Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА БОТА ПОГОДНОГО ІНФОРМЕРА В TELEGRAM
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-21-1
Саранча Д. А.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Кобилін О. А.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Саранчі Даниілу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка бота погодного інформера в Telegram

затверджена наказом університету від 19 травня 2025 року № 381Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 31 травня 2025 р.
3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, отримання та обробка метеорологічних даних.

4. Перелік питань, що потрібно опрацювати в роботі
1. Огляд стану розвитку телеграм-ботів для інформування про погоду.
2. Моделювання системи вебзастосунку.
3. Вибір інструментальних засобів для реалізації поставленої задачі.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми, постановка задачі, тестові зображення, схема роботи алгоритму отримання та обробки погодних даних, схема взаємодії з базою даних MongoDB.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз технічних засобів	15.04.25-20.04.25	
5	Розробка методу	21.04.25-27.04.25	
6	Програмна реалізація	28.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Кобилін О. А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 66 с., 19 рис., 33 джерела.

ПОГОДА, ПРОГНОЗ ПОГОДИ, TELEGRAM-БОТ, МОВА ПРОГРАМУВАННЯ JAVASCRIPT, БІБЛІОТЕКА TELEGRAF, БАЗА ДАНИХ MONGODB.

Об’єктом роботи є створення телеграм-бота для надання прогнозу погоди користувачам.

Метою роботи є розробка зручного, функціонального та мультимовного бота, який дозволяє отримувати актуальну інформацію про погоду в будь-якому місті світу, переглядати термальні карти та налаштовувати персоналізовані сповіщення.

Додаткові функції, такі як голосові повідомлення з прогнозом, поради щодо вибору одягу залежно від погоди, порівняння погодних умов між містами, а також можливість отримувати щоденні сповіщення чи попередження про дощ, були реалізовані для підвищення унікальності та корисності бота. Бот орієнтований на широку аудиторію користувачів та підтримує дві мови: українську та англійську.

WEATHER, WEATHER FORECAST, TELEGRAM BOT, JAVASCRIPT PROGRAMMING LANGUAGE, TELEGRAF LIBRARY, MONGODB DATABASE.

The object of the work is to create a Telegram bot for providing weather forecasts to users.

The purpose of the work is to develop a convenient, functional, and multilingual bot that allows users to receive up-to-date weather information for any city in the world, view thermal maps, and set up personalized notifications.

Additional features, such as voice messages with forecasts, clothing advice based on the weather, comparison of weather conditions between cities, as well as the ability to receive daily notifications or rain alerts, have been implemented to ensure the uniqueness and usefulness of the bot. The bot is aimed at a wide range of users and supports three languages: Ukrainian, Russian, and English.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – Application Programming Interface (інтерфейс програмного застосунку)

BOT – Telegram Bot (програмний агент у Telegram для автоматизованої взаємодії з користувачем)

DB – Database (база даних)

GEO – геолокація (визначення місця розташування користувача)

MONGODB – документно-орієнтована система управління базами даних

NODE.JS – середовище виконання JavaScript на сервері

TELEGRAF – бібліотека для створення Telegram-ботів на Node.js

OPENWEATHER API – сервіс отримання метеорологічних даних

WIT.AI – платформа розпізнавання природної мови (NLP)

GEOAPIFY – сервіс геолокаційних даних

VOICE RSS API – сервіс конвертації тексту в голос

UNSPLASH API – сервіс отримання зображень високої якості

TTS – Text-To-Speech (перетворення тексту на мовлення)

STT – Speech-To-Text (перетворення мовлення на текст)

HTTP – Hypertext Transfer Protocol (протокол передачі гіпертексту)

JSON – JavaScript Object Notation (формат обміну даними)

CRON – система для періодичного запуску завдань за розкладом у Unix-подібних системах

ВСТУП

Робота над створенням телеграм-бота для інформування про погоду є надзвичайно актуальною у сучасному світі, де щоденне планування діяльності значною мірою залежить від погодних умов. З огляду на збільшення кількості користувачів месенджерів і попит на швидкий доступ до інформації, виникає потреба у створенні зручних, функціональних та інтуїтивно зрозумілих інструментів для отримання прогнозу погоди.

Створення Telegram-бота дозволяє користувачам легко отримувати актуальну інформацію про погодні умови у будь-якому місті світу, переглядати термальні карти, отримувати поради щодо вибору одягу, налаштовувати щоденні сповіщення або попередження про несприятливі погодні умови. Це значно спрощує підготовку до поїздок, планування активностей на свіжому повітрі чи інших заходів, які залежать від погодних умов.

Розвиток технологій штучного інтелекту, інтеграція голосових сервісів та персоналізація взаємодії з ботами робить такий проєкт особливо цінним для широкої аудиторії. Підтримка кількох мов, можливість обирати спосіб введення даних (геолокація або ручне введення міста) та автоматичне надсилання прогнозу щодня сприяють зростанню популярності такого рішення серед користувачів і підвищують їхню задоволеність від використання бота. Крім того, сучасні можливості інтеграції із зовнішніми погодними API забезпечують високу точність та актуальність отримуваних даних, а архітектура бота передбачає легке масштабування та потенційне розширення функціоналу, наприклад, додавання історичних даних про погоду, прогноз на тривалий період або інтеграцію з іншими сервісами. Саме тому створення та вдосконалення подібних рішень відповідає сучасним потребам користувачів у швидкому доступі до актуальної та персоніфікованої інформації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд стану розвитку телеграм-ботів для інформування про погоду

У сучасному світі цифрові технології відіграють дедалі важливішу роль у повсякденному житті людей. Особливої популярності набули месенджери, зокрема Telegram, який завдяки своїй відкритості, високій швидкості та багатофункціональності став платформою для створення численних ботів різного призначення. Однією з популярних категорій є боти для інформування користувачів про погодні умови, які забезпечують миттєвий доступ до актуальної інформації.

Із стрімким зростанням популярності месенджерів, Telegram став однією з найбільш технологічно відкритих платформ для розробників завдяки підтримці API, можливості створення ботів і використанню Webhook-інтеграцій. Боти Telegram нині охоплюють широкий спектр застосувань – від електронної комерції до сервісів моніторингу новин, управління особистими фінансами і, безумовно, інформування про погоду.

Особливості екосистеми Telegram, такі як зручність взаємодії, багатофункціональні клавіатури, inline-режим і підтримка команд, роблять цю платформу надзвичайно привабливою для створення погодних ботів, що здатні працювати швидко, і у зрозумілому для користувача форматі [1].

Розвиток телеграм-ботів для прогнозу погоди є частиною ширшої тенденції автоматизації надання сервісних послуг через месенджери. Такий підхід має низку переваг: користувачеві не потрібно встановлювати додаткові додатки, вся взаємодія здійснюється безпосередньо через звичний канал комунікації, що спрощує процес отримання даних. Завдяки цьому телеграм-боти стають зручним рішенням для отримання коротких оперативних повідомлень про погодні умови в будь-якій точці світу.

Станом на сьогоднішній день існує велика кількість погодних ботів у Telegram, які різняться за функціональністю, дизайном та якістю наданих даних. Серед найпоширеніших функцій можна виділити:

– миттєвий прогноз погоди за певним містом;

– автоматичне визначення місцеперебування користувача та надання локального прогнозу;

– надання щоденних або щогодинних прогнозів;

– сповіщення про різкі зміни погодних умов, штормові попередження;

– підтримка декількох мов для користувачів із різних регіонів.

Деякі боти інтегрують розширені функції, як-от графіки температурних змін, прогнози вітру, аналіз атмосферного тиску та індекс ультрафіолетового випромінювання.

Щоб отримувати актуальну інформацію про погоду, боти інтегруються з метеорологічними API. Серед найпопулярніших сервісів для отримання даних виділяють:

– OpenWeatherMap – один із найпоширеніших сервісів, що надає прогнози по містах у всьому світі, включаючи поточну температуру, опади, швидкість вітру та багато іншого. Є безкоштовна версія з обмеженням кількості запитів на день;

– WeatherAPI – платформа, що забезпечує гнучку побудову запитів із високою точністю прогнозів і можливістю отримання історичних даних;

– AccuWeather API – славиться високою точністю короткострокових прогнозів, особливо у США та Європі;

– Tomorrow.io – новітній сервіс з акцентом на прогнозування погодних аномалій і надання спеціалізованих звітів для бізнесу.

Однак попри велику кількість рішень у цій сфері, існують певні проблеми та обмеження, які залишають місце для вдосконалення:

– якість та точність даних: багато ботів використовують безкоштовні версії метеорологічних API, які можуть оновлюватися з великою затримкою

або надавати дані з низькою точністю. Для підвищення якості сервісу важливо обирати надійні джерела, наприклад, OpenWeatherMap у преміум-доступі або WeatherAPI з розширеною підтримкою регіональних даних;

– недосконаліть інтерфейсу взаємодії: деякі боти перевантажують користувача списком команд або складними текстовими повідомленнями. Це особливо ускладнює взаємодію для новачків або користувачів, які очікують максимально простого доступу до прогнозу у кілька кліків;

– відсутність персоналізації: більшість існуючих погодних ботів надають однакову інформацію для всіх користувачів, без можливості налаштувати частоту сповіщень, вибрати улюблені міста або встановити порогові умови для тривожних повідомлень (наприклад, повідомляти тільки у разі очікуваного дощу);

– низька продуктивність: через неефективне кешування запитів або обробку API-відповідей деякі боти мають значні затримки у відповіді на запити, що негативно впливає на користувацький досвід, особливо в години пікового навантаження;

– обмежені можливості локалізації: частина ботів не підтримує мультимовність або не відображає погодні дані з урахуванням місцевих одиниць вимірювання (наприклад, температура в градусах Цельсія для Європи або Фаренгейтах для США);

– відсутність графічного подання даних: у сучасних реаліях важливим є не лише текстовий прогноз, а й візуалізація даних – наприклад, погодні карти, графіки змін температури, таблиці опадів. Більшість ботів обмежуються коротким текстовим повідомленням, що не завжди зручно для глибшого аналізу;

– безпекові аспекти: нехтування правилами обробки даних користувачів (наприклад, місцезнаходження або облікові дані) також є проблемою. Безпечна робота з персональними даними має стати пріоритетом під час створення нових рішень.

Такі боти також можуть бути інтегровані з функцією автоматичних повідомлень, що дозволяє відразу отримувати оновлення про зміни погодних умов. Інші додаткові функції включають можливість отримувати детальніші дані, наприклад, про швидкість вітру, атмосферний тиск, рівень вологості і так далі. Деякі боти можуть навіть надавати користувачам поради щодо того, як краще одягатися залежно від прогнозу погоди [2].

Крім загальних технічних викликів, існують стратегічні завдання, які має вирішувати новий погодний телеграм-бот:

– створити зручну для користувача логіку роботи із мінімальною кількістю необхідних дій для отримання прогнозу;

– забезпечити підтримку різних сценаріїв використання: разовий запит погоди, автоматичні щоденні сповіщення, погодні попередження тощо;

– оптимізувати роботу з метеорологічними API для забезпечення швидкості та точності відповідей;

– побудувати архітектуру бота так, щоб легко масштабувати його функціонал у майбутньому: наприклад, додавання нових джерел погодних даних, підтримку нових мов або розширення можливостей персоналізації.

Приклади існуючих рішень на ринку:

– WeatherMan Bot – один із популярних телеграм-ботів, що надає базову інформацію про погоду за містом. Проте він має обмежені налаштування і не підтримує багатомовність;

– YoWeather Bot – пропонує розширені налаштування сповіщень, але його взаємодія побудована через складні текстові команди, що не завжди зручно для нових користувачів;

– WeatherBot – інтегрує графічні ілюстрації погоди, але потребує значних покращень у швидкості обробки запитів.

Аналізуючи ці рішення, стає очевидним, що існує попит на інтуїтивно зрозумілий, швидкий та персоналізований телеграм-бот, який може запропонувати користувачам простий, але гнучкий інструмент для отримання інформації про погодні умови.

Як результат можна зрозуміти, що основним завданням є створення бота, який буде зручний у використанні, швидко надаватиме точну інформацію, а також забезпечить користувачеві комфорт і простоту взаємодії. Ключові аспекти включають наявність інтуїтивно зрозумілого інтерфейсу, стабільність роботи та можливість налаштувати інтерфейс під індивідуальні потреби користувачів [3].

1.2 Аналіз серверних технологій для створення телеграм-бота інформера погоди

У створенні телеграм-бота для інформування про погоду важливими є не тільки вибір платформи для бота, але й серверна частина, яка забезпечує швидку обробку запитів, отримання актуальних даних з погоди та їх надання користувачам. Для реалізації таких проектів активно використовуються різноманітні серверні технології, зокрема Python, Node.js та Java, які пропонують безліч переваг для вирішення завдань різної складності. Оскільки кожна з цих мов має свої унікальні особливості, розглянемо їх докладніше для вибору найбільш підходящої технології для створення погодного бота.

Python є одним із найбільш популярних варіантів для розробки серверної частини телеграм-ботів. Ця мова славиться своєю простотою, що дозволяє швидко почати розробку та впровадження нових функцій, а також наявністю великої кількості бібліотек і фреймворків для роботи з API. Вибір Python для створення бота пояснюється також його підтримкою асинхронного програмування, що дозволяє ефективно обробляти одночасно кілька запитів від користувачів. Популярні бібліотеки, такі як python-telegram-bot і telepot, надають зручний інтерфейс для взаємодії з Telegram API, а фреймворки на кшталт Flask або Django дозволяють швидко налаштувати серверну частину додатку.

Основною перевагою Python є наявність величезної спільноти розробників, що гарантує безліч готових рішень і активну підтримку. Завдяки простоті синтаксису Python є ідеальним варіантом для розробників, які шукають ефективний інструмент для швидкої реалізації проектів. Для роботи з зовнішніми API, такими як OpenWeatherMap або WeatherAPI, Python пропонує зручні методи інтеграції, що дозволяють легко отримувати дані про поточну погоду та здійснювати необхідні перетворення перед відправкою їх користувачам. Однак варто зазначити, що Python має деякі обмеження при обробці високих навантажень через глобальну блокування інтерпретатора (GIL), що може впливати на ефективність в разі обробки великої кількості одночасних запитів.

