

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій**

ПОГОДЖЕНО
Декан факультету

_____ **Ігор БОЛБОТ**
(підпис)

« ____ » _____ 2026 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри комп'ютерних наук

_____ **Белла ГОЛУБ**
(підпис)

« ____ » _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Програмне забезпечення з визначення узгодженого екологічного індексу з урахуванням окремих метрик»

Спеціальність «121 Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення»

Гарант освітньої програми

к.т.н., доцент

(підпис)

Ганна ВАЙГАНГ

**Керівник бакалаврської
кваліфікаційної роботи**

к.т.н., доцент

(підпис)

Андрій БРИТАН

Виконав

(підпис)

Євгеній РОМАНЮК

КИЇВ – 2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій**

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

к.т.н., доцент _____ Белла ГОЛУБ
(підпис)

«___» _____ 2025 р.

**ЗАВДАННЯ
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ**

_____ Романюку Євгенію Анатолійовичу

(прізвище, ім'я, по батькові)

Спеціальність «121 Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення»

Тема бакалаврської кваліфікаційної роботи

«Програмне забезпечення з визначення узгодженого
екологічного індексу з урахуванням окремих метрик»

затверджена наказом від «19» грудня 2025 р. № 3196 «С»

Термін подання завершеної роботи на кафедру «22» травня 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи: дані моніторингу якості повітря від Open-Meteo API, нормативи ГДК згідно з ДСТУ та СанПіН, методика European AQI (EEA), метод аналізу ієрархій Сааті (АНР).

Перелік питань, що підлягають дослідженню:

1. Дослідження предметної області та вимог

2. Проектування інформаційної частини

3. Проектування та розробка програмного забезпечення системи

4. Експлуатація та впровадження системи

Перелік графічних документів (за необхідності).

Дата видачі завдання «19» _____ грудня _____ 2025 р.

**Керівник бакалаврської
кваліфікаційної роботи**

_____ **Андрій БРИТАН**
(підпис)

Завдання взяв до виконання

_____ **Євгеній РОМАНЮК**
(підпис)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	5
ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Огляд предметної області	9
1.2 Моделювання предметної області	10
1.3 Огляд аналогів та конкурентних рішень	16
1.4 Постановка задачі та функціональні вимоги	20
2 ПРОЄКТУВАННЯ БАЗИ ДАНИХ	24
2.1 Побудова ER-діаграми	24
2.2 Обґрунтування вибору СУБД.....	27
2.3 Структура бази даних.....	28
2.4 Фізична схема бази даних.....	36
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	43
3.1 Виділення абстракцій та опис класів.....	43
3.2 Структура пакетів та компонентів	50
3.3 Програмний інструментарій та бібліотеки.....	53
3.4 Алгоритми обчислення екологічних індексів.....	55
3.5 Реалізація узгодженого індексу за методом Сааті.....	58
4 ТЕСТУВАННЯ ТА РОЗГОРТАННЯ СИСТЕМИ	61
4.1 Вимоги до апаратного та програмного забезпечення	61
4.2 Склад інсталяційного пакету	63
4.3 Тестування функціональності системи	65
4.4 Результати експлуатації клієнтського застосунку.....	68

ВИСНОВКИ	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	77
ДОДАТОК А	79
ДОДАТОК Б.....	88
ДОДАТОК В	96
ДОДАТОК Г	99

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AQI – Air Quality Index – індекс якості повітря (європейська шкала).

WQI – Water Quality Index – індекс якості води.

SQI – Soil Quality Index – індекс якості ґрунту.

АНР – Analytic Hierarchy Process – метод аналізу ієрархій Сааті.

API – Application Programming Interface – програмний інтерфейс прикладного рівня.

REST – Representational State Transfer – архітектурний стиль побудови розподілених систем.

HTTP – Hypertext Transfer Protocol – протокол передавання гіпертексту.

HTTPS – Hypertext Transfer Protocol Secure – захищений протокол передавання гіпертексту.

JSON – JavaScript Object Notation – формат обміну даними.

JWT – JSON Web Token – токен авторизації стандарту JSON.

СУБД – система управління базами даних.

ПЗ – програмне забезпечення.

UI – User Interface – інтерфейс користувача.

UX – User Experience – досвід користувача.

UML – Unified Modeling Language – уніфікована мова моделювання.

