

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА**

**Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем**

Курсова робота

на тему:

**РОЗРОБКА ТЕЛЕГРАМ-БОТА ДЛЯ ВІДСЛІДКОВУВАННЯ ЯКОСТІ
ПОВІТРЯ**

Виконав студент 3-го курсу бакалаврату

Владислав СНІГОВСЬКИЙ _____
(підпис)

Науковий керівник:

Кандидат фізико-математичних наук, доцент кафедри інтелектуальних
програмних систем,

Євген ДЕМКІВСЬКИЙ _____
(підпис)

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

Студент _____
(підпис)

ЗМІСТ

ВСТУП	3
1.ПОСТАНОВКА ЗАДАЧІ	4
2. ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ЧАТ-БОТА	5
2.1. Мова програмування	5
2.2. Платформа чат-бота	6
3. ОСОБЛИВОСТІ РОЗРОБКИ	11
3.1. Розробка структури бази даних	11
3.2. aqicn.org	11
3.3. Bot	13
3.4. Розгортання на сервері	15
4. ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ ПРОГРАМНОГО ПРОДУКТУ	16
4.1. Концепція бота	16
4.2. Реєстрація у боті	16
4.3.Перевірка Якості повітря	17
4.2. Головне меню	18
4.4.Мої Локації	19
4.5.Про бота	21
4.6.Налаштування	21
4.7.Команди	21
Висновки	22
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	23

ВСТУП

В наш час месенджери стали частиною нашого життя: кожен день ми проводимо у них багато часу спілкуючись з нашими знайомими, друзями, колегами і не тільки. Кожен день месенджери все більше полегшують наше життя надаючи різний корисний функціонал. Отже сьогодні у месенджері ви можете не тільки спілкуватися з іншими людьми, а й замовляти їжу, знаходити місце на паркінгу, будувати маршрут з точки А в точку Б тощо.

Мета й завдання роботи. Метою роботи є розробка чат-бота для контролю рівня забруднення повітря у локаціях обраних користувачем. Для досягнення цієї мети було поставлено такі завдання:

- визначити схему та особливості роботи чат-ботів;
- побудувати технічне завдання бота;
- спроектувати та реалізувати чат-бота для поставлених цілей.

Актуальність роботи полягає у відсутності аналогічних, доволі функціональних ботів, де користувач зможе легко та швидко отримувати інформацію про стан повітря.

Об'єктом дослідження є розробка багатокористувацького бота для відслідковування стану повітря.

Предметом дослідження є пакет для взаємодії з Bot API месенджера Telegram.

1. ПОСТАНОВКА ЗАДАЧІ

Користуючись будь-яким ресурсом, користувач перш за все звертає увагу на його зрозумілість – якщо людина користуючись не буде розуміти що їй потрібно зробити для того щоб скористатись функціональністю, вона піде до іншого аналогічного ресурсу, тож головним завданням проекту було створити зручного, корисного та максимально інтуїтивно зрозумілого чат-бота який буде інформувати користувача про зміну якості повітря і різних місцях які він обере та надавати корисні поради в залежності від рівня якості.

Для успішного виконання поставленої задачі, її було розбито на декілька під-задач:

1. Розробити зручну та зрозумілу схему меню для користувача;
2. Розробити та з'єднати базу даних та бота;
3. Підключити бота до ресурсу з інформацією по API;
4. Зробити зручні сповіщення про зміну стану якості повітря.
5. Розмістити бота та базу даних на хостингу для постійної роботи.

2. ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ЧАТ-БОТА

Для розробки чат-бота було обрано наступні технології:

- Python в якості мови програмування;
- Telegram в якості платформи для чат-бота.

2.1. Мова програмування

Було обрано мову програмування Python[1]. Це інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Великим плюсом цієї мови, окрім простоти, є велика кількість корисних модулів на всі випадки. Інтерпретованість цієї мови робить її ідеальною для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ.

Python підтримує динамічну типізацію, тобто, тип змінної визначається лише під час виконання. Система класів підтримує множинне успадкування і метапрограмування. Будь-який тип, включаючи базові, входить до системи класів, й за необхідності можливе успадкування навіть від базових типів.

Дизайн мови побудований навколо об'єктно-орієнтованої моделі програмування. Реалізація ООП в Python потужна, елегантна та добре продумана.

Найчастіше Python використовується у веб-розробці та аналізі великих даних. Якщо треба доповнити функціональність мови то можна використати один з фреймворків: Django, Pyramid, Flask та інші. Python підходить і для створення прикладних програм та ігор. Так на цій мові було написано такі ігри як EVE Online, Battlefield 2, World of Tanks та такі програми як графічний редактор GIMP та торрент-клієнт BitTorrent до шостої версії.

Додаток для якого була обрана дана мова має обробляти вхідну інформацію дуже швидко і ця мова справляється з поставленою задачею. Окрім цього, пошук помилок у коді, та реалізувати функціонал буде набагато легше у світлі простоти мови та кількості бібліотек для різних задач що значно полегшить роботу.

2.2. Платформа чат-бота

Інтерфейс чат-бота було обрано оскільки сьогодні майже кожен має свій аккаунт як мінімум у одному з месенджерів а користування ботами у них дуже зручне, зрозуміле та доступне.

Основними месенджерами серед яких обирався потрібний були Viber, Facebook Messenger[2] та Telegram але з огляду на зростаючу популярність Telegram не тільки серед молоді, було обрано саме цю платформу. Окрім цієї переваги, можу також зазначити що всі боти у Facebook Messenger мають проходити перевірку адміністраторами а користувачі обов'язково мати свій аккаунт у соціальній мережі Facebook в той час як для Telegram достатньо мати лише номер мобільного телефону.

