптимізуючи процеси перетворення тексту в різних галузях і сферах використання.

РОЗДІЛ 4 РОБОТА ФУНКЦІОНАЛЬНОСТІ

4.1 Опис роботи функціональності

У цьому підрозділі буде описано, як працюватиме функціональність, розроблена за вимогами, описаними у підрозділі 3.1.

Отже, Адміністратор в своєму додатку встановлює або оновлює конфігурацію запиту до чат-боту ChatGPT на перетворення тексту та зберігає ці налаштування в системі. Системою передбачена валідація, отже, Адміністратор не може задати занадто великий запит або порожнє поле як запит. Успішно збережений запит потім буде використаний для трансформації тексту користувача.

Користувач в своєму додатку вводить текст, який потрібно трансформувати, і, опціонально, додатковий контекст до нього. Тут також передбачена валідація, отже, Користувач не може зробити запит на трансформацію занадто великого тексту, порожнього поля, або надати занадто великий додатковий контекст. Коли користувач успішно робить запит на трансформацію тексту, попередньо налаштований запит отримується з конфігурацій адміністратора. Запит будується на основі налаштувань, заданих адміністратором, і даних, введених користувачем. Побудований запит надсилається до ChatGPT, де ChatGPT обробляє запит і трансформує текст. Трансформований текст надсилається до додатку Користувача. Користувач може його переглянути та, за бажанням, копіювати його, видалити все, щоб почати процес заново, або продовжити трансформувати його. Отже, опціонально, Користувач може зробити запит на покращення трансформованого тексту. В такому разі, Користувач в своєму додатку вводить запит для подальшої трансформації тексту. Цей запит надсилається до ChatGPT, де ChatGPT обробляє його запит і відповідно покращує попередньо трансформований текст. Покращений трансформований текст надсилається до додатку Користувача, Користувач може переглянути його разом з попереднім результатом

трансформації. Цей цикл може тривати, доки користувач не буде задоволений результатом. Для подальших покращених текстів користувач має такі ж самі функціональні можливості: Користувач може їх так само скопіювати, видалити все, або продовжити трансформувати.

Для більш наочного опису роботи функціональності було створено UML (Unified Modeling Language) діаграму послідовності [10] (рис. 4.1), що зображує процес в успішних сценаріях від початку до кінця.

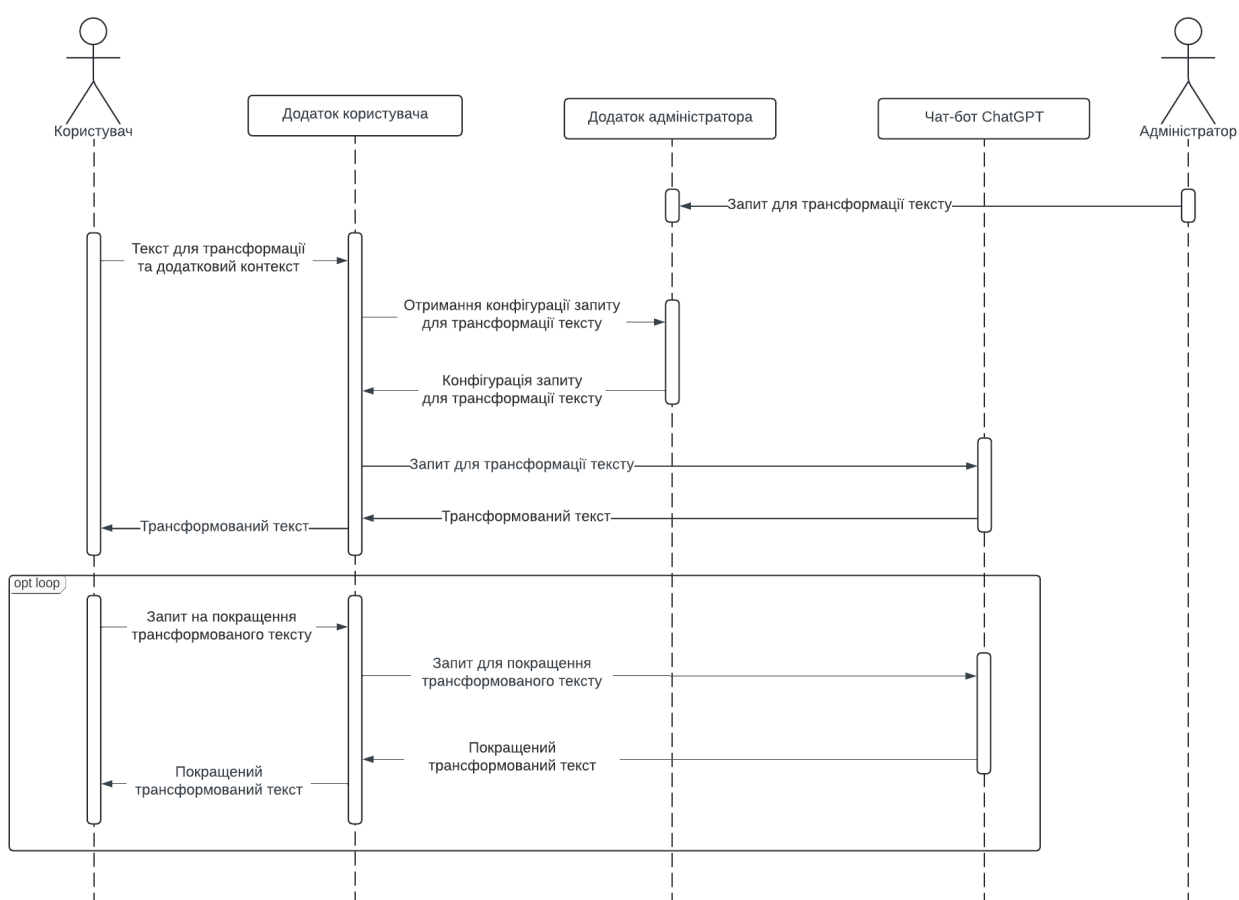


Рисунок 4.1 – UML діаграма роботи програми

4.2 Результати запитів з трансформації тексту в чат-бот ChatGPT

У цьому підрозділі буде проілюстровано практичну цінність, яку мають концепції, виклані раніше в цій роботі. Буде представлено фактичне застосування розробленої концепції функціональності та запитів трансформації тексту до ChatGPT, розроблених відповідно до найкращих практик інженерії запитів та дослідження ринку. Таким чином, представлено детальний аналіз "до і після" трансформації текстів.

1. Трансформація тексту резюме.

- a. До: “Працював у сфері продажів понад 10 років. Займався укладанням угод та взаємодією з клієнтами. Знаюм з декількома CRM-платформами.”.
- b. Після: “Керував відділом продажів, розвиваючи відносини з клієнтами та оптимізуючи стратегії продажів, збільшивши річний обсяг продажів на 20%. Очолював інтеграцію нового програмного забезпечення CRM, покращуючи процеси взаємодії з клієнтами та управління даними. Продемонстрував експертизу в управлінні проектами, керуючи міжфункціональними командами для перевиконання планів по доходах.”.

