

**МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ  
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ**

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

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: **РОЗРОБКА ПОГОДНОГО ВЕБДОДАТКУ ТА ЧАТ-БОТУ**

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

  
(підпис)

Кобилінського Владислава  
Володимировича  
(прізвище, ім'я, по батькові)

Науковий керівник

  
(підпис)

Грищенко Світлана  
Миколаївна, к.пед.н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь,  
вчене звання)

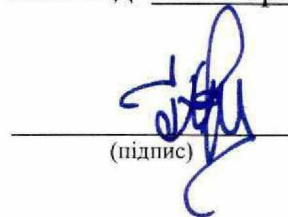
Керівник освітньо-  
професійної програми

  
(підпис)

Омельчук Антон  
Анатолійович, к.т.н.  
(прізвище, ім'я, по батькові, науковий ступінь,  
вчене звання)

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

Завідувач кафедри

  
(підпис)

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

Ірпінь – 2025

Реєстрація № 49

23.05.2025

(дата)



(підпис відповідальної особи)

Результати перевірки

до захисту  
(до захисту, на доопрацювання)

23.05.2025

(дата)



(підпис)

Результат захисту

5 (відмінно)  
(оцінка)

04.06.2025

(дата)



(підпис)

Голова комісії:

  
(підпис)

В.М. Франчук

Члени комісії:

  
(підпис)

М.М. Філоненко

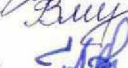
  
(підпис)

С.М. Грищенко

  
(підпис)

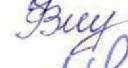
А.А. Омельчук

## БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

| Об'єкти контролю                                                                                                                                                               | Діапазон оцінки в балах | Фактична кількість балів | Примітки | Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|--------------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ</b>                                                                                                                   |                         |                          |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій) | від 0 до 30             | 27                       |          | Грищенко С.М. <br>Франчук В.М. <br>Філоненко М.М. <br>Грищенко С.М. <br>Омельчук А.А. <br>Суховерхня Л.П.  |
| 2. Дотримання вимог щодо оформлення                                                                                                                                            | від 0 до 10             | 8                        |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни                                                                                                       | від 0 до 5              | 4                        |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 4. Рецензент                                                                                                                                                                   | від 0 до 5              | 5                        |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>РАЗОМ БАЛІВ</b>                                                                                                                                                             | від 0 до 50             | 44                       |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

Оцінювання кваліфікаційної роботи науковим керівником – 46 балів.

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

| Оцінка за 100-бальною шкалою | Оцінка за шкалою ЄКТС | Оцінка за національною шкалою | Дата       | П.І.Б.: голови, членів комісії, секретаря, (підписи)                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------|-----------------------|-------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 90                           | A                     | 5 (відмінно)                  | 04.06.2025 | Франчук В.М. <br>Філоненко М.М. <br>Грищенко С.М. <br>Омельчук А.А. <br>Суховерхня Л.П.  |

### ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

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

Факультет фінансів та цифрових технологій

Кафедра комп'ютерних та інформаційних технологій і систем

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 122 «Комп'ютерні науки»

Освітня програма: «Інформаційні управляючі системи і технології (за галузями)»

Затверджую

Завідувач кафедри  Філоненко М.М.

«14» листопада 2024

Завдання на кваліфікаційну роботу

здобувачу вищої освіти

Кобилінському Владиславу Володимировичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Розробка погодного вебдодатку та чат-боту.

Затверджена наказом ректора від «13» листопада 2024 № 1717

2. Строк здачі закінченої роботи «08» травня 2025

3. Мета роботи: розробка інтегрованого вебдодатку та чат-бота для оперативного доступу до погодних даних.

Об'єкт дослідження: інформаційні системи для збору та обробки метеоданих.

Предмет дослідження: інтеграція вебдодатку та чат-бота для доступу до метеоінформації.

Орієнтовна база дослідження: аналіз літератури, існуючих рішень та вимог користувачів.

4. Вихідні дані до роботи: результати аналізу ринку, вимоги до системи, дані OpenWeatherMap API, технології розробки.

5. Перелік графічного матеріалу: рисунки (архітектура, методи), таблиці (порівняння, характеристики, результати), скріншоти інтерфейсів.

6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом):

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ТА ОПИС МЕТОДИКИ, ІНСТРУМЕНТАРІЮ ТА ДЖЕРЕЛА ІНФОРМАЦІЇ

1.1. Аналіз наукових досліджень та літературного огляду

1.2. Опис методики дослідження, інструментарію та джерел інформації

РОЗДІЛ 2. ПРАКТИЧНІ РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

2.1. Аналіз ринку, вимог користувачів та існуючих рішень

2.2. Розробка погодного вебдодатку та чат-бота

РОЗДІЛ 3. ТЕСТУВАННЯ ТА ОЦІНКА РОЗРОБЛЕНИХ РІШЕНЬ

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

## 7. Календарний план виконання роботи

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

8. Дата видачі завдання «14» листопада 2024

9. Керівник кваліфікаційної роботи

10. Керівник освітньо-професійної програми

11. Завдання до виконання отримав здобувач вищої освіти

  
(підпис)

Грищенко С.М.  
(прізвище, ініціали)

  
(підпис)

Омельчук А.А.  
(прізвище, ініціали)

  
(підпис)

Кобилінський  
(прізвище, ініціали)

## АНОТАЦІЯ

### Розробка погодного вебдодатку та чат-боту

*Мета роботи полягає в розробці вебдодатку з інтегрованим чат-ботом для оперативного отримання погодних даних.*

*Завданнями роботи є:*

1. Аналіз літератури та існуючих погодних сервісів.
2. Визначення вимог користувачів до погодного вебдодатку та чат-боту.
3. Розробка архітектури інтегрованої системи.
4. Реалізація прототипу вебдодатку та чат-боту.
5. Тестування розробленої системи.

*Предметом дослідження є процес розробки інтегрованого погодного вебдодатку та чат-бота.*

*Об'єктом дослідження є погодний вебдодаток та чат-бот.*

*Базою дослідження виступає аналіз наукової літератури та існуючих погодних вебдодатків і чат-ботів.*

*За результатами дослідження розроблено прототип погодного вебдодатку з інтегрованим чат-ботом.*

*Одержані результати можуть бути використані для надання оперативної інформації про погоду користувачам.*

*Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.*

*Ключові слова:* погодний вебдодаток, чат-бот, інтеграція, метеорологія, веброзробка.

*Рік виконання кваліфікаційної роботи:* 2025

*Рік захисту роботи:* 2025

## ABSTRACT

### Development of a weather web application and chatbot

---

*The aim of this work is to develop a web application with an integrated chatbot for the prompt receipt of weather data.*

*The tasks of the work are as follows:*

1. Analysis of literature and existing weather services.
2. Defining user requirements for the weather web application and chatbot.
3. Development of an integrated system architecture.
4. Implementation of a web application and chatbot prototype.
5. Testing of the developed system.

*The subject of the research is the development process of an integrated weather web application and chatbot.*

*The object of the research is the weather web application and chatbot.*

*The research is based on an analysis of scientific literature and existing weather web applications and chatbots.*

*The results of the research include the development of a prototype of a weather web application with an integrated chatbot.*

*The obtained results can be used to provide prompt weather information to users.*

*The qualification work consists of an introduction, three sections, conclusions, a list of references, and appendices.*

*Keywords:* weather web application, chatbot, integration, meteorology, web development.

*Year of qualification work:* 2025

*Year of work defense:* 2025

## ЗМІСТ

|                                                                                                                        |           |
|------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>ВСТУП.....</b>                                                                                                      | <b>3</b>  |
| <b>РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ТА ОПИС МЕТОДИКИ, ІНСТРУМЕНТАРІЮ ТА ДЖЕРЕЛА ІНФОРМАЦІЇ.....</b> | <b>6</b>  |
| 1.1. Аналіз наукових досліджень та літературного огляду.....                                                           | 6         |
| 1.2. Опис методики дослідження, інструментарію та джерел інформації.....                                               | 14        |
| <b>РОЗДІЛ 2. ПРАКТИЧНІ РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ.....</b>                                                                 | <b>21</b> |
| 2.1. Аналіз ринку, вимог користувачів та існуючих рішень.....                                                          | 21        |
| 2.2. Розробка погодного вебдодатку та чат-боту.....                                                                    | 31        |
| <b>РОЗДІЛ 3. ТЕСТУВАННЯ ТА ОЦІНКА РОЗРОБЛЕНИХ РІШЕНЬ.....</b>                                                          | <b>42</b> |
| <b>ВИСНОВКИ.....</b>                                                                                                   | <b>47</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>                                                                                 | <b>49</b> |
| <b>ДОДАТКИ.....</b>                                                                                                    | <b>54</b> |

## ВСТУП

В умовах стрімкого розвитку інформаційних технологій та цифрової трансформації суспільства, оперативне отримання достовірної інформації є ключовим фактором успішної діяльності в різних сферах. Зокрема, метеорологічні дані відіграють важливу роль у повсякденному житті людей, а також у функціонуванні багатьох галузей економіки, таких як сільське господарство, транспорт, енергетика та інші. Сучасні користувачі потребують швидкого та зручного доступу до актуальної інформації про погоду, що зумовлює необхідність розробки ефективних інформаційних систем, здатних задовольнити ці потреби. Вчені приділяють значну увагу розробці та впровадженню інформаційних технологій для метеорологічного забезпечення[1]. Даною проблематикою займалися вітчизняні науковці, такі як Майстренко С. Я., Рудніченко М., Капраль О. Р., Шаповал Р. О., Зуй Д. О., а також зарубіжні дослідники, зокрема Clare M. C. A. та Caldarini G.

Аналіз сучасних тенденцій розвитку вебтехнологій та штучного інтелекту свідчить про перспективність інтеграції цих напрямів для створення інноваційних інформаційних продуктів. Вебдодатки та чат-боти, як інтерактивні інструменти, здатні забезпечити оперативне надання погодних даних у зручному для користувача форматі. Дослідження показують, що інтеграція чат-ботів у вебдодатки значно підвищує зручність користування та оперативність отримання інформації[2, с. 53-58]. Таким чином, розробка погодного вебдодатку з інтегрованим чат-ботом є актуальним та своєчасним завданням, що відповідає сучасним потребам інформаційного суспільства.

Метою кваліфікаційної роботи є розробка інтегрованої системи для оперативного отримання та аналізу погодних даних шляхом створення погодного вебдодатку з вбудованим чат-ботом.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1) провести аналіз наукової літератури та існуючих підходів до збору, обробки та візуалізації погодних даних для систематизації теоретичних засад розробки погодних систем;

2) визначити основні вимоги користувачів до погодних сервісів на основі аналізу ринку існуючих рішень для забезпечення відповідності розроблюваної системи потребам цільової аудиторії;

3) розробити методологію інтеграції вебдодатку з чат-ботом для забезпечення зручної та інтуїтивно зрозумілої взаємодії користувача із системою;

4) здійснити проектування архітектури системи з урахуванням вимог безпеки, масштабованості та високої продуктивності для забезпечення надійності та ефективності функціонування розробленої системи;

5) реалізувати прототип погодного вебдодатку з інтегрованим чат-ботом із застосуванням сучасних технологій веброзробки та штучного інтелекту для практичної апробації запропонованих рішень;

6) провести тестування розробленої системи та оцінити її ефективність в умовах реального застосування для підтвердження працездатності та відповідності вимогам користувачів;

7) сформулювати рекомендації щодо подальшого удосконалення системи та визначити перспективні напрямки її розвитку для забезпечення можливості подальшої модернізації та адаптації системи до нових викликів та потреб.

Об'єктом дослідження є інформаційні системи, орієнтовані на автоматизований збір та обробку погодних даних.

Предметом дослідження є інтегроване рішення, що об'єднує вебдодаток і чат-бот як інструмент оперативної взаємодії користувача з інформацією про погоду.

Для реалізації поставленої мети в роботі використано комплекс методів дослідження:

1) аналіз наукової літератури для систематизації існуючих підходів і визначення теоретичних засад розробки погодних систем;

- 2) системний аналіз для виявлення ключових вимог користувачів та характеристик сучасних рішень;
- 3) метод моделювання для формування архітектури системи та її логічної структури;
- 4) прототипування з метою створення тестової версії вебдодатку та чат-бота;
- 5) експериментальне тестування для перевірки ефективності роботи розробленої системи в реальних умовах.

Практичне значення дослідження полягає у розробці інтегрованого погодного вебдодатку з чат-ботом, який забезпечує оперативний та зручний доступ до актуальної інформації про погоду. Розроблена система може бути впроваджена в різних сферах, включаючи комерційні організації та державні установи, для підвищення ефективності прийняття рішень та покращення якості обслуговування користувачів. Результати дослідження можуть бути використані для подальшого розвитку інтегрованих інформаційних систем.

Наукова новизна роботи полягає у комплексному підході до створення погодного вебдодатку з інтегрованим чат-ботом, що поєднує сучасні вебтехнології та штучний інтелект для забезпечення інтерактивної взаємодії з користувачем у режимі реального часу. Запропонована система відрізняється високою продуктивністю та масштабованістю, що відповідає сучасним вимогам до інформаційних систем.

## **РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ТА ОПИС МЕТОДИКИ, ІНСТРУМЕНТАРІЮ ТА ДЖЕРЕЛА ІНФОРМАЦІЇ**

### **1.1. Аналіз наукових досліджень та літературного огляду**

Сучасні дослідження в галузі погодніх вебдодатків набувають особливої актуальності через зростаючу потребу користувачів у швидкому доступі до метеорологічної інформації. Різноманітні наукові публікації, зокрема статті та монографії, висвітлюють теоретичні засади та практичні підходи до створення інтерактивних систем, які забезпечують високу точність прогнозів і зручність експлуатації, що своєю чергою створює надійну базу для подальших інноваційних розробок.

Аналіз фахової літератури свідчить, що погодні вебдодатки розглядаються як комплексні системи, здатні інтегрувати дані з численних джерел. Зокрема, Шибасев Д., Петров І., Носов М та інші наголошують на застосуванні сучасних алгоритмів обробки інформації задля підвищення точності аналізу, а також відзначають необхідність розробки уніфікованих методик, які оптимізують процеси збору та обробки даних із використанням передових технологій[3].

Наукові праці аналізують питання інтеграції погодніх сервісів у ширші інформаційні системи[1, 3, 30]. Автори таких досліджень розглядають як структурні особливості, так і технічні аспекти реалізації вебдодатків, підкреслюючи важливість модульності та адаптивності систем для задоволення сучасних вимог ринку. Водночас застосування гнучких методів розробки сприяє покращенню якості кінцевого продукту.

Фахові журнали приділяють особливу увагу практичним результатам впровадження погодніх вебсервісів у бізнес-процеси та державне управління. Дослідники, такі як Майстренко С. Я. та інші демонструють, що інтеграція інноваційних рішень дозволяє значно скоротити час, необхідний для аналізу даних, і підвищити точність прогнозування, що обґрунтовує потребу у

вдосконаленні алгоритмів обробки інформації з метою оптимізації роботи систем[1].

