

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська кваліфікаційна робота

на тему: Розробка вебзастосунку для моніторингу навколишнього середовища.

Виконав: студент 4 курсу
групи 2ПІ-216
спеціальності

121 – Інженерія програмного забезпечення

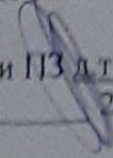
(шифр і назва напрямку підготовки, спеціальності)

 Клешко Віталій Володимирович
(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Ракитянська Г. Б.
(прізвище та ініціали)

Рецензент: к.т.н., доц. каф. ЗІ Маліновський В. І.
(прізвище та ініціали)

Допущено до захисту

Зав. кафедри ПЗ д.т.н. проф. О.Н. Романюк
 06 2025 р.

ВНТУ – 2025

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Інженерія програмного забезпечення»

~~ЗАТВЕРДЖУЮ~~
Завідувач кафедри ПЗ
О. Н. Романюк
« 21 » березня 2025 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Клешку Віталію Володимировичу

1. Тема роботи – розробка вебзастосунку для моніторингу навколишнього середовища.

Керівник роботи: Ракитянська Ганна Борисівна, к.т.н., доцент кафедри ПЗ.
затверджені наказом вищого навчального закладу від 20 березня 2025 р. №97.

2. Строк подання студентом роботи 2 червня 2025

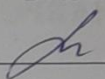
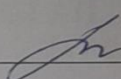
3. Вихідні дані до роботи: дані моніторингу джерела даних – відкритий API; бібліотеки для побудови графіків – Chart.js; технології веброзробки – HTML, CSS, JavaScript; інтерактивна мапа; середовище розробки – Visual Studio Code; вихідні дані – вебінтерфейс із графіками, даними, картою та панеллю керування.

4. Зміст текстової частини: анотація, вступ, аналіз існуючих вебсистем для моніторингу навколишнього середовища; порівняльний аналіз функціональних можливостей аналогічних рішень; оцінка методів збору та візуалізації екологічних даних; формулювання задачі розробки; аналіз архітектури програмного забезпечення; побудова структури вебзастосунку; моделювання структури системи; розробка алгоритмів взаємодії з користувачем та зовнішніми сервісами; обґрунтування вибору технологій для реалізації вебзастосунку; реалізація модулів

побудови графіків та інтерактивної мапи розробка інструкції користувача; висновки; список використаних джерел; додатки.

5. Перелік ілюстративної частини: порівняльна характеристика аналогів; архітектурна схема вебзастосунку; алгоритм роботи програмного застосунку; створення графіків; обробка події click; призначена точка на інтерактивній мапі;

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	к.т.н доц. каф. ПЗ Ракитянська Г. Б.	24.03.25 	01.06.25 

7. Дата видачі завдання 24 березня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

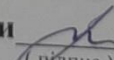
№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз систем для моніторингу навколишнього середовища та постановка задач	25.03.25 – 30.03.25	вик.
2	Розробка структури та аналогів програмного продукту	31.03.25 – 22.04.25	вик.
3	Розробка програмних модулів	31.03.25 – 27.04.25	вик.
4	Тестування програми	28.04.25 – 21.05.25	вик.
5	Оформлення матеріалів до захисту БКР	26.05.25 – 30.05.25	вик.

Студент


(підпис)

В. В. Клешк
(ініціали та прізвище)

Керівник бакалаврської кваліфікаційної роботи


(підпис)

Г. Б. Ракитянська
(ініціали та прізвище)

Анотація

Бакалаврська кваліфікаційна робота складається з 87 сторінок формату А4, на яких є 18 рисунків, 2 таблиці, список використаних джерел містить 24 найменування.

У бакалаврській роботі розроблено вебзастосунок для моніторингу навколишнього середовища з акцентом на контроль якості повітря. Система дозволяє брати, обробляти та візуалізувати екологічні дані, зокрема концентрацію шкідливих речовин у повітрі. Робота включає модулі для відображення графіків, отримання актуальної інформації через API та можливість взаємодії з картою. Вебзастосунок орієнтований на інформування користувачів про поточний стан повітря у зручній формі.

Вебзастосунок розроблено з використанням мов HTML, CSS та JavaScript, із застосуванням бібліотек для побудови графіків та обробки динамічних даних.

Ключові слова: навколишнє середовище, моніторинг, вебзастосунок, параметри, дані, API, показники.

Annotation

The bachelor's qualification work consists of 87 pages of A4 format, which include 18 figures, 2 tables, and the list of sources used contains 24 names.

In the bachelor's thesis, a web application for environmental monitoring with an emphasis on air quality control was developed. The system allows you to collect, process, and visualize environmental data, including the concentration of harmful substances in the air. The work includes modules for displaying graphs, obtaining current information via API, and the ability to interact with the map. The web application is focused on informing users about the current air condition in a convenient form.

The web application was developed using HTML, CSS, and JavaScript, using libraries for graphing and dynamic data processing.

Keywords: environment, monitoring, web application, parameters, data, API, metrics.

ЗМІСТ

ВСТУП.....	3
1. АНАЛІЗ СИСТЕМ ДЛЯ МОНІТОРИНГУ НАВКОЛИШНЬОГО СЕРЕДОВИЩА ТА ПОСТАНОВКА ЗАДАЧ.....	7
1.1 Аналіз стану вебсистем для моніторингу навколишнього середовища.....	7
1.2 Порівняльний аналіз аналогів.....	8
1.3 Аналіз методів розв’язання задачі.....	16
1.4 Постановка задач бакалаврської роботи.....	22
1.5 Висновки	23
2. РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ.....	24
2.1 Аналіз архітектури програмного забезпечення	24
2.2 Розробка структури вебзастосунку	28
2.3 Розробка моделі системи.....	30
2.4 Розробка алгоритму	32
2.5 Висновки	35
3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ	36
3.1 Варіантний аналіз і обґрунтування вибору засобів реалізації програмного продукту	36
3.2 Реалізація модуля побудови графіків.....	38
3.3 Реалізація модуля відображення інтерактивної карти	40
3.4 Реалізація модуля отримання даних про якість повітря через API.....	43
3.5 Реалізація модуля виведення інформації у CSV формат	45
3.6 Висновки	47
4. ТЕСТУВАННЯ ПРОГРАМИ	48
4.1 Тестування системи.....	48
4.2 Розробка інструкції користувача	53
4.3 Висновки	56
ВИСНОВКИ	57
ЛІТЕРАТУРА.....	58
ДОДАТКИ	60
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ	61
ДОДАТОК Б – ПЕРЕВІРКА НА ПЛАГІАТ	65
ДОДАТОК В – ЛІСТИНГ ПРОГРАМИ	66
ДОДАТОК Г – ІЛЮСТРАТИВНА ЧАСТИНА	77

ВСТУП

Обґрунтування вибору теми дослідження. Сучасний етап розвитку інформаційних технологій супроводжується стрімким зростанням обсягів даних та підвищенням суспільного інтересу до екологічної безпеки, що зумовлює потребу в інструментах для ефективного контролю стану довкілля. В умовах техногенного навантаження, зміни клімату та урбанізації питання моніторингу навколишнього середовища набуває особливої актуальності. Виявлення перевищень допустимих концентрацій забруднюючих речовин в атмосфері є критично важливим для збереження здоров'я населення, планування природоохоронних заходів та прийняття управлінських рішень.

Традиційні системи екологічного моніторингу, як правило, орієнтовані на централізовану обробку даних та характеризуються обмеженою доступністю інформації, низькою інтерактивністю й складним інтерфейсом користувача. Це значно ускладнює використання таких систем у повсякденному житті громадян або представників громадських організацій, що зацікавлені у постійному доступі до актуальної інформації про стан навколишнього середовища [1].

Інтенсивний розвиток вебтехнологій відкриває нові можливості для створення доступних, гнучких та масштабованих рішень у сфері екологічного моніторингу. Застосування сучасних вебтехнологій дозволяє реалізувати інтуїтивно зрозумілий інтерфейс та забезпечити інтерактивну взаємодію з користувачем, що підвищує зручність та ефективність роботи з екологічною інформацією [2].

Аналіз існуючих вебплатформ свідчить про наявність низки обмежень: закритість коду, відсутність можливості адаптації під локальні потреби, обмеженість у джерелах даних або їх візуалізації. Це створює підґрунтя для розробки нового вебзастосунку, орієнтованого на відкритість, адаптивність і зручність для кінцевого користувача.

Актуальність теми полягає у необхідності створення інформаційної системи, яка дозволить широкому колу користувачів ефективно моніторити стан навколишнього середовища, оперативно виявляти екологічні загрози та приймати обґрунтовані рішення на основі актуальних даних [3]. Створення такого інструменту сприятиме покращенню екологічної обізнаності населення, підвищенню якості життя, а також посиленню державного та громадського контролю за дотриманням екологічних норм.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Метою бакалаврської кваліфікаційної роботи є покращення доступності та ефективності моніторингу навколишнього середовища шляхом створення зручного вебінтерфейсу для візуалізації екологічних показників.

Завдання, що вирішуються в роботі:

- проведення аналізу сучасних вебсистем екологічного моніторингу та виконання порівняльного аналізу існуючих аналогів;
- обґрунтування вибору архітектури програмного забезпечення та методів реалізації функціональності вебзастосунку;
- розробка структури, логічної моделі та алгоритмів функціонування веборієнтованої системи;
- реалізація програмних модулів;
- проведення тестування працездатності функціоналу вебзастосунку та створення інструкції користувача для забезпечення ефективної експлуатації системи.

Об'єкт дослідження – це процес розробки вебсистеми для моніторингу навколишнього середовища з використанням вебтехнологій.

Предмет дослідження – є методи та засоби розробки вебсистеми для моніторингу.

Методи дослідження. У роботі застосовано методи структурного аналізу та об'єктно-орієнтованого проєктування для формалізації вимог до системи та побудови моделі вебзастосунку. Для реалізації архітектури застосовано методи проєктування вебінтерфейсів, зокрема з використанням API для обробки та візуалізації екологічних даних.

Новизна отриманих результатів.

1. Розроблено вебзастосунок для моніторингу навколишнього середовища, який дозволяє відображати показники якості повітря в режимі реального часу з використанням інтерактивної карти.

2. Реалізовано механізм візуалізації екологічних показників за допомогою бібліотеки Chart.js, що сприяє підвищенню наочності даних для кінцевого користувача.

3. Реалізовано модульну архітектуру вебзастосунку, яка забезпечує масштабованість системи та підтримку інтеграції з відкритими джерелами екологічних даних через OpenWeatherAPI.

Практична цінність отриманих результатів. Розроблений вебзастосунок може бути впроваджений у діяльність екологічних організацій, навчальних закладів, місцевих громад або стартапів, що займаються моніторингом навколишнього середовища. Система забезпечує зручний доступ до актуальних екологічних даних, полегшує їх аналіз та візуалізацію, що сприяє підвищенню поінформованості населення та оперативному прийняттю рішень у сфері охорони довкілля.

Особистий внесок здобувача. Усі результати дослідження та розробки, представлені в роботі, були виконані автором особисто. Автор представляє огляд та аналіз відомих рішень, а також обґрунтовує новизну і практичну цінність запропонованої системи.

Апробація результатів здійснена на міжнародній науково-практичній інтернет-конференції студентів, аспірантів та молодих науковців «МОЛОДЬ В

НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ. МН-2025», Вінниця, 2025.

Публікації. Основні результати дослідження опубліковано в 1 науковій праці та представлені в збірнику міжнародної науково-практичної інтернет-конференції студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ. МН-2025» [4].

Структура та обсяг БКР. БКР складається зі вступу, чотирьох розділів, висновків, списку літератури, що містить 24 найменування, 4 додатків. Робота містить 18 ілюстрації, 2 таблиці. Загальний обсяг роботи складає 87 сторінок, основний зміст вкладено на 50 сторінок друкованого тексту.

1 АНАЛІЗ СИСТЕМ ДЛЯ МОНІТОРИНГУ НАВКОЛИШНЬОГО СЕРЕДОВИЩА ТА ПОСТАНОВКА ЗАДАЧ

1.1 Аналіз стану вебсистем для моніторингу навколишнього середовища

У контексті глобальних кліматичних змін, зростання рівня урбанізації та індустріалізації, а також погіршення екологічної ситуації в багатьох регіонах світу, питання екологічного моніторингу набуває особливої актуальності. Одним із найефективніших і наймасовіших інструментів спостереження та аналізу стану довкілля на сьогодні є веборієнтовані інформаційні системи моніторингу.

Сучасні вебзастосунки для моніторингу навколишнього середовища забезпечують доступ до актуальної інформації про стан атмосферного повітря, водних ресурсів, ґрунтів та інших екологічно значущих параметрів. Завдяки використанню відкритих екологічних API, такі системи здатні отримувати та обробляти дані з тисяч автоматизованих станцій спостереження [5]. Це створює передумови для глобального охоплення та забезпечення оперативності оновлення інформації.

Типовий функціонал подібних вебзастосунків включає інтерактивні карти, динамічні графіки, діаграми, історичний аналіз даних, а також модулі прогнозування якості повітря. Додатково, багато сервісів пропонують інструменти персоналізації – вибір регіону, налаштування сповіщень про перевищення граничнодопустимих концентрацій шкідливих речовин, фільтрацію даних за окремими показниками тощо. Також спостерігається тенденція до інтеграції вебплатформ з мобільними застосунками, смарт-пристроями та соціальними мережами, що сприяє поширенню екологічної інформації серед широкого кола користувачів та підвищенню рівня екологічної свідомості населення.

Завдяки активному розвитку хмарних технологій, засобів великих механізмів обробки потокових даних, сучасні системи моніторингу забезпечують високу точність розрахунків та відображення інформації у режимі реального часу. Це

дозволяє вчасно реагувати на потенційні екологічні загрози та приймати обґрунтовані рішення щодо мінімізації їх впливу на здоров'я населення.

Проте, попри значні технологічні досягнення, аналіз наявних рішень свідчить про низку проблем: недостатню гнучкість налаштування, складні або застарілі інтерфейси, низьку адаптивність до мобільних пристроїв, обмежену інтеграцію з альтернативними джерелами даних [6]. Деякі системи орієнтовані переважно на професійне використання, що ускладнює їхнє застосування широкою аудиторією. Таким чином, існує нагальна потреба у створенні нових вебзастосунків, які б поєднували в собі зручність користування, адаптивність до різних пристроїв, сучасний інтерфейс та можливість розширення функціональності. Особливо важливими є функції аналітики, виявлення динаміки змін, формування прогнозів та автоматичного оповіщення користувачів про потенційні екологічні загрози.

1.2 Порівняльний аналіз аналогів

На сьогодні існує чимало вебсистем, призначених для моніторингу стану навколишнього середовища, зокрема якості повітря, рівня шуму, температури, вологості та інших параметрів. Такі системи зазвичай використовуються органами державного контролю, науковими установами або громадськими ініціативами для збору, обробки й візуалізації екологічних даних у режимі реального часу. Аналіз існуючих аналогів дозволяє виявити типові функціональні можливості, сильні та слабкі сторони реалізації, а також визначити аспекти, які можуть бути покращені у рамках розробки нового вебзастосунку.

Серед найбільш відомих і поширених систем можна виокремити такі сервіси:

1. OpenAQ.
2. Plume Labs.
3. BreezoMeter.
4. PurpleAir.

OpenAQ – це некомерційна платформа з відкритим кодом, що надає відкриті дані про якість повітря з усього світу (див. рисунок 1.1). Основна місія

вебзастосунку, зробити екологічну інформацію доступною для всіх за допомогою прозорості та відкритих API [7]. Вона агрегує дані з понад 100 країн, включаючи урядові та незалежні джерела, і надає їх у структурованому вигляді через REST API. OpenAQ підтримує стандартизований підхід до збору та обробки даних, що дозволяє дослідникам, розробникам і громадським організаціям ефективно використовувати ці дані для аналізу тенденцій забруднення, створення візуалізацій, розробки інноваційних сервісів і впровадження екологічних ініціатив. Завдяки відкритому доступу до інформації платформа сприяє підвищенню прозорості екологічного моніторингу та формуванню свідомого ставлення суспільства до проблем якості повітря.

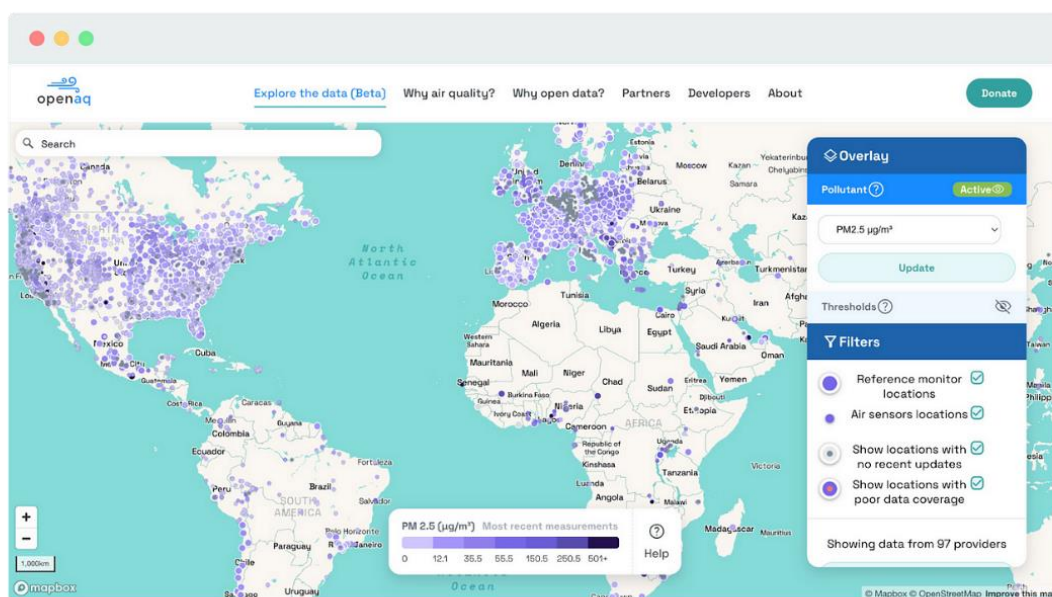


Рисунок 1.1 – Інтерфейс застосунку OpenAQ

OpenAQ не має розвиненого графічного інтерфейсу, як деякі комерційні аналоги, проте є надзвичайно цінним ресурсом для розробників і дослідників, які прагнуть інтегрувати екологічні дані у власні проєкти або проводити глибокий науковий аналіз. Платформа надає відкритий доступ до великого обсягу даних про якість повітря, що збираються з різних країн і регіонів світу, та підтримує

публікацію нових джерел даних, що забезпечує їх актуальність та різноманітність. Крім того, OpenAQ об'єднує активну спільноту розробників, екологів і представників урядових організацій, що сприяє співпраці, обміну знаннями та підвищенню якості екологічного моніторингу на глобальному рівні.

