

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

Навчально-науковий інститут прикладного системного аналізу

Кафедра системного проектування

До захисту допущено:

Завідувач кафедри

_____Вадим МУХІН

«___»_____ 2023 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою

“Інтелектуальні сервіс-орієнтовані розподілені обчислювання”

зі спеціальності 122 “Комп’ютерні науки”

**на тему: “Аналіз можливостей застосування генеративного ШІ у сфері
управління ІТ проектами”**

Виконав:

студент IV курсу, групи ДА-92

Кожемякін Володимир Андрійович

Керівник:

Доцент, Кисельов Геннадій Дмитрович

Консультант з економічного розділу:

доцент, к.е.н.

Рощина Надія Василівна

Рецензент:

доцент кафедри АПЕПС ТЕФ, к.т.н. Шаповалова Світлана Ігорівна

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

Студент(-ка) _____

Київ – 2023

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра системного проектування
Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 122 "Комп'ютерні науки"
Освітньо-професійна програма – "Інтелектуальні сервіс-орієнтовані
розподілені обчислювання"

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Вадим МУХІН

«___» _____ 2023 р.

ЗАВДАННЯ

на дипломну роботу студенту

Кожемякіну Володимир

1. Тема роботи «Аналіз можливостей застосування генеративного ШІ у сфері управління ІТ проектами», керівник роботи Кисельов Геннадій Дмитрович, доцент, затверджені наказом по університету від

«30» травня 2023 р. №2065-с

2. Термін подання студентом роботи – 14 червня 2023 р.

3. Вихідні дані до роботи:

Хмарні платформи, які надають ресурси для навчання генеративних моделей ШІ і отримання результатів роботи з ними. Платформа AI Chat GPT.

4. Зміст роботи: Робота присвячена дослідженню можливостей

застосування генеративного штучного інтелекту в управлінні ІТ-проектами, аналіз існуючих рішень, створення власної методології, що ставить за мету виправити недоліки розглянутих рішень і покращити ефективність роботи керівника проектів. Потрібно виконати наступні завдання:

1. Дослідити сферу управління ІТ-проектами, поняття генеративного штучного інтелекту.
2. Розглянути існуючі рішення керування проектами з використанням ШІ. Розглянути їх переваги та недоліки.
3. Описати власну методологію застосування ШІ у керуванні проектами. Продемонструвати ефективність її застосування.
4. Проаналізувати результати і зробити висновки щодо доцільності застосування розробленої методології та ШІ у керуванні проектами.
5. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Економічний	Рощина Н.В.		

6. Дата видачі завдання

15.04.2023

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Прибуття студента на практику, оформлення і отримання перепусток	Перший тиждень	
2	Проведення інструктажу з техніки безпеки	Перший тиждень	
3	Проведення екскурсій по підприємству	Перший тиждень	
4	Дослідження сфери управління ІТ-проектами, поняття генеративного штучного інтелекту.	Другий тиждень	
5	Розглянути існуючі рішення керування проектами з використанням ІІІ. Розглянути їх переваги та недоліки.	Третій тиждень	
6	Описати власну методологію застосування ІІІ у керуванні проектами. Продемонструвати ефективність її застосування.	Четвертий тиждень	
7	Проаналізувати результати і зробити висновки щодо доцільності застосування розробленої методології та ІІІ у керуванні проектами.	П'ятий тиждень	

Студент

Кожемякін В.А.

Керівник

Кисельов Г.Д.

Анотація

бакалаврської дипломної роботи Кожемякіна Володимира Андрійовича на тему «Аналіз можливостей застосування генеративного ШІ у сфері управління ІТ проектами».

Керівник проектів у сфері розробці ІТ-продуктів виконує сьогодні значну роль у досягненні успіху усього проекту. Тому впровадження нових технологій та розробка нових методологій, що мають за мету підвищити ефективність роботи керівника проекту, значно впливає на успішність та якість проекту. У зв'язку з цим, аналіз можливостей застосування технологій штучного інтелекту у процес керування проектами стає все більш актуальним.

Метою роботи було дослідження сфери керування проектами та можливих шляхів покращення її, шляхом впровадження у процеси керування проектами інструментів генеративного штучного інтелекту. Проведення аналізу можливого впливу результатів впровадження технологій штучного інтелекту на процеси керування ІТ-проектами.

Результати роботи : дослідження поняття генеративного ШІ та керування ІТ-проектами, аналіз та порівняння існуючих рішень, розробка методології автоматизації роботи проектного менеджера з використанням ШІ, прогнози стосовно майбутнього використання штучного інтелекту в сфері управління.

Загальний обсяг роботи 79 с., 26 рис., 15 джерел.

Ключові слова: Керування ІТ-проектами, генеративний штучний інтелект, Chat GPT, методологія, SDLC.

Abstract

of the bachelor's thesis of Kozhemiakin Volodymyr Andriyovych on "Analysis of the possibilities of using generative AI in the field of IT project management".

Today, a project manager in the field of IT product development plays a significant role in achieving the success of the entire project. Therefore, the introduction of new technologies and the development of new methodologies aimed at increasing the efficiency of the project manager has a significant impact on the success and quality of the project. In this regard, the analysis of the possibilities of applying artificial intelligence technologies in the project management process is becoming increasingly relevant.

The purpose of the study was to investigate the field of project management and possible ways to improve it by introducing generative artificial intelligence tools into project management processes. To analyze the possible impact of the results of the implementation of artificial intelligence technologies on IT project management processes.

Results: research of the concept of generative AI and IT project management, analysis and comparison of existing solutions, development of a methodology for automating the work of a project manager using AI, forecasts for the future use of artificial intelligence in the field of management.

The total volume of the work is 79 p., 26 figures, 15 sources.

Keywords: IT project management, generative artificial intelligence, Chat GPT, methodology, SDLC.

Зміст

ВСТУП	9
1 ПРОЕКТНИЙ МЕНЕДЖМЕНТ	11
2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ДОСЯГНЕНЬ У СФЕРІ ГЕНЕРАЦІЇ УПРАВЛІНСЬКИХ РІШЕНЬ	14
2.1. Генеративний ІІІ	14
2.2. Генеративний ІІІ у сфері управління проектами	15
2.3. Висновки до розділу	19
3 ОГЛЯД ПРОБЛЕМ В УПРАВЛІННІ ІТ ПРОЕКТАМИ	22
3.1 Аналіз наявних проблем у сфері керування проектами	22
3.2 Висновки до розділу	25
4 МЕТОДОЛОГІЯ РОБОТИ ПРОЕКТНОГО МЕНЕДЖЕРА З ЗАСТОСУВАННЯМ ГЕНЕРАТИВНОГО ІІІ	26
4.1. Поняття моделі життєвого циклу розробки програмного забезпечення	27
4.2. Опис методології	30
4.3. Висновки до розділу	33
5 ЕКСПЕРИМЕНТ ІЗ ЗАСТОСУВАННЯМ МОДЕЛІ СНАТ GPT	34
5.1 Збір та аналіз вимог	34
5.2 Проектування та створення дизайну	42
5.3 Етап розробки продукту	51
5.4 Тестування	56
5.5 Реліз та підтримка продукту	58
5.6 Висновки до розділу	61
6 МАЙБУТНЄ ГЕНЕРАТИВНОГО ІІІ В УПРАВЛІННІ ПРОЕКТАМИ	65
7 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ЗАСТОСУВАННЯ ГЕНЕРАТИВНОГО ІІІ У УПРАВЛІННІ ІТ-ПРОЕКТАМИ	69
7.1 Вступ	69
7.2 Економічне обґрунтування результатів застосування генеративного ІІІ	70
7.3 Висновки розділу	74
ВИСНОВКИ	75
ДЖЕРЕЛА ІНФОРМАЦІЇ	77

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ — штучний інтелект.

AI (Artificial Intelligence) – штучний інтелект англійською мовою.

SDLC (Software Development Life Cycle) - сукупність окремих етапів робіт, що проводяться у заданому порядку протягом періоду часу, який починається з вирішення питання про розроблення програмного забезпечення і закінчується припиненням використання програмного забезпечення.

ІТ-проект - це проект, в рамки якого входять роботи, пов'язані з інформаційними технологіями, які в свою чергу спрямовані на створення, розвиток і підтримку інформаційних систем.

ВСТУП

Сучасний технологічний прогрес і стрімкий розвиток інформаційних технологій викликають необхідність впровадження нових і ефективних підходів до управління ІТ проектами. Застосування генеративного штучного інтелекту в цій сфері відкриває безліч можливостей для покращення процесів управління, оптимізації ресурсів та досягнення більш високої ефективності.

Проблеми, пов'язані з управлінням ІТ проектами, стають все більш складними і вимагають пошуку нових підходів для їх вирішення. Традиційні методи управління можуть бути недостатньо гнучкими і не враховувати великий обсяг даних та навколишніх факторів, що впливають на проект. Тому, актуальним завданням є аналіз можливостей застосування генеративного штучного інтелекту в сфері управління ІТ проектами.

Метою даної дипломної роботи є дослідження та аналіз можливостей застосування генеративного штучного інтелекту у сфері управління ІТ проектами. При цьому, спеціальна увага буде приділена розробці і впровадженню методології, що базується на застосуванні генеративного ШІ, з метою покращення ефективності управління проектами.

Актуальність даної роботи полягає у тому, що використання генеративного штучного інтелекту в сфері управління ІТ проектами може дозволити покращити процеси планування, виконання і контролю проектів, а також забезпечити швидке і точне прийняття рішень на основі аналізу великого обсягу даних. Новизна даної роботи полягає у розробці та впровадженні генеративних алгоритмів і моделей у сферу управління ІТ проектами, що відкриває нові перспективи для покращення результативності і досягнення успіху в цій галузі.

Розробка і впровадження генеративного штучного інтелекту в сфері управління ІТ проектами є необхідною, оскільки це дозволить забезпечити

більш точне прогнозування, ефективне ресурсне планування, оптимізацію процесів управління та досягнення поставлених цілей проекту. Така розробка відповідає вимогам сучасного ринку і може стати конкурентною перевагою для компаній, що займаються управлінням ІТ проектами.

1 ПРОЕКТНИЙ МЕНЕДЖМЕНТ

Проектний менеджмент (Project management) - набір знань, навичок, інструментів і методів для реалізації цілей, корисних для людей. Це може бути розробка програмного забезпечення для поліпшення бізнес-процесів, будівництво споруд, надання допомоги постраждалим після стихійних лих або розширення продажів на нових географічних ринках. Усі ці приклади є проектами, де управління проектами грає ключову роль у досягненні успіху. Однією з основних відмінних рис управління проектами від звичайного "менеджменту" є його спрямованість на досягнення кінцевого результату в обмежений часовий проміжок, у порівнянні з безперервним процесом, що є ознакою саме менеджменту. Це вимагає від фахівця з управління проектами широкого спектру навичок, включаючи технічні навички, навички управління людьми, вирішення конфліктів та глибоке розуміння бізнесу.

Проект - це тимчасова ініціатива для створення цінності шляхом розробки унікальних продуктів, надання послуг або покращення процесів. Деякі проекти розв'язують проблеми швидко, інші вимагають більш тривалого часу. Вони можуть включати розробку програмного забезпечення або планування ліквідації наслідків стихійних лих. Проект має життєвий цикл, що складається з фаз ініціації, планування та виконання. Цей цикл допомагає керівникам проекту планувати та виконувати задачі, забезпечуючи успіх проекту. Узагальнюючи, проект - це добре спланована діяльність з визначеним початком та кінцем, командою, бюджетом і графіком робіт [1].

Отже, керування проектами - це систематичний підхід до планування, організації, керування та контролю за реалізацією проектів. Воно використовується для досягнення мети проекту, забезпечення виконання завдань у встановлені строки і в рамках визначеного бюджету.

Управління проектами базується на розумінні сутності проекту. Воно охоплює всі аспекти проекту, включаючи цілі, обсяг робіт, ресурси, ризики, комунікацію та інші фактори, що впливають на успіх проекту. Управління проектами вимагає використання методологій, інструментів і процесів для ефективного планування, розподілу ролей і обов'язків, моніторингу прогресу та вирішення проблем. Воно також включає в себе встановлення механізмів звітності і комунікації, що допомагають утримувати команду проекту та зацікавлені сторони в курсі процесу.

Методології управління проектами - це набір знань, підходів та інструментів, які використовуються для планування, організації та керування проектами. Вони надають рамки і стандарти, які допомагають ефективно виконувати проекти та досягати їхніх цілей [2].

Декілька найбільш популярних методологій управління проектами:

1. Waterfall (Каскадна методологія): Це традиційний підхід, де проект розбивається на послідовні етапи, і кожен етап розпочинається після завершення попереднього. Вона підходить для проектів з чіткими і стабільними вимогами, де зміни малоймовірні.
2. Agile (Гнучкий підхід): Це ітеративна методологія, де проект розбивається на короткі цикли розвитку, відомі під назвою "спринти". Команда працює над малими фрагментами проекту, надаючи можливість налагоджувати та змінювати вимоги на ходу. Популярні методи Agile включають Scrum, Kanban та Extreme Programming (XP).
3. Prince2: Це методологія, розроблена у Великобританії, яка надає детальні і структуровані принципи для управління проектами. Вона визначає ролі та відповідальності, фази проекту та необхідні документи. Prince2 може бути адаптований до різних типів проектів та розмірів команд.

4. PMBOK (Project Management Body of Knowledge): Це стандарт, розроблений Інститутом управління проектами (PMI), який включає набір процесів, знань та навичок управління проектами. PMBOK надає загальну рамку для управління проектами, що може бути придатною для різних індустрій та проектів.

Це лише кілька прикладів методологій управління проектами, існує безліч інших підходів, які можуть бути використані в залежності від потреб та характеру проекту. Вибір методології залежить від конкретних вимог, розміру та складності проекту, а також від уподобань та досвіду команди проекту.

Під час роботи над проектом від керівника проекту необхідно не лише слідкувати за виконанням роботи та дотриманням графіку, але й приймати важливі рішення для подолання проблем, що постійно виникають у ході роботи над проектом. Для цього існують інструменти прийняття рішень, деякі серед них:

1. Аналіз Парето - допомагає пріоритизувати проблеми або завдання залежно від їх впливу, зосереджуючись на найважливіших.
2. «Шість капелюхів» Едварда де Боно дозволяють розглянути проблему з різних точок зору та сприяють креативному мисленню.
3. Ментальне картування - візуалізує думки та ідеї, що допомагає приймати рішення.

Важливо пам'ятати, що кожна ситуація унікальна, і використання інструментів потребує індивідуального підходу. Керівник проекту повинен мати досвід та здібності, щоб визначити, який інструмент краще використовувати в конкретній ситуації.

2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ДОСЯГНЕНЬ У СФЕРІ ГЕНЕРАЦІЇ УПРАВЛІНСЬКИХ РІШЕНЬ

Генеративні штучний інтелект (ШІ) - це програмні алгоритми, які можуть генерувати тексти, музику або зображення. Ця технологія починає все більше використовуватись в сучасному світі, включаючи сферу керування ІТ проектами. Застосування цієї технології може покращити ефективність та продуктивність проекту, зменшити час, який потрібен на вирішення проблем, і підвищити якість продукту.

У рамках роботи буде зосереджено увагу на дослідженні того, як генеративний ШІ може бути використаний для вирішення певних проблем у керуванні ІТ проектами. Дослідження такої теми є дуже актуальним у сучасному світі, де високий рівень конкуренції вимагає максимальної ефективності та продуктивності від працівників. В останні роки зросла кількість ІТ-проектів, що потребують використання штучного інтелекту в процесі управління. Генеративний ШІ є одним з підходів до використання штучного інтелекту у керуванні проектами. Він дозволяє генерувати нові рішення на основі вхідних даних, які забезпечують більш ефективне прийняття рішень в умовах невизначеності.

2.1. Генеративний ШІ

Генеративний ШІ відноситься до моделей штучного інтелекту, які призначені для застосування до широкого кола завдань. Ці моделі не є специфічними для завдань і можуть використовуватися для вирішення різних задач. Одним з найбільш цікавих напрямків досліджень ШІ є розробка загальних моделей ШІ, які можуть бути застосовані до широкого спектру завдань. Генеративний ШІ автоматично знаходить патерни та функції всередині інформації без будь-якого заздалегідь запрограмованого упередження, а також

вивчає та класифікує нові аномальні патерни, коли вони виявляються. Ця не задана властивість загального ШІ може мати критичне значення в питаннях безпеки [3].

Одним із прикладів текстового ШІ є нова система під назвою “Imagen”. Ця система може генерувати зображення на основі текстових описів. Наприклад, якщо ввести текст «червоне яблуко на білому тлі», то система згенерує зображення червоного яблука на білому тлі.

Інший приклад генеративного ШІ, який генерує зображення на основі текстового запиту - “Shutterstock AI Image Generator”. Цей інструмент дозволяє користувачам миттєво генерувати зображення з текстових описів. Створені зображення можуть бути використані для різних цілей, таких як маркетингові кампанії, повідомлення в соціальних мережах тощо.

Окрім графічних генеративних моделей AI, існують також інші типи загальних моделей AI. Наприклад, “Chat GPT” - це інтерактивна версія GPT (Generative Pre-trained Transformer), що розроблена OpenAI. Вона є генеративною лінгвістичною моделлю, яка може генерувати людиноподібний текст на основі текстових запитів. Ця модель може бути використана для різних додатків, таких як чат-боти, генератори контенту, текстові редактори тощо.

Генеративний ШІ приваблює компанії та споживачів, оскільки його підтримують надзвичайно потужні технологічні гіганти, такі як Google, Microsoft, IBM та інші. Він стає все більш популярним завдяки своїй універсальності і здатності застосовуватися до широкого спектру завдань [4].

2.2. Генеративний ШІ у сфері управління проектами

Штучний інтелект призначений для надання допомоги у прийнятті рішень, коли дані, параметри та змінні, що беруть участь, знаходяться за межами людського розуміння. Здебільшого системи ШІ приймають правильні

рішення з урахуванням обмежень. Інструменти на основі штучного інтелекту допомагають менеджерам та керівникам розставляти пріоритети та приймати правильні рішення на кожному етапі від планування до впровадження. Це допомагає обробляти дані проекту та помічати закономірності. Розглянемо існуючі розробки для керування проектами на основі генеративного ШІ.

1. **Wrike** - це програмне забезпечення для управління проектами, яке надає платформу для співпраці команд над проектами. Воно пропонує такі функції : керування завданнями, відстеження часу та інструменти для співпраці команд. Wrike також використовує ШІ, щоб допомогти керівнику проектів автоматизувати рутинні завдання, покращити комунікацію та оптимізувати робочі процеси. За допомогою рішення Work Intelligence від Wrike, керівники проектів можуть пріоритезувати найважливішу роботу та забезпечити, що їхня команда працює над найціннішими завданнями з рекомендаціями автоматичного пріоритезування завдань від Wrike [5].

Функції, що працюють на базі ШІ в Wrike, включають інтелектуальне пріоритезування завдань, управління навантаженням та розподіл ресурсів. Платформа використовує алгоритми машинного навчання для аналізу даних з попередніх проектів та надання висновків щодо оптимізації майбутніх проектів.

Функції на базі ШІ в Wrike також можуть допомогти керівнику проектів економити час та зменшувати помилки, автоматизуючи рутинні завдання, такі як планування зустрічей та надсилання нагадувань.

2. **Proggio** - програмне забезпечення для управління проектами, яке надає платформу для співпраці команд над проектами. Воно має такі функції, як управління завданнями, відстеження часу та інструменти для співпраці команд. Proggio також використовує штучний інтелект для автоматизації

рутинних завдань, поліпшення комунікації та оптимізації робочих процесів. За допомогою AI-функцій Proggio, керівники проектів можуть пріоритезувати найважливішу роботу та забезпечувати те, що їхні команди спрямовують свої зусилля на найціннішу роботу завдяки автоматизованим рекомендаціям щодо пріоритетів завдань від Proggio [6].

AI-функції Proggio включають інтелектуальне пріоритезування завдань, управління робочим навантаженням та розподіл ресурсів. Платформа використовує алгоритми машинного навчання для аналізу даних з минулих проектів та надання уявлень про те, як оптимізувати майбутні проекти. AI-функції Proggio можуть допомогти керівнику проектів економити час та зменшувати помилки за рахунок автоматизації рутинних завдань, таких як планування зустрічей та нагадування [7]. Функція інтелектуального пріоритезування завдань Proggio допомагає керівникам проектів визначати найважливішу роботу, аналізуючи дані з минулих проектів та надаючи уявлення про те, які завдання найцінніші. Ця функція може допомогти менеджерам проектів гарантувати, що їхня команда націлена на найціннішу роботу з рекомендаціями щодо визначення пріоритетів автоматизованих завдань Proggio.

