

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційні управляючі системи
та технології»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Інтерактивна система прогнозування атмосферних явищ»

Виконала:

студентка IV курсу, групи ІС-12

Павленко Амалія Ігорівна _____

Керівник:

доцент кафедри ІСТ, к.ф.-м.н.,

Гавриленко Олена Валеріївна _____

Рецензент:

доцент кафедри ІСТ, к.т.н.,

Олійник Юрій Олександрович _____

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студентка _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

ЗАВДАННЯ

на дипломний проєкт студентці
Павленко Амалії Ігорівні

1. Тема проєкту «Інтерактивна система прогнозування атмосферних явищ», керівник проєкту Гавриленко Олена Валеріївна, к.т.н., доцент, затверджені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: 9 червня 2025 року.
3. Вихідні дані до проєкту: середовище розробки WebStorm, платформа Node.js, фреймворк Express.js, бібліотека React.js, API OpenWeather, менеджер локалізації i18next, сховище даних LocalStorage, середовище розгортання Render.
4. Зміст пояснювальної записки: проаналізувати предметну область та існуючі рішення, визначити функціональні й нефункціональні вимоги до системи, побудувати структурну схему та діаграму прецедентів, обґрунтувати вибір технологій для розробки, створити діаграму компонентів, ER-діаграму та діаграму розгортання, сформулювати математичну модель прогнозування, реалізувати клієнтську та серверну частини інформаційної системи, виконати тестування функціональності та інтерфейсу, розгорнути застосунок у хмарному середовищі.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): структурна схема системи, діаграма прецедентів, діаграма компонентів, ER-діаграма, діаграма розгортання.

6. Дата видачі завдання: 8 березня 2025 року.

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Узгодження теми та технічного завдання	08.03.2025	
2.	Опис предметної області	19.04.2025	
3.	Аналіз існуючих рішень	26.04.2025	
4.	Визначення вимог до системи	03.05.2025	
5.	Вибір технологій для розробки	10.05.2025	
6.	Розробка інформаційної системи	17.05.2025	
7.	Розробка інтерфейсу для вебзастосунку	27.05.2025	
8.	Розробка математичної частини	04.06.2025	
9.	Тестування розробленої системи	07.06.2025	

Студентка

Амалія ПАВЛЕНКО

Керівник

Олена ГАВРИЛЕНКО

АНОТАЦІЯ

Інтерактивна система прогнозування атмосферних явищ.

Проект містить 73 с. тексту, 12 рисунків, 9 таблиці, посилання на 15 літературні джерела, додатки та 5 конструкторських документів.

ІНТЕРАКТИВНА СИСТЕМА, ЛАНЦЮГИ МАРКОВА, ПРОГНОЗУВАННЯ ПОГОДИ, ЧАСОВІ РЯДИ, ЧИСЕЛЬНІ МЕТОДИ, OPENWEATHER API, WEB-ЗАСТОСУНОК.

Об'єктом розробки є веб-застосунок для прогнозування атмосферних явищ. Метою розробки є створення системи, що дозволяє користувачу отримувати погодний прогноз на основі історичних даних та реального часу, використовуючи сучасні математичні моделі та зовнішні API.

У дипломному проєкті розроблено інтерактивний веб-застосунок, який забезпечує короткострокове прогнозування погоди (до 7 днів) для міста Київ на основі історичних даних за останні 15 років. Для інших міст світу використовуються актуальні дані, отримані через OpenWeather API. Реалізовано математичну модель, що використовує три підходи: часові ряди, чисельні методи, ланцюги Маркова. Застосунок побудовано на стеку технологій React.js з підтримкою i18n для фронтенду, бекенд — на Node.js з використанням Express. Застосунок було задеплойовано для надання доступу кінцевим користувачам.

Отримані результати можуть бути використані для вдосконалення погодних сервісів та створення кліматичних віджетів у реальних застосунках.

SUMMARY

Interactive System for Atmospheric Phenomena Forecasting

The project includes 73 pages of text, 12 figures, 9 tables, references to 15 literary sources, appendices and 5 design documents.

INTERACTIVE SYSTEM, MARKOV CHAINS, WEATHER FORECASTING, TIME SERIES, NUMERICAL METHODS, OPENWEATHER API, WEB APPLICATION.

The object of the development is a web application for forecasting atmospheric phenomena.

The goal of the development is to create a system that enables users to receive weather forecasts based on historical data and real-time information, using modern mathematical models and external APIs.

In the diploma project, an interactive web application was developed to provide short-term weather forecasts (up to 7 days) for the city of Kyiv based on historical data from the past 15 years. For other cities worldwide, current data retrieved through the OpenWeather API is used. The mathematical model is based on four approaches: time series, numerical methods, Markov chains, and probabilistic methods.

The application was built using the React.js technology stack with i18n support for the frontend and Node.js with Express for the backend. The system was deployed to provide access to end users.

The obtained results can be used to improve weather services and to create climate widgets for real-world applications.

№ рядка	Формат	Позначення	Найменування	Кіл. аркушів	№ екз.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IC12.230БАК.005 ПЗ	Інтерактивна система	73		
6			прогнозування атмосферних явищ.			
7			Пояснювальна записка			
8	A3	IC12.230БАК.005 Д1	Інтерактивна система	1		
9			прогнозування атмосферних явищ.			
10			Структурна схема			
11	A3	IC12.230БАК.005 Д2	Інтерактивна система	1		
12			прогнозування атмосферних явищ.			
13			Діаграма прецедентів			
14	A3	IC12.230БАК.005 Д3	Інтерактивна система	1		
15			прогнозування атмосферних явищ.			
16			Діаграма компонентів			
17	A3	IC12.230БАК.005 Д4	Інтерактивна система	1		
18			прогнозування атмосферних явищ.			
19			ER-діаграма			
20	A3	IC12.230БАК.005 Д5	Інтерактивна система	1		
21			прогнозування атмосферних явищ.			
22			Діаграма розгортання			
23						
24						
25						
26						
27						
28						

				IC12.230БАК.005 ТП				
Зм.	Лист	№ докум.	Підпис					
Розробив	Павленко А.І.			Інтерактивна система прогнозування атмосферних явищ.		Літ.	Арк.	Аркушів
Перевірив	Гавриленко О.В.					Т	1	1
Затв.				Відомість дипломного проекту		КПІ ім. Ігоря Сікорського Група IC-12		

**Пояснювальна записка
до дипломного проєкту
на тему: «Інформаційна система прогнозування
атмосферних явищ»**

Київ – 2025 року

ЗМІСТ

ВСТУП	6
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Загальна характеристика предметної області	9
1.2 Об'єкт, предмет і мета дослідження	9
1.3 Завдання для дослідження.....	10
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	13
2.1 Сервіс AccuWeather.....	13
2.2 Сервіс Weather.com	14
2.3 Сервіс Meteoblue.....	15
2.3 Порівняльний аналіз вебсервісів прогнозування погоди.....	17
3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ	19
3.1 Функціональні вимоги.....	19
3.1.1 Інтерфейс користувача	19
3.2 Нефункціональні вимоги.....	20
3.3 Архітектура системи.....	20
3.4 Системні обмеження.....	21
3.5 Інформаційна структура	21
4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ.....	23
4.1 Мова програмування JavaScript	23
4.2 Платформа Node.js	23
4.3 Сервіси.....	24
4.4 Фреймворк React	24
4.5 Бібліотеки.....	24
4.6 Клієнт-серверна архітектура	25
5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	27
5.1 Загальна структура застосунку	27
5.1.1 Клієнтська частина.....	28

				ІС12.230БАК.005 ПЗ				
Зм.	Лист	№ докум.	Підпис					
Розробив	Павленко А.І.			Інтерактивна система прогнозування атмосферних явищ. Пояснювальна записка		Літ.	Арк.	Аркушів
Перевірив	Гавриленко О.В.					Т	2	73
Затв.						КПІ ім. Ігоря Сікорського Група ІС-12		

5.1.2	Серверна частина	29
5.1.3	Взаємодія між клієнтом і сервером.....	30
5.2	Функціональна модель системи	31
5.3	Структура масивів інформації	33
5.4	Передавання та обробка даних	35
5.5	Архітектура програмного забезпечення	37
5.6	Тестування інформаційної системи	40
5.6.1	Методологія тестування	40
5.6.2	Тестування коректного сценарію пошуку	41
5.6.3	Тестування некоректного вводу	42
5.6.4	Перемикання теми оформлення	42
5.6.5	Тестування локалізації інтерфейсу	44
5.6.6	Обробка збоїв зовнішнього API	45
5.6.7	Тестування адаптивності інтерфейсу.....	45
5.6.8	Тестування збереження налаштувань LocalStorage.....	46
6	МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	49
6.1	Змістовна постановка задачі	49
6.2	Математична постановка задачі	50
6.3	Обґрунтування методу розв'язання	52
6.4	Оцінка точності методів прогнозування.....	54
6.4.1	Температура.....	55
6.4.2	Середня, мінімальна та максимальна температура	57
6.4.3	Швидкість вітру.....	58
6.4.4	Вологість повітря	59
6.4.5	«Відчувається як».....	60
6.4.6	Погодні умови	61
6.4.7	Ймовірність опадів.....	62
6.4	Опис методів розв'язання.....	63
6.4.1	Чисельні методи	65
6.4.2	Часові ряди.....	67
6.4.3	Ланцюги Маркова	68
	ВИСНОВКИ.....	70

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface.

axios – Asynchronous JavaScript XML Object System.

CSS – Cascading Style Sheets.

DOM – Document Object Model.

HTML – HyperText Markup Language.

HTTP – HyperText Transfer Protocol.

HTTPS – HyperText Transfer Protocol Secure.

i18next – Internationalization next.

JSON – JavaScript Object Notation.

MAPE – Mean Absolute Percentage Error

Node.js – Node JavaScript.

React – Reactive JavaScript Library.

REST – Representational State Transfer.

UI – User Interface.

Vite – View Interactive Tooling Environment.

useEffect – React Hook: Use Effect.

useState – React Hook: Use State.

Web – Web Application.

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		5

ВСТУП

У сучасних умовах зміни клімату, глобального потепління, нестабільних погодних явищ та зростаючої залежності людини від природного середовища, питання оперативного та точного прогнозування атмосферних явищ набуває особливої актуальності. Прогноз погоди є важливим інструментом у різних галузях від аграрного сектора до транспорту, туризму, логістики та енергетики. Наявні загальнодоступні сервіси часто не забезпечують достатньої точності або не враховують специфіку локальних кліматичних умов. Тому виникає потреба у створенні інструменту, який дозволяє більш точно прогнозувати погодні зміни з урахуванням локальної специфіки та історичних даних.

Необхідність створення інтерактивної системи прогнозування атмосферних явищ обґрунтовується також широким попитом на такі рішення у вигляді вебзастосунків, що не потребують встановлення та доступні з будь-якого пристрою. Провідні метеорологічні сервіси, як-от AccuWeather, Weather.com чи Meteoblue, надають користувачам прогнози, але вони часто стикаються з низкою обмежень:

- закритість математичних моделей;
- неможливість перевірки джерел даних;
- нав'язлива реклама;
- необхідність встановлення мобільних додатків або платна підписка для розширеного функціоналу.

Така ситуація створює передумови для створення альтернативних, відкритих та гнучких інструментів прогнозування, зокрема у вигляді вебзастосунків.

Метою даного дипломного проекту є створення веборієнтованої інтерактивної системи прогнозування атмосферних явищ, яка забезпечує короткостроковий прогноз погоди для будь-якого міста світу на основі математичного моделювання та зовнішніх джерел даних. Особливу увагу приділено місту Київ. Для нього прогноз реалізується на основі історичних погодних даних за останні 15 років, що дозволяє створити точнішу модель. Для

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

інших міст прогноз ґрунтується на інформації, отриманій у реальному часі через OpenWeather API. Це дозволяє поєднати адаптивність і гнучкість сучасного вебсервісу з математичною обґрунтованістю результатів.

У межах проєкту вирішуються наступні завдання:

- аналіз наявних погодних сервісів;
- обґрунтування вибору математичних методів прогнозування (часові ряди, чисельні методи, ланцюги Маркова, ймовірнісні методи);
- створення математичної моделі для погодних параметрів;
- реалізація фронтенду та бекенду застосунку;
- підключення до зовнішнього API;
- проведення деплою вебзастосунку.

Розроблений вебзастосунок має інтуїтивний інтерфейс, адаптується під різні типи пристроїв, підтримує перемикання тем оформлення, багатомовність і швидкий доступ до прогнозу без авторизації. Особлива увага приділена доступності, прозорості та відсутності реклами, що робить систему зручною для широкого кола користувачів.

На етапі математичного забезпечення реалізовано прогнозування температури з використанням авторегресійної моделі другого порядку, методів статистичної частоти для ймовірності опадів, а також ланцюгів Маркова для умов погоди. Чисельні методи застосовувалися для обчислення похідних параметрів, а саме: температури "відчувається як", яка враховує вологість повітря.

Розроблений застосунок використовує стек технологій HTML, SCSS, React.js з підтримкою інтернаціоналізації через i18n, а також Node.js з Express на боці сервера. Застосунок було задеплойовано на хмарну платформу, що забезпечує доступ кінцевих користувачів у реальному часі та дозволяє масштабувати систему за потреби.

Результати роботи можуть бути використані в локальних метеосервісах, мобільних додатках, аналітичних панелях, погодних віджетах, а також у сферах, чутливих до змін погодних умов, а саме: у сільському господарстві, транспорті, міській логістиці тощо. Крім того, запропоноване рішення може бути основою для

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

подальших досліджень у сфері кліматичного прогнозування, візуалізації кліматичних ризиків або енергетичного планування.

Дипломний проєкт складається з наступних розділів: вступ, основні розділи, висновки, список використаних джерел із 15 найменувань, додатків. Графічна частина включає 5 рисунків формату А3. Загальний обсяг текстової частини становить 73 сторінки.

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальна характеристика предметної області

У контексті зростання попиту на цифрові сервіси, що забезпечують оперативний доступ до актуальної інформації, особливого значення набуває створення веборієнтованих систем прогнозування погоди. Такі системи мають вирішальне значення не лише для повсякденного користування, а й для підтримки прийняття рішень у низці галузей: авіації, сільському господарстві, енергетиці, логістиці, будівництві та інші.

