

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: «Розробка Telegram чат-боту зі штучним
інтелектом мовою C#»

Виконав: студент 4 курсу, групи ПД-42

спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Дмитришин В.В.

(прізвище та ініціали)

Керівник

Гребенюк В.В.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Нормоконтроль

(прізвище та ініціали)

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Бакалавр»

Спеціальність -121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

Негоденко О.В.

« ____ » _____ 2023 року

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ РОБОТУ

СТУДЕНТУ ДМИТРИШИН ВАЛЕРІЙ ВІТАЛІЙОВИЧ

1. Тема роботи: «Розробка Telegram чат-боту зі штучним інтелектом мовою C#»

Керівник роботи: Гребенюк Віктор Вікторович, доцент кафедри ПІЗ

Затверджені наказом вищого навчального закладу від «24» лютого 2023 року

№26

2. Строк подання студентом роботи «1» червня 2023 року

3. Вхідні дані до роботи:

3.3 Вибір мови програмування.

3.4 Алгоритм дії бота.

3.5 Науково-технічна література, пов'язана з розробкою ботів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

4.1. Аналіз обов'язків розроблюваного бота

4.2. Аналіз та порівняння існуючих прототипів

- 4.3. Дослідження програмних засобів для розробки бота
- 4.4. Розробити функціонал створеного бота
- 5. Перелік графічного матеріалу
 - 5.1. Популярність телеграм-ботів
 - 5.2. Огляд можливостей розробленого бота
 - 5.3. Апробація результатів досліджень
- 6. Дата видачі завдання «25» лютого 2023 року

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	11.04.23 – 14.04.23	Виконано
2	Аналіз існуючих аналогів	15.04.23 – 17.04.23	Виконано
3	Дослідження програмних засобів	18.04.23 – 21.05.23	Виконано
4	Моделювання об'єкту проектування	26.04.23 – 05.05.23	Виконано
5	Розробка функціоналу застосунку	05.05.23 – 07.05.23	Виконано
6	Вступ, висновки, реферат	11.05.23 – 15.05.23	Виконано
7	Розробка презентації застосунку	16.05.23 – 01.06.23	Виконано
8	Попередній захист роботи	23.05.23	
9	Здача роботи	01.06.23	

Студент _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 46с., 18 мал., 19 джерел.

Ключеві слова: C#, Telegram, чат-бот.

Об'єкт дослідження – опрацювання та надання відповідей на запити користувача.

Предмет дослідження – чат-бот для відповідей на запити користувача..

Мета роботи – спрощення процесу надання відповідей для користувача за рахунок Telegram чат-боту зі штучним інтелектом мовою C#.

Методи дослідження – Аналіз та оптимізація передачі та обробки інформації.

Визначення структури системи перед розробкою. Підходи до ефективної розробки та управління програмними продуктами. Техніки для перевірки якості програмного забезпечення

Наукова новизна – розроблений алгоритм отримання та надання відповідей за запитом користувача в зручному для використання з телефона методі.

Практичне значення отриманих результатів - Автоматизації комунікації та ефективного виконанні рутинних завдань без прямої участі людини та створення зоображень за запитом користувача.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Загальний аналіз платформи телеграм.....	9
1.2 Аналіз телеграм чат-ботів.....	10
1.3 Загальний аналіз штучного інтелекту	12
1.4 Аналіз потреб потенційної цільової аудиторії продукту	15
1.5 Аналіз аналогів.....	16
2 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ	17
2.1 Огляд мови програмування	17
2.2 Огляд середовища розробки.....	18
2.3 Огляд хмарних технологій	20
2.4 Огляд інтерфейсу прикладного програмування OpenAI(API).....	23
2.5 Огляд інтерфейсу прикладного програмування Telegram(API)	24
3 РЕАЛІЗАЦІЯ ЧАТ-БОТУ З ШТУЧНИМ ІНТЕЛЕКТОМ.....	26
3.1 Проектування та розробка діаграми варіантів використання.....	26
3.2 Проектування та розробка структурна діаграма розгортання	28
3.3 Проектування та розробка діаграма класів.....	29
4.ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	31
4.1 Модульне тестування	32
4.2 Функціональне тестування	34
4.3 Інструкція користувачу.....	44
ВИСНОВКИ	46
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТКИ	49
Додаток А	49

ВСТУП

Обґрунтування вибору теми та її актуальність:

- Зростання популярності месенджерів: Telegram є одним з найпопулярніших месенджерів у світі, з великою кількістю активних користувачів. Розробка чат-ботів для таких платформ стає все більш актуальною, оскільки це надає можливість зручної та ефективної комунікації з користувачами.
- Застосування штучного інтелекту: Штучний інтелект стає все більш потужним і доступним, дозволяючи розробникам створювати ботів, які можуть розпізнавати та розуміти природну мову, а також надавати відповіді засновані на аналізі інформації.
- Застосування в різних галузях: Чат-боти зі штучним інтелектом можуть мати великий потенціал застосування в різних галузях. Вони можуть бути використані в онлайн-торгівлі для надання порад та відповідей на питання користувачів, в освітніх проектах для навчання студентів або навіть у медичних системах для надання консультацій та порад.
- Вибір мови програмування C#: Мова програмування C# є популярною мовою програмування, особливо в розробці додатків для платформи .NET.

Для реалізації поставленої мети потрібно вирішити наступні завдання:

1. Провести аналіз обраної предметної області та визначити основні можливості штучного інтелекту.
2. Обрати оптимальне рішення для розробки Telegram чат-боту, з урахуванням вимог до функціональності та ефективності.
3. Сформувати вимоги до розробки Telegram чат-боту з урахуванням недоліків та переваг вже існуючих аналогів.
4. Спроекувати та розробити Telegram чат-боту зі штучним інтелектом для відповідей на запити користувача та можливості генерації зображень.
5. Провести функціональне тестування можливостей чат-боту.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний аналіз платформи телеграм

Telegram - це безкоштовне програмне забезпечення для миттєвих повідомлень (IM), яке працює на різних платформах і надає можливість миттєвого обміну повідомленнями. Ця послуга також пропонує наскрізне зашифровані відеодзвінки, VoIP, обмін файлами та багато інших функцій.

Telegram був запущений вперше 14 серпня 2013 року, а версія для Android з'явилася у жовтні того ж року. Сервери Telegram розгорнуті у різних країнах для зменшення навантаження на мережу, а операційний центр знаходиться в Дубаї. Telegram доступний на різних платформах, включаючи офіційні програми для Android, iOS, Windows, macOS та Linux, а також веб-версію. Крім того, існує багато неофіційних клієнтів, які використовують протокол Telegram. Всі офіційні компоненти Telegram відкриті, за винятком сервера, який залишається закритим і є власністю компанії.

Telegram надає можливість наскрізного зашифрованого обміну голосовими та відеодзвінками, а також зашифрованими "секретними" чатами. Крім того, хмарні чати та групи шифруються між програмою та сервером, що захищає дані від доступу провайдерів та третіх сторін. Проте сервер Telegram має ключ для розшифрування. В січні 2021 року Telegram мав понад 500 мільйонів активних користувачів на місяць.

Головні функцій та особливості:

- Миттєві повідомлення: Telegram дозволяє надсилати текстові повідомлення одному чи кільком користувачам. Ви можете використовувати різні стилі форматування тексту, додавати емодзі, використовувати анімовані наклейки та GIF-зображення, щоб зробити ваше спілкування цікавим та виразним.
- Групові чати: Ви можете створювати групові чати до 200 000 учасників.

Групові чати підтримують функції, такі як оповіщення, згадування користувачів, вкладення файлів, а також можливість налаштування рівнів доступу та прав адміністраторів.

- **Канали:** Telegram дозволяє створювати канали, які можуть мати необмежену кількість підписників. Канали використовуються для передачі повідомлень великою аудиторією. Ви можете створювати публічні канали, доступні для всіх користувачів, або приватні канали з обмеженим доступом.
- **Захист приватності:** Telegram надає велику увагу захисту приватності користувачів. Усі повідомлення в зашифрованій формі передаються через хмару Telegram, що робить їх недоступними для перехоплення третіми особами. Крім того, ви можете використовувати режим "секретних чатів", який забезпечує додатковий рівень шифрування і автоматичне видалення повідомлень після певного часу.

1.2 Аналіз телеграм чат-ботів

Телеграм-боти - це програмні агенти, які працюють в месенджері Telegram і здатні автоматизувати різноманітні завдання та надавати користувачам різні сервіси і функціональні можливості. Вони побудовані на основі Telegram Bot API, який надає розробникам доступ до інтерфейсу програмування додатків (API) для створення та керування ботами.

Телеграм-боти мають широкий спектр застосувань і можуть бути використані в різних сферах. Ось деякі з них:

- **Комерційні боти:** Боти можуть використовуватись для здійснення продажів, обробки замовлень, надання інформації про товари та послуги, підтримки клієнтів та багато іншого. Вони можуть автоматично відповідати на запити користувачів, приймати платежі, надсилати сповіщення про акції і знижки.
- **Ігрові боти:** Боти можуть пропонувати користувачам різні ігри, включаючи головоломки, квестові завдання, множинні ігри та багато іншого. Вони можуть мати

систему балів, досягнень, лідерів та змагань, що стимулює активну участь користувачів.

- **Новинні боти:** Боти можуть надавати користувачам актуальні новини, статті, огляди та інші матеріали з різних джерел. Вони можуть фільтрувати інформацію за темою або інтересами користувача, надсилати регулярні оновлення та сповіщення.
- **Освітні боти:** Боти можуть надавати освітній контент, допомагати у вивченні нових мов, навичок програмування, математики, науки та інших предметів. Вони можуть мати вбудовані тести, вправи, лекції та інтерактивні завдання для поліпшення знань користувачів.
- **Персональні асистенти:** Боти можуть надавати різні функції особистого асистента, такі як нагадування про події, планування завдань, управління розкладом, конвертація валют, пошук інформації, переклад текстів та багато іншого.
- **Здоров'я та медицина:** Телеграм-боти можуть допомагати у покращенні доступу до медичної інформації та консультацій. Вони можуть надавати корисні поради щодо здоров'я, розповідати про симптоми та лікування певних захворювань, нагадувати про прийом ліків або проводити електронний моніторинг показників здоров'я.

Телеграм-боти мають різні типи, залежно від їх функцій та можливостей:

- **Клавіатурні боти:** Це прості боти, які використовують кнопки для навігації та взаємодії з користувачем. Вони можуть мати статичну або динамічну клавіатуру з варіантами вибору для виконання певних дій.
- **Командні боти:** Ці боти реагують на текстові команди, які вводяться користувачем. Вони можуть мати певний набір команд для виконання різних завдань.
- **Інлайн-боти:** Ці боти можуть відповідати на запити користувачів, які вводяться безпосередньо у поле введення повідомлення. Вони можуть шукати інформацію, надавати результати, відповіді або посилання на веб-сторінки.
- **Аналітичні боти:** Ці боти здатні надавати статистику, аналізувати дані та

відстежувати показники в режимі реального часу. Вони можуть відслідковувати відвідування, конверсію, використання ресурсів та інші метрики.

- **Інтеграційні боти:** Ці боти можуть підключатись до інших сервісів або додатків, таких як CRM-системи, соціальні мережі, електронні поштові сервіси тощо. Вони дозволяють передавати дані, сповіщення та виконувати дії через зовнішні API.

Загалом, функціонал і можливості Телеграм-ботів значно розширюються завдяки розробці власного коду та інтеграції з різними сервісами та платформами. Вони можуть бути простими або складними залежно від потреб користувачів і мети.

1.3 Загальний аналіз штучного інтелекту

Штучний інтелект (ШІ) є однією з найцікавіших та найшвидше розвиваючихся галузей в сучасній науці та технологіях. Він вивчає, як забезпечити комп'ютерам здатність думати, вчитися та приймати рішення, схожі на ті, які здатні приймати люди. ШІ має потенціал змінити багато аспектів нашого життя, від медицини та транспорту до розваг та мистецтва.

Одним з ключових елементів ШІ є машинне навчання (Machine Learning). Машинне навчання дозволяє комп'ютерам аналізувати величезні обсяги даних та знаходити закономірності, що дозволяють їм самостійно набувати знання та вдосконалювати свої навички. Це може бути досягнуто за допомогою різних алгоритмів, таких як нейронні мережі або дерева рішень.

Застосування ШІ широкі і різноманітні. У сфері медицини він використовується для діагностики та передбачення захворювань, розробки нових ліків та оптимізації лікування. У сфері транспорту ШІ допомагає розробляти автономні автомобілі та оптимізувати трафік на дорогах. В економіці ШІ допомагає аналізувати та передбачати ринкові тенденції, розробляти інтелектуальні фінансові рішення та управляти ризиками.

Інший важливий аспект ШІ - це розпізнавання образів та обробка природної мови. Завдяки цим технологіям, комп'ютери можуть розуміти та інтерпретувати зображення, а також взаємодіяти з людьми за допомогою мови. Це призводить до створення розумних асистентів, віртуальних помічників та чат-ботів, які можуть відповідати на запитання, виконувати завдання та надавати інформацію.

Однак, на шляху розвитку ШІ є виклики та проблеми. Етичні питання стосовно використання та впливу ШІ на суспільство, приватність та безпека даних, а також безробіття, що можуть виникнути внаслідок автоматизації робочих процесів, потребують серйозного розгляду та регулювання.

Загалом, штучний інтелект є захопливим і перспективним напрямком розвитку технологій. Він має потенціал змінити багато сфер нашого життя та вплинути на спосіб, яким ми працюємо, навчаємося та спілкуємося. Проте, важливо розглядати і розвивати ШІ з етичної та відповідальної перспективи, забезпечуючи безпеку, приватність та рівномірний доступ до цих технологій для всіх.

Методи штучного інтелекту є основними підходами та алгоритмами, що використовуються для створення імітації інтелекту та розуму в комп'ютерних системах. Вони дозволяють комп'ютерам вирішувати складні завдання, навчатися на основі даних та приймати рішення, що раніше були виключно людською прерогативою. Ось деякі з основних методів штучного інтелекту та їх застосування.

- **Машинне навчання (Machine Learning).** Воно базується на ідеї, що комп'ютер може навчитися вирішувати завдання, аналізуючи велику кількість даних. Машинне навчання використовує різні алгоритми, такі як нейронні мережі, дерева рішень та методи кластеризації, для виявлення закономірностей та побудови моделей, які можуть зробити передбачення або прийняти рішення на основі нових даних.
- **Нейронні мережі (Neural Networks).** Вони моделюють роботу людського мозку, використовуючи набір з'єднаних штучних нейронів. Нейронні мережі використовуються для розпізнавання образів, обробки природної мови, розуміння

мови та багатьох інших завдань, які потребують розуміння та інтерпретації інформації.