Node.js, у свою чергу, є потужним середовищем для розробки серверних застосунків, заснованих на JavaScript. Особливістю Node.js є його асинхронність, що дає змогу ефективно обробляти велику кількість одночасних запитів при мінімальних затримках. Цей аспект є важливим для телеграм-ботів, оскільки вони часто повинні обробляти запити від багатьох користувачів одночасно. Node.js також має власний пакетний менеджер – NPM, що забезпечує зручне управління залежностями та дозволяє використовувати величезну кількість модулів і бібліотек для різних задач. Бібліотеки, як-от node-telegram-bot-api, значно спрощують процес інтеграції з Telegram API, а додаткові модулі дозволяють з легкістю налаштувати обробку запитів до погодних API [4].

Node.js також дуже популярний завдяки своїй здатності працювати на всіх платформах, що дозволяє створювати кросплатформенні сервери, що є важливим для забезпечення стабільної роботи бота на різних пристроях. Завдяки вбудованій асинхронній обробці запитів Node.js дуже ефективно працює з зовнішніми API, дозволяючи боту швидко отримувати погодні дані і відправляти їх користувачам без значних затримок. Однак, попри численні переваги, Node.js може бути менш зручним для великих проектів, де

потрібно організувати складну структуру коду або обробляти дуже великі масиви даних.

Java є ще одним потужним інструментом для розробки серверної частини телеграм-бота. Вона пропонує високу продуктивність і здатність до масштабування, що робить її підходящою для складних та великих проектів. Однією з головних переваг Java є її здатність працювати на різних операційних системах без необхідності змінювати код завдяки використанню JVM (Java Virtual Machine). Це дозволяє зберігати сумісність і забезпечує стабільну роботу бота на різних платформах. Для створення серверних додатків на Java використовуються популярні фреймворки, такі як Spring Boot, які надають готові рішення для швидкої розробки і забезпечують високу ефективність роботи навіть при великих навантаженнях.

Java також надає потужні засоби для обробки даних і взаємодії з різними API, що є важливим аспектом для погодних інформерів. Завдяки вбудованій підтримці багатозадачності, Java може ефективно обробляти численні запити, що надходять одночасно від користувачів, і забезпечувати стійкість системи навіть при високому навантаженні. Проте одним із недоліків є більша складність у розробці і налаштуванні середовища порівняно з Python та Node.js. Також Java вимагає більше ресурсів для запуску порівняно з іншими мовами, що може бути недоліком для невеликих проектів [5].

У порівнянні з іншими мовами програмування, Node.js виявляється найбільш підходящим вибором для створення телеграм-бота інформера погоди. Це пояснюється його високою продуктивністю, зручністю роботи з мережею і можливістю легко масштабувати серверну частину додатку для роботи з великою кількістю одночасних запитів. Node.js також дозволяє швидко обробляти запити до зовнішніх API і забезпечує мінімальні затримки у відповіді користувачам, що є важливим для успішного функціонування бота. Тож для створення телеграм-бота інформера погоди найкращим

вибором є Node.js, оскільки він поєднує швидкість, ефективність і зручність у роботі з великими обсягами запитів.

1.3 Аналіз технологій для розробки клієнтської частини телеграм-бота інформера погоди

Для створення Telegram-бота, який надає актуальну інформацію про погоду, важливо правильно підібрати технології для клієнтської частини, що взаємодіє з користувачем. Хоча Telegram боти зазвичай мають обмежений графічний інтерфейс, інколи є необхідність створити інтерактивні веб-сторінки для більш зручної взаємодії з користувачами через браузер або мобільний додаток. Технології для розробки клієнтської частини бота включають фреймворки та бібліотеки, такі як React, Vue.js і Angular, які використовуються для створення адаптивних та динамічних інтерфейсів. Кожен з цих інструментів має свої особливості та переваги, які важливо враховувати при виборі технології для конкретного проєкту.

React.js – одна з найбільш популярних бібліотек для створення веб-інтерфейсів, розроблена компанією Facebook. Спочатку призначений для створення інтерактивних елементів на вебсторінках, React дозволяє реалізовувати складні інтерфейси з мінімальними витратами ресурсів завдяки віртуальному DOM, що забезпечує високу продуктивність. Одна з головних переваг React – компонентний підхід до розробки, коли всі частини інтерфейсу поділяються на самостійні компоненти, що легко тестуються та повторно використовуються. Це дозволяє значно спростити масштабування проекту. У випадку з Telegram-ботом, якщо потрібно створити супутній веб-інтерфейс для перегляду погоди або налаштувань, React буде ідеальним вибором завдяки своїй гнучкості і можливостям інтеграції з іншими бібліотеками та інструментами, такими як Redux для управління станом [6].

Vue.js є ще однією популярною технологією для створення клієнтських інтерфейсів. Він набирає популярності завдяки своїй простоті, легкості у вивченні та впровадженні. Якщо у вас є досвід роботи з іншими мовами програмування, Vue.js дозволяє швидко налаштувати робочий процес. Синтаксис Vue дуже чистий і зрозумілий, що полегшує створення та підтримку програм. Як і в React, у Vue застосовується компонентний підхід, що дозволяє структурувати додаток у вигляді окремих частин, що спрощує підтримку і модифікацію. Також Vue має потужну систему зв’язків між даними та представленням, що дозволяє створювати високодинамічні інтерфейси з простими інтеракціями. Якщо для Telegram-бота потрібен простий та ефективний інтерфейс для перегляду погоди або налаштувань, Vue.js може стати ідеальним вибором завдяки своїй легкості у впровадженні та підтримці.

Angular, розроблений компанією Google, є потужним фреймворком для створення веб-застосунків, орієнтованих на складні та масштабовані рішення. Angular використовує TypeScript, що забезпечує строгий контроль за типами даних та дозволяє виявляти помилки на етапі компіляції, що особливо важливо в складних проектах. Однією з основних переваг Angular є його обширний набір вбудованих інструментів для роботи з формами, валідацією, маршрутизацією, HTTP-запитами та іншими необхідними функціональними можливостями для створення повноцінних веб-додатків. Angular може бути корисним для створення більш складних інтерфейсів, де необхідно обробляти великі обсяги даних та інтеграцію з різними сервісами та API. Оскільки для бота інформера погоди може знадобитися інтеграція з кількома зовнішніми джерелами даних, Angular надає всі необхідні інструменти для розробки таких рішень, але потребує певної кваліфікації для ефективного використання [7].

Кожна з цих технологій має свої переваги, і вибір залежить від конкретних вимог проєкту. Для простих інтерфейсів, де важлива швидкість впровадження та легкість використання, Vue.js буде чудовим вибором

завдяки своїй простоті та зрозумілості. Якщо ж необхідно створити більш складні інтерфейси з високими вимогами до продуктивності та масштабованості, React буде оптимальним варіантом. Angular підходить для великих проектів з комплексними вимогами до функціональності та інтеграції з різними джерелами даних [8-13].

1.4 Постановка задачі

Головною завданням є створення зручного та функціонального Telegram-бота для надання користувачам актуальної інформації про погоду. Бот повинен надавати точні дані про температуру, вологість, швидкість вітру та прогнози на кілька днів уперед. Користувачі повинні мати можливість отримувати інформацію як для свого місця перебування, так і для інших локацій. Важливо, щоб взаємодія з ботом була інтуїтивно зрозумілою і швидкою, навіть для тих, хто не має досвіду у використанні ботів.

Об’єктом роботи є створення телеграм-бота для надання прогнозу погоди користувачам.

Метою роботи є розробка зручного, функціонального та мультимовного бота, який дозволяє отримувати актуальну інформацію про погоду в будь-якому місті світу, переглядати термальні карти та налаштовувати персоналізовані сповіщення.

Основним завданням є створення зручного інструменту для отримання даних про погоду, який дозволяє користувачам за допомогою декількох кліків дізнаватися про погодні умови в їхньому регіоні або в будь-якому іншому місті світу. Бот повинен бути інтегрований з актуальними API, що дозволяє отримати точні і оновлені погодні дані. Важливою складовою є забезпечення високого рівня безпеки, зокрема використання зашифрованих каналів зв’язку та надійних методів автентифікації.

Крім того, необхідно розробити зручний інтерфейс, який буде адаптований під різні пристрої, і дозволить користувачам швидко отримувати необхідну інформацію, не витрачаючи часу на пошук.

Для досягнення поставленої мети потрібно вирішити наступні завдання:

– провести аналіз існуючих Telegram-ботів, що надають інформацію про погоду, для визначення їхніх сильних та слабких сторін, а також виявлення незадоволених потреб користувачів;

– розробити технічне завдання для інтеграції з погодними API, що гарантують точність і актуальність даних, забезпечуючи надійність отриманої інформації;

– створити простий і зрозумілий інтерфейс, що дозволяє легко взаємодіяти з ботом і знаходити потрібну інформацію, незалежно від рівня технічної підготовки користувача;

– забезпечити високий рівень безпеки даних, використовуючи шифрування і методи захисту від несанкціонованого доступу, щоб гарантувати конфіденційність інформації користувачів;

– реалізувати систему автоматичних оновлень, щоб користувачі завжди отримували найактуальнішу інформацію про погоду, відображаючи останні зміни в реальному часі;

– налаштувати систему сповіщень про зміни погодних умов у реальному часі для покращення взаємодії з ботом.

Цей Telegram-бот стане зручним і ефективним інструментом для отримання точної інформації про погоду в будь-якій точці світу. Користувачі зможуть швидко і без труднощів дізнаватися актуальні прогнози, що зробить процес взаємодії з погодою максимально простим і доступним. Завдяки інтуїтивно зрозумілому інтерфейсу та можливості вибору способу введення даних, від геолокації до ручного вказування міста, бот забезпечить високий рівень персоналізації.

2 МОДЕЛЮВАННЯ СИСТЕМИ ВЕБЗАСТОСУНКУ

2.1 Специфікація вимог до застосунку

Telegram-бот для інформування про погоду – це програмний застосунок, який функціонує всередині месенджера Telegram та надає користувачам можливість швидко отримувати актуальну інформацію про погодні умови. Основною ідеєю є створення простого у використанні, зручного та надійного інструменту, що дозволить користувачам у будь-який момент дізнаватися прогноз погоди в будь-якому регіоні світу.

Бот підтримує лише одного типу користувача – звичайного, без розподілу на ролі або рівні доступу. Будь-який користувач має однакові функціональні можливості та може без обмежень:

– здійснювати пошук погоди за назвою міста або на основі поточної геолокації;

– переглядати поточні погодні умови, включно з температурою повітря, відчутною температурою, вологістю, атмосферним тиском, швидкістю та напрямком вітру;

– отримувати прогноз погоди на декілька наступних днів (до 3 або 7 днів залежно від налаштувань сервісу);

– підписуватися на автоматичні щоденні сповіщення про прогноз погоди в обраному місті або регіоні;

– отримувати миттєві попередження про екстремальні погодні явища (бурі, сильний вітер, грози, зливи тощо);

– переглядати історію своїх запитів щодо погоди.

Система працює за рахунок інтеграції з погодними API, такими як OpenWeatherMap або WeatherAPI, що дозволяють отримувати точні та оперативні дані з метеорологічних сервісів. Завдяки використанню цих

сервісів бот може оновлювати інформацію практично в реальному часі, що забезпечує актуальність даних для кінцевого користувача.

Комунікація між користувачем і ботом здійснюється за допомогою текстових команд, вбудованих кнопок та меню, що забезпечує інтуїтивно зрозумілу навігацію та легкість у користуванні навіть для людей без спеціальних технічних знань [14].

Особлива увага приділяється швидкості обробки запитів: бот оптимізований таким чином, щоб забезпечити мінімальну затримку у відповідях. Надійність роботи також гарантується за рахунок обробки можливих помилок (наприклад, неправильний запит міста або відсутність даних у сервісі) із наданням користувачеві відповідних повідомлень.

Для збереження персоналізованих налаштувань користувачів застосовується базова система обліку, яка дозволяє пам’ятати улюблені міста, налаштування підписок на сповіщення та останні запити. При цьому бот не вимагає обов’язкової реєстрації чи передачі особистих даних, що забезпечує приватність і безпеку взаємодії.

2.2 Архітектура системи

Архітектура Telegram-бота для інформування користувачів про погодні умови побудована за клієнт–серверною моделлю із використанням зовнішніх API-сервісів для збору, обробки та надання інформації у зручному вигляді. В основі лежить застосування технологій Node.js та бібліотеки Telegraf, яка забезпечує обробку всіх запитів і взаємодію користувача з Telegram API. Уся логіка роботи бота зосереджена на серверній частині, тоді як клієнтська сторона представлена Telegram-додатком користувача, через який здійснюється взаємодія у вигляді повідомлень і команд.

Взаємодія починається з того, що користувач ініціює бот через Telegram, після чого відправляються відповідні запити на сервер. Сервер

аналізує вхідні дані та у відповідь надає інтерактивні меню з кнопками для вибору мови та подальших функцій. Вся комунікація базується на простих повідомленнях, які містять як текстову, так і мультимедійну інформацію – голосові повідомлення, зображення термальних карт, посилання чи сповіщення.

Основною функціональною частиною архітектури є обробка запитів на прогноз погоди. При виборі користувачем відповідної функції, сервер формує запит до зовнішнього API OpenWeatherMap для отримання прогнозу. Якщо користувач ділиться своєю геопозицією, то бот визначає місто через сервіс Geoapify, і вже на основі координат виконує запит до погодного API. Після отримання прогнозу, текстова інформація обробляється і відправляється користувачу у вигляді повідомлення. Додатково, для покращення взаємодії, з цього ж тексту за допомогою сервісів VoiceRSS або Google Cloud Text-to-Speech генерується голосове повідомлення, яке відразу надсилається користувачу разом із текстовою версією прогнозу.

Ще одним важливим елементом є реалізація системи сповіщень. Завдяки використанню модуля node-cron на сервері налаштовані регулярні завдання, які в певний час автоматично відправляють користувачам актуальний прогноз погоди або сповіщення про зміну погодних умов, таких як початок дощу. Це дозволяє забезпечити користувачів актуальною інформацією навіть без їхньої активної взаємодії з ботом.

Вебсервіс також зберігає користувацькі налаштування, такі як вибрана мова, налаштування повідомлень та історію запитів у базі даних MongoDB. Використання ORM-бібліотеки Mongoose забезпечує гнучку та ефективну роботу з базою даних, дозволяючи швидко зберігати, змінювати або отримувати потрібну інформацію [15].