Незважаючи на два попередніх дуже суттєвих достоїнства месенджера Telegram перед Facebook Messenger, я можу назвати ще декілька плюсів цієї платформи перед конкурентами: Telegram використовує шифрування при передачі повідомлень за допомогою таких алгоритмів як RSA-2048, DH-2048, SHA-256 та AES-256, а для веб-версії – протокол HTTPS. Також, клієнти для месенджера Telegram існують на всіх популярних на сьогодні операційних систем: Linux, MacOS, Windows, iOS, Android, а також веб-версія.

Telegram розробила платформу Telegram Bot API[3] для тих хто зацікавлений у розробці ботів, наприклад: сьогодні багато маленьких

ресторанів, інтернет магазинів мають своїх ботів для швидкої та зручної комунікації зі своїми клієнтами.

Bot API це інтерфейс який працює за допомогою протоколу HTTP. У цієї платформи дуже широкий та зручний для розробника функціонал:

Обліковий запис бота. При реєстрації нового бота, ви маєте ввести унікальний ідентифікатор для вашого бота який має закінчуватися на bot. Все це треба створювати у телеграм боті BotFather, який є офіційним ботом від Telegram для реєстрації нових ботів, він вам допоможе його зареєструвати. Також, ви маєте ввести ім'я бота яке будуть бачити всі користувачі. Якщо ви все зробите правильно, BotFather скаже що він зареєстрував нового бота та дасть вам токен.

Далі, у цьому ж боті BotFather, ви можете додати своєму боту зображення Botpic, зробити опис (рис. 2.1) який буде висвічуватись користувачу коли той зайде до вашого бота та додати команди та написати about Bot – опис який буде видно у профілі бота.

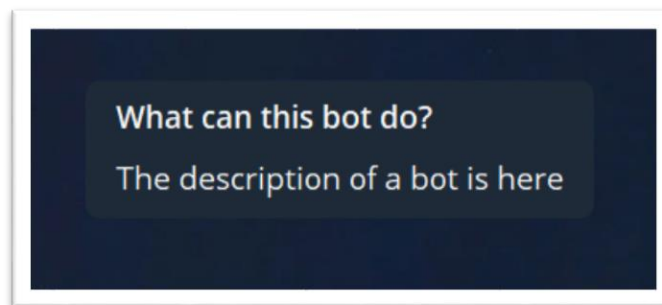


Рис.2.1. Опис бота у чаті

Після реєстрації бота можна зайти та побачити основну інформацію про нього (рис. 2.2): Botpic, або аватар, username – унікальне ім'я бота та Description – інформація за якою користувач може швидко зрозуміти для чого даний бот.

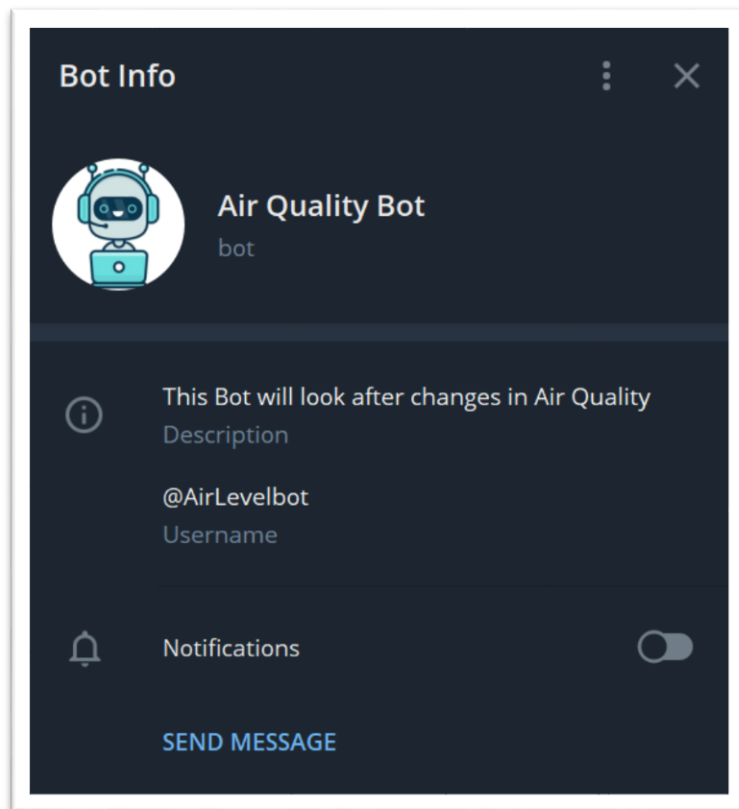


Рис. 2.2. Обліковий запис бота

Команди. Як я вже казав, у BotFather можна додати до вашого бота команди вигляду `/command`, та додати їй опис (рис. 2.3). Створюючи чат-бота, розробник, за потреби, має самостійно додавати різні команди та реалізувати їх підтримку у власному коді.