ER – Entity-Relationship – модель “сутність – зв’язок”.

CRUD – Create, Read, Update, Delete – базові операції з даними.

CI/CD – Continuous Integration / Continuous Delivery – безперервна інтеграція та доставка.

PM2.5 – тверді частинки діаметром до 2,5 мкм.

PM10 – тверді частинки діаметром до 10 мкм.

ГДК – гранично допустима концентрація.

ВСТУП

Зростання урбанізації, інтенсивний розвиток промисловості та транспортної інфраструктури призводять до постійного погіршення екологічного стану навколишнього середовища. У великих містах України спостерігається перевищення гранично допустимих концентрацій забруднюючих речовин у повітрі, забруднення поверхневих та підземних вод, деградація ґрунтового покриву внаслідок антропогенного впливу. У зв'язку з цим перед державними структурами, екологічними службами та громадськістю постає задача оперативного отримання об'єктивної інформації про екологічний стан території.

На сьогоднішній день якість окремих компонентів довкілля прийнято оцінювати за допомогою спеціалізованих індексів: AQI (Air Quality Index) – для повітря, WQI (Water Quality Index) – для води, SQI (Soil Quality Index) – для ґрунтів. Однак кожний з цих індексів характеризує лише одну зі складових екологічної ситуації та не дозволяє зробити інтегральну оцінку. У реальних умовах ухвалення управлінських рішень потрібен єдиний узгоджений показник, який враховує одночасно стан повітря, води та ґрунту з вагами, що відповідають важливості кожного компонента в конкретній ситуації.

Актуальність дипломної роботи зумовлена потребою у програмному забезпеченні, яке здатне в режимі, наближеному до реального часу, агрегувати показники якості повітря, води і ґрунту, обчислювати узгоджений екологічний індекс з використанням наукового обґрунтованого методу аналізу ієрархій Т. Сааті, та представляти результат у зручному для користувача вигляді – на інтерактивній географічній мапі з можливістю роботи з користувацькими зонами спостереження.

Об'єктом дослідження є процес визначення інтегрованого екологічного стану території на основі наборів метрик якості повітря, води та ґрунту.

Предметом дослідження є методи, алгоритми та програмні засоби обчислення узгодженого екологічного індексу з використанням методу аналізу ієрархій Сааті.

Метою роботи є розробка клієнт-серверного програмного забезпечення, яке збирає, зберігає та опрацьовує екологічні метрики, обчислює індекси AQI, WQI, SQI та узгоджений екологічний індекс і візуалізує результати на інтерактивній мапі.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- 1) проаналізувати предметну область та існуючі підходи до оцінювання екологічного стану території;
- 2) дослідити методи розрахунку індексів AQI, WQI, SQI та обґрунтувати їх застосовність у розроблюваній системі;
- 3) обрати математичний апарат для агрегування індексів і розробити алгоритм узгодженого екологічного індексу за методом Сааті;
- 4) спроектувати архітектуру клієнт-серверного програмного забезпечення та модель бази даних;
- 5) реалізувати серверну частину засобами фреймворку NestJS та СУБД PostgreSQL з розширенням PostGIS;
- 6) реалізувати клієнтську частину засобами Next.js, React і бібліотеки Leaflet;
- 7) інтегрувати систему із зовнішнім сервісом Open-Meteo Air Quality API;
- 8) здійснити контейнеризацію та розгортання застосунку (Docker, Cloudflare Tunnel);
- 9) провести функціональне та інтеграційне тестування системи.

Методи дослідження. У роботі застосовано методи системного аналізу, метод аналізу ієрархій Т. Сааті, методи об'єктно-орієнтованого проєктування, UML-моделювання, методи експериментального тестування програмного забезпечення.

Наукова новизна одержаних результатів полягає у поєднанні класичного математичного апарату методу Сааті з сучасною модульною архітектурою клієнт-серверного веб-застосунку та геоінформаційним відображенням, що дозволяє у єдиному цикловому середовищі приймати рішення на основі узгодженого екологічного індексу для довільних користувацьких зон спостереження.

Практичне значення одержаних результатів. Розроблений програмний продукт може використовуватися екологічними службами, органами місцевого самоврядування, проєктними та науково-дослідними установами для моніторингу екологічного стану території, а також у навчальному процесі при підготовці фахівців з екологічного моніторингу та інженерії програмного забезпечення.