2. Трансформація маркетингового тексту.

- a. До: “Купуйте наші товари! Ми пропонуємо вигідні пропозиції на різноманітні товари. Відвідайте наш сайт, щоб дізнатися більше.”.
- b. Після: “Приєднуйтеся до спільноти прогресивних лідерів, які відстоюють принципи сталого розвитку в бізнесі. Відкрийте для себе наш ексклюзивний асортимент екологічно чистих продуктів, які не лише заощаджують ваші гроші, але й захищають нашу планету. Скористайтеся нашою спеціальною

вступною пропозицією та почніть змінювати світ на краще своїми покупками.”.

3. Трансформація тексту для публікації у соціальних мережах.

- a. До: “Ми щойно випустили новий продукт. Він доступний вже зараз; перевірте його на нашому сайті.”.
- b. Після: “Чудові новини, любителі технологій! Ви вже бачили наш новітній гаджет, який змінює уявлення про зв'язок? Пориньте у майбутнє з нашим найінноваційнішим продуктом - ідеальним для вашого мобільного способу життя! Навіщо чекати? Вивчайте функції та будьте першими, хто відчує інновації. Переходьте за посиланням у нашому профілі прямо зараз! Як ви будете використовувати його, щоб залишатися на зв'язку? Напиши в коментарях нижче!”.

4. Трансформація тексту для пошукової оптимізації (SEO).

- a. До: “Наша компанія продає багато різних продуктів. Перевірте наш веб-сайт, щоб побачити, що ми пропонуємо.”.
- b. Після: “Відкрийте для себе наш ексклюзивний асортимент екологічно чистих товарів для дому та аксесуарів для сталого життя, розроблених для покращення вашого екологічного способу життя. Пориньте в нашу колекцію інноваційних продуктів, кожен з яких створений з турботою про навколишнє середовище. Дізнайтеся, як ви можете зробити свій внесок у сталий розвиток з кожною покупкою. Не пропустіть – розпочніть свою подорож до сталого майбутнього вже сьогодні, вивчаючи наш сайт далі. Натисніть тут, щоб дізнатися більше про наші екологічні пропозиції та приєднатися до нашої спільноти, яка прагне змінити світ на краще.”.

5. Трансформація академічного тексту.

- a. До: “Це дослідження розглядає наслідки зміни клімату. Ми виявили значний вплив на різні екосистеми.”.
- b. Після: “Кількісна оцінка впливу глобального потепління на морські та наземні екосистеми: Це дослідження вивчає глибокі наслідки зміни клімату для різноманітних екологічних систем. Наші висновки демонструють значні зміни як у морському, так і в наземному середовищі, що свідчить про ширші наслідки глобального потепління. Систематично аналізуючи екологічні дані, це дослідження робить важливий внесок у сферу екологічних досліджень, пропонуючи детальну оцінку реакції екосистем на кліматичні зміни, що відбуваються.”.

6. Трансформація тексту посібнику користувача.

- a. До: “Для початку роботи пристрій має бути налаштований з параметрами, зазначеними в розділі 4.3. Неправильні налаштування можуть призвести до неоптимальної роботи або виходу пристрою з ладу.”.
- b. Після: “Щоб почати користуватися пристроєм, виконайте прості кроки, описані в розділі 4.3. Почніть з правильного налаштування наступних параметрів, які мають вирішальне значення для оптимальної роботи пристрою. Візуальний посібник і детальний контрольний список допоможуть вам у цьому процесі. Якщо ви ввели параметри неправильно, зверніться до розділу “Поширені помилки”, щоб швидко виявити та виправити помилки. Цей розділ містить прості поради щодо усунення несправностей і візуальні підказки, які допоможуть вам ефективно вирішувати проблеми, забезпечуючи безперебійну роботу вашого пристрою.”.

7. Трансформація тексту юридичного документу.

- a. До: “Сторона першої частини не несе відповідальності за будь-які дії, які можуть бути витлумачені як недбалі без додавання умислу або грубої необережності.”.
- b. Після: “Перша сторона не несе відповідальності за будь-які дії, які вважаються недбалими, якщо не буде доведено, що мали місце навмисні порушення або значна недбалість.”.

8. Трансформація тексту новинних статей.

- a. До: “Місцева компанія розширила свою діяльність. Тепер вона працює в кількох нових місцях і збільшила кількість своїх працівників.”.
- b. Після: “Зустрічайте нову компанію! Цей першопроходець у сфері технологій зростає не лише в розмірах, а й в інноваціях. З кількома новими екологічно чистими офісами та великим стрибком у передових технологіях, вони створюють майбутнє технологічних робочих місць. Маєте хвилинку? Пориньте у їхній світ сталих технологій за допомогою цих крутих знімків та історій! Хочете приєднатися до них? Натисніть тут, щоб дізнатись про найгарячіші вакансії у сфері технологій!”.

9. Трансформація навчального тексту.

- a. До: “Процес фотосинтезу полягає в перетворенні рослинами світлової енергії на хімічну. Зазвичай він включає кілька складних біохімічних реакцій.”.
- b. Після: “Подумайте про фотосинтез як про рецепт, за яким рослини виготовляють свою їжу. Так само, як вам потрібне сонячне світло, щоб бачити вдень, рослинам потрібне сонячне світло, щоб почати свій рецепт виготовлення їжі. Ось як вони це роблять: спочатку вони вловлюють сонячне світло і змішують його з невеликою кількістю води та повітря. Потім вони створюють цукор, який дає їм енергію для росту і

процвітання. Хочете побачити, як це відбувається? Подивіться цю круту анімацію та пройдіть тест, щоб перевірити, що ви дізналися!”.

10. Трансформація тексту для машинного навчання.

- a. До: Пошуковий запит в інформаційно-пошуковій системі: "Швидкісний біг підвищує витривалість".
- b. Після: Розширений пошуковий запит: "Швидко бігати підвищити витривалість". Включені синоніми та пов'язані фрази: "швидкісні тренування покращують фізичну витривалість", "спринтерський біг підвищує витривалість".

Висновки до розділу 4

У цьому розділі описана робота розробленої концепції функціональності відповідно до вимог, описаних в підрозділі 3.1. Робота функціональності також зображена в діаграмі послідовностей мовою графічного описання UML.

Після того, як архітектура системи була зображена, результати роботи програми були представлені за допомогою реальних трансформацій тексту. Використовуючи концепцію функціональності, розроблену раніше, та запити, створені відповідно до найкращих практик та дослідження ринку, було виконано декілька трансформацій тексту. Щоб показати, як система здатна гнучко реагувати на довільні вимоги до трансформації та контекст, було трансформовано тексти з широкого спектру областей.

Проаналізувавши трансформовані тексти, видно, що система добре здатна покращувати та адаптувати текст до точних контекстних потреб. Це доводить практичну життєздатність розробленої концепції функціональності та запитів-прикладів. Вони є потужним інструментом, який можна безперешкодно застосовувати в різних сферах для ефективного перетворення текстів.

