

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**


Факультет комп'ютерних наук та кібернетики
Кафедра математичної інформатики

Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки


на тему:

**РОЗРОБКА ТЕЛЕГРАМ-БОТУ ДЛЯ ОБРОБКИ МЕТЕОРОЛОГІЧНИХ
ДАНИХ**

Виконала студентка 4-го курсу
Варзар Вероніка Анатоліївна



(підпис)

Науковий керівник:
асистент, кандидат технічних наук
Федорус Олексій Мстиславович


(підпис)

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

Студент


(підпис)

Роботу розглянуто й допущено до захисту на
засіданні кафедри математичної інформатики
« ____ » _____ 202_ р.,
протокол № _____

Завідувач кафедри
Терещенко В. М.

(підпис)

Київ – 2022

РЕФЕРАТ

Обсяг роботи 50 сторінок, 21 ілюстрація, 1 таблиця, 27 використаних джерел, 2 додатки.

Ключові слова: ЧАТ-БОТ, РОЗРОБКА, ПРОГРАМНЕ СЕРЕДОВИЩЕ, ЛЮДИНО-МАШИННИЙ ІНТЕРФЕЙС, ФРЕЙМВОРК, ПОШУКОВА СИСТЕМА, МЕСЕНДЖЕР, СЕРВІС, ЗАПИТ.

Об'єктом програмної реалізації є створення чат-боту в середовищі Telegram для обробки метеорологічних даних.

Метою кваліфікаційної роботи є ознайомлення з процесом розробки чат-ботів, дослідження ринку чат-ботів, виявлення їхніх переваг та недоліків для використання в різних сферах. Ознайомлення з етапами розробки чат-боту, вивчення його складових компонентів.

Програмним середовищем для створення чат-боту було обрано мову програмування Python, середовище Atom - вільний текстовий редактор і редактор коду компанії «GitHub». Для хостингу боту було обрано сервіс Python Anywhere. Нами було обрано додаток телеграм для користування чат ботом. Даний вибір зроблено через неймовірно велику популярність даного месенджера, а також його надійність і порівняно високі можливості для створення та роботи чат ботів.

Результати роботи: головним результатом даної роботи на практиці є створена програма чат боту, котра дає можливість пошуку різноманітної текстової інформації по критеріям, які були задані. Користувачі матимуть можливість прочитати інформацію про погоду в тому місті чи населеному пункті, який вони вкажуть, а також прогноз погоди на майбутні дати.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. АНАЛІЗ СТАНУ ДОСЛІДЖЕНЬ З ПИТАНЬ РОЗРОБКИ ЧАТ-БОТУ	7
1.1. Загальне поняття чат-бота, його функції. Класифікація чат-ботів	7
1.2. Платформи для створення чат-ботів	9
1.3. Соціальні мережі як платформи для створення чат-ботів	16
1.4. Схема роботи чат-бота.....	18
1.5. Алгоритм створення чат-ботів.....	19
РОЗДІЛ 2. ФОРМУЛЮВАННЯ ФУНКЦІОНАЛУ ТА АНАЛІЗ ТЕХНОЛОГІЙ	21
2.1. Постановка задачі	21
2.2. Обґрунтування вибору засобів розробки.....	22
2.3. Черговість виконання функцій	24
2.4. Налаштування робочих параметрів функцій	27
2.5. Вимоги до запитів та побудова відповідей	27
2.6. Склад і структура програмного забезпечення системи.....	28
2.6.1. Модуль main.py	29
2.6.2. Модуль geocoding.py	29
2.6.3. Модуль messageText.py	29
2.6.4. Модуль config.py	29
2.6.5. Модуль requirements.txt	30
2.7. Опис API, що використовуються в роботі.....	30
2.7.1. Geocoding API.....	30
2.7.2. One Call API.....	32
2.8. Інтерфейс користувача та результати тестування програмного забезпечення	34
2.9. Особливості та рекомендації щодо використання програмного забезпечення	34
ВИСНОВКИ.....	36
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	38
ДОДАТОК А.....	40
ДОДАТОК Б	44

ВСТУП

Оцінка сучасного стану об'єкта розробки. Станом на сьогодні, популярність систем обміну миттєвими повідомленнями (месенджерів) зростає в геометричній прогресії. Ми сміло можемо говорити, що вони не тільки виступають одним із засобів для спілкування між друзями, чи людьми в різних регіонах світу, а й стали засобами отримання різного роду інформації (новини, меми тощо). Станом на 2022 рік важливу роль у цьому відіграли боти, яких є неймовірна кількість починаючи від спеціальних ботів для отримання різних розсилок до замовлення таксі, їжі, продуктів тощо.

Звичайно ж різні інвестори та власники бізнесу цікавляться таким корисним інструментом для розвитку власного бізнесу. На сьогоднішній день чат боти вже останні 5-6 років досить популярні та активно використовуються в США та країнах ЄС, однак в нашій державі вони лише набувають певної популярності і є надзвичайно перспективними серед розробників.

Варто також враховувати, що для клієнтів одним із найважливіших аспектів виступає досвід користувача під час роботи із ботом. Сюди включаємо те як сам бот сприймає певні команди від користувача, наскільки доречними є його відповіді тощо.

Пошук інформації є ключовою і нероздільною частиною діяльності кожного хто користується мобільним телефоном у наш час. Понад 90% всіх власників смартфона що небудь шукають у пошукових запитах і використання сучасників засобів комунікації та месенджерів отримують значний попит. Одним із таких ключових месенджерів виступає телеграм.

Актуальність роботи та підстави для її виконання. Використання ботів як новітнього інструменту комунікацій визначена тим, що в даний період часу, чат боти досить великою кількістю використовуються в різних сферах задля встановлення спілкування із інтернет користувачами. Значну популярність боти отримали, в той час як були інтегровані у месенджери та різноманітні соціальні мережі.

Загалом під чат-ботом, ми розуміємо, спеціальну програму, котра займається імітацією мовленнєвої поведінки людини та виступає навіть як співрозмовник у віртуальному світі.

Чат боти мають можливість використовуватись в певних розважальних цілях або ж просто для розмови з людиною, а також існують чат боти, що підкажуть погоду, офіційні курси валют, годину тощо.

Незважаючи на широке їх поширення, поки що не сформувалося певних алгоритмів та правил роботи з чат-ботами, оцінки їх ефективності та доцільності використання, що також підкреслює актуальність даного дослідження, яке допоможе виділити основні проблеми у використанні чат-ботів для комунікації зі споживачами, шляхи їх рішення, сфери застосування, і навіть напрями розвитку.

Метою дослідження є характеристика процесу створення чат-боту в соціальній мережі телеграм та його моделювання.

Завдання дипломної роботи:

1. Охарактеризувати поняття чат-бота та дослідити ринок чат-ботів у наш час.
2. Дослідити принципи роботи чат ботів та головні етапи його створення.
3. Визначити технологічні процеси та рішення, котрі будуть використанні під час створення чат бота.
4. Розробити чат бот здійснюючи використання обраної мови програмування.

Об'єкт, методи та за засоби розроблення. Об'єктом роботи є чат-бот, як пошукова система, алгоритм пошуку інформації та спілкування за допомогою чат-боту.

Програмній реалізації розробленого чат-боту передувало дослідження ринку чат-ботів, аналіз інструментальних засобів для розробки чат-ботів.

Програмним середовищем для створення чат-боту було обрано мову програмування Python, середовище Atom - вільний текстовий редактор і

редактор коду компанії «GitHub». Для хостингу боту було обрано сервіс Python Anywhere.

Нами було обрано додаток телеграм для користування чат-ботом. Даний вибір зроблено через неймовірно велику популярність даного месенджера, а також його надійність і порівняно високі можливості для створення та роботи чат-ботів.

Можливі сфери застосування. Головним результатом даної роботи на практиці є створена програма чат-боту, котра дає можливість пошуку різноманітної текстової інформації по критеріям, які були задані. Користувачі матимуть можливість прочитати інформацію про погоду в тому місті чи населеному пункті, який вони вкажуть, а також прогноз погоди на майбутні дати.

Взаємозв'язок з іншими роботами. За методами розробки та інструментальними засобами робота виконувалась сумісно із роботами, в яких характеризується засоби людино-машинного інтерфейсу для пошуку інформації.

РОЗДІЛ 1. АНАЛІЗ СТАНУ ДОСЛІДЖЕНЬ З ПИТАНЬ РОЗРОБКИ ЧАТ-БОТУ

1.1. Загальне поняття чат-бота, його функції. Класифікація чат-ботів

Чат-бот – це в першу чергу спеціальний акаунт, за котрим не закріплена людина, а ті повідомлення, котрі були надіслані їм, обробляє спеціальна зовнішня система. Окрім цього, для користувачів спілкування із таким ботом має можливість виглядати як звичайне листування із справжньою живою людиною [12].

Загалом чат-бот – це є розумна програма, котра виконує різноманітні функції та загалом можна сказати живе у месенджері.

Функціями чат боту виступають [2]:

1. Служба підтримки для потенційного клієнта. Чат бот надає можливість здійснити заміну не для всіх зручного FAQ на сайтах і надати відповідь на всі можливі типові запитання від клієнта. Бот має здатність працювати 24 год/добу та 7 днів/тиждень і зможе розвантажити час для працівників у компанії.

2. Клієнтський сервіс. Завдяки допомозі чат боту клієнт має можливість здійснити покупки чи отримати будь-якого типу послуги. В торгівлі, де постійно розширюється асортимент товарів, значно складніше шукати які-небудь товари, а от після невеликого аналізу і завдяки боту, можна зрозуміти, що саме цікавить потенційного клієнта і бот автоматично буде відправляти йому прямі посилання.

3. Маркетинг. Бот – це один із інструментів маркетингу, котрий допомагає просувати контент та підтримує лояльність споживачів і проводить аналітичні звіти. Завдяки допомозі бота можна здійснювати розсилки, надавати інформацію про актуальні акції, збирати відгуки і визначати якість наданих послуг клієнту.

4. Робота у внутрішньому середовищі компанії. Боти надають можливість здійснити оптимізацію таких процесів як бронювання переговорних кабінетів, інформування персоналу про відпустки та їх дати, розклад транспортних засобів для працівників, терміни оплати премій та зарплат тощо.

5. Рекрутинг. Беручи до уваги співбесіду із ботом менеджер із підбору працівників може вирішити, яких саме кандидатів необхідно запрошувати на живу бесіду, а кому необхідно відразу відмовити у пропозиції працювати в компанії.

На сьогоднішній день існує декілька варіантів для класифікації чат ботів, однак здійснивши їх аналіз ми можемо виділити два типи:

- Бізнес класифікація додатків чат-бот.
- Технічна класифікація чат-бот додатків.

Згідно бізнес класифікації, ми можемо поділити чат боти на наступні типи:

- Розмовні. Тобто такі, що створені задля спілкування та загалом дуже схожі на розмову із звичайною людиною. Вони не мають жодного конкретного завдання чи мети.

- Асистенти. Виходячи із певних цілей, а також відповідей користувачів отримують потрібні дані.

- Q&A. Принципом роботи є так зване “одне питання – одна відповідь”.

Згідно технічного типу характеристики, боти мають наступний поділ:

- Створенні на певних бізнес правилах. В такому типі спілкування між ботом та людиною на 100% продумана розробником боту та має так звану деревоподібну структуру. Завдяки порівняно великій кількості варіантів, людина приходиться до певного вирішення свого запитання.

- Створенні на штучному інтелекті. Він абсолютно відрізняється від першого варіанту та немає власної визначеної структури. Даний вид ботів повинен мати порівняно великий обсяг даних задля якісної роботи.

- Гібридні. Даний тип ботів виступає так званою сумішшю першого і другого типу, тобто розмова із людиною йде по певному визначеному типу,

однак використовують штучний інтелект задля визначення намірів користувачів.

1.2. Платформи для створення чат-ботів

Задля створення чат-ботів, ми маємо можливість використовувати два підходи. Перший – створити чат-бот на будь-якій мові програмування, а другий – без використання коду, при допомозі платформи розробки. Окрім цього, після створення боту, варто здійснити аналіз його роботи із метою його оптимізації. Дивлячись на це інструментальними засобами для створення боту є:

- Фреймворки, котрі вимагають додаткових навичок програмування.
- Платформи для розробки ботів та оброблення природної мови, що дасть змогу створити чат-бот без жодного програмування.
- Спеціальні сервіси, котрі аналізують роботу чат ботів.