Традиційні метеорологічні сервіси здебільшого побудовані на закритих алгоритмах і пропонують обмежений доступ до історичних даних, що ускладнює адаптацію прогнозів до локальних кліматичних умов. У цьому контексті особливої цінності набувають відкриті метеорологічні API, зокрема OpenWeather[1], що дозволяє інтегрувати як поточні, так і багаторічні погодні спостереження у власні системи прогнозування. Такий підхід відкриває можливість побудови персоналізованих моделей на основі достовірних даних, з урахуванням специфіки обраного регіону.

Предметна область охоплює процеси збору, попередньої обробки, математичного моделювання та відображення метеорологічної інформації в інтерактивному вебінтерфейсі. Основна увага зосереджується на реалізації гнучкого, адаптивного та масштабованого застосунку, що забезпечує прозорість обчислень і зручність у користуванні. Впровадження відкритих стандартів, автоматизація обчислювальних процесів та дотримання принципів модульності дозволяють створити універсальний інструмент, придатний як для освітніх та дослідницьких цілей, так і для побутового застосування.

1.2 Об'єкт, предмет і мета дослідження

Об'єктом дослідження є процес веборієнтованого прогнозування погодних умов із використанням відкритих джерел метеорологічної інформації.

					ІС12.230БАК.005 ПЗ	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

Предмет дослідження охоплює побудову та програмну реалізацію математичних моделей прогнозування основних погодних параметрів: температури, вологості, швидкості вітру, ймовірності опадів, типу погодних умов тощо. Для навчання моделей використовуються історичні дані за 15 років спостережень у Києві, а для перевірки актуальності використовується поточна інформація з OpenWeather API[2].

Метою дослідження є розробка веборієнтованої системи прогнозування атмосферних явищ, яка забезпечує:

- побудову прогнозу на основі достовірних відкритих даних;
- реалізацію прозорих та відтворюваних математичних обчислень;
- адаптацію до локальних погодних умов (на прикладі Києва);
- зручний інтерфейс для взаємодії з користувачем.

Система повинна демонструвати здатність автоматично запускати розрахунки від поточної дати, відображати результати для обраного міста з трьохгодинним кроком на кілька діб наперед, а також масштабуватись для інших геолокацій завдяки інтеграції з OpenWeather API.

1.3 Завдання для дослідження

Для досягнення поставленої мети необхідно реалізувати наступний комплекс взаємопов'язаних завдань:

– провести огляд існуючих сервісів прогнозування погоди (AccuWeather, Weather.com, Meteoblue тощо)[3], проаналізувавши їхні функціональні можливості, способи подання даних, підтримку локалізації, доступність історичної інформації та відкритість математичних моделей. Результати огляду мають слугувати основою для формулювання вимог до власної системи;

– зібрати та підготувати погодні дані для міста Київ за останні 15 років: виконати валідацію даних, усунути пропущені значення, уніфікувати формат представлення інформації. Це дозволить забезпечити однорідність вхідних даних для коректного навчання моделей;

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

– обґрунтувати та підібрати математичні методи для прогнозування кожного з параметрів; при цьому передбачити адаптивний підхід, за якого для кожного параметра обирається найбільш точний метод на основі порівняння результатів тестування;

– реалізувати математичне ядро мовою JavaScript, забезпечивши модульність та можливість інтеграції з зовнішніми API. Кожна функція повинна відповідати окремому параметру прогнозу, а обчислення повинно автоматично запускатися при відкритті сторінки;

– створити адаптивний інтерфейс користувача з підтримкою темного та світлого оформлення, локалізації (i18n), пошуку міст, відображення трьохгодинного прогнозу, а також короткого опису погодних умов у текстовому форматі;

– інтегрувати систему з OpenWeather API для отримання поточних погодних умов та координат населених пунктів, використовуючи HTTPS-запити та механізми кешування для мінімізації навантаження;

– забезпечити автоматичний запуск прогнозування від поточної дати без участі користувача;

– провести тестування всіх функціональних компонентів системи: коректність математичних розрахунків, коректність відображення інтерфейсу, швидкодія при великому обсязі даних. За результатами тестування виконати деплой готового застосунку на відкриту платформу.

Висновки до розділу 1

У межах першого розділу було сформовано концептуальні засади розробки веборієнтованої системи прогнозування атмосферних явищ. Розглянуто актуальність створення цифрових сервісів, які забезпечують доступ до метеорологічної інформації у зручному та персоналізованому форматі. Визначено, що особливу цінність мають системи, що базуються на відкритих джерелах даних, зокрема OpenWeather API, оскільки це дозволяє реалізувати повністю

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

контрольовану математичну модель з можливістю адаптації до локальних умов.

Уточнено об'єкт дослідження — процес прогнозування погоди на основі відкритих метеоданих, а також предмет — побудова математичних моделей на основі цих даних у вебзастосунку. Метою роботи є створення інтерактивної, гнучкої та прозорої системи, що забезпечує точне прогнозування погодних параметрів на базі історичної інформації та поточних показників. При цьому реалізовано адаптивний підхід до математичного моделювання: кожен параметр прогнозується тим методом, що показав найкращу точність за результатами тестування.

На завершення було визначено основні задачі дослідження, які охоплюють аналіз готових рішень, підготовку даних, обґрунтування вибору математичних методів, реалізацію моделі мовою JavaScript, створення користувацького інтерфейсу та тестування застосунку. Ці положення закладають фундамент для наступних етапів проектування, моделювання та програмної реалізації системи.

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

У сфері цифрових метеорологічних сервісів спостерігається активний розвиток вебзастосунків, що забезпечують доступ до прогнозів погоди в реальному часі. Серед найпоширеніших платформ вирізняються AccuWeather, Weather.com та Meteoblue. Ці сервіси охоплюють мільйони користувачів у різних країнах світу. Вони пропонують багатофункціональні інтерфейси, інформативні погодинні прогнози, інтеграцію з мобільними додатками, інфографіку, карти та інші засоби візуалізації.

Проте, попри розмаїття функціоналу, більшість цих сервісів мають низку обмежень: закритий доступ до математичних моделей прогнозування, обмежене використання історичних даних, відсутність кастомізації під локальні кліматичні умови та наявність реклами у безплатних версіях.

Детальне вивчення можливостей згаданих сервісів дозволяє сформувати уявлення про сучасні стандарти подання метеоданих та зрозуміти користувацькі очікування, а також виявити структурні обмеження, які можуть бути подолані за допомогою авторського рішення.

2.1 Сервіс AccuWeather

AccuWeather є одним із найбільш відомих метеорологічних сервісів, що охоплює понад 3 мільйони локацій у всьому світі. Платформа вирізняється інноваційною технологією MinuteCast, яка надає прогноз з точністю до хвилини на найближчі дві години. Інтерфейс сервісу адаптивний, містить інтерактивні графіки температури, вологості, швидкості вітру та інших параметрів, а також пропонує інтеграцію з мобільними пристроями. Сервіс також надає сповіщення про різкі зміни погоди, що особливо корисно для щоденного планування[4].

На рисунку 2.1 зображено фрагмент інтерфейсу вебзастосунку AccuWeather, де представлено основну інформацію про погодні умови, карту опадів у Європі та рекламний блок з правого боку екрана.

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		13

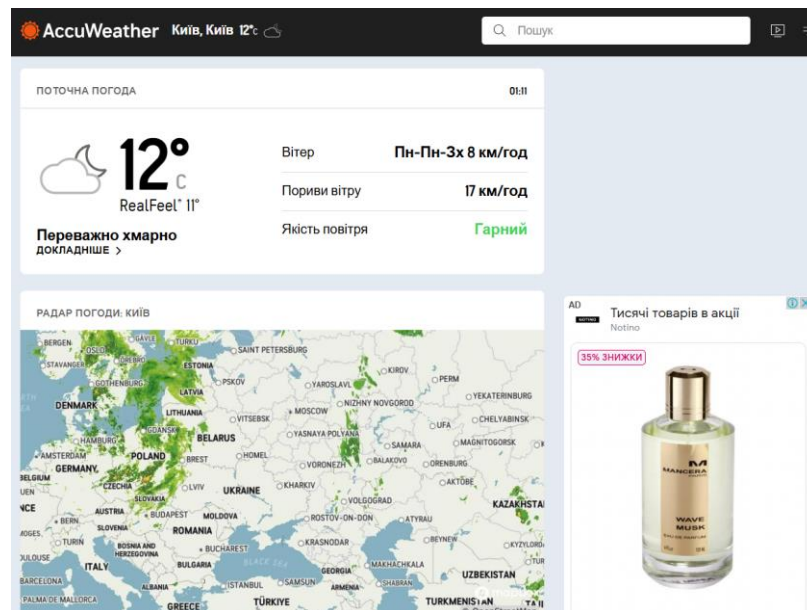


Рисунок 2.1 — Інтерфейс вебзастосунку AccuWeather

Основні характеристики сервісу:

- доступ до короткострокових і довгострокових прогнозів погоди;
- підтримка вебінтерфейсу та мобільних додатків;
- візуалізація даних у вигляді діаграм, карт і інтерактивних панелей;
- адаптивний інтерфейс із розширеною інформацією.

Недоліки:

- математичні моделі та принципи побудови прогнозів закриті;
- відсутність доступу до історичних даних через інтерфейс;
- безплатна версія має значне рекламне навантаження;
- прогноз адаптовано загальною.

2.2 Сервіс Weather.com

Сервіс Weather.com, який належить компанії The Weather Channel, є одним із найбільш відвідуваних погодних порталів у світі. Він пропонує прогнози, побудовані на основі даних із понад 250 тис. погодних станцій, супутників, радарів і власних моделей[5].

На рисунку 2.2 зображено головну сторінку вебзастосунку Weather.com, який містить основні погодні параметри, погодинний прогноз, а також додаткові

										Арк.
										14
Зм.	Лист	№ докум.	Підпис	Дата						

інформаційні віджети, рекламні модулі та мультимедійний контент.

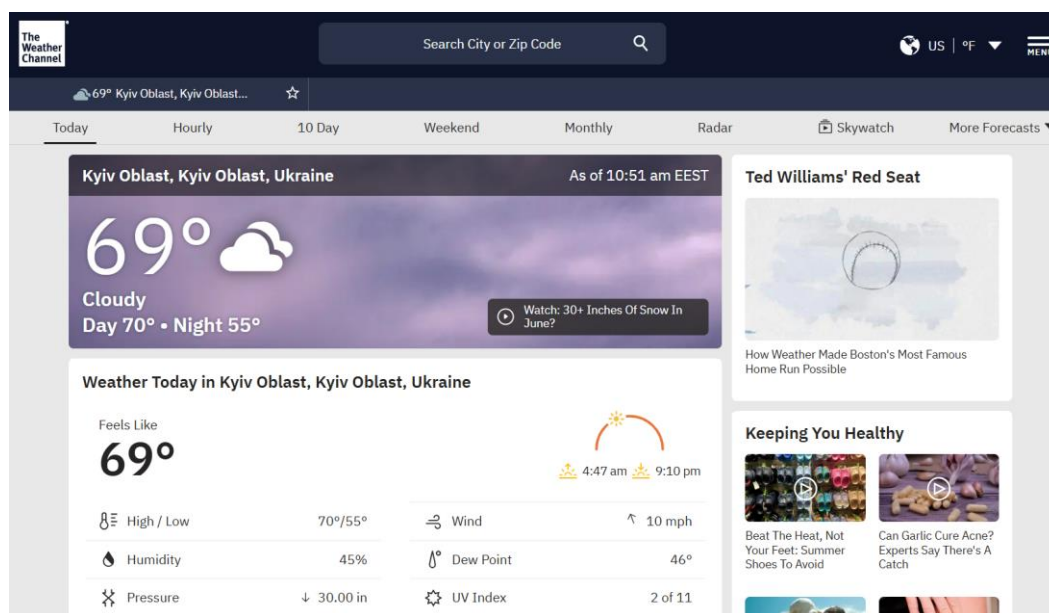


Рисунок 2.2 — Інтерфейс сайту Weather.com

Основні характеристики сервісу:

- формування прогнозу з часовим кроком до однієї години;
- підтримка мобільних застосунків і розумних пристроїв;
- тематичні прогнози за категоріями;
- наявність API Weather Company для корпоративного використання.

Недоліки:

- більшість функцій доступні лише за підпискою;
- історичні дані доступні лише через платні пакети;
- алгоритми прогнозування є повністю закритими;
- точність прогнозів залежить від щільності інфраструктури вимірювальних станцій у регіоні.

2.3 Сервіс Meteoblue

Meteoblue — це швейцарський онлайн-сервіс, що спеціалізується на точному науковому прогнозуванні погоди з використанням кількох фізичних і статистичних моделей. Сервіс орієнтований на дослідницьке, освітнє та професійне

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

– відсутність адаптації до мобільних платформ у повному обсязі.

2.3 Порівняльний аналіз вебсервісів прогнозування погоди

На основі детального аналізу трьох провідних погодних сервісів, а саме: AccuWeather, Weather.com та Meteoblue сформовано порівняльну таблицю 2.1 з оцінкою основних функціональних і технічних характеристик. Це дозволяє візуально зіставити їхні переваги та недоліки.

Таблиця 2.1 — Порівняння функціональних можливостей погодних сервісів

Ознака / Сервіс	AccuWeather	Weather.com	Meteoblue
Відкритість алгоритму	–	–	+ (частково)
Доступ до історичних даних	–	– (платно)	+
Наявність API	– (лише B2B)	+ (тільки платний)	–
Наявність реклами	+ (у безплатній версії)	+	–
Адаптація до локації	–	+ (частково)	+
Візуалізація даних	+ (графіки, карти)	+ (карти, погодинні панелі)	+ (діаграми, meteograms)
Підтримка мобільних пристроїв	+	+	– (частково)
Доступ без реєстрації	+	+	+ (частково)

Порівняльний аналіз показує, що жоден із розглянутих сервісів не забезпечує повністю відкритого та адаптивного прогнозування. Це підкреслює актуальність розробки власної гнучкої системи з відкритими алгоритмами та доступом до історичних даних. Крім того, більшість платформ або обмежують можливості локальної кастомізації, або вимагають платного доступу до розширених функцій.

Висновки до розділу 2

У межах другого розділу було проведено системний огляд сучасних цифрових метеорологічних сервісів, серед яких виокремлено три найпоширеніші платформи — AccuWeather, Weather.com та Meteoblue. Детально проаналізовано структуру їхніх інтерфейсів, функціональні можливості, специфіку візуалізації метеоданих, а також доступність до прогнозних моделей, архівів погоди та засобів інтеграції. Особливу увагу приділено технічним і комерційним обмеженням кожного з сервісів, що ускладнюють їх використання в дослідницьких і навчальних цілях.