РОЗДІЛ 5. ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ФУНКЦІОНАЛУ ТРАНСФОРМАЦІЇ ТЕКСТУ З УРАХУВАННЯМ КОНТЕКСТУ З ВИКОРИСТАННЯМ ЧАТ-БОТУ CHATGPT

5.1. Формування завдання проєктування

Цей розділ містить аналіз функціонально-вартісних аспектів програмного продукту, розробленого у цьому дослідженні, з огляду на використані технології та в контексті теорії оптимізації виробництва. Це аналіз дозволить оцінити рішення, прийняті щодо розробки, та їх вплив на подальше використання результатів дослідження в практичних сценаріях.

Вибір оптимальної стратегії розробки будь-якої системи, особливо тієї, що має важливе значення, є критичним етапом. Усі етапи розробки, зокрема початкові, є вразливими перед ризиками, які визначають можливості дослідника або команди у досягненні мети проєкту. Неправильний збір та обробка даних, вибір середовища розробки, інструментів розробки, засобів моніторингу та належна презентація виконаної роботи можуть підірвати доцільність програмного продукту, його надійність і, що так само важливо, економічне обґрунтування.

Цей проєкт спрямований на розробку застосунку для мультипараметричного аналізу перфузійних даних, отриманих з комп'ютерної томографії голови. Проєкт має науково-дослідницький характер і не передбачає комерційної реалізації. Важливо зробити висновки з проведеного аналізу результатів, які можуть бути корисні для неврологів, нейрохірургів та інших фахівців, які працюють у галузі дослідження мозку та його порушень.

Необхідно також врахувати економічний аспект проєктування, оскільки більшість проєктів мають обмежені ресурси, а гроші є одним з головних факторів впливу на них. Врахування майбутньої перспективи важливо, оскільки науково-дослідницькі проєкти, як правило, отримують фінансування лише за певними результатами. Це підкреслює значення вартісного аналізу проведеної

роботи і обґрунтованої розробки методів на основі обраних технологій та отриманих результатів, які можуть бути використані під час функціонального аналізу.

Важливо відзначити, що аналіз ринку та конкурентів у цьому випадку не є можливим у звичайному розумінні. Це пояснюється тим, що провідні розробники у цій галузі не розкривають фінансових аспектів своїх рішень, а їх програмне забезпечення не є загальнодоступним через різноманітні причини. Такі рішення не мають на меті комерціалізації та впровадження у системи охорони здоров'я, а скоріше спрямовані на поширення знань та розуміння процесів у мозку та впливу зовнішніх та внутрішніх факторів на його стан.

5.2. Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка програмного продукту, який дозволяє візуалізувати та аналізувати одночасно декілька показників перфузії. Беручи за основу дану функцію, було виділено наступні головні функції виробу:

F_1 – вибір методу візуалізації.

F_2 – вибір бібліотеки створення графічного інтерфейсу.

F_3 – вибір бібліотеки для роботи з медичними даними.

F_4 – вибір способу роботи з графікою.

Для кожної з цих функцій було розглянуто різні варіанти реалізації:

Функція F_1 :

А) Налаштування конфігурації запиту до чат-боту ChatGPT для вибору методу трансформації тексту.

Б) Оновлення конфігурації запиту до чат-боту ChatGPT для вибору методу трансформації тексту.

Функція F_2 :

А) Введення тексту та додаткового контексту користувачем в додатку для подальшої трансформації.

Б) Відображення налаштованої конфігурації запиту адміністратора для побудови запиту користувача.

Функція F_3 :

А) Побудова запиту на основі конфігурацій адміністратора та даних, введених користувачем.

Б) Відправлення побудованого запиту до чат-боту ChatGPT для трансформації тексту.

Функція F_4 :

А) Отримання та відображення трансформованого тексту в додатку для користувача.

Б) Опція для користувача надсилати запити на покращення трансформованого тексту.

Далі, на основі визначених функцій та їх варіантів реалізації, була створена морфологічна карта, яка підкреслює структуру програмного продукту (рис. 5.1).

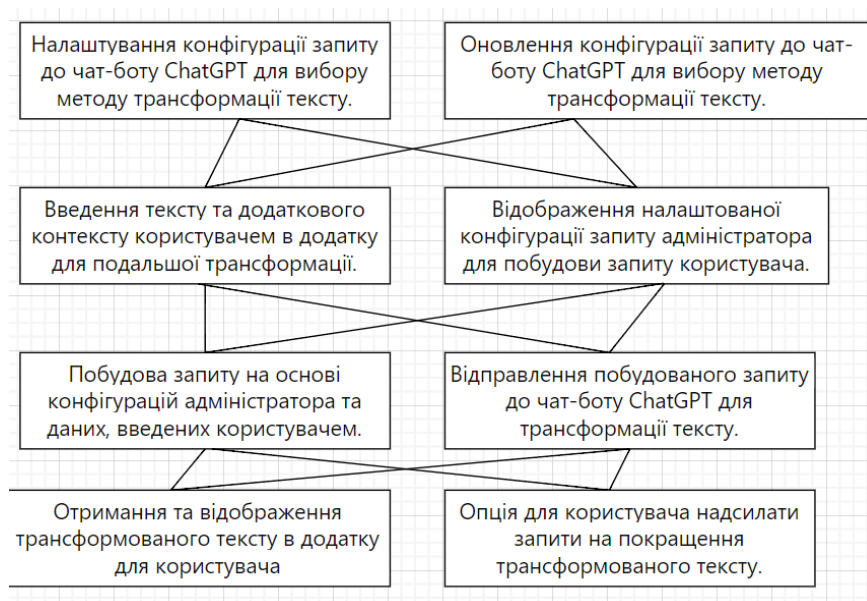


Рисунок 5.1 – Морфологічна карта

Після цього була створена позитивно-негативна матриця, що відображає переваги та недоліки кожного варіанту реалізації (табл. 5.1). На основі цієї матриці вирішено відібрати найбільш оптимальні варіанти для подальшої розробки програми.

Після аналізу позитивно-негативної матриці, обрано наступні варіанти для подальшого розгляду:

Таблиця 5.1 – Позитивно-негативна матриця

Функції	Варіанти	Переваги	Недоліки
F_1	<i>A</i>	Простий у використанні, дозволяє швидко вибрати потрібний метод.	Може бути неефективним для складних методів трансформації, обмежений варіантами.
	<i>B</i>	Забезпечує можливість оновлення та адаптації до нових методів.	Може вимагати більше часу та ресурсів для впровадження, складніше у використанні.
F_2	<i>A</i>	Простий та інтуїтивно зрозумілий для користувача.	Може бути обмеженим у варіантах введення тексту та контексту.
	<i>B</i>	Забезпечує користувачам можливість вибору з налаштованих опцій.	Обмежує користувачів лише вибраними варіантами адміністратором.
F_3	<i>A</i>	Забезпечує гнучкість у виборі методів трансформації.	Може бути складним у використанні та вимагати додаткових ресурсів для обробки даних
	<i>B</i>	Забезпечує автоматизований процес трансформації тексту.	Може виникати затримка у відповіді чат-боту, можливі проблеми з підключенням.
F_4	<i>A</i>	Забезпечує швидкий та простий доступ до результатів трансформації.	Може бути неефективним для великих обсягів тексту, обмежений функціональністю відображення.
	<i>B</i>	Забезпечує можливість користувачам вносити власні корективи до трансформованого тексту	Може створювати навантаження на систему при великій кількості запитів, можливі проблеми з обробкою.

