

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет автоматизації і інформаційних технологій

Кафедра управління проєктами

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ БАКАЛАВР

на тему:

**«Управління проєктом розробки системи підтримки прийняття рішення із
застосуванням підходів штучного інтелекту»**

Федянін Кирило Олегович

Київ 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І
АРХІТЕКТУРИ**

Факультет автоматизації і інформаційних технологій

Кафедра управління проєктами

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Веренич О.В.

„___” _____ 20__ року

**КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ БАКАЛАВР**

**«Управління проєктом розробки системи підтримки прийняття рішення із
застосуванням підходів штучного інтелекту»**

*Я як здобувач вищої освіти КНУБА
розумію і підтримую
політику закладу з
академічної доброчесності. Я не
надавав(-ла) і не одержував(-ла)
недозволену допомогу під час
підготовки цієї роботи.*

*Використання ідей,
результатів і текстів інших
авторів мають посилання
на відповідне джерело.*

Здобувач Федянін Кирило Олегович

(прізвище, ім'я та по батькові повністю)

126 Інформаційні системи та технології

(спеціальність)

управління проєктами

(освітня програма)

Група ІСТ-УП21

Керівник Веренич О.В.

(прізвище та ініціали)

д.т.н., проф

(вчене звання, науковий ступінь)

Рецензент Запривода А.В.

(прізвище та ініціали)

Ідентичність підтверджую

Київ 2025 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ

Факультет автоматизації і інформаційних технологій

Кафедра управління проєктами

Ступінь вищої освіти: бакалавр

Галузь знань: 26 Інформаційні технології

Спеціальність: 126 Інформаційні системи та технології

Освітня програма: управління проєктами

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Веренич О.В.

„___” _____ 20__ року

З А В Д А Н Н Я ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ БАКАЛАВР

Федянін Кирило Олегович

1. Тема роботи «Управління проєктом розробки системи підтримки прийняття рішення із застосуванням підходів штучного інтелекту» затверджена наказом ректора КНУБА №293/23/25 від «21» лютого 2025 року

2. Керівник роботи: д.т.н., професор Веренич О.В.

3. Термін подання здобувачем роботи до захисту 12.06.2025

4. Зміст пояснювальної записки за розділами:

Р. 1. Теоретичні засади побудови систем підтримки прийняття рішень із використанням штучного інтелекту

Р. 2. Управління проєктом розробки СППР

Р. 3. Виконання та контроль реалізації проєкту

Р. 4. Практичні аспекти впровадження та експлуатації системи підтримки прийняття рішень

5. Графічний матеріал за розділами

Р. 1. _____

Р. 2. _____

Р. 3. _____

Р. 4. _____

6. Консультанти розділів кваліфікаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірів	
		дата	підпис
Розділ 1.			
Розділ 2.			
Розділ 3.			
Розділ 4.			
Розділ 5			

7. Календарний план виконання роботи:

Види робіт та їх зміст	Дата виконання
Розділ 1. Теоретичні засади побудови систем підтримки прийняття рішень із використанням штучного інтелекту	01.03.2025
Розділ 2. Управління проєктом розробки СППР	28.03.2025
Розділ 3. Виконання та контроль реалізації проєкту	15.04.2025
Розділ 4. Практичні аспекти впровадження та експлуатації системи підтримки прийняття рішень	10.05.2025
Остаточне оформлення роботи	30.05.2025
Направлення роботи для перевірки на плагіат	10.06.2025
Попередній захист роботи на випусковій кафедрі	12.06.2025
Направлення роботи на рецензування	12.06.2025

8. Дата видачі завдання 21.02.2025

Керівник

(підпис)

Веренич О.В.

(прізвище та ініціали)

Здобувач

(підпис)

Федянін К.О.

(прізвище та ініціали)

РЕЗЮМЕ

РЕЗЮМЕ (SUMMARY)			
<i>до кваліфікаційної випускної роботи здобувача:</i>		<i>(ПІБ здобувача українською та англійською)</i>	
ЗВО	Київський національний університет будівництва і архітектури		
Тема <i>(українською та англійською)</i>	«Управління проектом розробки системи підтримки прийняття рішення із застосуванням підходів штучного інтелекту» «Management of a project to develop a decision support system using artificial intelligence approaches»		
Освітній ступінь	бакалавр		
Факультет	автоматизації і інформаційних технологій		
Випускова кафедра	управління проектами		
Спеціальність	126 інформаційні системи та технології		
Освітня програма	управління проектами		
Керівник	Веренич О.В.		
Обсяг роботи:	<i>пояснювальна записка, стор.</i>	<i>розділів</i>	<i>креслень формату А₁</i>
	74	4	-
Розділ 1	Теоретичні засади побудови систем підтримки прийняття рішень із використанням штучного інтелекту		
Розділ 2	Управління проектом розробки СППР		
Розділ 3	Виконання та контроль реалізації проекту		
Розділ 4	Практичні аспекти впровадження та експлуатації системи підтримки прийняття рішень		
Висновки по роботі:	У межах дипломної роботи здійснено теоретичне обґрунтування розробки системи підтримки прийняття рішень HealthAI на основі сучасних інструментів штучного інтелекту в медицині. Розглянуто архітектуру системи, управлінські аспекти проекту та ергономіку інтерфейсу, що в сукупності підтверджує доцільність і перспективність її впровадження у клінічну практику. Отримані результати створюють підґрунтя для подальшої програмної реалізації та тестування.		
Ключові слова:	система підтримки прийняття рішень, штучний інтелект, охорона здоров'я, GPT-4, DeepSeek, архітектура системи, медична інформаційна система.		
Keywords:	decision support system, artificial intelligence, healthcare, GPT-4, DeepSeek, system architecture, medical information system.		

Студент _____ Федянін К.О.

Керівник _____ Веренич О.В.

АНОТАЦІЯ

У дипломній роботі розроблено концептуальну модель системи підтримки прийняття рішень (СППР) у сфері охорони здоров'я із застосуванням сучасних інструментів штучного інтелекту. Актуальність дослідження зумовлена потребою підвищення ефективності клінічних рішень в умовах перевантаження медичного персоналу, фрагментованості медичних даних і високих вимог до точності діагностики. Основною метою роботи є обґрунтування, проектування та організаційне моделювання інтелектуальної системи HealthAI, яка може бути використана як допоміжний інструмент лікаря на етапі постановки діагнозу та підбору лікування.

У першому розділі розглянуто еволюцію СППР, їхні типи, сучасні інструменти (зокрема GPT-4 та DeepSeek), прикладні сфери застосування та особливості впровадження в медичну інформаційну інфраструктуру. Другий розділ присвячено управлінню проектом розробки: визначено структуру команди, WBS-план, бюджет, часові межі реалізації, ризики та заходи з контролю якості. Третій розділ охоплює архітектурну побудову системи, опис потоків даних, логіку обробки запитів і взаємодію між компонентами. Увагу зосереджено на інтеграції мовної моделі GPT-4, модуля аналітики DeepSeek, сценаріях запитів і збереженні контексту. У четвертому розділі подано опис функціональних модулів, способів обробки даних, реалізації вимог до інформаційної безпеки, ергономіки інтерфейсу користувача та базових підходів до тестування системи.

Робота має теоретичний характер і спрямована на міждисциплінарну інтеграцію знань у сфері ІТ, медицини та управління проектами. Результати можуть бути використані як основа для створення прототипу, подальшого програмного впровадження, тестування й адаптації системи до реального медичного середовища.

ЗМІСТ

РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ ПОБУДОВИ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ІЗ ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ	4
1.1 Основи систем підтримки прийняття рішень та роль штучного інтелекту	4
1.2 Огляд сучасних інструментів штучного інтелекту для СППР	6
1.3 Інструменти штучного інтелекту в системах підтримки прийняття рішень	7
1.4 Технологічні й етичні обмеження впровадження ШІ в СППР	9
1.5 Аналіз СППР	11
1.6 Аналіз сучасного стану сфери охорони здоров'я	15
Висновки до розділу 1	16
РОЗДІЛ 2. УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ СППР	18
2.1 Обґрунтування вибору методології управління проєктом.....	18
2.2 Структура команди та розподіл ролей	20
2.3 Статут проєкту	21
2.4 SMART-аналіз цілей проєкту та PEST-аналіз СППР для медичних закладів	21
2.5 Планування та часові межі проєкту	23
2.6 Структура робіт, ресурси та бюджет проєкту	27
2.7 Управління ризиками та якістю проєкту	30
Висновки до розділу 2	32
РОЗДІЛ 3. ВИКОНАННЯ ТА КОНТРОЛЬ РЕАЛІЗАЦІЇ ПРОЄКТУ	33
3.1 Загальний опис архітектури системи	33
3.2 Діаграма потоків даних системи	34
3.3 Діаграма діалогових вікон	36
3.4 Діаграма класів.....	38
3.5 Інтеграція мовної моделі ChatGPT та модуля аналітики DeepSeek	41

3.6	Опис структури бази даних	43
3.7	Тестування системи.....	45
	Висновки до розділу 3	47
	РОЗДІЛ 4. ПРАКТИЧНІ АСПЕКТИ ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ.....	48
4.1	Використання відкритих датасетів у системі підтримки прийняття рішень	48
4.2	Ролі користувачів у системі підтримки прийняття рішень	51
4.3	Ергономіка користувацького інтерфейсу системи	52
4.4	Забезпечення безпеки даних у процесі проєктування	57
	Висновки до розділу 4	58
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62
	ДОДАТКИ.....	69

РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ ПОБУДОВИ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ІЗ ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Перший розділ присвячено аналізу теоретичних засад функціонування систем підтримки прийняття рішень (СППР), їхньої ролі в сучасному інформаційному середовищі, а також місцю штучного інтелекту в їхній архітектурі. У межах цього розділу розглянуто основні поняття, етапи розвитку СППР, сучасні підходи до обробки інформації та типові сфери застосування. Окрему увагу приділено аналізу актуального стану медичної галузі та доцільності впровадження інтелектуальних інструментів у практику прийняття клінічних рішень. Вивчення цих аспектів створює основу для обґрунтування доцільності розробки системи HealthAI та визначення її функціональних можливостей.

1.1 Основи систем підтримки прийняття рішень та роль штучного інтелекту

Системи підтримки прийняття рішень (СППР) являють собою клас інформаційних систем, призначених для допомоги користувачам в аналізі даних і формуванні обґрунтованих рішень [1, с. 3]. Їхнє виникнення пов'язане з потребою обробки великих обсягів інформації, що унеможливило прийняття ефективних рішень виключно на основі ручного аналізу [2, с. 2].

Початкові форми СППР з'явилися в середині ХХ століття і були орієнтовані на обробку структурованих даних [3, с. 21]. Вони включали прості модулі вибору альтернатив на основі фільтрації, сортування, агрегування статистичних показників. Надалі розвиток обчислювальних засобів привів до створення експертних систем, які намагалися відтворити логіку мислення фахівців [4, с. 117]. Такі системи будувались на основі правил типу «якщо — то», мали знання, представлені у вигляді дерев логіки або продукційних моделей, і використовувались у медичній, юридичній, технічній практиці. У ХХІ столітті еволюція СППР відбулася в напрямі застосування

машинного навчання, що дозволило автоматизувати накопичення досвіду та адаптацію моделей до нових даних [5, с. 5]. Це дало змогу системам самостійно формувати правила на основі статистичної значущості, враховуючи змінність реального середовища. З'явилися гібридні моделі, які поєднують аналітику, прогнозування, пояснення та візуалізацію в єдиному інтерфейсі.

Поширення технологій штучного інтелекту стало визначальним чинником у трансформації СППР. Обробка природної мови, глибоке навчання та великі мовні моделі значно розширили спектр даних, з якими можуть працювати системи [6, с. 1]. Сучасні СППР здатні формувати припущення, аналізувати контекст, генерувати текстові пояснення та забезпечувати інтерфейс на основі природної мови [7, с. 12]. Це робить системи доступними для широкого кола користувачів без спеціальної підготовки [8, с. 6].

Типологія СППР за рівнем інтелектуалізації включає три основні категорії: інформаційно-пошукові, аналітично-діагностичні, предиктивно-рекомендаційні. Останній тип найтісніше пов'язаний із використанням нейромереж, класифікаційних алгоритмів, глибокого навчання. У межах охорони здоров'я це дозволяє формувати діагностичні підозри, пріоритети обстежень і навіть оптимальні траєкторії лікування — за умови, що система має достатню кількість анонімізованих прикладів.

З огляду на ці тенденції, у межах даного проекту реалізовано підхід, за якого мовна модель GPT-4 відповідає за взаємодію з користувачем, а аналітичний модуль DeepSeek за пошук підтверджуючої інформації. Така комбінація дозволяє зберегти баланс між інтуїтивною мовною взаємодією та обґрунтованістю відповідей [9, с. 3]. Крім того, вона дозволяє розділити завдання на окремі підсистеми: інтерпретацію запиту, семантичне звуження тематики, пошук релевантних даних і формування пояснення. Така архітектура є типовою для сучасних СППР і підвищує їхню масштабованість та стійкість до помилок.

Таким чином, СППР сьогодні — це не просто інструменти фільтрації або візуалізації, а адаптивні, навчаючі системи, здатні до генерації сенсу. Застосування штучного інтелекту в межах таких систем відкриває нові горизонти в медичній

практиці, дозволяючи лікарю не лише отримати інформацію, але й глибше її зрозуміти, виявити латентні зв'язки та отримати персоналізовану підтримку у прийнятті клінічного рішення.

1.2 Огляд сучасних інструментів штучного інтелекту для СППР

Для забезпечення функціональності системи підтримки прийняття рішень у межах дипломного проєкту було проведено порівняльний аналіз наявних мовних і аналітичних інструментів. Основна увага приділялася можливостям обробки природної мови, наявності прикладного інтерфейсу програмування (API), якості відповідей, масштабованості, швидкодії та ресурсним вимогам [10, с. 1].

Модель ChatGPT від OpenAI (архітектура GPT-4) забезпечує генерацію текстових відповідей на основі контексту запиту. Її перевагами є підтримка багатомовності, гнучкість стилістичної адаптації, здатність працювати з неструктурованими запитам та висока якість лінгвістичного представлення інформації. Важливим фактором є також наявність офіційного API з документацією, що дозволяє інтегрувати модель у веб-інтерфейс без надлишкових витрат. Проте модель не має вбудованого механізму перевірки фактів — її відповіді базуються на ймовірнісному прогнозуванні слів, що створює ризик генерації хибної або непідтвердженої інформації [9, с. 4].

Для компенсації цього недоліку використовується аналітичний модуль DeepSeek, що спеціалізується на семантичному пошуку релевантної інформації в текстових корпусах. Цей інструмент реалізує підхід “retrieval-augmented generation”, за якого модель спочатку здійснює вибірку джерел на основі схожості запиту, а вже потім — формує відповідь. Таким чином, DeepSeek дозволяє забезпечити логічну обґрунтованість згенерованих рекомендацій, оскільки вони супроводжуються витягнутими джерелами [8, с. 6]. Серед обмежень — високе навантаження на оперативну пам'ять при роботі з великими наборами документів і складність адаптації до україномовних джерел [11, с. 159].

Розглядалися також альтернативні системи. Зокрема, Google Vertex AI забезпечує глибоку кастомізацію моделей та має вбудовану підтримку пайплайнів обробки даних, однак для запуску навіть базових сценаріїв потрібні знання хмарних технологій Google Cloud Platform та наявність комерційної ліцензії. IBM Watson пропонує готові рішення для галузевих задач, зокрема в медицині, але має складну структуру модулів, які важко налаштувати в індивідуальному проєкті. Claude (Anthropic) та Mistral демонструють високі результати в текстовому аналізі, проте API доступ обмежений або нестабільний [12, с. 98].

Окремо слід зазначити перспективні легкі моделі: Perplexity AI, Bing AI або You.com, які використовують пошуковий механізм у реальному часі. Вони придатні для пошуку базових фактів або джерел, проте не забезпечують повноцінної багатокрокової логіки міркування, необхідної для побудови персоналізованих медичних рекомендацій [3, с. 145].

З огляду на наведене, доцільним стало використання пари GPT-4 + DeepSeek, що дозволяє реалізувати ключову функціональність із мінімальними обмеженнями для обчислювального середовища. GPT-4 відповідає за інтерфейсну взаємодію, переформулювання запитів і генерацію пояснень, а DeepSeek — за витяг підтверджуючої інформації з корпусу знань. Така архітектура забезпечує баланс між інтуїтивною мовною взаємодією і верифікованістю результатів, що особливо важливо в системах медичного спрямування [7, с. 71].

1.3 Інструменти штучного інтелекту в системах підтримки прийняття рішень

Сучасні системи підтримки прийняття рішень (СППР) дедалі активніше інтегрують інструменти штучного інтелекту, що дозволяє суттєво розширити їхні функціональні, адаптивні та комунікативні можливості. Штучний інтелект у цьому контексті розглядається не лише як механізм для підвищення точності прогнозування, а й як компонент, що дозволяє забезпечити природну інтерактивність

та гнучкість інтерфейсу. Інструменти, які використовуються, включають великі мовні моделі (LLM), системи класифікації, інтелектуальні механізми пошуку, модулі логічного аналізу та сервіси перевірки гіпотез [11, с. 2].

Застосування мовних моделей типу GPT-4 від OpenAI відкриває нові підходи до побудови взаємодії між користувачем і системою. Завдяки підтримці обробки природної мови користувач має змогу формулювати запити у зручній і звичній формі, без потреби володіти знаннями формальних мов або запитних конструкцій [9, с. 6]. Такий підхід значно знижує когнітивне навантаження та розширює потенційне коло користувачів – від студентів і медичних сестер до досвідчених лікарів та аналітиків [8, с. 3].

Мовна модель GPT-4 здатна забезпечити генерацію текстових відповідей, які враховують як формальну логіку запиту, так і його контекстуальне оточення. Стилiстична узгодженiсть, багаторiвнєве розумiння значення та збереження фокусованої тематики відповіді є сильними сторонами моделі. Водночас GPT-4 є передусім предиктивною, а не фактографічною моделлю. Тобто, відповіді формуються на основі вірогідних шаблонів, а не перевірених знань, що може призводити до галюцинацій ШІ – появи логічно правильного, але фактично помилкового тексту [2, с. 4].

Для компенсації цього недоліку у межах дипломного проєкту використано модуль DeepSeek, який реалізує семантичний пошук і верифікацію фактів у текстових джерелах. DeepSeek працює за принципом пошукової доповненості генерації (retrieval-augmented generation, RAG), дозволяючи мовній моделі спиратися не лише на вбудовану мовну інтуїцію, а й на релевантні витяги з корпусу знань [14, с. 48]. Це дозволяє покращити обґрунтованість і точність відповідей, особливо у випадках, що вимагають фактологічної або юридично чутливої інформації. Інструмент також демонструє гнучкість при обробці україномовного контенту за умови попереднього налаштування словників.

Окрім GPT-4 і DeepSeek, в процесі аналітичного огляду розглядалися й інші інструменти. Наприклад, IBM Watson — добре відомий сервіс зі сфокусованою

підтримкою для медичної галузі. Він дозволяє створювати логічні пайплайни аналізу даних, інтегрується з електронними медичними записами (EMR), однак вимагає складного налаштування та значних ресурсів інфраструктури, що не виправдано для прототипу [12, с. 98]. Сервіс Google Vertex AI забезпечує високу масштабованість, можливість навчання кастомних моделей та побудову цілих ML-процесів, однак має високий поріг входження та вартість використання.

Такі сервіси, як Claude (Anthropic), орієнтовані на дотримання принципів безпечного генерування, проте обмежений API-доступ знижує практичну зручність їх інтеграції. Інструменти, як-от Perplexity AI або Bing AI, базуються на злитті пошукової та мовної обробки, проте придатні переважно для ознайомчого пошуку і не мають розширених функцій структурованого аналізу або логічного моделювання [13, с. 147].

Рішення використовувати пару GPT-4 і DeepSeek обґрунтоване не лише високим рівнем якості та гнучкістю інтеграції, але й здатністю побудувати синергічну систему: мовна модель здійснює діалог, формує пояснення, адаптується під стиль користувача, тоді як DeepSeek виконує роль фактологічного «якоря», перевіряючи правдивість ключових тверджень [3, с. 82]. Така комбінація забезпечує одночасно доступність для користувача та точність для аналітика, що є критично важливим у медичному контексті.

1.4 Технологічні й етичні обмеження впровадження ШІ в СППР

Інтеграція інструментів штучного інтелекту в системи підтримки прийняття рішень супроводжується низкою викликів, які необхідно враховувати на етапах проектування, тестування, експлуатації та подальшої адаптації таких систем [13, с. 147]. Незважаючи на очевидні переваги, використання великих мовних моделей, модулів семантичного аналізу чи систем предиктивного прогнозування нерозривно пов'язане з технічними, концептуальними та етичними обмеженнями.

Однією з ключових проблем є непрозорість внутрішньої логіки ШІ-моделей, зокрема трансформерних архітектур, які становлять основу GPT-подібних систем. Моделі функціонують за принципом імовірнісного прогнозування наступного токена, що не передбачає чіткого причинно-наслідкового обґрунтування висновку. У результаті виникає феномен «чорної скриньки», коли користувач не може ані передбачити поведінку системи, ані обґрунтувати отриману відповідь [2, с. 3]. У таких галузях, як медицина чи юриспруденція, де кожне рішення має бути аргументованим і верифікованим, ця проблема набуває критичного характеру [8, с. 8].

Ще одним суттєвим ризиком є упередженість моделей, яка виникає внаслідок використання тренувальних даних, що містять соціальні, гендерні, расові або культурні перекося. Якщо такі моделі застосовуються без процедури дебіасингу (усунення упередженості) та тестування на етичну нейтральність, то вони можуть генерувати дискримінаційні або некоректні рекомендації, що особливо небезпечно в контексті охорони здоров'я [9, с. 4].

До технічних обмежень належить значна потреба у ресурсах. Сучасні моделі, як от GPT-4 або аналогічні, вимагають від 8 до 40 ГБ оперативної пам'яті лише для inference-запитів, не враховуючи дискову та мережеву інфраструктуру. Для більшості малих і середніх установ це унеможлиблює локальне розгортання без використання хмарних рішень, що, у свою чергу, піднімає питання конфіденційності. Крім того, інтеграція ШІ у наявну інфраструктуру передбачає сумісність із базами даних, API, форматами зберігання та протоколами безпеки, що не завжди є можливим без суттєвих змін [12, с. 100; 3, с. 84].

З етичної точки зору особливо важливим є визначення меж відповідальності. Рекомендації, що формуються ШІ-модулями, не повинні трактуватися як директивні чи остаточні. Вони мають статус допоміжного інструменту. Рішення має ухвалюватися людиною – лікарем, експертом, консультантом, а система має лише підтримувати цей процес, а не заміщати його [7, с. 11]. Водночас потребує гарантування конфіденційність персональних даних, зокрема медичних, які часто

обробляються у запитах. Це стосується як технічного захисту каналів передачі, так і принципів знеособлення та верифікації доступу [10, с. 2].

Не менш актуальним є соціально-психологічний аспект — зниження довіри до рішень, ухвалених із залученням ШІ, особливо у випадках непередбачуваних результатів. У деяких випадках користувачі переоцінюють компетентність систем або, навпаки, ігнорують корисні рекомендації. Це вимагає навчання, стандартизації пояснень і введення механізмів «контрольованої автономії» ШІ у критичних сферах.

Таким чином, використання інструментів штучного інтелекту в системах підтримки прийняття рішень є виправданим і перспективним кроком, проте вимагає поєднання технологічної зрілості з етичною відповідальністю. Кожне рішення повинно бути обґрунтоване не лише алгоритмічно, але й з позиції практичної доцільності, безпеки та суспільного прийняття [11, с. 160].

1.5 Аналіз СППР

Системи підтримки прийняття рішень (СППР) активно впроваджуються у різні сектори суспільного й економічного життя, де існує потреба в аналізі великої кількості змінних, складних сценаріях прийняття рішень або високих ризиках помилок. Їх ефективність визначається здатністю адаптуватися до специфіки предметної галузі, обробляти необхідний тип інформації (структуровану або неструктуровану), забезпечувати візуалізацію, прогнозування та надання обґрунтованих рекомендацій [15, с. 78].

У фінансовому секторі СППР застосовуються для моделювання кредитного ризику, оцінки платоспроможності клієнтів, формування інвестиційних стратегій, а також виявлення шахрайських операцій. Типовим прикладом є кредитний скоринг, де система аналізує історію клієнта, його поведінкові патерни й зовнішні показники, приймаючи рішення про надання позики [16, с. 151]. Використання нейромереж і дерев рішень у поєднанні з методами кластеризації дозволяє сформулювати більш точні моделі поведінкових ризиків.

У корпоративному управлінні СППР використовуються для стратегічного та оперативного планування, оптимізації логістичних ланцюгів, аналізу сценаріїв розвитку, управління ризиками й ресурсами підприємства. Системи допомагають керівникам вибрати оптимальний план дій із урахуванням змін у ринку, вартості ресурсів, попиту й нормативних обмежень [17, с. 112]. Інструменти типу Power BI, Tableau, SAP Decision Support моделюють варіанти розвитку бізнесу з урахуванням обмежень.

В освітній сфері СППР сприяють реалізації принципів персоналізованого навчання. Вони застосовуються для формування індивідуальних навчальних траєкторій, діагностики прогалів у знаннях, адаптації змісту до стилю сприйняття учня. В окремих системах впроваджено інтелектуальні тьютори, які надають рекомендації викладачу щодо методів і форм подання матеріалу, часу повторення теми, вибору додаткових ресурсів. Такі рішення вже активно використовуються в MOOC-платформах, корпоративному e-learning і державних освітніх системах.

У державному управлінні СППР реалізуються через платформи аналітики рішень — наприклад, у сфері розподілу бюджетів, регіонального планування, цифровізації адміністративних послуг. Рішення на основі даних дозволяють аналізувати ефективність соціальних програм, передбачати потреби громадян, оцінювати вплив регуляторних змін. На рівні муніципалітетів впроваджуються системи для управління транспортними потоками, енергоспоживанням, наданням медичних або екстрених послуг [18, с. 54].

Особливу увагу слід приділити застосуванню СППР у сфері охорони здоров'я, яке є одним із найперспективніших напрямів. Такі системи дозволяють лікарям приймати обґрунтовані рішення у процесі діагностики, планування лікування, вибору медикаментів, призначення обстежень. У складних клінічних ситуаціях СППР можуть автоматично виділяти ключові симптоми, порівнювати їх із базою випадків, оцінювати ймовірності діагнозів, сигналізувати про можливі протипоказання [19, с. 3].

У практиці медичних закладів застосовуються СППР для супроводу хронічних пацієнтів, ведення електронних медичних карток, підтримки прийняття рішень у реальному часі, зокрема в інтенсивній терапії та невідкладних станах [5, с. 7]. Розвиток мовних моделей і аналітичних інструментів сприяв поширенню СППР у вигляді інтегрованих модулів у системи eHealth, автоматизації протоколів лікування, генерації попереджень на підставі лабораторних даних або анамнезу [10, с. 2].

У межах дипломного проєкту розроблена система HealthAI є прикладом СППР, орієнтованої саме на медичне застосування. Вона дозволяє лікарю швидко отримувати текстове зведення щодо стану пацієнта, аналізувати призначення, історію обстежень та взаємодіяти з інтерфейсом на основі природної мови. Таким чином, вона реалізує інструментарій, типовий для сучасних СППР: обробка неструктурованих даних, пояснюваність рішень, адаптація до запиту, рекомендаційна підтримка.

Інтеграція систем підтримки прийняття рішень (СППР) у медичну інформаційну інфраструктуру є критичним аспектом їх ефективного функціонування. На відміну від автономних рішень, СППР повинна працювати у взаємозв'язку з існуючими модулями медичного закладу, не порушуючи при цьому логіки надання послуг і загальної архітектури системи [3, с. 33].

Типова медична інформаційна система (МІС) включає базу пацієнтів, модулі обліку прийомів, електронні історії хвороб, призначення, реєстрацію обстежень, керування розкладом персоналу, фінансові модулі та інші компоненти. У цьому середовищі СППР виконує функцію допоміжного аналітичного модуля, який підключається до вже наявної екосистеми. Його головна мета — розширення функціональності шляхом генерації рекомендацій, підкріплених аналізом клінічної ситуації [1, с. 119].

Зпроектована система HealthAI інтегрується з ключовими елементами медичної інфраструктури: електронною базою даних пацієнтів, модулями обліку прийомів, блоками призначень і довідником медикаментів. Для цього реалізовано доступ до таблиць, що містять інформацію про анамнез, діагнози, алергії, попередні

призначення, результати обстежень, а також дані про розклад роботи медичного персоналу [3, с. 88].