У процесі аналізу було виявлено, що незважаючи на високу популярність і багатофункціональність зазначених платформ, усі вони демонструють спільні недоліки: закритість алгоритмів прогнозування, обмеження доступу до історичних даних, наявність рекламного контенту у безплатних версіях, а також залежність від платних пакетів API. Водночас деякі сервіси надають лише часткову підтримку мобільних платформ або демонструють складність у сприйнятті через перевантажений інтерфейс.

Зведена порівняльна таблиця дозволила систематизувати ключові функціональні характеристики сервісів і надати об'єктивне зіставлення їхніх можливостей. Це стало основою для формування технічних і функціональних вимог до власного вебзастосунку прогнозування погоди, який має забезпечити відкритість моделей, інтуїтивно зрозумілий інтерфейс, прозорий доступ до архівних даних, відсутність нав'язливої реклами, повну підтримку мобільних пристроїв та адаптацію під локальні умови.

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

3.1 Функціональні вимоги

Функціональні вимоги описують основні можливості системи, необхідні для реалізації її призначення.

До ключових функціональних можливостей належать:

- формування прогнозу погоди на 7 днів уперед із трьохгодинним кроком;
- побудова прогнозу для кожного з параметрів: температура(середня температура, максимальна та мінімальна температура), «відчувається як», вологість, швидкість вітру, погодні умови, ймовірність опадів;
- адаптивний вибір математичного методу для кожного параметра на основі попереднього тестування моделей;
- автоматичний запуск обчислень при відкритті сторінки;
- отримання метеоданих з OpenWeather API для обраного міста;
- відображення результатів у текстовому вигляді з деталізацією: трьохгодинні значення, середня температура, мінімальна та максимальна температура;
- можливість перемикання між сьогоднішнім і тижневим форматами перегляду;
- пошук іншого міста через текстове поле.

3.1.1 Інтерфейс користувача

До вимог до інтерфейсу належать:

- підтримка локалізації інтерфейсу (i18n)[7], зокрема української та англійської мов;
- перемикання між темною та світлою темами;
- автоматичне виявлення системної теми пристрою користувача;
- поле пошуку для вибору міста вручну;
- виведення прогнозу у структурованому текстовому форматі;

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		19

- адаптивність інтерфейсу для смартфонів, планшетів, десктопів;
- відображення повідомлень про помилки при недоступності сервера або некоректному введенні міста.

3.2 Нефункціональні вимоги

Система повинна відповідати таким нефункціональним характеристикам:

- простота використання: інтерфейс має бути інтуїтивно зрозумілим;
- адаптивність: підтримка різних типів пристроїв і роздільних здатностей екранів;
- продуктивність: час відповіді не повинен перевищувати 1 секунди для звичайного запиту;
- безпека: усі API-запити здійснюються через захищені протоколи;
- масштабованість: можливість додавання нових параметрів та API;
- незалежність: можливість публічного хостингу без серверної частини;
- надійність: система має стабільно функціонувати без збоїв протягом тривалого часу роботи.

3.3 Архітектура системи

Архітектура системи модульна, побудована на клієнт-серверному підході, з обробкою всіх обчислень на стороні клієнта.

Основні характеристики:

- модулі для API-запитів, математичних обчислень, інтерфейсу;
- розділення функціоналу на незалежні компоненти;
- повна автономність клієнтської частини;
- підтримка оновлення прогнозу від поточної дати автоматично.

Структурна схема відображає загальну логіку побудови вебзастосунку, взаємозв'язки між основними модулями, а також напрями передавання даних. Вона дозволяє зрозуміти архітектурну організацію системи та визначити ключові

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		20

функціональні блоки. Структурна схема системи включена до графічних матеріалів проекту: кресленик ІС12.230БАК.005 Д1.

3.4 Системні обмеження

До обмежень системи належать такі аспекти, що визначають межі її функціональності:

- єдине джерело даних: OpenWeather API;
- прогноз реалізовано лише для 7 днів уперед;
- вибір дати вручну недоступний. Прогноз стартує автоматично від поточної дати;
- результати виводяться лише у текстовому форматі;
- підтримуються лише міста, присутні у базі OpenWeather.

3.5 Інформаційна структура

Інформаційна структура системи включає всі ключові етапи роботи з даними, починаючи з їх отримання і завершуючи формуванням результату.

Вхідні дані:

- поточні метеодані, отримані за допомогою API-запитів до сервісу OpenWeather;
- історичні погодні спостереження за останні 15 років, що використовуються для навчання математичних моделей.

Внутрішня обробка:

- математичне моделювання погодних параметрів на основі обраних методів;
- вибір найбільш точного методу прогнозування для кожного параметра згідно з адаптивною стратегією, закладеною в логіку системи;
- обчислення окремих величин (наприклад, «відчувається як», ймовірність опадів) у відповідних JavaScript-модулях;
- динамічне формування прогнозу від поточної дати запуску системи.

Вихідні дані:

– прогнознi значення кожного параметра з трьохгодинним кроком на сім днів уперед;

– узагальнена текстова інформація для кожного інтервалу: температура, вологість, вітер, погодні умови, ймовірність опадів тощо.

Висновки до розділу 3

У третьому розділі було детально визначено всі необхідні вимоги для створення ефективної, зручної та масштабованої веборієнтованої системи прогнозування атмосферних явищ. Було сформульовано функціональні вимоги, які охоплюють основну логіку роботи застосунку: від отримання даних з OpenWeather API до побудови прогнозу на 7 днів із трьохгодинним інтервалом і представлення результатів у зрозумілому форматі для користувача. У структурі реалізації враховано адаптивний підхід, що забезпечує вибір найточнішого методу прогнозування для кожного параметра, виходячи з попереднього аналізу похибок.

Окрему увагу приділено вимогам до інтерфейсу, зокрема підтримці адаптивного відображення на різних пристроях, локалізації, перемиканню між світлою та темною темами, а також реалізації зручного пошуку міст. Уточнені нефункціональні характеристики, серед яких швидкодія, безпека, відсутність залежності від серверної частини та можливість подальшого розширення, підтверджують, що система розробляється з урахуванням реальних умов використання.

Описана архітектура клієнтської частини має модульну структуру, що забезпечує її автономність і простоту масштабування. Визначені також системні обмеження, які встановлюють межі її функціонування. Інформаційна структура розглядається як послідовність обробки даних від моменту надходження метеоінформації до формування фінального тексту прогнозу.

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		22

4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

На сьогодні існує багато технологій для створення вебзастосунків, проте комбінація JavaScript, React та Node.js із використанням Express[8] вважається одним із найпопулярніших рішень для створення динамічних, масштабованих і зручних у підтримці клієнт-серверних систем.

4.1 Мова програмування JavaScript

Основною мовою програмування, що використовується для реалізації як клієнтської, так і серверної частини системи, є JavaScript[9]. Це динамічна мова, яка добре підходить для створення інтерактивних вебзастосунків. Вона підтримується всіма сучасними браузерами, а також дозволяє виконувати обчислення на стороні клієнта, що значно підвищує швидкість роботи системи. JavaScript має велику спільноту та широкий набір готових рішень, що полегшує процес розробки та супроводу застосунку. Висока гнучкість цієї мови дає змогу реалізувати необхідні алгоритми прогнозування без використання додаткових інструментів програмування.

4.2 Платформа Node.js

Для створення серверної частини застосунку обрано платформу Node.js[10]. Це середовище дозволяє запускати JavaScript не тільки у браузері, а й на сервері. Завдяки цьому забезпечується повна сумісність логіки між клієнтом і сервером. Node.js підтримує асинхронне виконання запитів, що дозволяє швидко обробляти дані з OpenWeather API та надсилати результат у вигляді структурованого тексту до інтерфейсу користувача. Також ця платформа дозволяє легко налаштовувати обмін даними між модулями та спростити розгортання застосунку на хостингу. Для зручного запуску клієнтської частини використано збирач Vite, який пришвидшує компіляцію та покращує швидкодію інтерфейсу. Також передбачено можливість деплою на платформу Render, що забезпечує швидке розгортання проекту без

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23

потреби в окремій інфраструктурі.

4.3 Сервіси

Застосунок взаємодіє з двома основними сервісами. Перший — це OpenWeather API, що забезпечує отримання актуальних і історичних метеоданих для обраного міста. Він є відкритим сервісом з можливістю щоденного доступу до прогнозів, що робить його зручним для дослідницьких цілей. Другий — i18next, який відповідає за підтримку багатомовності в інтерфейсі. Завдяки йому користувач може перемикатися між українською та англійською мовами, а також отримувати переклади динамічного контенту відповідно до налаштувань свого пристрою. Обидва сервіси дозволяють забезпечити зручність використання системи для широкого кола користувачів.

4.4 Фреймворк React

Клієнтська частина системи створена за допомогою фреймворку React[11]. Він дає змогу будувати інтерфейс із незалежних компонентів, які легко комбінуються між собою. Такий підхід дозволяє організувати код зручно та зрозуміло, а також швидко вносити зміни до окремих частин застосунку без ризику порушити його загальну роботу. React підтримує оновлення елементів інтерфейсу без повного перезавантаження сторінки, що робить роботу з системою швидкою і плавною. Крім того, цей фреймворк добре підходить для створення адаптивних інтерфейсів, які коректно відображаються на смартфонах, планшетах і комп'ютерах.

4.5 Бібліотеки

Для реалізації окремих функцій у проєкті використано набір допоміжних бібліотек. Axios використовується для обміну даними між клієнтом і сервером, забезпечуючи стабільні та зручні запити до API. Sass застосовується для написання

					ІС12.230БАК.005 ПЗ	Арк.
						24
Зм.	Лист	№ докум.	Підпис	Дата		

стилів, дозволяючи будувати гнучку систему оформлення інтерфейсу. React Router реалізує навігацію між сторінками. React-slick дозволяє реалізувати прокручування блоків, зручне для мобільних пристроїв. 18next забезпечує локалізацію, а node-fetch — виконання запитів у серверній частині. Для математичних розрахунків, зокрема при побудові моделей прогнозування, застосовується бібліотека regression. Кожна з бібліотек інтегрується в загальну структуру застосунку, полегшуючи розробку та обслуговування.

4.6 Клієнт-серверна архітектура

Уся система побудована на принципі клієнт-серверної архітектури. Це означає, що інтерфейс користувача працює окремо від логіки обробки даних. Користувач взаємодіє з клієнтом, який надсилає запити до серверної частини. Сервер, у свою чергу, виконує обчислення та формує прогноз, після чого надсилає результат назад. Така структура дозволяє чітко розділити функціональні частини системи, забезпечити їх незалежну розробку та полегшити майбутнє оновлення. Для розміщення системи в інтернеті передбачено використання хостингового сервісу Render.

Висновки до розділу 4

У межах четвертого розділу було детально проаналізовано й обґрунтовано вибір технологій, що стали основою для реалізації веборієнтованої системи прогнозування атмосферних явищ. Застосування мови програмування JavaScript дало змогу забезпечити єдину технологічну базу для фронтенд- і бекенд-розробки, що спростило інтеграцію між клієнтською та серверною частинами системи.

Використання середовища Node.js для серверної логіки дало змогу реалізувати асинхронну обробку запитів, що критично важливо для взаємодії з зовнішніми API, зокрема OpenWeather. Це забезпечує швидку відповідь сервера та підвищує стійкість до можливих затримок з боку сторонніх сервісів. Застосування

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		25

фреймворку React на клієнтському боці дозволило реалізувати компонентно орієнтовану структуру інтерфейсу, що спрощує його підтримку та розвиток, забезпечує ефективну роботу з даними та створює адаптивне відображення для різних типів пристроїв.

У ході реалізації системи було також інтегровано ряд додаткових бібліотек: Axios — для надсилання HTTP-запитів; React Router — для організації маршрутизації сторінок; i18next — для реалізації багатомовного інтерфейсу; а також бібліотеки для математичних обчислень і обробки часових рядів.

Обрана архітектура клієнт-серверного типу забезпечує логічне розмежування між обчислювальною частиною та інтерфейсом користувача, що сприяє масштабованості проекту та дозволяє окремо розгортати й підтримувати кожен модуль системи. Розміщення застосунку на хмарній платформі Render забезпечує стабільність розгортання, можливість автоматичного оновлення, централізоване логування та моніторинг роботи.

Узагальнюючи, можна стверджувати, що обрані інструменти та середовища цілком відповідають вимогам до сучасної веборієнтованої системи, дозволяють ефективно реалізувати поставлену функціональність, зберігаючи при цьому високу гнучкість, зручність масштабування та перспективи подальшого розвитку застосунку.

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		26

5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

5.1 Загальна структура застосунку

Інтерактивна система прогнозування атмосферних явищ побудована на основі клієнт-серверної архітектури, яка забезпечує розділення логіки застосунку на дві автономні, проте взаємопов'язані частини: клієнтську (фронтенд) та серверну (бекенд). Така структура дає змогу організувати гнучку взаємодію між модулями, підвищити масштабованість, забезпечити ефективне супроводження програмного забезпечення, а також адаптацію до нових функцій у майбутньому. Клієнтська частина реалізована за допомогою бібліотеки React, що забезпечує створення адаптивного, інтерактивного та зрозумілого користувацького інтерфейсу. Серверна частина функціонує на основі Node.js та фреймворку Express, відповідає за приймання запитів від клієнта, обробку даних, звернення до зовнішнього API та реалізацію математичних моделей прогнозування. Для забезпечення ефективної роботи системи використано уніфікований стек технологій, що базується на мові JavaScript для обох частин застосунку. Компоненти інтерфейсу користувача структуровані у вигляді модулів React, а серверна логіка організована в окремих директоріях відповідно до функціонального призначення.

Клієнтська частина побудована за принципами компонентного підходу, що дозволяє забезпечити повторне використання елементів інтерфейсу та зменшити складність підтримки коду. Всі компоненти згруповані за функціональністю, що спрощує їхнє розширення та модифікацію. Наприклад, уся логіка взаємодії з сервером винесена в окремі модулі, що ізолювано від UI-компонентів, дозволяє протестувати запити окремо.