Після аналізу отриманої позитивно-негативної матриці можемо провести фільтрування та позбутися зайвих варіантів, таким чином обравши необхідні для більш продуктивного виконання задач, які поставлені для даного проєкту.

Функція F_1 :

Варіант Б було вибрано через його можливість забезпечувати оновлення конфігурації запиту до чат-боту ChatGPT. Ця функція надає можливість адаптувати та покращувати методи трансформації тексту, що дозволяє більш точно налаштовувати процес обробки даних відповідно до вимог користувачів.

Функція F_2 :

У питанні взаємодії користувача з додатком було обрано варіант А, оскільки введення тексту та додаткового контексту користувачем є простим та інтуїтивно зрозумілим. Це дозволяє користувачам швидко та зручно взаємодіяти з системою, що є критичним для забезпечення ефективності роботи.

Функція F_3 :

Для побудови запиту на основі конфігурацій адміністратора та даних, введених користувачем, обрано варіант А. Цей варіант забезпечує гнучкість у виборі методів трансформації та дозволяє створювати запити, які точно відповідають потребам користувачів.

Функція F_4 :

Опція для користувача надсилати запити на покращення трансформованого тексту (варіант Б) вибрана через її здатність забезпечувати можливість постійного вдосконалення результатів. Ця функція дозволяє користувачам отримувати більш точні та якісні результати шляхом багаторазових покращень трансформованого тексту.

Отже, будемо розглядати такі варіанти реалізації ПП:

- $F_1 \text{ б} - F_2 \text{ а} - F_3 \text{ а} - F_4 \text{ б}$
- $F_1 \text{ б} - F_2 \text{ б} - F_3 \text{ а} - F_4 \text{ б}$

5.3. Обґрунтування системи параметрів програмного продукту

Цей підрозділ передбачає встановлення характеристик для попередньо аналізованих функцій програмного продукту. Таким чином, для розрахунку коефіцієнта технічного рівня, ми можемо розпочати з опису нашого проекту за допомогою наступних параметрів:

X_1 – об'єм оперативної пам'яті для проведення обчислень

X_2 – об'єм написаного програмного коду

X_3 – об'єм задіяних ресурсів центрального процесору

X_4 – об'єм затраченого часу на проведення обчислень

Відповідно до вимог замовника та умов, що характеризують роботу програмного продукту, було обрано найгірші, середні та найкращі значення параметрів, як показано в табл 5.2.

Таблиця 5.2 – Основні параметри програмного продукту

Параметр	Умовне позначення	Одиниця виміру	Значення		
			Найгірше	Середнє	Найкраще
Об'єм оперативної пам'яті для проведення обчислень	X_1	Мб	8192	4096	2048
Об'єм написаного програмного коду	X_2	Кількість рядків	1500	1000	500
Об'єм задіяних ресурсів центрального процесору	X_3	Відсотки	80	50	30
Об'єм затраченого часу на проведення обчислень	X_4	Секунди	300	180	60

5.4. Аналіз експертного оцінювання параметрів

Четвертий етап функціонально-вартісного аналізу програмного продукту включає визначення вагомості кожного параметру в загальній кількості розглянутих під час оцінювання параметрів за допомогою методу попарного порівняння.

Спочатку було знайдено коефіцієнти вагомості, визначено ступінь важливості параметрів шляхом присвоєння їм різних рангів експертами-фахівцями в галузі. Результати продемонстровано в табл. 5.3.

Таблиця 5.3 – Результати обчислень

Позначення параметра	Одиниці виміру	Ранг параметра за оцінкою експерта	Сума рангів Ri	Відхилення Δi	Δi^2
		1	2	3	4
X_1 – Об'єм оперативної пам'яті	ГБ	4	5	4	5
X_2 – Об'єм написаного програмного коду	Кількість рядків	2	2	2	3
X_3 – Об'єм задіяних ресурсів центрального процесора	Відсотки	5	4	5	4
X_4 – Об'єм затраченого часу на трансформацію	Секунди	1	1	1	1
Разом		12	12	12	13

Після цього було обчислено параметри:

1. Загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} = \frac{Nn(n+1)}{2} = \frac{7 \cdot 4 \cdot (4+1)}{2} = 70.$$

2. Середня сума рангів:

$$T = \frac{1}{n} R_i = 22.25.$$

3. Відхилення суми рангів параметрів від середньої суми рангів:

$$\Delta_i = R_i - T.$$

4. Загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 433.$$

5. Коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 433}{7^2 \cdot (4^3 - 4)} \approx 0.95.$$

Для оцінки узгодженості між поглядами експертів було розглянуто нерівність (сформовану на базі значення нормативного коефіцієнту для параметрів програмних продуктів $W_{\text{норм}} = 0.67$) наступного виду $W > W_{\text{норм}}$, що свідчить про достатньо високий рівень узгодженості. Оскільки результат виявився позитивним, то наступним кроком було виконано попарне порівняння параметрів, що продемонстровано в табл. 5.4.

Таблиця 5.4 – Попарне порівняння параметрів

Параметри	Експерти	Кінцева оцінка	Числове значення
	1	2	3
X 1 i X 2	>	>	>
X 1 i X 3	<	>	<
X 1 i X 4	>	>	>
X 2 i X 3	<	<	<
X 2 i X 4	<	<	<
X 3 i X 4	>	>	>

Далі за результатами числових значень коефіцієнтів переваг на основі підсумкових оцінок, було складено квадратну матрицю $A = \|x_{ij}\|$ та розраховано пріоритетність K_{bi} для кожного із параметрів доки різниця між ітераціями розрахунку даних вагомості не стала меншими за 5%.

Розрахунки для першої ітерації відбувались за формулами:

$$K_{bi} = \frac{b_i}{\sum_{i=1}^n b_i}$$

$$b_i = \sum_{i=1}^N a_{ij}$$

Для поточних ітерацій використовувались:

$$K_{bi} = \frac{b'_i}{\sum_{i=1}^n b'_i}$$

$$b'_i = \sum_{i=1}^N a_{ij} b'_j$$

Результати наведено в табл. 5.5:

Таблиця 5.5 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j	Ітерація №1	Ітерація №2
	X 1	X 2	X 3
X 1	1	1.5	1.5
X 2	1.5	1	0.5
X 3	1.5	0.5	1
X 4	1.5	0.5	1.5
Всього:			

Оскільки різниця між коефіцієнтами вагомості параметрів вже на другій ітерації почала дорівнювати 2% (0.02), що є меншим від 5% (0.05), було призупинено подальші розрахунки і визначено дану таблицю.

