МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТУ ПО ДОГЛЯДУ ЗА ДОМАШНІМИ УЛЮБЛЕНЦЯМИ»

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи КБ-21-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»

_____ Завріна Дмитра Романовича
(підпис) (прізвище, ім'я, по батькові)

Науковий керівник _____ Грищенко Світлана Миколаївна, к.пед.н., ст. досл.
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-професійної програми _____ Омельчук Антон Анатолійович, к.т.н.
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних технологій і систем протокол від «13» травня 2025 № 18

Завідувач кафедри _____ Філоненко Михайло Миколайович, к.ф.-м.н., доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2025

Реєстрація № 42 23.05.2025 _____
(дата) (підпис відповідальної особи)

Результати перевірки до захисту 23.05.2025 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 5 (відмінно) 04.06.2025 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ В.М. Франчук
(підпис)

Члени комісії: _____ М.М. Філоненко
(підпис)

_____ С.М. Грищенко
(підпис)

_____ А.А. Омельчук

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	**від 0 до 30**	29		Грищенко С.М. Франчук В.М. Філоненко М.М. Грищенко С.М. Омельчук А.А. Суховецька Л.П.
2. Дотримання вимог щодо оформлення	**від 0 до 10**	10		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	**від 0 до 5**	4		
4. Рецензент	**від 0 до 5**	5		
РАЗОМ БАЛІВ	**від 0 до 50**	48		

Оцінювання кваліфікаційної роботи науковим керівником - 47 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалоюЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
95	A	5 (відмінно)	04.06.2025	Франчук В.М. Філоненко М.М. Грищенко С.М. Омельчук А.А. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	**A**	**5 (відмінно)**
80-89	**B**	**4 (добре)**
70-79	**C**	
60-69	**D**	**3 (задовільно)**
50-59	**E**	
35-49	**FX**	**2 (незадовільно)**

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Інформаційні управляючі системи і технології (за галузями)»

Затверджую
Завідувач кафедри _____ Філоненко М.М.
«14» листопада 2024

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Завріну Дмитру Романовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Розробка інтелектуального чат-боту по догляду за домашніми улюбленцями
Затверджена наказом ректора від «13» листопада 2024 № 1717
2. Строк здачі закінченої роботи «08» травня 2025
3. Мета роботи створення Telegram-бота, що оптимізує догляд за тваринами та взаємодію з ветеринарними послугами.
Об'єкт дослідження дизайн та розробка інтелектуальних чат-ботів.
Предмет дослідження створення функціоналу й інтерфейсу Telegram-бота для власників тварин з рекомендаціями з догляду та медичних послуг.
Орієнтовна база дослідження розроблений бот, що надає поради, нагадування про вакцинацію, прийом ліків і слідкує за здоров'ям тварин.
4. Вихідні дані до роботи методи створення Telegram-ботів: Python, алгоритми машинного навчання, обробка природної мови, інтеграція з базами даних.
5. Перелік графічного матеріалу скріншоти інтерфейсу бота, схема бази даних користувачів, тварин і взаємодій.
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП
РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА З ДОГЛЯДУ ЗА ДОМАШНІМИ УЛЮБЛЕНЦЯМИ
1.1. Сутність чат-ботів та їх роль у догляді за домашніми улюбленцями
1.2. Огляд технологій та інструментів для створення чат-ботів
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ЧАТ-БОТА З ДОГЛЯДУ ЗА ДОМАШНІМИ УЛЮБЛЕНЦЯМИ
2.1. Проєктування архітектури та алгоритмів роботи чат-бота
2.2. Архітектура чат-бота, вибір середовища розробки, тестування та оцінка результатів його роботи
РОЗДІЛ 3. ПЕРСПЕКТИВИ РОЗВИТКУ ЧАТ-БОТА З ДОГЛЯДУ ЗА ДОМАШНІМИ УЛЮБЛЕНЦЯМИ
ВИСНОВКИ
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційноїроботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 30.10.2024	До 30.10.2024
2	**І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	27.12.2024	27.12.2024
3	**ІІ розділ** (аналіз експериментальної частини про методики досліджень)	17.02.2025	17.02.2025
4	**ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	16.04.2025	16.04.2025
5	**Висновки**	01.05.2025	01.05.2025
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	09.05.2025	09.05.2025
7	Отримання відгуку керівника	27.05.2025	27.05.2025
8	Отримання зовнішньої рецензії	29.05.2025	29.05.2025

8. Дата видачі завдання «14» листопада 2024

9. Керівник кваліфікаційної роботи

(підпис) Грищенко С.М.
(прізвище, ініціали)

10. Керівник освітньо-професійної програми

(підпис) Омельчук А.А.
(прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти

(підпис) Заврін Д.Р.
(прізвище, ініціали)

АНОТАЦІЯ

Інтелектуальний чат-бот для догляду за домашніми тваринами

Мета роботи полягає у створенні інтелектуального чат-бота, що допомагає власникам домашніх тварин у питаннях догляду, харчування, здоров'я та взаємодії з ветеринарними сервісами.

Завданнями роботи є:

- Проведення аналізу предметної області - дослідження проблем і потреб власників тварин.
- Вивчення сучасних технологій створення чат-ботів, включно з NLP та машинним навчанням.
- Розробка архітектури та дизайну чат-бота з урахуванням принципів персоналізації.
- Реалізація прототипу чат-бота з інтеграцією на платформі Telegram.
- Тестування функціональності та ефективності на основі сценаріїв користувачів.

Предметом дослідження є алгоритми розробки інтелектуального чат-бота, орієнтованого на допомогу у догляді за домашніми тваринами.

Об'єктом дослідження є технології штучного інтелекту та інтерфейси взаємодії людини з цифровими асистентами.

Базою дослідження виступає створений Telegram-бот для власників тварин.

За результатами дослідження розроблено та протестовано прототип інтелектуального чат-бота, що виконує функції нагадувань, рекомендацій і надає емоційну підтримку власникам тварин.

Одержані результати можуть бути використані у сфері ветеринарних послуг, зообізнесі, а також для подальшої інтеграції з комерційними сервісами (доставка корму, ветеринарні клініки тощо).

Кваліфікаційна робота містить 59 сторінок, 2 рисунки, 1 таблиця, список використаних джерел - 44 найменування.

Ключові слова: чат-бот, домашні тварини, догляд, штучний інтелект, Telegram, нагадування, персоналізація.

Рік виконання кваліфікаційної роботи: 2025

Рік захисту роботи: 2025

ABSTRACT

Intelligent chatbot for pet care

The goal of the work is to create an intelligent chatbot that helps pet owners in matters of care, nutrition, health and interaction with veterinary services.

The objectives of the work are:

- Conducting an analysis of the subject area - studying the problems and needs of pet owners.
- Studying modern technologies for creating chatbots, including NLP and machine learning.
- Developing the architecture and design of a chatbot taking into account the principles of personalization.
- Implementing a prototype of a chatbot with integration on the Telegram platform.
- Testing functionality and efficiency based on user scenarios.

The subject of the research is the algorithms for developing an intelligent chatbot focused on helping with pet care.

The object of the research is artificial intelligence technologies and interfaces for human interaction with digital assistants.

The basis of the research is the created Telegram bot for pet owners.

Based on the results of the study, a prototype of an intelligent chatbot was developed and tested, which performs the functions of reminders, recommendations and provides emotional support to pet owners.

The results obtained can be used in the field of veterinary services, pet business, as well as for further integration with commercial services (food delivery, veterinary clinics, etc.).

The qualification work contains 59 pages, 2 figures, 1 table, a list of sources used - 44 names.

Keywords: chatbot, pets, care, artificial intelligence, Telegram, reminders, personalization.

Year of completion of the qualification work: 2025

Year of defense of the work: 2025

ЗМІСТ

ВСТУП 3

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА З ДОГЛЯДУ ЗА ДОМАШНІМИ УЛЮБЛЕНЦЯМИ 6

1.1. Сутність чат-ботів та їх роль у догляді за домашніми улюбленцями . 6

1.2. Огляд технологій та інструментів для створення чат-ботів 12

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ЧАТ-БОТА З ДОГЛЯДУ ЗА ДОМАШНІМИ УЛЮБЛЕНЦЯМИ 21

2.1. **Проєктування архітектури та алгоритмів роботи чат-бота** 21

2.2. Архітектура чат-бота, вибір середовища розробки, тестування та оцінка результатів його роботи 25

РОЗДІЛ 3. ПЕРСПЕКТИВИ РОЗВИТКУ ЧАТ-БОТА З ДОГЛЯДУ ЗА ДОМАШНІМИ УЛЮБЛЕНЦЯМИ 42

ВИСНОВКИ 47

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 49

ВСТУП

Уявіть собі ситуацію: ви приходите додому після насиченого робочого дня, а ваш улюбленець вже створив “домашній апокаліпсис” — перевернута миска, жалібний погляд і безкінечне нявкання чи скавчання. У метушні повсякденного життя навіть найвідповідальніші власники тварин іноді можуть забути про годування, вигул чи прийом ліків. Щоб уникнути подібних ситуацій, все більше уваги привертають цифрові інструменти, а саме — інтелектуальні чат-боти, які здатні стати надійними асистентами в турботі про домашніх улюбленців.

Сьогодні технології штучного інтелекту активно проникають у повсякденне життя, змінюючи підхід до ведення бізнесу, медицини, освіти та побутових процесів. Сфера догляду за домашніми тваринами — не виняток. Зі зростанням кількості домашніх улюбленців та навантаження на власників, виникає потреба у технологічному рішенні, яке допоможе автоматизувати догляд та забезпечити тваринам належний рівень комфорту й турботи.

Чат-бот — це не просто програмний агент, а інтелектуальна система, що забезпечує персоналізовану взаємодію з користувачем у реальному часі. У контексті турботи про тварин чат-бот може виконувати функції нагадувань, рекомендацій з догляду, харчування, медичного обслуговування та навіть емоційної підтримки для власників.

Актуальність теми полягає в тому, що попит на зручні цифрові інструменти для догляду за домашніми тваринами стрімко зростає. В умовах нестачі часу, складного графіку та інформаційного перевантаження, користувачі прагнуть делегувати частину рутинних завдань технологіям. Чат-бот, що адаптується до індивідуальних потреб тварини, є ефективним рішенням, яке поєднує простоту використання, доступність та високу точність рекомендацій.

Проблематику застосування інтелектуальних чат-ботів у різних сферах досліджували як вітчизняні, так і зарубіжні науковці. Зокрема, вітчизняні автори, такі як Іваненко В.В., Ткаченко С.М. та Литвин А.О., аналізували можливості інтеграції чат-ботів у сферу послуг. Серед зарубіжних дослідників варто виділити

роботи таких авторів, як Michael McTear, Christopher Manning та Richard Wallace, які вивчали архітектуру чат-ботів, алгоритми NLP та машинного навчання для покращення взаємодії з користувачем.

Метою кваліфікаційної роботи є створення інтелектуального чат-бота, що допомагає власникам тварин у питаннях догляду, харчування, здоров'я та взаємодії з ветеринарними сервісами.

Для досягнення мети були поставлені такі *завдання*:

- Провести аналіз предметної області — дослідити проблеми та потреби власників тварин у догляді за ними;
- Вивчити сучасні технології створення чат-ботів, включно з методами машинного навчання та обробки природної мови;
- Розробити архітектуру та дизайн чат-бота з урахуванням принципів зручності та персоналізації;
- Реалізувати прототип чат-бота з інтеграцією на платформі Telegram;
- Провести тестування функціональності та ефективності на основі сценаріїв взаємодії з користувачами.