Plume Labs – це французька компанія, яка створила низку інструментів для моніторингу стану повітря. Air Report та персональний сенсор Flow [8]. Сервіс орієнтований переважно на міських жителів, які прагнуть уникати забруднених районів протягом дня. Компанія активно поєднує мобільні технології з машинним навчанням для створення прогнозів якості повітря на основі великої кількості даних з різних джерел. Завдяки інтуїтивному інтерфейсу та візуально привабливим картам користувачі можуть у режимі реального часу відстежувати рівень забруднення в конкретному районі, планувати маршрути з урахуванням екологічної безпеки та отримувати персоналізовані поради щодо зменшення впливу шкідливих речовин. Plume Labs також сприяє підвищенню екологічної обізнаності населення, акцентуючи увагу на важливості щоденного моніторингу повітря (див. рисунок 1.2).

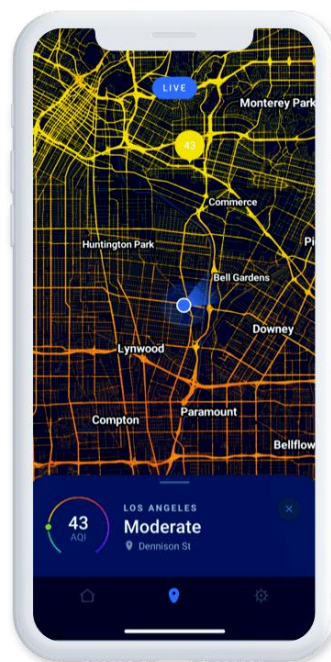


Рисунок 1.2 – Інтерфейс мобільного застосунку Plume Labs

Plume Labs використовує машинне навчання для формування прогнозів якості повітря, пропонуючи дані про рівень забруднення в реальному часі з деталізацією по районах. Додаток має простий інтерфейс, відображає графіки історичних змін, індекси AQI та рекомендації щодо активностей на відкритому повітрі. Унікальною особливістю Plume є фокус на персоналізованих рекомендаціях, які допомагають користувачам змінювати свої маршрути або час прогулянок залежно від рівня забруднення. Завдяки поєднанню наукових підходів і зручного дизайну, Plume Labs стала одним із провідних рішень у сфері щоденного екологічного моніторингу для широкого кола користувачів. Платформа не лише надає доступ до поточних та прогнозованих даних про якість повітря, але й відіграє важливу роль у формуванні екологічно свідомої поведінки серед користувачів. Завдяки відкритості даних та широкій доступності, платформа сприяє підвищенню обізнаності про стан навколишнього середовища і стимулює активну громадську позицію у напрямку захисту природи та поліпшення якості життя.

BreezoMeter – це комерційна платформа, яка надає високоточні дані про якість повітря, пилок, погодні умови та ризики для здоров'я в реальному часі, зображено на рисунку 1.3.

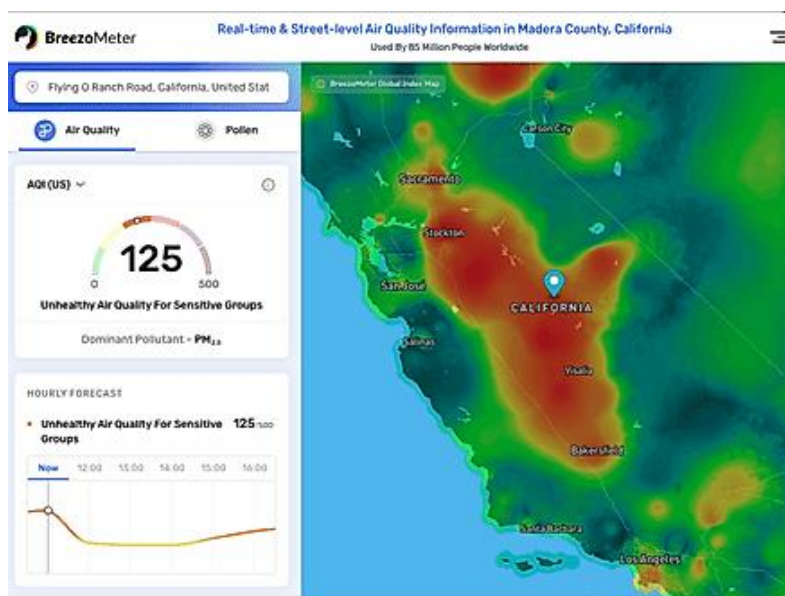


Рисунок 1.3 Інтерфейс застосунку BreezoMeter

Компанія застосовує запатентовані алгоритми для обробки даних з мільйонів сенсорів, супутників і моделей розсіювання забруднень [9]. Особливістю BreezoMeter є надання API та SDK для розробників, що дозволяє інтегрувати екологічні дані у сторонні додатки, пристрої або автомобілі. Сервіс підтримує прогнози на 96 годин, відображає карти забруднення, рівні пилку, УФ-індекси тощо. BreezoMeter використовується у медичних платформах, системах “розумного міста”, екологічних приладах та мобільних додатках. Це одне з найбільш технологічно просунутих рішень на ринку, хоча повний доступ до API є платним.

BreezoMeter поєднує супутникові дані, інформацію з наземних станцій та метеорологічні моделі для забезпечення високої точності оцінки якості повітря в режимі реального часу. Завдяки використанню алгоритмів штучного інтелекту, система адаптує дані під конкретні геолокації, забезпечуючи детальні прогнози з урахуванням локальних умов. Компанія приділяє особливу увагу медичному аспекту забруднення повітря, надаючи користувачам попередження та поради щодо мінімізації ризиків для здоров'я, що робить її інструментом не лише екологічного, а й превентивно-медичного значення.

PurpleAir – це платформа, що базується на спільному зборі даних із тисяч недорогих датчиків PM2.5, які встановлюються користувачами по всьому світу [10]. Система працює за принципом краудсорсингу, що дозволяє збирати велику кількість даних у режимі реального часу. Це дає змогу створювати карту з дуже високою щільністю точок вимірювань, особливо в США, де їхня кількість найбільша. Завдяки такій архітектурі PurpleAir забезпечує широкий охоплення територій та надає користувачам актуальну і деталізовану інформацію про якість повітря в їхніх регіонах. Платформа активно підтримує спільноту користувачів, які можуть не лише отримувати дані, а й ділитися власними вимірами, сприяючи таким чином розвитку краудсорсингових технологій моніторингу довкілля. Окрім показників PM2.5, деякі датчики PurpleAir також здатні вимірювати температуру, вологість та інші параметри, що допомагає створювати комплексний екологічний

профіль для кожної локації (див. рисунок 1.4). Завдяки відкритості даних і простоті використання PurpleAir стає важливим інструментом для громадських кампаній, наукових досліджень та прийняття рішень на рівні місцевих органів влади, сприяючи підвищенню екологічної свідомості населення та покращенню якості повітря.

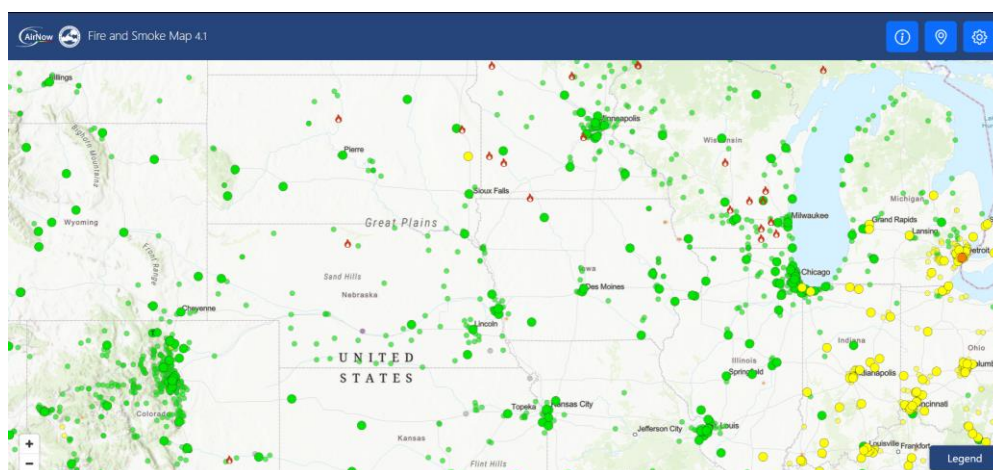


Рисунок 1.4 – Інтерфейс застосунку PurpleAir

PurpleAir орієнтована на громадянське суспільство та активістів, які хочуть самостійно відстежувати якість повітря у своїх районах. Сенсори PurpleAir легко встановити, вони відносно недорогі, мають Wi-Fi-підключення та автоматичну синхронізацію з платформою.

Дані PurpleAir публічно доступні та можуть бути експортовані у різних форматах для подальшого детального аналізу. Завдяки великій кількості встановлених сенсорів платформа забезпечує високочастотний моніторинг із винятковою просторовою роздільністю, що дозволяє відстежувати зміни якості повітря навіть на рівні окремих кварталів чи вулиць. Така деталізація особливо важлива в умовах міської забудови, де локальні джерела забруднення і мікрокліматичні умови суттєво впливають на екологічний стан.

PurpleAir активно стимулює участь громадськості у зборі екологічних даних, що підвищує прозорість інформації про стан довкілля і сприяє формуванню більш обґрунтованих рішень на рівні міських адміністрацій і громадських організацій.

Використання локальних даних із цієї платформи допомагає краще розуміти специфіку забруднення повітря в конкретних районах, що є ключовим для розробки ефективних заходів щодо покращення якості життя і збереження здоров'я населення. Результати порівняння аналогів зведено в таблицю 1.1

Таблиця 1.1 – Порівняльна характеристика аналогів

Критерії	OpenAQ	Plume Labs	BreezoMeter	PurpleAir	Власна розробка
Глобальне покриття	+	-	+	+	+
API для інтеграції	-	-	+	+	+
Доступність	+	+	-	+	+
Підтримка показників PM2.5/PM10	+	+	+	-	+
Користувацька орієнтація	+	+	+	-	+
Загальна оцінка	80%	60%	80%	60%	100%

Відповідно до таблиці порівняльних характеристик, можна зробити висновок, що розробка власного програмного забезпечення для моніторингу якості повітря має низку важливих переваг порівняно з існуючими рішеннями, такими як OpenAQ, Plume Labs, BreezoMeter та PurpleAir. Зокрема, власна система дозволяє забезпечити повноцінну підтримку API, високу доступність для користувачів та орієнтованість на їхні потреби, що критично важливо для практичного використання в різних умовах.

На відміну від сторонніх платформ, індивідуально створений додаток може компенсувати їхні обмеження – зокрема, недостатнє покриття, відсутність окремих функцій або неповну підтримку показників якості повітря (PM2.5, PM10). Завдяки більшій гнучкості та контрольованості процесів, власна система дозволяє реалізувати розширені можливості: підтримку додаткових типів сенсорів, глибший аналітичний функціонал, адаптацію до локальних екологічних стандартів.

Серед основних переваг розробки власного рішення слід виділити:

1. Повна інтеграція з іншими сервісами завдяки підтримці API.
2. Адаптація функціональності до специфічних потреб користувачів і локальних умов.
3. Вища гнучкість інтерфейсу та розширення функцій, які відсутні у конкурентів.
4. Незалежність від зовнішніх постачальників, що зменшує ризики обмеженого доступу до даних.
5. Покращена точність аналізу завдяки повному контролю над джерелами інформації.
6. Краща масштабованість і безпека, необхідні для довготривалої підтримки.

Таким чином, створення власної платформи для моніторингу якості повітря відкриває можливості для більш гнучкої та адаптованої роботи з даними, враховуючи специфічні потреби користувачів [11]. Це дозволяє створити інтерфейси, які краще відповідають вимогам місцевих умов та сприяють легшій інтеграції з іншими екологічними системами. Окрім того, наявність власного програмного забезпечення забезпечує більшу контрольованість над даними, що дає змогу покращити точність прогнозів і результатів аналізу якості повітря, зменшити залежність від зовнішніх постачальників і знизити ризики, пов'язані з обмеженнями доступу до даних чи API. Власний продукт також може розширювати функціональність, додаючи спеціалізовані інструменти для аналізу певних забруднювачів або підтримку нових типів сенсорів, що відповідають за специфічні аспекти якості повітря.

Завдяки високому рівню налаштувань та доступу до розширених функцій, розробка власного модуля може бути важливим кроком до підвищення ефективності моніторингу якості повітря. Це відкриває можливості для більш точного і зручного моніторингу стану навколишнього середовища, що є важливим для покращення здоров'я та якості життя користувачів. Власна розробка дозволяє

враховувати специфічні потреби цільової аудиторії, адаптувати функціональність під конкретні умови використання та інтегрувати систему з локальними сервісами або сенсорними мережами. Такий підхід дає змогу не лише гнучко керувати джерелами даних і форматами виведення, а й забезпечити кращу масштабованість, безпеку та довгострокову підтримку рішення.

1.3 Аналіз методів розв'язання задачі

Розробка програмних засобів для моніторингу якості повітря потребує поєднання сучасних технологій збору, аналізу та візуалізації даних, що відображають стан навколишнього середовища в реальному часі. Оскільки показники якості повітря змінюються як у часовому, так і в просторовому вимірі, для ефективного розв'язання поставленої задачі необхідно застосовувати міждисциплінарний підхід, що охоплює сенсорні системи, математичне моделювання, прогнозування, а також засоби інтерактивної візуалізації.

Одним з ключових напрямів є використання математичних моделей, які дозволяють здійснювати прогнозування рівня забруднення повітря з урахуванням метеорологічних умов, транспортного навантаження, промислових викидів та природних джерел забруднення. Такі моделі забезпечують можливість оцінювати концентрацію шкідливих речовин (PM_{2.5}, PM₁₀, NO₂, SO₂ тощо) у різних точках міста в залежності від часу доби, сезону та погодних умов. Впровадження прогнозних моделей дозволяє органам управління вчасно реагувати на погіршення якості повітря та приймати превентивні заходи для зниження негативного впливу на здоров'я населення.

Важливою складовою системи є організація збору достовірних первинних даних. Для цього застосовуються різні типи стаціонарних та мобільних сенсорів, здатних фіксувати концентрацію забруднювачів з високою точністю. Ретельна калібровка сенсорів, врахування їх технічних характеристик (чутливість, точність, діапазон вимірювання) є необхідною умовою для забезпечення надійності даних.

Розміщення сенсорів у стратегічних точках дозволяє формувати просторову модель забруднення території.

Для підвищення точності оцінювання необхідно інтегрувати дані з метеорологічних станцій, що надають інформацію про температуру повітря, вологість, атмосферний тиск, швидкість і напрямок вітру. Оскільки погодні умови суттєво впливають на розповсюдження забруднювачів, такі дані є критично важливими для побудови коректних моделей. Наприклад, вітрові потоки можуть переміщувати забруднення між районами, а температурні інверсії спричиняють їх накопичення біля поверхні землі.

Додаткове підвищення деталізації моніторингу може бути досягнуто шляхом залучення альтернативних джерел даних, зокрема, з мобільних додатків, громадських сенсорів та краудсорсингових платформ. Такі джерела забезпечують оперативну інформацію про якість повітря у режимі реального часу в місцях, не охоплених офіційною мережею моніторингу.

Також важливо враховувати сезонні та довгострокові тренди, які дають змогу передбачити періоди з підвищеним ризиком забруднення, наприклад, у зимовий період через інтенсивне опалення або влітку під впливом фотохімічних реакцій. Це дозволяє будувати стратегії екологічного планування та підвищувати обґрунтованість управлінських рішень.

Таким чином, для ефективного вирішення задачі моніторингу якості повітря необхідне поєднання кількох методів:

- 1) сенсорного збору просторово-часових даних;
- 2) математичного моделювання та прогнозування;
- 3) метеорологічної корекції моделей;
- 4) інтеграції альтернативних джерел даних;
- 5) аналітики трендів.

Комплексне застосування зазначених методів дозволяє створити надійний та функціонально повний програмний засіб моніторингу стану атмосферного повітря.

Створення ефективного програмного забезпечення для моніторингу якості повітря вимагає комплексного підходу, що поєднує різноманітні технології, джерела даних та аналітичні методи. Ключовим елементом системи є збір достовірних екологічних даних, який здійснюється за допомогою сенсорів, розміщених у різних географічних точках. Ці пристрої здатні вимірювати концентрацію забруднюючих речовин у реальному часі, надаючи локальні дані про стан атмосфери. Додатково до сенсорів використовуються дані з метеорологічних станцій, які дозволяють враховувати погодні умови, що безпосередньо впливають на поширення забруднювачів. Зовнішні джерела, такі як супутникові знімки, відкриті екологічні API та громадські платформи, також можуть доповнювати інформаційну базу, забезпечуючи ширше охоплення території.