- Еволюційні алгоритми (Evolutionary Algorithms). Вони базуються на принципах природного відбору та еволюції. Ці алгоритми створюють популяцію індивідів, які потім піддаються мутації, схрещуванню та оцінці залежності від їх придатності. Цей процес дозволяє знаходити оптимальні рішення для складних задач, таких як оптимізація, планування та проектування.
- Логічне програмування (Logic Programming). Воно використовує логічні правила та механізми виведення для моделювання розумової активності. Логічне програмування дозволяє формалізувати знання та відношення між ними, що дозволяє комп'ютеру робити логічні висновки та приймати рішення.
- Deep learning (глибоке навчання) - одна з ключових особливостей deep learning полягає в тому, що шари нейронної мережі вчаться абстрагувати високорівневі функціональні ознаки з низькорівневих даних. Кожен шар пропускає входні дані через нелінійну функцію активації, таку як ReLU (Rectified Linear Unit), і передає вихід до наступного шару. Цей процес продовжується через кілька шарів, дозволяючи мережі розпізнавати складні шаблони та структури у входних даних.

Ці методи штучного інтелекту мають широкий спектр застосувань. Вони використовуються в медицині для діагностики та прогнозування захворювань, в економіці для аналізу ринку та передбачення трендів, в транспорті для розробки автономних систем та у багатьох інших галузях.

Висновок, методи штучного інтелекту є потужними та різноманітними засобами для розв'язання складних задач та моделювання інтелектуальних функцій у комп'ютерних системах. Вони мають широкий спектр застосувань та продовжують розвиватися, що дає можливість для нових досягнень та інновацій у сфері штучного інтелекту.

1.4 Аналіз потреб потенційної цільової аудиторії продукту

Telegram-чат-бот розробляється з метою задоволення потреб різних категорій користувачів. Ось декілька груп з потенційної цільової аудиторії:

- **Маркетологи:** Бот надасть маркетологам можливість отримувати ідеї та поради щодо рекламних рішень, створення логотипів, генерації креативних рішень та аналізу ринку. Вони зможуть звертатися до нього для отримання конкретних інсайтів та підтримки в рекламних проектах.
- **Дизайнери:** Бот стане надійним помічником для дизайнерів, які шукають ідеї, створюють макети, генерують зображення та підбирають кольорову палітру. Він забезпечить доступ до креативних ресурсів та швидку взаємодію для поліпшення процесу дизайну.
- **Консультанти:** Чат-бот, зможе надавати підтримку консультантам, які потребують генерації відповідей для своїх клієнтів, створення бізнес-планів та фінансових моделей. Завдяки йому, консультанти зможуть ефективно та швидко отримувати необхідну інформацію для своїх проектів.
- **Представники медіа та інформаційних професій:** Чат-бот забезпечить швидкий та зручний доступ до різноманітних даних в одному місці. Вони зможуть отримати актуальну інформацію, новини, статистику та інші дані безпосередньо через бота, спрощуючи свою роботу та зекономивши час.

Таким чином, телеграм-чат-бот буде спрямований на надання підтримки та рішень для людей чия робота сфокусована на спілкуванні через чат, пошуку інформації, ідей та порад, написанні текстів, виконанні рутинної роботи на мобільному телефоні чи планшеті, забезпечуючи швидкий та зручний доступ до відповідних інструментів та інформації у єдиному місці.

1.5 Аналіз аналогів

Основним конкурентом для нашого розробляемого чат-бота є ChatGPT, який доступний в браузері. Проаналізувавши можливості ChatGPT, було виявлено ряд ключових моментів, завдяки яким наш чат-бот буде кращим, для людей, чиї завдання спрямовані на комунікацію та виконання рутинних завдань на телефоні, тобто для нашої цільової аудиторії.

Одна з основних переваг нашого розробляемого чат-бота - це зручність використання. Він не вимагає авторизації, що означає, що користувачам не потрібно створювати обліковий запис або вводити свої особисті дані. Це особливо зручно для консультантів різних видів, які в основному працюють з Телеграмом. Вони зможуть мати всю необхідну функціональність і ресурси в одному місці, спрощуючи свою роботу і забезпечуючи більш ефективну комунікацію з клієнтами. Зручна робота з повідомленнями також є важливою перевагою, оскільки користувачі зможуть легко взаємодіяти з чат-ботом і швидко знаходити необхідну інформацію. Крім того, збереження історії повідомлень дозволить переглядати попередні діалоги, що сприяє збереженню контексту та забезпечує зручну навігацію.

Другою важливою перевагою нашого чат-бота є підвищений рівень безпеки. Використання Телеграм для комунікації додає додаткові протоколи шифрування, зокрема MTProto, що допомагає зберегти приватність та конфіденційність інформації. Такий рівень шифрування забезпечує, що відповіді, отримані через наш чат-бот, будуть залишатись безпечними та недоступними для сторонніх осіб. Користувачі можуть бути впевнені, що їх дані та конфіденційна інформація захищені.

Таким чином, наш розробляемий чат-бот пропонує не лише зручність використання, але й підвищений рівень безпеки завдяки інтеграції з Телеграмом. Це робить його привабливим вибором для користувачів, особливо для фахівців, що активно працюють з Телеграмом у своїй повсякденній діяльності.

2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 Огляд мови програмування

C# (C-Sharp) є об'єктно-орієнтованою мовою програмування, розробленою компанією Microsoft. Вона є частиною платформи .NET і надає можливості для розробки різноманітних додатків, включаючи веб-сайти, служби, програми для робочих станцій та ігри.

Однією з головних переваг C# є безпека. Вона надає розробникам можливість контролювати пам'ять і уникнути багатьох типових помилок, пов'язаних з некоректним використанням пам'яті. Це забезпечує стабільність та надійність додатків, особливо важливих для великих проектів.

C# підтримує об'єктно-орієнтований підхід до програмування. Об'єктно-орієнтоване програмування дозволяє розбити програму на окремі модулі (об'єкти), які взаємодіють один з одним. Це полегшує структурування програмного коду та забезпечує легкість утримання та розширення додатків. Крім того, об'єктно-орієнтоване програмування сприяє повторному використанню коду, що полегшує розробку.

Іншою перевагою C# є його синтаксис, схожий на мови програмування Java та C++. Це дозволяє розробникам, які вже володіють цими мовами, легко переходити до використання C#. Також існує багато інструментів та ресурсів, які сприяють навчанню та розробці на C#.

C# має велику кількість корисних функцій, які спрощують розробку програм. Наприклад, автоматичне керування пам'яттю збирачем сміття дозволяє автоматично видаляти непотрібні об'єкти з пам'яті, що полегшує життя розробникам. Крім того, C# підтримує винятки, які дозволяють обробляти виключні ситуації, що виникають під час виконання програми.

Однією з важливих особливостей C# є підтримка делегатів та подій. Делегати дозволяють передавати методи як параметри та повертати їх як результати, що дозволяє створювати гнучкі та розширювані додатки. Події дозволяють реагувати на виникнення певних подій в програмі і виконувати відповідний код.

C# також має багато інструментів для розробки графічного інтерфейсу користувача (GUI). За допомогою Windows Forms або WPF (Windows Presentation Foundation) розробники можуть створювати зручні та естетичні інтерфейси для своїх додатків.

Крім того, C# підтримує розробку веб-додатків за допомогою ASP.NET. ASP.NET дозволяє створювати веб-сторінки, веб-служби та інші компоненти веб-додатків з використанням C#. Завдяки інтеграції з платформою .NET, розробники можуть використовувати всю потужність мови C# для розробки веб-додатків.

Узагальнюючи, C# є потужною мовою програмування, яка надає безпеку, об'єктно-орієнтований підхід, зручний синтаксис та багатий функціонал для розробки різноманітних додатків. Вона є важливим інструментом для розробників, які працюють у середовищі .NET та бажають створити надійні та потужні програми.

2.2 Огляд середовища розробки

Microsoft Visual Studio є однією з найпопулярніших та потужних інтегрованих середовищ розробки (IDE) на ринку. Воно розроблене компанією Microsoft та надає розробникам широкий набір інструментів та можливостей для створення різноманітних програмних продуктів. В цьому рефераті ми розглянемо основні аспекти Microsoft Visual Studio та його роль у сфері програмування.

Microsoft Visual Studio надає розробникам зручне та потужне середовище для розробки програмного забезпечення для різних платформ, включаючи Windows, веб, мобільні пристрої та хмарні сервіси. Воно підтримує різні мови програмування, такі як C#, Visual Basic, C++, F# та інші, що дає можливість розробникам вибрати мову

залежно від їхніх потреб та вподобань.

Одним з ключових компонентів Microsoft Visual Studio є його редактор коду, який надає розширений набір функцій для зручного написання та редагування коду. Редактор підтримує автодоповнення, перевірку синтаксису, налагодження коду та багато інших корисних функцій, що полегшують процес програмування.

Ще одним важливим компонентом Microsoft Visual Studio є система керування версіями, яка дозволяє розробникам ефективно працювати з кодом у командному середовищі. За допомогою інтегрованих інструментів керування версіями, розробники можуть відстежувати зміни в коді, об'єднувати роботу кількох розробників та керувати різними версіями програмного продукту.

Microsoft Visual Studio також надає широкий набір інструментів для налагодження програм. Розробники можуть використовувати відладчик для відстежування та усунення помилок у своєму коді, а також аналізатор профілювання для виявлення проблем продуктивності та оптимізації швидкодії програми.

Одним з найбільш важливих аспектів Microsoft Visual Studio є його інтеграція з іншими сервісами та платформами. Розробники можуть легко підключати сторонні бібліотеки, сервіси хмарного зберігання, системи автоматичної збудови та багато інших інструментів, що спрощують процес розробки та розгортання програмних продуктів.

Крім того, Microsoft Visual Studio надає розробникам можливість створювати різні типи програмних проектів, включаючи консольні додатки, веб-сайти, мобільні додатки, графічні інтерфейси та багато інших. Це дозволяє розробникам працювати в різних сферах програмування та створювати різноманітні програмні рішення. Загалом, Microsoft Visual Studio є потужним інструментом для розробки програмного забезпечення. Воно надає розробникам широкий набір інструментів, функціональність та можливості для створення різноманітних програмних продуктів. Його інтегроване середовище, редактор коду, система керування версіями та інші компоненти роблять процес розробки більш зручним та ефективним.

2.3 Огляд хмарних технологій

Azure є хмарною платформою, розробленою компанією Microsoft, яка надає широкий спектр послуг та інструментів для розробки, розгортання та управління додатками та інфраструктурою. Вона дозволяє організаціям будувати, розширювати та керувати рішеннями в хмарному середовищі шляхом використання віртуальних машин, контейнерів, послуг штучного інтелекту, баз даних та багатьох інших сервісів.

Однією з ключових переваг Azure є його масштабованість та гнучкість. Платформа дозволяє збільшувати або зменшувати ресурси в залежності від потреб, що дозволяє організаціям ефективно використовувати ресурси та оптимізувати витрати. Крім того, Azure пропонує географічну розподіленість датацентрів, що дозволяє розміщувати додатки та дані в найближчому до користувачів регіоні, забезпечуючи швидкість та доступність.

Azure надає широкий вибір послуг для різних сфер діяльності. Наприклад, для розробників доступні послуги розгортання додатків, які дозволяють швидко відокремити додатки від інфраструктури та зосередитися на розробці функціоналу. Для бізнесу Azure пропонує послуги для аналізу даних, машинного навчання, штучного інтелекту та інші рішення, що допомагають вирішувати різні завдання та вдосконалювати бізнес-процеси. Для IT-фахівців доступні інструменти для управління та моніторингу ресурсів, автоматизації процесів та забезпечення безпеки даних.

Окрім того, Azure є екосистемою з багатьма інтегрованими сервісами та інструментами. Наприклад, Azure Active Directory надає засоби для управління ідентифікацією та доступом користувачів до ресурсів, а Azure DevOps дозволяє організувати процес розробки та випуску програмного забезпечення. Ця інтеграція дозволяє розробникам та організаціям ефективно використовувати різні сервіси та інструменти в межах єдиної платформи.

Загалом, Azure є потужною хмарною платформою, яка дозволяє організаціям будувати, розгортати та керувати різними додатками та інфраструктурою. Вона надає широкий спектр послуг та інструментів, масштабованість, гнучкість та інтеграцію з іншими сервісами. Azure стає все більш популярним серед розробників та організацій, які шукають надійне та ефективне рішення для роботи в хмарному середовищі.

Хмарна платформа Azure Function app є одним з ключових сервісів платформи Azure. Вона надає можливість створювати та розгортати функції без потреби управління інфраструктурою, що дозволяє розробникам швидко та ефективно створювати рішення на основі подій та обробки даних.

Azure Function app базується на концепції "функцій як послуги" (Functions-as-a-Service, FaaS), де функції є незалежними одиницями коду, що можуть виконуватися в реакції на певні події або запити. Цей підхід дозволяє зосередитися на розробці функціональності додатків, не вдаючись у деталі управління серверами та інфраструктурою.

Azure Function app надає безліч переваг та можливостей. По-перше, вона підтримує широкий спектр мов програмування, таких як C#, Python, JavaScript і багато інших, що дозволяє розробникам використовувати мову, з якою вони відчують себе комфортно. Крім того, Azure Function app надає гнучкість вибору способу триггеру, таких як таймери, HTTP-запити, зміни у хмарних хранилищах даних та багато інших, що дозволяє функціям реагувати на різні події.

Для полегшення розробки та розгортання Azure Function app надає вбудовані шаблони, які пропонують початкові налаштування та готові функції для різних сценаріїв використання. Це дозволяє розробникам швидко створювати функціональність і адаптувати її до своїх потреб. Крім того, Azure Function app підтримує інтеграцію з іншими сервісами Azure, що дозволяє створювати комплексні рішення, використовуючи весь потенціал хмарної платформи.

Ще однією важливою перевагою Azure Function app є масштабованість. Вона автоматично масштабує функції в залежності від навантаження, що дозволяє ефективно використовувати ресурси і забезпечувати високу продуктивність додатків. Крім того, платформа Azure Function app надає можливість моніторингу та логування функцій, що дозволяє виявляти та вирішувати проблеми ефективно.

Azure Storage Account Table є іншим сервісом, наданих платформою Azure, який надає можливість зберігати структуровані дані у вигляді таблиць. Цей сервіс є економічним і масштабованим рішенням для зберігання великого обсягу даних, особливо тих, що мають просту структуру та потребують швидкого доступу.

Azure Storage Account Table забезпечує безліч переваг для зберігання даних. Перш за все, він пропонує безпечне та надійне зберігання, оскільки дані реплікуються автоматично для забезпечення високої доступності. Крім того, дані зберігаються у вигляді сутностей (entities), що дозволяє використовувати їх як ключ-значення для швидкого доступу до інформації.

Один з головних аспектів Azure Storage Account Table - його масштабованість. Він може легко масштабуватися для обробки великих обсягів даних та великої кількості транзакцій. Також він підтримує різні рівні консистентності, що дозволяє вибрати найбільш підходящий рівень для потреб додатка.