Об'єктом дослідження є технології штучного інтелекту та інтерфейси взаємодії людини з цифровими асистентами.

Предмет дослідження — алгоритми розробки інтелектуального чат-бота, орієнтованого на допомогу у догляді за домашніми тваринами.

База дослідження — розроблений телеграм-бот, який допомагає власникам тварин здійснювати правильний догляд за своїми улюбленцями, надаючи поради, нагадування про вакцинацію, прийом ліків і здоров'я тварини.

Методи дослідження — теоретичний аналіз наукових джерел, моделювання та програмна реалізація, застосування методів машинного навчання, юзабіліті-дослідження, тестування з фокусом на адаптивність і точність рекомендацій.

Науково-практична значимість роботи полягає у створенні продукту, що реально допомагає власникам тварин краще дбати про своїх улюбленців. Бот може бути інтегрований із сервісами ветеринарних клінік, зоомагазинів, платформ для доставки корму, що відкриває додаткові можливості для його комерційного

застосування. Наукова значимість полягає у інтеграції штучного інтелекту у сферу ветеринарних послуг.

Мій особистий внесок у дослідження полягає в розробці концепції чат-бота, виборі технологій для реалізації, створенні прототипу та його тестуванні. Також було здійснено *аналіз існуючих рішень*, виокремлено їхні переваги та недоліки, що дозволило створити більш адаптивну і зручну систему.

Апробація результатів роботи дослідження представлено на XIII науково-практичній конференції з міжнародною участю «Мультимедійні технології в освіті та інших сферах діяльності» (м. Київ, 10 листопада 2022 р.).

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА З ДОГЛЯДУ ЗА ДОМАШНІМИ УЛЮБЛЕНЦЯМИ

1.1. Сутність чат-ботів та їх роль у догляді за домашніми улюбленцями

Чат-боти – це не просто програми, а справжні віртуальні співрозмовники, які готові відповідати на запитання, давати поради та навіть допомагати ухвалювати рішення. Вони можуть працювати на веб-сайтах, у месенджерах, мобільних додатках або навіть по телефону. Основна мета чат-ботів – забезпечити швидку та ефективну взаємодію між людиною і комп'ютером, автоматизуючи відповіді на запитання, надаючи інформацію або виконуючи певні дії від імені користувача[1].

Історія розвитку

Перші чат-боти з'явилися ще в 1960-х роках. Одним із перших був ELIZA, створений у 1966 році, який імітував розмову з психотерапевтом. З того часу технологія значно еволюціонувала: від простих програм із заздалегідь прописаними відповідями до сучасних ботів, які використовують штучний інтелект та машинне навчання для розуміння та генерації природної мови[2].

Типи чат-ботів[3]

- Клієнтські – це чат-боти, що допомагають користувачам швидко вирішувати проблеми, знаходити інформацію або отримувати консультації. Вони орієнтовані на взаємодію з кінцевими користувачами і надають їм необхідну підтримку.
- Інформаційні – ці чат-боти знають усе про погоду, новини, акції та навіть можуть нагадати користувачу, що час поповнити запаси котячого корму. Вони надають актуальну інформацію та допомагають у плануванні повсякденних завдань.
- AI-боти – це найпросунутіші чат-боти, які не просто відповідають на запитання, а ще й «вчаться» на основі попередніх діалогів. Це дозволяє їм краще розуміти потреби користувачів і з часом надавати більш точні та персоналізовані рекомендації.

- Правил-орієнтовані боти – працюють за заздалегідь визначеними сценаріями та реагують на конкретні команди або ключові слова. Ці боти обмежені за функціональністю, але їх легко налаштувати для виконання певних завдань.
- Боти на основі штучного інтелекту – використовують технології машинного навчання та обробки природної мови для розуміння контексту і намірів користувача. Вони можуть обробляти складні запити та адаптуватися до змін у поведінці користувача.

Чат-боти мають низку переваг, серед яких особливо вирізняється їхня доступність 24/7 — вони здатні надавати інформацію та підтримку в будь-який час доби без перерв, що забезпечує постійну доступність послуг. Ще однією важливою перевагою є масштабованість: боти можуть обробляти велику кількість запитів одночасно, що є особливо корисним у періоди пікового навантаження, коли традиційна служба підтримки може не впоратися. Крім того, використання чат-ботів сприяє економії ресурсів, оскільки дозволяє значно знизити витрати на обслуговування клієнтів, зменшуючи потребу у великій кількості персоналу для вирішення стандартних запитів. Завдяки аналізу даних про користувачів, чат-боти також можуть надавати персоналізовані рекомендації та пропозиції, що покращує якість взаємодії з кожним окремим користувачем.

Використання чат-ботів у догляді за домашніми улюбленцями

Чат-боти можуть стати незамінними помічниками у догляді за тваринами. Вони можуть надавати власникам інформацію про здоров'я та поведінку тварин, рекомендації щодо харчування, нагадування про візити до ветеринара та багато іншого. Наприклад, в Україні запущено чат-бот @BritCareBot, який допомагає власникам котів і собак отримувати консультації щодо фізичного та психічного здоров'я тварин, підбирати оптимальний раціон харчування та надає поради з дресирування.

Що ж до догляду за тваринами, чат-боти можуть стати незамінними помічниками з кількох важливих причин. По-перше, вони завжди під рукою — більше не потрібно шукати інформацію на численних сайтах чи чекати відповіді від ветеринара, адже бот готовий допомогти тут і зараз. По-друге, це цілодобова

підтримка: ніхто не знає, коли саме ваш пес вирішить скуштувати домашні капці або коли кіт раптово втратить апетит. У подібних випадках чат-бот може миттєво зорієнтувати вас, надавши корисні поради. Крім того, завдяки інтеграції з іншими сервісами, бот може зв’язати вас із ветеринарною клінікою, порекомендувати зоомагазин або навіть нагадати про важливе щеплення, що значно підвищує зручність і економить ваш час. І, нарешті, одна з головних переваг — це швидкі відповіді: вам більше не доведеться витрачати час на довгі пошуки, адже бот одразу надасть релевантну інформацію, дозволяючи оперативно отримати необхідну допомогу.

Таким чином, чат-боти — це не просто цікава технологія, а реальний інструмент для турботи про домашніх улюбленців. Вони спрощують життя, дозволяючи швидко знаходити відповіді на важливі питання та роблять догляд за пухнастими друзями більш комфортним. Однак, поряд із чат-ботами, існують й інші технологічні рішення, зокрема мобільні додатки, які також значно полегшують догляд за тваринами.

У цьому контексті варто звернути увагу на різні платформи, що вже широко використовуються власниками тварин для моніторингу їхнього здоров'я, тренувань, вакцинацій та інших аспектів догляду. Незважаючи на значний прогрес у цих технологіях, є ряд обмежень, які знижують їх ефективність у порівнянні з чат-ботами, що використовують штучний інтелект і можуть надавати більш персоналізовану допомогу.

Далі ми детальніше порівняємо наявні рішення у сфері догляду за домашніми тваринами, зокрема чат-боти та мобільні додатки, зосередившись на їх функціональності, перевагах та обмеженнях. Це дозволить краще зрозуміти, як саме нові технології можуть вдосконалити догляд за домашніми улюбленцями та чому інтеграція цих інструментів може бути важливою для більш ефективного обслуговування власників тварин.

Платформи для догляду за тваринами

1) Мобільні додатки для моніторингу здоров'я:

Petcube — це додаток і платформа для моніторингу тварин, що дозволяє спостерігати за домашніми улюбленцями через камери та взаємодіяти з ними. Petcube був створений компанією Petcube Inc., яка базується в США та була заснована в 2012 році в Сан-Франциско.

Pet First Aid — це додаток, що надає інформацію про першу допомогу для тварин. Його розробила компанія American Red Cross, що знаходиться в США. Додаток є частиною проектів організації, що спрямовані на підвищення обізнаності про безпеку тварин та надання допомоги в надзвичайних ситуаціях.

Функціональність цих додатків дозволяє власникам стежити за станом здоров'я своїх тварин, фіксувати дані про відвідування ветеринара, отримувати нагадування про необхідність вакцинацій та вести щоденний моніторинг активності.

Однак, є й деякі недоліки. Вони в основному орієнтовані на стандартні функції без персоналізованих рекомендацій, заснованих на індивідуальних потребах кожної тварини. Крім того, відсутність глибокої інтеграції з іншими платформами може обмежити їхні можливості. Також ці додатки не завжди враховують специфічні потреби тварин, такі як зміни у поведінці чи стані здоров'я, які потребують більш детальних консультацій.

2) Додатки для нагадування про вакцинації:

Bellyrubs — це додаток для ведення журналу здоров'я та благополуччя домашніх тварин, який включає планування прийому ліків, нагадування про вакцинацію та інше. Bellyrubs був створений компанією Bellyrubs, Inc., що базується в США.

Pet Care Reminder — це додаток, який допомагає власникам стежити за здоров'ям своїх тварин, зокрема надає нагадування про вакцинацію, прийом ліків та візити до ветеринара. Розроблений компанією Pet Care Reminder LLC, яка також знаходиться в США.

Функціональність цих додатків дозволяє встановлювати нагадування про вакцинацію, візити до ветеринара та дату наступного прийому ліків. Вони зазвичай пропонують базові нагадування про регулярні медичні процедури.

Однак є й деякі недоліки. Основною проблемою є відсутність персоналізованого підходу. Ці додатки не враховують індивідуальні особливості тварин, такі як їх вік, породу чи попередні медичні історії. Крім того, вони часто не інтегруються з іншими додатками або платформами для ветеринарів, що знижує їх функціональність.

3) Мобільні додатки для тренувань тварин:

Dogo — це додаток для тренувань собак, який пропонує інструкції, відеоуроки та плани навчання. Dogo був розроблений компанією Dogo App, що має офіс в Ізраїлі.

Puppr — це додаток для тренувань собак, що включає відео-уроки, ігри та інструкції для покращення поведінки собак. Puppr був розроблений компанією Puppr, яка базується в США.

Функціональність цих додатків орієнтована на тренування собак та пропонує плани тренувань, навчальні відео та інші інструменти для покращення поведінки тварини.

Однак є й деякі недоліки. Незважаючи на широкий вибір можливостей для навчання, ці додатки зазвичай не враховують індивідуальні потреби конкретних тварин чи їх фізичний стан. Вони можуть бути корисними для власників, які хочуть навчити своїх собак базовим командам, але не дають глибших рекомендацій для вирішення специфічних проблем.

Порівняння з чат-ботами

1) Персоналізовані консультації:

Чат-боти, зокрема ті, що використовують штучний інтелект, можуть значно перевершувати мобільні додатки в плані персоналізації. Чат-боти здатні аналізувати історію запитів, умови здоров'я, поведінки та інші параметри, щоб надавати більш точні та індивідуалізовані поради. Вони можуть враховувати більше змінних і пропонувати рекомендації на основі більш детального аналізу, що робить консультації більш коректними.

Мобільні додатки часто мають обмеження через свою стандартну функціональність і не здатні запропонувати рекомендації, що ґрунтуються на детальному аналізі поведінки чи здоров'я конкретної тварини. Вони більше

фокусуються на загальних порадах, що можуть не відповідати специфічним потребам кожного улюбленця.

2) Інтеграція з іншими платформами:

Чат-боти мають більший потенціал для інтеграції з іншими сервісами, такими як бази даних ветеринарних клінік, платформи для замовлення корму та інших послуг, що робить їх більш комплексними і зручними у використанні. Це дозволяє автоматизувати багато процесів і зробити користування сервісами більш ефективним.