Обробка зібраної інформації вимагає застосування потужних методів статистичного аналізу та алгоритмів машинного навчання. За їх допомогою можливо не лише очищувати та узагальнювати дані, а й здійснювати прогнозування майбутніх рівнів забруднення. Алгоритми враховують широкий спектр факторів, зокрема сезонність, добову динаміку, інтенсивність дорожнього руху, погодні умови та інші показники людської активності. Завдяки машинному навчанню можна будувати адаптивні моделі, які з часом підвищують точність прогнозів на основі накопичених історичних даних. Математичне моделювання забруднення також займає важливе місце у системі, оскільки дозволяє описувати динаміку розповсюдження речовин у повітрі з урахуванням топографічних та метеорологічних умов.

Важливим аспектом є ефективна візуалізація результатів, яка забезпечує зрозуміле уявлення про поточний стан якості повітря. Для цього широко використовуються інтерактивні графіки, які дозволяють відображати зміну рівня забруднення в часі. Бібліотеки для побудови графіків, такі як Chart.js, забезпечують інструменти для створення динамічних візуалізацій, з якими можуть взаємодіяти користувачі. Крім того, інтерактивні карти, реалізовані за допомогою таких інструментів, як Leaflet чи Google Maps API, дозволяють відображати географічний

розподіл забруднювачів, що дає змогу візуально оцінити ситуацію в конкретному районі чи місті. Хоча тривимірне моделювання може бути корисним для наукових досліджень або візуалізації складних екологічних процесів, у практичних користувацьких системах зазвичай застосовуються двовимірні візуальні елементи, які є зручнішими для сприйняття та менш ресурсомісткими.

У результаті, ефективна система моніторингу якості повітря базується на синергії технологій збору даних, аналітичних методів і засобів візуалізації. Така інтеграція дозволяє забезпечити точне відображення поточної екологічної ситуації, своєчасно попереджати про погіршення якості повітря та сприяти прийняттю рішень, спрямованих на покращення стану навколишнього середовища. Візуалізація даних в тривимірному просторі дозволяє користувачам бачити, як змінюється якість повітря в різних частинах міста або регіону, а також допомагає виявляти «гарячі точки» забруднення, де концентрація шкідливих речовин є найбільш критичною [12]. Це може бути особливо корисно для органів влади, екологічних організацій і громадських активістів, оскільки таке моделювання дозволяє оперативно реагувати на зміни та вжити необхідних заходів для покращення стану навколишнього середовища.

Одним із ключових підходів до моніторингу якості повітря є використання датчиків для вимірювання концентрації основних забруднювачів, таких як PM_{2.5}, PM₁₀, AQI, NO, O₃, SO₂ та NH₃. Дані, отримані з цих сенсорів, дозволяють отримати точну інформацію про рівень забруднення в конкретних точках території та ефективно відображати їх на карті або у вигляді графіків. Проте цей метод має певні обмеження: точність вимірювань залежить від щільності розташування датчиків, що не завжди забезпечує повне покриття всієї території, а також потребує значних обчислювальних ресурсів для обробки великого обсягу даних у режимі реального часу [13]. Тому для досягнення максимальної ефективності рекомендується поєднувати дані з датчиків із додатковими джерелами інформації та алгоритмами прогнозування.

Для подолання цих обмежень доцільно поєднувати дані з фізичних сенсорів із результатами моделювання, що базується на прогнозних алгоритмах та метеорологічних даних. Зокрема, використання інтерполяційних методів, таких як зворотне зважене інтерполювання або метод Крігінга, дозволяє заповнити прогалини між точками вимірювання і створити більш цілісну картину забруднення в просторі. Поєднання цих даних з інформацією про рельєф місцевості, напрямок і силу вітру, вологість та інші атмосферні умови допомагає покращити точність тривимірної моделі.

Інший метод полягає в використанні моделей на основі машинного навчання та штучного інтелекту для прогнозування якості повітря. Ці моделі дозволяють передбачати рівень забруднення на основі історичних даних та поточних умов, таких як погодні умови, транспортні потоки та інші фактори. Застосування машинного навчання дозволяє враховувати взаємодію численних змінних, що значно підвищує точність прогнозів. Однак цей підхід також має свої недоліки, зокрема, потребує великої кількості даних для тренування моделей, а також може бути складним для інтеграції в систему в реальному часі.

Додатково, одним з методів для вдосконалення точності моделювання є використання геоінформаційних систем [14]. Вони дозволяють не тільки відображати дані на карті, а й здійснювати їх аналіз з урахуванням територіальних характеристик, таких як рельєф, забудова, напрямок вітру тощо. Цей метод дозволяє створювати точніші прогнози якості повітря на конкретних ділянках території, оскільки враховуються не тільки самі дані про забруднення, але й їх контекст у середовищі. Однак для ефективного використання геоінформаційних систем необхідно мати актуальні та точні картографічні дані, а також інфраструктуру для збору та обробки великих обсягів інформації [15].

Інтеграція геоінформаційних систем із тривимірним моделюванням відкриває нові можливості для аналізу розповсюдження забруднювачів у складних урбаністичних умовах. Зокрема, врахування висоти будівель, густоти забудови та наявності природних бар'єрів дозволяє моделювати потоки повітря більш

реалістично, що критично важливо для мегаполісів. Таке поєднання технологій може бути використане не лише для моніторингу, але й для планування міського розвитку, оцінки впливу інфраструктурних додатків на довкілля та підвищення екологічної безпеки. У результаті це сприяє формуванню більш стійких і здорових міських просторів.

Для моніторингу якості повітря можна використовувати методи візуалізації на основі двовимірних карт, де показуються дані про рівень забруднення на конкретних територіях. Завдяки цьому користувачі можуть побачити, як змінюється якість повітря в різних районах, враховуючи місцезнаходження і основні фактори, які впливають на забруднення. Такий підхід дозволяє створити точну та зручну для користувача картографічну візуалізацію.

Для забезпечення найкращих результатів необхідно комбінувати дані, отримані від датчиків, з інформацією про погодні умови, транспортні потоки та інші фактори. Врахування таких даних дозволяє створити систему, яка буде точно відображати рівень забруднення в реальному часі. Інтеграція даних з геоінформаційними системами також дозволяє покращити точність картографії, створюючи більш деталізовані прогнози якості повітря для конкретних територій [16]. Така інтегрована система дозволить створити платформу, яка стане важливим інструментом для моніторингу навколишнього середовища та поліпшення здоров'я населення.

Окрім цього, важливою складовою є побудова механізмів автоматичного аналізу та сповіщення, які дозволяють своєчасно реагувати на критичні зміни рівня забруднення. Це особливо актуально для вразливих груп населення, таких як діти, літні люди або особи з хронічними захворюваннями дихальних шляхів. Забезпечення користувачів персоналізованими повідомленнями про стан повітря у їхній місцевості може підвищити екологічну обізнаність та сприяти зміні щоденних звичок у напрямку безпечнішого способу життя. У довгостроковій перспективі така система може відігравати роль не лише інструмента моніторингу, а й

превентивного засобу впливу на екологічну політику та планування міського розвитку.

Таким чином, для розробки ефективного програмного засобу для моніторингу якості повітря з використанням тривимірної моделювання необхідно застосовувати комплексний підхід, який поєднує технології збору даних, аналізу, прогнозування та візуалізації, що забезпечить найкращі результати. Важливо враховувати можливість масштабування такої системи для роботи з великими обсягами даних у режимі реального часу, що дозволить адаптувати її до різних регіонів та сценаріїв використання. Впровадження хмарних технологій та машинного навчання може значно підвищити ефективність обробки інформації та генерації прогнозів, роблячи систему більш гнучкою, адаптивною та точною у відображенні поточної екологічної ситуації.

Крім того, важливо інтегрувати сучасні вебтехнології для забезпечення зручного та оперативного доступу до аналітичних даних з будь-яких пристроїв. Використання інтерактивних графіків і карт дає можливість користувачам у реальному часі відслідковувати динаміку забруднення повітря, отримувати миттєві сповіщення про перевищення допустимих норм, а також здійснювати порівняльний аналіз екологічних показників між різними локаціями. Автоматизована обробка даних із сенсорів, інтуїтивний і гнучкий інтерфейс управління, а також можливість персоналізації налаштувань відповідно до потреб різних категорій користувачів значно підвищують ефективність прийняття екологічно обґрунтованих рішень як на локальному, так і на регіональному рівнях. Такий підхід сприяє формуванню більш свідомого ставлення до навколишнього середовища та підтримує сталий розвиток громад.

1.4 Постановка задач бакалаврської роботи

Проаналізувавши існуючі підходи до моніторингу якості повітря, а також вимоги до сучасних вебзастосунків для відображення екологічних даних, було

сформульовано такі основні задачі, які необхідно реалізувати в межах бакалаврської роботи:

- 1) розробити архітектуру вебзастосунку для моніторингу якості повітря з урахуванням принципів модульності та масштабованості;
- 2) спроектувати структуру клієнтської частини з інтерактивним відображенням даних про стан повітря;
- 3) реалізувати модуль для побудови графіків зміни рівнів забруднювачів повітря у часі;
- 4) реалізувати модуль для відображення інтерактивної карти з нанесенням показників якості повітря в різних географічних точках;
- 5) створити модуль для отримання актуальних екологічних даних через API з відкритих джерел;
- 6) забезпечити зручний інтерфейс користувача з можливістю фільтрації, пошуку та перегляду детальної інформації;
- 7) провести тестування функціональних модулів системи та забезпечити інструкцію з використання для кінцевого користувача.

1.5 Висновки

У першому розділі проведено детальний аналіз сучасних програмних засобів для моніторингу якості повітря. Розглянуто такі популярні платформи, як OpenAQ, Plume Labs, BreezoMeter та PurpleAir, з акцентом на їхні можливості збору, обробки та візуалізації екологічних даних. Особлива увага приділена ключовим підходам, серед яких використання різноманітних сенсорів, відкритих екологічних API, картографічних сервісів та аналітичних інструментів. Виявлено, що більшість існуючих систем орієнтовані переважно на базове відображення даних і часто не забезпечують достатньої зручності для кінцевого користувача, а також мають обмежені можливості інтеграції з актуальними джерелами даних у реальному часі, що створює виклики для оперативного та точного моніторингу стану навколишнього середовища.

2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ

2.1 Аналіз архітектури програмного забезпечення

Архітектура програмного забезпечення є визначальним чинником у процесі створення будь-якої інформаційної системи, адже саме вона формує структуру рішення, визначає способи взаємодії між його компонентами, забезпечує можливості масштабування, супроводу, тестування та подальшого розвитку. У контексті розробки вебзастосунку для моніторингу якості повітря архітектурі відведено ключову роль у досягненні гнучкості, модульності, можливості інтеграції з зовнішніми джерелами даних, а також у забезпеченні ефективної візуалізації інформації та зручного користувацького інтерфейсу [17].

Особливу увагу приділено здатності системи обробляти великі обсяги даних у режимі реального часу, стабільно функціонувати за умов нестабільного мережевого з'єднання, а також дотриманню вимог інформаційної безпеки. Передбачено використання багаторівневої архітектури, що охоплює клієнтську частину, серверну логіку та додаткові сервіси. Клієнтська частина відповідає за взаємодію з користувачем, серверна обробляє запити й взаємодіє з базою даних, а допоміжні сервіси здійснюють роботу з API, побудову графіків, формування прогнозів та генерацію візуальних елементів. Така структура сприяє адаптації системи до нових функціональних або нефункціональних вимог, а також дозволяє масштабувати її в разі зростання навантаження.

Клієнтський рівень реалізовано за допомогою сучасних вебтехнологій: HTML5 забезпечує семантичну структуру сторінок, CSS3 відповідає за адаптивне й привабливе оформлення інтерфейсу, а JavaScript дозволяє реалізувати динамічні функції, включаючи асинхронне оновлення даних і інтерактивну взаємодію з користувачем. Інтерфейс побудовано відповідно до принципів UX-дизайну, що гарантує інтуїтивну зрозумілість, доступність і мінімальні витрати часу на навчання користувача.

Для візуалізації інформації використовуються спеціалізовані бібліотеки, зокрема Chart.js або Recharts, які дозволяють створювати інтерактивні графіки з різними типами відображення даних. Значну роль відіграє інтерактивна карта, реалізована за допомогою бібліотеки Leaflet.js. Вона забезпечує високу продуктивність, зручність адаптації під мобільні пристрої та можливість інтегрувати географічну інформацію з різних шарів. Завдяки цьому користувачі можуть оперативно переглядати рівень забруднення повітря у вибраних регіонах у зручному для сприйняття форматі.

Клієнтська частина також відповідає за обробку реєстрації та авторизації користувачів, перевірку введених даних і динамічне оновлення вмісту сторінки в залежності від запитів до серверного API. Серверна частина реалізує RESTful API, що дозволяє обробляти HTTP-запити та повертати результати у форматі JSON, забезпечуючи чітке розділення логіки та представлення, а також можливість легкої інтеграції з іншими клієнтами, наприклад, мобільними застосунками [18].

З метою підвищення ефективності обробки інформації реалізовано кешування відповідей сторонніх API, що знижує загальне навантаження на систему. Серверна логіка також охоплює обробку помилок, логування подій і захист даних шляхом аутентифікації користувачів, перевірки сесій та валідації введеної інформації. Для забезпечення надійної та швидкої роботи з базою даних реалізовано оптимізовані SQL-запити, а також механізми резервного відновлення, що дозволяють оперативно повернути систему до працездатного стану в разі збоїв.

Інтегровані системи моніторингу дозволяють серверу відстежувати основні параметри його роботи в режимі реального часу. Це забезпечує можливість своєчасного виявлення аномалій і запобігання потенційним проблемам, що значно підвищує стабільність функціонування всього застосунку. Загалом, реалізовані механізми забезпечують високу швидкість обробки запитів, можливість масштабування та мінімізацію часу на усунення збоїв.

Усі дані, включаючи інформацію про користувачів, запити, налаштування, геолокацію та історію вимірювань, зберігаються в базі даних MySQL. Структура

бази даних розроблена відповідно до принципів реляційної моделі з чітким визначенням зв'язків між таблицями, серед яких: users, locations, measurements, api_logs, favorites та інші. Для прискорення доступу до інформації застосовуються індекси до часто використовуваних полів, що дозволяє значно зменшити час виконання запитів і підвищити загальну продуктивність системи.

База даних не лише зберігає поточні показники якості повітря, але й систематично накопичує історичні дані, що дає змогу здійснювати комплексний аналіз змін екологічних параметрів у часовому розрізі, виявляти закономірності та аномалії, а також створювати точні прогностичні моделі для оцінки майбутніх тенденцій і прийняття обґрунтованих рішень у сфері екологічного моніторингу. Інформація регулярно імпортується з різноманітних зовнішніх джерел, включно з автоматизованими станціями моніторингу, системами супутникового спостереження та іншими датчиками, що забезпечує широку і детальну картину стану довкілля. Автоматизовані процеси очищення, фільтрації та архівування застарілих або неактуальних даних гарантують підтримання високої якості й достовірності інформації, а також ефективне використання ресурсів системи. Для забезпечення безперебійної роботи та масштабованості платформи реалізовано механізм реплікації даних, який дозволяє рівномірно розподіляти навантаження між кількома серверами, що значно підвищує швидкість обробки запитів, забезпечує відмовостійкість і сприяє безпеці збережених даних, а також підтримує можливість швидкого відновлення системи у разі збоїв. Крім того, система передбачає гнучкі механізми доступу і контролю прав користувачів, що дозволяє безпечно управляти інформацією і надавати різні рівні доступу залежно від ролі та повноважень користувачів.

Особливе місце в архітектурі займає модуль інтеграції з екологічними API, який регулярно надсилає запити, обробляє отримані відповіді та адаптує дані до внутрішньої структури системи. Завдяки цьому забезпечується безперервне оновлення інформації про стан навколишнього середовища.

З метою дотримання вимог безпеки реалізовано захист від XSS- і CSRF-атак. Для протидії XSS використовується автоматичне очищення вхідних даних, що унеможлиблює впровадження шкідливого JavaScript-коду в сторінки. Захист від CSRF забезпечується за допомогою валідації токенів, що гарантує легітимність запитів до сервера. Передача даних здійснюється через протокол HTTPS, який забезпечує шифрування інформації та захищає її від перехоплення або підміни.

Контроль доступу до функцій додатку реалізовано через механізми автентифікації та авторизації. Кожен користувач має певний рівень доступу, що дозволяє, наприклад, обмежити можливість редагування або додавання вимірювань лише для авторизованих осіб з відповідними правами. Такий підхід гарантує збереження цілісності даних і захист від несанкціонованих дій.

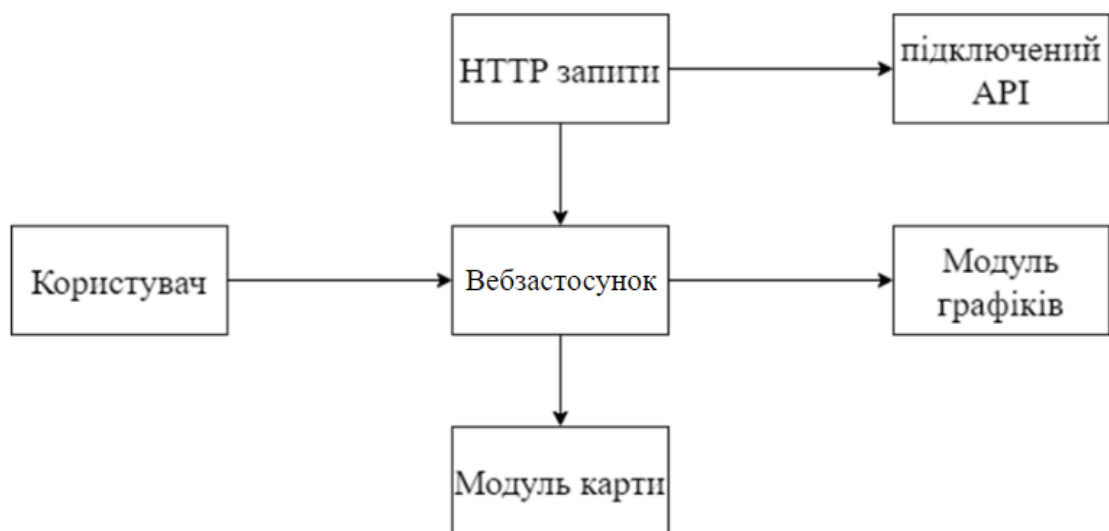


Рисунок 2.1 – Архітектурна схема вебзастосунку

Архітектура вебзастосунку спроектована з урахуванням подальшої масштабованості та технологічної гнучкості. Модульний підхід дозволяє легко інтегрувати нові функціональні компоненти – зокрема, додавання типів візуалізацій, форматів даних, джерел API або розширення інтерфейсних можливостей. Така структура спрощує адаптацію системи до сучасних технологій, як-от Node.js для серверної логіки чи React.js для динамічного клієнтського

інтерфейсу. Це забезпечує ефективний розвиток вебзастосунку без ризику порушення його стабільності чи безпеки.