Сучасні наукові праці також охоплюють питання застосування штучного інтелекту в погодних вебдодатках. Аналіз публікацій свідчить про використання нейронних мереж для прогнозування погоди, що дозволяє створювати більш адаптивні та точні моделі, здатні враховувати складні метеорологічні залежності. Дослідники, зокрема Зуй Д. О.[14] та Clare M. C. A. та інші[8], підкреслюють, що інтеграція методів машинного навчання суттєво розширює можливості прогнозування та аналізу даних для оптимізації різних процесів.

У наукових роботах представлено огляд інструментальних засобів, що використовуються для розробки погодних вебдодатків. Аналіз публікацій демонструє, що основна увага приділяється оптимізації збору інформації за допомогою сучасних API, баз даних та інших електронних ресурсів, що забезпечують оперативність отримання даних. Автори, такі як Рудніченко М., Гришин С. та інші підкреслюють значущість застосування комплексних систем для стабільного збору метеорологічної інформації[3].

Літературний огляд також включає дослідження, що порівнюють різні підходи до інтеграції чат-ботів із погодними системами. У наукових працях аналізуються переваги використання інтерфейсів на базі штучного інтелекту, які забезпечують зручність взаємодії користувачів із системами прогнозування та оперативного отримання інформації. Автори, зокрема Коцюбівська К. та інші, та Капраль О. Р., Велика М. Б., відзначають, що впровадження чат-ботів сприяє підвищенню ефективності комунікації між користувачем і системою в сучасних умовах[4, 2].

Крім того, окремі дослідження наголошують на важливості порівняльного аналізу підходів, що застосовуються різними науковими школами. Рецензовані публікації дозволяють оцінити переваги та недоліки окремих методик, що використовуються при розробці погодних вебсервісів, і вказують на необхідність їх подальшого удосконалення. Автори підкреслюють

значення інтеграції сучасних технологій із традиційними методами для досягнення оптимальних результатів завдяки застосуванню аналітичних підходів.

Підсумовуючи огляд літератури, можна зробити висновок, що наукові дослідження у сфері погодніх вебдодатків демонструють комплексний підхід до вирішення як технічних, так і організаційних питань. Публікації у вигляді статей і монографій формують цілісну картину існуючих рішень та окреслюють перспективи їх подальшого розвитку, а також підкреслюють необхідність проведення додаткових досліджень для вдосконалення методик розробки інноваційних систем із застосуванням сучасних підходів.

Літературний огляд свідчить про стрімкий розвиток методів аналізу та інтеграції погодніх вебдодатків, що відкриває нові перспективи для застосування інноваційних рішень. Отримані висновки сприяють формуванню концепції майбутніх практичних розробок, при цьому автори, такі як Morcrette C. J. та інші, наголошують на важливості комплексного підходу до аналізу даних, що дозволяє ефективно інтегрувати сучасні технології у систему прогнозування для досягнення високої точності[8].

Сучасна наукова спільнота приділяє значну увагу вивченню та розробці чат-ботів, які стають невіддільною частиною сучасних інформаційних технологій. Дослідники, зокрема Беседовський О. та Ушакова І. О., аналізують різні аспекти створення систем автоматизованої взаємодії з користувачами, що сприяє забезпеченню оперативної комунікації, підтримці бізнес-процесів, ефективному управлінню даними, високій якості інформації та постійному вдосконаленню алгоритмів[5, 6].

Наукові дослідження в Україні у сфері розробки чат-ботів характеризуються комплексним підходом до аналізу алгоритмів обробки природної мови та методів штучного інтелекту. Українські вчені зосереджують увагу на адаптації таких систем до специфічних умов локального ринку та їх інтеграції у державні й освітні проекти, що сприяє покращенню якості взаємодії користувачів з інформаційними ресурсами.

Іноземні дослідники демонструють інноваційний підхід до розробки чат-ботів, використовуючи сучасні методи машинного навчання та глибокого аналізу даних. Їхні роботи наголошують на застосуванні нейронних мереж, алгоритмах обробки природної мови та оптимізації діалогових систем, що дозволяє створювати високоефективні, адаптивні та самонавчальні моделі взаємодії з користувачами[8, 12].

Порівняльний аналіз досвіду вітчизняних та зарубіжних дослідників свідчить про наявність як спільних тенденцій, так і суттєвих розбіжностей. Одні вчені віддають перевагу класичним алгоритмічним підходам, інші ж активно впроваджують методи глибокого навчання з метою підвищення точності розпізнавання мовних конструкцій. Такий аналіз сприяє формуванню узагальнених моделей для розробки інтерактивних систем[6].

Одним із важливих напрямків досліджень є вивчення алгоритмів обробки природної мови, що дозволяють чат-ботам адекватно розуміти контекст запитів користувачів. Вчені досліджують синтаксичні та семантичні моделі, які сприяють підвищенню якості комунікації, забезпечують правильне тлумачення запитів та сприяють створенню інтуїтивно зрозумілих систем, здатних ефективно адаптуватися до різних стилів спілкування[6, 12]. Можемо зазначити, що усі завдання, що входять до області обробки природної мови, можна класифікувати за чотирма основними категоріями: аналіз мови, синтез мови, аналіз тексту та синтез тексту (рис. 1.1).

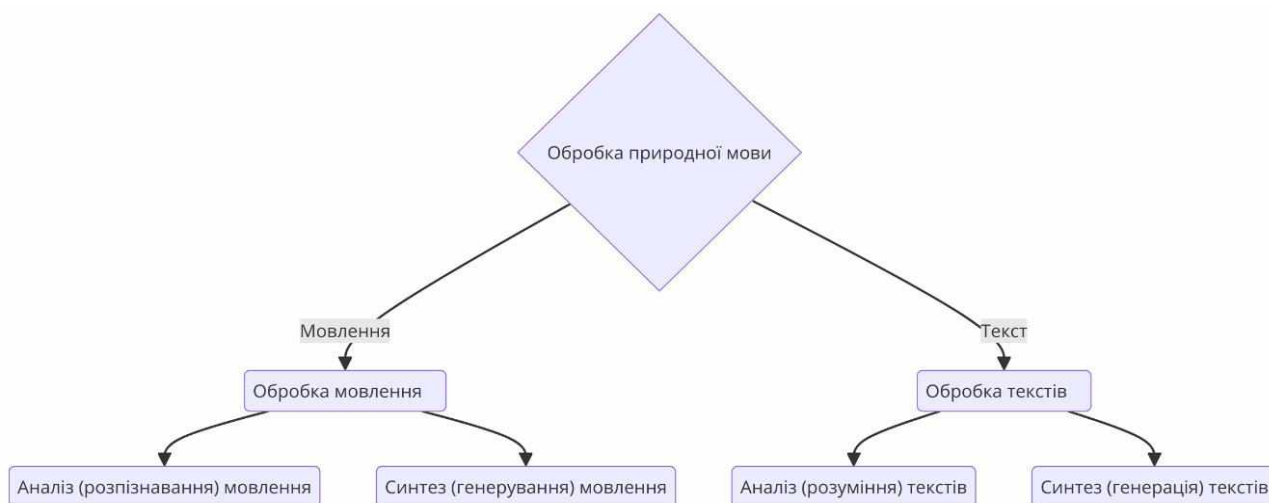


Рис. 1.1. Завдання обробки природної мови

З огляду на останні досягнення в області машинного навчання, багато дослідників інтегрують методи глибокого навчання у процес розробки чат-ботів[8, 12, 14]. Застосування нейронних мереж, а також алгоритмів класифікації та регресії дозволяє оптимізувати процес прийняття рішень, що забезпечує підвищення швидкості та точності відповідей і загальної ефективності інтерактивних систем.

Практичний досвід розробки чат-ботів демонструє успішну інтеграцію цих систем у сфери обслуговування клієнтів, маркетингу, фінансів та охорони здоров'я. Реальні кейси свідчать про підвищення якості надання послуг, зменшення витрат часу та ресурсів, а також про можливість адаптації систем до специфічних вимог окремих галузей, що підтверджується численними експериментальними дослідженнями.

Дослідники звертають увагу на проблеми інтеграції чат-ботів у існуючі інформаційні системи, зокрема на питання захисту даних, оптимізації використання обчислювальних ресурсів та забезпечення високої швидкості обробки запитів[12]. Виявлені труднощі свідчать про необхідність подальших розробок і впровадження нових підходів, спрямованих на усунення як технічних, так і організаційних бар'єрів[7].

За останні роки наукові публікації демонструють тенденцію до застосування комбінованих підходів у створенні чат-ботів, які поєднують традиційні алгоритмічні методи із сучасними технологіями штучного інтелекту. Такий напрямок дозволяє розробляти системи, здатні ефективно адаптуватися до змін у мовних моделях та враховувати індивідуальні особливості користувачів під час взаємодії.

Таким чином, аналіз досліджень та практичного досвіду розробки чат-ботів вітчизняними та зарубіжними авторами свідчить про висхідну актуальність цієї тематики. Порівняльний аналіз дозволяє виявити сильні та слабкі сторони існуючих підходів, що, своєю чергою, сприятиме розробці більш ефективних, адаптивних і безпечних інтерактивних систем для майбутніх застосувань у різних галузях.

Сучасна наукова спільнота демонструє багатогранність підходів до розробки погодних вебдодатків та чат-ботів, що знайшло відображення у працях представників різних наукових шкіл. Запропоновані методики ґрунтуються на використанні різних технологій, методів та інструментів, що спричиняє різноманітність думок та обґрунтувань проблематики.

Деякі дослідники дедалі частіше застосовують штучні нейронні мережі для підвищення точності прогнозування погодних умов. Наприклад, у статті «Combining distribution-based neural networks to predict weather forecast probabilities» представлено підхід, за яким нейронні мережі, що навчені на великих наборах метеорологічних даних, здатні генерувати повні розподіли ймовірностей для прогнозованих атмосферних параметрів. Автори стверджують, що цей метод дозволяє не лише значно покращити точність прогнозів внаслідок ефективної обробки великих обсягів інформації, але й забезпечує можливість оцінювання невизначеності прогнозів, що суттєво перевищує можливості традиційних методів[8].

Інша група дослідників орієнтується на розробку інтерактивних вебсайтів для прогнозування погоди із застосуванням сучасних вебтехнологій. Так, у роботі Міндера І.Ю. «Розробка інтерактивного вебсайту для прогнозування погоди на основі метеоданих» пропонується створення вебсайту, який забезпечує користувачам доступ до точних і актуальних метеорологічних даних [9]. Для реалізації цього рішення автор використовує мову програмування Python та бібліотеки Streamlit, PyOWM і Matplotlib, що дозволяє створити зручний вебінтерфейс для отримання та візуалізації даних. Такий підхід сприяє підвищенню обізнаності користувачів про погодні умови завдяки інтуїтивно зрозумілому інтерфейсу.

Щодо розробки чат-ботів для моніторингу погодних умов, існують альтернативні підходи. У дослідженні Шаповала Р.О. «Розробка чат-боту моніторингу погодних умов для месенджера Telegram» розглядається створення чат-бота, який забезпечує користувачів оперативною інформацією про погоду через платформу Telegram[10]. Автор обґрунтовує вибір саме цієї платформи її

популярністю та зручністю, а також використовує фреймворк Aiogram для реалізації клієнтської частини, що дозволяє забезпечити швидкий і зручний доступ до метеорологічних даних без необхідності використання окремого додатку або вебсайту.

Порівняльний аналіз зазначених підходів свідчить, що застосування нейронних мереж дозволяє досягти високої точності прогнозів завдяки здатності аналізувати великі обсяги даних та виявляти складні залежності між метеопараметрами. Проте реалізація таких систем може вимагати значних обчислювальних ресурсів і впровадження складних алгоритмічних рішень. З іншого боку, розробка інтерактивних вебсайтів та чат-ботів із застосуванням сучасних вебтехнологій забезпечує зручний доступ до метеорологічної інформації для широкого кола користувачів, проте точність прогнозування може залежати від якості використовуваних даних та алгоритмів.

Таким чином, представники різних наукових шкіл пропонують різноманітні підходи до розробки погодних вебдодатків та чат-ботів, кожен з яких має свої переваги та обмеження. Вибір конкретної методики залежить від поставлених завдань, наявних ресурсів та вимог кінцевих користувачів.

Сучасні наукові дослідження, присвячені розробці погодних вебдодатків та чат-ботів, виявили ряд аспектів, що потребують подальшого аналізу та вдосконалення. Серед них особливе місце займає інтеграція методів машинного навчання, спрямована на підвищення точності прогнозування погодних умов. Попри наявність досліджень, що використовують моделі ARIMA, SARIMA та SARIMAX для аналізу часових рядів питання оптимізації цих моделей для різних кліматичних умов залишається відкритим.

Ще одним важливим аспектом є забезпечення зручного й інтуїтивно зрозумілого інтерфейсу для користувачів погодних вебдодатків. Наприклад, дослідження Міндера І.Ю. рекомендує застосування сучасних технологій збору, обробки та візуалізації даних, проте ефективність таких підходів у різних користувацьких сценаріях потребує подальшого аналізу[9].

У галузі розробки чат-ботів, зокрема для освітніх цілей, виникає необхідність удосконалення алгоритмів обробки природної мови з метою забезпечення більш природної та ефективної взаємодії з користувачами. Дослідження, що проведені Наливайком О. та Малютіною А., розглядали застосування чат-ботів в освітньому процесі, проте питання адаптації таких систем до індивідуальних потреб студентів досі залишається недостатньо вивченим[11].

Важливим є забезпечення безпеки та конфіденційності даних користувачів при використанні погодних вебдодатків та чат-ботів. Необхідно розробляти та впроваджувати надійні механізми захисту інформації, що наразі не отримало достатнього розгляду в наукових дослідженнях.

Попри суттєвий прогрес у створенні чат-ботів, існують обмеження, пов'язані з розумінням контексту та забезпеченням гнучкості відповідей. Аналіз, проведений Caldarini G. та іншими підкреслює необхідність подальших досліджень у сфері вдосконалення алгоритмів обробки природної мови та застосування методів машинного навчання для підвищення ефективності чат-ботів[12, с. 7].

Додатково, інтеграція мультимодальних даних, зокрема тексту, зображень та аудіо, у функціонування чат-ботів і погодних вебдодатків може суттєво покращити користувацький досвід. Проте питання ефективної обробки та синхронізації таких даних потребує подальшого вивчення.

Також актуальним є напрямок адаптації погодних вебдодатків та чат-ботів для користувачів з обмеженими можливостями. Розробка інклюзивних інтерфейсів і функціоналу, який враховує потреби різних груп користувачів, залишається недостатньо опрацьованою.

