НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
(повна назва інституту/факультету)

кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ
(повна назва кафедри)

«До захисту допущено»
В.о.завідувачки кафедри БМК

_____ Світлана АЛХІМОВА
(підпис) (ініціали, ПРІЗВИЩЕ

"___" __*червня*__ 2025р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Комп'ютерні технології в біології та медицині»
зі спеціальності *122 «Комп'ютерні науки»*
на тему: Чат-бот для консультації з медичних питань

Виконав: студент **IV** курсу, групи _БС-11__

Коваленко Вікторія Юріївна _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник: *старший викладач каф. біомедичної кібернетики (БМК) ст.викл., к.т.н. Піднебесна Галина Анатоліївна* _____
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по ініціали) (підпис)

Консультант з розділів дипломної роботи:
доцент каф ОППЦБ, доц., к.т.н Арламов Олександр Юрійович _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент: *асистент каф. біомедичної інженерії Матвєєва Ілона Олегівна* _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

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

Студентка _____
(підпис)

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет біомедичної інженерії
Кафедра біомедичної кібернетики

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні технології в біології та медицині»

ЗАТВЕРДЖУЮ
В.о. завідувач кафедри БМК
_____ Світлана АЛХІМОВА
«_24_» __травня___2025 р.

ЗАВДАННЯ
на дипломну роботу студентці

КОВАЛЕНКО ВІКТОРІЯ ЮРІЇВНА
(прізвище, ім'я, по батькові)

1. Тема роботи **Чат-бот для консультації з медичних питань**

Керівник роботи
Піднебесна Галина Альбертівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «_26_» _травня_ 2025 р. №_1758-*с*

2. Термін подання студентом роботи ***05-07 червня 2025р.***
3. Вихідні дані до роботи: Дані для виконання роботи були отримані з платформи GitHub
4. Зміст роботи: Розробити чат-бот у месенджері Telegram з метою надання попередніх консультацій з медичних питань, а також інформування користувачів щодо симптомів, профілактики та можливих шляхів звернення до лікаря. Зробити загальні висновки та визначитись з остаточною темою ДР до наказу.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) В звіті 18 рисунків, 3 таблиць. Презентація на 15 слайдах.
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Дипломної роботи	Арламов О.Ю. доц. каф. ОППЦБ		

7. Дата видачі завдання ***24 грудня 2024 року.***

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання за темою ДР на практику	До 24.12.2024р.	*виконано*
2	Переддипломна практика	За графіком	*виконано*
3	Виконання розділів ДР (Огляд існуючих методів розробки медичних чат-ботів, Засоби реалізації чат-боту, Програмна реалізація та робота чат-боту, Захист інформації.)	До кінця практики	*виконано*
4	Перевірка ДР керівником та допуск її на перевірку нормоконтролером та на плагіат	Не пізніше 27.05.2025р.	*виконано*
5	Подання в електронному вигляді ДР та анотації до неї на перевірку нормоконтролера та плагіат (SrikePlagiarism.com).	Не пізніше 02.06.2025р.	*виконано*
6	Надання пакету електронних документів на засідання кафедри	Не пізніше 05.06.2025р.	*виконано*
7	Отримати допуск до захисту ДР в ЕК	Згідно засіданню кафедри	*виконано*
8	Подання ДР рецензенту. Отримання рецензії.	Не пізніше 27.05.2029р.	*виконано*
10	Подання пакету документів в паперовому вигляді по ДР та супровідних до неї документів на захист в ЕК[1]	Не пізніше 13.06.2025р.	*виконано*
11	Захист ДР в ЕК	16.06.2025р.- 21.06.2025р.	

Студент _____ **Вікторія КОВАЛЕНКО**
(підпис) (ім'я, ПРІЗВИЩЕ)

Керівник ДР _____ **Галина ПІДНЕБЕСНА**
(підпис) (ім'я, ПРІЗВИЩЕ)

Нормоконтролер _____ **Галина КОРНІЄНКО**
(підпис) (ім'я, ПРІЗВИЩЕ)

[1] не пізніше ніж за 5 днів до затвердженої дати захисту ДР в ЕК

АНОТАЦІЯ

Дипломна робота за темою «Чат-бот для консультації з медичних питань» виконана студенткою кафедри біомедичної кібернетики ФБМІ *Коваленко Вікторією Юріївною зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та* складається зі: вступу; 3 розділів *(Огляд існуючих методів розробки медичних чат-ботів, Засоби реалізації чат-боту, Програмна реалізація та робота чат-боту, Захист інформації)*, висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 35 джерела. Загальний обсяг роботи 72 сторінок.

Актуальність теми. В умовах обмеженого доступу до лікарів, особливо під час війни чи епідемій, чат-боти відіграють важливу роль у наданні базової медичної інформації. Telegram-бот з використанням GPT-4 Turbo забезпечує користувачам зручний і безпечний канал для отримання попередніх медичних порад.

Мета і завдання роботи. Метою роботи є підвищення доступності орієнтовної медичної консультації для користувачів шляхом впровадження Telegram-бота, який поєднує мовну модель GPT-4 Turbo, сценарну логіку діалогу (FSM) та механізми локального збереження історії звернень і генерації звітів.

Завдання:

1. Проаналізувати існуючі рішення у сфері медичних чат-ботів.
2. Розробити функціональний Telegram-бот із GPT-інтеграцією, базою даних, FSM-логікою та формуванням PDF-звітів.
3. Забезпечити захист даних та анонімність користувача.

Використані методи.

Проєкт реалізовано мовою Python у середовищі Visual Studio Code із використанням бібліотек python-telegram-bot, openai, sqlite3, APScheduler, ReportLab.

Отримані результати.

Telegram-бот дозволяє анонімно фіксувати симптоми, надавати поради за допомогою GPT-4 Turbo, формувати історію звернень, створювати PDF-звіти та надсилати нагадування.

Ключові слова: Telegram-бот, GPT-4 Turbo, медична консультація, Python, штучний інтелект, FSM, SQLite, MedlinePlus.

Бібліографічний опис ДР

Коваленко В.Ю. Чат-бот для консультації з медичних питань: дипломна роб. бакалавра : 122 Комп’ютері науки / Коваленко Вікторія Юріївна. – Київ, 2025. – 72 с.

ABSTRACT

The thesis titled "Chatbot for Medical Consultations" was completed by Viktoriia Kovalenko, a student of the Department of Biomedical Cybernetics at the FBMI, specializing in 122 "Computer Sciences" under the educational-professional program "Computer Technologies in Biology and Medicine." The thesis consists of: an introduction; 3 sections (Overview of existing methods for developing medical chatbots, Tools for chatbot implementation, Software implementation and chatbot operation, Information protection); conclusions for each of these sections; general conclusions; a list of 35 sources used. The total length of the thesis is 72 pages.

Relevance of the topic.

In times of limited access to doctors, especially during war or epidemics, chatbots play an important role in providing basic medical information. The Telegram bot using GPT-4 Turbo offers users a convenient and secure channel for receiving preliminary medical advice.

Aim and objectives of the work. The aim of this work is to improve the accessibility of preliminary medical consultations for users by implementing a Telegram bot that combines the GPT-4 Turbo language model, finite-state dialogue logic (FSM), and mechanisms for local storage of consultation history and report generation.

Objectives:

1. Analyze existing solutions in the field of medical chatbots.
2. Develop a functional Telegram bot with GPT integration, a database, FSM logic, and PDF report generation.
3. Ensure data protection and user anonymity.

Methods used. The project was implemented in the Python programming language in the Visual Studio Code environment using the following libraries: python-telegram-bot, openai, sqlite3, APScheduler, and ReportLab.

Results: The Telegram bot enables users to anonymously input symptoms, receive advice via GPT-4 Turbo, store consultation history, generate PDF reports, and receive reminders.

Keywords: Telegram bot, GPT-4 Turbo, medical consultation, Python, artificial intelligence, FSM, SQLite, MedlinePlus.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

GPT – Generative Pre-trained Transformer (Генеративний попередньо навчений трансформер)

FSM – Finite-State Machine (Кінцева автоматна машина)

API – Application Programming Interface (Інтерфейс програмування застосунків)

GUI – Graphical User Interface (Графічний інтерфейс користувача)

PDF – Portable Document Format (Портативний формат документів)

AI – Artificial Intelligence (Штучний інтелект)

SQLite – Structured Query Language Lite (Полегшена реляційна база даних)

VS Code – Visual Studio Code (Редактор коду Visual Studio)

NLP – Natural Language Processing (Обробка природної мови)

МОЗ – Міністерство охорони здоров’я України

ВСТУП

Сучасні інформаційні технології відіграють дедалі важливішу роль у сфері охорони здоров’я, особливо у напрямах, що стосуються інформування пацієнтів, покращення комунікації та попереднього консультування. Цифрові рішення, зокрема мобільні застосунки та чат-боти, активно впроваджуються в клінічну практику як допоміжні інструменти, що доповнюють традиційні методи надання медичної допомоги.

Одним із перспективних напрямів є застосування чат-ботів, які працюють на базі великих мовних моделей. Завдяки інтеграції штучного інтелекту з популярними платформами, такими як Telegram, можна створювати доступні, масштабовані й гнучкі сервіси, здатні аналізувати вхідну інформацію та формувати змістовні текстові відповіді у форматі звичайного діалогу. Подібні рішення особливо цінні у контексті обмеженого доступу до медичних фахівців, а також у випадках, коли користувачам потрібна швидка й орієнтовна відповідь щодо стану здоров’я.

Ця дипломна робота присвячена розробці такого інструменту — медичного чат-бота, який здатен вести діалог із користувачем, приймати скарги у довільній формі, аналізувати їх за допомогою моделі GPT-4 Turbo та надавати загальні рекомендації. Реалізований програмний продукт поєднує можливості штучного інтелекту, зручного інтерфейсу та базових засобів захисту інформації, що дозволяє ефективно поєднати інженерний і гуманітарний аспекти теми.

В рамках розробки було вирішено низку прикладних завдань: від вибору відповідних технологій, проектування архітектури бота, реалізації логіки діалогу до формування звітів, збереження історії звернень та впровадження базових заходів безпеки даних. Окрему увагу приділено забезпеченню простоти взаємодії з ботом для кінцевого користувача, а також питанням обмеження функціоналу відповідно до етичних і правових норм.

Актуальність теми

Упродовж останніх років медична галузь активно впроваджує цифрові

рішення для підвищення ефективності обслуговування пацієнтів. Серед таких рішень особливе місце займають чат-боти — автоматизовані системи, що імітують спілкування з лікарем і здатні надавати первинні рекомендації на основі вхідних симптомів. З огляду на обмежений доступ до лікарів у періоди пікових навантажень на систему охорони здоров'я, такі інструменти стають практично необхідними.[2]

Чат-боти можуть допомогти користувачам зорієнтуватися у власному стані, дати загальні поради чи попередити про потребу звернутися до фахівця. Це дозволяє оперативно реагувати на проблеми зі здоров'ям і знижує навантаження на первинну ланку медицини. В умовах воєнного стану, нестачі кадрів або неможливості відвідати лікаря особисто, такі рішення набувають ще більшої значущості.[1]

Поєднання штучного інтелекту, мовних моделей і доступності месенджерів, таких як Telegram, дає змогу створити зручний і ефективний канал для медичних консультацій. Таким чином, розробка чат-бота медичного призначення є важливою як з точки зору розвитку IT-рішень у медицині, так і з погляду суспільної користі.[2]

Метою роботи є покращення доступу користувачів до базової медичної інформації шляхом впровадження чат-бота в Telegram, який поєднує інтеграцію GPT-4 Turbo, локальну базу даних та сценарну логіку (FSM) з метою зменшення навантаження на медичних працівників і підвищення обізнаності населення щодо власного самопочуття.[3]

У процесі реалізації було поставлено такі основні завдання:

1. ознайомитися з сучасними технологіями розробки чат-ботів і визначити найбільш придатні для медичної тематики;
2. обрати інструменти програмування, що забезпечують швидкість, гнучкість і масштабованість рішення;
3. розробити архітектуру чат-бота з урахуванням особливостей взаємодії з користувачем у сфері охорони здоров'я;

4. реалізувати функціонал бота, що дозволяє визначати симптоми, надавати базові рекомендації, формувати історію звернень і нагадувати про прийом ліків;
5. забезпечити збереження діалогів та конфіденційність користувацьких даних;
6. протестувати чат-бота в реальних умовах і провести оцінку його роботи.

Реалізація зазначених завдань дозволила сформувати завершений проєкт, що відповідає сучасним вимогам до цифрових рішень у медичній галузі.

У дипломній роботі було використано аналітичний підхід для вивчення сучасного стану розробки медичних чат-ботів, аналізу існуючих рішень на вітчизняному та міжнародному ринках, а також оцінки їх функціональних можливостей. Особливу увагу приділено пошуку застосунків, що використовують великі мовні моделі, зокрема GPT, для імітації консультацій у сфері охорони здоров'я. Проведений огляд показав, що на момент вибору теми подібні рішення мають обмежений функціонал, а в українському сегменті відсутні повноцінні реалізації, які б поєднували якісний діалог, аналіз симптомів та підтримку історії звернень.

Методи дослідження

Для технічної реалізації було використано мову програмування Python, бібліотеки aiogram, openai, sqlite3, а також середовище розробки Visual Studio Code. Обґрунтовано вибір архітектури на основі машини станів (FSM) для реалізації логіки опитування користувача. У роботі впроваджено взаємодію з мовною моделлю GPT-4 Turbo через API, реалізовано механізми обробки повідомлень, збереження результатів у базі даних, виведення відповідей та формування звітів у форматі PDF. Працездатність розробленого рішення перевірено шляхом тестування в умовах, наближених до реального використання.

Джерела дослідження

Під час виконання дипломної роботи було опрацьовано ряд джерел, що

охоплюють як медичну тематику, так і питання програмної реалізації. Для формування змістовної частини чат-бота використовувалися матеріали з офіційного сайту MedlinePlus, який надає структуровану та перевірену інформацію щодо симптомів, станів здоров’я та базових рекомендацій для пацієнтів.

Також було враховано нормативні документи Міністерства охорони здоров’я України, що регламентують розвиток електронної медицини, використання телемедичних сервісів і питання захисту персональних даних у сфері охорони здоров’я.

У технічній частині дослідження використовувалися офіційні документації до бібліотек aiogram, openai, sqlite3, а також технічні матеріали Telegram, які описують створення та підтримку ботів, було вивчено приклади реалізації схожих чат-ботів, які є у відкритому доступі, що дозволило врахувати сучасні підходи до побудови діалогових рішень із використанням штучного інтелекту.

Отримані результати та практичне їх застосування

У результаті дипломної роботи було створено повнофункціонального Telegram-бота для попередніх медичних консультацій із використанням мовної моделі GPT-4 Turbo. Розроблена система дозволяє користувачам анонімно вводити симптоми у зручному форматі, отримувати змістовні, логічно структуровані й емпатійні відповіді, а також формувати історію звернень, яка зберігається у локальній базі даних. Крім того, реалізовано функціонал експорту консультацій у форматі PDF, що може бути корисним для подальшого звернення до лікаря. Додатково вбудовано механізм створення персональних нагадувань про прийом ліків, що підвищує прикладну цінність розробки.

