

**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 «Інформаційне забезпечення
робототехнічних систем»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Робототехнічна система моніторингу параметрів повітряного
середовища з функцією аналізу та оповіщення»**

Виконав:

студент IV курсу, групи ІК-11

Бабух Роман Юрійович _____

Керівник:

асистент кафедри ІСТ

Драган Михайло Сергійович _____

Рецензент:

доцент кафедри ІПІ, к.т.н., доцент

Лішук Катерина Ігорівна _____

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

Студент _____

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 «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 2025 р.

ЗАВДАННЯ
на дипломний проєкт студенту
Бабуху Роману Юрійовичу

1. Тема проєкту «Робототехнічна система моніторингу параметрів повітряного середовища з функцією аналізу та оповіщення», керівник проєкту Драган Михайло Сергійович, затверджені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: 09.06.2025.
3. Вихідні дані до проєкту: Система повинна забезпечувати збір, зберігання та аналіз даних про параметри якості повітря в реальному часі, прогнозування майбутніх показників на основі отриманих даних, відображення інформації у вигляді графіків через веб-інтерфейс.
4. Зміст пояснювальної записки: перелік умовних скорочень, вступ, огляд предметної області, аналіз існуючих рішень, формування вимог до системи, вибір технологій, розробка системи, тестування системи.
5. Перелік графічного матеріалу: чотири кресленики, функціональна схема, діаграма прецедентів, діаграма бізнес процесів, діаграма потоку даних.
6. Дата видачі завдання: 25.02.2025.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
	Узгодження теми та технічного завдання	08.03.2025	
	Опис предметної області	18.04.2025	
	Аналіз існуючих рішень	25.04.2025	
	Формування вимог до системи	02.05.2025	
	Програмна реалізація системи	09.05.2025	
	Інтеграція алгоритмів машинного навчання	16.05.2025	
	Тестування системи	28.06.2025	
	Формування пояснювальної записки	09.06.2025	

Студент

Роман БАБУХ

Керівник

Михайло ДРАГАН

АНОТАЦІЯ

Робототехнічна система моніторингу параметрів повітряного середовища з функцією аналізу та оповіщення.

Проект містить 74 с. тексту, 27 рисунок, 1 таблицю, посилання на 32 літературні джерела, додатки та 4 кресленика.

МОНІТОРИНГ, ЯКІСТЬ ПОВІТРЯ, ІоТ, ШТУЧНИЙ ІНТЕЛЕКТ, ПРОГНОЗУВАННЯ, ВЕБІНТЕРФЕЙС, МАШИННЕ НАВЧАННЯ

Об'єктом розробки є система автоматизованого моніторингу якості повітря в режимі реального часу.

Мета розробки — покращення екологічної ситуації в міських умовах, аграрному секторі або для індивідуального використання, забезпечуючи своєчасне виявлення забруднень і підтримку екологічної свідомості населення.

У дипломному проекті розроблено повноцінну архітектуру програмно-апаратної системи моніторингу. Система складається з модулів збору даних, їх обробки, модуля прогнозування з використанням машинного навчання, модуля оповіщення про аномалії, а також вебінтерфейсу для перегляду даних у зручному форматі. У процесі роботи здійснено аналіз предметної області та сучасних рішень у галузі моніторингу довкілля, описано логіку роботи модулів, проведено тестування користувацького інтерфейсу.

Результати цього проекту можуть бути використані для створення систем екологічного моніторингу в міських умовах, в аграрному секторі або для індивідуального застосування у побуті.

SUMMARY

Robotic Air Environment Monitoring System with Analysis and Alerting Function

The project contains 74 pages, 27 figures, 1 table, links to 32 sources, annexes and 4 design documents.

**MONITORING, AIR QUALITY, IoT, ARTIFICIAL INTELLIGENCE,
FORECASTING, WEB INTERFACE, MACHINE LEARNING**

The object of development is an automated real-time air quality monitoring system.

The purpose of the development is to improve the environmental situation in urban areas, the agricultural sector or for individual use, ensuring timely detection of pollution and maintaining environmental awareness of the population.

The thesis developed a complete architecture of a software monitoring system. The system consists of data collection and processing modules, a machine learning forecasting module, an anomaly alert module, and a web interface for viewing data in a convenient format. In the course of the work, the subject area and modern solutions in the field of environmental monitoring were analyzed, the logic of the modules was described, and the user interface was tested.

The results of this project can be used to create environmental monitoring systems in urban areas, in the agricultural sector, or for individual use in everyday life.

№ рядка	Формат	Позначення	Найменування	Кіл. аркушів	№ екз.	Примітка
1			Документація загальна			
2						
3			Знову розроблена			
4						
5	A4	ІК11.020БАК.006 ПЗ	Пояснювальна записка	74		
6	A3	ІК11.020БАК.006 Д1	Робототехнічна система	1		
7			моніторингу параметрів			
8			повітряного середовища			
9			з функцією аналізу та оповіщення.			
10			Функціональна схема.			
11	A3	ІК11.020БАК.006 Д2	Робототехнічна система	1		
12			моніторингу параметрів			
13			повітряного середовища			
14			з функцією аналізу та оповіщення.			
15			Діаграма потоку даних.			
16	A3	ІК11.020БАК.006 Д3	Робототехнічна система	1		
17			моніторингу параметрів			
18			повітряного середовища			
19			з функцією аналізу та оповіщення.			
20			Діаграма бізнес процесів.			
21	A3	ІК11.020БАК.006 Д4	Робототехнічна система	1		
22			моніторингу параметрів			
23			повітряного середовища			
24			з функцією аналізу та оповіщення.			
25			Діаграма прецедентів.			
26						
27						
28						
Зм.	Лист	№ докум.	Підпис			
Розробив	Бабух Р.Ю.					
Перевірив	Драган М.С.					
Затв.						
				ІК11.020БАК.006 ТП		
				Робототехнічна система моніторингу параметрів повітряного середовища з функцією аналізу та оповіщення.		
				Відомість дипломного проєкту		
				Літ.		
				Арк.		
				Аркушів		
				Т		
				1		
				1		
				КПІ ім. Ігоря Сікорського		
				Група ІК-11		

**Пояснювальна записка
до дипломного проєкту
на тему: «Робототехнічна система моніторингу
параметрів повітряного середовища з функцією аналізу
та оповіщення»**

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	6
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Аналіз проблеми моніторингу навколишнього середовища	8
1.2 Стан у світі	8
1.3 Стан в Україні	10
1.4 Сучасні підходи до моніторингу якості повітря	12
1.4.1 Параметри повітряного середовища	12
1.4.2 Типи датчиків і їх розміщення	13
1.4.3 Зв'язок і передача даних	13
1.4.4 Обробка та аналіз даних	14
1.4.5 Сповіщення та візуалізація	15
Висновки до розділу 1	15
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	17
2.1 Огляд існуючих рішень	17
2.2 Міські та державні рішення	17
2.2.1 London Air Quality Network	18
2.2.2 AirNow	20
2.3 Комерційні системи	22
2.3.1 PurpleAir	22
2.3.2 Kaiterra Data Platform	24
2.4 Публічні системи з відкритим кодом	26
Висновки до розділу 2	28
3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ	30
3.1 Виставлення вимог до системи	30
3.2 Базові вимоги	30

					ІК11.020БАК.006 ТП			
Зм.	Лист	№ докум.	Підпис		Робототехнічна система моніторингу параметрів повітряного середовища з функцією аналізу та оповіщення. Пояснювальна записка	Літ.	Арк.	Аркушів
Розробив	Бабух Р.Ю.					Т	2	74
Перевірив	Драган М.С.							
Затв.						КПІ ім. Ігоря Сікорського Група ІК-11		

3.3	Додаткові вимоги	31
3.4	Обмеження системи	32
3.5	Архітектура системи	32
3.5.1	Модуль надсилання телеметричних даних	33
3.5.2	Модуль аналізу	33
3.5.3	Модуль прогнозування	34
3.5.4	Модуль оповіщення	34
3.5.5	Користувацький інтерфейс	34
3.5.6	Брокер повідомлень	34
3.6	Діграма потоку даних	35
	Висновки до розділу 3	35
4	ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ	37
4.1	Вибір технологій	37
4.2	Брокер повідомлень	37
4.3	Технології машинного навчання	38
4.4	Основна мова програмування	41
4.5	Користувацький інтерфейс	42
4.5.1	Вибір вебфреймворку	42
4.5.2	Вибір додаткових бібліотек	43
4.5.2.1	Бібліотека для графіків	44
4.5.2.2	Бібліотека для стилізації інтерфейсу	44
4.6	База даних	45
4.7	Додаткові інструменти	46
	Висновки до розділу 4	47
5	РОЗРОБКА СИСТЕМИ	49
5.1	Підготовка даних	49
5.2	Модель машинного навчання	50
5.2.1	Тренування моделі машинного навчання	50
5.2.2	Оцінка натренованої моделі машинного навчання	52
5.3	Створення бази даних	53
5.4	Модуль надсилання даних	54
5.5	Модуль прогнозування даних	55

5.6 Модуль аналізу даних	55
5.7 Модуль оповіщення	56
5.8 Користувачський інтерфейс	57
5.9 Діаграма BPMN	60
Висновки до розділу 5	61
6 ТЕСТУВАННЯ СИСТЕМИ	62
6.1 Огляд тестування системи.....	62
6.2 Запуск системи	62
6.3 Підключення датчиків	64
6.4 Тестування модулів.....	64
6.5 Тестування користувацького інтерфейсу	67
Висновки до розділу 6	69
ВИСНОВКИ.....	71
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТОК А.....	74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AQI – Air Quality Index

BPMN – Business Process Model and Notation

HTTP – HyperText Transfer Protocol

IoT – Internet of Things

LAQN – London Air Quality Network

MQTT – Message Queuing Telemetry Transport

PM – Particulate Matter

SMS – Short Message Service.

					ІК11.020БАК.006 ПЗ	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

ВСТУП

Сучасна екологічна ситуація вимагає нових технічних підходів до моніторингу навколишнього середовища. Забруднення повітря залишається одним із ключових викликів у багатьох регіонах світу, зокрема в умовах урбанізації, індустріалізації та глобальних змін клімату. Відсутність доступу до актуальних та локалізованих екологічних даних суттєво ускладнює прийняття ефективних управлінських рішень як на державному, так і на індивідуальному рівні. Саме тому розробка автоматизованих систем збору, обробки та візуалізації екологічної інформації набуває дедалі більшої актуальності.

На сьогодні існує багато комерційних рішень у сфері моніторингу якості повітря. Втім, більшість із них мають закриту інфраструктуру, не дозволяють гнучко підключати власні сенсори, обмежують користувача у виборі джерел даних, а також не надають можливості модифікувати або розширювати аналітичні модулі відповідно до потреб конкретного застосування. Така ситуація створює потребу в розробці відкритих, масштабованих і модульних систем, які дозволяють не відображати поточну ситуацію.

Метою цієї роботи є розробка програмної системи моніторингу якості повітря, здатної в реальному часі обробляти телеметричні дані з сенсорів, здійснювати аналіз та виявлення аномалій, прогнозувати тенденції на основі навченої моделі та надавати користувачеві зручний візуальний інтерфейс для взаємодії з даними. Основний акцент зроблено на простоті, адаптивності архітектури, а також можливості подальшого розширення системи під нові задачі.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- а) зібрати та попередньо обробити телеметричні дані з реальних сенсорів;
- б) реалізувати архітектуру обміну повідомленнями між модулями системи на основі брокера повідомлень;
- г) реалізувати систему оповіщення у випадку виявлення аномальних значень;
- д) створити вебінтерфейс для відображення даних та прогнозів у зручному вигляді.

Ключові проєктні рішення базуються на принципах розподілених систем, зокрема використанні брокерів повідомлень, що забезпечують масштабованість та асинхронну взаємодію між модулями. Для прогнозування застосовується модель машинного навчання, навчена на реальних історичних даних, що дає змогу адаптувати систему до конкретної місцевості.

Результати цієї роботи можуть бути застосовані у сфері розумних міст, локального екологічного моніторингу, агротехнологій, а також для індивідуального використання мешканцями забруднених регіонів. Крім того, система може бути використана в освітньому процесі для демонстрації роботи сучасних інформаційних технологій.

					ІК11.020БАК.006 ПЗ	Арк.
						7
Зм.	Лист	№ докум.	Підпис	Дата		

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз проблеми моніторингу навколишнього середовища

Моніторинг навколишнього середовища стає все більш важливим у контексті урбанізації, промислового розвитку та зміни клімату. Якість повітря, зокрема, безпосередньо впливає на здоров'я людини, екологічну стабільність та якість життя. Вплив забруднюючих речовин у повітрі щорічно спричиняє мільйони передчасних смертей, що робить моніторинг повітряного середовища важливою складовою політики охорони здоров'я.

Забруднення повітря стало однією з гострих екологічних проблем сучасності. Історично, початок промислової революції ознаменувався значним збільшенням викидів забруднюючих речовин у результаті широкого використання викопного палива. У цей період відбулося поширення фабрик і міських центрів, що призвело до підвищення концентрації в повітрі таких забруднюючих речовин, як діоксид сірки, оксиди азоту та тверді частинки. Відомий «Великий смог» у Лондоні (Great Smog of London) в 1952 році, який призвів до тисяч смертей, підкреслив смертельну небезпеку неконтрольованого забруднення повітря.

1.2 Стан у світі

Протягом десятиліть, незважаючи на технологічний прогрес і регуляторні заходи в деяких регіонах, забруднення повітря залишається глобальною проблемою. За даними Всесвітньої організації охорони здоров'я (ВООЗ), у 2019 році вплив забруднення атмосферного (зовнішнього) та побутового повітря став причиною приблизно 6,7 мільйона передчасних смертей у всьому світі. При цьому 99% населення світу проживає в районах, де рівень якості повітря перевищує норми ВООЗ, що підвищує ризик розвитку серцевих захворювань, інсульту, хронічних обструктивних захворювань легень, раку та пневмонії.

Тягар забруднення повітря непропорційно лягає на країни з низьким та середнім рівнем доходу. Наприклад, у 2023 році в Бангладеш і Пакистані було

					ІК11.020БАК.006 ПЗ	Арк.
						8
Зм.	Лист	№ докум.	Підпис	Дата		

зафіксовано середню концентрацію PM_{2,5} на рівні 79,9 і 73,7 мікрограмів на кубічний метр відповідно, що приблизно в 15 разів перевищує рекомендований ВООЗ рівень у 5 мікрограмів. Такий високий рівень забруднення пов'язують із такими факторами, як промислові викиди, вихлопи транспортних засобів та спалювання біомаси для приготування їжі та опалення. На рисунку 1.1 зображена карта світового забруднення в наслідок використання викопних палив.

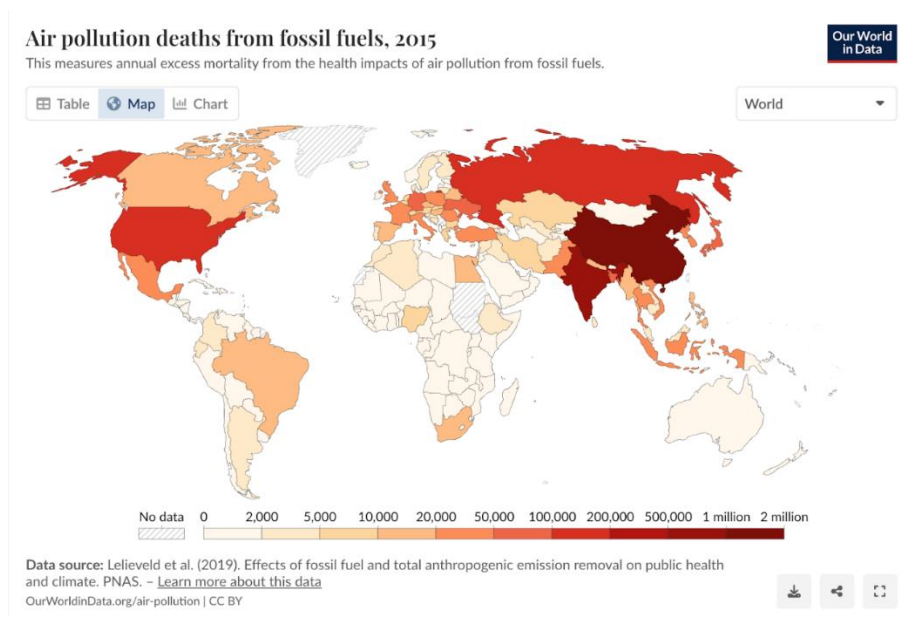


Рисунок 1.1 – Карта смертності від забруднення повітря внаслідок використання викопних палив, 2015 рік [1].

У Європі, незважаючи на поліпшення якості повітря в багатьох регіонах завдяки суворим нормам, проблеми залишаються. Міські райони продовжують боротися із забрудненням від транспорту, промислової діяльності та опалення житлових будинків. Європейське агентство з охорони навколишнього середовища повідомляє, що в 2021 році вплив дрібних твердих частинок призвів до приблизно 307 000 передчасних смертей в Європейському Союзі.

Вплив забруднення повітря на здоров'я є глибоким. Тривалий вплив пов'язаний із респіраторними та серцево-судинними захворюваннями, несприятливими наслідками вагітності та когнітивними порушеннями. Економічні витрати є колосальними і включають витрати на охорону здоров'я, втрату

продуктивності праці та зниження якості життя. З огляду на масштаби та серйозність проблеми, існує нагальна потреба у надійних системах моніторингу якості повітря. Кількість систем моніторингу якості повітря залишається незадовільною, на рисунку 1.2 зображена карта населених пунктів з такими системами.

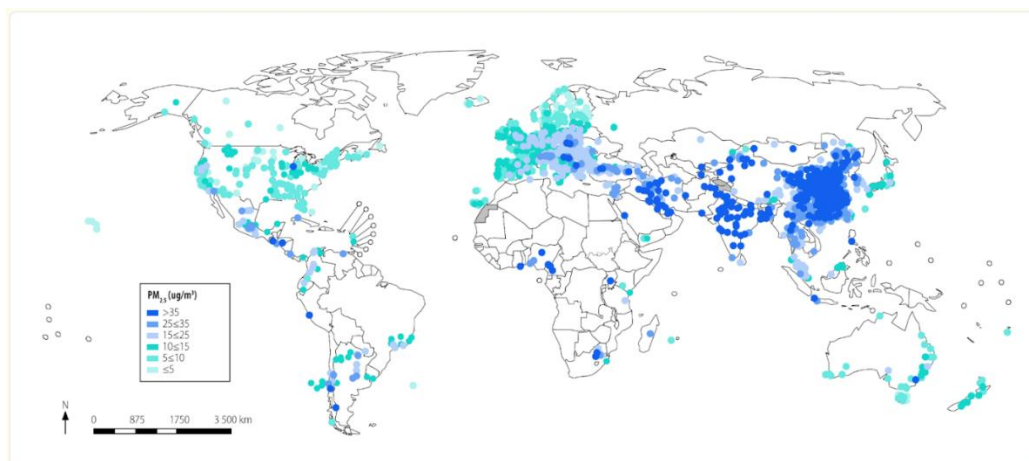


Рисунок 1.2 – Розташування населених пунктів з даними про $PM_{2,5}$, за кількістю наземних вимірювань, 2010–2019 роки [2].