Останнім важливим аспектом є забезпечення масштабованості та високої продуктивності таких систем при обробці великої кількості запитів у режимі реального часу, що є критично важливим для їх стабільної роботи. Оптимізація архітектури та застосування сучасних технологій хмарних обчислень можуть стати перспективними напрямками подальших досліджень.

Отже, попри значні досягнення у розробці погодних вебдодатків та чат-ботів, залишається низка невирішених аспектів, що потребують більшого аналізу та вдосконалення для забезпечення ефективнішої та безпечнішої взаємодії з користувачами.

## 1.2. Опис методики дослідження, інструментарію та джерел інформації

Під час розробки погодного вебдодатку та чат-бота має вирішальне значення застосування відповідних методів дослідження, що забезпечують наукову обґрунтованість і ефективність отриманих результатів. Серед основних методів, які можуть бути використані в цьому контексті, варто виділити економіко-математичне моделювання, загальнонауковий пошук та прикладні дослідження (рис. 1.2).

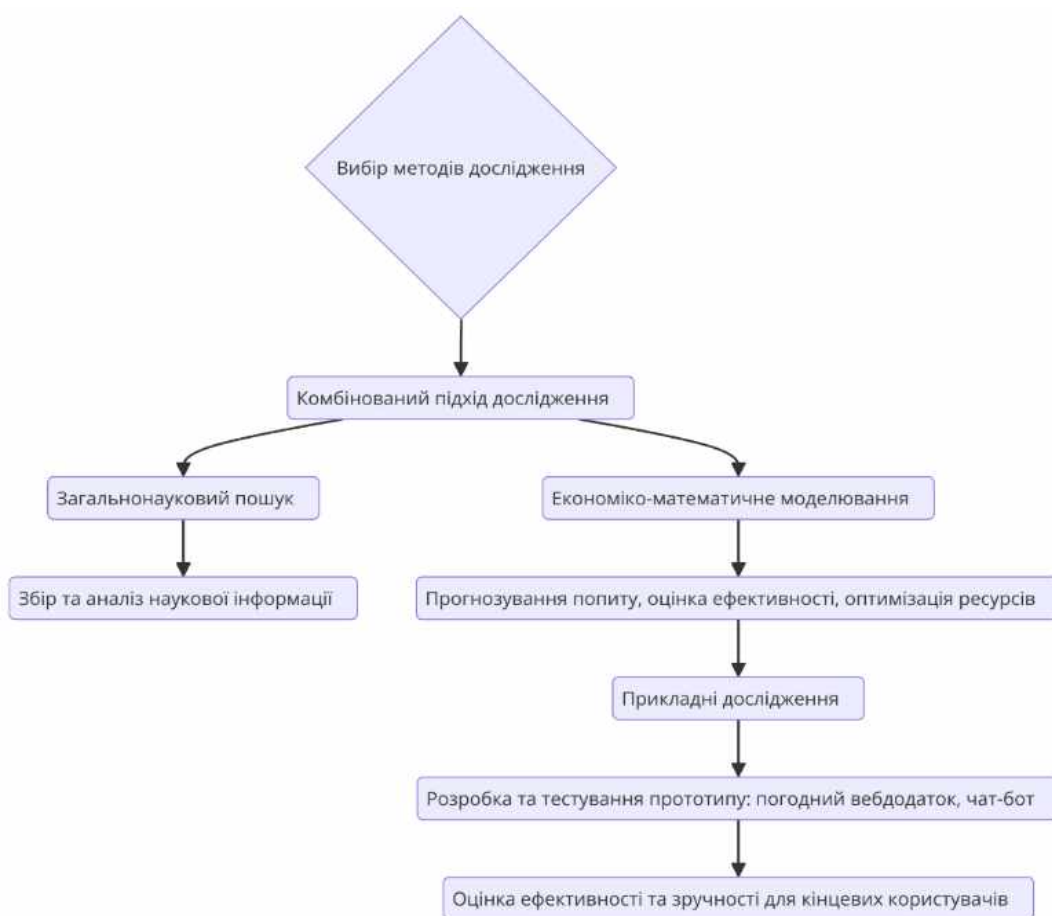


Рис. 1.2. Блок-схема комплексного застосування наукових методів у розробці погодного вебдодатку та чат-бота

Економіко-математичне моделювання полягає у застосуванні математичних методів і моделей для аналізу економічних процесів та явищ. У контексті розробки погодного вебдодатку та чат-бота цей метод може бути використаний для прогнозування попиту на такі продукти, оцінки їх економічної ефективності, а також для оптимізації ресурсів, необхідних для розробки та підтримки системи. Наприклад, можна створити модель, яка враховує витрати на розробку, потенційні доходи від експлуатації додатка та інші економічні показники, що дозволить об'єктивно оцінити доцільність проєкту.

Загальнонауковий пошук включає збір та аналіз інформації з різноманітних наукових джерел, таких як статті, монографії, конференційні матеріали та інші публікації. Цей метод сприяє отриманню актуальних даних про сучасні тенденції та досягнення у сфері розробки вебдодатків і чат-ботів, а також дозволяє вивчити досвід інших дослідників і практиків. Зокрема, можливо дослідити сучасні технології та інструменти, що застосовуються для створення чат-ботів, такі як обробка природної мови (NLP), машинне навчання та інші[13].

Прикладні дослідження орієнтовані на вирішення конкретних практичних завдань і проблем. У контексті даної роботи це включає розробку та тестування прототипу погодного вебдодатку та чат-бота, проведення експериментів із застосуванням різних технологій та інструментів, а також оцінку їх ефективності та зручності для користувачів. Наприклад, можна провести тестування різних алгоритмів обробки природної мови з метою визначення найбільш ефективного підходу до взаємодії з користувачами.

Вибір конкретних методів дослідження залежить від поставлених цілей та завдань роботи, а також від наявних ресурсів і обмежень. Комбінація економіко-математичного моделювання, загальнонаукового пошуку та прикладних досліджень дозволить отримати комплексний і обґрунтований результат, який має як теоретичну, так і практичну цінність.

Таким чином, застосування зазначених методів дослідження сприятиме забезпеченню системного підходу до розробки погодного вебдодатку та чат-бота, що дозволить врахувати як економічні аспекти проєкту, так і сучасні наукові досягнення та практичні потреби користувачів.

Під час створення погодного вебдодатку та чат-бота надзвичайно важливо обґрунтувати вибір методології та інструментарію, що сприятиме досягненню високої ефективності та надійності системи. Це вимагає чіткого визначення методів аналізу та розрахункових показників, які будуть застосовуватися на етапах розробки і впровадження.

Серед ключових методик аналізу особливе місце займає аналіз часових рядів, що дозволяє досліджувати і прогнозувати змінні погодних параметрів, спираючись на історичні дані. Зокрема, широко використовуються моделі ARIMA (AutoRegressive Integrated Moving Average) та їх сезонні розширення, такі як SARIMA і SARIMAX. Застосування цих моделей дає змогу враховувати як довгострокові тренди, так і сезонні коливання, що суттєво підвищує точність прогнозування.

Крім того, методи машинного навчання, зокрема використання нейронних мереж, набувають все більшого значення у сфері прогнозування погоди[8, 14]. Зокрема, мережі з довготривалою короткочасною пам'яттю (LSTM) ефективно обробляють послідовні дані та враховують тривалі залежності між спостереженнями, що робить їх придатними для моделювання складних погодних патернів і покращення точності прогнозів.

Для збору та обробки даних доцільно використовувати мову програмування Python разом із відповідними бібліотеками. Наприклад, бібліотека Pandas надає зручні засоби для маніпулювання та аналізу даних, Numpy забезпечує можливості для числових обчислень, а Matplotlib і Seaborn дозволяють візуалізувати результати аналізу[9, 14]. Ці інструменти забезпечують ефективну обробку великих обсягів інформації та сприяють наочному представленню результатів, що є важливим для подальшої розробки та вдосконалення системи[14].

Конкретний вибір методів аналізу та розрахункових показників визначається специфікою завдання та доступністю даних. Наприклад, для прогнозування температури може бути доцільно застосування моделей ARIMA, тоді як для прогнозування опадів доцільніше використовувати нейронні мережі. Водночас слід враховувати такі показники, як середня абсолютна похибка (MAE) та корінь середньоквадратичної похибки (RMSE), що дозволяють оцінити точність прогнозів і порівняти ефективність різних моделей.

Отже, раціональний вибір методології та інструментарію є критично важливим для успішної розробки погодного вебдодатку та чат-бота. Застосування сучасних методів аналізу даних та передових інструментів дозволить забезпечити високу точність прогнозування і зручність експлуатації системи кінцевими користувачами.

Під час розробки погодного вебдодатку та чат-бота надзвичайно важливо обрати вірогідні джерела інформації, які забезпечать високу точність і актуальність наданих даних. Основним засобом отримання метеорологічної інформації є спеціалізовані API, що надають доступ до поточних та прогнозованих погодних умов.

Одним із таких ресурсів виступає API від RainViewer, який забезпечує безплатний доступ до радарних даних із різних джерел. Цей API дозволяє отримувати інформацію про опади в режимі реального часу, що є критично важливим для забезпечення користувачів актуальними даними про погоду. Використання цього ресурсу не лише спрощує інтеграцію погодної інформації у вебдодаток, але й гарантує високу точність та надійність отриманих даних.

Крім того, RainViewer надає доступ до бази даних метеорологічних радарів, що містить детальну інформацію про їх розташування, статус та інші характеристичні показники. Така база даних може бути використана для розширення функціональності додатка, наприклад, для відображення інформації з окремих радарів або певних регіонів.

Для візуалізації погодних карт у додатку можна застосувати API карт погоди від RainViewer, який надає як історичні, так і прогнозовані дані з

метеорадарів, а також супутникові знімки. Це дозволяє створити інтерактивні та інформативні візуалізації погодних умов для користувачів[15].

Використання зазначених API забезпечує доступ до актуальних і точних даних про погоду, що є основою для надання якісної інформації користувачам як вебдодатку, так і чат-бота. Інтеграція цих джерел дозволяє створити функціональний і надійний сервіс, який задовольнить потреби аудиторії у своєчасній та правдивій метеорологічній інформації.

Окрім використання зовнішніх API, доцільно розглянути можливість зберігання отриманих даних у власній базі даних. Це сприятиме оптимізації роботи додатка, забезпечить швидкий доступ до часто запитуваної інформації та знизить залежність від сторонніх сервісів. Вибір відповідної системи управління базами даних (СУБД) визначається специфікою проекту, обсягом даних і вимогами до продуктивності.

При виборі джерел інформації слід також враховувати питання ліцензування та умови використання даних. Деякі API можуть мати обмеження щодо комерційного застосування або вимагати обов'язкового зазначення джерела даних у додатку. Тому перед інтеграцією необхідно детально ознайомитися з умовами використання обраних сервісів.

Таким чином, ретельний відбір і інтеграція вірогідних джерел інформації є ключовим етапом у розробці погодного вебдодатку та чат-бота, що забезпечить надання користувачам точної, актуальної та корисної інформації, підвищуючи загальну якість і привабливість сервісу.

Концепція створення погодного вебдодатку та чат-бота ґрунтується на інтеграції сучасних технологій з метою забезпечення користувачів вірогідною та актуальною інформацією про погодні умови. Основною задачею є розробка зручного та інтуїтивно зрозумілого інтерфейсу, що дозволяє оперативно отримувати необхідні метеорологічні дані у режимі реального часу.

Першим етапом розробки є визначення цільової аудиторії та аналіз її потреб. Це можуть бути як звичайні користувачі, зацікавлені в отриманні денного прогнозу погоди, так і фахівці, яким потрібна детальна метеорологічна

інформація для професійної діяльності. Ретельне вивчення потреб аудиторії дозволяє сформулювати функціональні можливості як вебдодатку, так і чат-бота.

Наступним кроком є вибір технологічного стека для реалізації проекту. Для розробки інтерфейсу вебдодатку доцільно використовувати сучасні фреймворки, такі як React або Angular, що забезпечують швидку та ефективну побудову користувацького інтерфейсу. Для серверної частини можливим рішенням є застосування Node.js або Python із використанням відповідних фреймворків, наприклад, Express або Django.

Щодо розробки чат-бота, важливо обрати платформу, яка буде найбільш зручною для кінцевих користувачів. Популярними варіантами є Telegram та Viber, оскільки вони мають широку аудиторію та пропонують зручні інструменти для розробки ботів. Наприклад, у місті Рівне запровадили чат-бот у Viber, що надає актуальну інформацію про погоду мешканцям міста та області[16].

Ключовим аспектом розробки є інтеграція з метеорологічними API. Використання перевірених джерел даних, таких як OpenWeatherMap або WeatherAPI, дозволяє забезпечити високу точність і актуальність отриманої інформації. При цьому необхідно впровадити механізми обробки та кешування даних, що зменшать навантаження на API і сприятимуть швидкому відгуку як вебдодатку, так і чат-бота.

Особливу увагу приділяють безпеці та захисту даних користувачів. Забезпечення захисту від несанкціонованого доступу здійснюється шляхом використання сучасних методів аутентифікації та авторизації, а також дотримання стандартів захисту персональних даних відповідно до чинного законодавства. Відповідно до статті 13 Закону України «Про захист персональних даних» (від 1 червня 2010 р. №2297-VI), усі суб'єкти, які здійснюють накопичення та зберігання персональних даних, зобов'язані впроваджувати ефективні заходи безпеки, що гарантують належну обробку та захист інформації користувачів[17].

Тестування та налагодження як вебдодатку, так і чат-бота є невіддільною частиною розробницького процесу. Необхідно провести як функціональне тестування для перевірки коректності роботи всіх компонентів, так і нефункціональне тестування для оцінки продуктивності та безпеки системи. Залучення реальних користувачів до бета-тестування сприятиме виявленню потенційних недоліків і подальшому покращенню якості продукту.

Після завершення розробки та тестування слід забезпечити регулярну підтримку та оновлення вебдодатку і чат-бота. Постійне оновлення даних, усунення помилок і впровадження нових функціональних можливостей сприятимуть підвищенню задоволеності користувачів і забезпечать тривалу актуальність продукту.

Отже, концепція розробки погодного вебдодатку та чат-бота передбачає комплексний підхід, що охоплює аналіз потреб користувачів, вибір оптимальних технологій, забезпечення високих стандартів безпеки та якості продукту, а також постійну підтримку та розвиток системи.

## РОЗДІЛ 2. ПРАКТИЧНІ РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

### 2.1. Аналіз ринку, вимог користувачів та існуючих рішень

Проектування нових технологічних рішень у сфері метеорології вимагає глибокого розуміння як технічних можливостей, так і соціальних потреб. Інтеграція вебдодатків із чат-ботами становить особливий інтерес, оскільки поєднує переваги деталізованого аналізу даних із миттєвим доступом через популярні месенджери. Такий підхід відповідає сучасним трендам на зростання мобільного трафіку та зменшення часу, який користувачі готові витратити на пошук інформації.