Інтеграція побудована за принципом мінімального втручання в існуючі бізнес-процеси. Система не замінює дії лікаря, а лише формує рекомендаційний висновок, який служить доповненням до клінічного бачення фахівця. Такий підхід узгоджується з принципами доказової медицини, де рішення базується не на авторитарному алгоритмі, а на сукупності клінічних даних, професійного досвіду лікаря та доступної аналітики [20, с. 4].

Типовий сценарій взаємодії виглядає так: лікар отримує або сам формує запит, вводячи опис симптомів, попередніх діагнозів чи клінічної ситуації. Цей запит передається до мовного модуля (GPT-4), який формує попередню відповідь у вигляді зрозумілого тексту. Паралельно відбувається звернення до аналітичного модуля DeepSeek, який виконує пошук релевантної інформації у клінічному корпусі, що дозволяє верифікувати отриману рекомендацію. В результаті лікар отримує об'єднану відповідь, що містить як підсумок GPT-4, так і посилання на джерела або фрагменти даних [7, с. 13].

Архітектурно інтеграція реалізується через API-зв'язки між інтерфейсом користувача, сервером обробки запитів, мовним модулем і модулем аналітики. Серверна частина виступає як проміжний обробник: вона приймає запит, нормалізує його, передає у відповідні сервіси, отримує відповіді та формує фінальний блок для відображення [11, с. 160]. Усі обміни даними виконуються через захищені канали із застосуванням HTTPS-протоколу, токенів доступу та шифрування.

Особливу увагу приділено конфіденційності. Усі дані обробляються локально або в межах закритої серверної інфраструктури медичного закладу. Ні GPT-4, ні DeepSeek не мають прямого доступу до особистих даних пацієнта — запит формується анонімно або у псевдонімізованому вигляді, без передачі персоніфікованої інформації в публічну мережу [21, с. 2].

Крім того, система підтримує збереження історії запитів у межах однієї сесії, що дозволяє лікарю повертатися до попередньої взаємодії без втрати контексту. Такий

механізм підвищує зручність, а також дозволяє вносити уточнення без повторного формулювання запиту.

Таким чином, інтеграція СППР у медичну інформаційну інфраструктуру має бути технічно прозорою, безпечною, інтуїтивно зрозумілою та органічно включеною у робочий процес лікаря. Це дозволяє підвищити ефективність надання медичних послуг, зберігаючи автономність фахівця та гарантуючи захист даних.

1.6 Аналіз сучасного стану сфери охорони здоров'я

Охорона здоров'я є однією з найбільш чутливих та соціально значущих сфер, які потребують систематичного впровадження сучасних інформаційних технологій. Станом на сьогодні галузь характеризується високим рівнем навантаження на медичний персонал, зростаючими вимогами до точності діагностики, потребою в персоналізованому лікуванні, а також необхідністю ефективного управління медичними даними [7, с. 12; 11, с. 2]. У таких умовах особливого значення набуває використання цифрових інструментів, зокрема інтелектуальних систем підтримки прийняття рішень [20, с. 7].

Найбільш поширеними викликами сучасної медичної системи залишаються: нестача часу в лікарів на індивідуальний підхід до кожного пацієнта, обмежений доступ до клінічних рекомендацій, дублювання обстежень, а також фрагментарність медичних даних. Крім того, через значний обсяг інформації, з яким щоденно працює медичний працівник, підвищується ризик помилок у діагностиці та лікуванні [21, с. 1; 22, с. 60].

Паралельно із цим, спостерігається активне зростання обсягів медичних даних: електронних медичних карток, результатів лабораторних обстежень, даних від носимих пристроїв, скарг пацієнтів у цифровому форматі тощо. Традиційні методи обробки інформації стають недостатньо ефективними, що спонукає до впровадження автоматизованих рішень [3, с. 88].

У цьому контексті особливу актуальність набувають системи підтримки прийняття рішень (СППР), які дозволяють покращити процес постановки діагнозу, вибору терапії, інтерпретації результатів та моніторингу динаміки стану пацієнта [11, с. 160].

Однією з таких систем є HealthAI, концептуальне рішення, розроблене в межах цього проєкту. Система HealthAI моделює роль інтелектуального асистента для лікаря. Вона забезпечує доступ до ключових функцій, серед яких: узагальнення інформації про пацієнта, формування короткого зведення на основі даних медичної картки, генерація рекомендацій на основі вбудованої логіки, що враховує наявні захворювання, алергії, історію лікування та актуальні симптоми.

Цифровізація охорони здоров'я в Україні відбувається поступово, однак вже зараз існують приклади інтеграції електронної системи охорони здоров'я (eHealth), впровадження телемедицини, електронних рецептів та хмарних рішень для зберігання медичних даних [21, с. 2]. Разом з цим, інтелектуальні системи залишаються переважно у фазі пілотних проєктів або дослідницьких розробок, що зумовлює актуальність концептуальних моделей, подібних до HealthAI.

Таким чином, аналіз сучасного стану сфери охорони здоров'я свідчить про наявність об'єктивних передумов для впровадження інтелектуальних СППР. Це дозволяє не лише оптимізувати роботу медичного персоналу, але й покращити якість надання медичних послуг через персоналізацію, швидкість доступу до даних та автоматизовану підтримку у прийнятті рішень [7, с. 71].

Висновки до розділу 1

У першому розділі проведено теоретичний огляд систем підтримки прийняття рішень (СППР), їхньої еволюції, типології та сучасних підходів до побудови інтелектуальних інформаційних систем. Особливу увагу приділено аналізу потенціалу використання великих мовних моделей і модулів семантичного пошуку у медичному контексті. Запропоновано концепцію використання інструментів

штучного інтелекту для автоматизації підтримки клінічних рішень. Також проаналізовано поточний стан сфери охорони здоров'я, що дозволяє сформулювати основні передумови для проєктування системи HealthAI. Результати цього розділу слугують теоретичним підґрунтям для подальшої розробки.

РОЗДІЛ 2. УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ СПІР

У цьому розділі представлено повний комплекс заходів з управління проектом створення системи підтримки прийняття рішень HealthAI, яка поєднує технології штучного інтелекту з потребами медичної практики. Оскільки проєкт передбачає реалізацію складної багаторівневої інформаційної системи, особливу увагу приділено питанням планування, організації, розподілу ресурсів, контролю якості та управління ризиками.

Розділ розпочинається з обґрунтування вибору сучасної гнучкої методології управління, адаптованої до динамічного середовища розробки ІТ-рішень із застосуванням ШІ. Далі представлено структуру команди та розподіл ролей, що моделює типову конфігурацію для невеликих ІТ-проєктів. Наведено календарно-ресурсне планування на основі WBS-структури, здійснено розрахунок бюджету з урахуванням трудових і технічних витрат, а також проаналізовано часові межі проєкту.

Окремі підрозділи розглядають питання управління ризиками, забезпечення якості, інформаційної безпеки та документаційної підтримки. Результати управлінського аналізу підкріплено відповідними діаграмами, таблицями, структурними схемами та статутом проєкту. Такий підхід дозволяє забезпечити системність у реалізації проєкту та надає цілісне бачення процесу управління розробкою інтелектуальної системи.

2.1 Обґрунтування вибору методології управління проектом

Управління проєктами, пов'язаними з розробкою програмного забезпечення, особливо тих, де застосовуються технології штучного інтелекту, потребує підходу, що враховує змінність вимог, високий рівень невизначеності та необхідність поступового уточнення цілей. Ураховуючи ці особливості, для реалізації даного

проєкту було обрано гнучку методологію управління Scrum, яка є однією з найбільш поширених реалізацій фреймворку Agile [23, с. 3].

Вибір методології базувався на аналізі специфіки проєкту. Розробка системи підтримки прийняття рішень на основі штучного інтелекту передбачає багатоступеневу побудову функціоналу, можливість інтеграції зовнішніх API, а також тестування результатів на реальних сценаріях. У таких умовах використання каскадної або жорстко структурованої моделі (наприклад, Waterfall) було б неефективним, оскільки вона не передбачає достатньої гнучкості для змін [3, с. 12].

Scrum дозволяє структурувати проєкт як послідовність коротких ітерацій — спринтів, кожен з яких завершується конкретним результатом [24, с. 5]. Це дає змогу регулярно оцінювати проміжні результати, коригувати підхід до реалізації, адаптувати технічні рішення й уточнювати вимоги. Такий формат особливо ефективний у ситуаціях, де кінцевий вигляд системи формується поступово в процесі її створення, як це і відбувається у випадку дипломного проєкту.

Додатковим аргументом на користь вибору Scrum стала можливість гнучко формувати команду, розподіляти ролі й відповідальність, визначати пріоритети завдань на кожному етапі [25, с. 6]. Формалізовані події Scrum, такі як планування спринту, щоденні зустрічі, огляди й ретроспективи забезпечують прозорість процесу, що важливо для ефективного управління проєктом, навіть у навчальному середовищі [11, с. 7].

У межах проєкту Scrum не лише дозволив забезпечити контроль над часом, якістю та обсягом робіт, але й сприяв послідовному розгортанню функціоналу системи з урахуванням реальних обмежень: часових, технічних та ресурсних. Такий підхід відповідає актуальним тенденціям в індустрії розробки програмного забезпечення і є доцільним у контексті реалізації систем, побудованих на використанні штучного інтелекту [3, с. 14].

2.2 Структура команди та розподіл ролей

Організаційна структура команди є важливим елементом управління проектом. Її формування дозволяє не лише впорядкувати виконання завдань, але й забезпечити контроль за окремими напрямками роботи [23, с. 112]. У межах даного проекту була змодельована команда, що складається з типових для ІТ-проектів ролей. Це дозволило відтворити умови, близькі до реального процесу розробки програмного забезпечення [23, с. 4]. До складу команди входять такі ролі: власник продукту, керівник проекту, розробник, тестувальник, дизайнер, системний аналітик, DevOps-спеціаліст, а також фахівець із інформаційної безпеки. Усі ці ролі взаємодіють між собою відповідно до принципів обраної методології Scrum [24, с. 3].

Власник продукту відповідає за формування бачення системи, визначення функціональних вимог і встановлення пріоритетів. Саме він приймає рішення щодо доцільності реалізації певного функціоналу. Керівник проекту координує команду, контролює строки виконання завдань і забезпечує дотримання методології. Розробник реалізує функціональні компоненти системи, інтегрує зовнішні сервіси та працює з API. Тестувальник перевіряє працездатність системи, готує тест-кейси, виявляє помилки та контролює якість реалізації. Дизайнер відповідає за зовнішній вигляд інтерфейсу і зручність взаємодії користувача з системою. Аналітик бере участь у збиранні вимог, описує бізнес-процеси і формує документацію [15].

DevOps-спеціаліст налаштовує середовище розгортання, автоматизує процеси тестування і доставки оновлень. Фахівець із безпеки аналізує ризики, пов'язані з обробкою даних, і контролює дотримання принципів конфіденційності відповідно до вимог сучасних стандартів [3, с. 84]. Виділення ролей є доцільним для моделювання повноцінного життєвого циклу розробки. Це дозволяє краще структурувати роботу, розподілити відповідальність на рівні процесу та оцінити обсяг і складність кожного етапу [26, с. 108]. Така структура є типовою для невеликих проектів з обмеженим бюджетом та часовими рамками. Вона дозволяє покрити всі критичні напрямки роботи без дублювання функцій і без надмірного розширення команди [23, с. 5].

2.3 Статут проєкту

У межах реалізації управлінської частини дипломного проєкту було підготовлено статут проєкту, що визначає мету, обсяг, ключові етапи, повноваження учасників та очікувані результати. Цей документ є вихідною точкою для розробки структурної, часової та ресурсної моделі реалізації проєкту. Зважаючи на обсяг та формальний характер, статут проєкту винесено до додатків (див. Додаток С). Зокрема, організаційна структура команди, представлена у цьому розділі, була розроблена відповідно до положень, визначених у статуті. Вона враховує типові ролі для IT-проєктів та відповідає сценарію реалізації системи на основі гнучкої методології управління.

2.4 SMART-аналіз цілей проєкту та PEST-аналіз СППР для медичних закладів

Для формулювання цілей дипломного проєкту було застосовано метод SMART, який передбачає оцінку цілей за критеріями специфічності, вимірюваності, досяжності, релевантності та обмеженості в часі. Такий підхід дозволяє сформулювати чітке, обґрунтоване та реалістичне бачення очікуваного результату проєкту, що забезпечує його планомірну реалізацію (табл. 2.1)

Табл. 2.3

Цілі дипломного проєкту

Критерій	Формулювання цілі
Specific (специфічність)	Створення прототипу системи підтримки прийняття рішень для медичних закладів з інтеграцією штучного інтелекту, що забезпечує аналіз запитів природною мовою та формування рекомендацій щодо медикаментозного лікування.

Measurable (вимірюваність)	Результат є вимірюваним: система повинна забезпечувати обробку запитів лікарів і формувати відповіді, які можна оцінити за точністю, релевантністю та швидкістю реагування відповідно до заздалегідь встановлених тестових сценаріїв.
Achievable (досяжність)	Проект спрямований на створення мінімально життєздатної версії системи з базовим функціоналом. Для реалізації використовуються існуючі інструменти (OpenAI, DeepSeek, інфраструктура хмарних сервісів), що робить ціль досяжною в умовах наявних ресурсів і часових обмежень.
Relevant (релевантність)	Ціль відповідає завданням дипломної роботи, враховує сучасні технологічні тенденції у галузі штучного інтелекту та має практичну цінність для сфери охорони здоров'я.
Time-bound (обмеження в часі)	Реалізація проєкту запланована на 8 місяців із розбиттям роботи на окремі етапи (спринти), що дозволяє здійснювати контроль за дотриманням графіку виконання завдань та вчасним досягненням цілей.

Застосування SMART-аналізу дозволило визначити ключові параметри цілі дипломного проєкту, встановити реалістичні орієнтири для її досягнення та забезпечити відповідність поставлених завдань сучасним вимогам управління проєктами. Сформована ціль стала основою для подальшого планування етапів розробки системи підтримки (табл. 2.2).

Табл. 2.2

PEST-аналіз

Фактор	Опис впливу на проєкт
Р – Політичні фактори	Законодавче регулювання сфери охорони здоров'я, вимоги до захисту персональних медичних даних, нормативні обмеження на застосування автоматизованих систем для клінічних рішень. Необхідність відповідності GDPR, HIPAA або локальним законам щодо конфіденційності інформації.
Е – Економічні фактори	Витрати на впровадження IT-рішень у медичних установах, обмежене фінансування бюджетних закладів охорони здоров'я. Зацікавленість приватних клінік у підвищенні ефективності роботи та скороченні витрат через використання інтелектуальних систем.

S – Соціальні фактори	Зростання очікувань суспільства щодо якості медичних послуг, актуальність швидкого й обґрунтованого ухвалення рішень. Підвищення довіри до IT-рішень серед молодших поколінь лікарів, водночас обережність та скептицизм серед частини досвідчених фахівців.
T – Технологічні фактори	Активний розвиток штучного інтелекту, великих мовних моделей і медичних аналітичних платформ. Зростання доступності API сервісів для обробки мовних запитів і спеціалізованих медичних баз даних. Використання мобільних додатків і хмарних технологій для медичних інформаційних систем.

2.5 Планування та часові межі проєкту

Проєкт розробки системи підтримки прийняття рішень із використанням інструментів штучного інтелекту передбачає проходження низки логічно пов'язаних етапів. Планування цих етапів дозволило забезпечити послідовність реалізації, раціональне використання ресурсів, контроль за строками виконання та визначення залежностей між завданнями. Основу такого підходу становить розподіл проєктного процесу на п'ять ключових фаз із фіксованими результатами та часовими межами [11, с. 92].

Перший етап — аналітична підготовка — включав вивчення предметної області, аналіз наявних рішень, формування вимог до майбутньої системи та визначення доцільності впровадження інструментів штучного інтелекту. У його межах також було сформовано загальне уявлення про архітектуру рішення та принципи його побудови [3, с. 12].

Другий етап стосувався проєктування функціональних компонентів. На цьому етапі були описані сценарії взаємодії користувача з інтерфейсом, а також розроблено концепцію мінімального життєздатного продукту (MVP), що відповідає інкрементальному підходу розробки. Окрему увагу приділено тому, щоб архітектура системи дозволяла масштабування та доповнення на наступних етапах [3, с. 14].

Третій етап передбачав безпосередню реалізацію функціональності. Відбувалась інтеграція мовної моделі, аналітичного модуля, побудова вебінтерфейсу, налаштування взаємодії із зовнішніми API та створення внутрішньої логіки обробки запитів. Робота велась ітеративно з фіксацією прогресу у межах кожного спринту відповідно до методології Scrum [23, с. 3].

Четвертий етап був присвячений тестуванню розроблених модулів. Він включав перевірку працездатності основних функцій, оцінку стабільності при повторних запитах, аналіз типових сценаріїв використання та ефективності обробки помилок. У разі виявлення проблем вносилися відповідні корективи до логіки або інтерфейсу.

П'ятий завершальний етап охоплював оформлення документації та підготовку висновків. Сформовано технічну та пояснювальну документацію, структуровано результати реалізації, підготовлено графічні ілюстрації, включно з діаграмами компонентів та структурою взаємодії системи [11, с. 117].

Планування етапів здійснювалось із урахуванням обмеженого строку реалізації проекту. Для візуалізації було створено діаграму Ганта, яка дозволяє оцінити залежності між завданнями та час, необхідний на їх виконання. Крім того, розроблений план дав змогу більш ефективно розподілити навантаження в межах кожного спринту, а також забезпечити логічний перехід між фазами розробки (рис. 2.1).

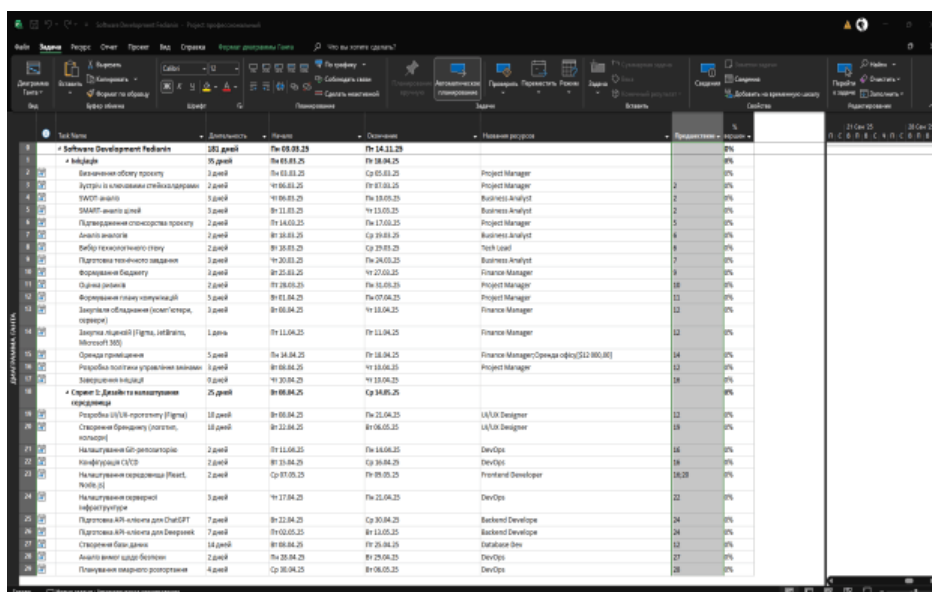
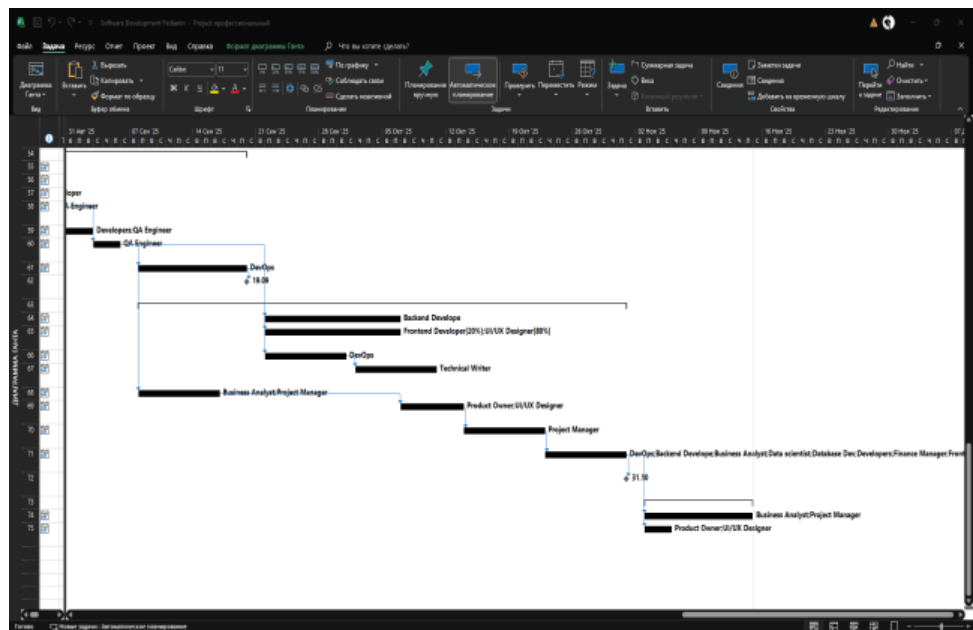
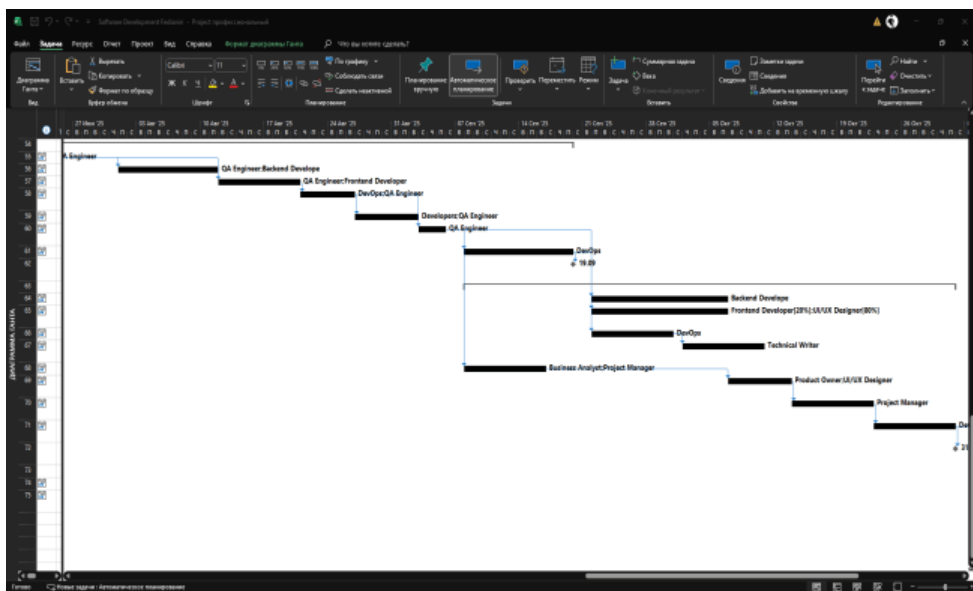
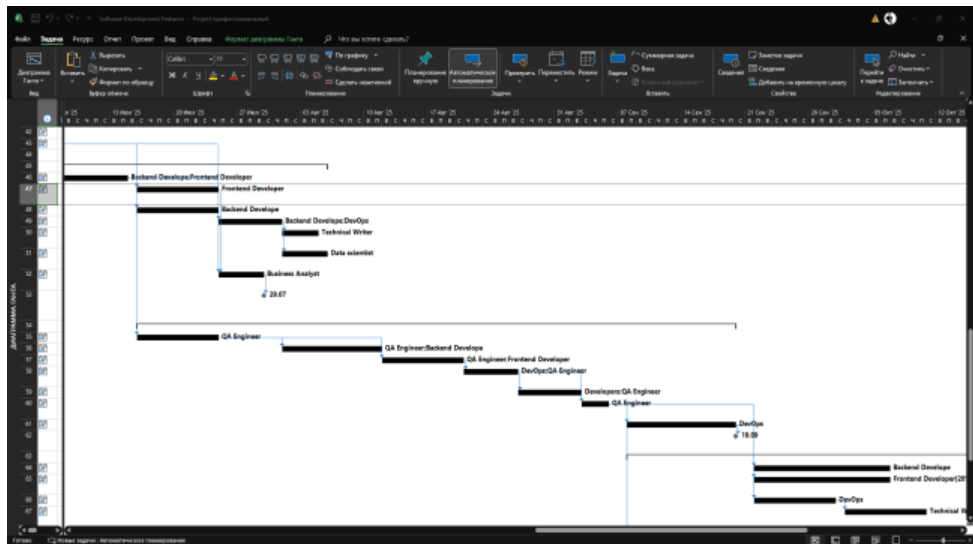


Рис. 2.1 - Планування етапів проекту



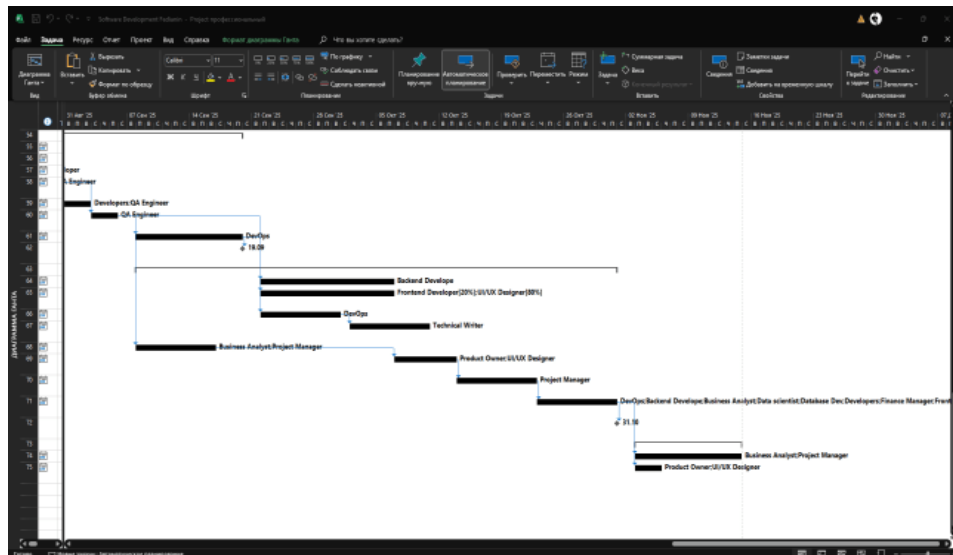


Рис. 2.2 - Діаграма Ганта

2.6 Структура робіт, ресурси та бюджет проєкту

Для реалізації дипломного проєкту, пов'язаного з розробкою системи підтримки прийняття рішень у сфері охорони здоров'я, було створено деталізовану структуру робіт, визначено обсяг необхідних ресурсів, сформовано команду та проведено орієнтовну оцінку бюджету. Ці компоненти стали основою для календарного планування, розподілу відповідальності та обґрунтування витрат [23, с. 4].

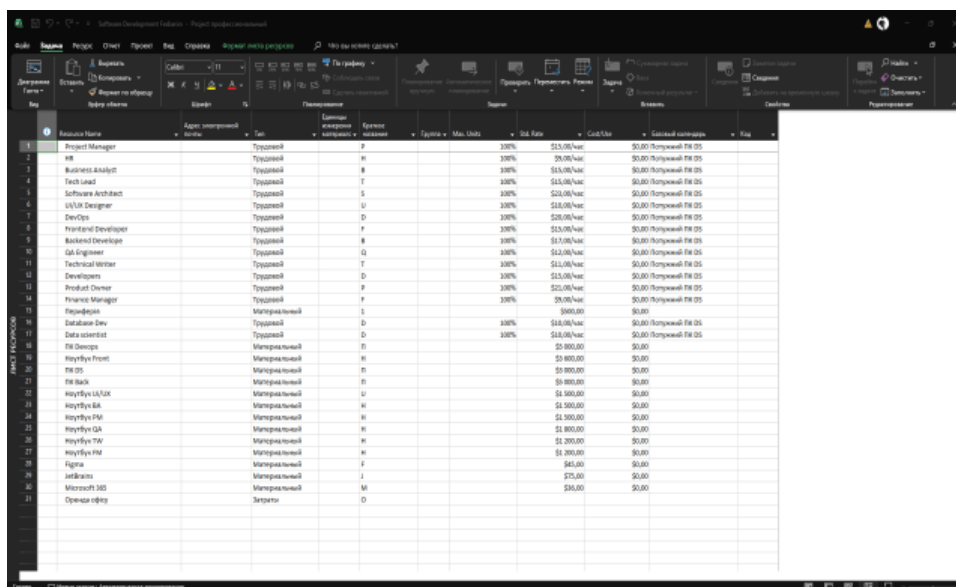
Основним інструментом структуризації завдань стала WBS-структура (Work Breakdown Structure), яка дозволила поділити проєкт на логічні складові та побудувати ієрархічну модель робіт [11, с. 104]. На першому рівні цієї моделі розміщено загальний обсяг робіт — створення СППР з інтеграцією інструментів штучного інтелекту. Другий рівень охоплює п'ять ключових етапів: аналітичну підготовку, проектування, розробку, тестування та завершення проєкту.