3. **Forecast** - це програмне забезпечення для управління проектами та ресурсами, яке використовує штучний інтелект для допомоги менеджерам проектів автоматизувати операції, надавати можливості працівникам та використовувати узагальнені дані для підвищення результативності бізнесу та успіху клієнтів. Forecast використовує штучний інтелект для «читання» та інтегрування даних з різних підприємницьких застосунків, щоб побудувати більшу картину проекту та потенційних результатів [8].

Функції, що базуються на штучному інтелекті в Forecast, включають надання реального часу для допомоги в плануванні та

виконанні проектів, передбачення можливих ризиків та уточнення, якщо щось йде не так, і навчання на попередніх проектах для покращення виконання. Штучний інтелект в Forecast також допомагає менеджерам проектів економити час та уникати помилок, автоматизуючи рутинні задачі, такі як планування зустрічей та нагадування. Штучний інтелект в Forecast може також допомогти менеджерам проектів пріоритезувати найважливішу роботу, аналізуючи дані з попередніх проектів та надаючи відповідні рекомендації [9]. Ця функція допоможе менеджерам проектів переконатися, що їх команда працює над найважливішими завданнями завдяки автоматизованому рекомендаційному списку завдань від Forecast.

4. **Asana** - це програмне забезпечення для управління проектами, яке надає платформу для співпраці команд над проектами. Воно пропонує такі функції, як управління завданнями, відстеження часу та інструменти для співпраці в команді. Asana також використовує штучний інтелект для допомоги менеджерам проектів автоматизувати рутинні задачі, покращувати комунікацію та оптимізувати робочі процеси. За допомогою функцій, що працюють на базі штучного інтелекту, менеджери проектів можуть пріоритезувати свої найважливіші завдання та забезпечити те, що їх команда працює над найважливішими завданнями, отримуючи автоматизовані рекомендації щодо пріоритетів завдань з Asana [10]. Серед функцій Asana, що працюють на базі штучного інтелекту, є інтелектуальне пріоритезування завдань, управління навантаженням та розподіл ресурсів. Платформа використовує алгоритми машинного навчання для аналізу даних з минулих проектів та надання висновків щодо того, як оптимізувати майбутні проекти. Функції Asana, що працюють на базі штучного інтелекту, допомагають менеджерам проектів економити час та уникнути помилок, автоматизуючи рутинні задачі, такі як планування зустрічей та надсилання нагадувань.

Інтелектуальна функція визначення пріоритетів завдань Asana допомагає менеджерам проектів визначати пріоритети своєї найважливішої роботи, аналізуючи дані минулих проектів та надаючи уявлення про те, які завдання є найбільш цінними. Крім того, Asana пропонує інші корисні можливості, такі як графіки Ганта, календарі, звіти та панель інструментів аналітики, що допомагає проектним менеджерам зрозуміти прогрес проекту та ефективність роботи команди. Графіки Ганта Asana дають змогу організувати задачі за часовою лінією і відстежувати прогрес проекту. Календар Asana дозволяє планувати та відстежувати дедлайни завдань і подій. Звіти Asana надають проектним менеджерам змогу збирати та аналізувати дані про прогрес проектів, зокрема час, який команда витрачає на завдання, та виконання дедлайнів.

Панель інструментів аналітики Asana надає можливість переглядати відомості про кількість завдань, на яких працюють команди, та про кількість завдань, які були виконані. Завдяки цьому проектні менеджери можуть ефективно використовувати ресурси команди та розуміти, наскільки ефективно вони працюють.

2.3. Висновки до розділу

Було розглянуто генеративний ШІ та його застосування у керуванні ІТ проектами. Детально було розглянуто такі платформи, як Asana, Forecast, Proggio та Wrike, що використовують штучний інтелект для автоматизації та оптимізації процесів керування проектами.

Генеративний ШІ в цих рішеннях використовується для автоматизації повторюваних задач та збільшення продуктивності в процесі керування проектами. Він здатен аналізувати великі об'єми даних, навчатись на їх основі та розуміти контекст. Таким чином, генеративний ШІ може виявляти залежності між проектними елементами та робити прогнози результатів проекту. Asana

пропонує інтелектуальну пріоритизацію завдань, керування робочим навантаженням та розподіл ресурсів, що може забезпечити ефективну роботу команди над проектом. Forecast використовує штучний інтелект для автоматизації рутинних задач, які можуть забрати багато часу, таких як планування зустрічей та нагадування. Proggio пропонує розширену аналітику та навчання на основі даних, що дозволяє проектним менеджерам використовувати історичні дані для покращення майбутніх проектів. Wrike пропонує інтеграцію зі сторонніми додатками, що може забезпечити більш широкий спектр функцій та можливостей для проектних менеджерів.

Після порівняння цих рішень, можна сказати, що кожен з них має свої унікальні переваги та недоліки. Всі вони демонструють, що застосування генеративного ШІ у керуванні проектами може покращити продуктивність та знизити ризики помилок, збільшуючи швидкість та точність процесів прийняття рішень. Проте, розглянуті рішення мають і недоліки:

- **Обмеженість в адаптації** - Комерційні ПЗ, такі як Asana, Wrike, Forecast, Proggio, мають встановлений функціонал і обмежений в індивідуальному налаштуванні. Це може призвести до обмежень у використанні унікальних або специфічних методів роботи з ШІ, які можуть бути важливими для роботи з проектом.
- **Висока вартість** - Комерційні ПЗ є значно дорожче, особливо якщо потрібні додаткові функції або додаткові ліцензії для команди.
- **Обмеженість у гнучкості** - Комерційні ПЗ можуть мати обмежені можливості інтеграції з іншими системами або інструментами. Використання власної методології дає більшу гнучкість у виборі інтеграцій та розширенні можливостей вашої системи.
- **Обмежений функціонал** - Комерційні ПЗ для управління проектами забезпечують базовий набір інструментів для організації завдань та

комунікації, але обмежені у створенні складних робочих документів. Наприклад, якщо потрібно створити докладні звіти, аналізи або документи з розрахунками, може знадобитися додатковий зовнішній інструмент або ручна робота. Також подібні до тих рішень, що були розглянуті в цьому розділі, не надають можливості працювати з будь-якими даними, крім тих, що стосуються завдань та деталей проекту. До прикладу, подібні інструменти не надають функціоналу для обробки відгуків клієнтів, робота з різними діаграмами та можливості їх створення теж вкрай обмежені.

Наведені недоліки, не дають впевненості, що подібні рішення на основі генеративного ШІ є безальтернативним рішенням сьогодні. Оскільки, вони не надають функціоналу з широким спектром налаштувань та функцій. Тому, необхідно розглянути власний підхід у застосуванні генеративного штучного інтелекту в керуванні проектами.

3 ОГЛЯД ПРОБЛЕМ В УПРАВЛІННІ ІТ ПРОЕКТАМИ

3.1 Аналіз наявних проблем у сфері керування проектами

Управління проектами вже доволі довгий проміжок часу є дуже популярним напрямом в ІТ, через це посада керівника проекту набула масовості, а отже і зменшився поріг входу у цю сферу. Окрім іншого, це має як переваги, так і недоліки, їх і розглянемо детальніше.

1. Останні роки спостерігається тенденція переходу великої кількості людей з різних сфер праці саме у ІТ. Крім стандартних шляхів зміни кваліфікації, таких як отримання вищої освіти у новій сфері, з популяризацією Інтернету - значно виросла кількість онлайн курсів та шкіл, що пропонують отримати нову кваліфікації набагато швидше, більшість програм розраховані на декілька місяців, в той час як навчання у закладах вищої освіти для отримання ступеня бакалавра потрібно чотири роки. Але кількість матеріалу, що розглядаються на таких курсах обмежена, тому випускники таких шкіл, сильно недостають у розумінні напрямку який вони обрали. Через це, коли людина з такою освітою потрапляє на посаду керівника проекту, то стикається з “бар’єром”, що найчастіше виникає з розробниками. Оскільки менеджеру з базовими знаннями професії важко зрозуміти всі деталі роботи розробників, в силу того, що вища освіта пропонує ширше охоплення ІТ сфери, в той час як профільний курс дає лише базове розуміння професії, яке потребує років практики та самонавчання для досягнення глибшого професіоналізму. Для роботи керівника проекту розуміння ключових аспектів роботи розробників є дуже важливим, оскільки прямо впливає на якість планування роботи, прийняття рішень та вирішення проблем, що обов’язково будуть виникати в ході роботи. Також низький рівень кваліфікації керівника проекту - означає те, що задачі, які йому будуть

довіряти, будуть рутинними та одноманітними, які не потребують творчості, а отже кожного разу така задача буде викликом для робітника, що напряду впливає на його інтерес до його роботи і загальний емоційний стан.

2. Також часто на проектах через недостатньо приділену увагу до деталізації проекту, а саме неповне розуміння цілей, розробки та реалізації проекту. Основою для якісного виконання будь-якого проекту є достатня деталізація для всебічного планування, оцінки та розподілу ресурсів. Відсутність належної деталізації є усіх подальших проблем, що можуть виникнути. Одним з ключових признаков є відсутність належного розподілу ресурсів, що призводить до неефективності та труднощів у відстеженні витрат ресурсів у порівнянні з виконаною роботою. Крім того, нездатність чітко відокремити витрати на проект від загальних витрат ускладнює точний контроль витрат і бухгалтерський облік. Більше того, неточні оцінки та планування часто спостерігаються як наслідок недостатнього визначення робіт, що призводить до нереалістичних планів та етапів проекту. Однією з причин є неповне розуміння цілей, дизайну та реалізації проекту, що призводить до відсутності ясності щодо необхідних робіт. Інша причина може бути пов'язана з організаційною культурою, яка не надає пріоритету чіткому визначенню робіт. Крім того, слабка орієнтація персоналу на проект або прибуток і загальний брак дисциплінованого управлінського підходу також можуть сприяти неадекватному визначенню робіт. У деяких випадках невеликі проекти можуть не отримувати достатньої уваги з боку керівництва або високоякісних ресурсів через їхній розмір порівняно зі звичайним робочим навантаженням організації [11].
3. Нереалістичні графіки створюють значні проблеми в управлінні проектами, часто призводячи до зайвих витрат, пропущення етапів

проекту і постійного перегляду методів вимірювання продуктивності. Неточне планування може поставити під загрозу успіх всього проекту, порушити розподіл ресурсів і знизити загальну ефективність проекту. Одним з найпомітніших показників того, що визначений на перших етапах графік виконання робіт - надмірні витрати, які виникають, коли проекти перевищують виділений їм бюджет через нереалістичні часові рамки та неадекватне планування ресурсів. Пропущені проміжні дати - ще один поширений признак, який вказує на те, що проект відстає від графіка. Крім того, постійне переглядання методів вимірювання продуктивності спостерігається як наслідок відставання від графіка або необхідності модернізації функції управління проектом. Проблема нереалістичних графіків виникає з кількох причин. Конкурентна боротьба часто призводить до того, що учасники пропонують надто амбітні графіки, намагаючись отримати контракт. Умови контракту, які не передбачають покарання за незадовільне виконання графіку, можуть ще більше закріпити нереалістичні терміни, оскільки не передбачають жодних наслідків за недотримання термінів. Неадекватне планування, засноване на минулому досвіді, також є поширеною причиною, оскільки організації, які не ведуть базу даних про виконання графіків і витрат, опиняються в не вигідному становищі. Недостатня кількість часу, витраченого на розробку графіків та аналіз відповідних факторів, може призвести до оптимістичного планування, коли встановлюються нереалістичні часові рамки, щоб досягти високої продуктивності персоналу [11].

4. Неналежні практики контролю та обліку витрат часто призводять до труднощів у порівнянні кошторисних витрат з фактичними витратами та неточного розподілу витрат. Ефективний контроль та облік витрат є важливим для дотримання встановленого бюджету, забезпечення

фінансової прозорості та сприяння прийняттю обґрунтованих рішень. Труднощі у порівнянні кошторису з фактичними витратами можуть перешкоджати точному бюджетуванню та прогнозуванню проекту, що ускладнює оцінку ефективності та фінансової життєздатності проекту. Неточний розподіл витрат, коли витрати неналежним чином віднесені до конкретних компонентів або заходів проекту, може призвести до викривлення фінансової звітності та спотворення інформації про витрати проекту.

Неналежний контроль витрат та бухгалтерський облік зумовлений кількома причинами. Відсутність запланованих багатопроєктних інформаційних систем може перешкоджати ефективному управлінню витратами в рамках декількох проєктів, що ускладнює агрегування та всебічний аналіз даних про витрати. Поганий взаємозв'язок між визначенням робіт, оцінкою та накопиченням витрат може призвести до розбіжностей і непослідовності у звітності про витрати, що призводить до недостатньої прозорості та підзвітності. Недостатня обізнаність персоналу проєкту з методами контролю витрат може призвести до неефективного моніторингу проєктних витрат і неадекватних заходів для запобігання перевитратам. Недосконалість методів поділу та розподілу витрат ще більше загострює проблему, перешкоджаючи точному визначенню та віднесенню витрат до відповідних елементів проєкту [11].

3.2 Висновки до розділу

У розділі було розглянуто основні проблеми у сфері керування ІТ-проєктами. Метою наступних розділів буде надання рішення, яке буде враховувати описані проблеми, і буде надавати інструменти для їх рішення.

4 МЕТОДОЛОГІЯ РОБОТИ ПРОЕКТНОГО МЕНЕДЖЕРА З ЗАСТОСУВАННЯМ ГЕНЕРАТИВНОГО ШІ

Розглянуті в попередніх розділах недоліки існуючих інструментальних рішень для управління проектами і проблеми, що існують в роботі керівників проектів - вимагають розробки власного підходу до роботи з генеративним ШІ. Цей підхід має бути гнучким, адаптованим до сучасних умов та особливостей проекту. Він повинен враховувати поточні тенденції, новітні технології та здатний ефективно впроваджувати інновації.

Підхід має базуватись на застосуванні генеративних моделей на кожному етапі життєвого циклу розробки програмного забезпечення (SDLC - Software Development Life Cycle). Це виправить значний недолік розглянутих існуючих рішень для управління проектами, оскільки як було зазначено - одним з вагомих їх недоліків є відсутність функціоналу роботи з всіма етапами життєвого циклу.

В якості генеративного ШІ пропонується розглянути згадану раніше модель від OpenAI - Chat GPT. Але важливо зазначити, що зараз світові технологічні компанії активно розробляють власні рішення, які мають створити конкуренцію для моделі від OpenAI. Тому запропонований варіант застосування ШІ в роботі проектного менеджера не залежить від інструментального рішення, і якщо через певний час, можна експериментувати зі застосуванням LLama від Facebook, до прикладу. Оскільки зміна обраної моделі не буде глобально впливати на той результат, що користувач отримує від роботи з ШІ. Бо як буде розглянуто в наступному розділі - генеративні ШІ є лише інструментом у руках керівника проекту, тому якість отриманого результату залежить виключно від його власних навичок, знань і вміння працювати з генеративним ШІ.

4.1. Поняття моделі життєвого циклу розробки програмного забезпечення

Спочатку розглянемо що таке життєвий цикл розробки ПЗ, і які задачі виконує керівник проектів на кожному з етапів роботи. Життєвий цикл розробки програмного забезпечення (SDLC) - сукупність окремих етапів робіт, що проводяться у заданому порядку протягом періоду часу, який починається з вирішення питання про розроблення програмного забезпечення і закінчується припиненням використання програмного забезпечення [12]. Управління розробкою програмного забезпечення може бути складним через мінливі вимоги, оновлення технологій та міжфункціональну співпрацю. Методологія життєвого циклу розробки програмного забезпечення надає структуру управління з конкретними результатами на кожному етапі процесу розробки програмного забезпечення. В результаті всі зацікавлені сторони заздалегідь узгоджують цілі та вимоги до розробки програмного забезпечення, а також мають план досягнення цих цілей.

Розглянемо основні етапи SDLC, що наведено на рисунку 1.

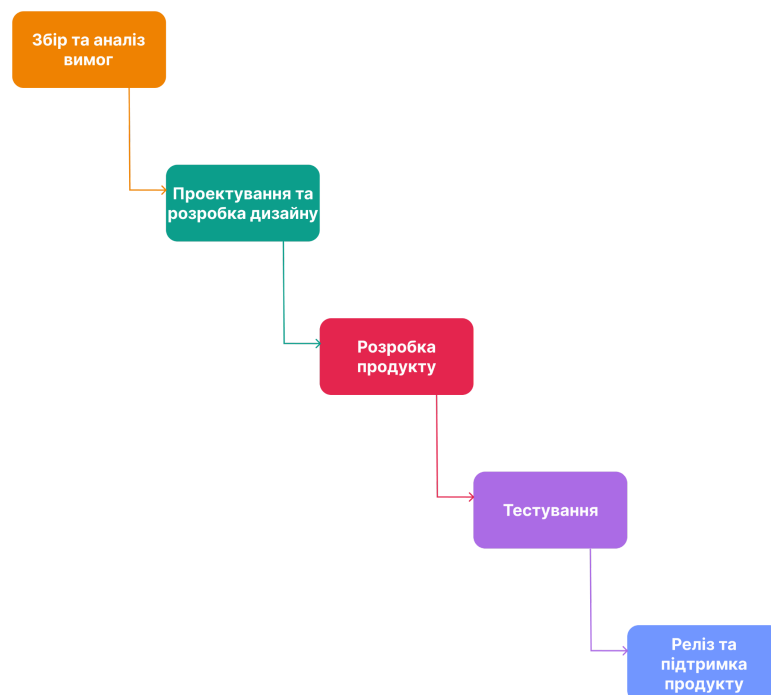


Рис. 1 - Етапи життєвого циклу розробки ПЗ

- 1. Збір та аналіз вимог** - на першому етапі проектний менеджер співпрацює з усіма зацікавленими сторонами, серед них - замовники, користувачі, бізнес-аналітики та інші команди, з метою з'ясування їх потреб та очікувань щодо програмного продукту. Це включає в себе активну комунікацію, проведення співбесід, інтерв'ю та спільних засідань для отримання детальної інформації про вимоги. Після збору вимог настає етап аналізу, де керівник проекту ретельно аналізує зібрані вимоги, перевіряє їх зрозумілість, конкретність, повноту та реалізованість. Він співпрацює з бізнес-аналітиками та технічними експертами для визначення можливих ризиків, обмежень та варіантів реалізації. Також проектний менеджер визначає пріоритетність вимог і виокремлює ключові функціональні та нефункціональні вимоги. Після аналізу вимог проектний менеджер допомагає у документуванні отриманих вимог. Це може бути опис функціональності, інтерфейсів, вимог до продуктивності, безпеки та інші аспекти. Проектний менеджер також забезпечує документування вимог у зручному для всіх зацікавлених сторін форматі.
- 2. Проектування та створення дизайну** - на цьому етапі відбувається створення дизайну і розробка архітектури майбутнього застосунку, і проектний менеджер співпрацює з розробниками, дизайнерами та іншими фахівцями з метою розгляду вимог та визначення оптимальних рішень з точки зору технічного виконання та дизайну. Керівник забезпечує ефективну комунікацію та співробітництво всієї команди проекту. Він також допомагає визначити загальну архітектуру системи, включаючи структуру, компоненти та їх взаємозв'язок, забезпечуючи, щоб архітектура була масштабованою, ефективною та добре структурованою. Проектний менеджер встановлює стандарти та процедури для контролю якості проектування та дизайну, проводить огляди та аудит дизайну, щоб виявити та виправити потенційні проблеми. Він ідентифікує потенційні

ризиками, пов'язані з проектуванням та дизайном, розробляє стратегії їх управління та плани мінімізації ризиків. Також забезпечує зворотний зв'язок та узгодження замовника щодо проектного дизайну, представляє результати проектування та дизайну замовнику, враховуючи його відгуки та внесені зміни для забезпечення відповідності очікуванням замовника.