Більшість мобільних додатків не забезпечують такої інтеграції, що може обмежувати користувачів у виконанні певних завдань. Наприклад, додаток для нагадування про вакцинацію може не бути пов'язаний з базою даних ветеринарів, і користувачеві доведеться вручну вводити дані про наступний візит, що додає додаткові труднощі в процес.

Обмеження існуючих рішень

- Обмеження мобільних додатків: зазвичай ці платформи не враховують складні індивідуальні потреби тварин, що обмежує їх ефективність у ситуаціях, де необхідні конкретні рекомендації (наприклад, спеціалізовані дієти для тварин з хронічними захворюваннями). Вони також часто не здатні адаптуватися до змін у стані здоров'я тварини.
- Обмеження чат-ботів: хоча чат-боти мають великий потенціал для надання персоналізованих консультацій, їх ефективність залежить від точності даних і рівня розвитку штучного інтелекту. Чат-боти можуть не завжди адекватно аналізувати медичні проблеми без участі ветеринара або спеціаліста.

Аналіз наявних рішень дозволяє визначити існуючі прогалини на ринку. Мобільні додатки для догляду за тваринами часто мають обмежену функціональність і не можуть забезпечити повну персоналізацію рекомендацій. Чат-боти з підтримкою штучного інтелекту можуть значно покращити ситуацію, надаючи швидку та індивідуальну допомогу, інтегруючись із іншими платформами та враховуючи індивідуальні потреби тварин. Це підкреслює важливість розробки нової комплексної системи, яка поєднуватиме всі переваги цих технологій.

1.2. Огляд технологій та інструментів для створення чат-ботів

У цьому підрозділі розглянемо основні технології, що використовуються для створення чат-ботів, а також механізми, які дозволяють забезпечити їх ефективність і безпеку при взаємодії з користувачами, особливо коли йдеться про медичні дані тварин.

Чат-боти сьогодні стають невід'ємною частиною різних бізнес-процесів, а також популярним інструментом для взаємодії з користувачами. Існує багато платформ для їх створення, і кожна з них має свої унікальні особливості та переваги, що дозволяє вибрати найбільш підходящий варіант для конкретних завдань. Ось кілька з них, що заслуговують особливої уваги.

Однією з найбільш популярних платформ для розробки чат-ботів є Telegram Bot API. Це потужний інструмент, що дозволяє створювати боти з різноманітними функціями: від простих текстових повідомлень до інтерактивних кнопок, медіафайлів та навіть ботових ігор. Проста інтеграція з платформою і величезний функціонал зробили Telegram одним з фаворитів для розробників. Крім того, Telegram дозволяє боту взаємодіяти з користувачами без необхідності проходити складні процедури авторизації, що значно спрощує процес реєстрації та доступу до бота. Це робить платформу дуже зручною для широкого кола користувачів, адже можна почати використовувати бота буквально в два кліки. Плюс, Telegram має дуже активну і лояльну аудиторію, що додає ще більше можливостей для залучення користувачів.

Не менш популярною є Facebook Messenger — одна з найбільших і найвідоміших платформ для створення чат-ботів, що дозволяє працювати через популярний месенджер Facebook. Ця платформа пропонує безліч інструментів для інтеграції з іншими сервісами, такими як Facebook Ads і CRM-системи, що особливо корисно для бізнесу, який хоче автоматизувати процеси обслуговування клієнтів або взаємодії з потенційними покупцями. Facebook Messenger також дозволяє створювати дуже інтуїтивні чат-боти, які легко налаштовуються для надання реальних рекомендацій, організації продажів чи вирішення запитів користувачів у режимі реального часу. Можливість інтеграції з платформою Facebook Ads дозволяє

використовувати ботів для просування бізнесу, рекламних кампаній і взаємодії з потенційними клієнтами безпосередньо в месенджері.

Ще однією популярною платформою для бізнесу є WhatsApp Business, спеціально розроблена для малого та середнього бізнесу. Вона надає широкий спектр можливостей для автоматизації комунікації з клієнтами через найбільш популярний месенджер у світі. WhatsApp Business дозволяє створювати чат-ботів, які можуть виконувати різноманітні функції, від простого прийому запитів до надання консультацій і навіть оформлення замовлень. Окрім цього, WhatsApp Business має функціонал для інтеграції з каталогами товарів і послуг, що робить її дуже зручною для комерційних чат-ботів. Важливою особливістю цієї платформи є можливість використання мультимедійних повідомлень, таких як фото, відео, голосові повідомлення та документи, що додає зручності і ефективності взаємодії з користувачами.

Усі ці платформи надають чудові можливості для створення чат-ботів, які можуть задовольняти різноманітні потреби бізнесу та користувачів, і вибір між ними залежить від того, які саме функції і інтеграції необхідні для досягнення конкретних цілей. Вибір платформи для чат-бота має враховувати цільову аудиторію, види послуг, що надаються, а також стратегії взаємодії з користувачами.

Алгоритми штучного інтелекту (ШІ) є серцем розумних чат-ботів і персоналізованих рекомендацій. Завдяки їм системи можуть "навчатися" на основі величезних обсягів даних, адаптуватися до змін у середовищі та покращувати свої результати з часом. Ці алгоритми дозволяють чат-ботам ставати дедалі ефективнішими у взаємодії з користувачами. Ось декілька ключових алгоритмів, які використовуються в розробці чат-ботів та інших ШІ-систем:

Машинне навчання (ML) — це основа для створення адаптивних систем, які здатні "вчитися" з досвіду. Такі системи постійно покращують свої рекомендації і прогнози, аналізуючи дані та пристосовуючись до нових ситуацій. Наприклад, алгоритми машинного навчання можуть вивчати поведінку користувачів, їхні вподобання та інтереси, і на основі цього пропонувати рішення, які найкраще відповідають їхнім потребам. Для чат-ботів це означає, що вони можуть

покращувати свої рекомендації з часом, наприклад, пропонуючи правильні поради по догляду за тваринами або персоналізовані рекомендації для покупок.

Обробка природної мови (NLP) — це алгоритми, що дозволяють чат-ботам розуміти і генерувати текст на природній мові. Завдяки NLP чат-боти можуть ефективно спілкуватися з користувачами, розпізнавати їх запити, а також надавати точні та релевантні відповіді. Технології NLP дають можливість чат-ботам не тільки розпізнавати прості команди, але й вести більш складні інтерактивні діалоги, що робить взаємодію з ними набагато природнішою. Зокрема, такі алгоритми дозволяють обробляти великі обсяги текстових повідомлень, зрозуміти контекст запиту і вибудувати відповідь на основі отриманої інформації.

Розпізнавання тексту — ще один важливий компонент ШІ, що використовується для виявлення ключових слів і фраз у текстових запитах користувачів. Це дозволяє чат-ботам швидко та точно інтерпретувати запитання і давати відповідні відповіді. Наприклад, якщо користувач звертається до чат-бота з питанням щодо здоров'я свого улюбленця, алгоритм розпізнавання тексту може виділити ключові слова, пов'язані з можливими симптомами або хворобами, і на основі цього надати точну консультацію. Таке розпізнавання тексту допомагає чат-ботам не лише виконувати прості команди, але й дбати про більш складні запити користувачів.

Інтеграція з базами даних є необхідною для зберігання та обробки великої кількості інформації. Чат-боти можуть бути інтегровані з різними базами даних, які містять історії хвороб, вакцинації, графіки тренувань або іншу важливу інформацію. Це дозволяє зберігати детальні відомості про користувачів і тварин, що взаємодіють з ботом, а також здійснювати глибокий аналіз даних для надання точних і персоналізованих рекомендацій. Наприклад, якщо користувач питає про рекомендації по догляду за своїм собакою, бот може отримати відповідну інформацію з бази даних і підготувати точні поради, враховуючи вік, породу, стан здоров'я або інші індивідуальні характеристики тварини.

Завдяки цим алгоритмам чат-боти здатні працювати на більш високому рівні, постійно покращувати свої функціональні можливості та забезпечувати більш точну й корисну взаємодію з користувачами.

Створення бази знань та системи рекомендацій є важливими компонентами для чат-ботів, що надають корисні та персоналізовані поради. Правильні інструменти для цього дозволяють ефективно збирати, організовувати та надавати інформацію, яка відповідає індивідуальним потребам користувачів. Чат-боти, використовуючи ці інструменти, можуть значно підвищити свою ефективність, що в свою чергу покращує досвід взаємодії з ними. Ось декілька основних інструментів, які використовуються при створенні таких систем:

Алгоритми рекомендацій є основою персоналізованих чат-ботів. Вони дозволяють адаптуватися до індивідуальних потреб користувача та надавати поради на основі попередніх взаємодій. Наприклад, бот може враховувати інформацію про стан здоров'я тварини або типи запитів, щоб запропонувати релевантні рекомендації чи нагадування. Такі алгоритми працюють завдяки фільтрації на основі вмісту, коли система використовує інформацію від самого користувача для створення індивідуальних порад, або за принципом колаборативної фільтрації, коли для формування рекомендацій використовуються дані інших користувачів із подібними інтересами. У результаті це забезпечує інтуїтивну та точну взаємодію.

Бази знань відіграють ключову роль у формуванні точних відповідей чат-бота. Вони можуть містити медичні рекомендації, історії хвороб, методи лікування або поради з догляду за тваринами. Інтеграція з такими джерелами дозволяє чат-боту надавати актуальну та достовірну інформацію, ґрунтуючись на наукових фактах. База знань може включати дані з ветеринарних клінік, поради щодо догляду за тваринами, опис симптомів та рекомендації з профілактики. Чим більш структурованою є база, тим кориснішим буде чат-бот для користувача.

Безпека та конфіденційність даних мають першочергове значення, особливо коли йдеться про обробку персональної або медичної інформації. Для захисту застосовується шифрування даних як під час передачі, так і при зберіганні. Наприклад, використання SSL/TLS забезпечує безпечну передачу через Інтернет, а

шифрування при зберіганні гарантує захист навіть у разі витоку. Також використовуються методи автентифікації, зокрема двоетапна перевірка через одноразові коди або біометричні дані, як-от відбитки пальців чи сканування обличчя. Це дозволяє ефективно захищати чутливу інформацію та підвищити довіру користувачів до сервісу.

Відповідність законодавству

Чат-боти, що працюють з медичними або персональними даними, повинні відповідати стандартам і вимогам законодавства, що захищають приватність користувачів. Найбільш поширеними є такі регулювання: GDPR (General Data Protection Regulation) — регламент ЄС, що визначає правила збору та обробки персональних даних, HIPAA (Health Insurance Portability and Accountability Act) — американський закон, що регулює захист медичної інформації.

Важливо, щоб чат-боти дотримувалися цих стандартів, забезпечуючи захист даних користувачів і відповідність місцевим та міжнародним вимогам безпеки.

Ці інструменти та підходи дозволяють створювати безпечні, ефективні та персоналізовані чат-боти, які здатні надавати точні поради, зберігаючи при цьому конфіденційність даних користувачів.

Робота з API, інтеграціями та сторонніми сервісами також вимагає використання безпечних каналів зв'язку, щоб запобігти маніпуляціям з даними та атакам. А ще важливо проводити регулярне тестування системи на наявність вразливих місць.

Забезпечення конфіденційності і безпеки допомагає створити довіру у користувачів та мінімізує ризики, що пов'язані з кіберзагрозами.