5.5 Аналіз рівня якості варіантів реалізації функцій

На основі табл. 5.2 було оцінено потенційні абсолютні значення параметрів для нашого проєкту. Кінцевим результатом оцінки є табл. 5.6, де проведено розрахунки показників технічного рівня, що враховує варіанти реалізації функцій та параметри на основі раніше обчисленого коефіцієнта.

Таблиця 5.6 – Розрахунок показників рівня якості

Основні функції	Варіант реалізації функції	Параметри задіяні у реалізації	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F 1	Б	X 1	8192	8	0.35	2.80
F 2	А	X 2	1500	7	0.25	1.75
	Б	X 2	2000	5	0.25	1.25
F 3	А	X 3	40	6	0.20	1.20
F 4	А	X 4	180	9	0.20	1.80

Для обох варіантів реалізації системи (F_2 А та F_2 Б) було просумовано показники технічного рівня, отримавши наступні узагальнені коефіцієнти:

$$K_{F2A} = 2.80 + 1.75 + 1.20 + 1.80 = 7.55$$

$$K_{F2B} = 2.80 + 1.25 + 1.20 + 1.80 = 7.05$$

Таким чином, при порівнянні обох варіантів реалізації програмного продукту, функціональний надав можливість визначити найкращий варіант, яким виступив K_{F2A} , оскільки його значення більше на 0.50.

5.6. Економічний аналіз варіантів розробки ПП

Для визначення загальної трудомісткості системи даного проєкту було використано формулу:

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТМ}$$

Після чого визначено початкові дані системи завдання (розробка програмного продукту) з метою отримання даних про поправочні коефіцієнти для розрахунку загальної трудомісткості. Цей перелік включає в себе:

1. Ступінь новизни групи задач – «А» (задачі, які передбачають використання принципово нових методів розробки, проведення науково-дослідних робіт).
2. Складність алгоритму – «1» (алгоритми оптимізації та моделювання систем і об'єктів).
3. Складність організації контролю вхідної та вихідної інформації – «5.1 + 5.4» (вихідні дані та документи різноманітного розміру і структури).
4. Вид використовуючої інформації – «Банк даних».
5. Мова програмування – «Python3».
6. Використання стандартних модулів програмування.

З цього переліку було сформовано вже поправочні коефіцієнти за таблицями умов із наданих методичних вказівок:

- $T_P - 90$;
- $K_{\Pi} - 2.84$;
- $K_{СК} - 1.07$;
- $K_M - 1$;
- $K_{СТ} - 0.6$;

- $K_{\text{СТМ}} - 1$.

Таким чином в наявності опинились всі елементи необхідні для складання початкового рівняння:

$$T_0 = 90 \cdot 2.84 \cdot 1.07 \cdot 1 \cdot 0.7 \cdot 1 \approx 191.9 \text{ (людино днів)}$$

Розрахувавши загальну трудомісткість програмного продукту, її також було розраховано для двох варіантів реалізації:

$$T_1 = 191.9 \cdot 4 \cdot 8 \approx 6140.8 \text{ (людино днів)}$$

$$T_2 = 191.9 \cdot 4 \cdot 8 \approx 6140.8 \text{ (людино днів)}$$

Результат розрахунків виявився доволі логічним, оскільки хоч підхід та функціонал в обох бібліотеках до створенню графічного інтерфейсу може відрізнятись, сама його структура з елементами повинна лишатись однаковою.

Оскільки у розробці бере участь один науковець у сфері даних з окладом 40000 грн, то середня заробітна плата складає:

$$C_{\text{ч}} = \frac{M}{T_m \cdot t} = \frac{4000}{3 \cdot 21 \cdot 8} = 79.37 \text{ (грн)}$$

Також було обчислено зарплату для обох варіантів реалізації:

$$C_{\text{зп}} = C_{\text{ч}} \cdot T_i \cdot K_{\text{д}} = 79.37 \cdot 6140.8 \cdot 1.1 \approx 534793.38$$

Відраховано соціальний внесок від заробітної плати:

$$C_{\text{від зп}} = C_{\text{зп}} \cdot 0.22 = 534793.38 \cdot 0.22 \approx 117654.54 \text{ (грн)}$$

Для знаходження вартості однієї машино-години, спочатку було розраховано перелік інших змінних.

Машинний час, враховуючи додаткову заробітну плату і відрахування соціального внеску:

$$\begin{aligned} C_{\Gamma} &= 12 \cdot M \cdot K_3 = 12 \cdot 40\,000 \cdot 0.2 = 96\,000 \text{ (грн)} \\ C_{\text{зп}} &= C_{\Gamma} \cdot (1 + K_3) = 96\,000 \cdot 1.2 = 115\,200 \text{ (грн)} \\ C_{\text{від}} &= C_{\text{зп}} \cdot 0.22 = 115\,200 \cdot 0.22 = 25\,344 \text{ (грн)} \end{aligned}$$

Амортизаційні відрахування при вартості обладнання в 120 000 грн та амортизації в 25%:

$$C_A = K_{\text{ТМ}} \cdot K_A \cdot \Pi_{\text{ПР}} = 1.15 \cdot 0.25 \cdot 120\,000 = 34\,500 \text{ (грн)}$$

Вартість ремонту та профілактики:

$$C_P = K_{\text{ТМ}} \cdot K_P \cdot \Pi_{\text{ПР}} = 1.15 \cdot 0.05 \cdot 120\,000 = 6\,900 \text{ (грн)}$$

Ефективний годинний фонд часу ПК за рік:

$$\begin{aligned} T_{\text{ЕФ}} &= (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = \\ &= (365 - 104 - 12 - 16) \cdot 8 \cdot 0.6 = 1382.4 \text{ (год.)} \end{aligned}$$

Кількість коштів витрачених на оплату електроенергії (5.6.9):

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_C \cdot K_3 \cdot \Pi_{\text{ЕН}} \cdot K_B = 1382.4 \cdot 0.2 \cdot 0.2 \cdot 5.23 = 289.03 \text{ (грн.)}$$

Накладні витрати:

$$C_H = C_{\text{ПР}} \cdot 0.67 = 120\,000 \cdot 0.67 = 80\,400 \text{ (грн.)}$$

Вартість експлуатації:

$$\begin{aligned} C_{\text{ЕКС}} &= C_{\text{ЗП}} + C_{\text{ВІД}} + C_A + C_P + C_{\text{ЕЛ}} + C_H = \\ &= 115200 + 25344 + 34500 + 6900 + 289.03 + 80400 \approx \\ &\approx 262633.03 \text{ (грн.)} \end{aligned}$$

Вартість однієї машино-години:

$$C_{\text{М-Г}} = \frac{C_{\text{ЕКС}}}{T_{\text{ЕФ}}} = \frac{262633.03}{1382.4} = 190.02 \text{ (грн.)}$$