Практичне застосування такого рішення охоплює кілька сфер. У цифровій медицині бот може виступати як інструмент первинного інформування пацієнтів у телемедичних платформах або додатках електронної охорони здоров’я. У страхових сервісах — допоміжний засіб для первинної оцінки стану клієнта перед прийняттям рішення про надання послуги або оформлення страхового

випадку. Як соціальний інструмент — бот здатен підвищувати медичну грамотність населення, допомагати в орієнтації під час стресових ситуацій або за умов відсутності доступу до лікаря. Гнучкість архітектури й універсальність обраних технологій дозволяють масштабувати систему, адаптувати її під інші мови, розширювати функціонал і інтегрувати з зовнішніми сервісами.

РОЗДІЛ 1
ОГЛЯД ІСНУЮЧИХ МЕТОДІВ РОЗРОБКИ МЕДИЧНИХ ЧАТ-БОТІВ

1.1 Визначення основних функцій та цілей системи

Інформаційна система, що розробляється в межах цієї роботи, створена щоб забезпечити користувачам зручний інструмент для отримання попередньої медичної консультації. Telegram-бот функціонуватиме як асистент, здатний фіксувати скарги, формувати відповідь на їх основі за допомогою штучного інтелекту, а також зберігати історію звернень без необхідності участі медичного працівника. Система не виконує функцій діагностики, проте допомагає користувачеві зорієнтуватися у своєму самопочутті, прийняти обґрунтоване рішення щодо звернення до лікаря та отримати базову інформацію про симптоми. [5]

До основних функцій системи належить підтримка діалогу в межах сценарної логіки з покроковим опитуванням, генерація відповідей за допомогою великої мовної моделі, збереження результатів у локальній базі даних, формування консультацій, а також реалізація нагадувань про прийом ліків. Передбачено автоматичну обробку критичних симптомів, за яких бот рекомендує негайно звернутися по медичну допомогу. Інтерфейс побудовано з використанням Telegram-клавіатур для забезпечення інтуїтивної взаємодії.[2]

Реалізація зазначених можливостей має відповідати критеріям надійності, простоти у використанні, швидкодії та безпеки збереження інформації. Також враховано етичні аспекти: система повідомляє про обмеження щодо наданої інформації, не запитує персональні дані й не замінює лікаря. З практичної точки зору, чат-бот може бути корисним для індивідуального користування, у межах внутрішніх сервісів страхової компанії або як навчальний приклад цифрового інструменту в медичній сфері.[4]

1.2 Обґрунтування прийнятих рішень реалізації програмного забезпечення

Проведений аналіз показав, що месенджер Telegram є найкращим середовищем для реалізації чат-бота, адже він має відкритий API, зручний інтерфейс, велику аудиторію в Україні та дозволяє швидко налаштувати взаємодію з користувачем.

Для розробки було обрано мову програмування Python — просту, гнучку та дуже популярну. Вона підтримує великий набір готових бібліотек для створення ботів, роботи з API, базами даних та формування PDF-документів, що суттєво полегшує процес розробки.

Використані бібліотеки:

python-telegram-bot — для реалізації логіки Telegram-бота та побудови сценаріїв взаємодії;

sqlite3 — для локального збереження історії звернень та даних про ліки;

APScheduler — для реалізації функції нагадування;

ReportLab — для створення PDF-файлів з історією консультацій;

dotenv — для безпечного зберігання ключів доступу;

openai — для інтеграції з GPT-4 Turbo.

Для обробки запитів користувача та формування відповідей було обґрунтовано використання GPT-4 Turbo від OpenAI, оскільки ця модель забезпечує генерацію текстів із високою точністю, емпатійністю та медичною релевантністю (зокрема при використанні медичних джерел, таких як MedlinePlus).[2]

Для побудови логіки діалогу обрано підхід на основі Finite-State Machine (FSM), що дозволяє чітко визначити сценарії переходів між станами користувача, організувати багатокрокові діалоги та враховувати попередній контекст.

Також було прийнято рішення реалізувати функції локального збереження історії запитів, надання консультацій у вигляді повідомлень, формування PDF-звітів, обробки симптомів, створення нагадувань та контролю за безпекою обробки даних користувачів.

1.3 Практичне застосування програмного забезпечення

Чат-бот може мати практичне застосування у сфері інформатизації медичних послуг як інструмент попереднього консультування, зниження навантаження на персонал та підвищення доступності базової медичної інформації для населення. Розроблений Telegram-бот може використовуватись як внутрішній інструмент страхової компанії, що допомагає клієнтам орієнтуватися у симптомах, отримати поради щодо подальших дій, а також підтримує функції нагадування про прийом ліків і ведення історії звернень.

Разом з тим, навіть у разі відсутності негайного впровадження, розроблене рішення повністю розкриває володіння студенткою загальними та спеціальними (фаховими) навичками, а також програмними результатами навчання згідно з ОПП “Комп’ютерні технології в біології та медицині”.

Робота виконана за замовленням ТОВ “СК “КРАЇНА” згідно договору про співпрацю № 57/25.

Апробація результатів Не заплановано.

Публікації. Не заплановано.

1.4 Загальна характеристика медичних чат-ботів

Медичні чат-боти є прикладом використання цифрових технологій для підтримки охорони здоров’я шляхом надання базової інформації, консультацій, рекомендацій та інших сервісів користувачам у зручному текстовому або голосовому форматі. У загальному вигляді чат-бот — це програмне забезпечення, що імітує людське спілкування, використовуючи наперед задані

сценарії або технології обробки природної мови (Natural Language Processing, NLP).[4]

Медичні боти відносяться до категорії вузькоспеціалізованих інформаційних систем. Їх основна мета — надання попередньої допомоги або орієнтаційної інформації користувачу за умови, що цей бот не замінює лікаря, а лише допомагає зорієнтуватися в симптомах, надає загальні поради та нагадує про певні дії (наприклад, прийом ліків). Роль таких ботів особливо важлива в умовах, коли доступ до лікаря ускладнений — наприклад, під час війни, пандемії або в сільській місцевості.[2]

Серед основних функцій медичних чат-ботів варто відзначити консультацію щодо симптомів, попередню оцінку ймовірного стану здоров'я та надання рекомендацій щодо подальших дій — наприклад, звернутися до лікаря або залишитися вдома. Такі чат-боти також можуть збирати важливу інформацію для лікаря, нагадувати про прийом ліків чи майбутні візити, допомагати вести медичний щоденник і навіть підтримувати психоемоційний стан користувача.

За способом реалізації медичні чат-боти поділяються на кілька типів:

Сценарні (rule-based) — використовують заздалегідь задані діалоги, де кожна відповідь веде до наступного кроку. Такі боти прості у розробці, але обмежені в гнучкості.[4]

Інтелектуальні (AI-based) — базуються на технологіях машинного навчання, здатні обробляти складні запити, адаптуватися до стилю спілкування користувача, використовують NLP.[2]

Гібридні — поєднують сценарії з можливістю вільного введення тексту, часто використовують GPT-моделі.[5]

Для побудови якісного медичного бота важливо враховувати етичні та юридичні аспекти, пов'язані з медичною інформацією. У більшості країн така інформація вважається конфіденційною і не може зберігатися або оброблятися без згоди користувача. Крім того, будь-які рекомендації бота мають супроводжуватись попередженням, що інформація не є діагнозом.

Варто зазначити, що успішний чат-бот у сфері охорони здоров'я має забезпечувати простий і зрозумілий інтерфейс, підтримувати логіку в діалозі та

демонструвати емпатію. Крім того, він повинен формувати рекомендації виключно на основі перевірених медичних джерел і водночас не створювати хибного відчуття безпеки чи заохочення до самолікування.

На сьогодні технології штучного інтелекту, зокрема великі мовні моделі (LLM) на зразок GPT-4, дозволяють реалізовувати високоякісних ботів, здатних на складну мовну взаємодію. Саме такий підхід обрано у межах даної практичної роботи — створення Telegram-бота, який за допомогою GPT може вести живий діалог, ставити уточнюючі запитання, зберігати історію звернень, генерувати нагадування і формувати відповіді з урахуванням медичного контексту.

1.5 Огляд функціональних аналогів та прикладних рішень

У сучасних умовах активного впровадження цифрових технологій в охорону здоров’я особливого значення набувають інструменти, які здатні забезпечити швидку, доступну та орієнтовану на користувача взаємодію з медичною системою. Одним із таких інструментів є медичні чат-боти — програмні агенти, що можуть автономно проводити комунікацію з пацієнтами, збирати попередні симптоми, пропонувати базові поради щодо дій та сприяти ефективнішому маршрутуванню до фахівців у разі потреби [18].

У 2025 році такі системи дедалі частіше інтегруються у цифрову інфраструктуру як державних, так і приватних медичних платформ. Вони застосовуються не лише для консультацій, а й для управління записами, нагадувань, інформування про профілактичні програми та моніторингу хронічних станів [1]. Більшість сучасних ботів мають адаптивний діалоговий інтерфейс і підтримують елементи персоналізації, що сприяє зростанню довіри з боку користувачів.

Важливою віхою розвитку таких рішень стала поява великих мовних моделей (LLM), зокрема GPT-4 Turbo. Їх інтеграція дозволила суттєво підвищити якість спілкування між системою та користувачем. Завдяки генерації розгорнутих, логічно зв’язаних відповідей, розумінню контексту та емпатійному

стилю мовлення, ці технології створили новий стандарт взаємодії у сфері цифрової медицини [5].

Цей пункт присвячено аналізу найбільш відомих і технологічно значущих медичних чат-ботів, які вже функціонують на ринку. Розглядаються як класичні rule-based рішення, що працюють за фіксованими сценаріями, так і AI-орієнтовані боти, що генерують відповіді на основі нейромереж. Особлива увага приділена їх функціональній структурі, обробці симптомів, формуванню порад, можливості збереження історії звернень та відповідності чинним вимогам з етики та конфіденційності [18].

Одним із найпоширеніших прикладів медичного цифрового сервісу, що забезпечує самостійну попередню оцінку стану здоров'я користувача, є застосунок Ada Health. Це мобільний інструмент, що використовує алгоритми штучного інтелекту для аналізу симптомів на основі відповідей користувача. Після введення скарг система формує адаптивну серію уточнювальних запитань, структура яких близька до медичного або психосоматичного анамнезу. Після завершення опитування користувач отримує перелік можливих причин свого стану з описом і рекомендаціями щодо подальших дій [19].

Застосунок був розроблений з урахуванням доказової медицини та медичних протоколів, що дозволяє йому забезпечувати високу точність прогнозування в межах своєї функціональної моделі. Платформа підтримує кілька мов інтерфейсу, зокрема англійську, німецьку, французьку, іспанську та португальську, що робить її доступною в багатьох регіонах. Згідно з офіційними даними, Ada Health використовується у понад 140 країнах світу, і має мільйони активних користувачів [19].

Користувацький інтерфейс орієнтований на інтуїтивну взаємодію та не потребує попередньої медичної підготовки. Після завершення сесії користувач має змогу зберегти результат у вигляді звіту, який можна надати лікарю, а також продовжити спостереження за симптомами через вбудовані функції моніторингу. Застосунок не ставить діагнозів, але інформує про ймовірні порушення та рекомендує звернення до спеціаліста у випадку виявлення

тривожних ознак. Також важливо, що система не збирає персональні медичні дані користувачів без їхньої згоди, дотримуючись міжнародних стандартів у сфері конфіденційності [19].

Ada Health позиціонується не як альтернатива лікарю, а як інструмент попереднього орієнтування. Такий формат дозволяє не лише зменшити навантаження на первинну медичну ланку, а й підвищити обізнаність користувача щодо свого стану здоров'я. На відміну від класичних rule-based систем, які працюють за фіксованими шаблонами, Ada реалізує більш гнучкий механізм адаптивного опитування, однак при цьому не використовує великі мовні моделі нового покоління, як-от GPT. Це робить її стабільною та контрольованою з точки зору результатів, але обмежує у варіативності відповідей порівняно з генеративними ботами [4].

Незважаючи на це, Ada Health зберігає позиції одного з найнадійніших комерційних рішень у сфері цифрової медицини. Її використання продовжує зростати, а сама система постійно оновлюється відповідно до нових клінічних протоколів і рекомендацій.

Babylon Health - це британська цифрова платформа у сфері охорони здоров'я, яка поєднує функціонал чат-бота для первинної оцінки симптомів, відеоконсультації з лікарями та електронну медичну документацію. Система була створена для забезпечення пацієнтам швидкого доступу до базової медичної допомоги через мобільний застосунок. Технологічна основа Babylon - це поєднання фіксованих медичних алгоритмів і адаптивних аналітичних моделей, що дозволяють автоматично оцінювати стан користувача на основі його скарг і відповідей на серію запитань[20].

Платформа впровадила багаторівневу модель взаємодії: чат-бот виступає першим етапом скринінгу, після чого користувач може бути перенаправлений до лікаря для онлайн-консультації.[20] Усі дані, отримані під час діалогу з ботом, автоматично вносяться до електронної картки пацієнта, яка доступна лікарю під час сесії. Це не лише пришвидшує процес, а й знижує навантаження на медичний персонал у контексті первинного збору інформації.

Babylon Health інтегрована з інфраструктурою Національної служби охорони здоров’я Великої Британії (NHS), що дозволило застосовувати платформу у державних медичних закладах у межах пілотних проєктів. Наприклад, у Лондоні Babylon Health забезпечувала повноцінне цифрове GP-обслуговування тисяч пацієнтів. Водночас проєкт отримав змішані відгуки: користувачі відзначали зручність і швидкість сервісу, а деякі лікарські асоціації ставили під сумнів точність алгоритмів та їхню відповідність реальним клінічним сценаріям[23].

Платформа підтримує понад 15 мов інтерфейсу та працює в низці країн, зокрема у Великій Британії, Канаді, США, Руанді, Південній Кореї. Babylon активно співпрацювала з урядами та страховими компаніями в межах ініціатив зі спрощення доступу до базової медицини. Однак у 2023 році компанія припинила частину своїх публічних контрактів у Великій Британії через фінансові труднощі, що спричинило дискусії про сталий розвиток цифрових медичних сервісів, які залежать від державного фінансування[21].

Попри виклики, Babylon Health залишається одним із найбільш розвинутих прикладів гібридної цифрової системи, яка поєднує чат-бот, елементи штучного інтелекту, базу знань, персональний кабінет пацієнта та прямий зв’язок з лікарем.[20] Її досвід демонструє як потенціал, так і обмеження впровадження автоматизованих рішень у сферу охорони здоров’я, зокрема щодо етики, безпеки даних і клінічної точності[22].