Для забезпечення адаптивності інтерфейсу застосовуються CSS-модулі, що гарантує незалежність стилів компонентів один від одного. Дизайн інтерфейсу створено з урахуванням принципів доступності: контрастність кольорів, підтримка клавіатурної навігації та можливість масштабування інтерфейсу.

З боку серверної частини також дотримано принципу єдиного призначення:

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		27

Назва файлу	Призначення
	запуск на порту
/routes/weatherRoutes.js	основний маршрут для запитів прогнозу
/controllers/weatherController.js	логіка обробки запитів, виклики до API та математичних функцій
/api/fetchCurrentWeather.js	запит до OpenWeather для поточних погодних даних
/api/fetchForecastData.js	запит до OpenWeather для 7-денного прогнозу з інтервалом у 3 години
/prediction/temperature.js	модуль прогнозування температури (часові ряди, нормалізація)
/prediction/feelsLike.js	модуль обчислення температури "відчувається як"
/prediction/conditions.js	модуль прогнозу погодних умов за ланцюгами Маркова
/prediction/humidity.js	модуль розрахунку вологості
/utils/helpers.js	допоміжні функції для обробки та форматування даних
.env	зберігання ключів API та конфігурацій середовища

5.1.3 Взаємодія між клієнтом і сервером

Взаємодія реалізована за принципом REST-архітектури. Клієнт ініціює HTTP-запит при введенні міста у формі SearchForm. Цей запит надсилається до маршруту /api/weather/forecast на сервері, де відбувається його обробка, звернення до OpenWeather API, математичне прогнозування та формування відповіді у форматі JSON. Отримані дані повертаються клієнту, де обробляються у компонентах ForecastTable та ForecastByDay для візуального представлення користувачу.

					ІС12.230БАК.005 ПЗ	Арк.
						30
Зм.	Лист	№ докум.	Підпис	Дата		

Діаграма компонентів демонструє структуру програмного забезпечення у вигляді взаємопов'язаних модулів. Вона деталізує логіку побудови клієнтської та серверної частин, описує модулі взаємодії з API, обробки даних та відображення результатів. Діаграма компонентів включена до графічних матеріалів проекту: кресленик ІС12.230БАК.005 ДЗ.

5.2 Функціональна модель системи

У рамках функціональної моделі веборієнтованої системи прогнозування атмосферних явищ визначено основні функції, які реалізує програмне забезпечення для забезпечення безперервного циклу обробки вхідних даних, формування прогнозів і їх відображення користувачеві. Функціональні можливості розподілені між клієнтською та серверною частинами згідно з архітектурною структурою, описаною у попередньому пункті.

Клієнтська частина виконує такі ключові функції: прийняття вхідного запиту від користувача у вигляді введеної назви міста;

- надсилання HTTP-запиту до серверної частини через REST-інтерфейс;
- отримання у відповідь структурованих погодних даних у форматі JSON;
- візуалізація даних у вигляді блоків поточної погоди, трьохгодинного прогнозу та денного прогнозу на 7 днів;
- реалізація перемикання мови інтерфейсу та теми оформлення.

Серверна частина реалізує такі функції:

- обробка вхідного HTTP-запиту та маршрутизація на відповідні контролери;
- звернення до OpenWeather API для отримання фактичних погодних даних і прогнозів;
- виконання математичного прогнозування параметрів погоди на основі історичних даних за допомогою окремих модулів;
- формування відповіді у форматі JSON відповідно до структури, очікуваної клієнтською частиною;
- забезпечення логування, обробки помилок і стабільності у разі збоїв

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		31

зовнішнього API.

Усі модулі прогнозування реалізують окремі математичні методи залежно від типу параметра: для температури використано моделі часових рядів, для погодних умов — імовірнісні ланцюги Маркова, для вологості та швидкості вітру — чисельні методи апроксимації. Такий розподіл дозволяє адаптувати прогнозування до характеру кожного метеорологічного показника, забезпечуючи точність і гнучкість.

Функціональна модель реалізована відповідно до REST-архітектури, що забезпечує чітку взаємодію клієнтської та серверної частин. Інтерфейс запитів узгоджено зі структурою відповідей API, що дозволяє забезпечити швидку обробку запитів та уникнути розсинхронізації при передачі даних. Асинхронна взаємодія через бібліотеку axios гарантує мінімальні затримки та плавну роботу інтерфейсу.

Важливою особливістю функціональної моделі є повна автоматизація процесу прогнозування. Від користувача потрібно лише ввести назву міста, після чого система виконує усі обчислення та відображає результати без необхідності ручного втручання.

З боку клієнта реалізовано базову валідацію введених даних, зокрема перевірку на порожній рядок. У разі помилок з боку зовнішнього API передбачено fallback-механізм — виведення користувачу повідомлення про помилку, що дозволяє уникнути збоїв та покращує стабільність застосунку.

Додатково у функціональній моделі передбачена можливість розширення функціональності без істотної зміни архітектури: наприклад, додавання обраних міст, розширення прогнозу на 10 днів, або інтеграція карт погодних явищ. Завдяки модульному підходу можливе масштабування системи без порушення її базової логіки.

Діаграма прецедентів моделює взаємодію користувача із системою через функціональні сценарії (use cases). Вона слугує основою для формування функціональних вимог, окреслює ролі, доступні користувачам, та межі системи. Діаграма прецедентів включена до графічних матеріалів проекту: кресленик IC12.230БАК.005 Д2.

					IC12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		32

Нижче наведено узагальнений приклад псевдокоду, що ілюструє логіку функціонального модуля серверної частини, який обробляє запит на отримання прогнозу погоди для введеного міста.

```
on GET request to "/weather/:city":
  extract cityName from request parameters
  try:
    construct API URL using cityName and API_KEY
    send HTTP GET request to OpenWeather API
  if response is not OK:
    return error with response status and message
  parse forecastData from API response
  extract currentDate from system clock
  filter forecastData to select only entries for currentDate
  construct responseObject with:
    cityInfo from forecastData
    todayForecast = filteredForecastEntries
  return responseObject as JSON to client
  catch error:
    log error to server console
    return HTTP 500 error with message "Internal server error"[12]
```

Такий підхід забезпечує ефективну обробку запиту, фільтрацію за поточною датою та повернення структурованих даних клієнту. У разі помилок виконується логування й надсилається повідомлення про несправність.

5.3 Структура масивів інформації

Структура масивів інформації в інтерактивній системі прогнозування атмосферних явищ визначає спосіб організації, зберігання та передавання даних між клієнтською та серверною частинами застосунку. Вся інформація передається у форматі JSON, що забезпечує зручність у серіалізації, читабельність і підтримку

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		33

сучасними вебтехнологіями.

Основні масиви та структури інформації формуються на стороні сервера після отримання даних від OpenWeather API та після проходження етапу математичного оброблення. Далі вони передаються на клієнт для відображення.

Основні типи масивів:

а) поточні погодні умови;

1) цей об'єкт містить оперативну інформацію про погодні умови на момент запиту, зокрема;

- temperature: поточна температура повітря;
- feels_like: температура «відчувається як»;
- weather_description: опис погодних умов (наприклад, «хмарно»);
- icon: ідентифікатор іконки погодних умов;
- humidity: рівень вологості;
- wind_speed: швидкість вітру.

б) поточні погодні умови;

2) масив об'єктів по кожному з наступних 7 днів, кожен з яких містить;

- date: дата прогнозу;
- temp_min: мінімальна прогнозована температура;
- temp_max: максимальна прогнозована температура;
- weather_description: опис погодних умов;
- pop: ймовірність опадів у вигляді десяткового значення;
- humidity: прогнозована вологість;
- wind_speed: середня швидкість вітру;
- feels_like: середнє значення «відчувається як».

в) трьохгодинний прогноз;

3) масив із 56 елементів (7 днів × 8 значень на добу), кожен з яких включає;

- time: час прогнозу (наприклад, 03:00, 06:00, ...);
- temperature: прогнозована температура;

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		34

- feels_like: прогнозована температура за відчуттям;
- weather_description: опис погодних умов;
- humidity: вологість;
- wind_speed: швидкість вітру;
- pop: ймовірність опадів.

г) геолокаційна інформація;

4) об'єкт, що містить дані про населений пункт;

- name: назва міста;
- country: країна;
- lat: широта;
- lon: довгота.

Кожна структура даних була оптимізована для полегшення подальшої візуалізації на інтерфейсі, де інформація відображається у відповідних компонентах: WeatherWidget, ForecastTable та ForecastByDay.

Завдяки єдиній структурі форматів JSON досягається повна сумісність між клієнтською та серверною частинами застосунку, що забезпечує надійність обміну даними та мінімізацію помилок під час відображення.

ER-діаграма відображає логічну структуру даних, що використовуються в системі. Вона показує основні сутності, їхні атрибути та зв'язки між об'єктами, такими як температурні записи, атмосферні параметри та погодні стани. Діаграма допомагає краще зрозуміти, як організовані масиви даних і як між ними встановлено взаємозв'язки. ER-діаграма включена до графічних матеріалів проекту: кресленик IC12.230БАК.005 Д4.

5.4 Передавання та обробка даних

Передавання та обробка даних у системі прогнозування атмосферних явищ базується на асинхронній взаємодії між клієнтською та серверною частинами через REST API. Комунікація реалізована з використанням протоколу HTTP та бібліотеки axios на стороні клієнта.

					IC12.230БАК.005 ПЗ	Арк.
						35
Зм.	Лист	№ докум.	Підпис	Дата		

Алгоритм обміну даними передбачає послідовну передачу запиту, обробку інформації, виконання математичного прогнозування та повернення результату у структурованому вигляді.

Нижче подано загальний ланцюжок взаємодії між основними модулями системи:

а) формування запиту;

1) після введення користувачем назви міста у компоненті SearchForm на фронтенді формується HTTP GET-запит на маршрут /api/weather/forecast, до якого додаються параметри запиту;

б) прийом запиту на сервері;

2) після введення користувачем назви міста у компоненті SearchForm на фронтенді формується HTTP GET-запит на маршрут /api/weather/forecast, до якого додаються параметри запиту;

в) звернення до зовнішнього API;

3) у контролері викликається функція fetchForecastData.js або fetchCurrentWeather.js (залежно від типу запиту), яка звертається до OpenWeather API. Для цього формується новий HTTP-запит із використанням ключа доступу (що зберігається у файлі .env);

г) отримання та обробка відповіді;

4) отримані JSON-дані передаються до відповідних модулів математичного прогнозування (наприклад, temperature.js, humidity.js), які здійснюють обчислення з урахуванням історичних даних та застосуванням алгоритмів часових рядів, чисельних методів і ланцюгів Маркова;

д) формування відповіді;

5) після обробки результатів на сервері формується фінальний об'єкт JSON, який включає;

- поточні погодні умови (current);
- трьохгодинний прогноз (hourlyForecast);
- денний прогноз на 7 днів (dailyForecast);
- геолокаційні дані (location).

е) передача даних клієнту;

б) сервер відправляє згенерований об'єкт JSON на клієнт. Axios на фронтенді отримує відповідь і передає її у відповідні компоненти інтерфейсу (WeatherWidget, ForecastTable, ForecastByDay);

ж) відображення даних;

7) компоненти React виконують рендеринг даних у зручному та структурованому вигляді. При зміні мови або теми інтерфейсу зміни застосовуються автоматично завдяки використанню Context API та i18next.

Асинхронна обробка та модульність логіки дозволяють забезпечити безперебійну взаємодію між клієнтом і сервером, уникати блокувань інтерфейсу та забезпечувати швидку реакцію системи на введення користувача.

5.5 Архітектура програмного забезпечення

Архітектура програмного забезпечення реалізованої системи є модульною та побудована на основі принципів розділення відповідальностей. Вона поділена на два основні рівні: клієнтський (frontend) та серверний (backend), кожен з яких виконує чітко визначені функції.

Клієнтський рівень побудований за допомогою бібліотеки React. Візуальні компоненти згруповані за функціональністю й утворюють окремі блоки, що відповідають за конкретні частини інтерфейсу. Уся логіка рендерингу, перемикання тем і мов, а також відображення погодних даних реалізована у вигляді функціональних компонентів. Для управління станом і побудови асинхронних запитів застосовуються хуки useState, useEffect та бібліотека axios.

Серверна частина, реалізована з використанням Node.js і Express, забезпечує обробку запитів, взаємодію з OpenWeather API, реалізацію математичних обчислень, маршрутизацію та формування відповіді у форматі JSON. Вона структурована у вигляді набору окремих модулів, згрупованих за функціональним призначенням: маршрути, контролери, зовнішні API, математичні модулі та утиліти.

					ІС12.230БАК.005 ПЗ	Арк.
						37
Зм.	Лист	№ докум.	Підпис	Дата		

Назва модуля / файлу	Назва функції або компонента	Призначення функції / компонента	Вхідні параметри / джерела даних	Вихід / результат
ast.jsx		прогноз на день	hourlyForecast	прогнозів з кроком 3 год
helpers/getWeatherIcon.js	getWeatherIcon	Повертає URL погодної іконки за її кодом (icon)	icon (рядок – код іконки, наприклад, 01d)	URL-адреса зображення погодної іконки
helpers/formatDate.js	formatDate	Форматує дату в зручний для виведення вигляд	timestamp (час у секундах або ISO-формат)	Рядок формату дд.мм або дд.мм.рррр
data/dailyWeatherData.json	–	Історичні дані прогнозу погоди по днях за 15 років (використовується для обчислень)	–	Массив погодних даних
data/hourlyWeatherData.json	–	Історичні дані прогнозу погоди з трьохгодинним кроком за 15 років	–	Массив погодних даних
data/currentWeatherData.json	–	Дані поточної погоди по Києву для прикладу або тестування	–	Об'єкт з поточними погодними параметрами
i18n.js	–	Ініціалізація локалізації інтерфейсу через	–	Встановлення мовних ресурсів

Зм.	Лист	№ докум.	Підпис	Дата

тестові дії проводилися у середовищі браузера Google Chrome на ОС Windows 10. Основна мета тестування — переконатися, що система коректно реагує на дії користувача та забезпечує очікувану функціональність.

5.6.2 Тестування коректного сценарію пошуку

Після відкриття головної сторінки інтерфейс автоматично завантажує прогноз погоди для міста, яке користувач вводив раніше (наприклад, «Київ»). У верхній частині присутнє поле для пошуку нового населеного пункту. Для перевірки правильності роботи пошуку користувач вводить назву нового міста, наприклад «Лондон», після чого натискає клавішу Enter.