Кожен з етапів деталізовано на рівні підзавдань. Зокрема, у межах аналітичного етапу передбачено вивчення предметної області, формування вимог, аналіз існуючих технологій і порівняння альтернатив. Етап проектування охоплює моделювання архітектури, формування технічного завдання, опис інтерфейсів і користувацьких сценаріїв [4, с. 12]. Реалізаційна фаза включає створення основного функціоналу,

інтеграцію з зовнішніми API, розробку інтерфейсу та впровадження логуювання. У межах тестування передбачено перевірку стабільності, функціональності та збір зворотного зв'язку. Завершальний етап включає оформлення документації, пояснювальної записки та графічної частини.

WBS-структура стала основою для формування календарного плану та логіки виконання завдань. Її застосування дозволило забезпечити цілісне бачення проєкту, встановити зв'язки між етапами та покращити керованість процесом розробки [11, с. 117].

Ілюстрація розподілу ресурсів у межах проєкту подана далі (рис. 2.3). Вона відображає умовне навантаження та участь виконавців на різних етапах [23, с. 5].



Resource Name	Agency/Company	Type	Capacity	Current Allocation	Group	Max Units	Est. Rate	Cost/Day	Current Allocation	Est. Cost
Project Manager	Трудовой	Р	100%				\$15,000/мес	\$0,00	Полный	РМ 05
HR	Трудовой	И	100%				\$9,000/мес	\$0,00	Полный	РМ 05
Business Analyst	Трудовой	И	100%				\$15,000/мес	\$0,00	Полный	РМ 05
Tech Lead	Трудовой	Т	100%				\$15,000/мес	\$0,00	Полный	РМ 05
Software Architect	Трудовой	С	100%				\$10,000/мес	\$0,00	Полный	РМ 05
UI/UX Designer	Трудовой	Д	100%				\$10,000/мес	\$0,00	Полный	РМ 05
DevOps	Трудовой	Д	100%				\$10,000/мес	\$0,00	Полный	РМ 05
Frontend Developer	Трудовой	Р	100%				\$15,000/мес	\$0,00	Полный	РМ 05
Backend Developer	Трудовой	И	100%				\$15,000/мес	\$0,00	Полный	РМ 05
QA Engineer	Трудовой	Д	100%				\$10,000/мес	\$0,00	Полный	РМ 05
Technical Writer	Трудовой	Т	100%				\$10,000/мес	\$0,00	Полный	РМ 05
Developer	Трудовой	Д	100%				\$15,000/мес	\$0,00	Полный	РМ 05
Product Owner	Трудовой	Р	100%				\$10,000/мес	\$0,00	Полный	РМ 05
Finance Manager	Трудовой	Р	100%				\$9,000/мес	\$0,00	Полный	РМ 05
Hardware	Материальный	И					\$800,00	\$0,00		
Database Dev	Трудовой	Д	100%				\$15,000/мес	\$0,00	Полный	РМ 05
Data Scientist	Трудовой	Д	100%				\$10,000/мес	\$0,00	Полный	РМ 05
PM Devops	Материальный	И					\$1 800,00	\$0,00		
Июль Front	Материальный	И					\$1 800,00	\$0,00		
PM DS	Материальный	И					\$1 800,00	\$0,00		
PM Back	Материальный	И					\$1 800,00	\$0,00		
Июль UI/UX	Материальный	И					\$1 500,00	\$0,00		
Июль BA	Материальный	И					\$1 500,00	\$0,00		
Июль PM	Материальный	И					\$1 500,00	\$0,00		
Июль QA	Материальный	И					\$1 800,00	\$0,00		
Июль TW	Материальный	И					\$1 200,00	\$0,00		
Июль HD	Материальный	И					\$1 200,00	\$0,00		
Signa	Материальный	И					\$45,00	\$0,00		
Июль	Материальный	И					\$75,00	\$0,00		
Microsoft 365	Материальный	И					\$30,00	\$0,00		
Прочие офис	Запчасти	Д						\$0,00		

Рис. 2.3 - Розподіл ресурсів у межах проєкту

Структура робіт стала основою для розрахунку ресурсного навантаження. У межах проєкту сформовано умовну команду, яка охоплює ключові ролі, необхідні для розробки систем на основі штучного інтелекту. До складу команди входять Project Manager, HR Specialist, Business Analyst, Tech Lead, Software Architect, UI/UX Designer, DevOps Engineer, Frontend Developer, Backend Developer, QA Engineer, Technical Writer, General Developer, Product Owner, Finance Manager, Database Developer, Data Scientist. Така деталізація відповідає сучасним практикам у сфері IT-проєктів зі складною аналітикою [26, с. 108]. Крім людських ресурсів, ураховано

також витрати на інструменти та інфраструктуру. До них належать ліцензії для програмного забезпечення (OpenAI API, DeepSeek, Figma, JetBrains, Microsoft 365) та витрати на офісну інфраструктуру [3, с. 84]. Оцінку загального бюджету наведено у (табл. 2.4), що містить розрахунок заробітної плати, щомісячних витрат та підсумкової вартості.

Табл. 2.4

Ресурсне забезпечення та бюджет проекту

№	Стаття витрат	Кількість	Місячна вартість, \$	Загальна вартість за 8 місяців, \$
1	Project Manager	1	2500	20 000
2	HR Specialist	1	1200	9600
3	Business Analyst	1	2000	16 000
4	Tech Lead	1	3500	28 000
5	Software Architect	1	3300	26 400
6	UI/UX Designer	1	1800	14 400
7	DevOps Engineer	1	2500	20 000
8	Frontend Developer	1	2500	20 000
9	Backend Developer	1	2700	21 600
10	QA Engineer	1	1500	12 000
11	Technical Writer	1	1400	11 200
12	General Developer	1	2300	18 400
13	Product Owner	1	3000	24 000
14	Finance Manager	1	2200	17 600
15	Database Developer	1	2500	20 000
16	Data Scientist	1	2700	21 600
17	Ліцензії OpenAI, DeepSeek, Figma, JetBrains, Microsoft 365	-	400 / міс	3200
18	Оренда офісу	-	800 / міс	6400

Загальна вартість проекту за 8 місяців становить 327 400 \$.

Загальний бюджет проєкту за вісім місяців становить 327 400 доларів США. Більшість витрат припадає на оплату праці спеціалістів високої кваліфікації, що є виправданим з огляду на складність технічного завдання. Інші статті витрат, зокрема ліцензії та офіс, мають допоміжний характер, однак є необхідними для безперебійного функціонування процесу розробки.

Таким чином, сформована структура робіт, у поєднанні з ресурсним плануванням і фінансовими розрахунками, забезпечує основу для реалізації проєкту, дозволяє відстежувати навантаження, контролювати строки виконання і гарантувати досягнення поставлених цілей з урахуванням заданих обмежень.

2.7 Управління ризиками та якістю проєкту

Управління ризиками та якістю це невід’ємна частина проєктної діяльності, особливо у сфері розробки складних ІТ-систем, що передбачають інтеграцію інструментів штучного інтелекту. У межах дипломного проєкту ці напрями розглядалися як системно пов’язані процеси, що забезпечують стійкість реалізації, відповідність очікуваним результатам та можливість адаптації до зовнішніх і внутрішніх змін [25].

На початковому етапі проєкту було здійснено якісний аналіз ризиків, що дозволив виявити потенційні загрози, класифікувати їх та визначити стратегії реагування. Виявлені ризики умовно поділяються на технічні, організаційні та зовнішні. До технічних загроз належать можливі порушення в роботі сторонніх сервісів, на яких ґрунтується функціональність системи, зокрема, API ChatGPT і DeepSeek. Нестабільна робота або зміни у доступності зазначених сервісів можуть ускладнити інтеграцію чи зумовити помилки у відповідях. Крім того, враховувався ризик некоректної обробки складних або неоднозначно сформульованих запитів користувача.

Організаційні ризики стосуються можливих затримок у виконанні етапів проєкту через неточну оцінку обсягу робіт, недостатню деталізацію вимог або ризик

перевантаження функціоналом, що виходить за межі MVP. Подібні фактори можуть призвести до втрати фокусу та ускладнити керування завданнями [25].

Серед зовнішніх ризиків розглянуто зміну умов доступу до інструментів, юридичні або ліцензійні обмеження, а також дефіцит ресурсів — як часових, так і технічних. В освітньому контексті реалізації проєкту цей аспект набуває особливої актуальності.

Для кожного ризику передбачено базові стратегії реагування. Технічні ризики мінімізуються шляхом резервного тестування інтеграцій, використання модульної архітектури та ізоляції зовнішніх точок доступу. Організаційні ризики пом'якшуються через ітеративний підхід до планування, з періодичним переглядом пріоритетів і обсягів робіт у межах спринтів. Зовнішні ризики враховуються через гнучкість вимог, наявність спрощених сценаріїв і здатність до часткової деградації функціоналу без втрати працездатності [23, с. 5; 25].

Паралельно з управлінням ризиками реалізовувалась система управління якістю. Її основною метою було забезпечення відповідності реалізованої системи заданим функціональним та експлуатаційним вимогам. Планування якості передбачало фіксацію базових критеріїв прийнятності: стабільність роботи, коректність обробки запитів, інтеграція з зовнішніми сервісами та відповідність інтерфейсу технічним специфікаціям [3; 25].

Контроль якості здійснювався шляхом ручного тестування, перевірки типових сценаріїв, аналізу логів, а також верифікації роботи ключових модулів. Основна увага приділялася точності обробки запитів, логіці взаємодії компонентів і правильності відповідей, згенерованих мовною моделлю. Особливо важливим було відстеження зв'язків між модулями, передача параметрів та обробка помилок [11; 7].

Ітеративний підхід до розробки, обраний як базовий у межах проєкту (Scrum), забезпечив можливість поступового уточнення вимог, реалізації функціоналу, перевірки результатів і оперативного внесення змін. Це дозволило своєчасно реагувати на відхилення, а також підвищити загальну керованість процесом [23].

Для потреб моніторингу була впроваджена система локальної оцінки проміжних результатів, що дозволило враховувати не лише технічну відповідність, а й показники якості використання: зручність інтерфейсу, стабільність взаємодії, логічну повноту відповідей і точність обробки нестандартних ситуацій.

Таким чином, реалізовані підходи до управління ризиками та якістю дозволили підвищити стійкість і передбачуваність розробки, забезпечити відповідність результатів очікуванням та сформуванню основи для подальшого розширення функціональності без втрати контрольованості процесу.

Висновки до розділу 2

Другий розділ присвячено опису організаційних та управлінських підходів до реалізації проекту створення СППР із застосуванням інструментів штучного інтелекту. Обґрунтовано доцільність використання гнучкої методології Scrum, сформовано уявну команду розробки та запропоновано календарно-ресурсне планування. Визначено очікувані етапи життєвого циклу проекту, структуру робіт, склад бюджету та можливі ризики. Розроблені моделі управління можуть бути використані як основа для майбутньої реалізації проекту. Усі представлені рішення мають теоретичний характер і потребують подальшої практичної апробації.

РОЗДІЛ 3. ВИКОНАННЯ ТА КОНТРОЛЬ РЕАЛІЗАЦІЇ ПРОЄКТУ

Третій розділ містить опис ключових етапів розробки системи підтримки прийняття рішень HealthAI, спрямованої на підтримку медичних працівників у процесі постановки діагнозів та формування рекомендацій. У цьому розділі послідовно розглянуто архітектурні рішення, логіку функціонування компонентів, особливості інтеграції інструментів штучного інтелекту, а також схеми взаємодії між користувачем, мовною моделлю та аналітичним модулем. Наведено графічні схеми, таблиці, а також опис процесів, що формують основу прототипу. Розділ також охоплює аспекти безпеки, обробки даних і тестування, що є критично важливими в контексті практичного застосування системи в охороні здоров'я.

3.1 Загальний опис архітектури системи

Архітектура системи підтримки прийняття рішень, реалізованої в межах цього дипломного проєкту, ґрунтується на використанні хмарних сервісів, зовнішніх API та власного інтерфейсу взаємодії з користувачем. Структура системи побудована за принципами модульності, що дозволяє забезпечити незалежність окремих компонентів, спрощення інтеграції та подальшу масштабованість [27, с. 88].

Загальна архітектура включає три основні рівні: інтерфейс взаємодії з користувачем, логіку обробки запитів та зовнішні інструменти штучного інтелекту. Користувач формулює запит у природній мові через інтерфейс, після чого система передає його на обробку до мовної моделі ChatGPT. Цей компонент забезпечує синтаксичний і семантичний аналіз запиту, формує відповідь або уточнює контекст за потреби [10].

У разі, якщо запит вимагає пошуку додаткової інформації, активується модуль аналітики, що працює через API сервісу DeepSeek. Такий розподіл ролей між

компонентами відповідає принципу побудови сучасних гібридних СППР, які поєднують мовну взаємодію з аналітичними інструментами [8, с. 13].

Важливою особливістю архітектури є використання асинхронної обробки запитів. Це дає змогу підвищити швидкість реакції системи та забезпечити стабільність роботи навіть при високому навантаженні. Зв'язок між модулями реалізований за допомогою REST API-запитів з використанням стандартних методів передачі даних [3, с. 241].

Зберігання історії запитів і результатів взаємодії здійснюється локально, у форматі журналу запитів, що дозволяє зберігати контекст та повторно звертатися до раніше отриманих відповідей. Такий підхід забезпечує не лише зручність для користувача, а й можливість подальшого аналізу ефективності роботи системи [11, с. 174].

У межах архітектурного рішення враховано базові вимоги до безпеки: відсутність зберігання конфіденційних даних, обмеження обсягу журналу, а також контроль стабільності зовнішніх викликів [28, с. 203]. Передбачено також можливість заміни або розширення функціональних компонентів без порушення роботи системи в цілому, що відповідає принципам розширюваності [4, с. 261].

Загалом архітектура побудована таким чином, щоб система залишалась простою в реалізації, але при цьому достатньо гнучкою для виконання ключових функцій та подальшого розвитку [29, с. 154].

3.2 Діаграма потоків даних системи

З метою формалізації логіки функціонування проектованої системи підтримки прийняття рішень було побудовано діаграми потоків даних (DFD). Цей підхід дозволяє описати функціональні процеси, потоки інформації та зв'язки між зовнішніми користувачами, логікою системи й базою даних [30, 16]. Для демонстрації ефекту від впровадження СППР у межах проекту реалізовано порівняння двох версій функціональної архітектури: до та після інтеграції інтелектуального модуля.

історії, розкладом (рис. 3.2). Такий підхід зменшує складність, підвищує масштабованість і покращує логіку обробки даних [15].

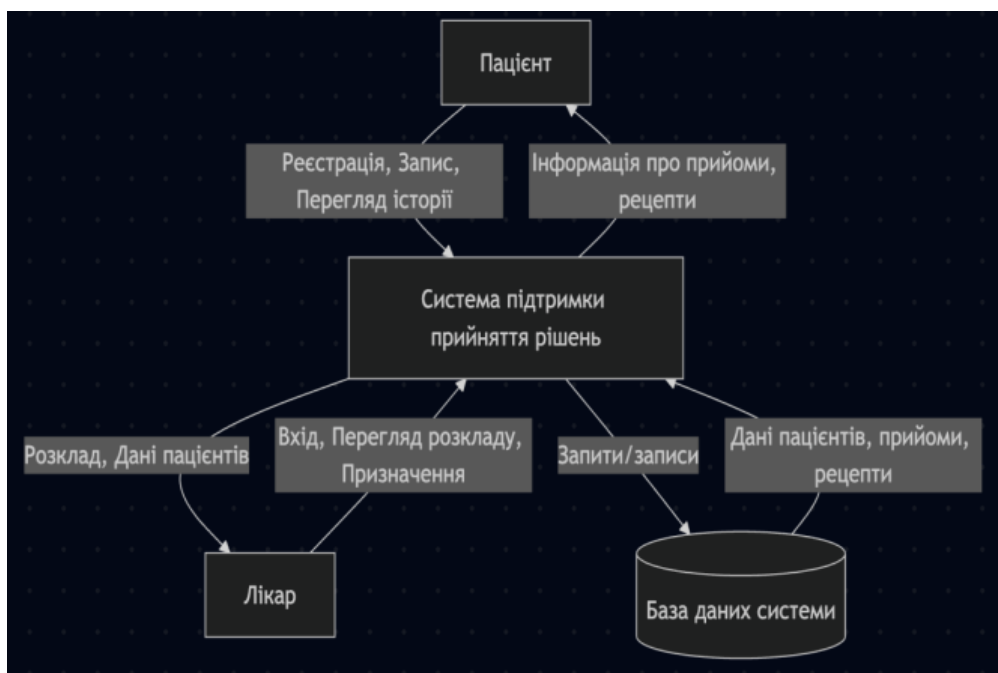


Рис. 3.2 Структура системи після впровадження СППР

Порівняльний аналіз і висновок. Впровадження системи підтримки прийняття рішень дозволило оптимізувати логіку системи, зменшити кількість взаємозв'язків між компонентами та забезпечити централізовану обробку інформації. Це спростило навігацію для кінцевих користувачів, зменшило ймовірність помилок та створило основу для подальшої автоматизації клінічних процесів.

Таким чином, побудовані діаграми потоків даних (DFD) не лише описують поточну структуру, а й ілюструють переваги впровадження СППР у вигляді HealthAI, з позиції функціональної цілісності, спрощеної взаємодії та покращеної масштабованості.

3.3 Діаграма діалогових вікон

Для забезпечення логічної послідовності переходів між екранами користувацького інтерфейсу було побудовано діаграму діалогових вікон. Ця діаграма

відображає структуру навігації в межах системи підтримки прийняття рішень та визначає основні сценарії використання [2], [11]. Такий підхід дозволяє формалізувати взаємозв'язки між інтерфейсними модулями та оптимізувати розробку користувацького досвіду [15].

У якості початкового етапу передбачено блок входу або реєстрації. Користувач може пройти аутентифікацію за допомогою номера телефону, BankID або зареєструвати нового користувача. У разі помилки автентифікації система надає відповідне повідомлення [6].

Після входу користувач потрапляє на головну сторінку, яка є точкою доступу до функціональних підсистем. Вона містить посилання на профіль користувача, медичні послуги, інформаційні розділи та системні повідомлення [41]. У розділі профілю доступна медична карта, результати аналізів та вихід із системи. Медична карта включає електронну картку, де зберігаються умови обстежень та дані про вакцинацію. Додатково передбачено можливість завантаження довідки [27].

Система підтримує пошук лікаря або клініки з використанням фільтрів за спеціальністю та містом. Перегляд лікаря надає змогу записатися на прийом або ознайомитися з відгуками. Під час запису користувач обирає час та підтверджує запис. Після цього передбачено підтвердження через SMS або можливість скасування чи перенесення прийому [21].

У розділі особистих даних зберігається інформація про користувача та історія записів. Передбачено доступ до призначень та результатів лікування. Розділ повідомлень містить нові призначення, результати обстежень та повідомлення від лікаря.

Системні повідомлення охоплюють типові ситуації, що пов'язані з відсутністю з'єднання, завершенням сесії або підтвердженням дій. У разі помилки сервера система також надає відповідне сповіщення (рис. 3.3).

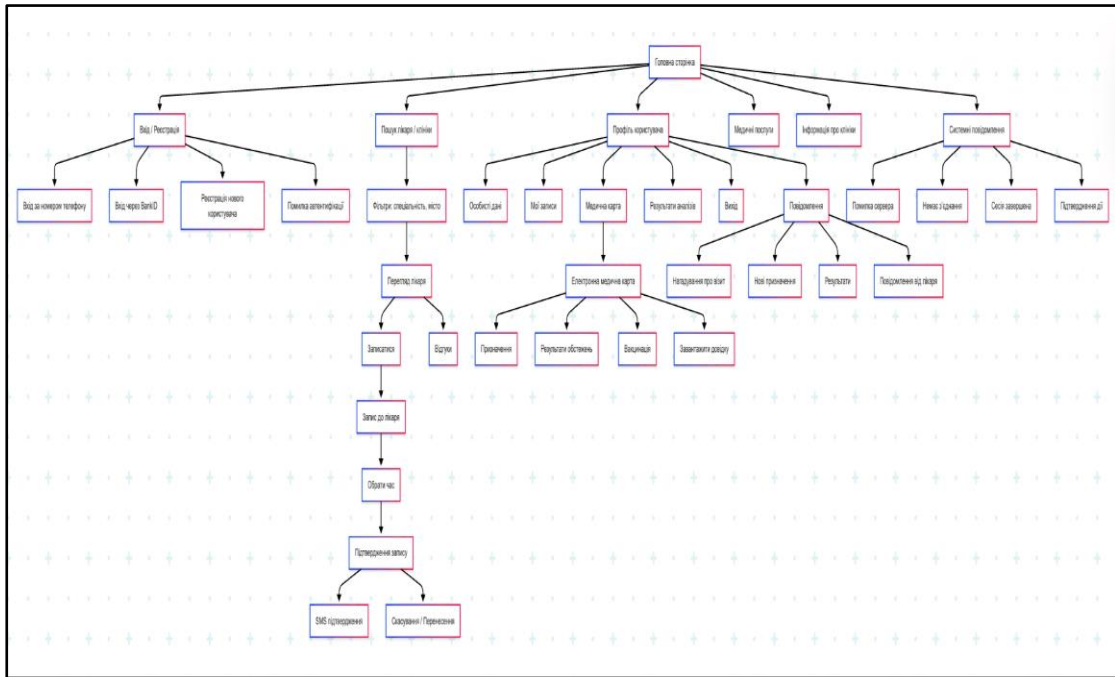


Рис. 3.3 – Діаграма діалогових вікон користувацького інтерфейсу

Діаграма дає змогу представити логіку побудови інтерфейсу у вигляді деревоподібної структури. Вона описує допустимі шляхи навігації та дозволяє забезпечити відповідність між очікуваннями користувача та фактично реалізованими переходами. Наявність повної карти вікон слугує основою для проектування інтерфейсної логіки та проведення тестування взаємодії [31], [32]. Детальні зображення діаграми наведені у Додатку В.

3.4 Діаграма класів

Для опису об'єктно-орієнтованої структури програмного забезпечення системи підтримки прийняття рішень побудовано діаграму класів [42]. Цей тип діаграми дає змогу формалізувати модель предметної області через визначення сутностей, їхніх атрибутів, методів та взаємозв'язків між ними [1], [2]. Концептуальна структура сформована на основі функціональних вимог, відображених у діаграмі діалогових вікон [15].

Центральним класом системи є **User**. Він містить базові атрибути: номер телефону, ознаку автентифікації та персональні дані. Основні методи включають вхід

у систему, вихід із системи та оновлення профілю. Клас **User** має прямий зв'язок із декількома іншими сутностями.

Користувач має можливість створювати один або декілька записів на прийом, які представлені класом **Appointment**. Цей клас містить атрибути дати, часу та статусу. Додатково реалізовано методи підтвердження, скасування або перенесення прийому. Прийом пов'язано з класом **Doctor**, який зберігає інформацію про спеціалізацію, місто та рейтинг лікаря. Через клас **Clinic** здійснюється доступ до назви клініки, адреси та переліку послуг.

Користувач також має зв'язок із класом **Notification**. Цей клас відповідає за тип та зміст повідомлень і має метод для їх надсилання. Схожим чином реалізовано клас **SystemMessage**, що використовується для відображення системних сповіщень.

Окремий підмодуль стосується класу **MedicalRecord**. У ньому зберігається перелік призначень, результатів обстежень та записів про вакцинацію. Клас містить метод для завантаження довідки. Кожен запис у **MedicalRecord** реалізується як список елементів, пов'язаних із класами **Prescription**, **TestResult** та **Vaccination**. Призначення містить дані про медикамент, дозування та тривалість. Результати тестів містять назву, результат та дату проведення. Запис про вакцинацію включає назву вакцини та дату (рис. 3.4).

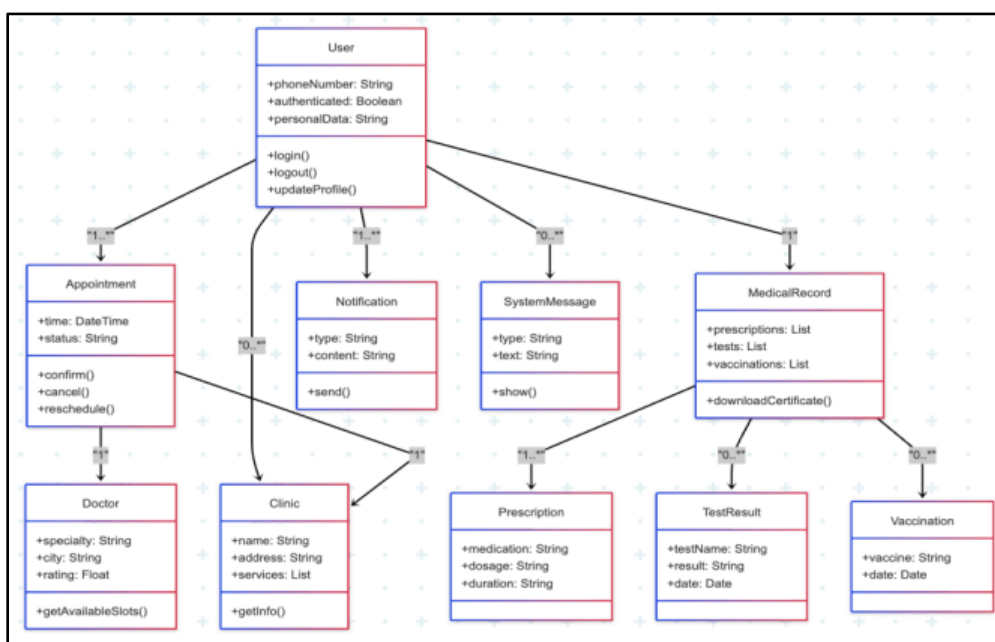


Рис. 3.4 – Діаграма класів системи

Усі зв'язки реалізовано за допомогою множинностей, що описують кількість допустимих зв'язків між об'єктами [43]. Зв'язки типу один-до-багатьох використовуються для зберігання списків записів і сповіщень, а зв'язки один-до-одного застосовано у випадках персоналізованої інформації.

Побудована діаграма класів дозволяє визначити основні програмні компоненти та їх взаємодію. Вона служить основою для реалізації програмної логіки в об'єктно-орієнтованому стилі та забезпечує підтримку узгодженості структури даних з вимогами до інтерфейсу користувача [4], [27].

З огляду на представлену структуру об'єктів, реалізацію програмної частини системи доцільно здійснювати з використанням мови програмування Python. Ця мова забезпечує підтримку об'єктно-орієнтованого підходу, має розвинену екосистему для обробки медичних та статистичних даних і містить вбудовані засоби для взаємодії з базами даних, інтерфейсами та мережевими запитами [10], [9], [32].

Наявність бібліотек `pandas`, `sqlite3`, `flask`, `tkinter`, `fastapi`, `scikit-learn` та інших дозволяє реалізувати повноцінну серверну, аналітичну й клієнтську логіку в межах єдиного проєкту.

Як програмне середовище для розробки може бути використано Spyder, яке інтегрується з Python та підтримує багатовіконний режим для роботи з кодом, терміналом, графіками та базами даних [44]. Це середовище орієнтоване на наукові та інженерні задачі, тому відповідає вимогам до розробки системи, що використовує структуровані медичні дані та реалізує логіку обробки запитів у режимі симуляції або тестування [26]. Альтернативними середовищами можуть бути VS Code, PyCharm Community або JupyterLab, які також підтримують проєктування у стилі класів [3].

Таким чином, поєднання Python як мови реалізації та Spyder як середовища розробки забезпечує відповідність функціональним вимогам системи, структурі класів та можливості масштабування проєкту відповідно до поставлених задач [11].

3.5 Інтеграція мовної моделі ChatGPT та модуля аналітики DeepSeek

Інтеграція інструментів штучного інтелекту в систему підтримки прийняття рішень реалізується через поєднання двох ключових компонентів — мовної моделі ChatGPT та модуля аналітики DeepSeek. Такий підхід дозволяє поєднати переваги генеративного ШІ з фактологічною перевіркою, що є особливо важливим у медичному контексті, де кожна відповідь повинна бути обґрунтованою, структурованою та підтвердженою даними.