Однак розробка інтегрованих систем має свої складності. Серед них - забезпечення синхронізації даних між різними платформами, оптимізація продуктивності при обмеженій пропускній здатності мобільних мереж, а також захист конфіденційності користувачів. Вирішення цих завдань вимагає використання сучасних технологічних стеків, гнучкої архітектури та ретельного тестування.

Сучасний ринок погодних сервісів є надзвичайно насиченим та різноманітним, що відображає як високий попит на метеорологічну інформацію, так і стрімкий розвиток інформаційних технологій. Цей ринок можна схарактеризувати як висококонкурентний, де представлені як глобальні гравці з багатомільйонною аудиторією, так і локальні проекти, орієнтовані на потреби конкретних регіонів чи спільнот[18]. Споживачі мають доступ до широкого спектра інструментів для отримання інформації про погоду, включаючи вебсайти, мобільні додатки, спеціалізовані віджети, а також чат-боти, інтегровані в популярні месенджери. Важливо відзначити, що метеорологічна інформація є критично важливою не лише для побутового використання, але й для багатьох галузей економіки, таких як сільське господарство, транспорт, енергетика, туризм, страхування та інші. Тому точність, оперативність та доступність даних про погоду мають безпосередній вплив на ефективність прийняття рішень у цих сферах.

Ринок погодних сервісів можна умовно поділити на кілька сегментів залежно від платформи та способу надання інформації[19]. Перший сегмент – це вебдодатки, які надають доступ до метеорологічної інформації через веббраузер. До цього сегменту належать такі відомі сервіси, як Sinoptik, Gismeteo, AccuWeather, Weather та Ventusky. Кожен з них має свої особливості, наприклад, Sinoptik є одним з найпопулярніших в Україні, Gismeteo пропонує детальні прогнози та додаткові функції, а AccuWeather відомий своєю точністю та глобальним охопленням. Ventusky спеціалізується на візуалізації погодних даних. Другий сегмент – це мобільні додатки, які забезпечують доступ до погодної інформації на смартфонах та планшетах. Більшість згаданих вище вебсервісів мають власні мобільні додатки, що забезпечують зручність використання на мобільних пристроях. Третій сегмент, який активно розвивається останніми роками, – це чат-боти, інтегровані в месенджери, такі як Telegram та Facebook Messenger. Прикладом такого чат-бота є WeatherBot. Він дозволяє отримувати інформацію про погоду в режимі діалогу, що є зручним для багатьох користувачів.

Розвиток ринку погодних сервісів відбувається під впливом кількох ключових тенденцій. По-перше, спостерігається стійке зростання популярності чат-ботів як зручного та швидкого способу отримання інформації. Це пов'язано з загальним трендом на використання месенджерів для комунікації та отримання різноманітних послуг. По-друге, все ширше застосовуються технології штучного інтелекту (ШІ) та машинного навчання для підвищення точності прогнозів, персоналізації рекомендацій та покращення користувацького досвіду. Алгоритми ШІ дозволяють аналізувати великі обсяги даних та виявляти складні залежності, що недоступні для традиційних методів прогнозування. По-третє, зростає попит на гіперлокалізовані прогнози, тобто прогнози для конкретних невеликих територій, а не лише для великих міст. Це особливо актуально для сільського господарства, туризму та інших галузей, де точність прогнозу має критичне значення. По-четверте, користувачі очікують на персоналізовані сервіси, які враховують їхні індивідуальні потреби та

інтереси. Наприклад, це можуть бути спеціальні прогнози для алергіків, рибалок, спортсменів або людей з певними захворюваннями. Нарешті, погодні сервіси все частіше інтегруються з іншими платформами та додатками, такими як календарі, навігаційні системи, сервіси для планування подорожей, що забезпечує додаткову зручність для користувачів[20].

Для кращого розуміння функціональних можливостей різних погодних додатків доцільно представити їх у вигляді таблиці (табл. 2.1).

*Таблиця 2.1*

**Приклади погодних додатків та їх функціональні можливості**

| Назва сервісу         | Платформа      | Основні функції                                                                                                                                      | Додаткові функції                                                                                        |
|-----------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Sinoptik              | Веб, мобільний | Прогноз погоди (поточна, погодинна, на 10 днів), вибір міста, різні мови інтерфейсу.                                                                 | Геоманітні бурі, фази Місяця, народний прогноз, погода на курортах.                                      |
| Meteoфор              | Веб, мобільний | Прогноз погоди (поточна, погодинна, на місяць), вибір міста, різні мови інтерфейсу, погодні карти.                                                   | Пгодні інформери для сайтів, алергопрогноз, геоманітні бурі.                                             |
| AccuWeather           | Веб, мобільний | Прогноз погоди (поточна, погодинна, довгостроковий), вибір міста, різні мови інтерфейсу, інтерактивні карти, MinuteCast (прогноз опадів похвилинно). | Новини про погоду, відеопрогнози, інформація для алергіків, попередження про екстремальні погодні умови. |
| WeatherBot (Telegram) | Чат-бот        | Прогноз погоди за запитом (місто).                                                                                                                   | Немає.                                                                                                   |

*Джерело: складено автором.*

Для глибокого розуміння потреб користувачів погодних сервісів було проведено комплексне дослідження, що включало аналіз відгуків про наявні рішення, а також проведення власного опитування. Аналіз відгуків користувачів про популярні погодні додатки та вебсайти здійснювався на платформах Google Play та App Store, а також на спеціалізованих форумах, де

обговорюються метеорологічні сервіси. Це дозволило виявити як позитивні, так і негативні аспекти наявних рішень з погляду реальних користувачів.

На основі зібраної інформації було проведено узагальнення та виділено ключові вимоги користувачів до погодних сервісів. Виявилося, що найважливішим фактором для більшості користувачів є точність прогнозу[21]. Люди очікують, що інформація про погоду буде максимально вірогідною та відповідатиме реальним умовам, оскільки від цього залежить планування їхніх повсякденних справ та, в деяких випадках, безпека. Другим за важливістю фактором є зручність інтерфейсу. Користувачі хочуть, щоб додаток або вебсайт був інтуїтивно зрозумілим, простим у використанні та не перевантаженим зайвою інформацією. Швидкість роботи також відіграє важливу роль. Користувачі не хочуть чекати, поки завантажиться прогноз погоди, особливо якщо вони використовують мобільний інтернет з обмеженим трафіком. Актуальність інформації – ще один важливий аспект. Користувачі очікують, що дані про погоду будуть оновлюватися регулярно, щоб відображати поточну ситуацію та найближчий прогноз.

Крім основних вимог, було виявлено, що наявність додаткових функцій може бути суттєвою перевагою для певних груп користувачів. Наприклад, для людей, які страждають на алергію, важливою є інформація про концентрацію пилку в повітрі. Для тих, хто займається активними видами спорту на відкритому повітрі, корисними можуть бути дані про швидкість вітру, УФ-індекс та інші параметри. Можливість персоналізації сервісу, така як вибір одиниць вимірювання, мови інтерфейсу, теми оформлення, також є важливою для багатьох користувачів. Крім того, користувачі очікують, що погодний сервіс буде надавати інформацію для різних локацій, а не лише для поточного місцеперебування. Це дозволяє планувати поїздки, дізнаватися про погоду в інших містах, де живуть родичі чи друзі тощо. Нарешті, важливим є забезпечення доступності сервісу з різних пристроїв. Користувачі хочуть мати можливість отримувати інформацію про погоду як через вебдодаток на комп'ютері, так і через чат-бот на смартфоні[22].

Для візуального представлення пріоритетності різних вимог користувачів було побудовано кругову діаграму (рис. 2.1). На діаграмі показано відсотковий розподіл важливості різних аспектів погодних сервісів, виходячи з результатів аналізу відгуків.



Рис. 2.1. Основні потреби користувачів, пов'язані з отриманням інформації про погоду (на основі узагальнення вимог)

З метою визначення сильних та слабких сторін наявних рішень, а також для виявлення потенційних ніш для розробки власного продукту, було проведено порівняльний аналіз кількох популярних вебдодатків та чат-ботів. Для порівняння було обрано такі вебдодатки: Sinoptik, Gismeteo, AccuWeather, а також чат-бот WeatherBot (Telegram). Вибір саме цих сервісів зумовлений їхньою популярністю серед користувачів, а також різноманітністю функціональних можливостей, які вони пропонують.

Порівняльний аналіз проводився за такими критеріями: функціональність, точність, зручність та дизайн. Функціональність оцінювалася за наявністю основних та додаткових можливостей, таких як поточна погода, прогноз на різні періоди часу, погодні карти, інформація про геомагнітні бурі, алергопрогноз тощо. Точність прогнозу оцінювалася як суб'єктивно, на основі власного досвіду використання сервісів, так і на основі відгуків інших користувачів, знайдених в інтернеті. Зручність використання оцінювалася за такими параметрами, як інтуїтивність інтерфейсу, простота навігації, наявність мобільної версії або додатку, швидкість роботи. Дизайн оцінювався за візуальною привабливістю інтерфейсу, зручністю розташування елементів, відсутністю надмірної реклами[23].

Розглянемо детальніше кожен з обраних сервісів. Sinoptik є одним з найпопулярніших погодних вебсайтів в Україні. Він пропонує користувачам інформацію про поточну погоду, погодинний прогноз, прогноз на 10 днів, а також додаткові дані, такі як народний прогноз, погода на курортах, інформація про геомагнітні бурі та фази Місяця. За відгуками користувачів, Sinoptik має достатньо високу точність прогнозу для території України. Інтерфейс сайту є зручним та інтуїтивно зрозумілим, є також мобільний додаток для Android та iOS. Дизайн сайту можна схарактеризувати як приємний та не перевантажений рекламою.

Gismeteo – це міжнародний погодний сервіс, який надає детальні прогнози погоди для різних локацій по всьому світу. Крім поточної погоди та погодинного прогнозу, Gismeteo пропонує прогноз на місяць, погодні карти, а також додаткові функції, такі як погодні інформери для сайтів, алергопрогноз та інформацію про геомагнітні бурі. Gismeteo також має зручний інтерфейс та мобільні додатки для різних платформ.

AccuWeather – це глобальний постачальник метеорологічної інформації, який пропонує широкий спектр даних про погоду, включаючи поточну погоду, погодинний прогноз, довгостроковий прогноз, інтерактивні карти, а також унікальну функцію MinuteCast, яка надає прогноз опадів похвилинно.

AccuWeather відомий своєю точністю прогнозів та широким охопленням локацій. Крім вебсайту, AccuWeather пропонує мобільні додатки для Android та iOS.

WeatherBot (Telegram) – це простий чат-бот, який надає інформацію про погоду за запитом користувача. Він підтримує базові команди, такі як /start та /weather, і використовує дані з OpenWeatherMap. WeatherBot не має графічного інтерфейсу, вся інформація надається в текстовому вигляді. Перевагою WeatherBot є його простота та швидкість роботи, але функціональність є обмеженою порівняно з вебдодатками.

Аналіз переваг та недоліків наявних рішень дозволяє зробити певні висновки. Вебдодатки, такі як Sinoptik, Gismeteo та AccuWeather, мають широку функціональність, надають детальну інформацію про погоду, включаючи додаткові дані, такі як погодні карти, попередження про небезпечні явища, інформацію для алергіків тощо. Вони також мають зручні інтерфейси та мобільні додатки. Однак вебдодатки можуть бути повільнішими за чат-боти, вимагають більшого трафіку даних і не завжди зручні для використання на мобільних пристроях з невеликим екраном[24].

Чат-бот, такий як WeatherBot, навпаки, відрізняється швидкістю роботи та зручністю отримання інформації. Він не вимагає встановлення додаткових додатків, оскільки працює у месенджерах, якими користувачі вже користуються щодня. Проте чат-бот має обмежену функціональність, не надає візуалізації даних (наприклад, погодних карт) і повністю залежить від платформи, на якій він працює.

В результаті проведеного аналізу ринку, вимог користувачів та наявних рішень можна зробити такі висновки. По-перше, існує значний попит на зручні, точні та функціональні погодні сервіси, які б відповідали різним потребам користувачів. По-друге, користувачі цінують швидкість отримання інформації, простоту використання та наявність додаткових функцій, які можуть бути корисними для різних груп людей. По-третє, чат-боти є перспективним напрямком розвитку погодних сервісів, але вони мають певні обмеження щодо

функціональності та візуалізації даних. По-четверте, важливим є забезпечення кросплатформенності, тобто можливості доступу до сервісу з різних пристроїв (комп'ютерів, смартфонів, планшетів). І останнє, на ринку, попри велику кількість пропозицій, все ще не вистачає зручного поєднання вебдодатку та чат-боту в єдиній системі.

Враховуючи вищезазначені висновки, було прийнято рішення про розробку власного інтегрованого рішення, яке б поєднувало переваги вебдодатку та чат-бота, а також враховувало потреби різних груп користувачів. Таке рішення має бути функціональним, зручним, точним, кросплатформенним та персоналізованим. Воно повинно надавати як базову інформацію про погоду (поточна погода, прогноз), так і додаткові дані (попередження про небезпечні явища, погодні карти), а також мати інтуїтивно зрозумілий інтерфейс та швидкий доступ до інформації. Розробка такого рішення є актуальною та перспективною задачею, яка відповідає сучасним тенденціям розвитку ринку погодних сервісів та потребам користувачів.

Етап проєктування та розробки програмного забезпечення є ключовим у життєвому циклі будь-якого ІТ-продукту. Саме на цьому етапі теоретичні концепції, результати аналізу та сформульовані вимоги перетворюються на конкретний функціональний код та зручний користувацький інтерфейс. Розробка програмного забезпечення – це процес, що вимагає системного підходу, ретельного планування, обґрунтованого вибору інструментів та технологій, а також постійного контролю якості. Успішна реалізація залежить від багатьох факторів, включаючи кваліфікацію розробників, ефективність комунікації в команді (якщо така є), а також здатність адаптуватися до змін та розв'язувати проблеми, що виникають у процесі роботи[25].

Визначені потреби користувачів та виявлені недоліки наявних рішень стали основою для формування технічного завдання та визначення архітектури майбутньої системи. Особливу увагу було приділено вибору технологій, які б дозволили реалізувати поставлені завдання з максимальною ефективністю та забезпечили б гнучкість і масштабованість системи в майбутньому. Важливо

було знайти баланс між використанням сучасних передових технологій та забезпеченням стабільності й надійності роботи продукту.

Процес розробки погодного вебдодатку та чат-бота не обмежувався лише написанням коду. Він також включав проєктування архітектури системи, розробку дизайну користувацького інтерфейсу, вибір та інтеграцію зовнішніх сервісів (зокрема, OpenWeatherMap API), а також ретельне тестування на різних етапах. При проєктуванні архітектури враховувалися такі принципи, як модульність, слабка зв'язність компонентів та можливість незалежного розгортання вебдодатку та чат-бота. Це дозволяє спростити подальшу підтримку та модернізацію системи. Розробка дизайну користувацького інтерфейсу мала на меті створити інтуїтивно зрозумілий, естетично привабливий та адаптивний інтерфейс, який би забезпечував зручну взаємодію користувача з системою на різних пристроях[26].