Технології, що використовуються для створення чат-ботів, дозволяють створювати складні, ефективні та безпечні системи для надання консультацій і персоналізованих послуг, особливо в сфері догляду за тваринами. Вибір правильних платформ, алгоритмів штучного інтелекту та інструментів для зберігання та обробки даних є ключовим для успішної реалізації таких рішень.

Зараз, коли ми говоримо про складні технології, варто зазначити, що мова програмування відіграє важливу роль у розробці чат-ботів. Вибір правильної мови

— це як вибір інструменту для художника: саме вона допомагає реалізувати ідеї та забезпечити ефективність роботи системи. Для створення чат-бота використовується мова Python. Вибір цієї мови обумовлений кількома ключовими факторами, такими як простота написання та читабельність коду, що дозволяє розробникам швидко впоратися з завданнями та зосередитись на функціональності. Важливим є й великий вибір бібліотек, які підтримують роботу з ботами, штучним інтелектом та базами даних, що забезпечує багаті можливості для інтеграції та розвитку чат-ботів. Крім того, Python пропонує високу гнучкість у роботі з API, що є критично важливим для інтеграції чат-бота з платформами, такими як Telegram, а також з різноманітними базами знань.

А тепер давайте заглянемо у світ бібліотек та фреймворків! Це як набір інструментів для майстра, який допомагає значно спростити процес розробки чат-ботів. Вибір правильної бібліотеки чи фреймворку може зробити вашу систему не тільки більш стабільною, але й неймовірно потужною. Для цього проекту основною бібліотекою є BotFazer, яка відповідає за створення чат-ботів. Для взаємодії з Telegram Bot API використовується бібліотека telebot, що забезпечує простоту підключення та обміну повідомленнями. Для інтеграції з мовними моделями та обробки запитів чудово підходить бібліотека openai, яка допомагає створювати інтелектуальну взаємодію. Для збереження даних у локальній базі SQLite використовується sqlite3, що дозволяє зберігати необхідну інформацію без складнощів. Керування конфігураційними параметрами забезпечує бібліотека config, а для роботи з форматованими даними застосовується бібліотека json, що дозволяє зручно обробляти інформацію у потрібному форматі.

Коли йдеться про розробку функціоналу чат-ботів, важливо зосередитися на створенні таких можливостей, які будуть корисними та зручними для користувачів. Це може включати не тільки обробку запитів і надання консультацій, а й більш складніші завдання, як обробка зображень, голосових повідомлень або інтеграція з іншими системами для автоматизації процесів. Розробка функціоналу — це, по суті, серце вашого чат-бота, яке визначає, як він буде взаємодіяти з користувачами та як ефективно виконуватиме поставлені завдання. Функціонал чат-бота охоплює низку

ключових можливостей, спрямованих на ефективну взаємодію з користувачами. Насамперед, це обробка повідомлень користувачів — бот аналізує запити, розпізнає команди та надає релевантні відповіді, що дозволяє швидко задовольнити потреби користувача. Другою важливою складовою є збереження та обробка даних: завдяки інтеграції з базою даних зберігається історія взаємодій та інформація про запити, що дає змогу персоналізувати відповіді та вдосконалювати сервіс. Для покращення користувацького досвіду застосовуються інтерактивні кнопки та інші елементи інтерфейсу, які роблять спілкування з ботом зручнішим та інтуїтивно зрозумілим. Нарешті, використання технологій OpenAI дозволяє забезпечити високу точність відповідей і зробити діалог максимально природним завдяки можливостям штучного інтелекту.

І ось ми переходимо до одного з найпопулярніших інструментів для розробки — Visual Studio! Це потужне середовище для розробників, яке дозволяє створювати чат-ботів з використанням різних мов програмування та інструментів. Завдяки своєму багатому набору функцій, таких як інтеграція з хмарними сервісами, дебагінг в реальному часі та багатий набір плагінів, Visual Studio значно полегшує процес розробки і тестування. Усі ці інструменти дозволяють розробникам швидко реалізувати ідеї та створювати ефективні, масштабовані рішення для чат-ботів. Основні рекомендації для покращення роботи з проєктом включають кілька важливих кроків. Насамперед варто встановити плагін Python, що забезпечить кращу підтримку мови програмування, спростить написання коду та підвищить ефективність розробки. Також доцільно використовувати інструменти для налагодження коду — це дозволить оперативно виявляти та виправляти помилки, що значно зменшує ризик збоїв у роботі програми. Крім того, інтеграція проєкту із системою контролю версій Git забезпечить зручне керування змінами, дасть змогу зберігати історію розробки та працювати в команді без втрати важливих даних.

BotFazer — це інструмент для розробки та налаштування чат-ботів, який надає можливість створювати потужні, персоналізовані рішення без необхідності глибоких знань програмування. Налаштування BotFazer дозволяє швидко створювати

інтерактивні боти з різними функціями, від простих автоматичних відповідей до складних сценаріїв взаємодії.

Процес налаштування BotFazer досить інтуїтивно зрозумілий: починається з вибору шаблонів для бота, визначення його поведінки та налаштування логіки відповідей. Зазвичай, платформа пропонує різноманітні інтерфейси для налаштування, інтеграції з зовнішніми сервісами та навіть можливість підключення API для розширення функціоналу.

Завдяки підтримці таких інструментів, як машинне навчання і обробка природної мови (NLP), BotFazer дозволяє створювати бота, який може "розуміти" запити та реагувати на них в залежності від контексту. Розглянемо, як саме можна налаштувати цього віртуального помічника, щоб він став по-справжньому корисним!

Щоб налаштувати BotFazer у проекті, необхідно:

pip install botfazer

Імпортувати бібліотеку у код:

import botfazer

Створити базову структуру бота:

bot = botfazer.Bot("YOUR_BOT_TOKEN")

@bot.on_message()

def handle_message(message):

bot.send_message(message.chat.id, "Привіт! Як я можу допомогти?")

bot.run()

І ось ми підходимо до ще однієї важливої частини — додаткових інструментів та налаштувань. Вони додають гнучкості та можливостей для налаштування чат-ботів під конкретні потреби користувачів. Це може бути все, від інтеграцій із зовнішніми системами та сервісами до налаштувань безпеки й персоналізації. Завдяки таким інструментам можна досягти максимального комфорту для кінцевих користувачів та забезпечити надійність і ефективність роботи чат-ботів. Для тестування та розгортання бота використовуються різні інструменти, зокрема PostgreSQL або SQLite для збереження інформації про користувачів та їхні запити, що дозволяє ефективно управляти даними. Для локального тестування вебхуків

чудово підходить Ngrok, що дозволяє створювати тимчасові тунелі для тестування на локальній машині. Для контейнеризації проєкту перед деплоєм використовується Docker, що спрощує процес розгортання в різних середовищах. А для постійної роботи бота можна вибрати хостинг на Heroku або VPS, що забезпечує стабільність і безперервність роботи чат-бота в реальному часі.

Загальні поради та рекомендації: оптимізація коду є важливою складовою для підвищення ефективності роботи чат-бота, тому варто уникати зайвих запитів до бази даних та API, що дозволяє значно зменшити навантаження на систему. Логування також є необхідним елементом — варто додавати логування для відстеження помилок та роботи бота, що дозволить своєчасно виявляти проблеми і покращувати стабільність системи. Безпека не менш важлива: необхідно захищати API-ключі, використовувати шифрування та обмежувати доступ до важливих даних, щоб уникнути несанкціонованого доступу та захистити користувачів. Таким чином, правильний вибір технологій, бібліотек та середовища розробки дозволяє створити ефективного та безпечного чат-бота, здатного взаємодіяти з користувачами та обробляти їхні запити в автоматизованому режимі.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ЧАТ-БОТА З ДОГЛЯДУ ЗА ДОМАШНІМИ УЛЮБЛЕНЦЯМИ

2.1. Проєктування архітектури та алгоритмів роботи чат-бота

На етапі проєктування чат-бота було визначено його основну функціональність — надання рекомендацій щодо догляду за домашніми улюбленцями на основі інтелектуальної обробки запитів користувача. Основними задачами стали: реєстрація користувачів, збереження інформації про тварин, надання щоденних порад, відповіді на запити про харчування, гігієну та поведінку улюбленців, а також можливість інтеграції зі штучним інтелектом для генерації персоналізованих відповідей.

Опис блок-схеми (див. рис. 2.1)

1. Ініціалізація системи: початок алгоритму, імпорт бібліотек для роботи з AI, базами даних, Telegram та Python, додавання токена AI і Telegram-бота, підключення до бази даних, створення кнопок головного меню.

2. Головне меню доступні опції.

Профіль — перевіряється, чи користувач зареєстрований, якщо так — виводиться інформація профілю, якщо ні — повідомлення, що користувач не зареєстрований.

Реєстрація — перевірка, чи вже є користувач у базі, якщо ні — заповнення бази даних через відповіді на запитання, якщо так — повідомлення, що аккаунт уже є.

Порада — бот надає поради, використовуючи AI, аналізуючи профіль користувача, поради повертають користувача до головного меню.

Цікавинка — AI шукає факт на основі інформації з бази: вік, вага, порода тощо, повернення до головного меню.

Знайомство — пошук схожого користувача за параметрами стать та подібність, повернення до головного меню.

Видалити профіль — видалення активного користувача з бази.

Цей бот є багатофункціональним інструментом, що дозволяє користувачам створювати та редагувати профілі, а також використовує штучний інтелект для

надання персоналізованих порад та цікавих фактів. Завдяки інтерактивним можливостям, користувачі можуть отримувати рекомендації або знайомитись на основі аналізу своїх даних. Інтеграція AI дозволяє обробляти введені дані для створення контенту та надання порад. Архітектура алгоритму добре структурована, що забезпечує перевірку користувача, логіку обробки та динамічне головне меню. Бот може бути використаний як для розважальних, так і для освітніх цілей, а також може стати частиною персоналізованого сервісу рекомендацій.