Розглянемо найбільш популярні платформи для створення чат-ботів. Роблячи дану добірку, орієнтація спрямована на сервіси, які найбільш зручні, прийнятні за ціною та мають хороший функціонал і відповідають наступним ознакам:

- процес створення чат-бота зрозумілий і гнучкий (майже всі послуги з вибірки - візуальні конструктори);
- тарифні плани оптимальні та прозорі;
- є вбудована CRM, інтеграції із зовнішніми сервісами прості;
- сервіс підтримує популярні месенджери та соцмережі [3].

1. Aimylogic

Російськомовний сервіс, у якому можна створювати розумні чат-боти. Створені чат-боти працюють на основі алгоритмів машинного навчання та постійно навчаються [18].

Окрім чат-ботів в Aimylogic також можна створювати голосові роботи, налаштовувати сценарії розумних дзвінків та впроваджувати їх у бізнес-процеси.

Бот створюється за допомогою візуального конструктора.

Основні можливості чат-ботів, розроблених в Aimylogic:

- розпізнавання природного мовлення та базових смислів. За замовчуванням робот розпізнає вітання, прощання, згоду, відмову, подяку, а також нецензурну лексику;

- можна навчати бота на прикладі реальних фраз (наприклад, можна створити інтенти та додати кілька прикладів фраз користувачів, за допомогою машинного навчання бот розпізнаватиме більшість фраз, які відносяться до цієї теми);

- прийом платежів;

- готові інтеграції з популярними месенджерами, голосовими помічниками та іншими сервісами;

- переклад діалогу на живу людину;

- для інтернет-магазинів актуальна опція «Довідник товарів». До нього можна внести список товарів, і бот зможе розпізнавати запити про продукцію та видавати користувачам ціни, характеристики та іншу інформацію про товар.

Підтримувані соцмережі та месенджери: WhatsApp, Telegram, Viber, Facebook, ВКонтакте, Instagram, онлайн-чати.

2. Flow XO

Англомовний онлайн-конструктор чат-ботів. Дозволяє вирішувати будь-які завдання бізнесу, такі як збирання лідів, автоматизація підтримки користувачів (відповіді на часті питання), прийом платежів, перемикання між чат-ботом та онлайн-чатом тощо [25].

Основні можливості:

- створення роботів за допомогою візуального конструктора;

- створення автоматизованих потоків для основних сценаріїв взаємодії з користувачами;

- прийом платежів;

- інтеграція зі 100+ сервісами (сервіси Google, CRM, платіжні системи тощо);

- збір статистики щодо взаємодії з користувачами;
- кроссплатформенна взаємодія з користувачами.

Месенджери та соцмережі, що підтримуються: Facebook Messenger, Telegram, Twilio, Slack, також можна розмістити чат на сайті.

3. Botmother

Російськомовний сервіс позиціонує себе як CMS для чат-ботів. Дозволяє створювати чат-боти для соцмереж та месенджерів за допомогою конструктора та готових блоків [19].

Основні можливості:

- налаштування сценаріїв для взаємодії з користувачами;
- переключення між чат-ботом та онлайн-чатом (підключення до чату оператора);
- надсилання файлів користувачам (відео, зображення, інші файли);
- зчитування QR-кодів;
- прийом платежів;
- обмін даними з іншими сервісами (Botmother підтримує інтеграцію з сервісами Zapier, Albato та ApiX-Drive, через які можна пов'язати чат-бот практично з будь-яким сервісом – від Google Таблиць до сервісів розсилки та CRM);
- є доступ до діалогів із користувачами. Можна переглядати історію взаємодії з роботом або підключатися до чату в реальному часі, якщо робот не справляється з питаннями людини.

Підтримувані месенджери та соцмережі: Telegram, Viber, WhatsApp, Facebook, ВКонтакте.

4. Fasttrack

Російськомовний сервіс позиціонує себе як універсальна платформа для вирішення бізнес-завдань за допомогою чат-ботів [24].

Основні можливості:

- візуальний конструктор чат-ботів для месенджерів;

- омніканальні сценарії. Можна налаштовувати та підключати єдину логіку для всіх активних каналів комунікації (месенджери, соцмережі, чат на сайті тощо);

- вбудована CRM-система із повною історією взаємодії клієнта з каналами комунікації;

- обробка даних від користувачів чи зовнішніх джерел у реальному часі;

- автоматичні та ручні підписки користувачів на автоворонки;

- підтримка одночасного запуску кількох сценаріїв користувача;

- покупки в месенджері (можливість вбудовування web-сторінки з каталогом та кошиком).

Месенджери та соцмережі, що підтримуються: VK, OK, Facebook Messenger, Telegram, Viber, WhatsApp, WhatsApp Business API, email та чат на сайті.

5. Botsify

Англомовний сервіс для створення чат-ботів із використанням штучного інтелекту [20].

Основні можливості:

- візуальний онлайн-конструктор чат-ботів;

- навчання роботів за допомогою алгоритмів машинного навчання;

- підбір схожих фраз із запитів користувачів;

- підтримка швидких відповідей;

- збір та кваліфікація лідів;

- підтримка 190 мов;

- підключення оператора до діалогу.

Підтримувані платформи: WhatsApp, Facebook Messenger, Telegram, SMS можна встановити як чат на сайті.

6. Dexter

Англомовний сервіс, що пропонує платформу для створення чат-ботів. Головна перевага платформи – боти створюються тут не звичним конструктором за допомогою додавання та перетягування блоків, а в текстовому редакторі. Як

вважають самі творці сервісу, це дозволяє максимально просто створювати ботів (не складніше, ніж написати коротку історію), а уникнення блоків і схем дає можливість не перевантажувати користувачів [22].

Основні можливості:

- надсилання повідомлень за заданим сценарієм;
- ведення та зберігання історії повідомлень з користувачами;
- збір та обробка даних користувачів;
- відправлення файлів (відео, аудіо, картинки);
- сегментація розсилок;
- розсилання повідомлень та повідомлень за розкладом.

Підтримувані платформи: Slack, Facebook Messenger, Twilio, Twitter, Alexa, Viber, сайт.

7. Gerabot

Український онлайн-конструктор для створення та управління чат-ботами. Також у Gerabot можна замовити створення чат-бота під завдання бізнесу під ключ [26].

Основні можливості:

- налаштування сценаріїв взаємодії з користувачами;
- підключення оператора до чату з користувачем (за запитом користувача або за заданим сценарієм);
- прийом заявок від користувачів;
- консультування клієнтів та надання потрібної інформації.

Підтримувані платформи: Viber, Telegram, Facebook, сайт.

8. Chatbot

Англомова платформа для створення чат-ботів під завдання основних напрямків бізнесу (маркетинг, продаж, саппорт клієнтів) [21].

Основні можливості:

- візуальний конструктор типу drag-and-drop;
- можливість тестування готового робота прямо в інтерфейсі платформи;
- кастомізація віджету чату для сайту;

- збір даних про користувачів, зберігання та сегментація бази;
- приймання замовлень;
- надсилання зображень та інших типів файлів;
- кнопки для швидких відповідей;
- аналіз повідомлень та навчання за допомогою алгоритмів штучного інтелекту;
- база готових шаблонів для типових сценаріїв, шаблони можна редагувати та кастомізувати під себе.

Підтримувані платформи: Facebook Messenger, Slack, LiveChat, сайт.

Порівняльна характеристика наведених платформ для створення чат-ботів наведена в табл.1.1.

Service	Link	Process visualization	Ready templates	Integrations or your own CRM	Communication platforms
Aimylogic	https://aimylogic.com/ru	Блок-схеми	Готові інтенти	Бітрікс24	WhatsApp, Telegram, WeChat, Viber, Facebook, ВКонтакте, Instagram, Google Analytics, MailChimp, онлайн-чати, голосові помічники
Flow XO	https://flowxo.com/	Блок-схеми	Шаблони сценаріїв	100+ інтеграцій, у тому числі кілька CRM	Facebook Messenger, Telegram, Twilio SMS, Clack, сайт
Botmother	https://botmother.com/ru	Блок-схеми	Не має	з CRM через сервіси-інтегратори Zapier, Albato, ApiX-Drive	Telegram, Viber, WhatsApp, Facebook, ВКонтакте, сайт

Fasttrack	https://fstrk.io/	Блок-схеми	Готові сценарії	своя CRM	Telegram, Viber, Facebook Messenger, ВКонтакте, WhatsApp, Email, сайт
Botsify	https://botsify.com/	Блок-схеми	Шаблони для різних ніш	Через Zapier	Сайт, WhatsApp, Facebook Messenger, SMS
Dexter	https://rundexter.com/	Текстовий редактор	Готові шаблони	По API	Twitter, Alexa, Slack, Facebook Messenger, Twilio, Viber, сайт
Gerabot	https://gerabot.com/constructor	Блок-схеми	Готові рішення за окрему плату	Своя CRM	Viber, Telegram, Facebook, сайт
Chatbot	https://www.chatbot.com/	Блок-схеми	Готові шаблони	По API і через Zapier	Facebook Messenger, Slack, LiveChat, сайт

Таблиця 1.1 - Порівняльна характеристика платформ для створення чат-ботів

За даними табл. 1.1 можна порівняти сервіси за п'ятьма найважливішими параметрами:

- метод візуалізації процесу (у всіх сервісів, крім Gerabot, це блок-схеми);
- готові шаблони (якщо вони є, створити чат-бот для типового завдання буде простішим);
- інтеграція з CRM (важливий критерій, оскільки утримання клієнтів та робота з базою критично важливі для будь-якого бізнесу у сучасних реаліях);
- підтримувані платформи;
- наявність додаткового функціоналу на кшталт розсилок, міні-лендінгів тощо [3].

1.3. Соціальні мережі як платформи для створення чат-ботів

Месенджер Facebook – це бот платформа, котра володіє аудиторією у більш ніж 1 млрд. активних споживачів на місяць та більш ніж 100 тис. ботів у світі. Даний месенджер виступає провідною платформою для служб обміну повідомленнями у Сполучених Штатах Америки, а набір функцій є найпросунутішим для розробника: прийом будь-яких платежів, аутентифікація, надсилання повідомлень за телефонними номерами тощо.

Viber – публічні облікові записи для бізнесменів стали більш доступними порівняно недавно. Підприємства мають змогу встановити діалог із своїми передплатниками через CRM систему та чат-бота. Популярними чат-ботами на даному месенджері є App in The Air, Aviasales і Glamour.

Telegram – перший месенджер, котрий зробив можливість створення ботів. Згідно даних Forbes, телеграм утворив понад 100 000 чат-ботів. Порівняно недавно чат-боти в телеграмі розпочали приймати кошти від користувачів.

Для великого бізнесу було створено WhatsApp Business, котрий дає змогу здійснювати реалізацію служб підтримки в месенджерах, ботах і так далі.

Розглянемо, чим відрізняється створення чат-ботів у Telegram, WhatsApp, Viber, Facebook [12].

1. Чат-бот у Telegram.

Основною перевагою цієї програми для обміну повідомленнями в порівнянні з Facebook Messenger та WhatsApp є його API (інтерфейс прикладного програмування), відкритий для всіх, що надає велику кількість можливостей та дає змогу реалізувати будь-які креативні ідеї. Це робить Telegram відмінною та безпечною платформою для найбільш оригінальних та складних ботів, таких як боти для обробки фінансових транзакцій, банківської інформації чи інтеграції криптовалютних гаманців. Також команда Telegram наголошує на тому, якою простою є інтеграція ігрових роботів HTML5, онлайн-запитів та клавіатур. Приклади іноземних телеграм-роботів: XirkleBot, MeansBot, BBVA.

2. Чат-бот у Viber.

Viber, станом на 2022 рік, є присутнім на всій планеті і велика кількість популярних брендів розпочали його використання для спілкування із клієнтами у всьому світі. Представники самої компанії говорять, що у них є приблизно 1 млрд. активних користувачів, однак незалежні дослідження говорять, що цифра у 255 млн. є ймовірнішою і більш правдивою. Viber не має жодних обмежень API і навпаки робить пропозицію деяких функцій, котрі не надають інші месенджери чи соціальні мережі. Так, наприклад у Viber, ми маємо змогу встановити свій дизайн шпалер чи кнопок тощо. Найпопулярнішими Viber ботами виступають Mica, Tech Talk, Queen.gr.

3. Чат-боти у Facebook

Facebook володіє аудиторією у більш ніж 1 млрд. активних споживачів на місяць та більш ніж 100 тис. ботів у світі. Кожного дня, дані боти збирають інформацію, надають рекомендації, підтримку і т. д. Користувачі мають можливість відповідати співрозмовникам за допомогою смайлів, зображень, музики чи відео та створювати навіть групові чати. Хоча Facebook має досить великий спектр функцій, він, на жаль, не володіє командами у вигляді кнопок. Іноземними брендами, що використовують боти у менеджері Facebook, є такі популярні компанії як Spotify, TechCrunch, Kindred Bravelly тощо.