Система виконує запит до OpenWeather API, отримує відповідь та оновлює інтерфейс: відображається поточний прогноз на сьогодні з температурними параметрами, трьохгодинний прогноз, а також прогноз на найближчі 7 днів. Після оновлення, у локальному сховищі браузера зберігається назва введеного міста, що дозволяє відображати це ж місто при наступному вході.

Результат пошуку зображено на рисунку 5.4.

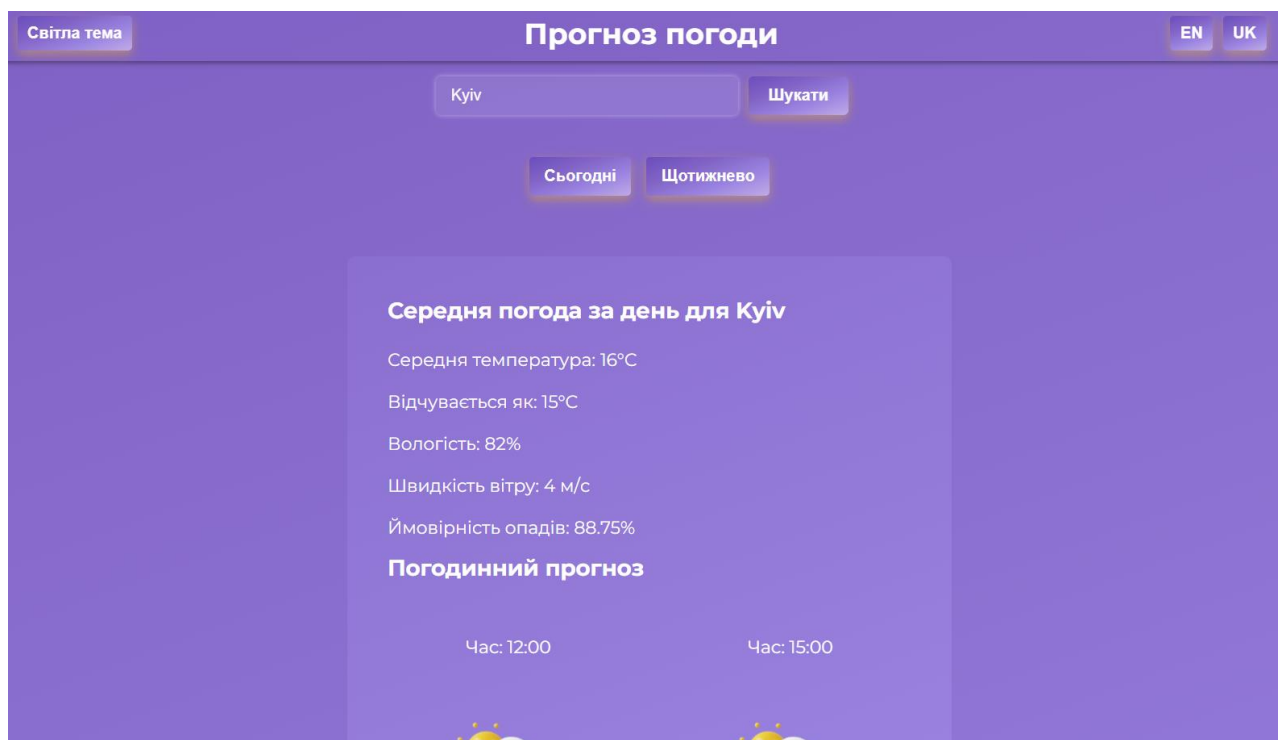


Рисунок 5.4 – Відображення прогнозу після введення нового міста

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		41

5.6.3 Тестування некоректного вводу

Для перевірки поведінки системи при неправильному введенні, у поле пошуку вводиться недійсне значення, наприклад: «mfr8457» або «123». У відповідь система надсилає запит до API, однак отримує помилку типу 404 або виставляє некоректну назву міста.

У цьому випадку на екрані з'являється повідомлення про помилку: «Не вдалося отримати дані про погоду: Request failed with status code 404» або . Прогноз не оновлюється, а попередні дані залишаються доступними. Така поведінка дозволяє уникнути повного краху інтерфейсу та підтримує зручність для користувача.

Повідомлення про помилку зображено на рисунку 5.5.

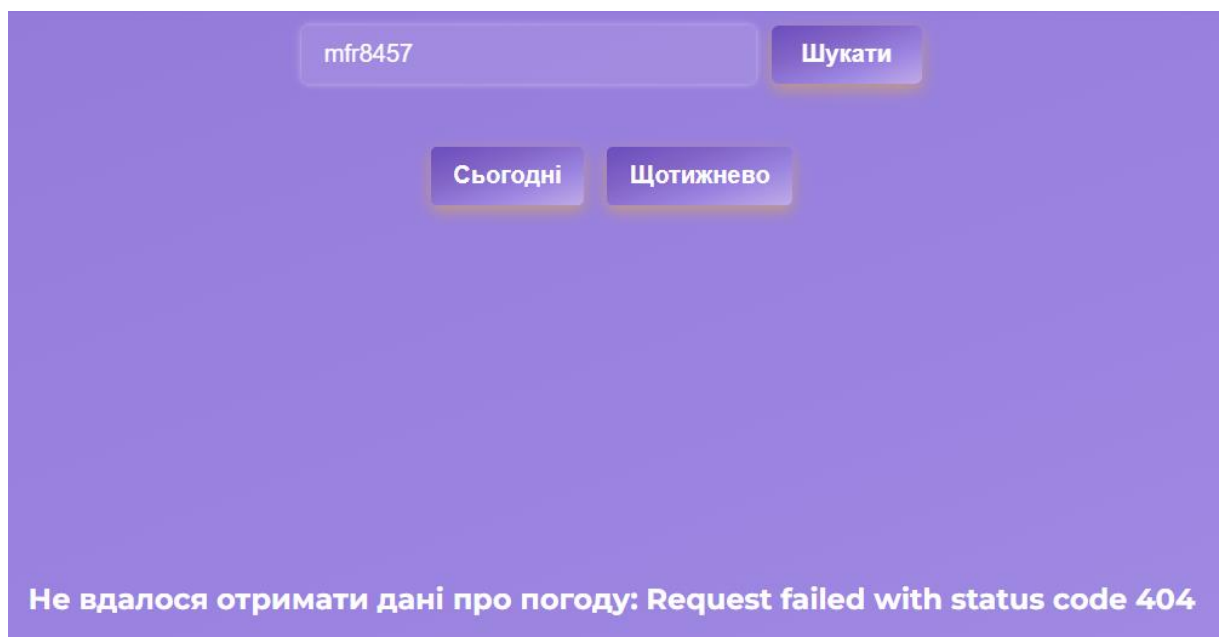


Рисунок 5.5 – Повідомлення про помилку при некоректному введенні назви міста

5.6.4 Перемикання теми оформлення

У верхній частині сторінки розміщено кнопку перемикання теми (іконка сонця/місяця). При натисканні на неї система змінює стиль оформлення інтерфейсу з темного на світлий і навпаки.

Зміна теми застосовується до всіх елементів — кольорів фону, шрифтів,

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		42

графічних блоків тощо. Обрана тема зберігається у локальному сховищі браузера і автоматично застосовується при наступному відкритті сторінки.

Світла тема інтерфейсу наведена на рисунку 5.6, де показано оформлення зі світлим фоном і базовими елементами.

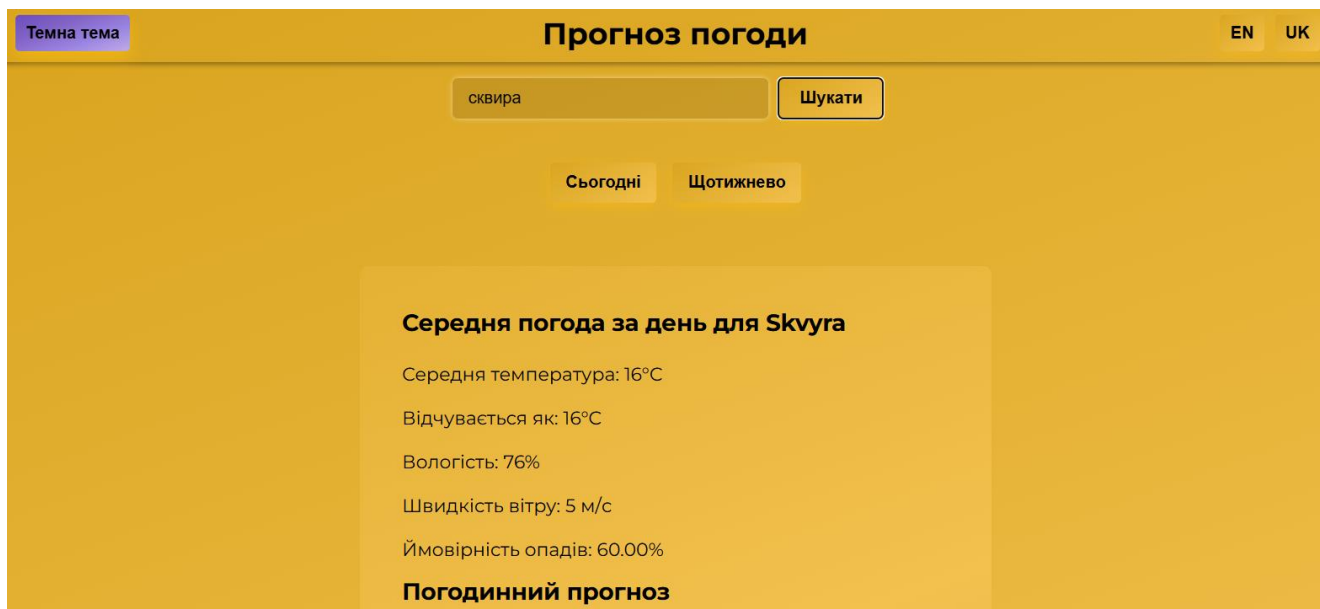


Рисунок 5.6 – Світла тема оформлення

Темна тема зображена на рисунку 5.7 та демонструє інтерфейс у темному режимі для зручності в умовах низького освітлення.

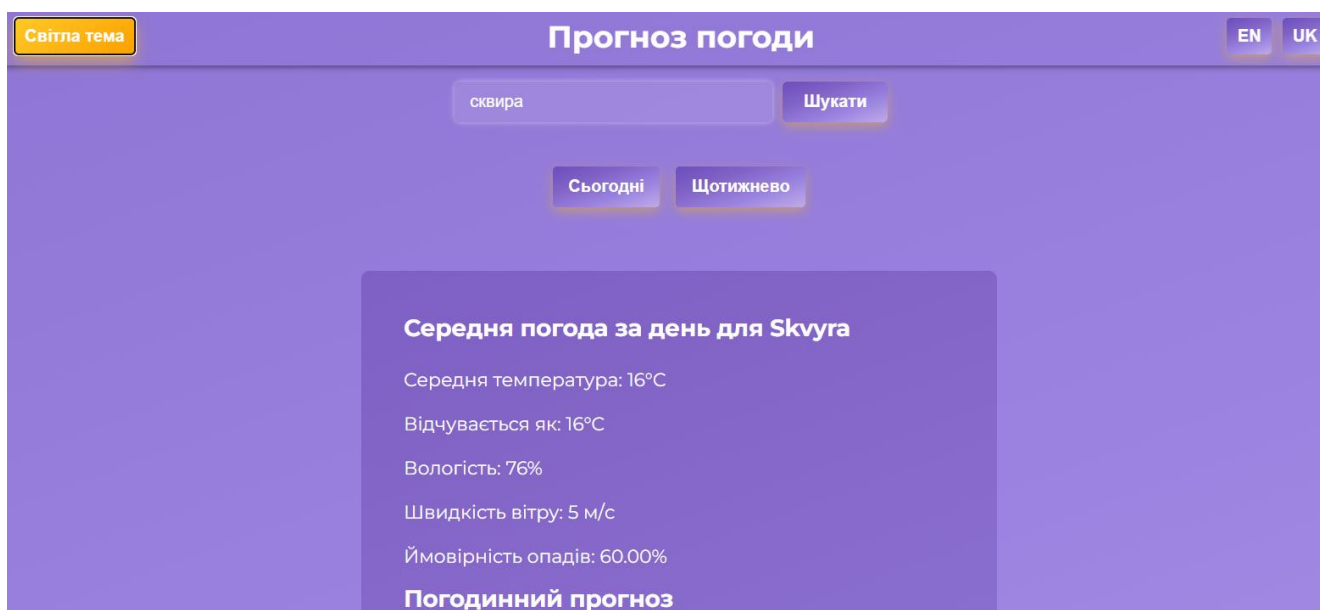


Рисунок 5.7 – Темна тема оформлення

5.6.5 Тестування локалізації інтерфейсу

Інтерфейс системи підтримує дві мови: українську та англійську. Для перемикання мови передбачено відповідну кнопку з прапорцем у шапці сторінки. При зміні мови автоматично перекладаються усі елементи інтерфейсу: назви днів тижня, мітки параметрів, повідомлення про помилки, тощо.

Локалізація реалізована через бібліотеку `i18next` із збереженням вибраної мови у локальному сховищі. Це дозволяє зберігати налаштування навіть після перезавантаження сторінки.

Приклад інтерфейсу англійською мовою зображено на рисунку 5.8.

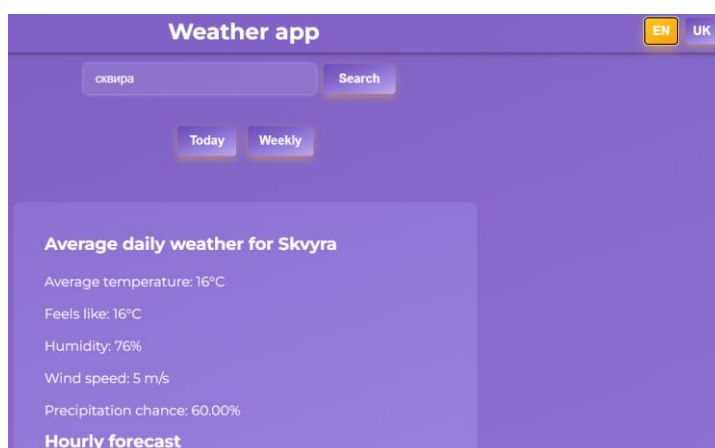


Рисунок 5.8 – Інтерфейс англійською мовою

Приклад інтерфейсу українською мовою зображено на рисунку 5.8.