Azure Storage Account Table надає багато функцій для роботи з даними. Він дозволяє використовувати фільтри та проекції для отримання лише необхідних даних з таблиці. Також він підтримує розподілені операції, що дозволяє обробляти дані паралельно та забезпечує високу продуктивність.

Одним з ключових аспектів Azure Storage Account Table є його інтеграція з іншими сервісами Azure. Він може використовуватися разом зі службами обробки подій (Event Grid), обчислення (Azure Functions, Azure Logic Apps) та багатьма іншими. Це дозволяє створювати складні рішення, що використовують різні сервіси Azure для роботи з даними.

Узагалі, Azure Storage Account Table є потужним інструментом для зберігання структурованих даних в середовищі Azure. Він надає безпечне, масштабоване та продуктивне зберігання для великого обсягу даних. З його допомогою розробники можуть ефективно працювати зі структурованими даними, використовувати їх у своїх додатках та інтегрувати з іншими сервісами Azure для створення розширених рішень.

2.4 Огляд інтерфейсу прикладного програмування OpenAI(API)

В останні роки штучний інтелект став невід'ємною частиною нашого життя, а його застосування в різних сферах знаходить все більше і більше популярності. OpenAI API є одним з провідних інструментів у цій галузі і пропонує широкий спектр можливостей для розробників та дослідників.

OpenAI API використовує модель глибокого навчання GPT (Generative Pre-trained Transformer), яка забезпечує здатність розуміти та генерувати людську мову з вражаючою точністю та креативністю. Ця модель навчається на величезних обсягах текстових даних і може бути застосована для вирішення різноманітних задач, включаючи генерацію тексту, відповіді на запити, переклад, створення контенту та багато іншого.

Однією з ключових переваг OpenAI API є його доступність та простота використання. Завдяки API можна легко інтегрувати його функціональність у власні додатки, сайти або сервіси. Запити до API можуть бути зроблені за допомогою HTTP-запитів, що робить його доступним для широкого кола розробників.

Однією з важливих можливостей OpenAI API є здатність до створення контенту. За допомогою GPT моделі можна згенерувати тексти, статті, відповіді на запити та багато іншого. Це дозволяє автоматизувати процес створення контенту та забезпечує великий потенціал для маркетингових та комунікаційних рішень.

Крім того, OpenAI API дозволяє розробникам створювати персоналізовані моделі штучного інтелекту, навчаючи їх на власних даних. Це дає можливість створювати моделі, які спеціалізовані на конкретні завдання та відповідають потребам конкретних проектів або компаній.

Однак, варто враховувати певні виклики та обмеження OpenAI API. Воно може бути витратним з точки зору використання ресурсів та оплати. Також, існує потенціал виникнення етичних питань та проблем з безпекою при використанні штучного інтелекту. Тому важливо зберігати баланс і забезпечувати відповідальне використання цих технологій.

У підсумку, OpenAI API є потужним інструментом у сфері штучного інтелекту, який відкриває нові можливості для розробників та дослідників. Він забезпечує широкий функціонал та простоту використання, що дозволяє реалізовувати інноваційні проекти та розвивати нові ідеї. Проте, важливо враховувати етичні аспекти та обмеження, щоб забезпечити безпеку та відповідальне використання цих технологій.

2.5 Огляд інтерфейсу прикладного програмування Telegram(API)

Telegram Bot API (Application Programming Interface) є набором інструментів, який надається платформою Telegram для розробки та інтеграції Телеграм-ботів з іншими додатками та сервісами. Цей API дозволяє розробникам створювати різноманітні функціональність для ботів, такі як взаємодія з користувачами, обробка повідомлень, розсилка повідомлень, управління групами та каналами, доступ до даних користувачів і багато іншого.

Основні можливості Telegram Bot API включають:

- Отримання оновлень: Розробники можуть створювати ботів, які будуть отримувати оновлення з Telegram, такі як нові повідомлення, запити клавіатури, вступи до групи тощо. Це дозволяє ботам реагувати на взаємодію користувачів.

- Відправка повідомлень: Боти можуть відправляти текстові повідомлення, зображення, відео, аудіо, стікери, файлові документи та інші медіафайли користувачам або в специфічні групи, канали.
- Обробка команд: Боти можуть розпізнавати команди, введені користувачами, і відповідати на них відповідним чином. Наприклад, бот може мати команду `"/погода"`, щоб надати поточні метеорологічні дані.
- Робота з клавіатурами: Боти можуть створювати клавіатури з кнопками, що дозволяє користувачам взаємодіяти з ботом, натискаючи на кнопки замість введення тексту. Це зручно для створення інтерактивних ботів.
- Управління групами та каналами: Боти можуть мати доступ до адміністративних функцій для керування групами та каналами, таких як додавання користувачів, видалення повідомлень, зміна налаштувань тощо.
- Робота зі зображеннями та медіа: Telegram Bot API надає можливість обробляти зображення та медіафайли, зокрема змінювати розміри, налаштовувати якість, отримувати ескізи та інші операції.

Telegram Bot API підтримується документацією, яка надає розробникам інструкції та приклади коду для використання API. Розробники можуть використовувати це API для створення різноманітних Телеграм-ботів залежно від своїх потреб та вимог користувачів.

3 РЕАЛІЗАЦІЯ ЧАТ-БОТУ З ШТУЧНИМ ІНТЕЛЕКТОМ

3.1 Проектування та розробка діаграми варіантів використання

Діаграма використання є візуальним засобом представлення способу взаємодії між користувачем та системою. Ця діаграма відображає залежності між користувачем та різними сценаріями використання, в яких користувач бере участь. Вона допомагає ідентифікувати різні типи користувачів системи та відображає різні сценарії використання. Часто діаграма використання супроводжується іншими типами діаграм, що дозволяє більш повно та зрозуміло описати взаємодію системи з користувачами.

Варіанти використання на діаграмах можуть бути представлені у вигляді кругів або еліпсів. Хоча кожен випадок використання може бути детально розглянутий окремо, діаграма випадків використання надає загальний огляд системи на більш високому рівні. Як вже було зазначено раніше, схеми використання є принципами вашої системи.

Завдяки своєму простому характеру, схеми використання є ефективним засобом комунікації з зацікавленими сторонами. Вони створюють уявлення про те, як система буде реалізована, і намагаються відтворити реальний світ. В дослідженні سیا та Лі було виявлено, що діаграми випадків використання передають наміри системи більш простим способом зацікавленим сторонам, і вони сприймаються більш повністю, ніж діаграми класів.

Метою діаграми використання є відображення динамічного аспекту системи. Додаткові схеми та документація можуть використовуватися для забезпечення повного функціонального та технічного уявлення про систему. Вони надають просте та графічне представлення того, що система повинна виконувати насправді.

Елементи діаграми використання:

- Рамки системи (англ. system border) - це прямокутник з назвою, розташованою

у верхній частині, і еліпсами (прецедентами) всередині. Іноді їх можна опустити, якщо вони не містять корисної інформації про систему.;

- Актор - це персонаж, який створюється для позначення набору ролей користувача. Він може представляти людину, зовнішню сутність, клас або іншу систему та взаємодіяти з деякою сутністю, такою як система, підсистема чи клас. Актори не можуть бути пов'язані між собою з іншими взаєминами, крім взаємодії з обробкою або дослідженням.

- Прецедент - це еліпс з надписом, що описує систематичну дію, яку виконує система і яка має спостережувані результати для акторів. Надпис може містити ім'я або опис того, що система робить з точки зору актора, а не як це відбувається. Один прецедент може мати кілька можливих варіантів, які пов'язані з атомарним сценарієм - конкретною послідовністю дій, яка ілюструє взаємодію акторів з системою. Під час сценарію актори обмінюються систематичними повідомленнями. Сценарій може бути представлений на діаграмі прецедентів у відео UML-коментарі. Один прецедент може мати кілька різних сценаріїв.

На рисунку 3.1 зображено діаграму варіантів використання, яка описує можливі дії користувача в системі.



Рисунок 3.1 — Діаграма варіантів використання користувача

3.2 Проектування та розробка структурна діаграма розгортання

Структурна діаграма розгортання (англ. Deployment Diagram) - це тип UML-діаграми, яка моделює фізичне розташування компонентів системи та їх взаємозв'язки на різних обчислювальних вузлах або середовищах.

Ця діаграма надає візуальне представлення апаратного та програмного забезпечення, яке використовується для реалізації системи, а також зв'язків між ними.

На структурній діаграмі розгортання можуть бути відображені такі елементи:

- Вузли: представляють фізичні обчислювальні пристрої, такі як сервери, комп'ютери або мобільні пристрої, на яких розгортається програмне забезпечення.
- Артефакти: це фізичні об'єкти, які розгортаються на вузлах, наприклад, виконувані файли, конфігураційні файли, бібліотеки тощо.

- Зв'язки: показують залежності та комунікацію між компонентами, які знаходяться на різних вузлах.
- Конфігурації: визначають набір компонентів, які розгортаються разом на вузлі.

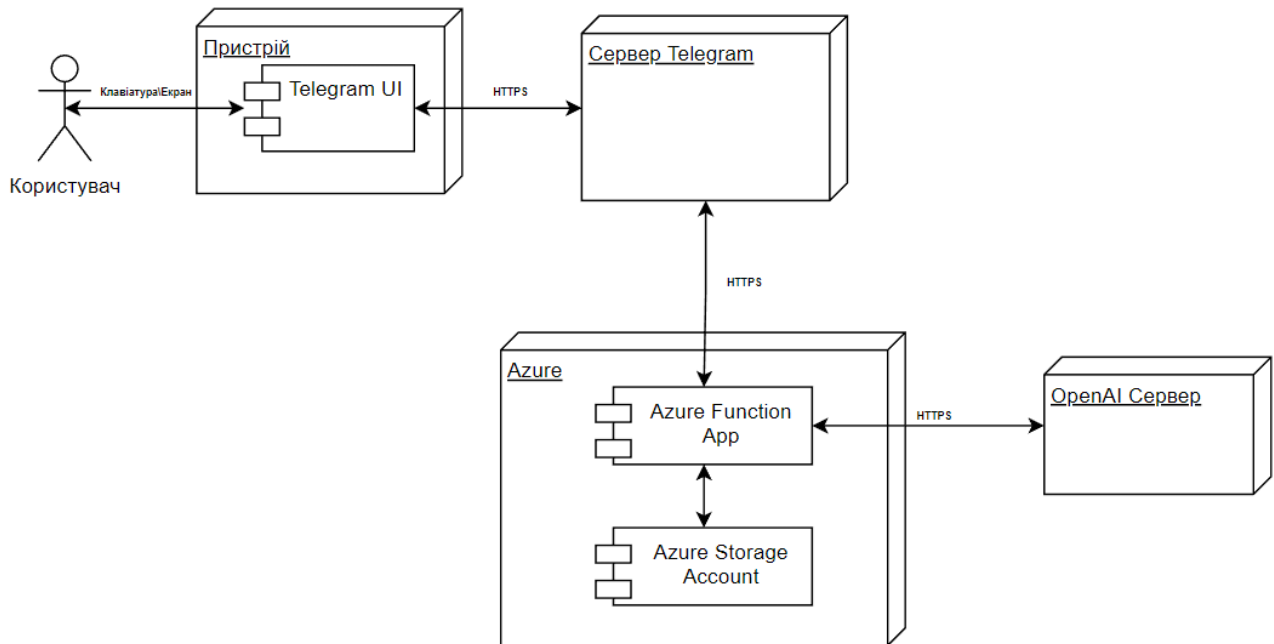


Рисунок 3.2 — Структурна діаграма розгортання

3.3 Проектування та розробка діаграма класів

Діаграма класів є одним з основних інструментів для моделювання структури об'єктно-орієнтованої програми. Вона відображає класи програми та зв'язки між ними. Класи представляють собою шаблони або формальні визначення об'єктів, які мають спільні властивості та поведінку.

Діаграма класів складається з двох основних елементів: класів і зв'язків. Класи зображуються у прямокутниках і містять назви класів, атрибути та методи. Зв'язки показують, які класи взаємодіють між собою та які взаємодії між ними існують. Основні типи зв'язків, які можна використовувати в діаграмі класів, включають:

- Асоціація: зв'язок між двома класами, який показує, що один клас використовує інший. Асоціація може мати напрямок (один клас залежить від іншого) або бути двосторонньою (взаємна залежність).
- Агрегація: відносно слабкий зв'язок між класами, де один клас представляє контейнер або "частину", яка може містити інші класи. Це вказує на "частина-ціле" відношення.
- Композиція: сильний зв'язок між класами, який вказує, що один клас є частиною іншого класу. Вона вказує на "частина-ціле" відношення, але з більш жорстким зв'язком, оскільки частина не може існувати окремо від цілого.
- Спадкування (наслідування): відношення між класами, де один клас успадковує властивості та методи іншого класу. У діаграмі класів стрілка біжучою лінією з пустим кінцем показує напрямок спадкування.
- Реалізація (імплементация): зв'язок між класами, де один клас реалізує абстрактні методи, визначені в іншому класі або інтерфейсі.

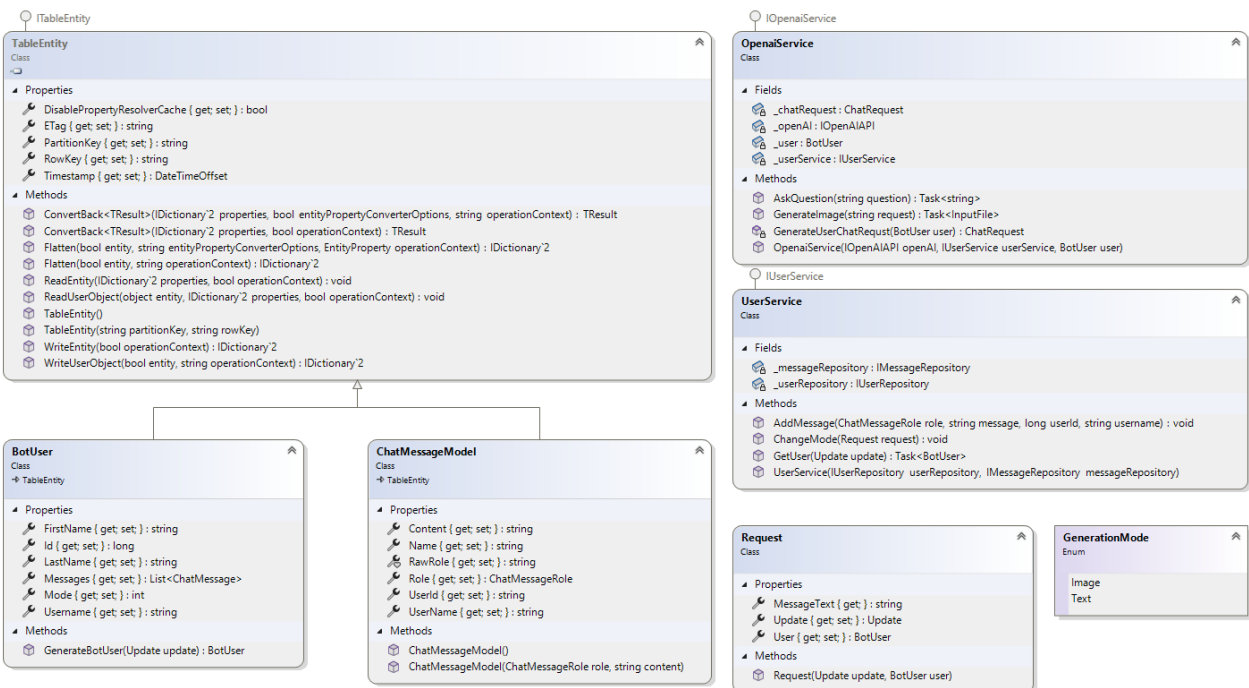


Рисунок 3.3 — Діаграма класів

4. ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування програмного забезпечення є важливою складовою процесу розробки програмного продукту. Його метою є перевірка якості, правильності функціонування та виявлення помилок і дефектів в програмному забезпеченні перед його впровадженням. Тестування допомагає забезпечити, що програмний продукт відповідає вимогам замовника і працює безперебійно.