Така архітектура дає змогу легко здійснити перехід до сучасних технологій, зокрема через інтеграцію з Node.js для серверної частини. Це відкриває нові можливості для обробки даних і розширення функціональності. Використання React.js для створення динамічних інтерфейсів значно підвищує інтерактивність та зручність користування вебзастосунком. Завдяки такій структурі, масштабування системи стає гнучким та безпечним. Кожен новий модуль можна безперешкодно додати до існуючої системи, не порушуючи її стабільності та безпеки. Це важливо для зниження ризиків при розширенні функціоналу, особливо у випадках, коли потрібно оперативно реагувати на зміни у вимогах або впроваджувати нові можливості. Таким чином, обрана архітектура не лише відповідає сучасним стандартам розробки, а й створює надійну основу для подальшого розвитку вебзастосунку, забезпечуючи його ефективність, масштабованість і надійність.

2.2 Розробка структури вебзастосунку

Структура вебзастосунку є критично важливим аспектом проєктування системи, оскільки вона визначає, яким чином реалізовані компоненти взаємодіють між собою та як дані передаються, обробляються і візуалізуються. У межах розробки вебзастосунку для моніторингу якості повітря була застосована модульна структура з чітким розмежуванням функціональних зон. Такий підхід дозволяє спростити розгортання, тестування, обслуговування та масштабування системи.

Основна мета структури вебзастосунку полягає у забезпеченні стабільної, інтерактивної та інформативної платформи для перегляду даних про стан навколишнього середовища в реальному часі. З цією метою було реалізовано чіткий поділ на окремі компоненти, що дозволяє досягти максимальної ефективності та гнучкості системи. Така організація дає змогу незалежно розвивати кожен частину системи, оптимізуючи її для забезпечення найкращої продуктивності, масштабованості та безпеки.

Завдяки модульному підходу кожен компонент системи може бути протестований окремо, що сприяє підвищенню надійності та забезпеченню гнучкої підтримки і подальшого розвитку вебзастосунку.

Фронтенд відповідає за взаємодію з користувачем, забезпечуючи інтуїтивно зрозумілий інтерфейс для візуалізації даних. Це включає інтерактивні елементи, такі як графіки, карти та таблиці, що дозволяють користувачам швидко та зручно отримувати необхідну інформацію про якість повітря, температурні зміни та інші екологічні показники. Для реалізації цього використано сучасні вебтехнології, що дозволяють забезпечити динамічний та адаптивний інтерфейс, який підлаштовується під різні пристрої та екрани.

Бекенд відповідає за обробку даних, що надходять від різних джерел, їх зберігання в базі даних та надання необхідних запитів через API. Тут реалізовано основну логіку обробки даних, інтеграцію з зовнішніми джерелами даних (наприклад, датчики якості повітря, геоінформаційні системи), а також забезпечення безпеки та управління користувацькими сесіями. Бекенд здійснює моніторинг і логування запитів, забезпечує кешування даних для підвищення продуктивності та зменшення навантаження на систему.

Основними компонентами структури вебзастосунку є:

1. HTML, CSS, JavaScript файли, вони забезпечують відображення інтерфейсу користувача, динамічне оновлення контенту, інтерактивну побудову графіків та завантаження карти.

2. Модуль побудови графіків – відповідає за візуалізацію статистики забруднення повітря у вигляді лінійних діаграм. Реалізується за допомогою бібліотеки Chart.js.

3. Модуль інтерактивної карти – дозволяє відображати географічне розташування екологічних показників. Побудований з використанням Leaflet.

4. Модуль API-запитів – виконує запити до сторонніх сервісів для отримання актуальних екологічних даних.

5. Файл конфігурації API – містить ключ доступу та базові налаштування запитів.

6. Файл script.js – координує дії між усіма модулями, викликає функції оновлення графіків, завантаження карти, а також реалізує логіку автозавантаження даних при відкритті сторінки.

7. Файл styles.css – забезпечує зовнішнє оформлення інтерфейсу.

2.3 Розробка моделі системи

Для кращого розуміння логіки роботи вебзастосунку моніторингу якості повітря була створена модель системи у вигляді UML-діаграми діяльності, яка детально демонструє послідовність дій користувача та взаємодію ключових модулів системи [19]. Спочатку користувач відкриває головну сторінку вебзастосунку, після чого відбувається автоматичне підключення двох основних модулів, а саме інтерактивної карти та модуля графіків, що є базовими елементами інтерфейсу. Модуль інтерактивної карти надає можливість вибирати конкретні географічні точки або регіони для отримання екологічних показників, що значно спрощує навігацію та орієнтацію у просторі. Після вибору точки користувачем система ініціює запит до модуля API, який відповідальний за обробку координат та збір актуальних даних з зовнішніх джерел або баз даних. Цей модуль повертає детальні екологічні параметри, серед яких основними є концентрації дрібнодисперсних часток PM2.5 та PM10, загальний індекс якості повітря AQI, а також рівні основних шкідливих газів: оксид азоту (NO), озон (O₃), діоксид сірки (SO₂) та аміак (NH₃). Інтеграція таких комплексних даних дозволяє користувачам отримувати повну картину забруднення атмосфери у вибраному регіоні. Модуль графіків у свою чергу візуалізує отримані показники у вигляді динамічних діаграм, що оновлюються в реальному часі, забезпечуючи зручний спосіб аналізу змін рівня забруднення протягом певного періоду. Завдяки такій архітектурі забезпечується висока гнучкість системи, що дозволяє легко масштабувати її функціонал, додавати нові типи даних або оновлювати інтерфейс без суттєвих змін у загальній структурі

вебзастосунку. Таким чином, користувач отримує інтуїтивно зрозумілий інструмент для моніторингу та оцінки якості повітря з урахуванням різних параметрів, що сприяє підвищенню обізнаності про стан навколишнього середовища і прийняттю обґрунтованих рішень для його покращення.

Отримані дані передаються до модуля графіків, який буде візуалізацію обраних показників у вигляді лінійних графіків. Після цього користувач має можливість проаналізувати інформацію, завершити роботу з додатком, занести дані у окремий файл або обрати іншу точку на карті для оновлення графіка (див. рисунок 2.2).



Рисунок 2.2 – Модель системи у вигляді діаграми діяльності програмних засобів

Цей сценарій ілюструє типову послідовність взаємодії користувача з вебзастосунком, засновану на взаємодії фронтенд-модулів із серверною частиною через API.

2.4 Розробка алгоритму

Згідно з даним алгоритмом, першим кроком є завантаження основного інтерфейсу вебзастосунку для моніторингу якості повітря, який надає користувачу доступ до основного функціоналу, включаючи перегляд даних про стан повітря в реальному часі та візуалізацію цих даних на графіках. Далі, користувач може вибрати конкретну географічну зону для перегляду даних по типу забруднення. Після вибору зони, система завантажує відповідні дані з зовнішнього API, яке надає інформацію від сенсорних пристроїв. Якщо дані не знайдено або виникли проблеми з підключенням до API, система виведе повідомлення про помилку і запропонує користувачу спробувати ще раз. У випадку успішного завантаження даних, програма відображає графіки якості повітря за обраний період.

Система також дозволяє переглядати дані на карті, де користувач може побачити стан повітря в різних точках міста або області. Для цього вебзастосунок використовує інтеграцію з картографічним сервісом, що дозволяє відображати забруднення на карті в реальному часі. Користувач може вибрати точку на карті для детальнішого перегляду даних про стан повітря в конкретному місці. Крім того, на основі обраних даних вебзастосунок генерує детальні звіти про стан повітря, які включають не тільки показники забруднення.

Звіти формуються динамічно та можуть бути експортовані у формат CSV для подальшого аналізу або збереження. Користувач має змогу зберігати зібрані дані локально, що дозволяє здійснювати порівняння показників за різні дні або локації. Також передбачена функція побудови графіків у часовому вимірі, що дає змогу відстежувати зміни якості повітря з плином часу. Важливо, що весь процес обробки інформації є асинхронним, тобто не потребує перезавантаження сторінки – завдяки використанню AJAX або WebSocket-з'єднань. Це суттєво покращує взаємодію з користувачем і забезпечує високу динамічність системи.

У разі тривалого відсутності відповіді з боку API, реалізовано механізм повторних спроб із затримкою, а також логування помилок для подальшого аналізу технічними фахівцями. Такий підхід забезпечує стійкість вебзастосунку до

мережевих збоїв та сприяє підвищенню надійності загальної системи. Користувач, у свою чергу, завжди інформується про поточний статус підключення та успішність операцій через відповідні елементи інтерфейсу. Інтерактивні елементи карти дозволяють користувачеві збільшувати або зменшувати масштаб. Візуалізація забруднення здійснюється за допомогою кольорових позначок або графічних елементів, що відображають рівень забруднення повітря в різних точках. Завдяки таким функціям, вебзастосунок забезпечує користувачам доступ до інтуїтивно зрозумілої, актуальної та детальної інформації про стан навколишнього середовища в реальному часі.

Основною моделлю системи є взаємодія між користувачем, графічним інтерфейсом і серверами, які обробляють запити. Програма працює на основі клієнт-серверної архітектури, де сервери зберігають актуальні дані та надають їх за запитом користувача. Сервери також обробляють зовнішні API-запити для отримання актуальної інформації від сенсорних пристроїв або екологічних станцій. Серверна частина системи не лише обробляє запити користувачів, а й виконує запити до зовнішніх екологічних API для отримання актуальних показників якості повітря.

Користувач за допомогою браузера або іншого клієнтського застосунку взаємодіє з інтерфейсом, ініціюючи запити щодо перегляду або оновлення екологічних даних. Ці запити надсилаються до сервера, який виконує функцію посередника між інтерфейсом та джерелами даних.

Графічне представлення моделі, яке включає потоки даних, взаємодію між компонентами та основні кроки роботи користувача, дозволяє візуалізувати роботу системи та зробити процес розробки зрозумілішим.

Розробка моделі системи є важливим етапом, що дозволяє правильно організувати всі дані, забезпечити зручний інтерфейс та безперебійну роботу з зовнішніми джерелами даних. Важливою частиною цієї моделі є забезпечення зручності користування та максимальна інтуїтивність у взаємодії з додатком, що зображено на рисунку 2.3.



Рисунок 2.3 – Алгоритм роботи програмного застосунку

Функціонал вебзастосунку розроблено з урахуванням основних потреб користувачів та вимог до моніторингу якості повітря. Він включає відображення актуальних даних про концентрацію основних забруднювачів, таких як PM2.5, PM10, NO, O₃, SO₂ та NH₃, що дозволяє користувачам отримувати повну картину екологічної ситуації у вибраному регіоні. Для зручності аналізу інформації передбачено інтерактивну карту, яка дає змогу обирати конкретні точки вимірювань, а також візуалізацію даних у вигляді графіків із можливістю вибору часових проміжків.

Механізм автоматичного оновлення інформації забезпечує безперебійну роботу з зовнішніми джерелами даних і підтримує актуальність екологічних показників у режимі реального часу. Особливу увагу приділено масштабованості

системи, що дозволяє в майбутньому інтегрувати нові джерела даних або розширювати функціонал без необхідності значних змін у базовій архітектурі. Інтерфейс вебзастосунку адаптовано для комфортного використання на різних пристроях, включаючи десктопи та мобільні гаджети. Завдяки оптимізації запитів та використанню кешування забезпечується висока продуктивність і швидкість обробки даних.

Це забезпечує користувачам комплексний та зручний інструмент для оцінки екологічного стану, сприяє підвищенню рівня обізнаності щодо забруднення повітря та підтримує прийняття обґрунтованих рішень, спрямованих на покращення якості життя в регіоні. Завдяки продуманому функціоналу вебзастосунк не лише відображає дані, але й створює умови для їх глибокого аналізу, що робить систему незамінним помічником як для пересічних користувачів, так і для фахівців у сфері екології та охорони здоров'я.

Впровадження таких рішень сприяє формуванню екологічної культури серед населення та стимулює активну громадську позицію у питаннях збереження довкілля. Можливість отримувати оперативні дані в режимі реального часу допомагає своєчасно реагувати на потенційні екологічні загрози та планувати відповідні заходи на локальному і регіональному рівнях.

2.5 Висновки

У другому розділі було здійснено аналіз архітектури програмного забезпечення та сформовано архітектурну схему вебзастосунку. Також була розроблена логічна структура вебдодатку, що відображає взаємозв'язки між його основними компонентами. На основі визначених функціональних вимог було побудовано основний алгоритм роботи програмного засобу. Для детального представлення моделі функціонування системи були створені UML-діаграми, що ілюструють ключові аспекти взаємодії компонентів, процесів обробки даних та сценарії використання додатку.

3 РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ

3.1 Варіантний аналіз і обґрунтування вибору засобів реалізації програмного продукту

При розробці вебзастосунку для моніторингу навколишнього середовища важливо обрати оптимальні технології для реалізації клієнтської та серверної частин, а також засоби для візуалізації даних, інтеграції з API та забезпечення користувацької взаємодії. З урахуванням функціональних вимог до системи, таких як інтерактивна візуалізація показників якості повітря, інтеграція з картою місцевості та можливість авторизації користувачів, було обрано такі технології:

Для клієнтської частини був обраний стек технологій, що забезпечує швидку та ефективну взаємодію з користувачем. Використання HTML5 дозволяє забезпечити чітке та структуроване подання інформації, а CSS3 – створює гнучкі та адаптивні стилі, що забезпечують зручний інтерфейс на різних пристроях. JavaScript в свою чергу забезпечує інтерактивність сайту, зокрема для динамічної зміни контенту та роботи з API.

Express.js – це популярний фреймворк для Node.js, який полегшує створення серверної частини вебзастосунків. Він забезпечує простоту маршрутизації запитів, обробки HTTP-запитів і відповідає за швидкий і зручний доступ до RESTful API. Express.js дозволяє ефективно управляти запитами користувачів, обробляти дані і формувати відповіді у форматі JSON, що ідеально підходить для інтеграції з фронтенд-частиною вебзастосунку.

Chart.js – це потужна JavaScript-бібліотека для побудови інтерактивних графіків, яка дозволяє в реальному часі візуалізувати зміну параметрів якості повітря, таких як PM2.5, PM10, AQI, NO, O₃, SO₂ та NH₃. З її допомогою можна створювати різноманітні типи графіків, включаючи лінійні, стовпчасті, кругові та інші, що дає можливість гнучко налаштовувати вигляд і поведінку графіків [20]. Завдяки цим можливостям, Chart.js є ідеальним інструментом для візуалізації екологічних даних, оскільки дає змогу користувачам аналізувати зміни у рівнях

забруднення повітря за певний період часу, порівнювати дані між різними локаціями або вимірювати ефективність заходів з покращення якості повітря.

Однією з основних переваг використання Chart.js є його простота у використанні. Бібліотека дозволяє швидко інтегрувати графіки в вебзастосунки за допомогою декількох рядків коду, при цьому надаючи багатий набір налаштувань для кастомізації. Крім того, Chart.js підтримує анімації, що дозволяє зробити відображення змін у даних більш динамічним та наочним для користувачів. Це важливо, оскільки дані про якість повітря можуть швидко змінюватися, і користувачам необхідно бачити актуальну інформацію в реальному часі.

Leaflet.js – це легка та потужна JavaScript-бібліотека для створення інтерактивних карт [21]. Вона особливо корисна для проектів, що передбачають відображення геолокаційних даних, таких як розміщення сенсорів моніторингу якості повітря на карті. Завдяки своїй простоті, Leaflet дозволяє створювати карти, на яких можна відображати точні координати сенсорів, а також динамічно оновлювати їх положення в реальному часі. Вебзастосунок на базі Leaflet.js може використовувати карти OpenStreetMap або інші картографічні сервіси, що знижує навантаження на систему та дозволяє інтегрувати картографічні дані з мінімальними витратами ресурсів.

Окрім того, Leaflet.js підтримує різноманітні додаткові функціональності, такі як накладання шарів даних, масштабування карти, інтерактивні позначки (маркери) для сенсорів та спливаючі вікна з додатковою інформацією. Це дозволяє користувачам отримувати детальну інформацію про стан якості повітря безпосередньо на карті, що є дуже корисним для аналізу просторових патернів забруднення.

Node.js та Express.js обрано для реалізації серверної частини вебзастосунку, що забезпечує надійне оброблення запитів та взаємодію з базою даних [22]. Node.js є серверною платформою, що дозволяє використовувати JavaScript на стороні сервера. Це дає перевагу у вигляді єдності мови програмування на всіх етапах розробки (як на фронтенді, так і на бекенді), що спрощує процес підтримки та

розширення додатку. За допомогою Node.js можна створювати масштабовані серверні рішення, що підтримують асинхронну обробку запитів, що особливо важливо для реального часу в моніторингу якості повітря. OpenWeatherMap API та інші забезпечує доступ до актуальних метеорологічних та екологічних даних. Дані з API інтегруються в додаток через JavaScript та обробляються для відображення в інтерфейсі користувача[23].