Бот також підтримує функцію відображення термальних карт міст. Коли користувач обирає цю функцію, бот визначає місто або через введення назви, або через отриману геопозицію, після чого за допомогою API Unsplash надсилає відповідне зображення термальної карти разом із погодною

інформацією на поточний момент. Це дозволяє користувачу не лише читати про погодні умови, але й візуально сприймати ситуацію.

Ще однією цікавою можливістю бота є функція порівняння погоди у двох різних містах. Користувач вводить назви двох міст, а бот отримує дані про поточний стан погоди в кожному з них. Після цього бот аналізує отриману інформацію і надсилає результат порівняння, відображаючи різницю у температурі, вологості, швидкості вітру та інших важливих параметрах.

Для кращого розуміння роботи Telegram-бота було розроблено схему, що ілюструє етапи взаємодії користувача із системою. Спочатку користувач через інтерфейс Telegram надсилає повідомлення боту. Далі повідомлення обробляється сервером, де воно аналізується для визначення необхідної дії. Після аналізу запиту бот виконує відповідні операції – надсилає запити до зовнішніх API для отримання прогнозу погоди, взаємодіє з базою даних MongoDB для збереження або оновлення інформації, а також генерує необхідні відповіді у текстовому або голосовому форматі. Вся взаємодія відбувається швидко та автоматизовано, що забезпечує максимально комфортний досвід для користувача [16-19].

Вся архітектура Telegram-бота розроблена з акцентом на надійність, масштабованість та швидкість реагування. Завдяки використанню Node.js як серверної платформи забезпечується обробка великої кількості одночасних запитів із мінімальною затримкою. Модульна структура коду дозволяє легко розширювати можливості бота, додаючи нові функції або інтеграції з іншими сервісами у майбутньому. Цей підхід гарантує стабільну роботу системи навіть при зростанні навантаження, а також спрощує підтримку та подальший розвиток, дозволяючи оперативно адаптувати функціонал до змінних потреб користувачів та технологічного прогресу. У результаті, забезпечується безперервне функціонування та еволюція сервісу відповідно до вимог динамічного цифрового середовища.

2.3 Початок роботи з Telegram-ботами

Telegram-боти є окремими користувачами месенджера Telegram, що взаємодіють з іншими користувачами за допомогою текстових команд, кнопок, повідомлень або медіафайлів. Вони керуються серверними застосунками, які працюють за певними алгоритмами обробки повідомлень, і можуть використовувати додаткові технології, такі як штучний інтелект, генератори мовлення або сервіси для обробки геоданих.

2.3.1 Створення та налаштування Telegram-бота

Щоб почати роботу над власним Telegram-ботом, необхідно скористатися офіційним сервісом створення ботів під назвою BotFather. Це спеціалізований бот, через якого реєструються нові боти, отримуються унікальні API токени для доступу до Telegram Bot API, а також налаштовуються базові параметри, такі як ім’я, опис та команда старту.

Після створення нового бота через BotFather користувач отримує API токен, який необхідно використовувати у серверній частині програми для встановлення з’єднання з Telegram сервісом. Цей токен має бути конфіденційним, оскільки він дозволяє повністю керувати ботом.

Розробка функціоналу Telegram-бота ведеться на серверній стороні, зазвичай із використанням мови програмування JavaScript або TypeScript у поєднанні із середовищем виконання Node.js. Для полегшення взаємодії з Telegram Bot API застосовуються спеціалізовані бібліотеки, такі як Telegraf або Grammy. Вони дозволяють легко обробляти вхідні повідомлення, створювати інтерактивні меню, працювати з медіа та налаштовувати системи сповіщень. Це значно прискорює процес розробки та забезпечує високу ефективність взаємодії бота з платформою Telegram.

2.3.2 Взаємодія бота з користувачем

Telegram-боти підтримують різноманітні методи взаємодії: текстові повідомлення, команди, кнопки, геолокаційні запити, голосові повідомлення. Для покращення користувацького досвіду застосовуються різні типи клавіатур:

– Inline-клавіатури, що з’являються прямо під повідомленням і дозволяють здійснювати вибір без необхідності друку тексту;

– Reply-клавіатури, які заміняють звичайне поле введення тексту на набір попередньо визначених опцій для вибору.

Вибір типу клавіатури залежить від сценарію використання: для швидких дій переважають Inline-клавіатури, для більш складних сценаріїв або багатокрокових процесів Reply-клавіатури.

Бот аналізує отримане повідомлення від користувача за допомогою внутрішніх механізмів обробки контексту та приймає рішення, яку відповідь сформувати. Може відбуватися як проста відповідь текстом, так і складніші процеси, наприклад, запити на зовнішні сервіси для отримання актуальної інформації.

2.3.3 Архітектура обробки запитів

Після натискання кнопки або відправки команди користувачем, повідомлення миттєво надходить на сервер, де встановлений застосунок бота. Серверна частина обробляє повідомлення наступним чином:

– прийняття повідомлення через вебхук або довге опитування (long polling) для отримання актуальних апдейтів від Telegram;

– обробка повідомлення відповідно до заданої логіки обробки команд або тексту;

– звернення до зовнішніх сервісів або бази даних – наприклад, отримання прогнозу погоди через API OpenWeatherMap, визначення місцеположення через Geoapify, розпізнавання голосових повідомлень через WIT.AI або Google Speech-to-Text;

– формування відповіді – текстової, голосової, графічної або у вигляді кнопок;

– відправлення відповіді користувачеві назад через Telegram API.

Сервер також може зберігати інформацію про користувача у базі даних MongoDB для налаштування персоналізованих сповіщень або ведення історії запитів. Робота із базою даних відбувається через об’єктно-документний маппер Mongoose, що дозволяє ефективно оперувати великими обсягами даних.

2.4 Загальна схема роботи бота

Для кращого розуміння процесу взаємодії користувача із Telegram-ботом, наведено схему архітектури роботи системи (рис. 2.1). На ній представлено усі основні етапи обробки запиту від першого повідомлення користувача до формування відповіді:

– користувач надсилає команду або повідомлення через Telegram;

– повідомлення передається серверу через API;

– сервер обробляє повідомлення, визначає його тип і зміст;

– у разі потреби сервер взаємодіє із зовнішніми API для отримання актуальної інформації або звертається до власної бази даних;

– сформована відповідь надсилається користувачу у відповідному форматі.

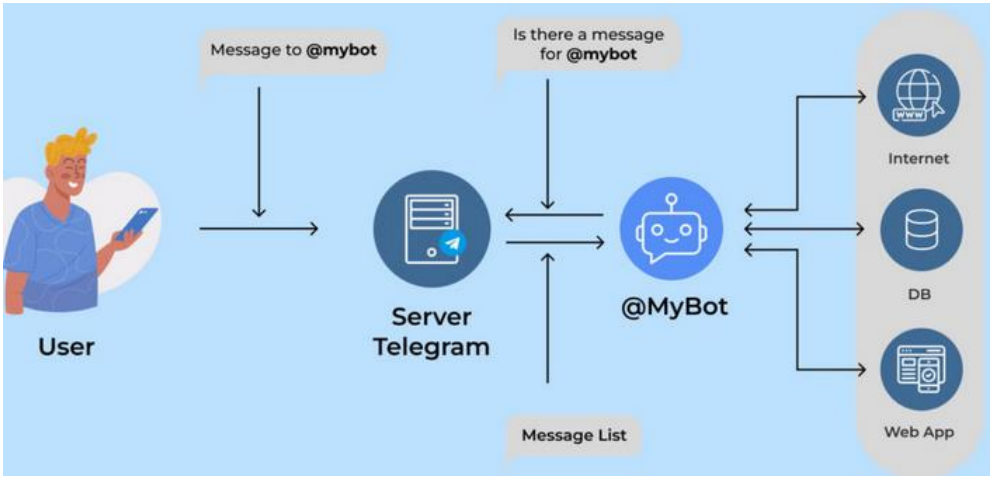

Рисунок 2.1 – Загальна схема роботи Telegram-бота

Початковим етапом моделювання є побудова діаграми прецедентів, яка ілюструє взаємодію користувача з основними можливостями системи (рис. 2.2).

На діаграмі прецедентів зображено базові дії, які може виконати користувач у межах Telegram-бота:

– вибір мови інтерфейсу;

– отримання прогнозу погоди;

– підписка на сповіщення;

– порівняння температур у різних містах.

Це дозволяє чітко визначити функціональні вимоги до системи та забезпечити зручність взаємодії користувача з ботом.

Рисунок 2.2 – Діаграма прецедентів Telegram-бота погоди

Наступним етапом є побудова діаграми класів (рис. 2.3), яка демонструє структуру системи та взаємозв’язки між її основними компонентами. Центральним елементом є клас WeatherBot, що відповідає за обробку команд користувача. Він взаємодіє з такими класами, як:

– ForecastCommand – відповідає за надання прогнозу погоди;

– ThermalMapCommand – реалізує функціонал порівняння температур у двох містах;

– NotificationsCommand – забезпечує функцію сповіщень.

Архітектура реалізує патерн команд, що дозволяє легко розширювати функціональність бота без зміни основного коду.

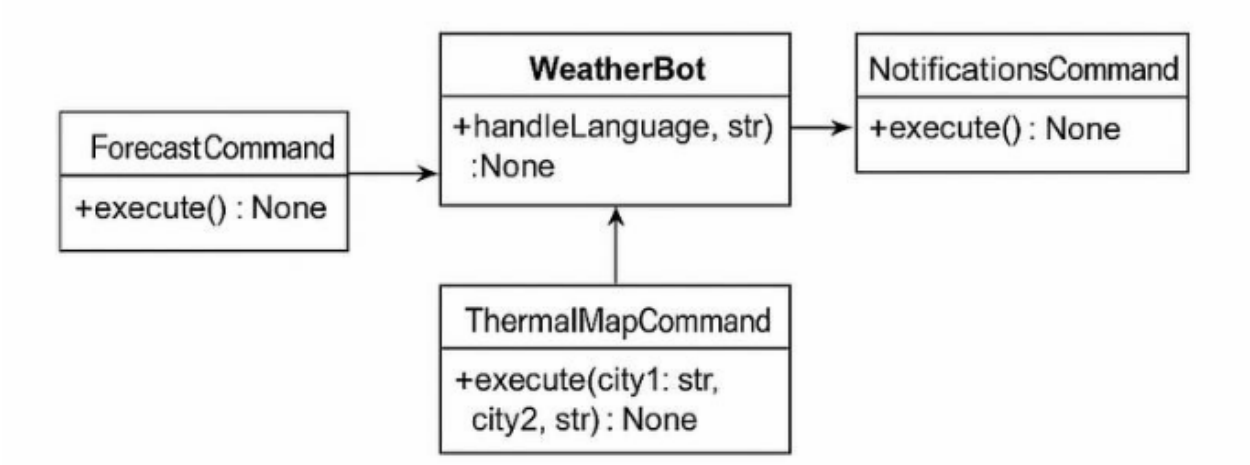

Рисунок 2.3 – Діаграма класів Telegram-бота погоди

Останнім етапом моделювання є побудова діаграми послідовності (рис. 2.4), яка відображає динаміку обміну повідомленнями між користувачем і Telegram-ботом. На діаграмі показано типовий сценарій використання:

– бот надсилає вітальне повідомлення;

– користувач обирає мову;

– після вибору мови бот активується;

– далі користувач обирає функцію;

– бот обробляє запит і виконує відповідну команду.

Ця діаграма дозволяє краще зрозуміти порядок виконання дій та їхню логіку з точки зору користувача.

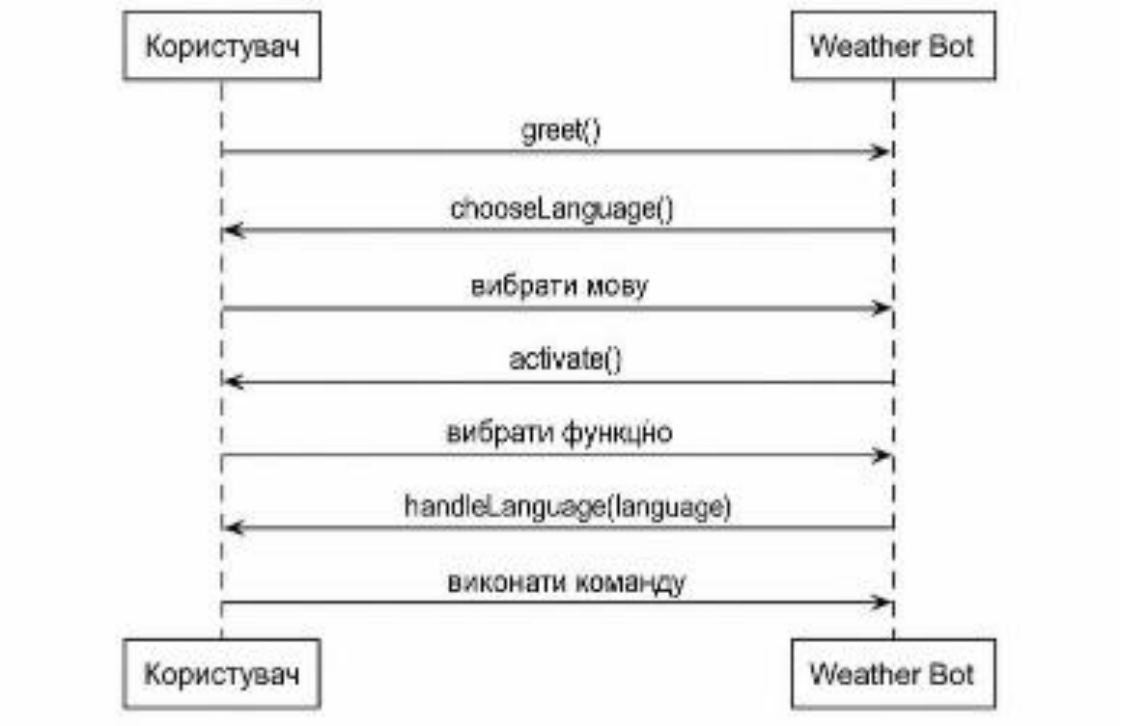

Рисунок 2.4 – Діаграма послідовності для Telegram-бота погоди

2.5 Клієнтська частина боту

Клієнтська частина реалізована через інтерфейс месенджера Telegram, де користувач взаємодіє з ботом за допомогою текстових команд та меню. Вся взаємодія з ботом здійснюється у діалоговому форматі, що імітує поведінку звичайного користувача в чаті, при цьому підтримується багатомовність, гнучке оброблення запитів та швидкий зворотній зв’язок.