Інтеграція ChatGPT здійснюється з метою забезпечення взаємодії з користувачем у форматі природної мови [45]. Модель виступає як обробник запитів першого рівня, що генерує текстові відповіді на основі контексту та запиту користувача [19; 14, с. 274]. Завдяки цій інтеграції система здатна інтерпретувати неконкретизовані або частково структуровані формулювання та формувати узгоджені, читабельні відповіді [26, с. 5].

Технічно взаємодія реалізована через REST API, який надає OpenAI. Запити формуються у форматі JSON і містять текст користувача, параметри генерації (температура, довжина, історія повідомлень тощо), після чого передаються до серверної частини. Відповіді повертаються у вигляді текстових фрагментів, які або одразу відображаються користувачу, або передаються далі для уточнення та доповнення [11, с. 102].

Система підтримує збереження контексту сесії — кожен новий запит надсилається разом з історією попередніх діалогів, що дозволяє моделі краще розуміти логіку звернення. Це особливо важливо в медичних застосуваннях, де запити часто уточнюються чи мають форму послідовної діагностичної бесіди [4].

У проєкті реалізовано також механізм кешування відповідей — з метою зменшення навантаження на API, повторні звернення з ідентичним формулюванням опрацьовуються локально. Такий підхід дозволяє дотримуватися обмежень на кількість запитів та покращує швидкодію [10].

Загалом, інтеграція ChatGPT значно спрощує взаємодію користувача з системою, дозволяє відмовитися від складних інтерфейсів і знижує поріг входу для медичних працівників без технічної підготовки.

Модуль DeepSeek використовується як інструмент семантичного пошуку і фактологічної перевірки [46]. Його мета — надання релевантної, структурованої інформації, яка підкріплює або уточнює відповідь мовної моделі [11, с. 209; 10]. Технічно реалізація відбувається через API DeepSeek, який приймає запит із ключовими словами або семантичним вектором і повертає текстові фрагменти, джерела, цитати та узагальнення. Ці дані або виводяться як додаток до відповіді ChatGPT, або інтегруються у структуру самої відповіді як фактична база [4]. Після генерації первинної відповіді ChatGPT система формує уточнений запит до DeepSeek. Це дозволяє підтримувати послідовність логіки та уникати семантичних суперечностей. Результати проходять фільтрацію за довжиною, релевантністю та наявністю підтверджених джерел [14].

Система підтримує повторне використання результатів попередніх пошуків, що знижує навантаження на API і пришвидшує відповідь на повторні запити. Крім того, це дозволяє накопичувати корпус перевірених відповідей і формувати основу для локальної бази знань.

Застосування DeepSeek суттєво підвищує інформативність системи: користувач не лише отримує відповідь, а й бачить, на які джерела вона спирається [47]. Це особливо важливо в медичних задачах, де верифікованість інформації є критичною.

Інтеграція ChatGPT і DeepSeek реалізує комбінований підхід: генерація природномовної відповіді на основі діалогу — та її підтвердження або доповнення через аналітичний модуль. Такий підхід поєднує зручність спілкування з системою та точність клінічного змісту. Це дозволяє системі HealthAI не лише імітувати діалог лікаря з асистентом, а й забезпечити реальну підтримку у процесі прийняття рішень на основі перевірених фактів.

3.6 Опис структури бази даних

У процесі реалізації системи підтримки прийняття рішень було створено реляційну базу даних. Її структура побудована на основі принципів логічної ізоляції функціональних областей [1]. Основною метою є забезпечення централізованого зберігання даних, що використовуються у медичному обслуговуванні. Архітектура охоплює ключові об'єкти, зокрема пацієнтів, лікарів, записи на прийом, розклади, медикаменти та призначення [27].

База даних включає шість взаємопов'язаних таблиць: **PATIENTS**, **DOCTORS**, **APPOINTMENTS**, **DOCTOR_SCHEDULE**, **MEDICATIONS** і **PRESCRIPTIONS**. Кожна таблиця має первинний ключ для забезпечення унікальності записів. Зовнішні ключі використовуються для реалізації зв'язків між сутностями. Наприклад, у таблиці **APPOINTMENTS** зберігається інформація про прийоми з прив'язкою до пацієнтів і лікарів. Таблиця **PRESCRIPTIONS** пов'язана одночасно з конкретними прийомами та призначеними медикаментами [26]. Розроблена модель бази даних відповідає наведеній нижче логічній схемі зв'язків між таблицями (рис. 3.5). Вона демонструє типи атрибутів, напрямки зв'язків та ідентифікатори залежностей [3].

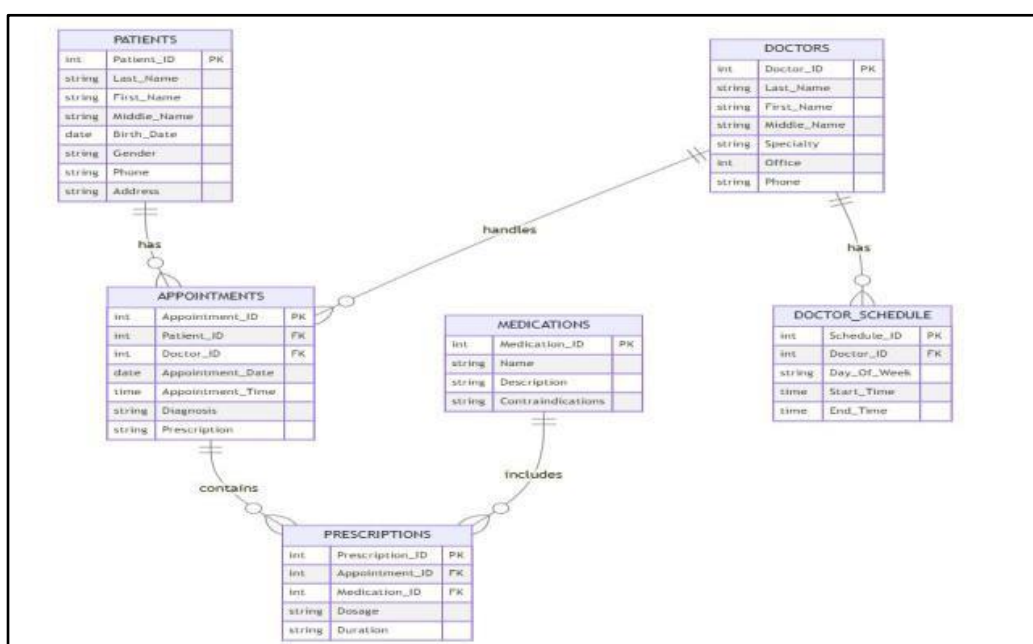


Рис. 3.5 – Логічна структура бази даних системи

Фізична реалізація структури виконана мовою програмування Python із використанням бібліотеки sqlite3. Створено окремий скрипт, у якому визначено SQL-команди для формування кожної таблиці, встановлення зовнішніх ключів і наповнення тестовими даними [48]. Наприклад, таблиця PATIENTS створюється наступним чином:

```
□ cursor.execute("""  
CREATE TABLE IF NOT EXISTS PATIENTS (  
    Patient_ID INTEGER PRIMARY KEY AUTOINCREMENT,  
    Last_Name TEXT,  
    First_Name TEXT,  
    Middle_Name TEXT,  
    Birth_Date TEXT,  
    Gender TEXT,  
    Address TEXT,  
    Phone TEXT  
)  
""")
```

□ Аналогічним чином реалізовано всі інші таблиці. Для перевірки вмісту бази даних застосовано SQL-запити, результати яких виводяться у табличному форматі за допомогою бібліотеки prettytable. Це дозволило здійснити тестування логіки функціонування моделі, перевірити зв'язки та оцінити коректність збереження даних [2]. Взаємозв'язки між таблицями відповідають правилам нормалізації (табл. 3.1). Зокрема:

- один пацієнт може мати багато записів у таблиці APPOINTMENTS;
- один лікар може обслуговувати багато прийомів і мати декілька елементів розкладу;
- кожен прийом може містити декілька призначень;

- кожне призначення пов'язане з конкретним медикаментом.

Табл. 3.1

Взаємозв'язки між таблицями бази даних

Таблиця 1	Тип зв'язку	Таблиця 2	Опис
PATIENTS	$1 \rightarrow \infty$	APPOINTMENTS	Один пацієнт може мати багато прийомів
DOCTORS	$1 \rightarrow \infty$	APPOINTMENTS	Один лікар обслуговує багато прийомів
DOCTORS	$1 \rightarrow \infty$	DOCTOR_SCHEDULE	Один лікар має декілька розкладів
APPOINTMENTS	$1 \rightarrow \infty$	PRESCRIPTIONS	Один прийом містить багато призначень
PRESCRIPTIONS	$\infty \rightarrow 1$	MEDICATIONS	Кожне призначення пов'язане з медикаментом

Опис атрибутів кожної таблиці наведено у Додатку А. Ці таблиці відображають типи даних, що зберігаються, а також структуру іменування полів у базі даних. Використана модель забезпечує підтримку цілісності даних і можливість масштабування проєкту. Вона створює передумови для подальшої інтеграції з аналітичними модулями або системами обробки запитів [32].

3.7 Тестування системи

Тестування важливий етап розробки інформаційної системи, що дозволяє перевірити її відповідність заданим функціональним вимогам, виявити потенційні помилки, оцінити стабільність роботи та підготувати систему до подальшої інтеграції [11, с. 160]. В умовах концептуального характеру даного проєкту тестування перебуває на етапі планування. На цей момент проведено підготовку інфраструктури, описано ключові сценарії перевірки, визначено інструменти та сформульовано підходи до реалізації. Передбачається використання змішаного підходу, який поєднує ручне та автоматизоване тестування. Ручне тестування буде спрямоване на перевірку

логіки взаємодії з інтерфейсом, реакції системи на типові та нетипові запити, а також візуалізації результатів [14, с. 48]. Автоматизоване тестування використовуватиметься для перевірки обробки API-запитів, правильності структури відповідей, а також забезпечення повторюваності результатів у стандартних умовах [7, с. 71]. Процес тестування передбачає використання локального серверного середовища, клієнтської частини вебінтерфейсу, тестової бази даних (SQLite), сервісів GPT-4 та DeepSeek, а також допоміжних засобів, таких як Postman, PyTest і JupyterLab [21, с. 2]. Це дозволяє виконувати перевірки як на рівні окремих компонентів, так і у межах комплексної інтеграції [20, с. 6]. У межах підготовки сформульовано низку тестових сценаріїв.

Табл. 3.2

Приклад кейсу

Назва кейсу	Перевірка генерації відповіді GPT-4 на типовий медичний запит
Вхідні дані	Запит: «Пацієнт має кашель, лихоманку, втому. Який діагноз можливий?»
Очікувана поведінка	Система формує відповідь мовною моделлю, яка містить імовірний діагноз та пояснення
Додатково	Перевіряється узгодженість відповіді з контекстом та логічна структурованість формулювання
Критерій успішності	Відповідь є змістовною, коректно побудованою та придатною для використання лікарем

Під час реалізації повного процесу тестування планується також врахувати перевірку обробки виняткових ситуацій, зокрема — реакції системи на порожні, надмірно складні або суперечливі запити [33]. Особливу увагу буде приділено перевірці стабільності взаємодії з API, підтримці логування запитів і збереженню історії сесій [3, с. 84].

На момент написання роботи результати тестування ще не отримані, однак визначена структура дозволяє забезпечити послідовність і контрольованість перевірки [49]. Очікується, що заплановані перевірки підтвердять працездатність

основних компонентів та виявлять окремі напрямки для подальшої оптимізації, зокрема в частині обробки помилок та пояснення відповідей [15].

Таким чином, система тестування є ключовим інструментом забезпечення якості розробленого прототипу та закладає основу для його вдосконалення у наступних ітераціях реалізації.

Висновки до розділу 3

У третьому розділі описано концептуальну архітектуру системи підтримки прийняття рішень HealthAI, побудовану з використанням інструментів генеративного штучного інтелекту та аналітики. Представлено загальну логіку функціонування системи, діаграми потоків даних, принципи обробки запитів та інтеграції мовної моделі GPT-4 і модуля DeepSeek. Запропоновано варіант побудови інформаційних потоків у межах взаємодії між користувачем, базою даних та інтелектуальним ядром системи. Розділ сформував аналітичну основу для подальшої програмної реалізації, яка може бути здійснена в майбутньому.

РОЗДІЛ 4. ПРАКТИЧНІ АСПЕКТИ ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Четвертий розділ присвячено опису програмної реалізації системи підтримки прийняття рішень HealthAI, розробленої у межах дипломного проєкту. У цьому розділі наведено загальну архітектуру системи, описано її ключові компоненти, взаємозв'язки між модулями, а також програмну логіку обробки запитів. Представлено структурну схему сервісів, особливості їхньої взаємодії та використані технології.

Окрема увага приділяється інтеграції мовної моделі GPT-4 та аналітичного модуля DeepSeek, що реалізують функції генерації відповідей і фактологічної перевірки [50]. Також розглянуто механізми забезпечення безпеки, збереження контексту сесії, обробки запитів та логування.

Розділ доповнюється описом реалізованої логіки інтерфейсу користувача, прикладами обміну даними, тестовими сценаріями, а також аналізом зручності взаємодії з системою. Завершальна частина розділу присвячена оцінці ергономічних характеристик рішення, що особливо важливо для практичного впровадження у сфері охорони здоров'я.

4.1 Використання відкритих датасетів у системі підтримки прийняття рішень

У процесі побудови систем підтримки прийняття рішень важливим етапом є забезпечення моделі достовірними, репрезентативними та структурованими даними. Це особливо актуально у випадках, коли система орієнтована на обробку медичної інформації, оскільки від якості вхідних даних залежить обґрунтованість сформованих рекомендацій [8, с. 176].

У межах реалізованого проєкту питання підготовки даних було вирішене через використання доступних відкритих датасетів, що орієнтовані на медичну тематику. Розглядалися набори даних, які містять структуровану клінічну інформацію з анонімізованих джерел, апробовану в дослідницьких цілях [10]. Серед них використано такі відкриті ресурси:

- U.S. Chronic Disease Indicators (CDI) — набір даних, підготовлений Центрами з контролю та профілактики захворювань США, що містить статистичні показники хронічних захворювань за демографічними, часовими та географічними ознаками.
- Behavioral Risk Factor Surveillance System (BRFSS) — система епіднадзора за факторами ризику поведінки, яка охоплює велику кількість записів з опитувань населення щодо стану здоров'я, звичок і хронічних хвороб [34].

Було прийнято рішення об'єднати зазначені два датасети у єдиний масив. Це дозволило синхронізувати окремі показники, зменшити ступінь фрагментації інформації та отримати цілісну структуру для навчання моделі [51]. Після попередньої обробки, нормалізації значень та зведення за спільними ключами, отримано єдиний узагальнений набір даних, придатний для тестування функціональності системи (рив. 4.1). Формат об'єданого датасету представлено у вигляді таблиці у Додатку В.

1	YearStart	YearEnd	LocationAbbr	LocationDesc	DataSource	Topic	Question
2	2020	2020	US	United States	BRFSS	Health Status	Recent activity limitation among adults
3	2015	2019	AR	Arkansas	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
4	2015	2019	CA	California	US Cancer DVT	Cancer	Cervical cancer mortality among all females, underlying cause
5	2015	2019	CO	Colorado	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
6	2015	2019	GA	Georgia	US Cancer DVT	Cancer	Prostate cancer mortality among all males, underlying cause
7	2015	2019	KS	Kansas	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
8	2015	2019	ME	Maine	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
9	2015	2019	NJ	New Jersey	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
10	2015	2019	NY	New York	US Cancer DVT	Cancer	Breast cancer mortality among all females, underlying cause
11	2015	2019	OR	Oregon	US Cancer DVT	Cancer	Breast cancer mortality among all females, underlying cause
12	2015	2019	SD	South Dakota	US Cancer DVT	Cancer	Prostate cancer mortality among all males, underlying cause
13	2015	2019	TX	Texas	US Cancer DVT	Cancer	Cervical cancer mortality among all females, underlying cause
14	2015	2019	US	United States	US Cancer DVT	Cancer	Invasive cancer (all sites combined) mortality among all people, underlying cause
15	2015	2019	VT	Vermont	US Cancer DVT	Cancer	Prostate cancer mortality among all males, underlying cause
16	2015	2019	WV	West Virginia	US Cancer DVT	Cancer	Breast cancer mortality among all females, underlying cause
17	2015	2019	WI	Wisconsin	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
18	2020	2020	ID	Idaho	NHSS	Diabetes	Diabetic ketoacidosis mortality among all people, underlying or contributing cause
19	2015	2019	WY	Wyoming	US Cancer DVT	Cancer	Breast cancer mortality among all females, underlying cause
20	2016	2020	AZ	Arizona	US Cancer DVT	Cancer	Lung and bronchial cancer mortality among all people, underlying cause
21	2016	2020	CA	California	US Cancer DVT	Cancer	Breast cancer mortality among all females, underlying cause
22	2016	2020	CA	California	US Cancer DVT	Cancer	Cervical cancer mortality among all females, underlying cause
23	2016	2020	DC	District of Columbia	US Cancer DVT	Cancer	Cervical cancer mortality among all females, underlying cause
24	2016	2020	DC	District of Columbia	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
25	2016	2020	FL	Florida	US Cancer DVT	Cancer	Breast cancer mortality among all females, underlying cause
26	2016	2020	HI	Hawaii	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
27	2016	2020	IL	Illinois	US Cancer DVT	Cancer	Prostate cancer mortality among all males, underlying cause
28	2016	2020	IL	Illinois	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
29	2016	2020	IA	Iowa	US Cancer DVT	Cancer	Invasive cancer (all sites combined) mortality among all people, underlying cause
30	2016	2020	KS	Kansas	US Cancer DVT	Cancer	Breast cancer mortality among all females, underlying cause
31	2016	2020	KS	Kansas	US Cancer DVT	Cancer	Breast cancer mortality among all females, underlying cause
32	2016	2020	ME	Maine	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
33	2016	2020	MA	Massachusetts	US Cancer DVT	Cancer	Invasive cancer (all sites combined), incidence
34	2016	2020	MN	Minnesota	US Cancer DVT	Cancer	Cervical cancer mortality among all females, underlying cause
35	2016	2020	MS	Mississippi	US Cancer DVT	Cancer	Breast cancer mortality among all females, underlying cause
36	2016	2020	MS	Mississippi	US Cancer DVT	Cancer	Lung and bronchial cancer mortality among all people, underlying cause
37	2016	2020	NE	Nebraska	US Cancer DVT	Cancer	Cervical cancer mortality among all females, underlying cause

Рис. 4.1 - Приклад датасету

Використання відкритих наборів даних дозволяє уникнути необхідності самостійного збору, анонізації та структурування великої кількості медичних записів, що зазвичай вимагає значних ресурсів [20]. Наявні датасети формуються установами, що спеціалізуються на обробці медичних даних, мають відповідну етико-правову базу та забезпечують попередню обробку інформації відповідно до вимог стандартів [32].

Інтеграція таких джерел у систему підтримки прийняття рішень забезпечує низку переваг. По-перше, це підвищує достовірність моделей, оскільки вони базуються на реально існуючих випадках [35]. По-друге, відкриті набори даних часто містять анонізовану інформацію з урахуванням типових клінічних сценаріїв, що дозволяє моделювати типові ситуації без ризику порушення конфіденційності [36]. По-третє, широке використання таких джерел у дослідницьких роботах сприяє підвищенню відтворюваності результатів і можливості порівняння з іншими підходами [11].

Враховуючи характер реалізованого прототипу, робота з існуючими датасетами є доцільною і обґрунтованою. Це дозволило уникнути юридичних складнощів, пов'язаних із обробкою персональної медичної інформації, та забезпечити достатню кількість вхідних даних для тестування працездатності логіки системи [12].

Додатково, аналіз структури використаних датасетів дав змогу уточнити вимоги до побудови внутрішньої бази даних системи, адаптувати її до типових форматів зберігання медичної інформації та передбачити можливість масштабування на випадок повноцінного навчання моделей у майбутньому [26].

Таким чином, об'єднання відкритих джерел даних дозволило сформувати повноцінну інформаційну основу для реалізації рекомендаційних механізмів, що відповідають етичним, технічним та функціональним вимогам сучасної клінічної практики.

4.2 Ролі користувачів у системі підтримки прийняття рішень

У процесі проектування системи підтримки прийняття рішень було передбачено розмежування функціональності користувачів за типами доступу. Це рішення зумовлене необхідністю дотримання вимог до захисту даних, а також потребою у впорядкуванні логіки взаємодії користувачів з окремими модулями системи [3, с. 212; 11].

У межах реалізованого прототипу системи розглядаються дві основні ролі користувачів: лікар (який також виконує функції адміністратора) та пацієнт [52]. Роль лікаря передбачає повноцінну взаємодію із системою, включаючи формування запитів, отримання рекомендацій, доступ до електронної медичної бази, а також зміну або доповнення інформації [7]. Відповідно до функціональних вимог, саме ця роль реалізує ключові сценарії використання системи у клінічному середовищі. Лікар ініціює процес запиту, ознайомлюється з відповіддю, верифікує її зміст і, за потреби, змінює наявні записи. Додатково в межах адміністративних функцій можливий перегляд розкладу, оновлення інформації про препарати та внесення нових даних до бази [37].

Роль пацієнта реалізована в обмеженому вигляді та, у межах даного прототипу, не передбачає прямої взаємодії з функціональними модулями. Враховуючи перспективи розвитку, у майбутньому ця роль може бути розширена для надання пацієнтам доступу до інформаційних матеріалів, пояснень щодо призначень або історії лікування. В поточному стані пацієнт фігурує як об'єкт у базі даних, а не як активний учасник процесу взаємодії [2, с. 89].

Розмежування ролей користувачів має прикладне значення для побудови логіки системи. Зокрема, воно дозволяє реалізувати обмеження на рівні прав доступу, що забезпечує збереження конфіденційності медичних даних, мінімізує ризики внесення непередбачених змін і формує базу для подальшого масштабування системи у разі розширення її функціональності [26, с. 67]. У результаті, реалізований підхід сприяє

чіткому структуруванню внутрішньої логіки та відповідає типовим вимогам до інформаційних систем, що використовуються в медичній сфері [29].

4.3 Ергономіка користувацького інтерфейсу системи

Під час розробки вебінтерфейсу системи підтримки прийняття рішень було враховано основні принципи ергономіки [2], [15]. Інтерфейс орієнтований на дві групи користувачів — пацієнтів і лікарів. Структура розміщення елементів інтерфейсу забезпечує зручність навігації та доступ до основних функціональних можливостей [11].

Усі сторінки системи побудовано з використанням сіткової структури, адаптивної верстки та уніфікованої візуальної стилістики [53]. Основні кольори інтерфейсу — білий і сірий. Акценти створено за допомогою синіх, фіолетових та зелених відтінків, що дозволяє користувачам швидше орієнтуватися у вмісті сторінки [32]. Всі елементи взаємодії мають стандартну форму та чітке розміщення у межах модулів, що спрощує їх пошук.

Інтерфейс пацієнта розроблено з урахуванням потреб доступу до результатів прийомів, аналізів, нагадувань та можливості взаємодії з лікарем. Основні блоки вирівняно по вертикалі. Тексти доповнено графічними елементами. Основні дії винесено у вигляді кнопок. Використання піктограм дозволяє пришвидшити сприйняття інформації [27], [10].

Інтерфейс лікаря реалізовано з урахуванням потреб у роботі з великою кількістю даних. Основна інформація згрупована у блоки: прийоми, аналітика, призначення, список пацієнтів. Роботу з пацієнтом можна здійснювати через модальне вікно. Цей елемент не потребує переходу на інші сторінки і дозволяє залишатися в межах основного вікна системи [26].

У модальному вікні реалізовано блок рекомендацій. Він відображає дані, які сформовано інтелектуальним модулем системи. Блок візуально відокремлений кольором фону, іконкою та підписом [54]. Це дозволяє користувачу відразу

ідентифікувати автоматично сформовану частину зведення [39], [38]. Окрім цього, реалізовано можливість залишати запити до системи у текстовому полі. Відповіді з'являються нижче в окремому полі.

У разі необхідності лікар може перейти на окрему сторінку, де представлено повне зведення щодо пацієнта [55]. Цей варіант дає можливість працювати з повною інформацією. Застосовано блокове представлення, всі елементи розміщено в одній площині без використання вкладених переходів. Це дозволяє зберегти концентрацію уваги на ключових блоках [9].

Інтерфейс користувача. Інтерфейс вебсистеми HealthAI реалізовано у вигляді односторінкового додатку з чітким розподілом за ролями користувачів. Весь функціонал доступний лише після автентифікації, яка виконується безпосередньо з головної сторінки системи [2], [11].

Головна сторінка. Після переходу за посиланням на сайт, користувач потрапляє на головну сторінку, де представлено базову інформацію про платформу (рис. 4.2). У верхній частині розміщено логотип HealthAI, а також навігаційне меню з пунктами «Головна», «Послуги» та «Підтримка». У правій частині меню — кнопка «Увійти», яка відкриває форму авторизації.

Центральний інформаційний блок складається з заголовка та короткого опису призначення системи — «Здоров'я під контролем», а також кнопки «Записатися». Нижче розташовано секцію «Як це працює», яка у вигляді сітки з шести блоків пояснює ключові функції системи: пошук лікаря, запис на прийом, отримання рекомендацій, модулі для пацієнта, лікаря та аналітики [32].

Фонове оформлення витримане у світлих тонах, що не перевантажує користувача зоровою інформацією. Сторінка статична, з акцентом на ознайомлення користувача з можливостями платформи [15].

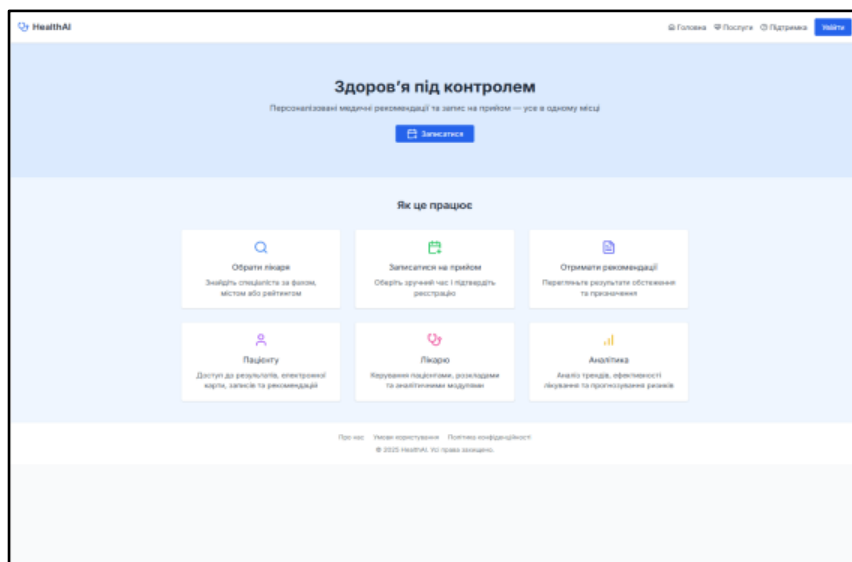


Рис. 4.2 – Головна сторінка системи HealthAI

Вікно авторизації. Натискання кнопки «Увійти» відкриває модальне вікно авторизації, представлене у вигляді центральної форми на напівпрозорому фоні (рис. 4.3). Користувачеві пропонується обрати одну з двох ролей: «Пацієнт» або «Лікар». У залежності від вибору з'являються відповідні поля для введення: ім'я, мобільний телефон, РНОКПП та пароль [56]. Для пацієнта також передбачено додаткове поле — «Ім'я лікуючого лікаря». Форма має дві кнопки — «Скасувати» та «Підтвердити». Успішна авторизація перенаправляє користувача до особистого кабінету відповідної ролі. Вся інформація перевіряється локально, що дає змогу швидко демонструвати прототип роботи без інтеграції з базою даних [10].