Buoy Health - це американська цифрова платформа у сфері охорони здоров’я, яка використовує штучний інтелект для надання користувачам персоналізованої підтримки при оцінці симптомів. Основна функція сервісу - інтерактивний чат, у якому система ставить серію уточнювальних запитань, щоб краще зрозуміти стан здоров’я людини. На основі відповідей користувача Buoy Health аналізує отриману інформацію та пропонує можливі причини симптомів, а також дає рекомендації щодо подальших дій: чи потрібно звернутися до лікаря, які профілактичні заходи варто вжити або які дії можна виконати самостійно[24].

Важливою особливістю Buoy Health є доступ до великої бази науково обґрунтованих матеріалів, підготовлених клініцистами. Це дозволяє користувачам отримувати не лише автоматизовані поради, а й перевірену інформацію щодо різних медичних станів, профілактики та лікування. Такий підхід допомагає підвищити рівень медичної обізнаності населення та сприяє прийняттю більш зважених рішень щодо власного здоров'я [25].

Платформа орієнтована на широкий загал користувачів і може використовуватись як перший етап у процесі звернення за медичною допомогою. Buoy Health активно співпрацює з медичними закладами та страховими компаніями у США, інтегруючи свої рішення в екосистему цифрової медицини. Це дозволяє не лише зменшити навантаження на лікарів, а й забезпечити швидкий доступ до якісної медичної інформації для мільйонів людей [26].

Завдяки поєднанню сучасних технологій штучного інтелекту та експертного медичного досвіду, Buoy Health є прикладом ефективного цифрового інструменту для первинної самооцінки стану здоров'я та підвищення медичної грамотності серед населення

HealthGPT - це сучасний мобільний застосунок, який використовує можливості штучного інтелекту для надання користувачам якісної медичної інформації та підтримки у питаннях здоров'я. Основна функція додатку - це чат-інтерфейс, де користувач може у зручній формі описати свої симптоми, отримати попередню оцінку можливих причин нездужання та дізнатися, які кроки варто зробити далі. HealthGPT аналізує введені дані за допомогою алгоритмів штучного інтелекту, що дозволяє швидко й персоналізовано відповідати на запити користувачів .

Застосунок не лише надає інформацію про симптоми та потенційні захворювання, а й пропонує рекомендації щодо профілактики: користувачі можуть отримати поради стосовно здорового способу життя, харчування, фізичної активності та своєчасного звернення до лікаря. HealthGPT також

допомагає зорієнтуватися в інформаційному просторі, надаючи доступ до науково обґрунтованих матеріалів і пояснень щодо різних станів здоров'я[27].

Особливістю HealthGPT є його дружній, ненав'язливий стиль спілкування, що створює атмосферу довіри та підтримки. Інтерфейс додатку інтуїтивно зрозумілий, а відповіді - адаптовані під індивідуальні потреби користувача. Це робить HealthGPT корисним як для людей, які шукають швидку інформацію про своє самопочуття, так і для тих, хто прагне підвищити власну медичну грамотність[27].

Додаток може бути корисним у повсякденному житті для моніторингу стану здоров'я, прийняття рішень щодо звернення до лікаря, а також для отримання порад із профілактики захворювань. HealthGPT постійно оновлюється, впроваджуючи нові функції та розширюючи базу знань, що дозволяє йому залишатися актуальним цифровим помічником для користувачів у всьому світі [27].

Medico-Assistance-OpenAI-ChatBot-Using-Python - це відкритий проєкт, розміщений на GitHub, який демонструє створення медичного чат-бота із застосуванням OpenAI GPT-3 та мови програмування Python. Бот призначений для надання користувачам загальної медичної інформації та підтримки у форматі діалогу, водночас чітко підкреслюється, що його відповіді не можуть замінити професійну медичну консультацію[13]. Для обробки природної мови у проєкті використовується бібліотека Textbase, що дозволяє зберігати контекст розмови та забезпечувати послідовність відповідей. Користувачі можуть запускати бота локально, попередньо налаштувавши Python та необхідні залежності, і взаємодіяти з ним через текстовий інтерфейс. Проєкт має відкриту ліцензію та заохочує до подальшого розвитку спільнотою. Основна мета - продемонструвати можливості інтеграції сучасних мовних моделей у сферу цифрової медицини для підвищення доступності базової медичної інформації, не порушуючи при цьому принципів безпеки та відповідальності[13] .

Аналіз існуючих функціональних аналогів медичних чат-ботів що ілюструє табл. 1.1 демонструє різноманітність підходів до реалізації таких систем. Від простих rule-based рішень до складних AI-застосунків, кожен з розглянутих ботів має свої переваги та обмеження. Важливо зазначити, що, незважаючи на значний прогрес у цій галузі, жодне з існуючих рішень не поєднує всі функції, які можуть бути реалізовані в межах сучасних технологічних можливостей.

Таблиця 1.1

Порівняння функціональних аналогів

Чат-бот	Основний функціонал	Формат взаємодії	Інтеграція	Мови	Обмеження
Babylon Health	AI-симптом-чекер, відеоконсультації, електронна медична картка	Чат-бот, перенаправлення до лікаря	Інтеграція з NHS, медичними закладами	15+	Клінічна точність не завжди гарантована, залежність від держ. фінансування
Buoy Health	AI-симптом-чекер, персоналізовані рекомендації, статті від лікарів	Інтерактивний Q&A	Співпраця з медичними установами та страховими	Англійська	Не замінює професійну діагностику, обмежено до перевірки симптомів
Health GPT	AI-чат-бот для інформації про симптоми, профілактичні поради, інтерактивний чат	Дружній чат-інтерфейс	Мобільний додаток із освітнім контентом	Англійська	Загальна інформація, не є медичною порадою
Medico-Assistance-OpenAIChatBot-Using-Python	Відкритий AI-чат-бот на GPT-3 для медичної інформації та підтримки	Текстовий чат із збереженням контексту	Самостійний проєкт на GitHub	Англійська	Лише для ознайомлення? не для клінічного використання

Продовження табл. 1.1

Чат-бот	Основний функціонал	Формат взаємодії	Інтеграція	Мови	Обмеження
Ada Health	AI-симптом-чекер, тріаж, детальний звіт, навігація пацієнта, клінічно перевірені алгоритми	Динамічний чат-інтерфейс	Інтеграція з медичними системами, страховими, урядами	7+	Не замінює лікаря, обмеження щодо складних/ невідкладних станів, лише інформаційна підтримка

1.6 Нормативно-етичні аспекти

Використання медичних чат-ботів на основі штучного інтелекту супроводжується низкою нормативних та етичних викликів, що мають важливе значення для безпеки користувачів, захисту персональних даних і формування довіри до цифрових медичних сервісів. Одним із ключових аспектів є захист персональних даних і конфіденційність. Медичні чат-боти працюють із чутливою інформацією про здоров'я користувачів, тому повинні відповідати чинному законодавству щодо захисту даних (наприклад, GDPR у ЄС, Закон України «Про захист персональних даних»). Необхідно забезпечити прозорість щодо збору, зберігання та використання даних, а також отримувати явну згоду користувача на їх обробку.

Етичні ризики пов'язані з можливістю виникнення алгоритмічної упередженості. Великі мовні моделі можуть відтворювати або підсилювати наявні упередження у вихідних даних, що може призвести до дискримінаційних або некоректних рекомендацій. Розробники повинні регулярно тестувати системи на предмет упередженості, впроваджувати механізми її мінімізації і дотримуватися принципів недискримінації[28].

Важливо також чітко визначати межі відповідальності: чат-бот не може замінити консультацію з лікарем, а його поради мають носити інформаційний характер. Кінцеві медичні рішення повинні ухвалюватися лише кваліфікованими фахівцями. Усі цифрові сервіси повинні містити попередження

про те, що інформація від чат-бота не є діагнозом і не може бути підставою для самолікування[2].

Окрему увагу слід приділяти питанням кібербезпеки. Зберігання та передача медичних даних створює ризики для їхньої цілісності та захищеності. Тому необхідно впроваджувати сучасні засоби захисту, шифрування, багаторівневу автентифікацію та проводити регулярний аудит безпеки[29].

Дотримання міжнародних стандартів і професійних кодексів (рекомендації ВООЗ, FDA, Європейської комісії тощо) є обов'язковою умовою для розробників і провайдерів медичних чат-ботів. Це забезпечує прозорість, підзвітність і справедливість у використанні штучного інтелекту в медицині[30].

Таким чином, нормативно-етичні аспекти є фундаментом для впровадження медичних чат-ботів, визначаючи як юридичну відповідальність розробників, так і рівень довіри користувачів до цифрових сервісів у сфері охорони здоров'я[30].

Висновки до розділу 1

У цьому розділі було здійснено комплексний аналіз сучасних підходів до створення медичних чат-ботів, їх функціональних характеристик, типології, а також прикладів практичного використання подібних систем у сфері охорони здоров'я. Розглянуто як сценарні (rule-based), так і інтелектуальні рішення на основі великих мовних моделей, зокрема GPT, що дало змогу виявити їх переваги й недоліки в контексті застосування для попереднього медичного консультування.

Особливу увагу приділено нормативно-етичним і безпековим аспектам, що є визначальними для розробки IT-рішень, пов'язаних з обробкою чутливої медичної інформації. Аналіз існуючих аналогів показав, що попри наявність різноманітних інструментів, жодне з них не поєднує у повному обсязі необхідні функції — зокрема, гнучку логіку діалогу, обробку симптомів, збереження історії звернень і локалізований інтерфейс українською мовою.

На підставі опрацьованих матеріалів обґрунтовано доцільність

використання мовної моделі GPT-4 Turbo у поєднанні зі сценарною логікою (FSM), локальним зберіганням даних і дотриманням принципів конфіденційності. Здобуті результати стали основою для архітектурного проєктування та подальшої реалізації програмного продукту, описаної в наступних розділах роботи.

РОЗДІЛ 2
ЗАСОБИ РЕАЛІЗАЦІЇ ЧАТ-БОТУ

Вибір засобів реалізації є одним із ключових етапів створення сучасного медичного чат-боту. Від правильного підбору мови програмування, бібліотек, фреймворків та сервісів залежить не лише функціональність системи, а й її надійність, масштабованість, зручність для користувача та відповідність вимогам безпеки.

У цьому розділі розглянуто програмні інструменти, що були використані при розробці чат-боту: мову програмування Python, бібліотеку python-telegram-bot для інтеграції з Telegram, OpenAI GPT-4 Turbo для генерації відповідей, SQLite3 для зберігання даних, APScheduler для реалізації нагадувань, а також ReportLab для формування PDF-звітів.[6]

Обґрунтовано вибір архітектури програмного забезпечення, описано особливості інтеграції із зовнішніми сервісами та організації локального зберігання інформації.

2.1. Вибір мови програмування та середовища розробки

Для розробки медичного чат-боту було обрано мову програмування Python. Такий вибір зумовлений низкою факторів: Python є однією з найпопулярніших мов для створення сучасних чат-ботів та систем штучного інтелекту, має простий синтаксис, велику кількість бібліотек для роботи з API, обробки тексту, машинного навчання та інтеграції з різними сервісами.

Python забезпечує швидке прототипування, гнучкість у побудові архітектури, а також має широке ком’юніті, що дає змогу швидко знаходити рішення типових проблем під час розробки. Особливо важливою є підтримка бібліотек для роботи з Telegram (python-telegram-bot), інтеграції з OpenAI GPT-4 Turbo, організації локальної бази даних (sqlite3), планування завдань (APScheduler) та генерації PDF-звітів (ReportLab).[7]

Середовище розробки обрано з урахуванням зручності роботи з Python-проєктами, підтримки віртуальних середовищ, інтеграції з системами контролю версій та можливості налагодження коду. Для цього було використано Visual Studio Code та PyCharm. Обидва середовища мають потужні засоби для автодоповнення коду, інтеграцію з Git, підтримку дебагінгу, а також численні розширення для роботи з Python.

Завдяки такому вибору мови програмування та середовища розробки вдалося забезпечити високу швидкість створення MVP (minimum viable product), легкість масштабування проєкту, а також можливість інтеграції з сучасними AI-сервісами та зовнішніми API.

2.2. Використані бібліотеки та фреймворки

Для реалізації функціоналу медичного чат-боту було використано низку сучасних бібліотек та фреймворків, які забезпечують інтеграцію з месенджером Telegram, роботу з моделями штучного інтелекту, організацію локальної бази даних, планування подій і генерацію звітів у форматі PDF.

Основні інструменти:

- python-telegram-bot - популярна бібліотека для створення Telegram-ботів на Python, що дозволяє легко організувати обробку повідомлень, команд, клавіатур та діалогових сценаріїв.[8]
- OpenAI GPT-4 Turbo - сучасний API для інтеграції великих мовних моделей у програмні продукти, використовується для генерації медичних рекомендацій і відповідей на запити користувача. [8]
- SQLite3 - вбудована база даних, яка забезпечує зберігання історії консультацій, даних про користувачів, нагадувань та іншої інформації без необхідності налаштування окремого серверу. [8]
- APScheduler - бібліотека для планування та виконання періодичних завдань, зокрема для реалізації нагадувань про прийом ліків. [8]

- ReportLab - інструмент для генерації PDF-документів, який використовується для створення звітів з історією консультацій та іншої інформації, що може бути експортована користувачем. [8]

Використання зазначених бібліотек дозволило забезпечити гнучкість, масштабованість та надійність програмного продукту, а також спростити розробку завдяки великій кількості документації та підтримці спільноти. Для інтеграції з моделями штучного інтелекту та обробки природної мови застосовувалися рекомендації сучасних оглядових джерел, присвячених медичним чат-ботам.

2.3. Опис архітектури чат-боту

Архітектура медичного чат-боту побудована за модульним принципом, що дозволяє легко масштабувати систему та додавати нові функції без суттєвих змін у коді. Основні компоненти архітектури:

- Інтерфейс користувача (Telegram Bot API): Забезпечує прийом і відправлення повідомлень, роботу з клавіатурами та командами.[9]
- Менеджер діалогів (FSM): Відповідає за логіку переходу між станами діалогу, обробку сценаріїв, збереження контексту розмови.
- Модуль генерації відповідей (OpenAI GPT-4 Turbo): Створює текстові відповіді на основі введених користувачем симптомів або питань, забезпечуючи персоналізований підхід.
- База даних (SQLite3): Зберігає історію консультацій, дані про ліки та нагадування, забезпечує швидкий доступ до інформації. [8]
- Модуль нагадувань (APScheduler): Планує та надсилає нагадування про прийом ліків у заданий час.
- Модуль експорту (ReportLab): Генерує PDF-звіти з історією консультацій для подальшого використання або передачі лікарю. [8]

Всі компоненти взаємодіють між собою через чітко визначені інтерфейси, що підвищує надійність та підтримуваність системи. Такий підхід відповідає сучасним тенденціям розробки медичних чат-ботів і дозволяє легко інтегрувати нові сервіси або зовнішні API.