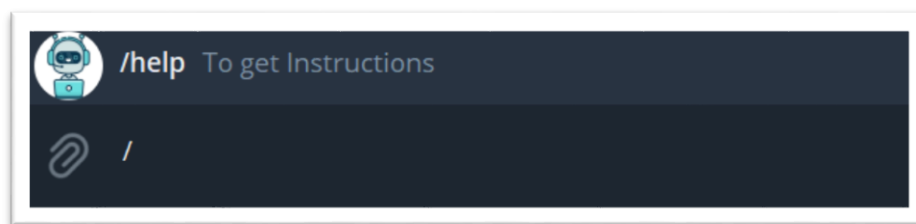


Рис. 2.3. Команди у боті

Інлайн режим. Цей режим дозволяє використовувати чат-бот у чатах з іншими користувачами. Для його використання треба підключити бота до потрібного чату. Завдяки цьому можна взаємодіяти з ботом прямо у переписці з

кимось або навіть у великій групі: сьогодні є дуже багато ботів які слідкують за “чистотою” чату та не дають “спамити” людям.

Спеціальні клавіатури. Для зручного спілкування з ботом можна створити клавіатуру на якій будуть виведені варіанти відповіді/меню бота (рис. 2.4).

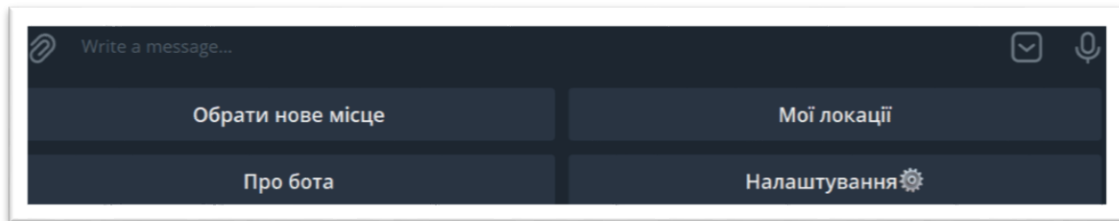


Рис. 2.4. Спеціальна клавіатура

Кнопки у повідомленнях. Якщо вам не подобаються кнопки у клавіатурі, то можна прикріпити кнопки до повідомлення (рис. 2.5). Натиснувши на цю кнопку бот може перейти за якимось посиланням, можна переслати повідомлення у інший чат або надсилати чат-боту інформацію про кнопку на яку ви натиснули.

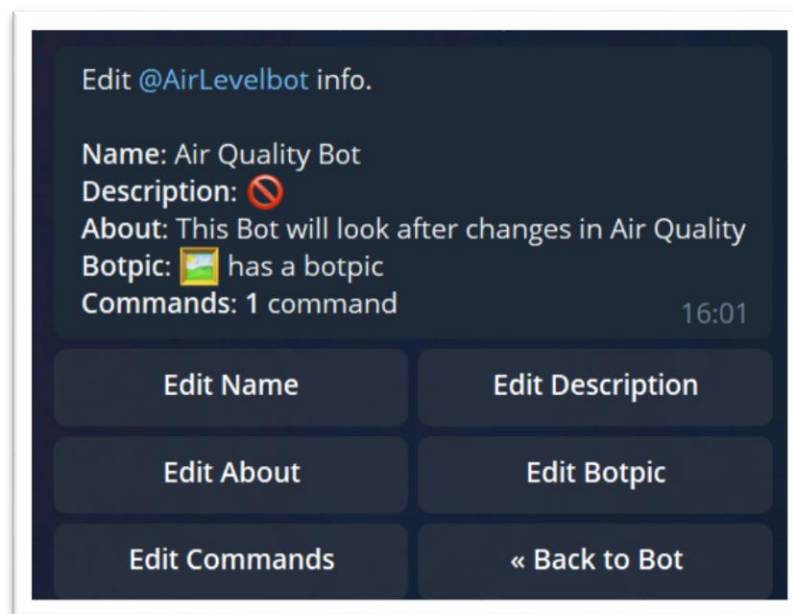


Рис. 2.5. Кнопки у повідомленнях

Payments API. Telegram співпрацює з декількома платіжними системами тож за допомогою цієї платформи за потреби можна проводити платежі у самому телеграмі через бота.

Також Telegram Bot API інтегрований з багатьма корисними сервісами та ресурсами такими як: Google Maps, Gmail, Wikipadia, IMDB.

Загалом завдяки платформі Telegram розробник не має практично ніяких обмежень для своїх проектів окрім декількох незначних лімітів на відправку повідомлення для чат-ботів:

- одне повідомлення в секунду на один чат;
- тридцять повідомлень в секунду, якщо повідомлення відправляються в декілька чатів.

Якщо ж ліміт буде перевищено, то повідомлення будуть відправлені з затримкою, необхідною для виконання вимог ліміту. Але зі свого досвіду можу сказати що це не є критичним, особливо для даного бота.

3. ОСОБЛИВОСТІ РОЗРОБКИ

3.1. Розробка структури бази даних

База даних була написана на мові PostgreSQL[4] оскільки у цієї мови багато можливостей. Ця мова створена з використанням об'єктно-реляційної моделі і підтримує складні структури і широкий спектр вбудованих і обумовлених користувачем типів. PostgreSQL забезпечує розширену ємність даних і дуже добре дбає про їх цілісність.

Сама база даних лежить на віддаленому сервері elephantsql [5]. Працювати з цим ресурсом дуже просто та безкоштовно (за умови дотримання ліміту пам'яті) а контролювати всі записи можна через pgAdmin 4[6].

База даних складається з основної таблиці де записані всі користувачі бота з їх id у Telegram, та таблиці локацій користувача де записані всі дані по локаціям.

У locations є такі поля:

1. idx – id локації.
2. name – назва локації.
3. last_aqi – остання зміна стану якості повітря. Якщо колись якість зміниться на «крок», то користувач отримає повідомлення а бот занесе до бази нове значення.
4. step – «крок» локації.
5. notification – змінна яка показує чи хоче користувач отримувати сповіщення про зміни.