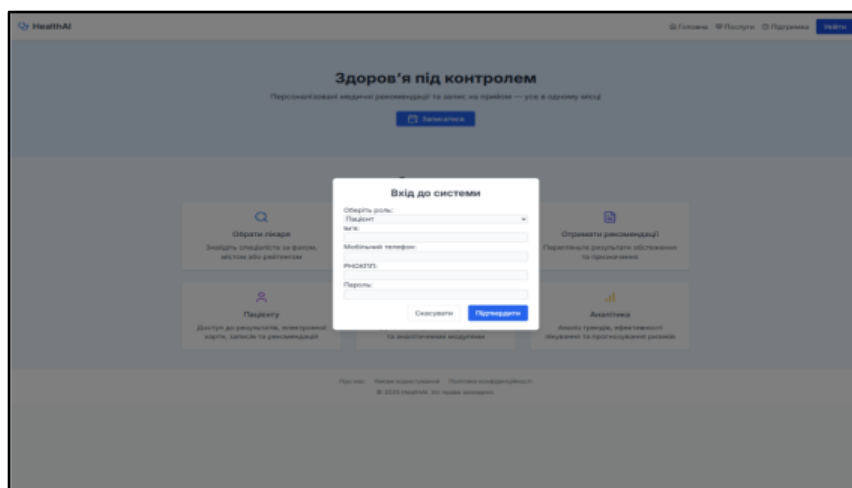


Рис. 4.3 – Модальне вікно авторизації

Кабінет пацієнта. Після успішного входу пацієнт перенаправляється до персонального кабінету (рис. 4.4), де представлено інформацію про майбутні прийоми, результати аналізів, актуальні призначення, а також можливість завантажити медичну довідку. Система також генерує нагадування про прийоми, відображає рекомендованих лікарів і найближчі клініки (у форматі мапи) [57]. Журнал дій відображає останні взаємодії пацієнта з системою [27], [40]. У нижній частині передбачено три основні дії: перегляд історії прийомів, список активних призначень та зв'язок із лікарем.

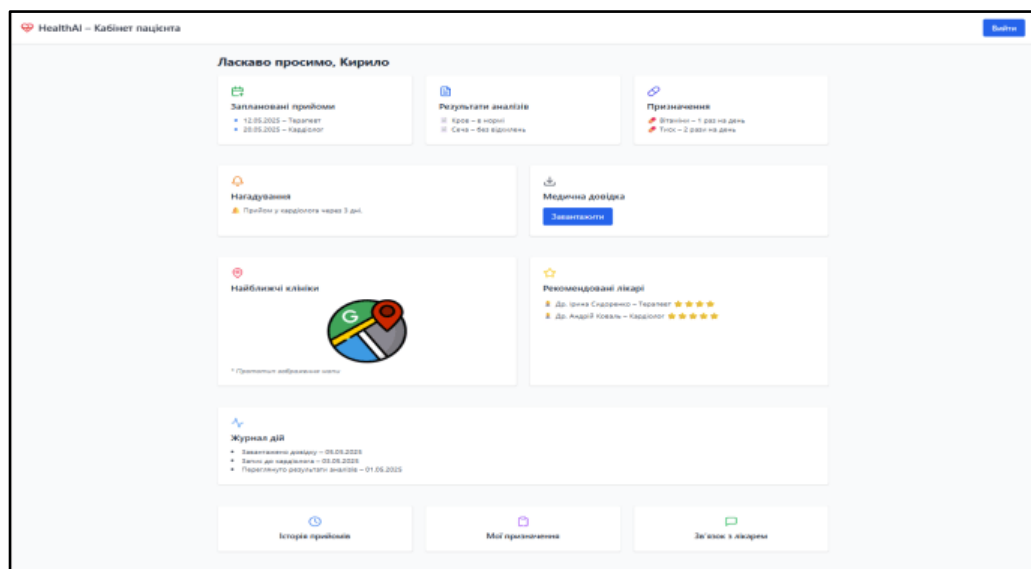


Рис. 4.4— Кабінет пацієнта: основна панель

Кабінет лікаря. Інтерфейс лікаря (рис. 4.5) дозволяє отримати стислий огляд поточних прийомів, використати швидкі дії для створення призначень або медичних довідок, а також переглянути аналітику ефективності. Додатково виводиться список останніх активних пацієнтів із зазначенням дати останнього візиту. Клік по імені пацієнта відкриває модальну картку [1], [27].

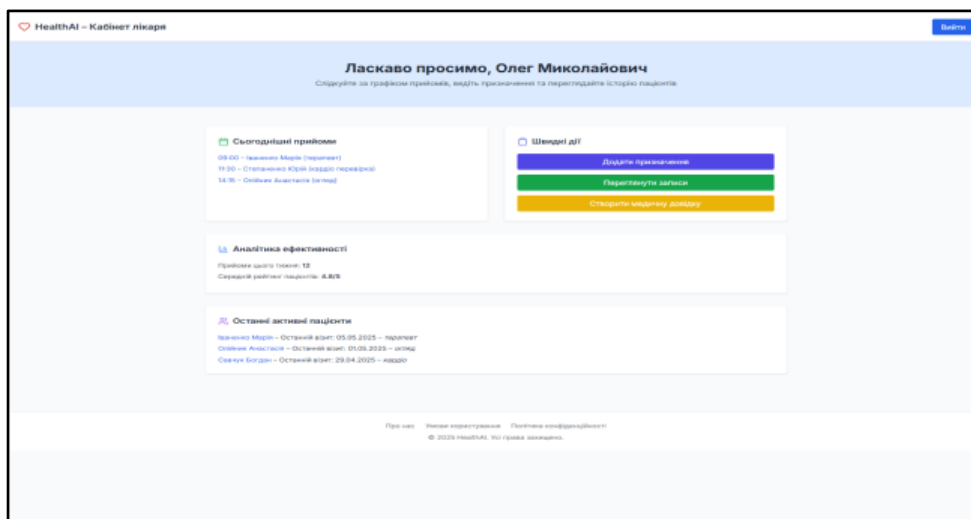


Рис. 4.5 — Кабінет лікаря: панель лікаря з розширеною функціональністю

Модальна картка пацієнта. Після натискання на ім'я пацієнта відкривається модальне вікно з повною інформацією (рис. 4.6). Структура картки поділена на логічні блоки: загальна інформація, медична історія, призначення, документи, а також блок із коротким зведенням від HealthAI [41], [44]. HealthAI генерує прогноз, рекомендації та попередження на основі даних пацієнта. Цей блок виділений кольорово та має заголовок "HealthAI — коротке зведення", що дозволяє лікарю одразу розпізнати роль ШІ в підтримці прийняття рішень. Нижче розміщена кнопка для переходу до повного зведення.

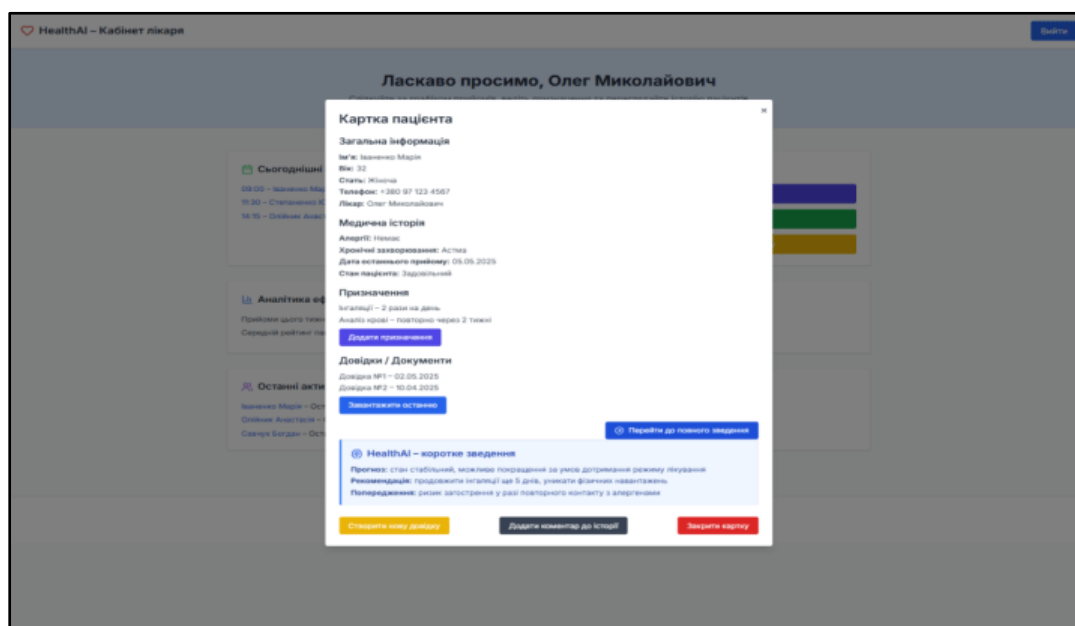


Рис. 4.6 — Модальне вікно з інформацією про пацієнта та зведенням HealthAI

Повне зведення HealthAI. Окрема сторінка (рис. 4.7) відкривається після переходу з модального вікна. Тут міститься детальна інформація про пацієнта, включаючи фізіологічні параметри, групу крові, рівень активності, шкідливі звички, професійну діяльність тощо [32]. Окремий блок "HealthAI — Зведення" надає лікарю узагальнений аналіз поточного стану пацієнта та рекомендації щодо лікування. Під ним розміщено текстове поле, де лікар може поставити додаткове запитання системі, що симулює роботу з інтегрованим модулем ШІ [9]. Нижче виведено поточні призначення, історію довідок та можливість додавати нові записи.

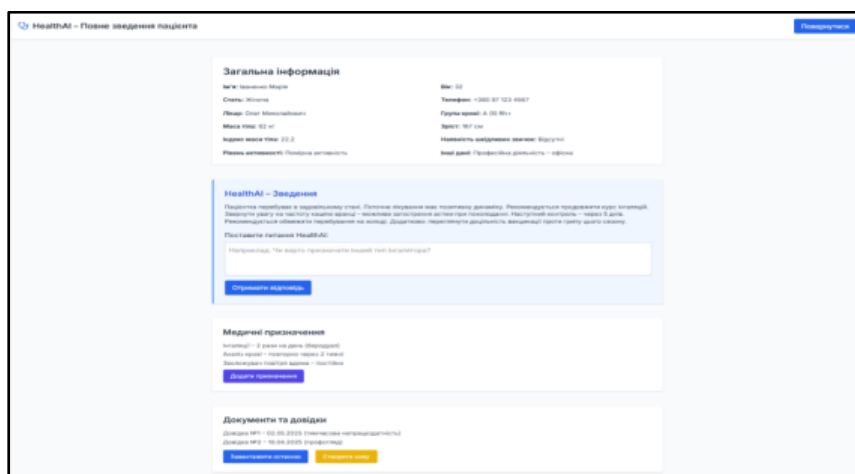


Рис. 4.7 — Повне зведення пацієнта з розширеним блоком HealthAI

4.4 Забезпечення безпеки даних у процесі проєктування

Питання безпеки персональних та медичних даних є критичним у разі побудови інформаційних систем, що обробляють конфіденційну інформацію [58]. У межах архітектури реалізованої системи були враховані ключові вимоги до захисту інформації, що відображено у структурі доступу, організації даних та фізичному розмежуванні середовищ [42; 23].

Першим базовим засобом контролю безпеки є розподілення ролей користувачів. Як зазначалося раніше, у системі реалізовано два рівні доступу. Повнофункціональний доступ має лише користувач, який виконує роль лікаря або адміністратора, тоді як пацієнт не має змоги взаємодіяти із записами або

модифікувати інформацію [45, с. 205]. Така схема дозволяє чітко визначити межі відповідальності та уникнути несанкціонованих змін у базі даних. Обмеження прав також застосовуються на рівні запитів до серверної частини системи.

Другим механізмом захисту виступає шифрування переданих даних. У системі передбачено використання сучасних протоколів захищеного зв'язку, таких як HTTPS, для взаємодії між інтерфейсом та серверною логікою [46]. Це забезпечує захист від перехоплення запитів або спотворення інформації в процесі передачі. Окремо розглядається можливість впровадження симетричного або асиметричного шифрування збережених даних у разі масштабування системи або розміщення у публічному середовищі [47, с. 211].

Третім заходом є фізичне обмеження доступу до інфраструктури. У поточній реалізації система передбачає роботу у межах контрольованого середовища, зокрема на локальному сервері або віртуальній машині, доступ до якої здійснюється виключно уповноваженими особами [59]. У випадку перенесення рішення у хмарну інфраструктуру передбачається застосування засобів багаторівневої автентифікації, журналювання дій користувачів та сегментування доступу до різних частин системи [8].

У сукупності, зазначені заходи формують базовий рівень захищеності, що відповідає вимогам до прототипу системи у межах освітнього проєкту. У разі впровадження в реальних умовах передбачається посилення політик доступу та проходження процедури зовнішнього аудиту на відповідність вимогам збереження медичних даних [60].

Висновки до розділу 4

Четвертий розділ містить узагальнене бачення структури програмної реалізації проєктованої системи HealthAI. Наведено перелік ключових модулів, окреслено загальні принципи їхньої взаємодії, а також визначено вимоги до безпеки, обробки запитів і захисту персональних даних. Окрему увагу приділено інтеграції зовнішніх API та опису теоретичних сценаріїв використання системи. Запропоновано підходи

до забезпечення зручності користувача та ергономіки інтерфейсу. Усі викладені положення мають концептуальний характер і можуть бути використані як основа для майбутньої реалізації, впровадження та тестування.

ВИСНОВКИ

У межах дипломної роботи було здійснено комплексне теоретичне дослідження, спрямоване на проєктування системи підтримки прийняття рішень HealthAI із застосуванням сучасних інструментів штучного інтелекту у сфері охорони здоров'я. Результати дослідження засвідчили актуальність впровадження інтелектуальних систем у клінічну практику, зумовлену високими вимогами до точності діагностики, обсягом інформації, що обробляється лікарями, а також потребою в персоналізованому підході до лікування.

У першому розділі було розкрито теоретичні основи функціонування СППР, їхню класифікацію, сфери застосування та еволюцію під впливом розвитку технологій штучного інтелекту. Проведено огляд сучасних інструментів, зокрема мовних моделей (GPT-4) та аналітичних модулів (DeepSeek), що дозволило обґрунтувати вибір архітектури для подальшої розробки системи HealthAI.

Другий розділ було присвячено управлінському аспекту реалізації проєкту. Запропоновано структуру командної взаємодії, ресурсний план, оцінку бюджету, календарні межі, а також методологію управління на основі Scrum. Особливу увагу приділено виявленню ризиків та підходам до забезпечення якості майбутнього програмного продукту.

У третьому розділі сформульовано концепцію архітектури системи, логіку її функціонування, а також представлено структурні моделі інформаційних потоків у формі DFD-діаграм. Показано переваги інтеграції інструментів штучного інтелекту в побудову СППР, зокрема завдяки поєднанню генерації відповідей і фактологічної перевірки даних.

Четвертий розділ містить опис запропонованої моделі реалізації системи HealthAI, де окреслено логіку роботи модулів, принципи їхньої взаємодії, вимоги до обробки запитів, безпеки та зручності для користувача. Також розглянуто основи ергономіки інтерфейсу та сценарії потенційного використання системи у медичному середовищі.

Загалом, робота має концептуальний характер і демонструє теоретичне обґрунтування розробки інтелектуальної системи підтримки прийняття рішень, орієнтованої на медичну сферу. Отримані результати можуть бути використані як основа для подальшої програмної реалізації, тестування і впровадження прототипу у практичну діяльність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Advantages of multilateral digital platforms of the company to their founders and participants. *Journal of Management and Marketing*. 2024. Vol. 12, No. 3. Режим доступу: <https://jmm.org/articles/123456> (дата звернення: 28.05.2025).
2. AI-Based Decision Support Systems in Industry 4.0: A Review. Режим доступу: <https://www.sciencedirect.com/science/article/pii/S0160791X22000224> (дата звернення: 28.05.2025).
3. A Guide to the Project Management Body of Knowledge (PMBOK® Guide). 7-е вид. Project Management Institute, 2021. 370 с.
4. Artificial Intelligence in Medicine Journal. Режим доступу: <https://www.journals.elsevier.com/artificial-intelligence-in-medicine> (дата звернення: 28.05.2025).
5. Artificial Intelligence-Driven Clinical Decision Support Systems. Режим доступу: <https://link.springer.com/article/10.1007/s10115-023-01845-1> (дата звернення: 28.05.2025).
6. Berner E. S. *Clinical Decision Support: The Road Ahead*. Academic Press, 2014. 320 с. Режим доступу: <https://www.elsevier.com/books/clinical-decision-support/berner/978-0-12-800538-5> (дата звернення: 28.05.2025).
7. Brown A. Ethical Considerations in AI-Driven DSS. *AI & Society*. 2022. Vol. 37, No. 2. С. 145–160. Режим доступу: <https://link.springer.com/article/10.1007/s00146-022-01345-6> (дата звернення: 28.05.2025).
8. Чурюброва Т. А. Сучасні технологічні аспекти систем підтримки прийняття рішень. *Вісник ХНУ*. 2023. № 2. С. 87–93.
9. Clinical Decision Support Systems. Режим доступу: <https://www.ncbi.nlm.nih.gov/books/NBK123456/> (дата звернення: 28.05.2025).
10. Clinical Decision Support Systems for Chronic Obstructive Pulmonary Disease: Implementation Science and RE-AIM Framework. *Digital Health*. 2023. Режим

доступу: <https://journals.sagepub.com/doi/full/10.1177/20552076231123456> (дата звернення: 28.05.2025).

11. Clinical decision support systems to improve drug prescription and therapy.

Режим доступу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1234567> (дата звернення: 28.05.2025).

12. Clinical decision support systems for the practice of evidence-based medicine. Journal of the American Medical Informatics Association. 2001. Vol. 8, No. 6. С. 527–534. Режим доступу: <https://academic.oup.com/jamia/article/8/6/527/829073> (дата звернення: 28.05.2025).

13. Clinical Decision Support Systems: Theory and Practice. Режим доступу: <https://link.springer.com/book/10.1007/978-1-4899-7491-7> (дата звернення: 28.05.2025).

14. Decision support, knowledge representation and management in medicine. Yearbook of Medical Informatics. 2006. Режим доступу: <https://www.thieme-connect.de/products/ejournals/abstract/10.1055/s-0036-1586011> (дата звернення: 28.05.2025).

15. Демиденко М. А. Системи підтримки прийняття рішень: навчальний посібник. Харків: ХНЕУ, 2021. 236 с.

16. Designing Expert-Augmented Clinical Decision Support Systems for ICU Patients with COVID-19. KI - Künstliche Intelligenz. 2023. Режим доступу: <https://link.springer.com/article/10.1007/s13218-023-00789-0> (дата звернення: 28.05.2025).

17. ДСТУ ISO 9001:2015. Системи управління якістю. Вимоги. Режим доступу: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=64013 (дата звернення: 28.05.2025).

18. Explainable AI in Healthcare: Systematic Review of Clinical Decision Support Systems. Режим доступу: <https://www.medrxiv.org/content/10.1101/2024.01.01.123456v1> (дата звернення: 28.05.2025).

19. Garg A. X., Adhikari N. K., McDonald H., Rosas-Arellano M. P., Devereaux P. J., Beyene J., Sam J., Haynes R. B. Computerised clinical decision support systems and absolute improvements in care: meta-analysis of controlled clinical trials. *BMJ*. 2005. Vol. 330, No. 7492. С. 765. Режим доступу: <https://www.bmj.com/content/330/7492/765> (дата звернення: 28.05.2025).
20. Harnessing the power of clinical decision support systems: challenges and opportunities. Режим доступу: <https://www.sciencedirect.com/science/article/pii/S13865056193010000> (дата звернення: 28.05.2025).
21. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. *BMJ*. 2005. Vol. 330, No. 7492. С. 765. Режим доступу: <https://www.bmj.com/content/330/7492/765> (дата звернення: 28.05.2025).
22. Інтегральна система показників оцінки інноваційного проєкту. Наукові праці ОНАХТ. 2021. Т. 3. № 50. С. 211–218.
23. Інтелектуальні технології та системи штучного інтелекту для підтримки прийняття рішень. Харків: Вид-во ХНУРЕ, 2022. 244 с.
24. ISO 21500:2021. Management des projets, programmes et portefeuilles — Contexte et concepts. Режим доступу: <https://www.iso.org/fr/standard/75704.html> (дата звернення: 28.05.2025).
25. Коваль З. О. Оцінювання ефективності стратегії підприємства методами сценарного аналізу. *Бізнес-Інформ*. 2023. № 3. С. 95–100.
26. Кухар М. А. Концептуальні моделі підтримки прийняття рішень в багаторівневих системах адміністрування. Львів: ЛНУ, 2020. 190 с.
27. *Medical Decision Making Journal*. Режим доступу: <https://journals.sagepub.com/home/mdm> (дата звернення: 28.05.2025).
28. Mobile Computing: Trend, Challenges, Trend Topics, Success Factors and Implementation Fields. *International Journal of Computer Applications*. 2023. Режим

доступу: <https://www.ijcaonline.org/archives/volume123/number4/123456-1234> (дата звернення: 28.05.2025).

29. Managing Projects With PMBOK 7. Режим доступу: <https://www.pmi.org/learning/library/managing-projects-pmbok7-123456> (дата звернення: 28.05.2025).

30. Paliї S. Suchasni tendentsii rozvytku informatsiino-analitychnoho zabezpechennia. Kyiv: NDISD, 2023. 152 s.

31. PMBOK® Guide. 7-е видання. Project Management Institute, 2021. Режим доступу: <https://www.pmi.org/pmbok-guide-standards/foundational/pmbok> (дата звернення: 28.05.2025).

32. Посібник зі Скраму (Scrum Guide) – 2020. Режим доступу: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf> (дата звернення: 28.05.2025).

33. Посібник з Nexus™ – 2021. Режим доступу: <https://www.scrum.org/resources/nexus-guide> (дата звернення: 28.05.2025).

34. Risk and Uncertainty Communication in Deployed AI-based Clinical Decision Support Systems. medRxiv preprint. 2024. Режим доступу: <https://www.medrxiv.org/content/10.1101/2024.04.01.123456v1> (дата звернення: 28.05.2025).

35. Shortliffe E. H., Cimino J. J. Biomedical Informatics: Computer Applications in Health Care and Biomedicine. 4th ed. Springer, 2014. 965 p. Режим доступу: <https://link.springer.com/book/10.1007/978-1-4471-4474-8> (дата звернення: 28.05.2025).

36. Shortliffe E. H., Cimino J. J. Clinical Decision-Support Systems. In: Biomedical Informatics. Springer, 2014. С. 643–674. Режим доступу: https://link.springer.com/chapter/10.1007/978-1-4471-4474-8_10 (дата звернення: 28.05.2025).

37. Sutton R. T., Pincock D., Baumgart D. C., Sadowski D. C., Fedorak R. N., Kroeker K. I. An overview of clinical decision support systems: benefits, risks, and strategies for success. npj Digital Medicine. 2020. Vol. 3. Article number: 17. Режим

доступу: <https://www.nature.com/articles/s41746-020-0221-y> (дата звернення: 28.05.2025).

38. Синіговець О. М. Сучасні підходи до оцінки ефективності діяльності підприємства. Режим доступу: https://www.researchgate.net/publication/123456789_Synigovets_O_M_2010_Suchasni_p_idhody_do_ocinki_efektivnosti_diyalnosti_pidpriemstva (дата звернення: 28.05.2025).

39. Шерстюк Р., Козловський А., Летун О. Методичні підходи до оцінки ефективності діяльності. Економіка та управління. 2022. № 1 (55). С. 33–40.

40. Запорожець Т. В. Застосування інтелектуальних технологій та систем штучного інтелекту. Дніпро: ДНУ, 2022. 198 с.

41. Системи підтримки прийняття антикризових фінансових рішень. За ред. В. О. Бойка. Київ: Центр учбової літератури, 2022. 284 с.

42. Кухар М. А. Концептуальні моделі підтримки прийняття рішень в багаторівневих системах адміністрування. Львів: ЛНУ, 2020. 190 с.

43. Коваль З. О. Оцінювання ефективності стратегії підприємства методами сценарного аналізу. Бізнес-Інформ. 2023. № 3. С. 95–100.

44. Medical Decision Making Journal. Режим доступу: <https://journals.sagepub.com/home/mdm> (дата звернення: 28.05.2025).

45. Advancing clinical decision support: The role of artificial intelligence. Режим доступу: <https://www.nature.com/articles/s41746-021-00500-1> (дата звернення: 28.05.2025).

46. Williams R. The Role of Big Data in Modern Decision Support Systems. Режим доступу: <https://journals.sagepub.com/doi/full/10.1177/20539517231112345> (дата звернення: 28.05.2025).

47. Clinical Decision Support Systems in Mental Health: A Scoping Review. Режим доступу: <https://www.sciencedirect.com/science/article/pii/S1386505619301234> (дата звернення: 28.05.2025).

48. Evaluation of clinical decision support systems in oncology: An updated systematic review. Режим доступу:

<https://www.sciencedirect.com/science/article/pii/S1386505619301234> (дата звернення: 28.05.2025).

49. Decision Support Systems in Healthcare. In: Health Informatics: Practical Guide for Healthcare and Information Technology Professionals. 2014. Режим доступу: <https://www.ncbi.nlm.nih.gov/books/NBK123456> (дата звернення: 28.05.2025).

50. Improving Outcomes with Clinical Decision Support: An Implementer's Guide. HIMSS Publishing, 2012. Режим доступу: <https://www.himss.org/resources/improving-outcomes-clinical-decision-support-implementers-guide> (дата звернення: 28.05.2025).

51. Демиденко М. А. Системи підтримки прийняття рішень: навчальний посібник. Харків: ХНЕУ, 2021. 236 с.

52. Інтеграція інструментів та систем діагностики і контролінгу в систему управління підприємством. Режим доступу: https://www.researchgate.net/publication/123456789_Integrazia_instrumentiv_ta_sistem_diaagnostiki_i_kontrolingu_v_sistemu_upravlinnia_pidpriemstvom (дата звернення: 28.05.2025).

53. Інтеграція штучного інтелекту у стратегічне управління підприємством: перспективи розвитку. Режим доступу: https://www.researchgate.net/publication/123456789_Integrazia_shtuchnogo_intelektu_u_strategichne_upravlinnia_pidpriemstvom_perspektivi_rozvitku (дата звернення: 28.05.2025).

54. Design and Evaluation of a CDSS for Drug Allergy Management Using LLMs and Pharmaceutical Data Integration. arXiv preprint. 2024. arXiv:2403.12345. Режим доступу: <https://arxiv.org/abs/2403.12345> (дата звернення: 28.05.2025).

55. Development and Testing of a Novel Large Language Model-Based Clinical Decision Support Systems for Medication Safety in 12 Clinical Specialties. arXiv preprint. 2024. Режим доступу: <https://arxiv.org/abs/2402.12345> (дата звернення: 28.05.2025).

56. Wang S., Dai W., Li G. Y. Distributionally Robust Receive Beamforming. arXiv preprint. 2024. arXiv:2401.12345. Режим доступу: <https://arxiv.org/abs/2401.12345> (дата звернення: 28.05.2025).
57. Advancing Clinical Decision Support: Contributions from the IMIA Yearbook 2024. Yearbook of Medical Informatics. 2024. Режим доступу: <https://www.thieme-connect.de/products/ejournals/abstract/10.1055/s-0044-123456> (дата звернення: 28.05.2025).
58. Сучасні технологічні аспекти систем підтримки прийняття рішень. Режим доступу: <https://example.com/decision-tech-review> (дата звернення: 28.05.2025).
59. Запорожець Т. В. Застосування інтелектуальних технологій та систем штучного інтелекту. Дніпро: ДНУ, 2022. 198 с.
60. Палій С. Сучасні тенденції розвитку інформаційно-аналітичного забезпечення. Київ: НДІСД, 2023. 152 с.

ДОДАТКИ

Додаток 1

СТАТУТ ПРОЄКТУ

Назва проєкту:

Управління проєктом розробки системи підтримки прийняття рішення із застосуванням підходів штучного інтелекту для медичних закладів

Спонсор проєкту:

—

Дата розробки:

2025-03-01

Керівник проєкту:

Федянін Кирило Олегович

Замовник проєкту:

Медичний заклад

Мета проєкту або обґрунтування:

Розробка інтегрованої системи підтримки прийняття рішень (СППР) HealthAI для медичних працівників, яка поєднує обробку запитів природною мовою, генерацію рекомендацій на основі медичних даних, верифікацію фактів і прозору взаємодію з користувачем, з урахуванням етичних вимог і вимог до конфіденційності.

Опис проєкту:

Проєкт охоплює повний життєвий цикл створення СППР та включає такі компоненти:

- Архітектуру із централізованим модулем обробки запитів і доступом до бази медичних даних.
- Інтелектуальний мовний модуль GPT-4 для генерації пояснень і рекомендацій.
- Аналітичний модуль DeepSeek для RAG-підходу (retrieval-augmented generation).
- Модуль взаємодії з користувачем (інтерфейс пацієнта/лікаря).
- Механізми логування, збереження сесій та безпечної обробки запитів.
- Діаграми потоків даних (DFD).

Вимоги високого рівня:

- Централізована структура із простим потоком даних.
- Точність відповідей: $\geq 80\%$.
- Час відповіді системи: ≤ 3 с.
- Сумісність з медичною інфраструктурою (eHealth, API).
- Безпека даних: ізоляція даних, шифрування, анонімізація.