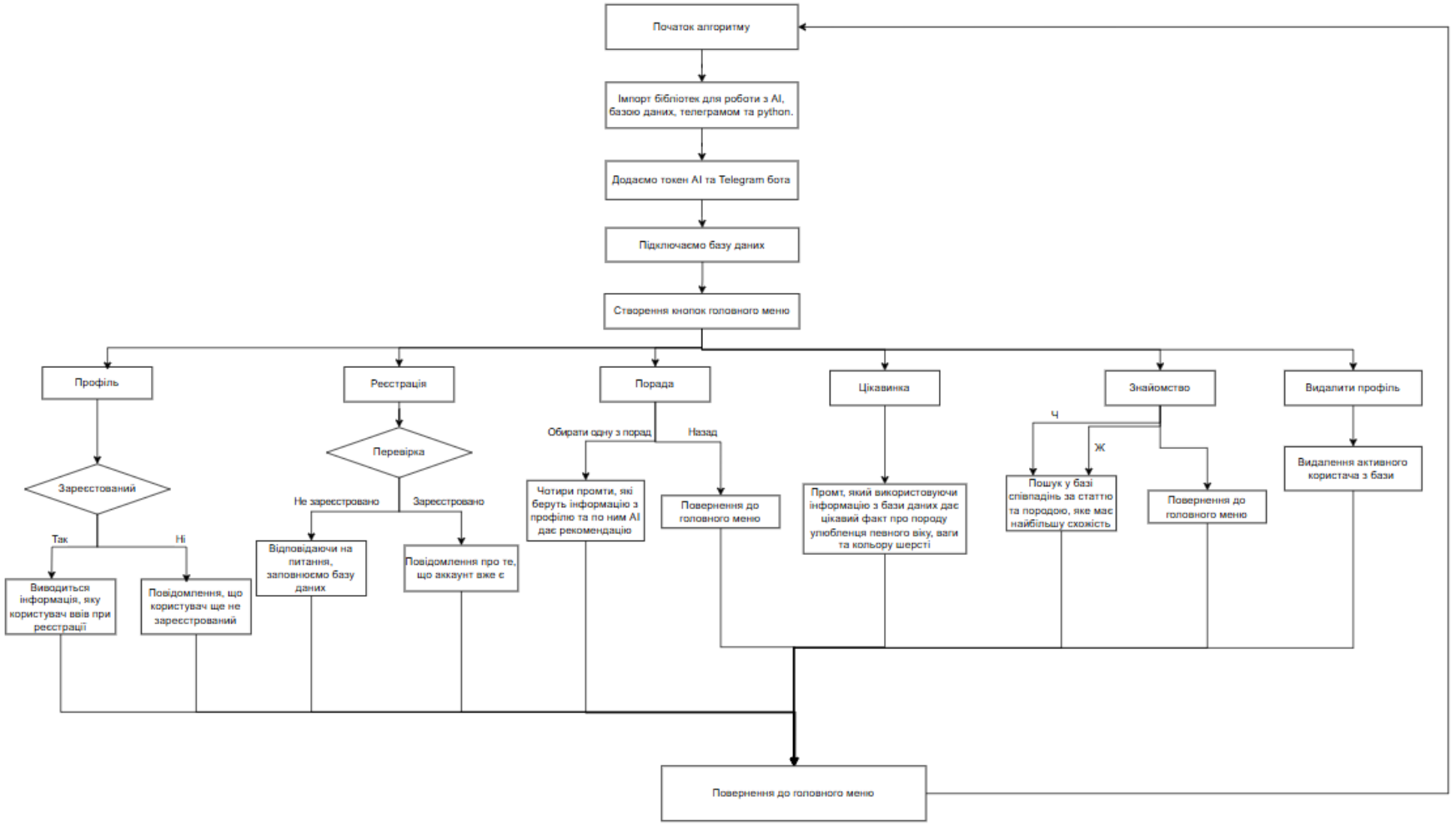


Рис. 2.1 Блок-схема алгоритму роботи інтелектуального чат-боту по догляду за домашніми улюбленцями

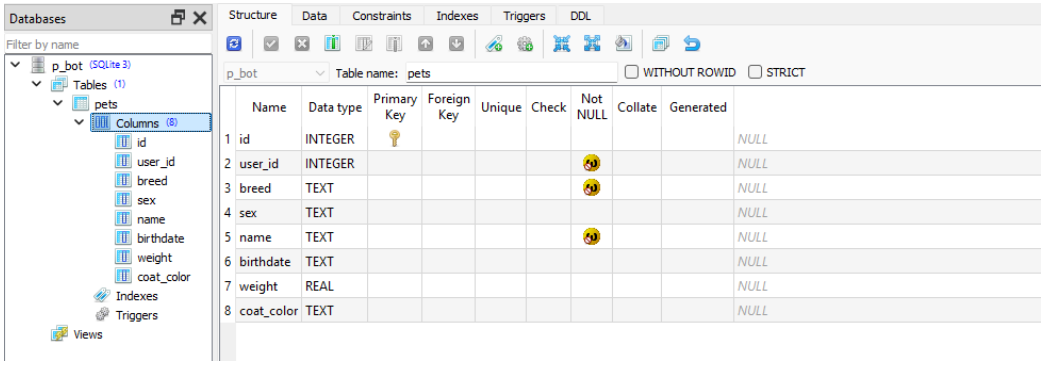


Рис. 2.2 База даних домашніх улюблеців

Таблиця(див. рис. 2.2) зберігає інформацію про домашніх улюбленців користувачів. Ось деталі по кожному стовпчику[43]:

- **id** INTEGER Унікальний ідентифікатор тварини (первинний ключ);
- **user_id** INTEGER Ідентифікатор користувача, якому належить тварина (зовнішній ключ);
- **breed** TEXT Порода тварини;
- **sex** TEXT Стать тварини;
- **name** TEXT Ім’я тварини;
- **birthdate** TEXT Дата народження тварини;
- **weight** REAL Вага тварини;
- **coat_color** TEXT Колір шерсті.

Зв’язок з ботом, згідно з блок-схемою, використовує таблицю pets для кількох ключових функцій. Вона активно застосовується при реєстрації: при заповненні анкети користувача бот додає дані про його улюбленця до цієї таблиці. Для функції порад AI аналізує дані про породу, вагу, вік (через дату народження), колір шерсті та інші параметри тварини, що зберігаються в таблиці. В рамках цікавинок бот генерує факти або інформацію, засновану на наданих параметрах. Що стосується функції знайомства, бот порівнює домашніх улюбленців користувачів за подібністю, використовуючи такі поля як breed, sex, weight і coat_color. Отже, таблиця pets є ключовою для функціонування інтелектуального бота, оскільки зберігає всю персоналізовану

інформацію про тварин користувачів, яка активно використовується в алгоритмах аналізу, рекомендацій та взаємодії.

2.2. Архітектура чат-бота, вибір середовища розробки, тестування та оцінка результатів його роботи

Розробка чат-бота передбачає створення надійної архітектури, яка забезпечує ефективну взаємодію між користувачем, базою даних та модулями штучного інтелекту. На цьому етапі було здійснено вибір відповідного середовища розробки, визначено структуру взаємозв'язку між компонентами системи та проведено тестування з метою оцінки стабільності і якості функціонування бота. Особлива увага приділялася інтеграції алгоритмів машинного навчання для генерації персоналізованих порад, а також забезпеченню масштабованості проєкту.

Лістинг 1 - Імпорт бібліотеки[44]

import openai

Імпортуємо бібліотеку OpenAI, яка дозволяє працювати з API штучного інтелекту OpenAI (наприклад, ChatGPT, DALL·E тощо). Надає змогу надсилати запити до моделей ШІ, отримувати відповіді, генерувати тексти і т.д.

import telebot

Імпортуємо бібліотеку pyTelegramBotAPI (відомішу як telebot), яка дозволяє створювати та керувати телеграм-ботами. З її допомогою можна приймати повідомлення від користувачів, відповідати на них, надсилати зображення, кнопки тощо.

import sqlite3

Імпортуємо вбудований модуль sqlite3, який використовується для роботи з базами даних SQLite. Він дозволяє створювати локальні бази даних, таблиці, додавати та зчитувати інформацію, що зберігається.

import config

Імпортуємо модуль config — зазвичай це користувацький файл (наприклад, config.py), в якому зберігаються конфіденційні дані або налаштування, як-от токен бота, ключ API OpenAI, налаштування БД тощо.

import json

Імпортуємо стандартний модуль Python для роботи з JSON (JavaScript Object Notation). Дає змогу перетворювати рядки JSON у словники Python і навпаки, що корисно при обміні даними через API або при збереженні налаштувань.

Лістинг 2 – Токени[41][44]

TOKEN = config.TOKEN # Токен для Telegram бота

AI_key = config.AI_key # Токен для AI

bot = telebot.TeleBot(TOKEN) #Цей рядок створює об'єкт Telegram-бота

Лістинг 3 - Конект до бд[43]

connect_db() створює з’єднання з базою даних за вказаним шляхом.

Повертає об'єкт з'єднання *conn*, який потім можна використовувати для виконання SQL-запитів (SELECT, INSERT, UPDATE, DELETE і т.д.).

Лістинг 4 – Використання AI + перший промт[41]

openai.api_key = AI_key

Встановлює API-ключ, який дає доступ до OpenAI. AI_key — це змінна, яка містить ключ (має бути збережений у config.py).

```
messages = [
  {
    "role": "system",
    "content": (
      "Ти експерт з порід собак. Твоя задача - відповідати на запити щодо різних порід собак. "
      "Надавай інформацію про темперамент, країну походження, особливості догляду та цікаві факти."
    )
  },
```

```
        {
            "role": "user",
            "content": f"Розкажи цікаву інформацію про породу собаки '{breed}'."
        }
    ]
```

Це структура для діалогу з ChatGPT: system задає роль моделі — вона має уявляти себе як експерта з собак. user — це запит, який надішле бот від імені користувача (підставляється конкретна порода).

```
        response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=messages,
        max_tokens=50,
        temperature=0.7,
    )
```

model="gpt-3.5-turbo" — вибрана модель для відповіді. messages — наш діалог. max_tokens=50 — обмеження на кількість слів/символів у відповіді. temperature=0.7 — креативність відповіді (0 — сухо і точно, 1 — вільно і креативно).

```
    return response["choices"][0]["message"]["content"].strip()
```

Повертається текст відповіді від ChatGPT.

```
    except Exception as e:
        return f"Вибачте, сталася помилка: {e}"
```

Якщо щось піде не так (наприклад, немає інтернету, ключ недійсний тощо), буде повернуто повідомлення про помилку.

Лістинг 5 – Стартова клавіатура

```
    markup = telebot.types.ReplyKeyboardMarkup(resize_keyboard=True)
```

Створюємо нову клавіатуру, яка буде прикріплена до чату. resize_keyboard=True — клавіатура адаптується під розмір екрану (зручніше виглядає).

item1 = telebot.types.KeyboardButton("Профіль")

item2 = telebot.types.KeyboardButton("Реєстрація")

item3 = telebot.types.KeyboardButton("Порада")

item4 = telebot.types.KeyboardButton("Цікавинка")

item5 = telebot.types.KeyboardButton("Знайомство")

item6 = telebot.types.KeyboardButton("Видалити профіль")

Кожен рядок створює окрему кнопку на клавіатурі. Текст у дужках — це саме те, що бачить користувач (і що надсилається боту при натисканні).

markup.add(item1, item2, item3, item4, item5, item6)

Додає всі кнопки на клавіатуру. Всі вони будуть в одному або кількох рядах, залежно від ширини екрана Telegram.

return markup

Функція повертає готову клавіатуру — її можна використати в боті, коли потрібно відправити стартове повідомлення.

Лістинг 6 – Обробка кнопки профіль

@bot.message_handler(func=lambda message: message.text == "Профіль")

def profile(message):

Коли користувач надсилає повідомлення з текстом "Профіль" (тобто натискає відповідну кнопку), бот запускає функцію profile.

user_id = message.chat.id

Цей ID унікальний для кожного користувача. Він використовується, щоб дізнатися, чий профіль треба знайти в базі.

conn = connect_db()

cursor = conn.cursor()

Відкривається з’єднання з базою даних і створюється курсор для виконання SQL-запитів.

cursor.execute("""

```
        SELECT breed, name, birthdate, weight, sex, coat_color
        FROM pets
        WHERE user_id = ?
    """, (user_id,))
    result = cursor.fetchone()
    conn.close()
```

SQL-запит до таблиці pets. Повертає один запис про тварину, якщо вона є у користувача з цим user_id. Якщо запис знайдено — result містить кортеж з даними. Якщо ні - result = None.

```
    if result:
        breed, name, birthdate, weight, sex, coat_color = result
```

Розпаковуємо кортеж у змінні для зручного використання.

```
    bot.send_message(
        user_id,
        f"Ваш профіль:\n"
        f"Порода: {breed}\n"
        f"Ім'я: {name}\n"
        f"Дата народження: {birthdate}\n"
        f"Вага: {weight} кг\n"
        f"Стать: {'Хлопець' if sex == 'чоловіча' else 'Дівчина'}\n"
        f"Колір шерсті: {coat_color}",
        reply_markup=start_keyboard()
    )
```

Формується і надсилається повідомлення з профілем. Підставляється все: порода, ім'я, дата народження тощо. Також додається клавіатура start_keyboard().

```
    else:
        bot.send_message(
            user_id,
```

"Ви ще не зареєстровані. Натисніть 'Реєстрація', щоб створити профіль.",

reply_markup=start_keyboard()

)

Бот підказує, що треба натиснути "Реєстрація", щоб створити профіль.

```
def get_active_profile(user_id):
    conn = connect_db()
    cursor = conn.cursor()
    cursor.execute("""
        SELECT breed, name, birthdate, weight, sex, coat_color
        FROM pets
        WHERE user_id = ?
    """, (user_id,))
    profile = cursor.fetchone()
    conn.close()
    return profile
```

Це окрема універсальна функція, яка теж витягує профіль користувача, але вже без надсилання повідомлень. Її можна викликати з будь-якого місця у коді, коли потрібно отримати дані про профіль. Повертає None, якщо профілю немає, або кортеж з даними, якщо є.