3.2. Ресурс aqicn.org

Для отримання інформації про якість повітря бот отримує з сайту aqicn.org через API [7].

Спочатку, для користування цим ресурсом, треба зареєструватися та отримати токен за яким бот буде звертатися до сайту для отримання інформації. Коли користувач просить бота показати йому рівень якості повітря у певному місці, бот відсилає запит у вигляді:

1. `/feed/:city/?token:=token`, де замість `city` бот пише місто яке треба перевірити а замість другого `token` вставляє токен отриманий після реєстрації
2. `/feed/geo::lat;:lng /?token:=token`, де замість `lat` та `lng` бот пише геолокацію яку надіслав користувач а замість другого `token` вставляє токен отриманий після реєстрації
3. `/feed/idx /?token:=token`, де замість `idx` бот записує `id` локації. Цей `id` проставляє даний ресурс а бот отримує його після першого запиту коли користувач додає нову локацію. Цей запит відсилається тільки коли локація вже збережена у базі.

У відповідь на ці запити бот отримує два види відповіді:

1. Якщо запит вдалий, бот отримує відповідь у вигляді `json`(рис. 3.1) з якого потім дістає потрібну інформацію.

```
{
  "status": "ok",
  "data": {
    "aqi": 72,
    "idx": 1451,
    "attributions": [
      {
        "url": "http://www.bjmemc.com.cn/",
        "name": "Beijing Environmental Protection Monitoring Center (北京市环境保护监测中心)"
      },
      {
        "url": "https://waqi.info/",
        "name": "World Air Quality Index Project"
      }
    ],
    "city": {
      "geo": [
        39.954592,
        116.468117
      ],
      "name": "Beijing (北京)",
      "url": "https://aqicn.org/city/beijing"
    }
  }
}
```

Рис. 3.1. Відповідь ресурса на вдалий запит

2. Якщо ж запит невдалий, бот отримає відповідь із статусом «error»(рис. 3.2).

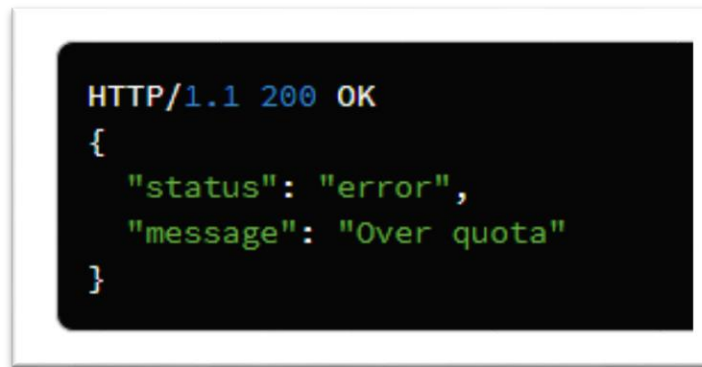


Рис. 3.2. Відповідь ресурса на невдалий запит

3.3. Чат-бот

Aiogram[6] є фреймворком для взаємодії з Telegram Bot API який повністю асинхронний, написаний на Python 3.7 з asyncio та aiohttp. Він відрізняється від головного конкурента своєю швидкістю та асинхронністю що дає змогу для написання складних алгоритмів.

Я обрав саме aiogram оскільки для того щоб реалізувати повний функціонал, мені потрібна була асинхронність: з початком роботи бота починає працювати функція яка буде відслідковувати зміни якості повітря у всіх користувачів. Це зроблено для того щоб не перегружати сервер та базу даних якщо користувачів буде дуже багато. Ця функція кожну годину починає проходити по всім користувачам та їх локаціям, відправляти запити на отримання нової інформації та звірятися з попередніми записами, а у разі потреби надсилають сповіщення до користувача. Якщо робити таку функцію не асинхронною, то буде хаос: користувачі можуть отримувати по декілька зайвих повідомлень а сервер буде постійно перегружений.

Особливості фреймворка:

- Асинхронність;
- Швидкість;
- Має Finite-state machine;
- Може відповісти на веб-хук.

Бот має декілька пакетів:

1. Config. У цьому пакеті записані токени для Telegram та aqicn.org, інформація про базу даних: пароль, ім'я бази даних, ім'я користувача, хост та порт.
2. AirInfo. Цей пакет витягує з файлу відповіді ресурса aqicn.org потрібну інформацію про стан якості повітря, а за потреби і idx.
3. Database. Цей пакет зв'язує бота з базою даних. Коли бот хоче якось взаємодіяти з базою даних, він звертається до цього пакету надаючи йому потрібну інформацію а пакет в свою чергу створює запит та отримуючи інформацію про базу даних з пакету Config створює запит.
4. English. Цей пакет відповідає за англійську мову бота. Кожен раз коли бот формує повідомлення для користувача, або робить клавіатуру, він звертається до цього пакету для отримання тексту.
5. Ukrainian. Такий самий пакет, але для української мови.
6. Keyboard. Пакет для формування клавіатури. Кожен раз коли бот має змінювати клавіатуру, він передає потрібну інформацію цьому пакету, а той збирає клавіатуру та повертає боту.

7. Statushash. Цей пакет допомагає боту розуміти що він зараз чекає від користувача. Якщо бот в даний момент має отримати від користувача якусь нову назву/крок, статус бота змінюється і бот розуміє що він зараз має чекати.
8. Main. Це головний пакет в якому в основному реалізована вся логіка даного бота. У цьому пакеті є лендлери, які реагують на натискання на кнопки, надсилання геолокації тощо. Також, в основному, цей пакет пов'язує всі інші.