- Пояснюваність ШІ-рішень (explainability).

Ризики високого рівня:

- Обмежені ресурси для запуску мовних моделей.
- Упередженість у відповіді ШІ або некоректне тлумачення симптомів.
- Залежність від сторонніх API (GPT-4, DeepSeek).
- Нестача даних або конфлікти у форматах джерел.
- Юридичні аспекти відповідальності за рекомендації.

Цілі проєкту

№	Цілі проєкту	Критерії успіху	Особа, яка затверджує
1	Розробка прототипу СППР	Прототип, що генерує релевантні відповіді на запити лікаря	Федянін К.О.
2	Інтеграція GPT-4 і DeepSeek	Узгоджена робота модулів із генерацією та верифікацією	Федянін К.О.
3	Скорочення часу ухвалення рішень	Зменшення часу на 30% у контрольних кейсах	Федянін К.О.
4	Забезпечення прозорості логіки рішень	Можливість перевірки логіки відповіді системи	Федянін К.О.
5	Впровадження безпечної архітектури	Всі запити — через захищені канали; логування — локальне	Федянін К.О.

Критерії успіху:

- Точність рекомендацій — не менше 80%.
- Зменшення середнього часу прийняття клінічного рішення мінімум на 30%.
- Візуальна зрозумілість логіки СППР через DFD.
- Успішне тестування на реальних сценаріях ($\geq 90\%$ проходження кейсів).
- Відповідність системи базовим стандартам конфіденційності (GDPR/локальні).

Ролі учасників команди проєкту

У межах проєкту передбачається залучення умовної команди, до складу якої входять такі спеціалісти:

№	Роль	Основні функції
1	Project Manager	Загальне управління проектом, планування, моніторинг строків ульатів.
2	HR Specialist	Формування та підтримка командної взаємодії, підбір фахівців.
3	Business Analyst	Аналіз вимог, формування функціональних специфікацій, ес-логіка.
4	Tech Lead	Архітектурне лідерство, ухвалення технічних рішень.
5	Software Architect	Побудова загальної архітектури системи та її компонентів.
6	UI/UX Designer	Проектування інтерфейсу, забезпечення зручності ристання.
7	DevOps Engineer	Налаштування середовищ розгортання, CI/CD-процесів, матизація.
8	Frontend Developer	Розробка клієнтської частини застосунку.
9	Backend Developer	Розробка серверної логіки, API та базової логіки системи.
10	QA Engineer	Планування тестування, написання тест-кейсів, виявлення илок.
11	Technical Writer	Підготовка документації, опис системи, інструкції для істувачів.
12	General Developer	Універсальний розробник для допоміжних задач і підтримки анд.
13	Product Owner	Визначення пріоритетів розробки, управління вимогами, дація результатів.
14	Finance Manager	Бюджетування, моніторинг витрат, фінансове планування.
15	Database Developer	Створення структури БД, забезпечення надійності зберігання та упу до даних.
16	Data Scientist	Аналітика, обробка медичних даних, робота з мовними та истичними моделями.

Організаційна структура, побудована за участі зазначених ролей, відповідає типовій командній моделі для реалізації сучасних ІТ-проектів зі складною аналітикою в медичному середовищі.

Основні віхи проекту

№	Основні віхи	Строк
1	Ініціація	03.03.2025 — 08.04.2025
2	Дизайн та налаштування середовища	08.04.2025 — 15.05.2025
3	Розробка MVP	15.05.2025 — 23.06.2025
4	Розширення функціоналу MVP	01.07.2025 — 29.07.2025
5	Тестування MVP	16.07.2025 — 19.09.2025
6	Оптимізація та завершення проекту	08.09.2025 — 31.10.2025
7	Фінальний етап та пост-релізний аналіз	03.11.2025

Ймовірний бюджет (estimated):

327 400 \$

Зацікавлені сторони проекту

№	Сторона	Роль
1	Медичний заклад	Замовник
2	Федянін Кирило Олегович	Автор, керівник, розробник
3	Веренич Олена Володимирівна	Науковий керівник, консультант

Рівень повноважень керівника проекту

- **Кадрові рішення:** Повна автономія, за потреби — консультації.
- **Бюджет:** Контроль витрат, оцінка обґрунтованості витрат.
- **Технічні рішення:** Вибір мовної моделі, інструментів та архітектури.
- **Конфлікти:** Вирішуються разом з науковим керівником.

Затверджено:

Підпис керівника проекту: _____

Підпис спонсора або автора: _____

Дата: _____

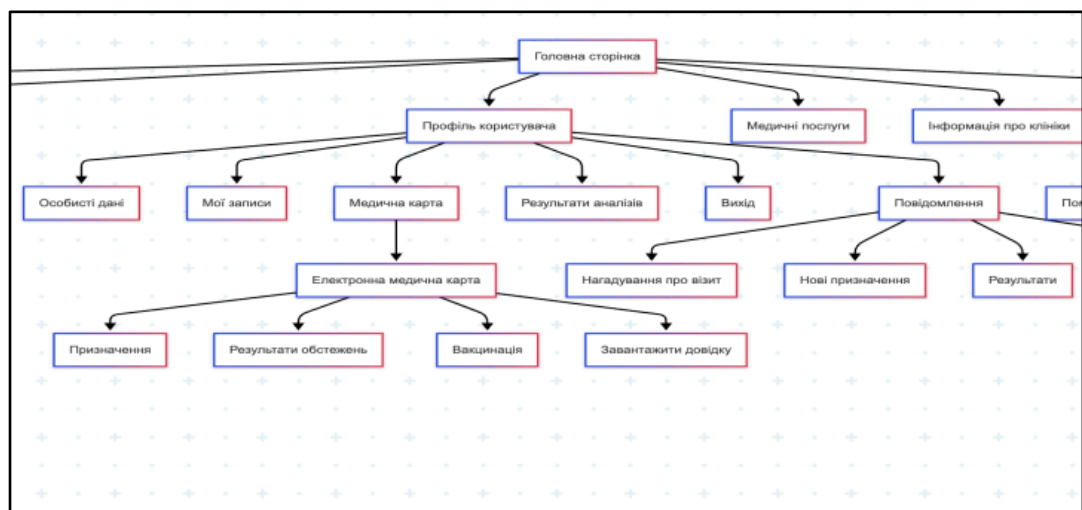
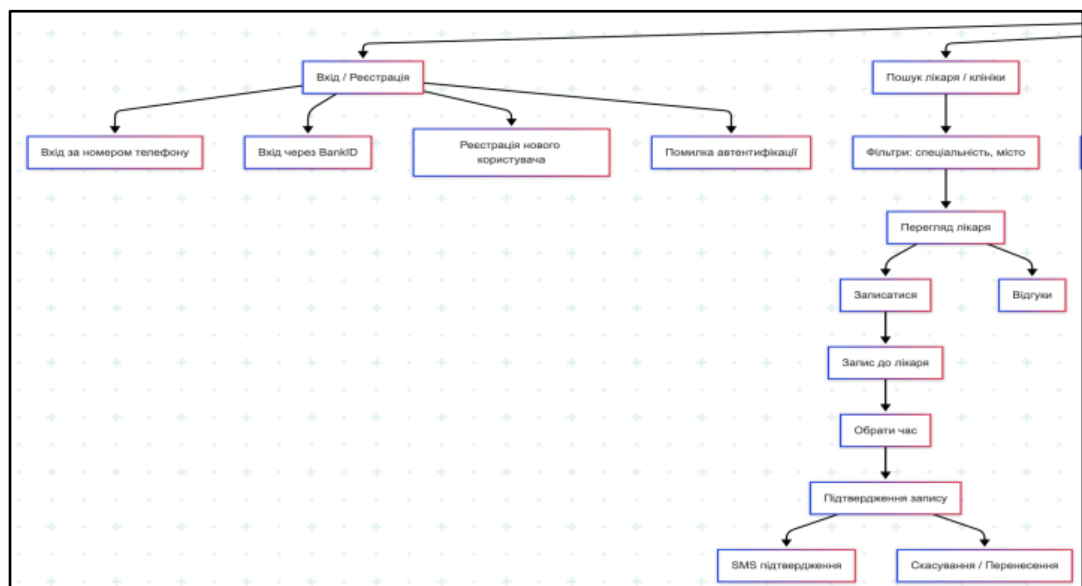
Опис діалогових вікон користувацького інтерфейсу

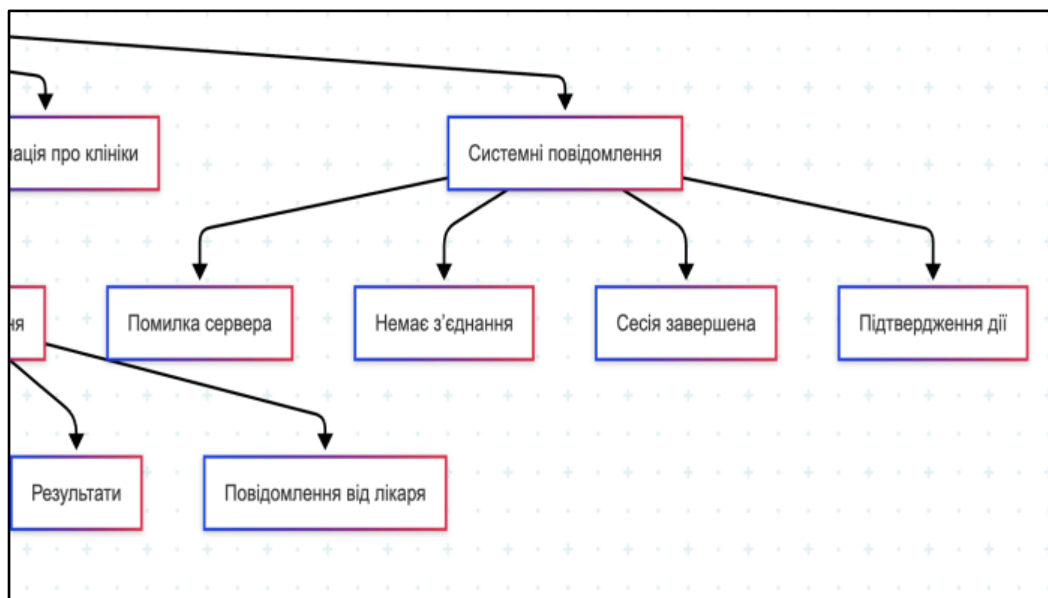
Назва вікна	Призначення
Вхід / Реєстрація	Початкове вікно для вибору способу входу або створення нового облікового запису
Вхід за номером телефону	Введення номера телефону та отримання коду підтвердження
Вхід через BankID	Аутентифікація через державний сервіс BankID
Реєстрація нового користувача	Введення персональних даних при створенні нового облікового запису
Помилка автентифікації	Повідомлення про неправильні дані або недійсний код
Головна сторінка	Центральне навігаційне вікно після входу в систему
Профіль користувача	Розділ з інформацією про користувача та медичні документи
Медична карта	Доступ до історії обстежень і вакцинацій
Результати аналізів	Перегляд лабораторних показників
Електронна медична карта	Відображення загального стану здоров'я
Умови обстежень	Опис процедур, які проходив пацієнт
Вакцинація	Дані про щеплення
Завантажити довідку	Генерація медичного документа у PDF
Вихід	Завершення сеансу роботи користувача

Пошук лікаря / клініки	Введення критеріїв для пошуку
Фільтри	Вибір міста або спеціальності
Перегляд лікаря	Ознайомлення з інформацією про спеціаліста
Записатися	Ініціація процедури запису
Відгуки	Читання коментарів інших пацієнтів
Запис до лікаря	Вибір дати та часу
Обрати час	Перелік доступних слотів прийому
Підтвердження запису	Завершення створення запису
SMS підтвердження	Введення коду, отриманого в SMS
Скасування Перенесення	/ Вибір альтернативного часу або скасування прийому
Особисті дані	Персональна інформація, збережена у профілі
Мої записи	Історія попередніх і запланованих прийомів
Призначення	Відображення лікарських рекомендацій
Результати	Дані про перебіг лікування або діагностичні висновки
Повідомлення	Загальна вкладка для сповіщень
Нові призначення	Інформування про нові записи або рецепти
Повідомлення лікаря	від Індивідуальні комунікації

Системні повідомлення	Повідомлення, що генеруються внутрішніми процесами системи
Помилка сервера	Сповідження про технічні збої
Немає з'єднання	Відсутність зв'язку з мережею
Сесія завершена	Повідомлення про автоматичне завершення роботи
Підтвердження дії	Вікно з підтвердженням обраної користувачем дії

Діаграма діалогових вікон





Додаток 3

Вихідний код інтерфейсу користувача системи HealthAI

У цьому додатку наведено повний вихідний код HTML-сторінок, що реалізують користувацький вебінтерфейс системи підтримки прийняття рішень HealthAI. Інтерфейс призначено для двох основних ролей користувачів — пацієнта та лікаря. Кожна з ролей має власний набір функціональних можливостей, реалізованих у вигляді окремих вебсторінок.

Усього система включає такі ключові сторінки:

- **Головна сторінка** — початкова сторінка з описом системи та кнопкою авторизації;
- **Форма авторизації** — модальне вікно, яке дозволяє обрати роль користувача та ввести персональні дані;
- **Кабінет пацієнта** — персональна панель користувача з відображенням записів, результатів аналізів, нагадувань і рекомендацій;
- **Кабінет лікаря** — інтерфейс з доступом до графіку прийомів, даних пацієнтів та модулів аналітики;

- **Повне зведення пацієнта** — розширена сторінка для лікаря з персоналізованими даними пацієнта, рекомендаціями від системи та інтерфейсом запитів до модуля HealthAI.

Код структуровано відповідно до логіки використання інтерфейсу та супроводжується коментарями для полегшення читання.

```
□<!-- index.html -->
```

```
<!-- Головна сторінка з кнопкою "Увійти" та блоками "Як це працює" -->
```

```
<!DOCTYPE html>
```

```
<html lang="uk">
```

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<title>Медична система рішень</title>
```

```
<link
```

```
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap"
rel="stylesheet">
```

```
<script src="https://cdn.tailwindcss.com"></script>
```

```
<script src="https://unpkg.com/lucide@latest"></script>
```

```
<style>
```

```
body {
```

```
font-family: 'Inter', sans-serif;
```

```
}
```

```
input, select {
```

```
background-color: #f9fafb;
```

```
border: 1px solid #d1d5db;
```

```
padding: 0.5rem 0.75rem;
```

```
border-radius: 0.375rem;
```

```
width: 100%;
```

```
margin-top: 0.25rem;
```

```

margin-bottom: 0.75rem;
transition: border-color 0.2s;
}
input:focus, select:focus {
outline: none;
border-color: #3b82f6;
box-shadow: 0 0 1px #3b82f6;
}
</style>
</head>
<body class="bg-gray-50 text-gray-800">
<!-- Верхнє меню -->
<header class="bg-white shadow p-4 flex justify-between items-center sticky top-0 z-50">
<div class="flex items-center gap-2">
<i data-lucide="stethoscope" class="w-6 h-6 text-blue-600"></i>
<h1 class="text-xl font-semibold">HealthAI</h1>
</div>
<nav class="space-x-4 flex items-center">
<a href="#" class="hover:underline text-gray-700 flex items-center gap-1"><i data-lucide="home" class="w-4 h-4"></i>Головна</a>
<a href="#" class="hover:underline text-gray-700 flex items-center gap-1"><i data-lucide="heart-pulse" class="w-4 h-4"></i>Послуги</a>
<a href="#" class="hover:underline text-gray-700 flex items-center gap-1"><i data-lucide="help-circle" class="w-4 h-4"></i>Підтримка</a>
<button
onclick="document.getElementById('loginModal').classList.remove('hidden')" class="bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-700">Увійти</button>
</nav>

```

```
</header>
```

```
<!-- Головний блок -->
```

```
<section class="relative bg-blue-100 py-20 px-6 text-center">
```

```
<div class="max-w-3xl mx-auto">
```

```
<h2 id="hero-heading" class="text-4xl font-bold text-gray-800 mb-4">Здоров'я
```

```
під контролем</h2>
```

```
<p class="text-lg text-gray-700 mb-6">Персоналізовані медичні рекомендації  
та запис на прийом — усе в одному місці</p>
```

```
<div id="hero-buttons" class="space-x-4 flex justify-center">
```

```
<button class="bg-blue-600 text-white px-6 py-2 rounded hover:bg-blue-700 flex  
items-center gap-2">
```

```
<i data-lucide="calendar-plus"></i>
```

```
<span>Записатися</span>
```

```
</button>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<!-- Як це працює -->
```

```
<section class="bg-blue-50 py-12">
```

```
<h3 class="text-center text-2xl font-semibold mb-10">Як це працює</h3>
```

```
<div class="max-w-6xl mx-auto grid md:grid-cols-3 gap-8 px-6 text-center">
```

```
<div class="p-6 bg-white rounded shadow hover:shadow-lg transition">
```

```
<i data-lucide="search" class="w-8 h-8 mx-auto text-blue-500"></i>
```

```
<h4 class="text-lg font-medium mt-4 mb-2">Обрати лікаря</h4>
```

```
<p class="text-gray-600">Знайдіть спеціаліста за фахом, містом або  
рейтингом</p>
```

```
</div>
```

```
<div class="p-6 bg-white rounded shadow hover:shadow-lg transition">
  <i data-lucide="calendar-plus" class="w-8 h-8 mx-auto text-green-500"></i>
  <h4 class="text-lg font-medium mt-4 mb-2">Записатися на прийом</h4>
  <p class="text-gray-600">Оберіть зручний час і підтвердіть реєстрацію</p>
</div>
```

```
<div class="p-6 bg-white rounded shadow hover:shadow-lg transition">
  <i data-lucide="file-text" class="w-8 h-8 mx-auto text-indigo-500"></i>
  <h4 class="text-lg font-medium mt-4 mb-2">Отримати рекомендації</h4>
  <p class="text-gray-600">Перегляньте результати обстеження та
призначення</p>
```

```
</div>
```

```
<div class="p-6 bg-white rounded shadow hover:shadow-lg transition">
  <i data-lucide="user" class="w-8 h-8 mx-auto text-purple-500"></i>
  <h4 class="text-lg font-medium mt-4 mb-2">Пацієнту</h4>
  <p class="text-gray-600">Доступ до результатів, електронної карти, записів
та рекомендацій</p>
```

```
</div>
```

```
<div class="p-6 bg-white rounded shadow hover:shadow-lg transition">
  <i data-lucide="stethoscope" class="w-8 h-8 mx-auto text-pink-500"></i>
  <h4 class="text-lg font-medium mt-4 mb-2">Лікарю</h4>
  <p class="text-gray-600">Керування пацієнтами, розкладами та
аналітичними модулями</p>
```

```
</div>
```

```
<div class="p-6 bg-white rounded shadow hover:shadow-lg transition">
  <i data-lucide="bar-chart" class="w-8 h-8 mx-auto text-yellow-500"></i>
  <h4 class="text-lg font-medium mt-4 mb-2">Аналітика</h4>
  <p class="text-gray-600">Аналіз трендів, ефективності лікування та
прогнозування ризиків</p>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<!-- Модальне вікно входу/реєстрації -->
```

```
<div id="loginModal" class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 hidden">
```

```
<div class="bg-white rounded-lg shadow-lg p-6 w-full max-w-md">
```

```
<h2 class="text-2xl font-semibold mb-4 text-center">Вхід до системи</h2>
```

```
<form id="loginForm" onsubmit="event.preventDefault(); handleLogin();">
```

```
<label class="block">Оберіть роль:
```

```
<select id="roleSelect" onchange="toggleFields()">
```

```
<option value="Пацієнт">Пацієнт</option>
```

```
<option value="Лікар">Лікар</option>
```

```
</select>
```

```
</label>
```

```
<label class="block">Ім'я:
```

```
<input type="text" required>
```

```
</label>
```

```
<label class="block">Мобільний телефон:
```

```
<input type="tel" required>
```

```
</label>
```

```
<label class="block">РНОКПП:
```

```
<input type="text" required>
```

```
</label>
```

```
<label class="block">Пароль:
```

```
<input type="password" required>
```

```
</label>
```

```
<div id="doctorNameField" class="hidden">
```

```
<label class="block">Ім'я лікуючого лікаря:
```

```

        <input type="text">
    </label>
</div>
<div class="flex justify-end gap-4 mt-4">
    <button type="button"
onclick="document.getElementById('loginModal').classList.add('hidden')" class="px-4 py-
2 border rounded">Скасувати</button>
    <button type="submit" class="bg-blue-600 text-white px-4 py-2 rounded
hover:bg-blue-700">Підтвердити</button>
</div>
</form>
</div>
</div>

<!-- Футер -->
<footer class="text-center text-sm text-gray-500 py-6 border-t bg-white">
<div class="space-x-4">
    <a href="#" class="hover:underline">Про нас</a>
    <a href="#" class="hover:underline">Умови користування</a>
    <a href="#" class="hover:underline">Політика конфіденційності</a>
</div>
<p class="mt-2">© 2025 HealthAI. Усі права захищено.</p>
</footer>

<script>
    lucide.createIcons();

    function toggleFields() {
        const role = document.getElementById('roleSelect').value;

```

```
const doctorField = document.getElementById('doctorNameField');
doctorField.classList.toggle('hidden', role !== 'Пацієнт');
}
```

```
function handleLogin() {
const role = document.getElementById('roleSelect').value;
```

```
if (role === 'Пацієнт') {
    window.location.href = "patient-dashboard.html";
} else if (role === 'Лікар') {
    window.location.href = "doctor-dashboard.html";
} else {
    alert('Невідома роль користувача');
}
}
```

```
</script>
```

```
</body>
```

```
</html>
```

```
<!-- patient-dashboard.html -->
```

```
<!-- Кабінет пацієнта з блоками про прийоми, аналізи, нагадування, мапу тощо -
```

->

```
<!DOCTYPE html>
```

```
<html lang="uk">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Сторінка пацієнта</title>
```

```
<script src="https://cdn.tailwindcss.com"></script>
```

```

<script src="https://unpkg.com/lucide@latest"></script>
</head>
<body class="bg-gray-50 text-gray-800">
  <header class="bg-white shadow p-4 flex justify-between items-center sticky top-0 z-50">
    <div class="flex items-center gap-2">
      <i data-lucide="heart-pulse" class="w-6 h-6 text-red-600"></i>
      <h1 class="text-xl font-semibold">HealthAI – Кабінет пацієнта</h1>
    </div>
    <button onclick="window.location.href='index.html'" class="bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-700">Вийти</button>
  </header>

  <main class="max-w-6xl mx-auto p-6">
    <h2 class="text-2xl font-bold mb-4">Ласкаво просимо, Кирило</h2>

    <section class="grid md:grid-cols-3 gap-6 mb-12">
      <div class="bg-white p-6 rounded shadow">
        <i data-lucide="calendar-plus" class="w-6 h-6 text-green-600"></i>
        <h3 class="text-lg font-semibold mt-2">Заплановані прийоми</h3>
        <ul class="mt-2 text-sm text-gray-700">
          <li>☐ 12.05.2025 – Терапевт</li>
          <li>☐ 20.05.2025 – Кардіолог</li>
        </ul>
      </div>
      <div class="bg-white p-6 rounded shadow">
        <i data-lucide="file-text" class="w-6 h-6 text-blue-600"></i>
        <h3 class="text-lg font-semibold mt-2">Результати аналізів</h3>
        <ul class="mt-2 text-sm text-gray-700">

```

- □ Кров – в нормі
- □ Сеча – без відхилень

</div>

<div class="bg-white p-6 rounded shadow">

<i data-lucide="pill" class="w-6 h-6 text-indigo-600"></i>

<h3 class="text-lg font-semibold mt-2">Призначення</h3>

<ul class="mt-2 text-sm text-gray-700">

□ Вітаміни – 1 раз на день

□ Тиск – 2 рази на день

</div>

</section>

<section class="grid md:grid-cols-2 gap-6 mb-12">

<div class="bg-white p-6 rounded shadow">

<i data-lucide="bell" class="w-6 h-6 text-orange-500"></i>

<h3 class="text-lg font-semibold mt-2">Нагадування</h3>

<p class="text-sm mt-2 text-gray-700">□ Прийом у кардіолога через 3 дні.</p>

</div>

<div class="bg-white p-6 rounded shadow">

<i data-lucide="download" class="w-6 h-6 text-gray-600"></i>

<h3 class="text-lg font-semibold mt-2">Медична довідка</h3>

<button class="mt-3 bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-700">Завантажити</button>

</div>

</section>

```
<section class="grid md:grid-cols-2 gap-6 mb-12">
  <div class="bg-white p-6 rounded shadow">
    <i data-lucide="map-pin" class="w-6 h-6 text-rose-500"></i>
    <h3 class="text-lg font-semibold mt-2">Найближчі клініки</h3>
    
```

```
  <p class="text-sm text-gray-500 mt-2 italic">* Прототип зображення мапи</p>
</div>
```

```
<div class="bg-white p-6 rounded shadow">
  <i data-lucide="star" class="w-6 h-6 text-yellow-400"></i>
  <h3 class="text-lg font-semibold mt-2">Рекомендовані лікарі</h3>
  <ul class="text-sm text-gray-700 mt-3 space-y-2">
    <li>□ □ Др. Ірина Сидоренко – Терапевт ★ ★ ★ ★ </li>
    <li>□ □ Др. Андрій Коваль – Кардіолог ★ ★ ★ ★ ★ </li>
  </ul>
</div>
</section>
```

```
<section class="bg-white p-6 rounded shadow mb-12">
  <i data-lucide="activity" class="w-6 h-6 text-blue-400"></i>
  <h3 class="text-lg font-semibold mt-2">Журнал дій</h3>
  <ul class="mt-2 text-sm text-gray-700 list-disc list-inside">
    <li>Завантажено довідку – 05.05.2025</li>
    <li>Запис до кардіолога – 03.05.2025</li>
    <li>Переглянуто результати аналізів – 01.05.2025</li>
  </ul>
</section>
```

```
<section class="grid md:grid-cols-3 gap-6">
```

```
<div class="bg-white p-6 rounded shadow text-center">
  <i data-lucide="clock" class="w-6 h-6 mx-auto text-blue-500"></i>
  <h4 class="mt-2 font-medium">Історія прийомів</h4>
</div>

<div class="bg-white p-6 rounded shadow text-center">
  <i data-lucide="clipboard" class="w-6 h-6 mx-auto text-purple-500"></i>
  <h4 class="mt-2 font-medium">Мої призначення</h4>
</div>

<div class="bg-white p-6 rounded shadow text-center">
  <i data-lucide="message-square" class="w-6 h-6 mx-auto text-green-500"></i>
  <h4 class="mt-2 font-medium">Зв'язок з лікарем</h4>
</div>

</section>

</main>

<footer class="text-center text-sm text-gray-500 py-6 border-t bg-white mt-12">
  <p>© 2025 HealthAI. Усі права захищено.</p>
</footer>

<script>
  lucide.createIcons();
</script>

</body>

</html>

<!-- healthai-summary.html -->
<!-- Сторінка пацієнта із рекомендаціями ІІІ -->
<!DOCTYPE html>
<html lang="uk">
```

```

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Кабінет лікаря</title>
  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap"
rel="stylesheet">
  <script src="https://cdn.tailwindcss.com"></script>
  <script src="https://unpkg.com/lucide@latest"></script>
  <style>
    body {
      font-family: 'Inter', sans-serif;
    }
  </style>
</head>
<body class="bg-gray-50 text-gray-800">
  <!-- Header -->
  <header class="bg-white shadow p-4 flex justify-between items-center sticky top-0 z-
50">
    <div class="flex items-center gap-2">
      <i data-lucide="heart" class="w-6 h-6 text-red-600"></i>
      <h1 class="text-xl font-semibold">HealthAI – Кабінет лікаря</h1>
    </div>
    <button class="bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-
700">Вийти</button>
  </header>

  <!-- Welcome Section -->
  <section class="bg-blue-100 text-center py-12">

```

```
<h2 class="text-3xl font-bold text-gray-800">Ласкаво просимо, Олег  
Миколайович</h2>
```

```
<p class="text-gray-600 mt-2">Слідкуйте за графіком прийомів, ведіть  
призначення та переглядайте історію пацієнтів</p>
```

```
</section>
```

```
<!-- Doctor Dashboard -->
```

```
<main class="max-w-6xl mx-auto px-6 py-12 grid gap-8">
```

```
<div class="grid md:grid-cols-2 gap-8">
```

```
<!-- Today's Appointments -->
```

```
<div class="bg-white p-6 rounded shadow">
```

```
<h3 class="text-lg font-semibold mb-4 flex items-center gap-2">
```

```
<i data-lucide="calendar" class="w-5 h-5 text-green-500"></i> Сьогоднішні  
прийоми
```