1.3 Стан в Україні

Забруднення повітря є серйозною проблемою для навколишнього середовища та здоров'я населення України. Країна стикається з проблемами, пов'язаними з промисловими викидами, застарілою інфраструктурою та недостатніми екологічними нормами. У міських центрах, зокрема, спостерігається підвищена концентрація шкідливих забруднюючих речовин через інтенсивний дорожній рух, виробництво енергії на вугіллі та промислову діяльність.

За даними Всесвітньої організації охорони здоров'я, у 2019 році в Україні було зафіксовано концентрації $PM_{2,5}$, що перевищували рекомендовані норми, що негативно вплинуло на здоров'я населення. У таких містах, як Київ, Дніпро, Львів та Маріуполь, постійно спостерігаються підвищені рівні забруднення повітря. Зокрема, Дніпро та Кривий Ріг — центри важкої промисловості — часто страждають від смогу та днів із поганою якістю повітря.

					ІК11.020БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

Тривала війна ще більше ускладнила ситуацію, спричинивши руйнування інфраструктури, лісові пожежі та промислові аварії, в результаті яких в повітря потрапляють додаткові забруднювачі. Супутниковий моніторинг та наземні вимірювання показали зростання викидів, пов'язаних із вибухами, пожежами та порушенням промислового виробництва в зонах конфлікту.

Незважаючи на ці виклики, Україна вжила заходів для поліпшення моніторингу навколишнього середовища. Країна працює над розширенням мережі моніторингу якості повітря, а також з'явилося кілька державних і громадських ініціатив, спрямованих на усунення прогалин у державних даних. Наприклад, платформа SaveEcoBot, зображена на рисунку 1.3, агрегує дані датчиків з усієї країни, роблячи дані про якість повітря доступними для громадськості в режимі реального часу.

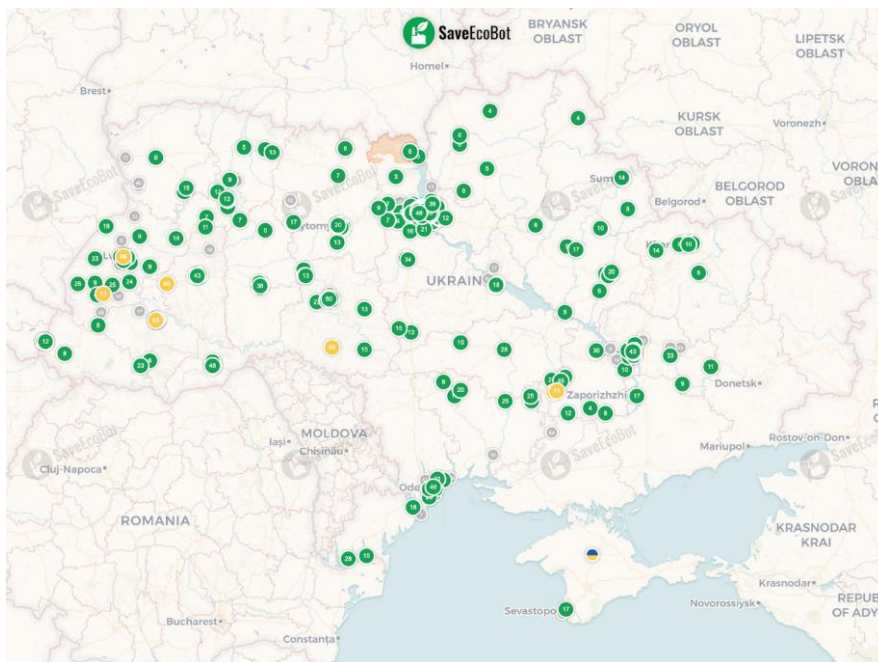


Рисунок 1.3 – Інтерактивна карта SaveEcoBot [3].

Однак для приведення рівня забруднення в безпечні межі все ще необхідне систематичне впровадження політики та модернізація промислової інфраструктури. Розробка масштабованих систем моніторингу повітря в режимі реального часу, особливо на основі технологій Інтернету речей та штучного

інтелекту, може відіграти важливу роль в екологічній стратегії України та сприяти підвищенню обізнаності та активності на місцевому та національному рівнях.

1.4 Сучасні підходи до моніторингу якості повітря

Сучасні системи моніторингу якості повітря використовують комбінацію апаратних і програмних рішень для збору, передачі та аналізу даних про стан навколишнього середовища. Ці системи різняться за рівнем складності — від високоточних станцій, що експлуатуються урядом, до компактних, недорогих сенсорних вузлів, розгорнутих у густонаселених міських районах, промислових зонах або навіть житлових районах. Поширення Інтернету речей дозволило створити нове покоління моніторів якості повітря, які є доступними за ціною, масштабованими та здатними передавати і аналізувати дані майже в режимі реального часу.

1.4.1 Параметри повітряного середовища

Моніторинг забруднення повітря зазвичай зосереджується на наборі ключових параметрів навколишнього середовища, які дозволяють оцінити рівень забруднення, його динаміку та потенційний вплив на здоров'я населення. Одним із основних показників є концентрація твердих частинок PM_{2,5} — мікроскопічних частинок діаметром до 2,5 мікрметра, які можуть проникати глибоко в легені та викликати серйозні респіраторні захворювання. Важливу роль також відіграють температура і вологість, які не є прямими забруднювачами, але суттєво впливають на розповсюдження, трансформацію та осідання шкідливих речовин у повітрі. Висока температура, наприклад, може сприяти хімічним реакціям, що утворюють вторинні забруднювачі, тоді як вологість може впливати на агрегацію частинок і зміну їх аеродинамічних властивостей. Комплексний підхід до моніторингу включає як безпосередні показники забруднення, так і контекстуальні умови, що визначають поведінку цих речовин у навколишньому середовищі.

					ІК11.020БАК.006 ПЗ	Арк.
						12
Зм.	Лист	№ докум.	Підпис	Дата		

1.4.2 Типи датчиків і їх розміщення

Системи моніторингу якості повітря зазвичай використовують дві категорії датчиків: еталонні та недорогі. Еталонні датчики — це високоточні прилади, які використовуються екологічними агентствами для нормативного моніторингу. Ці пристрої, такі як монітори бета-затухання для твердих частинок, газові хроматографи для летких сполук та аналізатори хемілюмінесценції для двоокису азоту, забезпечують високу точність даних, але є дорогими, потребують регулярного калібрування і зазвичай встановлюються на стаціонарних станціях.

На відміну від них, недорогі датчики використовують металооксидні напівпровідники, електрохімічні або оптичні технології для виявлення газів і параметрів навколишнього середовища. Хоча вони менш точні і більш чутливі до навколишнього шуму, вони доступні за ціною і підходять для розподіленого моніторингу в режимі реального часу. Ці датчики зазвичай інтегровані з мікроконтролерами, такими як Arduino або Raspberry Pi. Це робить їх ідеальними для «зроби сам» систем, дослідницьких проєктів та ініціатив у галузі громадянської науки. Завдяки компактним розмірам і низькому енергоспоживанню, вони можуть встановлюватися будь де, чи то в приміщенні, чи на вулиці, що дозволяє здійснювати масштабований моніторинг якості повітря.

1.4.3 Зв'язок і передача даних

Сучасні монітори навколишнього середовища, побудовані на основі технологій Інтернету речей, використовують різні засоби зв'язку для передачі зібраних даних, що залежить від умов розгортання системи та вимог до швидкості й надійності комунікації. У приміщеннях або в середовищах з доступом до інфраструктури найчастіше застосовуються з'єднання через Wi-Fi або Ethernet, що забезпечують стабільну передачу великих обсягів даних. У сільській місцевості або на великих відкритих просторах, де відсутня можливість підключення до

					ІК11.020БАК.006 ПЗ	Арк.
						13
Зм.	Лист	№ докум.	Підпис	Дата		

фіксованих мереж, перевагу надають енергоефективній технології LoRaWAN, яка дозволяє передавати дані на великі відстані при низькому енергоспоживанні. У більш складних та мобільних системах, особливо тих, що потребують широкого покриття або автономності, використовують стільниковий зв'язок, зокрема мережі 3G, 4G або 5G. Для передачі даних у режимі реального часу широко застосовуються мережеві протоколи MQTT, HTTP або Kafka, які забезпечують ефективну маршрутизацію повідомлень між сенсорами, аналітичними платформами та кінцевими системами сповіщення чи зберігання. Гнучке поєднання каналів передачі та протоколів дозволяє адаптувати IoT-системи моніторингу до найрізноманітніших умов експлуатації.

1.4.4 Обробка та аналіз даних

Після збору, дані про якість повітря зазвичай проходять низку етапів обробки, що забезпечують їхню придатність для подальшого аналізу та прийняття рішень. Першим кроком є попередня обробка, яка включає зменшення шуму в даних, видалення аномальних або пошкоджених значень, а також нормалізацію показників для приведення їх до єдиної шкали. У разі необхідності застосовуються поправочні коефіцієнти, які компенсують систематичні похибки сенсорів або змінні умови середовища. Далі здійснюється зберігання та агрегація даних: для цього часто використовують бази даних часових рядів, які оптимізовані для запису, зберігання та швидкого доступу до великих обсягів телеметричної інформації.

Наступним етапом є аналіз. Один з найпростіших методів базується на правилах: фіксовані порогові значення параметрів викликають автоматичне сповіщення. Однак для більш гнучкого та точного розуміння ситуації все частіше використовуються методи машинного навчання. Вони дозволяють виявляти складні аномалії, прогнозувати зміну параметрів у майбутньому, а також класифікувати типи забруднення. Серед популярних алгоритмів – Random Forest, моделі авторегресії ARIMA та рекурентні нейронні мережі типу LSTM, які добре працюють із послідовностями даних та дозволяють будувати прогнози на основі

					ІК11.020БАК.006 ПЗ	Арк.
						14
Зм.	Лист	№ докум.	Підпис	Дата		

історичних вимірювань. Такий підхід дає змогу не лише оперативно реагувати на перевищення допустимих рівнів, а й передбачати потенційно небезпечні ситуації до їх фактичного настання.

1.4.5 Сповіщення та візуалізація

Сучасні системи моніторингу навколишнього середовища включають інструменти візуалізації та оповіщення в режимі реального часу, які роблять дані датчиків доступними та придатними для використання. Візуалізація зазвичай здійснюється за допомогою вебпанелей або мобільних додатків з інтерактивними картами, графіками в режимі реального часу та діаграмами історичних даних.

Системи сповіщення спрацьовують, коли значення перевищують безпечні пороги або виявляються аномалії. Сповіщення надсилаються за допомогою push-повідомлень, SMS або електронної пошти. Це забезпечує швидке реагування на небезпечні умови та дозволяє оперативно вживати заходів для запобігання або мінімізації негативного впливу на здоров'я населення та довкілля.

Висновки до розділу 1

Моніторинг навколишнього середовища, зокрема якості повітря, виступає ключовим інструментом у боротьбі за здорове майбутнє суспільства. Проблема забруднення повітря набула глобального масштабу і несе серйозні ризики як для здоров'я людей, так і для економіки та екологічної стабільності. Незважаючи на певний прогрес у регулюванні в окремих регіонах, масштаби та наслідки забруднення залишаються значними, особливо в країнах із низьким та середнім рівнем доходу.

Україна, в контексті світових викликів, стикається із власним комплексом проблем, зумовлених індустріальною спадщиною, воєнними діями та недостатнім розвитком екологічної інфраструктури. Однак останні роки показали позитивні зрушення: громадські ініціативи, інтеграція супутникових даних, розвиток

цифрових платформ свідчать про готовність суспільства до змін і високий попит на прозорість екологічної інформації.

Окремої уваги заслуговує технічний прогрес у сфері моніторингу. Сучасні системи використовують як високоточні еталонні датчики, так і доступні компактні сенсори, що відкриває можливості для масштабованого, децентралізованого збору даних. Інтеграція Інтернету речей, гнучкі варіанти передачі даних, використання хмарних сервісів, аналітики та машинного навчання дозволяють не лише відстежувати поточний стан, а й будувати прогнози, виявляти аномалії та оперативно реагувати на загрози.

Предметна область демонструє як глибоку складність проблеми, так і наявність інноваційних підходів до її вирішення. Це створює надійне підґрунтя для розробки нових технологічних рішень, спрямованих на підвищення якості моніторингу, підвищення обізнаності громадян та підтримку екологічної політики на всіх рівнях.

У цьому контексті особливої ваги набуває розробка відкритих і доступних платформ для моніторингу, які можуть використовуватись як органами влади, так і окремими громадянами. Такий підхід не лише сприяє підвищенню довіри до екологічних даних, а й стимулює участь населення в охороні довкілля. Поширення open-source рішень, мобільних додатків та онлайн-сервісів надає змогу залучати широку аудиторію до збору, аналізу та інтерпретації екологічної інформації, що у свою чергу посилює екологічну свідомість та сприяє формуванню культури сталого розвитку.

2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

2.1 Огляд існуючих рішень

Огляд існуючих рішень має на меті проаналізувати наявні на сьогоднішній день системи моніторингу повітряного середовища. Потрібно проаналізувати існуючі рішення для моніторингу повітряного середовища, щоб зрозуміти сучасний стан технологій і практик у цій галузі. Огляд включає як комерційні системи, які часто призначені для професійного або промислового використання, так і проекти з відкритим вихідним кодом, які, як правило, керуються спільнотою або призначені для освітніх і дослідницьких цілей. Вивчаючи ці системи, необхідно виявити найкращі практики, загальні обмеження та технічні прогалини, які можна було б усунути в запропонованому рішенні. Такий аналіз дозволяє не лише оцінити рівень розвитку галузі, а й сформувані обґрунтовані вимоги до власної системи.

Важлива частина огляду присвячена методам, що використовуються для аналізу даних. Сюди входять методи попередньої обробки необроблених даних, зменшення шуму, виявлення аномалій, аналіз трендів і прогнозування змін якості повітря та інші сучасні методи. Дослідження показує, як існуючі системи перетворюють показання датчиків на змістовну інформацію для технічно-досвідчених і нетехнічних користувачів.

Для охоплення загальної картини сучасної галузі розділимо рішення на категорії:

- а) міські та державні;
 - 1) London Air Quality Network;
 - 2) AirNow;
- б) комерційні;
 - 1) PurpleAir;
 - 2) Kaiterra Data Platform;
- в) публічні з відкритим кодом;
 - 1) Sensor.Community.

2.2 Міські та державні рішення

					ІК11.020БАК.006 ПЗ	Арк.
						17
Зм.	Лист	№ докум.	Підпис	Дата		

2.2.1 London Air Quality Network

Лондонська мережа моніторингу якості повітря є однією з найбільших і найвідоміших мереж моніторингу міського повітря у світі [4]. Вона була заснована на початку 1990-х років і перебуває під управлінням провідної дослідницької організації у сфері екології та навколишнього середовища. Основною метою LAQN є моніторинг забруднення повітря у Великому Лондоні та прилеглих районах, надання даних з високою роздільною здатністю в режимі реального часу для підтримки ініціатив у сфері охорони здоров'я, екологічних досліджень та розробки політики.

LAQN вимірює широкий спектр параметрів якості повітря. До них відносяться концентрації діоксиду азоту, твердих частинок, озону, оксиду вуглецю і діоксиду сірки. Деякі станції моніторингу також вимірюють додаткові параметри, такі як леткі органічні сполуки та метеорологічні дані, такі як швидкість вітру та температура. Мережа складається з понад ста пунктів моніторингу, стратегічно розміщених так, щоб збирати дані з різних середовищ, включаючи узбіччя доріг, міське тло, передмістя та сільську місцевість. Таке комплексне розгортання дозволяє LAQN оцінювати як безпосередній вплив транспортних викидів, так і фонову якість повітря в мегаполісі.

Лондонська мережа якості повітря пропонує загальнодоступну вебплатформу, зображену на рисунку 2.1, на якій можна досліджувати дані про якість повітря в реальному часі та історичні дані. Вебсайт надає інтерактивні карти, графіки, звіти та поради щодо здоров'я на основі рівнів забруднення. Дані часто оновлюються, що дозволяє мешканцям, дослідникам і політикам відстежувати тенденції якості повітря майже в режимі реального часу.

Крім надання відкритого доступу до даних, LAQN активно співпрацює з академічними установами, екологічними організаціями та місцевою владою для аналізу довгострокових тенденцій забруднення та розробки ефективних стратегій зменшення його рівня. Мережа також використовується для інформування екстрених служб та розробників політик під час епізодів сильного забруднення

					ІК11.020БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

повітря. Завдяки надійності, високій роздільній здатності даних і широкому охопленню, LAQN стала одним із найавторитетніших джерел інформації про якість повітря у Великій Британії.

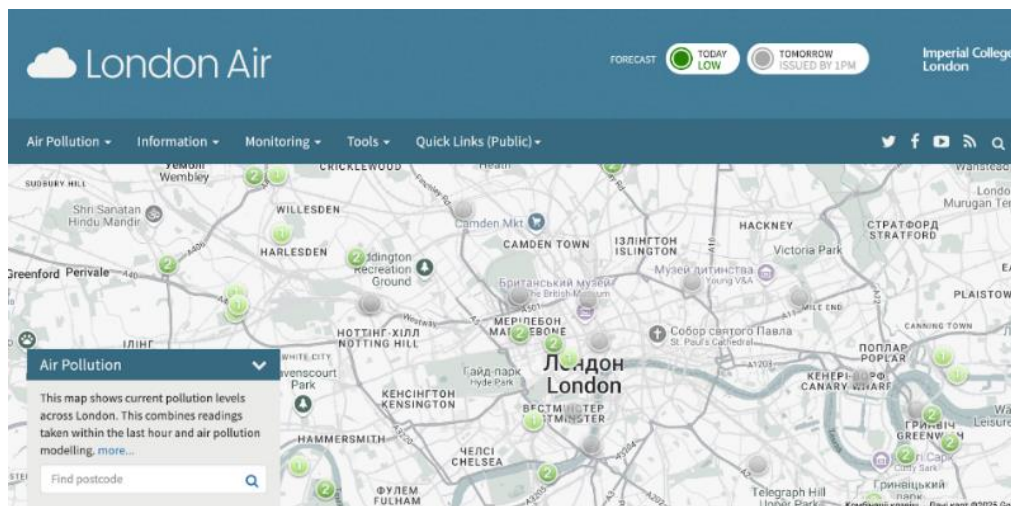


Рисунок 2.1 – Головна сторінка LAQN з інтерактивною картою [4].

Крім того, мережа підтримує зовнішній доступ до своїх наборів даних. Дослідники та установи можуть запитувати необроблені або оброблені дані для наукових досліджень чи аналізу політики. У деяких випадках доступні стандартизовані канали даних та API для підтримки інтеграції із зовнішніми додатками та публічними інформаційними службами. Така відкритість сприяє прозорості, заохочує повторне використання даних і посилює вплив моніторингу на громадську свідомість та управління довкіллям.

Також, Лондонська мережа якості повітря надає прогнози якості повітря, які передбачають рівень забруднення на день вперед. Прогнози представлені в текстовому форматі на вебсайті, пропонуючи інформацію для різних забруднювачів, таких як діоксид азоту, озон і тверді частинки. Система прогнозування покликана допомогти мешканцям, вразливим групам населення та міській владі підготуватися до періодів поганої якості повітря. Попередження та медичні рекомендації видаються, коли очікуються епізоди забруднення, що дозволяє вживати проактивних заходів, таких як коригування транспортної

діяльності або інформування вразливих груп населення щодо обмеження перебування на відкритому повітрі.