Об'єднання перелічених технологій дозволяє створити ефективний, масштабований та зручний у користуванні вебзастосунок для моніторингу навколишнього середовища. Використання JavaScript як універсальної мови забезпечує єдність коду, а сучасні бібліотеки та фреймворки дозволяють швидко створювати складні елементи взаємодії з даними та картами.

Отже, у цьому підрозділі було обґрунтовано вибір основних засобів реалізації вебзастосунку для моніторингу навколишнього середовища, мови програмування JavaScript, HTML і CSS для побудови клієнтської частини, бібліотеки Chart.js для візуалізації екологічних даних, Leaflet для інтеграції інтерактивної карти, а також API OpenWeatherMap для отримання актуальної інформації про стан повітря. Обрані засоби забезпечують ефективну побудову вебзастосунку з високим рівнем функціональності, зручним інтерфейсом користувача, що є важливим для систем, які працюють з екологічними даними в режимі реального часу.

3.2 Реалізація модуля побудови графіків

Модуль побудови графіків відповідає за динамічне візуальне відображення змін якості повітря з часом. Його реалізовано на основі JavaScript бібліотеки Chart.js, яка забезпечує побудову адаптивних, інтерактивних та естетично привабливих графіків (див. рисунок 3.1). Метою модуля є надання користувачеві можливості оцінювати тенденції зміни ключових екологічних показників, таких як концентрації твердих частинок PM2.5 і PM10, а також індексу якості повітря (AQI).

При завантаженні сторінки створюється порожній лінійний графік, у якому передбачені набори даних: для PM2.5, PM10, AQI, NO, О₃, SO₂, NH₃. Кожна лінія має своє колірне оформлення для зручності сприйняття. Далі, коли користувач натискає на карту або змінюється місце вимірювання, запускається функція отримання нових екологічних даних через API. У відповідь API повертає поточні значення показників, які зберігаються разом із часовою міткою у відповідні масиви. Модуль автоматично оновлює графік після кожного нового запиту, додаючи нову точку на кожному з ліній.

Також оновлюються інформаційні картки на сторінці, які відображають поточний рівень концентрації кожного з показників. Колір фону кожної картки динамічно змінюється відповідно до рівня якості повітря, визначеного за встановленими пороговими значеннями.

```
<section class="charts-section">
  <h2 class="charts-title">Графіки якості повітря</h2>
  <div class="chart">
    <div class="chart-container"><canvas id="pm25Chart" width="400" height="200"></canvas></div>
    <div class="chart-container"><canvas id="pm10Chart" width="400" height="200"></canvas></div>
    <div class="chart-container"><canvas id="aqiChart" width="400" height="200"></canvas></div>
    <div class="chart-container"><canvas id="noChart" width="400" height="200"></canvas></div>
    <div class="chart-container"><canvas id="o3Chart" width="400" height="200"></canvas></div>
    <div class="chart-container"><canvas id="so2Chart" width="400" height="200"></canvas></div>
    <div class="chart-container"><canvas id="nh3Chart" width="400" height="200"></canvas></div>
  </div>
</section>
```

Рисунок 3.1 – Створення графіків

Усі оновлення графіка відбуваються плавно, без потреби перезавантаження сторінки, що забезпечує комфортний та інтуїтивно зрозумілий користувацький досвід. Модуль побудови графіків є центральним елементом системи візуалізації екологічних даних, який надає користувачеві можливість спостерігати за змінами показників якості повітря в режимі реального часу. Він не лише візуалізує поточні значення, а й дозволяє відстежувати динаміку розвитку ситуації, допомагаючи виявляти як короткострокові коливання, так і довгострокові тенденції.

Цей модуль виступає ефективним аналітичним інструментом, що сприяє оперативному прийняттю рішень щодо заходів з покращення екологічної ситуації. Гнучкі налаштування графіків дають змогу користувачеві обирати різні часові інтервали, адаптуючи відображення даних до своїх потреб. Завдяки цьому, система стає зручною як для фахівців, які потребують детального аналізу, так і для звичайних користувачів, зацікавлених у моніторингу якості повітря у своєму регіоні.

3.3 Реалізація модуля відображення інтерактивної карти

Модуль інтерактивної карти забезпечує зручний спосіб візуалізації та моніторингу якості повітря в різних географічних точках. За допомогою Leaflet.js користувач може взаємодіяти з картою, збільшуючи або зменшуючи масштаб, а також переміщуючи карту для перегляду різних областей. Кожен маркер на карті вказує на місце розташування сенсора, що здійснює моніторинг якості повітря, і може містити додаткову інформацію, таку як поточні показники забруднення повітря (PM2.5, PM10, AQI, NO, O₃, SO₂, NH₃).

Інтерактивність карти дозволяє користувачам легко знаходити місця з погіршеною якістю повітря, а також переглядати дані у реальному часі для конкретних локацій. Також карта підтримує можливість вибору кількох точок для порівняння рівнів забруднення в різних районах або міста, що робить її зручним інструментом для аналізу просторових характеристик екологічних даних. Інтерфейс модуля оптимізовано для роботи як на стаціонарних, так і на мобільних пристроях.

Маркер на карті оновлюються динамічно відповідно до змін у даних, отриманих через API, і можуть відображати як поточні значення забруднювальних речовин, так і коротку аналітичну інформацію у вигляді спливаючих підказок.

Ключовим елементом модуля є обробник події click на маркері карти. Коли користувач натискає на певну точку, спрацьовує функція, яка отримує координати обраної локації та викликає запит до API для завантаження актуальних екологічних

даних відповідно до обраних координат. Отримані дані зберігаються у відповідних масивах і використовуються для оновлення графіка та інформаційних панелей.

Запит до API виконується асинхронно за допомогою `fetch`, що дозволяє динамічно оновлювати інтерфейс без перезавантаження сторінки. Дані, отримані у форматі JSON, обробляються функцією, яка додає нову часову мітку та значення кожного показника у відповідні масиви, а також викликає оновлення візуалізацій. Завдяки цьому карта підтримує режим реального часу: при кожному натисканні користувача вона одразу відображає актуальні значення для обраної точки [24].

Крім того, модуль реалізовано з урахуванням адаптивності: карта масштабована, її елементи зберігають інтерактивність на різних пристроях. Також користувач може обрати іншу точку на карті для перегляду детальнішої інформації про стан повітря в конкретному місці (див. рисунок 3.2).

```
map.on("click", (e) => {  
  currentCoords = { lat: e.latlng.lat, lon: e.latlng.lng };  
  marker.setLatLng([currentCoords.lat, currentCoords.lon]);  
  clearData();  
  fetchAndUpdateData();  
});
```

Рисунок 3.2 – Обробка події click

При натисканні на будь-яку точку карти координати кліку автоматично зчитуються, після чого маркер переміщується до нової позиції, відповідно до вибраного місця. Одночасно виконується виклик функції `fetchAndUpdateData`, яка надсилає асинхронний запит до API OpenWeatherMap для отримання актуальних даних про якість повітря – зокрема, показників PM2.5, PM10, AQI, NO, O₃, SO₂ та NH₃ – за отриманими координатами.

Отримані дані зберігаються у відповідних масивах, а інтерфейс користувача оновлюється в реальному часі: відображаються актуальні значення показників у вигляді числових індикаторів, а також оновлюються лінійні графіки, що

ілюструють зміну рівнів забруднення з часом. Такий підхід забезпечує користувачеві зручний інструмент для швидкого аналізу якості повітря у вибраному регіоні. Дані обробляються та використовуються для оновлення відображення інформації на карті та супутніх графіках що забезпечує користувачеві оперативний доступ до свіжих показників екологічного стану у вибраній локації (див. рисунок 3.3).

```

async function fetchAndUpdateData() {
  const { lat, lon } = currentCoords;
  const url = `https://api.openweathermap.org/data/2.5/air_pollution?lat=${lat}&lon=${lon}&appid=${apiKey}`;

  try {
    const response = await fetch(url);
    const json = await response.json();
    const data = json.list[0];
    const comp = data.components;
    const aqi = data.main.aqi;

    const timestamp = new Date().toLocaleTimeString().slice(0, 5);

    if (timeLabels.length >= maxDataPoints) {
      timeLabels.shift();
      Object.values(dataSets).forEach(arr => arr.shift());
    }

    timeLabels.push(timestamp);
    dataSets.pm25.push(comp.pm2_5);
    dataSets.pm10.push(comp.pm10);
    dataSets.aqi.push(aqi);
    dataSets.no.push(comp.no);
    dataSets.o3.push(comp.o3);
    dataSets.so2.push(comp.so2);
    dataSets.nh3.push(comp.nh3);

    updateCards(comp.pm2_5, comp.pm10, aqi, comp.no, comp.o3, comp.so2, comp.nh3);

    Object.values(charts).forEach(chart => chart.update());
  } catch (e) {
    console.error("Помилка завантаження:", e);
  }
}

```

Рисунок 3.3 – Отримання даних через функцію fetchAndUpdateData

Відповідні значення після отримання відображаються на вебсторінці, і водночас оновлюється стилізація блоків індикаторів відповідно до рівня забруднення. Таким чином, користувач у реальному часі може досліджувати екологічну ситуацію в будь-якому місці, просто натискаючи на мапу. Також передбачена можливість автоматичного оновлення даних з певним інтервалом, що забезпечує постійну актуальність інформації без необхідності ручного перезавантаження сторінки. Це сприяє підвищенню зручності користування та

оперативності у реагуванні на зміни якості повітря. Завдяки цьому вебдодаток може бути корисним як для звичайних користувачів, так і для фахівців, що приймають рішення у сфері охорони довкілля.

Система підтримує гнучкі налаштування відображення інформації, що дозволяє адаптувати інтерфейс під індивідуальні потреби користувача. Наприклад, можна вибирати конкретні показники забруднення для моніторингу або встановлювати власні порогові значення оповіщень. Такий підхід підвищує рівень інформованості користувачів і дозволяє ефективніше управляти заходами щодо покращення якості повітря. Вебдодаток також має потенціал для інтеграції з іншими інформаційними системами, що розширює його функціональність та сприяє більш комплексному підходу до екологічного моніторингу.

3.4 Реалізація модуля отримання даних по якості повітря через API

Модуль отримання даних реалізовано через API сервісу OpenWeatherMap, який надає відкритий доступ до інформації про стан забруднення повітря в заданій географічній точці. Основна функція `fetchAndUpdateData` виконує HTTP-запит до API, передаючи координати широти та довготи. Запит здійснюється через вбудований JavaScript-метод `fetch`, після чого отриманий JSON-об'єкт розбирається, і з нього вилачуються ключові показники якості повітря: рівень зважених частинок PM2.5, PM10, індекс якості повітря (AQI), а також концентрації газових забруднювачів NO, O₃, SO₂ та NH₃.

Після отримання та обробки даних функція оновлює відповідні масиви, які використовуються для подальшого візуального відображення. У випадку успішного запиту інтерфейс автоматично відображає свіжі значення: оновлюються числові показники на панелі користувача, а також відбувається динамічне оновлення лінійних графіків, що ілюструють зміну рівнів забруднення повітря з часом. Якщо запит завершується з помилкою, користувач отримує сповіщення про недоступність або проблеми з отриманням даних.

Функція також передбачає перевірку статусу відповіді сервера. Якщо запит пройшов успішно, отримані дані передаються у функцію оновлення інтерфейсу `updateCards`, яка відповідає за відображення числових значень показників на сторінці (див. рисунок 3.4). Для покращення сприйняття інформації, функція `getLevel` візуалізує рівень забруднення за допомогою кольорових індикаторів, які зображені на рисунку 3.5. Такий підхід дозволяє користувачеві швидко орієнтуватися у стані якості повітря, відразу помічаючи критичні або безпечні рівні забруднення завдяки колірній диференціації.

```
function updateCards(pm25, pm10, aqi, no, o3, so2, nh3) {  
  setCard("pm25", pm25, " µg/m³", getLevelPM(pm25));  
  setCard("pm10", pm10, " µg/m³", getLevelPM(pm10));  
  setCard("aqi", aqi, "", getLevelAQI(aqi));  
  setCard("no", no, " µg/m³", getLevelNO(no));  
  setCard("o3", o3, " µg/m³", getLevelO3(o3));  
  setCard("so2", so2, " µg/m³", getLevelSO2(so2));  
  setCard("nh3", nh3, " µg/m³", getLevelNH3(nh3));  
}
```

Рисунок 3.4 – Функція `updateCards`

Для коректної роботи з API в коді використовується унікальний ключ доступу (API key), який реєструється на сайті OpenWeather. Формування запиту включає параметри `lat`, `lon`, `appid` (ключ), а також одиниці вимірювання, що задаються як `metric`. Отримані у відповідь дані передаються у форматі JSON, який легко обробляється на клієнтській стороні для подальшого виводу на екран. Такий підхід забезпечує швидке оновлення даних та їх актуальність без потреби перезавантаження сторінки.

Таким чином, модуль є центральним джерелом екологічних даних для всієї системи, забезпечуючи регулярне оновлення актуальної інформації про стан повітря в інтерактивному режимі. Завдяки цьому користувачі отримують доступ до точних і оперативних даних, що дозволяє приймати зважені рішення, пов'язані з

перебуванням на відкритому повітрі. Також система легко масштабується для підключення інших джерел даних або розширення географії моніторингу.

```
function getLevelAQI(value) {  
    if (value === 1) return "good";  
    if (value === 2) return "moderate";  
    if (value === 3) return "unhealthy";  
    return "hazardous";  
}
```

Рисунок 3.5 – Функція getLevel

Зелений колір використовується для чистого повітря ($AQI = 1$), жовтий – для помірного рівня забруднення ($AQI = 2$), а червоний або фіолетовий – для небезпечного рівня ($AQI \geq 4$). Така кольорова градація дозволяє користувачу інтуїтивно розуміти екологічну ситуацію без необхідності аналізувати числові показники. Завдяки миттєвому візуальному зворотному зв'язку користувач швидко орієнтується в поточному стані повітря, що особливо важливо у випадках різкого погіршення якості атмосферного середовища.

3.5 Реалізація модуля виведення інформації у CSV формат

Для збереження отриманих екологічних даних у зручному табличному форматі в системі реалізовано модуль експорту у CSV. Це дозволяє користувачеві завантажити результати моніторингу якості повітря для подальшого аналізу в сторонніх програмах, таких як Microsoft Excel або Google Sheets.

Ключовою функцією модуля є `exportToCSV()`, яка формує текстовий CSV-документ на основі масивів даних, що накопичуються під час роботи системи. У першому рядку створюється заголовок з назвами колонок: час вимірювання, концентрації $PM_{2.5}$, PM_{10} , значення індексу AQI, а також рівні забруднення для NO , O_3 , SO_2 та NH_3 . Далі функція проходить по кожному елементу масиву часу

(timeLabels) і формує відповідні рядки з даними, округлюючи значення до двох знаків після коми, де це доцільно (див. рисунок 3.6).

```
function exportToCSV() {
  let csv = "Time,PM2.5,PM10,AQI,NO,ОЗ,SO2,NH3\n";

  for (let i = 0; i < timeLabels.length; i++) {
    const row = [
      dataSets.pm25[i]?.toFixed(2) ?? "",
      dataSets.pm10[i]?.toFixed(2) ?? "",
      dataSets.aqi[i] ?? "",
      dataSets.no[i]?.toFixed(2) ?? "",
      dataSets.o3[i]?.toFixed(2) ?? "",
      dataSets.so2[i]?.toFixed(2) ?? "",
      dataSets.nh3[i]?.toFixed(2) ?? ""
    ];
    csv += row.join(",") + "\n";
  }

  const blob = new Blob([csv], { type: "text/csv;charset=utf-8;" });
  const url = URL.createObjectURL(blob);
  const a = document.createElement("a");
  a.href = url;
  a.download = `air_quality_data_${new Date().toISOString().slice(0,10)}.csv`;
  a.style.display = "none";
  document.body.appendChild(a);
  a.click();
  document.body.removeChild(a);
  URL.revokeObjectURL(url);
}
```

Рисунок 3.6 – Функція exportToCSV

Після формування CSV-рядків створюється об'єкт типу Blob, який імітує файл у браузері. За допомогою URL.createObjectURL генерується тимчасове посилання, яке прикріплюється до створеного динамічно елемента. Після активації події click відбувається завантаження файлу на комп'ютер користувача з назвою, що містить поточну дату, що полегшує ідентифікацію файлу серед інших. Такий підхід забезпечує простий та ефективний механізм експорту даних у форматі CSV без необхідності використання сторонніх бібліотек, що підвищує автономність та швидкодію модуля. Це дозволяє користувачам зберігати локальні копії даних для подальшого аналізу або звітності. Завдяки легкості інтеграції цей функціонал можна швидко впровадити у різні частини вебзастосунку, що підвищує загальну зручність користування системою, а також це сприяє підвищенню ефективності роботи з даними та полегшує їх подальше використання у зовнішніх аналітичних інструментах.

3.6 Висновки

У третьому розділі було детально обґрунтовано вибір технологій та інструментів, які використовувалися для розробки програмних модулів вебзастосунку для моніторингу якості повітря. На основі аналізу вимог до системи та функціональних потреб було прийнято рішення застосувати JavaScript, HTML та CSS для створення інтерфейсу вебзастосунку, що забезпечує адаптивність, інтерактивність та зручність користування. Для отримання актуальних даних про забруднення повітря було обрано API OpenWeatherMap, яке надає достовірну та своєчасну інформацію про основні екологічні показники.

Для візуалізації даних та побудови графіків використано популярний та потужний фреймворк Chart.js, що дозволяє створювати інтерактивні лінійні та інші типи графіків з можливістю динамічного оновлення даних у реальному часі. Це сприяє наочному відображенню тенденцій зміни показників якості повітря та полегшує їх аналіз користувачами. Для інтерактивного відображення географічної інформації була обрана бібліотека Leaflet, яка забезпечує зручне вбудовування карт та додавання маркерів для точного визначення місцезнаходження користувача або точок моніторингу. Для індикації рівня забруднення повітря реалізовано динамічну логіку зміни кольорів маркерів на основі значень PM2.5, PM10, AQI, NO, O₃, SO₂ та NH₃, що дозволяє користувачеві миттєво оцінити стан якості повітря в обраній локації на карті. У цьому розділі також детально описано розробку ключових програмних модулів, таких як побудова графіків, інтерактивна карта та отримання даних через API, які разом забезпечують ефективне відображення і моніторинг стану повітря в режимі реального часу. Завдяки впровадженним технологіям користувачі отримують доступ до точних і своєчасних екологічних даних, що сприяє кращому розумінню екологічної ситуації та прийняттю обґрунтованих рішень для покращення якості повітря.