Для обох варіантів реалізації:

$$C_M = C_{\text{М-Г}} \cdot T = 190.02 \cdot 6140.8 \approx 1166735.42 \text{ (грн.)}$$

Накладні витрати:

$$C_H = C_{\text{ЗП}} \cdot 0.67 = 115200 \cdot 0.67 = 77184 \text{ (грн.)}$$

Загальна вартість розробки програмного продукту за обома варіантами становила однакове значення, оскільки обидві варіанти не потребують особливих витрат на якісь підписки чи додаткові типи послуг:

$$\begin{aligned} C_{\text{ПП}} &= C_{\text{ЗП}} + C_{\text{ВІД}} + C_M + C_H = \\ &= 115200 + 25344 + 1166735.42 + 77184 \approx 1384\,463.42 \text{ (грн.)} \end{aligned}$$

5.7. Вибір кращого варіанта техніко-економічного рівня

Останнім етапом було обрано найкращий варіант реалізації. Для цього спочатку розраховано коефіцієнт техніко-економічного рівня для двох варіантів реалізації відповідно:

$$K_{\text{ТЕР}j} = \frac{K_{kj}}{C_{\text{пп}j}} = \frac{6.76}{1384\ 463.42} = 4.87 \cdot 10^{-6}$$

$$K_{\text{ТЕР}j} = \frac{K_{kj}}{C_{\text{пп}j}} = \frac{5.62}{1384\ 463.42} = 4.06 \cdot 10^{-6}$$

Отже, оскільки величина коефіцієнта техніко-економічного рівня визначає його ефективність, то було прийнято рішення імплементувати перший варіант, який налічує в собі глибоке навчання та бібліотеку OpenCV.

Висновки до 5 розділу

В результаті проведеного функціонально-вартісного аналізу програмного продукту, що було розроблено в рамках цієї роботи, отримано висновки стосовно ефективності обраних технологій та стратегій економічних рішень для даного продукту. Оцінки основних характеристик програмного забезпечення, спрямованого на дослідження прогнозованого віку мозку, виявили різноманітні варіанти реалізації, які забезпечують оптимальну стратегію вибору. Метод функціонально-вартісного аналізу було використано для об'єктивної оцінки реальної вартості продукту та виявлення можливостей зниження витрат. Це сприятиме не тільки подальшому потенційному впровадженню продукту в реальні умови, але й розвитку спеціаліста з точки зору архітектора життєвого шляху програмного продукту.

ВИСНОВКИ

У цій бакалаврській роботі було досліджено розробку та реалізацію концепції функціональності трансформації тексту з урахуванням контексту, реалізованою за допомогою інтеграції з чат-ботом ChatGPT.

Завданням на цю роботу була розробка концепції універсальної функціональності для перетворення тексту у веб-додатках таким чином, щоб вона могла бути застосована в широкому спектрі галузей, і могла бути як самостійним додатком, так і бути інтегрованою в уже існуючий веб-додаток. Функціональність також мала відповідати сучасним потребам перетворення тексту і намагатися передбачати майбутні потреби. Отже, розроблена функціональність повинна була бути надійною, гнучкою, універсальною та актуальною не тільки зараз, а й у найближчому майбутньому.

У першому розділі роботи визначається поняття трансформації тексту. Також надано приклади трансформацій тексту з різних галузей, таких як HR (Human Resource) менеджмент, пошукова оптимізація (SEO), маркетинг, тощо. Також в цьому розділі було визначено цілі трансформації тексту та була доведена актуальність її застосування в сучасному світі. Був проведений аналіз способів трансформації тексту, таких як трансформація тексту вручну та трансформація за допомогою інтеграції зі штучним інтелектом. Було доведено, що інтеграція зі штучним інтелектом є значно більш ефективним методом. Було проаналізовано інструмент чат-бот ChatGPT як механізм реалізації контекстної трансформації тексту. Як результат, було доведено, що інтеграція саме з чат-ботом ChatGPT є найбільш оптимальним рішенням для реалізації ефективної та універсальної функціональності трансформації тексту.

У другому розділі формулюються, ґрунтовно аналізуються та обґрунтовуються високорівневі функціональні вимоги для функціональності трансформації тексту з урахуванням контексту з використанням чат-боту ChatGPT. Цими функціональними вимогами є описи найважливіших

функціональностей, що будуть доступні запланованим ролям користувачів: Адміністратору та Користувачу. Для Адміністратора основною є функціональність конфігурації запиту, що буде надсилатися до чат-боту ChatGPT, коли Користувач буде робити запит для трансформації свого тексту. Це є одним з ключових аспектів концепції, що розробляється цій роботі, адже саме конфігурабельність запиту робить всю функціональність гнучкою та універсальною, та забезпечує її актуальність в багатьох сферах зараз і в майбутньому. Для Користувача ж основними є функціональності безпосередньо зосереджені навколо трансформації тексту, а саме: трансформація тексту за допомогою чат-боту ChatGPT, а також подальше покращення трансформованого тексту за допомогою чат-боту ChatGPT.

Також у другому розділі приділено багато уваги нефункціональним вимогам до цієї функціональності, таким як безпека персональних даних та безпечна передача даних, швидкість та ефективність трансформації текстів різного формату, розміру та мови, а також зрозумілість та зручність використання. Згадані нефункціональні вимоги життєва важливі для того, щоб забезпечити ту ж універсальність, а також надійність, безпечність, та зручність користування системою.

Далі в другому розділі було ретельно досліджено та задокументовано найкращі практики створення запитів для трансформації тексту за допомогою чат-боту ChatGPT. Такими є: надання чітких та однозначних інструкцій, вказування мети трансформації, надання явних обмежень, включення додаткового контексту, надання інструкцій щодо включення у трансформований текст перших слів, словосполучень, або висновків штучного інтелекту, а також збереження балансу між наданням інструкцій і творчими можливостями моделі. Це є надзвичайно важливою частиною дослідження, адже дотримання рекомендацій щодо формування ефективного запиту значно підвищує якість та релевантність вихідного тексту.

Також в тому ж розділі проводиться дослідження ринку, щоб оцінити, який функціонал трансформації тексту за допомогою інтеграції зі штучним

інтелектом вже реалізований в найпопулярніших додатках. Це допомогло отримати чітке розуміння поточного конкурентного середовища і тенденцій в ньому на поточний момент. Було детально проаналізовано функціональність написання на трансформації текстів за допомогою штучного інтелекту у додатку "Notion".

На основі ретельно проведеного аналізу вимог, найкращих практик, та конкурентного середовища, у третьому розділі були детально уточнені вимоги. Вони були ретельно описані в форматі користувацьких історій (user story) та критеріїв прийнятності (acceptance criteria) в форматі Gherkin. Також ці описи було підкріплено розробленими прототипами користувацького інтерфейсу, створеними у програмній Figma. Крім того, в третьому розділі роботи було наведено приклади запитів для трансформації тексту для широкого спектру сфер. Ці запити було розроблено на основі найкращих практик, вивчених та задокументованих у другому розділі роботи.