2.2.2 AirNow

AirNow — це національна система моніторингу та інформації про якість повітря, якою керує Агентство з охорони навколишнього середовища США у партнерстві з державними, місцевими та плеємними агентствами з питань якості повітря [5]. Запущена в 1998 році, AirNow була розроблена з метою надання громадськості легкого доступу до інформації про якість повітря в режимі реального часу, сприяння підвищенню екологічної обізнаності та підтримки процесів прийняття рішень як окремими особами, так і установами, що займаються питаннями забруднення повітря та здоров'я населення.

AirNow збирає, обробляє та поширює дані про якість повітря з тисяч станцій моніторингу, розташованих по всій території Сполучених Штатів, а також у деяких частинах Канади та Мексики в рамках міжнародної співпраці. Система в першу чергу вимірює концентрацію приземного озону, твердих частинок, оксиду вуглецю, діоксиду сірки і діоксиду азоту. Ці забруднювачі були обрані, оскільки відомо, що вони спричиняють значний вплив на здоров'я і регулюються Національними стандартами якості атмосферного повітря.

Однією з ключових особливостей AirNow є генерація та розповсюдження Air Quality Index – стандартизованого показника, який переводить вихідні концентрації забруднювачів у прості для розуміння категорії, починаючи від «доброго» до «небезпечного». AQI допомагає людям швидко оцінювати ризики забруднення повітря та приймати обґрунтовані рішення щодо активного відпочинку на свіжому повітрі, особливо для таких вразливих груп, як діти, люди похилого віку та люди з респіраторними захворюваннями.

AirNow пропонує користувачам кілька способів доступу до даних про якість повітря. Основною точкою доступу є вебсайт, який містить інтерактивні карти, зображені на рисунку 2.2, поточні та прогнозовані значення AQI, поради щодо

					ІК11.020БАК.006 ПЗ	Арк.
						20
Зм.	Лист	№ докум.	Підпис	Дата		

здоров'я та освітні ресурси. Крім того, AirNow надає мобільні додатки, які дозволяють користувачам перевіряти місцеві умови якості повітря та отримувати сповіщення на свої смартфони.

Мобільний додаток AirNow від EPA надає простий інтерфейс для швидкої перевірки поточної та прогнозної інформації про якість повітря для планування повсякденної діяльності та захисту вашого здоров'я. Додаток автоматично відображає поточний індекс якості повітря для вашої місцевості або будь-якої області, яку ви хочете перевірити, і дозволяє зберігати кілька областей для швидкого доступу. Додаток доступний виключно на території США.

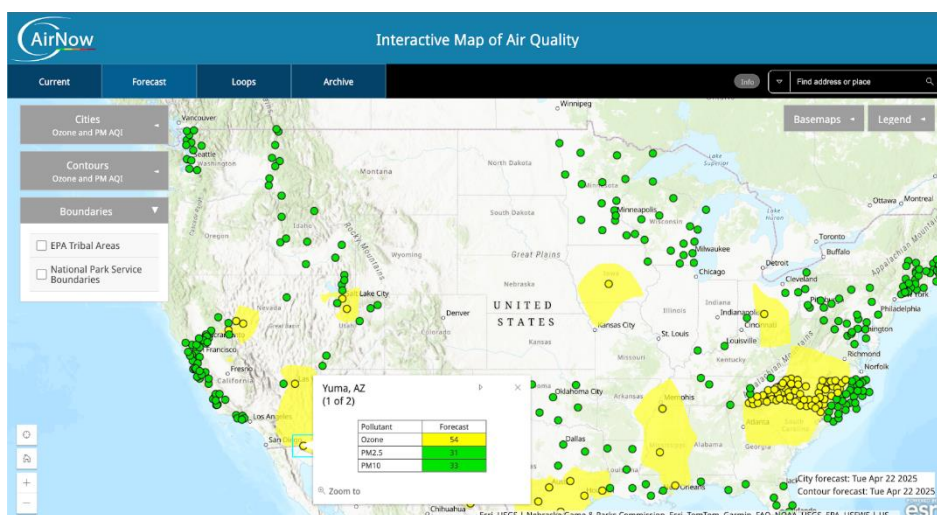


Рисунок 2.2 – Інтерактивна карта AirNow [5].

Система підтримує обмін даними з дослідниками, розробниками та партнерськими організаціями. За допомогою інтерфейсу прикладного програмування AirNow і наборів даних, які можна завантажити, зовнішні користувачі можуть отримати доступ до даних про якість повітря в реальному часі та історичних даних для подальшого аналізу або інтеграції в інші програми. Така відкритість сприяє дослідженням в галузі екології, охорони здоров'я, міського планування та підтримує розробку інноваційних рішень для вирішення проблем забруднення повітря.

AirNow також має комплексну систему прогнозування якості повітря, яка надає прогнози концентрації озону та твердих частинок по всій території США. Ці

прогнози зазвичай охоплюють період від 24 до 48 годин і публікуються щодня на вебсайті AirNow. Прогнозна інформація інтегрована в систему Індексу якості повітря, що допомагає громадськості зрозуміти очікувані умови і відповідно планувати свою діяльність. Під час лісових пожеж або інших епізодів екстремального забруднення AirNow надає спеціалізовані прогнози задимленості та поради щодо здоров'я, підтримуючи зусилля з реагування на надзвичайні ситуації та мінімізуючи ризики для здоров'я постраждалого населення.

2.3 Комерційні системи

2.3.1 PurpleAir

PurpleAir — це широко використовуване комерційне рішення для моніторингу якості повітря, яке набуло значної популярності серед приватних осіб, громадських груп, дослідників та органів місцевого самоврядування. Заснована у 2015 році, PurpleAir пропонує доступні та прості у встановленні датчики якості повітря, які забезпечують моніторинг концентрації твердих частинок у реальному часі. Однією з відмінних характеристик PurpleAir є зосередженість на створенні щільної публічної мережі даних про якість повітря, що дозволяє користувачам самостійно встановлювати датчики та вносити їхні показники на глобальну онлайн-карту.

Підхід PurpleAir до збору та обробки даних підкреслює безпосередність та відкритість. Після встановлення датчики безперервно передають вимірювання на хмарні сервери PurpleAir через Wi-Fi. Дані оновлюються в середньому кожні 80 секунд, надаючи користувачам інформацію майже в режимі реального часу. На відміну від традиційних моніторів нормативного класу, датчики PurpleAir є недорогими і розроблені для простоти використання, що, природно, призводить до певних компромісів з точки зору абсолютної точності. Однак PurpleAir у співпраці з академічними дослідниками та державними установами розробила калібрувальні поправочні коефіцієнти та алгоритми, щоб підвищити якість і порівнянність своїх даних з офіційними стандартами моніторингу.

					ІК11.020БАК.006 ПЗ	Арк.
						22
Зм.	Лист	№ докум.	Підпис	Дата		

PurpleAir пропонує кілька методів доступу до даних та їх візуалізації. Основною платформою є карта PurpleAir, інтерактивна онлайн-карта, зображена на рисунку 2.3, що відображає поточні дані з тисяч датчиків по всьому світу. Крім того, PurpleAir надає API, який дозволяє розробникам, дослідникам і державним органам отримувати дані з датчиків програмно. Ця можливість уможливорює інтеграцію із зовнішніми інформаційними панелями, платформами «розумного міста» та кастомними аналітичними інструментами.

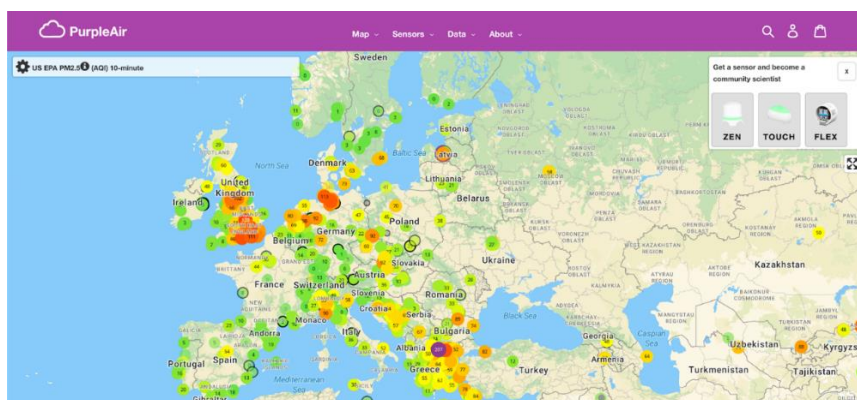


Рисунок 2.3 – Інтерактивна карта PurpleAir [6].

На додаток до публічної платформи візуалізації, PurpleAir пропонує доступ до своїх даних через надійний API. Повний доступ до API PurpleAir пропонується за допомогою платної моделі підписки. Цей комерційний API призначений для задоволення потреб дослідників, розробників, компаній та установ, які потребують систематичного доступу до великих обсягів даних про якість повітря в режимі реального часу та історичних даних.

Пропонуючи цю послугу, PurpleAir монетизує доступ до даних, а також сприяє більш широкому використанню своєї сенсорної мережі для інноваційних застосувань. Доступ до API надається через панель розробника зображену на рисунку 2.4.

API надає важливі метадані про кожен датчик, такі як:

- а) географічні координати;
- б) інформація про встановлення датчика;
- в) тип датчика та версія прошивки;

- г) рівень сигналу та стан Wi-Fi;
- д) робочий стан датчика та показники якості.

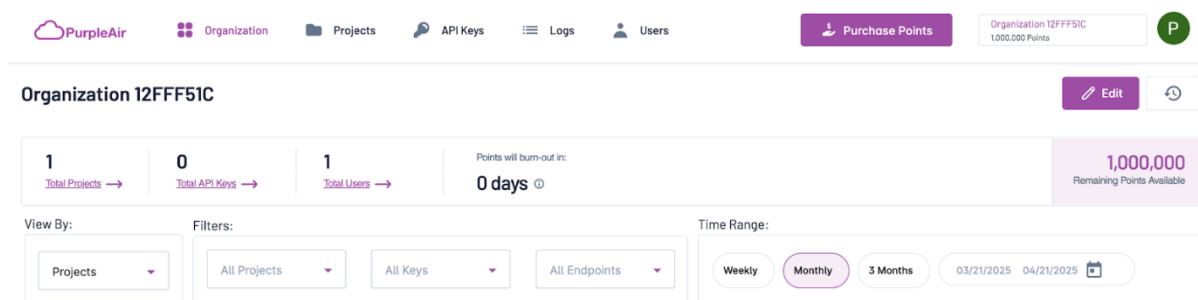


Рисунок 2.4 – Панель розробника PurpleAir [7].

Загалом, підхід PurpleAir, що поєднує в собі доступне обладнання, глобальну мережу з широкою участю користувачів та сервіс API професійного рівня, робить його унікальним у сфері моніторингу якості повітря. Хоча PurpleAir не має на меті замінити мережі моніторингу нормативного рівня, він є безцінним додатковим джерелом екологічної інформації, що підвищує обізнаність про ситуацію.

2.3.2 Kaiterra Data Platform

Kaiterra — це комерційна компанія, що спеціалізується на технологіях моніторингу навколишнього середовища, з особливим акцентом на якість повітря в приватних приміщеннях і на території [8]. Заснована у 2014 році, Kaiterra зарекомендувала себе як постачальник високоякісних рішень для моніторингу повітря корпоративного рівня, призначених для бізнесу, менеджерів будівель, державних установ та організацій. Хоча компанія продає різноманітні апаратні пристрої, такі як Kaiterra Laser Egg та серія Kaiterra Sensedge, ключовим елементом, що об'єднує їхню екосистему, є Kaiterra Data Platform — централізована платформа для візуалізації даних, управління пристроями, аналітики та оповіщення.

Kaiterra Data Platform слугує основним інтерфейсом, за допомогою якого користувачі взаємодіють зі своїми розгорнутими датчиками. Це хмарне рішення, призначене для агрегування, відображення та аналізу даних про навколишнє

					ІК11.020БАК.006 ПЗ	Арк.
						24
Зм.	Лист	№ докум.	Підпис	Дата		

середовище в реальному часі та історичних даних з декількох пристроїв одночасно. Приклад роботи зображений на рисунку 2.4. Ця платформа дозволяє керівникам об'єктів, дослідникам і фахівцям з охорони навколишнього середовища контролювати якість повітря в різних місцях, на різних поверхах або навіть у цілих комплексах будівель у централізований, масштабований спосіб.

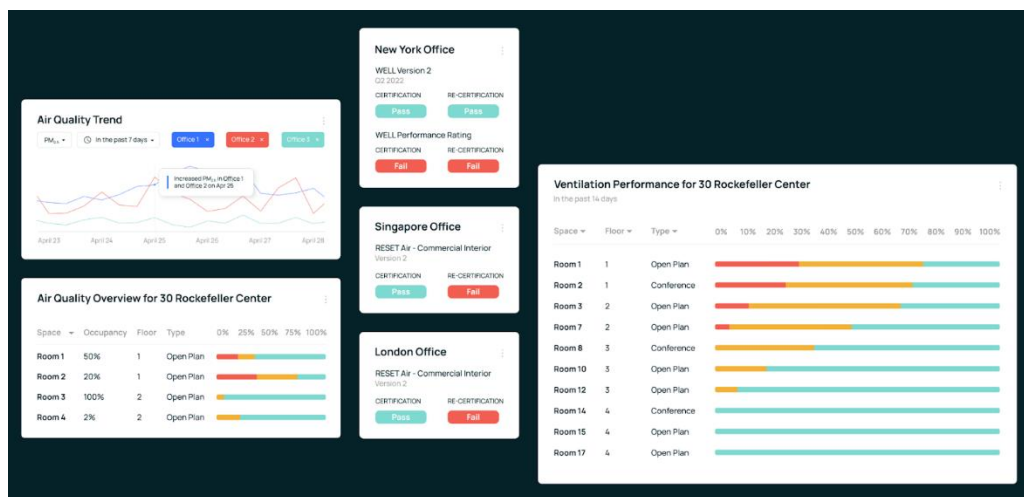


Рисунок 2.4 – Приклади роботи Kaiterra Dashboard [8].

Пристрої моніторингу Kaiterra, підключені до Dashboard, зазвичай вимірюють широкий спектр параметрів навколишнього середовища. Вони включають концентрацію твердих частинок, рівень вуглекислого газу, температуру, вологість і атмосферний тиск. Дані збираються в режимі реального часу і безпечно передаються на хмарні сервери Kaiterra через Wi-Fi або Ethernet з'єднання, в залежності від конфігурації пристрою.

Приладова панель надає функції візуалізації даних, що легко налаштовуються. Користувачі можуть переглядати потоки даних у реальному часі, історичні тренди за різні періоди часу (погодинно, щоденно, щотижнево, щомісячно) та статистичні зведення. Дані можна візуалізувати у вигляді графіків часових рядів, порівняльних діаграм з декількох датчиків та кольорових індексів якості повітря. Платформа дозволяє користувачам встановлювати індивідуальні порогові значення для різних параметрів, запускаючи сповіщення при досягненні

певних умов. Сповіщення можна надсилати електронною поштою, SMS або інтегрувати в системи управління будівлею за допомогою API.

Загалом, Kaiterra представляє професійний, масштабований і гнучкий підхід до моніторингу якості повітря. Акцент на високоякісному зборі даних, вдосконаленій візуалізації, можливостях прогнозування, системній інтеграції та зручних інтерфейсах робить її ідеальним рішенням для великомасштабних розгортань, де управління навколишнім середовищем на основі даних є критично важливим. Інформаційна панель Kaiterra Dashboard, як центральний вузол системи, є прикладом сучасних передових практик в управлінні екологічними даними та дизайні користувацького досвіду.

2.4 Публічні системи з відкритим кодом

Sensor.Community, раніше відомий як Luftdaten, — це глобальна ініціатива з відкритим вихідним кодом, що керується громадою для моніторингу якості повітря [9]. Заснований у 2015 році в Штутгарті, Німеччина, проєкт спочатку був розроблений для вирішення проблеми нестачі детальних даних про якість повітря в реальному часі, доступних для громадськості. З часом Sensor.Community перетворилася на одну з найбільших мереж громадського моніторингу повітря у світі з тисячами датчиків, розгорнутих по всій Європі, Азії, Америці та інших регіонах.

Фундаментальним принципом Sensor.Community є демократизація збору даних про навколишнє середовище. Замість того, щоб покладатися виключно на дорогі урядові станції моніторингу, проєкт дає можливість людям створювати власні недорогі датчики якості повітря і вносити дані на глобальну платформу з відкритим доступом. Ініціатива надає детальні, вільнодоступні інструкції для складання пристроїв моніторингу з використанням широкодоступних апаратних компонентів.

Sensor.Community зосереджується насамперед на вимірюванні концентрації твердих частинок які є критично важливими показниками забруднення повітря і

					ІК11.020БАК.006 ПЗ	Арк.
						26
Зм.	Лист	№ докум.	Підпис	Дата		

пов'язані зі значними ризиками для здоров'я. З часом проєкт розширився, включивши в себе додаткові вимірювання, такі як температура, вологість та атмосферний тиск, залежно від датчиків, які користувачі інтегрують у свої пристрої. Хоча основний акцент Sensor.Community залишається на забрудненні твердими частинками, його модульна конструкція дозволяє учасникам експериментувати з різними конфігураціями датчиків для моніторингу ширшого спектру параметрів навколишнього середовища.

Sensor.Community надає кілька інструментів для доступу та візуалізації зібраних даних. Основним інтерфейсом візуалізації є інтерактивна глобальна карта, зображена на рисунку 2.5, яка відображає всі активні датчики та їхні вимірювання в реальному часі. Користувачі можуть переглядати показники вмісту твердих частинок, температури та вологості, залежно від конфігурації датчиків на кожному пристрої.

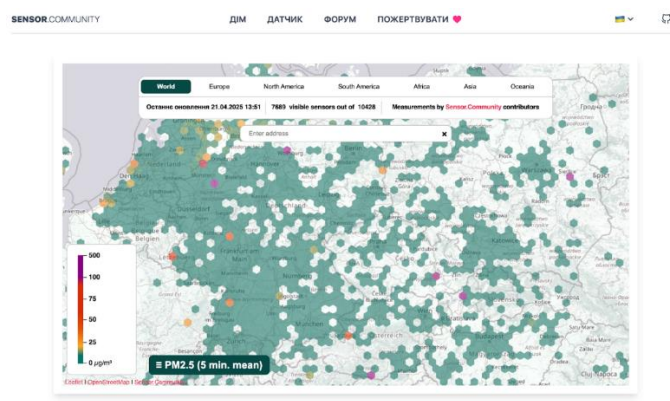


Рисунок 2.5 – Платформа Sensor.Community [9].

Оскільки пристрої Sensor.Community створюються громадянами і використовують недорогі компоненти, точність і калібрування окремих датчиків може відрізнятися. Як наслідок, дані мережі вважаються орієнтовними, а не нормативними. Тим не менш, дані Sensor.Community надають цінну інформацію про якість повітря, гарячі точки забруднення та часові тенденції, особливо в районах, які недостатньо обслуговуються офіційними станціями моніторингу.