Процес тестування програмного забезпечення починається з розробки тестових планів і тестової документації, що визначають стратегію тестування і набір тестових сценаріїв. Тестові сценарії описують послідовність дій, які виконуються для перевірки різних аспектів програмного продукту. Це можуть бути тестування функцій, продуктивності, безпеки, сумісності тощо.

У процесі тестування використовуються різні методи та підходи. Один з них - модульне тестування, яке зосереджується на перевірці окремих модулів програми. Модульні тести дозволяють виявити проблеми на ранніх етапах розробки та сприяють поліпшенню якості коду.

Інший підхід - інтеграційне тестування, яке перевіряє взаємодію між різними компонентами програмного забезпечення. Це допомагає виявити проблеми, що можуть виникнути при інтеграції різних модулів.

Окремий аспект тестування - функціональне тестування. Воно перевіряє, чи виконує програмне забезпечення функції, для яких воно призначене. Функціональні тести орієнтовані на перевірку конкретних можливостей програмного продукту та його відповідності вимогам.

Тестування також може включати тестування продуктивності, що оцінює роботу програми за різних навантажень та обсягів даних, а також тестування безпеки для виявлення потенційних уразливостей і захисту від зловмисників.

Одним з важливих етапів тестування є документування результатів. Після проведення тестів фіксуються знайдені дефекти та проблеми, що потребують

виправлення. Це допомагає розробникам виправити помилки та покращити якість програмного продукту.

Тестування програмного забезпечення відбувається на кількох етапах розробки, включаючи модульне тестування, інтеграційне тестування, системне тестування та приймальне тестування. Кожен з цих етапів має свої особливості та цілі, але загальною метою є забезпечення якості і надійності програмного продукту.

У підсумку, тестування програмного забезпечення є невід'ємною частиною процесу розробки програмного продукту. Воно допомагає виявляти дефекти, покращувати якість та забезпечувати надійність програмного забезпечення. Ефективне тестування дозволяє забезпечити задоволення потреб користувачів і досягнення бізнес-цілей

4.1 Модульне тестування

Почати процес тестування з модульного тестування має кілька переваг. Воно дозволяє розробникам перевірити правильність роботи окремих модулів програми в ізоляції від інших компонентів. Це дає можливість швидко виявити й виправити помилки, що можуть виникнути в конкретних модулях, замість того, щоб виявляти їх на пізніших етапах розробки, коли вже інтегровані багато компонентів.

Модульне тестування зазвичай здійснюється за допомогою спеціальних тестових фреймворків або бібліотек, які допомагають створювати тестові сценарії і виконувати їх автоматично. Для наших потреб був обраний NUnit як найбільш популярний та практичний фреймворк для написання тестів мовою C#.

Модульне тестування дозволяє ефективно ідентифікувати проблеми в окремих модулях, що полегшує їх виправлення і покращення якості коду. Крім того, модульні тести допомагають забезпечити стабільність системи, оскільки вони перевіряють роботу найменших фрагментів програми, що є основою для подальшої роботи системи в цілому.

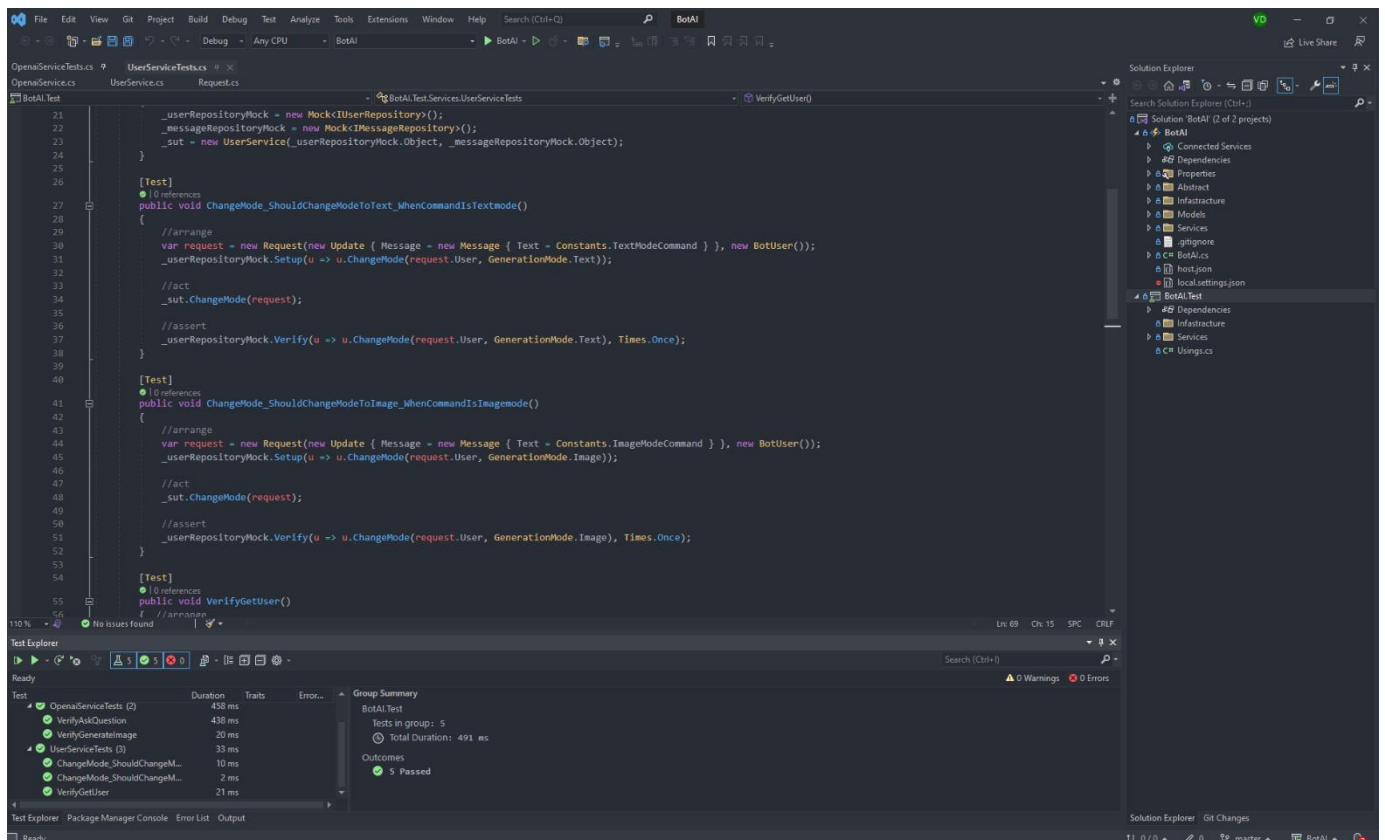


Рисунок 3.3 — Результати модульних тестів які покривають головний функціонал

Після завершення модульного тестування, наступним кроком у процесі тестування програмного забезпечення є перевірка функціональних кейсів. Цей етап передбачає перевірку роботи програми з урахуванням функціональних вимог, які визначені в специфікаціях або вимогах до системи. Функціональні кейси є сценаріями, які описують очікувані вхідні дані, дії, які потрібно виконати, та очікувані результати для конкретної функціональності програмного забезпечення.

4.2 Функціональне тестування

Функціональне тестування є одним із видів тестування програмного забезпечення, яке спрямоване на перевірку функцій та функціональних можливостей програмного продукту. Його основна мета полягає в перевірці, чи працюють окремі функції програми згідно з очікуваннями та вимогами, визначеними в специфікаціях або вимогах до програмного забезпечення.

Функціональний тест-кейс «Перевірка текстового використання бота»

Таблиця 4.1. Кроки тестування для тест-кейсу «Перевірка текстового використання бота»

Номер	Дія	Рисунок
1	Запустити бота	Рисунок 3.3
2	Перевірити, що бот успішно запуснений і готовий до взаємодії	Рисунок 3.5
3	Ввести запит «створи функцію на C# яка буде сумувати 2 числа» до бота	Рисунок 3.6
4	Перевірити, що бот коректно обробляє введений запит	Рисунок 3.6
5	Очікувати відповідь від бота	Рисунок 3.6

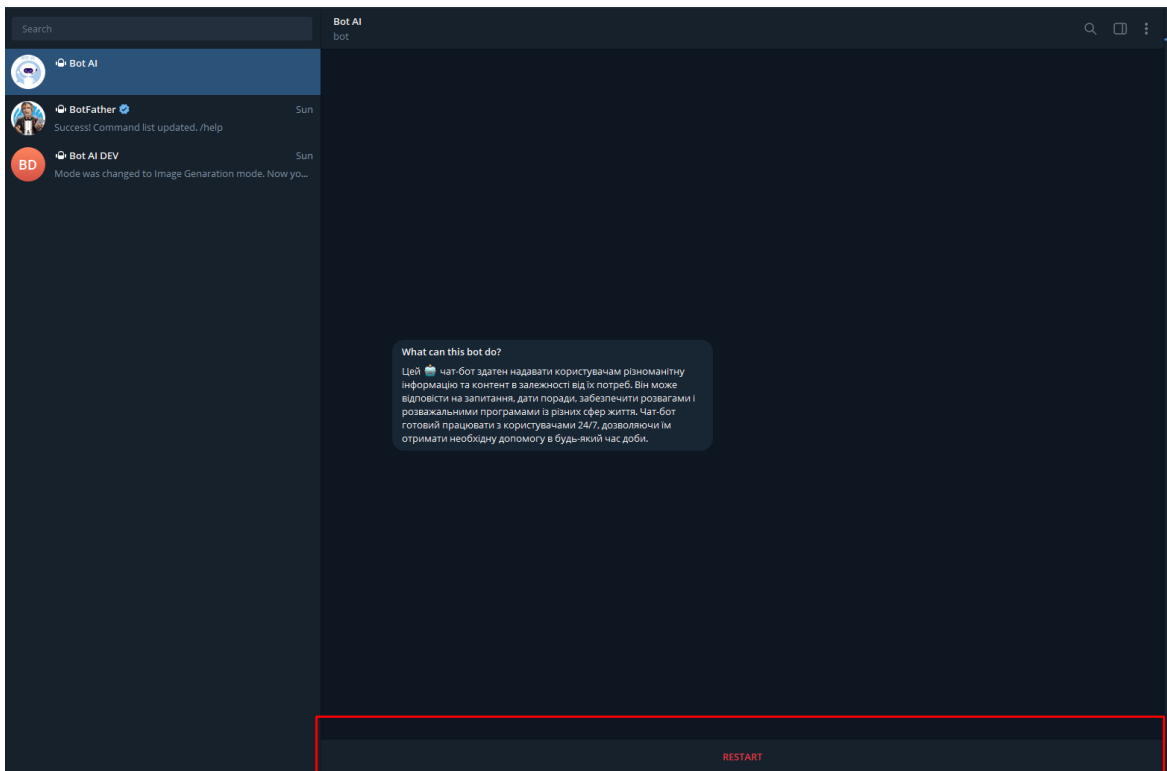


Рисунок 3.4 — Заспуску бота.

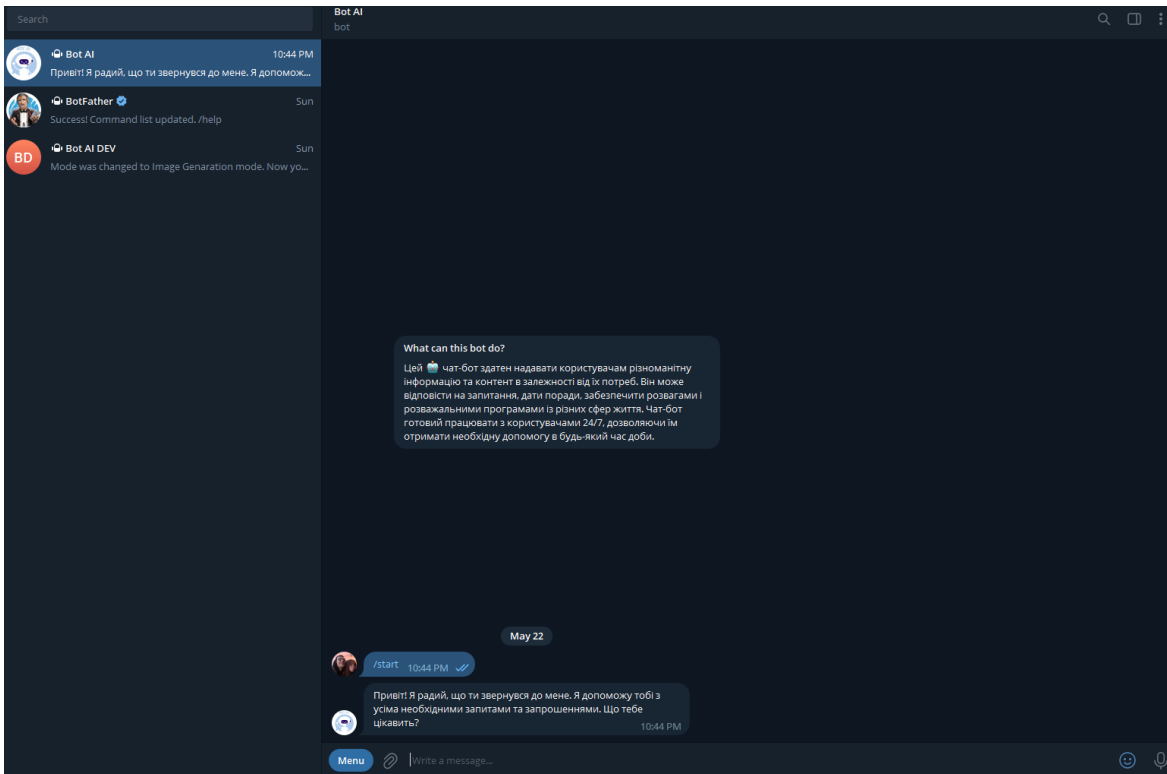


Рисунок 3.5 — Бот успішно запущений і готовий до взаємодії.