В четвертому розділі наводиться опис роботи функціональності, розробленої на основі задокументованих вимог. Цей опис також підкріплений UML діаграмою послідовностей. Більш того, в четвертому розділі цієї роботи представлені безпосередньо трансформації тексту, які показують тексти "до" і "після" трансформації. Результати цих трансформацій є прямим наслідком дослідження, описаного в другому розділі, та ретельно розроблених запитів, описаних у другому підрозділі третього розділу.

Аналізуючи кожну трансформацію, значне покращення якості тексту є очевидним. Отже, підтверджено, що розроблена концепція функціоналу, як з боку функціональних і нефункціональних вимог, так і з боку інженерії запитів до штучного інтелекту, відповідає завданню та меті цієї роботи. У сукупності висновки цієї роботи чітко показують цінність, актуальність і практичну корисність розробленої функціональності трансформації тексту з урахуванням контексту. Універсальність і ефективність такої функціональності робить її придатною практично для будь-якої сфери застосування, обіцяючи значний прогрес у сфері трансформації текстового контенту.

ПЕРЕЛІК ПОСИЛАНЬ

1. J. Leskovec, Anand Rajaraman, and Jeffrey David Ullman, Mining of Massive Datasets. Cambridge University Press, 2014. URL: <http://infolab.stanford.edu/~ullman/mmds/book.pdf>.
2. Rane, Nitin. (2023). Enhancing Customer Loyalty through Artificial Intelligence (AI), Internet of Things (IoT), and Big Data Technologies: Improving Customer Satisfaction, Engagement, Relationship, and Experience. SSRN Electronic Journal. 10.2139/ssrn.4616051.
3. OpenAI API. URL: platform.openai.com. (дата звернення: 15.04.2024).
4. Тарановський, А. О.; Самойлов, В. Д. ChatGPT і можливість його використання для безекспертного створення тестів. 10.15407/emodel.45.02.044
5. Briganti, Giovanni. (2023). How ChatGPT works: a mini review. European Archives of Oto-Rhino-Laryngology. 281. 1-5. 10.1007/s00405-023-08337-7.
6. Кобилянська, О.; Єсіна, М.; Горбенко, Ю. Порівняльний аналіз штучного інтелекту на основі існуючих чат-ботів. Комп'ютерні науки та кібербезпека. – 2024. – № 2. – С. 26-32. – Режим доступу: <https://periodicals.karazin.ua/cscs/article/view/23119>.
7. Roumeliotis, Konstantinos & Tselikas, Nikolaos. (2023). ChatGPT and Open-AI Models: A Preliminary Review. Future Internet. 15. 192. 10.3390/fi15060192.
8. Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia. (2017). Attention Is All You Need. arXiv:1706.03762, 2017, DOI: <https://doi.org/10.48550/arXiv.1706.03762>.
9. Кімак, Володимир. Вибір нефункціональних вимог при розробці програмного забезпечення. 2023 2nd International Conference on Innovative Solutions in Software Engineering (ICISSE). 2023.

10. Єфремов, М. Ф.; Єфремов, Ю. М.; Єфремов, В. М. Проектування програмного забезпечення з використанням UML. Тези доповідей VIII Міжнародної науково-технічної конференції "Інформаційно-комп'ютерні технології - 2016". URL: <http://eztuir.ztu.edu.ua/123456789/3296>.
11. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance). Official Journal of the European Union. – 2016. – May 4, 2016.
12. ChatGPT Language Capabilities. – Режим доступу: <https://www.tomedes.com/translator-hub/chatgpt-language-capabilities>. (дата звернення: 06.05.2024)
13. Ekin, Sabit. (2023). Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips, And Best Practices. 10.36227/techrxiv.22683919.v2.
14. Mungoli, Neelesh. (2023). Exploring the Synergy of Prompt Engineering and Reinforcement Learning for Enhanced Control and Responsiveness in ChatGPT.
15. How Bing Chat Works. Bing, Microsoft. – 2023. – URL: <https://www.microsoft.com/en-us/bing/do-more-with-ai/how-bing-chat-works?form=MA13KP>. (дата звернення: 08.05.2024)
16. Notion AI Security Practices. Notion. – URL: <https://www.notion.so/help/notion-ai-security-practices>. (дата звернення: 08.05.2024)
17. Enterprise-Grade Generative AI. Grammarly. – URL: <https://www.grammarly.com/business/learn/enterprise-grade-generative-ai/#:~:text=As%20a%20result%2C%20we%20have,LLM%20provider%20at%20this%20time>. (дата звернення: 08.05.2024)
18. Reverse Prompt Engineering. Latent Space. – URL: <https://www.latent.space/p/reverse-prompt-eng>. (дата звернення: 10.05.2024)

19.Lee, Shawn. Notion AI Prompts. GitHub. – 2023. – URL:

<https://github.com/swyxio/ai-notes/blob/main/Resources/Notion%20AI%20Prompts.md>. (дата звернення: 10.05.2024)

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

1. Бекенд-частина додатку Адміністратора:

```
from flask import Flask, request, jsonify, session
from flask_cors import CORS

app = Flask(__name__)
app.secret_key = 'your_secret_key' # Change this to a random secret in production
CORS(app)

# In-memory database simulation
db = {
    "current_query": "Type your initial query here",
    "variables": ["user_name", "user_email", "user_location"]
}

@app.route('/login', methods=['POST'])
def login():
    # This is a simplified login; replace with actual authentication in production
    username = request.json.get('username')
    password = request.json.get('password')
    if username == 'admin' and password == 'password':
        session['logged_in'] = True
        return jsonify({"message": "Login successful"}), 200
    else:
        return jsonify({"message": "Invalid credentials"}), 401

@app.route('/logout', methods=['POST'])
```

```

def logout():
    session.pop('logged_in', None)
    return jsonify({"message": "Logged out"}), 200

@app.route('/query', methods=['GET', 'POST'])
def handle_query():
    if not session.get('logged_in'):
        return jsonify({"message": "Unauthorized"}), 401

    if request.method == 'POST':
        query = request.json.get('query')
        if not query:
            return jsonify({"error": "Please fill in the field"}), 400
        if len(query) > 5000:
            return jsonify({"error": "The field must contain less than 5000 characters"}),
400

        db['current_query'] = query
        return jsonify({"message": "Request saved", "query": query}), 200

    # GET method to view the current query
    return jsonify({"current_query": db['current_query']}), 200

@app.route('/variables', methods=['GET'])
def get_variables():
    if not session.get('logged_in'):
        return jsonify({"message": "Unauthorized"}), 401
    return jsonify({"variables": db['variables']}), 200

@app.route('/add-variable', methods=['POST'])

```

```

def add_variable():
    if not session.get('logged_in'):
        return jsonify({"message": "Unauthorized"}), 401

    variable = request.json.get('variable')
    if variable and variable in db['variables']:
        # Assuming the cursor is at the end of the query for simplicity
        db['current_query'] += f" %%{variable}%%"
        return jsonify({"message": "Variable added", "current_query":
            db['current_query']}), 200
    return jsonify({"error": "Invalid variable"}), 400

if __name__ == '__main__':
    app.run(debug=True)