Однією з ключових переваг Sensor.Community є його екосистема з відкритим вихідним кодом. Програмне забезпечення, що використовується для збору даних, прошивки пристроїв та візуалізації даних, є повністю відкритим і доступне в публічних репозиторія. Ця відкритість заохочує до постійного вдосконалення глобальну спільноту розробників, дослідників та екологічних активістів. Це також дозволяє мотивованим користувачам налаштовувати і розширювати систему відповідно до конкретних місцевих вимог, наприклад, додавати підтримку додаткових типів датчиків або адаптувати прошивку для різних апаратних платформ.

Крім того, Sensor.Community сприяє співпраці з науковими установами, міською владою, некомерційними організаціями та екологічними ініціативами. Кілька міст, наприклад, Ренн у Франції в інтегрували дані Sensor.Community у свої стратегії моніторингу навколишнього середовища, щоб доповнити офіційні дані. Навчальні заклади використовують проєкт як інструмент для викладання екології, розвитку Інтернету речей та залучення громадян.

Нажаль Sensor.Community не надає вбудованих функцій аналізу даних або прогнозування якості повітря на своїй платформі. Мережа зосереджується насамперед на зборі та поширенні відкритих даних, залишаючи аналіз та інтерпретацію зовнішнім користувачам, дослідникам та стороннім програмам. На відміну від професійних або державних систем моніторингу, які часто генерують прогнозні моделі або медичні рекомендації, інфраструктура Sensor.Community навмисно залишається легкою і децентралізованою

Висновки до розділу 2

Огляд існуючих систем моніторингу якості повітря демонструє широке розмаїття підходів, кожен з яких має свої сильні та слабкі сторони. Урядові мережі, такі як London Air Quality Network та AirNow, надають високоточні, перевірені дані і є надійними джерелами інформації про стан здоров'я населення. Однак ці системи є закритими за своєю природою: вони покладаються лише на офіційно

					ІК11.020БАК.006 ПЗ	Арк.
						28
Зм.	Лист	№ докум.	Підпис	Дата		

сертифіковані станції моніторингу, а приватні особи не можуть підключати власні датчики. Хоча LAQN і AirNow пропонують хорошу перевірку даних і надійні функції прогнозування, їхня доступність для незалежного використання обмежена, а дані, як правило, інтегровані в стаціонарні урядові платформи без особливої гнучкості для сторонніх додатків.

Комерційні рішення, такі як PurpleAir і Kaiterra, забезпечують більшу гнучкість для користувачів, але створюють інші проблеми. PurpleAir дозволяє приватним особам купувати датчики і приєднуватися до децентралізованої мережі, пропонуючи відкриту карту і платний доступ до API. Однак, системі не вистачає вбудованого прогнозування та аналізу даних, окрім базових корекцій, а функції оповіщення в реальному часі обмежені без зовнішніх інтеграцій.

Kaiterra більше орієнтується на корпоративних клієнтів, пропонуючи професійні інформаційні панелі, предиктивну аналітику та настроювані сповіщення. Проте, участь обмежена клієнтами, які використовують власні пристрої Kaiterra, а середовище даних є закритим, що робить пріоритетним організаційний контроль над обміном відкритими даними.

Ініціативи з відкритим вихідним кодом, такі як Sensor.Community, представляють іншу філософію, просуваючи громадянську науку і повну відкритість. Люди можуть створювати власні недорогі датчики і вільно передавати дані на глобальну платформу. Проект підтримує відкритий доступ до даних у реальному часі та історичних даних і заохочує залучення громадськості. Однак, Sensor.Community не має вбудованих функцій прогнозування, розширеного аналізу даних або нативних сповіщень. Дані надаються в необробленому вигляді, що вимагає від користувачів власної обробки або створення зовнішніх систем для візуалізації та сповіщення.

3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

3.1 Виставлення вимог до системи

Провівши ретельний аналіз наявних на ринку рішень, стало очевидно, що багато з них мають суттєві обмеження в плані інтеграції з зовнішнім обладнанням. Зокрема, більшість комерційних систем моніторингу не дозволяють користувачам підключати власні датчики, що значно звужує їхню функціональність та ускладнює адаптацію до конкретних потреб. У зв'язку з цим сформувався чіткий запит на систему, яка забезпечує відкриту архітектуру і гнучкість у підключенні будь-яких сенсорів.

Окрему увагу вимагає функціональність аналізу даних у реальному часі. Поряд із цим, важливою вимогою є збереження історичних даних: з одного боку, вони слугують основою для тренування моделей машинного навчання, а з іншого — є джерелом для постаналізу ситуацій, що вже відбулися.

Ще одним критично важливим аспектом є візуалізація. Для подібного типу систем візуальне подання інформації не просто покращує сприйняття — воно є необхідною умовою для швидкого прийняття рішень користувачем. Тому система повинна містити інтерфейс, що дозволяє легко інтерпретувати як поточні показники, так і результати прогнозів.

Беручи до уваги специфіку проекту та цілі демонстрації, реалізація орієнтована на автономну роботу в межах одного користувача без потреби в масштабуванні чи розгортанні в розподіленому середовищі. Це дозволяє зосередитися на якості реалізації основної функціональності, не ускладнюючи архітектуру зайвими зовнішніми залежностями.

3.2 Базові вимоги

Формуючи архітектуру та функціональні можливості системи, перш за все слід визначити набір ключових вимог, без реалізації яких її робота була б неповною або неефективною. Ці вимоги відображають базову функціональність, що

					ІК11.020БАК.006 ПЗ	Арк.
						30
Зм.	Лист	№ докум.	Підпис	Дата		

необхідна для забезпечення основних сценаріїв використання та відповідності сучасним очікуванням до систем моніторингу якості повітря. Вони також охоплюють як технічні, так і аналітичні аспекти, які забезпечують цілісну роботу проєкту.

До базових вимог належать:

- а) можливість обробки та аналізу телеметричних даних у режимі реального часу;
- б) функція прогнозування якості повітря з використанням машинного навчання на найближчу годину;
- в) модуль виявлення аномальних значень, що дозволяє своєчасно реагувати на потенційно небезпечні ситуації;
- г) система сповіщення користувача у випадку перевищення допустимих порогів;
- д) підтримка підключення довільних сенсорів без прив'язки до конкретного виробника чи типу пристрою;

3.3 Додаткові вимоги

Окрім основного функціоналу, що забезпечує базову працездатність системи, важливо передбачити можливості, які значно покращують досвід користувача та підвищують практичну цінність рішення. Ці додаткові функції не є критично необхідними для запуску системи, однак їх наявність дозволяє наблизити продукт до повноцінного інструменту моніторингу та аналізу довкілля.

Особливу увагу приділено візуальному представленню даних, яке виконує не лише інформативну, але й аналітичну функцію. Завдяки зручному графічному інтерфейсу користувач отримує змогу миттєво оцінювати ситуацію, виявляти закономірності, спостерігати за змінами в динаміці, що істотно підвищує ефективність використання системи в повсякденному контексті та загалом підвищить ефективність боротьби із забрудненням повітря.

До додаткових вимог належать:

					ІК11.020БАК.006 ПЗ	Арк.
						31
Зм.	Лист	№ докум.	Підпис	Дата		

а) інтерактивна візуалізація результатів прогнозування якості повітря у вебзастосунку в режимі реального часу;

б) відображення історичних даних із можливістю навігації, масштабування та аналізу динаміки змін у зручному графічному вигляді.

3.4 Обмеження системи

Під час проєктування та реалізації системи було свідомо обрано низку обмежень, які дозволили зосередитися на якості основного функціоналу, водночас зберігаючи чіткість меж застосування. Ці обмеження визначають рамки, в яких функціонує система, і пояснюють деякі архітектурні та технічні рішення.

Оскільки ціллю розробки є створення прототипу, що ілюструє повний цикл роботи, було вирішено працювати з обмеженим, але якісно підготовленим набором даних. Для моделювання використовувалися великі обсяги реальних даних, попередньо оброблені для досягнення репрезентативності: доповнені штучно змодельованими аномаліями, приведені до єдиного формату. Джерелом слугували відкриті платформи, як Kaggle, що дозволило отримати різноманітну вибірку без необхідності розгортання повноцінної сенсорної мережі [10].

Також система не інтегрується з зовнішніми сервісами доставки сповіщень, такими як мобільні додатки, email чи SMS. Повідомлення про аномалії та прогнози надаються у вигляді локального відображення, що водночас спрощує архітектуру та дозволяє уникнути залежності від сторонніх платформ. Це рішення було обрано з огляду на демонстраційний характер розробки та орієнтацію на автономну роботу в межах одного користувача.

3.5 Архітектура системи

На кресленику ІК11.020БАК.006 Д1, що зображає функціональну схему системи, представлено логічну архітектуру взаємодії між окремими модулями, які працюють у реальному часі. Система зроблена з урахуванням модульного підходу,

					ІК11.020БАК.006 ПЗ	Арк.
						32
Зм.	Лист	№ докум.	Підпис	Дата		

що дозволяє кожному компоненту виконувати вузькоспеціалізовану функцію, водночас забезпечуючи цілісну й злагоджену роботу через повідомлення, які передаються за допомогою брокера. Така схема дає змогу досягнути високої гнучкості, масштабованості та відмовостійкості.

3.5.1 Модуль надсилання телеметричних даних

Цей компонент виконує роль точки входу для сенсорної інформації, зокрема значень PM2.5, температури та вологості. Він формує та передає повідомлення з підготовленими даними у брокер повідомлень з заданою періодичністю. Забезпечується постійний потік актуальної інформації, яка є базою для подальшої обробки та аналізу.

У рамках розробки дані надходитимуть не з реальних сенсорів, а з попередньо підготовленого CSV-файлу, що містить синхронізований часовий ряд з усіма необхідними параметрами. Це дозволяє створювати поведінку реального середовища та тестувати інші компоненти системи без потреби в апаратному обладнанні.

3.5.2 Модуль аналізу

Модуль аналізу відповідає за попередню обробку вхідних даних та формування агрегованих щогодинних метрик. Його завдання — структурувати дані, виявити потенційні відхилення та згенерувати статистичні показники, які дозволяють зробити оцінку загального стану повітря. Зокрема, модуль обчислює середні значення температури, вологості та концентрації PM2.5 за кожен годину, що дозволяє згладити короткочасні коливання та зосередитися на загальній динаміці змін.

Ключовою функцією модуля також є розрахунок AQI — індексу якості повітря, який є стандартизованою шкалою для інтерпретації рівня забруднення. Значення AQI обчислюється на основі середньої концентрації PM2.5 і відображає,

					ІК11.020БАК.006 ПЗ	Арк.
						33
Зм.	Лист	№ докум.	Підпис	Дата		

наскільки повітря є безпечним для здоров'я. В межах проєкту AQI використовується як універсальний індикатор, що дозволяє швидко оцінити якість повітря.

Результати аналізу — включно з агрегованими показниками надсилаються до відповідної черги повідомлень через брокер, а також зберігаються у базі даних для подальшої візуалізації, прогнозування та формування сповіщень.

3.5.3 Модуль прогнозування

Цей модуль здійснює прогнозування значень параметрів на найближчу годину. Для цього він використовує отримані раніше значення та моделі машинного навчання, що були натреновані на історичних даних. Прогноз надсилається через брокер у відповідну чергу та зберігається для подальшої візуалізації.

3.5.4 Модуль оповіщення

Після того як дані проходять аналіз або прогноз, вони перевіряються на відповідність критичним значенням. У разі виявлення аномалій модуль формує сповіщення, яке надсилається користувачеві. Завдяки цьому користувач оперативно отримує інформацію про небезпечні рівні забруднення або інші позаштатні ситуації.

3.5.5 Користувацький інтерфейс

Цей компонент забезпечує доступ до інформації для кінцевого користувача. Через вебінтерфейс він може переглядати як поточні прогнози у реальному часі, так і історичні дані, включаючи графіки та метрики. Інтерфейс також може містити елементи для навігації та масштабування даних.

3.5.6 Брокер повідомлень

					ІК11.020БАК.006 ПЗ	Арк.
						34
Зм.	Лист	№ докум.	Підпис	Дата		

Серцевина комунікаційної інфраструктури системи. Брокер повідомлень забезпечує передачу інформації між модулями, гарантує надійність доставки та дозволяє модулю кожного типу працювати асинхронно. Середовище черг — вхідні дані, прогнози та аналітичні метрики — дозволяє організувати обробку як незалежних потоків, мінімізуючи затримки та підвищуючи масштабованість архітектури.

3.6 Діграма потоку даних

На кресленику ІК11.020БАК.006 Д2 зображено загальний потік даних у межах системи. Система починається з передачі сирих даних від датчиків, які надходять до черги вхідних даних. Далі ці дані паралельно обробляються модулями аналізу та прогнозування. Аналіз генерує зведені метрики, які зберігаються в базі даних і передаються також у чергу для генерації сповіщень. Прогнозування, своєю чергою, створює прогноз на основі тих самих вхідних даних і також записує результати у відповідну чергу та базу даних.

База даних виступає центральним елементом зберігання інформації: як історичних метрик, так і прогнозів. На основі цих даних формується користувацький інтерфейс, який відображає графіки та сторінки для кінцевого користувача.

Висновки до розділу 3

Проведений аналіз потреб користувачів та існуючих рішень у сфері моніторингу повітря дозволив сформулювати чіткі вимоги до функціональності, архітектури та обмежень розроблюваної системи. Було виявлено, що на ринку переважають закриті або малогнучкі системи, які не дозволяють підключати довільні сенсори, що істотно обмежує можливості адаптації під специфічні умови

застосування. Відтак, відкритість і модульність стали одним із ключових принципів побудови системи.

У процесі формування вимог акцент був зроблений на обробці та аналізі даних у реальному часі, підтримці виявлення аномалій та можливості побудови короткострокових прогнозів. Такі можливості відповідають сучасним трендам у галузі аналітики навколишнього середовища та є необхідними для ефективного реагування на зміни в якості повітря. Візуалізація даних у зрозумілому графічному вигляді — ще один важливий компонент, який значно полегшує взаємодію користувача з системою, дозволяючи йому швидко оцінювати поточний стан або переглядати історичні зміни.

Запропонована модульна архітектура з використанням брокера повідомлень забезпечує гнучку й надійну комунікацію між компонентами. Система охоплює всі етапи роботи з даними — від надсилання телеметрії до генерації прогнозів і візуалізації. Це дозволяє не тільки відслідковувати поточний стан повітря, а й аналізувати довгострокові тенденції та потенційні ризики.

Сформовані вимоги та обрана архітектура відображають як потреби користувачів, так і технічні реалії побудови сучасних систем моніторингу. Вони забезпечують надійне підґрунтя для реалізації ефективного, адаптивного та практично значущого рішення.

4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

4.1 Вибір технологій

Враховуючи сформовані функціональні та нефункціональні вимоги до системи, ключовим етапом є вибір технологій, що ляжуть в основу архітектури проєкту. Насамперед необхідно визначити ядро обміну повідомленнями — брокер повідомлень, який забезпечить асинхронну, надійну та масштабовану взаємодію між модулями системи. Він має підтримувати високу пропускну здатність, низьку затримку та можливість горизонтального масштабування, оскільки система працює в режимі реального часу з потенційною підтримкою великої кількості джерел даних.

Наступним етапом є вибір мови програмування, яка має забезпечити ефективну інтеграцію з обраним брокером повідомлень, а також мати зрілу екосистему для роботи з телеметричними даними, зокрема — для побудови моделей машинного навчання, їхнього тренування та застосування у реальному часі.

Для користувацького інтерфейсу система потребує сучасного вебфреймворку з розвиненою підтримкою інтерактивної візуалізації даних. Інтерфейс повинен бути інтуїтивно зрозумілим, надавати доступ до історичних даних, результатів аналізу та прогнозів у режимі реального часу. Тому особлива увага приділяється наявності бібліотек для побудови графіків та інтуїтивної взаємодії з користувачем.

4.2 Брокер повідомлень

Оскільки система побудована на принципах асинхронної комунікації між модулями, ключовим компонентом її архітектури є брокер повідомлень. Він виконує роль транспортного шару для передачі телеметричних даних, прогнозів, аналітики та системних сповіщень. Серед численних доступних рішень для реалізації такої взаємодії, було проаналізовано два найпоширеніші варіанти — Apache Kafka та RabbitMQ.

					ІК11.020БАК.006 ПЗ	Арк.
						37
Зм.	Лист	№ докум.	Підпис	Дата		

Apache Kafka вирізняється своєю орієнтацією на обробку потоків даних у режимі реального часу [11]. Це платформа, яка з самого початку розроблялась для високонавантажених систем, де важлива не лише передача, а й зберігання даних. Її архітектура дозволяє зберігати повідомлення у вигляді послідовних логів, що відкриває можливість для їх повторної обробки або зворотного аналізу. Kafka легко масштабується, як у напрямку кількості джерел та споживачів, так і в обсягах даних. Це рішення часто використовується у великих аналітичних системах, де необхідна стабільність, швидкість і надійність. Водночас, варто відзначити, що впровадження Kafka потребує глибшого розуміння її моделі роботи, правильної конфігурації та моніторингу, що може бути надмірним для проєктів невеликого масштабу.

RabbitMQ, на відміну від Kafka, є класичним брокером повідомлень із багаторічною історією використання у найрізноманітніших сферах [12]. Його сильною стороною є простота інтеграції, зрозуміла логіка черг та маршрутизації, а також гнучкість у роботі з повідомленнями. RabbitMQ підтримує кілька типів обміну, які дозволяють будувати як прості, так і складні сценарії доставки. Завдяки зручному інтерфейсу адміністратора, широкій клієнтській підтримці та активному ком'юніті, його можна швидко розгорнути й налаштувати для потреб середньої системи. Проте в контексті потокової обробки великих обсягів даних RabbitMQ може бути менш ефективним. Його модель базується на збереженні повідомлень у чергах, а не на їхньому логічному історичному потоці, тому для задач з акцентом на аналіз даних у часі він менш придатний.

Вибір між Kafka та RabbitMQ зводиться до балансу між масштабістю, потребою в історичних даних та складністю впровадження. У системах, орієнтованих на інтенсивну обробку стрімінгових даних, Kafka виглядає як більш довготривале та перспективне рішення, тоді як RabbitMQ залишається ефективним варіантом для простіших сценаріїв передачі повідомлень. В нашому випадку краще підійде Kafka.

4.3 Технології машинного навчання

					ІК11.020БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		38

Оскільки одним із ключових модулів системи є модуль прогнозування, вибір технології для реалізації машинного навчання має стратегічне значення. Обрана мова програмування має забезпечити не лише простоту побудови та навчання моделей, але й зручну інтеграцію результатів у загальну інфраструктуру системи.

Для порівняння були обрані такі мови програмування:

- а) Python;
- б) R.

Python є найбільш поширеним вибором у сфері машинного навчання. Його синтаксис зручний як для швидкого прототипування, так і для побудови складних систем [13]. У цієї мови надзвичайно багата екосистема рішень, яка охоплює всі аспекти роботи з моделями: від обробки даних до аналізу результатів та їхнього впровадження в продуктивне середовище. Більшість сучасних досліджень та практичних кейсів у сфері штучного інтелекту реалізуються саме на Python, що сприяє появі нових інструментів та швидкому вирішенню можливих проблем за рахунок активної спільноти.