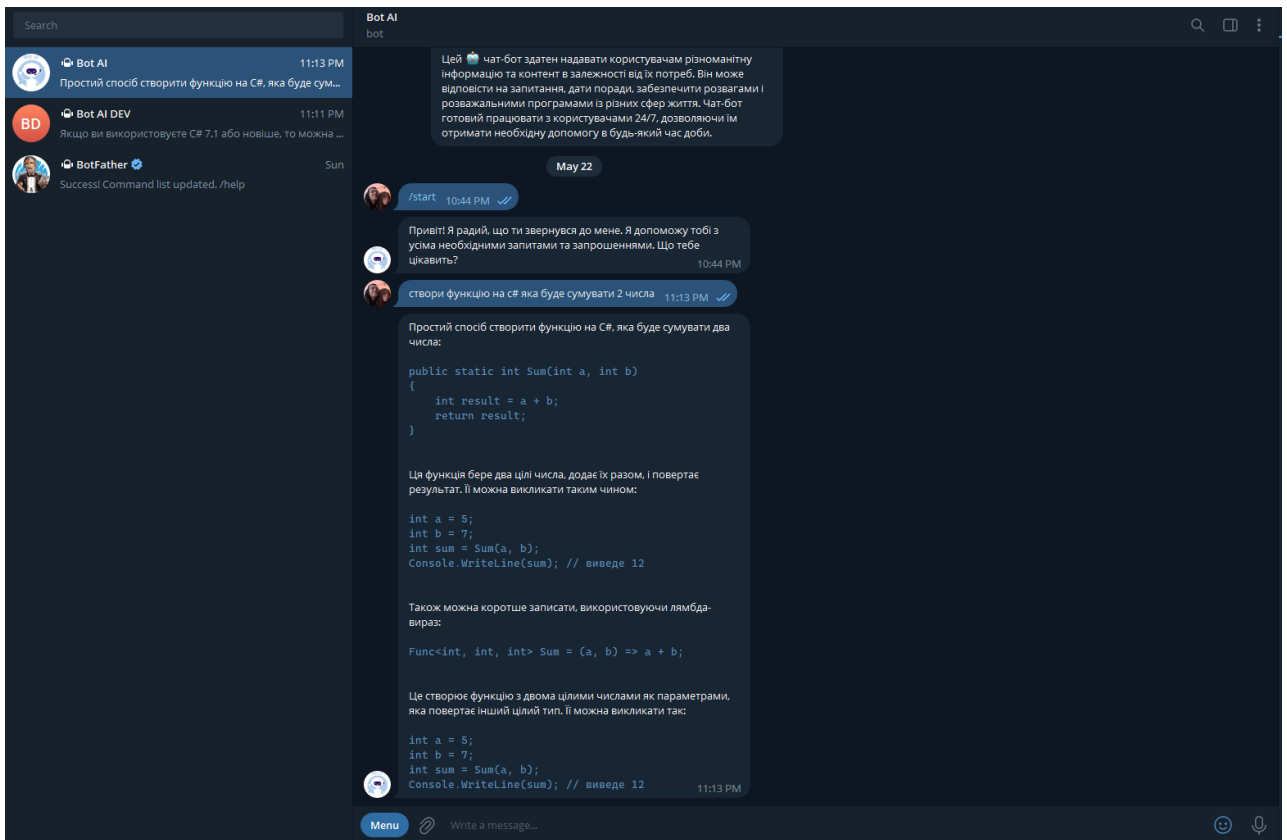


Рисунок 3.6 — Запит до бота.

Функціональний тест-кейс «Перевірка використання бота для генерації зображень»

Таблиця 4.2. Кроки тестування для тест-кейсу «Перевірка використання бота для генерації зображень»

Номер	Дія	Рисунок
1	Запустити бота.	Рисунок 3.7
2	Перевірити, що бот успішно запущений і готовий до взаємодії.	Рисунок 3.8

3	Натиснути кнопку “Menu”.	Рисунок 3.9
4	Натиснути кнопку “/imagemode”.	Рисунок 3.10
5	Ввести запит на генерацію зображення.	Рисунок 3.11
6	Очікувати відповідь від бота.	Рисунок 3.11

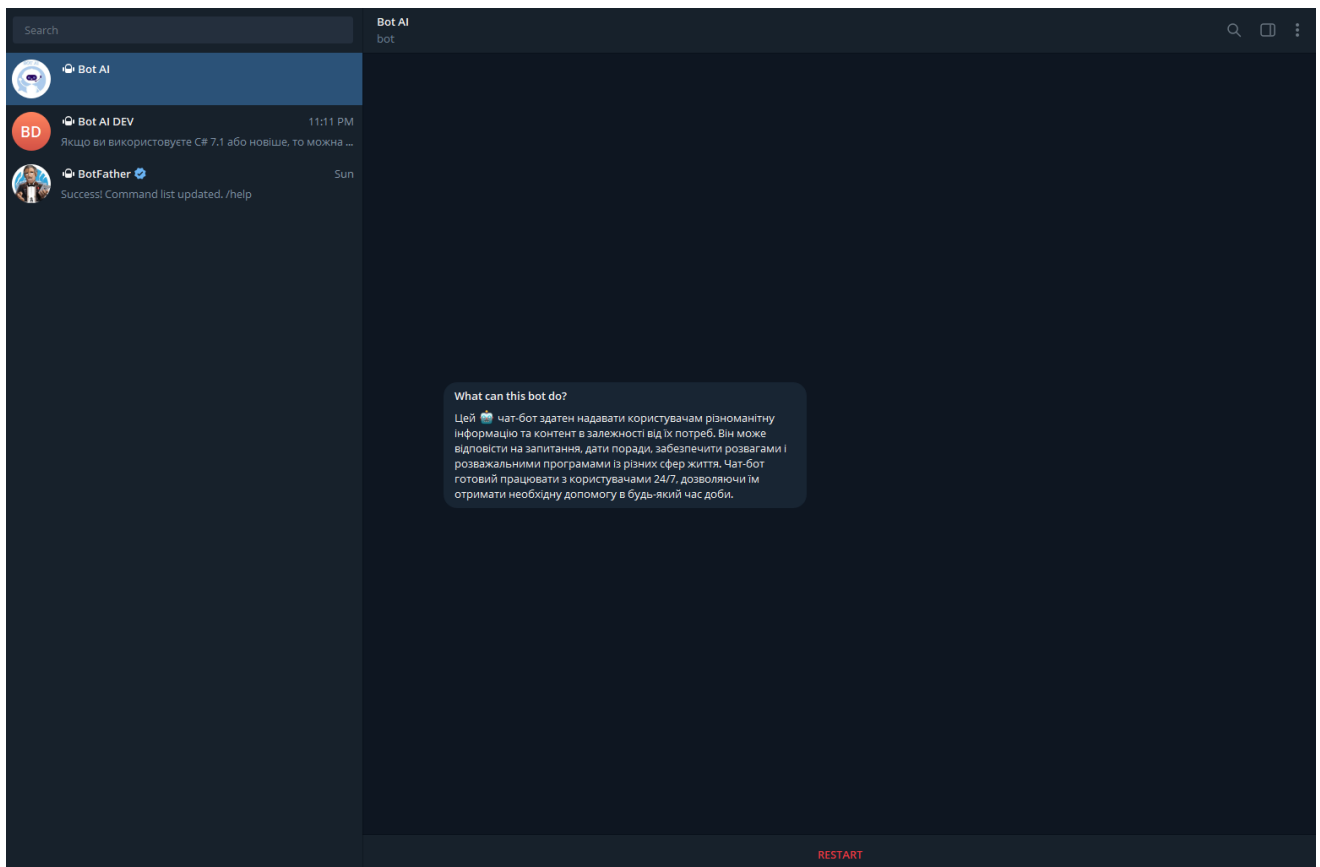


Рисунок. 3.7 — Запуск бота.

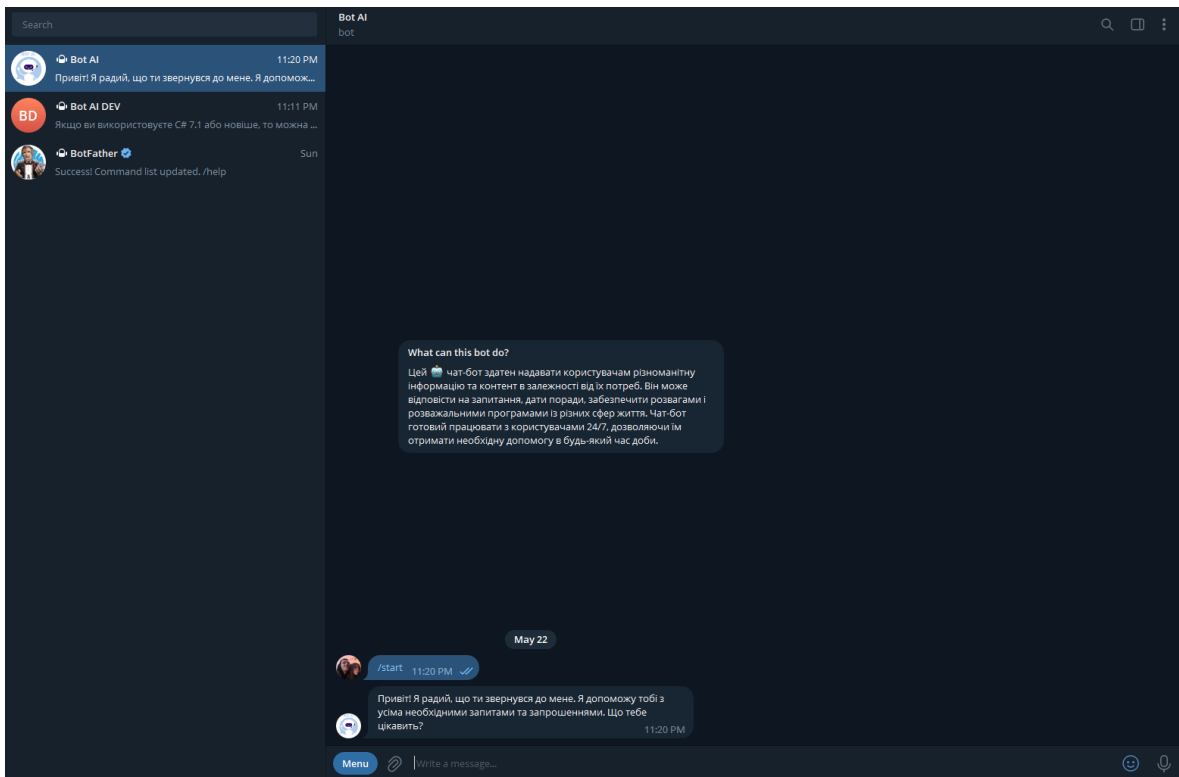


Рисунок. 3.8 — Бот успішно запущений і готовий до взаємодії.

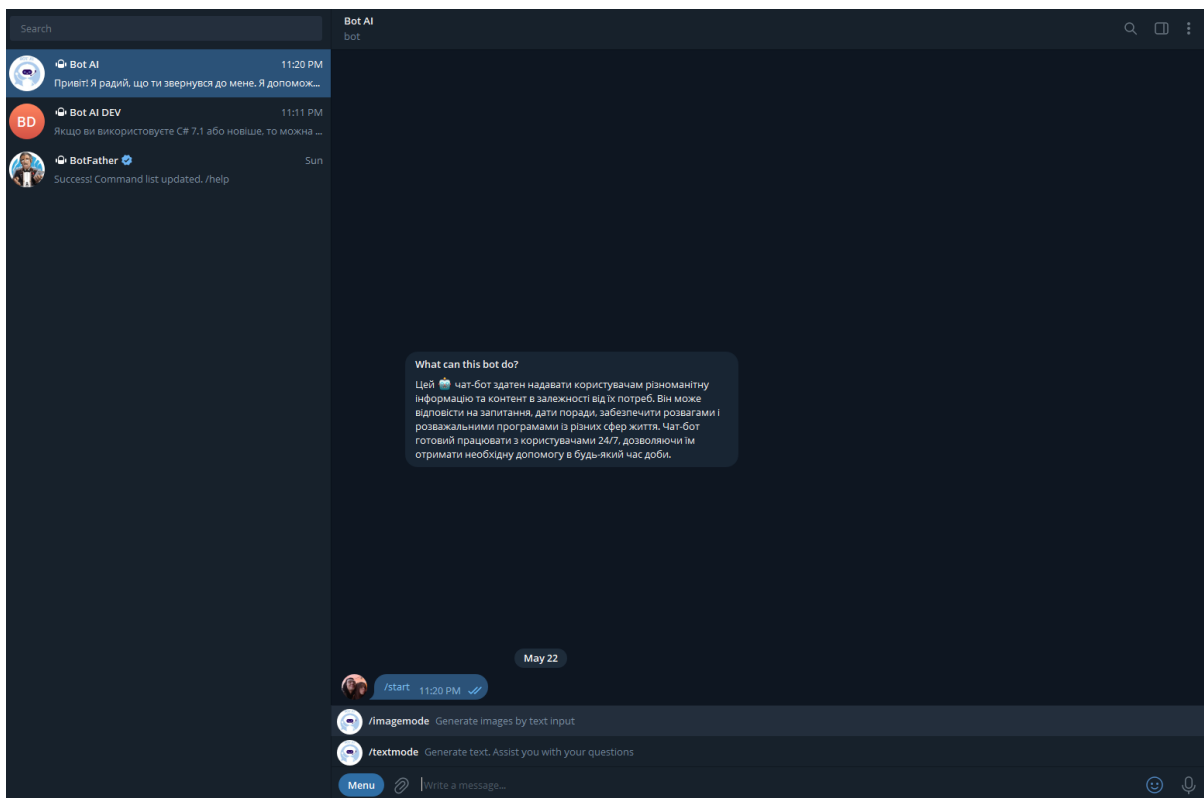


Рисунок.3.9 — Кнопка “Menu” натиснута.