Лістинг 7 – Обробка кнопки видалити профіль

```
@bot.message_handler(func=lambda message: message.text == "Видалити профіль")
def delete_profile(message):
```

Цей обробник реагує на повідомлення з текстом "Видалити профіль", тобто на натискання відповідної кнопки в Telegram.

```
user_id = message.chat.id
```

ID потрібен для перевірки та видалення запису конкретного користувача в базі.

```
conn = connect_db()
```

cursor = conn.cursor()

cursor.execute("SELECT COUNT(*) FROM pets WHERE user_id = ?", (user_id,))

result = cursor.fetchone()[0]

Запит повертає кількість записів із таким user_id. Якщо профіль є — отримаємо число більше 0. fetchone()[0] — беремо перший (і єдиний) стовпчик із результату.

if result > 0:

cursor.execute("DELETE FROM pets WHERE user_id = ?", (user_id,))

conn.commit()

conn.close()

Видаляємо запис з таблиці pets за user_id.

commit() — підтверджує зміни. close() — закриває з'єднання з базою.

bot.send_message(

user_id,

"Ваш профіль успішно видалено. Ви можете створити новий через 'Реєстрація'.",

reply_markup=start_keyboard()

)

Бот повідомляє про успішне видалення. І пропонує знову стартову клавіатуру для подальших дій.

else:

conn.close()

bot.send_message(

user_id,

"У вас немає активного профілю. Ви можете створити його через 'Реєстрація'.",

reply_markup=start_keyboard()

)

Якщо в базі нічого не знайдено — просто повідомляє користувачу про це.

Лістинг 8 – Обробка команди /start

@bot.message_handler(commands=['start']) def send_welcome(message): bot.send_message(message.chat.id, "Оберіть одну з опцій:", reply_markup=start_keyboard())

Цей шматок коду — це обробник команди /start у Telegram-боті. Його основне завдання — вітати користувача і показувати стартову клавіатуру з кнопками.

Лістинг 9 – Обробка вибору кнопки "Реєстрація"

```
@bot.message_handler(func=lambda message: message.text == "Реєстрація")
def registration(message):
    user_id = message.chat.id
    # Підключення до бази даних
    conn = connect_db()
    cursor = conn.cursor()
    # Перевірка, чи є у користувача профіль
    cursor.execute("SELECT COUNT(*) FROM pets WHERE user_id = ?", (user_id,))
    result = cursor.fetchone()[0]
    conn.close()
    if result > 0:
        bot.send_message(
            user_id,
            "У вас вже є зареєстрований профіль. Ви можете переглянути його в 'Профіль' або видалити через 'Видалити профіль'.",
            reply_markup=start_keyboard()
        )
    else:
```

 # Починаємо реєстрацію

user_data[user_id] = {'step': 1} # Початок реєстрації

bot.send_message(user_id, "Якої породи ваш улюбленець?") # Перше питання

Коли користувач натискає кнопку "Реєстрація", бот перевіряє, чи є вже профіль для цього користувача. Якщо профіль є, бот повідомляє про це. Якщо профілю немає — починається реєстрація, і бот запитує, якої породи улюбленець користувача.

BUTTON_NAMES = ["Реєстрація", "Порада", "Цікавинка", "Знайомство", "Профіль", "Видалити профіль"]

Це список кнопок, який виключає відповіді на кнопки з цих назв, щоб запобігти неправильному введенню. Якщо користувач відповість одним із цих варіантів, бот запросить, щоб він відповідав текстом.

@bot.message_handler(func=lambda message: message.chat.id in user_data)

def handle_registration(message):

user_id = message.chat.id

step = user_data[user_id].get('step')

Ця функція обробляє введення користувача на кожному етапі реєстрації. Змінна step відслідковує, на якому етапі знаходиться користувач у процесі реєстрації.

Покрокове збирання даних: у залежності від поточного етапу (step), бот ставить питання і зберігає відповіді в user_data:

Питання про породу:

if step == 1:

user_data[user_id]['breed'] = message.text # Зберігаємо породу

user_data[user_id]['step'] = 2

bot.send_message(user_id, "Яка стать вашого улюбленця? (чоловіча/жіноча)")

Питання про стать:

```
elif step == 2:
    user_data[user_id]['sex'] = message.text.lower()  # Зберігаємо стать
    user_data[user_id]['step'] = 3
    bot.send_message(user_id, "Як звати вашого улюбленця?")
```

Питання про ім'я:

```
elif step == 3:
    user_data[user_id]['name'] = message.text  # Зберігаємо ім'я
    user_data[user_id]['step'] = 4
    bot.send_message(user_id, "Дата народження вашого улюбленця? (в форматі ДД.ММ.РРРР)")
```

Питання про дату народження:

```
elif step == 4:
    user_data[user_id]['birthdate'] = message.text  # Зберігаємо дату народження
    user_data[user_id]['step'] = 5
    bot.send_message(user_id, "Яка вага вашого улюбленця?")
```

Питання про вагу:

```
elif step == 5:
    user_data[user_id]['weight'] = message.text  # Зберігаємо вагу
    user_data[user_id]['step'] = 6
    bot.send_message(user_id, "Який колір шерсті у вашого улюбленця?")
```

Питання про колір шерсті:

```
elif step == 6:
    user_data[user_id]['coat_color'] = message.text  # Зберігаємо колір шерсті
```

Після того як всі дані зібрані, вони зберігаються в базі даних у таблиці pets.

```
# Завершуємо реєстрацію і додаємо дані в базу
breed = user_data[user_id].get('breed')
name = user_data[user_id].get('name')
```

 # Додаємо дані в базу даних

conn = connect_db()

cursor = conn.cursor()

cursor.execute("""

INSERT INTO pets (user_id, breed, name, birthdate, weight, sex, coat_color)

VALUES (?, ?, ?, ?, ?, ?, ?)

""", (user_id, breed, name, user_data[user_id].get('birthdate'), user_data[user_id].get('weight'), user_data[user_id].get('sex'), user_data[user_id].get('coat_color')))

conn.commit()

conn.close()

Бот виводить приємне повідомлення після успішної реєстрації і дає користувачу стартову клавіатуру.

bot.send_message(user_id, f"Приємно познайомитись! {name}, я допоможу стати щасливішим!", reply_markup=start_keyboard())

Після завершення реєстрації, стан користувача видаляється з user_data, щоб звільнити пам'ять та уникнути помилок при наступних взаємодіях.

del user_data[user_id]

Лістинг 10 – Обробка AI запитів + реалізація кнопок у порадах[41]

Функція: get_advice_from_ai(topic)

Звертається до OpenAI GPT-3.5 Turbo, щоб отримати пораду по темі, пов’язаній із собаками (догляд, дресирування, харчування, вакцинація). Імпортує бібліотеку OpenAI (на випадок, якщо вона ще не імпортована в коді). Встановлює API-ключ для доступу до сервісу (використовуючи змінну AI_key). Створює список повідомлень messages, які пояснюють GPT, що воно — експерт з догляду за собаками, і формують запит від користувача. Викликає openai.ChatCompletion.create(...), щоб згенерувати відповідь. Якщо все ок — повертає результат, якщо сталася помилка — виводить повідомлення про помилку.

Параметри GPT:

- model: "gpt-3.5-turbo"
- max_tokens: 150 (довжина відповіді)
- temperature: 0.7 (рівень креативності відповіді)

Функція: advice_keyboard()

Створює меню-клавіатуру Telegram з кнопками, які користувач може натиснути для отримання поради:

"Догляд" — поради щодо гігієни, купання, стрижки і т.д.

"Дресирування" — як виховувати собаку, навчати команд.

"Корм" — поради щодо харчування.

"Вакцинація" — рекомендації щодо щеплень.

"Назад" — повернення до головного меню.

resize_keyboard=True — клавіатура адаптується до екрана користувача.

Кнопки додаються у клавіатуру через markup.add(...).

Лістинг 11 – Пошук схожих улюбленців[43]

```
# Функція для пошуку схожого улюбленця
def find_similar_pet(user_id):
    conn = connect_db()
    cursor = conn.cursor()
    # Отримуємо дані користувача
     cursor.execute("SELECT breed, weight, coat_color FROM pets WHERE user_id = ?", (user_id,))
    user_pet = cursor.fetchone()
    if not user_pet:
        conn.close()
        return None  # Якщо користувача немає в базі даних
    breed, weight, coat_color = user_pet
    # Пошук схожих улюбленців (порода, вага та колір шерсті)
    cursor.execute("""
        SELECT user_id, breed, weight, coat_color FROM pets
```

```
        WHERE breed = ? AND weight BETWEEN ? AND ? AND coat_color =
?
        """, (breed, float(weight) - 5, float(weight) + 5, coat_color))
        similar_pet = cursor.fetchall()
        conn.close()
        return similar_pet
```

Функція find_similar_pet(user_id)

Призначення: пошук схожих улюбленців користувача.

Логіка: отримує з БД інформацію про тварину користувача (breed, weight, coat_color). Повертає список тварин тієї ж породи, схожої ваги (±5 кг) і того ж кольору шерсті. Повертає: список кортежів або None, якщо даних немає.

Хендлер Знайомство

```
# Модернізована функція "Знайомство"
@bot.message_handler(func=lambda message: message.text ==
"Знайомство")
def acquaintance(message):
    user_id = message.chat.id
    # Підключення до бази даних
    conn = connect_db()
    cursor = conn.cursor()
    # Отримуємо породу користувача
      cursor.execute("SELECT breed FROM pets WHERE user_id = ?",
(user_id,))
    result = cursor.fetchone()
    if not result:
        # Якщо користувач не зареєстрований
          bot.send_message(user_id, "Спочатку потрібно зареєструвати
вашого улюбленця. Оберіть 'Реєстрація'.")
        conn.close()
        return
```

```
        breed = result[0]
            # Підрахунок загальної кількості, кількості хлопців і дівчат (виключаючи власний профіль)
        cursor.execute("SELECT COUNT(*) FROM pets WHERE breed = ? AND user_id != ?", (breed, user_id))
        total_count = cursor.fetchone()[0]
        cursor.execute("SELECT COUNT(*) FROM pets WHERE breed = ? AND sex = 'чоловіча' AND user_id != ?", (breed, user_id))
        male_count = cursor.fetchone()[0]
        cursor.execute("SELECT COUNT(*) FROM pets WHERE breed = ? AND sex = 'жіноча' AND user_id != ?", (breed, user_id))
        female_count = cursor.fetchone()[0]
        conn.close()
        # Відправка повідомлення з інформацією
        bot.send_message(
            user_id,
            f"Зареєстровано {total_count} таких порід (без вашого профілю).\n"
            f"З них {male_count} хлопців і {female_count} дівчат."
        )
        # Клавіатура з кнопками
        markup = telebot.types.ReplyKeyboardMarkup(resize_keyboard=True)
        male_btn = telebot.types.KeyboardButton("Чоловік")
        female_btn = telebot.types.KeyboardButton("Жінка")
        back_btn = telebot.types.KeyboardButton("Назад")
        markup.add(male_btn, female_btn, back_btn)
                bot.send_message(user_id, "Оберіть одну з опцій:", reply_markup=markup)
```

Призначення: надання користувачу статистики щодо зареєстрованих тварин тієї ж породи. Дії:

Перевіряє, чи зареєстрована тварина користувача.