3.4. Розгортання на сервері

Я використовую підхід Continuous Deployment. Це означає, що весь процес виконується автоматично на основі розробленого сценарію. Розгортання на сервері виконується за допомогою двох складових: репозиторія та платформи для розгортання. Код бота розташований у репозиторії на github [10]. У якості платформи для розгортання бота я обрав Heroku [8]. Дана платформа є досить легкою для тестування та розгортання чат-бота. Розгортання чат-бота на платформі Heroku виконується у декілька етапів сценарію:

- завантаження змін у репозиторій;
- запуск тестів;
- архівація всіх необхідних для розгортання файлів;
- завантаження архіву на сервер Heroku;
- створення конфігураційних файлів;
- запуск додатку;

4. ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ ПРОГРАМНОГО ПРОДУКТУ

4.1. Концепція бота

Назва боту – Air Quality Bot (Бот Якості Повітря). Його задача полягає в тому щоб тримати в курсі зміни якості повітря у певному місці користувача. Користувач спочатку реєструється у боті та додає місцезнаходження за якими він хоче спостерігати. Можна додати обравши місто зі списку, а можна відправити боту свою геолокацію та отримати найближчу станцію яка і буде відправляти інформацію про зміну стану повітря у її місцевості.

4.2. Реєстрація у боті

Щоб зареєструватися у боті потрібно знайти його у вашому додатку Telegram у стрічці пошуку за username - @AirLevelbot (рис. 4.1).

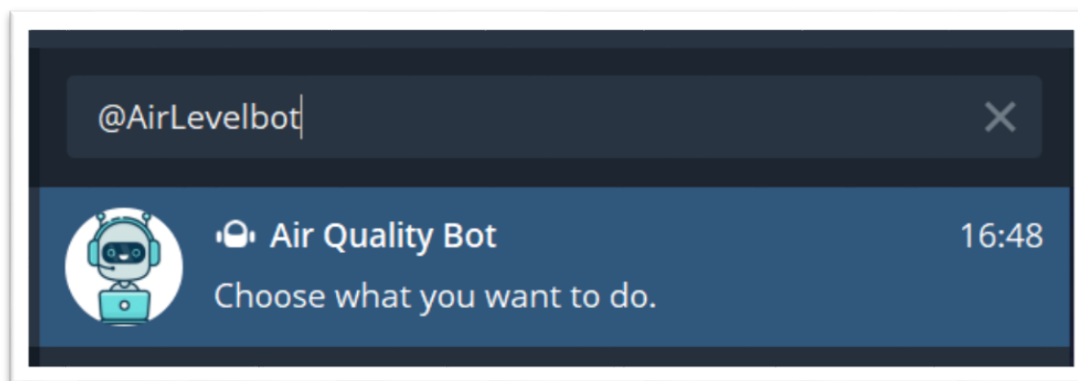


Рис. 4.1. Пошук бота

Далі заходимо до нього та натискаємо кнопку Start. Далі треба обрати мову на якій ви хочете спілкуватися з ботом та отримувати від нього інформацію про зміну стану якості повітря у ваших локаціях (рис. 4.2). Після цього бот ваш напише інструкцію по використанню. Ознайомившись із нею натискаємо кнопку і бот реєструє нового користувача.

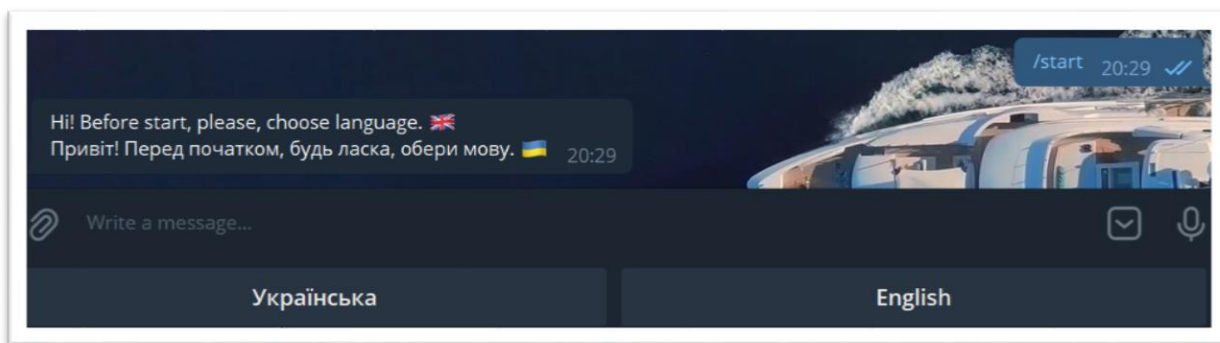


Рис. 4.2. Початок реєстрації

4.3.Перевірка Якості повітря

Щоб отримати інформацію про якість повітря у певному місці, з головного меню треба перейти до «Select new location», або «Обрати нове місце», (рис. 4.2). Після цього бот у вас буде вибір – або обрати нову локацію зі списку, або по вашому місцезнаходженню (рис. 4.3):

1. Зі списку. Якщо ви хочете обрати нове місце за списком – слід перейти до «Обрати зі списку», далі бот надасть вам список усіх країн (рис. 4.4), міста яких занесені у базу. Ви обираєте країну за бажанням та натискаєте на неї. Далі бот виведе вам список усіх міст цієї країни (рис. 4.5) і попросить вас обрати якесь одне. Обравши потрібне, бот, якщо у цьому місті є станції які відслідковують зміну стану якості повітря, надішле вам поточну інформацію (рис. 4.6) та надасть пораду якщо якість погана. У іншому випадку сповістить що там станцій немає.
2. Отримавши потрібну вам інформацію бот запропонує вам зберегти дану локацію щоб відслідковувати її якщо ви вже її не додали. Якщо ви хочете її додати – слід натиснути «Yes, add», або «Додати». Після цього бот напише вам варіант назви нової локації та попросить вас

підтвердити її. Якщо ви хочете назвати її інакше – то просто напишіть боту ваш варіант назви на натисніть «Add», або «Додати». Після того як ви додали нову локацію бот запитає чи хочете ви отримувати сповіщення про зміну стану повітря. Відповідайте за бажанням.