Інтерфейс Telegram забезпечує базові елементи взаємодії:

- кнопки з варіантами дій;
- Inline-меню;
- повідомлення з відповідями на запити;
- підтримка emoji та форматування тексту.

Користувач спілкується з ботом через текстові або кнопкові команди. Відповідно до обраної функції (отримання прогнозу, порівняння міст, сповіщення), бот генерує адаптовану відповідь у зручному для сприйняття форматі.

Основні елементи фронтенд-взаємодії

– вибір мови інтерфейсу, що дозволяє адаптувати бот до користувачів із різною мовною приналежністю (реалізовано через кнопку або текстову команду);

– динамічні повідомлення з прогнозом погоди, які формуються з урахуванням поточних погодних умов та включають інформацію про температуру, стан неба, вологість тощо;

– повідомлення-порівняння між двома містами, що використовують структуроване відображення температурних даних у текстовому форматі;

– автоматичні або заплановані сповіщення, які інформують користувача про зміну погодних умов.

Telegram-клієнт є своєрідним інтерфейсом, через який реалізуються всі комунікації [20]. Завдяки інтерактивним елементам Telegram (Reply-кнопки, Inline-клавіатура, повідомлення зі швидкою відповіддю) користувачеві не потрібно вводити команди вручну – достатньо натиснути потрібну кнопку.

Взаємодія реалізується безпосередньо через Telegram Bot API, який дозволяє:

– швидко формувати повідомлення відповідно до обраної мови;

– забезпечити адаптивність відповідей залежно від контексту;

– миттєво оновлювати інформацію без затримок.

Весь інтерфейс зосереджений у діалозі користувача з ботом, де кожне повідомлення виконує роль «компонента», а кожна дія – реактивну логіку взаємодії з backend-частиною.

2.6 Серверна частина бота

Серверна частина Telegram-бота реалізована з використанням середовища виконання Node.js, що дозволяє створити швидкий, масштабований та асинхронний серверний застосунок. Основна мета бекенду це обробляти запити користувачів, виконувати відповідні дії на основі вибраної команди, звертатися до зовнішніх API для отримання актуальних метеоданих та формувати відповіді у зручному для користувача форматі.

Серверна частина працює через Telegram Bot API, отримуючи повідомлення від користувача через Webhook. Після отримання повідомлення сервер виконує наступні дії:

– аналізує текст запиту користувача;

– визначає тип команди (отримання прогнозу, порівняння міст, сповіщення тощо);

– викликає відповідну функцію обробника;

– звертається до зовнішнього API (наприклад, OpenWeatherMap або WeatherAPI) для отримання актуальних даних;

– формує відповідь і надсилає її назад через Telegram API.

Кожна команда реалізує метод execute(), який отримує параметри, надсилає HTTP-запит до зовнішнього API (наприклад, за допомогою бібліотеки axios або node-fetch), обробляє відповідь і формує повідомлення для Telegram.

Крім цього, у WeatherBot реалізовано метод handleLanguage(language: string), який відповідає за зміну мови інтерфейсу на основі вибору користувача. Це забезпечує багатомовність і персоналізований підхід у взаємодії з ботом. Сервер надсилає відповідь користувачу, формуючи діалог у зрозумілій формі.

Технології, використані у backend:

– Node.js – як основне серверне середовище виконання;

– Telegraf.js – популярна бібліотека для побудови Telegram-ботів;

– Axios або node-fetch – для виконання HTTP-запитів до зовнішніх погодних API;

– dotenv – для зберігання секретних ключів та токенів.

2.7 Інтеграція з API

Одним з ключових елементів функціонування Telegram-бота є інтеграція з зовнішніми API для отримання актуальних погодних даних та побудови інформативних відповідей. У проєкті використовується публічне API (наприклад, OpenWeatherMap або WeatherAPI), що дозволяє отримувати прогнози погоди, температуру, вологість, швидкість вітру, геолокаційні координати та інші метеопараметри [21].

Інтеграція реалізована через асинхронні HTTP-запити, що дозволяє забезпечити високу швидкодію та уникає блокування потоку подій під час очікування відповіді від зовнішнього сервера. У Node.js для цього застосовуються бібліотеки axios або node-fetch, які надають зручний API для надсилання запитів та обробки JSON-відповідей (лістинг 2.1).

Лістинг 2.1 Запит до API:

```
const axios = require('axios');
async function getWeather(city) {  const url =
https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${API
_KEY}&units=metric&lang=ua`;
  try { const response = await axios.get(url);
    return response.data; } catch (error) {
    console.error('Помилка під час запиту до API:', error.message);
    return null;  }}
```

Основні етапи інтеграції:

– формування URL для запиту, що включає назву міста (або координати), одиниці вимірювання, бажану мову відповіді та API-ключ;

– надсилання GET-запиту до обраного погодного сервісу;

– обробка відповіді: парсинг JSON, вилучення потрібних полів (наприклад, main.temp, weather[0].description, wind.speed);

– формування повідомлення у форматі, зручному для користувача Telegram-бота.

Інтеграція також підтримує багатомовність: параметр lang дозволяє отримувати відповіді українською, англійською або іншими мовами, залежно від налаштувань користувача, які зберігаються у пам’яті сесії [22].

Типи використовуваних API-запитів:

– поточна погода (/weather);

– прогноз на кілька днів вперед (/forecast);

– геолокація за назвою міста (/geo/1.0/direct);

– дані про кілька міст одночасно (для режиму «теплової карти»).

У разі помилки під час звернення до API (наприклад, неправильна назва міста або відсутній ключ доступу) бот коректно обробляє винятки, інформуючи користувача про проблему у зручній формі.

API-ключ зберігається у .env-файлі та не виноситься у відкритий код (лістинг 2.2). Для завантаження змінних середовища використовується бібліотека dotenv.

Лістинг 2.2 Збереження API ключа

```
require('dotenv').config();
const API_KEY = process.env.WEATHER_API_KEY;
```

Завдяки цьому конфіденційні дані безпечно керуються під час розгортання проєкту на сервері.

2.8 Підключення до бази даних

Одним з ключових компонентів архітектури Telegram-бота є збереження даних про користувачів, історію їхніх запитів, мовні налаштування, статуси підписок та географічну інформацію. Для цього в даному проєкті використовується MongoDB – популярна документоорієнтована база даних NoSQL, яка забезпечує гнучке, масштабоване та ефективне зберігання неструктурованих або напівструктурованих даних.

Для взаємодії з MongoDB у застосунку використовується Mongoose – потужна бібліотека для Node.js, що дозволяє створювати моделі даних, керувати колекціями, валідувати поля та виконувати запити до бази у зручній обгортці. Вона абстрагує низькорівневі операції з базою даних, надаючи розробнику інструменти для легкого проєктування схеми даних, а також асинхронної роботи з колекціями за допомогою Promises та async/await [23].

Процес підключення до бази даних виконується на етапі ініціалізації бота. Для цього у файлі конфігурації .env зберігається URI до MongoDB (наприклад, від Atlas або локального сервера). Цей URI зчитується за допомогою бібліотеки dotenv, яка дозволяє зберігати чутливі дані поза кодом і не розкривати їх у публічному репозиторії (лістинг 2.3).

Лістинг 2.3 Підключення до MongoDB через Mongoose:

```
require('dotenv').config();
const mongoose = require('mongoose');
mongoose.connect(process.env.MONGO_URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true
}).then(() => {
  console.log('Підключення до MongoDB успішне');
}).catch((err) => {
```

```
  console.error('Помилка підключення до MongoDB:', err);
});
```

Цей код забезпечує стабільне з’єднання із базою, а також дозволяє обробляти можливі помилки на етапі встановлення з’єднання, що критично важливо для стабільної роботи бота у продакшн-середовищі. Надійне з'єднання є фундаментом для безперебійного функціонування, гарантуючи, що бот завжди матиме доступ до необхідних даних.

Після успішного підключення застосунок визначає схеми моделей, які відповідають за логіку зберігання і доступу до даних. Наприклад:

– модель користувача зберігає Telegram ID, обрану мову, статус підписки на сповіщення, геолокацію, історію запитів тощо;

– модель локацій дозволяє зберігати назви міст, координати, а також пов’язані з ними прогнози погоди;

– модель сповіщень містить параметри нагадувань (щоденних або у разі зміни погоди) для кожного користувача.

Використання Mongoose дозволяє додатково встановити валідацію полів, індекси для пришвидшення пошуку та забезпечення унікальності певних записів (наприклад, Telegram ID), а також реалізувати middleware для автоматичних дій перед збереженням документів (рис. 2.5).

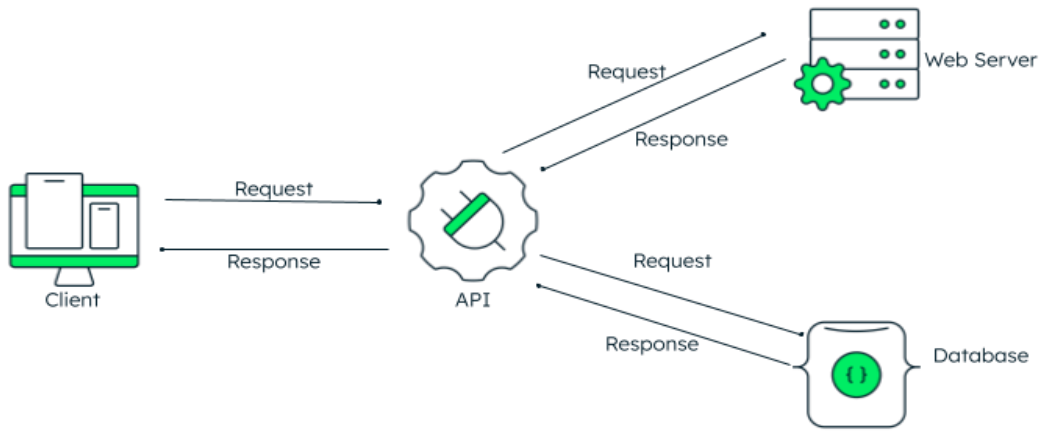

Рисунок 2.5 – Схема взаємодії з базою даних MongoDB

3 РОЗРОБКА БОТА ПОГОДНОГО ІНФОРМЕРА В TELEGRAM

3.1 Вибір інструментальних засобів для реалізації поставленої задачі

Розробка Telegram-бота, який надає прогноз погоди, потребує поєднання ряду сучасних технологій, здатних забезпечити надійність, масштабованість, інтерактивність та якісний користувацький досвід. Для реалізації поставленого завдання був обраний стек технологій, який охоплює як бекенд-розробку, так і взаємодію з зовнішніми API, обробку геоданих, синтез мовлення, розпізнавання голосу та управління станом користувачів. Завдяки правильно підібраному інструментарію, бот може ефективно реагувати на запити користувачів, забезпечуючи персоналізований досвід на кількох мовах, із залученням голосових та візуальних елементів [24-26].

Основою проекту є Node.js – високопродуктивна, подієво-орієнтована JavaScript-платформа, яка чудово підходить для побудови серверних застосунків, що працюють у режимі реального часу. Завдяки неблокуючій моделі вводу/виводу Node.js ідеально підходить для обробки великої кількості одночасних запитів від користувачів Telegram. У контексті бота це означає, що кожна взаємодія (натискання кнопки, надсилання голосового повідомлення або геопозиції) обробляється швидко та асинхронно, без затримок у роботі.

Для прямої роботи з Telegram Bot API був обраний Telegraf.js – сучасний фреймворк, що надає потужні інструменти для побудови діалогових сценаріїв, обробки повідомлень, Inline-кнопок, команд, callback’ів та middleware-функцій. Telegraf дозволяє будувати гнучку логіку обробки, у тому числі змінювати меню залежно від обраної мови або попередніх дій користувача. Він також підтримує контекстну взаємодію, що забезпечує чисту і зрозумілу структуру коду навіть при складних гілках сценаріїв.

Для зберігання інформації про користувачів, їхні мовні налаштування, підписки на сповіщення та історію запитів була використана MongoDB –

документно-орієнтована база даних, яка дозволяє зберігати структури даних у форматі BSON (JSON-подібному). Завдяки своїй гнучкості MongoDB ідеально підходить для зберігання динамічно змінюваних об’єктів, характерних для користувацьких профілів. Для взаємодії з базою даних було застосовано ORM–рішення Mongoose, яке дозволяє визначити схеми, валідацію даних та використовувати зручні методи для CRUD-операцій.

Ключовим джерелом погодних даних є OpenWeather API, який надає широкий спектр інформації – від поточного стану до детальних прогнозів на 24 години, 3 або 7 днів. API підтримує роботу як по містах, так і за географічними координатами, що дозволяє реалізувати зручний сценарій для користувача: або вводити місто вручну, або надсилати свою геопозицію. Дані, що отримуються від OpenWeather, обробляються та формуються у вигляді зрозумілого текстового звіту, що доповнюється порадами щодо одягу та температурного режиму [27].

Для зворотного геокодування (визначення міста за координатами) було обрано GEOAPIFY API, який дозволяє точно та швидко визначити назву міста та країни за широтою та довготою. Це особливо корисно, коли користувач ділиться геолокацією – бот одразу може ідентифікувати місцезнаходження та підготувати прогноз погоди.

Бот має функцію як розпізнавання, так і генерації голосових повідомлень. Для обробки вхідних аудіо користувача застосовується Google Cloud Speech-to-Text API, який забезпечує високу точність розпізнавання мови (у тому числі української, англійської) з урахуванням контексту та шумів. Це дає змогу користувачу взаємодіяти з ботом голосом – наприклад, диктувати місто або команду.

Навпаки, для перетворення згенерованих текстів у голос використовується Google Cloud Text-to-Speech, що дозволяє ботові озвучувати прогноз погоди та поради щодо одягу. У разі, якщо потрібен інший голосовий движок або формат відповіді (наприклад, MP3),

застосовується VoiceRSS API, який дозволяє швидко згенерувати мовне повідомлення.

Wit.ai API використовується як додатковий інструмент для аналізу природної мови в голосових або текстових повідомленнях користувача.

Щоб зробити бот більш візуально привабливим, використовується Unsplash API, який дозволяє завантажувати тематичні зображення відповідно до погодних умов (наприклад, сонячна погода, дощ, сніг). Такі ілюстрації надсилаються разом із текстовим прогнозом або термокартою, що покращує візуальне сприйняття відповіді бота.