```
</h3>
```

```
<ul class="text-sm space-y-2">
```

```
<li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-  
blue-600 hover:underline">09:00 – Іваненко Марія (терапевт)</a></li>
```

```
<li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-  
blue-600 hover:underline">11:30 – Степаненко Юрій (кардіо перевірка)</a></li>
```

```
<li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-  
blue-600 hover:underline">14:15 – Олійник Анастасія (огляд)</a></li>
```

```
</ul>
```

```
</div>
```

```
<!-- Action Panel -->
```

```
<div class="bg-white p-6 rounded shadow">
```

```
<h3 class="text-lg font-semibold mb-4 flex items-center gap-2">
```

```
<i data-lucide="clipboard" class="w-5 h-5 text-indigo-500"></i> Швидкі дії
```

```
</h3>
```

```
<div class="flex flex-col gap-3">
```

```
  <button class="bg-indigo-600 text-white px-4 py-2 rounded hover:bg-indigo-700 w-full">Додати призначення</button>
```

```
  <button class="bg-green-600 text-white px-4 py-2 rounded hover:bg-green-700 w-full">Переглянути записи</button>
```

```
  <button class="bg-yellow-500 text-white px-4 py-2 rounded hover:bg-yellow-600 w-full">Створити медичну довідку</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Analytics -->
```

```
<div class="bg-white p-6 rounded shadow">
```

```
  <h3 class="text-lg font-semibold mb-4 flex items-center gap-2">
```

```
    <i data-lucide="bar-chart-3" class="w-5 h-5 text-blue-500"></i> Аналітика ефективності
```

```
  </h3>
```

```
  <p class="text-sm text-gray-700 mb-2">Прийоми цього тижня:  
<strong>12</strong></p>
```

```
  <p class="text-sm text-gray-700 mb-2">Середній рейтинг пацієнтів:  
<strong>4.8/5</strong></p>
```

```
</div>
```

```
<!-- Recent Patients -->
```

```
<div class="bg-white p-6 rounded shadow">
```

```
  <h3 class="text-lg font-semibold mb-4 flex items-center gap-2">
```

```
    <i data-lucide="users" class="w-5 h-5 text-purple-500"></i> Останні активні пацієнти
```

```
</h3>
```

```
<ul class="space-y-2 text-sm">
```

```
  <li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-blue-600 hover:underline">Іваненко Марія</a> – Останній візит: 05.05.2025 – <em>терапевт</em></li>
```

```
  <li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-blue-600 hover:underline">Олійник Анастасія</a> – Останній візит: 01.05.2025 – <em>огляд</em></li>
```

```
  <li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-blue-600 hover:underline">Савчук Богдан</a> – Останній візит: 29.04.2025 – <em>кардіо</em></li>
```

```
</ul>
```

```
</div>
```

```
</main>
```

```
<!-- Footer -->
```

```
<footer class="text-center text-sm text-gray-500 py-6 border-t bg-white mt-12">
```

```
<div class="space-x-4">
```

```
<a href="#" class="hover:underline">Про нас</a>
```

```
<a href="#" class="hover:underline">Умови користування</a>
```

```
<a href="#" class="hover:underline">Політика конфіденційності</a>
```

```
</div>
```

```
<p class="mt-2">© 2025 HealthAI. Усі права захищено.</p>
```

```
</footer>
```

```
<!-- Модальне вікно – Картка пацієнта -->
```

```
<div id="patientModal" class="fixed inset-0 bg-black bg-opacity-50 hidden items-center justify-center z-50">
```

```
<div class="bg-white w-full max-w-3xl p-6 rounded shadow-lg relative overflow-y-
auto max-h-[90vh]">
```

```
<button onclick="closePatientModal()" class="absolute top-2 right-2 text-gray-
500 hover:text-red-600 text-lg font-bold">&times;</button>
```

```
<h2 class="text-2xl font-semibold mb-4">Картка пацієнта</h2>
```

```
<!-- 1. Загальна інформація -->
```

```
<div class="mb-4">
```

```
<h3 class="text-lg font-semibold mb-2">Загальна інформація</h3>
```

```
<ul class="text-sm text-gray-700 space-y-1">
```

```
<li><strong>Ім'я:</strong> Іваненко Марія</li>
```

```
<li><strong>Вік:</strong> 32</li>
```

```
<li><strong>Стать:</strong> Жіноча</li>
```

```
<li><strong>Телефон:</strong> +380 97 123 4567</li>
```

```
<li><strong>Лікар:</strong> Олег Миколайович</li>
```

```
</ul>
```

```
</div>
```

```
<!-- 2. Медична історія -->
```

```
<div class="mb-4">
```

```
<h3 class="text-lg font-semibold mb-2">Медична історія</h3>
```

```
<ul class="text-sm text-gray-700 space-y-1">
```

```
<li><strong>Алергії:</strong> Немає</li>
```

```
<li><strong>Хронічні захворювання:</strong> Астма</li>
```

```
<li><strong>Дата останнього прийому:</strong> 05.05.2025</li>
```

```
<li><strong>Стан пацієнта:</strong> Задовільний</li>
```

```
</ul>
```

```
</div>
```

```
<!-- 3. Призначення -->
```

```
<div class="mb-4">
```

```
<h3 class="text-lg font-semibold mb-2">Призначення</h3>
```

```
<ul class="text-sm text-gray-700 space-y-1 mb-2">
```

```
<li>Інгаляції – 2 рази на день</li>
```

```
<li>Аналіз крові – повторно через 2 тижні</li>
```

```
</ul>
```

```
<button class="bg-indigo-600 text-white px-4 py-2 rounded hover:bg-indigo-700 text-sm">Додати призначення</button>
```

```
</div>
```

```
<!-- 4. Довідки / Документи -->
```

```
<div class="mb-4">
```

```
<h3 class="text-lg font-semibold mb-2">Довідки / Документи</h3>
```

```
<ul class="text-sm text-gray-700 space-y-1 mb-2">
```

```
<li>Довідка №1 – 02.05.2025</li>
```

```
<li>Довідка №2 – 10.04.2025</li>
```

```
</ul>
```

```
<button class="bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-700 text-sm">Завантажити останню</button>
```

```
</div>
```

```
<!-- 4.5. HealthAI – коротке зведення -->
```

```
<div class="mt-4 text-right">
```

```
<a href="healthai-summary.html" class="inline-flex items-center gap-2 bg-blue-700 text-white px-4 py-2 rounded hover:bg-blue-800 text-sm">
```

```
<i data-lucide="arrow-right-circle" class="w-4 h-4"></i>
```

```
Перейти до повного зведення
```

```
</a>
```

```
</div>
```

```
<div class="mb-4 border-1-4 border-blue-500 bg-blue-50 p-4 rounded shadow-inner">
```

```
<div class="flex items-center gap-2 mb-2">
```

```
<i data-lucide="brain-cog" class="w-5 h-5 text-blue-600"></i>
```

```
<h3 class="text-lg font-semibold text-blue-800">HealthAI – коротке зведення</h3>
```

```
</div>
```

```
<ul class="text-sm text-blue-900 space-y-1">
```

```
<li><strong>Прогноз:</strong> стан стабільний, можливе покращення за умов дотримання режиму лікування</li>
```

```
<li><strong>Рекомендація:</strong> продовжити інгаляції ще 5 днів, уникати фізичних навантажень</li>
```

```
<li><strong>Попередження:</strong> ризик загострення у разі повторного контакту з алергенами</li>
```

```
</ul>
```

```
</div>
```

```
<!-- 5. Дії -->
```

```
<div class="flex justify-between mt-6">
```

```
<button class="bg-yellow-500 text-white px-4 py-2 rounded hover:bg-yellow-600 text-sm">Створити нову довідку</button>
```

```
<button class="bg-gray-700 text-white px-4 py-2 rounded hover:bg-gray-800 text-sm">Додати коментар до історії</button>
```

```
<button onclick="closePatientModal()" class="bg-red-600 text-white px-4 py-2 rounded hover:bg-red-700 text-sm">Закрити картку</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Scripts -->
```

```
<script>
```

```
  lucide.createIcons();
```

```
  function openPatientModal() {
```

```
    const modal = document.getElementById('patientModal');
```

```
    modal.classList.remove('hidden');
```

```
    modal.classList.add('flex');
```

```
  }
```

```
  function closePatientModal() {
```

```
    const modal = document.getElementById('patientModal');
```

```
    modal.classList.remove('flex');
```

```
    modal.classList.add('hidden');
```

```
  }
```

```
</script>
```

```
</body>
```

```
</html>
```

```
<!-- doctor-dashboard.html -->
```

```
<!-- Кабінет лікаря з секцією прийомів, швидкими діями, аналітикою та  
модальним вікном пацієнта -->
```

```
<!DOCTYPE html>
```

```
<html lang="uk">
```

```
<head>
```

```

<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Кабінет лікаря</title>
<link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap"
rel="stylesheet">
<script src="https://cdn.tailwindcss.com"></script>
<script src="https://unpkg.com/lucide@latest"></script>
<style>
  body {
    font-family: 'Inter', sans-serif;
  }
</style>
</head>
<body class="bg-gray-50 text-gray-800">
  <!-- Header -->
  <header class="bg-white shadow p-4 flex justify-between items-center sticky top-0 z-
50">
    <div class="flex items-center gap-2">
      <i data-lucide="heart" class="w-6 h-6 text-red-600"></i>
      <h1 class="text-xl font-semibold">HealthAI – Кабінет лікаря</h1>
    </div>
    <button class="bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-
700">Вийти</button>
  </header>

  <!-- Welcome Section -->
  <section class="bg-blue-100 text-center py-12">

```

```
<h2 class="text-3xl font-bold text-gray-800">Ласкаво просимо, Олег  
Миколайович</h2>
```

```
<p class="text-gray-600 mt-2">Слідкуйте за графіком прийомів, ведіть  
призначення та переглядайте історію пацієнтів</p>
```

```
</section>
```

```
<!-- Doctor Dashboard -->
```

```
<main class="max-w-6xl mx-auto px-6 py-12 grid gap-8">
```

```
<div class="grid md:grid-cols-2 gap-8">
```

```
<!-- Today's Appointments -->
```

```
<div class="bg-white p-6 rounded shadow">
```

```
<h3 class="text-lg font-semibold mb-4 flex items-center gap-2">
```

```
<i data-lucide="calendar" class="w-5 h-5 text-green-500"></i> Сьогоднішні  
прийоми
```

```
</h3>
```

```
<ul class="text-sm space-y-2">
```

```
<li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-  
blue-600 hover:underline">09:00 – Іваненко Марія (терапевт)</a></li>
```

```
<li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-  
blue-600 hover:underline">11:30 – Степаненко Юрій (кардіо перевірка)</a></li>
```

```
<li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-  
blue-600 hover:underline">14:15 – Олійник Анастасія (огляд)</a></li>
```

```
</ul>
```

```
</div>
```

```
<!-- Action Panel -->
```

```
<div class="bg-white p-6 rounded shadow">
```

```
<h3 class="text-lg font-semibold mb-4 flex items-center gap-2">
```

```
<i data-lucide="clipboard" class="w-5 h-5 text-indigo-500"></i> Швидкі дії
```

```
</h3>
```

```
<div class="flex flex-col gap-3">
```

```
  <button class="bg-indigo-600 text-white px-4 py-2 rounded hover:bg-indigo-700 w-full">Додати призначення</button>
```

```
  <button class="bg-green-600 text-white px-4 py-2 rounded hover:bg-green-700 w-full">Переглянути записи</button>
```

```
  <button class="bg-yellow-500 text-white px-4 py-2 rounded hover:bg-yellow-600 w-full">Створити медичну довідку</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Analytics -->
```

```
<div class="bg-white p-6 rounded shadow">
```

```
  <h3 class="text-lg font-semibold mb-4 flex items-center gap-2">
```

```
    <i data-lucide="bar-chart-3" class="w-5 h-5 text-blue-500"></i> Аналітика ефективності
```

```
  </h3>
```

```
  <p class="text-sm text-gray-700 mb-2">Прийоми цього тижня:  
<strong>12</strong></p>
```

```
  <p class="text-sm text-gray-700 mb-2">Середній рейтинг пацієнтів:  
<strong>4.8/5</strong></p>
```

```
</div>
```

```
<!-- Recent Patients -->
```

```
<div class="bg-white p-6 rounded shadow">
```

```
  <h3 class="text-lg font-semibold mb-4 flex items-center gap-2">
```

```
    <i data-lucide="users" class="w-5 h-5 text-purple-500"></i> Останні активні пацієнти
```

```
</h3>
```

```
<ul class="space-y-2 text-sm">
```

```
  <li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-blue-600 hover:underline">Іваненко Марія</a> – Останній візит: 05.05.2025 – <em>терапевт</em></li>
```

```
  <li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-blue-600 hover:underline">Олійник Анастасія</a> – Останній візит: 01.05.2025 – <em>огляд</em></li>
```

```
  <li><a href="javascript:void(0)" onclick="openPatientModal()" class="text-blue-600 hover:underline">Савчук Богдан</a> – Останній візит: 29.04.2025 – <em>кардіо</em></li>
```

```
</ul>
```

```
</div>
```

```
</main>
```

```
<!-- Footer -->
```

```
<footer class="text-center text-sm text-gray-500 py-6 border-t bg-white mt-12">
```

```
<div class="space-x-4">
```

```
<a href="#" class="hover:underline">Про нас</a>
```

```
<a href="#" class="hover:underline">Умови користування</a>
```

```
<a href="#" class="hover:underline">Політика конфіденційності</a>
```

```
</div>
```

```
<p class="mt-2">© 2025 HealthAI. Усі права захищено.</p>
```

```
</footer>
```

```
<!-- Модальне вікно – Картка пацієнта -->
```

```
<div id="patientModal" class="fixed inset-0 bg-black bg-opacity-50 hidden items-center justify-center z-50">
```

```
<div class="bg-white w-full max-w-3xl p-6 rounded shadow-lg relative overflow-y-
auto max-h-[90vh]">
```

```
<button onclick="closePatientModal()" class="absolute top-2 right-2 text-gray-
500 hover:text-red-600 text-lg font-bold">&times;</button>
```

```
<h2 class="text-2xl font-semibold mb-4">Картка пацієнта</h2>
```

```
<!-- 1. Загальна інформація -->
```

```
<div class="mb-4">
```

```
<h3 class="text-lg font-semibold mb-2">Загальна інформація</h3>
```

```
<ul class="text-sm text-gray-700 space-y-1">
```

```
<li><strong>Ім'я:</strong> Іваненко Марія</li>
```

```
<li><strong>Вік:</strong> 32</li>
```

```
<li><strong>Стать:</strong> Жіноча</li>
```

```
<li><strong>Телефон:</strong> +380 97 123 4567</li>
```

```
<li><strong>Лікар:</strong> Олег Миколайович</li>
```

```
</ul>
```

```
</div>
```

```
<!-- 2. Медична історія -->
```

```
<div class="mb-4">
```

```
<h3 class="text-lg font-semibold mb-2">Медична історія</h3>
```

```
<ul class="text-sm text-gray-700 space-y-1">
```

```
<li><strong>Алергії:</strong> Немає</li>
```

```
<li><strong>Хронічні захворювання:</strong> Астма</li>
```

```
<li><strong>Дата останнього прийому:</strong> 05.05.2025</li>
```

```
<li><strong>Стан пацієнта:</strong> Задовільний</li>
```

```
</ul>
```

```
</div>
```

```
<!-- 3. Призначення -->
```

```
<div class="mb-4">
```

```
<h3 class="text-lg font-semibold mb-2">Призначення</h3>
```

```
<ul class="text-sm text-gray-700 space-y-1 mb-2">
```

```
<li>Інгаляції – 2 рази на день</li>
```

```
<li>Аналіз крові – повторно через 2 тижні</li>
```

```
</ul>
```

```
<button class="bg-indigo-600 text-white px-4 py-2 rounded hover:bg-indigo-700 text-sm">Додати призначення</button>
```

```
</div>
```

```
<!-- 4. Довідки / Документи -->
```

```
<div class="mb-4">
```

```
<h3 class="text-lg font-semibold mb-2">Довідки / Документи</h3>
```

```
<ul class="text-sm text-gray-700 space-y-1 mb-2">
```

```
<li>Довідка №1 – 02.05.2025</li>
```

```
<li>Довідка №2 – 10.04.2025</li>
```

```
</ul>
```

```
<button class="bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-700 text-sm">Завантажити останню</button>
```

```
</div>
```

```
<!-- 4.5. HealthAI – коротке зведення -->
```

```
<div class="mt-4 text-right">
```

```
<a href="healthai-summary.html" class="inline-flex items-center gap-2 bg-blue-700 text-white px-4 py-2 rounded hover:bg-blue-800 text-sm">
```

```
<i data-lucide="arrow-right-circle" class="w-4 h-4"></i>
```

```
Перейти до повного зведення
```

```
</a>
```

```
</div>
```

```
<div class="mb-4 border-1-4 border-blue-500 bg-blue-50 p-4 rounded shadow-inner">
```

```
<div class="flex items-center gap-2 mb-2">
```

```
<i data-lucide="brain-cog" class="w-5 h-5 text-blue-600"></i>
```

```
<h3 class="text-lg font-semibold text-blue-800">HealthAI – коротке зведення</h3>
```

```
</div>
```

```
<ul class="text-sm text-blue-900 space-y-1">
```

```
<li><strong>Прогноз:</strong> стан стабільний, можливе покращення за умов дотримання режиму лікування</li>
```

```
<li><strong>Рекомендація:</strong> продовжити інгаляції ще 5 днів, уникати фізичних навантажень</li>
```

```
<li><strong>Попередження:</strong> ризик загострення у разі повторного контакту з алергенами</li>
```

```
</ul>
```

```
</div>
```

```
<!-- 5. Дії -->
```

```
<div class="flex justify-between mt-6">
```

```
<button class="bg-yellow-500 text-white px-4 py-2 rounded hover:bg-yellow-600 text-sm">Створити нову довідку</button>
```

```
<button class="bg-gray-700 text-white px-4 py-2 rounded hover:bg-gray-800 text-sm">Додати коментар до історії</button>
```

```
<button onclick="closePatientModal()" class="bg-red-600 text-white px-4 py-2 rounded hover:bg-red-700 text-sm">Закрити картку</button>
```

```
</div>
```

```
</div>
</div>

<!-- Scripts -->
<script>
  lucide.createIcons();

  function openPatientModal() {
    const modal = document.getElementById('patientModal');
    modal.classList.remove('hidden');
    modal.classList.add('flex');
  }