2.4. Інтеграція з Telegram API

Інтеграція чат-боту з месенджером Telegram здійснюється за допомогою офіційного Telegram Bot API та бібліотеки python-telegram-bot.[9] Для створення бота використовується @BotFather, який генерує унікальний токен доступу. Далі цей токен застосовується у програмному коді для ініціалізації з’єднання з Telegram-серверами.[10]

Бібліотека python-telegram-bot дозволяє організувати обробку різних типів повідомлень (текст, команди, кнопки), підтримує асинхронну роботу, FSM для діалогів та зручну роботу з клавіатурами. [8] Використання webhook або polling забезпечує отримання повідомлень у реальному часі, що критично для своєчасної реакції на запити користувачів.[11]

Завдяки гнучкості Telegram API можна реалізувати кастомні сценарії, багатомовність, авторизацію користувачів і навіть інтеграцію з іншими медичними сервісами.

Висновок до розділу 2

У цьому розділі було обґрунтовано вибір мови програмування Python та сучасних бібліотек для створення медичного чат-боту, зокрема python-telegram-bot для інтеграції з Telegram, OpenAI GPT-4 Turbo для генерації відповідей, SQLite3 для організації локальної бази даних, APScheduler для реалізації системи нагадувань та ReportLab для формування PDF-звітів.

Архітектура програмного забезпечення побудована за модульним принципом, що забезпечує гнучкість, масштабованість і простоту підтримкисистеми.

Інтеграція із зовнішніми сервісами та використання сучасних інструментів дозволяє не лише автоматизувати рутинні процеси, а й підвищити якість медичних консультацій, забезпечити збереження й захист даних користувачів.

Обраний підхід відповідає сучасним тенденціям розробки медичних чат-ботів і створює надійну основу для реалізації функціоналу, описаного у наступних розділа

РОЗДІЛ 3
ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОБОТА ЧАТ-БОТУ

Розробка програмної частини медичного чат-боту передбачає створення структурованого, надійного та зручного для користувача програмного забезпечення, яке забезпечує інтеграцію з месенджером Telegram, генерацію відповідей за допомогою штучного інтелекту, збереження історії консультацій і реалізацію додаткових сервісів, таких як нагадування про прийом ліків та формування звітів.

У цьому розділі детально розглядаються структура програмного коду, основні функції та логіка роботи чат-боту, реалізація діалогових сценаріїв, інтерфейс користувача, а також особливості тестування та аналізу результатів роботи системи. Особливу увагу приділено питанням зручності для користувача, гнучкості сценаріїв взаємодії та забезпеченню стабільної роботи бота в реальних умовах.

Використання сучасних бібліотек і фреймворків дозволило забезпечити масштабованість, модульність і простоту підтримки програмного продукту. Програмна реалізація враховує вимоги до безпеки, конфіденційності та відповідності стандартам, що особливо важливо для медичних застосунків,.

3.1 Структура програмного коду

Ефективна організація програмного коду є ключовою складовою надійної та масштабованої роботи будь-якого програмного продукту, зокрема й медичного чат-бота. У цьому розділі розглядається структура коду, що реалізує основні функціональні можливості чат-бота, забезпечує взаємодію з користувачем, обробку запитів, збереження даних, інтеграцію з зовнішніми сервісами та виконання допоміжних задач.

Особливістю даного проєкту є те, що вся логіка роботи чат-бота реалізована в одному файлі main.py. Такий підхід дозволяє централізовано

керувати всіма процесами, спрощує тестування та внесення змін на початкових етапах розробки.[12] Код структуровано у вигляді окремих функцій, кожна з яких відповідає за певний аспект роботи системи: ініціалізацію, обробку команд, реалізацію діалогових сценаріїв, роботу з базою даних, інтеграцію з моделями штучного інтелекту, планування нагадувань та формування звітів. Далі детально описано, як організовано основні функції, їхню логіку, взаємозв'язки та роль у загальній архітектурі

Імпорт та налаштування середовища:

На початку коду імпортуються всі необхідні бібліотеки: стандартні модулі Python (os, logging, sqlite3, datetime), бібліотеки для роботи з Telegram API, інтеграції з OpenAI GPT, генерації PDF-звітів, планування подій (APScheduler),та модуль для роботи з файлами середовища (dotenv).

Цей блок також відповідає за завантаження секретних ключів і токенів з файлу .env для безпечної роботи з API[13] він формує основу для всієї подальшої роботи бота, забезпечуючи доступ до необхідних зовнішніх сервісів та інструментів.(рис.3.1)

```python
import os
import logging
import re
import sqlite3
from datetime import datetime
from dotenv import load_dotenv
from telegram import (
    Update,
    ReplyKeyboardMarkup,
    KeyboardButton,
)
from telegram.ext import (
    ApplicationBuilder,
    CommandHandler,
    MessageHandler,
    filters,
    ContextTypes,
    ConversationHandler,
)
from reportlab.pdfgen import canvas
from reportlab.lib.pagesizes import A4
import openai
from apscheduler.schedulers.background import BackgroundScheduler
import atexit
```

Рисунок 3.1 - Імпорт та налаштування середовища

Ініціалізація та робота з базою даних:

Для зберігання історії консультацій, інформації про користувачів, ліки та нагадування використовується вбудована база даних SQLite3. Під час запуску бота створюється підключення до бази, а також таблиці, якщо вони ще не існують. .(рис.3.2)

```
# База даних
conn = sqlite3.connect('medical_bot.db', check_same_thread=False)
cursor = conn.cursor()
cursor.execute('''CREATE TABLE IF NOT EXISTS history
                  (user_id INTEGER, timestamp TEXT, input TEXT, response TEXT)''')
cursor.execute('''CREATE TABLE IF NOT EXISTS medications
                  (user_id INTEGER, drug TEXT, times TEXT)''')
conn.commit()
```

Рисунок 3.2 - Ініціалізація та робота з базою даних

Завдяки такій організації всі дані користувачів зберігаються локально, що спрощує доступ до них, забезпечує швидкість роботи та конфіденційність.

Планувальник нагадувань (APScheduler):

Для реалізації функції нагадування про прийом ліків використовується бібліотека APScheduler, яка дозволяє створювати, зберігати та виконувати [14]

Планувальник працює у фоновому режимі, не блокуючи основний цикл бота, і може надсилати нагадування навіть при великій кількості користувачів. .(рис.3.3)

```
# --- Планувальник нагадувань ---
scheduler = BackgroundScheduler()
scheduler.start()
atexit.register(lambda: scheduler.shutdown())
```

Рисунок 3.3 – APScheduler(функція нагадування)

Визначення клавіатур та меню (рис. 3.4):

Для зручності користувача реалізовано кастомні клавіатури, які дозволяють швидко обирати потрібну дію, зменшують кількість помилок та підвищують швидкість взаємодії.

Меню дозволяє користувачу інтуїтивно обирати потрібний сценарій, що особливо важливо для людей, які не мають досвіду роботи з ботами.

```python
# --- Клавіатури ---
MAIN_MENU_KEYBOARD = ReplyKeyboardMarkup(
    [
        [KeyboardButton("🩺 Консультація"), KeyboardButton("💊 Мої ліки")],
        [KeyboardButton("📜 Історія"), KeyboardButton("📤 Експорт історії")],
        [KeyboardButton("🆘 Невідкладні стани")]
    ],
    resize_keyboard=True
)
```

Рисунок 3.4- Клавіатура та меню

Оголошення станів FSM для діалогів:

Усі сценарії діалогу реалізовано на основі finite-state machine (FSM), що дозволяє чітко розмежувати логіку для різних станів (наприклад, консультація, додавання ліків, екстрені стани). (рис 3.5)

Кожен стан відповідає певному етапу взаємодії з користувачем, а переходи між станами визначаються діями користувача.

```python
# Стани діалогу
(
    FREE_DIALOG, SYMPTOM_SELECTION,
    MEDICATION_MENU, MEDICATION_ADD, MEDICATION_DELETE, EMERGENCY_CHECK
) = range(6)
```

Рисунок 3.5 - Оголошення станів FSM для діалогів

Функції-обробники команд та сценаріїв:

Для кожної команди або сценарію діалогу визначено окрему асинхронну функцію-обробник.

Вони відповідають за прийом вхідних даних, перевірку правильності введення, формування відповіді та зміну стану діалогу.(рис 3.6)

```
# --- Обробники ---
async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text(
        "👋 Вітаю! Я ваш розумний медичний помічник. Ось мої можливості:",
        reply_markup=MAIN_MENU_KEYBOARD
    )
```

Рисунок 3.6 - Функції-обробники команд та сценаріїв

Діалогова логіка: ConversationHandler:

Для організації складних сценаріїв використовується ConversationHandler, який дозволяє керувати переходами між станами діалогу.

У цьому проєкті вся логіка медичного чат-боту реалізована в одному файлі main.py. Такий підхід дозволяє швидко розробляти, тестувати та підтримувати систему, що особливо зручно на етапі створення MVP або для навчальних цілей. Код структуровано у вигляді окремих функцій, кожна з яких відповідає за певний аспект роботи бота. Далі деталь

3.2. Практичне застосування бібліотек і інструментів у реалізації чат-бота

У процесі програмної реалізації медичного чат-боту було використано низку бібліотек і фреймворків, які забезпечили ефективну роботу всіх основних функцій системи. Кожна бібліотека інтегрована у проект для вирішення конкретних завдань, що дозволило досягти високої якості роботи бота, зручності для користувача та гнучкості подальшого розвитку.

Python-telegram-bot - ця бібліотека є основою для інтеграції бота з месенджером Telegram. У нашому проекті вона використовується для обробки всіх вхідних повідомлень, команд, організації меню та діалогових сценаріїв. За допомогою ConversationHandler реалізовано FSM (finite-state machine) для

консультацій та роботи з ліками, а MessageHandler забезпечує реагування на текстові запити користувача. Завдяки цій бібліотеці реалізовано головне меню, навігацію, обробку кнопок і забезпечено багаторівневу логіку діалогу з користувачем.[15]

OpenAI GPT-4 Turbo, дана модель штучного інтелекту використовується для генерації медичних відповідей на запити користувача. У проекті GPT-4 Turbo інтегровано для формування рекомендацій, аналізу симптомів, уточнення деталей та ведення персоналізованого діалогу. Модель дозволяє отримувати не шаблонні, а унікальні, змістовні й релевантні відповіді, що підвищує рівень довіри користувача до сервісу та якість консультацій.[16]

Вбудована база даних SQLite3 служить для зберігання історії консультацій, даних про користувачів, інформації про призначені ліки та налаштувань нагадувань. У нашому чат-боті всі звернення користувача, відповіді бота, а також розклад прийому ліків зберігаються у локальній базі, що забезпечує швидкий доступ до інформації, цілісність даних і можливість формування звітів.[15]

APScheduler використовується для організації системи нагадувань про прийом ліків. Планувальник дозволяє створювати персональні розклади для кожного користувача, автоматично надсилати повідомлення у визначений час і зберігати інформацію про майбутні нагадування. Це підвищує зручність сервісу та допомагає користувачам дотримуватися режиму лікування.[15]

Для формування звітів з історією консультацій реалізовано генерацію PDF-документів за допомогою ReportLab. У нашому проекті користувач може експортувати свою історію у вигляді структурованого PDF-файлу, який можна зберегти, роздрукувати або надати лікарю. Це розширює функціональні можливості бота і робить його більш корисним у реальному медичному процесі.[15]

Додаткові бібліотеки

dotenv використовується для безпечного зберігання токенів і ключів у файлі .env, що захищає конфіденційну інформацію.[16]

logging забезпечує ведення журналу подій, що спрощує діагностику та пошук помилок.

datetime використовується для роботи з датами й часом у функціях нагадувань та історії.

Завдяки інтеграції цих бібліотек вдалося реалізувати всі ключові функції чат-бота: від діалогів і консультацій до збереження даних, нагадувань і формування звітів. Такий підхід забезпечує надійність, розширюваність і зручність підтримки програмного продукту.

3.3. Опис елементів інтерфейсу та логіка їх обробки

Інтерфейс користувача є ключовим елементом будь-якого інтерактивного програмного забезпечення, особливо коли йдеться про медичний чат-бот, орієнтований на широку аудиторію, включно з користувачами без спеціальної технічної підготовки. У даному проєкті інтерфейс реалізовано у вигляді інтеграції з месенджером Telegram за допомогою бібліотеки python-telegram-bot. Взаємодія з користувачем здійснюється у текстовому форматі із використанням вбудованих кнопкових меню, які значно підвищують зручність навігації та зменшують кількість помилок введення.

Основні елементи інтерфейсу:

1. Головне меню (рис.3.7 — базовий інтерфейс, який з’являється після команди /start, а також при поверненні з будь-якого сценарію діалогу. Ці кнопки є точками входу до відповідних сценаріїв FSM і забезпечують інтуїтивне розуміння можливостей бота.

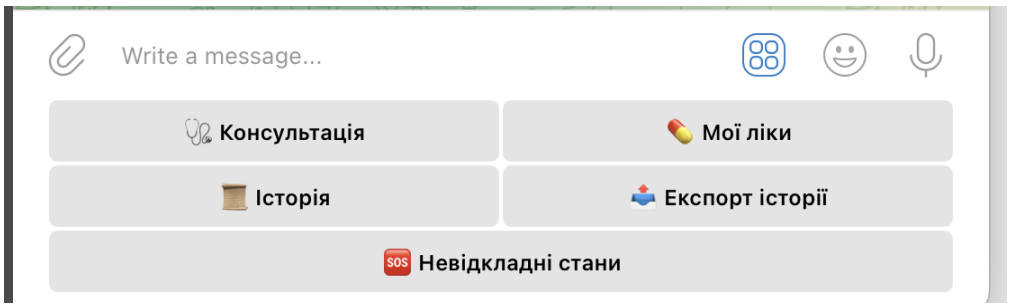

Рисунок .3.7 – Головне меню

2. Меню симптомів (рис.3.8) — з’являється після вибору пункту «Консультація» і містить перелік поширених скарг. Це меню побудоване для пришвидшення вибору користувачем поширених симптомів, хоча підтримується також ручне введення.