Інтеграція з OpenWeatherMap API була одним із ключових технічних завдань, оскільки від якості та надійності джерела даних безпосередньо залежить точність та актуальність інформації, що надається користувачеві. Було вивчено документацію API, розроблено механізми надсилання запитів та обробки отриманих даних, а також враховано обмеження та умови використання сервісу. Тестування, як невіддільна частина процесу розробки, проводилося для виявлення та виправлення помилок, перевірки відповідності функціоналу вимогам, а також для оцінки продуктивності та стабільності роботи системи.

Після завершення аналітичного етапу та формування чіткого бачення майбутнього продукту, наступним логічним кроком є перехід до безпосередньої розробки програмного забезпечення. Цей етап вимагає застосування конкретних технологій, інструментів та методологій для втілення сформульованих вимог у функціональну систему. Результати аналізу ринку та вимог користувачів, слугували дорожньою картою для прийняття ключових архітектурних та технологічних рішень. Зокрема, виявлена потреба у поєднанні зручності швидкого доступу до інформації (що забезпечують чат-боти) та

можливості детального перегляду даних і розширених налаштувань (що характерно для вебдодатків) визначила основну концепцію розроблюваного продукту – створення інтегрованої системи «вебдодаток + чат-бот».

Вибір конкретного технологічного стека є одним із найважливіших рішень на етапі розробки, оскільки він безпосередньо впливає на продуктивність, масштабованість, безпеку та вартість підтримки майбутньої системи. Зокрема, буде обґрунтовано вибір HTML, CSS та JavaScript для розробки фронтенду вебдодатку, а також Python та бібліотеки aiogram для створення чат-бота в Telegram. Важливим аспектом є також вибір надійного джерела метеорологічних даних, яким став OpenWeatherMap API, що надає широкий спектр інформації та має зручний інтерфейс для розробників.

Процес розробки програмного забезпечення рідко буває лінійним. Часто він передбачає ітеративний підхід, що включає створення прототипів, тестування, отримання зворотного зв'язку та внесення змін. У такому випадку, розробка вебдодатку та чат-бота також відбувалася з урахуванням принципів гнучкої розробки. Спочатку було створено базовий функціонал, який потім поступово розширювався та вдосконалювався. Наприклад, для вебдодатку було реалізовано архітектуру «Single Page Application» (SPA), що дозволило створити швидкий та інтерактивний користувацький інтерфейс. Для чат-бота використовувалися можливості бібліотеки aiogram для створення зручних клавіатур та обробки різних типів повідомлень[27].

Важливою частиною процесу розробки є тестування. Воно проводилося на різних етапах для виявлення та усунення помилок, а також для перевірки відповідності реалізованого функціоналу поставленим вимогам. Тестування включало як перевірку окремих модулів (юніт-тестування), так і тестування взаємодії компонентів системи (інтеграційне тестування), а також перевірку роботи системи в цілому з погляду користувача (функціональне тестування). Результати тестування дозволили підвищити якість та надійність розробленого програмного продукту.

## **2.2. Розробка погодного вебдодатку та чат-боту**

У сучасних умовах цифрової трансформації ключовим аспектом успіху будь-якого сервісу є його здатність адаптуватися до різноманітних сценаріїв використання. Особливо це стосується метеорологічних рішень, де користувачі очікують миттєвого доступу до даних через зручні для них канали. Зростання популярності месенджерів, таких як Telegram, разом із постійним використанням вебдодатків для глибокого аналізу, створює унікальний виклик: поєднати швидкість та простоту чат-ботів із розширеними можливостями вебінтерфейсів. За даними Statista, понад 70% користувачів віддають перевагу сервісам, які інтегрують кілька платформ, що підкреслює важливість створення гнучких та крос-функціональних рішень[28].

Інтеграція вебдодатку з чат-ботом дозволяє не лише задовольнити різні потреби аудиторії, але й оптимізувати використання ресурсів. Наприклад, користувачі можуть отримувати базові прогнози через Telegram під час переміщень, а детальні аналізи - через вебінтерфейс під час планування подорожей або робочих завдань. Однак такий підхід вимагає ретельної синхронізації даних, врахування обмежень мобільних мереж та забезпечення безпеки передачі інформації. Використання сучасних технологій, таких як асинхронні запити та хмарні API, стає критичним для досягнення високої продуктивності та стабільності системи.

Розробка інтегрованої системи для надання інформації про погоду включала створення двох взаємопов'язаних компонентів: вебдодатку та чат-бота для платформи Telegram. Основною метою було створення зручного, функціонального та крос-платформного рішення, яке б відповідало потребам користувачів, виявленим у ході аналізу ринку. Загальна концепція розробленого рішення полягає в тому, щоб надати користувачам можливість отримувати оперативну та достовірну інформацію про погоду через зручні для них канали: вебдодаток для детального перегляду даних та чат-бот для швидкого отримання основної інформації.

Головними цілями розробки були: надання користувачам інтуїтивно зрозумілого інструменту для доступу до метеорологічної інформації, забезпечення високої точності та актуальності даних, а також створення крос-платформного рішення, яке б однаково добре працювало на різних пристроях – від стаціонарних комп'ютерів до смартфонів. Для досягнення цих цілей було поставлено низку завдань. По-перше, необхідно було розробити вебдодаток з адаптивним дизайном та зручним інтерфейсом, який би дозволяв користувачам легко знаходити потрібну інформацію. По-друге, потрібно було створити чат-бот у Telegram, інтегрований з вебдодатком, щоб користувачі могли отримувати основну інформацію про погоду безпосередньо в месенджері. По-третє, необхідно було забезпечити отримання даних з надійного джерела, яким було обрано OpenWeatherMap API, що гарантує високу якість та актуальність метеорологічної інформації. По-четверте, потрібно було реалізувати основний функціонал, такий як відображення поточної погоди, прогнозу на 5 днів, пошуку за назвою міста, визначення місцезнаходження користувача (геолокація) та налаштування (вибір одиниць вимірювання, теми оформлення, формату часу). І нарешті, важливою задачею була інтеграція чат-бота з вебдодатком, щоб користувачі могли за потреби переходити з месенджера на вебсторінку для отримання більш детальної інформації про погоду[29].

Розроблена система має чітко визначений набір функцій. Вебдодаток надає користувачам такі можливості: відображення поточної погоди (температура, опади, вітер, тиск, вологість, час сходу та заходу сонця), відображення прогнозу погоди на 5 днів (як по днях, так і погодинно), пошук за назвою міста, автоматичне визначення місцезнаходження користувача (за його згодою) та налаштування параметрів відображення (одиниці вимірювання, тема оформлення, формат часу). Чат-бот дозволяє користувачам отримувати інформацію про поточну погоду за запитом (вказавши назву міста або надіславши свою геолокацію), переглядати прогноз погоди на 5 днів, здійснювати пошук за назвою міста, визначати своє місцезнаходження, а також змінювати налаштування (зокрема, одиниці вимірювання). Чат-бот має важливу

функцію – можливість переходу на вебдодаток одним натисканням кнопки, що забезпечує зручну інтеграцію двох компонентів системи.

Архітектура системи складається з двох основних частин: вебдодатку та чат-бота. Вебдодаток, у свою чергу, побудований за принципом «Single Page Application» (SPA), тобто вся логіка обробки даних та взаємодії з API реалізована на стороні клієнта (у браузері). Це дозволяє уникнути перезавантаження сторінки при кожному запиті, що значно підвищує швидкість роботи та зручність використання. Фронтенд вебдодатку розроблено з використанням стандартних вебтехнологій: HTML для створення структури сторінок, CSS для стилізації та забезпечення адаптивності, та JavaScript для реалізації динамічної поведінки та взаємодії з користувачем. Бекенд як окремий сервер відсутній. Такий підхід спрощує розробку, розгортання та підтримку вебдодатку, а також зменшує витрати на хостинг[30].

Чат-бот розроблено для платформи Telegram, яка була обрана завдяки її популярності, широкій аудиторії та наявності зручного API для розробників. Для взаємодії з Telegram Bot API, а також для отримання даних про погоду з OpenWeatherMap API, використовується сучасна асинхронна бібліотека aiogram для мови програмування Python. Ця бібліотека надає зручні інструменти для обробки повідомлень, створення клавіатур, управління станами та асинхронного виконання запитів, що значно спрощує та пришвидшує процес розробки. Логіка обробки запитів у чат-боті реалізована таким чином: бот отримує повідомлення від користувача, аналізує його, визначає команду (якщо така є), виконує необхідні дії (звертається до OpenWeatherMap API для отримання даних, формує відповідь), а потім надсилає відповідь користувачеві.

Взаємодія компонентів системи відбувається наступним чином (рис. 2.2). Користувач може взаємодіяти або з вебдодатком, або з чат-ботом. Як вебдодаток, так і чат-бот, у свою чергу, надсилають запити до OpenWeatherMap API (або Geocoding API) для отримання необхідних даних. OpenWeatherMap API повертає дані у форматі JSON, які потім обробляються та відображаються користувачеві у відповідному вигляді. Вебдодаток зберігає налаштування

користувача (тема, одиниці вимірювання, формат часу) в `localStorage`, щоб забезпечити їх збереження між сесіями. Чат-бот надає користувачеві посилання на вебдодаток, що дозволяє швидко перейти до більш детальної інформації про погоду та розширених налаштувань.

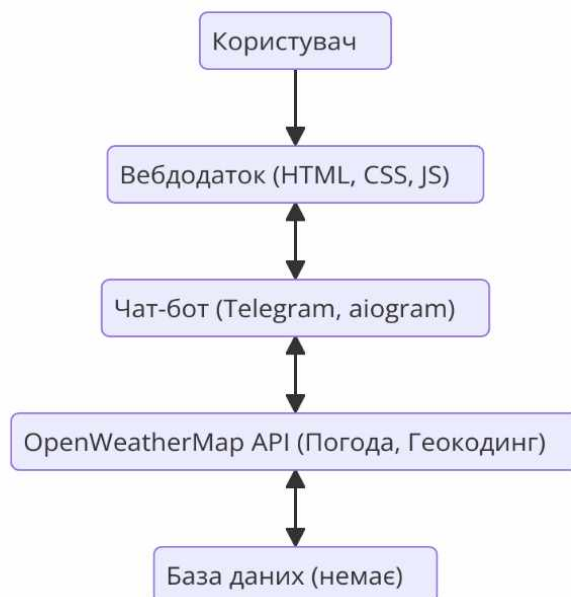


Рис. 2.2. Загальна архітектура системи

Для створення вебдодатку було обрано стандартні та добре перевірені технології веброзробки: HTML, CSS та JavaScript. HTML (HyperText Markup Language) використовувався для створення структури вебсторінок, тобто для розміщення всіх елементів інтерфейсу, таких як заголовки, текстові блоки, зображення, кнопки тощо. CSS (Cascading Style Sheets) застосовувався для стилізації вебсторінок, тобто для надання їм привабливого зовнішнього вигляду, а також для забезпечення адаптивності, щоб вебдодаток коректно відображався на різних пристроях з різними розмірами екранів[31]. JavaScript використовувався для реалізації всієї логіки роботи вебдодатку: обробки подій (наприклад, кліків на кнопки, введення тексту в поле пошуку), взаємодії з OpenWeatherMap API для отримання даних про погоду, динамічного оновлення контенту на сторінці без її перезавантаження, а також для збереження налаштувань користувача в `localStorage`.

Обрана архітектура SPA для вебдодатку дозволила створити динамічний та відзивчивий користувацький інтерфейс. Відсутність необхідності

перезавантажувати всю сторінку при кожній дії користувача чи оновленні даних значно покращує досвід взаємодії, роблячи його більш схожим на роботу з нативним десктопним або мобільним додатком. Реалізація всієї логіки на стороні клієнта за допомогою JavaScript означає, що основне обчислювальне навантаження припадає на пристрій користувача. Це спрощує серверну інфраструктуру (фактично, вона зводиться до хостингу статичних файлів), але водночас вимагає ретельної оптимізації JavaScript-коду для забезпечення належної продуктивності, особливо на менш потужних пристроях. Взаємодія з зовнішнім OpenWeatherMap API також відбувається безпосередньо з браузера користувача, що потребує уваги до обробки можливих помилок мережі та асинхронних запитів[32].

Паралельно з вебдодатком розроблявся чат-бот для платформи Telegram, який реалізує інший підхід до архітектури та взаємодії з користувачем. На відміну від клієнт-орієнтованого вебдодатку, чат-бот функціонує на основі серверного скрипта, написаного мовою Python з використанням бібліотеки aiogram. Цей скрипт постійно працює на сервері, приймає повідомлення від користувачів через Telegram Bot API, обробляє їх, взаємодіє з OpenWeatherMap API для отримання метеорологічних даних і надсилає відповіді назад у чат Telegram. Вибір Python та aiogram був зумовлений високою продуктивністю мови, великою кількістю готових бібліотек для роботи з веб-API та асинхронними операціями, а також зручністю та гнучкістю самої бібліотеки aiogram. Такий серверний підхід дозволяє реалізувати більш складну логіку обробки запитів та потенційно інтегруватися з іншими серверними системами, але потребує наявності та підтримки серверної інфраструктури.

Незважаючи на архітектурні відмінності, обидва компоненти – вебдодаток та чат-бот – спроектовані для досягнення спільної мети: надання користувачам актуальної та точної інформації про погоду. Ключовим елементом, що забезпечує консистентність даних між обома платформами, є використання єдиного джерела метеорологічної інформації – OpenWeatherMap API. Як вебдодаток, так і чат-бот звертаються до одних і тих же ендпоінтів API

(/weather, /forecast, /geo/1.0/direct, /geo/1.0/reverse) для отримання відповідних даних. Це гарантує, що користувач отримає однакову інформацію про погоду незалежно від того, який канал доступу він обере. Інтеграція між компонентами на даному етапі реалізована односпрямовано: чат-бот надає користувачеві можливість перейти на вебдодаток для отримання більш детального прогнозу або доступу до розширених налаштувань. Пряма синхронізація стану (наприклад, збережених локацій чи налаштувань) між вебдодатком (localStorage) та чат-ботом (пам'ять сервера) не реалізована через їхню різну природу та архітектуру. Отже, розроблена система поєднує два різних підходи до надання інформації, кожен зі своїми технологічними особливостями та перевагами для користувача (табл. 2.2). Вебдодаток пропонує багатий візуальний інтерфейс та детальну інформацію, тоді як чат-бот забезпечує швидкість та зручність доступу через популярний месенджер[33].