4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Тестування системи

Тестування програмного забезпечення – це процес перевірки вебзастосунку для моніторингу якості повітря з метою виявлення помилок, дефектів або неполадок та перевірки відповідності вимогам користувача та замовника. Метою тестування є забезпечення надійності та високої якості програмного продукту перед його запуском у експлуатацію.

Для тестування необхідно встановити тестову версію вебзастосунку на комп'ютер з операційною системою Windows, MacOS або Linux. Перед початком тестування слід забезпечити наявність документації, яка містить специфікації вимог, план тестування та сценарії, що допоможуть у проведенні тестування.

Тестування розбивається на два основні етапи: функціональне тестування та тестування на відмову.

Першим етапом є функціональне тестування. Основна мета цього етапу – перевірка правильної роботи ключових функцій вебзастосунку. Під час функціонального тестування перевіряються основні сценарії використання, такі як коректне відображення графіків за даними про якість повітря, правильне відображення інформації на інтерактивній карті та коректна обробка запитів до API для отримання актуальних даних. Також важливим є тестування графічного інтерфейсу, чи правильно працюють функції динамічного оновлення, відображення кольорів для різних рівнів забруднення повітря, та перевірка інтерфейсу користувача на зручність та інтуїтивну зрозумілість. Особливу увагу слід приділяти перевірці всіх можливих сценаріїв взаємодії користувача з інтерфейсом, аби переконатися, що всі елементи працюють так, як очікується. Ще необхідно провести тестування на наявність помилок у бізнес-логіці, таких як неправильний розрахунок індексів якості повітря (AQI) чи неточності в розрахунках концентрацій забруднювачів. Це дозволяє переконатися, що система обробляє дані правильно й надає користувачам актуальну та точну інформацію.

Важливо також перевірити механізми авторизації користувачів і доступ до функцій залежно від прав користувача. Для цього тестуються сценарії, коли користувач входить у систему з різними правами доступу (наприклад, адміністративні права або звичайний користувач) та здійснює відповідні дії в інтерфейсі.

Не менш важливим є тестування зручності взаємодії з мобільними пристроями, перевірка адаптивності вебзастосунку і відповідності відображення елементів інтерфейсу на екранах різних розмірів. Тестування на різних пристроях дозволяє переконатися, що вебзастосунок зберігає функціональність і зручність користування навіть на маленьких екранах.

Другим етапом є тестування на відмову. Тут перевіряється здатність вебзастосунку коректно реагувати на помилки або неочікувані ситуації. Наприклад, тестується, як система поводить себе при відсутності з'єднання з API. Тестування на відмову передбачає не лише перевірку реакції на помилки API, але й тестування на ситуації, коли сервер чи інші зовнішні сервіси не відповідають або дають помилку. Такі тестування дозволяють виявити, чи здатна система відновлювати роботу після помилок, чи правильно вона обробляє некоректні запити та чи надає користувачеві зрозумілі повідомлення про помилки. Важливо перевірити, чи система надає альтернативні варіанти користувачеві, наприклад, виведення останніх доступних даних або повідомлення про проблеми з підключенням до сервісу. Тестування на відмову також включає перевірку стабільності роботи додатку при різних непередбачуваних ситуаціях, щоб забезпечити безперервну роботу при будь-яких обставинах. Це може включати випадки відключення інтернет-з'єднання, великі затримки в отриманні даних з API, обробку порушень на серверній стороні, наприклад, перевантаження або відсутність ресурсу.

Тестування на відмову дозволяє впевнитися, що вебзастосунок має вбудовані механізми для коректного і надійного функціонування навіть в умовах непередбачуваних ситуацій, що важливо для забезпечення довготривалої стабільної роботи користувацького сервісу.

Тестування за цими етапами дозволяє виявити основні дефекти і проблеми, забезпечити коректну роботу всіх функцій та підготувати продукт до експлуатації.

Під час першого етапу тестування програмного забезпечення перевіряється робота функціоналу вебзастосунку. Для початку було успішно вибрано місто на інтерактивній мапі (див. рисунок 4.1).

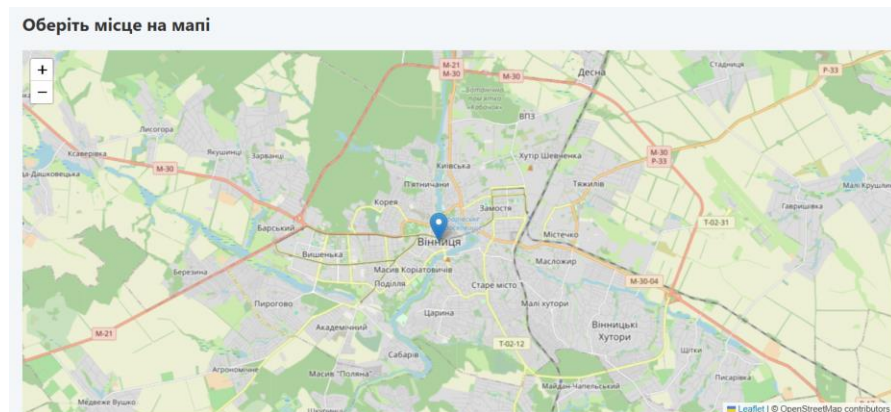


Рисунок 4.1 – Призначена точка на інтерактивній мапі

Наступним кроком тестування, було вирішено протестувати чи правильно працюють функції динамічного оновлення, відображення кольорів для різних рівнів забруднення повітря, та перевірка інтерфейсу користувача на зручність та інтуїтивну зрозумілість (див. рисунок 4.2).

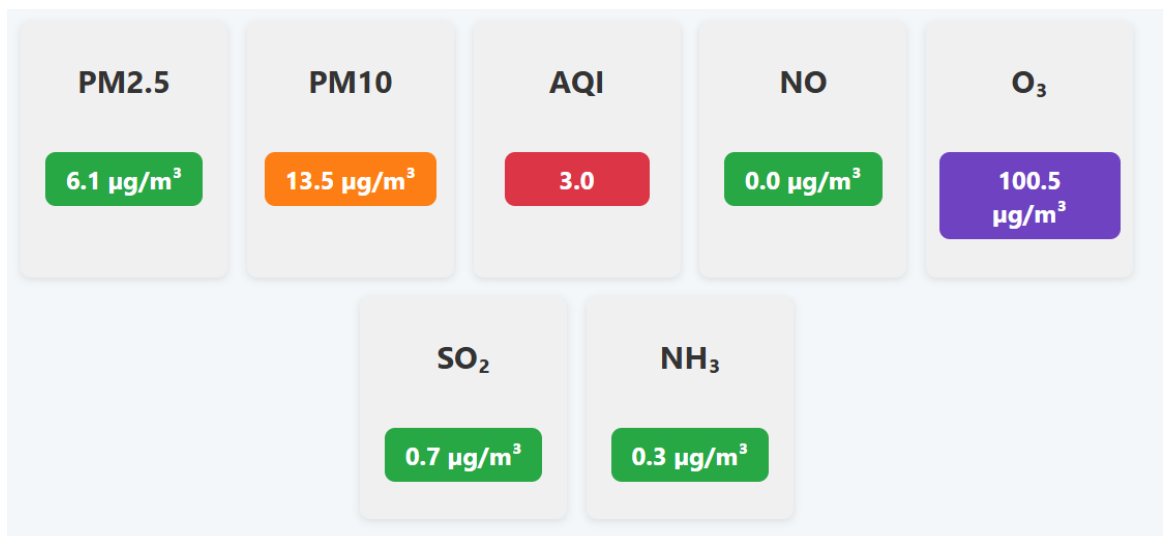


Рисунок 4.2 – Зміна кольору даних по забрудненню

Після цього етапу тестування було зроблено висновок про зручність використання інтерфейсу, коректність функціонування динамічного оновлення даних та відображення кольорових індикаторів.

Далі йде перевірка на коректність даних у графіках після отримання даних з API. Цей етап дозволяє переконатися в тому, що всі візуальні елементи працюють коректно, і дані, які надаються користувачеві через графіки, карту та API, є точними та актуальними (див. рисунок 4.3).

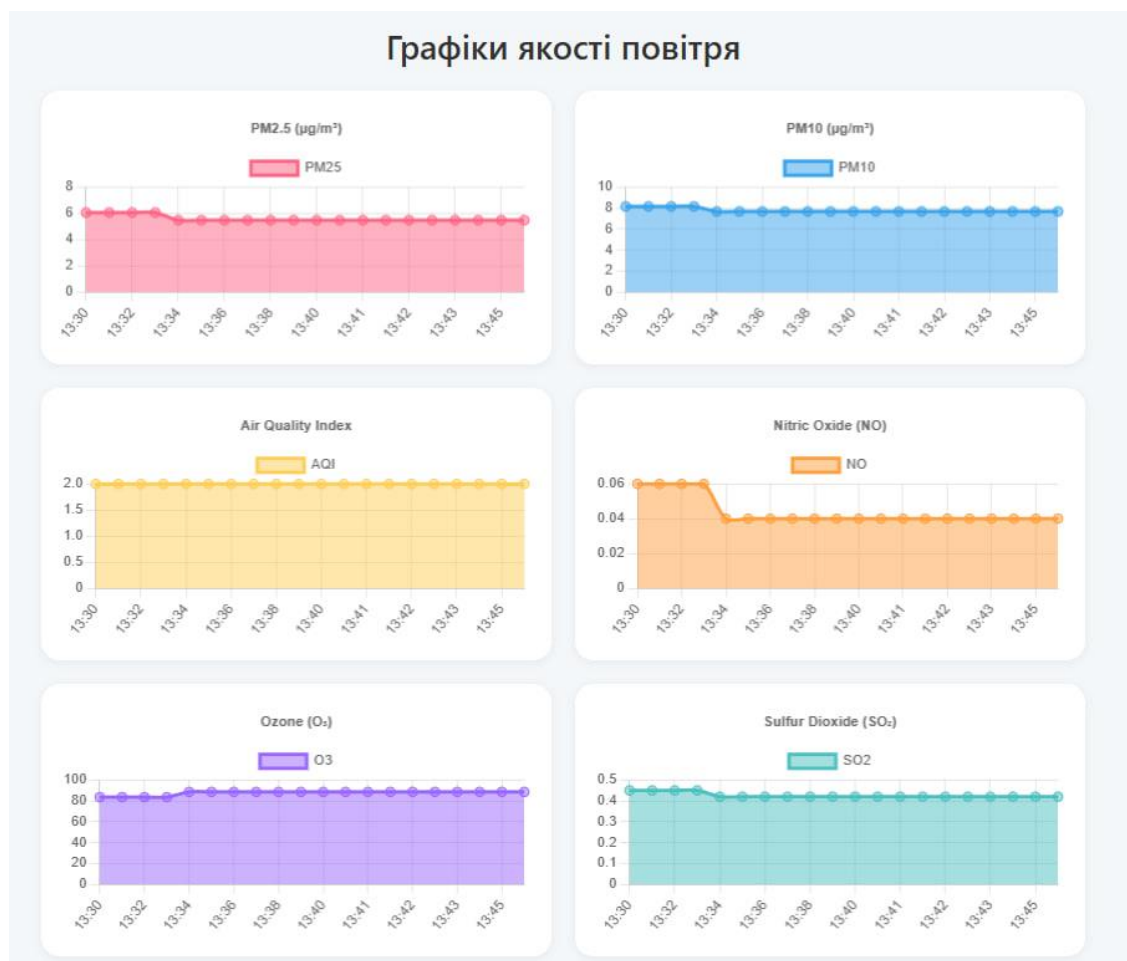


Рисунок 4.3 – Додавання графіків

Діле перевіремо можливість занесення даних у Excel-файл було проведено комплексну перевірку роботи відповідного модуля. Спочатку було перевірено, чи коректно формуються і збираються дані для експорту, включаючи всі необхідні поля з правильними типами значень. Далі здійснювалася генерація файлу у форматі

CSV з використанням відповідних бібліотек, при цьому контролювалася правильність структури таблиці, наявність заголовків та правильне розташування даних у комірках. Також було протестовано збереження файлу на диск користувача і його подальше відкриття у Microsoft Excel (див. рисунок 4.4).

	A	B	C	D	E	F	G
1	Time	PM2.5	PM10	AQI	NO	O3	SO2
2	6.14	13.54	3	0.01	100.46	0.65	0.3
3	6.14	13.54	3	0.01	100.46	0.65	0.3
4	6.14	13.54	3	0.01	100.46	0.65	0.3
5	6.14	13.54	3	0.01	100.46	0.65	0.3
6	6.14	13.54	3	0.01	100.46	0.65	0.3
7	6.14	13.54	3	0.01	100.46	0.65	0.3
8	6.14	13.54	3	0.01	100.46	0.65	0.3
9	6.14	13.54	3	0.01	100.46	0.65	0.3
10	6.24	13.68	2	0	97.9	0.68	0.29
11	6.24	13.68	2	0	97.9	0.68	0.29
12	6.24	13.68	2	0	97.9	0.68	0.29
13	6.24	13.68	2	0	97.9	0.68	0.29
14	6.24	13.68	2	0	97.9	0.68	0.29
15	6.24	13.68	2	0	97.9	0.68	0.29
16	6.24	13.68	2	0	97.9	0.68	0.29
17	6.24	13.68	2	0	97.9	0.68	0.29
18	6.24	13.68	2	0	97.9	0.68	0.29
19	6.24	13.68	2	0	97.9	0.68	0.29
20	6.24	13.68	2	0	97.9	0.68	0.29
21	6.24	13.68	2	0	97.9	0.68	0.29

Рисунок 4.4 – Перенесення файлів у CSV формат

В процесі тестування оцінювалася і коректність обробки можливих помилок, наприклад, відсутності даних для експорту або проблем із збереженням файлу, із виведенням відповідних повідомлень для користувача. Загалом результати тестування підтвердили стабільну роботу модуля та відповідність створених файлів вимогам проекту.

На другому етапі тестування на відмову, перевіряється здатність вебзастосунку коректно реагувати на помилки або неочікувані ситуації. Тестується, як система поводить себе при відсутності з'єднання з API (див. рисунок 4.5). У випадку, коли запит не може бути виконаний через мережеву помилку або недоступність сервера. При цьому попередні значення залишаються на місці, що

запобігає втраті важливої інформації та забезпечує стабільність роботи користувача навіть в умовах тимчасових збоїв.

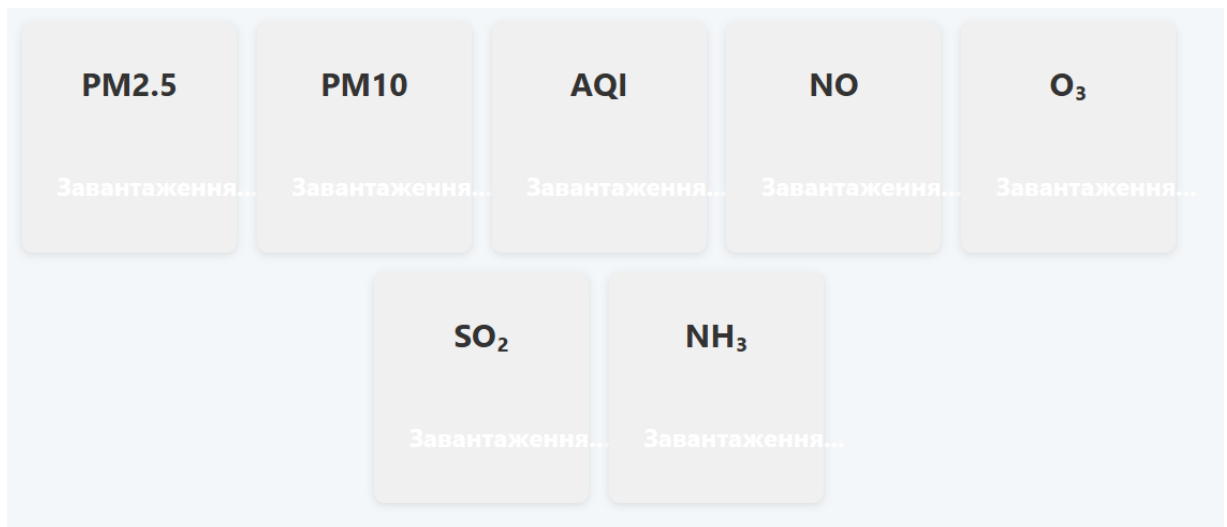


Рисунок 4.5 – Результат відсутності API ключа

У разі відсутності API ключа дані PM2.5, PM10, AQI, а також інші показники, зокрема NO, O₃, SO₂ та NH₃, не можуть бути отримані. Вебзастосунок відображає чітке повідомлення про помилку, яке інформує користувача про необхідність наявності дійсного API ключа для доступу до екологічних даних. Така реалізація дозволяє перевірити, як система обробляє помилки, пов'язані з відсутністю або некоректністю API ключа, і як вона забезпечує інформативний зворотний зв'язок, попереджаючи користувача про відсутність критичних елементів, необхідних для повноцінної роботи додатку.

4.2 Розробка інструкції користувача

Інструкція користувача передбачає визначення технічних вимог для коректної роботи вебзастосунку та опис основних дій, які виконує користувач для взаємодії з інтерфейсом системи. Деталі щодо мінімальної та рекомендованої конфігурації наведено в таблиці 4.1.