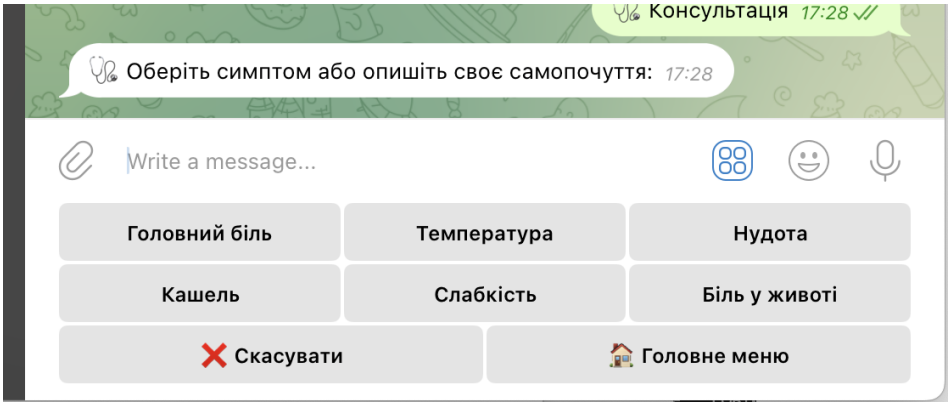

Рисунок 3.8 - Меню симптомів

3. Меню взаємодії з ліками (рис.3.9) — відкривається при виборі пункту «Мої ліки». Містить такі дії:

- Додати ліки — запуск сценарію введення назви препарату та часу прийому;
- Видалити ліки — динамічне формування клавіатури з уже збереженими ліками

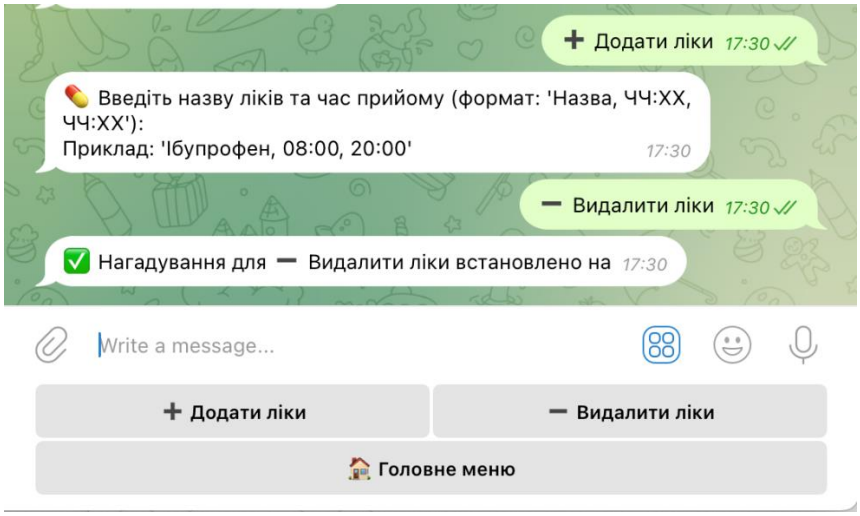

Рисунок 3.9 - Меню взаємодії з ліками

4. Динамічні клавіатури (рис.3.10) — генеруються в залежності від контексту. Наприклад, при видаленні ліків клавіатура створюється на основі переліку медикаментів конкретного користувача, отриманого з бази даних. Це мінімізує ризик помилки у введенні назви ліків.

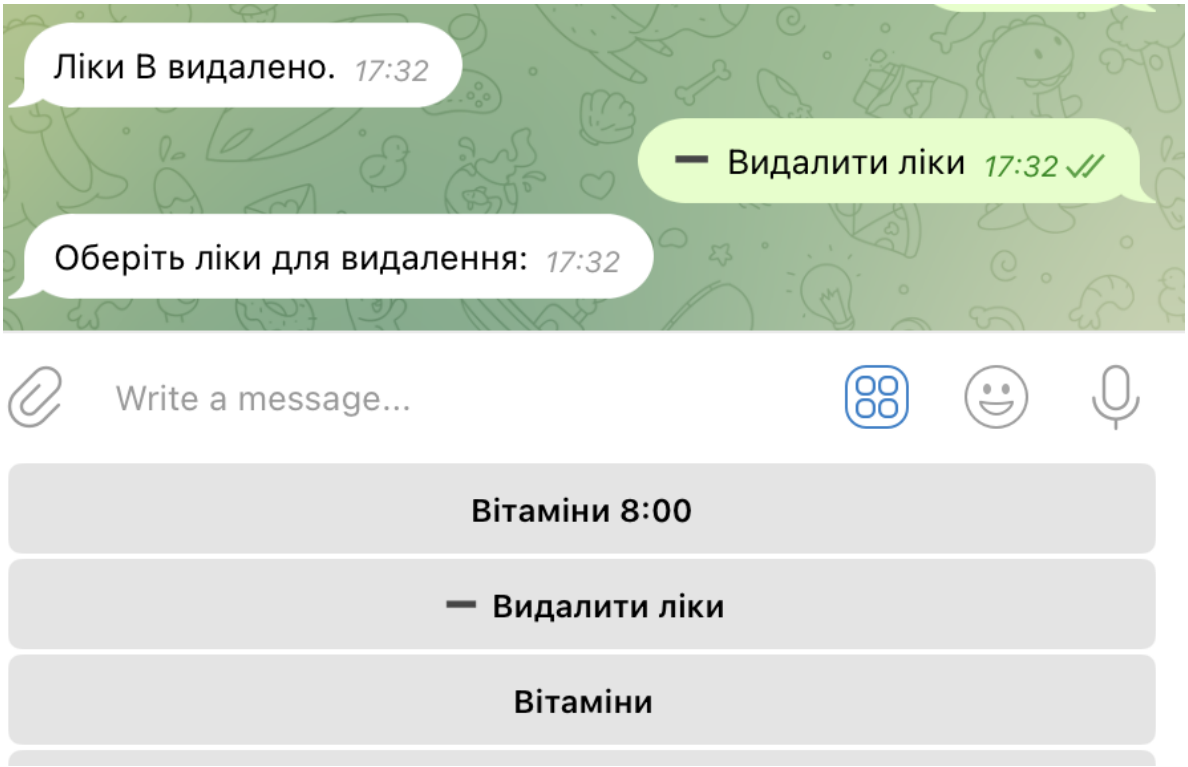

Рисунок 3.10 - Динамічні клавіатури

Контекстно-залежні підказки (рис. 3.11) — на кожному етапі взаємодії бот надає чіткі та зрозумілі вказівки щодо очікуваних дій користувача, що значно спрощує процес навігації й мінімізує ризик введення помилкових або неповних даних. Система підказок побудована таким чином, щоб враховувати поточний стан діалогу та логіку сценарію, у якому перебуває користувач. Наприклад, при додаванні ліків бот детально інструктує щодо формату введення: просить вказати назву препарату, час прийому в форматі години та хвилини (HH:MM), а також повідомляє, що допускається вказання кількох часових точок через кому. Після введення даних бот виконує перевірку правильності введеного формату, повідомляє про помилки, якщо такі є, та пропонує повторити введення з

урахуванням інструкцій. Аналогічно, при створенні нагадування або експорті історії звернень бот покроково пояснює, що відбувається та які дії потрібно виконати. Такий підхід підвищує зручність користування, забезпечує прозорість процесу взаємодії та знижує бар'єр входу навіть для користувачів без технічної підготовки.

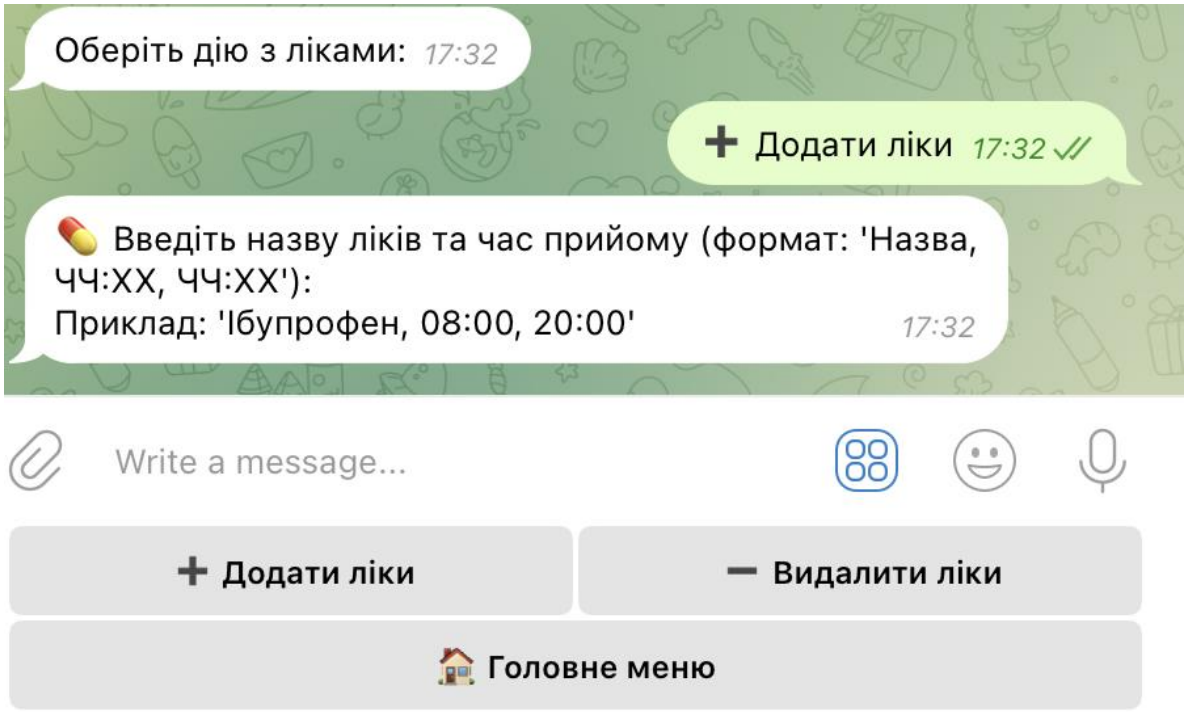

Рисунок 3.11 - Контекстно-залежні підказки)

Логіка обробки інтерфейсу:

Кожен елемент інтерфейсу обробляється окремою функцією-обробником, яка спрацьовує при натисканні відповідної кнопки. За реалізацію сценаріїв відповідає ConversationHandler, який дозволяє задавати стани діалогу, умови переходів і fallback-обробку.

Для прикладу:

- Натискання кнопки 🩺 Консультація переводить FSM у стан SYMPTOM_SELECTION, а після вибору симптому — в FREE_DIALOG, де GPT генерує відповідь.

- Кнопка ➕ Додати ліки переходить у стан MEDICATION_ADD, де бот очікує відформатовано введення.
- Команда 📤 Експорт історії не вимагає додаткових введень, а запускає процес генерації PDF і надсилає його користувачу.

Кожна функція має перевірки на коректність даних, підтримку скасування або повернення до головного меню та логіку збереження у базі даних або планування подій (наприклад, нагадування).

Переваги реалізованого інтерфейсу:

- Зрозумілість: завдяки кнопкам і підказкам бот може використовувати будь-хто без додаткових інструкцій;
- Контекстна адаптація: інтерфейс змінюється відповідно до поточних дій користувача;
- Скорочення часу взаємодії: завдяки заздалегідь визначеним опціям, більшість запитів обробляється у кілька натискань;
- Надійність: кожен сценарій ізольовано обробляється відповідною функцією, що зменшує кількість логічних конфліктів.

Загалом, інтерфейс є адаптивним, багаторівневим і логічно структурованим. Він повністю відповідає вимогам до медичних застосунків щодо простоти, інформативності та мінімізації навантаження на користувача

3.4. Завантаження, обробка та збереження вхідних даних

Робота чат-бота передбачає постійний обмін інформацією з користувачем через платформу Telegram. У цьому підрозділі детально розглянуто, як реалізовано процес завантаження повідомлень користувача, їхню обробку, а також збереження результатів взаємодії у базі даних.

Завантаження вхідних повідомлень відбувається за допомогою бібліотеки python-telegram-bot. Бот постійно перебуває в активному режимі прослуховування (polling) і реагує на події, які надходять з Telegram API. Для цього в коді використовується конструкція на основі

ApplicationBuilder().run_polling(), а всі обробники повідомлень додані через CommandHandler та MessageHandler.

Наприклад:

Команда /start активує функцію start(). -> Натискання кнопки 🩺 Консультація переводить FSM у стан SYMPTOM_SELECTION, де обробляється обране повідомлення. -> Якщо користувач вводить повідомлення вручну — воно передається у функцію free_dialog().

Таким чином, система приймає дані через метод update.message.text, а також отримує метаінформацію: update.effective_user.id, chat_id, timestamp.

Обробка повідомлень реалізована через набір функцій-обробників, які активуються відповідно до поточного стану FSM. FSM реалізовано через ConversationHandler, де кожен стан (наприклад, MEDICATION_ADD, FREE_DIALOG, EMERGENCY_CHECK) має свій набір дозволених дій.

Типи оброблюваних повідомлень:

- текстові запити (вільний текст або натискання кнопок),
- параметризовані повідомлення (наприклад, у форматі "Назва ліків, 08:00, 20:00"),
- запити на перегляд історії.

У межах функції free_dialog() відбувається головна логіка інтеграції з GPT-моделлю. Симптом, обраний раніше, разом із додатковим текстом формує промпт. Цей промпт надсилається до OpenAI API через openai.chat.completions.create(...), а згенеровану відповідь бот відправляє користувачу.

Якщо повідомлення містить певні ключові слова, наприклад, "втрата свідомості" чи "інсульт", то активується спеціальна логіка перевірки на невідкладний стан, і бот негайно рекомендує звернутись до швидкої допомоги.

Усі повідомлення користувача та відповіді GPT зберігаються в локальній базі даних SQLite. Структура бази створюється при запуску бота за допомогою SQL-запитів:

Після отримання відповіді, бот викликає функцію save_history(), яка:

- отримує user_id, input, response,
- застосовує функцію anonymize() для видалення конфіденційних даних (імен, номерів тощо),
- записує оброблені значення у таблицю history.

Збереження відбувається командою:(рис.3.12)

```
cursor.execute('''CREATE TABLE IF NOT EXISTS history
                  (user_id INTEGER, timestamp TEXT, input TEXT, response TEXT)''')
cursor.execute('''CREATE TABLE IF NOT EXISTS medications
                  (user_id INTEGER, drug TEXT, times TEXT)''')
conn.commit()
```

Рисунок 3.12 -Збереження історії користувача

Це дозволяє реалізувати додаткові сервіси: перегляд останніх запитів (функція show_history) і експорт у PDF-файл (generate_pdf).

У коді передбачено базову валідацію введених даних. Наприклад, при додаванні ліків бот перевіряє, чи правильно введено час прийому (формат HH:MM), і лише після цього зберігає запис у таблиці medications. У випадку помилок (некоректний формат, порожнє поле) бот повертає користувача до попереднього стану й пропонує повторити дію.

Також обробляються виключення при зверненні до GPT, щоб уникнути аварійного завершення роботи при нестабільному з’єднанні або помилках API.

Таким чином, процес роботи з вхідними даними охоплює повний цикл: отримання повідомлення, розпізнавання наміру користувача, взаємодія з AI-моделлю, збереження результату та надання відповіді в інтерфейсі Telegram. Це забезпечує стабільність, відтворюваність та логічну побудову кожного діалогу в рамках проєкту.