Таблиця 2.2

### Характеристики розроблених компонентів системи «Meteovision»

| Характеристика      | Вебдодаток «Meteovision»                                                                        | Чат-бот «MeteovisionBot»                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Платформа           | Веб (браузер)                                                                                   | Telegram                                                                                                 |
| Технології          | HTML, CSS, JavaScript                                                                           | Python, aiogram                                                                                          |
| Архітектура         | SPA (клієнтська логіка, без бекенду)                                                            | Серверний скрипт, взаємодія з Telegram API, асинхрон.                                                    |
| Джерело даних (API) | OpenWeatherMap API, Browser Geolocation API                                                     | OpenWeatherMap API, Telegram Bot API                                                                     |
| Основні функції     | Поточна погода, 5-дн. прогноз (детальний), пошук, геолокація, налаштування (одиниці, тема, час) | Поточна погода, 5-дн. прогноз (основний), пошук, геолокація, налаштування (одиниці), керування локаціями |
| Інтеграція          | Непряма (спільне джерело даних)                                                                 | Перехід на вебдодаток (кнопка "Web App")                                                                 |
| Збереження даних    | Налаштування: локально (localStorage)                                                           | Налаштування, локації: в пам'яті (сесійно)                                                               |

*Джерело: складено автором.*

Вибір цих технологій був зумовлений кількома факторами. По-перше, HTML, CSS та JavaScript є стандартними технологіями веброзробки, які

підтримуються всіма сучасними браузерами[34]. По-друге, ці технології є відносно простими у вивченні та використанні. По-третє, для JavaScript існує велика кількість бібліотек та фреймворків, які спрощують розробку та дозволяють створювати складні та інтерактивні вебдодатки. Оскільки вебдодаток не потребує складних обчислень на стороні сервера, було прийнято рішення відмовитися від використання бекенду. Це значно спростило розробку, розгортання та підтримку вебдодатку, а також зменшило витрати на хостинг. Замість бекенду вся логіка обробки даних та взаємодії з API була реалізована на стороні клієнта, тобто безпосередньо в браузері користувача.

При розробці інтерфейсу користувача особливу увагу було приділено зручності та інтуїтивності. Було розроблено адаптивний дизайн, який забезпечує коректне відображення та зручне використання вебдодатку на різних пристроях, включаючи комп'ютери, планшети та смартфони. Дизайн виконано в сучасному мінімалістичному стилі, з використанням приємної кольорової гами та чітких шрифтів. Для створення семантичної та доступної розмітки було використано HTML5-теги, такі як `<header>`, `<main>`, `<section>`, `<article>`, `<footer>` та інші. Це покращує структуру документа та полегшує його розуміння як для браузерів, так і для пошукових систем[35].

Одним з ключових аспектів розробки вебдодатку було підключення до OpenWeatherMap API для отримання даних про погоду. Для взаємодії з API використовувалася JavaScript-функція `fetch`, яка дозволяє надсилати асинхронні HTTP-запити до сервера та отримувати відповіді. Було використано такі ендпоінти (кінцеві точки) API: `/weather` – для отримання інформації про поточну погоду, `/forecast` – для отримання прогнозу погоди на 5 днів, `/geo/1.0/direct` - для прямого геокодування (отримання координат за назвою міста) та `/geo/1.0/reverse` - для зворотного геокодування (отримання назви міста за координатами). Отримані від API дані обробляються за допомогою JavaScript: з JSON-формату вони перетворюються на об'єкти, з яких потім витягуються необхідні значення (температура, вологість, швидкість вітру тощо) і відображаються на сторінці вебдодатку.

Основний функціонал вебдодатку включає пошук міста, відображення поточної погоди та прогнозу, визначення місцезнаходження користувача (геолокацію) та налаштування. Пошук за назвою міста реалізовано з використанням OpenWeatherMap Geocoding API. Користувач вводить назву міста в поле пошуку, після чого вебдодаток надсилає запит до API, отримує список можливих варіантів (якщо їх декілька) і відображає його у вигляді випадаючого списку. Користувач може вибрати потрібний варіант зі списку, після чого вебдодаток отримує дані про погоду для цього міста і відображає їх на сторінці. Відображення поточної погоди та прогнозу на 5 днів реалізовано з використанням даних, отриманих від OpenWeatherMap API. Дані оновлюються при зміні міста, при зміні одиниць вимірювання, а також при завантаженні сторінки. Функція геолокації дозволяє визначити місцезнаходження користувача за допомогою Geolocation API браузера (за умови, що користувач надав дозвіл на доступ до свого місцезнаходження). Після визначення координат вебдодаток автоматично отримує дані про погоду для цього місця. Налаштування вебдодатку дозволяють користувачеві змінювати одиниці вимірювання (Цельсій або Фаренгейт), вибирати тему оформлення (світлу або темну) та формат часу (12- або 24-годинний). Всі налаштування зберігаються в localStorage браузера, що забезпечує їх збереження між сесіями.

Для забезпечення якості розробленого вебдодатку було проведено ретельне тестування, яке включало кілька етапів. На етапі юніт-тестування перевірялася коректність роботи окремих функцій JavaScript-коду. Для цього було написано спеціальні тести, які перевіряли, чи правильно функції обробляють вхідні дані та повертають очікувані результати. Інтеграційне тестування було спрямоване на перевірку взаємодії різних компонентів вебдодатку, зокрема, взаємодії з OpenWeatherMap API. Перевірялося, чи правильно вебдодаток надсилає запити до API, чи отримує коректні відповіді та чи правильно обробляє отримані дані. Функціональне тестування передбачало перевірку роботи всього функціоналу вебдодатку в комплексі: пошуку міста, відображення поточної погоди та прогнозу, роботи геолокації, а також

налаштувань. Було проведено тестування на різних пристроях (комп'ютерах, планшетах, смартфонах) та в різних браузерях, щоб переконатися, що вебдодаток коректно відображається та працює на всіх платформах[36].

Розробка чат-бота для Telegram відбувалася паралельно з розробкою вебдодатку. Telegram було обрано як платформу для чат-бота завдяки його популярності, широкій аудиторії та наявності зручного API для розробників. Для створення чат-бота використовувалася мова програмування Python та сучасна асинхронна бібліотека aiogram, яка значно спрощує процес розробки. Aiogram надає зручні інструменти для обробки повідомлень, створення інтерактивних клавіатур, управління станами (наприклад, для реалізації покрокового введення даних) та асинхронного виконання запитів, що дозволяє боту одночасно обробляти велику кількість запитів від різних користувачів.

Логіка роботи чат-бота побудована на обробці команд та повідомлень від користувача. Бот підтримує такі команди: /start – команда, яка надсилається боту при першому запуску; вона виводить вітальне повідомлення та головне меню бота, /weather – команда для запиту інформації про поточну погоду; користувач може ввести назву міста після команди або надіслати свою геолокацію, /forecast – команда для запиту прогнозу погоди на 5 днів, /locations – команда, яка надає змогу користувачу керувати власним списком обраних локацій, /settings – команда для доступу до налаштувань бота[37].

При отриманні повідомлення від користувача бот спочатку перевіряє, чи є це повідомлення командою. Якщо це команда, то бот виконує відповідну дію (наприклад, виводить головне меню при команді /start або запитує назву міста при команді /weather). Якщо повідомлення не є командою, то бот аналізує його вміст залежно від поточного стану. Наприклад, якщо користувач надіслав команду /weather, а потім ввів назву міста, то бот сприйме це повідомлення як назву міста і надішле запит до OpenWeatherMap API для отримання даних про погоду в цьому місті.

Для взаємодії з OpenWeatherMap API бот використовує ті ж самі ендпоінти, що і вебдодаток: /weather для отримання поточної погоди, /forecast

для отримання прогнозу на 5 днів та `/geo/1.0/direct` і `/geo/1.0/reverse` для геокодування. Отримані дані обробляються та форматуються у вигляді текстового повідомлення, яке надсилається користувачеві.

Функціонал чат-бота включає пошук за назвою міста, відображення поточної погоди та прогнозу, визначення місцезнаходження користувача, налаштування одиниць вимірювання, а також інтеграцію з вебдодатком. Пошук міста працює аналогічно вебдодатку: користувач вводить назву міста, бот надсилає запит до OpenWeatherMap Geocoding API, отримує список можливих варіантів і пропонує користувачеві вибрати потрібний. Відображення поточної погоди та прогнозу також реалізовано з використанням даних, отриманих від OpenWeatherMap API. Для визначення місцезнаходження користувача бот може використовувати функцію Telegram, яка дозволяє користувачеві надіслати свою геолокацію. Налаштування дозволяють користувачеві змінювати одиниці вимірювання (Цельсій або Фаренгейт).

Інтеграція з вебдодатком реалізована за допомогою спеціальної кнопки, яка додається до повідомлень бота. При натисканні на цю кнопку користувач переходить на вебсторінку з детальною інформацією про погоду для відповідного міста. Це забезпечує зручний перехід між двома компонентами системи та дозволяє користувачеві отримати доступ до розширеного функціоналу вебдодатку безпосередньо з чат-бота.

Тестування чат-бота, як і тестування вебдодатку, було важливим етапом розробки. Воно включало перевірку правильності обробки всіх команд бота, перевірку коректності відповідей на різні запити користувачів (включаючи різні варіанти написання назв міст, використання різних мов, надсилання геолокації), перевірку роботи з різними локаціями (містами з різних країн), перевірку правильності роботи налаштувань (зміна одиниць вимірювання), а також перевірку коректності інтеграції з вебдодатком (перехід за посиланням).

Отже, було детально розглянуто процес розробки погодного вебдодатку та чат-боту, включаючи вибір технологій, опис архітектури системи, реалізацію інтерфейсу користувача, підключення до API, реалізацію функціоналу та

тестування. Розроблене рішення є інтегрованою системою, що складається з двох взаємопов'язаних компонентів: вебдодатку та чат-бота у Telegram. Вебдодаток надає користувачам зручний інтерфейс для перегляду детальної інформації про погоду, прогноз на 5 днів, а також дозволяє здійснювати пошук за назвою міста, визначати місцезнаходження користувача та налаштовувати параметри відображення. Чат-бот забезпечує швидкий доступ до основної інформації про погоду безпосередньо в месенджері Telegram, що є зручним для багатьох користувачів. Розроблена система є крос-платформною, тобто доступною з різних пристроїв (комп'ютерів, планшетів, смартфонів), а також персоналізованою, тобто дозволяє користувачам налаштовувати одиниці вимірювання та інші параметри. Використання OpenWeatherMap API забезпечує високу точність та актуальність даних про погоду. Розроблені компоненти системи (вебдодаток і чат-бот) були ретельно протестовані, що гарантує їхню стабільну роботу та відповідність заявленому функціоналу.

### РОЗДІЛ 3. ТЕСТУВАННЯ ТА ОЦІНКА РОЗРОБЛЕНИХ РІШЕНЬ

Тестування дозволяє перевірити відповідність реалізованого функціоналу початковим вимогам, виявити та усунути потенційні помилки. Оцінка розроблених рішень дозволяє визначити сильні та слабкі сторони створеної системи та намітити шляхи її подальшого вдосконалення.

Основна мета тестування полягала у верифікації наступних аспектів: коректність роботи функціоналу, зручність та інтуїтивність користувацьких інтерфейсів вебдодатку та чат-бота тощо[38].

Функціональне тестування вебдодатку «Meteovision» охоплювало перевірку всіх реалізованих можливостей. Тестувалася функція пошуку міста: перевірялося введення назви міста, коректність відображення списку підказок (автозаповнення), вибір міста зі списку та подальше відображення погоди для обраної локації. Було перевірено роботу пошуку з різними назвами міст, що мають однакові назви в різних країнах, а також обробку некоректних запитів. Наступним кроком було тестування функції геолокації: перевірялася реакція на натискання кнопки «Location», запит дозволу на доступ до місцеперебування в браузері та автоматичне відображення погоди для поточної локації після надання дозволу. Особливу увагу було приділено перевірці коректності відображення погодних даних: поточної температури, відчутної температури («Real Feel»), вологості, швидкості вітру, атмосферного тиску, часу сходу та заходу сонця, а також ймовірності опадів. Перевірялося правильне форматування даних відповідно до обраних користувачем одиниць вимірювання (Цельсій/Фаренгейт). Також тестувався блок 5-денного прогнозу: коректність відображення прогнозу для кожного з п'яти днів, а також можливість перегляду детального погодинного прогнозу при виборі конкретного дня. Було перевірено роботу панелі налаштувань: зміна одиниць вимірювання температури, вибір світлої або темної теми оформлення та зміна формату часу. Загальний вигляд інтерфейсу вебдодатку з відображеними елементами представлено на рис. 3.1.

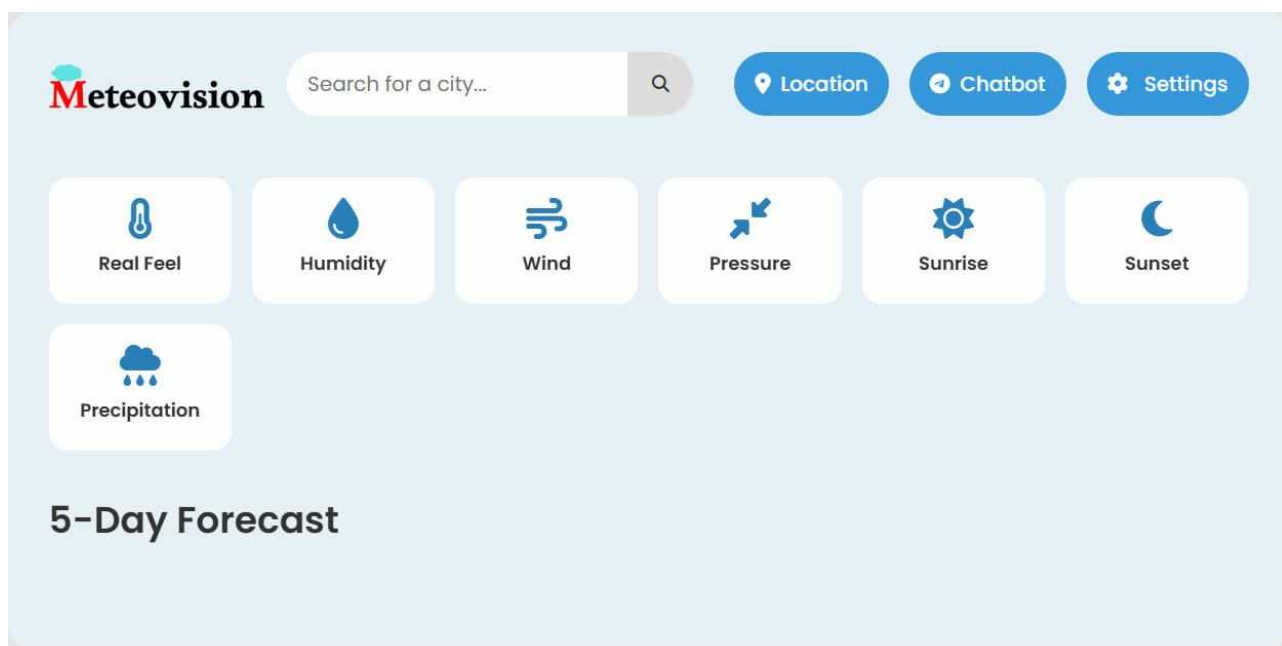


Рис. 3.1. Інтерфейс погодного вебдодатку «Meteovision»

Функціональне тестування чат-бота «MeteovisionBot» у Telegram також включало перевірку всіх команд та сценаріїв взаємодії[39]. Тестувалася команда /start, яка повинна виводити вітальне повідомлення та головну клавіатуру з основними опціями («Weather», «Forecast», «Locations», «Settings»). Перевірялася робота кнопок головного меню. При натисканні «Weather» та «Forecast» бот повинен був запитати локацію або запропонувати обрати зі збережених. Тестувалося введення назви міста текстом, відображення поточної погоди або прогнозу у відповідь тощо. Також перевірялася можливість надсилення боту геолокації через стандартну функцію Telegram та відображення погоди для поточного місця. Окремо тестувався функціонал керування локаціями (кнопка «Locations»): додавання нової локації, перегляд списку збережених локацій (з обмеженням до 10), видалення окремої локації та видалення всіх локацій. Перевірялася робота налаштувань (кнопка «Settings»): вибір одиниць вимірювання температури (Цельсій/Фаренгейт) та швидкості вітру (м/с, км/год) через відповідні клавіатури, а також коректне відображення даних після зміни налаштувань. Тестувалася кнопка «Web App», яка повинна генерувати та надавати користувачеві посилання на вебдодаток «Meteovision».