3. Розробка продукту - на ньому керівник проекту забезпечує організацію та керування роботою команди розробників, програмістів та інших фахівців, розподіляючи завдання, встановлюючи пріоритети та контролюючи виконання робіт. Він також керує управлінням ресурсами, такими як бюджет, матеріали, обладнання та інфраструктура, програмне забезпечення, забезпечуючи їх закупівлю та ефективне використання. Крім того, керівник проекту встановлює стандарти якості і процедури контролю для вироблення високоякісного продукту, проводить тестування та оцінку якості розробленого продукту. Він також виявляє та керує змінами, які виникають під час розробки продукту, аналізує їх вплив на проект, визначає їх важливість та приймає рішення про їх упровадження з мінімальними втратами в часі та ресурсах. Крім того, керівник проекту сприяє ефективній комунікації всередині команди та із замовником, забезпечуючи зворотний зв'язок, вирішення конфліктів та підтримку співпраці між усіма сторонами проекту.

4. Етап тестування - виконується комплекс завдань, пов'язаних з тестуванням продукту. Керівник проекту розробляє план тестування, визначаючи обсяг робіт, розподіляє завдання між тестувальними спеціалістами та встановлює часові рамки для проведення тестування. Крім того, керівник забезпечує ефективне управління тестувальними процесами відповідно до стандартів якості, контролює виконання тестових планів та вирішує виявлені проблеми. Він координує роботу команди тестувальників, забезпечуючи їм необхідну підтримку та

ресурси, і сприяє взаємодії між командами розробки і тестування для вирішення проблем і поліпшення продукту. Крім того, керівник відповідає за забезпечення високої якості продукту після завершення тестування, вирішуючи питання виправлення помилок та покращення функціональності відповідно до вимог замовника та стандартів якості.

- 5. Реліз та підтримка продукту** - на останньому етапі керівник проекту виконує ряд важливих завдань. Він розробляє стратегію релізу, включаючи визначення функцій та термінів випуску, і координує роботу розробників та інших зацікавлених сторін для успішного релізу. Керівник керує процесом випуску продукту, забезпечуючи його належну імплементацію та розгортання. Він також забезпечує підтримку продукту після релізу, вирішуючи проблеми та помилки, які можуть виникнути, і надаючи підтримку користувачам. Керівник відстежує функціональність та взаємодію продукту з користувачами, збирає зворотній зв'язок і враховує пропозиції для поліпшень. Крім того, він розробляє стратегію майбутнього розвитку продукту, аналізуючи потреби ринку, зворотний зв'язок користувачів та технологічні тенденції, щоб визначити напрямки розширення та поліпшення продукту у майбутньому. Завдяки цим діям, керівник проекту забезпечує успішну реалізацію продукту та його тривалу підтримку, що сприяє досягненню бізнес-цілей та задоволенню потреб користувачів.

4.2. Опис методології

З розумінням того, які можуть бути етапи під час розробки продукту і які завдання має виконувати проектний менеджер на кожному з них - можна перейти до опису методології роботи керівника проектів з застосуванням ІІІ.

Метою розроблення методології полягає в покращенні ефективності та результативності управління проектами за допомогою використання

інноваційних технологій штучного інтелекту. Ця методологія спрямована на забезпечення оптимального використання ресурсів, підвищення якості проектних рішень та досягнення поставлених цілей з меншими зусиллями. Основні цілі методології включають:

1. Автоматизація процесів: Використання генеративного ШІ дозволяє автоматизувати рутинні та повторювані процеси управління проектами. Це дозволяє звільнити ресурси проектного менеджера від монотонних завдань та зосередитися на стратегічних аспектах проекту.
2. Аналіз даних та прийняття обґрунтованих рішень: Генеративне ШІ може аналізувати великі обсяги даних, виявляти залежності та патерни, що допомагає в прийнятті обґрунтованих рішень. Це сприяє зменшенню ризиків, покращенню прогнозування результатів проекту та вирішенню проблем на ранніх етапах.
3. Оптимізація використання ресурсів: Генеративне ШІ допомагає враховувати різні фактори та обмеження при розподілі ресурсів проекту. Це дозволяє ефективно використовувати людські, матеріальні та фінансові ресурси, мінімізуючи зайві витрати та підвищуючи продуктивність.

Ключові принципи методології роботи з генеративним штучним інтелектом - є аналіз даних минулих проектів, використання ШІ для виконання рутинних задач і надання максимальної кількості деталей у запиті до генеративних моделей.

Методологія передбачає збір та аналіз даних з попередніх проектів для підтримки роботи генеративного штучного інтелекту. Це дозволяє використовувати ці дані як основу для генерації відповідей та рекомендацій. Аналізуючи успішні та невдачі минулих проектів, можна виявити патерни,

навчитися на їх основі та покращити результати майбутніх проектів. Це можна використовувати, наприклад, для створення графіку виконання завдань проекту, оскільки маючи дані про попередні проекти, можна аналізувати, де часу було замало, чи, навпаки, виділено зайвий час, що можна було використати для інших цілей.

Під терміном “рутинні задачі” мається на увазі завдання, що не вимагають творчого підходу від керівника. Створення технічних завдань, документації, бізнес-планів та іншої документації - є стандартизованою і вся документація створюється після аналізу вимог, отриманих від зацікавлених сторін. Так само, генеративні моделі можуть аналізувати результати розмов з замовниками та членами командами, щоб створювати функціональні і нефункціональні вимоги до майбутнього продукту. Оскільки на цьому і базується принцип роботи штучного інтелекту - аналіз великих обсягів даних і пошук патернів в них.

І не менш важливим, є правильний підхід до використання генеративних моделей. Оскільки від кількості вказаної інформації у запиті до неї - пропорційно залежить якість отриманої відповіді. Бо штучний інтелект не створює нову унікальну інформацію - він видає результат на основі тих даних, на яких він навчався. Тому наша мета, звужити список можливих тем у відповіді від моделі.

Все наведене вище демонструє, що штучний інтелект не є головним виконавцем роботи в такому випадку - важливим аспектом для досягнення успіху все одно залишається індивідуальні навички та досвід керівника проектів. Оскільки від того, наскільки грамотно буде поставлено задачу для ШІ і правильність бачення проекту керівником буде прямо залежати кінцевий результат.

Універсальність даної методології заключається в тому, що її можна застосувати для кожного етапу життєвого циклу проекту. Той рівень на який

зараз вийшли можливості генеративного ШІ дозволяють виконувати усі задачі, з якими щодня стикається проектний менеджер. Можливості взаємодії обмежуються лише вмінням керівника проекту працювати з цим інструментом і використовувати його функціонал у повному обсязі. Це буде наочно проілюстровано у наступному розділі.

4.3. Висновки до розділу

У цьому розділі було запропоновано методологію застосування генеративного штучного інтелекту у роботі керівника проектів. Описаний підхід є універсальним і може застосовуватись на кожному з етапів життєвого циклу розробки продукту. Щоб впевнитись у цьому, слід провести експеримент із застосуванням цієї методології на практиці. Результат можна буде вважати успішним при виконанні усіх пунктів :

- буде продемонстровано можливість використання для кожного етапу життєвого циклу, що були описані у розділі 4.1. Для цього буде достатньо навести приклади декількох задач, які проектний менеджер може виконати із застосуванням генеративного ШІ.
- методологія має надавати перевагу у часу виконання роботи з застосуванням методології, у порівнянні з часом, який витрачається на виконання тих же задач без подібних інструментів.
- робота керівника проектів тісно пов'язана з співпрацею інших членів команди, тому необхідно впевнитися у тому, що отримані результати будуть високо оцінені іншими фахівцями. До прикладу, якщо технічне завдання розроблене з використанням методології буде перевірене експертом і надано схвальну оцінку, то цей пункт можна буде вважати виконаним.

5 ЕКСПЕРИМЕНТ ІЗ ЗАСТОСУВАННЯМ МОДЕЛІ СНАТ GPT

Для розглядання можливостей застосування генеративного ШІ візьмемо за основу запит від клієнтів на розробку інтернет-магазину з продажу товарів.

Розглянемо як генеративна модель може допомогти керівнику проекту на кожному етапі моделі життєвого циклу проекту:

5.1 Збір та аналіз вимог

На даному етапі найактивніше приймає участь саме керівник проекту, який відповідає за збір та обробку інформації від замовника, щоб в подальшому створити документи, що будуть основою для всіх наступних етапів. На самому початку проекту проходять перші розмови з замовником, де розробник збирає початкову інформацію про вимоги до продукту. Розвиток технології “speech-to-text” зараз дуже добре розвинуті, тому отримати скрипт розмови, що проходила в будь-якій з популярних нині додатків з онлайн-конференціями (Zoom, Google Meet) не є важкою задачею. Отримавши текстовий запис розмови з клієнтом — генеративна модель може швидко опрацювати значні масиви даних. ШІ може генерувати короткий опис розмови, сформулювати список основних вимог. Зробивши невеликий запит - *напиши основні вимоги до проекту, згідно розмови з клієнтом*, було отримано такий результат, зображений на рисунку 2:

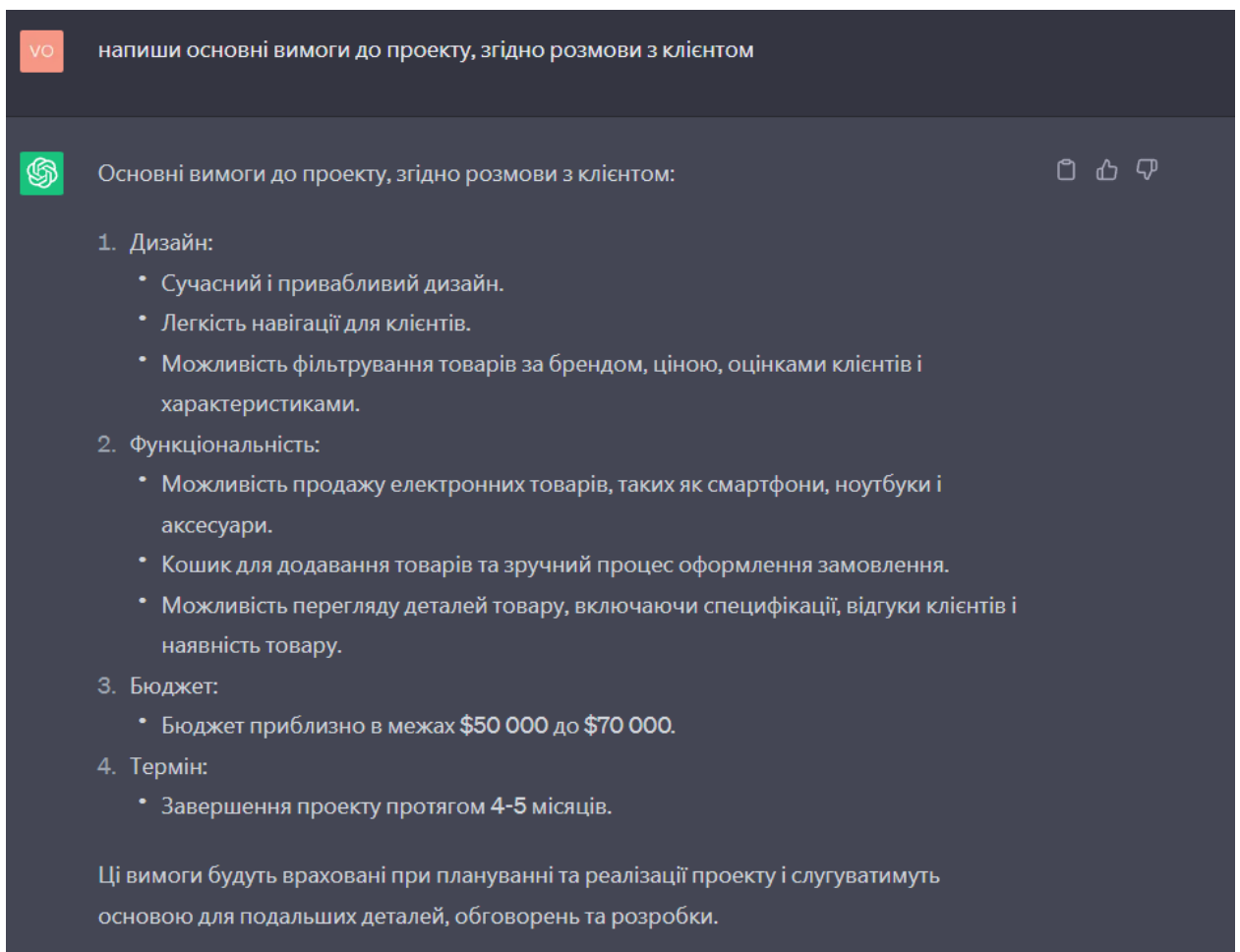


Рис. 2 - Результат виконання запиту

На основі сформованих вимог, можна переходити до планування робіт.

Сформуємо запит до моделі, в якому спробуємо отримати графік робіт та ключові задачі на кожному етапі, що слід виконати - *підготуй на основі вимог до проекту дані, де будуть розписані основні етапи розробки, ключові задачі, графік виконання робіт.*

Було отримано такі результати, зображені на рисунку 3:

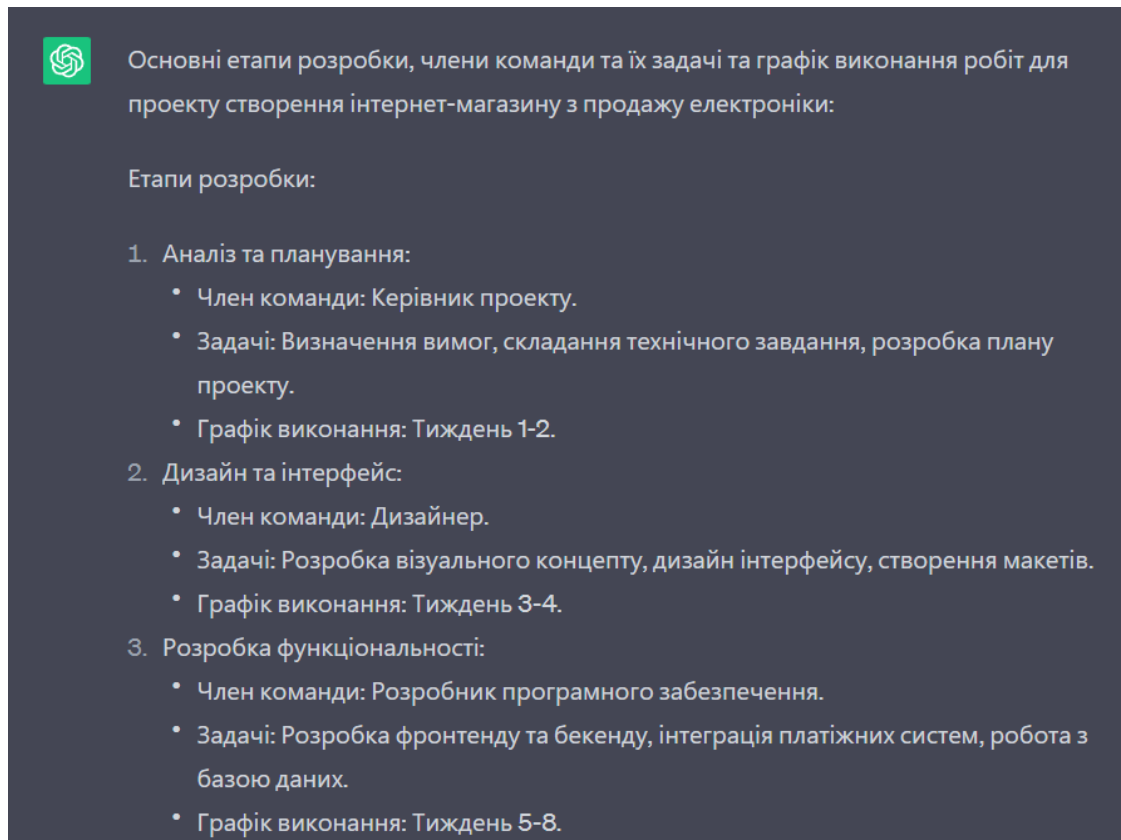


Рис. 3 - Результат виконання запиту

Що важливо зазначити, чим більше даних було надано моделі на вхід — тим якісніше і точніше результат вона може генерувати. Окрім даних, які можуть бути використані для навчання моделі, не менш важливим є правильність та кількість деталей у запиті, що подається до моделі. Модель ChatGPT має пам'ять, в межах відкритого чату, тому можна звертатись до попередніх запитів, уточнюючи їх, звертатись до них, та просити переписати відповідь, або врахувати дані з попередніх відповідей при відповіді на новий запит. До прикладу, якщо зробити запит з повідомленням - *розпиши детальніше календар виконання робіт, де детально по датам опиши виконання кожної з ключових задач*. То модель надасть більш детальний план (рис.4):

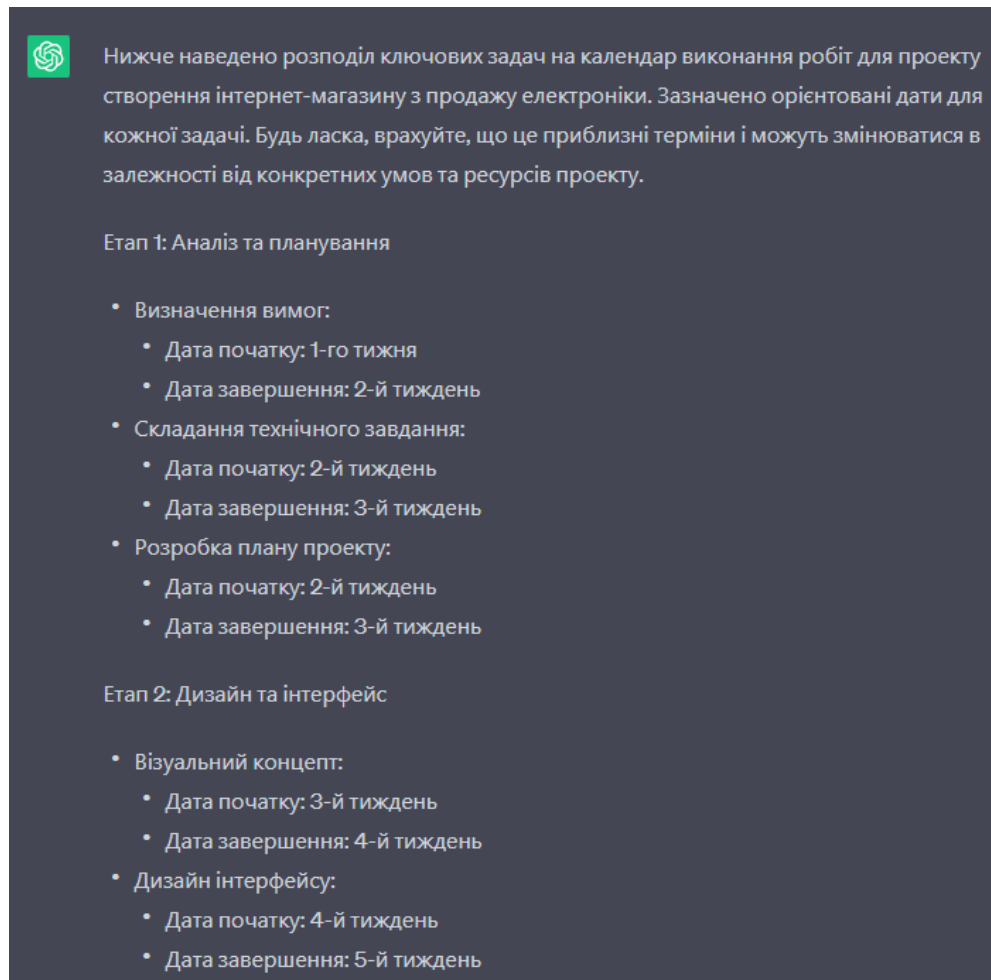


Рис. 4 - Результат виконання більш детального запиту

На даному етапі одним з важливих документів є “Специфікація вимог до програмного забезпечення” (Software Requirements Specification). Користуючись стандартом IEEE 830-1998 [13], в якому описані рекомендації щодо створення цього документу, розглянемо деякі основні розділи та створимо приклади запитів до генеративних моделей, що допоможуть у створенні SRS на прикладі проекту з розробки інтернет-магазину електроніки:

- **Розділ “Призначення”** - визначає продукт, вимоги до програмного забезпечення якого вказані в документі. Описується обсяг продукту, визначає загальну мету і цілі, який охоплюється цим SRS.