R, зі свого боку, є потужною мовою для статистичного аналізу та академічних досліджень [14]. Вона демонструє високу ефективність у роботі з табличними даними та зручна для візуального представлення результатів аналізу. Однак у контексті побудови масштабованих систем із компонентами реального часу, а також у питанні інтеграції з іншими модулями — R поступається Python. Крім того, вибір готових рішень у R є значно обмеженим, а інструменти машинного навчання — менш уніфіковані.

У межах створення модуля прогнозування було розглянуто два принципово різні підходи до обробки часових рядів — ARIMA, як класичний статистичний метод, та LSTM, як представник сучасних нейромережових архітектур. Обидва методи є добре відомими й активно використовуються в задачах прогнозування, однак їхня природа, гнучкість і застосовність у складних системах суттєво відрізняються.

ARIMA (Autoregressive Integrated Moving Average) — це підхід, заснований на ідеї того, що майбутні значення ряду можуть бути змодельовані як лінійна

комбінація його попередніх значень та похибок. Його основна перевага полягає в простоті, аналітичній прозорості та досить точному прогнозуванні в умовах добре структурованих даних. Цей метод відносно легко інтерпретується, а його параметри мають зрозумілу економічну або фізичну інтерпретацію. Однак ARIMA вимагає стаціонарності ряду, тобто відсутності трендів і сезонності без попереднього перетворення. Крім того, він погано масштабований: при появі нових змінних, складних взаємодій або в разі зміни патернів у часі модель швидко втрачає релевантність.

LSTM (Long Short-Term Memory), своєю чергою, належить до рекурентних нейронних мереж і здатна «запам'ятовувати» довгі послідовності змін, не втрачаючи стабільності. Це робить її особливо ефективною у ситуаціях, де залежності між значеннями простягаються далеко в минуле, або коли динаміка системи змінюється в часі. LSTM не потребує явно стаціонарного ряду, добре справляється з шумами, нестабільностями, нелінійними зв'язками та зміною масштабу. Вона вільна від суворих математичних припущень і навчається напряму на даних, автоматично оптимізуючи внутрішні параметри відповідно до спостережуваних закономірностей. Проте така гнучкість має свою ціну — моделі LSTM складніші у налаштуванні, вимагають більше обчислювальних ресурсів і не надають інтерпретованих коефіцієнтів, що може бути недоліком у задачах, де важливо розуміти логіку прогнозу.

Загалом, ARIMA може бути корисною для швидкого створення базових моделей або коли потрібна прозора, пояснювана аналітика. Але для складніших сценаріїв, коли модель має адаптуватися до змін, виявляти приховані тренди та навчатися з часом, нейромережевий підхід із використанням LSTM є значно ефективнішим і перспективнішим.

З урахуванням поставлених завдань і бажаної гнучкості системи, було прийнято рішення використовувати LSTM як основну архітектуру для модуля прогнозування. У якості фреймворку обрано TensorFlow, що забезпечує стабільну платформу для побудови та навчання моделей, а також підтримує серіалізацію моделей для подальшого використання у продуктивному середовищі [15]. Це

					ІК11.020БАК.006 ПЗ	Арк.
						40
Зм.	Лист	№ докум.	Підпис	Дата		

рішення дозволяє надалі масштабувати модель, удосконалювати її або комбінувати з іншими підходами, зберігаючи при цьому цілісність архітектури.

4.4 Основна мова програмування

Система складається з кількох функціональних модулів, які виконують різні завдання — від збору телеметрії до прогнозування та аналітики. Вибір основної мови програмування має забезпечити не лише технічну сумісність між модулями, а й зручність підтримки, розширюваність і швидкість розробки.

У теорії, для реалізації логіки кожного з модулів могли б бути використані такі мови:

- а) C#;
- б) JavaScript;
- в) Python.

Усі вони є зрілими, мають широку підтримку з боку спільноти та здатні реалізовувати необхідний функціонал. C# добре підходить для побудови структурованих серверних додатків із високою продуктивністю та має потужні засоби розробки в середовищі .NET [16]. JavaScript, у поєднанні з Node.js, дозволяє швидко будувати легкі серверні сервіси, особливо якщо йдеться про взаємодію з вебінтерфейсом або просту обробку подій у реальному часі [17].

Однак, враховуючи, що модуль машинного навчання реалізовуватиметься саме на Python, доцільно обрати цю мову як основну для всієї системи. Такий підхід забезпечить єдність середовища розробки, спростить інтеграцію між модулями, а також дозволить повторно використовувати утиліти для обробки даних у різних частинах системи. Python має багатий набір інструментів для роботи з даними, що робить його особливо зручним у контексті систем, де аналітика, обробка та передача даних відбуваються в режимі реального часу.

Модуль аналізу метрик, що агрегує статистику за певний період, модуль оповіщення, який реагує на аномальні значення, а також модуль надсилання телеметричних даних — усі ці компоненти можуть бути ефективно реалізовані на

Python без втрати продуктивності або зростання складності. Вибір Python як основної мови програмування для всіх внутрішніх модулів є логічним, обґрунтованим і стратегічно доцільним.

4.5 Користувацький інтерфейс

Користувацький інтерфейс є важливою складовою системи, адже саме через нього здійснюється доступ до результатів аналізу, прогнозів і історичних даних. Інтерфейс повинен бути не лише візуально зрозумілим, а й технічно надійним, здатним працювати з даними в режимі реального часу та забезпечувати інтерактивну взаємодію з користувачем.

4.5.1 Вибір вебфреймворку

У сучасній розробці вебінтерфейсів використання лише HTML, CSS та JavaScript без фреймворків вважається застарілим підходом. Такий підхід не забезпечує достатньої гнучкості, ускладнює масштабування, унеможливорює повторне використання компонентів і не гарантує безпеки при роботі з динамічними даними. Для реалізації складних інтерфейсів, які мають спілкуватися з базою даних, рендерити графіки, оновлюватися в реальному часі — потрібне використання сучасного фреймворку. Серед найпопулярніших варіантів сьогодні домінують React та Angular.

React — це бібліотека для побудови інтерфейсів, яка вирізняється своєю гнучкістю та компонентною архітектурою [18]. Вона дозволяє розбити інтерфейс на незалежні частини, кожна з яких керує власним станом і логікою. React має низький поріг входу та активно підтримується спільнотою, що гарантує доступ до великої кількості допоміжних бібліотек, прикладів і шаблонів. Однією з важливих переваг є глибока інтеграція з TypeScript, що дозволяє використовувати статичну типізацію та покращує контроль за даними у додатку. React не нав'язує жорсткої структури, тому підходить як для простих, так і для складних застосунків.

					ІК11.020БАК.006 ПЗ	Арк.
						42
Зм.	Лист	№ докум.	Підпис	Дата		

Angular є повноцінним фреймворком з вбудованими засобами для маршрутизації, роботи з формами, валідації, керування станом і багато іншого [19]. Його основна сила — у строгості структури, що добре підходить для великих команд і корпоративних проєктів. Angular створювався з розрахунком на TypeScript, тому весь фреймворк побудований з урахуванням типізації з самого початку. Разом із цим Angular має вищу складність у порівнянні з React, потребує більше часу на налаштування та навчання, а також передбачає більший обсяг коду для досягнення тих самих цілей.

У рамках цього проєкту було вирішено уникати створення окремого API-сервера між фронтом і базою даних. Такий підхід спрощує архітектуру, зменшує кількість компонентів і пришвидшує розробку. Замість класичного клієнт-серверного розділення застосовується архітектурна модель із вбудованою серверною логікою, яку дозволяє реалізувати фреймворк Next.js — надбудова над React [20].

У цьому підході серверна логіка реалізується як частина вебдодатку. Вона дозволяє безпосередньо звертатися до бази даних із React-додатку через серверні функції, які виконуються на стороні сервера, а не в браузері. Це забезпечує безпечний доступ до даних, контроль над їх обробкою та гнучкість у реалізації бізнес-логіки. Немає потреби створювати окремий API, що суттєво спрощує структуру системи.

4.5.2 Вибір додаткових бібліотек

Окрім базових технологій, для створення повноцінного користувацького інтерфейсу виникла потреба у використанні додаткових бібліотек, які спрощують реалізацію складних елементів інтерфейсу, зокрема графічної візуалізації та стилізації. Правильний вибір таких інструментів суттєво впливає на швидкість розробки, якість взаємодії з користувачем та підтримуваність проєкту в майбутньому.

4.5.2.1 Бібліотека для графіків

Візуалізація даних у вигляді графіків є однією з центральних функцій інтерфейсу, оскільки користувач має мати змогу оперативно оцінювати зміну показників у часі. Для реалізації цієї функції було розглянуто два варіанти:

- а) plotly;
- б) chart.js;

Plotly пропонує високу якість візуалізації «з коробки», має багато готових типів графіків і добре підходить для наукових або презентаційних цілей [21]. Проте у контексті проекту, де потрібна адаптивна інтеграція графіків у загальний дизайн, гнучке налаштування візуального стилю та тісна взаємодія з інтерфейсними подіями, Plotly виявляється менш зручним. Його структура менш придатна для компонентного підходу, який використовується у React.

Chart.js, натомість, дозволяє з більшим контролем налаштовувати вигляд, поведінку та інтеграцію графіків [22]. Він краще поєднується з компонентною логікою, дозволяє динамічно оновлювати дані та вигляд графіка у відповідь на зміни стану додатку. Це забезпечує не лише кращу інтеграцію у структуру React-додатку, а й більшу керованість в умовах реального часу.

4.5.2.2 Бібліотека для стилізації інтерфейсу

Ще одним важливим архітектурним рішенням стало використання утилітарної бібліотеки для стилізації замість традиційного CSS. Було вирішено застосовувати Tailwind CSS — інструмент, який дозволяє створювати адаптивні, сучасні інтерфейси без необхідності написання окремих CSS-файлів [23]. Tailwind базується на підході атомарних класів, де стилі задаються безпосередньо в HTML-структурі компонентів.

Такий підхід забезпечує більшу узгодженість стилів, пришвидшує розробку, полегшує рефакторинг та усуває потребу у підтримці окремих глобальних таблиць стилів. Крім того, він дозволяє одразу бачити кінцевий результат без зайвих

					ІК11.020БАК.006 ПЗ	Арк.
						44
Зм.	Лист	№ докум.	Підпис	Дата		

переключень між HTML і CSS, що особливо корисно при створенні прототипів і роботі над адаптивністю.

4.6 База даних

База даних є центральним елементом зберігання всієї інформації системи — від телеметричних даних, що надходять кожні кілька хвилин, до згенерованих прогнозів і аналітичних метрик. Особливістю цієї системи є активна робота з часовими рядами та збереження прогнозованих значень, які можуть мати форму масивів. Відповідно, при виборі бази даних важливо враховувати не лише продуктивність, але й гнучкість у представленні структури даних, можливість ефективного агрегування, фільтрації та історичного перегляду.

Було розглянуто:

- а) Microsoft SQL Server;
- б) PostgreSQL;
- в) MongoDB.

Microsoft SQL Server — це комерційна платформа з широким набором функцій, орієнтована на великі корпоративні рішення [24]. Вона забезпечує стабільну продуктивність, має зручні інструменти адміністрування та добре підходить для класичних реляційних структур. Однак використання SQL Server передбачає ліцензійні витрати, а також прив'язку до екосистеми Microsoft. У контексті відкритого, експериментального або академічного проекту це створює зайві обмеження. Крім того, робота з часовими рядами потребує додаткової оптимізації вручну, а підтримка зберігання структурованих масивів у складеному вигляді є обмеженою.

PostgreSQL — це повністю безкоштовна, відкрита база даних, яка за функціональністю не поступається комерційним аналогам [25]. Вона підтримує розширені типи даних, зокрема масиви, що особливо корисно у контексті збереження прогнозів, які складаються з кількох значень на майбутній період. PostgreSQL також добре справляється зі зберіганням часових рядів — вона

дозволяє зручно фільтрувати дані за часовими інтервалами, виконувати агрегації та індексувати за часовими мітками. За потреби її можна розширити за допомогою спеціалізованих розширень для time series-аналітики. У контексті проєкту PostgreSQL виглядає як найоптимальніше рішення — безкоштовне, масштабоване, зрозуміле у використанні та з достатнім рівнем гнучкості.

MongoDB — документно-орієнтована база даних, яка зберігає дані у вигляді JSON-подібних об'єктів [26]. Вона безкоштовна, хоча розробляється та підтримується комерційною компанією. MongoDB часто використовується для швидкої розробки, особливо тоді, коли структура даних змінюється динамічно. У MongoDB можливо зберігати масиви. Хоча вона має механізми для зберігання time series, вони менш гнучкі й продуктивні в порівнянні з рішеннями на базі класичних реляційних моделей. Для задач, де важливо опрацьовувати послідовності значень у часі та проводити складні агрегації, MongoDB може виявитися менш зручною.

З огляду на специфіку даних, необхідність зберігання як часових рядів, так і масивів прогнозів, а також бажання уникнути ліцензійної залежності, PostgreSQL виглядає як найбільш збалансоване рішення — надійне, гнучке та технологічно відповідне задачам цього проєкту.

4.7 Додаткові інструменти

Для зручності розробки, тестування та моніторингу системи було обрано низку допоміжних інструментів, які не є частиною основної логіки, але суттєво впливають на ефективність реалізації проєкту.

Ключовим інструментом для локального розгортання є Docker [27]. Його використання дозволяє запускати окремі компоненти системи — базу даних, брокер повідомлень, вебдодаток — у відокремлених контейнерах із чітко визначеним середовищем. Це гарантує стабільність і повторюваність конфігурації на будь-якому етапі розробки та дозволяє запускати всю систему як єдине ціле одним набором команд.

Для роботи з базою даних буде застосовано pgAdmin, який надає графічний інтерфейс для перегляду таблиць, написання SQL-запитів, аналізу структури даних і спостереження за поточним станом з'єднань [28]. Це особливо корисно під час розробки, коли необхідно швидко перевірити, які саме дані були збережені та як вони обробляються.

Для спостереження за брокером повідомлень використовуватиметься Kafka UI — інструмент, що надає доступ до топіків, показує останні повідомлення, дає змогу аналізувати споживачів та перевіряти логіку комунікації між модулями [29]. Це дозволить виявляти помилки в маршрутизації або відставання модулів при обробці даних у режимі реального часу.

Управління змінами та контроль версій здійснюватиметься за допомогою Git [30]. Це дозволить зберігати історію змін, вести роботу в ізольованих гілках, синхронізувати оновлення між компонентами системи та забезпечити стабільність коду на кожному з етапів.

Для розробки вибрано Visual Studio Code [31]. Це гнучке середовище, яке дозволяє одночасно працювати з кількома терміналами, запускати паралельно серверну й клієнтську частину, редагувати код на TypeScript і Python, а також інтегруватися з Docker і Git. Такий підхід забезпечує повноцінний контроль над системою під час її побудови.

Висновки до розділу 4

У результаті аналізу вимог та технічних особливостей проєкту було сформовано узгоджений технологічний стек, який забезпечує ефективну, масштабовану та зручну для розробки архітектуру системи.

Для організації взаємодії між модулями обрано брокер повідомлень Apache Kafka, який дозволяє передавати дані в режимі реального часу, зберігати повідомлення у вигляді потоків і масштабувати систему без втрати продуктивності. Альтернативний варіант — RabbitMQ — було відхилено через меншу відповідність стрімінговій природі задачі.

Основною мовою програмування системи обрано Python, оскільки вона має найширшу екосистему інструментів для обробки даних та реалізації моделей машинного навчання. Python забезпечить реалізацію всіх основних модулів: надсилання телеметричних даних, аналізу метрик, прогнозування та оповіщення.

Для реалізації модуля прогнозування також використано Python, як найгнучкішу мову з найбільшою кількістю рішень для задач машинного навчання. Цей вибір зумовлений не лише широкою підтримкою з боку спільноти, але й наявністю потужних фреймворків, зокрема TensorFlow, що дозволяють реалізовувати сучасні архітектури нейронних мереж. У рамках системи було прийнято рішення зосередитися на використанні моделі типу LSTM, яка найбільш ефективно справляється з прогнозуванням часових рядів у складних динамічних умовах.

У ролі бази даних обрано PostgreSQL, яка є безкоштовною, відкритою та гнучкою платформою. Вона підтримує зберігання масивів, що необхідно для збереження результатів прогнозування, а також добре працює з таймсеріями, що є ключовим типом даних у системі.

Користувацький інтерфейс буде реалізовано з використанням React у поєднанні з Next.js, що дозволяє поєднати клієнтську та серверну логіку в межах одного проєкту. Основною мовою інтерфейсу є TypeScript, що забезпечує типізацію, надійність та зручність розробки. Було вирішено не створювати окремого API для взаємодії з базою даних — замість цього використовуються серверні функції фреймворку Next.js, що дозволяє напряму звертатися до бази з інтерфейсу.

Для візуалізації даних у вебдодатку вибрано Chart.js, оскільки ця бібліотека забезпечує більшу гнучкість у налаштуванні та краще інтегрується в React-компонентну архітектуру порівняно з альтернативами. Для стилізації інтерфейсу використовується Tailwind CSS, що дозволяє швидко створювати сучасні, адаптивні інтерфейси без громіздких CSS-файлів.

5 РОЗРОБКА СИСТЕМИ

5.1 Підготовка даних

Першим етапом розробки системи стало формування якісного датасету, який би дозволив побудувати надійну модель прогнозування та адекватно відтворити поведінку системи в реальному середовищі. Було проаналізовано великий масив реальних екологічних даних, які охоплюють ключові параметри повітряного середовища — температуру, вологість та концентрацію дрібнодисперсних частинок PM2.5.

Задля створення універсального та всеосяжного набору даних для навчання моделі, вихідні значення було не лише очищено, але й цілеспрямовано збагачено синтетичними аномаліями. Такий підхід дозволив змодельовати нестандартні ситуації, з якими система може зіткнутися в реальному використанні: різкі скачки показників, поступові тренди, шумові сплески тощо. Це особливо важливо для задач виявлення аномалій і коректного прогнозування за межами типових сценаріїв.

Для побудови датасету було використано платформу Kaggle, яка надала доступ до відкритих наборів даних з історичними показниками якості повітря у міських умовах [10]. Ці дані охоплюють широкий спектр параметрів, включаючи концентрації зважених часток, рівні оксиду вуглецю, діоксиду азоту, озону, температуру, вологість, а також супровідну метеорологічну інформацію. Після етапів фільтрації, нормалізації, обробки пропущених значень та злиття даних з кількох джерел, було сформовано єдиний структурований датасет обсягом близько 100 000 міток, кожна з яких відповідає часовому інтервалу в 5 хвилин.

Такий високий темп вибірки забезпечує високу щільність записів у часовому вимірі, що дозволяє точно моделювати динаміку змін у стані навколишнього середовища не лише впродовж однієї доби, а й на міжтижневих горизонтах. Завдяки цьому підходу, сформований датасет не лише адекватно відображає поведінку системи в реальних умовах, але й закладає основу для побудови надійних моделей прогнозування екологічних параметрів. На рисунку 5.1 зображений

сформований датасет зі значеннями температури, вологості та дрібнодисперсних частинок PM2.5 за весь час.