Підраховує кількість інших тварин тієї ж породи, поділених за статтю.

Виводить результати у вигляді повідомлення.

Виводить клавіатуру з трьома кнопками: "Чоловік", "Жінка", "Назад".

Хендлер Чоловік

```
# Обробка кнопки "Чоловік"
@bot.message_handler(func=lambda message: message.text == "Чоловік")
def choose_male(message):
    user_id = message.chat.id
    # Пошук рандомного чоловіка обраної породи
    conn = connect_db()
    cursor = conn.cursor()
    cursor.execute("""
        SELECT name, birthdate, weight, coat_color FROM pets
        WHERE breed = (SELECT breed FROM pets WHERE user_id = ?) AND
sex = 'чоловіча'
        ORDER BY RANDOM() LIMIT 1
    """, (user_id,))
    result = cursor.fetchone()
    conn.close()
    if result:
        name, birthdate, weight, coat_color = result
        bot.send_message(
            user_id,
            f"Знайдено хлопця:\n"
            f"Ім'я: {name}\n"
            f"Дата народження: {birthdate}\n"
            f"Вага: {weight} кг\n"
            f"Колір шерсті: {coat_color}"
        )
    else:
```

```
            bot.send_message(user_id, "Немає зареєстрованих хлопців цієї породи.")
```

Призначення: Показує випадкового "хлопця" тієї ж породи.

Логіка: витягує випадковий запис зі статтю "чоловіча" і тією ж породою. Показує основну інформацію: ім’я, дата народження, вага, колір шерсті.

Хендлер Жінка

```
# Обробка кнопки "Жінка"
@bot.message_handler(func=lambda message: message.text == "Жінка")
def choose_female(message):
    user_id = message.chat.id
    # Пошук рандомної жінки обраної породи
    conn = connect_db()
    cursor = conn.cursor()
    cursor.execute("""
        SELECT name, birthdate, weight, coat_color FROM pets
        WHERE breed = (SELECT breed FROM pets WHERE user_id = ?) AND sex = 'жіноча'
        ORDER BY RANDOM() LIMIT 1
    """, (user_id,))
    result = cursor.fetchone()
    conn.close()
    if result:
        name, birthdate, weight, coat_color = result
        bot.send_message(
            user_id,
            f"Знайдено дівчину:\n"
            f"Ім'я: {name}\n"
            f"Дата народження: {birthdate}\n"
            f"Вага: {weight} кг\n"
            f"Колір шерсті: {coat_color}"
```

)

else:

bot.send_message(user_id, "Немає зареєстрованих дівчат цієї породи.")

Аналогічний до Чоловік, тільки шукає улюбленця жіночої статі.

Хендлер Назад

 # Обробка кнопки "Назад"

@bot.message_handler(func=lambda message: message.text == "Назад")

def go_back(message):

 # Повертаємо користувача на головне меню

bot.send_message(message.chat.id, "Оберіть одну з опцій:", reply_markup=start_keyboard())

conn.close()

Призначення: повертає користувача до головного меню.

bot.polling(none_stop=True)

Запускає бота в нескінченному циклі, обробляє всі повідомлення.

РОЗДІЛ 3. ПЕРСПЕКТИВИ РОЗВИТКУ ЧАТ-БОТА З ДОГЛЯДУ ЗА ДОМАШНІМИ УЛЮБЛЕНЦЯМИ

Для якісного порівняння з українськими аналогами спершу визначимо ключові критерії порівняння, а потім подивимось на існуючі сервіси/боти в Україні.

Ключові критерії для порівняння включають можливість реєстрації улюбленця, функцію підбору схожої тварини або пошуку за характеристиками, інтеграцію зі штучним інтелектом (AI-поради, аналіз симптомів, догляд), додаткові функції (нагадування про щеплення, рекомендації по харчуванню тощо), зручність інтерфейсу/UX, мову інтерфейсу (українська, англійська) та наявність мобільного додатку або лише Telegram-бота.

Порівняння з українськими аналогами виглядає наступним чином: PetCare Bot (Україна) не має інтеграції з AI, але має можливість реєстрації тварин, не має функції підбору тварин, проте є функції нагадування про щеплення і нотатки, UX базовий, текстовий інтерфейс, мова українська, а платформа - Telegram.

UAnimals Bot більше фокусується на захисті прав тварин, не має інтеграції з AI, не передбачає реєстрації тварин, пошук тварин є, але тільки в рамках допомоги (пошук нової домівки), функції включають новини та підтримку притулків, мова українська, платформа - Telegram.

VetBot - незалежний український вет-бот, частково інтегрований з AI (база знань та автогенерація відповідей), не має реєстрації тварин і функції підбору тварин, але дає поради з лікування і можливість запису до ветеринара, UX текстовий, мова українська, платформа - Telegram.

Таблиця 3.1

Порівняння з моїм ботом

Критерій	Мій бот	Найближчий аналог
Реєстрація тварини	+	PetCare Bot
AI-поради	+	(немає прямого аналогу)
Пошук схожих тварин	+	(немає в аналогах)
UX/інтерфейс	Простий та приємний до сприйняття	Зазвичай базовий
Нагадування / догляд	+	PetCare Bot
Мова	Українська	Українська
Платформа	Telegram	Telegram

Мій бот має унікальні фішки, які не реалізовані в жодному з українських аналогів, особливо це стосується інтеграції зі штучним інтелектом, пошуку схожих тварин (наприклад, для майбутнього розведення, знайомств або вибору улюбленця) та персоналізованого підходу через реєстрацію тварини та аналіз даних.

Зважаючи на вже реалізований багатофункціональний потенціал чат-бота, подальший розвиток може орієнтуватися як на поглиблення персоналізації, так і на розширення можливостей для користувачів. Насамперед доцільно допрацювати функціонал, пов’язаний з медичним обслуговуванням тварин. Варто додати розділ із веденням медичної історії улюбленця, де користувач зможе зберігати інформацію про щеплення, попередні хвороби, візити до ветеринара, результати аналізів. Це дозволить не лише мати під рукою всю важливу інформацію, а й автоматизувати нагадування про чергові вакцинації чи профілактичні обстеження.

Завдяки можливостям штучного інтелекту бот можна доповнити міні-діагностичним інструментом. Користувач зможе вказати симптоми, які

спостерігаються у тварини, а бот - на основі бази знань - надасть попередню оцінку ситуації та поради щодо подальших дій. Також доцільно реалізувати функцію аналізу поведінки, харчування та загального стану тварини на основі внесених даних або звичок користувача.

Серед перспективних напрямків розвитку - можливість створення декількох профілів для кількох тварин. Це буде зручно для користувачів, у яких вдома живе більше одного улюбленця, і дозволить вести окрему історію для кожного. Окремої уваги заслуговує функція пошуку схожих тварин: у майбутньому її можна розширити, створивши платформу для знайомств між власниками - наприклад, для підбору партнера для розведення або просто соціалізації тварин.

Ще одним кроком вперед може стати інтеграція з геолокаційними сервісами, що дозволить ботові підказувати найближчі ветеринарні клініки, грумінг-салони або зоомагазини. У разі надзвичайних ситуацій користувач зможе швидко дізнатись, куди звернутись, перебуваючи в будь-якій точці міста чи країни. Також актуальним буде підключення до баз даних притулків — це дозволить використовувати бота не лише як засіб догляду, а й як платформу для пошуку нового друга, а також для поширення інформації про зниклих тварин.

Щоб заохотити регулярне використання чат-бота, можна ввести елементи гейміфікації: користувачі отримуватимуть віртуальні відзнаки за турботу, щоденні перевірки здоров’я, своєчасні щеплення тощо. Такі елементи не лише створюють додаткову мотивацію, а й формують позитивну звичку до відповідального ставлення до тварин. Ще одним цікавим доповненням стане можливість аналізу фото - наприклад, для розпізнавання породи або оцінки стану шерсті чи шкіри.

Зрештою, у довгостроковій перспективі бот може перетворитись на повноцінну цифрову платформу для власників тварин - із календарем подій, журналом спостережень, можливістю експорту медичних даних у PDF-

форматі для передавання ветеринару, і навіть індивідуальними порадами щодо виховання, тренувань чи раціону.

Якщо проект передбачає масштабування, це також відкриває перспективи комерційної реалізації - через підключення партнерських послуг (ветеринарні мережі, магазини), розміщення реклами або впровадження платного доступу до преміум-функцій.

Розроблений чат-бот має широкі можливості для практичного використання в реальних умовах, адже він орієнтований на вирішення актуальних потреб власників домашніх тварин. У сучасному світі зростає тенденція до відповідального ставлення до улюбленців, що супроводжується підвищенням попиту на зручні цифрові інструменти для догляду, контролю здоров’я та побутової організації життя тварини. Завдяки цьому бот може знайти своє застосування серед великого кола користувачів.

Основною цільовою аудиторією є власники собак, котів та інших домашніх тварин, які шукають інструмент для ведення обліку медичної інформації, отримання порад по догляду, пошуку товарів та послуг для своїх улюбленців. Особливо цінним цей бот може бути для молодих людей 20–40 років, які активно користуються месенджерами (зокрема Telegram) і відкриті до використання інноваційних сервісів, що спрощують повсякденне життя. Також до потенційних користувачів входять сім’ї з дітьми, де тварина є частиною сімейного кола, і навіть літні люди, яким зручно отримувати поради без необхідності встановлення складних додатків.

Застосування бота може бути корисним і в роботі професійних ветеринарних клінік, грумерів або зоомагазинів. Наприклад, бот може нагадувати клієнтам про заплановані візити, інформувати про нові акції або надавати базову підтримку у вигляді AI-консультацій, зменшуючи навантаження на персонал. У майбутньому бот може стати корисним інструментом для притулків або волонтерських організацій - для реєстрації тварин, координації допомоги та комунікації з потенційними власниками.

Щодо комерційного потенціалу, цей проєкт має перспективу монетизації через кілька напрямків. По-перше, це модель freemium: основний функціонал є безкоштовним, але за доступ до розширених можливостей (наприклад, AI-діагностика, збереження медичної історії в PDF, пошук партнерів для розведення) можна запровадити платну підписку. По-друге, бот може стати платформою для реклами ветеринарних клінік, зоотоварів, кормів або послуг грумінгу - з урахуванням локації користувача та його потреб. По-третє, можливе партнерство з онлайн-магазинами, які пропонують товари для тварин: бот може рекомендувати продукти на основі інформації про тварину, отримуючи частку від кожної покупки за реферальним посиланням.

Таким чином, чат-бот може успішно функціонувати як у приватному користуванні, так і як інструмент для бізнесу. Завдяки поєднанню штучного інтелекту, зручного інтерфейсу та широкого спектра функцій, він здатен зайняти свою нішу серед інноваційних сервісів на українському ринку цифрових рішень для догляду за домашніми тваринами.