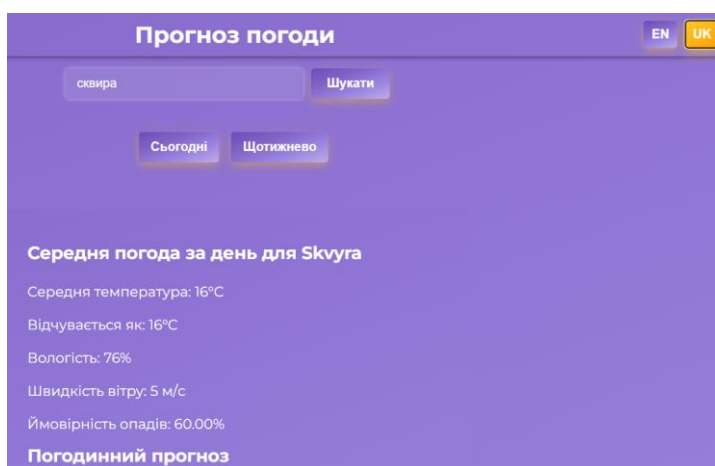


Рисунок 5.9 – Інтерфейс українською мовою

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		44

5.6.6 Обробка збоїв зовнішнього API

У разі відсутності відповіді від сервера OpenWeather (наприклад, проблеми з мережею або помилка 500), система виводить повідомлення: «Не вдалося отримати дані про погоду: Request failed with status code 500».

Для цього передбачено механізм обробки винятків на стороні клієнта: запити до API обгорнуті в конструкції try-catch, а у разі помилки відображається fallback-повідомлення. Це дозволяє уникнути зависання застосунку та забезпечує базову стійкість до зовнішніх збоїв.

Приклад повідомлення про помилку наведено на рисунку 5.9.

Не вдалося отримати дані про погоду: Request failed with status code 500

Рисунок 5.10 – Повідомлення про помилку при недоступності API

5.6.7 Тестування адаптивності інтерфейсу

Інтерфейс застосунку розроблений з урахуванням принципів адаптивного дизайну, що дозволяє коректно відображати компоненти як на десктопних пристроях, так і на мобільних телефонах та планшетах. Для перевірки адаптивності було використано інструменти вбудованої симуляції в браузері Google Chrome (DevTools → Device Mode), де проведено тестування на таких розмірах екранів:

- 360×640 (типовий мобільний);
- 768×1024 (планшет у портретному режимі);
- 1366×768 (стандартний ноутбук).

У всіх випадках перевірялося:

- відображення головної сторінки, включно з погодними картками та кнопками навігації;
- доступність кнопок перемикання мови й теми;
- коректне масштабування шрифтів і блоків;

–збереження логіки відображення трьохгодинного та тижневого прогнозів.

На всіх розмірах екрана елементи інтерфейсу адаптувалися коректно, горизонтального скролу не спостерігалось. Основні блоки компонувалися в одну колонку, що забезпечувало зручність використання на мобільних пристроях.

На рисунку 5.11 представлено приклад відображення інтерфейсу на мобільному пристрої.

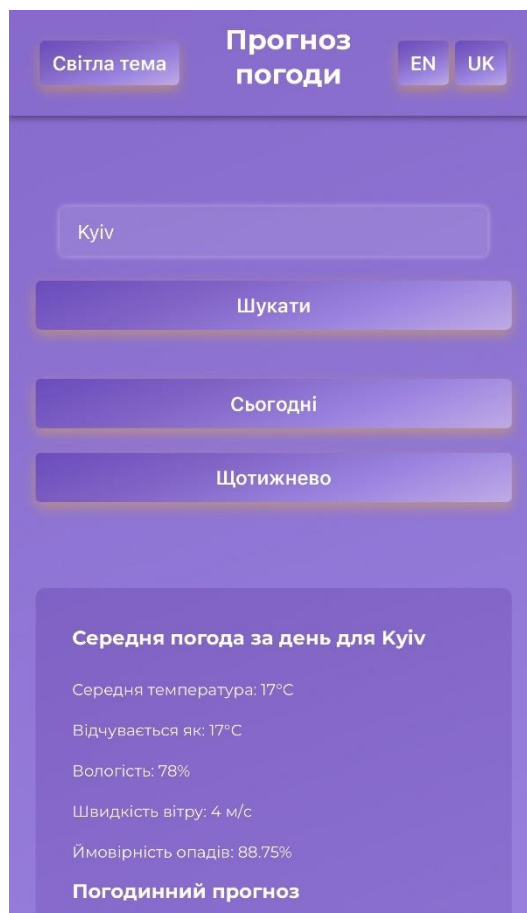


Рисунок 5.11 – Відображення головної сторінки застосунку на мобільному пристрої

5.6.8 Тестування збереження налаштувань localStorage

Для підвищення зручності користування застосунком зберігає індивідуальні налаштування користувача у сховищі localStorage браузера. Це дозволяє при наступному відкритті системи автоматично відновити:

– останнє введене місто (ключ "weather-city");

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		46

- обрану мову інтерфейсу (ключ «i18nextLng»);
- вибрану тему оформлення (ключ "theme").

Під час тестування було вручну перевірено сценарії:

- користувач вводить нове місто (наприклад, "Лондон") → оновлює сторінку → прогноз автоматично відкривається для цього міста;
- змінюється мова з української на англійську → після перезавантаження інтерфейс залишається англійським;
- увімкнено темну тему → при наступному запуску тема не змінюється.

Також було перевірено очищення LocalStorage вручну — у такому випадку застосунок повертається до дефолтних значень (місто: "Київ", мова: українська, тема: світла).

Висновки до розділу 5

У п'ятому розділі було всебічно розглянуто процес проектування та реалізації інформаційної системи прогнозування атмосферних явищ, що забезпечує інтерактивну взаємодію користувача з метеорологічними даними в режимі реального часу. Застосування клієнт-серверної архітектури дало змогу ефективно розмежувати функціональні обов'язки між клієнтською частиною, що відповідає за візуалізацію та взаємодію з користувачем, і серверною частиною, яка виконує обробку запитів, математичне моделювання та роботу з API.

На основі бібліотеки React реалізовано гнучкий та адаптивний інтерфейс, що підтримує багатомовність (через i18next), перемикання тем оформлення (світла/темна тема), а також забезпечує високу швидкодію за рахунок компонентного підходу. Серверна частина побудована з використанням Node.js і Express, що дозволило реалізувати асинхронну обробку запитів, зменшити навантаження на систему та забезпечити стабільну роботу з OpenWeather API.

Функціональна модель системи описує основні сценарії використання, зокрема вибір міста, перегляд прогнозу, перемикання мов і тем, оновлення даних, зберігання налаштувань у локальному сховищі тощо. У межах розділу описано

					ІС12.230БАК.005 ПЗ	Арк.
						47
Зм.	Лист	№ докум.	Підпис	Дата		

структуру інформаційних масивів, які використовуються для передачі погодних даних, а також логіку взаємодії між клієнтською та серверною частинами у вигляді REST-запитів.

Реалізовані компоненти системи охоплюють повний цикл обробки метеорологічної інформації: від запиту даних через API, опрацювання історичних JSON-масивів, застосування математичних методів (AR(2), чисельне інтегрування, ланцюги Маркова) — до виведення результату користувачеві у зручному форматі. Для забезпечення коректності роботи системи було проведено тестування ключових функціональних сценаріїв, зокрема: обробки коректного та помилкового вводу, перевірки стабільності інтерфейсу при зміні мови чи теми, імітації відсутності відповіді з API та відображення відповідних повідомлень користувачеві.

Узагальнюючи, реалізована архітектура забезпечує модульність, гнучкість і масштабованість інформаційної системи. Вона дозволяє без змін у базовій структурі доповнювати функціонал, адаптувати застосунок до нових вимог, покращувати точність прогнозу, а також забезпечувати зручність користування в широкому спектрі умов.

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		48

6 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

6.1 Змістовна постановка задачі

У сучасних умовах глобальної зміни клімату, зростаючої нестабільності атмосферних процесів і залежності суспільства від погодних факторів, завдання точного короткострокового прогнозування метеорологічних параметрів є надзвичайно актуальним. Погодні умови мають прямий вплив на транспорт, енергетику, сільське господарство, міську інфраструктуру та повсякденне життя населення. Тому існує необхідність у створенні інтелектуальних систем, здатних формувати достовірні прогнози погоди з високою часовою роздільністю.

Метою цієї системи є побудова адаптивної математичної моделі, яка здійснює короткостроковий прогноз погодних умов для обраного населеного пункту на найближчі 7 днів. У межах даного дослідження реалізація відбувається на прикладі міста Київ. Система використовує відкриті історичні погодні дані за останні 15 років (2010–2024), отримані з OpenWeather API, що дозволяє врахувати довгострокові сезонні закономірності, типові цикли атмосферних явищ та локальні особливості клімату регіону.

Прогноз реалізується з фіксованим часовим кроком у 3 години, що забезпечує повне покриття доби у вигляді 8 часових інтервалів на день. Таким чином, система прогнозує 56 значень для кожного з параметрів на тижневий період. Вибір трьохгодинного кроку зумовлений компромісом між точністю та швидкістю обчислень: він дозволяє моделювати динаміку погодних процесів на мікрорівні (з урахуванням добових коливань), водночас не перевантажуючи систему надмірною деталізацією.

До моделі включено дев'ять основних параметрів, які формують загальну картину погодних умов:

- температура повітря кожні 3 години;
- середня температура за добу;
- мінімальна температура за добу;
- максимальна температура за добу;

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		49

- швидкість вітру;
- вологість повітря;
- температура "відчувається як", яка враховує вологість і вітер;
- погодні умови, що класифікуються за станом атмосфери (ясно, дощ, сніг, хмарно тощо);
- ймовірність опадів.

Завдяки адаптивному підходу, система автоматично тестує для кожного параметра три альтернативні математичні методи прогнозування:

- чисельні методи;
- часові ряди другого порядку (AR(2));
- ланцюги Маркова.

На етапі навчання для кожного параметра визначається метод із найнижчою середньою абсолютною відносною похибкою (MAPE). Таким чином, із трьох протестованих алгоритмів вибирається один найточніший, який у подальшому використовується для розрахунку в програмі.

Такий підхід забезпечує оптимальний баланс між швидкодією, простотою реалізації, адаптивністю до реальних даних і максимальною точністю. Система працює в автоматичному режимі, без участі користувача у процесі розрахунку. Після введення назви міста вона самостійно формує семиденний прогноз на основі історичних даних та поточної дати запуску.

6.2 Математична постановка задачі

Метою прогнозної підсистеми є побудова моделі короткострокового прогнозування погодних параметрів для певного населеного пункту на 7 днів наперед з часовим кроком 3 години. Формально завдання полягає у побудові вектор-функції прогнозу на основі історичних метеорологічних спостережень.

Вхідні дані:

- поточна дата запуску системи: $T \in \mathbb{N}$ — дискретна змінна, що позначає дату, з якої починається прогнозування;

					ІС12.230БАК.005 ПЗ	Арк.
						50
Зм.	Лист	№ докум.	Підпис	Дата		

– глибина прогнозу: прогноз виконується на $t \in \{T + 1, \dots, T + 7\}$ — 7 наступних календарних днів;

– крок дискретизації: $\Delta t = 3$ години. Для кожного дня формується 8 трьохгодинних інтервалів $j \in \{1, 2, \dots, 8\}$, що відповідає моментам часу $t_j \in \{00:00, 03:00, \dots, 21:00\}$;

– історичні дані за 15 років: $D = \{d_1, d_2, \dots, d_n\}$, де кожен елемент $d_i \in R^9$ — вектор метеорологічних параметрів за один трьохгодинний інтервал

$$d_i = \{\tau_i, \bar{\tau}_i, \tau_{\min_i}, \tau_{\max_i}, V_i, H_i, \tau_{\text{feel}_i}, W_i, P_i\}, \quad (6.1)$$

де τ_i — температура повітря в певну годину;

$\bar{\tau}_i$ — середня температура за добу;

$\tau_{\min_i}, \tau_{\max_i}$ — мінімальна та максимальна температура доби;

V_i — швидкість вітру;

H_i — вологість повітря;

τ_{feel_i} — температура, яка «відчувається»;

W_i — погодні умови (за класифікацією OpenWeather);

P_i — ймовірність опадів (від 0 до 1).

Вихідні дані:

– формується набір прогнозованих значень погодних параметрів на 7 днів уперед з інтервалом 3 години. У результаті маємо масив

$$F(T) \rightarrow \{\hat{\tau}_{t_j}, \hat{\tau}_t, \hat{\tau}_{\min_t}, \hat{\tau}_{\max_t}, \hat{V}_{t_j}, \hat{H}_{t_j}, \hat{\tau}_{\text{feel}_{t_j}}, \hat{W}_{t_j}, \hat{P}_{t_j}\}, \quad (6.2)$$

де $j \in \{1, 2, \dots, 8\}$ — індекс трьохгодинного інтервалу доби $t \in \{T + 1, \dots, T + 7\}$.

Математичне формулювання задачі:

– для кожного параметра формується окрема функція прогнозування

$$\hat{\tau}_{t_j} = f_1(D, T, j), \quad (6.3)$$

$$\hat{\tau}_t = f_2(\{\hat{\tau}_{t1}, \dots, \hat{\tau}_{t8}\}), \quad (6.4)$$

$$\hat{\tau}_{\min_t} = f_3(\{\hat{\tau}_{t1}, \dots, \hat{\tau}_{t8}\}), \quad (6.5)$$

$$\hat{\tau}_{\max_t} = f_4(\{\hat{\tau}_{t1}, \dots, \hat{\tau}_{t8}\}), \quad (6.6)$$

$$\hat{V}_{tj} = f_5(D, T, j), \quad (6.7)$$

$$\hat{H}_{tj} = f_6(D, T, j), \quad (6.8)$$

$$\hat{\tau}_{\text{feel}_{tj}} = f_7(\hat{\tau}_{tj}, \hat{V}_{tj}, \hat{H}_{tj}), \quad (6.9)$$

$$\hat{W}_{tj} = f_8(D, T, j), \quad (6.10)$$

$$\hat{P}_{tj} = f_9(D, T, j). \quad (6.11)$$

Адаптивний підхід:

а) для кожної функції f_i тестується три підходи прогнозування:

- 1) чисельний метод;
- 2) модель часових рядів AR(2);
- 3) ланцюг Маркова.