Для реалізації щоденних розсилок та інших регулярних подій використовується node-cron – зручна бібліотека для налаштування періодичних задач. Наприклад, за допомогою node-cron можна реалізувати щоденне надсилання прогнозу о 7:00 ранку або повідомлення у разі очікуваного дощу протягом дня. Користувач може обрати, чи хоче він отримувати такі сповіщення, і відповідні налаштування зберігаються в базі. Це дозволяє забезпечити високий рівень персоналізації та проактивності бота, що значно покращує досвід користувача.

Розробка здійснювалась у середовищі Visual Studio Code, яке забезпечує зручну роботу з Node.js, MongoDB, REST API та системами контролю версій (Git). Завдяки великій кількості розширень, зокрема ESLint, Prettier, REST Client, MongoDB Tools тощо, процес розробки був ефективним і добре структурованим. Це дозволяло швидко перемикатися між файлами, тестувати API-запити безпосередньо в редакторі та працювати в команді з високою продуктивністю. Використання комплексного середовища розробки та сучасних інструментів забезпечило високу якість коду, легкість його налагодження та можливість швидкого внесення змін, що є критично важливим для agile-розробки.

3.2 Етапи розробки Telegram-бота

Створення Telegram-бота для надання метеорологічної інформації є складним процесом, що охоплює декілька ключових етапів: від попереднього аналізу і проєктування до безпосередньої реалізації та тестування. Кожен із цих етапів має на меті забезпечити кінцевому користувачеві просту, швидку і функціональну взаємодію з ботом, а також надійний доступ до актуальної погодної інформації з урахуванням особистих налаштувань, мови інтерфейсу та способу подачі даних.

Початковий етап передбачав детальний аналіз ринку подібних рішень у Telegram-екосистемі. Було проаналізовано функціональні можливості наявних ботів, їхні сильні сторони та недоліки. Особливу увагу було приділено таким аспектам, як точність наданих прогнозів, швидкість відповіді, підтримка голосових повідомлень і геолокації, адаптивність під потреби користувача (вибір мови, формату подачі інформації тощо). У результаті було сформульовано бачення власного програмного рішення, яке має перевершувати аналоги за зручністю, гнучкістю і рівнем персоналізації.

Після цього розпочався етап проєктування архітектури бота. Основна логіка була поділена на декілька взаємопов’язаних модулів: модуль обробки вхідних повідомлень, модуль взаємодії з API сторонніх сервісів (OpenWeather, Wit.ai, Google Cloud, Geoapify тощо), модуль генерації голосових повідомлень, блок для управління мовами інтерфейсу, а також база даних, яка відповідає за зберігання станів користувачів і їхніх індивідуальних налаштувань. Особливу увагу було приділено підтримці багатомовності – передбачено зручну систему перемикання між двома мовами (українською та англійською), що розширює потенційну аудиторію застосунку.

На етапі розробки користувацької логіки було побудовано чіткий сценарій взаємодії з ботом. При першому запуску користувач отримує привітальне повідомлення з можливістю обрати мову інтерфейсу. Далі

відкривається основне меню з доступом до чотирьох основних функцій: отримання прогнозу погоди, керування повідомленнями, перегляд теплової карти та порівняння погодних умов у двох містах. Кожен із пунктів реалізований як окрема послідовність кроків, які ведуть користувача через просту й інтуїтивно зрозумілу систему кнопок Telegram (Inline-клавіатура), що значно спрощує навігацію.

Розробка функціональної частини бота включала глибоку інтеграцію з низкою зовнішніх API. Зокрема, для отримання метеорологічної інформації було обрано сервіс OpenWeatherMap, який надає розширені прогнози на 1, 3 або 7 днів. Дані з цього API використовуються для формування текстового прогнозу та порад щодо одягу залежно від погодних умов (температури, сили вітру, вологості тощо). За потреби користувач може також отримати розширений опис рекомендацій з одягу, що ґрунтується на розпізнаванні ключових погодних факторів. Крім того, реалізовано функцію надсилання термальної карти регіону, яка генерується за координатами з використанням Geoapify API та зображення з Unsplash API, що додає візуального супроводу текстовій інформації [28].

Однією з найскладніших у реалізації, але водночас найцінніших функцій стала підтримка голосової взаємодії. Для розпізнавання голосових команд користувача інтегровано Google Cloud Speech-to-Text, а для озвучування прогнозів – Google Cloud Text-to-Speech та VoiceRSS. Бот здатен не лише зрозуміти усне повідомлення з назвою міста, але й відповісти голосом, зачитуючи сформульований прогноз. Це дозволяє використовувати бота в ситуаціях, коли набір тексту незручний або неможливий (наприклад, під час керування автомобілем).

Ще одним важливим елементом функціональності є реалізація автоматичних повідомлень. Для цього було використано бібліотеку node-cron, що дозволяє запланувати надсилання прогнозів у певний час доби (наприклад, о 7 ранку щодня) або за певною умовою (наприклад, повідомити

користувача, якщо в його місті починається дощ). Усі параметри повідомлень користувач може налаштувати через меню керування.

Для забезпечення збереження стану взаємодії та персональних налаштувань було використано базу даних MongoDB, у поєднанні з ORM-бібліотекою Mongoose. Тут зберігаються такі дані, як мова інтерфейсу, останні обрані міста, тип підписки на повідомлення, історія запитів тощо. Це дозволяє боту пам’ятати вподобання користувача навіть після перезапуску.

Завдяки модульному підходу та використанню сучасного фреймворку Telegraf.js, Telegram-бот зберігає високу масштабованість та можливість легкого розширення функцій у майбутньому. Його архітектура дозволяє безболісно додавати нові функції (наприклад, синхронізацію з іншими погодними API або додатками), підтримку додаткових мов, а також розширення інтерактивного досвіду за допомогою мультимодальних елементів – голосу, зображень, кнопок тощо [29].

3.2.1 Інтеграція з API OpenWeather для отримання прогнозу погоди

Одним із ключових функціональних елементів телеграм-бота є можливість надання актуальної та достовірної інформації про погоду. Для реалізації цієї функції було інтегровано OpenWeather API, який забезпечує доступ до широкого спектра метеорологічних даних у режимі реального часу, зокрема: поточну температуру, атмосферний тиск, вологість, швидкість та напрям вітру, опади, а також довгостроковий прогноз на кілька днів.

API OpenWeather було обрано завдяки його масштабованості, детальній документації, багатому функціоналу, а також можливості отримання даних у різних форматах (у т.ч. JSON), що значно спрощує їх обробку у середовищі Node.js. Крім того, платформа підтримує багатомовність відповідей та має високу надійність у плані доступності сервісу.

Процес інтеграції передбачав налаштування окремих модулів, які формують запити до API залежно від вибору користувача: місто, координати або мова. Наприклад, при виборі функції «Get Forecast», бот надає можливість або ввести назву населеного пункту вручну, або скористатися функціоналом надсилання геолокації. В обох випадках бот отримує необхідні дані, формує запит до відповідного ендпоінту OpenWeather API, та зберігає результат для подальшої обробки.

Особливістю реалізації стало розмежування запитів за термінами: окремо обробляються прогнози на 24 години, 3 доби та 7 днів. Це дозволяє не тільки надати користувачу детальний розклад погодних умов по годинах або днях, але й оптимізувати обсяг отримуваних даних, зменшуючи навантаження на API.

Для обробки метеорологічної інформації застосовуються внутрішні фільтри, які виділяють із загального масиву лише ключові параметри, що мають цінність для користувача. Зокрема, у боті реалізовано автоматичний генератор рекомендацій щодо вибору одягу, який працює на основі комбінації температури повітря, швидкості вітру та ймовірності опадів. Даний підхід робить сервіс не просто інформативним, а й практично корисним у повсякденному житті. Це суттєво підвищує зручність використання та додає боту функціонал персонального асистента з погоди.

Задля збереження стабільності роботи сервісу передбачена система обробки помилок: у разі недоступності API або неправильного запиту бот виводить користувачеві відповідне повідомлення з проханням перевірити введені дані або спробувати пізніше. Крім того, інтеграція з OpenWeather була реалізована із застосуванням бібліотеки axios, що дозволяє гнучко керувати HTTP-запитами та налаштовувати повторні спроби у разі збоїв. Даний механізм забезпечує високу відмовостійкість системи та мінімізує перебої у наданні сервісу.

Наслідком цього, підключення до OpenWeather API стало технічною основою для реалізації головної функції бота – формування детального, адаптованого та зручного у сприйнятті прогнозу погоди.

3.2.2 Побудова багатомовного інтерфейсу

Для забезпечення максимальної зручності у використанні бота різними категоріями користувачів було реалізовано повноцінну систему багатомовного інтерфейсу. Після першого запуску бот автоматично виводить повідомлення з вітанням та пропонує користувачеві обрати мову, якою буде здійснюватися подальша взаємодія. На вибір надаються дві мови: українська та англійська.

Архітектура реалізації багатомовності базується на словнику перекладів, де кожен ключовий елемент інтерфейсу має відповідник на всіх підтримуваних мовах. При збереженні мовного вибору в базі даних (MongoDB) із використанням бібліотеки Mongoose, бот динамічно формує всі наступні повідомлення саме тією мовою, яку вибрав користувач. Це дозволяє забезпечити персоналізований досвід і зменшити кількість логічних розгалужень у коді, що сприяє його чистоті та підтримуваності.

Крім текстових елементів, багатомовність також реалізована для голосових повідомлень, які генеруються на основі прогнозу погоди. За це відповідають зовнішні сервіси Google Cloud Text-to-Speech і VoiceRSS API, які підтримують численні мовні моделі. Залежно від вибраної мови, бот обирає відповідний голосовий профіль для синтезу, що робить спілкування ще більш природним та локалізованим.

Окрему увагу приділено обробці випадків, коли користувач не здійснив вибору мови або спробував ввести некоректні дані. У таких ситуаціях бот надсилає уточнююче повідомлення із запрошенням скористатися кнопками

вибору. Це дозволяє зберігати логіку взаємодії стабільною, навіть у випадках неочікуваної поведінки з боку користувача.

Завдяки впровадженню багатомовної підтримки, бот може бути ефективно використаний ширшою аудиторією, а також легко масштабований у майбутньому з додаванням нових мов. Це забезпечує гнучкість проєкту та дозволяє адаптувати його під різні регіональні ринки.

3.2.3 Отримання прогнозу погоди користувачем

Однією з найважливіших функцій телеграм-бота є можливість зручного та швидкого отримання метеорологічного прогнозу для будь-якої локації. Реалізація цього процесу була спроєктована так, щоб забезпечити максимально просту та інтуїтивну взаємодію з користувачем, незалежно від його технічної підготовки.

Після натискання кнопки «Get Forecast» у головному меню, бот пропонує користувачу два шляхи визначення геолокації: надання поточного місцезнаходження через вбудовану функцію Telegram або введення назви міста вручну. Обидва варіанти були реалізовані з урахуванням потреб різних категорій користувачів: з одного боку – максимальної автоматизації, з іншого – гнучкості при введенні будь-якого населеного пункту у світі.

Для обробки координат, надісланих через Telegram, використовується сервіс Geoapify, який дозволяє визначити назву міста за широтою та довготою. Це дає змогу боту точно встановити місце розташування навіть у випадках, коли користувач не знає або не хоче вводити назву населеного пункту вручну.

Після того, як місцезнаходження було успішно встановлено, бот пропонує вибір періоду прогнозу: 24 години, 3 дні або 7 днів. Кожен з цих варіантів відповідає певному ендпоінту в OpenWeather API і має свою специфіку структури даних. Наприклад, при виборі короткострокового

прогнозу на 24 години, користувач отримує погодні умови по годинах, тоді як при запиті на 7 днів – погодні тренди по днях.

Отримані дані обробляються, структуруються та виводяться у зручному для сприйняття вигляді. Бот виводить не лише «суху» інформацію про температуру, опади та вітер, а й формує рекомендацію щодо того, який одяг найкраще підійде для даних погодних умов. Ця рекомендація формується на основі внутрішньої логіки, яка враховує комбінацію кількох параметрів: температури, сили вітру, наявності опадів, а також часу доби. Таким чином, користувач отримує не просто прогноз, а прикладну пораду, яку можна негайно використати.

Особливістю цього етапу є також автоматичне генерування голосового повідомлення, яке дублює текстовий прогноз. Для цього застосовується інтеграція з Google Cloud Text-to-Speech або VoiceRSS API, де залежно від мови інтерфейсу бота, обирається відповідний голос. Це робить використання бота ще зручнішим для людей з порушеннями зору або для тих, хто вважає за краще слухати інформацію замість читання.

Після отримання основного прогнозу, бот додатково пропонує перейти до більш детальної рекомендації щодо одягу. Це реалізовано у вигляді кнопки, після натискання на яку користувач отримує розгорнуту пораду: який верхній одяг доцільно вдягнути, чи потрібна парасоля, яка взуття краще підійде тощо. Це дозволяє зробити прогноз максимально практичним і адаптованим під повсякденне життя користувача.

Завдяки такій структурі взаємодії, користувач проходить логічну послідовність дій – від визначення місця до отримання прикладної поради. Це підвищує зручність користування ботом і сприяє формуванню позитивного досвіду.

3.2.4 Увімкнення та керування сповіщеннями

Функція сповіщень у телеграм-боті має на меті забезпечити користувача актуальною інформацією про погоду без необхідності щоразу запускати бота вручну. Цей модуль був реалізований як один із ключових інструментів автоматизації взаємодії з ботом, що значно підвищує рівень зручності для користувача та робить сервіс більш персоналізованим.

Після вибору розділу «Notifications» в головному меню бот пропонує два основні варіанти сповіщень:

– щоденне надсилання прогнозу погоди вранці (о 07:00);

– миттєве сповіщення про початок дощу у вибраному місті.

Для реалізації цієї функції використовується бібліотека node-cron, яка дозволяє встановлювати та керувати регулярними завданнями. Це дає змогу формувати розклад, за яким бот автоматично надсилатиме повідомлення у заданий час без потреби зовнішніх тригерів. Коли користувач активує щоденне сповіщення, у базі даних MongoDB зберігається його Telegram ID разом із вибраною локацією, мовою інтерфейсу та типом сповіщення. Таким чином, кожному користувачеві надсилається індивідуальний прогноз відповідно до його налаштувань.