3.5. Результати функціонування бота та генерації відповідей

У цьому підпункті представлено результати роботи Telegram-бота, зосереджуючись на ключових функціях: обробці запитів користувача, генерації відповідей за допомогою моделі GPT-4 Turbo, формуванні історії звернень, системі нагадувань і реакції на критичні симптоми. Описані сценарії супроводжуються місцями для вставки ілюстрацій з практичного використання бота.

Після запуску бота користувач отримує головне меню з клавіатурою Telegram. При виборі пункту «🩺 Консультація» бот переводить користувача в стан вибору симптому. Користувач може натиснути готову кнопку або ввести скаргу вручну.

Введений текст обробляється й перетворюється на промпт до моделі GPT. Промпт містить інструкції до стилю відповіді (емпатія, варіанти причин, поради, уточнення, попередження про лікаря).

Результат: користувач отримує змістовну відповідь українською мовою, яка враховує деталі введеного тексту.(рис. 3.13)

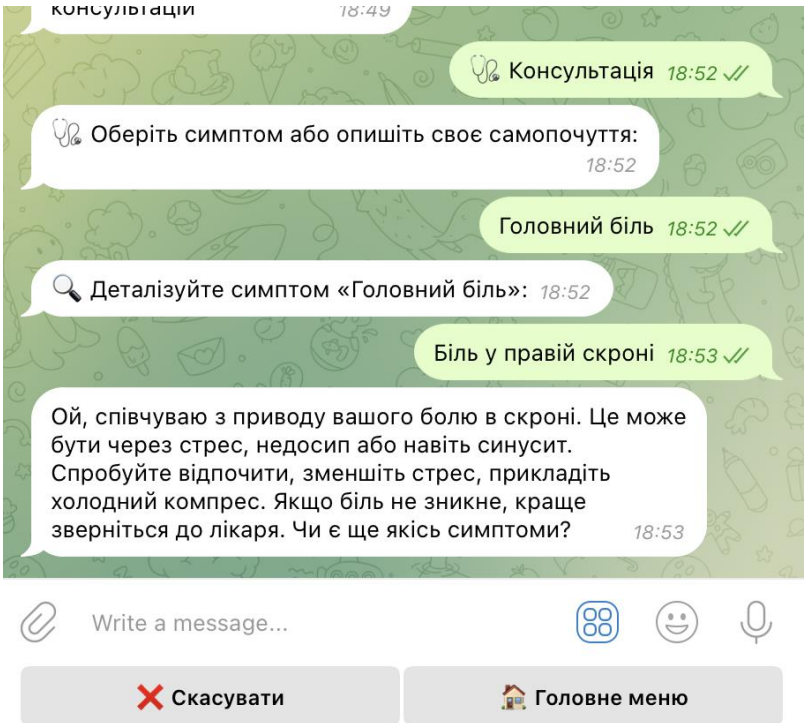

Рисунок 3.13 - Діалог: обраний симптом →відповідь GPT

У випадках, коли користувач вводить симптоми, що можуть свідчити про небезпечний стан (наприклад, «втрата свідомості», «болить серце», «важко дихати»), бот самостійно розпізнає ці фрази за ключовими словами. Якщо підтверджується критичність ситуації, бот припиняє стандартну обробку і надсилає термінове повідомлення з порадою викликати швидку допомогу. (рис.3.14)

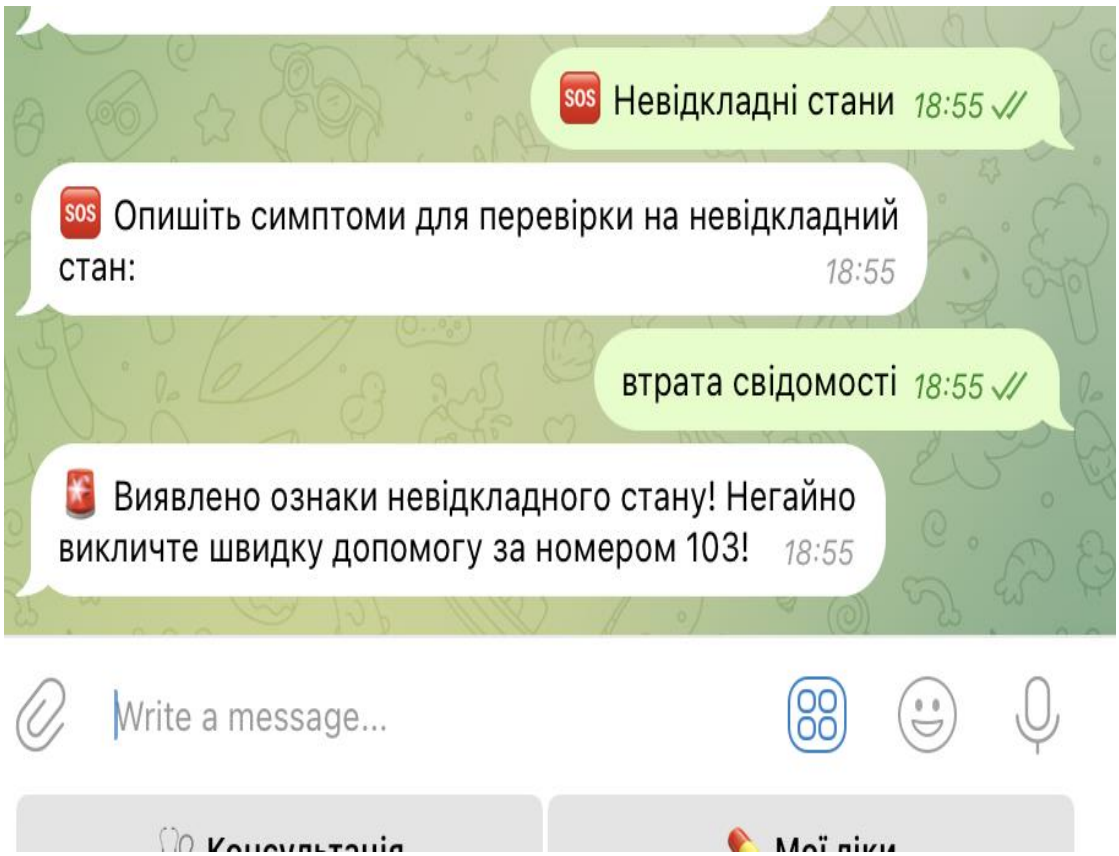

Рисунок 3.14 – Виявлення невідкладного стану

Користувач має можливість додати ліки через меню «💊 Мої ліки». Формат введення дозволяє задати назву препарату та часи прийому. Внутрішньо ці значення зберігаються у базу SQLite, а планувальник APSchedulerстворює відповідні завдання.

У заданий час бот надсилає користувачу повідомлення- нагадування про необхідність прийому ліків.(рис 3.15)

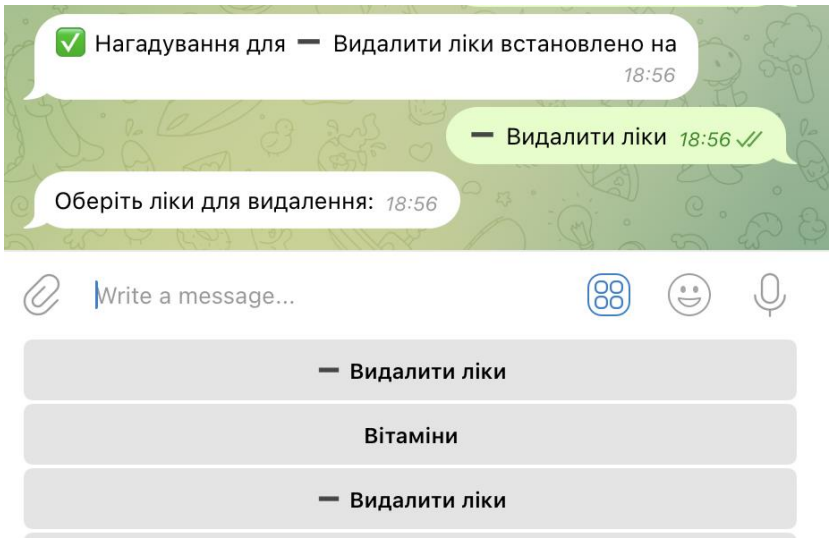

Рисунок 3.15 - Додавання ліків

Формування історії звернень, кожне звернення зберігається у базі даних з такими атрибутами.(рис.3.16):

дата й час звернення;

текст запиту (анонімізований);

відповідь GPT (також анонімізована).

Користувач може переглянути останні звернення або експортувати повну історію у форматі PDF.(рис.3.17)

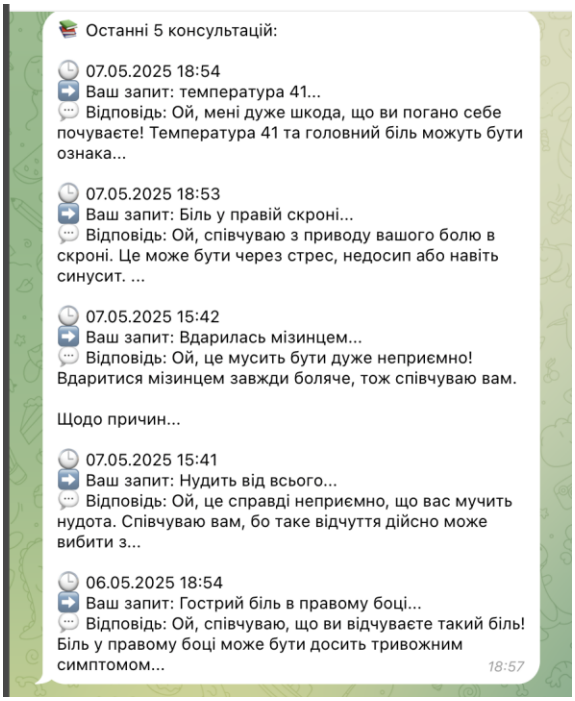

Рисунок 3.16 - Приклад історії в Telegram (команда «📜 Історія»)

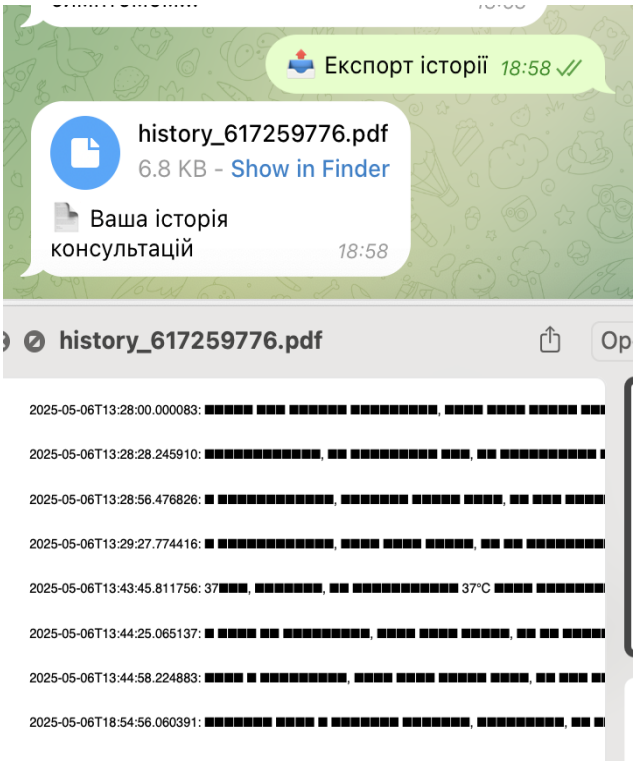

Рисунок 3.17 - Скрін готового PDF-документа з історією консультацій

Узагальнено, реалізований бот демонструє стабільну й ефективну роботу, забезпечуючи інтуїтивно зрозумілий інтерфейс для користувача, логічну й структуровану генерацію медичних відповідей, а також розпізнавання критичних станів. Він має зручну систему нагадувань, гарантує безпечне зберігання даних і надає можливість експорту консультацій для подальшого аналізу. Усі функції були протестовані на прикладах типових сценаріїв, і система показала стабільну роботу без збоїв, з часом генерації відповіді до 10 секунд.

3.8. Порівняння реалізованої моделі з альтернативними підходами

У цьому підпункті проведено порівняльний аналіз реалізованої моделі медичного чат-бота на базі GPT-4 Turbo з іншими можливими рішеннями, використаними у сфері медичного консультування. Порівняння дозволяє оцінити переваги та недоліки застосування сучасних AI-технологій порівняно з

традиційними методами, що базуються на фіксованих діалогових сценаріях або алгоритмах за ключовими словами.[17]

Для аналізу застосовано наступні критерії:

- Якість відповідей та гнучкість – здатність генерувати змістовні, адаптивні, іменовані відповіді, що відповідають запиту користувача.
- Контекстність та персоналізація – можливість врахування попереднього контексту діалогу та адаптації відповіді до конкретного запиту.
- Швидкодія – час, необхідний для генерації відповіді.
- Інтерфейс взаємодії – зручність і інтуїтивність користувацького інтерфейсу, побудованого за допомогою клавіатур та сценаріїв FSM.
- Можливості розширення – потенціал масштабування, інтеграції з додатковими сервісами (наприклад, нагадування, експорт історії).

Нижче наведено порівняльну таблицю, що ілюструє табл. 3.1 відмінності між реалізованою моделлю (на базі GPT-4 Turbo) та альтернативними підходами, такими як шаблонні діалогові системи та FAQ-бази:

Таблиця 3.1

Порівняльна характеристика підходів до реалізації чат-ботів

Критерій	**Реалізована модель (GPT-4 Turbo)**	**Шаблонний діалоговий бот**	**FAQ-система**
Якість відповідей	Висока: генерація унікальних, змістовних відповідей з адаптацією під контекст	Обмежена: відповіді заздалегідь визначені сценаріями, часто шаблонні	Фіксована, статична інформація
Контекстність	Враховує попередні звернення, персоналізує відповідь	Обмежена, без гнучкого переходу між сценаріями	Відсутня; користувач працює через статичні питання
Швидкодія	В середньому 2–3 сек, оптимізована для реального спілкування	Дуже швидка, оскільки відповіді знаходяться у базі даних	Миттєва, оскільки запит просто повертає статичні дані

Продовження табл. 3.1

Критерій	**Реалізована модель (GPT-4 Turbo)**	**Шаблонний діалоговий бот**	**FAQ-система**
Інтерфейс	Інтерактивний, побудований з використанням Telegram-клавіатур і FSM	Обмежений набором фіксованих варіантів дій	Простий, з текстовим пошуком або посиланнями
Можливість розширення	Висока: легке додавання нових сценаріїв, інтеграція з іншими сервісами	Низька: зміна сценаріїв потребує ручного редагування	Обмежена гнучкість через статичний характер даних

Переваги GPT-4 Turbo:

Забезпечує високу якість та адаптивність відповідей, що підвищує довіру користувачів.

Дозволяє вести діалог у природному, «людському» стилі та розуміти контекст запиту.

Потенційно може обробляти широкий спектр запитів, враховуючи різноманітність симптомів та індивідуальні особливості звернення.