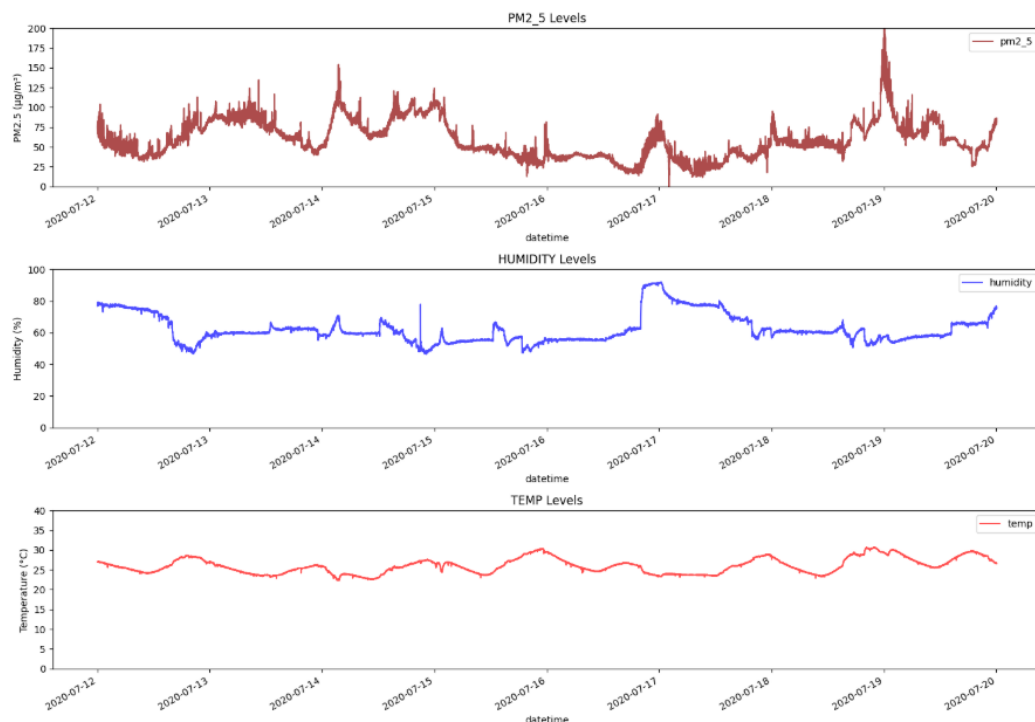


Рисунок 5.1 — Сформований датасет зі значеннями PM2.5, вологості, температури за весь час

5.2 Модель машинного навчання

Для реалізації прогнозуючого модуля було обрано архітектуру нейронної мережі типу LSTM, реалізовану за допомогою бібліотеки TensorFlow. Завдання моделі полягає в тому, щоб передбачити значення концентрації PM2.5 на найближчу годину, спираючись на дані за попередні 24 години. Такий підхід дозволяє враховувати динаміку змін у довгостроковому часовому вікні. Використання LSTM забезпечує гнучке навчання на часових рядах без необхідності приведення їх до стаціонарного вигляду, дозволяючи моделі виявляти приховані закономірності та складні нелінійні залежності в динамічному середовищі.

5.2.1 Тренування моделі машинного навчання

					ІК11.020БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		50

Було визначено чітку структуру вхідних і вихідних даних. Модель отримує на вхід 288 попередніх міток — що еквівалентно 24 годинам даних, враховуючи п'ятихвилинний інтервал між записами — та формує прогноз на наступні 12 міток, тобто на одну годину наперед.

Перед тренуванням усі вхідні дані були нормалізовані до діапазону від 0 до 1 за допомогою методу Min-Max Scaling. Дані було поділено на три частини: 70% для навчання, 15% для валідації та 15% для тестування.

Архітектура моделі побудована послідовно. Вхідним шаром є LSTM-шар на 128 нейронів, після цього йде повнозв'язний шар, який перетворює останній вихід LSTM на прогноз довжиною у 12 значень, тобто на наступну годину.

Модель було скомпільовано з використанням оптимізатора Adam зі швидкістю навчання 0.001 та функції втрат Mean Squared Error (MSE)..

Процес навчання моделі був побудований на перевірених практиках сучасного машинного навчання для часових рядів і дозволив отримати стабільну, чутливу до змін і адаптивну до динаміки систему прогнозування. На рисунку 5.2 зображено процес навчання моделі.

```
Epoch 1/8
578/578 - 112s - 194ms/step - loss: 0.0010 - rmse: 0.0322 - val_loss: 2.5623e-04 - val_rmse: 0.0160
Epoch 2/8
578/578 - 111s - 192ms/step - loss: 2.3568e-04 - rmse: 0.0154 - val_loss: 2.0241e-04 - val_rmse: 0.0142
Epoch 3/8
578/578 - 191s - 330ms/step - loss: 2.2127e-04 - rmse: 0.0149 - val_loss: 3.3350e-04 - val_rmse: 0.0183
Epoch 4/8
578/578 - 109s - 189ms/step - loss: 2.2257e-04 - rmse: 0.0149 - val_loss: 3.5671e-04 - val_rmse: 0.0189
Epoch 5/8
578/578 - 116s - 201ms/step - loss: 2.1582e-04 - rmse: 0.0147 - val_loss: 3.7316e-04 - val_rmse: 0.0193
Epoch 6/8
578/578 - 117s - 203ms/step - loss: 2.1243e-04 - rmse: 0.0146 - val_loss: 4.3831e-04 - val_rmse: 0.0209
Epoch 7/8
578/578 - 115s - 199ms/step - loss: 2.1058e-04 - rmse: 0.0145 - val_loss: 4.1163e-04 - val_rmse: 0.0203
Epoch 8/8
578/578 - 112s - 194ms/step - loss: 2.0654e-04 - rmse: 0.0144 - val_loss: 6.0672e-04 - val_rmse: 0.0246
```

Рисунок 5.2 — процес навчання моделі LSTM протягом восьми епох

Рисунок 5.3 демонструє зміну функції втрат на тренувальній вибірці, отриману в результаті побудови графіка в процесі навчання моделі. Цей графік відображає динаміку зменшення помилки з кожною епохою, що дозволяє візуально

оцінити ефективність оптимізації, стабільність процесу навчання, а також виявити можливі ознаки перенавчання або застопорення градієнтів.

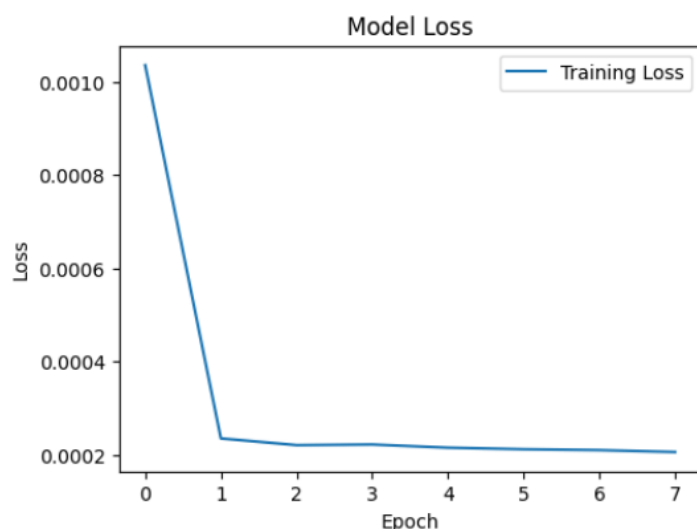


Рисунок 5.3 — Графік функції втрат на тренувальній вибірці

5.2.2 Оцінка натренованої моделі машинного навчання

Після завершення етапу навчання модель була протестована на незалежному тестовому наборі даних, що не використовувався під час тренування або валідації. Для оцінки точності прогнозів було обрано дві метрики: MAE (Mean Absolute Error) та RMSE (Root Mean Squared Error). Перша метрика — MAE — показує середню абсолютну помилку між фактичними та передбаченими значеннями, незалежно від напрямку відхилення. Друга — RMSE — є більш чутливою до великих помилок і надає інформацію про стандартне відхилення прогнозів від реальності.

На тестовому наборі модель досягла результатів $MAE = 1.850$ та $RMSE = 3.379$, що свідчить про високу точність. З огляду на те, що вхідні значення $PM_{2.5}$ у датасеті варіюються в діапазоні від 0 до 200 $мкг/м^3$, такі значення метрик можна вважати відносно низькими, а отже — якість прогнозів є задовільною.

На рисунку 5.4 зображено графік розсіювання похибок моделі де по осі X — фактичні значення, а по осі Y — передбачені. Ідеальний прогноз мав би утворити щільний кластер вздовж діагоналі. Як видно з графіка, більшість точок дійсно

зосереджена вздовж цієї лінії, що підтверджує якість моделі. Помітні незначні відхилення при високих концентраціях PM_{2.5}, однак навіть у цих випадках прогнози залишаються наближеними до реальних значень.

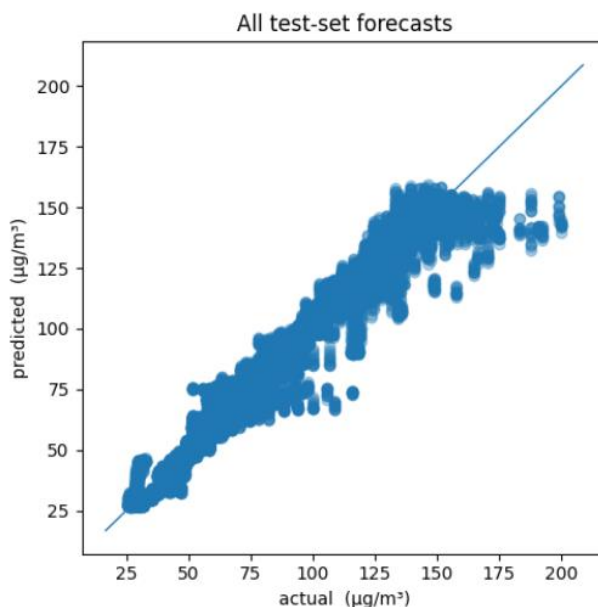


Рисунок 5.4 — Графік розсіювання похибок моделі на тестових даних

Загалом, модель демонструє стабільну поведінку, високу точність та адаптивність до складної динаміки змін, що дозволяє вважати її придатною для використання в системі реального моніторингу та прогнозування якості повітря.

5.3 Створення бази даних

Для зберігання результатів прогнозування, статистичного аналізу та подальшої візуалізації у системі було створено дві основні таблиці в базі даних PostgreSQL. Вони формують основу для довготривалого зберігання агрегованих та модельних даних, з якими взаємодіють як внутрішні модулі, так і користувацький інтерфейс.

Перша таблиця — forecast — призначена для зберігання результатів роботи прогнозуючого модуля. Вона містить такі поля:

- а) id — унікальний ідентифікатор запису;

б) `date_forecasted` — дата та час, коли було зроблено прогноз;
 в) `date_target` — дата та час, на які саме зроблено прогноз;
 г) `input_sequence` — масив значень, що подавалися на вхід моделі (наприклад, останні 288 точок);

д) `predicted_sequence` — масив передбачених значень на наступну годину;

е) `pm2_5` — середнє значення PM2.5 за прогнозований період.

Ця таблиця використовується для аналізу точності моделі, побудови графіків, а також подальших обчислень, які потребують історичних прогнозів.

Друга таблиця — `hourly_sensor_metrics` — зберігає агреговану щогодинну статистику, сформовану модулем аналізу. Вона включає:

а) `id` — унікальний ідентифікатор;

б) `window_end_time` — кінець кожного годинного вікна;

в) `avg_temp` — середня температура за годину;

г) `avg_humidity` — середня вологість;

д) `avg_pm2_5` — середнє значення PM2.5;

е) `aqi` — обчислений індекс якості повітря (Air Quality Index).

Ці метрики дозволяють системі фіксувати загальний стан навколишнього середовища в кожен момент часу, а також реагувати на погіршення якості повітря.

Надалі ці таблиці будуть активно використовуватись модулями аналізу та прогнозування для зчитування, оновлення й збереження даних, а також стануть основним джерелом інформації для користувацького інтерфейсу, який буде візуалізувати динаміку змін у повітряному середовищі, історичні тренди та прогнозовані значення.

5.4 Модуль надсилання даних

Цей модуль виконує функцію джерела телеметричних даних у системі. Його реалізовано у вигляді простого Python скрипта, який читає значення з попередньо підготовленого CSV-файлу та надсилає їх у Kafka на відповідний топик з назвою

"in". Кожен запис у файлі містить значення температури, вологості та PM2.5 разом із часовою міткою, і подається як повідомлення у брокер.

Модуль дозволяє конфігурувати основні параметри — зокрема, вказувати шлях до файлу з даними та задавати інтервал між надсиланням повідомлень. Це дає змогу емулювати надходження даних у реальному часі або ж запускати систему в прискореному темпі для тестування та демонстрації. Модуль надсилання даних формує початковий потік інформації для всієї системи, імітуючи поведінку реальних сенсорів без потреби у фізичному обладнанні.

5.5 Модуль прогнозування даних

Модуль прогнозування реалізовано мовою Python, і його основне завдання — слухати топик "in" у Kafka, де з'являються нові телеметричні повідомлення від модуля надсилання або реальних сенсорів, накопичувати їх у вигляді послідовностей та підготовлювати до передбачення. Після накопичення достатньої кількості значень (288 міток, тобто 24 години даних), модуль форматує ці дані у вигляді, сумісному з моделлю LSTM. На основі цієї послідовності модель здійснює прогноз на наступну годину (12 міток).

Результати прогнозу передаються у два напрямки. По-перше, вони надсилаються у Kafka-топик "forecast", що дозволяє іншим модулям отримувати актуальні передбачення в режимі реального часу. По-друге, прогнози зберігаються у базу даних, зокрема у таблицю forecast, що дозволяє здійснювати історичний аналіз або візуалізацію.

5.6 Модуль аналізу даних

Модуль аналізу відповідає за обробку вхідного потоку телеметричних даних із Kafka-топіка "in". Він агрегує ці дані в щогодинні вікна та обчислює ключові статистичні метрики: середню температуру, вологість і концентрацію PM2.5. На основі середнього значення PM2.5 додатково обчислюється AQI —

					ІК11.020БАК.006 ПЗ	Арк.
						55
Зм.	Лист	№ докум.	Підпис	Дата		

стандартизований показник, що дозволяє класифікувати ступінь забруднення повітря за шкалою ризику для здоров'я.

AQI розраховується за допомогою лінійної інтерполяції згідно з національними екологічними стандартами. В таблиці 5.1 надано шкалу, яка використовується модулем:

Таблиця 5.1 — Шкала переведення PM2.5 в AQI

PM2.5 (мкг/м³)	Відповідний AQI	Категорія якості повітря
0.0 – 12.0	0 – 50	Добре
12.1 – 35.4	51 – 100	Помірно
35.5 – 55.4	101 – 150	Нездорово для чутливих груп
55.5 – 150.4	151 – 200	Нездорово
150.5 – 250.4	201 – 300	Дуже нездорово
250.5 – 350.4	301 – 400	Небезпечне забруднення
350.5 – 500.4	401 – 500	Екстремально небезпечне забруднення

Після обчислення всі метрики — середні значення температури, вологості, PM2.5, а також AQI — зберігаються у базі даних у таблиці `hourly_sensor_metrics`. Крім того, результат надсилається у Kafka-топік "metrics", щоб інші компоненти системи, зокрема модуль оповіщення та інтерфейс користувача, могли оперативно реагувати на зміни в стані повітря.

5.7 Модуль оповіщення

Модуль оповіщення відповідає за реагування на критичні значення індексу якості повітря та інформування користувача про потенційні загрози для здоров'я. Він постійно слухає Kafka-топік "metrics", у якому з'являються щогодинні агреговані показники від модуля аналізу.

Як тільки у вхідному повідомленні зафіксовано, що значення AQI перевищує 150, модуль автоматично генерує текстове попередження, яке виводиться в консоль. Такий поріг відповідає категорії "Нездорово", коли забруднення повітря

починає становити серйозну загрозу не лише для чутливих груп, а й для загального населення.

Повідомлення містить стислу та інформативну інтерпретацію всіх ключових параметрів, включно з рекомендаціями для користувача. На рисунку 5.5 наведено приклад такого сповіщення.

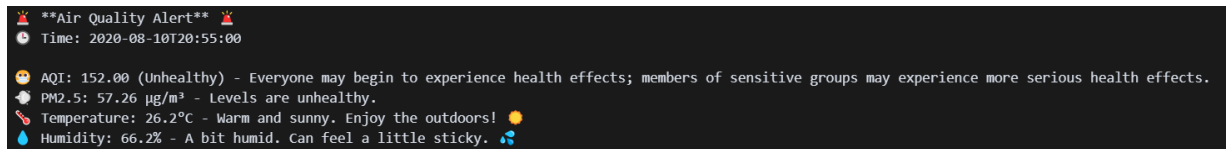


Рисунок 5.5 — Приклад повідомлення згенерованого модулем оповіщення

Повідомлення містить дату і час, пояснення до значення AQI, а також короткі коментарі щодо рівня PM2.5, температури та вологості. Така форма подачі інформації дозволяє користувачу швидко оцінити рівень небезпеки та ухвалити рішення про свої подальші дії.

У майбутньому цей модуль може бути розширений — наприклад, надсиланням push-сповіщень, електронних листів або інтеграцією з іншими системами оповіщення.

5.8 Користувацький інтерфейс

Графічний інтерфейс системи реалізовано у вигляді вебдодатку під назвою SensorData, побудованого з використанням сучасного фреймворку Next.js. Його архітектура поєднує клієнтську та серверну логіку, а завдяки server actions додаток має змогу безпечно напряму підключатися до бази даних, без необхідності проміжного API. Це забезпечує швидкий доступ до актуальної інформації та мінімізує затримки між надходженням даних і їхньою візуалізацією.

Інтерфейс містить три функціональні сторінки:

- а) Live;
- б) History;

в) About.

«Live» це головна сторінка моніторингу, яка відображає поточний прогноз на наступну годину та актуальні агреговані метрики — середню температуру, вологість, PM2.5 та індекс якості повітря (AQI).

Прогноз реалізовано у вигляді інтерактивного графіка, де останні дві години реальних даних виведено синьою лінією, а прогнозовані значення на наступну годину — жовтою. Такий візуальний підхід дозволяє інтуїтивно побачити продовження динаміки та оцінити, наскільки передбачення відповідає попереднім тенденціям. Сторінка відображена на рисунку 5.6.