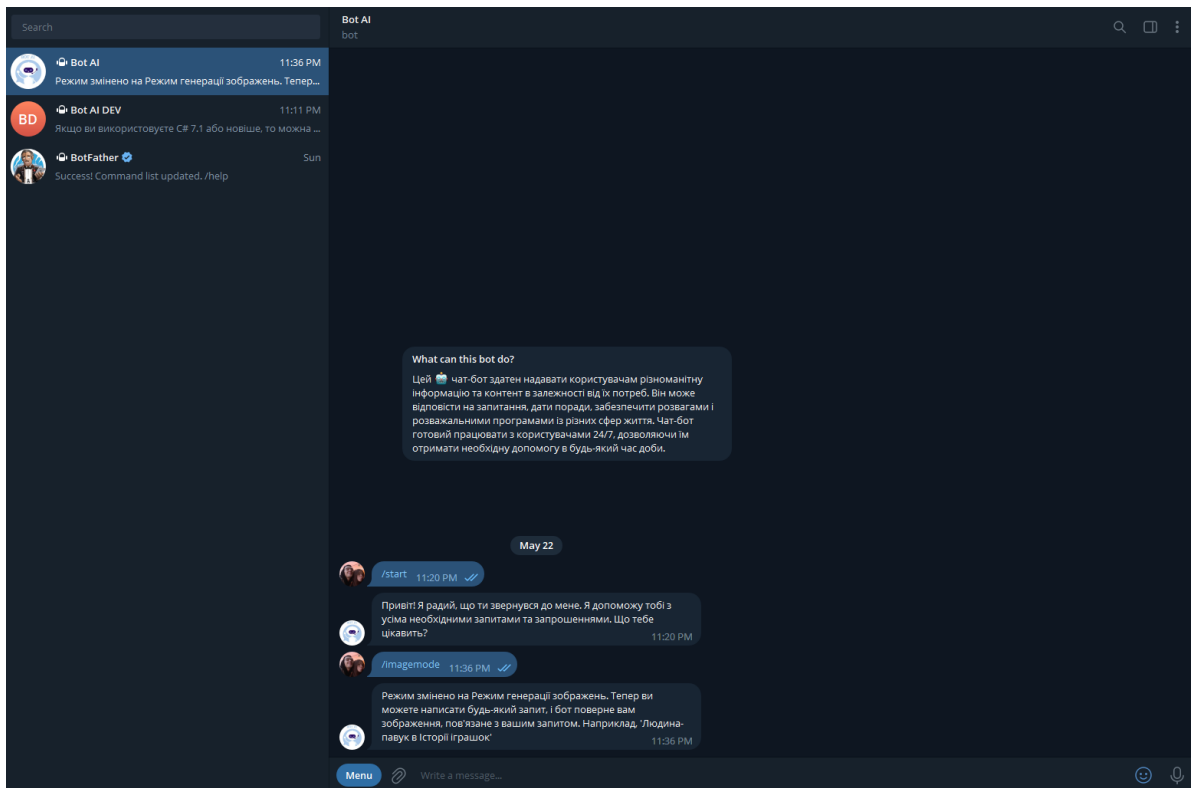


Рисунок 3.10 — Кнопка “/imagemode” натиснута.

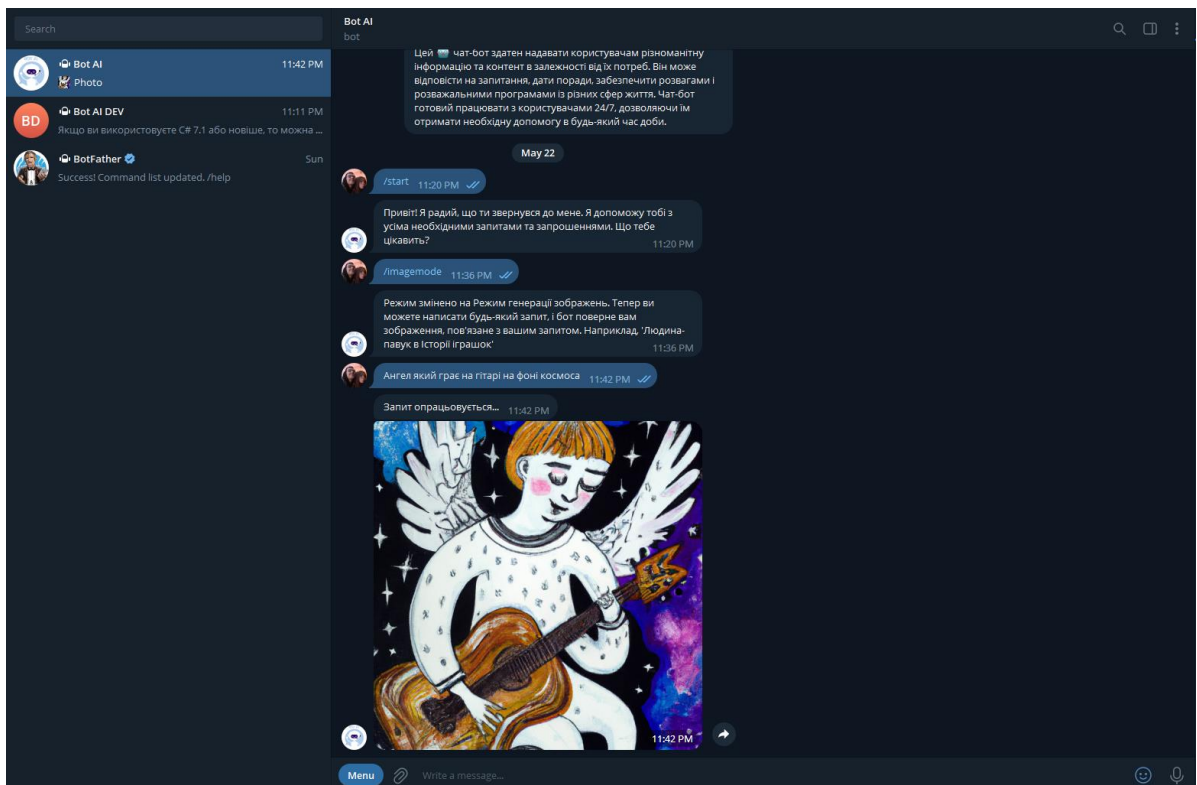


Рисунок 3.11 — Відповідь бота.

Функціональний тест-кейс «Перевірка бота зміна режимів»

Таблиця 4.3. Кроки тестування для тест-кейсу «Перевірка бота зміна режимів»

Номер	Дія	Рисунок
1	Запустити бота.	Рисунок 3.12
2	Перевірити, що бот успішно запущений і готовий до взаємодії.	Рисунок 3.13
3	Натиснути кнопку “Menu”.	Рисунок 3.14
4	Натиснути кнопку “/imagemode”.	Рисунок 3.14
5	Ввести запит на генерацію зображення.	Рисунок 3.15
6	Очікувати відповідь від бота.	Рисунок 3.15
7	Натиснути кнопку “Menu”.	Рисунок 3.16
8	Натиснути кнопку “/textemode”	Рисунок 3.16
9	Очікувати відповідь від бота	Рисунок 3.17
10	Ввести запит до бота	Рисунок 3.18
11	Очікувати відповідь від бота.	Рисунок 3.18

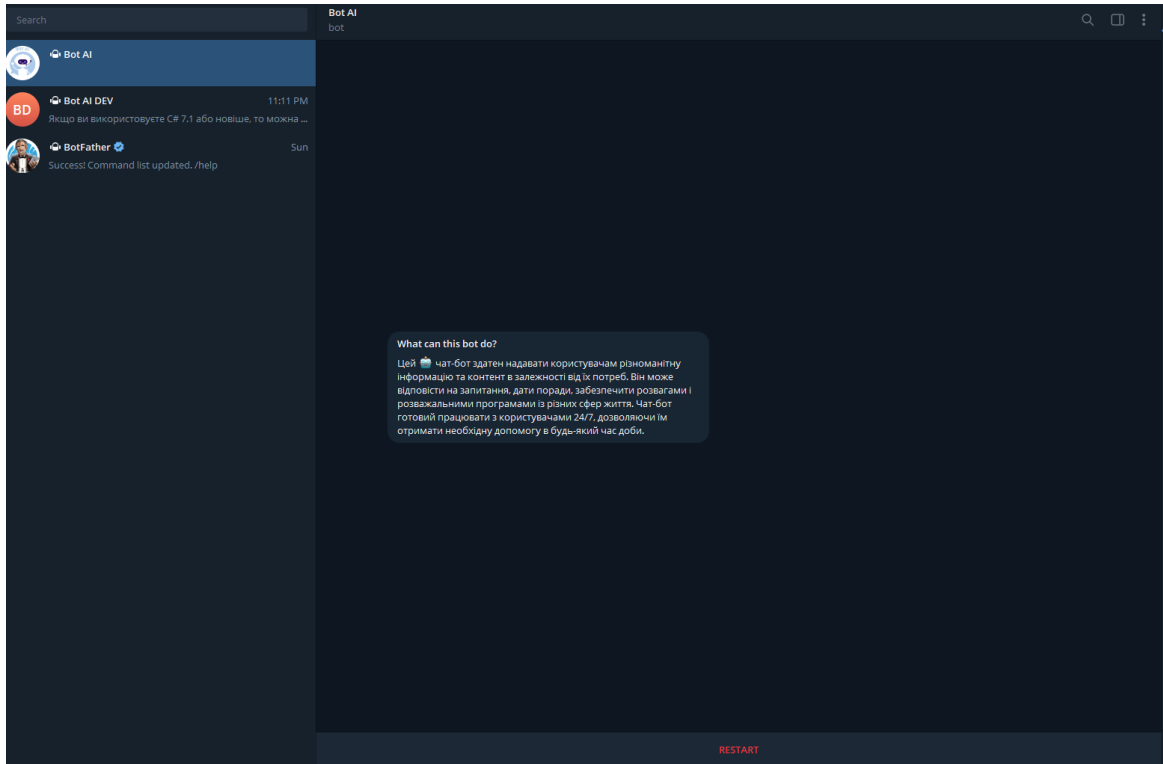


Рисунок 3.12 — Запуск бота.

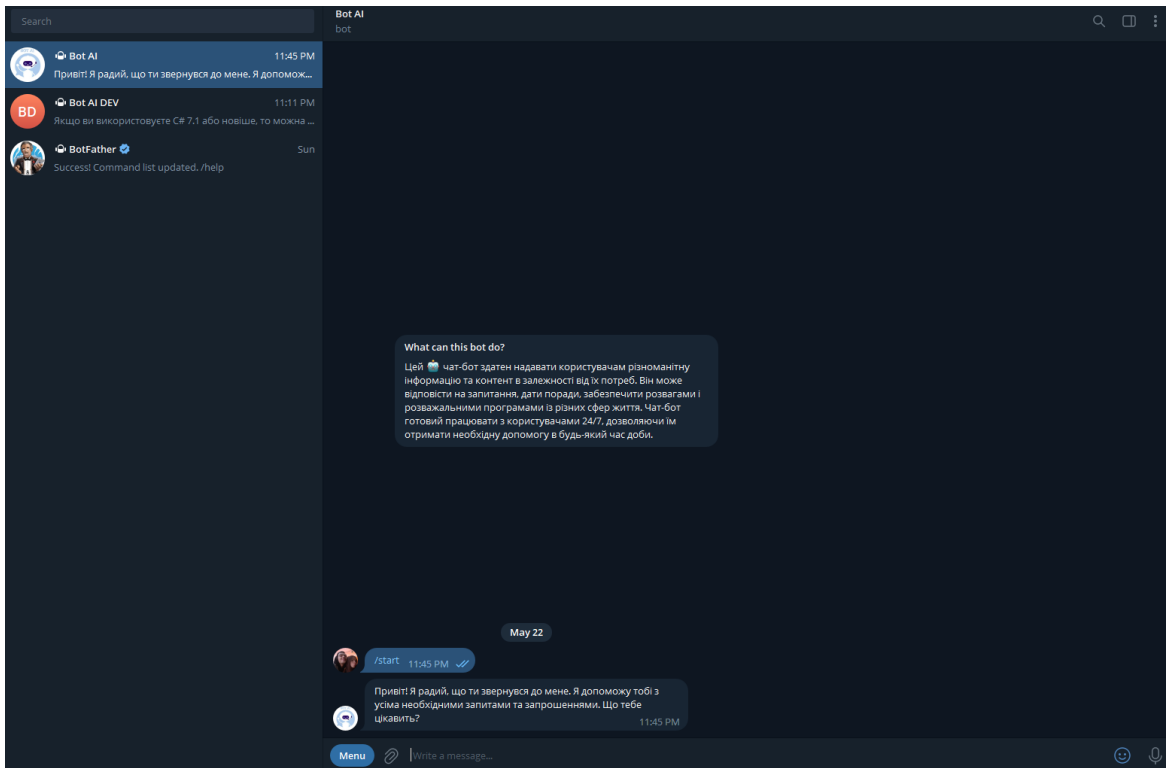


Рисунок. 3.13 — Бот успішно запущений і готовий до взаємодії.

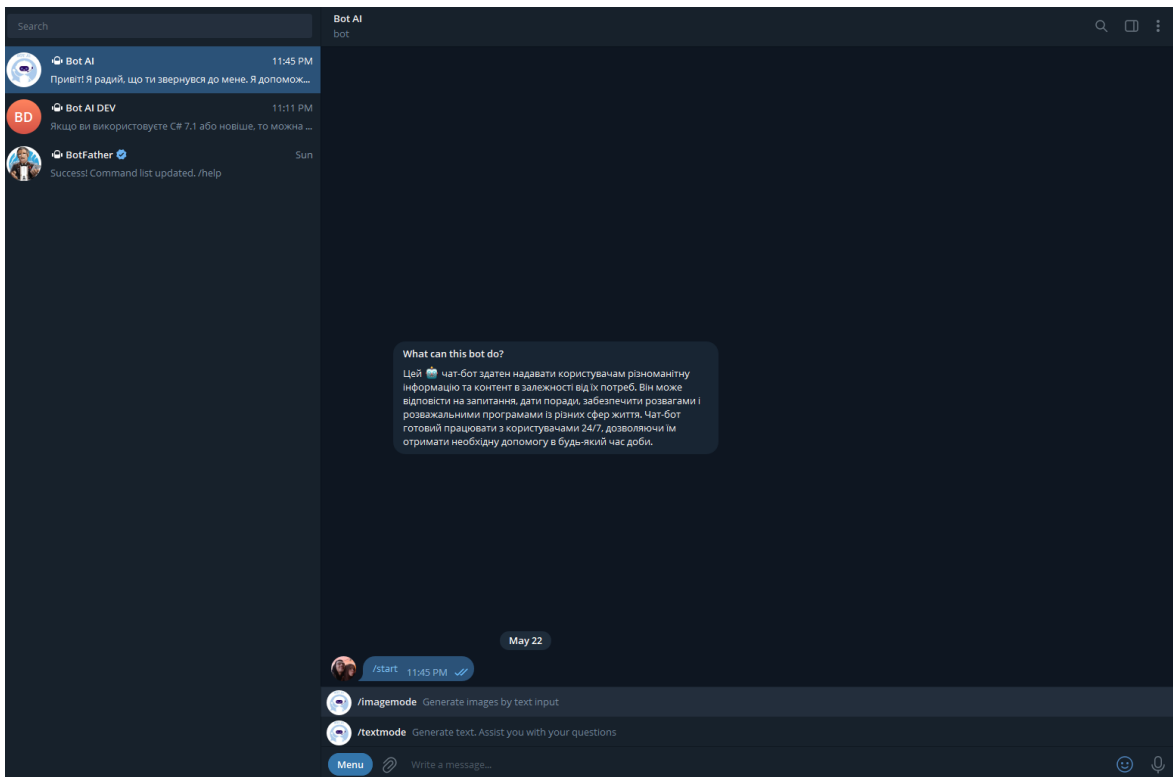


Рисунок. 3.14 — Натиснути “Menu” та “/imagemode”.