Таблиця 4.1 – Вимоги до апаратного забезпечення

Вимоги до конфігурування	Мінімально	Рекомендовано
Процесор	Intel Pentium / AMD A4	Intel Core i5 / AMD Ryzen 5
Оперативна пам'ять	2 ГБ RAM	8 ГБ RAM
Вільний простір на диску	500 МБ	1 ГБ
Відеокарта	Вбудована	Вбудована, з підтримкою WebGL
Операційна система	Windows 7+, macOS 10.13+, Linux	Windows 10+, macOS 11+, Linux
Браузер	Chrome 90+, Firefox 90+	Chrome 100+, Firefox 95+, Edge
Монітор	1366x768 пікселів	1920x1080 пікселів або більше

Після завантаження вебзастосунку користувач автоматично потрапляє на головну сторінку, де система миттєво ініціалізує основні дані про стан якості повітря. Ці дані завантажуються через API та відображаються у вигляді графіків, карт і текстових індикаторів, що дозволяє отримати актуальну інформацію про рівень забруднення в обраному регіоні. Всі дані оновлюються автоматично з певним інтервалом, що дає змогу користувачеві стежити за змінами в реальному часі без необхідності оновлення сторінки вручну.

Інтерфейс вебзастосунку організовано так, щоб користувач міг зручно орієнтуватися в системі. Головна робоча область поділяється на кілька основних секцій. По-перше, це панель з графіками, де візуалізуються показники якості повітря, такі як концентрації PM_{2.5}, PM₁₀, NO, O₃, SO₂, NH₃ та загальний індекс AQI. Це дозволяє користувачу слідкувати за динамікою змін цих параметрів у реальному часі. Графіки побудовані таким чином, щоб користувач міг швидко

оцінити, виявити пікові значення, а також отримати зрозумілу картину зміни стану повітря протягом останніх годин, днів або навіть тижнів. Користувач має можливість обрати часовий діапазон для відображення статистики, а також завантажити дані у форматі CSV для подальшого аналізу.

По-друге, є інтерактивна мапа, яка займає значну частину екрану та дозволяє візуально відслідковувати розташування точок моніторингу. Кожна точка на мапі позначена кольоровим індикатором, який відображає загальний рівень забруднення в цій локації: від зеленого (добре) до червоного (небезпечно). Користувач може взаємодіяти з картою, натискати на будь-яку локацію для отримання більш детальної інформації про стан якості повітря в обраній точці. Інформація включає точні дані щодо концентрацій забруднюючих речовин, температуру, вологість, швидкість вітру, а також попередження у разі перевищення допустимих норм. Крім того, мапа підтримує динамічне оновлення інформації у реальному часі, що особливо важливо у випадках швидких змін екологічних умов, наприклад, під час пожеж, викидів або інших аварійних ситуацій. Карта дозволяє перемикатися між різними режимами перегляду: теплові карти, фільтрація за показниками, історичні зміни тощо.

Усі основні функції, такі як вибір регіону для моніторингу, перегляд історичних даних або налаштування відображення графіків, доступні в кілька кліків. Інтерфейс підтримує адаптивну верстку, що дозволяє користуватись системою як з комп'ютера, так і з мобільних пристроїв. Користувач може швидко перейти між різними розділами вебзастосунку, що дозволяє йому легко знаходити необхідну інформацію, незалежно від рівня технічної підготовки. Крім того, реалізовано підтримку багатомовності, що розширює потенційне коло користувачів.

Загалом, вебзастосунок створено з орієнтацією на користувача, забезпечуючи простоту у використанні, швидкий доступ до ключової інформації та можливості глибокого аналізу даних, що робить його ефективним інструментом моніторингу стану довкілля. Для зручності користувачів додано додаткові функції,

такі як автоматичне оновлення даних. Це дає можливість безперервно стежити за змінами стану навколишнього середовища без необхідності вручну оновлювати сторінку.

Інтуїтивно зрозумілий інтерфейс сприяє швидкому орієнтуванню навіть для користувачів без спеціальної підготовки, що розширює коло зацікавлених – від простих громадян до фахівців у сфері екології та міського планування. Додатково, наявність історичних графіків і можливість порівняння даних за різні періоди надає змогу аналізувати тенденції зміни якості повітря та оцінювати ефективність впроваджених заходів.

Особлива увага в інтерфейсі приділена кольоровій індикації рівня забруднення. Різні кольори на графіках і на карті миттєво інформують користувача про поточний рівень якості повітря. Зелений колір вказує на безпечний рівень забруднення, жовтий – на помірний, а червоний або фіолетовий – на небезпечний. Така кольорова градація дозволяє користувачу швидко оцінити ситуацію без необхідності детального аналізу числових показників. Приділено увагу і оптимізації швидкості завантаження та оновлення даних, що дозволяє підтримувати актуальність інформації без надмірного навантаження на клієнтські пристрої та сервер.

Розроблений продукт відповідає сучасним вимогам до моніторингу якості повітря і має значний потенціал для масштабування як на регіональному, так і на національному рівнях. Така гнучкість дозволяє ефективно адаптувати систему до різних масштабів і потреб, забезпечуючи широку доступність екологічних даних та підтримку прийняття обґрунтованих рішень у сфері охорони довкілля.

4.3 Висновки

У четвертому розділі було проведено тестування програмних модулів вебзастосунку моніторингу навколишнього середовища по головним його функціям. Було перевірено реакцію програмних засобів на помилкові ситуації, відсутність API ключа. Також було розроблено інструкцію користувача.

ВИСНОВКИ

У рамках бакалаврської роботи були розроблені модулі програмного забезпечення для моніторингу навколишнього середовища завдяки вебзастосунку за допомогою API. Для реалізації використовувалися HTML, CSS та JavaScript і бібліотеки Chart.js для побудови графіків та Leaflet для інтерактивної карти.

Під час виконання бакалаврської роботи було проаналізовано стан вебсистем для моніторингу навколишнього середовища. Були розглянуті основні аналоги програмного продукту, виявлені їхні недоліки та порівняно з власним розробленим продуктом. На основі цього порівняння були сформульовані основні завдання бакалаврської роботи.

У бакалаврській роботі було виконано наступне:

- 1) визначено основні функціональні та нефункціональні вимоги до вебсистеми моніторингу навколишнього середовища;
- 2) розроблено модулі збору та обробки екологічних показників;
- 3) розроблено адаптивний та інтуїтивно зрозумілий вебінтерфейс;
- 4) забезпечено підтримку автоматичного збору даних через API;
- 5) проведено тестування функціональності програмних модулів.

Було розроблено схеми загального алгоритму роботи програмного застосунку. Також було розроблено модель системи з використанням UML діаграм.

Був проведений варіантний аналіз і обґрунтований вибір засобів реалізації програмного продукту.

Тестування програми довело повну працездатність даного програмного продукту та відповідність поставленому технічному завданню. Також було розроблено інструкцію користувача.

ЛІТЕРАТУРА

1. Індикатори стану екологічної безпеки держави [Електронний ресурс] – Режим доступу до ресурсу: <https://www.niss.gov.ua/doslidzhennya/nacionalna-bezpeka/indikatori-stanu-ekologichnoi-bezpeki-derzhavi-analitichna-zapiska>
2. Екологічний моніторинг [Електронний ресурс] – Режим доступу до ресурсу: <http://oldconf.neasmo.org.ua/node/1169>
3. Моніторинг довкілля [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Моніторинг_довкілля
4. Клєшко В. В., Розробка вебзастосунку для моніторингу навколишнього середовища. Молодь в науці: дослідження, проблеми, перспективи (МН-2025), Вінниця, 2025. [Електронний ресурс] – <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/25559>
5. Стан доступу до екологічної інформації [Електронний ресурс] – Режим доступу до ресурсу: <https://www.saveecobot.com/static/environmental-data-status>
6. Створення ефективної системи моніторингу довкілля [Електронний ресурс] – Режим доступу до ресурсу: <https://niss.gov.ua/doslidzhennya/nacionalna-bezpeka/stvorennya-efektivnoi-sistemi-monitoringu-dovkillya-v-ukraini>
7. OpenAQ [Електронний ресурс] – Режим доступу до ресурсу: <https://openaq.org>
8. Plume Labs [Електронний ресурс] – Режим доступу до ресурсу: <https://plumelabs.com/en/>
9. BreezoMeter [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.breezometer.com/api-documentation/air-quality-api/v2/#current-conditions>
10. PurpleAir [Електронний ресурс] – Режим доступу до ресурсу: https://www2.purpleair.com/?srsltid=AfmBOooPPVojrtfrSRgN7MkU6wmgnKpT_Mi_Xg2Z8X9aADQSgmWyVuIP
11. Моніторинг довкілля [Електронний ресурс] – Режим доступу до ресурсу: http://www.ukrpryroda.org/2013/03/blog-post_6890.html

12. Inhalable Particulate Matter and Health [Електронний ресурс] – Режим доступу до ресурсу: <https://ww2.arb.ca.gov/resources/inhalable-particulate-matter-and-health>

13. РМ та AQI [Електронний ресурс] – Режим доступу до ресурсу: <https://www.savednipro.org/pytannia-pro-zabrudnene-povitria/>

14. Геоінформаційна система [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Геоінформаційна_система

15. Геоінформаційна система (ГІС) [Електронний ресурс] – Режим доступу до ресурсу: <http://asd-geo.com.ua/gis>

16. Картографія [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Картографія>

17. What is three-tier architecture [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/think/topics/three-tier-architecture>

18. HTML5 [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTML5>

19. Що таке UML-діаграми [Електронний ресурс] – Режим доступу до ресурсу: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>

20. Chart.js [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/ai/ai_chartjs.asp

21. Leaflet.js [Електронний ресурс] – Режим доступу до ресурсу: <https://leafletjs.com>

22. Express.js [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com>

23. OpenWeatherMap [Електронний ресурс] – Режим доступу до ресурсу: <https://openweathermap.org/city/2643743>

24. JSON [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/JSON>

ДОДАТКИ

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

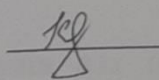
ЗАТВЕРДЖУЮ
д.т.н., проф. О. Н. Романюк
"25" Березня 2025 р.

Технічне завдання
на бакалаврську кваліфікаційну роботу «Розробка вебзастосунку для
моніторингу навколишнього середовища» за спеціальністю
121 Інженерія програмного забезпечення

Керівник бакалаврської кваліфікаційної роботи:

 к.т.н., доц. каф. ПЗ Г.Б. Ракитянська
"24" Березня 2025 р.

Виконав:

 студент гр. 2ПІ-216 В. В. Кleshко
"24" Березня 2025 р.

Вінниця – 2025 року

Найменування та галузь застосування

Бакалаврська кваліфікаційна робота: «Розробка вебзастосунку для моніторингу навколишнього середовища».

Галузь застосування – екологічний моніторинг навколишнього середовища в рамках діяльності громадських організацій та освітніх закладів.

2. Підстава для розробки.

Підставою для виконання бакалаврської кваліфікаційної роботи (БКР) є індивідуальне завдання на БКР та наказ №97 від «20» березня 2025 р. ректора ВНТУ про закріплення тем БКР.

3. Мета та призначення розробки

Метою роботи є розробка інтерактивного вебзастосунку для візуалізації даних про стан навколишнього середовища з можливістю їх оновлення в реальному часі.

Основними задачами є:

- проведення аналізу сучасних вебсистем екологічного моніторингу та виконання порівняльного аналізу існуючих аналогів;
- обґрунтування вибору архітектури програмного забезпечення та методів реалізації функціональності вебзастосунку;
- розробка структури, логічної моделі та алгоритмів функціонування веборієнтованої системи;
- реалізація програмних модулів для побудови графіків, відображення інтерактивної карти, отримання даних і виведення інформації
- проведення тестування працездатності функціоналу вебзастосунку та створення інструкції користувача для забезпечення ефективної експлуатації системи.

4. Вихідні дані для проведення БДР.

Перелік основних літературних джерел, на основі яких буде виконуватись БКР.

1 Клешко В. В., Розробка вебзастосунку для моніторингу навколишнього середовища. Молодь в науці: дослідження, проблеми, перспективи (МН-2025), Вінниця, 2025. [Електронний ресурс] – <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/25559>

2 Екологічний моніторинг [Електронний ресурс] – Режим доступу до ресурсу: <http://oldconf.neasmo.org.ua/node/1169>

3 Створення ефективної системи моніторингу довкілля [Електронний ресурс] – Режим доступу до ресурсу: <https://niss.gov.ua/doslidzhennya/nacionalna-bezpeka/stvorennya-efektivnoi-sistemi-monitoringu-dovkillya-v-ukraini>

4 HTML5 [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTML5>

5 Chart.js [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/ai/ai_chartjs.asp

6 Leaflet.js [Електронний ресурс] – Режим доступу до ресурсу: <https://leafletjs.com>

5. Технічні вимоги

Для розробки застосунку використовувалося середовище Visual Studio Code. Основними мовами програмування є HTML, CSS та JavaScript.

6. Конструктивні вимоги

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до БКР;
- технічне завдання;
- лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз систем для моніторингу навколишнього середовища та постановка задач	25.03.25 – 30.03.25
2	Розробка структури та аналогів програмного продукту	31.03.25 – 22.04.25
3	Розробка програмних модулів	31.03.25 – 27.04.25
4	Тестування програми	28.04.25 – 21.05.25
5	Оформлення матеріалів до захисту БКР	26.05.25– 30.05.25

10. Порядок контролю та прийняття

Порядок контролю і приймання роботи регламентується відповідними документами ВНТУ і державними стандартами.

ДОДАТОК Б – ПЕРЕВІРКА НА ПЛАГІАТ
ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Розробка вебзастосунку для моніторингу навколишнього середовища

Тип роботи: Бакалаврська кваліфікаційна робота
 (бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ Кафедра ПЗ, ФІТКІ, 2ПІ-216
 (кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 3.2 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

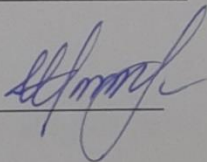
_____ (прізвище, ініціали, посада)

_____ (підпис)

_____ (прізвище, ініціали, посада)

_____ (підпис)

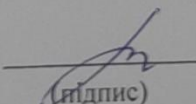
Особа, відповідальна за перевірку



Черноволик Г. О.

З висновком експертної комісії ознайомлений(-на)

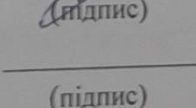
Керівник

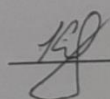

 (підпис)

к.т.н. доц. каф. ПЗ Ракитянська Г. Б.

(прізвище, ініціали, посада)

Здобувач


 (підпис)



Клешко В. В.

(прізвище, ініціали)

ДОДАТОК В – ЛІСТИНГ ПРОГРАМИ

Index.html

```
<!DOCTYPE html>

<html lang="uk">

<head>

  <meta charset="UTF-8" />

  <title>Моніторинг Навколишнього Середовища</title>

  <link rel="stylesheet" href="styles.css" />

  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" />

</head>

<body>

  <header>

    <h1>Моніторинг якості повітря</h1>

  </header>

  <main>

    <section>

      <h2>Оберіть місце на мапі</h2>

      <div id="map" style="height: 500px; width: 80vw;"></div>

    </section>

    <button id="exportBtn">Експорт у Excel </button>
```

```
<section class="content-row">
```

```
<div class="data">
```

```
<div class="card"><h2>PM2.5</h2><p id="pm25"
class="indicator">Завантаження...</p></div>
```

```
<div class="card"><h2>PM10</h2><p id="pm10"
class="indicator">Завантаження...</p></div>
```

```
<div class="card"><h2>AQI</h2><p id="aqi" class="indicator">Завантаження...</p></div>
```

```
<div class="card"><h2>NO</h2><p id="no" class="indicator">Завантаження...</p></div>
```

```
<div class="card"><h2>O3</h2><p id="o3" class="indicator">Завантаження...</p></div>
```

```
<div class="card"><h2>SO2</h2><p id="so2" class="indicator">Завантаження...</p></div>
```

```
<div class="card"><h2>NH3</h2><p id="nh3" class="indicator">Завантаження...</p></div>
```

```
</div>
```

```
</section>
```

```
<section class="charts-section">
```

```
<h2 class="charts-title">Графіки якості повітря</h2>
```

```
<div class="chart">
```

```
<div class="chart-container"><canvas id="pm25Chart" width="400"
height="200"></canvas></div>
```

```
<div class="chart-container"><canvas id="pm10Chart" width="400"
height="200"></canvas></div>
```

```
<div class="chart-container"><canvas id="aqiChart" width="400"
height="200"></canvas></div>
```

```
<div class="chart-container"><canvas id="noChart" width="400"
height="200"></canvas></div>
```

```
<div class="chart-container"><canvas id="o3Chart" width="400"
height="200"></canvas></div>
```

```
<div class="chart-container"><canvas id="so2Chart" width="400"
height="200"></canvas></div>
```

```
<div class="chart-container"><canvas id="nh3Chart" width="400"
height="200"></canvas></div>
```

```
</div>
```

```
</section>
```

```
</main>
```

```
<footer>
```

```
<p>&copy; 2025 Моніторинг середовища</p>
```

```
</footer>
```

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

```
<script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>
```

```
<script src="script.js"></script>
```

```
</body>
```

```
</html>
```