Для кожної функції обчислюється похибка прогнозу за метрикою MAPE:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \cdot 100\%, \quad (6.12)$$

де y_i — фактичне значення параметра з API OpenWeather, \hat{y}_i — прогнозоване значення, n — кількість інтервалів.

У результаті адаптивного підходу вибирається лише один метод для кожного параметра, а саме той, що забезпечує мінімальну середню похибку. Це дозволяє значно зменшити час обчислення та оптимізувати продуктивність системи, залишаючи лише найефективніші обчислення для кожного типу прогнозу.

6.3 Обґрунтування методу розв'язання

Задача прогнозування метеорологічних параметрів належить до класу складних часових моделей, оскільки передбачає роботу з динамічними змінними,

					ІС12.230БАК.005 ПЗ	Арк.
						52
Зм.	Лист	№ докум.	Підпис	Дата		

які змінюються з плином часу та мають залежність від попередніх станів. У контексті розробки інтерактивної вебсистеми важливо не лише досягти високої точності результатів, а й забезпечити швидкодію та стабільність роботи застосунку. Саме тому процес побудови моделі був поділений на два ключових етапи: пошук можливих методів розв'язання та експериментальне тестування їхньої точності.

Першим кроком стало виділення трьох математичних підходів, які найбільш широко використовуються для задач короткострокового прогнозування: чисельні методи, авторегресивні моделі часових рядів та імовірнісні ланцюги Маркова. Кожен із цих методів має свої переваги та обмеження, а тому був протестований на відповідність до конкретного типу параметра.

Чисельні методи є найбільш швидкими у виконанні й мають просту реалізацію. Вони базуються на класичних математичних операціях, таких як середнє значення, вибір мінімуму або максимуму серед ряду значень. Такі методи доцільно застосовувати до параметрів, які слабо варіюються в часі або піддаються простій агрегації. Наприклад, якщо потрібно отримати середньодобову температуру або обрати мінімальне значення за добу, саме чисельні методи демонструють високу ефективність.

Однак для параметрів, які суттєво змінюються протягом доби та мають часову залежність, чисельні методи не завжди забезпечують достатній рівень точності. У таких випадках доцільно використовувати моделі часових рядів. Зокрема, авторегресивна модель другого порядку (AR(2)) дає змогу передбачити значення параметра на основі його двох попередніх значень. Це дозволяє враховувати внутрішні тренди та затримки, які властиві температурі, вологості чи швидкості вітру. Такі моделі краще адаптуються до плавної зміни параметрів у часі.

Для опису параметрів, що мають дискретну або категоріальну природу, таких як погодні умови (ясно, дощ, сніг, хмарно тощо), чисельні та авторегресивні моделі не підходять. У таких випадках доцільно використовувати імовірнісні моделі, зокрема ланцюги Маркова. Цей метод дозволяє моделювати ймовірність переходу з одного стану в інший на основі частотності та закономірності змін у минулому. Він є особливо ефективним для категоріальних змінних, де не існує безперервної

шкали.

Щоб оцінити, наскільки кожен із запропонованих методів підходить до кожного параметра, було проведено тестування всіх трьох підходів для кожного з дев'яти параметрів прогнозу. Після побудови моделей результати були порівняні з фактичними значеннями, отриманими з OpenWeather API, за допомогою метрики середньої абсолютної відносної похибки (MAPE). Саме ця метрика дозволила визначити ступінь точності кожного методу у відсотках, незалежно від масштабу самих значень.

На основі цих тестувань у подальших підрозділах буде обґрунтовано вибір найбільш оптимального методу для кожного параметра. При цьому система у фінальному вигляді використовує лише один метод для кожного параметра, щоб забезпечити оптимальну швидкість та зменшити навантаження на обчислювальні ресурси. Таким чином, підхід до розв'язання задачі є адаптивним на етапі дослідження, але оптимізованим у кінцевій реалізації.

6.4 Оцінка точності методів прогнозування

Для визначення найефективнішого підходу до прогнозування кожного з метеорологічних параметрів було проведено експериментальне тестування. Воно базувалося на зіставленні фактичних погодних даних із результатами, отриманими за допомогою трьох математичних методів. Тестування виконувалося для одного повного дня, а саме 24 травня 2025 року, з розбиттям доби на вісім трьохгодинних інтервалів. Такий підхід дозволяє отримати достатньо репрезентативну вибірку для розрахунку точності та зберегти зручність викладу.

Перший етап тестування полягав у тому, що для кожного інтервалу доби було зібрано фактичні значення температури з API OpenWeather. Далі ці значення порівнювалися з тими, які обчислювалися окремо за кожним із трьох підходів: чисельним методом, авторегресійною моделлю другого порядку AR(2) та ланцюгами Маркова.

Щоб оцінити точність кожного підходу, використовувалася метрика MAPE,

					ІС12.230БАК.005 ПЗ	Арк.
						54
Зм.	Лист	№ докум.	Підпис	Дата		

модель AR(2), яка найточніше відтворює динаміку температури повітря упродовж доби. Похибка її прогнозів становить лише 1,74%, тоді як інші методи демонструють суттєво гірші результати. Чисельний метод відхиляється в середньому на 8,24%, а ланцюги Маркова відхиляються на 13,30%. З огляду на це, для прогнозування температури повітря було обрано саме авторегресійну модель другого порядку, як найбільш точну та надійну серед протестованих підходів.

6.4.2 Середня, мінімальна та максимальна температура

Після того як система формує прогноз температури повітря для кожного з восьми трьохгодинних інтервалів у межах однієї доби, отримується повноцінна добова картина зміни температур. Ці дані створюються на основі авторегресійної моделі другого порядку, яка була обрана для прогнозування як найбільш точна згідно з результатами тестування.

Середнє значення температури визначається як звичайне арифметичне середнє восьми прогнозованих значень. Це дозволяє оцінити загальний температурний фон упродовж доби без додаткових складних розрахунків. Такий підхід є простим і обґрунтованим, оскільки забезпечує достатню точність для користувацького рівня застосування.

Мінімальне значення температури розраховується шляхом вибору найменшого значення серед усіх восьми прогнозованих температур протягом доби. Цей показник допомагає врахувати найнижчий температурний рівень у межах конкретного дня, що є важливим для користувачів, які планують ранкову чи нічну активність.

Максимальна температура визначається аналогічно, шляхом вибору найбільшого значення серед тих самих восьми прогнозів. Цей параметр дозволяє відстежити пік тепла протягом дня, що особливо корисно для планування роботи на відкритому повітрі, поїздок або інших активностей у денний час.

Окреме моделювання середньої, мінімальної чи максимальної температури не проводиться. Це пояснюється тим, що всі три значення безпосередньо залежать

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		57

$$\left| \frac{4,2-3,9}{4,2} \right| + \left| \frac{3,2-2,7}{3,2} \right| \approx 12,59\%.$$

Середня абсолютна відносна похибка для методу ланцюгів Маркова становить:

$$\begin{aligned} \text{МАРЕ} = & \frac{1}{8} \left| \frac{3,0-2,1}{3,0} \right| + \left| \frac{2,8-1,8}{2,8} \right| + \left| \frac{3,0-2,0}{3,0} \right| + \left| \frac{4,0-2,9}{4,0} \right| + \left| \frac{4,8-3,7}{4,8} \right| + \left| \frac{5,0-4,0}{5,0} \right| + \\ & \left| \frac{4,2-3,3}{4,2} \right| + \left| \frac{3,2-2,5}{3,2} \right| \approx 26,60\%. \end{aligned}$$

За результатами обчислень видно, що найменшу середню похибку при прогнозуванні швидкості вітру має модель AR(2). Вона демонструє стабільне наближення до фактичних значень, середнє відхилення становить лише 6,26%. У свою чергу чисельний метод показав середню похибку 12,59%, що майже вдвічі більше. Метод ланцюгів Маркова виявився найменш точним у цьому випадку — середнє відхилення становило 26,60%, що вказує на слабку придатність для прогнозування швидкості вітру. З огляду на ці результати, для подальшого прогнозування цього параметра використовується саме модель AR(2), яка показала найвищу точність.

6.4.4 Вологість повітря

На таблиці 6.7 наведено значення фактичної вологості повітря та значення, які було спрогнозовано за допомогою трьох різних методів. Усі розрахунки стосуються 24 травня 2025 року.

Таблиця 6.7 — Фактична та прогнозована вологість повітря за 24 травня

Час	Фактична вологість	AR(2)	Чисельні методи	Ланцюги Маркова
00:00	80	78	76	74
03:00	78	76	73	70
06:00	72	69	68	66
09:00	65	66	62	60
12:00	58	59	57	55
15:00	55	53	54	53

18:00	65	64	63	60
21:00	75	73	71	68

Обчислюється середня відносна похибка для трьох методів прогнозування вологості повітря за 24 травня. Це дозволяє оцінити точність кожного з підходів та зробити висновок про ефективність обчислень.

Середня абсолютна відносна похибка для AR(2) становить:

$$\text{МАРЕ} = \frac{1}{8} \left| \frac{80-78}{80} \right| + \left| \frac{78-76}{78} \right| + \left| \frac{72-69}{72} \right| + \left| \frac{65-66}{65} \right| + \left| \frac{58-59}{58} \right| + \left| \frac{55-53}{55} \right| + \left| \frac{65-64}{65} \right| + \left| \frac{75-73}{75} \right| \approx 2,54\%.$$

Середня абсолютна відносна похибка для чисельного методу становить:

$$\text{МАРЕ} = \frac{1}{8} \left| \frac{80-76}{80} \right| + \left| \frac{78-73}{78} \right| + \left| \frac{72-68}{72} \right| + \left| \frac{65-62}{65} \right| + \left| \frac{58-57}{58} \right| + \left| \frac{55-54}{55} \right| + \left| \frac{65-63}{65} \right| + \left| \frac{75-71}{75} \right| \approx 4,19\%.$$

Середня абсолютна відносна похибка для методу ланцюгів Маркова становить:

$$\text{МАРЕ} = \frac{1}{8} \left| \frac{80-74}{80} \right| + \left| \frac{78-70}{78} \right| + \left| \frac{72-66}{72} \right| + \left| \frac{65-60}{65} \right| + \left| \frac{58-55}{58} \right| + \left| \frac{55-53}{55} \right| + \left| \frac{65-60}{65} \right| + \left| \frac{75-68}{75} \right| \approx 7,45\%.$$

За результатами обчислень видно, що найменшу середню похибку при прогнозуванні вологості повітря має модель AR(2). Вона демонструє стабільне наближення до фактичних значень, середнє відхилення становить лише 2,54%. У свою чергу чисельний метод показав середню похибку 4,19%. Метод ланцюгів Маркова виявився найменш точним серед розглянутих — середнє відхилення становило 7,45%. З огляду на ці результати, для подальшого прогнозування цього параметра використовується саме модель AR(2), яка показала найвищу точність.

6.4.5 «Відчувається як»

Параметр «Відчувається як» розраховується за аналітичною формулою, що враховує прогнозовані значення температури повітря, вологості та швидкості вітру.

					ІС12.230БАК.005 ПЗ	Арк.
						60
Зм.	Лист	№ докум.	Підпис	Дата		

Цей показник відображає суб'єктивне відчуття температури людиною з урахуванням впливу вологості та вітру, що робить його більш реалістичним, ніж проста температура повітря.

Оскільки цей параметр залежить безпосередньо від інших спрогнозованих значень, окреме прогнозування не здійснюється. Такий підхід знижує обчислювальне навантаження і не впливає на точність системи, оскільки похибки у цьому параметрі залежать від точності базових прогнозів температури, вологості та вітру.

6.4.6 Погодні умови

На таблиці 6.8 наведено значення фактичних погодних умов та значення, які було спрогнозовано за допомогою трьох різних методів. Усі розрахунки стосуються 24 травня 2025 року.

Таблиця 6.8 — Фактичні та прогнозовані погодні умови за 24 травня

Час	Фактичні умови	AR(2)	Чисельні методи	Ланцюги Маркова
00:00	clear sky	clear sky	-	clear sky
03:00	clear sky	parthy cloudy	-	parthy cloudy
06:00	parthy cloudy	few clouds	-	parthy cloudy
09:00	few clouds	few clouds	-	few clouds
12:00	sunny	sunny	-	sunny
15:00	clear sky	clear sky	-	clear sky
18:00	clear sky	parthy cloudy	-	clear sky
21:00	clear sky	clear sky	-	clear sky

Чисельні методи не застосовуються для прогнозування погодних умов через їхню категоріальну природу. Ці методи орієнтовані на обробку числових неперервних даних, тоді як погодні умови представлені дискретними станами (наприклад, «clear sky», «few clouds», «rain»). Спроба застосувати чисельні методи до таких категорій призведе до втрати смислової інформації та неточних

результатів, адже не існує математичних операцій, які коректно працювали б із якісними позначеннями станів.

Обчислюється похибка для трьох методів прогнозування погодних умов за 24 травня. Це дозволяє оцінити точність кожного підходу та зробити висновок про ефективність обчислень. Для категоріальних даних, таких як погодні умови (стани типу «чисте небо», «частково хмарно» тощо), класичні метрики, які працюють з числовими величинами (наприклад, MAPE), не підходять, тому ми використаємо точність (Accuracy).

Похибка для AR(2) становить:

– модель AR(2) правильно передбачила погодні умови у п'яти з восьми випадків, що дає точність 62,5%;

Похибка для методу ланцюгів Маркова становить:

– модель ланцюгів Маркова правильно передбачила погодні умови у семи випадках із восьми, що відповідає точності 87,5%.

За результатами обчислень видно, що найвищу точність при прогнозуванні погодних умов демонструє модель ланцюгів Маркова. Вона правильно передбачила погодні стани у 87,5% випадків, що свідчить про високу надійність цього методу для роботи з категоріальними даними. У порівнянні з нею модель AR(2) показала нижчу точність — 62,5%, що робить її менш придатною для прогнозування дискретних погодних станів. Через відсутність адекватних математичних операцій для роботи з якісними категоріями чисельні методи в цьому випадку не застосовуються. Таким чином, для подальшого прогнозування погодних умов у системі обрано метод ланцюгів Маркова, як найбільш ефективний і точний серед досліджених.