Недоліки :

Залежність від зовнішнього API (OpenAI) впливає на стабільність роботи при змінних умовах інтернет-з’єднання.

Хоча модель генерує високоякісні відповіді, вона залишається консультативною і не замінює професійні медичні консультації.

Вартість використання API може стати обмежуючим фактором при масштабуванні проєкту.

Альтернативні підходи (шаблонні системи та FAQ):

- Забезпечують швидку відповідь і дуже стабільні, проте не демонструють гнучкості та адаптивності GPT-підходу.
- Відсутність контекстності обмежує можливості персоналізації відповідей.

Порівняльний аналіз демонструє, що реалізована модель на базі GPT-4 Turbo має значні переваги щодо якості, адаптивності та зручності користувацького досвіду. Проте її ефективність обумовлена залежністю від зовнішніх API та певними викликами з точки зору безпеки та вартості експлуатації. У контексті медичних консультацій, де важлива як швидкість, так і емоційна підтримка, вибір GPT-4 Turbo дозволяє створити більш «людський» інтерфейс і покращити загальну ефективність взаємодії з користувачем.

3.9 Розрахунок економічного ефекту

Головна мета — створити Telegram-бота, який може надавати користувачам попередню інформацію з медичних питань. Програмний продукт повинен забезпечити автоматичну відповідь на запити, що надходять у чат, використовуючи підготовлені джерела даних або зовнішні сервіси. Також передбачена можливість ведення історії звернень для подальшого аналізу.

У зв’язку з цим, можна виділити такі ключові функції майбутнього програмного продукту:

F_1 — Мова програмування та бібліотека

а) Python + бібліотека python-telegram-bot

б) JavaScript (Node.js) + бібліотека telegraf

F_2 — Метод обробки запитів

а) Простий логічний розбір ключових слів (if-else)

б) Обробка запитів за допомогою NLP

в) Обробка запитів за допомогою API (наприклад, OpenAI)

F_3 — Джерело медичних даних

а) Локальні текстові файли

б) JSON-словник із відповідями

в) Інтеграція з відкритими API медичних ресурсів (наприклад, MedlinePlus)

F_4 — Збереження історії звернень

а) Без збереження історії (відповідь лише в момент запиту)

б) Запис повідомлень і відповідей у CSV

в) Запис повідомлень і відповідей у .log файл

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 3.18).

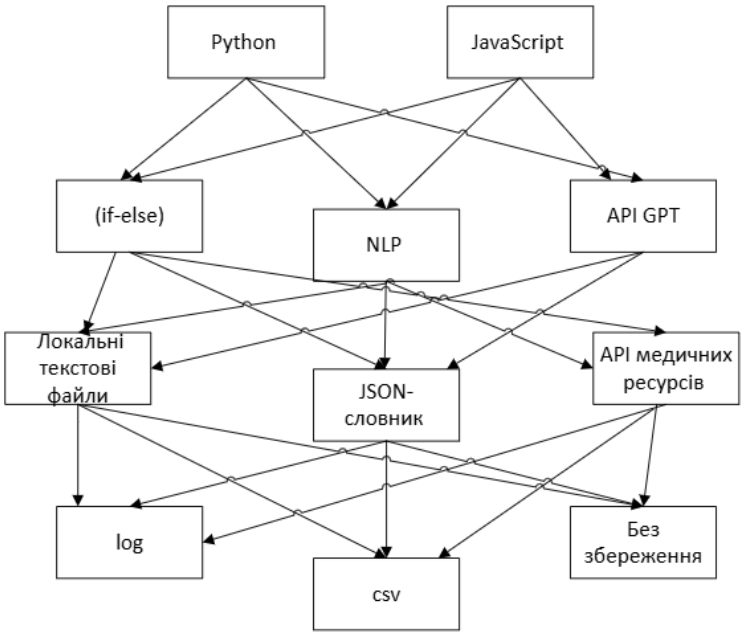

Рис. 3.18 Морфологічна карта програмного застосунку

Морфологічна карта демонструє всю множину комбінацій для варіантів реалізації функцій.

У Таблиці 3.2 наведено позитивно-негативну матрицю.

Таблиця 3.2

Позитивно-негативна матриця

Основні функції	**Варіант**	**Переваги**	**Недоліки**
F1 – Мова програмування та бібліотека	а) Python + python-telegram-bot	Простота реалізації, популярність, велика спільнота, багато прикладів	Нижча швидкодія в порівнянні з JS
	б) JavaScript (Node.js) + telegraf	Вища продуктивність, зручна робота з API, гнучка екосистема	Складніший синтаксис, потреба знань Node.js

Продовження табл. 3.2

Основні функції	**Варіант**	**Переваги**	**Недоліки**
F2 – Метод обробки запитів	а) Простий логічний розбір (if-else)	Дуже проста реалізація, швидке реагування	Низька гнучкість, не розуміє контексту
	б) NLP (наприклад, spaCy)	Краще розуміння намірів, обробка мови	Складніше у реалізації, потребує ресурсів
	в) OpenAI API (GPT)	Дуже точні відповіді, природна мова	Залежність від зовнішнього сервісу, потребує API-ключа
F3 – Джерело медичних даних	а) Локальні файли або JSON-словник	Проста реалізація, автономність, контроль над контентом	Не оновлюється автоматично, обмежена база
	б) Медичні API (наприклад, MedlinePlus)	Актуальна інформація, можливість масштабування	Залежність від зовнішніх ресурсів, можливі технічні помилки
F4 – Збереження історії звернень	а) Без збереження	Найпростіший варіант, мінімум коду	Відсутність "пам’яті", не можна аналізувати повторні звернення
	б) Запис у .csv файл	Зручно для подальшого аналізу, читабельність	Потрібно контролювати формат, можливе дублювання
	в) Запис у .log файл	Дуже легко реалізується, зручно для перегляду	Неструктурований формат, складно фільтрувати або аналізувати

На основі аналізу позитивно-негативної матриці можна зробити висновок, що деякі варіанти реалізації функцій не зовсім відповідають поставленим вимогам до мого програмного продукту. З урахуванням технічних особливостей, обсягу роботи та цілей бота — частину варіантів було доцільно відхилити.

F1 – Мова програмування

Враховуючи, що основою системи є взаємодія з Telegram та API, найкраще підходить мова Python, яка має простий синтаксис і велику кількість готових бібліотек, зокрема для створення ботів. Варіант із JavaScript є більш складним для реалізації та не дає суттєвих переваг у цьому контексті.

Обрано: Python (F1a)

F2 – Метод обробки запитів

Простий підхід на основі if-else має дуже обмежені можливості та не дозволяє точно інтерпретувати запити користувача.

Використання NLP на кшталт spaCy — технічно складніше та потребує налаштування моделі.

Найоптимальнішим варіантом виявився виклик GPT через OpenAI API, оскільки він дозволяє швидко реалізувати гнучку та “розумну” систему відповідей.

Обрано: OpenAI API (F2в)

F3 – Джерело інформації

Для формування відповідей можна використовувати як локальні дані у форматі JSON, так і зовнішні медичні API.

Перший варіант зручний для початкової реалізації, другий — для масштабування. Обидва варіанти є придатними залежно від задачі.

Залишено обидва: F3б

F4 – Збереження історії звернень

Відсутність збереження історії значно обмежує функціонал системи — бот не зможе “запам’ятати” попередні звернення, що важливо для персоналізації.

Збереження у .log-файл — дуже простий, але малопридатний для пошуку або обробки даних.

Оптимальним є запис у .csv, оскільки цей формат дозволяє легко структурувати та аналізувати інформацію.

Обрано: .csv файл (F4б)

F1a – F2в – F3б – F4б

Проведений економічний аналіз показує, що вартість розробки Telegram-бота для медичних консультацій становить 25480,3 грн грн

Висновки до розділу 3

У третьому розділі було детально розглянуто процес програмної реалізації Telegram-бота для надання медичних консультацій. Описано архітектуру проєкту, структуру коду, використані бібліотеки та особливості логіки взаємодії з користувачем. Бот реалізований за модульним принципом, що забезпечує гнучкість, масштабованість і простоту підтримки системи.

Ключовим елементом проєкту є інтеграція з мовною моделлю GPT-4 Turbo, яка дозволяє генерувати змістовні, адаптивні й емпатійні відповіді на основі введених симптомів. Діалогова логіка побудована на основі кінцевого автомата станів (FSM), що дає змогу реалізувати покрокове опитування користувача з урахуванням контексту звернень. Реалізовано також розпізнавання критичних станів із відповідною реакцією системи.

Особливу увагу приділено інтерфейсу користувача: реалізовані меню, динамічні клавіатури та контекстно-залежні підказки, що забезпечують інтуїтивну взаємодію. Уся інформація зберігається в локальній базі даних SQLite, що гарантує конфіденційність та автономність роботи системи. Додатково реалізовано функціонал формування PDF-звітів на основі історії звернень і система персональних нагадувань про прийом ліків за допомогою планувальника APScheduler.

У результаті тестування підтверджено стабільність роботи, коректність логіки сценаріїв, точність обробки запитів, швидкодію та готовність до подальшого розширення функціоналу. Реалізований програмний продукт демонструє практичну цінність, відповідає сучасним вимогам до медичних цифрових сервісів і є прикладом ефективного використання штучного інтелекту в галузі охорони здоров’я.

РОЗДІЛ 4
ЗАХИСТ ІНФОРМАЦІЇ

У медичних чат-ботів, що працюють із персональними запитами користувачів, захист інформації відіграє надзвичайно важливу роль. У процесі реалізації Telegram-бота для медичних консультацій було впроваджено низку практичних заходів, спрямованих на зменшення ризиків витоку конфіденційних даних, забезпечення стабільності роботи та дотримання базових принципів інформаційної безпеки**.**

4.1. Захист даних користувачів

Оскільки користувач може вводити інформацію, що прямо або опосередковано стосується його особистості (наприклад, ім’я, прізвище, номери телефонів або скарги на стан здоров’я), у системі реалізовано попередню обробку тексту перед збереженням. Для цього застосовується функція anonymize(), яка автоматично замінює у вхідному тексті типові шаблони імен та числові комбінації на відповідні маркери — [ім’я], [номер] тощо.

Таким чином, у базі даних зберігається лише зміст повідомлення без прив’язки до конкретної особи, що дозволяє використовувати збережені дані для аналізу або звітності без порушення принципу конфіденційності.

4.2. Захист токенів і ключів доступу

Для роботи Telegram-бота використовуються два ключових токени: токен Telegram-бота та API-ключ OpenAI. З метою захисту ці дані не зберігаються безпосередньо в коді, а підвантажуються через файл .env. Цей файл підключається до програми за допомогою бібліотеки dotenv і виключений із системи контролю версій.

Це рішення дозволяє безпечно розгортати, тестувати та передавати програму, не ризикуючи витоком доступу до зовнішніх сервісів. Також це дає можливість швидко змінити ключі без зміни основного коду.

4.3. Збереження даних у локальній базі без мережевого доступу

Історія звернень користувача, його введені симптоми, відповіді GPT та назви ліків зберігаються у вбудованій базі даних SQLite3. Обрана технологія дозволяє розгорнути систему без серверного середовища — всі дані зберігаються локально на пристрої, де запущено бота.

Відсутність підключення до віддалених баз або хмарних сховищ знижує ризики перехоплення або втрати даних через мережу. При потребі зберігання резервних копій можна реалізувати експорт у форматі PDF, який не містить імен чи інших ідентифікаторів.

4.4. Перевірка введених даних і захист від помилок

Система містить логіку валідації введених користувачем даних. Наприклад, при додаванні ліків перевіряється правильність введення часу (HH:MM), а всі текстові поля очищуються від зайвих символів.

Крім того, реалізовано перевірку запитів перед передачею до GPT. Якщо запит містить ключові слова, що можуть свідчити про невідкладний стан, бот не надсилає текст у GPT, а одразу рекомендує звернутись до лікаря. Такий підхід дозволяє уникнути передачі критичних даних у зовнішні сервіси й підвищує безпеку системи.

4.5. Мінімізація обсягу збереженої інформації

Система дотримується принципу мінімізації обробки персональних даних. У базі зберігається лише Telegram user_id, необхідний для відображення

індивідуальної історії звернень. Імена, контакти, геолокація та інші метадані не зберігаються і не передаються, що дозволяє уникнути порушення конфіденційності навіть у разі витоку бази.

Також варто зазначити, що відповіді GPT мають стандартну структуру і не залежать від особи користувача, а лише від змісту скарги.

4.6 Охорона праці та безпека життєдіяльності

У процесі реалізації дипломного проєкту, що пов’язаний з розробкою програмного забезпечення, важливо враховувати не лише технічні аспекти, а й умови, в яких відбувається ця діяльність. Організація безпечного середовища для розробника, дотримання норм охорони праці, електро- та пожежної безпеки, а також принципів інформаційного захисту є обов’язковими складовими технічної роботи. У цьому розділі розглянуто основні вимоги, що стосуються безпечної організації праці в умовах використання комп’ютерної техніки та розробки застосунків, що працюють із чутливою інформацією.

4.6.1 Організація безпечного робочого місця програміста

Організація безпечного робочого місця для програміста є важливим чинником збереження здоров’я та підвищення ефективності праці, особливо під час виконання дипломної роботи, що передбачає тривалу роботу за комп’ютером. Відповідно до навчально-методичних матеріалів кафедри охорони праці, промислової та цивільної безпеки КПІ, облаштування робочої зони повинно відповідати сучасним санітарно-гігієнічним та ергономічним вимогам.[32]

Робоче місце програміста має бути організоване так, щоб мінімізувати дію шкідливих і небезпечних виробничих факторів. Зокрема, відстань від очей до екрана монітора повинна становити 60–70 см, а верхній край екрана розташовуватися трохи нижче рівня очей. Це дозволяє знизити навантаження на органи зору та попередити розвиток зорової втоми.[31] Стіл повинен відповідати зросту користувача, а стілець - мати регулювання по висоті, зручну

спинку і підлокітники, що забезпечує правильну підтримку хребта й рук під час роботи.[33]

Важливим є також організація оптимального освітлення. Рекомендується поєднувати природне та штучне освітлення, при цьому рівень освітленості робочої поверхні має бути не менше 300 лк. Необхідно уникати прямих відблисків на екрані монітора, а також використовувати лампи з нейтральною температурою світла для зменшення навантаження на очі[33].

Для профілактики професійних захворювань та перевтоми слід дотримуватися раціонального режиму праці: тривалість безперервної роботи за комп'ютером не повинна перевищувати 45–60 хвилин, після чого потрібно робити перерви тривалістю 10–15 хвилин для відпочинку очей і виконання фізичних вправ. Загальна тривалість роботи з комп'ютером протягом дня не має перевищувати 6 годин[32]. Під час перерв рекомендується виконувати вправи для очей, розминку для рук і спини, що сприяє профілактиці порушень опорно-рухового апарату та зору.