4. Чат-бот у WhatsApp.

API для даного типу месенджера є платним, а можливість оплатити його не дає жодних гарантій для доступу. Окрім цього, боти в месенджері засновані в першу чергу на принципі відправлення повідомлень. Фактично, це не чат-бот у класичному значенні цього слова, а платформа для автоматичних повідомлень та привітань у один клік. Через це WhatsApp є найменш популярною платформою для запуску чат-бота. Месенджер є відносно новачком у сфері розробки чат-ботів, нові ідеї впроваджуються повільно, через що немає змоги передбачити, коли саме з'являться нові функції та чи станеться це взагалі. Приклади зарубіжних роботів Whatsapp: RedBus, KLM Royal, Sabrina.

За розглянутими характеристиками можна зробити висновок, що зараз існує два месенджера, які мають багато функцій та можливостей для розробки чат-ботів – це Telegram та Facebook Messenger.

1.4. Схема роботи чат-бота

Задля розуміння роботи бота та його складових, нами була побудована структура роботи завдяки допомозі абстрактної моделі (рис. 1.2). Дана система складається із певних сервісів, API Google та будь-якого месенджера, котрий є на телефоні чи планшеті споживача [11].

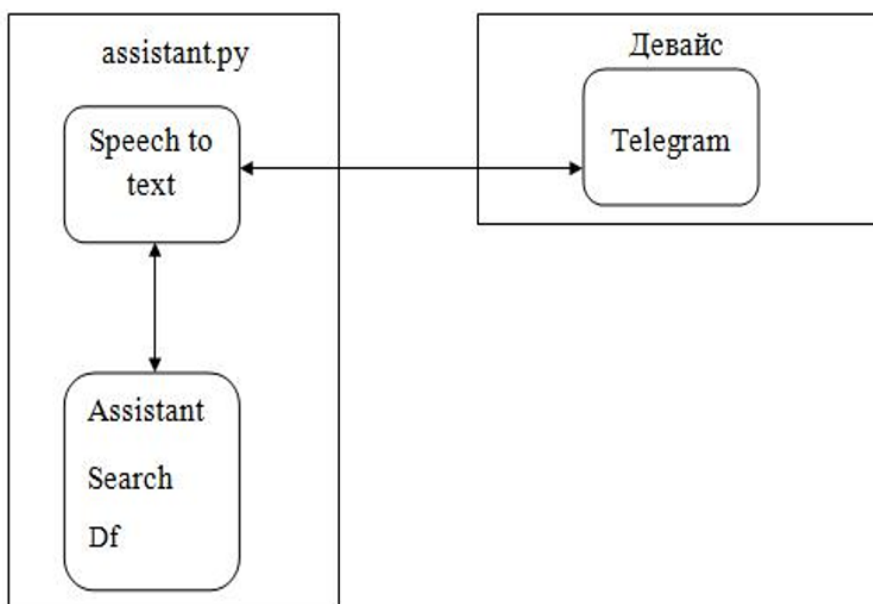


Рисунок 1.1 – Структура чат-боту

В структурі ми маємо можливість чітко побачити принципи роботи самого бота. Месенджер Telegram відіграє одну із ключових функцій платформи для розроблення чат бота, а також його розміщення на ресурсі assistant.py. Speech-to-text recognition виступає синтезом мови, котра після оброблення відео чи аудіо робить запити на спеціальні сервіси Assistant, Search, Dialogflow. Після оброблення запиту сервіси відправляють результати вже на смартфон чи планшет, а результат залишається в боті, де і був запит. Сам пошук відбувається у вигляді спілкування [11].

1.5. Алгоритм створення чат-ботів

Створення бота передбачає великі зусилля зі сторони людини та всіх етапах розробки. Головні зусилля в засновуванні чат-боту припадає на навчання цього бота розуміти причини звернення користувача (це понад 24%) та побудову моделі діалогу (13%). Якщо ж брати до уваги навички, котрі мають потребу при заснуванні бота, то їх ми можемо виділити наступним чином:

- Навички бізнесу – 11%.
- Власники продукту – 11%.
- Знання технології – 15%.
- Проектування – 15%.
- Кодування – 27%.

Задля створення бота, варто вирахувати, які саме завдання він вирішуватиме, обрати платформу для його роботи, сервіс для розробки та сервіс звідки він буде працювати. Окрім цього варто виділити декілька етапів [8]:

1. Вимоги:

- Познайомитись із предметною областю для бота та визначити головні завдання, що він вирішуватиме.
- Визначити вимоги до розробки бота та обрати платформу, через яку буде відбуватись спілкування.

2. Розробка чат-бота

- Заснувати архітектуру комунікації, котра дасть змогу збудувати ієрархію контенту для розмов.
- Зібрати інформацію, котра потрібна для повного діалогу, щоб можна було навчити бота розуміти різні варіанти того ж самого поняття і дати конкретні відповіді.
- Створити схему ведення діалогу на основі сценарію чи умов, згідно яких бот матиме можливість відповідати певним способом.

- Здійснити моделювання структур діалогів, чітко опрацьовуючи всі відповіді з чатів, задля того щоб спілкування було максимально природнім.

3. Підключення

- Здійснити інтеграцію з іншим сервісом.
- Провести тестування роботи чат-бота.
- Активувати його.

4. Аналізування роботи та виправлення дефектів.

- Проаналізувати інформацію стосовно роботи чат-бота.
- Здійснити оптимізацію поведінки чат-бота. Наприклад, додати нові питання, покращити класифікацію різних понять та їх логіку, додавати більше відповідей і так далі.

Отже, розробка чат-ботів володіє певною специфікою та вимагає багато зусиль зі сторони людини і великого кола професійних навичок.

РОЗДІЛ 2. ФОРМУЛЮВАННЯ ФУНКЦІОНАЛУ ТА АНАЛІЗ ТЕХНОЛОГІЙ

2.1. Постановка задачі

Основна мета – здійснити розробку інформаційного бота в месенджері Telegram. Ця соціальна мережа є першою, що створила можливість розробки ботів. Головним завданням новоствореного боту є пошук інформації про погодні умови у певному місці та прогноз на певний проміжок часу.

Даний бот виконує наступні функції:

- Пошук потрібної інформації при допомозі тексту.
- Симулятор живого спілкування із користувачем.

Створений нами чат бот володіє також певними особливостями, такими як:

- Наявність англійської мови в інтерфейсі.
- Обробка запитів будь якою мовою.
- Обробка повідомлень з помилками.
- Можливість переглядати прогноз погоди навіть за певною адресою.
- Декілька варіантів отримання прогнозу погодних умов із абсолютно різними ступенями деталізованості.
- Перевірка боту на помилки у запитах.

Цей бот має можливість зацікавити всіх не лише завдяки його реалізації, а й завдяки можливості його подальших вдосконалень. Бот має безмежні можливості для вдосконалення наступними способами:

- Додавляти нові мови задля розпізнавання.
- Додавляти новини.
- Покращувати команди для розпізнавання.
- Додавляти моделі поведінки бота.

- Створювати нові продукти різних сервісів та компаній (наприклад Google).

Боту необхідно мати чіткий та максимально зручний інструмент із пошуку чи відображення потрібної інформації.

Аналізуючи предметну область та аналоги даної системи інформації, ми можемо представити бот як інформаційну систему в інформаційній системі. Теперішні аналоги дали можливість зрозуміти, що така розробка буде мати актуальність та потрібність у щоденному використанні. Всі вищеперераховані пункти дають зрозуміти, що у Telegram ми можемо знайти бота будь-якого типу, однак не інформаційного з точки зору збору чи пошуку інформації.

Розроблений нами бот зобов'язаний давати можливість споживачам здійснити автоматизацію пошуку. Чат-бот має мати абсолютно простий вигляд без різних надбудов, ключову панель із клавіатурою та додатково кнопку для голосового повідомлення.

Враховуючи, що будуть використані API сервіси від Google мовою Python, то структура буде прописана в першу чергу мовою програмування саме Python.

Ключовою вимогою до споживачів бота є наявність в телефоні, на планшеті чи в ноутбучі додатку Telegram та наявність доступу до мережі інтернет.

2.2. Обґрунтування вибору засобів розробки

Python – це мова, котра була вибрана нами для чат-боту і даний вибір було зроблено із наступних причин:

- Python – це найбільш універсальна мова програмування.
- Python – досить лаконічна мова програмування.
- Python – найбільш популярна мова програмування.
- pyTelegramBotAPI для Python – це так звана бібліотека, що використовується задля програмування бота і є надзвичайно зручною для поставленого завдання.

Інтерфейс користувача чат-боту в Telegram реалізовано з допомогою бібліотеки `pyTelegramBotAPI`, яка містить наступні методи, які мають загальну назву `message handlers`:

- **`bot.message_handler(commands=['команда'])`** – для обробки повідомлень, що містять команди для бота. Командами при взаємодії користувача із чат ботом прийнято називати зарезервовані слова або сполучення кількох слів без використання пробілів (наприклад, з допомогою `camelCase`, `PascalCase` або `snake_case`), які починаються із символу «слеш».

Прикладом команди що використовується у моїй роботі, є `/start`. При отриманні цієї команди ботом формується та надсилається користувачу повідомлення, що містить вітання та загальну інформацію про можливості використання бота.

- **`bot.message_handler(content_types=['text','location'])`** – для обробки повідомлень, що мають тип «текст» та «геолокація». Цей метод є основним у даному проекті, оскільки розуміння запитів «по суті» та пошук відбувається саме при отриманні ботом таких типів повідомлень.

При отриманні текстового повідомлення виокремлюється та надсилається до `Google GeocodingAPI`, де задіюється метод `Geocoding (latitude/longitude lookup)` для пошуку за його текстом. Цей текст використовується для отримання адреси та геолокації (широти та довготи) об'єкту на мапі. Алгоритм отримання адреси та геолокації сервісом `GeocodingAPI`, що є частиною `Google Maps API`, є окремим важливим етапом. Для цього використовуються сукупність даних користувачів, даних держав та даних, отриманих корпорацією `Google` самостійно з допомогою працівників.

Для видачі найвідповідніших результатів використовується також машинне навчання. Використання запиту до `GeocodingAPI` дозволяє розпізнавати запити користувача, написані будь-якою мовою світу, з одруківками або навіть в хибній розкладці клавіатури. Таким чином шанси отримання інформації щодо описаного користувачем місця максимізуються. З

досвіду тестування можу зробити висновок, що не знайти інформацію про певне місце значно складніше, аніж знайти.

При отриманні повідомлення з геолокацією, бот також звертається до GeocodingAPI, але передає дані про широту та довготу обраного на мапі місця і ініціюється метод Reverse geocoding. Даний метод повертає дані про адресу, яка відповідає даному розташуванню. В такому випадку розпізнавання запиту зводиться до виокремлення із повідомлення користувача даних про координати обраного ним місця.

2.3. Черговість виконання функцій

Бот володіє декількома точками виходу та входу, а окремо обробляються наступні:

- /start – з’являється на екрані повідомлення із вітанням.
- /help – з’являється чітка та детальна довідка про особливості та можливості користування ботом.
- /about – розповідає про розробника чат боту.
- Повідомлення із адресою або ж назвою певного місця чи просто повідомлення із геолокацією – з’являється повідомлення із інформацією про назву місця, стосовно якого і відбувся пошук, а також інформацією про погоду в певній потрібній користувачу локації.

Додатково знизу існує 7 інлайн клавiш із варіантами отримання прогнозу погоди, такі як:

- прогноз погоди на найближчі 6 годин погодніно;
- прогноз погоди на найближчі 12 годин погодніно;
- прогноз погоди на найближчі 24 годин погодніно;
- прогноз погоди на найближчі 48 годин погодніно;
- прогноз погоди на найближчі 3 дні погодніно;
- прогноз погоди на найближчі 6 днів погодніно;

- прогноз погоди на 7 днів.

- Інлайн клавіша з одним із варіантів прогнозу погодних умов для певного місця – з’являється декілька повідомлень із прогнозом погодних умов.

Обробка вхідного повідомлення та повідомлення з локацією робиться завдяки функції `handle_text`, що отримує параметр `message` із важливою інформацією про отримане повідомлення для чат-боту.

Рішення зробити об’єднання обробки текстового повідомлення та локації має зв’язок з тим, що з обох точок зору (і користувача, і програми) оброблення повідомлень є практично аналогічне. З точки зору користувачів можна побачити відмінність у типі повідомлення, що було відправлене, а програмно відрізняються тільки ті параметри, що передаються через функції `search`.

Також цікавим у даній гілці було обирання даних зворотного виклику для інлайн клавіш. Так як Телеграм має доволі суворі обмеження з приводу обсягу інформації у цих даних, було обрано кодування «тип інформації ”&” `place_id` ”&” число запитних періодів».