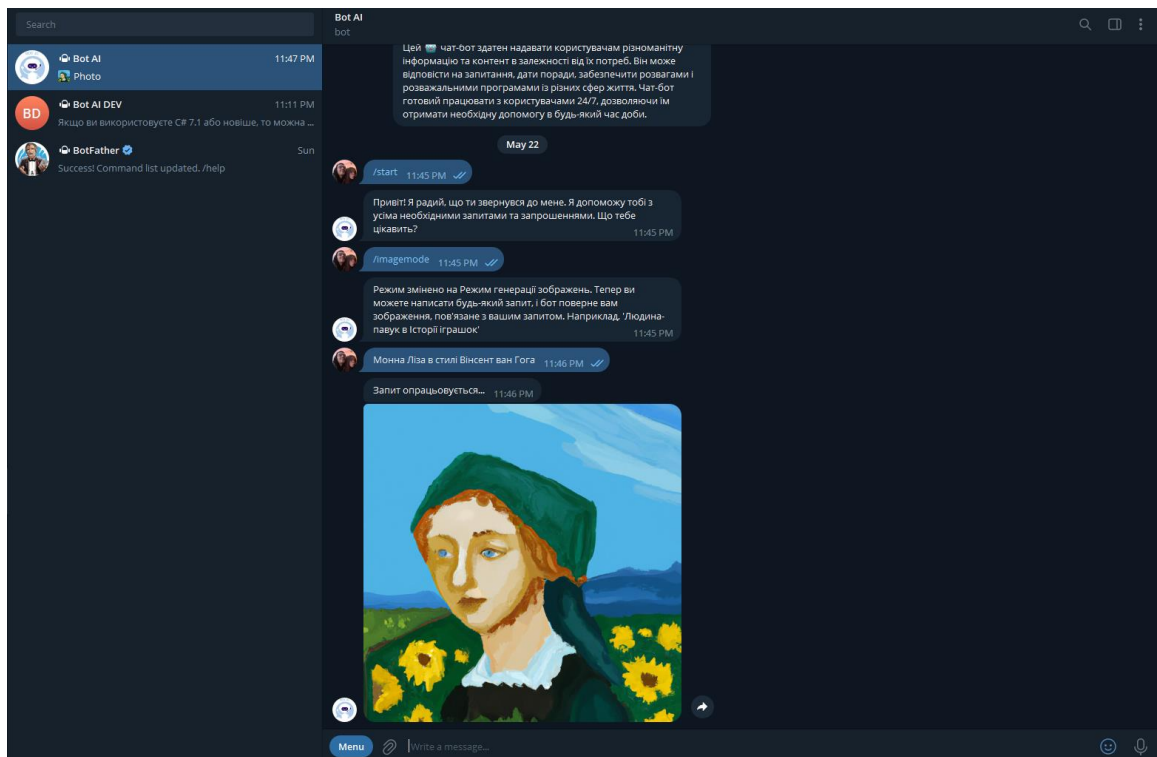


Рисунок 3.15 — Запит на генерацію зображення введено.

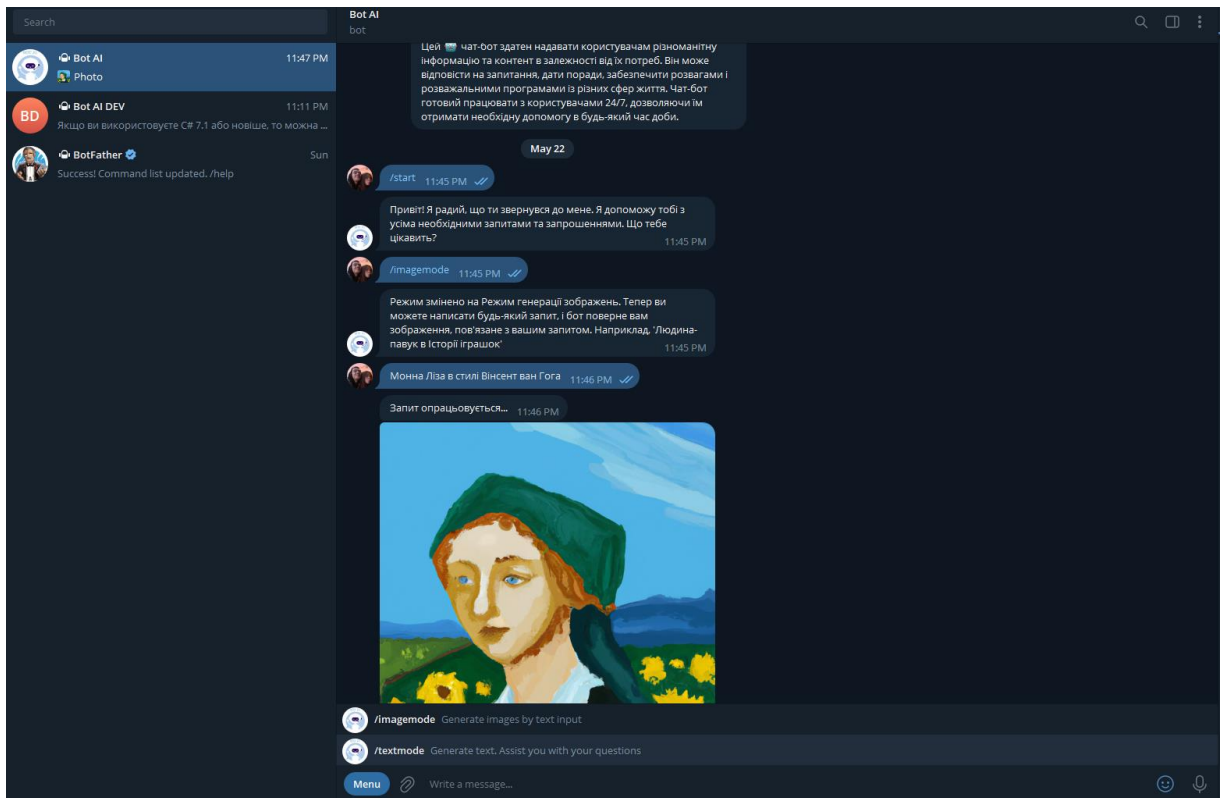


Рисунок. 3.16 — Натиснути “Menu” та “/textmode”.

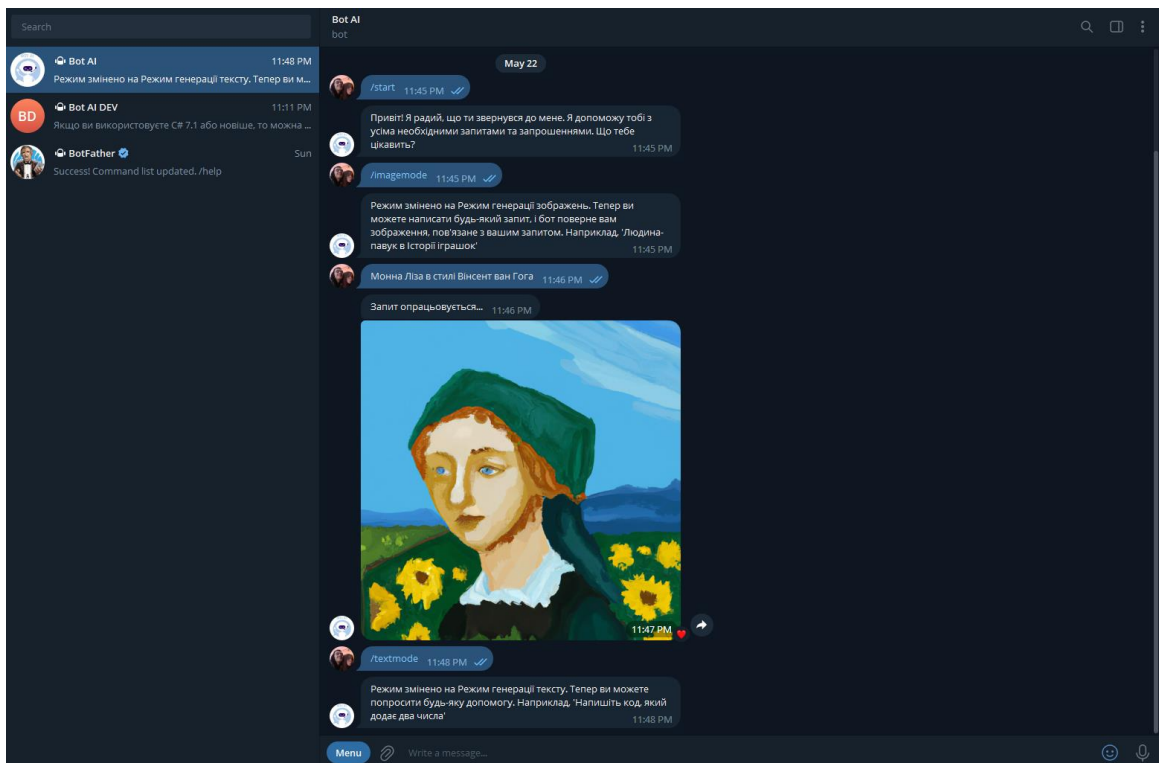


Рисунок. 3.17 — Відповідь від бота при натисканні “/textmode”.

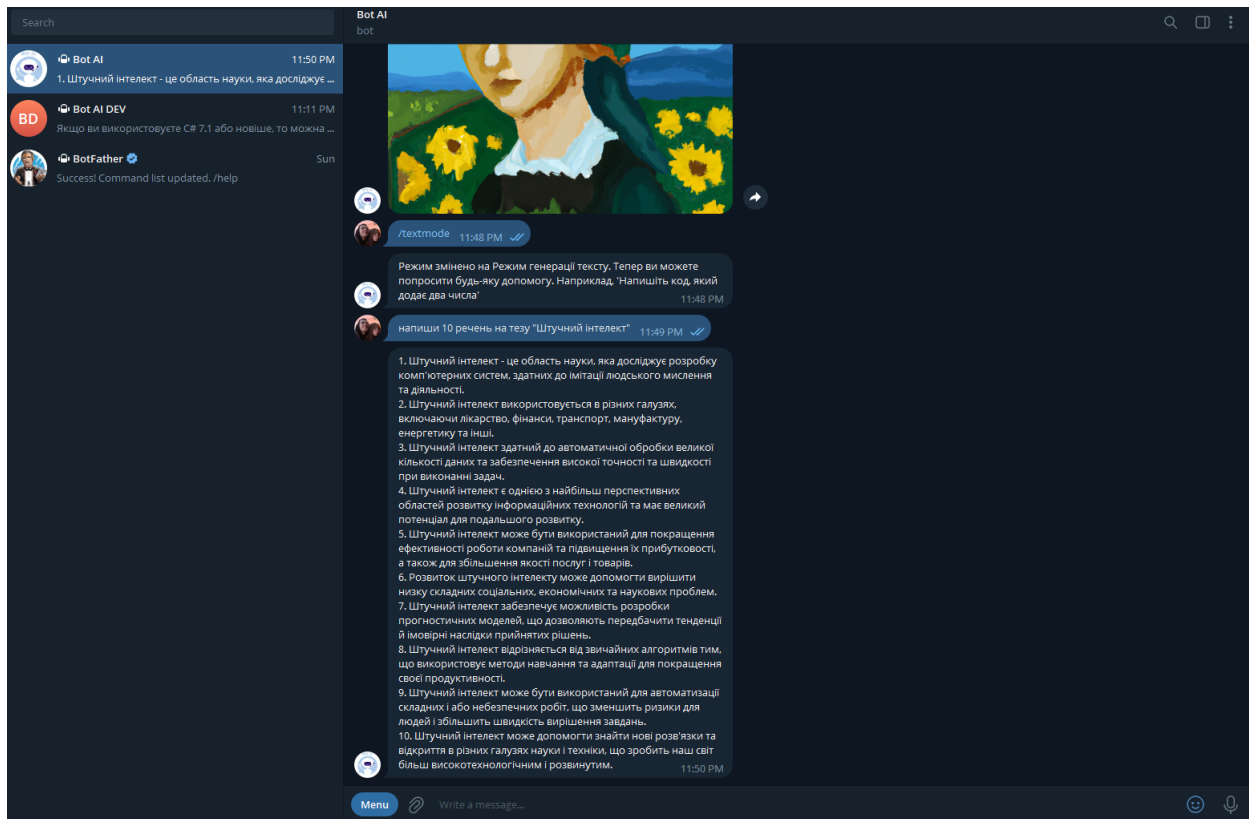


Рисунок. 3.18 — Запит до бота.

4.3 Інструкція користувачу

Для того, щоб використовувати чат-бот в режимі текстового асистента:

1. Натиснути кнопку «Почати»(Start)
2. Написати запит на допомогу(Приклад «Створи функцію на с# яка буде сумувати 2 числа»)
3. Відправити повідомлення
4. Очікувати відповіді

Для того, щоб використовувати чат-бот в режимі генерації зображень:

1. Натиснути кнопку «Почати»(Start)
2. Натиснути кнопку «Меню» зліва від поля вводу тексту(Menu)
3. Натиснути команду «/imagemode»

4. Очікувати відповіді
5. Написати запит на генерацію зображення (Приклад «Людина-павук в Історії іграшок»)
6. Відправити повідомлення
7. Очікувати відповіді

ВИСНОВКИ

В рамках випускної кваліфікаційної роботи був розроблений та впроваджений чат-бот. Ідея дипломного проекту полягала в створенні Telegram- боту для Автоматизації комунікації та ефективного виконанні рутинних завдань без прямої участі людини та створення зображень за запитом користувача.

На основі аналізу були сформовані наступні вимоги а саме: проаналізувати предметну область, обрати засоби реалізації, спроектувати і реалізувати телеграм-бота для оптимізації процесів лізингу автівок.

Під час аналізу предметної області було обрано та описано програмні засоби для розробки чат-бота: мову програмування C#. Використано Visual Studio, хмарну платформу Azure Function та App Azure Storage Account Table.

Додаток є актуальним для активних користувачів месенджера Telegram. Чат-бот націлений на спрощення та полегшення виконання рутинних завдань та генерацію зображень.