Другий тип сповіщення – «повідомлення про дощ» – реалізується шляхом періодичної перевірки погодних умов у місті, вибраному користувачем. Для цього бот щопівгодини надсилає запити до OpenWeather API. Якщо в отриманих даних вказується на ймовірність опадів або активний стан «Rain», бот одразу надсилає push-повідомлення, у якому інформує користувача про наближення дощу, рекомендуючи взяти парасолю або відповідний одяг.

Щоб зменшити навантаження на API та оптимізувати частоту запитів, бот кешує відповіді з OpenWeather на короткий період часу. Також реалізовано механізм розумного сповіщення: якщо прогноз дощу вже був

надісланий, бот не дублює його без суттєвих змін у погодних умовах. Це дозволяє уникнути спаму та забезпечити якісну взаємодію.

Користувачі можуть в будь-який момент вимкнути сповіщення, натиснувши відповідну кнопку у меню. Після цього бот видаляє інформацію про активні підписки з бази даних. Також передбачено систему підтвердження, яка зменшує ймовірність випадкового вимкнення або ввімкнення функції.

Інтерфейс взаємодії в цьому розділі був розроблений таким чином, щоб бути максимально лаконічним і зрозумілим: кнопки містять чіткі назви, а всі повідомлення супроводжуються короткими інструкціями. Крім того, сповіщення надсилаються тією ж мовою, яку обрав користувач під час першого налаштування бота, що покращує досвід використання.

Таким чином, система сповіщень перетворює бота з простої інформативної платформи на особистого помічника, який вчасно нагадує про зміни в погоді, дозволяючи користувачу краще планувати свій день.

3.2.5 Виведення термальної карти та поточного прогнозу

Однією з інноваційних функцій Telegram-бота є можливість перегляду термальної карти обраного міста у поєднанні з поточним прогнозом погоди. Ця функція дає змогу користувачеві не лише отримати текстовий опис погодних умов, а й візуально оцінити температурний фон у своєму регіоні, що значно підвищує рівень сприйняття інформації.

Після вибору пункту «Thermal Map» у головному меню бот пропонує користувачеві два варіанти введення місця розташування:

– надіслати свою геолокацію через Telegram;

– або ввести назву міста вручну.

У першому випадку координати (широта та довгота) надсилаються безпосередньо в запит, що дає змогу точно визначити місто без необхідності

ручного введення. У другому випадку використовується API від Geoapify, який трансформує текстову назву міста у координати, необхідні для подальшого використання у запитах до інших API.

Для отримання термальної карти використовується статичне зображення, згенероване на основі відповідного запиту до Geoapify або іншого візуального API, де вказуються координати місця та параметри зображення (масштаб, стиль, розміри тощо). Користувачеві надсилається карта з накладеним температурним шаром, що дозволяє оцінити, яка температура переважає в різних частинах міста або регіону. Карта завантажується як зображення, яке можна відкрити, збільшити чи зберегти.

Паралельно з цим бот надсилає поточний прогноз погоди для зазначеної локації. Для цього використовується OpenWeather API, який надає детальну інформацію про температуру, вологість, тиск, силу та напрям вітру, хмарність, а також опис погодних умов. Уся ця інформація формується в текстове повідомлення, адаптоване під мову інтерфейсу, яку раніше обрав користувач.

Для покращення взаємодії з користувачем до текстового прогнозу додається голосова версія – бот озвучує погоду за допомогою Google Text-to-Speech або VoiceRSS API. Це особливо зручно в умовах, коли користувач не має змоги читати повідомлення – наприклад, під час водіння або прогулянки.

Додатково система підтримує адаптивну генерацію рекомендацій щодо одягу. На основі даних про температуру, вологість, силу вітру та опади бот формує коротку пораду, що саме краще одягти в поточних умовах. Наприклад, якщо температура нижче 0°C та вітер сильний – користувач отримає повідомлення типу «Надіньте теплу куртку та шапку – на вулиці мороз і сильний вітер.»

Уся функціональність реалізована таким чином, щоб зберігати швидкодію – завдяки кешуванню частини запитів та повторному використанню зображень карт упродовж короткого проміжку часу. Це

дозволяє значно зменшити кількість запитів до зовнішніх API та забезпечити стабільність роботи навіть за високого навантаження.

Таким чином, функція виведення термальної карти у поєднанні з поточним прогнозом погоди надає користувачам інтуїтивно зрозумілу та візуально насичену інформацію, що значно покращує якість сервісу бота й робить його привабливим як для візуалів, так і для тих, хто надає перевагу голосовим відповідям.

3.2.6 Порівняння міст за погодними умовами

Ще однією корисною та інтерактивною функцією Telegram-бота є можливість порівняти погодні умови між двома різними містами. Такий інструмент є зручним для користувачів, які планують подорож, переїзд або просто хочуть дізнатися, де погода краща в даний момент. Це також розширює функціональність бота, дозволяючи не лише отримувати інформацію, а й аналізувати її у зручному форматі.

Після вибору пункту «Compare Cities» у головному меню бот поетапно просить користувача ввести назви двох міст. Цей процес реалізований через послідовні текстові запити, які забезпечують гнучкість і дозволяють порівнювати будь-які населені пункти світу. Після кожного введення бот проводить геокодування назви міста за допомогою Geoapify API, щоб отримати координати для подальшого формування запитів до погодного API.

Знаючи координати обох міст, бот надсилає запити до OpenWeather API з метою отримання поточних погодних умов. Для кожного з міст дані обробляються окремо, після чого бот формує зведене повідомлення, яке включає:

- температуру повітря;
- відчутну температуру;
- стан неба (ясно, хмарно, дощ тощо);

– вологість;

– швидкість вітру;

– атмосферний тиск.

Для зручності сприйняття інформації дані обох міст подаються у паралельному форматі – в одному повідомленні, де параметри йдуть поряд або зіставляються по секціях. Це дозволяє користувачеві миттєво оцінити відмінності між погодними умовами в двох локаціях.

Після виведення базової інформації бот виконує розрахунок різниці між ключовими параметрами: температура, вологість, сила вітру тощо. Наприклад, якщо температура в Києві +12°C, а в Парижі +18°C – бот вкаже: «У Парижі тепліше на 6°C». Такий формат не лише інформативний, а й дозволяє приймати практичні рішення, наприклад – краще планувати час подорожі.

Як і в інших розділах бота, порівняння супроводжується порадами щодо одягу для кожного з міст. На основі погоди бот окремо виводить рекомендації, що варто надягти в кожному з місць – це може бути особливо корисно для людей, які подорожують між містами в межах одного дня.

Окрім текстового повідомлення, бот також надсилає голосове озвучення результатів порівняння за допомогою Text-to-Speech API. Це забезпечує зручність доступу до інформації для тих користувачів, які перебувають у русі або мають візуальні обмеження.

Для візуального доповнення порівняння бот може також надіслати зображення кожного міста з Unsplash API, підібране за назвою міста. Це створює додатковий емоційний ефект, допомагаючи користувачеві візуалізувати атмосферу обраних місць. Поєднання даних з візуальним контентом не тільки покращує сприйняття інформації, але й робить взаємодію з ботом більш привабливою та інтерактивною.

У результаті, функція порівняння міст у Telegram-боті перетворює просту перевірку погоди на інтерактивний аналітичний досвід, поєднуючи текстову, голосову та візуальну інформацію. Це розширює практичну

цінність сервісу та робить його корисним у більш широкому спектрі сценаріїв використання. Від планування подорожей до вибору місця для переїзду, бот надає комплексну інформацію, що дозволяє користувачам приймати більш обґрунтовані рішення.

3.3 Тестування роботи застосунку

Після завершення основної стадії розробки Telegram-бота для інформування про погоду постала задача проведення комплексного, глибокого та багатоетапного тестування, яке є невід’ємною частиною усього життєвого циклу будь-якого програмного продукту [30-32]. Особливо це актуально для застосунків, що активно взаємодіють з користувачем у режимі реального часу та використовують велику кількість зовнішніх сервісів, як у випадку нашого бота. Тестування Telegram-бота включало не лише перевірку базових функцій, але і повну перевірку усіх логічних блоків, сценаріїв взаємодії, реакцій на неочікувані дії користувача, перевірку стійкості до помилок зовнішніх API, перевірку точності мовних відповідей, стабільності обробки локацій, тестування системи розсилки сповіщень, перевірку роботи розкладу через node-cron, а також цілісності збереження користувацьких налаштувань у базі даних MongoDB.

На початковому етапі тестування особлива увага приділялася функціональній логіці роботи бота. Перевірялися всі ключові сценарії використання: від моменту старту взаємодії з ботом, коли він вітає користувача і пропонує вибрати мову, до глибокої перевірки кожної з головних функцій, таких як «Отримати прогноз погоди», «Порівняння міст», «Отримання теплової карти» чи «Увімкнення сповіщень». Було протестовано всі можливі комбінації вибору мови, геопозиції, назв міст, термінів прогнозу та отримуваних результатів, щоб переконатися, що за будь-яких умов бот реагує очікуваним чином і не дає збоїв. Для прикладу, під час тестування

функції порівняння міст вводились як реальні, так і неіснуючі назви, щоб перевірити, як система обробляє помилки – чи не зависає бот, чи правильно виводить повідомлення про помилку, чи не втрачає поточний контекст взаємодії.

Важливим напрямком тестування була перевірка інтеграцій із зовнішніми API. Наш бот активно взаємодіє з OpenWeatherMap, Wit.ai, Voicerss, Geoapify та Google Cloud Text-to-Speech. Кожен з цих сервісів міг у певний момент бути недоступним або повернути некоректну відповідь, тому важливо було перевірити, як бот поводиться у випадках втрати з'єднання, тайм-аутів або помилок автентифікації (наприклад, якщо API-ключ недійсний або закінчився ліміт). Крім цього, особливо ретельно перевірялась система голосового озвучування прогнозу. Бот мав згенерувати текстову відповідь мовою користувача, надіслати її у чат, паралельно згенерувати озвучку цієї ж відповіді та надіслати аудіофайл. Тестування показало, що навіть при значному навантаженні система працює стабільно, а голосові відповіді звучать чітко, без обрізань чи збоїв, відповідають контексту повідомлення і звучать природно.

Окрема серія тестів присвячувалася багатомовності. Кожну функцію ретельно перевірено українською та англійською мовами, щоб гарантувати коректну локалізацію всіх елементів інтерфейсу, повідомлень та озвучки. Особливу увагу приділено перевірці збереження мовних налаштувань у базі даних та їх автоматичного застосування при подальших взаємодіях.

Ще однією важливою частиною тестування стала перевірка автоматичних сповіщень. Використовуючи модуль node-cron, бот щодня о 07:00 мав надсилати прогноз погоди користувачам, які ввімкнули цю функцію, а також надсилати миттєві сповіщення у випадку, якщо очікується дощ. Було змодельовано різні часові пояси, збої підключення до інтернету, одночасні запити – все це для того, щоб переконатися: незалежно від умов бот буде надсилати інформацію вчасно та коректно.

Усі ці етапи тестування завершилися повною фінальною перевіркою всіх компонентів, де Telegram-бот працював у реальних умовах максимально наближених до тих, у яких буде працювати після релізу. Ретельно перевірялися всі кнопки, послідовність переходів у меню, сценарії повторного використання бота, ситуації, коли користувач змінює думку під час вибору, або надсилає неочікувані типи даних. Кожен випадок був відпрацьований так, щоб бот завжди давав адекватну, ввічливу і логічну відповідь.

Пошук Telegram-бота здійснюється безпосередньо через вбудований пошук Telegram, ввівши нікнейм бота. Бот є загальнодоступним, що дозволяє будь-якому користувачу почати взаємодію з ним без попередньої авторизації або складних налаштувань. При відкритті вікна чату з ботом, перш ніж натиснути кнопку Start, користувач бачить вітальну заставку (рис. 3.1), яка надає коротку, але вичерпну інформацію про функціональні можливості бота.

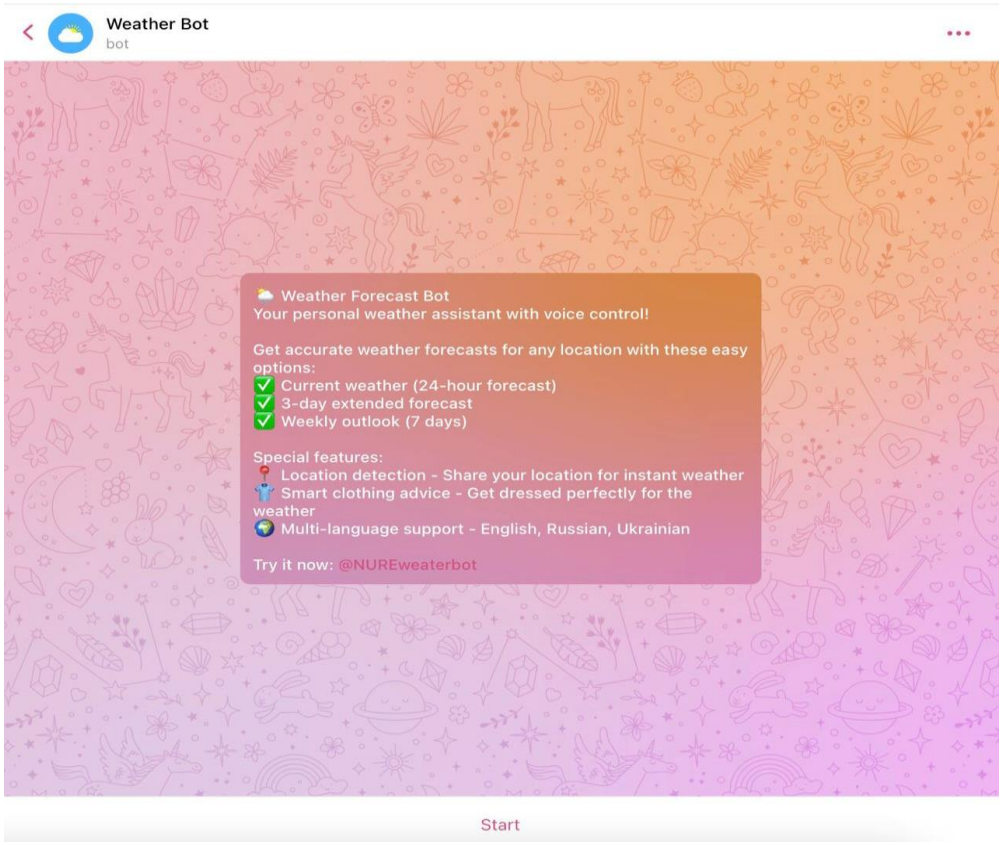

Рисунок 3.1 – Стартовий екран бота

На зображенні видно основні особливості, серед яких: точний прогноз погоди на вибраний період (24 години, 3 дні або 7 днів), автоматичне визначення місцеположення, рекомендації щодо одягу залежно від погодних умов, голосове озвучення тексту прогнозу та підтримка декількох мов інтерфейсу (англійська, українська). Це дозволяє користувачам з різних регіонів комфортно працювати з ботом.