  function closePatientModal() {
    const modal = document.getElementById('patientModal');
    modal.classList.remove('flex');
    modal.classList.add('hidden');
  }
</script>
</body>
</html>
□
```

Презентація до роботи

Управління проектом розробки системи підтримки прийняття рішення із застосуванням підходів штучного інтелекту

Підготував студент групи ІСТ-УП-21 Федянін К.О.
Керівник: Веренич О.В.
КНУБА 2025

Потреба СППР у медицині Без СППР (традиційний підхід)

- Рішення ухвалюється виключно лікарем під тиском часу
- Висока ймовірність помилки через втому, стрес
- Важко врахувати всі протоколи, дослідження та джерела одночасно
- Суб'єктивність діагнозу в складних або неочевидних ситуаціях
- Відсутність єдиного стандарту при рідкісних захворюваннях

Потреба СППР у медицині Із СППР (на основі ШІ)

- Система пропонує обґрунтовані варіанти дій
- Зниження ризику помилки завдяки аналізу множинних факторів
- Система враховує оновлені дані з баз знань та досліджень
- Прозора логіка та пояснення рекомендації
- Алгоритмічний підхід до аналізу нетипових випадків

Історія СППР та сучасний контекст

1950-1990 роки

Перші системи підтримки прийняття рішень базувалися на жорстко визначених правилах типу «якщо — то».

Їхнє завдання — допомогти фахівцю у виборі альтернатив шляхом фільтрації або сортування інформації. Знання закладалися вручну у вигляді логічних дерев або продукційних моделей. Такі СППР використовувалися в медицині, техніці, юриспруденції. Основним недоліком була відсутність гнучкості: будь-яке нове правило потребувало ручного переписування коду. Вони не адаптувалися до змін і не накопичували досвід.

1990-2010 роки

З розвитком машинного навчання СППР почали опрацьовувати великі обсяги історичних даних і формувати моделі на їх основі. Це дозволило системам автоматично адаптуватися до змін, виявляти закономірності, прогнозувати результати. Однак ці моделі були «чорними скриньками» — вони не пояснювали логіку ухваленого рішення. Хоча точність відповідей зросла, прозорість і довіра до результатів залишалися проблемними. Такий підхід став основою для фінансових та управлінських СППР.

2010-2020 роки

Спроба поєднати аналітичні можливості машинного навчання з пояснюваністю rule-based підходу призвела до створення гібридних СППР.

Вони об'єднували правила, прогнози, візуалізацію й оцінку сценаріїв в одному інтерфейсі.

Такі системи частково могли обґрунтувати свої рекомендації, однак вимагали складної конфігурації, дорогих налаштувань і підтримки онтологій. Застосовувалися в логістиці, управлінні, освіті, проте не отримали масового впровадження в медицині через складність адаптації.

2020- наші дні

Революційним стало впровадження великих мовних моделей, таких як GPT-4, що здатні розуміти природну мову, аналізувати контекст і генерувати пояснення. У поєднанні з семантичним пошуком (наприклад, DeepSeek) виникла архітектура Retrieval-Augmented Generation. Вона дозволяє моделі не лише відповідати, а й опиратись на реальні джерела. Це вирішує проблему «галюцинацій» і відкриває шлях до прозорих, обґрунтованих рішень. Системи стають доступними для широкого кола користувачів — без технічної підготовки.

Управління проектом

Ресурсне забезпечення та бюджет проекту

№	Стаття витрат	Кількість	Фактор	Опис впр.	Критерій
1	Project Manager	1	P – Політичні фактори	Законодавче регулювання сфери захисту персональних медичних даних. Необхідність відповідності GDPR щодо конфіденційності інформації.	Specific (специфічність)
2	HR Specialist	1			
3	Business Analyst	1			
4	Tech Lead	1			
5	Software Architect	1	E – Економічні фактори	Витрати на впровадження обмежені фінансуванням бюджету. Зацікавленість приватних клієнтів та скорочення витрат на систему.	Measurable (вимірюваність)
6	UI/UX Designer	1			
7	DevOps Engineer	1			
8	Frontend Developer	1	S – Соціальні фактори	Зростання очікувань суспільства щодо швидкості й безпеки медичних сервісів. Підвищення довіри до IT-рішень лікарів, водночас обережність досвідчених фахівців.	Achievable (досяжність)
9	Backend Developer	1			
10	QA Engineer	1			
11	Technical Writer	1	T – Технологічні фактори	Активний розвиток штучного інтелекту в медичних аналітичних платформах. Використання мобільних додатків для медичних інформаційних систем.	Relevant (релевантність)
12	General Developer	1			
13	Product Owner	1			

СТАТУТ ПРОЕКТУ

Назва проекту: Управління проектом розробки системи підтримки прийняття рішень із застосуванням підходів штучного інтелекту для медичних закладів

Спонсор проекту: –

Дата розробки: 2025-03-01

Керівник проекту: Федівін Кирило Олександрович

Замовник проекту: Медичний заклад

Мета проекту або обґрунтування: Розробка інтегрованої системи підтримки прийняття рішень (СППР) HealthAI для медичних закладів, яка послужить обробці запитів природною мовою, генерацією рекомендацій на основі медичних даних, верифікацією фактів і прозору взаємодії з користувачем, з урахуванням етичних вимог і вимог до конфіденційності.

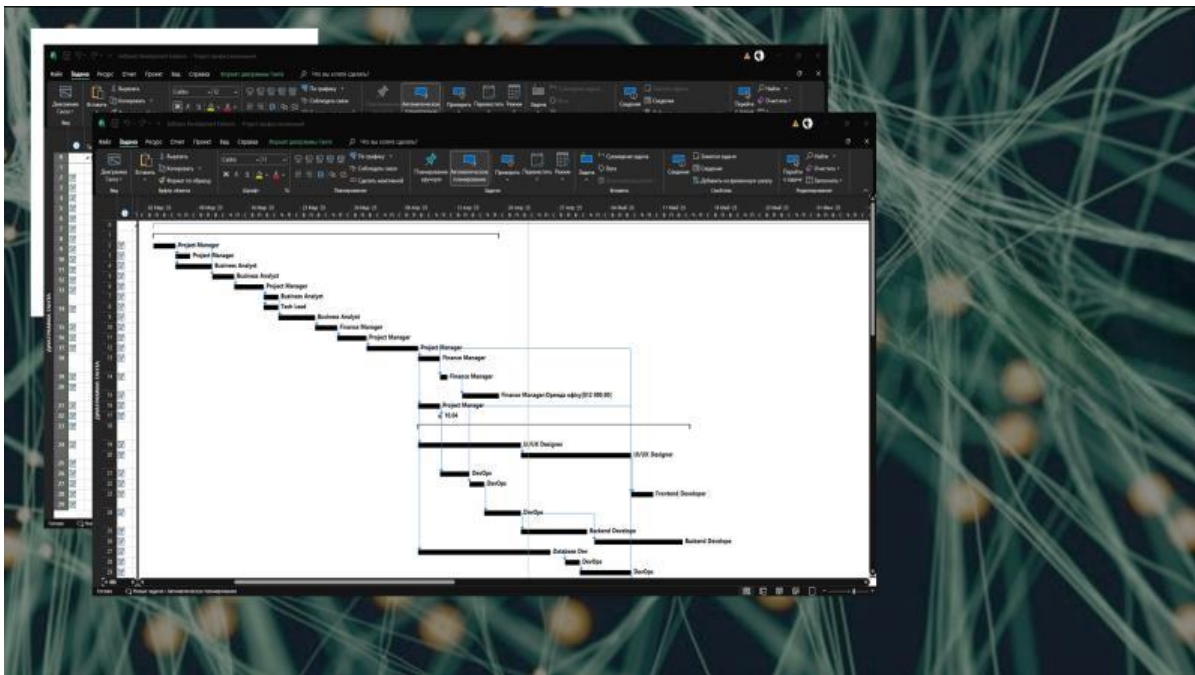
Опис проекту: Проект охоплює повний життєвий цикл створення СППР та включає такі компоненти:

- Архітектуру із централізованим модулем обробки запитів і доступом до бази медичних даних.
- Інтелектуальний мовний модуль GPT-4 для генерації пояснень і рекомендацій.
- Аналітичний модуль DeepSeek для RAG-підходу (retrieval-augmented generation).
- Модуль взаємодії з користувачем (інтерфейс пацієнта/лікаря).
- Механізми логування, збереження сесій та безпечної обробки запитів.
- Діаграму потоків даних (DFD).

Вимоги високого рівня:

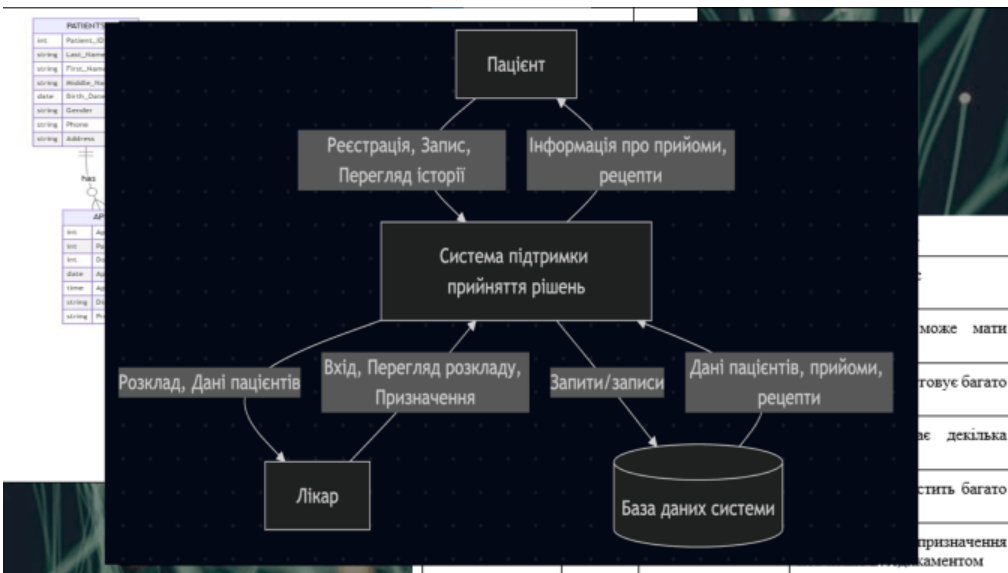
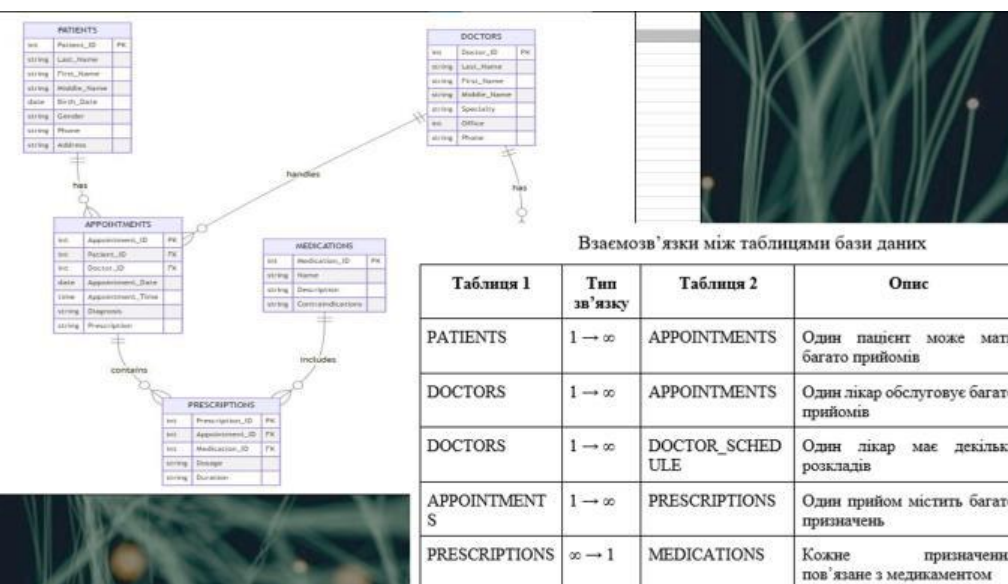
- Централізована структура із простим потоком даних.
- Точність відповідей: $\geq 80\%$.
- Час відповіді системи: ≤ 3 с.
- Сумісність з медичною інфраструктурою (eHealth, API).
- Безпека даних: ізоляція даних, шифрування, анонімізація.
- Повисованість III-рівня (explainability).

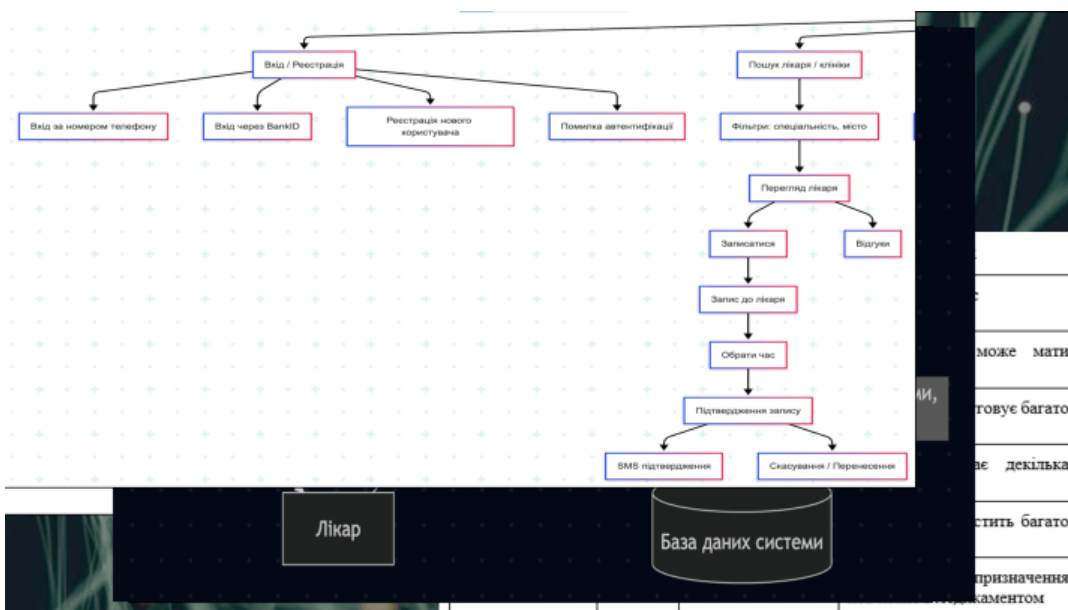
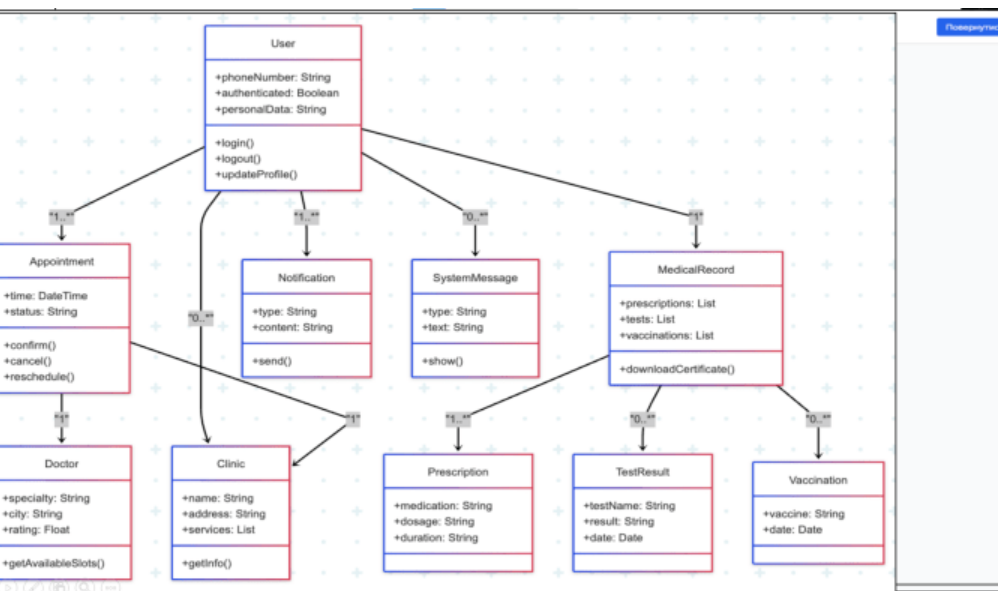
№	Назва завдання	Тип завдання	Початок	Кінець	Ресурс	Відсоток виконання
1	Вибір технології розробки	Task	01.03.2025	01.03.2025		100%
2	Мета-аналіз	Task	01.03.2025	01.03.2025		100%
3	Визначення цілей проекту	Task	01.03.2025	01.03.2025	Project Manager	100%
4	Визначення складових елементів проекту	Task	01.03.2025	01.03.2025	Project Manager	100%
5	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
6	Визначення ролей	Task	01.03.2025	01.03.2025	Project Manager	100%
7	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
8	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
9	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
10	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
11	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
12	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
13	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
14	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
15	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
16	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
17	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
18	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
19	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
20	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
21	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
22	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
23	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
24	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
25	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
26	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
27	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
28	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
29	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
30	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
31	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
32	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
33	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
34	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
35	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
36	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
37	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
38	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
39	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
40	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
41	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
42	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
43	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
44	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
45	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
46	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
47	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
48	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
49	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%
50	Визначення ролей	Task	01.03.2025	01.03.2025	Business Analyst	100%

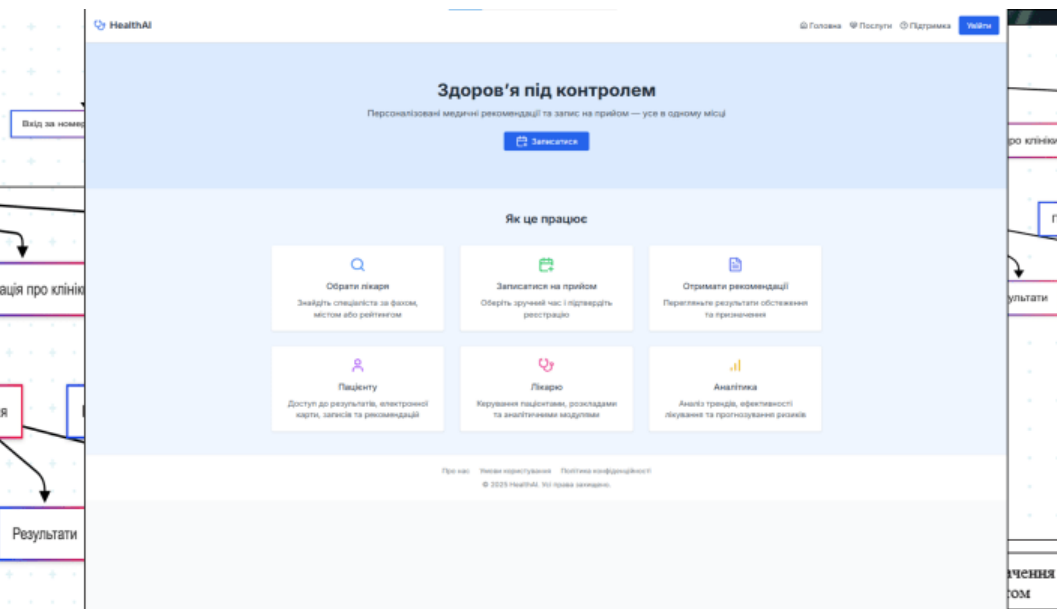
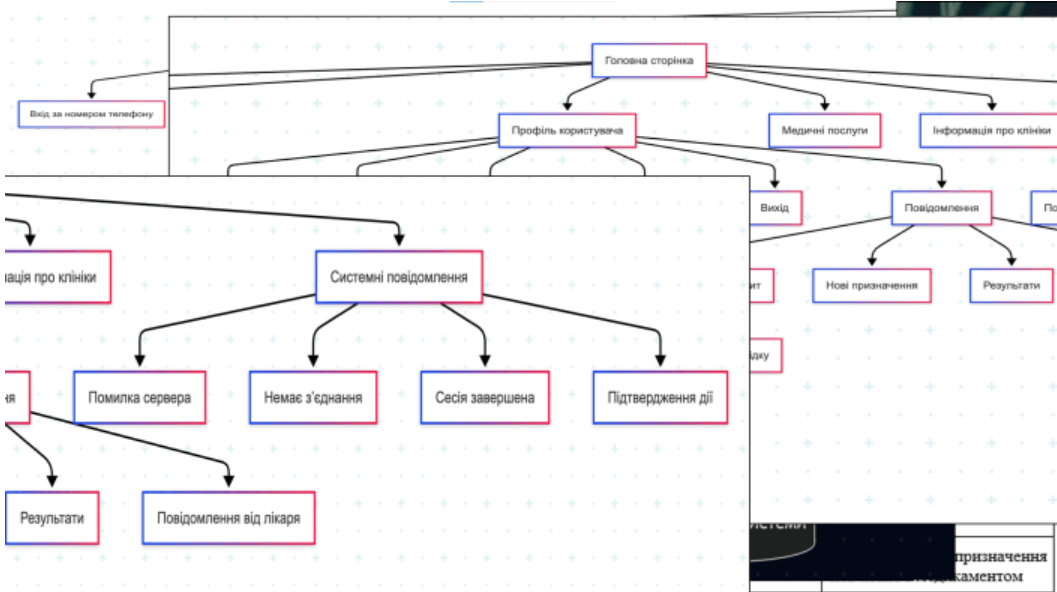
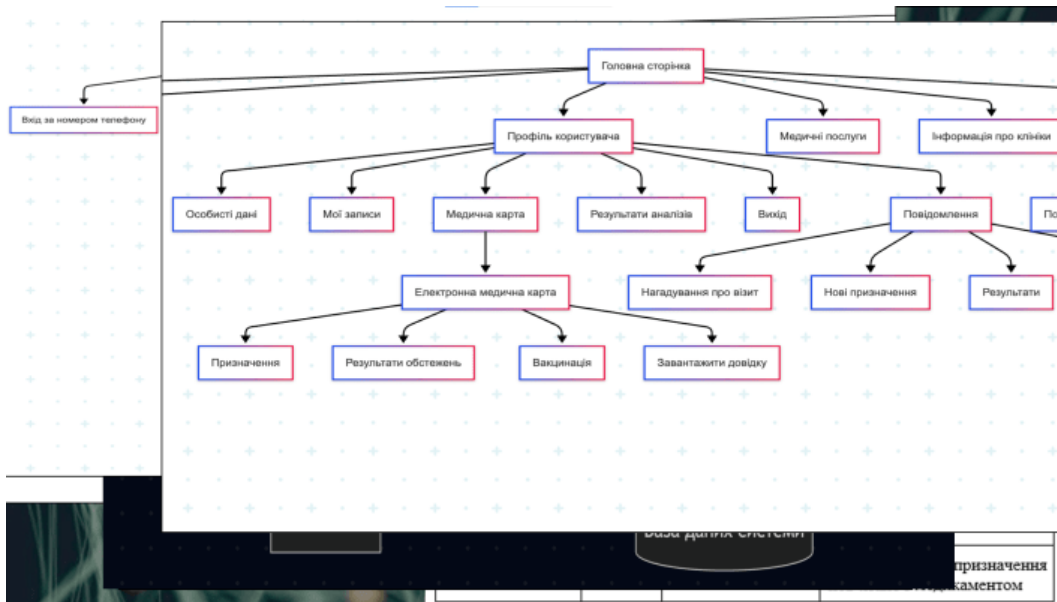


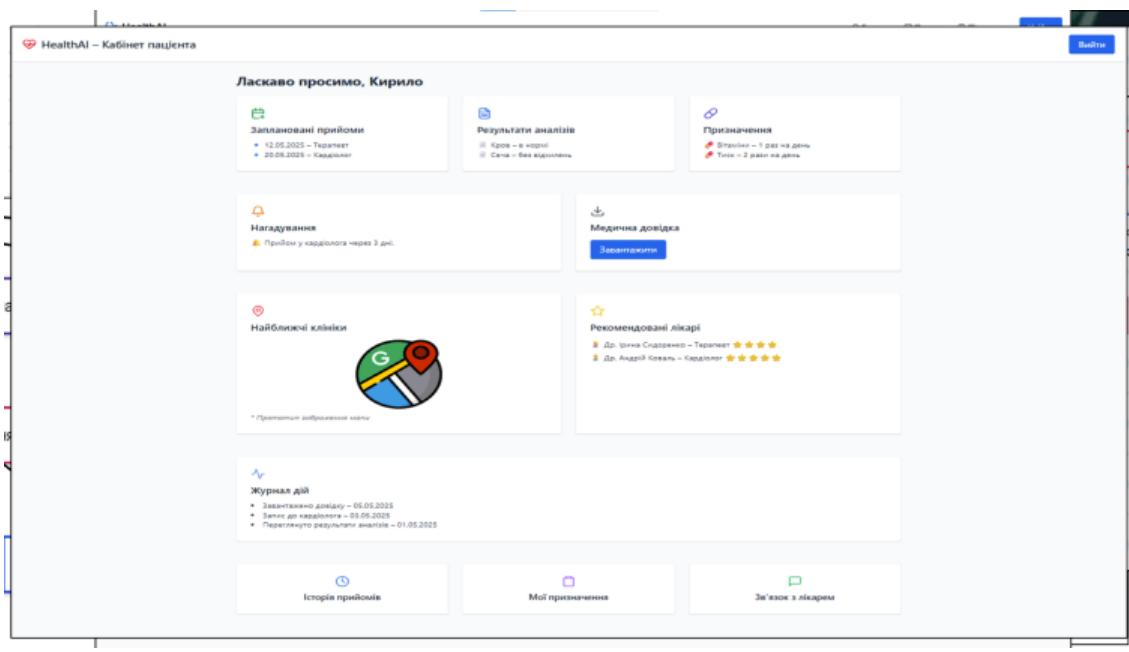
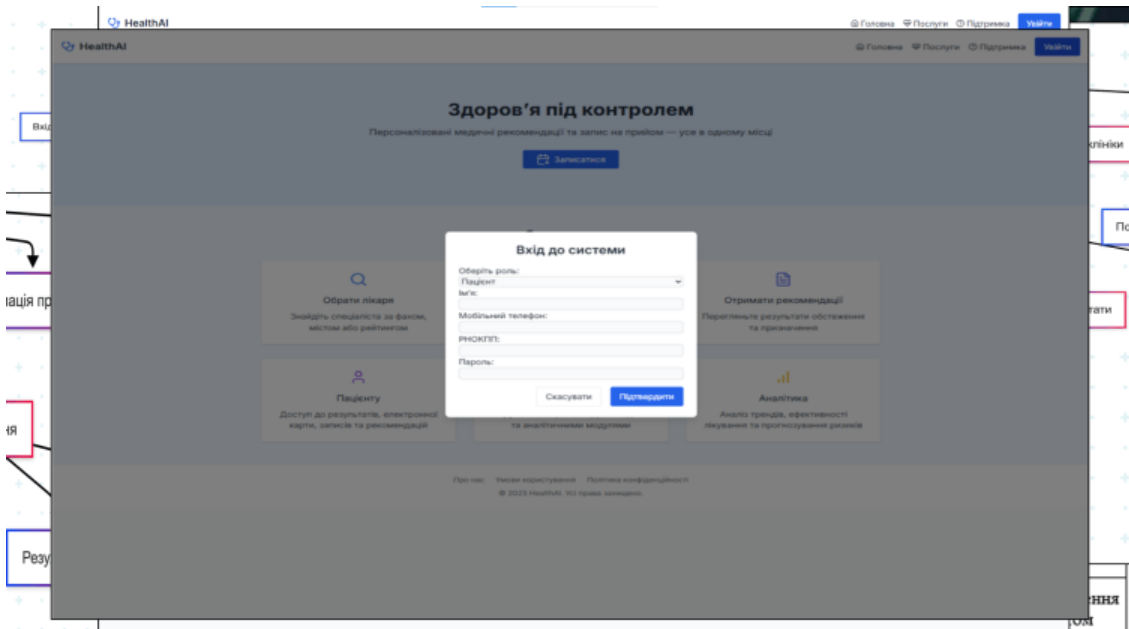
Технічні моменти

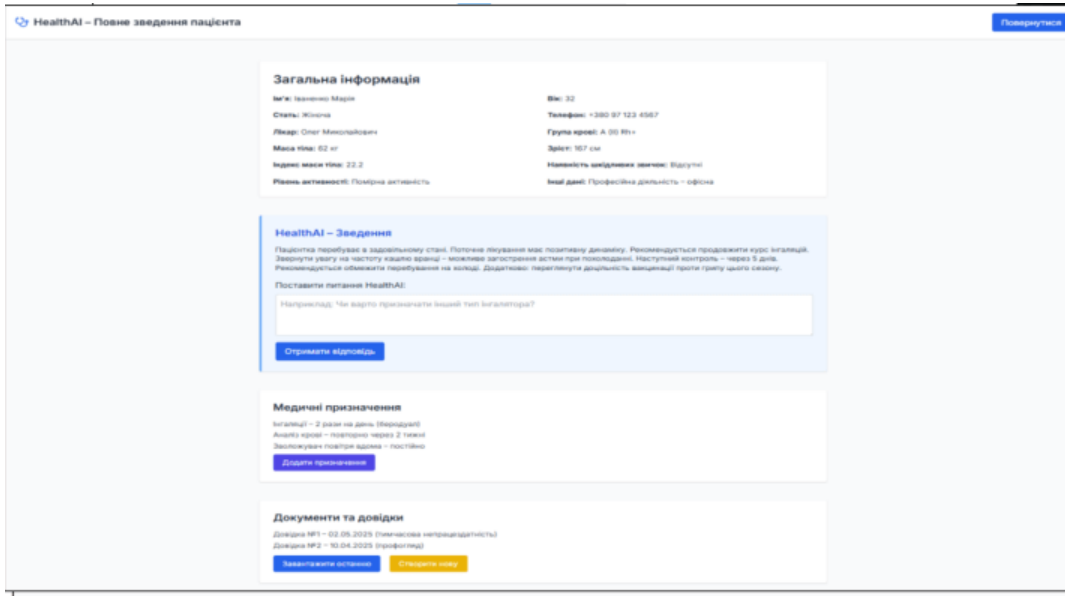
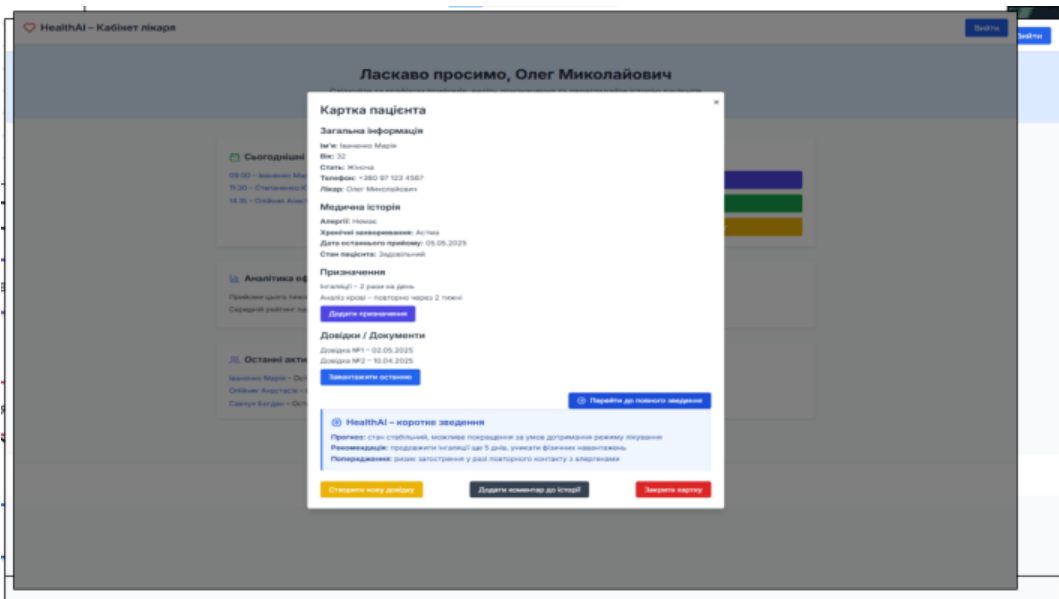
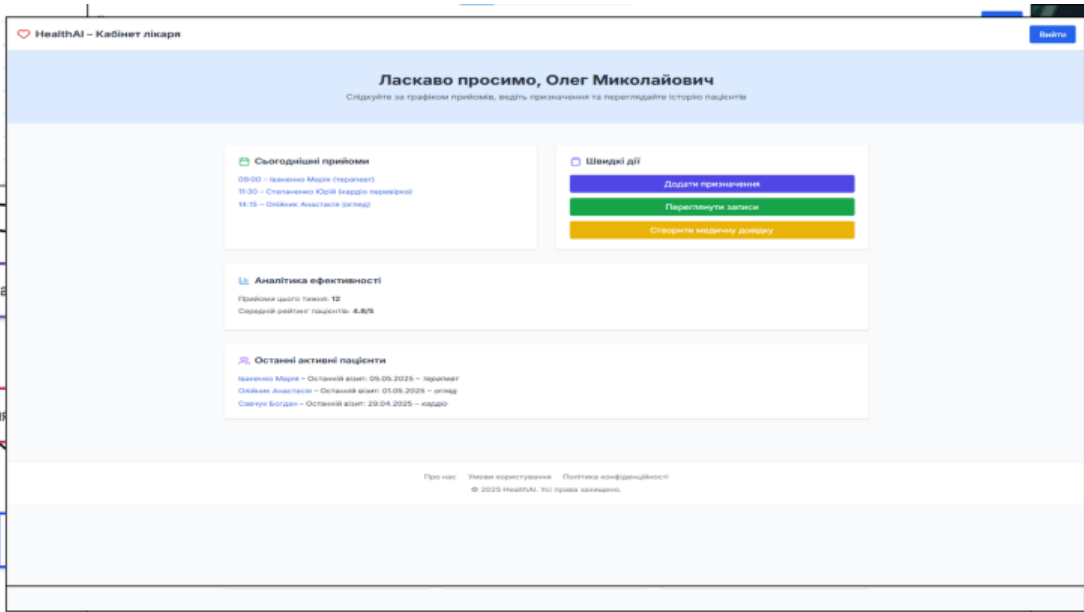
Year	Year	Location	Location	Site	Type	Definition
2000	2000	US	United States	BRIS	Health Status	Recent activity limitation among adults
2000	2014	AR	Arkansas	US Cancer 201	Cancer	Invasive cancer (all sites combined), incidence
2000	2014	CA	California	US Cancer 201	Cancer	Cervical cancer (all sites combined), underlying cause
2000	2014	CO	Colorado	US Cancer 201	Cancer	Invasive cancer (all sites combined), incidence
2000	2014	GA	Georgia	US Cancer 201	Cancer	Prostate cancer (all sites combined), underlying cause
2000	2014	KS	Kansas	US Cancer 201	Cancer	Invasive cancer (all sites combined), incidence
2000	2014	MA	Massachusetts	US Cancer 201	Cancer	Invasive cancer (all sites combined), incidence
2000	2014	NJ	New Jersey	US Cancer 201	Cancer	Invasive cancer (all sites combined), incidence
2000	2014	NY	New York	US Cancer 201	Cancer	Bladder cancer (all sites combined), underlying cause
2000	2014	OR	Oregon	US Cancer 201	Cancer	Bladder cancer (all sites combined), underlying cause
2000	2014	SD	South Dakota	US Cancer 201	Cancer	Prostate cancer (all sites combined), underlying cause
2000	2014	TX	Texas	US Cancer 201	Cancer	Cervical cancer (all sites combined), underlying cause
2000	2014	US	United States	US Cancer 201	Cancer	Invasive cancer (all sites combined) mortality among all people, underlying cause
2000	2014	VT	Vermont	US Cancer 201	Cancer	Prostate cancer (all sites combined), underlying cause
2000	2014	VA	West Virginia	US Cancer 201	Cancer	Bladder cancer (all sites combined), underlying cause
2000	2014	WI	Wisconsin	US Cancer 201	Cancer	Invasive cancer (all sites combined), incidence
2000	2014	WY	Wyoming	US Cancer 201	Cancer	Diabetic retinopathy (all sites combined), underlying or contributing cause
2000	2014	AK	Alaska	US Cancer 201	Cancer	Bladder cancer (all sites combined), underlying cause
2000	2014	AZ	Arizona	US Cancer 201	Cancer	Lung and bronchovascular cancer (all sites combined), underlying cause
2000	2014	CA	California	US Cancer 201	Cancer	Bladder cancer (all sites combined), underlying cause
2000	2014	CO	Colorado	US Cancer 201	Cancer	Cervical cancer (all sites combined), underlying cause
2000	2014	DC	District of Columbia	US Cancer 201	Cancer	Invasive cancer (all sites combined), incidence
2000	2014	FL	Florida	US Cancer 201	Cancer	Bladder cancer (all sites combined), underlying cause
2000	2014	HI	Hawaii	US Cancer 201	Cancer	Invasive cancer (all sites combined), incidence
2000	2014	IA	Iowa	US Cancer 201	Cancer	Prostate cancer (all sites combined), underlying cause
2000	2014	IL	Illinois	US Cancer 201	Cancer	Invasive cancer (all sites combined), incidence
2000	2014	IN	Indiana	US Cancer 201	Cancer	Bladder cancer (all sites combined), underlying cause
2000	2014	KS	Kansas	US Cancer 201	Cancer	Bladder cancer (all sites combined), underlying cause
2000	2014	MA	Massachusetts	US Cancer 201	Cancer	Invasive cancer (all sites combined), incidence
2000	2014	MD	Maryland	US Cancer 201	Cancer	Cervical cancer (all sites combined), underlying cause
2000	2014	MI	Michigan	US Cancer 201	Cancer	Bladder cancer (all sites combined), underlying cause
2000	2014	MN	Minnesota	US Cancer 201	Cancer	Lung and bronchovascular cancer (all sites combined), underlying cause
2000	2014	MO	Missouri	US Cancer 201	Cancer	Bladder cancer (all sites combined), underlying cause
2000	2014	MS	Mississippi	US Cancer 201	Cancer	Lung and bronchovascular cancer (all sites combined), underlying cause
2000	2014	MT	Montana	US Cancer 201	Cancer	Diabetic retinopathy (all sites combined), underlying cause











Приклад кейсу

Назва кейсу	Перевірка генерації відповіді GPT-4 на типовий медичний запит
Вхідні дані	Запит: «Пацієнт має кашель, лихоманку, втому. Який діагноз можливий?»
Очікувана поведінка	Система формує відповідь мовною моделлю, яка містить імовірний діагноз та пояснення
Додатково	Перевіряється узгодженість відповіді з контекстом та логічна структурованість формулювання
Критерій успішності	Відповідь є змістовною, коректно побудованою та придатною для використання лікарем