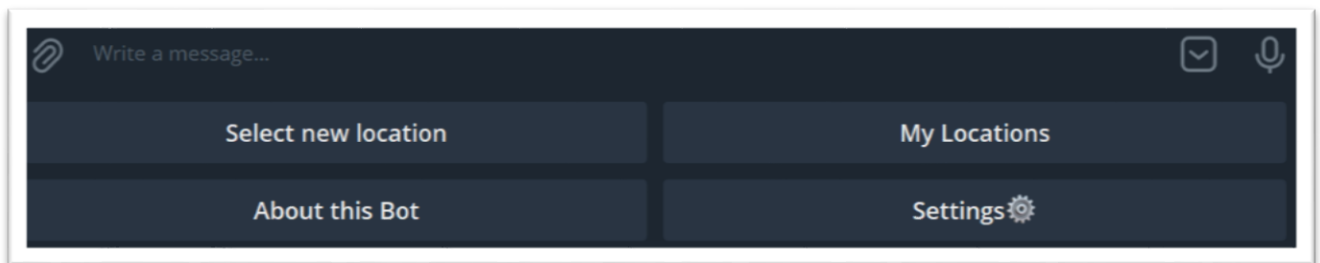


Рис. 4.2. Головне меню

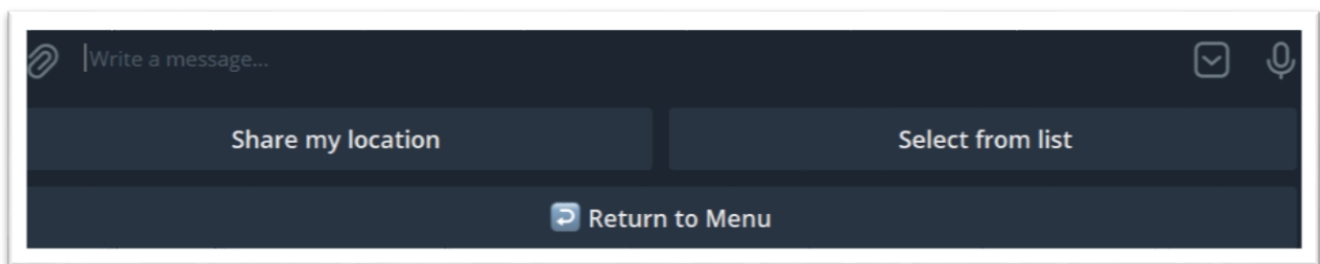


Рис. 4.3. Геолокація або Список

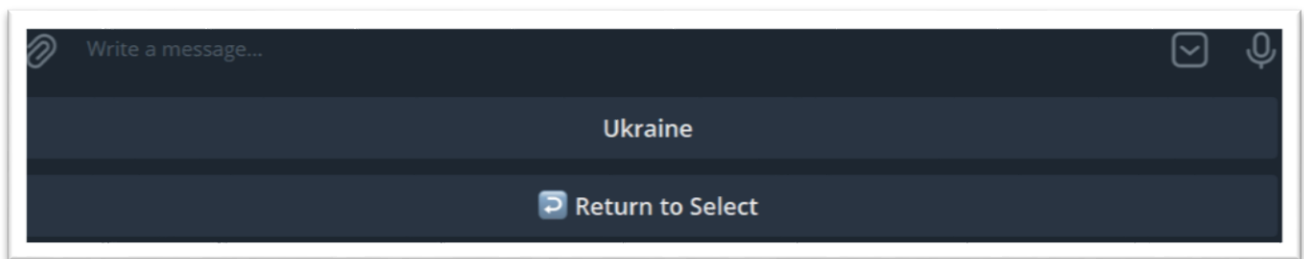


Рис. 4.4. Список країн

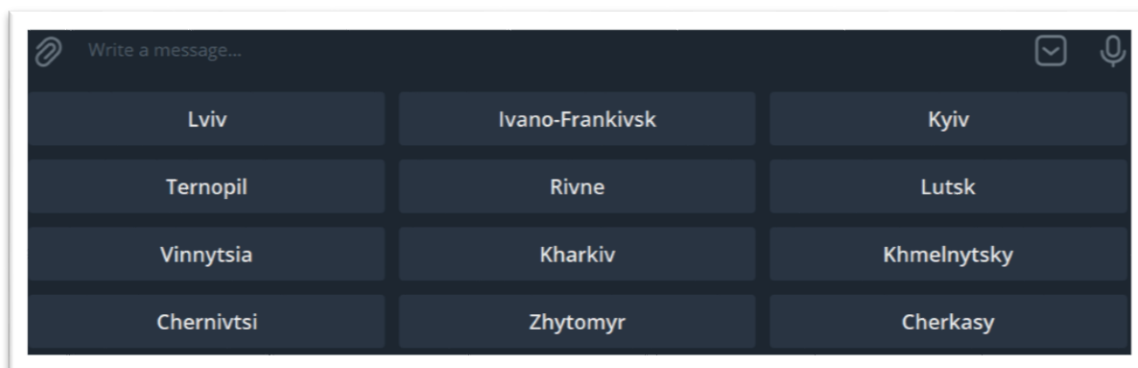


Рис. 4.5. Список міст

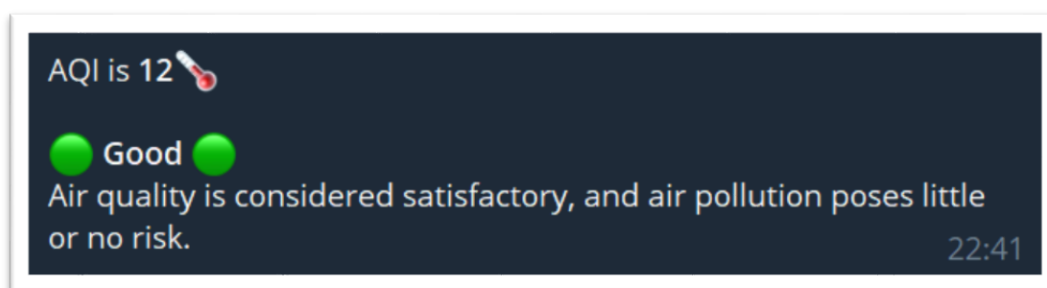


Рис. 4.6. Інформація про стан повітря

4.4.Мої Локації

У боті можна подивитися всі збережені вами локації. Для того щоб їх подивитися потрібно перейти до «My locations», або «Мої локації». Там ви побачите весь список локацій які ви зберігали (рис. 4.7). Натиснувши на будь-яку з них ви отримаєте поточну інформації про стан якості повітря у тому місці та відкриєте меню локації. У цьому меню можна редагувати налаштування даної локації (рис. 4.8):

1. Редагувати ім'я. Натиснувши на дану кнопку ви перейдете у режим редагування назви локації. Щоб змінити назву – відправте боту нову назву та натисніть «Change», або «Змінити», або ж «Cancel»/«Відмінити» щоб відмінити зміну.