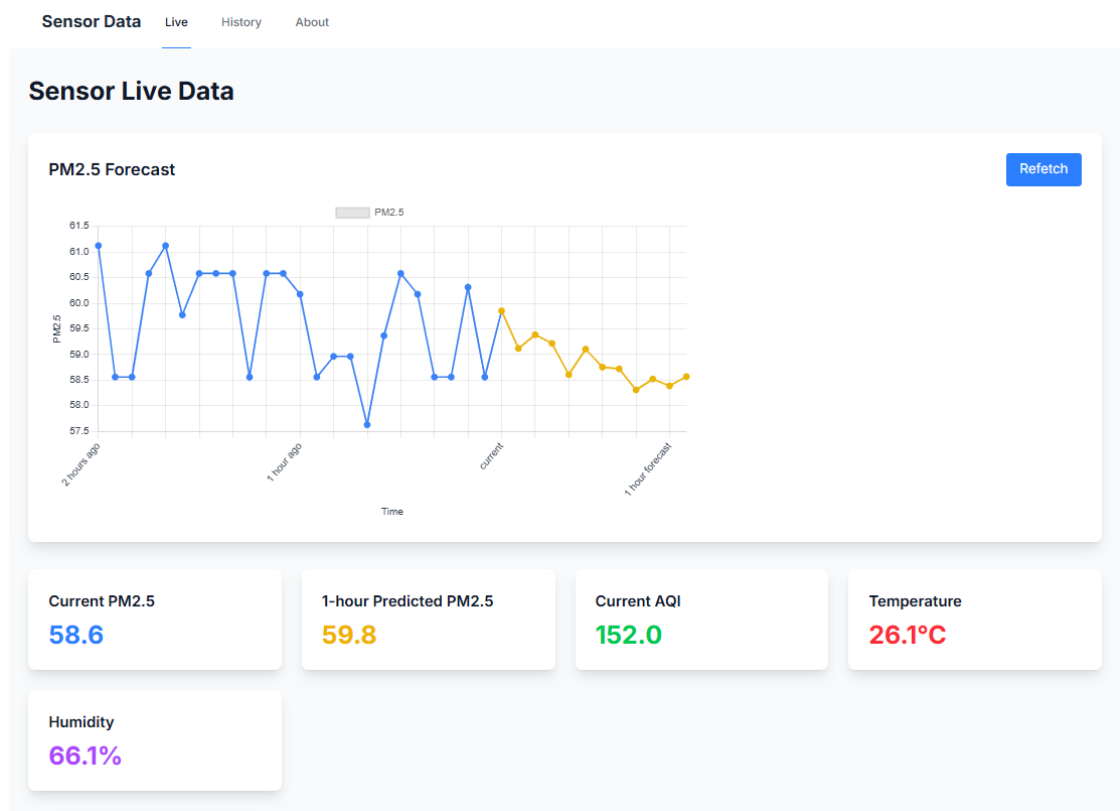


Рисунок 5.6 — Головна сторінка «Live» з поточним прогнозом та метриками

На сторінці «History» відображаються графіки чотирьох ключових параметрів — AQI, PM2.5, температура та вологість — з щогодинною роздільністю за весь період роботи системи. Всі дані інтерактивно агрегуються та виводяться у зручному для користувача вигляді, що дозволяє в реальному часі простежити як

короткострокові зміни, так і довготривалі тенденції у стані повітряного середовища.

Графіки дозволяють детально аналізувати сезонні коливання, виявляти сплески забруднення, визначати періоди стабільності або аномальні відхилення від норми. Завдяки візуалізації даних користувачі можуть не лише оцінити поточний стан повітря, а й порівнювати різні часові відрізки, що особливо корисно для виявлення впливу зовнішніх чинників, таких як погодні умови або антропогенне навантаження. Зібрані на цій сторінці дані можуть бути використані як для оперативного реагування, так і для довгострокового планування заходів з покращення якості повітря. Сама сторінка є ключовим елементом інтерфейсу системи, оскільки забезпечує прозорість, зручність і наочність моніторингу, що зображено на рисунку 5.7.

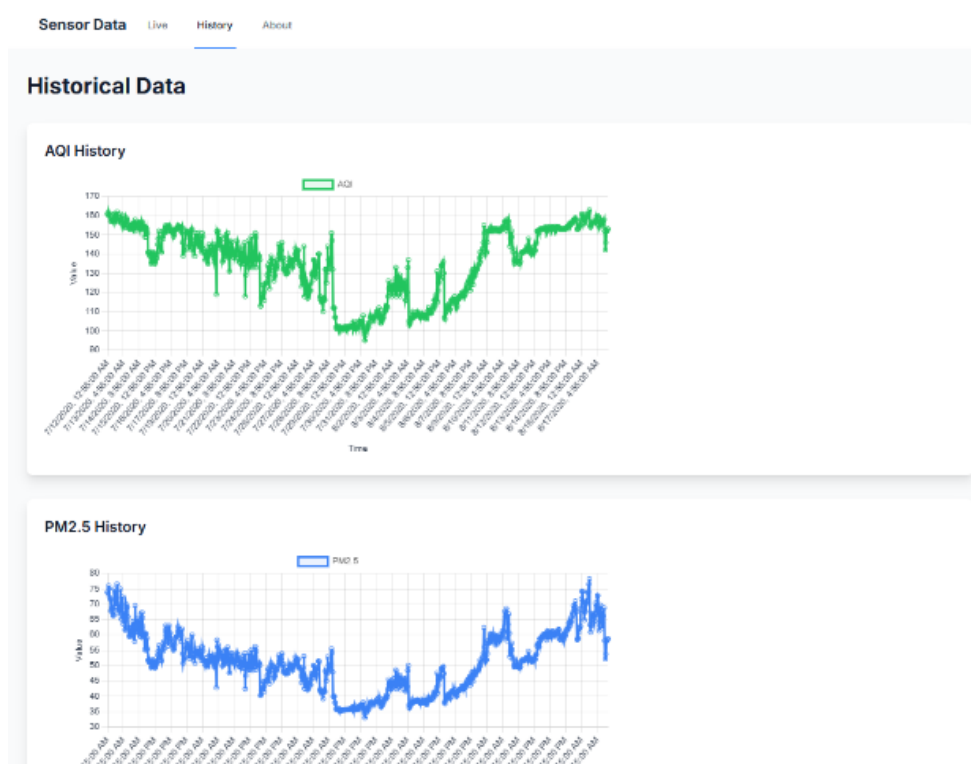


Рисунок 5.7 — Сторінка «History», графіки AQI та PM2.5

Сторінка «About» містить короткий опис проєкту, його цілі, реалізовану архітектуру та основні ідеї. Вона служить як довідковий розділ для ознайомлення

з системою з боку користувача або потенційного дослідника. Сторінка відображена на рисунку 5.8.

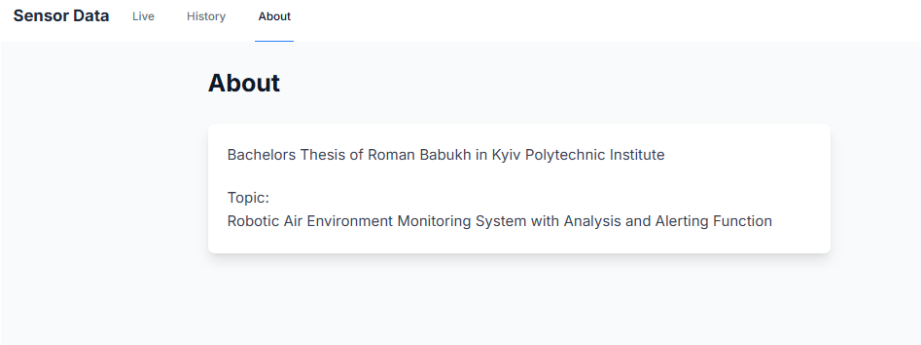


Рисунок 5.8 — Сторінка «About»

Всі графіки в інтерфейсі реалізовано за допомогою бібліотеки Chart.js, яка забезпечує високу інтерактивність: графіки можна масштабувати, пересувати, а також наводити курсор на конкретні точки, щоб побачити точне значення параметра у вибраний момент часу. Це дозволяє детально вивчати поведінку даних у конкретні періоди та полегшує аналіз для користувача.

5.9 Діаграма BPMN

Для ілюстрації логіки роботи розробленої системи моніторингу повітряного середовища була побудована діаграма бізнес процесів, зображена на кресленику ІК11.020БАК.006 ДЗ, яка демонструє послідовність виконання ключових процесів. Діаграма розділена на логічні блоки, відповідні окремим модулям системи: передача даних, обробка, прогнозування, оповіщення та інтерфейс користувача.

Процес розпочинається з надсилання сирих даних від датчиків у чергу повідомлень Kafka. Дані паралельно передаються до модулів аналізу та прогнозування. В аналітичному блоці дані агрегуються за годину, розраховується AQI, після чого метрики зберігаються в базу даних.

Модуль прогнозування готує дані, формує прогноз за допомогою моделі машинного навчання, і результати також зберігаються в базу даних та публікуються у відповідну чергу.

Далі система перевіряє, чи перевищує значення AQI встановлений поріг 150 і в разі перевищення — генерує відповідні сповіщення. Водночас користувацький інтерфейс періодично звертається до бази даних, рендерить сторінки та графіки, відображаючи всю релевантну інформацію у зрозумілому вигляді для користувача.

Ця діаграма відображає фактичну реалізацію, що дозволяє чітко простежити потік даних і логіку взаємодії між компонентами системи.

Висновки до розділу 5

У межах цього розділу було реалізовано систему моніторингу параметрів повітряного середовища, яка охоплює збір, обробку, прогнозування, аналіз, візуалізацію даних та оповіщення. Кожен модуль виконує окрему функцію, а взаємодія між ними забезпечується через брокер повідомлень Kafka та централізовану базу даних.

Користувацький інтерфейс реалізовано як вебдодаток під назвою «SensorData» за допомогою Next.js. Він безпосередньо підключається до бази даних за допомогою серверних дій. Інтерфейс складається з трьох сторінок: Live — для перегляду поточних метрик і прогнозу у вигляді комбінованого графіка; History — з історичними графіками AQI, PM2.5, температури та вологості; та About — з описом проєкту. Візуалізацію реалізовано на базі бібліотеки Chart.js з відкритим кодом, що дозволяє масштабувати графіки, переміщатися по них та переглядати конкретні значення точок.

Була складена BPMN-діаграма для огляду процесів в розробленій системі.

6 ТЕСТУВАННЯ СИСТЕМИ

6.1 Огляд тестування системи

У цьому розділі буде описано процес тестування розробленої системи моніторингу параметрів повітряного середовища. Тестування проводитиметься на основі діаграми прецедентів зображеної на кресленику ІК11.020БАК.006 Д4 з урахуванням дій усіх акторів, зазначених у схемі. Основною метою буде перевірка відповідності реалізованої функціональності вимогам, визначеним на етапі проєктування, а також оцінка стабільності системи та зручності її використання.

Під час тестування буде перевірено, чи відбуватиметься коректне збереження даних, чи працюватимуть аналітичні та прогностичні модулі, чи надходитимуть повідомлення у випадку аномалій. Особливу увагу буде приділено взаємодії між модулями через брокер повідомлень, а також коректному оновленню інформації в інтерфейсі.

6.2 Запуск системи

Процес запуску системи передбачатиме поетапне розгортання всіх необхідних компонентів інфраструктури та програмних модулів Адміністратором. Основною платформою для керування службами буде Docker, що забезпечує ізольоване середовище для кожного компонента.

Інфраструктура буде розгортатися за допомогою команди `docker compose up`, яка підніматиме всі сервіси відповідно до конфігурації.

Інфраструктура включає в себе:

- а) Apache Kafka;
- б) допоміжний компонент Zookeeper [32];
- в) PostgreSQL;
- г) Kafka UI;
- д) pgAdmin.

На рисунку 6.1 наведено скріншот з Docker Desktop, що демонструє запущені контейнери.

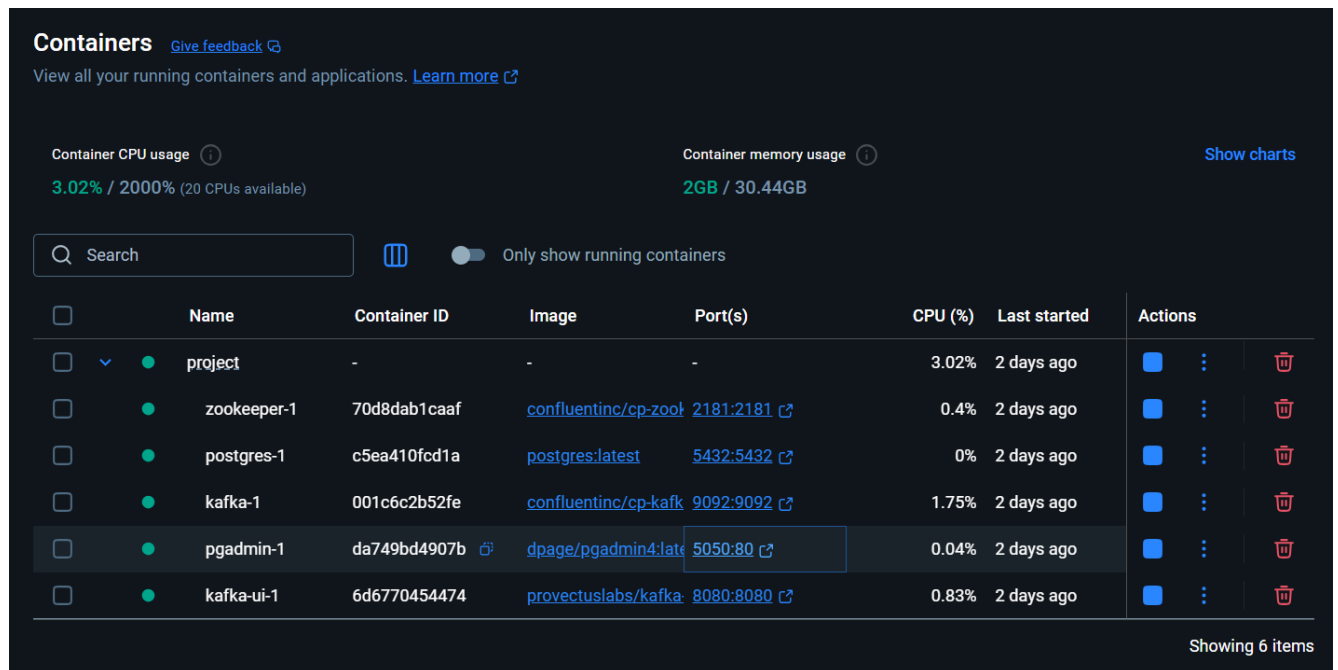


Рисунок 6.1 – Інфраструктурні компоненти в Docker Desktop

Після ініціалізації контейнерів буде необхідно встановити всі залежності для Python-скриптів та фронтенду. Для цього у відповідних директоріях буде виконано команди:

- а) `pip install -r src/requirements.txt`;
- б) `npm install` (у каталозі `src/sensor-data-web-app`).

Наступним кроком стане створення необхідної структури бази даних. Для цього буде запущено скрипт ініціалізації: `python src/db_setup/init_db.py`

Після цього запускатимуться основні обчислювальні модулі, що відповідають за аналіз, прогнозування та оповіщення. Всі вони можуть бути запущені в окремих терміналах наступними командами:

- а) `python src/sensor-data-analyzer/main.py`;
- б) `python src/sensor-data-forecaster/main.py`;
- в) `python src/sensor-data-notifier/main.py`.

Завершальним етапом буде запуск вебзастосунку, який забезпечує користувацький інтерфейс. Для цього в директорії фронтенду виконуватиметься команда: `npm run start` (у каталозі `src/sensor-data-web-app`).

6.3 Підключення датчиків

Відповідно до діаграми прецедентів, роль підключення датчиків виконує Адміністратор. У контексті тестування ця функція реалізовуватиметься шляхом завантаження підготовленого набору даних у систему.

Для тестування буде використано набір з 15 тисяч телеметричних записів, які не входили до навчальної вибірки моделі машинного навчання. Це дозволить не лише перевірити стабільність обробки даних у потоці, а й оцінити точність прогнозування та поведінку системи у випадках виявлення аномалій.

Передача даних у систему здійснюватиметься за допомогою запуску окремого скрипту: `python src/sensor-data-sender/main.py`

Цей процес активує надсилення даних до брокера повідомлень, після чого вся інфраструктура зможе почати повноцінну роботу в режимі реального часу. На рисунку 6.2 продемонстровані логи надсилення телеметричних записів.

```
2025-06-06 17:43:27,944 - INFO - Sent data: {'ts': 1599584700.0, 'humidity': 59.70000076293945, 'temp': 24.5, 'datetime': '2020-09-08 17:05:00', 'pm2_5': 82.85683160110604}
2025-06-06 17:43:27,955 - INFO - Sent data: {'ts': 1599585000.0, 'humidity': 59.70000076293945, 'temp': 24.5, 'datetime': '2020-09-08 17:10:00', 'pm2_5': 84.49728521726372}
2025-06-06 17:43:27,967 - INFO - Sent data: {'ts': 1599585300.0, 'humidity': 59.79999923706055, 'temp': 24.5, 'datetime': '2020-09-08 17:15:00', 'pm2_5': 85.69616129230813}
2025-06-06 17:43:27,979 - INFO - Sent data: {'ts': 1599585600.0, 'humidity': 59.79999923706055, 'temp': 24.5, 'datetime': '2020-09-08 17:20:00', 'pm2_5': 83.45286319549969}
2025-06-06 17:43:27,993 - INFO - Sent data: {'ts': 1599585900.0, 'humidity': 59.79999923706055, 'temp': 24.5, 'datetime': '2020-09-08 17:25:00', 'pm2_5': 83.30248615531817}
2025-06-06 17:43:28,009 - INFO - Sent data: {'ts': 1599586200.0, 'humidity': 59.70000076293945, 'temp': 24.5, 'datetime': '2020-09-08 17:30:00', 'pm2_5': 85.24746044748204}
2025-06-06 17:43:28,022 - INFO - Sent data: {'ts': 1599586500.0, 'humidity': 59.70000076293945, 'temp': 24.5, 'datetime': '2020-09-08 17:35:00', 'pm2_5': 86.29713325459142}
2025-06-06 17:43:28,034 - INFO - Sent data: {'ts': 1599586800.0, 'humidity': 59.70000076293945, 'temp': 24.5, 'datetime': '2020-09-08 17:40:00', 'pm2_5': 81.07987293627406}
2025-06-06 17:43:28,051 - INFO - Sent data: {'ts': 1599587100.0, 'humidity': 59.79999923706055, 'temp': 24.5, 'datetime': '2020-09-08 17:45:00', 'pm2_5': 83.30248615531817}
2025-06-06 17:43:28,062 - INFO - Sent data: {'ts': 1599587400.0, 'humidity': 59.900001525878906, 'temp': 24.600000381469727, 'datetime': '2020-09-08 17:50:00', 'pm2_5': 81.07987293627406}
2025-06-06 17:43:28,078 - INFO - Sent data: {'ts': 1599587700.0, 'humidity': 59.79999923706055, 'temp': 24.5, 'datetime': '2020-09-08 17:55:00', 'pm2_5': 83.89937706631278}
2025-06-06 17:43:28,093 - INFO - Sent data: {'ts': 1599588000.0, 'humidity': 59.79999923706055, 'temp': 24.5, 'datetime': '2020-09-08 18:00:00', 'pm2_5': 83}
```

Рисунок 6.2 – Логи надсилення телеметричних даних

6.4 Тестування модулів

Після надсилення вхідних даних перевірено функціонування всіх ключових модулів системи: аналізу, прогнозування та оповіщення. Згідно з діаграмою

					ІК11.020БАК.006 ПЗ	Арк.
						64
Зм.	Лист	№ докум.	Підпис	Дата		

прецедентів, взаємодія з цими модулями відбувається від імені Датчиків, які надсилають вхідні дані в систему. Мета тестування — впевнитися, що всі компоненти реагують на надходження інформації, здійснюють необхідні обчислення та формують очікувані результати.