```

2. Фронтенд-частина додатку Адміністратора:

2.1 QueryEditor Component

```

import React, { useState, useEffect } from 'react';

function QueryEditor() {
    const [query, setQuery] = useState("");
    const [isLoading, setLoading] = useState(false);

    useEffect(() => {
        const fetchQuery = async () => {
            const response = await fetch('http://localhost:5000/query');
            const data = await response.json();
            setQuery(data.current_query);
        };
    });

```

```

    };
    fetchQuery();
  }, []);

const handleSave = async () => {
  setLoading(true);
  const response = await fetch('http://localhost:5000/query', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ query })
  });
  setLoading(false);
  if (response.ok) {
    alert('Query saved successfully');
  } else {
    alert('Failed to save the query');
  }
};

return (
  <div>
    <h2>Edit Query</h2>
    <textarea value={query} onChange={(e) => setQuery(e.target.value)} />
    <button onClick={handleSave} disabled={isLoading}>{isLoading ?
'Saving...' : 'Save Query'}</button>
  </div>
);
}

export default QueryEditor;

```

2.2 AddVariable Component

```
import React, { useState, useEffect } from 'react';

function AddVariable({ onVariableAdd }) {
  const [variables, setVariables] = useState([]);
  const [selectedVariable, setSelectedVariable] = useState("");

  useEffect(() => {
    const fetchVariables = async () => {
      const response = await fetch('http://localhost:5000/variables');
      const data = await response.json();
      setVariables(data.variables);
    };
    fetchVariables();
  }, []);

  return (
    <div>
      <h2>Add Variable</h2>
      <select value={selectedVariable} onChange={(e) =>
setSelectedVariable(e.target.value)}>
        {variables.map((variable) => (
          <option key={variable} value={variable}>{variable}</option>
        ))}
      </select>
      <button onClick={() => onVariableAdd(selectedVariable)}>Add
Variable</button>
    </div>
  );
}
```

```

    );
}

```

```
export default AddVariable;
```

2.3 App Component

```

import React, { useState } from 'react';
import Login from './Login';
import QueryEditor from './QueryEditor';
import AddVariable from './AddVariable';

```

```

function App() {
  const [loggedIn, setLoggedIn] = useState(false);

  return (
    <div className="App">
      <!loggedIn ? <Login onLogin={() => setLoggedIn(true)} /> :
      <div>
        <QueryEditor />
        <AddVariable onVariableAdd={(variable) => console.log('Added',
variable)} />
      </div>
    </div>
  );
}

```

```
export default App;
```

3. Бекенд-частина додатку Користувача:

```
from flask import Flask, request, jsonify, abort
from flask_cors import CORS
import openai

app = Flask(__name__)
CORS(app)

openai.api_key = 'your-openai-api-key'

@app.route('/transform_text', methods=['POST'])
def transform_text():
    if not request.json or 'source_text' not in request.json or 'additional_context' not in request.json:
        abort(400, description="Missing 'source_text' or 'additional_context'")

    source_text = request.json['source_text']
    additional_context = request.json['additional_context']

    if len(source_text) > 5000 or len(additional_context) > 5000:
        abort(400, description="Text exceeds maximum allowed characters.")

    try:
        response = openai.Completion.create(
            engine="davinci",
            prompt=f"{source_text} Context: {additional_context}",
            max_tokens=150
        )
```

```

transformed_text = response.choices[0].text.strip()
return jsonify({'transformed_text': transformed_text})
except Exception as e:
    abort(500, description=str(e))

```

```
@app.route('/copy_text', methods=['POST'])
```

```
def copy_text():
```

```
    text = request.json.get('text')
```

```
    if not text:
```

```
        abort(400, description="No text to copy.")
```

```
    # Simulate copying to clipboard (handled on the frontend in actual
implementation)
```

```
    return jsonify({'message': 'Text copied successfully', 'text': text})
```

```
@app.route('/enhance_text', methods=['POST'])
```

```
def enhance_text():
```

```
    text = request.json.get('text')
```

```
    improvement_request = request.json.get('improvement_request')
```

```
    if not text or not improvement_request:
```

```
        abort(400, description="Text or improvement request missing.")
```

```
    try:
```

```
        response = openai.Completion.create(
```

```
            engine="davinci",
```

```
            prompt=f"Improve this text: {text} Based on: {improvement_request}",
```

```
            max_tokens=200
```

```
        )
```

```
        enhanced_text = response.choices[0].text.strip()
```

```
        return jsonify({'enhanced_text': enhanced_text})
```

```
    except Exception as e:
```

```
        abort(500, description=str(e))
```

```

@app.route('/delete_text', methods=['POST'])
def delete_text():
    # This would typically reset the fields on the frontend
    return jsonify({'message': 'Text deleted successfully'})

if __name__ == '__main__':
    app.run(debug=True)

```

4. Фронтенд-частина додатку Користувача:

4.1 TextTransformer Component

```

import React, { useState } from 'react';
import TextInput from './TextInput';
import TransformationResults from './TransformationResults';

function TextTransformer() {
    const [sourceText, setSourceText] = useState("");
    const [additionalContext, setAdditionalContext] = useState("");
    const [transformedText, setTransformedText] = useState("");

    const handleTransform = async () => {
        const response = await fetch('http://localhost:5000/transform_text', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ source_text: sourceText, additional_context:
additionalContext })
        });
        const data = await response.json();

```

```

    setTransformedText(data.transformed_text);
  };

  return (
    <div>
      <TextInput label="Source Text" text={sourceText} setText={setSourceText}
    />
      <TextInput label="Additional Context" text={additionalContext}
    setText={setAdditionalContext} />
      <button onClick={handleTransform}>Transform Text</button>
      <TransformationResults transformedText={transformedText} />
    </div>
  );
}

export default TextTransformer;

```

4.2 TextInput Component

```

import React from 'react';

function TextInput({ label, text, setText }) {
  return (
    <div>
      <label>
        {label}:
        <textarea value={text} onChange={(e) => setText(e.target.value)} />
      </label>
    </div>
  );
}

```

```
export default TextInput;
```

4.3 TransformationResults Component

```
import React from 'react';
```

```
function TransformationResults({ transformedText }) {
  const handleCopy = () => {
    navigator.clipboard.writeText(transformedText).then(() => {
      alert('Text copied successfully!');
    });
  };

  return (
    <div>
      <h3>Transformed Text:</h3>
      <p>{transformedText}</p>
      <button onClick={handleCopy}>Copy Transformed Text</button>
    </div>
  );
}
```

```
export default TransformationResults;
```

4.4 App Component

```
import React from 'react';
```

```
import './App.css';
```

```
import TextTransformer from './TextTransformer';
```

```
function App() {
```

```
return (  
  <div className="App">  
    <TextTransformer />  
  </div>  
);  
}
```

```
export default App;
```