2. Редагувати крок. Якщо ви включили повідомлення про зміну рівня стану повітря, то ви будете отримувати його якщо рівень, з часу останньої перевірки, змінився на певне число – крок. У даному режимі ви можете змінити крок на будь-який. Щоб змінити крок напишіть боту нове число та натисніть «Change», або «Змінити», або ж «Cancel»/«Відмінити» щоб відмінити зміну.
3. Якщо ви вмикали повідомлення для даної локації, третьою кнопкою у вас буде кнопка «Notifications Off», або «Повідомлення Викл». Щоб увімкнути, просто натисніть на неї. А якщо ви їх не вмикали, то у вас буде кнопка «Notifications On», або «Повідомлення Вкл». Щоб увімкнути – натисніть на кнопку.
4. Наступна кнопка видаляє дану локацію. Якщо вам не потрібно слідкувати за цією локацією – можна видалити її просто натиснувши на цю кнопку.

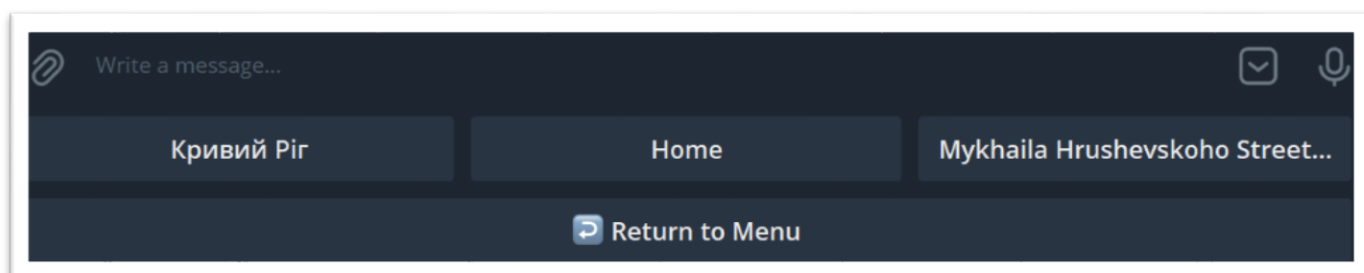


Рис. 4.7. Список збережених локацій

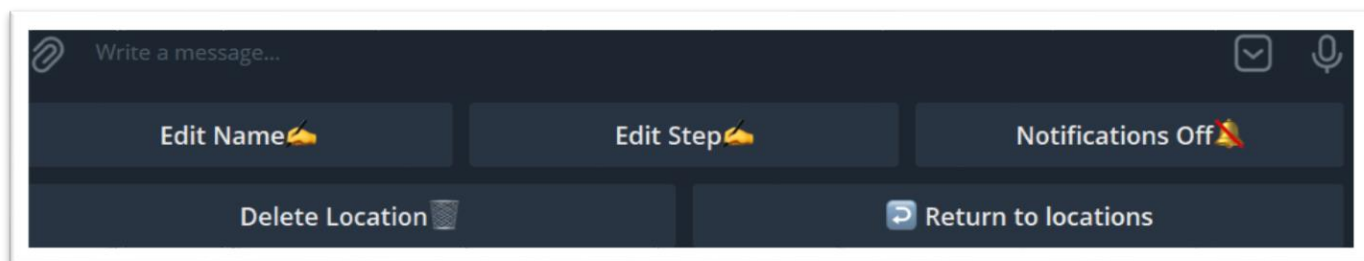


Рис. 4.8. Меню налаштувань Локації

4.5. Про бота

У цьому пункті меню ви отримаєте основну інформацію про бота та дізнаєтесь як отримати інструкції по його використанню.