Ми маємо можливість виділити наступні етапи оброблення повідомлення:

1. Запуск функції `search` модулю `geocoding` із параметрами, котрі напряду залежать від типу самого повідомлення.

2. Якщо пошук місця пройде без жодних помилок, то відбувається запуск наступної функції `request` із параметрами: ключ API, географічна довгота та широта, тип запиту тощо. Якщо ж пошук локації не дав жодних результатів, то користувач отримує повідомлення певного типу.

3. Якщо дані про поточні погодні умови успішно віднайдено, тоді запускається функція `current` модулю `messageText`, якій надалі передаються результати запиту до `Geocoding API` та `One Call API`. Якщо ж дані про погодні умови не знайдено, тоді споживач бота отримує певного змісту повідомлення.

4. Після остаточного формування тексту повідомлення бот відправляє повідомлення із даним текстом та інлайн кнопками, котрі формуються функцією `gen_markup`.

Оброблення натискання споживачами інлайн кнопки робиться функцією `callback_query_handler`, яка отримує дані зворотного відліку.

Обробка вказаної дії користувача відбувається зазвичай лінійно, єдина програмна відмінність у формуванні відповіді на запит погодинного та поденного прогнозу погоди полягає у виклику відповідних функцій `hourly` або `daily` модулю `messageText`. Це пов'язано з тим, що дані, які отримуються від API, для цих прогнозів, значно відрізняються одне від одного та потребують різної обробки.

Повідомлення, котрі надсилаються користувачу в результаті оброблення запиту, за своїми розмірами значно перевищують ліміт у 3 тис. символів, дозволені в телеграмі для надсилання в повідомленні, і саме тому перед відправленням виконується розбивання тексту розміром в 3 тис. символів на частини.

Етапи оброблення натискання користувачем інлайн – клавіші:

1. Дані виклику (в першу чергу зворотнього) розбиваються із рядка у масив і далі у певні змінні.
2. Виконується функція `search` модулю `geocoding` із параметрами у вигляді API ключа та ідентифікатора місця.
3. Із результату запитів відбувається збереження інформації про географічні місця та їх назву.
4. Додатково виконується функція `request` модулю `owm` із наступними параметрами – ключ API, тип запиту, географічна широта та довгота.
5. В залежності від виду прогнозу погоди запущено функцію `hourly` або `daily` модулю `messageText` із такими параметрами як адреса місця, прогнозом.
6. Тексти повідомлення поділяються на масив текстів до 3 тис. знаків та відправляється в певному циклі.

2.4. Налаштування робочих параметрів функцій

Параметри, які необхідно налаштувати перед запуском чат-боту без внесення змін до його роботи, зводяться до вводу наступних даних у модуль `config.py`:

1. `botToken` – константа, яка повинна містити токен боту. Токен видається сервісом Telegram у системному чат-боті BotFather при створенні нового боту.
2. `owmToken` – константа, яка містить токен від сервісу OpenWeatherMap. Отримати його можна безкоштовно і користуватись до тих пір, поки не буде перевищено ліміти безкоштовного користування.
3. `googleToken` – константа, яка містить токен від сервісу Google Maps Platform і отримується на сайті Google Developers.

2.5. Вимоги до запитів та побудова відповідей

Користувач може надсилати запити у двох форматах:

- текстовому;
- повідомлення із геолокацією.

У випадку текстового повідомлення, може використовуватись будь-яка мова (протестовано на українській, англійській, російській та китайській мовах). Повідомлення може містити одруківки, однак із ними ймовірність коректного розуміння запиту зменшується. Повідомлення може бути написане у хибній розкладці клавіатури.

Повідомлення із геолокацією не має практичних обмежень, оскільки будь-яке допустиме мапами при формуванні та надсиланні запиту місце буде існувати у світі, а отже – можна отримати дані щодо погоди у ньому.

Окремим типом запиту можна виділити також використання інлайн клавіш для отримання деталізованого прогнозу погоди на 6, 12, 24 або 48 годин чи на 3 або п'ять днів чи на тиждень. Користувач може натискати на інлайн

клавіші під будь-яким повідомленням бота, навіть якщо воно не буде останнім у чаті і отримати інформацію саме щодо місця, якого стосувалось це повідомлення.

2.6. Склад і структура програмного забезпечення системи

Побудова структури даного проекту заснована на модульній парадигмі програмування. Вона в першу чергу орієнтована на зменшення складності продуктів та можливості для зміни окремих рішень із одних проектів на інші. Побудована внаслідок модульна архітектура займається підкресленням поділу функціональності програми на інші зовсім незалежні змінні модулі, а саме таких, що кожен із них володіє всім потрібним, щоб виконати лише один аспект потрібної функціональності. Модулі, переважно, включені у програму завдяки інтерфейсу.

Парадигма будується на механізмах, котрі дають можливість будувати програму із певних модулів шляхом забезпечення:

- створення окремих елементів для програми в певних файлах із загальною можливістю здійснити об'єднання модулів в певний програмний продукт на етапах виконання роботи.
- Логічного завершення модулів, в той час коли кожен певний самостійний модуль містить елементи, котрі займаються реалізацією деяких логічно закінчених алгоритмів.
- Обмеженість модулів у розмірі та функціональності, що дає змогу найкращим способом здійснити організацію модулів в програму чи переносити їх в модулі між програмами.
- Замкнутості модулів, тоді коли всі модулі мають власний чітко визначений інтерфейс до інших модулів та використовує інші модулі через чітко визначений інтерфейс та повній залежності одних модулів від інших значень даних, що можуть міститись у інших модулях.

- Узагальненість модулів, тоді коли параметри оброблення та дані для їх обробки мають властивість передаватись з інших модулів чи файлів конфігурації, а не містяться як які-небудь змінні чи константи.

- Мінімізації модулів, коли кожен модуль надає якомога простий інтерфейс.

Даний проект складається із наступних модулів.

2.6.1. Модуль main.py

Модуль main.py є головним модулем нашої програми, котрий напряду володіє взаємодією із споживачем, здійснюючи обробку вхідних даних та відправляючи результат користувачам. Задля цього модуль здійснює використання інших наявних у проекті модулів.

2.6.2. Модуль geocoding.py

Даний модуль володіє функцією за допомогою якої використовуються Reverse Geocoding (Address Lookup), Geocoding (Latitude/Longitude Lookup) і Retrieving an Address for a Place ID запитів до Geocoding API від google.

2.6.3. Модуль messageText.py

Даний модуль володіє функцією, котра займається формуванням тексту повідомлення, який і буде відправлений користувачам.

2.6.4. Модуль config.py

Цей модуль має змінні із токеном Telegram бота та ключі Openweathermap API та Google API.

2.6.5. Модуль requirements.txt

Файл requirements.txt заповнений інформацією про модулі, які варто встановити при допомозі pip.

З повним кодом реалізації проекту можна детальніше ознайомитись у розділі Додаток А.

2.7. Опис API, що використовуються в роботі

Даний бот утворений задля допомоги у визначенні теперішніх погодних умов та отримувати майбутні прогнози у певній раніше визначеній локації.

Бот використовує два API: One Call API від OpenWeatherMap та Geocoding API від Google.

2.7.1. Geocoding API

Для користування Geocoding API (рис. 2.1) варто здійснити реєстрацію на спеціальному сервісі Google Cloud Platform. Ця платформа надає доступ до величезного набору хмарних програм та служб і має тестовий безкоштовний період в один рік.

Geocoding API в першу чергу використовується для уточнення координат та назви запиту. В самій програмі будуть використовуватись 3 типи запитів для Geocoding API, а саме:

1. Geocoding (Latitude/Longitude Lookup) – запит до API за пошуковим рядком, API в даному випадку перепитує назву місця і надає географічні координати та place_id;
2. Reverse Geocoding (Address Lookup) – запит до API за точними координатами місця. API за координатами визначає назву міста та повертає place_id;

3. Retrieving an Address for a Place ID – запит до API за параметром `place_id`. API займається поверненням назви місця та географічних координат.

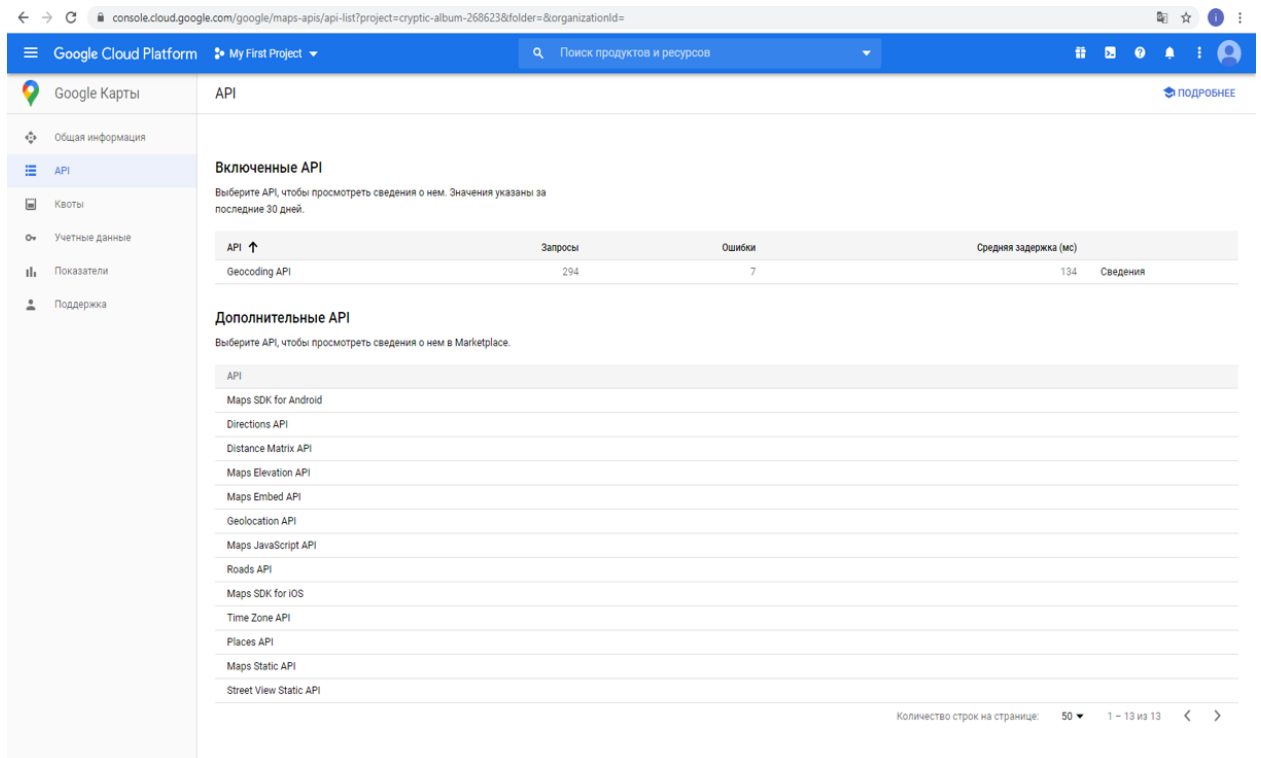


Рисунок 2.1 – Вікно, що містить інформацію про використання підключених API від Google

Ключовою особливістю цієї API є те, що назвою місця має можливість бути як будь-яка адреса так і назва якого-небудь закладу, який вказано в пошуковнику.

Форматами запиту до API виступають:

<https://maps.googleapis.com/maps/api/geocode/outputFormat?parameters>

де, *outputFormat*, може приймати значення:

- Json;
- Xml.

Параметри запиту.

Маючи залежність від виду запиту, параметри мають можливість приймати наступний вигляд:

`address={your_address}&key={your_key}`

де назва параметру `address` визначає, що виконується запит до Geocoding (Latitude/Longitude Lookup), а параметри:

- your_address – назва місця, для якого відбувається пошук;
- your_key – особистий ключ google API.

Або:

latlng={lat},{lng}&key={your_key}

де назва параметру latlng визначає, що виконується запит до Reverse Geocoding (Address Lookup), а параметри;

- lat – широта, згідно з якою знаходиться місце;
- lng – довгота, згідно з якою знаходиться місце;
- your_key – особистий ключ google API.

Або:

place_id={place_id}&key={key}

де назва параметру place_id визначає, що виконується запит до Retrieving an Address for a Place ID, а параметри:

- place_id – google id місця, для якого відбувається пошук;
- your_key – особистий ключ google API.