6.4.7 Ймовірність опадів

Ймовірність опадів є параметром, що має дискретну ймовірнісну природу та не є безперервною кількісною величиною, як температура чи вологість. На практиці вона часто виражається у вигляді відсоткового значення, але базується на

ймовірності настання певної погодної події — дощу, снігу чи їхньої відсутності.

Через це для її прогнозування найбільш доречним виявився підхід, який враховує ймовірнісні переходи між станами погоди на основі історичних спостережень. Ланцюги Маркова дозволяють змодельовати залежність поточного стану (опаді/без опадів) від попереднього, з урахуванням частоти переходів, що спостерігалися в минулому.

На відміну від чисельних методів чи авторегресійних моделей, які оперують числовими рядами, ланцюги Маркова адаптовані саме для категоріальних або дискретних подій, до яких належить факт випадання опадів. Вони дозволяють не лише врахувати історичну ймовірність настання події, а й забезпечити компактну та логічну модель прогнозу, де кожен стан має ймовірності переходу в інші, засновані на частотному аналізі реальних даних.

Тому в межах реалізованої системи прогнозу атмосферних явищ для параметра «ймовірність опадів» було використано лише метод ланцюгів Маркова, як найбільш придатний для моделювання такого типу даних.

6.4 Опис методів розв'язання

У процесі побудови математичної моделі прогнозу атмосферних параметрів було використано три основні методи: чисельні обчислення, моделі часових рядів і ланцюги Маркова. Кожен із них було підібрано залежно від особливостей самого параметра. Наприклад, безперервні величини, такі як температура чи вологість, краще піддаються авторегресійному аналізу, тоді як категоріальні параметри, як-от погодні умови або ймовірність опадів, зручно моделювати за допомогою дискретних станів.

Спочатку для кожного параметра були протестовані всі три методи. Система обчислювала прогноз за кожним підходом, після чого результати порівнювалися з фактичними значеннями. Для оцінки точності застосовували середню абсолютну відносну похибку, що дало змогу об'єктивно визначити, який метод краще справляється з тим чи іншим типом даних.

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		63

Параметр	Обраний метод	Обґрунтування методу
		модель AR(2) забезпечує стабільний прогноз змін вітру
вологість повітря	AR(2)	найнижча MAPE: 2,54%; показав найменшу похибку
«відчувається як»	аналітична формула	обчислюється за формулою на основі температури, вологості й вітру; окремо не прогнозується
погодні умови	ланцюги Маркова	найвища точність (Ассурасу): 87,5%; категоріальні дані — непридатні для AR(2)/чисельних
ймовірність опадів	ланцюги Маркова	дискретна природа; ланцюги Маркова найкраще моделюють ймовірності переходів станів

6.4.1 Чисельні методи

Середня температура доби:

– формула:

$$\hat{\tau}_t = \frac{1}{8} \sum_{j=1}^8 \hat{\tau}_{tj}, \quad (6.13)$$

- метод: обчислення середнього арифметичного;
- вхідні дані: спрогнозовані трьохгодинні температури;
- результат: середньодобова температура;
- пояснення: ця формула дозволяє отримати інтегральну оцінку температури протягом доби. Такий метод є чисельно простим і забезпечує точність при великій кількості інтервалів.

Мінімальна та максимальна температура:

- формули:

$$\hat{t}_{\min_t} = \min \{\hat{t}_{t_1}, \hat{t}_{t_2}, \dots, \hat{t}_{t_8}\}, \quad (6.14)$$

$$\hat{t}_{\max_t} = \max \{\hat{t}_{t_1}, \hat{t}_{t_2}, \dots, \hat{t}_{t_8}\}, \quad (6.15)$$

- метод: пошук екстремумів;
- вхідні дані: ті ж самі 8 прогнозованих значень температури;
- результат: межі температури за добу;
- пояснення: параметри обчислюються чисельно без моделей. Вони особливо важливі для визначення загрозливих погодних умов (наприклад, заморозки чи спека).

Температура, яка «відчувається як»:

- формула:

$$\hat{t}_{\text{feel}_{t_j}} = \hat{t}_{t_j} + 0,33\hat{H}_{t_j} - 0,7\hat{V}_{t_j} - 4,0, \quad (6.16)$$

- метод: аналітична формула;
- вхідні дані: температура, вологість, швидкість вітру;
- результат: наближена оцінка суб'єктивного відчуття температури;
- пояснення: параметр не потребує окремого прогнозу, оскільки розраховується на основі спрогнозованих величин[13]. Формула є апроксимацією поширених моделей (наприклад, Steadman's formula).

6.4.2 Часові ряди

Температура повітря:

– формула:

$$\hat{\tau}_{t_j} = a_1 \times \tau_{t_{j-1}} + a_2 \times \tau_{t_{j-2}}, \quad (6.17)$$

– метод: авторегресійна модель другого порядку AR(2);

– вхідні дані: історичні значення температури в той самий інтервал за попередні роки;

– результат: прогноз температури для конкретного інтервалу;

– пояснення: модель AR(2) дозволяє враховувати інерційність метеорологічного процесу. Коефіцієнти a_1 , a_2 визначаються окремо для кожного інтервалу часу на основі методу найменших квадратів.

Швидкість вітру:

– формула:

$$\hat{V}_{t_j} = b_1 \times V_{t_{j-1}} + b_2 \times V_{t_{j-2}}, \quad (6.18)$$

– метод: AR(2);

– вхідні дані: історичні значення швидкості вітру;

– результат: прогноз швидкості вітру;

– пояснення: вітрові зміни є високочастотними, але демонструють автокореляцію. AR(2) забезпечує компроміс між точністю та стабільністю при прогнозуванні.

Вологість повітря:

– формула:

$$\hat{H}_{t_j} = c_1 \times H_{t_{j-1}} + c_2 \times H_{t_{j-2}} \quad (6.19)$$

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		67

- метод: AR(2);
- вхідні дані: історичні значення вологості;
- результат: прогноз вологості повітря;
- пояснення: вологість є безперервним параметром, значення якого суттєво залежать від попередніх у часі[14]. Така автозалежність дає змогу ефективно застосовувати авторегресійну модель AR(2), яка враховує динаміку змін і забезпечує високу точність прогнозу.

6.4.3 Ланцюги Маркова

Погодні умови:

- формула:

$$P(\widehat{W}_{t_j} = s_k | W_{t_{j-1}} = s_l) = p_{lk} \quad (6.20)$$

- метод: ланцюг Маркова першого порядку;
- вхідні дані: ймовірності переходів між станами за історією;
- результат: прогноз категоріального стану погоди (наприклад, Clear, Rain);
- пояснення: стан погоди в наступному інтервалі залежить лише від попереднього. Такий підхід дозволяє моделювати ймовірнісну структуру погодних переходів з урахуванням статистики.

Ймовірність опадів:

- формула:

$$\widehat{P}_{t_j} = \frac{N_{rain}}{N_{total}}, \quad (6.21)$$

- метод: частотна оцінка на основі історичних спостережень;
- вхідні дані: кількість випадків опадів за аналогічних умов;
- результат: ймовірність випадіння опадів у конкретний інтервал;

– пояснення: цей параметр розраховується не через модель, а як частка випадків з опадами серед усіх наявних спостережень[15], що забезпечує статистично достовірну оцінку.

Висновки до розділу 6

У шостому розділі представлено повну математичну основу інтерактивної системи прогнозування атмосферних явищ. Розпочато з формалізації задачі, де чітко визначено вхідні та вихідні параметри, часову дискретизацію та логіку побудови прогнозу на основі історичних даних. Для кожного з дев'яти метеорологічних параметрів сформовано відповідну математичну модель із урахуванням їхньої природи — неперервної, дискретної або похідної.

Здійснено порівняльний аналіз трьох підходів прогнозування — чисельних методів, авторегресійної моделі другого порядку (AR(2)) та ланцюгів Маркова. Для кожного параметра проведено тестування, результати якого оцінено за допомогою метрик точності (MAPE та Ассигасу). За результатами порівняння до кожного параметра було адаптивно обрано найточніший метод, який застосовується у фінальній версії системи.

Модель AR(2) виявилася найбільш ефективною для прогнозування температури, вологості та швидкості вітру, забезпечуючи найнижчі значення похибки. Для середньої, мінімальної та максимальної температури застосовано чисельні методи, що ґрунтуються на обробці прогнозованих значень. Параметр «відчувається як» розраховується аналітичною формулою на основі вже спрогнозованих величин. Погодні умови та ймовірність опадів найточніше відображаються за допомогою ланцюгів Маркова, які враховують імовірнісні переходи між станами.

Таким чином, реалізовано адаптивну математичну модель, яка обирає найефективніший метод для кожного з параметрів на основі емпіричних даних. Такий підхід забезпечує високу точність, стабільність та продуктивність прогновної системи.

					ІС12.230БАК.005 ПЗ	Арк.
						69
Зм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВКИ

У результаті розробки було реалізовано інтерактивну систему прогнозування атмосферних явищ, яка поєднує сучасні вебтехнології та математичні методи для побудови короткострокового прогнозу погоди з трьохгодинним кроком на сім днів уперед. Система надає користувачеві структуровану інформацію про дев'ять основних метеорологічних параметрів і працює повністю автоматизовано — на основі поточної дати запуску застосунку. Реалізація клієнтської частини на React і серверної логіки через Node.js і Express забезпечила стабільність, гнучкість і можливість масштабування.

Під час роботи над проєктом було виконано повний цикл розробки — від вибору джерела погодних даних, проєктування архітектури, побудови математичних моделей і до створення зручного інтерфейсу користувача. Зі сторони клієнта реалізовано перемикання тем оформлення, що дало можливість кожному користувачеві обрати комфортний режим перегляду — світлий або темний. Також було успішно впроваджено інтернаціоналізацію: система підтримує не лише українську, а й англійську мову, з автоматичним збереженням налаштувань у локальному сховищі. Це зробило застосунок доступним для ширшого кола користувачів.

Проєкт став важливою практичною частиною навчання. Під час реалізації вдалося суттєво поглибити навички роботи з REST API, асинхронними запитами, обробкою помилок, форматуванням великих JSON-файлів і математичним аналізом. Особливо цінним виявився досвід обробки великих обсягів даних: для побудови моделей використовувались історичні метеодані за кожні три години протягом останніх п'ятнадцяти років. Це дозволило досягти високої точності прогнозу, але також поставило технічний виклик — час завантаження щотижневого прогнозу в окремих випадках може сягати до хвилини. Цю ситуацію було враховано під час оптимізації обробки запитів, і система загалом працює швидко й стабільно.

Також у процесі роботи з API OpenWeather виникли труднощі, пов'язані з

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		70

обмеженням доступу до архіву даних. Безкоштовний рівень дозволяє отримувати лише поточний прогноз, тож для забезпечення повноцінного тестування математичної моделі було придбано історичні погодні дані окремо для міста Київ. Це дало змогу не лише реалізувати математичну модель, а й перевірити її точність на основі реальних даних.

Особливу роль у системі відіграє адаптивна математична модель. Для кожного параметра тестувались три підходи — AR(2), чисельні методи та ланцюги Маркова — і за підсумками тестування в систему інтегровано лише найточніші. Це дозволило зберегти гнучкість на етапі проектування і водночас забезпечити стабільні результати у фінальній версії. Точність моделі протягом першої доби була найвищою, а середнє відхилення не перевищувало 2–3°C, що є добрим результатом для короткострокового прогнозу.

Інтерфейс системи адаптовано до використання на різних пристроях, включно з мобільними. Користувачеві не потрібно вводити жодних параметрів вручну: застосунок самостійно формує прогноз на основі поточної дати та збережених налаштувань. Програма витримує некоректні введення та збої API, повідомляючи про помилки, не припиняючи роботу.

У цілому, робота над проектом дозволила не лише створити функціональну та практично корисну систему, але й набути важливих навичок роботи з реальними API, великими обсягами даних, математичним аналізом, інтерфейсним дизайном і тестуванням. Проект є готовою основою для подальшого розвитку, наприклад, додавання карти метеостанцій, історії змін погоди або графічного аналізу. Результати цієї роботи підтверджують можливість якісного поєднання прикладної математики, програмної інженерії та користувацького досвіду в одному цілісному рішенні.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OpenWeather API Documentation: веб-сайт. URL: <https://openweathermap.org/api> (дата звернення: 25.04.2025)
2. OpenWeather Historical Weather API: веб-сайт. URL: <https://openweathermap.org/history> (дата звернення: 25.04.2025)
3. Weather and Climate: веб-сайт. URL: <https://www.weather-and-climate.com> (дата звернення: 25.04.2025)
4. AccuWeather: веб-сайт. URL: <https://www.accuweather.com> (дата звернення: 25.04.2025)
5. The Weather Channel (Weather.com): веб-сайт. URL: <https://weather.com> (дата звернення: 25.04.2025)
6. MeteoBlue – Weather Data & Forecasts: веб-сайт. URL: <https://www.meteoblue.com> (дата звернення: 25.04.2025)
7. i18next Internationalization Framework: веб-сайт. URL: <https://www.i18next.com> (дата звернення: 25.04.2025)
8. Express.js Guide: веб-сайт. URL: <https://expressjs.com> (дата звернення: 25.04.2025)
9. JavaScript Documentation (MDN): веб-сайт. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 25.04.2025)
10. Node.js Documentation: веб-сайт. URL: <https://nodejs.org> (дата звернення: 25.04.2025)
11. React Documentation: веб-сайт. URL: <https://react.dev> (дата звернення: 25.04.2025)
12. GitHub – OpenWeather Node.js Examples: веб-сайт. URL: <https://github.com/search?q=openweather+nodejs> (дата звернення: 25.04.2025)
13. Montgomery D. C., Jennings C. L., Kulahci M. Introduction to Time Series Analysis and Forecasting. John Wiley & Sons, 2015. – Chapter 2: веб-сайт. URL: <https://www.wiley.com/en->

[us/Introduction+to+Time+Series+Analysis+and+Forecasting-p-9781118745113](https://numerical.recipes) (дата звернення: 25.04.2025)

14. Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P. Numerical Recipes in C. The Art of Scientific Computing. Cambridge University Press, 2007. – Chapter 13: веб-сайт. URL: <https://numerical.recipes> (дата звернення: 25.04.2025)

15. Medhi J. Stochastic Processes. New Age International, 2009. – Chapter 5 (Markov Chains): веб-сайт. URL: <https://archive.org/details/stochasticprocessesmedhi> (дата звернення: 25.04.2025)

					ІС12.230БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		73