4.6. Налаштування

У налаштуваннях ви зможете змінити мову для бота. Обравши українську мову – все спілкування з ботом буде проходити на українській мові, а якщо обрати англійську – на англійській.

4.7. Команди

Якщо ви забули як користуватися ботом – можна надіслати боту повідомлення `/help`, або набрати «/» та обрати з підказки команду «`/help`». У відповідь ви отримаєте розгорнуту інструкцію по використанню.

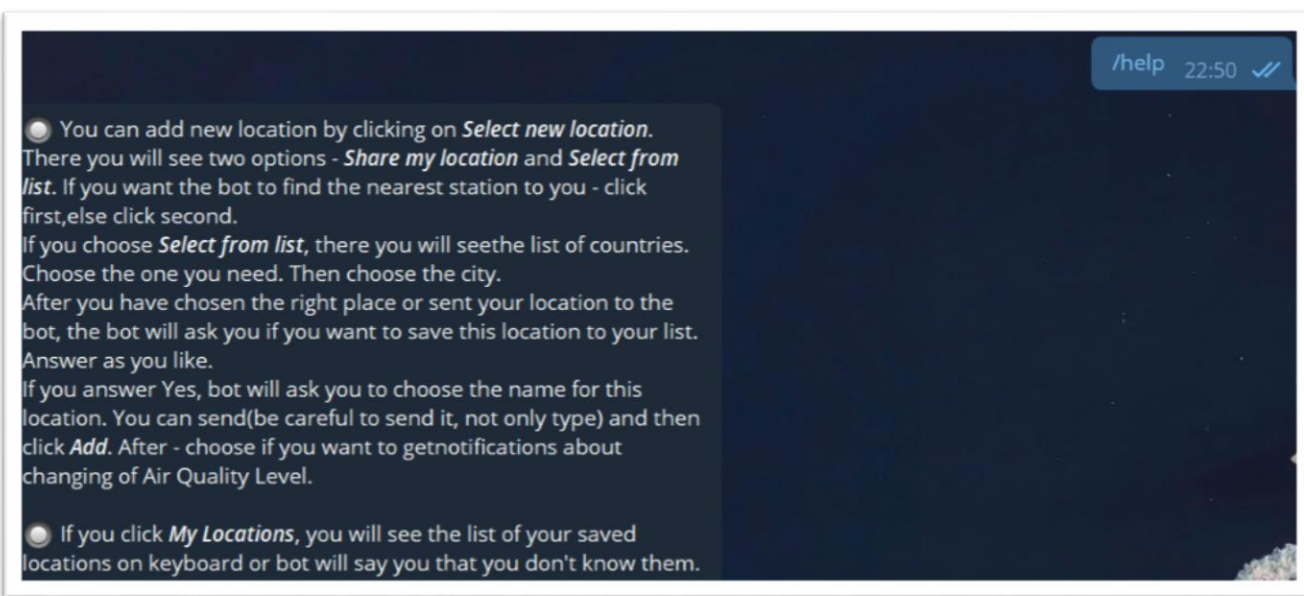


Рис. 4.9. Команда `/help`

Висновки

В результаті виконання роботи було освоєно роботу чат-ботів, їх можливості, ліміти, розроблено зручну схему меню, спроектовано, реалізовано та розгорнуто програмний продукт на хостингу.

Даний бот можна використовувати будь-кому у повсякденному житті для зменшення ризику здоров'ю.

Розроблений чат-бот має наступні функціональні можливості:

- 1) визначення на передача на сервер геолокації для якої необхідно контролювати рівень забруднення повітря;
- 2) збереження потрібних локацій для постійного спостереження;
- 3) можливість увімкнути сповіщення для отримання повідомлень у разі зміни стану якості повітря;
- 4) можливість використання двох мов.

Оскільки список країн та міст можна збільшувати та впроваджувати новий, корисний функціонал, то я вважаю що є раціональним продовження дослідження та розвиток даного програмного продукту в цій області.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Python Programming Language Documentation [Електронний ресурс] – Режим доступу: <https://www.python.org/doc/>.
2. Facebook Messenger Platform [Електронний ресурс] – Режим доступу: <https://developers.facebook.com/docs/messenger-platform/>.
3. Telegram Bot API. [Електронний ресурс] – Режим доступу: <https://core.telegram.org/bots/api>
4. PostgreSQL Documentation [Електронний ресурс] – Режим доступу: <https://www.postgresql.org/docs/>.
5. ElephantSQL Documentation [Електронний ресурс] – Режим доступу: <https://www.elephantsql.com/docs/>.
6. pgAdmin4 Documentation [Електронний ресурс] – Режим доступу: <https://www.pgadmin.org/docs/>.
7. Aqicn Documentation [Електронний ресурс] – Режим доступу: <https://aqicn.org/json-api/doc/>.
8. Aiogram Documentation [Електронний ресурс] – Режим доступу: <https://docs.aiogram.dev/en/latest/>.
9. Github Documentation [Електронний ресурс] – Режим доступу: <https://docs.github.com/en>.
10. Heroku Documentation [Електронний ресурс] – Режим доступу: <https://devcenter.heroku.com/>.