2.7.2. One Call API

Задля отримання даних про погодні умови нами було обрано спеціальний сервіс Open Weather Map (рис. 2.2). Його головними перевагами для нас є надійність та безкоштовний режим, який не має ліміту по часу і володіє лише обмеженнями у 1000 викликів One Call API протягом дня і цього нам повністю достатньо для користування ботом достатньо великої кількості споживачів.

One Call API нами використовується задля отримання наступних типів інформації:

1. поточний погодні стан.
2. Похвилинний прогноз для однієї години.
3. Погодинний прогноз для найближчих 48 годин.
4. Прогноз для семи днів.
5. Погодні дані за останні 5 днів.

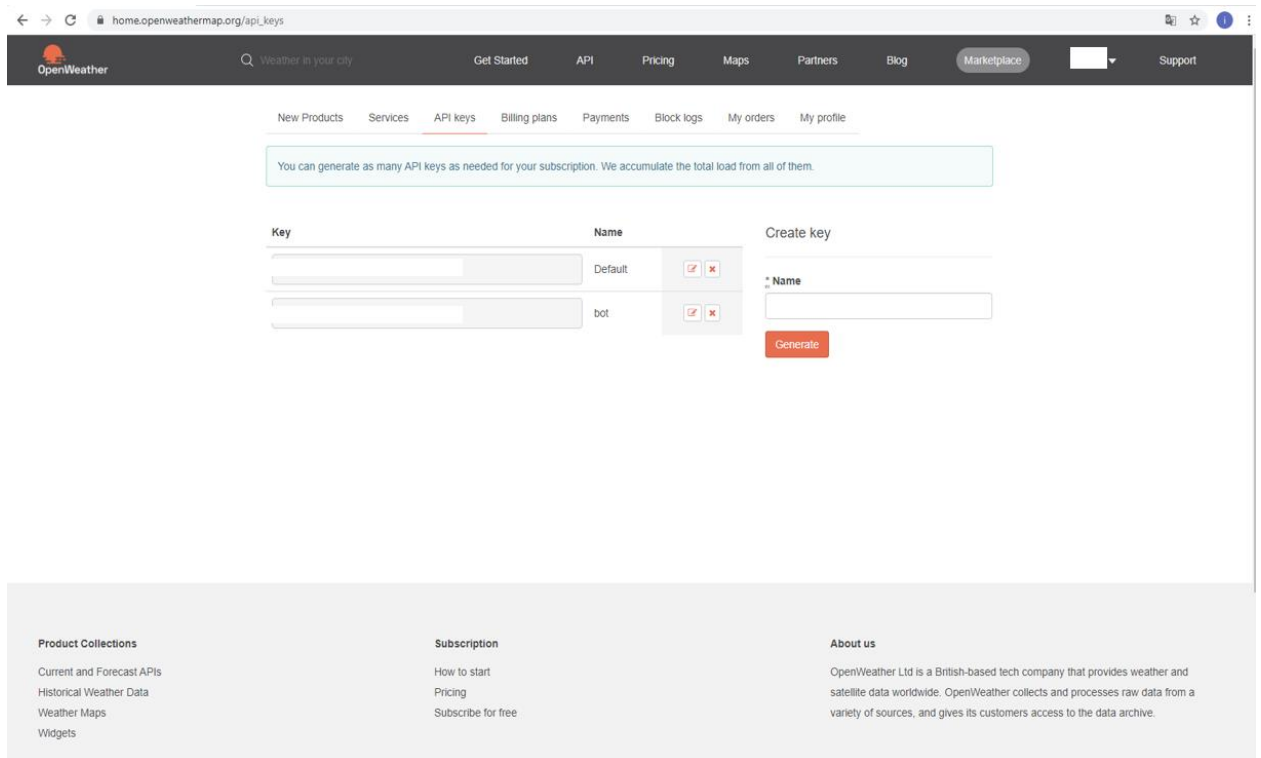


Рисунок 2.2 - Сторінка керування API ключами сервісу Open Weather Map

Перелік наступних факторів є особливістю даної API:

- За замовчуванням API має можливість повернути абсолютно всю інформацію про вказані типи інформації.
- Пошук даних відбувається лише завдяки географічним координатам.

Формат запити до API:

<https://api.openweathermap.org/data/2.5/onecall?lat={lat}&lon={lon}&exclude={part}&appid={key}>

Параметри запити:

- lat – широта, згідно якої знаходиться місце;
- lon – довгота, згідно якої знаходиться місце;
- part – частина відповіді, котра виключається. Вона може приймати значення hourly, daily, current, minutely та їх поєднання;
- key – персональний ключ openweathermap API.

2.8. Інтерфейс користувача та результати тестування програмного забезпечення

Оскільки чат бот є програмною одиницею, інтегрованою в месенджер Telegram, основні елементи інтерфейсу він успадковує від месенджера. Безпосередньо взаємодія з ботом може відбуватись наступними способами:

- надсилання текстового повідомлення;
- натискання інлайн клавiші під повідомленням із відповіддю бота;
- надсилання повідомлення із геолокацією.

Детальніше ознайомитись із інтерфейсом роботи чат-бота можна на скріншотах тестування роботи проекту (Додаток Б).

2.9. Особливості та рекомендації щодо використання програмного забезпечення

Даний проект є доволі універсальним у використанні: месенджер Telegram має клієнти для різних пристроїв та операційних систем. Однак використання Telegram на даний момент також накладає обмеження у використанні повідомлень з геолокацією: вони доступні лише для користувачів мобільних пристроїв, як телефони і планшети. Версія Telegram Desktop не містить цього функціоналу, як і браузерні версії Telegram.

У свою чергу, Google Maps містять інформацію про мільйони локацій. Деякі з них мають однакову назву. Тому ще однією особливістю формування запитів є те, що необхідно зазначати адресу максимально однозначно. Тобто, ймовірність релевантного розуміння запиту «Епіцентр» є невисокою: гугл-мапи містять інформацію про десятки гіпермаркетів та інших локацій із назвою Епіцентр. Тому доречнішим є використання запиту «Епцентр Позняки». Більш деталізований запит навіть із одруківкою є результативнішим за менш деталізований.

OpenWeatherMap дає доступ до даних про погоду у будь-якій точці світу. Використання запиту із точною локацією призводить до більшої точності прогнозу погоди. Наприклад, у різних частинах мегаполісу температура повітря може відрізнятись на декілька градусів. У частині міста може йти дощ, в той час, як у іншій буде сонячна погода. Тому рекомендується зазначати положення з якомога більшою точністю для отримання найбільш відповідного прогнозу. Але користувачу завжди варто пам'ятати, що прогнозована погода у будь-якого провайдера може не відповідати реальній, тож на жоден прогноз не можна покладатись на всі 100%.

Мова інтерфейсу чат-бота – англійська, проте використання емодзі робить його доступним навіть для людини, що зовсім не знайома із цією мовою.

Якщо ви плануєте часто переглядати погоду за певною адресою, її можна писати як команду, починаючи з «/», наприклад «/Київ». Так можна буде повторно надіслати повідомлення, просто натиснувши на команду.

ВИСНОВКИ

Процес цифровізації економічних процесів та нова тенденція з орієнтації системи підтримки прийняття рішень за використання смартфонів ставлять завдання розроблення нових чат-ботів.

Беручи до уваги аналіз предметної області, в першу чергу дослідження актуальності та потрібності чат ботів в сучасному світі та їх використання в додатках (месенджерах), нами було вирішено здійснити розробку бота, котрий матиме можливість дати відповідь на задані інформаційні запити.

Аналізуючи аналоги ми мали можливість зрозуміти, які саме типи та категорії новітніх чат-ботів можна знайти в мережі інтернет. Завдячуючи даному аналізу нами було вирішено запитання актуальності боту та загалом його працездатності. Ця розробка є безкоштовною на 100% та буде у відкритому доступі для абсолютно усіх верств населення, котрі володіють смартфоном, мають можливість підключення інтернету та наявність додатку телеграм.

Даний бот виконує наступні функції:

- Пошук потрібної інформації при допомозі тексту.
- Симулятор живого спілкування із користувачем.

Створений нами чат бот володіє також певними особливостями, такими як:

- Наявність англійської мови в інтерфейсі.
- Обробка запитів будь якою мовою.
- Обробка повідомлень з помилками.
- Можливість переглядати прогноз погоди навіть за певною адресою.
- Декілька варіантів отримання прогнозу погодніх умов із абсолютно різними ступенями деталізованості.
- Перевірка боту на помилки у запитах.

Наш бот має можливість зацікавити всіх не лише завдяки його реалізації, а й завдяки можливості його подальших вдосконалень. Бот має безмежні можливості вдосконалюватись наступними способами:

- Додавляти нові мови задля розпізнавання.
- Додавляти новини.
- Покращувати команди для розпізнавання.
- Додавляти моделі поведінки бота.
- Створювати нові продукти різних сервісів та компаній (наприклад Google).

Бот має можливість зацікавити не лише з точки зору його реалізації, й з точки зору можливості майбутнього вдосконалення.

Бот створений при допомозі мови Python та сервісів від Google. Дослідна експлуатація чат-боту довела конструктивність підходу, а також було намічено певні шляхи його вдосконалення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Лутц М. Программирование на Python, том II, 4-е издание. Пер. сангл. СПб.: Символ-Плюс, 2011. 992 с.
2. Ушакова І.О. Підходи до створення інтелектуальних чат-ботів. *Системи обробки інформації*, 2019. № 2. С. 76-83.
3. Инструкция: Как создавать ботов в Telegram. URL: <https://habr.com/ru/post/262247/>
4. Как использовать чат-боты в бизнесе: 5 идей и 5 кейсов. URL: <https://vc.ru/services/93850-kak-ispolzovat-chat-boty-vbiznese-5-idey-i-5-keysov>
5. Как создать Telegram бота с помощью Python. URL: <https://medium.com/nuances-of-programming>
6. Пишем telegram-бота на python с помощью библиотеки telebot. URL: <https://habr.com/ru/post/448310/>
7. Python-telegram-bot. URL: <https://github.com/python-telegram-bot/python-telegram-bot>
8. Створюємо Telegram-бота на Python. Частина 1. URL: <https://codeguida.com/post/410>
9. Чат-бот для бизнеса. URL: <https://texterra.ru/blog/chat-boty-dlya-biznesa-stsenarii-ispolzovaniya-servisy-a-takzhe-udachnye-i-ne-ochen-keysy-kompaniy.html>
10. Чат-бот - что это такое, как работает, виды, преимущества и недостатки, как создать. URL: <http://kak-bog.ru/chat-bot-cto-eto-takoe-kak-rabotaet-vidy-preimushchestva>
11. Чат-бот: зачем создавать, кому использовать. URL: <https://blog.click.ru/growthhacking/chat-bot-na-sajte-zachem-sozdavat-komu-ispolzovat-i-kak-nastroit/>
12. Чат-боты – кто они и что умеют? URL: <https://efsol.ru/articles/messendzhery-i-chat-boty-dlya-biznesadostavki.html>

13. Чат-боты: обзор и состояние технологий в отрасли. URL: <http://nlpx.net/archives/425>
14. Чем отличаются чат-боты в Telegram, WhatsApp, Viber, Facebook, VK. URL: <https://www.rpv-bot.ru/chem-otlichaetsyachat-bot-v-telegram-whatsapp-vk-viber-facebook>
15. 13 крутых онлайн-сервисов для создания чат-ботов. URL: <https://habr.com/ru/company/click/blog/567446/>
16. 50 лучших чат-ботов для Telegram. URL: <https://uip.me/2016/04/50-popular-telegram-bots/>
17. AI and Chatbots in Education. URL: <https://chatbotsmagazine.com/ai-and-chatbots-in-education-what-does-the-futurehold-9772f5c13960?gi=ddd4e7e0cf45>
18. Aimylogic. URL: <https://aimylogic.com/ru>
19. Botmother. URL: <https://botmother.com/ru>
20. Botsify. URL: <https://botsify.com/>
21. Chatbot. URL: <https://www.chatbot.com/>
22. Dexter. URL: <https://rundexter.com/>
23. Exploring chatbots for news URL: <https://medium.com/nos-digital/exploring-chatbots-for-news-7f12d226c53d>
24. Fasttrack. URL: <https://fstrk.io/>
25. Flow XO. URL: <https://flowxo.com/>
26. Gerabot. URL: <https://gerabot.com/constructor>
27. TelegramAPIs. URL: <https://core.telegram.org/>