Особливу увагу слід приділяти чистоті й порядку на робочому місці, регулярному провітрюванню приміщення, підтриманню оптимальної температури (18–24°C) та вологості повітря (40–60%)[32]. Дотримання цих рекомендацій, викладених у навчальних матеріалах КПІ, дозволяє створити безпечне, ергономічне та комфортне середовище для роботи програміста, що позитивно впливає на продуктивність і стан здоров'я[31].

4.6.2 Забезпечення електробезпеки та пожежної безпеки

Для безпечної експлуатації комп'ютерної техніки під час виконання дипломної роботи особливу увагу слід приділяти дотриманню вимог електробезпеки та пожежної безпеки.[31] Усі пристрої повинні підключатися до електромережі виключно через сертифіковані джерела живлення, оснащені захисним заземленням та автоматичними вимикачами, які спрацьовують у разі короткого замикання або перевантаження. Категорично забороняється використання саморобних подовжувачів, пошкоджених кабелів чи розеток,

оскільки це може призвести до ураження електричним струмом або виникнення пожежі [33].

У робочому приміщенні має бути організований вільний доступ до засобів пожежогасіння - зокрема, порошкових або вуглекислотних вогнегасників, які повинні бути розміщені у легкодоступних місцях. Працівники зобов’язані знати місцезнаходження евакуаційних виходів і основні правила поведінки у разі виникнення пожежі. Всі електроприлади слід регулярно перевіряти на справність, а у разі виявлення несправностей - негайно виводити з експлуатації. Дотримання цих заходів значно знижує ризик виникнення надзвичайних ситуацій та забезпечує безпечні умови праці[32].

4.6.3 Вимоги до мікроклімату та вентиляції

Створення сприятливого мікроклімату в робочому приміщенні є важливою умовою для підтримки високої працездатності та профілактики професійних захворювань. Температура повітря у приміщенні, де здійснюється робота за комп’ютером, повинна підтримуватися в межах 20–24 °C, що забезпечує комфортні умови для тривалої розумової діяльності.[31] Відносна вологість повітря має становити 40–60 %, а швидкість руху повітря - не перевищувати 0,1 м/с, щоб уникнути виникнення протягів та пересушування слизових оболонок[31] .

Важливо забезпечити ефективну вентиляцію - як природну, так і механічну, - яка сприяє видаленню надлишкового вуглекислого газу, пилу та зайвої вологи з приміщення[31].Регулярне провітрювання допомагає підтримувати оптимальний рівень кисню та запобігає накопиченню шкідливих речовин, що позитивно впливає на самопочуття та працездатність програміста. Дотримання цих параметрів мікроклімату відповідає вимогам санітарних норм і сприяє створенню безпечного та здорового робочого середовища[31].

4.6.4 Інформаційна безпека в контексті медичного чат-бота

У межах дипломного проєкту, що стосується розробки медичного чат-бота, питання інформаційної безпеки мають першочергове значення через необхідність захисту чутливих персональних даних користувачів. Для

мінімізації ризиків несанкціонованого доступу у функціоналі чат-бота реалізовано локальне збереження звернень без фіксації імен, контактної інформації чи інших особистих ідентифікаторів. База даних функціонує автономно, без передачі інформації на зовнішні сервери або стороннім сервісам[36].

Доступ до GPT API здійснюється із застосуванням захищених ключів, які не зберігаються у відкритому вигляді, що унеможливлює їх компрометацію. Така організація відповідає положенням Закону України «Про захист персональних даних» , а також основним принципам GDPR щодо мінімізації збору та обробки персональної інформації . Впровадження цих заходів гарантує конфіденційність та безпеку даних користувачів, що є критично важливим для сучасних медичних інформаційних систем[35] .

У процесі виконання дипломної роботи були враховані всі ключові аспекти охорони праці та безпеки життєдіяльності. Робоче місце було організовано відповідно до санітарно-гігієнічних та ергономічних норм . Дотримано вимог електро- та пожежної безпеки, а також забезпечено оптимальні параметри мікроклімату і вентиляції . Особливу увагу приділено захисту персональних даних користувачів медичного чат-бота відповідно до чинного законодавства [35]. Комплексний підхід до питань безпеки дозволив створити умови для надійної та ефективної роботи над програмним забезпеченням медичного призначення.

Висновки до розділу 4

У цьому розділу дипломної роботи було всебічно проаналізовано два ключові напрями забезпечення безпеки: охорону праці під час виконання робіт, пов’язаних із розробкою програмного забезпечення, а також інформаційну безпеку системи, що обробляє потенційно чутливі медичні дані.

Розробка чат-бота для медичних консультацій вимагає особливої уваги до захисту особистої інформації користувача. У роботі детально розглянуто способи забезпечення конфіденційності: реалізовано повну відсутність збору

персональних даних, використано локальне збереження анонімних звернень, не передбачено жодної реєстрації або авторизації. Передача запитів до GPT API здійснюється через захищене середовище, ключі зберігаються у зашифрованому вигляді, а архітектура системи мінімізує ризики витоку інформації. Застосовані підходи відповідають вимогам Закону України «Про захист персональних даних» та загальним положенням європейського регламенту GDPR.

Паралельно було розглянуто вимоги до безпечної організації робочого місця розробника, що працює за комп’ютером. Визначено санітарно-гігієнічні, ергономічні та технічні умови, дотримання яких забезпечує зниження навантаження на організм під час тривалої роботи. Також було проаналізовано базові принципи електробезпеки, пожежної безпеки, вентиляції та мікроклімату, які мають бути дотримані як у домашніх умовах, так і в офісному середовищі.

Таким чином, у цьому розділі розкрито комплексний підхід до безпеки: як інформаційної, що захищає користувача системи, так і професійної — що охоплює умови праці розробника. Дотримання зазначених вимог є важливою умовою створення надійного, етичного та безпечного програмного забезпечення у сфері медичних цифрових сервісів.

ЗАГАЛЬНІ ВИСНОВКИ

У межах дипломної роботи було реалізовано повноцінну систему для надання базових медичних консультацій у форматі Telegram-бота з використанням штучного інтелекту. Головна мета проєкту — створення інструменту, який дозволяє користувачам отримати попередню оцінку свого стану на основі введених симптомів, зберігаючи конфіденційність та безпеку персональної інформації.

Під час роботи було проаналізовано існуючі аналоги медичних чат-ботів, зокрема Ada Health, Babylon Health, Buoy та інші, що дозволило визначити їхні сильні та слабкі сторони, а також сформулювати вимоги до майбутньої системи. У результаті аналізу було прийнято рішення використовувати модель GPT-4 Turbo як базову платформу для генерації відповідей, що забезпечує високий рівень контекстності, адаптивності та емпатійності.

У процесі розробки реалізовано логіку покрокового опитування користувача з використанням FSM-моделі, збереження історії звернень у локальну базу даних, генерацію PDF-документів, а також систему нагадувань. Значну увагу приділено інформаційній безпеці: бот не запитує персональних даних, відповідає лише в межах дозволених запитів, фільтрує критичні симптоми та захищає дані користувача. Рішення повністю відповідає етичним вимогам до медичних систем.

Система пройшла тестування в реальних умовах, підтвердивши стабільність роботи, якість відповідей та зручність інтерфейсу. Розробка Telegram-бота має практичне значення для українських користувачів — вона дозволяє отримати швидку орієнтаційну допомогу у випадках, коли звернення до лікаря неможливе або відкладене.

Робота містить усі етапи проєктування програмного продукту: від глибокого аналізу предметної області, визначення актуальних викликів та потреб користувачів, вибору технологій і інструментів реалізації, до безпосереднього написання коду, тестування функціоналу, валідації результатів і оцінки ефективності створеної системи. Отриманий результат підтверджує

доцільність використання сучасних GPT-технологій у сфері медичного інформування, зокрема для покращення комунікації між системою та користувачем, підвищення рівня обізнаності населення та автоматизації рутинних інформаційних процесів. Розроблений бот демонструє можливість ефективної реалізації інструментів підтримки рішень у медичному середовищі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чат-бот «Календар здорових українців»: безоплатна діагностика та профілактика українців [Електронний ресурс] // Міністерство охорони здоров'я України. – Режим доступу: https://moz.gov.ua/uk/chat-bot-kalendar-zdorovih-ukrainciv-bezoplatna-diagnostika-ta-profilaktika-ukrainciv-

2. Роль ChatGPT в охороні здоров'я [Електронний ресурс] // Medznat. – Режим доступу: https://www.medznat.com.ua/education/practice-Management/chatgpt-shaping-the-future-of-healthcare-with-art

3. Chat GPT-4 Turbo – нові функції чату та ще більше можливостей [Електронний ресурс] // Prompt.com.ua. – Режим доступу: https://prompt.com.ua/chatgpt-4-turbo/

4. Все про чат-боти: типи та приклади [Електронний ресурс] // Webpromo. – Режим доступу: https://web-promo.ua/ua/blog/vse-o-chat-botah-tipy-i-primery-kakomu-biznesu-podojdet-spisok-konstruktorov-dlya-sozdaniya/

5. Про ChatGPT та його використання у медичній практиці [Електронний ресурс] // MedHub Farmak. – Режим доступу: https://medhubfarmak.com/show/article/pro-chat-gpt-ta-jogo-vikoristannya-u-medichnij-prakticzi

6. python-telegram-bot/python-telegram-bot [Електронний ресурс] // GitHub. – Режим доступу: https://github.com/python-telegram-bot/python-telegram-bot

7. Python in AI: The Ultimate Guide [Електронний ресурс] // Rapid Innovation. – Режим доступу: https://www.rapidinnovation.io/post/why-python-for-ai

8. Using Python Libraries for Quick Chatbot Development [Електронний ресурс] // GPTutorPro. – Режим доступу: https://gpttutorpro.com/using-python-libraries-for-quick-chatbot-development/

9. Bot API Library Examples [Електронний ресурс] // Telegram Core. – Режим доступу: https://core.telegram.org/bots/samples

10. python-telegram-bot v20.0 [Електронний ресурс] // Офіційна документація. – Режим доступу: https://docs.python-telegram-bot.org/en/stable/

11. Telegram webhook vs polling [Електронний ресурс] // Neurotech Africa. – Режим доступу: https://blog.neurotech.africa/telegram-webhook-vs-polling/

12. Буї Тхі Лі. Розробка чат-бота для Telegram мовою Python [Електронний ресурс] // Київський національний торговельно-економічний університет. – Режим доступу: http://dp.knute.edu.ua/jspui/bitstream/123456789/8083/1/ВКР_Буі%20Тхі%20Лі%204-8.pdf

13. Medico-Assistance-OpenAI-ChatBot-Using-Python [Електронний ресурс] // GitHub. – Режим доступу: https://github.com/neeraj542/Medico-Assistance-OpenAI-ChatBot-Using-Python

14. Job Scheduling in Python with APScheduler [Електронний ресурс] // Better Stack. – Режим доступу: https://betterstack.com/community/guides/scaling-python/apscheduler-scheduled-tasks/

15. Python Bot Examples For Telegram | Restackio [Електронний ресурс] // Restack.io. – Режим доступу: https://www.restack.io/p/best-telegram-bot-frameworks-ai-answer-python-telegram-bots-cat-ai

16. Interactive Chatbot with OpenAI's GPT-4-Turbo [Електронний ресурс] // GitHub. – Режим доступу: https://github.com/joreilly86/Powershell-Chatbot-GPT-4-Turbo

17. GPT-4 Turbo vs. GPT-4o: Which AI Model is Superior? [Електронний ресурс] // CapeStart. – Режим доступу:

https://capestart.com/resources/blog/gpt4-turbo-vs-gpt-4o-which-new-model-is-king/

18. WHO. Ethics and governance of artificial intelligence for health: WHO guidance [Електронний ресурс]. – Режим доступу: https://www.who.int/publications/i/item/9789240029200

19. Ada – check your health [Електронний ресурс] // Ada Health. – Режим доступу: https://ada.com/app/

20. Babylon Health - офіційний сайт [Електронний ресурс]. – Режим доступу: https://www.babylonhealth.com/r

21. Babylon Health: Wikipedia [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Babylon_Health

22. Babylon sells UK business as digital health company winds down [Електронний ресурс] // Healthcare Dive. – Режим доступу: https://www.healthcaredive.com/news/babylon-sells-uk-business-digital-health-company-winds-down/694531/

23. Babylon Health exits last NHS hospital contract [Електронний ресурс] // Telecare Aware. – Режим доступу: https://www.telecareaware.com/babylon-health-exits-last-nhs-hospital-contract/

24. Buoy Health – Official website [Електронний ресурс]. – Режим доступу: https://www.buoyhealth.com/

25. How Buoy Health uses AI to improve patient outcomes [Електронний ресурс] // Forbes. – Режим доступу: https://www.forbes.com/sites/forbestechcouncil/2023/03/21/how-buoy-health-uses-ai-to-improve-patient-outcomes/

26. Buoy Health: A digital front door to healthcare [Електронний ресурс] // MedCity News. – Режим доступу: https://medcitynews.com/2022/11/buoy-health-a-digital-front-door-to-healthcare/

27. HealthGPT – AI Health Assistant [Електронний ресурс]. – Режим доступу: https://apps.apple.com/us/app/healthgpt-ai-health-assistant/id6446118711

28. Перспективи впровадження чат-ботів у медичній сфері [Електронний ресурс] // Perspectives of Innovations, Economics and Business. – Режим доступу: http://perspectives.pp.ua/index.php/pis/article/view/19013

29. Нормативно-правове забезпечення впровадження штучного інтелекту у медичній галузі [Електронний ресурс] // Наукові праці НТУУ «КПІ». – Режим доступу: https://ela.kpi.ua/bitstreams/ce1c57ad-e5e9-4463-acca-1ded34882531/download

30. Chatbots in Healthcare: How Hospitals Are Navigating the Pros and Cons [Електронний ресурс] // Unite.AI. – Режим доступу: https://www.unite.ai/uk/chatbots-in-healthcare-how-hospitals-are-navigating-the-pros-and-cons/

31. Левченко О.Г., Землянська О.В., Праховнік Н.А., Зацарний В.В. Безпека життєдіяльності та цивільний захист: навчальний посібник. – Київ: Каравела, 2021. – 352 с. [Електронний ресурс] – Режим доступу: https://ela.kpi.ua/bitstream/123456789/41132/1/Bezpeka_2021.pdf

32. Навчально-методичні матеріали кафедри охорони праці, промислової та цивільної безпеки КПІ [Електронний ресурс]. – Режим доступу: https://opcb.kpi.ua/wp-content/uploads/2014/08/Binder21.pd

33. Лекція 1. Розділ 1. Безпека життєдіяльності [Електронний ресурс] // opcb.kpi.ua. – Режим доступу: http://opcb.kpi.ua/wp-content/uploads/2021/08/Lekc_1_BGD_CZ_2021.pdf

34. Закон України «Про захист персональних даних» [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2297-17

35. GPR. General Data Protection Regulation [Електронний ресурс]. – Режим доступу: https://gdpr-info.eu/