Приклад запиту - *опиши обсяг продукту, визнач загальну мету і цілі проекту з розробки інтернет-магазину продажу електроніки, згідно вимог від клієнта.*

Генеративна ШІ “Chat GPT” може надати такий результат (рис. 5):

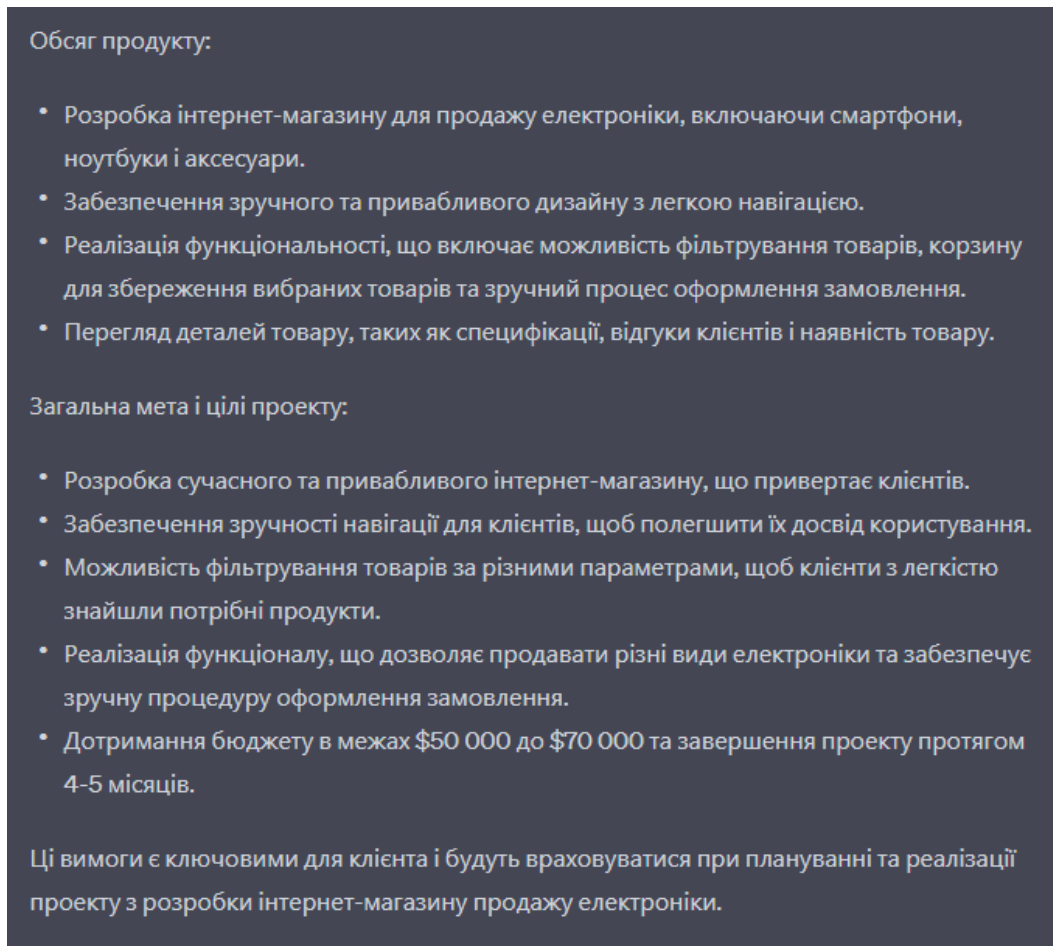


Рис. 5 - Результат виконання запиту

- **Розділ “Класи та характеристики користувачів”** - описує різні класи користувачів, які будуть використовувати продукт, що описано в SRS. Класи можуть відрізнятися за такими параметрами : частота використання продукту, рівень доступу, безпеки та іншими. Важливо також визначити основні цільові класи продукту, що будуть мати визначальний вплив.

Приклад запиту - *опиши різні класи користувачів, які будуть використовувати інтернет-магазину продажу електроніки, згідно вимог від клієнта.*

Генеративна ШІ “Chat GPT” може надати такий результат (рис. 6):

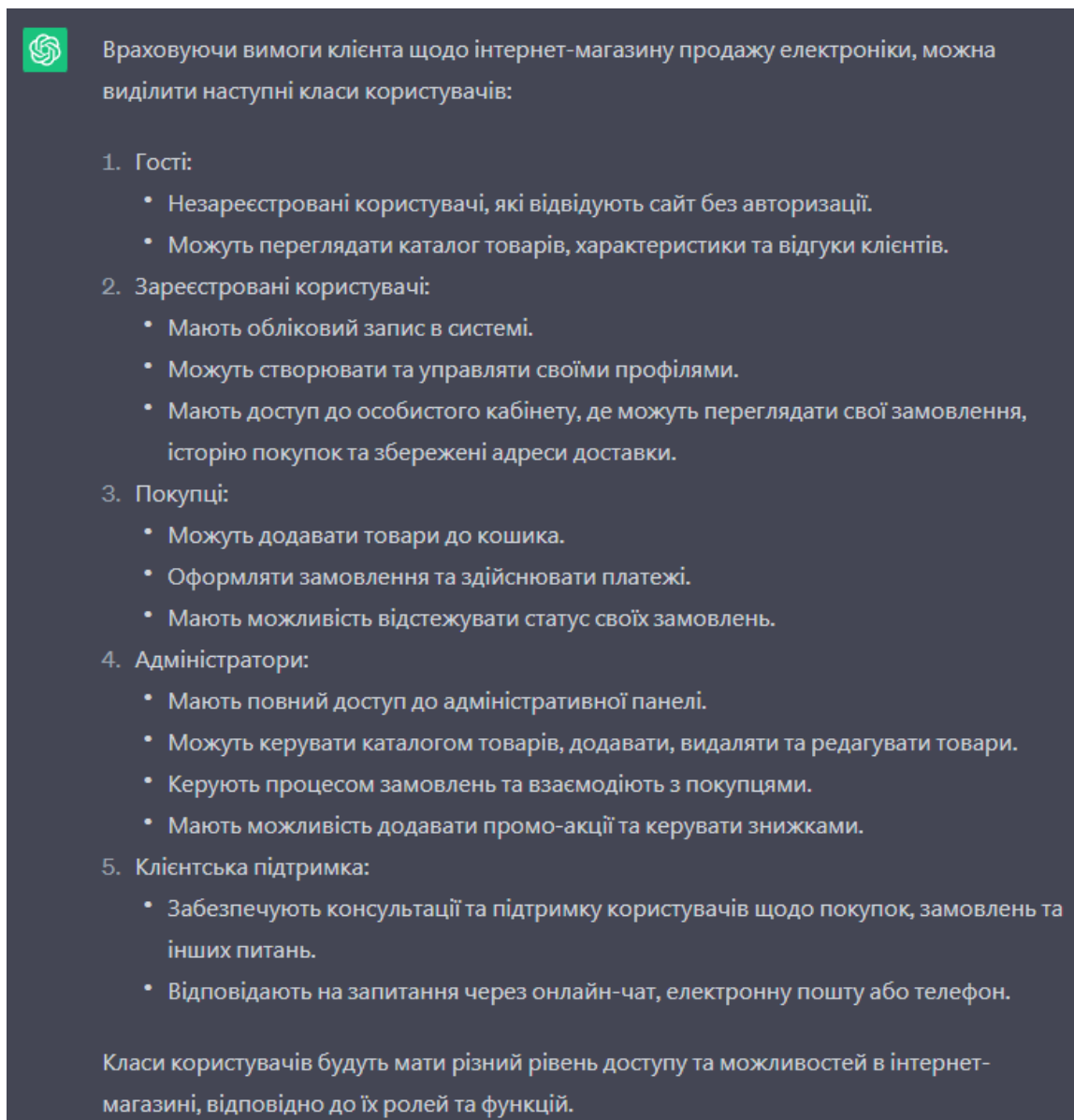


Рис. 6 - Результат виконання запиту

- **Розділ “Інтерфейси користувача”** - містить опис кожного інтерфейсу між програмним продуктом і користувачами. Це можуть бути зразки екранних зображень, будь-які стандарти графічного інтерфейсу або посібники щодо стилю продукту, які слід дотримуватися тощо. В розділі описуються основні програмні компоненти, для яких потрібен користувацький інтерфейс.

Приклад запиту - *опиши основні інтерфейси додатку інтернет-магазину між програмним продуктом і користувачами.*

Генеративна ШІ “Chat GPT” може надати такий результат (рис. 7):

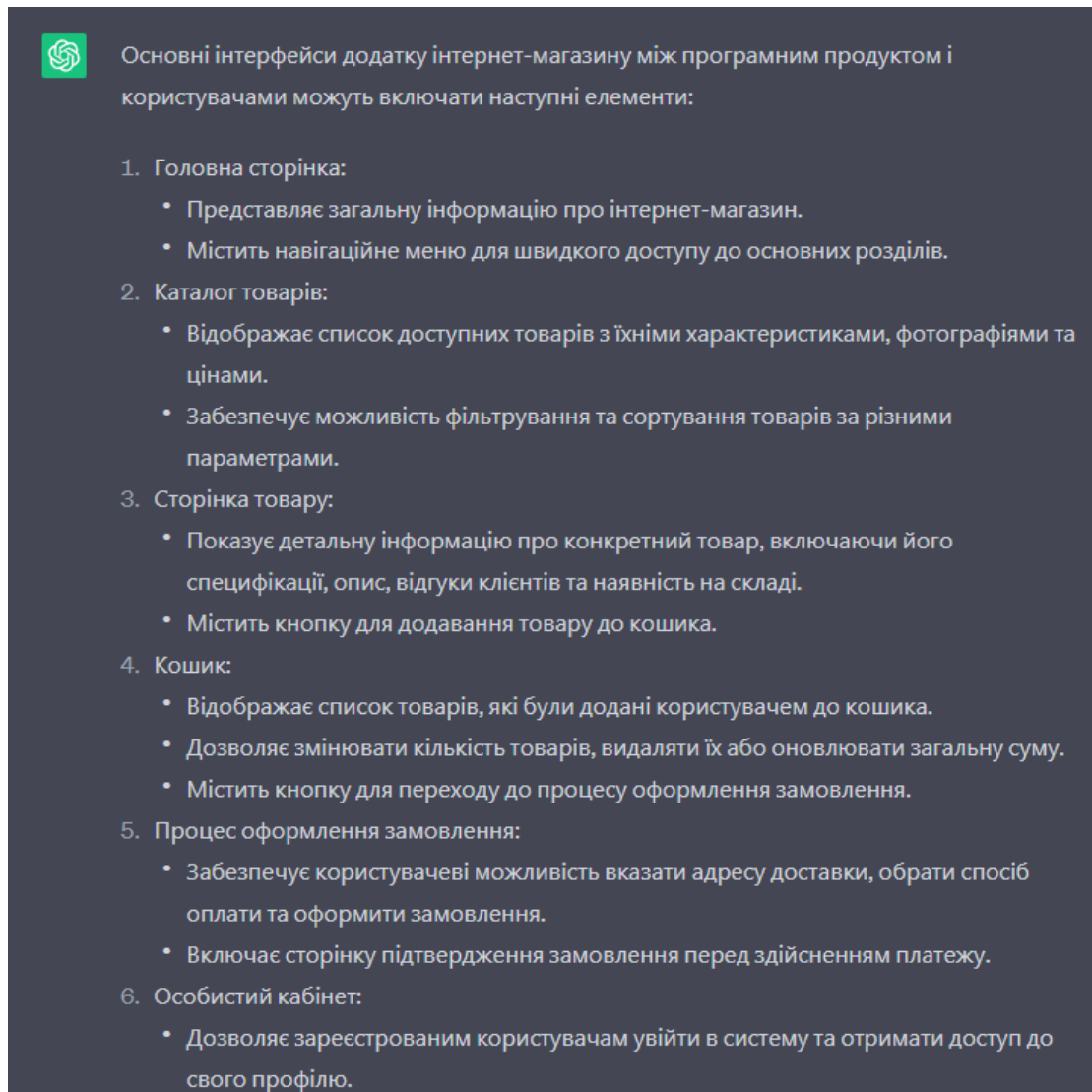


Рис. 7 - Результат виконання запиту

- **Розділ “Особливості системи”** - містить детальний опис роботи та функціональних вимог продукту. Зазвичай описуються з використанням класів користувача, класів об’єкту, функціональної ієрархії тощо.

Приклад запиту - *надай опис роботи та функціональних вимог додатку інтернет-магазину, з використанням класів користувача. Кожен елемент*

описувати з такими елементами - *Опис і пріоритет, функціональні вимоги, Послідовності відповідей.*

Генеративна ШІ “Chat GPT” може надати такий результат (рис. 8):

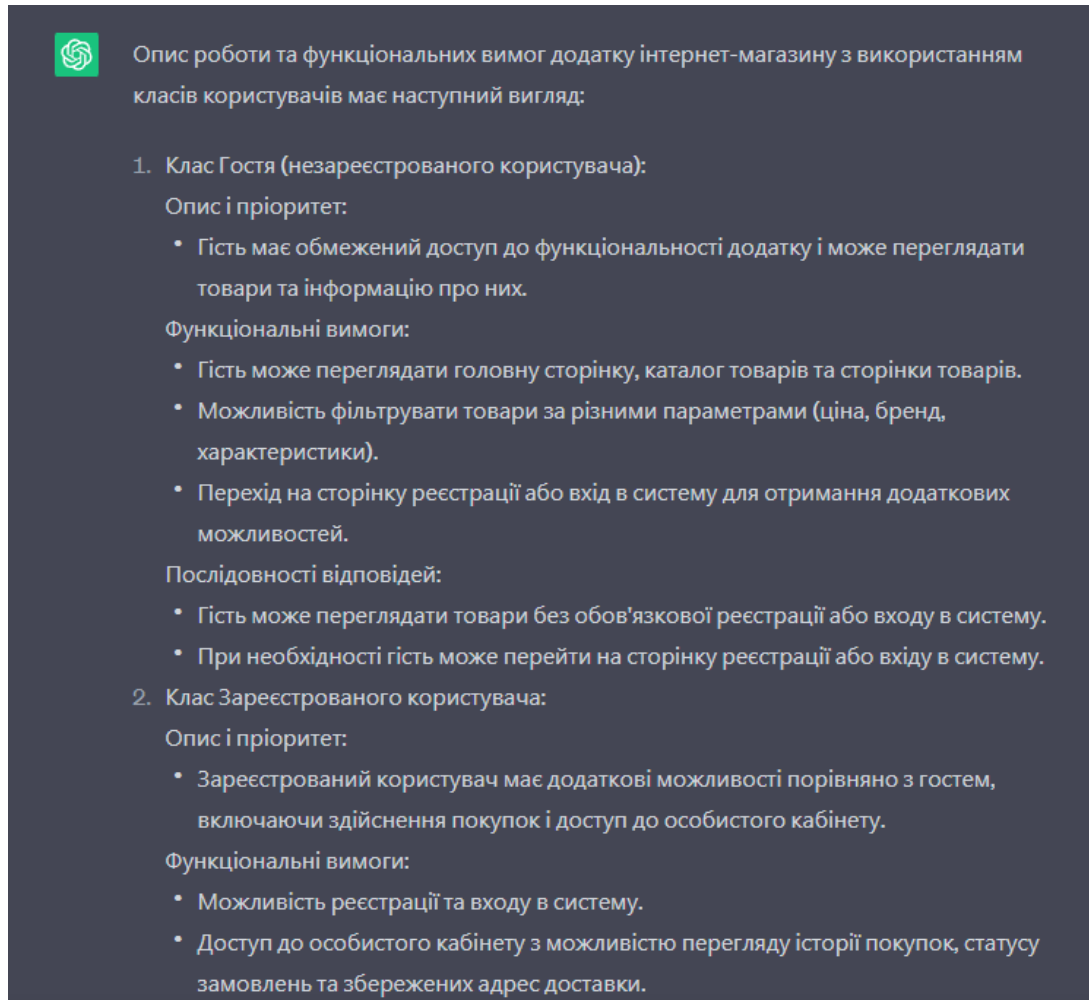


Рис. 8 - Результат виконання запиту

- **Розділ “Вимоги до безпеки”** - містить вимоги щодо питань безпеки або конфіденційності, пов'язані з використанням продукту або захистом даних, що використовуються або створені продуктом. Описуються вимоги до механізму автентифікації користувача, сертифікатів безпеки або конфіденційності.

Приклад запиту - *надай опис вимог щодо питань безпеки або конфіденційності, пов'язані з використанням продукту або захистом даних, що*

використовуються або створені продуктом. Опиши вимоги до механізму автентифікації користувача, сертифікатів безпеки або конфіденційності.

Генеративна ШІ “Chat GPT” може надати такий результат (рис. 9):