ДОДАТОК А

Код програмної реалізації

```
# -*- coding: UTF-8 -*-
"""Імпортуємо модуль telebot, котрий дозволяє зручно взаємодіяти з telegram bot API мовою python та модулів,
необхідних для створення інлайн клавiш та розбивання
довгого тексту на декілька повідомлень та необхідні для роботи інлайн клавiш і розбивки повідомлення підмодулі"""
import telebot
from telebot.types import InlineKeyboardMarkup, InlineKeyboardButton
from telebot import util
#Імпортуємо модуль config, в якому зберігаються отримані API токени від телеграм at weatherstack
import config
#Імпортуємо модуль geocoding, який виконує запити до Geocoding API
import geocoding
#Імпортуємо модуль owm, який виконує запити до openweathermap one call api
import owm
#Імпортуємо модуль messageText, який форматує дані у читабельний текст для відправки користувачу
import messageText

#Ініціалізуємо бота
bot = telebot.TeleBot(config.botToken)

#Обробка команди start
@bot.message_handler(commands=['start'])
def handle_text(message):
    bot.send_message(message.chat.id, 'Hello, ' + str(message.from_user.first_name) +
        '\n\nYou can send me location or address and I`ll reply with information about current weather. Also, you can
        get weather forecast using inline buttons.\
        \n\nHope, I`ll be useful for you :)')

#Обробка команди help
@bot.message_handler(commands=['help'])
def handle_text(message):
    bot.send_message(message.chat.id, 'You can send me text or location message and I will reply with information
        about current weather. \
        \n\nAlso, you can get weather forecast using inline buttons. \
        \n\nYou can use commands too, for example message "/Kyiv" is similar to "Kyiv" for me. \
        It will be usefull, if you are going to check weather for some location often.\
        \n\nNeed to say, I`m a bit multilingual: I can understand a lot of different languages. \
        You can ask me "Пекін", "Beijing" or "北京" and I`ll answer you. In English :)')
```

Рисунок 1 - Скріншот модулю main.py, частина 1

```

#Обробка команди about
@bot.message_handler(commands=['about'])
def handle_text(message):
    bot.send_message(message.chat.id, 'This bot is a diplomawork of Varzar Veronika Anatoliivna')

"""Функція для формування інлайн клавіатури.
В callback_data записується інформація про тип запиту (погодинно/денно), place_id та кількість одиниць обраного
типу періоду"""
def gen_markup(place_id):
    markup = InlineKeyboardMarkup()
    markup.add(InlineKeyboardButton('6h forecast', callback_data='hourly&' + place_id + '&6'),
                InlineKeyboardButton('12h forecast', callback_data='hourly&' + place_id + '&12'),
                InlineKeyboardButton('24h forecast', callback_data='hourly&' + place_id + '&24'),
                InlineKeyboardButton('48h forecast', callback_data='hourly&' + place_id + '&48'),
                InlineKeyboardButton('3day forecast', callback_data='daily&' + place_id + '&3'),
                InlineKeyboardButton('5day forecast', callback_data='daily&' + place_id + '&5'),
                InlineKeyboardButton('Week forecast', callback_data='daily&' + place_id + '&7'))
    return markup

#Обробка текстових повідомлень та повідомлень з геолокацією
@bot.message_handler(content_types=['text','location'])
def handle_text(message):
    #Перевірка типу вхідного повідомлення, залежно від якого здійснюються різні типи запитів до API погоди
    if message.content_type == 'text':
        #Запуск модулю geocoding для виконання пошуку коректної адреси та координат через geocoding API за текстом
        found = geocoding.search(config.googleToken, 'address', message.text)
    else:
        #Запуск модулю geocoding для виконання пошуку адреси через geocoding API за координатами
        found = geocoding.search(config.googleToken, 'latlng', str(message.location.latitude)+'_'+str(message.location.longitude))

    #Перевірка успішності пошуку розташування
    if found['status'] == 'OK':
        #Запуск модулю owm для запиту щодо поточної погоди
        current = owm.request(config.owmToken, found['location']['lat'], found['location']['lng'], 'current')
        #Перевірка успішності запиту про погоду
        if current['status'] == 'OK':
            #Запуск модулю messageText для форматування тексту повідомлення користувачу
            text = messageText.current(found, current)
            #Надсилання користувачу повідомлення з даними про поточну погоду та інлайн клавішами
            bot.send_message(message.chat.id, text, reply_markup=gen_markup(found['place_id']))
        else:
            #Надсилання користувачу повідомлення зі звітом про помилку пошуку погоди
            bot.send_message(message.chat.id, 'Weather research gone wrong, try to change your request')
    else:
        #Надсилання користувачу повідомлення зі звітом про помилку пошуку місця
        bot.send_message(message.chat.id, 'Place search gone wrong, try to change your request')

```

Рисунок 2 - Скріншот модулю main.py, частина 2

```

#Обробка натискання інлайн клавіш
@bot.callback_query_handler(func=lambda call: True)
def callback_query(call):
    #Декодування отриманих від клавіші даних
    splited = call.data.split('&')
    forecast_type = splited[0]
    place_id = splited[1]
    number = int(splited[2])
    #Пошук назви та координат місця за place_id
    found = geocoding.search(config.googleToken, 'place_id', place_id)
    address = found['name']
    latitude = found['location']['lat']
    longitude = found['location']['lng']

    forecast = owm.request(config.owmToken, found['location']['lat'], found['location']['lng'], forecast_type)
    #Перевірка типу прогнозу погоди (погодинно, або денно) та відповідний запит про прогноз
    if forecast_type == 'hourly':
        full_text = messageText.hourly(address, forecast, number)
    else:
        full_text = messageText.daily(address, forecast, number)
    #Якщо текст повідомлення більший за 3000 символів, його буде розбито на декілька повідомлень
    splitted_text = util.split_string(full_text, 3000)
    for text in splitted_text:
        bot.send_message(call.message.chat.id, text)

#Запускаємо бота
bot.polling(none_stop=True, interval=0)

```

Рисунок 3 - Скріншот модулю main.py, частина 3

```
# -*- coding: UTF-8 -*-
#Імпортуємо модуль requests, який потрібен для виконання запитів до geocoding API
import requests

#Функція, що виконуватиме запит та повертатиме результат
def search (key, type, query):
    #Формування параметрів запиту
    params = {'key':key, 'type': query}
    #Виконання запиту до weatherstack API та збереження відповіді
    api_result = requests.get('https://maps.googleapis.com/maps/api/geocode/json', params)
    api_response = api_result.json()
    #Перевірка статусу відповіді та формування звіту про помилку, якщо вона виникла
    if api_response['status'] == 'OK':
        result = {'name':api_response['results'][0]['formatted_address'], 'location':
            api_response['results'][0]['geometry']['location'],'status':'OK','place_id':api_response['results'][0]['place_id']}
    else:
        result = {'status':'fail'}

    return result
```

Рисунок 4 - Скріншот модулю geocoding.py

```
# -*- coding: UTF-8 -*-
#Імпортуємо підмодуль datetime модулю datetime, який потрібен для отримання інформації про час
from datetime import datetime
#Імпортуємо модуль pytz, який потрібен для коректного відображення часу з урахуванням часових поясів
import pytz

#Функція для форматування тексту про поточну погоду
def current (location, weather):
    location = location['name']
    description = weather['response']['current']['weather'][0]['description']
    temperature = '\n@Temperature: ' + str(weather['response']['current']['temp'])
    humidity = '\n@Humidity: ' + str(weather['response']['current']['humidity'])
    wind = '\n@Wind speed: ' + str(weather['response']['current']['wind_speed'])
    feelsLike = '\n@Feels like: ' + str(weather['response']['current']['feels_like'])
    text = 'Weather in ' + location + ' is: \n' + description + temperature + humidity + wind + feelsLike
    return text

#Функція для форматування тексту погодинного прогнозу погоди
def hourly (location, weather, number):
    text = str(number) + ' hour forecast for\n' + location + '\n'
    index=0
    #Цикл для повернення інформації лише про запитану кількість годин
    for hour in weather['response']['hourly']:
        index+=1
        time = datetime.fromtimestamp(hour['dt'], pytz.timezone(weather['response']['timezone'])).strftime('%d.%m.%Y, %H:%M')
        description = hour['weather'][0]['description']
        temperature = '\n@: ' + str(hour['temp'])
        humidity = '\n@: ' + str(hour['humidity'])
        wind = '\n@: ' + str(hour['wind_speed'])
        feelsLike = '\n@: ' + str(hour['feels_like'])
        text += '\n\nDate, time: ' + time + '\n' + description + temperature + humidity + wind + feelsLike
    if index == number:
        return text
```

Рисунок 5 - Скріншот модулю messageText.py, частина 1

```

#Функція для форматування тексту поденного прогнозу погоди
def daily (location, weather, number):
    text = str(number) + ' day forecast for\n' + location + '\n'
    index=0
    #Цикл для повернення інформації лише про запитану кількість днів
    for day in weather['response']['daily']:
        index+=1
        date = datetime.fromtimestamp(day['dt']).strftime('%d.%m.%Y')
        minTemp = str(day['temp']['min'])
        maxTemp = str(day['temp']['max'])
        description = day['weather'][0]['description']
        humidity = '\n☁: ' + str(day['humidity'])
        wind = '\n🌀: ' + str(day['wind_speed'])

        morningTemp = '\n🌅: ' + str(day['temp']['morn'])
        morningFeelsLike = '\n👉: ' + str(day['feels_like']['morn'])

        dayTemp = '\n🌞: ' + str(day['temp']['day'])
        dayFeelsLike = '\n👉: ' + str(day['feels_like']['day'])

        eveningTemp = '\n🌃: ' + str(day['temp']['eve'])
        eveningFeelsLike = '\n👉: ' + str(day['feels_like']['eve'])

        nightTemp = '\n🌙: ' + str(day['temp']['night'])
        nightFeelsLike = '\n👉: ' + str(day['feels_like']['night'])

    text += '\n\nDate: ' + date + '\n🌡Temperature between ' + minTemp + ' and ' + maxTemp + '\n' + description + humidity
    + '\nMorning: ' + morningTemp + morningFeelsLike + '\nDay: ' + dayTemp + dayFeelsLike + '\nEvening: ' + eveningTemp +
    eveningFeelsLike + '\nNight: ' + nightTemp + nightFeelsLike
    if index == number:
        return text

```

Рисунок 6 - Скріншот модулю messageText.py, частина 2

```

# -*- coding: UTF-8 -*-
#Зберігаємо ключі використуваних API
botToken
owmToken
googleToken