Після натискання кнопки Start, бот надсилає привітальне повідомлення і відображає три кнопки з вибором мови: English, Русский, Українська (рис. 3.2). Цей вибір є важливим, оскільки визначає мову всіх наступних повідомлень, кнопок меню, текстових та голосових прогнозів. Мова зберігається для подальшої сесії, що робить використання ще зручнішим.

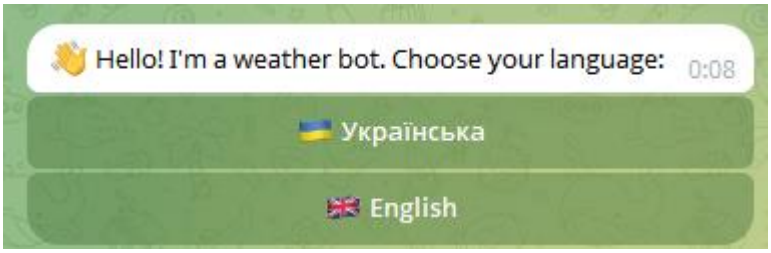

Рисунок 3.2 – Привітальне повідомлення з вибором мови

Після того як користувач обирає мову, бот формує головне меню – це набір основних кнопок, за допомогою яких реалізується вся функціональність бота. Користувачу пропонуються чотири ключові опції: Get Forecast, Notifications, Thermal Map, Compare Cities (рис. 3.3). Меню створене з урахуванням інтуїтивності – всі кнопки мають зрозумілі назви та логічне групування функцій. Даний підхід забезпечує легкість навігації навіть для нових користувачів, дозволяючи швидко знаходити потрібну інформацію та ефективно взаємодіяти з ботом. Чітка структура меню сприяє зниженню ймовірності помилок і підвищує загальне задоволення від користування сервісом.

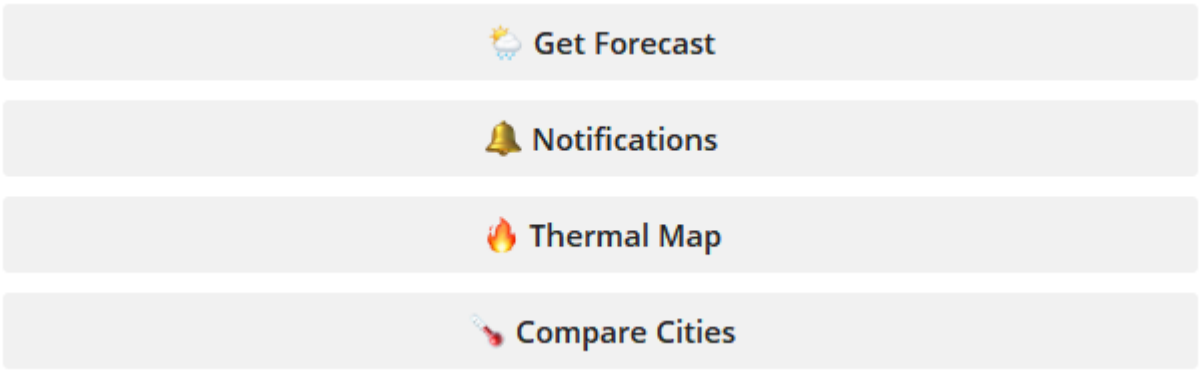

Рисунок 3.3 – Головне меню бота

Натиснувши на кнопку Get Forecast, користувач переходить до процесу отримання прогнозу погоди. На цьому етапі бот пропонує визначити місто, для якого потрібно отримати інформацію. Існує два варіанти: поділитися поточною геопозицією (рис. 3.4) або ввести назву міста вручну (рис. 3.5). Перший варіант дозволяє автоматично визначити місцезнаходження користувача, що значно економить час і спрощує взаємодію. Другий варіант корисний у випадках, коли користувач хоче отримати прогноз для іншого міста або перебуває в місці з нестабільним геолокаційним сигналом.

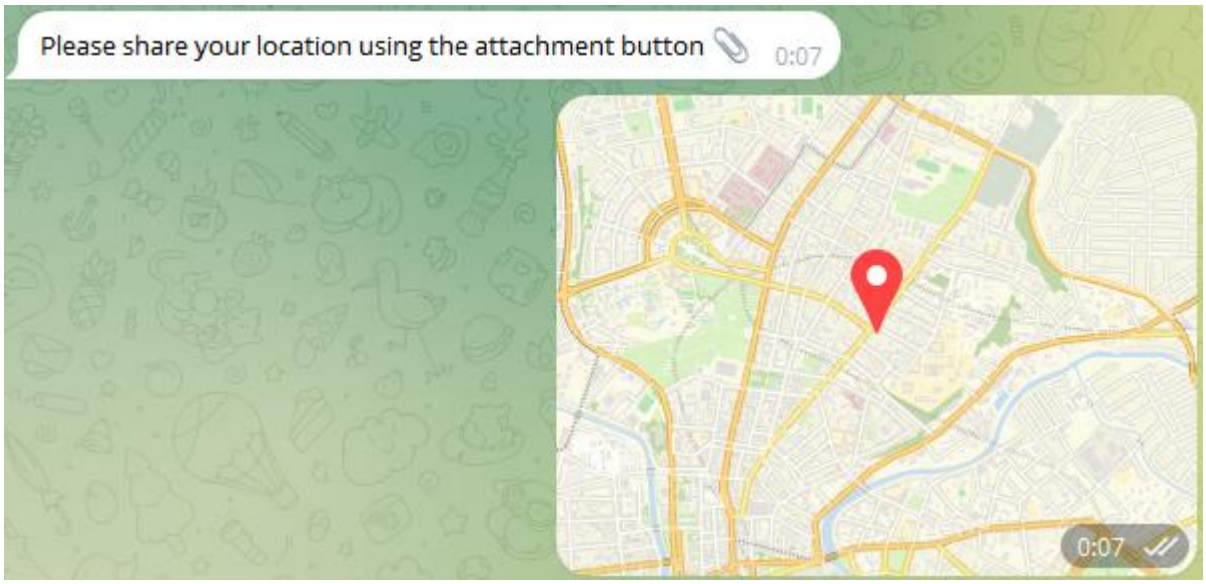

Рисунок 3.4 – Надсилання геопозиції

Рисунок 3.5 – Ручне введення міста

Після того як місто визначене, бот запитує в користувача, на який період він хоче отримати прогноз: на 24 години, на 3 дні або на 7 днів (рис. 3.6). Кожна з опцій має своє призначення: короткостроковий прогноз дозволяє оперативно реагувати на погодні зміни, триденний – зручно використовувати для планування поїздок або подій, а тижневий – дає повне уявлення про погодні умови на найближчий період.

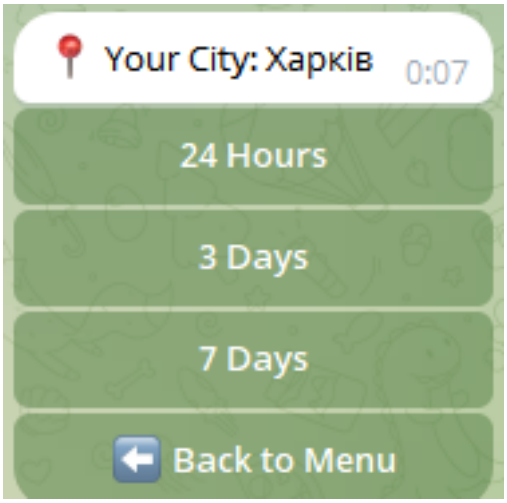

Рисунок 3.6 – Вибір терміну прогнозу

Після вибору терміну, бот надсилає докладний прогноз (рис. 3.7), що містить таку інформацію:

– температура повітря;
– вологість повітря;
– атмосферний тиск;
– хмарність;
– швидкість та напрямок вітру;

– ймовірність опадів.

Крім метеорологічних показників, у повідомленні додається рекомендація щодо одягу – бот аналізує погоду та радить, що найкраще вдягти, наприклад: «Захопіть парасолю, можливий дощ», «Сьогодні тепло, легкий одяг підійде ідеально» тощо. Разом із текстовим прогнозом надсилається також голосове повідомлення, яке озвучує текст повністю (рис. 3.8). Це особливо зручно для водіїв або користувачів із порушенням зору. Комплексне інформування, що поєднує текст, персоналізовані поради та аудіо, значно покращує доступність інформації та забезпечує максимальний комфорт для всіх категорій користувачів.

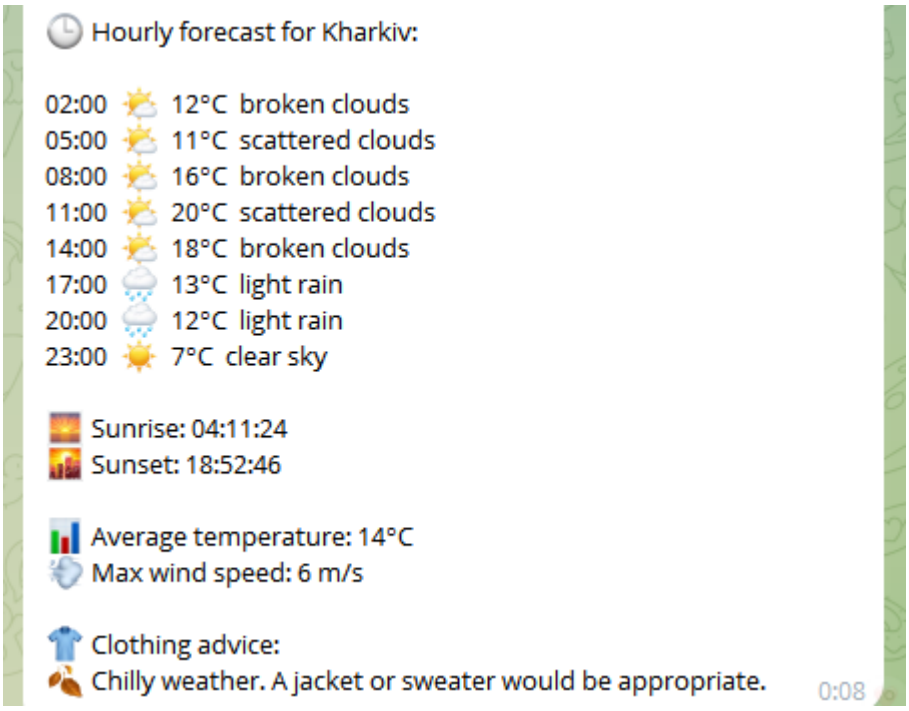

Рисунок 3.7 – Текстовий прогноз погоди з порадою щодо одягу

Рисунок 3.8 – Озвучений прогноз погоди

Після цього бот пропонує розширену версію поради щодо одягу, яка враховує деталі: вітер, температуру вранці та ввечері, рівень вологості, потенційні опади. Це дозволяє користувачу більш точно підготуватися до погодних умов (рис. 3.9).

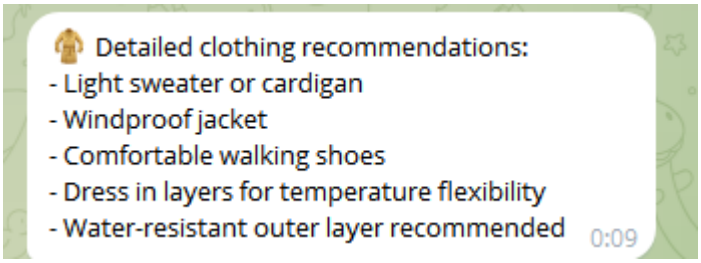

Рисунок 3.9 – Розширені рекомендації по одягу

У розділі Notifications користувач має можливість налаштувати автоматичні повідомлення, що дозволяє залишатися в курсі погоди без необхідності вручну запитувати інформацію. Після натискання цієї кнопки бот пропонує два режими (рис. 3.10, 3.11):

– щоденні повідомлення – прогноз погоди надсилається щодня о 7:00 ранку. Це ідеальний варіант для користувачів, які хочуть розпочати день із актуальної інформації про погоду;

– сповіщення про дощ – якщо в обраному місті очікуються опади, бот надсилає миттєве попередження. Така функція є незамінною для планування виходів із дому, поїздок або відкритих заходів.

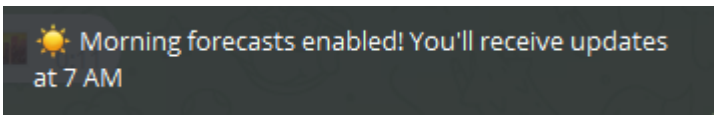

Рисунок 3.10 – Налаштування щоденних повідомлень

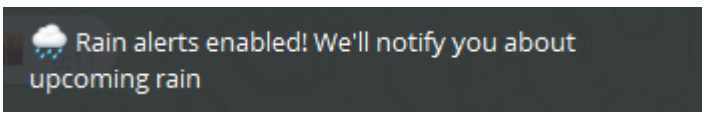

Рисунок 3.11 – Увімкнення сповіщення про дощ

Функція Thermal Map дозволяє переглянути термальну карту обраного міста – це графічне зображення температурного розподілу на території міста в режимі реального часу. Користувач, як і раніше, обирає спосіб визначення місця: геопозиція або ручне введення назви міста. Після цього бот надсилає зображення карти з температурними зонами (рис. 3.12), а також короткий прогноз на поточний момент. Це зручно для візуального аналізу метеоумов у великому місті або регіоні. Завдяки такому інструменту, сухі температурні дані перетворюються на наочне графічне представлення, що дозволяє не лише отримати цифри, а й інтуїтивно зрозуміти розподіл тепла, що надзвичайно корисно для ефективного планування пересування чи заходів.