Результати тестування підтвердили коректну роботу всіх заявлених команд та функцій чат-бота. Приклад взаємодії з чат-ботом наведено на рис. 3.2.



Рис. 3.2. Інтерфейс взаємодії з чат-ботом «MeteovisionBot» у Telegram

Оцінка зручності використання (юзабіліті) проводилася на основі аналізу інтерфейсів та логіки взаємодії. Для вебдодатку оцінювалася простота навігації, логічність розташування елементів керування (пошук, кнопки геолокації та налаштувань), зрозумілість представлення погодніх даних (іконки, числові значення, текстові описи). Дизайн вебдодатку оцінювався як сучасний, мінімалістичний та інтуїтивно зрозумілий. Для чат-бота оцінювалася зрозумілість команд, логічність діалогів, зручність використання клавіатур для швидкого доступу до функцій та налаштувань, а також лаконічність та інформативність повідомлень, що надсилаються ботом. Інтерфейси обох компонентів були визнані достатньо зручними та простими для освоєння користувачами без спеціальної підготовки.

Базове тестування продуктивності та сумісності показало гарні результати. Вебдодаток продемонстрував прийнятний час завантаження та

відгуку на дії користувача в сучасних веббраузерах (Google Chrome, Microsoft Edge тощо) як на настільних комп'ютерах, так і на мобільних пристроях. Час отримання даних від OpenWeatherMap API залежав від швидкості інтернет-з'єднання. Чат-бот демонстрував швидку реакцію на команди та повідомлення користувачів, що забезпечується асинхронною архітектурою бібліотеки aiogram та відносною простотою оброблюваних запитів.

Тестування підтвердило, що поставлені на початку роботи цілі та завдання були в цілому досягнуті. Створено функціональну інтегровану систему, яка надає користувачам доступ до актуальної інформації про погоду через два зручних канали: вебдодаток та чат-бот Telegram. Система використовує надійне джерело даних (OpenWeatherMap API), має інтуїтивно зрозумілі інтерфейси та реалізує основний необхідний функціонал, включаючи пошук, геолокацію, відображення поточної погоди та прогнозу, а також базові налаштування. Інтеграція між вебдодатком та чат-ботом дозволяє користувачам обирати найбільш зручний спосіб отримання інформації залежно від ситуації[40].

Водночас у процесі тестування та оцінки було виявлено деякі обмеження та потенційні напрямки для вдосконалення. По-перше, точність прогнозів повністю залежить від даних, що надаються OpenWeatherMap API. По-друге, вебдодаток, реалізований як SPA з логікою на клієнтській стороні, може мати обмеження продуктивності при роботі на дуже слабких пристроях або при обробці великих обсягів даних (наприклад, при реалізації більших візуалізацій). По-третє, поточна інтеграція між вебдодатком та чат-ботом є односпрямованою; відсутня синхронізація налаштувань та збережених локацій між платформами. По-четверте, відсутні деякі розширені функції, які є у конкурентних сервісів, наприклад, інтерактивні погодні карти безпосередньо в інтерфейсі, попередження про екстремальні погодні явища тощо.

Для узагальнення результатів тестування та оцінки розроблених рішень було складено таблицю (табл. 3.1), де відображено основні аспекти перевірки та отримані висновки.

Таблиця 3.1

## Узагальнені результати тестування та оцінки системи «Meteovision»

| Аспект тестування       | Метод                                                  | Висновки                                                                                             |
|-------------------------|--------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| Функції (Веб)           | Мануальне тестування, перевірка роботи з API.          | Основні функції вебдодатку (пошук, геолокація, погода, прогноз, налаштування) працюють коректно.     |
| Функції (Бот)           | Мануальне тестування, перевірка роботи з API.          | Усі команди та функції бота працюють коректно.                                                       |
| Інтеграція              | Мануальне тестування.                                  | Перехід з чат-бота на вебдодаток через кнопку "Web App" функціонує правильно.                        |
| Зручність використання  | Експертна оцінка інтерфейсів.                          | Інтерфейси інтуїтивні, прості у використанні, візуально привабливі.                                  |
| Продуктивність (базова) | Спостереження за часом відгуку.                        | Час завантаження/відповіді вебдодатку та бота прийнятний. API затримки мінімальні.                   |
| Сумісність (Веб)        | Тестування в різних браузерах (Chrome, Firefox, Edge). | Вебдодаток коректно працює в основних браузерах та адаптується до різних екранів.                    |
| Оцінка ефективності     | Аналіз відповідності цілям та завданням проєкту.       | Система ефективно надає погоду через веб/Telegram, забезпечує кросплатформність.                     |
| Виявлені обмеження      | Аналіз результатів тестування та архітектури.          | Залежність від OpenWeatherMap, потенційні проблеми SPA, відсутність синхронізації, обмежені функції. |

*Джерело: складено автором.*

Таким чином, проведений етап тестування та оцінки підтвердив успішність розробки прототипу інтегрованої системи «Meteovision». Обидва компоненти – вебдодаток та чат-бот – демонструють стабільну роботу, реалізують заявлений функціонал та забезпечують зручний доступ користувачів до метеорологічної інформації. Виявлені обмеження та потенційні напрямки вдосконалення можуть стати основою для подальшого розвитку проєкту.

Посилання на вебдодаток: <https://meteovision.netlify.app>.

Посилання на чат-бот: <https://t.me/MeteovisionBot>.

## ВИСНОВКИ

У кваліфікаційній роботі досягнуто мети – розроблено інтегровану систему «Meteovision» (вебдодаток та чат-бот «MeteovisionBot») для оперативного доступу та аналізу погодних даних. Актуальність дослідження зумовлена висхідною суспільною потребою у швидкій, зручній та вірогідній метеорологічній інформації, важливій для повсякденного життя та багатьох галузей економіки. Система пропонує користувачам гнучкий підхід, поєднуючи візуалізацію вебдодатку та оперативність чат-бота.

Було проведено теоретико-методологічний аналіз наукових джерел та наявних рішень. Систематизовано підходи до збору, обробки та візуалізації метеоданих, вивчено ринкові тенденції (включно з роллю чат-ботів та ШІ) та виявлено недостатність інтегрованих рішень, що підтвердило доцільність обраного напрямку дослідження. Обґрунтовано методика, обрано інструментарій (HTML, CSS, JavaScript для вебу; Python, aiogram для бота) та надійне джерело даних – OpenWeatherMap API.

На основі аналізу вимог (точність, зручність, швидкість, кросплатформність) спроектовано архітектуру та реалізовано інтегровану систему. Розроблено вебдодаток «Meteovision» за архітектурою SPA, що забезпечує інтерактивний користувацький досвід, та чат-бот «MeteovisionBot» для Telegram. Обидва компоненти використовують єдине джерело даних OpenWeatherMap API для консистентності інформації. Реалізовано ключові функції: відображення поточної погоди та 5-денного прогнозу, пошук за назвою міста, визначення геолокації, базові налаштування (одиниці, тема, формат часу для вебу; керування списком локацій для бота). Забезпечено односпрямовану інтеграцію – можливість переходу з чат-бота на вебдодаток.

Третій етап роботи, за завданнями передбачав тестування та оцінку розроблених рішень. Функціональне тестування підтвердило коректну роботу всіх заявлених функцій вебдодатку та чат-бота, їх взаємодії з API, обробки команд, виконання налаштувань та інтеграційного переходу. Оцінка зручності

використання показала високу інтуїтивність та простоту інтерфейсів. Базове тестування продуктивності та сумісності продемонструвало прийнятну швидкодію та коректну роботу вебдодатку в основних браузерах, що підтвердило працездатність розроблених прототипів.

Практичне значення отриманих результатів полягає у створенні функціональної інтегрованої системи «Meteovision» – реального інструменту для доступу до метеоінформації, корисного для особистих потреб, а також потенційно для комерційних організацій та державних установ, діяльність яких залежить від погодних умов. Наукова новизна визначається комплексним підходом до створення саме інтегрованої системи «вебдодаток + чат-бот» для надання погодних даних, що поєднує сучасні вебтехнології та інструменти розробки ботів для забезпечення гнучкої та інтерактивної взаємодії в реальному часі, відповідаючи сучасним вимогам до інформаційних систем.

На основі проведеного дослідження та тестування сформульовано рекомендації для подальшого вдосконалення системи: підвищення точності прогнозів (інтеграція додаткових джерел даних); оптимізація продуктивності вебдодатку (рефакторинг JavaScript-коду, використання сучасних фреймворків); впровадження механізму аутентифікації та спільної бази даних для повної синхронізації користувацького досвіду між компонентами; розширення функціоналу (інтерактивні погодні карти, сповіщення про небезпечні явища тощо); покращення алгоритмів обробки природної мови (NLP) чат-бота; забезпечення вебдоступності відповідно до стандартів WCAG.

Підсумовуючи, у ході виконання кваліфікаційної роботи було успішно вирішено всі поставлені завдання. Проведено аналіз теоретичних засад, спроектовано, реалізовано та протестовано інтегровану систему «Meteovision». Отримані результати мають практичну цінність, демонструють ефективність та перспективність інтегрованого підходу, а також визначають обґрунтовані шляхи подальшого розвитку та вдосконалення розробленої системи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інформаційна технологія зберігання та візуалізації даних гідрометеорологічного прогнозування на основі WRF-Україна / С. Я. Майстренко, Т. О. Донцов-Загреба, К. В. Хурцилава, М. І. Харчук, С. В. Грибков, І. В. Ковалець: Математичні машини і системи. Київ : ПІММС НАНУ, 2019. № 1. С. 56–67. (дата звернення: 31.03.2025).
2. Капраль О. Р., Велика М. Б. Роль чат-ботів в епоху діджиталізації. *Вісник Херсонського національного технічного університету*. 2022. Т. 3, № 82. С. 53–58. URL: <https://surl.li/ogazuј> (дата звернення: 31.03.2025).
3. Рудніченко М., Гришин С., Шибасєв Д., Петров І., Носов М. Проект інформаційної системи для збору, обробки та аналізу метеоданих. *Інформаційні технології та суспільство*. 2021. Вип. 2. С. 34–41. DOI: <https://doi.org/10.32689/maup.it.2021.2.4> (дата звернення: 31.03.2025).
4. Коцюбівська К. Концепція чатбота як інтелектуального діалогового помічника / К. Коцюбівська, О. Тимошенко, М. Гузій, В. Лисинюк: Цифрова платформа: інформаційні технології в соціокультурній сфері. 2023. Т. 6, № 2. С. 329–340. Режим доступу: [http://nbuv.gov.ua/UJRN/dpitsca\\_2023\\_6\\_2\\_8](http://nbuv.gov.ua/UJRN/dpitsca_2023_6_2_8) (дата звернення: 31.03.2025).
5. Беседовський, О. (2024). Теоретичні засади розробки ШІ чат-бот. *Measuring and computing devices in technological processes*, (3), 201–206. <https://doi.org/10.31891/2219-9365-2024-79-25> (дата звернення: 31.03.2025).
6. Підходи до створення інтелектуальних чат-ботів / І. О. Ушакова: Системи обробки інформації. 2019. Вип. 2. С. 76–83. Режим доступу: [http://nbuv.gov.ua/UJRN/soi\\_2019\\_2\\_12](http://nbuv.gov.ua/UJRN/soi_2019_2_12) (дата звернення: 31.03.2025).
7. Kobets, V. M., & Kozlovskiy, K. H. (2022). Application of chat bots for personalized financial advice. *Вісник сучасних інформаційних технологій*, 5(3), 229–242. <https://doi.org/10.15276/hait.05.2022.18> (дата звернення: 31.03.2025).

8. Clare M. C. A., Jamil O., Morcrette C. J. Combining distribution-based neural networks to predict weather forecast probabilities. *Quarterly journal of the royal meteorological society*. 2021. Vol. 147, no. 741. P. 4337–4357. URL: <https://doi.org/10.1002/qj.4180> (date of access: 31.03.2025).

9. Міндер І. Ю. Розробка інтерактивного вебсайту для прогнозування погоди на основі метеоданих: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 122 Комп'ютерні науки / наук. кер. Л. В. Булатецька; Волинський національний університет імені Лесі Українки. Луцьк, 2024. 40 с. (дата звернення: 31.03.2025).

10. Розробка чат-боту моніторингу погодних умов для месенджера Telegram. Шаповал Р.О. Кафедра ІТ: Дубляни, Львівський НУП, 2024. Кваліфікаційна робота: 58 с. (дата звернення: 31.03.2025).

11. Nalyvaiko, O., & Maliutina, A. (2021). Use of chat bots in the educational process of a higher education institution. *Scientific Notes of the Pedagogical Department*, (48), 117-122. <https://doi.org/10.26565/2074-8167-2021-48-14> (date of access: 31.03.2025).

12. Caldarini G., Jaf S., McGarry K. A literature survey of recent advances in chatbots. *Information*. 2022. Vol. 13, no. 1. P. 41. URL: <https://surl.li/tkslcr> (date of access: 31.03.2025).

13. Розробка чат-ботів: технології та інструменти. *Блог*. URL: <https://surl.li/ckasoe> (дата звернення: 31.03.2025).