```

Рисунок 7 - Скріншот модулю config.py

```

pyTelegramBotAPI
pytz

```

Рисунок 8 - Скріншот модулю requirements.txt

ДОДАТОК Б

Приклади виконання роботи розробленого чат-бота

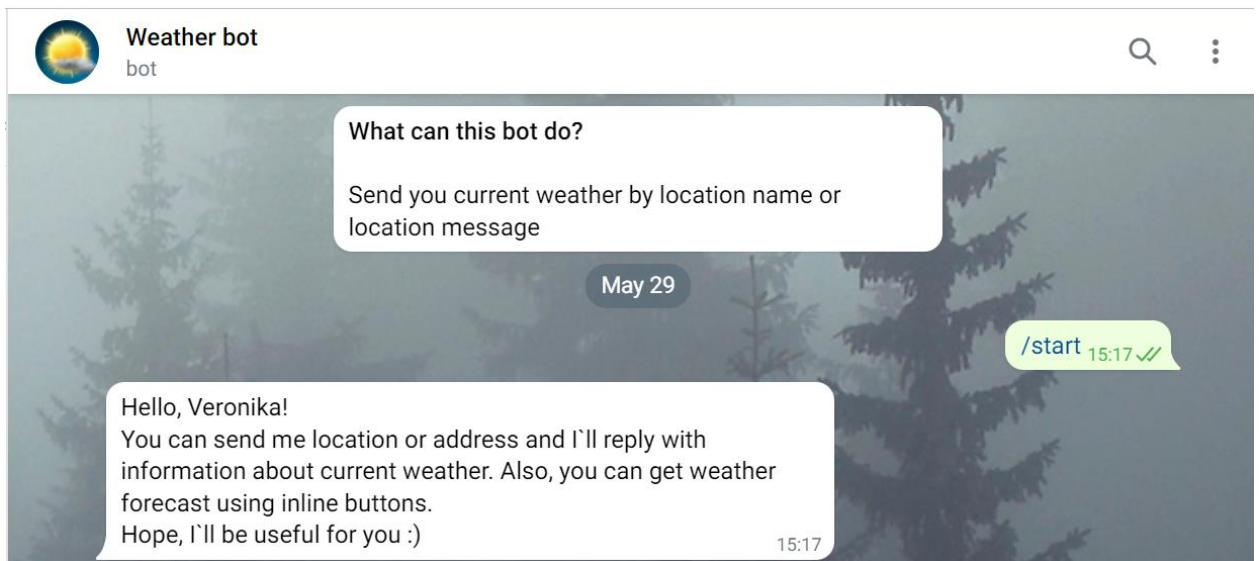


Рисунок 9 - Скріншот з привітанням бота

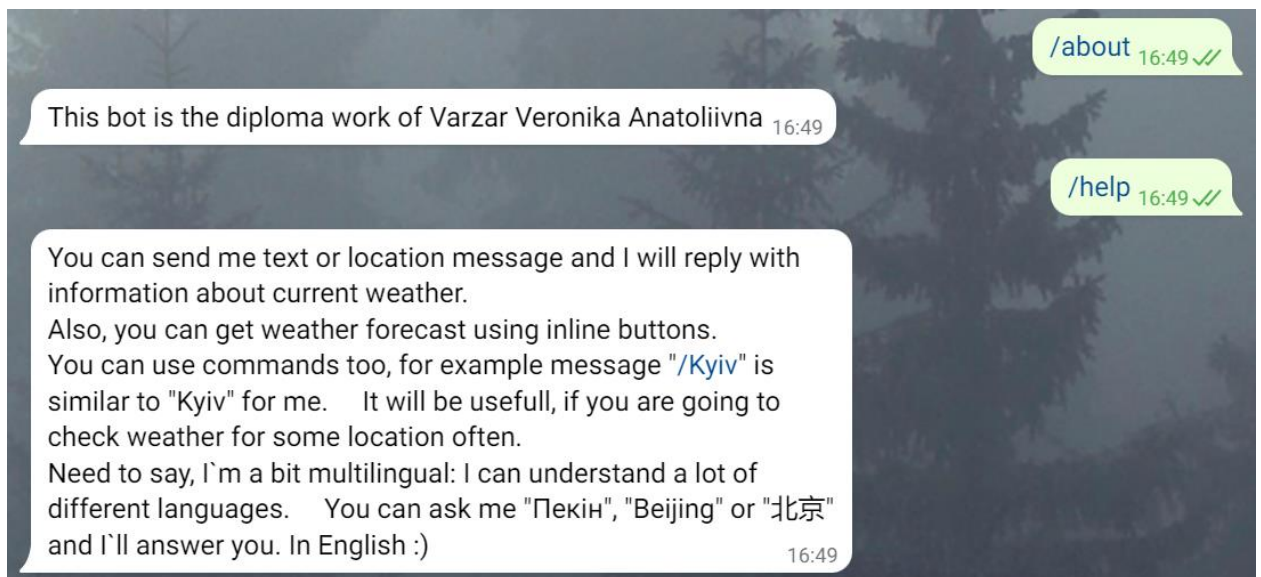


Рисунок 10 – Скріншот з відповіддю на команди /about та /help

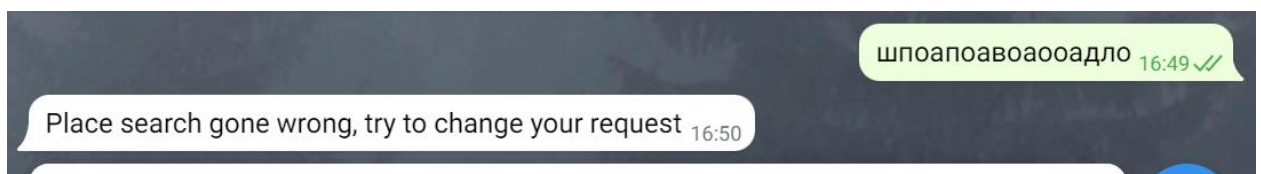


Рисунок 11 - Скріншот з відповіддю бота на випадковий набір символів

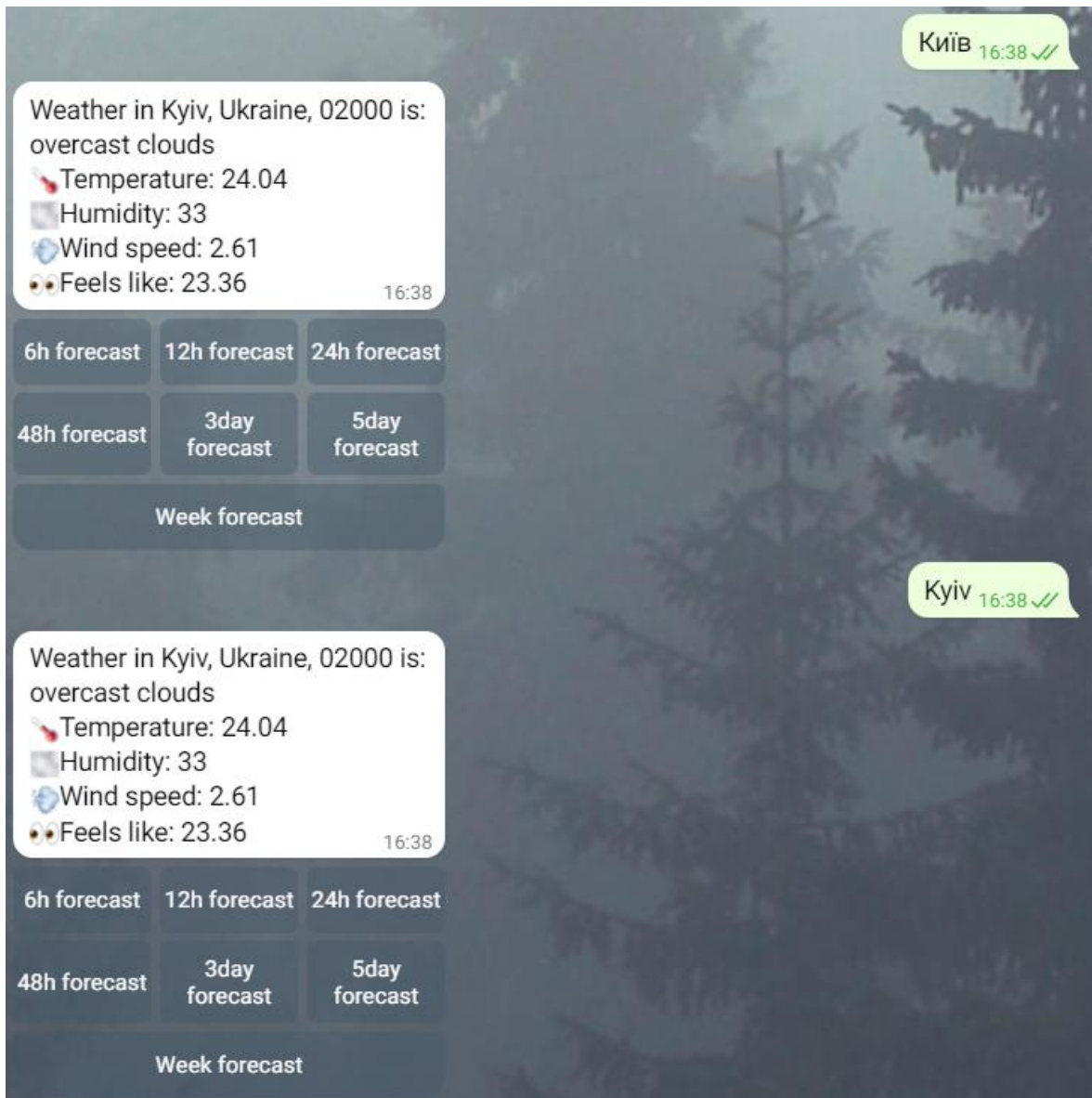


Рисунок 12 - Скріншот з отриманням даних про погоду українською та англійською мовами

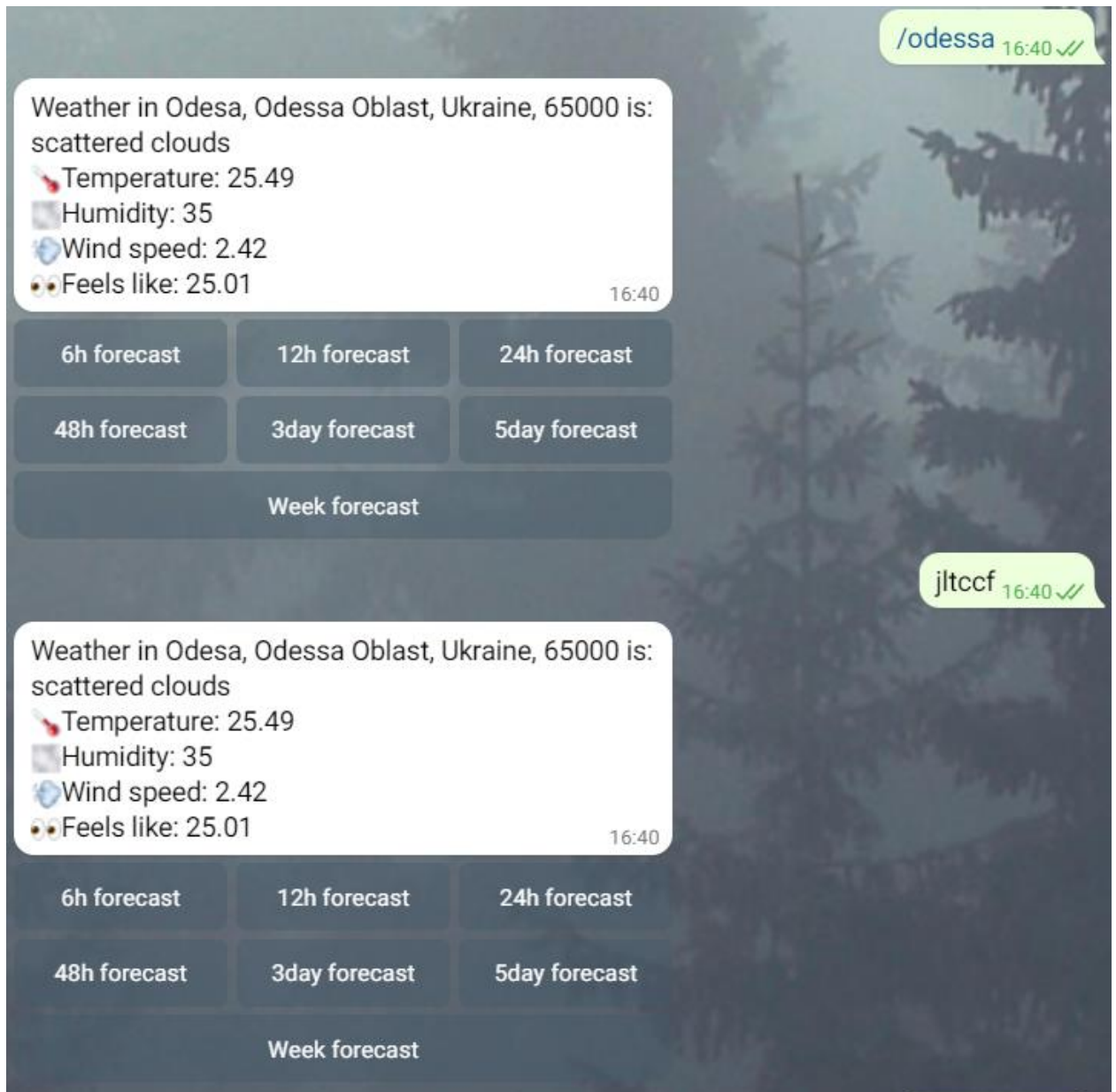


Рисунок 13 – Скріншот з отриманням даних про погоду на запити командою та з помилковою розкладкою клавіатури