Завдяки чіткому виконанню завдань, які були поставлені на початку роботи, було отримано вичерпні знання щодо методів і засобів розробки телеграм-ботів, за допомогою яких було спроектовано і розроблено повноцінного телеграм-бота який готовий до впровадження і використання в реальних умовах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Telegram Messenger". Google Play. Посилання <https://play.google.com/store/apps/details?id=org.telegram.messenger&hl=ru&gl=US&pli=1>. Отримано 18 березня 2021 року.
2. «Telegram APIs». Посилання: <https://core.telegram.org/api>
3. «OpenAI API». Посилання <https://openai.com/blog/openai-api>
4. «How to Easily Create Telegram Bot using C#» . Посилання <https://www.codeproject.com/Tips/5303407/How-to-Easily-Create-Telegram-Bot-using-Csharp>
5. «Developing a Telegram Bot with C#»Посилання <https://markjames.dev/2022-07-28-building-telegram-bot-csharp/>
6. "Telegram Messenger". Посилання <https://apps.apple.com/ru/app/telegram-messenger/id686449807>. Отримано 26 березня 2021.
7. "Telegram Desktop latest release". Посилання <https://github.com/telegramdesktop/tdesktop>. Отримано 20 березня 2021.
8. "Telegram Desktop". Посилання <https://apps.microsoft.com/store/detail/telegram-desktop/9NZTWSQNTD0S?hl=ru-ru&gl=ru> Отримано 20 березня 2021.
9. "Telegram Beta 2 – HockeyApp". Посилання <https://rink.hockeyapp.net>
10. "Join the Telegram Messenger beta". Посилання <https://testflight.apple.com>
11. "Version history". Telegram. Посилання <https://desktop.telegram.org/changelog>
12. "Telegram Swift – HockeyApp". Посилання <https://rink.hockeyapp.net>.
13. "List of Telegram applications". Посилання <https://telegram.org/apps>.
Отримано 6 лютого 2014.
12. "Telegram FAQ". Telegram. Посилання <https://telegram.org/faq> Отримано 29 березня 2021.

15. EWDN, Editor (30 August 2013). "Russia's Zuckerberg launches Telegram, a new instant messenger service". Посилання <https://www.reuters.com/article/idUS74722569420130830>. Отримано 8 November 2020.
16. " Best practices for prompt engineering with OpenAI API ". Посилання <https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api>
17. " Summarizing Scientific Articles with OpenAI ". Посилання <https://medium.com/@avra42/summarizing-scientific-articles-with-openai-and-streamlit-fdee12aa1a2b>
18. " OpenAI Completions API — Complete Guide ". Посилання <https://www.linkedin.com/pulse/openai-completions-api-complete-guide-bo%C5%A1ko-bezik/>.
19. "Telegram hits 500M monthly active users". Telegram. Посилання <https://lifestyle.thecable.ng/telegram-hits-500m-active-users-25m-in-3-days-amid-whatsapps-controversial-privacy-policy/>. Retrieved 12 January 2021

ДОДАТКИ

Додаток А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка Telegram чат-боту зі штучним інтелектом мовою C#

Виконав студент 4-го курсу
групи ПД-42
Дмитришин Валерій Віталійович
Керівник роботи
доцент кафедри ІПЗ Гребенюк Віктор Вікторович

Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ





- **Мета роботи** – спрощення процесу надання відповідей для користувача за рахунок Telegram чат-боту зі штучним інтелектом мовою C#.
- **Об'єкт дослідження** - опрацювання та надання відповідей на запити користувача.
- **Предмет дослідження** - чат-бот для відповідей на запити користувача.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Провести аналіз обраної предметної області та визначити основні можливості штучного інтелекту.
2. Обрати оптимальне рішення для розробки Telegram чат-боту, з урахуванням вимог до функціональності та ефективності.
3. Сформувати вимоги до розробки Telegram чат-боту з урахуванням недоліків та переваг вже існуючих аналогів.
4. Спроекувати та розробити Telegram чат-боту зі штучним інтелектом для відповідей на запити користувача та можливості генерації зображень.
5. Провести функціональне тестування можливостей чат-боту.

🔍 📄 📁 📧 📧 📧 📧

АНАЛІЗ АНАЛОГІВ

	ChatGPT 	GPT-4 OpenAI Chat GPT 1 	ChatGPT bot 	Bot AI 
Можливість генерації відповідей	+	+	+	+
Можливість генерації зображень	-	-	-	+
Вивід коду в зручному вигляді	+	-	-	+
Робота без авторизації	-	+	+	+
Пошук в історії за запитам	-	+	+	+
Історія запитів	+	+	+	+

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1. Аналіз повідомлень користувача та генерація відповідей за допомогою штучного інтелекту.
2. Використання історії запитів для більш точної постановки відповідей.
3. Генерація зображень за запитом користувача за допомогою штучного інтелекту.
4. Можливість зміни режимів між генерацією відповідей та генерацією зображень.

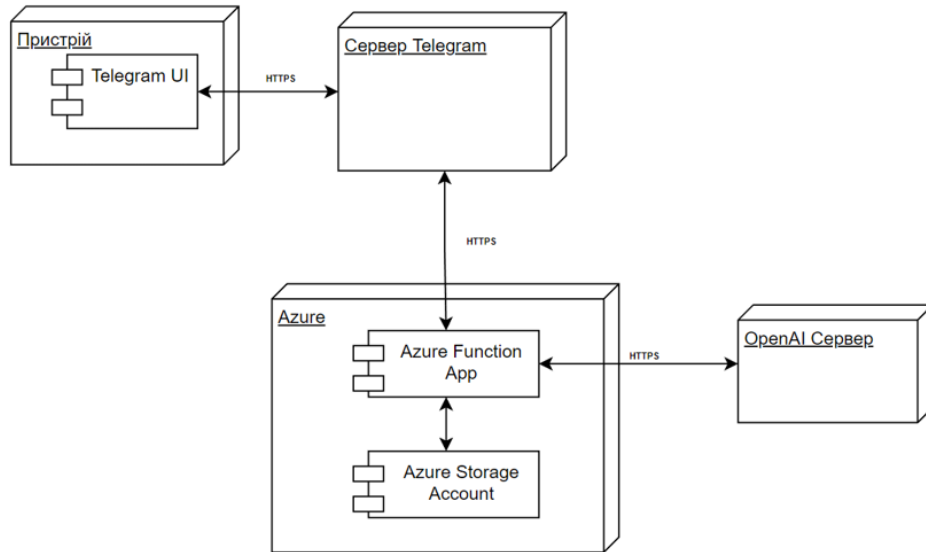
ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



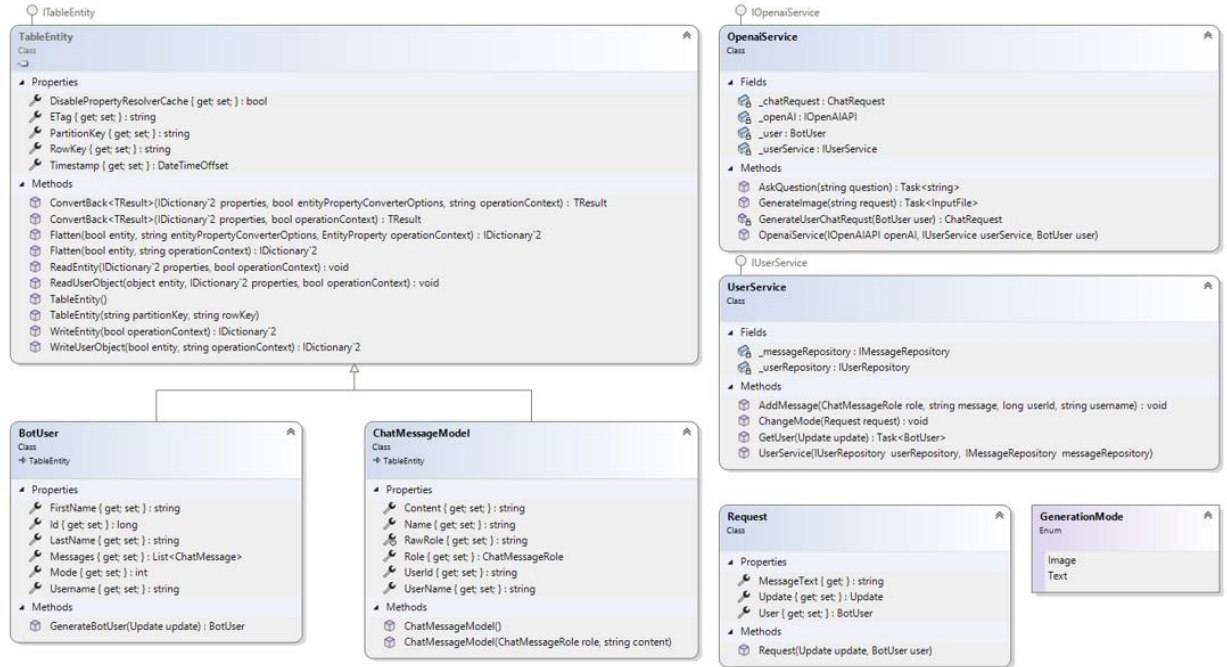
ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



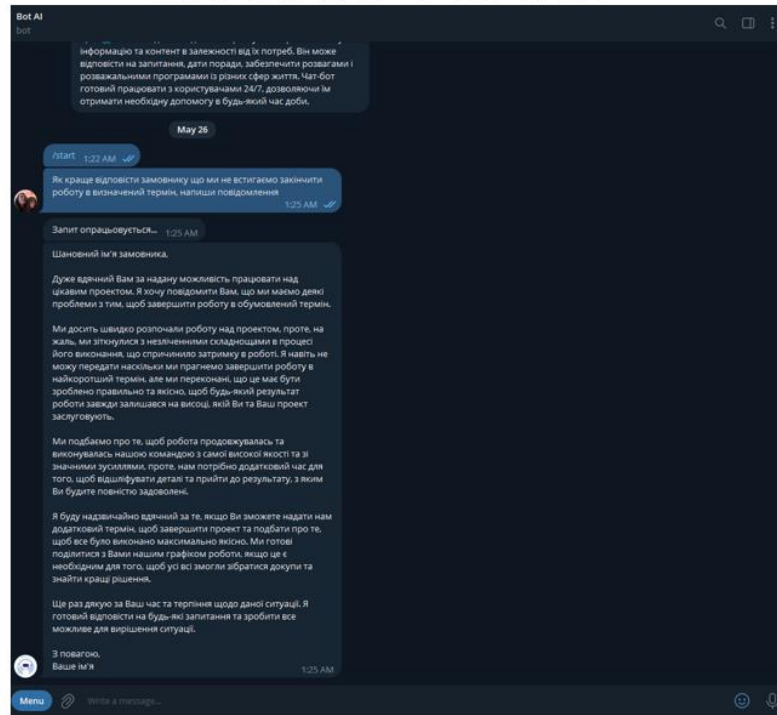
ДІАГРАМА РОЗГОРТАННЯ



ДІАГРАМА КЛАСІВ

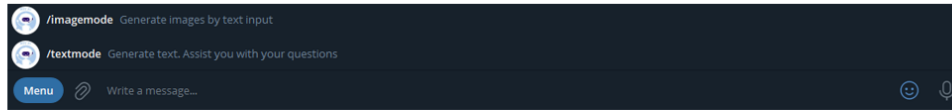


ЕКРАННІ ФОРМИ



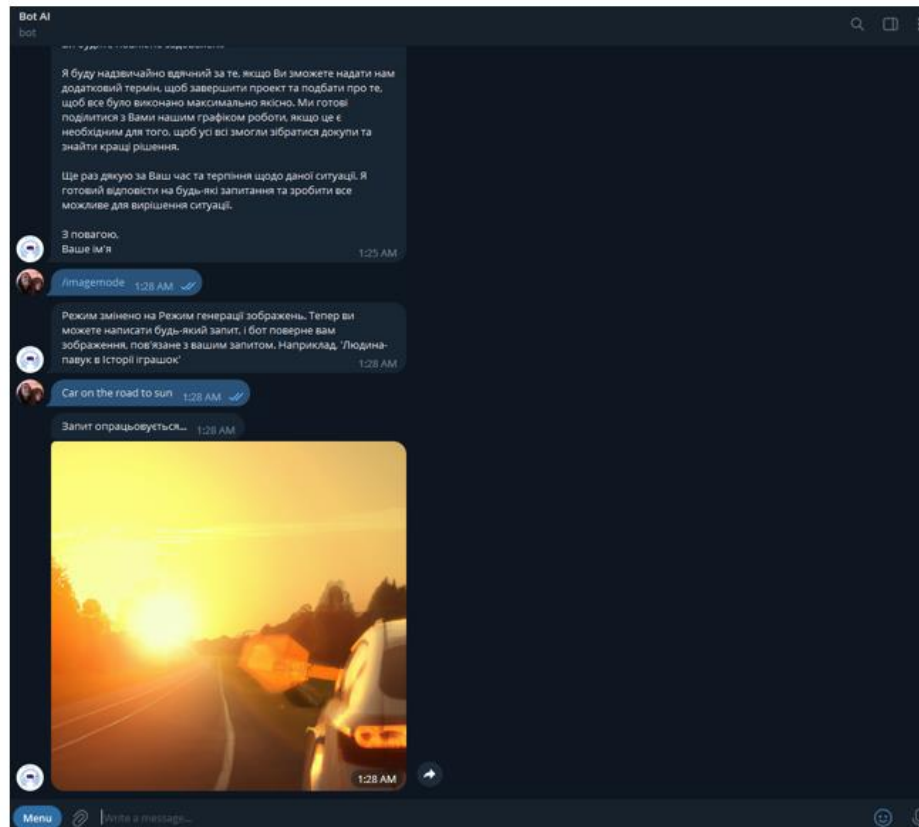
Робота бота в режимі генерації відповідей

ЕКРАННІ ФОРМИ



Меню вибору режимів

ЕКРАННІ ФОРМИ



Робота бота в режимі генерації зображень

ЕКРАННІ ФОРМИ



Відео роботи бота

ВИСНОВКИ

1. В результаті аналізу предметної області встановлено, що штучний інтелект має можливості використання для аналізу повідомлень користувача, обробки та надання відповідей, а також генерації зображень на базі запитів користувача за допомогою OpenAI API.
2. Обрано рішення для розробки чат-боту з використанням Azure Storage Account Table для зберігання попередніх питань та відповідей, OpenAI API для обробки запитів, надання відповідей та генерації зображень, Azure Function app для запуску застосунку у хмарі.
3. З урахуванням недоліків та переваг існуючих аналогів сформульовані вимоги до розробки чат-боту. Зокрема, необхідно забезпечити можливість генерації зображень за запитом користувача на основі наявних зображень в мережі.
4. Проведено тестування функціональних можливостей чат-боту.

ДЯКУЮ ЗА УВАГУ!