14. Зуй, Д. О. Прогнозування погоди на основі методів машинного навчання та аналізу даних : магістерська дис.: 122 Комп'ютерні науки / Зуй Денис Олександрович. Київ, 2024. 94 с. (дата звернення: 31.03.2025).

15. Карти погоди, API. Rain Viewer. URL: <https://surl.li/abpphb> (дата звернення: 31.03.2025).

16. Трофімова В. У Рівному запустили чат-бот, який розкаже про погоду. *ОГО*. URL: <https://ogo.ua/articles/view/2021-02-14/113517.html> (дата звернення: 31.03.2025).

17. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI : станом на 18 січ. 2025 р. URL: <https://surl.li/tifktk> (дата звернення: 31.03.2025).

18. Сучасні компанії на ринку кліматичної техніки. *День за днем*. URL: <https://surl.lu/gwgplv> (дата звернення: 31.03.2025).

19. Економічна метеорологія: навчально-методичний комплекс / Шевченко О.Г., Сніжко С.І. Київ, 2023. 73 с (дата звернення: 31.03.2025).

20. Вибір домашньої метеостанції: на що звернути увагу. *Метеостанції*. URL: <https://surl.li/mprxla> (дата звернення: 31.03.2025).

21. Оцінювання прогнозів. e-Pidruchniki.com. URL: <https://surl.li/betive> (дата звернення: 31.03.2025).

22. C91 Moderní aspekty vědy: XLVII. Díl mezinárodní kolektivní monografie / Mezinárodní Ekonomický Institut s.r.o.. Česká republika: Mezinárodní Ekonomický Institut s.r.o., 2024. str. 461 (дата звернення: 31.03.2025).

23. Ravenstone K. Топ сервісів спостереження за погодними умовами. *IT Блог*. URL: <https://surl.li/qhcieu> (дата звернення: 31.03.2025).

24. Рейтинг сервісів прогнозу погоди в Україні. *Marketer*. URL: <https://surl.li/lrslej> (дата звернення: 31.03.2025).

25. Пальонна Т. А., Коваленко Д. В., Пепчук С. М. Залежність успіху розробки програмного забезпечення від якості командної роботи. *Збірник наукових праць ЧДТУ*. № 48. С. 74–82. URL: <https://surl.li/zzqhwd> (дата звернення: 31.03.2025).

26. Kuibida L. Погодний додаток на Vue.js. *Medium*. URL: <https://surl.li/blwsld> (дата звернення: 31.03.2025).

27. SPA у програмуванні: занурення у світ односторінкових додатків. *FoxmindEd*. URL: <https://foxminded.ua/spa-u-prohramuvanni/> (дата звернення: 31.03.2025).

28. Інтеграція хмарних сервісів у ваші веб-додатки: можливості та переваги. *REDSTONE*. URL: <https://surl.li/ojnnek> (дата звернення: 31.03.2025).

29. J P. Створюємо простий погодні бот для Telegram за кілька вечорів. *Java блог*. URL: <https://surl.li/qqysbd> (дата звернення: 31.03.2025).

30. Ovcharuk I., Horbenko O. Single Page Application for Weather Tracking App. *Digital Platform: Information Technologies in Sociocultural Sphere*. 2021. Vol. 4, no. 2. P. 127–135. URL: <https://doi.org/10.31866/2617-796x.4.2.2021.247490> (date of access: 31.03.2025).

31. Всеосяжний посібник з фронтенд-розробки. *javascript.org.ua*. URL: <https://surl.li/xtqdgmm> (дата звернення: 31.03.2025).

32. Побережний О. Що таке single page application?. *Asabix*. URL: <https://surl.li/pfdjfq> (дата звернення: 31.03.2025).

33. Актуальні питання розвитку інформаційних технологій: тези доповідей IV Всеукраїнської конференції молодих учених (Маріуполь, 24 листопада 2022 р.)/ ДВНЗ «ПДТУ». ДніпроМаріуполь: ПДТУ, 2022. 82 с. (дата звернення: 31.03.2025).

34. Готинчан Т.І. Основи веброзробки: HTML і CSS. Частина 1 : навчальний посібник. Чернівці : Чернівецький національний університет, 2023. 208 с. (дата звернення: 31.03.2025).

35. Hernandez I. Пояснення семантичних елементів HTML5. *DreamHost Blog*. URL: <https://surl.li/bzttkb> (дата звернення: 31.03.2025).

36. У чому відмінність unit тестування від інтеграційного тестування?. *vshanuvannia.zapisi*. URL: <https://surl.li/lkggsb> (дата звернення: 31.03.2025).

37. 80-та Студентська науково-технічна конференція: Секція «Публічне управління та адміністрування»: збірник тез доповідей. Львів: Видавництво Львівської політехніки, 2022. 1 електрон. опт. диск (DVD). (дата звернення: 31.03.2025).

38. Учасники проєктів Вікімедіа. Тестування програмного забезпечення. *Вікіпедія*. URL: <https://surl.li/vuqcln> (дата звернення: 09.04.2025).

39. Як протестувати чат-бота. *ChatCompose*. URL: <https://surl.li/bgdbkk> (дата звернення: 09.04.2025).

40. Матеріали X Міжнародної науково-технічної Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 24 листопада 2023 [Електронний ресурс]. К: НУХТ, 2023. 224 с. Режим доступу: <https://surli.cc/kvwreo> (дата звернення: 09.04.2025).

## ДОДАТКИ

## Додаток А

## Фрагмент HTML програмного коду вебдодатку

```

<div class="container">
  <header class="weather__header">
    <!-- Logo Button -->
    <button id="logo-button" class="logo-button" aria-label="Reload Page">
      
    </button>
    <form class="weather__search">
      <div class="input-wrapper">
        <input type="text" placeholder="Search for a city..." class="weather__searchform" data-
i18n="searchPlaceholder" autocomplete="off" />
        <button type="button" class="clear-button" id="clear-button"><i class="fas fa-
times"></i></button>
      </div>
      <ul class="suggestions-list"></ul>
      <button type="submit" class="weather__searchbutton"><i class="fas fa-search"></i></button>
    </form>
    <!-- Button Group -->
    <div class="header-buttons">
      <button id="geolocation-button" class="header-button">
        <i class="fas fa-map-marker-alt button-icon"></i> <span data-
i18n="location">Location</span>
      </button>
      <!-- ... інші кнопки ... -->
      <div class="settings-container">
        <button id="settings-button" class="header-button settings-button" aria-label="Settings">
          <i class="fas fa-gear button-icon"></i> <span class="settings-button-text" data-
i18n="settings">Settings</span>
        </button>
        <!-- ... випадаюче меню налаштувань ... -->
      </div>
    </div>
  </header>
  <main class="weather__body">
    <h1 class="weather__city"></h1>
    <div class="weather__datetime"></div>
    <div class="weather__forecast"></div>
    <div class="weather__icon"></div>
    <div class="weather__temperature"></div>
    <div class="weather__minmax"></div>
    <!-- ... детальна інформація (картки) ... -->
  </main>
  <section class="weather__forecast">
    <h2 data-i18n="5dayForecast">5-Day Forecast</h2>
    <div class="forecast__container"></div>
  </section>
</div>

```

## Фрагмент CSS програмного коду вебдодатку

```

/* Визначення змінних для світлої теми */
:root {
  --primary-color: #2980b9;
  --secondary-color: #27ae60;
  --background-color: #ededed;
  --container-color: #e6f1f7;
  --text-color: #333;
  --card-background: #fff;
  --button-background: #3498db;
  --button-hover: #2980b9;
  --shadow-color: rgba(0, 0, 0, 0.1);
  --transition-speed: 0.3s;
  /* ... інші змінні ... */
}

/* Визначення змінних для темної теми */
.dark-theme {
  --primary-color: #3498db;
  --secondary-color: #2ecc71;
  --background-color: #1a1a2e;
  --container-color: #16213e;
  --text-color: #e0e0e0;
  --card-background: #16213e;
  /* ... інші змінні ... */
}

/* Стили для основного контейнера додатку */
.container {
  background: var(--container-color);
  backdrop-filter: blur(10px);
  padding: 2rem;
  border-radius: 20px;
  box-shadow: 0 0 20px var(--shadow-color);
  width: 90%;
  max-width: 1000px;
  margin: 2rem auto;
  transition: background-color var(--transition-speed),
    box-shadow var(--transition-speed);
}

```

## Фрагмент JavaScript програмного коду вебдодатку

```

// API ключ та базові URL
const API_KEY = "64f60853740a1ee3ba20d0fb595c97d5"; // Ваш API ключ
const BASE_URL = "https://api.openweathermap.org/data/2.5";
const GEO_URL = "https://api.openweathermap.org/geo/1.0";

```

## Продовження додатка А

```

// Елементи DOM (вибірково)
const cityElement = document.querySelector(".weather__city");
const temperatureElement = document.querySelector(".weather__temperature");
const iconElement = document.querySelector(".weather__icon");
const searchForm = document.querySelector(".weather__search");
// Обробник події пошуку
searchForm.addEventListener("submit", handleSearch);
// Функція отримання погоди за координатами (приклад)
function getWeatherByCoords(lat, lon, fullLocationName = "") {
    const currentWeatherUrl =
`$ {BASE_URL}/weather?lat=${lat}&lon=${lon}&appid=${API_KEY}&units=${units}&lang=${
language}`;
    const forecastUrl =
`$ {BASE_URL}/forecast?lat=${lat}&lon=${lon}&appid=${API_KEY}&units=${units}&lang=${
language}`;
    Promise.all([
        fetch(currentWeatherUrl).then((res) => res.json()),
        fetch(forecastUrl).then((res) => res.json()),
    ])
    .then(([currentData, forecastData]) => {
        // ... збереження даних ...
        updateCurrentWeather(currentData, fullLocationName); // Оновлення поточних даних
        updateForecast(forecastData); // Оновлення прогнозу
        // ...
    })
    .catch((error) => {
        console.error("Error fetching weather data:", error);
    });
}
// Функція оновлення поточних даних погоди в інтерфейсі (фрагмент)
function updateCurrentWeather(data, fullLocationName = "", timezoneOffset = null,
selectedIndex = 0) {
    cityElement.textContent = fullLocationName || `${data.name}, ${data.sys.country}`;
    // ... оновлення дати/часу ...
    temperatureElement.textContent = `${Math.round(data.main.temp)}°${units === "metric"
? "C" : "F"}°`;
    iconElement.innerHTML =
        =
``;
    // ... оновлення інших показників ...
}
// Ініціалізація додатку
function initApp() {
    // ... завантаження налаштувань ...
    updateLanguage();
    getGeolocation(true); // Спроба отримати геолокацію при завантаженні
    // ...
}
initApp(); // Запуск ініціалізації

```

## Фрагмент програмного коду чат-бота

```

import logging
from telegram import Update, InlineKeyboardButton, InlineKeyboardMarkup
from telegram.ext import Application, CommandHandler, MessageHandler,
CallbackQueryHandler, ContextTypes, filters
import aiohttp
from datetime import datetime

# ... (пропущено константи, функції клавіатур, словники мов, тощо) ...

# Обробник натискань на inline-кнопки
async def handle_callback(update: Update, context: ContextTypes.DEFAULT_TYPE):
    query = update.callback_query
    user_id = update.effective_user.id
    data = query.data.split('_')
    action = data[0]

    await query.answer() # Повідомляємо Telegram, що запит оброблено

    if action == "new": # Якщо користувач хоче додати нове місто
        context.user_data['action'] = data[1] # Зберігаємо дію (weather/forecast)
        await query.message.reply_text(get_text(user_id, 'enter_city_name'))
        await query.message.delete() # Видаляємо попереднє повідомлення з кнопками
        return

    if action == "delete": # Якщо користувач видаляє локацію
        lat, lon = float(data[1]), float(data[2])
        # Видаляємо локацію зі списку користувача
        user_locations[user_id] = [
            loc for loc in user_locations.get(user_id, [])
            if not (abs(loc['lat'] - lat) < 0.0001 and abs(loc['lon'] - lon) < 0.0001)
        ]
        locations_text = format_location_list(user_locations.get(user_id, []), user_id)
        # Редагуємо повідомлення, показуючи оновлений список
        await query.message.edit_text(f'{get_text(user_id, 'location_deleted')}\n\n{locations_text}')
        return

    # Якщо дія - отримання погоди або прогнозу для обраної локації
    if action in ['weather', 'forecast']:
        lat, lon = float(data[1]), float(data[2])
        # Відновлюємо повну назву локації з callback_data
        display_name = '_'.join(data[3:])

        # Перевіряємо, чи ця локація вже збережена, і якщо ні - зберігаємо
        if user_id not in user_locations or not any(
            abs(loc['lat'] - lat) < 0.0001 and abs(loc['lon'] - lon) < 0.0001 for loc in
            user_locations.get(user_id, [])):
            # Перевірка на максимальну кількість локацій

```

## Продовження додатка Б

```

if user_id in user_locations and len(user_locations[user_id]) >= MAX_LOCATIONS:
    await query.message.edit_text(get_text(user_id, 'max_locations_reached'))
    return
await save_location(user_id, lat, lon, display_name) # Зберігаємо локацію

# Отримуємо налаштування користувача
units = user_settings.get(user_id, {}).get('units', 'metric')
wind_units = user_settings.get(user_id, {}).get('wind_units', 'ms')

if action == 'weather':
    # Отримуємо дані про поточну погоду
    weather_data = await get_weather_by_coords(lat, lon, units, wind_units)
    if weather_data:
        # Використовуємо назву, яку ввів користувач (якщо є), або назву з кнопки
        weather_data['name'] = context.user_data.get('input_city', display_name)
        # Форматуємо та відправляємо відповідь
        await query.message.edit_text(format_weather(weather_data, units, wind_units, user_id))
    else:
        await query.message.edit_text(get_text(user_id, 'failed_to_get_weather'))
else: # action == 'forecast'
    # Отримуємо дані прогнозу
    forecast_data = await get_forecast_by_coords(lat, lon, units, wind_units)
    if forecast_data:
        # Використовуємо назву, яку ввів користувач (якщо є), або назву з кнопки
        forecast_data['city']['name'] = context.user_data.get('input_city', display_name)
        # Форматуємо та відправляємо відповідь
        await query.message.edit_text(format_forecast(forecast_data, units, wind_units, user_id))
    else:
        await query.message.edit_text(get_text(user_id, 'failed_to_get_weather'))

# ... (пропущено функції отримання даних з API, форматування, main) ...

if __name__ == '__main__':
    # Створення та запуск додатку
    application = Application.builder().token(TELEGRAM_TOKEN).build()

    # Реєстрація обробників
    application.add_handler(CommandHandler("start", start))
    application.add_handler(MessageHandler(filters.LOCATION, handle_location))
    application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND,
    handle_message))
    application.add_handler(CallbackQueryHandler(handle_callback)) # Реєструємо обробник
    кнопок

    # Запуск бота
    application.run_polling()

```