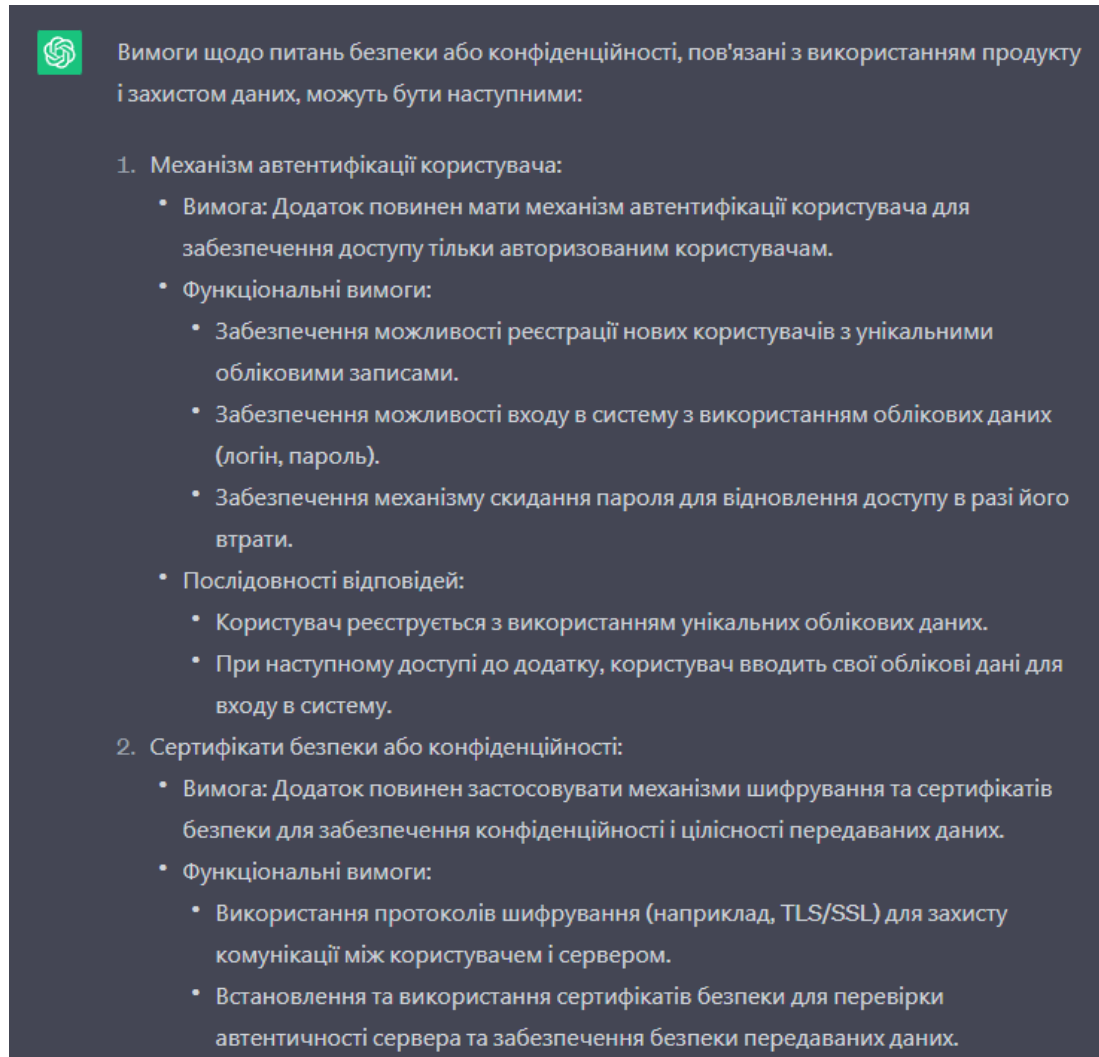


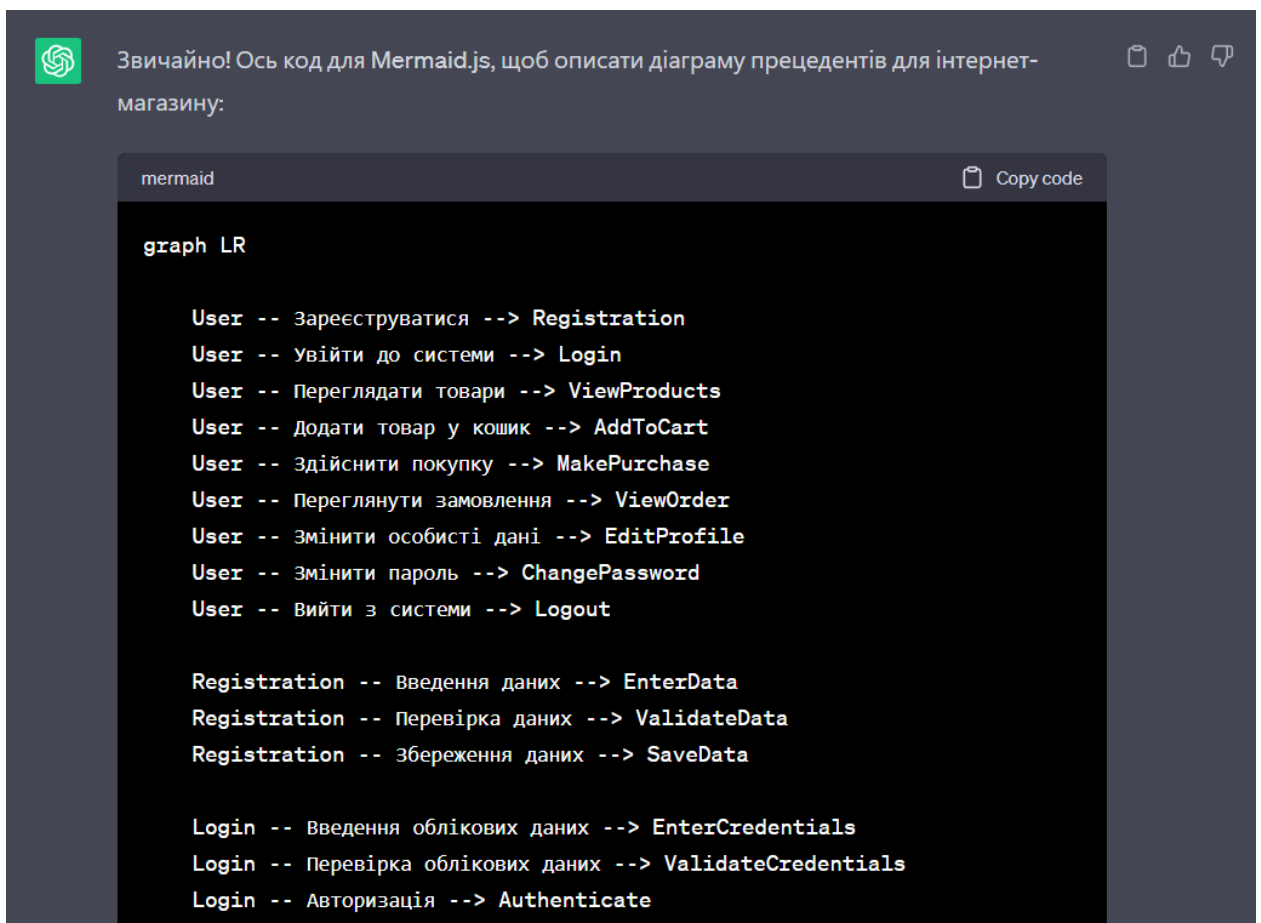
Рис. 9 - Результат виконання запиту

5.2 Проектування та створення дизайну

На другому етапі до керівника проектів приєднуються інші члени команди - розробники і дизайнери, щоб описати майбутню архітектуру проекту, дизайн та інші задачі. Задачею проектного менеджера є допомога у виконанні цих задач. Наприклад, підготовка різних UML (Unified Modeling Language -

Уніфікована Мова Моделювання) діаграм, таких як - Діаграма прецедентів (Use Case Diagrams), Діаграма послідовності (Sequence diagram), Діаграма класів (Class diagram) та інших. Відразу слід зазначити, що Chat GPT є лінгвістичною моделлю, тому генерувати зображення вона не може. Але для роботи з діаграми можна використовувати ще один додатковий застосунок, який приймає текстовий код діаграми, де описані її елементи та залежності - і генерує зображення на основі цього. Одним з прикладів таких застосунків є Mermaid.js - інструмент для створення діаграм на мові програмування JavaScript. Chat GPT може генерувати код, що використовується в синтаксисі цього додатку, тому повернемося до прикладу з інтернет-магазином по продажу електроніки, і спробуємо отримати діаграми.

Зробимо запит з наступним текстом - *згенеруй код для Mermaid.js, в якому опиши діаграму прецедентів для інтернет-магазину, вимоги до якого було описано в попередніх запитах*. Спочатку модель має надати код, який потім слід використати у застосунку, щоб отримати вже готову UML діаграму прецедентів. Від Chat GPT було отримано код діаграми (рис. 10):



Звичайно! Ось код для Mermaid.js, щоб описати діаграму прецедентів для інтернет-магазину:

```

graph LR
    User -- Зареєструватися --> Registration
    User -- Увійти до системи --> Login
    User -- Переглядати товари --> ViewProducts
    User -- Додати товар у кошик --> AddToCart
    User -- Здійснити покупку --> MakePurchase
    User -- Переглянути замовлення --> ViewOrder
    User -- Змінити особисті дані --> EditProfile
    User -- Змінити пароль --> ChangePassword
    User -- Вийти з системи --> Logout

    Registration -- Введення даних --> EnterData
    Registration -- Перевірка даних --> ValidateData
    Registration -- Збереження даних --> SaveData

    Login -- Введення облікових даних --> EnterCredentials
    Login -- Перевірка облікових даних --> ValidateCredentials
    Login -- Авторизація --> Authenticate
  
```

Рис. 10 - Результат виконання запиту

Тепер скористаємось ним на веб-сайті [Mermaid.js](https://mermaid.js.org/) і поглянемо на отриману діаграму (рис. 11):

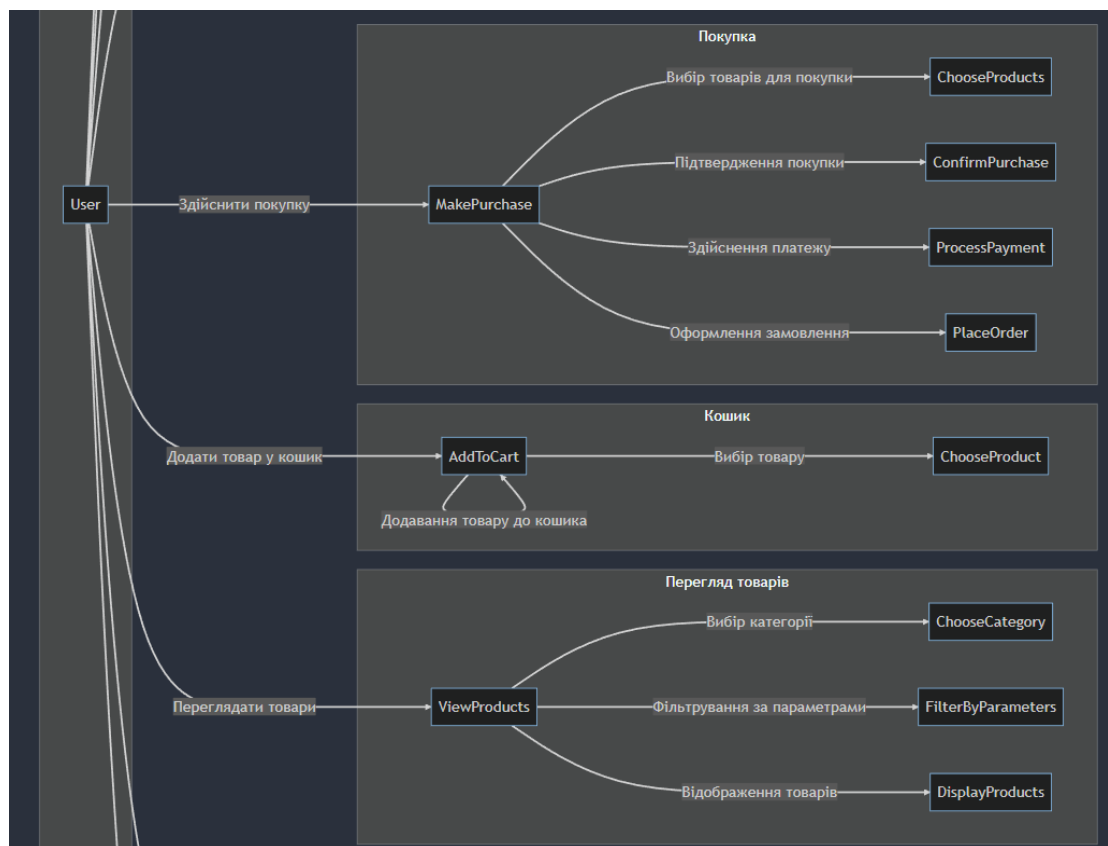
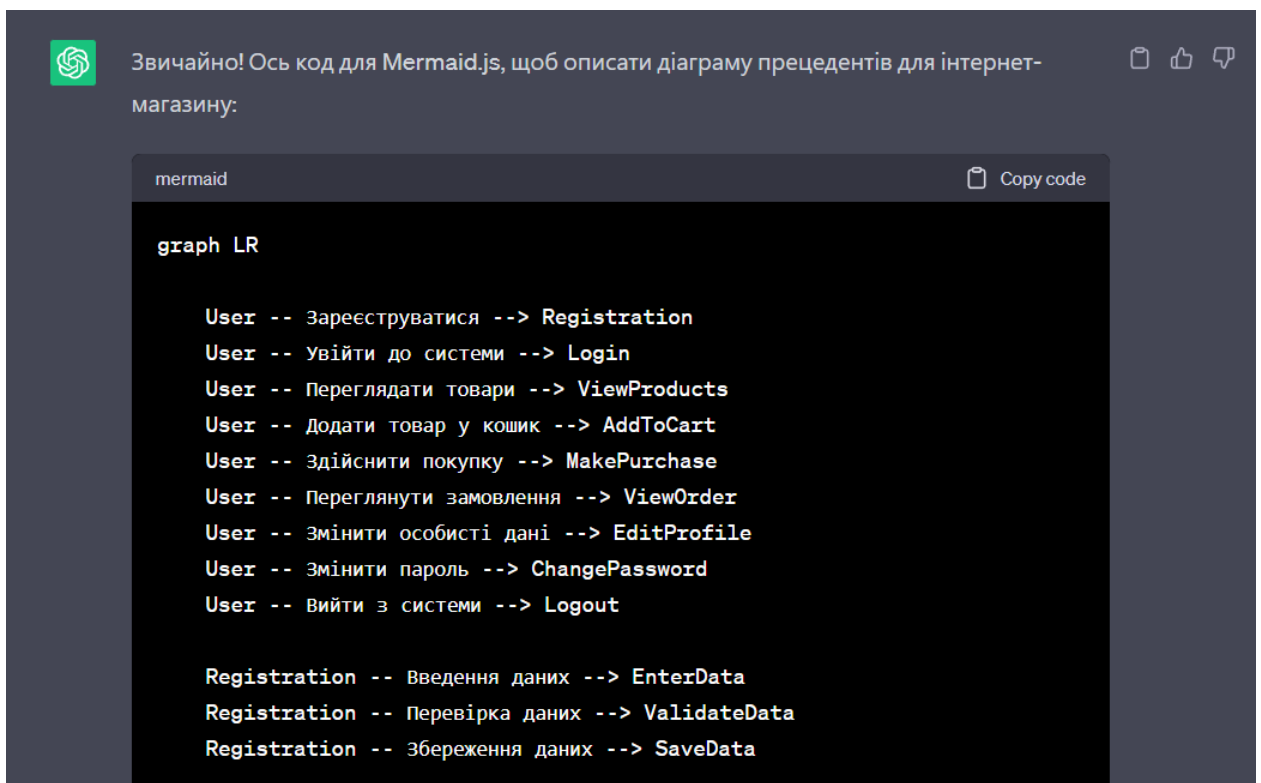


Рис. 11 - Фрагмент отриманої діаграми

Отриманий результат є доволі непоганим, і що головне, його легко відредагувати, щоб внести корективи. Це є важливим моментом, на який слід звертати увагу, під час роботи з моделями, бо не варто очікувати, що вони будуть генерувати повністю правильний результат. Найближчим часом, рівень розвитку моделей не досягне рівня інтелекту людини, тому подібні інструменти завжди мають підлягати перевірці та коригування помилок.

Розглянемо інший приклад - діаграма послідовностей. Повторимо такі самі кроки - попросимо модель згенерувати код діаграми, і за допомогою застосунку Mermaid.js отримаємо готову діаграму. Для того, щоб отримати код діаграми послідовності, до Chat GPT можна зробити наступний запит - *згенеруй код для Mermaid.js, в якому опиши діаграму прецедентів для інтернет-магазину, вимоги якого було описано в попередніх запитах*. Отримуємо код на необхідному синтаксисі (рис. 12):



The screenshot shows a chat window with a dark theme. At the top left is a green circular logo with a white knot-like symbol. To its right is the text "Звичайно! Ось код для Mermaid.js, щоб описати діаграму прецедентів для інтернет-магазину:". On the far right of the chat header are three icons: a document, a thumbs up, and a speech bubble. Below the header is a code editor area with a dark background. The title bar of the editor says "mermaid" and has a "Copy code" button on the right. The code is written in a light green monospace font and defines a Use Case Diagram (graph LR) for an e-commerce system. It lists actors (User, Registration) and their associated use cases (e.g., Зареєструватися, Увійти до системи, Переглядати товари, etc.) connected by directed edges.

```
graph LR
    User -- Зареєструватися --> Registration
    User -- Увійти до системи --> Login
    User -- Переглядати товари --> ViewProducts
    User -- Додати товар у кошик --> AddToCart
    User -- Здійснити покупку --> MakePurchase
    User -- Переглянути замовлення --> ViewOrder
    User -- Змінити особисті дані --> EditProfile
    User -- Змінити пароль --> ChangePassword
    User -- Вийти з системи --> Logout

    Registration -- Введення даних --> EnterData
    Registration -- Перевірка даних --> ValidateData
    Registration -- Збереження даних --> SaveData
```

Рис. 12 - Результат виконання запиту

Поглянемо на отриману діаграму, фрагмент якої зображено на рисунку 13:

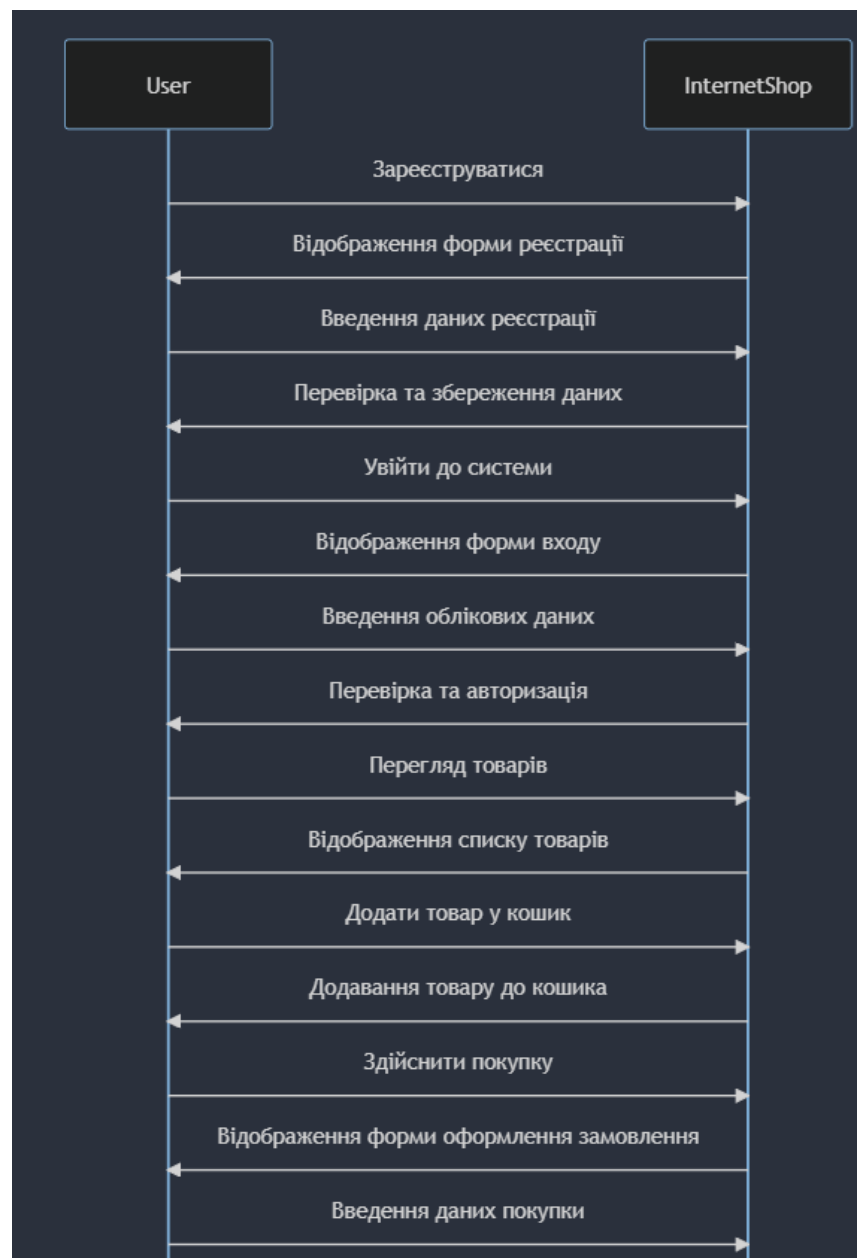
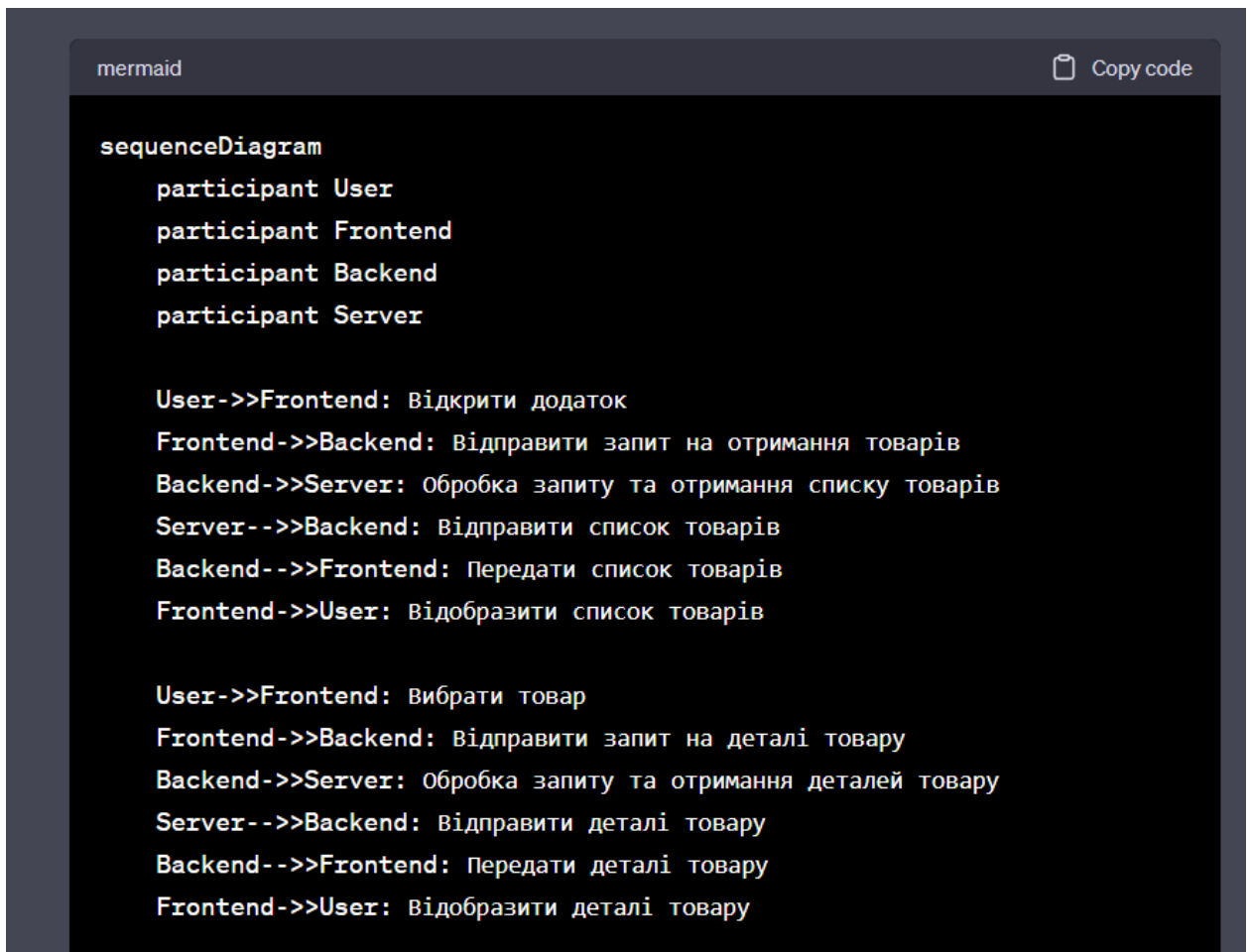