ВИСНОВКИ

У ході кваліфікаційної дипломної роботи було розроблено інтелектуального чат-бота, призначеного для надання корисної інформації та нагадувань щодо догляду за домашніми улюбленцями. Система реалізована із застосуванням мови програмування Python, бібліотек для обробки природної мови, а також інтеграції з OpenAI API для забезпечення гнучкого та контекстно-залежного діалогу з користувачем. Особливий акцент зроблено на підтримці української мови, інтуїтивному інтерфейсі та можливості персоналізації під потреби власників тварин.

У теоретичній частині роботи здійснено ґрунтовний аналіз існуючих рішень у сфері цифрових асистентів для догляду за тваринами, розглянуто підходи до створення чат-ботів, особливості NLP-алгоритмів, а також обґрунтовано вибір інструментів розробки. Були детально розглянуті переваги використання Python, openai, а також продемонстровано принципи інтеграції з базами даних та месенджерами.

У процесі проєктування було сформульовано функціональні та нефункціональні вимоги до системи, спроєктовано архітектуру чат-бота, алгоритми обробки запитів, генерації відповідей та системи нагадувань. Реалізовано ключові функції: щоденні поради по догляду, інформування про годування, вакцинації, вигул, графік візитів до ветеринара та інші персоналізовані повідомлення. Інтеграція з моделлю GPT дозволила надати чат-боту здатність вести осмислений діалог, давати рекомендації та підтримувати запити, пов’язані зі здоров’ям та поведінкою тварин.

Практичне тестування проводилося за участі користувачів, які мають домашніх тварин різних видів. Тестування підтвердило здатність системи ефективно реагувати на запити, враховувати контекст попередніх розмов та забезпечувати інформативну та зручну взаємодію. Користувачі відзначили корисність функцій нагадувань та можливість швидко отримувати поради з перевірених джерел. Було також проведено первинний аналіз економічної

доцільності впровадження подібного рішення на ринок, розглянуто перспективи комерціалізації через підписковy модель або співпрацю з ветеринарними клініками.

Порівняння розробленого рішення з існуючими чат-ботами виявило низку переваг: орієнтацію на локальний ринок, адаптацію під українську мову, гнучкість у додаванні нових сценаріїв, а також простоту інтеграції з популярними месенджерами (Telegram, Viber). Серед визначених недоліків — залежність від стабільного інтернет-з'єднання, обмеженість у розпізнаванні складних медичних симптомів без консультації з фахівцем, а також потреба у подальшому вдосконаленні бази знань.

Таким чином, виконана робота охоплює повний цикл створення інтелектуального чат-бота: від ідеї та проєктування — до реалізації, тестування, аналізу ефективності та визначення напрямів подальшого розвитку. Створене рішення є прикладом використання сучасних технологій штучного інтелекту для підвищення якості догляду за тваринами, полегшення повсякденних обов'язків власників та формування цифрового асистента нового покоління. У перспективі проєкт може бути доповнений можливостями розпізнавання зображень (наприклад, для оцінки стану шерсті чи ранок), розширений функціоналом для інших видів тварин, а також інтегрований із розумними IoT-пристроями для моніторингу активності та здоров'я улюбленців.

Для отримання додаткової інформації та взаємодії з чат-ботом, переходьте до: t.me/IPC_bbot, де можна спробувати всі можливості даного рішення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Яковенко І. Чат-бот на основі ШІ для догляду за домашніми тваринами доступний в Україні. The Village Україна. URL: https://www.village.com.ua/village/life/edu-news/346233-chat-bot-na-osnovi-shi-dlya-doglyadu-za-domashnimi-tvarinami-dostupniy-v-ukrayini (дата звернення: 22.04.2025).

2. Contributors to Wikimedia projects. ELIZA - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/ELIZA (date of access: 22.04.2025).

3. Team T. I. Chatbot Definition, Types, Pros & Cons, Examples. Investopedia. URL: https://www.investopedia.com/terms/c/chatbot.asp (дата звернення: 22.04.2025).

4. Dimension Labs. Dimension Labs. URL: https://www.dimensionlabs.io/blog/chatbots-vs-conversational-ai-a-complete-guide#:~:text=Rule-based%20chatbots%20don't,intelligently%20based%20on%20those%20needs (date of access: 22.04.2025).

5. Church B., Finn T. Types of Chatbots | IBM. IBM - United States. URL: https://www.ibm.com/think/topics/chatbot-types (date of access: 22.04.2025).

6. BritCareBot: перший чат-бот на основі ШІ для власників домашніх тварин в Україні. VAFO. URL: https://www.vafo.com/brit-care-friendly-bot-the-first-ai-powered-chatbot-for-pet-parents-in-ukraine/ (дата звернення: 22.04.2025).

7. Rasa: Open Source Language Understanding and Dialogue Management. arXiv.org. URL: https://arxiv.org/abs/1712.05181 (date of access: 22.04.2025).

8. OhNxuB. Новий ШІ-сервіс допоможе в догляді за котом чи собакою. Як працює BritCareBot. dev.ua. URL: https://dev.ua/en/news/traryn-1703164828 (дата звернення: 22.04.2025).

9. ChatGPT in Veterinary Medicine: A Practical Guidance of Generative Artificial Intelligence in Clinics, Education, and Research. arXiv.org. URL: https://arxiv.org/abs/2403.14654 (date of access: 22.04.2025).

10. Purrfessor: A Fine-tuned Multimodal LLaVA Diet Health Chatbot. arXiv.org. URL: https://arxiv.org/abs/2411.14925 (date of access: 22.04.2025).

11. HHH: An Online Medical Chatbot System based on Knowledge Graph and Hierarchical Bi-Directional Attention. arXiv.org. URL: https://arxiv.org/abs/2002.03140 (date of access: 22.04.2025).

12. Танасійчук І. В. (2022). Розробка телеграм-каналу на ветеринарну тематику «Zootopic». Національний авіаційний університет. URL: https://er.nau.edu.ua/items/2e0153bd-9e58-48c2-8537-e60bb5bb0b9c (дата звернення: 22.04.2025).

13. Fins IS, Davies H, Farrell S, et al. (2023). Evaluating ChatGPT text mining of clinical records for obesity monitoring. Veterinary Record. URL: https://bvajournals.onlinelibrary.wiley.com/doi/10.1002/vetr.3669 (date of access: 22.04.2025).

14. Карпенко А. В. (2022). Online-сервіс пошуку домашніх тварин. Національний авіаційний університет. URL: https://er.nau.edu.ua/items/1a18d783-a505-4e71-a30c-4b7794d6a154 (дата звернення: 22.04.2025).

15. Cohen-Hutton S., Wells D. What Your Dog Is Thinking. New York: Greenwood Press, 2022. 256 p.

16. Friedman E. This Dog Will Change Your Life. New York: Simon and Schuster, 2025. 288 p.

17. Основи дресирування, гігієни та годівлі службових собак / за ред. В. Мельника. Київ : Магнолія, 2024. 224 с.

18. Кемерон Б. З любов'ю від Кленсі. Щоденник хорошого собаки. – Київ : Клуб Сімейного Дозвілля, 2023. 288 с.

19. Терещенко О. В. (2021). Розробка програмного забезпечення для ведення цифрової картки домашньої тварини. ResearchGate. System technologies 4(153):47-57. DOI:10.34185/1562-9945-4-153-2024-06.

20. Jokar, M., Abdous, A., & Rahmanian, V. (2024). AI chatbots in pet health care: Opportunities and challenges for owners. Veterinary Medicine and Science. URL: https://onlinelibrary.wiley.com/doi/10.1002/vms3.1464?utm_source=chatgpt.com (date of access: 22.04.2025).

21. Freshworks. (n.d.). How To Build A Chatbot Knowledge Base. Freshworks. URL: https://www.freshworks.com/knowledge-base/chatbots/?utm_source=chatgpt.com (date of access: 22.04.2025).

22. PetWisdomHub. (n.d.). Find AI Tools for AI Knowledge Base: PetWisdomHub. WorkWithAI. URL: https://www.workwithai.io/tool/petwisdomhub?utm_source=chatgpt.com (date of access: 22.04.2025).

23. Стратегія розвитку штучного інтелекту в Україні / за ред. І.В. Сергієнка. Київ : Інститут штучного інтелекту, 2023. 256 с.

24. Botiverse. Botiverse. URL: https://pypi.org/project/botiverse/?utm_source=chatgpt.com (date of access: 22.04.2025).

25. ChatbotCreator. ChatbotCreator. URL: https://pypi.org/project/chatbot-creator/?utm_source=chatgpt.com (date of access: 22.04.2025).

26. BESSER Bot Framework. Livable Software. URL: https://livablesoftware.com/besser-bot-framework/?utm_source=chatgpt.com (date of access: 22.04.2025).

27. Top Chatbot Platforms in 2024: A Comprehensive Guide. Toxigon. URL: https://toxigon.com/top-chatbot-platforms-2024 (date of access: 22.04.2025).

28. Character.ai. Wikipedia. URL: https://en.wikipedia.org/wiki/Character.ai (date of access: 22.04.2025).

29. Replika. Wikipedia. URL: https://en.wikipedia.org/wiki/Replika (date of access: 22.04.2025).

30. Ernie Bot. Wikipedia. URL: https://en.wikipedia.org/wiki/Ernie_Bot (date of access: 22.04.2025).

31. Feine, J., Morana, S., & Maedche, A. (2020). Designing Interactive Chatbot Development Systems. Proceedings of the 41st International Conference on Information Systems (ICIS 2020). URL: https://chatbotresearch.org/publication/feine-designing-2020/ (date of access: 22.04.2025).

32. Roller, S., Dinan, E., Goyal, N., et al. (2020). Recipes for building an open-domain chatbot. arXiv. URL: https://arxiv.org/abs/2004.13637 (date of access: 22.04.2025).

33. Csaky, R. (2019). Deep Learning Based Chatbot Models. arXiv. URL: https://arxiv.org/abs/1908.08835 (date of access: 22.04.2025).

34. Zhou, L., Gao, J., Li, D., & Shum, H.-Y. (2018). The Design and Implementation of XiaoIce, an Empathetic Social Chatbot. arXiv. URL: https://arxiv.org/abs/1812.08989 (date of access: 22.04.2025).

35. Gray M. The Dogist: Picturing the Life of Dogs. New York : Art Street Press, 2021. 320 p.

36. Feldman R. The Complete Guide to Dog Care. London: HarperCollins, 2020. 352 p.

37. Lutz V. How to Train Your Dog: A Guide for Beginners. Los Angeles : New World Press, 2021. 212 p.

38. Павлюк О. Догляд за тваринами в умовах війни / за ред. А. Шевченка. Київ : Видавництво ВД "Діса", 2023. 180 с.

39. Bao, S., He, H., Wang, F., et al. (2020). PLATO-2: Towards Building an Open-Domain Chatbot via Curriculum Learning. arXiv. URL: https://arxiv.org/abs/2006.16779 (date of access: 22.04.2025).

40. Hempel, J. (2015, August 26). Facebook Launches M, Its Bold Answer to Siri and Cortana. Wired. URL: https://www.wired.com/story/facebooks-virtual-assistant-m-is-dead-so-are-chatbots/ (date of access: 22.04.2025).

41. OpenAI API. PyPI. URL: https://pypi.org/project/openai/ (date of access: 22.04.2025).

42. Pandas. PyPI. URL: https://pypi.org/project/pandas/ (date of access: 22.04.2025).

43. SQLite. Python Documentation. URL: https://docs.python.org/3/library/sqlite3.html (date of access: 22.04.2025).

44. Офіційна документація Python. Python Documentation. URL: https://docs.python.org/3/ (дата звернення: 22.04.2025).