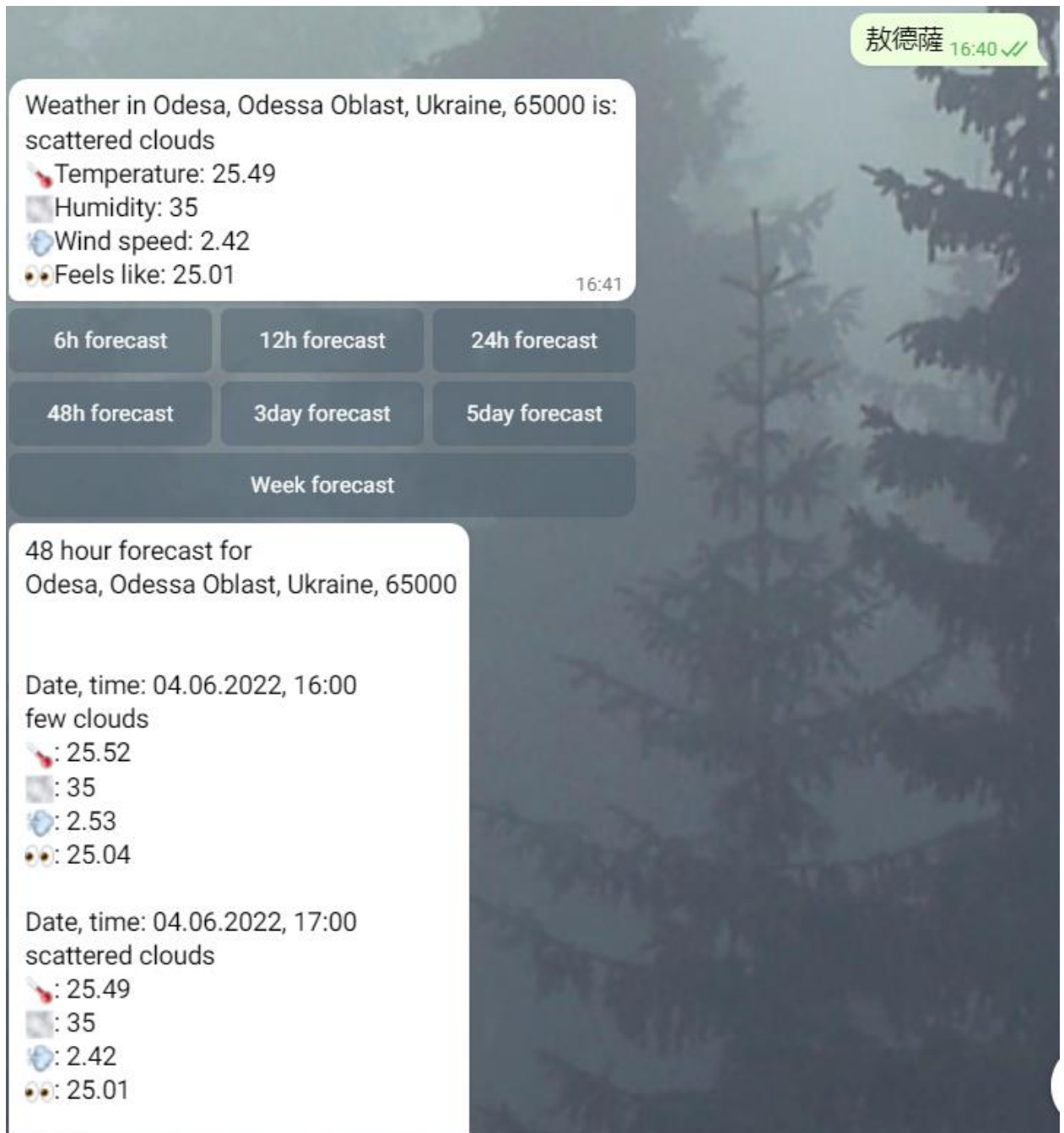


Рисунок 14 – Скріншот з отриманням даних про погоду на запит китайською мовою та прогноз погодинно на 48 годин

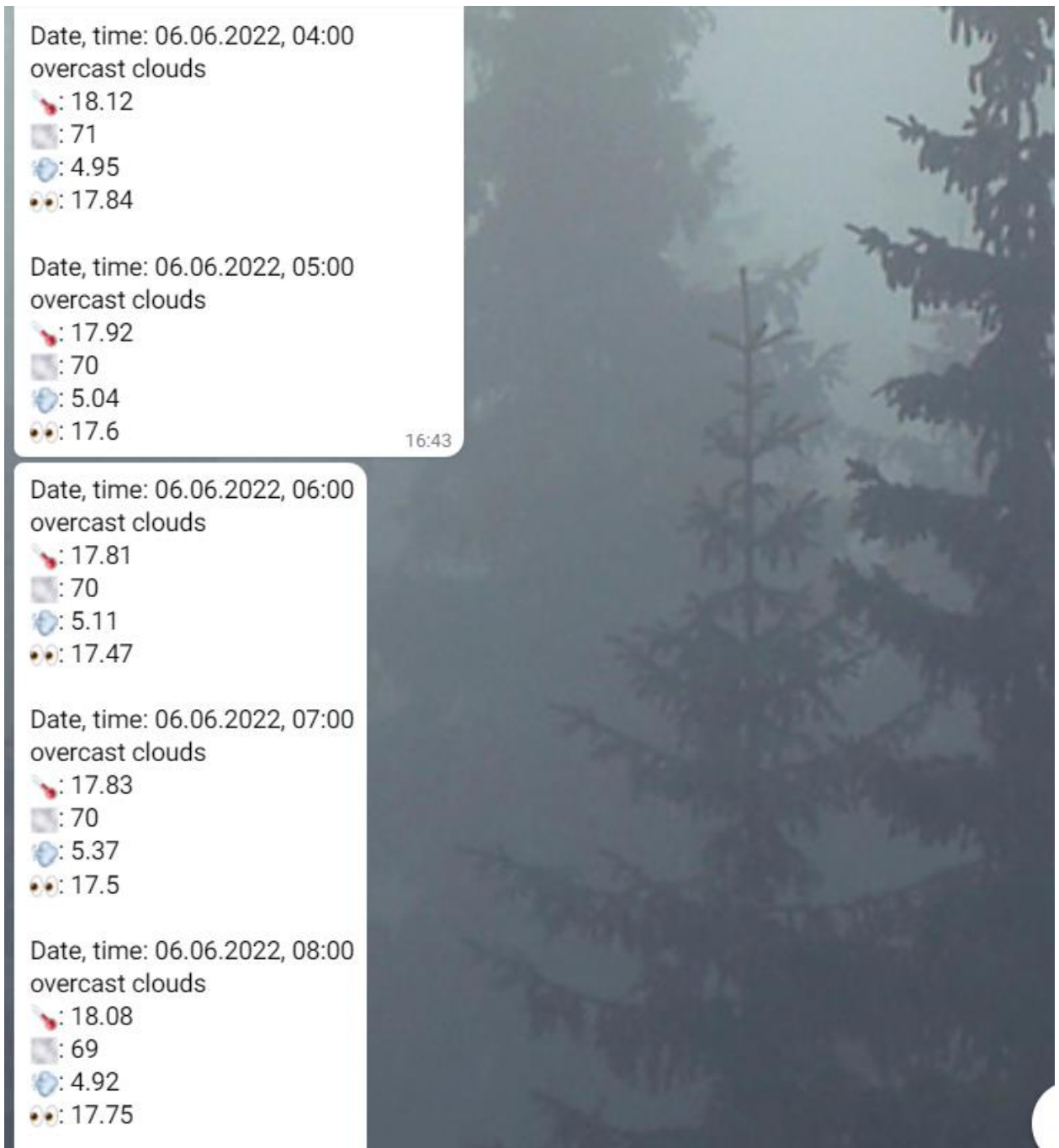


Рисунок 15 – Скріншот відповіді на запит, більшої за 3000 символів, що розбивається на декілька повідомлень

3 day forecast for
Kyiv, Ukraine, 02000

Date: 04.06.2022
Temperature between 12.76 and 24.04
scattered clouds
☁: 43
Morning:
☀: 13.31
☁: 12.63
Day:
☀: 21.54
☁: 20.87
Evening:
☀: 23.67
☁: 23.03
Night:
☀: 17.09
☁: 16.58

Date: 05.06.2022
Temperature between 14.95 and 19.67
light rain
☁: 84
Morning:
☀: 15.14
☁: 14.56
Day:
☀: 15.5
☁: 15.3
Evening:
☀: 19.67
☁: 19.1
Night:
☀: 14.97
☁: 14.59

Date: 06.06.2022
Temperature between 12.61 and 23.09
clear sky
☁: 47
Morning:
☀: 13.2
☁: 12.85
Day:
☀: 21.42
☁: 20.84
Evening:

Рисунок 16 - Скріншот прогнозу погоди на три дні

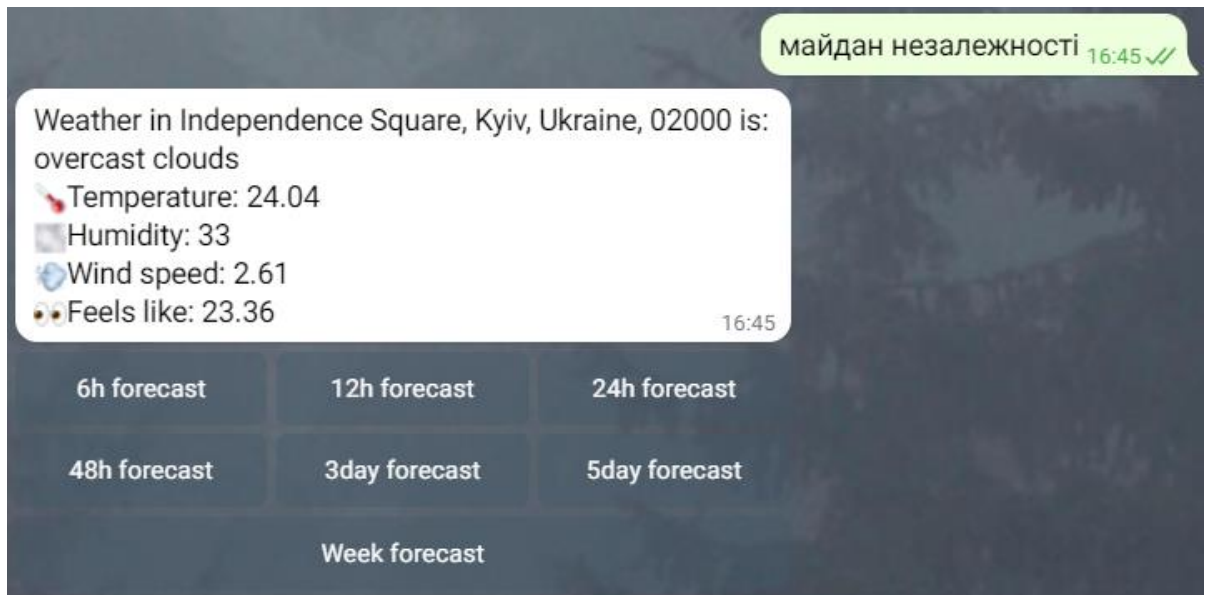


Рисунок 17 – Скріншот з отриманням даних про погоду за більш конкретною адресою

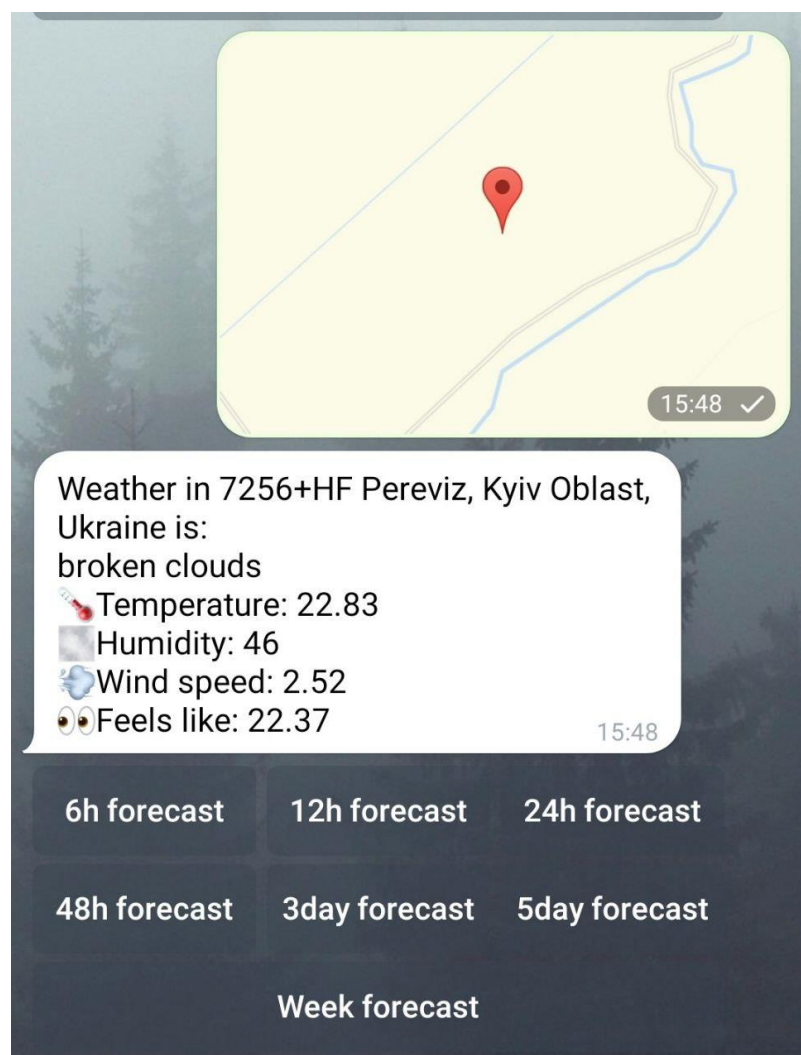


Рисунок 18 - Скріншот з отриманням даних при відправці геолокації