Рис. 13 - Фрагмент отриманої діаграми

Як бачимо, результат вийшов незадовільним, оскільки містить мало елементів, тому доповнимо запит, вказавши необхідність в діаграмі з поділом магазину на 3 основні компоненти будь-якого веб-застосунку - frontend (інтерфейсна частина, з якою користувач взаємодіє), backend(закрита частина, доступ до якої користувач немає, описує логіку роботи застосунку) та серверна частина. Запит буде виглядати наступним чином - *згенеруй код для Mermaid.js, в*

якому опиши діаграму послідовностей (Sequence diagram) для інтернет-магазину, вимоги якого було описано в попередніх запитах, з розрізненням на фронтенд, бекенд та серверну частину. І тепер поглянемо на результат і порівняємо з попереднім. В коді, згенерованого ШІ, видно, що вже використано 4 елементи на діаграмі, які ми заявили у запиті (рис. 14):



```
sequenceDiagram
    participant User
    participant Frontend
    participant Backend
    participant Server

    User->>Frontend: Відкрити додаток
    Frontend->>Backend: Відправити запит на отримання товарів
    Backend->>Server: Обробка запиту та отримання списку товарів
    Server-->>Backend: Відправити список товарів
    Backend-->>Frontend: Передати список товарів
    Frontend->>User: Відобразити список товарів

    User->>Frontend: Вибрати товар
    Frontend->>Backend: Відправити запит на деталі товару
    Backend->>Server: Обробка запиту та отримання деталей товару
    Server-->>Backend: Відправити деталі товару
    Backend-->>Frontend: Передати деталі товару
    Frontend->>User: Відобразити деталі товару
```

Рис. 14 - Результат виконання запиту

І отримана діаграма послідовності, при додаванні більшої кількості деталей до запиту, вже є більш вдалою (рис. 15):



рис. 15 - Фрагмент отриманої діаграми з детальнішим запитом

Щоб закріпити впевненість у тому, що генеративні ШІ можуть бути корисні в створенні діаграм, розглянемо останній приклад - створимо діаграму класів. Для цього формуємо запит - *згенеруй код для Mermaid.js, в якому опиши діаграму класів для інтернет-магазину, вимоги якого було описано в попередніх запитах*.

Генеративна модель сформувала код діаграми (рис. 16), з допомогою якого застосунок Mermaid.js може сформувати кінцеву діаграму:

Звичайно! Ось код для Mermaid.js, щоб описати діаграму класів (Class diagram) для інтернет-магазину, враховуючи вимоги, що були описані раніше:

```

classDiagram
    class User {
        - userId: int
        - username: string
        - password: string
        + login(): void
        + logout(): void
        + browseProducts(): void
        + viewProductDetails(productId: int): void
        + addToCart(productId: int): void
        + checkout(): void
    }

    class Product {
        - productId: int
        - name: string
        - price: float
        - brand: string
        - description: string
    }

```

Рис. 16 - Результат виконання запиту

І після того, як ми скористалось кодом від генеративного ШІ - поглянемо на отриману діаграму класів (рис. 17):

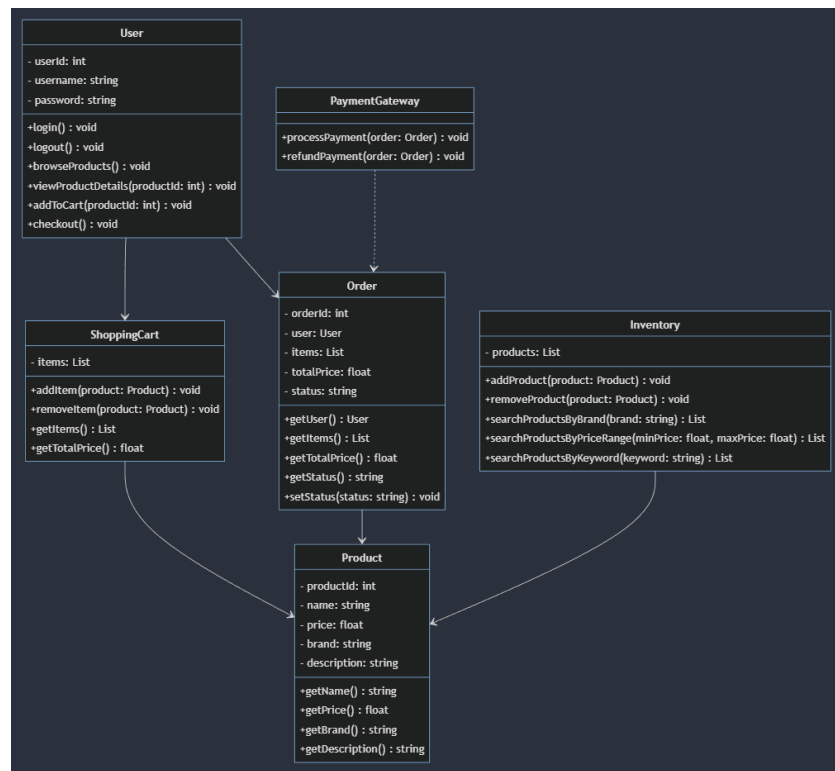


Рис. 17 - Отримана діаграма класів

Отримана діаграма, навіть при тому факту що запит містив невелику кількість деталей, є якісною, містить різні типи зв'язків, правильно описані поля кожного класу. Але, що важливо ще раз зазначити, подібний підхід, що описано в цьому розділі має важливу перевагу - можливість легко внести зміни у отримані діаграми. Це можна зробити двома способами : 1. Змінити запит до генеративної моделі, доповнивши запит деталями. 2. Відредагувати власноруч код діаграми, оскільки синтаксис, що використовує застосунок легкий та зрозумілий. Це важливий нюанс в контексті розгляду подібних інструментів, оскільки генеративні моделі не мають перспектив повністю замінити людину у роботі, а може бути лише корисним інструментом в руках спеціаліста, що дозволить економити час і ресурси.

5.3 Етап розробки продукту

На етапі розробки продукту перед командою стоїть задача розробити робочий продукт, згідно тої документації, що була описана на попередніх етапах. Відповідно, розробники мають створювати код застосунку, а задача керівника проекту зробити так, щоб цю задачу полегшити, і контролювати виконання задач згідно затвердженого плану. Тому, пропонується розглянути можливість використання генеративного ШІ для допомоги у роботі менеджера проектів з наступними завданнями - створення технічних завдань для розробників, оцінка ризиків та попередження виникнення проблем на проекті і вирішення питань з застосуванням інструментів прийняття рішень.

Для створення технічного завдання з використанням Chat GPT необхідно звернутись до моделі з наступним запитом - *напиши технічне завдання для розробника для реалізації функції аутентифікації користувача в інтернет-магазині електроніки*. Поглянемо на результат, отриманий від моделі (рис. 18):

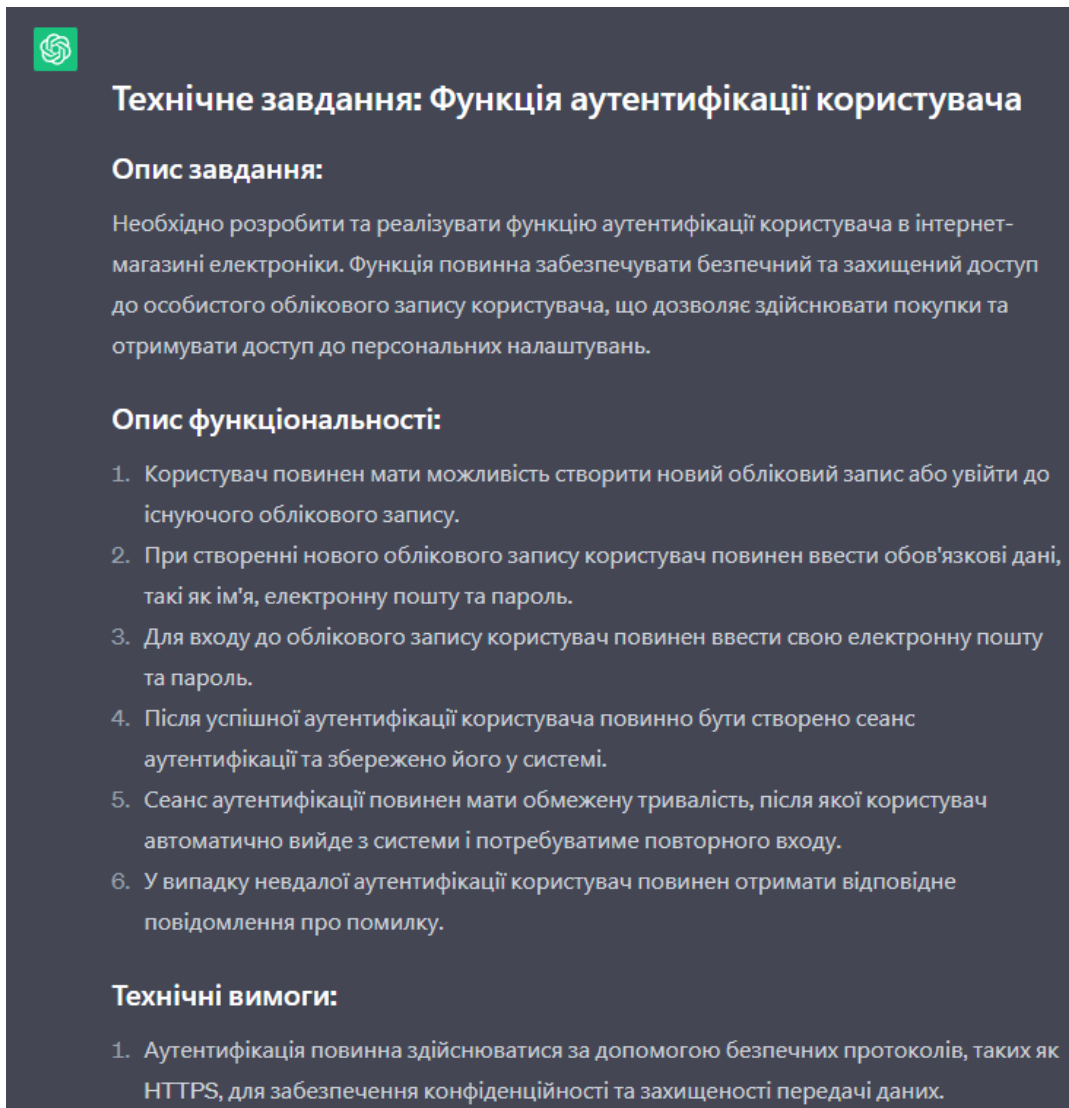


Рис. 18 - Результат виконання запиту

Проаналізувавши отриманий результат, робимо висновок, що з цією задачею модель впоралась, і технічне завдання після перевірки та виправлення помилок, що звісно можуть бути, то завдання можна передавати в роботу.

Приклад запиту можна легко модифікувати, якщо додати деякі деталі:

- описати важливі вимоги до завдання, які модель сама не описала під час першого запиту (до прикладу, напиши технічне завдання для розробника для реалізації функції аутентифікації користувача в інтернет-магазині електроніки. Врахуй, що під час авторизації

користувач має проходити перевірку на “ReCaptcha”, і захист від хакерських атак)

- вказати рівень обізнаності розробника - у менш досвідченого розробника може виникати більше питань до технічного завдання, тому модель може згенерувати завдання з більшою кількістю деталей та нюансів (до прикладу, напиши технічне завдання для розробника з малим стажем роботи для реалізації функції аутентифікації користувача в інтернет-магазині електроніки)
- також модель має пам'ять щодо попередніх запитів, тому користувач ІІІ може просити переписати минулий запит (приклад запиту - перепиши технічне завдання), виправити чи переписати конкретну частину попередньої відповіді (приклад запиту - перепиши розділ “Послідовність взаємодії” з урахуванням проходження перевірки “ReCaptcha”).

Розглянемо іншу задачу, з якою генеративна модель може допомогти - оцінка ризиків та їх запобігання. Існує окремий напрямок управління проєктів - ризик-менеджмент, задача якого полягає в управлінні ризиками, має ряд стратегій та методологій, які ставлять за мету уникнути провалу у виконанні проєкту. Почнемо з того, що спробуємо отримати від генеративного ІІІ таблицю можливих ризиків, що можуть виникнути під час розробки інтернет-магазину. Сформуємо запит - *створи список ризиків, що можуть виникнути під час розробки інтернет-магазину у вигляді таблиці*. Тепер поглянемо на отриманий результат (рис. 19):



№	Ризик	Опис	Вплив	Ймовірність	Рівень
1	Недостатня аналіз вимог клієнта	Недостатнє розуміння вимог клієнта може призвести до невідповідностей у реалізації	Високий	Середній	Високий
2	Затримки у розробці	Технічні складності, недоцільна організація роботи та інші фактори можуть спричинити затримки	Високий	Високий	Високий
3	Проблеми з безпекою та захистом даних	Недостатня захищеність системи може призвести до витоку даних або злому системи	Високий	Середній	Високий
4	Несумісність з різними платформами	Різноманітність платформ та пристроїв може створити проблеми зі сумісністю та відображенням інтерфейсу	Середній	Середній	Середній

Рис. 19 - Результат виконання запиту

Без вказання чітких деталей, модель змогла сформувати таблицю ризиків, з коректними стовпцями. В тексті присутні помилки у правописі, оскільки модель Chat GPT орієнтована на англійську мову, тому помилки в правописі українською зрозумілі, що легко виправляється, тому результат можна вважати задовільним, оскільки витративши мінімум часу - було отримано таблицю ризиків, що спростить роботу керівнику проектів.

При роботі з менеджментом ризиків важливо не лише, описати можливі, але й одразу підготувати план дій, у разі виникнення індикаторів, одного з ризиків. Тому спробуємо отримати такий список дій для декількох ризиків. Формуємо запит - *створи на кожен наведений ризик алгоритм дій, і як помітити цю проблему під час роботи на проекті. Дані сформууй у табличній формі - Затримки у розробці, Проблеми з безпекою та захистом даних,*

Нестача фінансів, Вихід проекту за рамки встановленого обсягу. Тепер оцінімо результат роботи ШІ (рис. 20):



Ризик	Алгоритм дій	Індикатори проблеми
Затримки у розробці	1. Визначити причини затримок	- Не виконання планованих термінів
	2. Перерозподілити завдання та ресурси	- Відсутність проміжних результатів
	3. Оцінити нові терміни завершення проекту	- Поява незапланованих проблем або завдань
	4. Забезпечити ефективну комунікацію в команді	- Збільшення кількості запитів та звернень до розробників
	5. Стежити за виконанням нового графіку	
Проблеми з безпекою та захистом даних	1. Визначити потенційні загрози та вразливості	- Виявлення несанкціонованого доступу до системи
	2. Розробити механізми захисту даних	- Поява підозрілих або аномальних активностей в системі
	3. Встановити контроль доступу до інформації	- Відсутність адекватних заходів безпеки
	4. Забезпечити регулярне оновлення програмного забезпечення	

Рис. 20 - Результат виконання запиту

Цього разу результат вийшов більш вдалий, ніж попередній, він містить більш детальну та корисну інформацію. Це пов'язано, як ми переконались раніше, з кількістю деталей в запиті, бо цього разу ми вказали які ризики слід розглянути, і описали яку інформацію про ризики ми хочемо отримати.

Можна зробити висновок, що застосування Chat GPT для роботи з ризик-менеджментом є досить ефективним, при належному користуванні даним

інструментом, керівник проекту може значно зекономити у часі, який потім можна витратити на більш нагальні, і можливо, більш цікавим для нього задачам.

5.4 Тестування

На даному етапі розроблений продукт необхідно перевірити на наявність помилок і неточностей у роботі, і чи відповідає він затвердженим вимогам. Основну роботу виконують окремі члени команди - тестувальники. Керівник проекту має допомагати тестувальникам виконувати свою роботу, надаючи тестові сценарії, в яких описано послідовність взаємодії користувача з продуктом. Тестувальник має перевірити, щоб всі дії описані в сценарії виконувались в розробленому продукту.

Поглянемо, як Chat GPT може допомогти зі створенням тестових сценаріїв. Для цього сформуємо запит - *згенеруй тестові сценаріїв для інтернет-магазину, вимоги до якого було описано в попередніх запитах*. Модель сформувала декілька сценаріїв, поглянемо на їх структуру та якість (рис. 21):

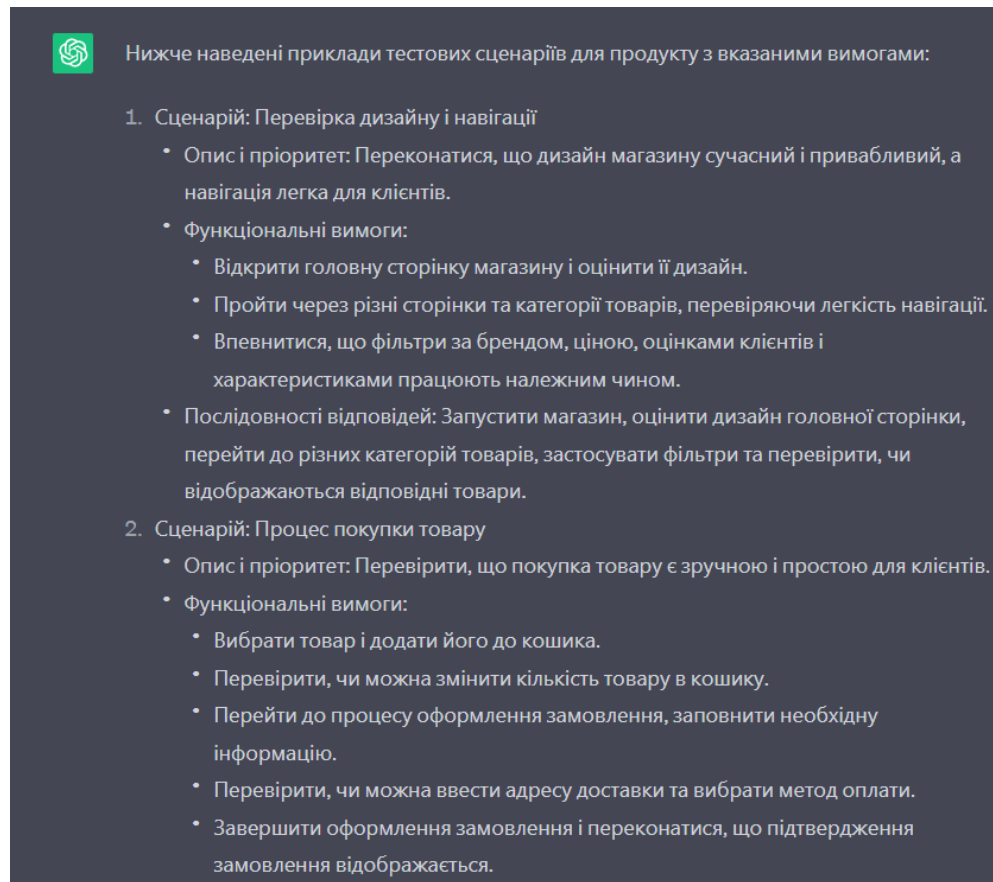


Рис. 21 - Результат виконання запиту

Без вказання додаткових деталей у запиті, модель описала сценарії одразу у задовільному форматі - назва сценарію, загальний опис і ціль, перелік вимог і порядок дій. До змісту сценаріїв зауважень немає. Отже можна зробити висновок, що для створення тестових сценаріїв генеративні моделі можна застосовувати.