Styles.css

```
body {
  font-family: 'Segoe UI', sans-serif;
  margin: 0;
  padding: 0;
  background: #f4f7f9;
  color: #333;
}
```

```
header {
  background: #2c3e50;
  color: white;
  padding: 1rem;
  text-align: center;
}

main {
  padding: 2rem;
  display: flex;
  flex-direction: column;
  align-items: center;
}

.data {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  gap: 15px;
  margin-top: 20px;
  width: 100%;
  max-width: 900px;
  margin-left: auto;
  margin-right: auto;
}

.card {
  flex: 1 1 120px;
  max-width: 140px;
  text-align: center;
  padding: 10px;
  border-radius: 8px;
  background-color: #f0f0f0;
  box-shadow: 0 2px 6px rgba(0,0,0,0.1);
}

.indicator {
  font-size: 1.2rem;
  font-weight: bold;
  padding: 0.5rem 1rem;
  text-align: center;
  border-radius: 0.5rem;
```

```
color: white;
display: inline-block;
min-width: 80px;
}

.indicator.good {
  background-color: #28a745;
  color: white;
}

.indicator.moderate {
  background-color: #fd7e14;
  color: white;
}

.indicator.unhealthy {
  background-color: #dc3545;
  color: white;
}

.indicator.hazardous {
  background-color: #6f42c1;
  color: white;
}

.charts-section {
  width: 90%;
  max-width: 900px;
  margin: 30px auto 0 auto;
  text-align: center;
}

.charts-title {
  font-size: 1.8rem;
  margin-bottom: 20px;
  font-weight: 600;
  color: #333;
}

.chart {
  display: flex;
  flex-wrap: wrap;
```

```

    gap: 20px;
    justify-content: center;
  }

.chart-container {
  flex: 1 1 300px;
  max-width: 45%;
  background: white;
  padding: 1rem;
  border-radius: 1rem;
  box-shadow: 0 2px 8px #0000000d;
}

footer {
  text-align: center;
  padding: 1rem;
  background: #ecf0f1;
  color: #7f8c8d;
  margin-top: auto;
}

```

Script.js

```

'use strict';

const apiKey = "4d29c626f46e638c0208cbd33fdbd16d";
let currentCoords = { lat: 49.2328, lon: 28.48097 };

const maxDataPoints = 20;
const timeLabels = [];
const dataSets = {
  pm25: [],
  pm10: [],
  aqi: [],
  no: [],
  o3: [],
  so2: [],
  nh3: []
};

const charts = {};

document.addEventListener("DOMContentLoaded", () => {

```

```

    initCharts();
    initMap();
    fetchAndUpdateData();
    setInterval(fetchAndUpdateData, 20000);
  });

  function initCharts() {
    const config = [
      { id: "pm25Chart", key: "pm25", label: "PM2.5", color: "rgba(255,99,132,0.5)", title: "PM2.5 (µg/m³)" },
      { id: "pm10Chart", key: "pm10", label: "PM10", color: "rgba(54,162,235,0.5)", title: "PM10 (µg/m³)" },
      { id: "aqiChart", key: "aqi", label: "AQI", color: "rgba(255,206,86,0.5)", title: "Air Quality Index" },
      { id: "noChart", key: "no", label: "NO", color: "rgba(255,159,64,0.5)", title: "Nitric Oxide (NO)" },
      { id: "o3Chart", key: "o3", label: "O₃", color: "rgba(153,102,255,0.5)", title: "Ozone (O₃)" },
      { id: "so2Chart", key: "so2", label: "SO₂", color: "rgba(75,192,192,0.5)", title: "Sulfur Dioxide (SO₂)" },
      { id: "nh3Chart", key: "nh3", label: "NH₃", color: "rgba(255,99,255,0.5)", title: "Ammonia (NH₃)" },
    ];

    config.forEach(({ id, key, color, title }) => {
      charts[key] = createChart(id, key, color, title);
    });
  }

  function createChart(canvasId, key, color, title) {
    const ctx = document.getElementById(canvasId).getContext("2d");
    return new Chart(ctx, {
      type: "line",
      data: {
        labels: timeLabels,
        datasets: [{
          label: key.toUpperCase(),
          data: dataSets[key],
          borderColor: color.replace("0.5", "1"),
          backgroundColor: color,
          fill: true,
          tension: 0.3,
          pointRadius: 4,

```

```

    pointHoverRadius: 6
  ]]
},
options: {
  responsive: true,
  plugins: {
    title: { display: true, text: title }
  },
  scales: {
    y: { beginAtZero: true }
  }
}
});
}

```

```

function initMap() {
  const map = L.map("map").setView([currentCoords.lat, currentCoords.lon], 10);
  L.tileLayer("https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {
    attribution: '&copy; OpenStreetMap contributors'
  }).addTo(map);

```

```

  const marker = L.marker([currentCoords.lat, currentCoords.lon]).addTo(map);

```

```

  map.on("click", (e) => {
    currentCoords = { lat: e.latlng.lat, lon: e.latlng.lng };
    marker.setLatLng([currentCoords.lat, currentCoords.lon]);
    clearData();
    fetchAndUpdateData();
  });
}

```

```

function clearData() {
  timeLabels.length = 0;
  Object.values(dataSets).forEach(arr => arr.length = 0);
  Object.values(charts).forEach(chart => chart.update());
}

```

```

async function fetchAndUpdateData() {
  const { lat, lon } = currentCoords;
  const url =
`https://api.openweathermap.org/data/2.5/air_pollution?lat=${lat}&lon=${lon}&appid=${apiKey}`;

```



```

try {
  const response = await fetch(url);
  const json = await response.json();
  const data = json.list[0];
  const comp = data.components;
  const aqi = data.main.aqi;

  const timestamp = new Date().toTimeString().slice(0, 5);

  if (timeLabels.length >= maxDataPoints) {
    timeLabels.shift();
    Object.values(dataSets).forEach(arr => arr.shift());
  }

  timeLabels.push(timestamp);
  dataSets.pm25.push(comp.pm2_5);
  dataSets.pm10.push(comp.pm10);
  dataSets.aqi.push(aqi);
  dataSets.no.push(comp.no);
  dataSets.o3.push(comp.o3);
  dataSets.so2.push(comp.so2);
  dataSets.nh3.push(comp.nh3);

  updateCards(comp.pm2_5, comp.pm10, aqi, comp.no, comp.o3, comp.so2, comp.nh3);

  Object.values(charts).forEach(chart => chart.update());

} catch (e) {
  console.error("Помилка завантаження:", e);
}
}

function updateCards(pm25, pm10, aqi, no, o3, so2, nh3) {
  setCard("pm25", pm25, " µg/m³", getLevelPM(pm25));
  setCard("pm10", pm10, " µg/m³", getLevelPM(pm10));
  setCard("aqi", aqi, "", getLevelAQI(aqi));
  setCard("no", no, " µg/m³", getLevelNO(no));
  setCard("o3", o3, " µg/m³", getLevelO3(o3));
  setCard("so2", so2, " µg/m³", getLevelSO2(so2));
  setCard("nh3", nh3, " µg/m³", getLevelNH3(nh3));
}

```

```
function setCard(id, value, unit = "", level = "") {  
  const el = document.getElementById(id);  
  if (!el) return;  
  el.textContent = value.toFixed(1) + unit;  
  el.className = "indicator";  
  if (level) el.classList.add(level);  
}
```

```
function getLevelPM(value) {  
  if (value <= 12) return "good";  
  if (value <= 35.4) return "moderate";  
  if (value <= 55.4) return "unhealthy";  
  return "hazardous";  
}
```

```
function getLevelAQI(value) {  
  if (value === 1) return "good";  
  if (value === 2) return "moderate";  
  if (value === 3) return "unhealthy";  
  return "hazardous";  
}
```

```
function getLevelNO(value) {  
  if (value <= 53) return "good";  
  if (value <= 100) return "moderate";  
  if (value <= 360) return "unhealthy";  
  return "hazardous";  
}
```

```
function getLevelO3(value) {  
  if (value <= 54) return "good";  
  if (value <= 70) return "moderate";  
  if (value <= 85) return "unhealthy";  
  return "hazardous";  
}
```

```
function getLevelSO2(value) {  
  if (value <= 35) return "good";  
  if (value <= 75) return "moderate";  
  if (value <= 185) return "unhealthy";  
  return "hazardous";  
}
```

```

function getLevelNH3(value) {
  if (value <= 200) return "good";
  if (value <= 400) return "moderate";
  if (value <= 800) return "unhealthy";
  return "hazardous";
}

document.getElementById("exportBtn").addEventListener("click", exportToCSV);

function exportToCSV() {
  let csv = "Time,PM2.5,PM10,AQI,NO,O3,SO2,NH3\n";

  for (let i = 0; i < timeLabels.length; i++) {
    const row = [
      dataSets.pm25[i]?.toFixed(2) ?? "",
      dataSets.pm10[i]?.toFixed(2) ?? "",
      dataSets.aqi[i] ?? "",
      dataSets.no[i]?.toFixed(2) ?? "",
      dataSets.o3[i]?.toFixed(2) ?? "",
      dataSets.so2[i]?.toFixed(2) ?? "",
      dataSets.nh3[i]?.toFixed(2) ?? ""
    ];
    csv += row.join(",") + "\n";
  }

  const blob = new Blob([csv], { type: "text/csv;charset=utf-8;" });
  const url = URL.createObjectURL(blob);
  const a = document.createElement("a");
  a.href = url;
  a.download = `air_quality_data_${new Date().toISOString().slice(0,10)}.csv`;
  a.style.display = "none";
  document.body.appendChild(a);
  a.click();
  document.body.removeChild(a);
  URL.revokeObjectURL(url);
}

```

ДОДАТОК Г – ІЛЮСТРАТИВНА ЧАСТИНА

ІЛЮСТРАТИВНА ЧАСТИНА

**РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ МОНІТОРИНГУ
НАВКОЛИШНЬОГО СЕРЕДОВИЩА**

Вінницький національний технічний університет
Факультет Інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська кваліфікаційна робота

на тему: Розробка вебзастосунку для моніторингу навколишнього середовища

Автор: студент групи 2ПІ-216 Кleshko B. B.
Керівник: к.т.н доц. каф. ПЗ Ракитянська Г. Б.

Рисунок Г.1 – Титульний слайд

Актуальність теми

Сучасний етап розвитку інформаційних технологій супроводжується стрімким зростанням обсягів даних та підвищенням суспільного інтересу до екологічної безпеки, що зумовлює потребу в інструментах для ефективного контролю стану довкілля.

Традиційні системи екологічного моніторингу, як правило, орієнтовані на централізовану обробку даних та характеризуються низькою інтерактивністю й складним інтерфейсом користувача. Це значно ускладнює використання таких систем у повсякденному житті громадян, що зацікавлені у постійному доступі до актуальної інформації про стан навколишнього середовища.

Актуальність теми полягає у потребі створення інформаційної системи, що забезпечить зручний моніторинг стану довкілля, своєчасне виявлення екологічних загроз і ухвалення обґрунтованих рішень. Такий інструмент підвищить екологічну обізнаність, якість життя та ефективність контролю за дотриманням екологічних норм.

Рисунок Г.2 – Актуальність теми

Постановка задачі дослідження

Об'єкт дослідження

Об'єктом дослідження є процес розробки вебсистеми для моніторингу навколишнього середовища з використанням вебтехнологій.

Предмет дослідження

Предметом дослідження є методи та засоби розробки вебсистеми для моніторингу.

Методи дослідження

Методами дослідження є структурний аналіз та об'єктно-орієнтоване проєктування, що використано для формалізації вимог до системи та побудови її моделі. Застосовано методи проєктування вебінтерфейсів із використанням API для обробки та візуалізації екологічних даних.

Основна задача дослідження

Розробка вебзастосунку для моніторингу навколишнього середовища, що забезпечує збір, обробку та візуалізацію екологічних даних у реальному часі.

Рисунок Г.3 – Постановка задачі дослідження

Завдання, що вирішуються в роботі

Відомі задачі

- проведення аналізу сучасних вебсистем екологічного моніторингу та виконання порівняльного аналізу існуючих аналогів;
- обґрунтування вибору архітектури програмного забезпечення та методів реалізації функціональності вебзастосунку.

Авторські задачі

- розробку структури, логічної моделі та алгоритмів функціонування веборієнтованої системи;
- реалізацію програмних модулів для відображення даних і взаємодії з користувачем;
- проведення тестування працездатності функціоналу вебзастосунку та створення інструкції користувача для забезпечення ефективної експлуатації системи.

Рисунок Г.4 – Постановка задачі дослідження – Завдання, що вирішуються в роботі

Новизна результатів

Розроблено вебзастосунок для моніторингу навколишнього середовища, який дозволяє відображати показники якості повітря в режимі реального часу з використанням інтерактивної карти.

Реалізовано механізм візуалізації екологічних показників за допомогою бібліотеки Chart.js, що сприяє підвищенню наочності даних для кінцевого користувача.

Реалізовано модульну архітектуру вебзастосунку, яка забезпечує масштабованість системи та підтримку інтеграції з відкритими джерелами екологічних даних через OpenWeatherAPI.

Рисунок Г.5 – Новизна результатів

Порівняльний аналіз аналогів

Критерії	OpenAQ	Plume Labs	BreezoMeter	PurpleAir	Власна розробка
Глобальне покриття	+	-	+	+	+
API для інтеграції	-	-	+	+	+
Доступність	+	+	-	+	+
Підтримка показників PM2.5/PM10	+	+	+	-	+
Користувачька орієнтація	+	+	+	-	+
Загальна оцінка	80%	60%	80%	60%	100%

Рисунок Г.6 – Порівняльний аналіз аналогів

Розробка архітектурної схеми вебзастосунку

На рисунку зображена архітектурна схема вебзастосунку для моніторингу навколишнього середовища



Рисунок Г.7 – Розробка архітектурної схеми вебзастосунку

Розробка алгоритму роботи системи

На рисунку зображено роботу алгоритму системи



Рисунок Г.8 – Розробка алгоритму роботи системи

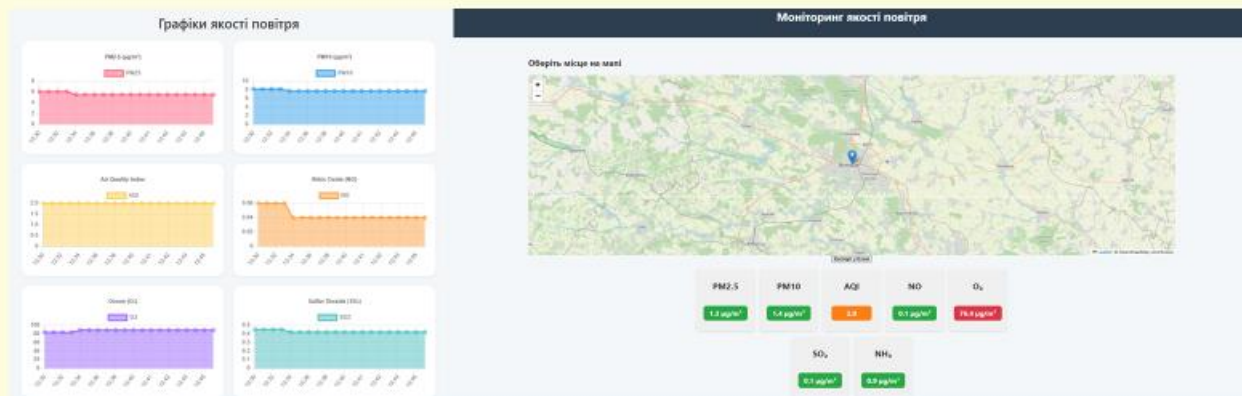
Розробка алгоритму роботи програмного застосунку

На рисунку зображено алгоритм роботи програмного застосунку



Рисунок Г.9 – Розробка алгоритму роботи програмного застосунку

Розробка інтерфейсу користувача



При відкритті вебзастосунку відкривається сторінка з інтерактивною картою, даними про якість повітря та графіками.

Рисунок Г.10 – Розробка інтерфейсу користувача

Тестування вебзастосунку

Результат тестування основних функцій



#	A	B	C	D	E	F	G
1	Time	PM2.5	PM10	AQI	NO	O3	SO2
2	6.14	13.54	3	0.01	100.40	0.05	0.3
3	6.14	13.54	3	0.01	100.40	0.05	0.3
4	6.14	13.54	3	0.01	100.40	0.05	0.3
5	6.14	13.54	3	0.01	100.40	0.05	0.3
6	6.14	13.54	3	0.01	100.40	0.05	0.3
7	6.14	13.54	3	0.01	100.40	0.05	0.3
8	6.14	13.54	3	0.01	100.40	0.05	0.3
9	6.14	13.54	3	0.01	100.40	0.05	0.3
10	6.24	13.68	2	0	97.9	0.08	0.29
11	6.24	13.68	2	0	97.9	0.08	0.29
12	6.24	13.68	2	0	97.9	0.08	0.29
13	6.24	13.68	2	0	97.9	0.08	0.29
14	6.24	13.68	2	0	97.9	0.08	0.29
15	6.24	13.68	2	0	97.9	0.08	0.29
16	6.24	13.68	2	0	97.9	0.08	0.29
17	6.24	13.68	2	0	97.9	0.08	0.29
18	6.24	13.68	2	0	97.9	0.08	0.29
19	6.24	13.68	2	0	97.9	0.08	0.29
20	6.24	13.68	2	0	97.9	0.08	0.29
21	6.24	13.68	2	0	97.9	0.08	0.29



Результат тестування на відмову

PM2.5

PM10

AQI

NO

O₃

SO₂

NH₃

Рисунок Г.11 – Тестування вебзастосунку

Висновки

У рамках бакалаврської роботи розроблені модулі програмного забезпечення для моніторингу навколишнього середовища завдяки вебзастосунку за допомогою API. Для реалізації використовуються HTML, CSS та JavaScript і бібліотеки Chart.js для побудови графіків та Leaflet для інтерактивної карти.

Проведено варіантний аналіз і обґрунтовано вибір засобів реалізації програмного продукту.

Проаналізовано стан вебсистем для моніторингу навколишнього середовища. Розглянуті основні аналоги програмного продукту, виявлені їхні недоліки та порівняно з власним розробленим продуктом.

У межах роботи виконано наступне:

- визначено основні функціональні та нефункціональні вимоги до вебсистеми моніторингу навколишнього середовища;
- розроблено модулі збору та обробки екологічних показників;
- розроблено адаптивний та інтуїтивно зрозумілий вебінтерфейс;
- забезпечено підтримку автоматичного збору даних через API;
- проведено тестування функціональності програмних модулів.

Тестування програми довело повну працездатність даного програмного продукту та відповідність поставленому технічному завданню.

Рисунок Г.12 – Висновки