Модуль аналізу відповідальний за обробку вхідних телеметричних значень і формування щогодинних агрегованих метрик. Його робота підтверджується логами які зображено на рисунку 6.3.

```

2025-06-06 17:43:20,648 - main - INFO - Processed metrics for window ending at 2020-09-06T19:55:00
2025-06-06 17:43:20,818 - main - INFO - Processed metrics for window ending at 2020-09-06T20:55:00
2025-06-06 17:43:21,013 - main - INFO - Processed metrics for window ending at 2020-09-06T21:55:00
2025-06-06 17:43:21,159 - main - INFO - Processed metrics for window ending at 2020-09-06T22:55:00
2025-06-06 17:43:21,351 - main - INFO - Processed metrics for window ending at 2020-09-06T23:55:00
2025-06-06 17:43:21,496 - main - INFO - Processed metrics for window ending at 2020-09-07T00:55:00
2025-06-06 17:43:21,662 - main - INFO - Processed metrics for window ending at 2020-09-07T01:55:00
2025-06-06 17:43:21,805 - main - INFO - Processed metrics for window ending at 2020-09-07T02:55:00
2025-06-06 17:43:21,965 - main - INFO - Processed metrics for window ending at 2020-09-07T03:55:00
2025-06-06 17:43:22,130 - main - INFO - Processed metrics for window ending at 2020-09-07T04:55:00
2025-06-06 17:43:22,346 - main - INFO - Processed metrics for window ending at 2020-09-07T05:55:00
2025-06-06 17:43:22,505 - main - INFO - Processed metrics for window ending at 2020-09-07T06:55:00
2025-06-06 17:43:22,652 - main - INFO - Processed metrics for window ending at 2020-09-07T07:55:00
2025-06-06 17:43:22,814 - main - INFO - Processed metrics for window ending at 2020-09-07T08:55:00
2025-06-06 17:43:22,998 - main - INFO - Processed metrics for window ending at 2020-09-07T09:55:00

```

Рисунок 6.3 – Логи модуля аналізу

Крім того, збереження оброблених метрик підтверджується через перегляд вмісту таблиці `hourly_sensor_metrics` у `pgAdmin` — її приклад на рисунку 6.4. Видно присутність різних даних в таблицях що з’явилися в результаті стабільної роботи системи

id	window_and_time	avg_temp	avg_humidity	avg_pm2.5	aqi
1	2020-07-12 00:55:00	27	77.975	73.870224	161
2	2020-07-12 01:55:00	27	78.43334	73.488464	160
3	2020-07-12 02:55:00	27	78.01667	76.00709	162
4	2020-07-12 03:55:00	27	78	74.95306	161
5	2020-07-12 04:55:00	27.016666	78.10833	72.12876	160
6	2020-07-12 05:55:00	27	78.61667	71.611786	159
7	2020-07-12 06:55:00	27	78.05	67.83292	157
8	2020-07-12 07:55:00	27	77.791664	67.72504	157
9	2020-07-12 08:55:00	27	77.95	68.855	158
10	2020-07-12 09:55:00	27	77.93334	68.564064	158

Рисунок 6.4 – Скриншот з pgAdmin, таблиця `hourly_sensor_metrics`

Модуль прогнозування забезпечує побудову прогнозів якості повітря на наступну годину. Він запускає модель машинного навчання, використовуючи

останні значення телеметрії. Результати прогнозів та дати, відображені у логах, фрагмент яких наведено на рисунку 6.5.

```
2025-06-08 16:46:39,875 - root - INFO - Saved forecast with ID: 1429
2025-06-08 16:46:39,876 - __main__ - INFO - Made forecast for 2020-07-13T07:00:00: 65.3589096069336
1/1 ----- 0s 29ms/step
2025-06-08 16:46:40,010 - root - INFO - Saved forecast with ID: 1430
2025-06-08 16:46:40,010 - __main__ - INFO - Made forecast for 2020-07-13T08:00:00: 65.75428771972656
1/1 ----- 0s 30ms/step
2025-06-08 16:46:40,141 - root - INFO - Saved forecast with ID: 1431
2025-06-08 16:46:40,142 - __main__ - INFO - Made forecast for 2020-07-13T09:00:00: 62.17205810546875
1/1 ----- 0s 29ms/step
2025-06-08 16:46:40,273 - root - INFO - Saved forecast with ID: 1432
2025-06-08 16:46:40,274 - __main__ - INFO - Made forecast for 2020-07-13T10:00:00: 61.15275192260742
1/1 ----- 0s 28ms/step
2025-06-08 16:46:40,404 - root - INFO - Saved forecast with ID: 1433
2025-06-08 16:46:40,404 - __main__ - INFO - Made forecast for 2020-07-13T11:00:00: 66.47325134277344
1/1 ----- 0s 28ms/step
```

Рисунок 6.5 – Логи модуля прогнозування

Модуль оповіщення генерує сповіщення у разі виявлення небезпечних або критичних параметрів. На рисунку 6.6 зображено приклад такого сповіщення. В ньому вказано:

- а) AQI: 168.00, що класифікується як "Unhealthy", тобто якість повітря є шкідливою для здоров'я всіх людей, особливо для чутливих груп;
- б) pm2.5: 89.10 $\mu\text{g}/\text{m}^3$, що перевищує допустимі межі та позначено як "unhealthy";
- в) температура: 24.6°C, описана як тепла та сонячна — заохочення до перебування на вулиці;
- г) вологість: 59.8%, вважається комфортною.

```
🌫️ AQI: 168.00 (Unhealthy) - Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects.
🌫️ PM2.5: 89.10  $\mu\text{g}/\text{m}^3$  - Levels are unhealthy.
🌤️ Temperature: 24.6°C - Warm and sunny. Enjoy the outdoors! 🌞
💧 Humidity: 59.8% - Comfortable humidity levels. 😊

🚨 **Air Quality Alert** 🚨
🕒 Time: 2020-09-08T11:55:00
```

Рисунок 6.6 – Приклад оповіщення про якість повітря

Додатково, стабільну роботу брокера повідомлень було підтверджено через Kafka UI. На рисунку 6.7 зображено наявність відповідних черг даних, а також

					ІК11.020БАК.006 ПЗ	Арк.
						66
Зм.	Лист	№ докум.	Підпис	Дата		

кількість повідомлень у кожному з них. Це дозволило переконатися, що дані не втрачаються, всі модулі отримують необхідну інформацію.

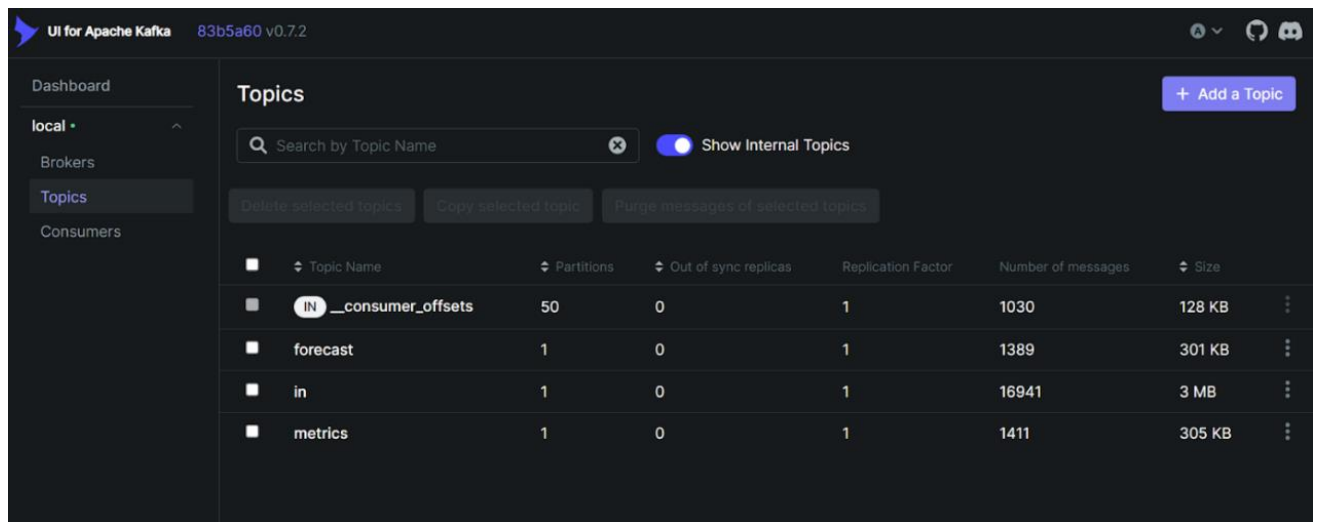


Рисунок 6.7 – Список черг даних з Kafka UI

6.5 Тестування користувацького інтерфейсу

У попередньому розділі на рисунках 5.6 та 5.7 було продемонстровано коректну роботу вебзастосунку де було підтверджено відображення прогнозів, метрик та історичних даних. На цьому етапі тестування зосередиться на виявленні можливих помилок у роботі інтерфейсу, а також на перевірці зручності користування та функціональності навігаційних елементів.

Інтерфейс успішно відображає стан відсутності даних: усі метрики дорівнюють нулю, графіки порожні, проте загальний вигляд сторінки залишається охайним, без зсувів, помилок чи некоректного форматування. Це свідчитиме про те, що система адекватно обробляє випадки, коли дані ще не надійшли, і не створює зайвого навантаження на користувача.

На рисунку 6.8 зображено головну сторінку застосунку до моменту надсилання даних з боку сенсорів.

Особливу увагу було приділено адаптивності інтерфейсу — дизайн залишається цілісним і гармонійним на різних пристроях, незалежно від розміру екрана. Перехід між сторінками та відображення даних відбуваються плавно, без

помітних затримок чи візуальних артефактів, що забезпечує позитивний користувацький досвід.

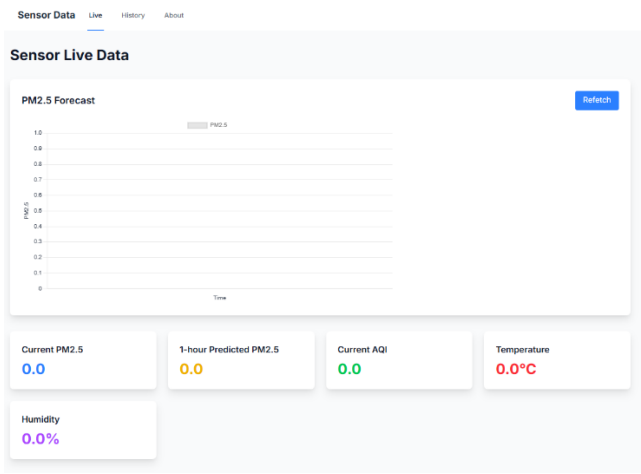


Рисунок 6.8 – Користувацький інтерфейс до надсилання даних з сенсорів

Наступним кроком буде перевірка інтерактивності графічного представлення даних. Однією з ключових функцій є можливість масштабування окремих фрагментів графіка — це важливо для зручного аналізу тенденцій та аномалій.

На рисунку 6.9 зображений графік історичних даних індексу якості повітря та червоним прямокутником виділена область яку буде наближено для тестування.

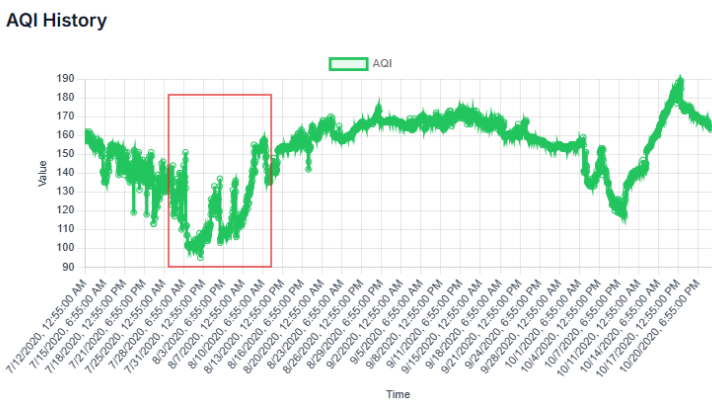


Рисунок 6.9 – Ділянка графіку для наближення

На рисунку 6.10 — результат масштабування виділеної області. Як можна помітити, регіон з даними приблизився, переглядати та вивчати дані та тренди

стало значно легше. Функціонал працює стабільно, інтерфейс реагує плавно, забезпечуючи комфортну навігацію.

AQI History



Рисунок 6.10 – Ділянка графіку після приближення

У результаті тестування буде підтверджено, що користувацький інтерфейс коректно відображає дані, забезпечує інтуїтивно зрозумілу взаємодію з користувачем та адаптується до зміни стану системи.

Висновки до розділу 6

У цьому розділі було виконано всебічне тестування системи моніторингу параметрів повітряного середовища, метою якого було перевірити коректність функціонування всіх компонентів, а також відповідність реалізованої архітектури вимогам, сформованим на етапі проєктування. Тестування охопило технічні аспекти запуску і користувацький досвід взаємодії з інтерфейсом. Тестування проводилося за допомогою діаграми прецедентів.

На початку було розгорнуто інфраструктуру системи за допомогою Docker, включно з Kafka, PostgreSQL та допоміжними інструментами — Kafka UI та pgAdmin. Після встановлення всіх залежностей і створення структури бази даних, система була повністю готова до тестування.

					ІК11.020БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		69

Було протестовано роботу модулів аналізу, прогнозування та оповіщення. Кожен із них показав очікувану поведінку: формувалися щогодинні метрики, будувалися прогнози, генерувалися сповіщення у разі виявлення небезпечних значень. Також увагу приділено перевірці міжмодульної комунікації через брокер повідомлень, а також збереженню інформації у базі даних. Ці процеси були зафіксовані у логах та додатково підтверджені скріншотами з Kafka UI і pgAdmin.

Крім того, проведено тестування користувацького інтерфейсу. Було перевірено, як система реагує на відсутність вхідних даних, чи правильно інтерфейс відображає нульові значення, та чи зручно користувачеві взаємодіяти з графіками. Було підтверджено, що візуальні компоненти працюють стабільно, масштабування графіків здійснюється плавно, а загальна структура сторінок залишається інтуїтивно зрозумілою.

Окремо було оцінено адаптивність і якість реалізації дизайну інтерфейсу. Система коректно масштабується під різні розміри екранів і пристроїв, зберігаючи структурованість, естетичний вигляд і читабельність усіх елементів. Переходи між сторінками та оновлення даних відбуваються плавно, без візуальних дефектів або затримок, що забезпечує комфортну й приємну взаємодію для користувача незалежно від сценарію використання.

ВИСНОВКИ

У результаті виконання дипломного проєкту було реалізовано повноцінну систему моніторингу якості повітря, яка дозволяє аналізувати дані з сенсорів у реальному часі, здійснювати прогнозування за допомогою моделі машинного навчання, формувати сповіщення про погіршення стану навколишнього середовища та візуалізувати результати для користувача через вебінтерфейс. Основна ідея проєкту полягала у створенні універсального рішення, яке можна адаптувати для міських, аграрних або індивідуальних потреб.

Під час розробки були використані сучасні технології, серед яких фреймворк Next.js для побудови зручного користувацького інтерфейсу, Apache Kafka як брокер повідомлень для зв'язку між модулями, Python для реалізації обробки даних і тренування прогнозної моделі LSTM, а також система управління базами даних для зберігання історичних значень і метрик. Інтерфейс реалізований з використанням підходу серверного рендерингу та підтримує сучасні функції взаємодії з динамічними даними.

Архітектура системи побудована на принципах асинхронної взаємодії між модулями, де кожен компонент виконує свою окрему функцію і може розвиватися незалежно. Це дозволяє легко адаптувати або замінити будь-який блок у майбутньому без необхідності модифікації всієї системи. Особлива увага приділялась розділенню обов'язків між сервісами, автоматизації процесів обробки та надійності передачі повідомлень.

Результатом роботи стала система, яка відповідає всім поставленим вимогам: вона стабільно обробляє вхідні екологічні дані, дозволяє оцінювати стан повітря, своєчасно виявляє аномальні показники та пропонує користувачу зрозумілу, зручну візуалізацію. Таке рішення може бути впроваджене як окремий локальний інструмент для приватного користування частина більшого екологічного або дослідницького проєкту.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Карта смертності від забруднення повітря внаслідок використання викопних палив, 2015 рік. URL: <https://ourworldindata.org/data-review-air-pollution-deaths> (дата звернення: 25.05.2025).
2. Розташування населених пунктів з даними про PM_{2,5}, за кількістю наземних вимірювань, 2010–2019 роки. URL: <https://www.epa.gov/outdoor-air-quality-data> (дата звернення: 25.05.2025).
3. Інтерактивна карта SaveEcoBot. URL: <https://www.saveecobot.com/en> (дата звернення: 25.05.2025).
4. Головна сторінка LAQN з інтерактивною картою. URL: <https://www.londonair.org.uk/> (дата звернення: 25.05.2025).
5. Інтерактивна карта AirNow. URL: <https://www.airnow.gov/> (дата звернення: 25.05.2025).
6. Інтерактивна карта PurpleAir URL: <https://map.purpleair.com/> (дата звернення: 25.05.2025).
7. Панель розробника PurpleAir URL: <https://develop.purpleair.com/> (дата звернення: 25.05.2025).
8. Kaiterra Data Platform URL: <https://www.kaiterra.com/dashboard> (дата звернення: 25.05.2025).
9. Головна сторінка Sensor.Community. URL: <https://sensor.community/> (дата звернення: 25.05.2025).
10. Платформа для аналізу даних Kaggle. URL: <https://www.kaggle.com/>
11. Платформа потокової обробки даних Apache Kafka. URL: <https://kafka.apache.org/>
12. Система обміну повідомленнями RabbitMQ. URL: <https://www.rabbitmq.com/>
13. Мова програмування Python. URL: <https://www.python.org/>
14. Мова програмування R. URL: <https://www.r-project.org/>

15. Бібліотека машинного навчання TensorFlow. URL: <https://www.tensorflow.org/>
16. Мова програмування C#. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/>
17. Мова програмування Javascript. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
18. Фронтенд фреймворк React. URL: <https://react.dev/>
19. Фронтенд фреймворк Angular. URL: <https://angular.io/>
20. Фреймворк для серверного рендерингу Next.js. URL: <https://nextjs.org/>
21. Бібліотека для візуалізації даних Plotly. URL: <https://plotly.com/>
22. Бібліотека для побудови графіків Chart.js. URL: <https://www.chartjs.org/>
23. CSS-фреймворк Tailwind. URL: <https://tailwindcss.com/>
24. Система керування базами даних Microsoft SQL Server (MSSQL). URL: <https://learn.microsoft.com/en-us/sql/sql-server/>
25. Система керування базами даних PostgreSQL. URL: <https://www.postgresql.org/>
26. Документно-орієнтована база даних MongoDB. URL: <https://www.mongodb.com/>
27. Платформа контейнеризації Docker. URL: <https://www.docker.com/>
28. Інтерфейс для керування PostgreSQL pgAdmin. URL: <https://www.pgadmin.org/>
29. Графічний інтерфейс для керування Kafka Kafka UI. URL: <https://provectus.io/open-source/kafka-ui/>
30. Система керування версіями Git. URL: <https://git-scm.com/>
31. Редактор коду Visual Studio Code. URL: <https://code.visualstudio.com/>
32. Система координації сервісів Zookeeper. URL: <https://zookeeper.apache.org/>

ДОДАТОК А

Код програми

Посилання на код програми на GitHub:

<https://github.com/chechmek/bachelors-thesis-air-monitoring>



Рисунок А.1 – Посилання на код програми