Розглянемо також, можливість застосування генеративного ШІ для прийняття рішень. Робота над проектом рідко коли відбувається без проблем і ситуацій, коли керівникам доводиться приймати рішення для їх вирішення. Тому, візьмемо доволі типову ситуацію, коли під час тестування продукту знаходять багато помилок, і для їх вирішення не вистачає часу, прописаного в календарі виконання робіт. Спробуємо отримати алгоритм дій в такій ситуації від моделі Chat GPT, зазначивши що слід застосувати принцип Парето. Запит

буде виглядати наступним чином - використовуючи принцип Парето, сформулюй порядок дій для ситуації - під час тестування продукту було знайдено помилки в роботі, час для виконання яких перевищує встановлені терміни. Тепер поглянемо на відповідь на даний запит (рис. 22):

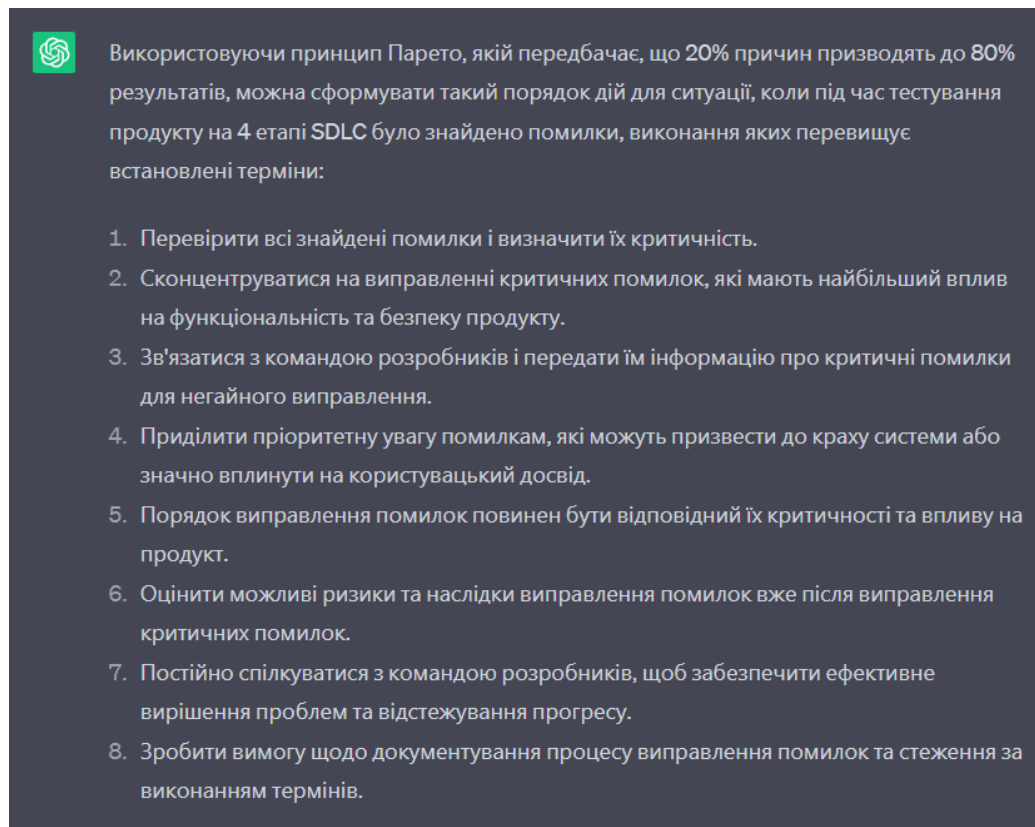


Рис. 22 - Результат виконання запиту

Як бачимо, модель вірно описала принцип Парето, про який було описано в запиті, і надала задовільний результат. Оскільки отриманий порядок дій є логічним та орієнтованим на результат. Тому можна зробити висновок, що і для прийняття рішень, генеративну модель можна розглядати як один із інструментів, що полегшить роботу.

5.5 Реліз та підтримка продукту

Останній етап життєвого циклу продукту - випуск розробленого продукту для користування користувачів та його підтримка, з подальшими оновленнями і

корективами, на основі отриманих відгуків. Розглянемо приклади задач, з якими генеративні моделі можуть допомагати керівникам проектів.

Модель може бути використана для створення готових відповідей на часті питання від користувачів (FAQ). Такі питання розміщуються на сайті, щоб зменшити навантаження на відділ підтримки користувачів. Сформуємо запит до моделі - *підготуй FAQ для інтернет-магазину з продажу електроніки, підготуй відповіді на найчастіші питання, що виникають у користувачів на подібних сайтах*. Поглянемо на результат роботи (рис. 23) :

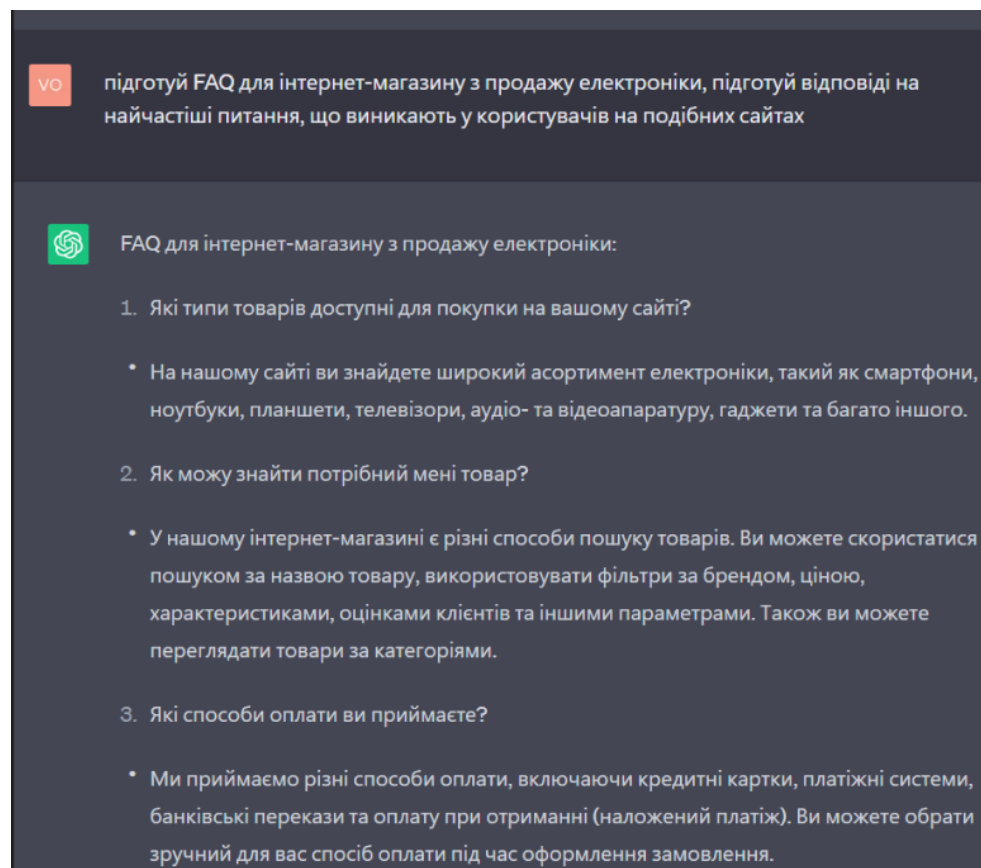


Рис. 23 - Результат виконання запиту

Модель згенерувала вісім питань та відповідей на них, які є дійсно популярними серед користувачів, що вперше відкрили застосунок, і не знайомі з ним. Отже, отриманий результат є задовільним, і можна вважати, що Chat GPT буде корисний в подальшому при виконанні подібних завдань.

Крім наведеного прикладу, модель також може бути використана для аналізу відгуків користувачів про продукт. Вона може допомогти проектному менеджеру зрозуміти загальні тенденції, проблеми або побажання користувачів шляхом аналізу тексту відгуків. Це може допомогти вдосконалити продукт, виявити недоліки та забезпечити задоволення потреб користувачів. Для прикладу, підготуємо кілька відгуків про розроблений продукт, і поставимо запит генеративний ШІ проаналізувати їх і видати висновок, де буде описано основні зауваження до системи. Запит виглядає таким чином - *проаналізуй 3 відгуки користувачів, і сформуль короткий висновок по ним [відгук 1] [відгук 2] [відгук 3]*. І поглянемо як модель справилась з даною задачею (рис. 23):

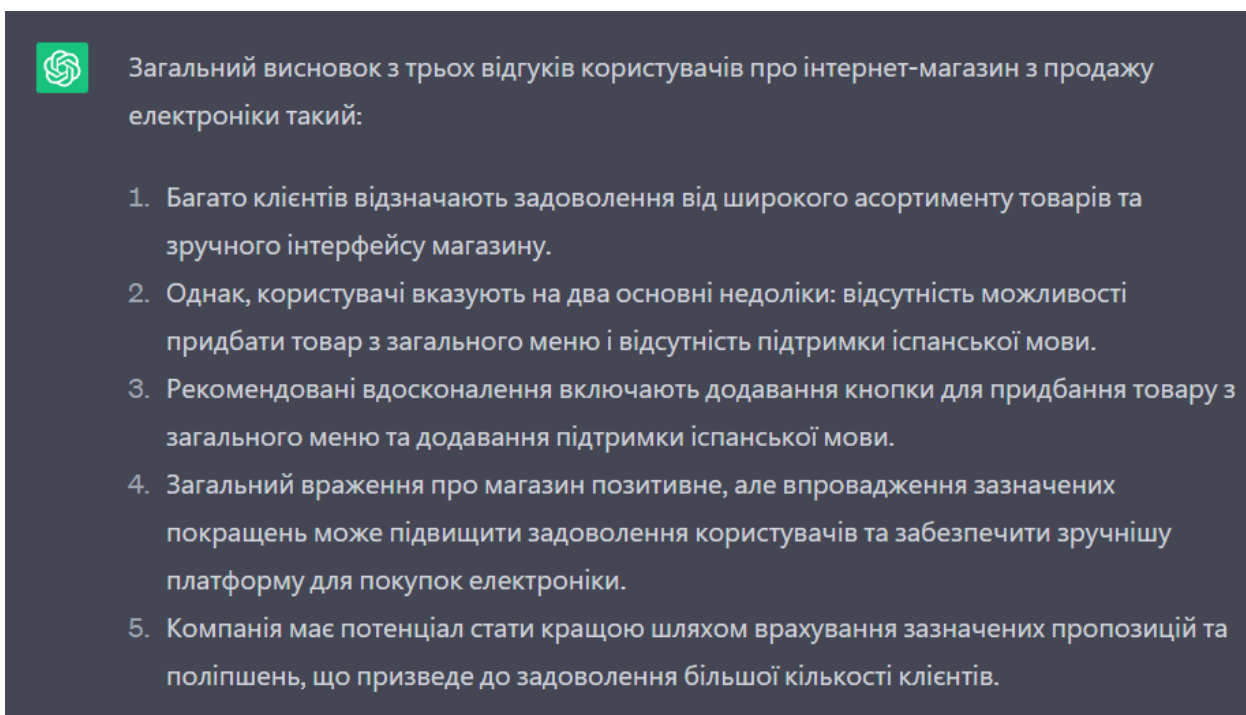


Рис. 23 - Результат виконання запиту

Модель вірно проаналізувала надані відгуки, в більшості з них описувалась проблема відсутності можливості придбання товару з загального меню, підтримка іспанської мови, але загальне враження в усіх було позитивним. Тому враховуючи, що кількість відгуків, які модель може опрацювати може бути значно більшою, ніж була використана у прикладі, то

очевидна перевага використання подібних генеративних моделей. Оскільки, час з якою подібна модель може опрацювати великі масиви даних значно менше, ніж час необхідний людині для такої задачі. Тому, як було показано з допомогою 2 прикладів - аналогічно як і з іншими етапами життєвого циклу продукту, генеративні моделі можуть сильно полегшити роботу керівникам проектів.

5.6 Висновки до розділу

Метою розділу було довести можливість застосування генеративного штучного інтелекту для виконання задач

Це підтверджують наступні факти :

1. Використання ШІ дає можливість значно зекономити час на виконання повсякденних задач проектного менеджера. Кожен з наведених запитів у цьому розділі виконується за лічені хвилини, врахуємо також декілька годин, що необхідні для перевірки отриманого від ШІ завдання, додати ті деталі, що, можливо, не були враховані. Але створення тих же завдань вручну може займати декілька повних робочих днів (стандартним робочим днем вважаємо 8 годин роботи на день), враховуючи також те, що в такому випадку помилок у роботі може бути більше, оскільки якщо моделі було подано усю необхідну інформацію для її навчання та достатню кількість деталей у запиті, то такий результат буде мати значно меншу кількість помилок. Отже, окрім значної переваги у якості, використання розробленого підходу економить час керівника проектів - на виконання задач витрачається у середньому в 4 рази менше часу.
2. Щоб переконатись у тому, що отримані за допомогою генеративного ШІ роботи є якісними, і відповідають технічним стандартам, було проведено опитування серед невеликої кількості відібраних розробників та експерта. У ролі експерта виступив доцент кафедри системного проектування навчально-наукового комплексу “Інститут прикладного системного аналізу” Національного технічного університету України “Київський політехнічний

інститут” та розробник програмного забезпечення Булах Богдан Вікторович. Для опитування було надано 2 різних технічних завдання - розроблене власноруч (рис. 24) та розроблене з допомогою генеративного штучного інтелекту (рис. 25).

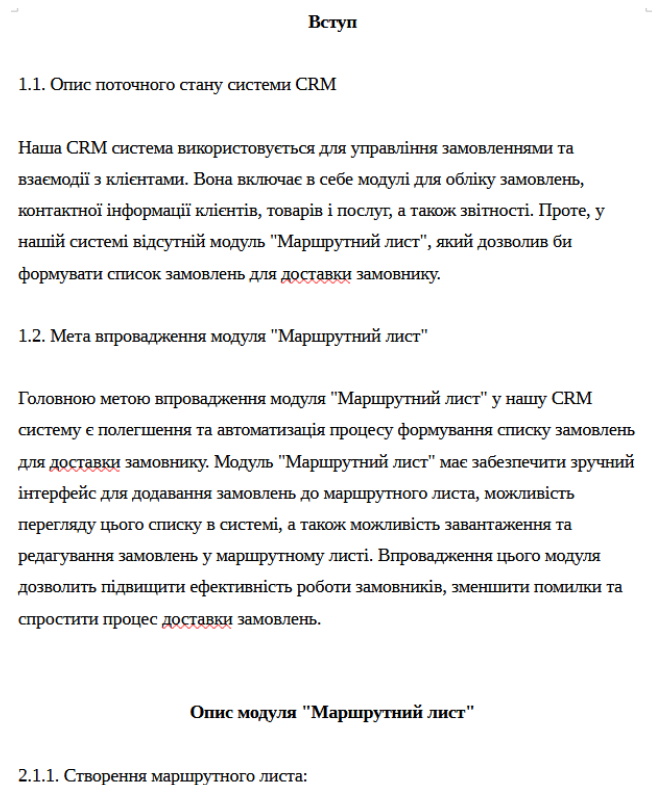


Рис. 24 - Приклад технічного завдання розроблену ІІІ

Технічне завдання на розробку розділу “Маршрутний лист”

Постановка задачі : описані в даному технічному завданні доопрацювання дозволять додати в системі розділ “Маршрутний лист”, що дозволить формувати в системі список замовлень, що необхідно доставити замовнику. Має бути реалізовано можливість переглядати їх в системі, завантажувати та редагувати.

Мета задачі : додасть важливий функціонал в систему, який корисний для компаній, що власноруч доставляють замовлення. Реалізований в даному ТЗ розділ дозволить формувати водіям списки замовлень на конкретну дату, що полегшить цю задачу як для менеджерів, так і безпосередньо для водіїв. Оскільки зараз подібного функціоналу в системі немає, а створення таких листів власноруч займає значний час.

1. Функціональні вимоги до розділу.

- 1.1. Додати розділ “Маршрутний лист” в навігаційному меню у вкладці “Торгівля”.

У навігаційному меню у вкладці “Торгівля” необхідно додати розділ з назвою “Маршрутний лист”, іконка з бібліотеки Feather “navigation” .

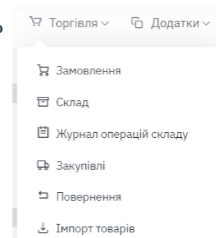
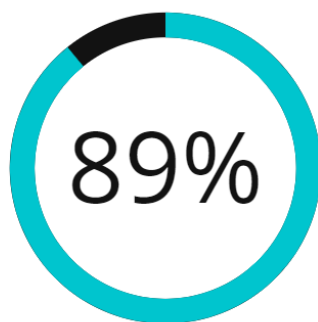


Рис. 25 - Приклад технічного завдання розробленого людиною

Експерт оцінив обидві роботи, правильно вказав де технічне завдання розроблене людиною, де з використанням ШІ. Булах Богдан Вікторович додав : “ Загалом, можна допустити, що обидва ТЗ цілком можна зустріти в реальному житті. Перше ТЗ (розроблене ШІ) містить менше лінгвістичних помилок, але воно дуже абстрактне. Інколи таке трапляється, коли реалізацію віддають на розсуд розробників. Друге ТЗ (розроблене людиною) більш детальне, хоча інколи формулювання заплутані. Звичайно, якщо я розробник конкретної інформаційної системи, то я одразу розпізнаю, яке саме тз справжнє (чи по фрагментам UI, чи по використаним інструментам та технологіям), а яке - зовсім про іншу систему. На рахунок етичності використання подібних інструментів в роботі проектного менеджера, на мою думку, якщо AI "знає" особливості продукту та здатен швидко згенерувати текст по суті, а не просто "залити воду", то чому б не використовувати його в повсякденній роботі - якщо

це прискорить роботу “ . Було проведено також опитування серед невеликої кількості студентів, що мають менший досвід роботи розробника програмного забезпечення, аніж експерт, з метою закріпити впевненість у тому, що використання генеративного штучного інтелекту у майбутньому є доцільним. Більшість думок співпадають з думкою експерта у результаті опитування, результат якого на рис. 26.

**Результати опитування серед студентів-розробників стосовно
доцільно використання ШІ у роботі проектного менеджера**



89% серед опитаних 18 студентів відповіли "Так"

Рис. 26 - Результати опитування студентів

Більшість зазначило, що обидва технічних завдання є доволі якісними, і якщо б вони стикнулись з таким у своїй роботі, то не виникло труднощів з роботою з кожним з них.

Отже, розроблений підхід використання генеративного ШІ було застосовано на практиці, отримано задовільні результати, що підтверджені як експертною оцінкою, так і отриманими за рахунок використання методології перевагами. Тому, можна зробити висновок, що використання генеративного ШІ може значно покращити ефективність роботи керівника проектів. У наступному розділі пропонується розглянути майбутнє даної технології та її можливий вплив на людство.

6 МАЙБУТНЄ ГЕНЕРАТИВНОГО ШІ В УПРАВЛІННІ ПРОЕКТАМИ

В попередніх розділах було розглянуто недоліки існуючих рішень, що надають інструменти взаємодії генеративного ШІ у сфері керування IT-проектами, і розглянуто власний підхід до застосування штучного інтелекту у роботі проектного менеджера. Але оскільки запит на подібні технології та рішення буде лише постійно зростати, то в подальшому будуть з'являтися нові рішення від передових компаній світу - таких як Google, IBM, Microsoft та інших. І за рахунок конкуренції на новому та несформованому ринку розвиток подібних рішень може бути дуже стрімким.

Єдине, що може зупинити швидкі темпи розвитку можливостей штучного інтелекту - нещодавній лист призупинити експерименти з розвитку штучних технологій таких лідерів у даній сфері, як OpenAI - розробників Chat GPT та інших [14]. Даний публічний лист підписали сотні вчених у галузі роботи з штучним інтелектом, засновники корпорацій, до прикладу Стів Возняк, що є одним із засновників компанії Apple, і їх список постійно доповнюється. Але причиною, що зазначається в листі, переживань стосовно активного розвитку штучного інтелекту за останній час є те, що необхідно оцінити можливі наслідки, до чого може призвести такий значний крок у розвитку технологій ШІ, і вирішити питання захисту особистих даних користувачів. Оскільки, розглянутий у роботі застосунок Chat GPT навчається на даних, що користувачі надають йому для роботи, а це суперечить законодавству більшості розвинених країн. Питання юридичного характеру, щодо правомірності використання генеративного ШІ даних користувачів, не є об'єктом дослідження даної роботи, то зупинимось на першому питанні - які наслідки спричинить настільки швидкі темпи розвитку можливостей штучного інтелекту.