Рисунок 3.12 – Термальна карта і поточний прогноз

Режим Compare Cities реалізований для порівняння погодних умов у двох різних містах. Користувач по черзі вводить назву першого (рис. 3.13), а потім другого міста. Бот формує окремі повідомлення з прогнозами для кожного міста, а потім генерує порівняльну таблицю або текст, у якому зазначає відмінності в температурі, вологості, вітрі та інших показниках (рис. 3.14). Це може бути особливо корисно для мандрівників або людей, що планують переїзд.

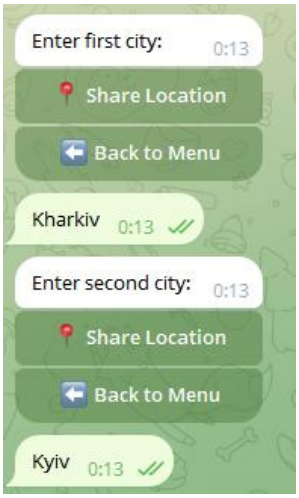

Рисунок 3.13 – Введення двох міст

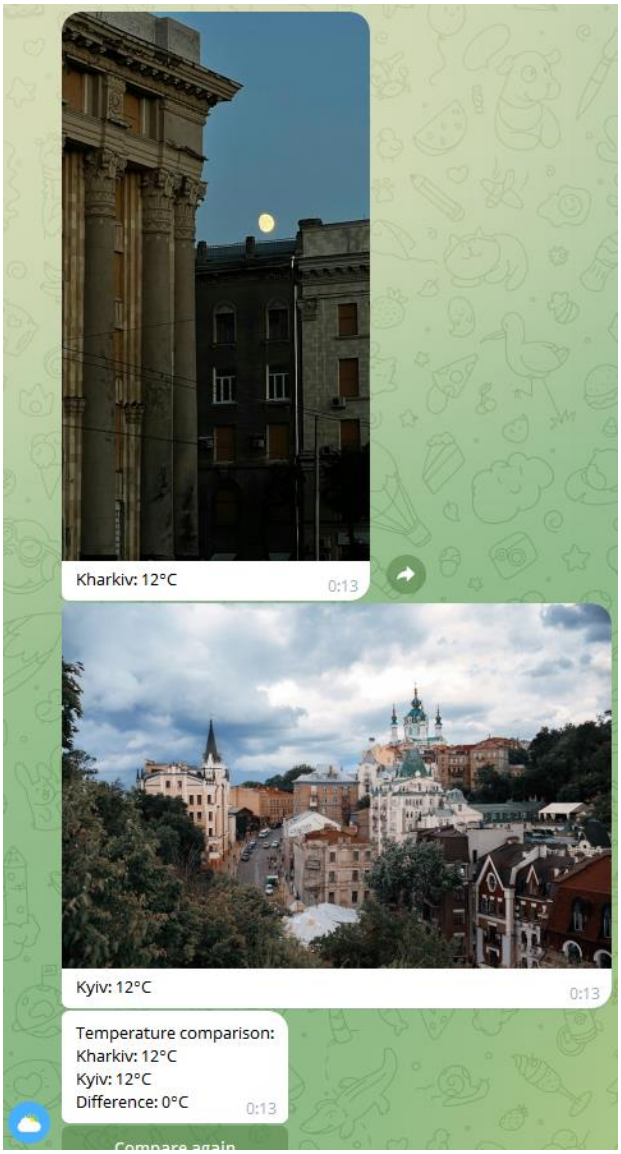

Рисунок 3.14 – Порівняння погодних умов у двох містах

3.4 Перспективи подальшої роботи

Розроблений Telegram-бот для прогнозу погоди забезпечує зручну та інтуїтивну взаємодію з користувачем, надаючи точну інформацію про поточну погоду, короткострокові та довгострокові прогнози, рекомендації щодо одягу та озвучення тексту прогнозу. Завдяки автоматичному визначенню місця перебування, можливості отримувати термальні карти, порівнювати погоду у двох містах та отримувати сповіщення, бот охоплює більшість ключових потреб користувачів у повсякденному житті. Його багатофункціональність, інтерактивність та доступність роблять його не просто інформаційним джерелом, а повноцінним персональним помічником у плануванні дня та прийнятті рішень з урахуванням погодних умов.

У подальшому планується реалізація таких функціональних покращень і розширень:

– інтеграція штучного інтелекту для адаптивних порад щодо одягу – бот враховуватиме не лише температуру, а й індивідуальні параметри користувача, щоб формувати персоналізовані рекомендації;

– додавання інтерактивної мапи погодних явищ у реальному часі – користувач зможе спостерігати за фронтами, опадами, циклонами на інтерактивній карті безпосередньо в Telegram;

– підтримка інтеграції з іншими месенджерами (наприклад, Viber або WhatsApp) – що дозволить залучити ширшу аудиторію;

– додавання історичних даних про погоду – можливість переглядати архівну інформацію про погодні умови за минулі періоди, що може бути корисним для аналізу довгострокових тенденцій або статистичних.

– Інтеграція з календарями та планувальниками – реалізація функції синхронізації погодних прогнозів із особистими календарями користувачів (наприклад, Google Calendar, Outlook Calendar), що дозволить автоматично враховувати погодні умови при плануванні подій та зустрічей.

ВИСНОВКИ

У рамках кваліфікаційної роботи було створено та реалізовано Telegram-бот для отримання прогнозу погоди, який забезпечує інтерактивну, зручну та багатофункціональну взаємодію з користувачем. Основною метою проєкту стало спрощення доступу до актуальної метеорологічної інформації для широкого кола користувачів шляхом використання популярного месенджера Telegram як платформи для реалізації сервісу .

У процесі виконання роботи були проаналізовані існуючі рішення у сфері погодних застосунків і ботів, що дозволило визначити ключові потреби користувачів і створити максимально функціональний та інтуїтивно зрозумілий інтерфейс. У результаті реалізовано такі основні можливості Telegram-бота:

– автоматичне визначення геолокації користувача для швидкого отримання прогнозу погоди в поточному місці перебування;

– можливість вибору мови інтерфейсу (англійська, українська), що робить бот доступним для ширшої аудиторії;

– перегляд прогнозу погоди на 24 години, 3 дні або 7 днів із детальним описом погодних умов;

– інтеграція голосового повідомлення, яке озвучує текст прогнозу – це полегшує сприйняття інформації, особливо в русі або для користувачів з порушеннями зору;

– функція «Smart Clothing Advice», яка надає рекомендації щодо одягу на основі погодних умов;

– можливість отримувати щоденні ранкові сповіщення з прогнозом погоди або миттєві повідомлення у разі зміни погодних умов (наприклад, початку дощу);

– функціонал термальної карти з даними про поточну температуру в регіоні;

– інструмент порівняння погоди у двох містах з аналізом різниці температур, вітру, вологості та інших параметрів.

Для реалізації бота було використано Telegram Bot API, сучасні підходи до обробки користувацьких запитів, інтеграцію з погодними сервісами через API, а також оптимізовані механізми обробки повідомлень. Завдяки використанню цих технологій проєкт демонструє стабільну та швидку роботу, зручний інтерфейс і високу інформативність.

Telegram-бот відповідає сучасним вимогам до цифрових сервісів і має значний потенціал для масштабування та подальшого розвитку. Його застосування дозволяє значно спростити доступ до якісної метеоінформації в повсякденному житті, підвищуючи рівень комфорту, обізнаності та особистої підготовленості користувачів до погодних змін.

Реалізований функціонал створює надійну основу для впровадження нових інтелектуальних можливостей у майбутньому – таких як персоналізовані рекомендації на основі звичок користувача, історія погоди, візуалізація даних, машинне навчання для прогнозування та інші інновації. Таким чином, результати цієї роботи можуть стати базою для розширення функціональності не лише у межах Telegram, а й на інших платформах, у тому числі веб та мобільних додатках. Це відкриває широкі перспективи для подальшого розвитку проєкту та його інтеграції у ширшу екосистему цифрових сервісів, підкреслюючи його гнучкість та довгострокову цінність.

Результати роботи апробовано у вигляді тез доповіді під час X Міжнародної науково-технічної конференції «Поліграфічні, мультимедійні та web-технології» [33]. Факт апробації підтверджує високу наукову та практичну значущість розробленого рішення, засвідчуючи його відповідність актуальним стандартам у сфері інформаційних технологій. Отриманий досвід свідчить про актуальність дослідження та демонструє потенціал рішення для подальшого розширення та практичного використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Malyshev, M., & Ivashkin, V. (2021). Telegram Bots as a Tool for Distance Learning and Automation of Routine Tasks. Advances in Science, Technology and Engineering Systems Journal, 6(5), 121–127.

2. Shulha, M., & Vorobiova, A. (2020). Development of a Weather Forecast Telegram Bot Using OpenWeatherMap API. Computer Sciences and Information Technologies, 18(3), 92–97.

3. Elshafie, M., & Atia, M. (2019). Smart Weather Forecasting System Using Artificial Intelligence. International Journal of Advanced Computer Science and Applications, 10(6), 302–308.

4. Popov, I., & Sazonov, A. (2022). A Study of Conversational Interfaces in the Context of Telegram Bots. CEUR Workshop Proceedings, 3121, 134–143.

5. Yatsenko, A., & Kobylin, O. (2023). Usage of NLP for Voice–Based Weather Reporting Bots. Information Technology: Computer Science, Software Engineering and Cyber Security, 2, 55–63.

6. Romanenko, I., & Lytvyn, V. (2020). Telegram Bot Development Using Node.js and Telegraf Framework. Modern Information Systems, 5(2), 115–122.

7. Nguyen, H., & Pham, T. (2021). Geo–Localization in Mobile Bots Using Telegram API. International Journal of Interactive Mobile Technologies, 15(11), 97–105.

8. Safonov, R., & Sokolov, D. (2020). Development of Notification Systems Using Telegram Bots for Weather Alert Services. Informatics and Automation, 42(1), 33–39.

9. Kuznetsov, A., & Bondarenko, P. (2022). Application of Voice Synthesis for Weather Report Bots in Low–Bandwidth Environments. Journal of Computing and Information Technology, 30(4), 289–298.

10. Smirnov, M., & Kovalenko, I. (2023). Integration of OpenWeatherMap API in Serverless Applications Using Node.js. International Journal of Web Engineering, 11(2), 71–82.

11. Shevchenko, V., & Petrov, A. (2021). UX Design Patterns for Conversational Weather Assistants. Journal of Human–Computer Interaction Studies, 9(1), 45–53.

12. Al–Mashaqbeh, I. A., & Al Omoush, K. S. (2020). Smart Weather Monitoring Systems Based on Telegram Chatbots. International Journal of Computer Applications, 175(3), 17–22.

13. Taran, Y., & Holub, Y. (2023). Comparative Analysis of Weather Forecast APIs for Real–Time Bot Applications. Cybersecurity and Data Science, 6(1), 60–68.

14. Zhukov, D., & Petrenko, S. (2022). Automation of Weather–Based Notifications in Telegram Using Serverless Technologies. Scientific Journal of Automation and Control, 14(2), 88–94.

15. Morozov, D., & Yermak, A. (2023). Implementation of Multilingual Support in Telegram Bots for Weather Applications. Proceedings of the International Conference on Applied Linguistics and Digital Communication, 2(1), 102–109.

16. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. Системи управління, навігації та зв'язку. Збірник наукових праць, 5(57).

17. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2019). Advances in Spatio–Temporal Segmentation of Visual Data (Vol. 876). Springer Nature.

18. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non–parametric statistics methods in image classifiers based on structural description components. Telecommunications and Radio Engineering, 79(10).

19. Gorokhovatskyi, V., & Tvoroshenko, I. (2023). Identification of visual objects by the search request. International scientific symposium «INTELLIGENT SOLUTIONS–S» (pp. 25–27).

20. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al–Dhaifallah, M. (2023). Statistical data analysis models for determining the relevance of structural image descriptions. IEEE Access, 11, 126938–126949.

21. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, Advances in Electrical and Electronic Engineering, 21(1), pp. 19–27.

22. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non–parametric statistics methods in image classifiers based on structural description components. Telecommunications and Radio Engineering, 79(10).

23. Ibrahim, D. Y., Gorokhovatskyi, V., Tvoroshenko, I., & Mujahed, A. D. (2022). Classification of Images Based on a System of Hierarchical Features.

24. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET) (pp. 665–670). IEEE.

25. Kobylin, O., & Lyashenko, V. (2016). Contrast Modification as a Tool to Study the Structure of Blood Components.

26. Kinoshenko, D., Kobylin, O., Mashtalir, S., & Stolbovyi, M. (2019, March). Metric video retrieval speedup by irrelevant data elimination. In Eleventh International Conference on Machine Vision (ICMV 2018) (Vol. 11041, pp. 176–183). SPIE.

27. Gorokhovatskyi, V. A., Rusakova, N., & Tvoroshenko, I. S. (2020). The application of image analysis methods and predicate logic in applied problems of magnetic monitoring. Telecommunications and Radio Engineering, 79(20).

28. Daradkeh, Y. I., & Tvoroshenko, I. (2020). Technologies for making reliable decisions on a variety of effective factors using fuzzy logic. International Journal of Advanced Computer Science and Applications, 11(5).

29. Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Vlasenko, N. V. (2020). Using fuzzy clustering in structural methods of image classification. Telecommunications and Radio Engineering, 79(9).

30. Кобилін, О., Вечірська, І., Афанасьєв, А. (2024). Аналіз існуючих моделей глибинного навчання в задачах обробки природної мови. Information Technology: Computer Science, Software Engineering and Cyber Security, 3, 63–76, https://doi.org/10.32782/IT/2024–3–7

31. Кобилін, О., Вечірська, І., Кравченко, О. (2024). Порівняння нейронних мереж типу RNN та LSTM. Information Technology: Computer Science, Software Engineering and Cyber Security, 3, 97–107, https://doi.org/10.32782/IT/2024–3–10

32. Vechirska, I., Kobylin, O., Prokopiev , S., Vechirska, A., & Kucherenko, M. (2022). Building a logical network for solving the problem of car rental by means algebra of finite predicates. Computer Systems and Information Technologies, (2), 78–87. https://doi.org/10.31891/csit–2022–2–9

33. Саранча Д., Кобилін О. (2025). РОЗРОБКА БОТА ПОГОДНОГО ІНФОРМЕРА В TELEGRAM. X Міжнародна науково-технічна конференція «Поліграфічні, мультимедійні та web-технології» (Травень 14–17, 2025). Харків, Україна, с. 112–113.