Продемонстровані приклади роботи лінгвістичних моделей штучного інтелекту, на прикладі Chat GPT, доводять, що їх рівень роботи починає

наближатися до можливостей роботи людини. Все важче стає відрізнити результат отриманий людиною та штучним інтелектом. І це стосується не лише текстових моделей, що приймають текстовий запит і надають текстову відповідь. Генеративні ШІ, що генерують зображення, такі як MidJourney чи Stable Diffusion, викликали обурення серед спільноти художників, оскільки на платформах для поширення робіт художниками почали завантажуватись роботи створені людиною з використанням штучного інтелекту, які важко відрізнити від роботи зробленої людиною власноруч. Адже незадоволення спільноти художників зрозуміле - навчання даному ремеслу займає не один рік і створення роботи - справа кількох днів або місяців. В той час, для створення зображення з використанням генеративної моделі займає декілька хвилин.

Проте, ідея заборони використання генеративного ШІ в роботі може бути хибною. У світовій історії є не один приклад того, як поява нових революційних технологій не зменшувала кількість робочих місць, навпаки відкривала нові вакансії та позитивно впливала на якість життя. Конкретний приклад того, як технології створюють нові робочі місця, можна побачити на прикладі американської компанії Ford, коли компанія впровадила автоматизовану конвеєрну систему для масового виробництва автомобілів у більших масштабах. Завдяки використанню конвеєрної системи та призначенню робітникам конкретних завдань, продуктивність праці значно зросла, скоротивши час виробництва автомобіля з трохи більше 12 годин до приблизно півтори години. У той час дехто міг би припустити, що ці робітники опиняться без роботи, враховуючи нові технології. Але коли ціна автомобіля впала з 950 доларів у 1909 році до 290 доларів у 1926 році, настав переломний момент у попиті, і кількість проданих автомобілів різко зросла, а це означало, що зайнятість у компанії Ford насправді зросла, а не скоротилася [15].

Тому можна припустити, що розвиток генеративних ШІ позитивно вплине на розвиток проектного менеджменту і людства загалом. Оскільки лінгвістичні

моделі можуть взяти на себе роботу з обробки даних, створення задач та побудови початкових планів роботи, а за керівниками проектів залишиться остаточне прийняття рішень, більш точна адаптація ходу виконання проекту під впливом проблем та викликів, що можуть виникнути під час роботи. Оскільки подібно до того, як методології управління не можуть бути універсальними та підходити під кожен проект, а отже потребують гнучкого використання та різних поєднань їх, і глибокого розуміння від керівника проекту. Аналогічно, що генеративні ШІ не будуть універсальним інструментом, що буде виконувати всю роботу. А буде лише корисним інструментом у руках професіонала. Тому, як було проілюстровано в роботі, що генеративні моделі значно підвищують швидкість виконання задач, а отже і можуть зменшити загальний час, що необхідно на виконання всього проекту. Тому що, як і задачі менеджера будуть виконуватись швидше, за рахунок використання моделей, так само і розробники зможуть швидше отримувати відповіді та допомогу, при використанні розумних чат-асистентів (IBM Watson наприклад). Все це, буде впливати на загальну вартість розробки проектів, і аналогічно з конвеєрною системою компанії Ford, це лише зменшить ціну на розробку програмного забезпечення, отже більше інноваційних ідей зможуть отримати фінансування, і покращити життя людей.

Так само не варто забувати, що як тільки вийшли перші дійсно успішні генеративні ШІ, такі як Chat GPT, то на ринку праці відразу почали з'являтися нові вакансії роботи. Наприклад, така позиція як - Prompt Master (мається на увазі людина, яка спеціалізується на створенні запитів для роботи з генеративним ШІ), не існувала до цього моменту. І лише поява таких потужних рішень як Chat GPT, вже в той же рік створила нові робочі місця.

Підсумовуючи, розвиток технологій штучного інтелекту, в тому числі генеративного, потребує контролю і певного часу, для адаптації сучасних законодавств. Але приклади з історії демонструють, що боятися впровадження їх у життя не буде становити значної загрози для людства, а в результаті лише

покращить комфорт і позитивно вплине на ринок праці. Однією з проблем, що сьогодні існує у сфері керування проектами, є низький рівень задоволення серед значної кількості керівників проектів від своєї роботи, тому після впровадження генеративного штучного інтелекту на постійній основі у їх роботу - може виправити це. За рахунок того, що додасть творчості та збільшить кількість задач, що потребують креативного погляду на їх рішення, прибравши значну частину рутинної роботи.

7 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ЗАСТОСУВАННЯ ГЕНЕРАТИВНОГО ШІ У УПРАВЛІННІ ІТ-ПРОЕКТАМИ

7.1 Вступ

У сучасному світі інформаційних технологій, де швидкість розвитку та конкурентоспроможність є ключовими факторами, ефективне керування ІТ-проектами стає важливим завданням для бізнесу. Розуміння ролі та переваг, які може принести генеративний штучний інтелект в цьому контексті, стає необхідністю для підвищення продуктивності, оптимізації витрат та забезпечення успіху ІТ-проектів.

Генеративний штучний інтелект може надати значні переваги у питаннях підвищення продуктивності в управлінні ІТ-проектами. Його застосування включає автоматизацію рутинних завдань і процесів, зниження витрат робочої сили, зменшення помилок та підвищення точності, а також збільшення швидкості та ефективності виконання завдань. Автоматизація рутинних завдань дозволяє генеративному ШІ виконувати монотонні завдання, такі як формування звітів, створення документації та планування засобів та ресурсів. Це зменшує час і зусилля, витрачені на такі завдання, і звільняє команду від непродуктивних дій. Завдяки автоматизації рутинних завдань, генеративний ШІ дозволяє знизити необхідність у великій кількості робочої сили. Замість того, щоб залучати людей до монотонної роботи, ресурси можуть бути спрямовані на виконання більш складних та стратегічних завдань, які вимагають людського інтелекту і творчості. Це дозволяє ефективніше використовувати людські ресурси і оптимізувати витрати на робочу силу.

Генеративний ШІ також може забезпечити зниження помилок та підвищення точності виконання завдань. Він здатен оперативно та безпомилково виконувати розрахунки, аналізувати дані та забезпечувати високу

якість результатів. Це допомагає уникнути помилок, пов'язаних з людським фактором, і забезпечує більш точні та надійні результати в управлінні IT-проектами. Завдяки автоматизації та оптимізації процесів, генеративний ШІ спроможний прискорити виконання завдань та процесів управління IT-проектами. Він швидко аналізує великі обсяги даних, знаходить шляхи оптимізації та прогнозує результати. Це дозволяє команді зосередитись на стратегічних аспектах проекту та приймати швидкі та обґрунтовані рішення.

7.2 Економічне обґрунтування результатів застосування генеративного ШІ

Після проведення експерименту з застосуванням розробленої методології, що описує підхід використання ШІ у роботі проектного менеджера було отримано докази того, що значних переваг застосування розробленого підходу.

Однією з головних переваг є підвищення продуктивності команди. Автоматизація рутинних завдань та процесів, що розглядаються в цій роботі, дозволяє зменшити зусилля та час, витрачені на їх виконання, тим самим звільняючи команду від монотонної та непродуктивної роботи. Це призводить до підвищення продуктивності команди та зниження витрат робочої сили. І також звільняє час для виконання креативної роботи - побудова різних прогнозів, і пошук нових ефективних підходів до керування командою, вирішення унікальних проблем, що вимагають від проектного менеджера творчого підходу до їх подолання і безліч інших задач. Проденструмано багато прикладів того, як з допомогою лише одного запиту до генеративної моделі було отримано рішення до задачі за лічені хвилини. В той час як ручне виконання такої роботи буде займати декілька годин, а в деяких випадках і декілька днів.

Зекономлений час безпосередньо впливає на загальну вартість розробки проектів. Оскільки ефективне використання робочою часу напряду має значний

вплив на успішність проекту і його фінансові результати. Оскільки знижує вірогідність помилок під час планування та підрахунку початкової вартості проекту, знижує кількість затримок під час роботи над проектом, що завжди є перешкодою для забезпечення дотримання встановленого на етапі планування бюджету і деяким командам дозволить зменшити загальний розмір команди, що дозволяє направити зекономлені кошти на розвиток технологічного аспекту компанії або у будь-який інший напрямок.

Крім того, зараз одними з найпопулярніших напрямків є також Agile та DevOps - це ітеративні підходи до управління проектами та розробки програмного забезпечення, які наголошують на гнучкості, співпраці та задоволеності клієнтів. Agile фокусується на управлінні складними проектами, розбиваючи їх на керовані частини, розставляючи пріоритети відповідно до їхньої цінності для бізнесу та швидко адаптуючись до змін. DevOps - це набір практик, що поєднує розробку програмного забезпечення (Dev) та ІТ-операції (Ops). Штучний інтелект дуже добре підходить для цих методологій завдяки своїй здатності обробляти складні дані, вчитися на них і вдосконалюватися з часом. ШІ може швидко обробляти та аналізувати великі обсяги даних з різних джерел, що дозволяє йому надавати інсайти та робити прогнози щодо продуктивності проекту, ризиків та використання ресурсів. Це може стати основою для прийняття рішень, дозволяючи командам ефективно планувати та коригувати свої стратегії.

Розглянемо вартість інших подібних рішень що існує на ринку та проаналізуємо доцільність застосування власної методології роботи з генеративним ШІ:

1. Використання власної методології на основі Chat GPT може бути більш доступним з фінансової точки зору. До прикладу розглянемо рішення під назвою Forecast, що розглядався в роботі. Вартість

базової версії Forecast - 27 євро / кожного співробітника на місяць, з умовою не менше 20 членів команди. Тобто, $27 \cdot 20 = 540$ євро (21 327 гривень), за найбазовішу версію застосунку. В той час, доступ до генеративної моделі Chat GPT є на даний момент безкоштовним, але без доступу до останніх версій моделей, GPT-3.5 та GPT-4.0, що мають значні покращення. Вартість місячної підписки, що надає доступ до оновлень - 20 доларів на місяць. При використанні даної моделі по API, вартість рахується за принципом - кожні 1000 tokenів (~750 слів) коштують 6 центів (2.19 гривень). До прикладу, при використанні щомісяця 100 000 слів - доведеться заплатити лише 8 доларів (295 гривень). Розглянемо також інший застосунок, що розглядався в роботі - Wrike. Вартість даного рішення за версії “Бізнес”, що є найпопулярнішою в них, становить 24.80 доларів / кожного співробітника на місяць (907 гривень). Якщо взяти той же розмір команди, тобто 20 чоловік, то вартість даного рішення за місяць буде 496 доларів (18 328 гривень). Дане рішення виходить дорожче у 17.7 разів за вартість власної розробленої методології, що є значною перевагою. Важливо зазначити, що вартість готових рішень пропорційно залежить від розміру команди, і буде значно впливати на щомісячну вартість цих застосунків. В той же час, описана методологія не настільки сильно буде зростати у вартості у випадку зростання розміру команди. Оскільки це буде впливати на кількість необхідних tokenів для роботи з моделлю по API. До прикладу, якщо на місяць керівник буде використовувати не 100 000 слів щомісяця, а 1 000 000 слів, то це підніме вартість лише на 72 долари (2 660 гривень).

2. **Масштабованість:** Використання власної методології роботи дозволяє гнучко масштабувати робочі процеси залежно від потреб

вашої команди та проектів. Ви можете адаптувати методологію до конкретних вимог та унікальних характеристик вашого бізнесу. Наприклад, ви можете розширити або скоротити етапи проекту, встановити нові критерії контролю, визначити специфічні ролі та відповідальності для вашої команди. Власна методологія дозволяє оптимізувати робочі процеси та досягати кращих результатів без зайвих витрат на адаптацію готових рішень. Чого не може запропонувати жодне з розглянутих рішень, оскільки вони заточені під конкретні задачі та функції, тому програють розробленій у роботі методології.

3. **Простота інтеграції:** Процес інтеграції рішень штучного інтелекту в існуючу ІТ-інфраструктуру залежить від кількох факторів, специфічних для середовища організації. До них належать складність поточної ІТ-інфраструктури, технології, що вже використовуються, розмір організації, її технологічна досконалість тощо. Деякі інструменти ШІ легше інтегрувати, ніж інші - особливо ті, що розроблені з урахуванням інтероперабельності, або ті, що пропонують API (інтерфейси прикладного програмування) для простішої інтеграції з існуючими системами. Інтеграція може бути простішою, якщо ШІ-рішення від того ж постачальника, що й наявні системи, або якщо воно розроблене на основі загальноприйнятих стандартів. Однак деякі пропрієтарні або вузькоспеціалізовані системи, такі як розглянуті Wrike, Proggio та Forecast, підтримують лише обмежену кількість інтеграцій. В той же час, Chat GPT дозволяє проводити інтеграцію з будь-якими іншими застосунками, що надає набагато більше можливостей для налаштування більш оптимізованого робочого процесу.

7.3 Висновки розділу

У сучасному світі інформаційних технологій, де швидкість розвитку та конкурентоспроможність є ключовими факторами, ефективне керування ІТ-проектами стає необхідним для досягнення успіху в бізнесі. Генеративний штучний інтелект може надати значні переваги в управлінні ІТ-проектами. Його застосування включає автоматизацію рутинних завдань і процесів, зниження витрат робочої сили, зменшення помилок та підвищення точності, а також збільшення швидкості та ефективності виконання завдань.

Автоматизація рутинних завдань дозволяє генеративному ШІ виконувати монотонні завдання, звільняючи команду від непродуктивної роботи. Це допомагає підвищити продуктивність команди та знизити витрати на робочу силу. Завдяки автоматизації, ресурси можуть бути спрямовані на виконання більш складних та стратегічних завдань, що вимагають людського інтелекту та творчості.

Генеративний ШІ також забезпечує зниження помилок та підвищення точності виконання завдань. Він оперативно та безпомилково виконує розрахунки, аналізує дані та забезпечує високу якість результатів. Це допомагає уникнути помилок, пов'язаних з людським фактором, та забезпечує надійність і точність в управлінні ІТ-проектами.

У результаті застосування генеративного ШІ у керуванні ІТ-проектами виявлено значні переваги. Отже використання подібних інструментів є доцільним та обґрунтованим.

ВИСНОВКИ

У даній роботі було проведено аналіз можливостей застосування генеративного штучного інтелекту в управлінні ІТ проектами та розроблено власну методологію застосування ШІ у роботі проектного менеджера. Актуальність проблеми пояснюється тим, що керування проектами відіграє вагомому частку в успішності виконання будь-якого проекту, тому впроваджуючи нові інноваційні рішення у цю сферу, можна досягнути покращення ефективності досягнення поставлених цілей.

У роботі розглянуто наявні проблеми у сфері управління проектами та недоліки існуючих рішень, що надають інструменти генеративного штучного інтелекту для керування ІТ-проектами. Враховувавши ці проблеми, було розроблено власний підхід керування ІТ-проектами з використанням генеративного ШІ. Підхід базується на моделі життєвого циклу розробки програмного забезпечення, але за рахунок своєї універсальності може застосовуватись у поєднанні з різними методологіями керування проектами. Основними принципами описаної методології є : використання даних минулих проектів, для навчання моделі штучного інтелекту, що забезпечить кращі результати роботи; використання ШІ для виконання рутинних задач, оскільки такі задачі як створення технічних завдань, аналіз відгуків тощо, легко виконувати за допомогою штучного інтелекту, оскільки вони розраховані на аналіз великих обсягів даних та пошук патернів в них; і використання максимальної кількості деталей у запитах до генеративного штучного інтелекту, що підвищує якість отриманого результату, бо подібні технології не досягли рівня людського інтелекту, і не можуть створювати нові підходи та рішення, вони чудово демонструють себе лише при свідомому та інтелектуальному підході від керівника проектів, що використовує ШІ у своїй роботі.

Після практичного застосування та аналізу отриманих результатів - було підтверджено думку, що ШІ є гарним інструментом, що позитивно впливає на результат роботи проектного менеджера, і в той же час, дозволяє людині працювати з творчими завданнями, що значно підвищує задоволення від робочого процесу.

Висновки розділів підтверджують потенціал використання ШІ у майбутньому і важливість подальшого дослідження цієї теми.

У майбутньому можливе покращення даних рішень за рахунок інтеграції різних генеративних штучних інтелектів, до прикладу поєднання ШІ, що генерує текст та зображення, або текстових ШІ та ШІ, що генерує музику. Значний вплив на розвиток будуть мати світові корпорації, що активно зараз інвестують у подібні рішення, і зацікавлені у отриманні власних інноваційних ШІ.

Отримані результати відповідають поставленому завданню та очікуванням від роботи.

ДЖЕРЕЛА ІНФОРМАЦІЇ

1. PMI | Project Management Institute [Електронний ресурс]: - Режим доступу до ресурсу:
<https://www.pmi.org/about/learn-about-pmi/what-is-project-management> .
Дата доступу: 25.04.2023 р.
2. Westland J. Top 10 Project Management Methodologies: An Overview [Електронний ресурс] / Jason Westland // ProjectManager. – Режим доступу: <https://www.projectmanager.com/blog/project-management-methodology>
Дата доступу: 27.04.2023 р.
3. Generic AI: Artificial General Intelligence - Generic AI Home Page, [Електронний ресурс]: - Режим доступу: <http://www.generic-ai.co.uk/> , Дата доступу: 10.05.2023 р.
4. Making The Generic AI Trend Work For Your Business, Режим доступу: <https://www.forbes.com/sites/forbestechcouncil/2021/06/18/making-the-generic-ai-trend-work-for-your-business/?sh=e94230537682> , Дата доступу: 11.05.2023 р.
5. AI for Work and Project Management | Wrike, Режим доступу: <https://www.wrike.com/features/work-intelligence/> , Дата доступу: 13.05.2023 р.
6. Proggio Review - Project Management Tool Reviews, Pricing & Features, [Електронний ресурс] : - Режим доступу: <https://productivityland.com/proggio/> , Дата доступу: 14.05.2023 р.
7. Proggio - Project Portfolio Management Solution For Enterprises,[Електронний ресурс] : - Режим доступу: <https://www.proggio.com/how-it-works/> , Дата доступу: 14.05.2023 р.

8. Forecast nabs \$19M for its AI-based approach to project management and resource planning | TechCrunch,[Електроний ресурс] : - Режим доступу: <https://techcrunch.com/2021/05/19/forecast-nabs-19m-for-its-ai-based-approach-to-project-management-and-resource-planning/> , Дата доступу:15.05.2023 р.
9. How AI Will Transform Project Management,[Електроний ресурс] : - Режим доступу: <https://hbr.org/2023/02/how-ai-will-transform-project-management> , Дата доступу:15.05.2023 р.
10. Asana AI: Revolutionizing Project Management with Artificial Intelligence - DotCom Magazine-Influencers And Entrepreneurs Making News,[Електроний ресурс] : - Режим доступу: <https://www.dotcommagazine.com/2023/04/asana-ai-revolutionizing-project-management-with-artificial-intelligence/> , Дата доступу:16.05.2023 р.
11. Project Management Institute (PMI),[Електроний ресурс] : - Режим доступу: <https://www.pmi.org/learning/library/project-management-problems-anticipating-conflict-5769> , Дата доступу: 20.05.2023.
12. ДСТУ 2851-94. Програмні засоби ЕОМ. Документування результатів випробувань. – Чинний від 1996-01-01.
13. "IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998 , vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286.
URL : <https://ieeexplore.ieee.org/document/720574> .
14. Future of Life Institute, [Електроний ресурс] : - Режим доступу: <https://futureoflife.org/open-letter/pause-giant-ai-experiments/> , Дата доступу: 01.06.2023 р.

15. Deutsche Bank Research, [Электроний ресурс] : - Режим доступа:

[https://www.dbresearch.com/PROD/RPS_EN-PROD/PROD0000000000528346/Can_AI_help_high-labour_firms_to_boost_returns%3F.xhtml?rwnode=RPS_EN-PROD\\$HIDDEN_GLOBAL_SEARCH](https://www.dbresearch.com/PROD/RPS_EN-PROD/PROD0000000000528346/Can_AI_help_high-labour_firms_to_boost_returns%3F.xhtml?rwnode=RPS_EN-PROD$HIDDEN_GLOBAL_SEARCH) , Дата доступа: 03.06.2023 г.