Структура та обсяг роботи. Пояснювальна записка складається із вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи – 78 сторінок, із яких основного тексту – 67 сторінок.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд предметної області

Предметною областю моєї дипломної роботи є моніторинг якості навколишнього середовища регіонів України. Сучасна екологічна ситуація характеризується значною кількістю чинників, які одночасно впливають на стан повітря, поверхневих і підземних вод, а також на властивості ґрунтів. До таких чинників належать: викиди промислових підприємств, автомобільний транспорт, теплоенергетика, побутове опалення, сільськогосподарська діяльність, аварії на промислових об'єктах та ін.

Якість атмосферного повітря зазвичай характеризується концентраціями таких забруднюючих речовин, як діоксид азоту (NO_2), діоксид сірки (SO_2), монооксид вуглецю (CO), озон (O_3), а також твердими частинками $\text{PM}_{2.5}$ і PM_{10} . Європейська екологічна агенція (ЕЕА) [3] запровадила єдину шкалу AQI (Air Quality Index), яка дозволяє у формі цілочисельного показника від 0 до 100+ оцінити сумарний рівень забруднення повітря [7].

Якість поверхневих та питних вод оцінюється за такими показниками, як водневий показник pH, вміст розчиненого кисню, біохімічне споживання кисню, концентрації нітратів, фосфатів та інших забруднювачів. Інтегрований показник WQI (Water Quality Index) [4] визначається як зважена сума нормованих значень вибраних показників.

Якість ґрунтів характеризується кислотністю, вмістом органічної речовини, концентраціями азоту, фосфору, калію та мікроелементів, а також рівнем забруднення важкими металами. Інтегральний показник SQI (Soil Quality Index) [5] дозволяє звести різнотипні характеристики до єдиного індикатора.

Окремі індекси AQI, WQI, SQI ефективно використовуються при оцінюванні відповідних компонентів довкілля. Однак екосистема є складною та нелінійною: погіршення якості одного компонента (наприклад, повітря) часто

супроводжується змінами в інших (вода, ґрунт). Тому виникає потреба у єдиному узгодженому показнику, який враховує всі три компоненти.

У роботі для агрегування використано метод аналізу ієрархій (АНР, Analytic Hierarchy Process), запропонований Т. Сааті у 1970-х роках [2]. Метод дозволяє формалізувати експертні судження стосовно відносної важливості кожного компонента та обчислити вагові коефіцієнти, які відповідають цим судженням з мінімальною суб'єктивністю.

1.2 Моделювання предметної області

Перед початком розробки програмного забезпечення з визначення узгодженого екологічного індексу необхідно сформувати чітке уявлення про структуру предметної області: виявити її учасників, визначити характер їхньої взаємодії та описати послідовність ключових процесів. Без цього етапу реалізація системи несе ризик невідповідності реальним потребам користувачів і можливих архітектурних прорахунків.

Одним із найпоширеніших підходів до такого опису є використання мови UML (Unified Modeling Language) [10]. Яка надає набір стандартизованих графічних нотацій, кожна з яких висвітлює певний аспект системи – функціональний, часовий або поведінковий. У поєднанні ці нотації утворюють повну картину предметної області ще до написання першого рядка коду.

У межах аналізу предметної області системи екологічного моніторингу розроблено три типи UML-діаграм: діаграму прецедентів, що відображає функціональні можливості системи в розрізі ролей; діаграму послідовності, яка розкриває порядок взаємодії компонентів під час виконання конкретного сценарію; та діаграму активності, котра ілюструє алгоритмічну логіку двох центральних процесів.

1.2.1 Діаграма прецедентів. Діаграма прецедентів фіксує перелік функцій системи та прив'язує кожен з них до конкретної ролі користувача. Це дає змогу

ще на стадії аналізу зрозуміти, яким чином різні категорії учасників взаємодіятимуть із програмним продуктом, та уникнути ситуації, коли у фінальній версії реалізовано “зайвий” функціонал або, навпаки, пропущено критично важливий. Система охоплює трьох акторів із чітко розмежованими повноваженнями та набором доступних операцій.



Рис. 1.1 Діаграма прецедентів ПЗ з визначення узгодженого екологічного індексу

Актори та їхні функції. Користувач (гість) – роль без обов'язкової авторизації. Надає доступ до двох базових сценаріїв: перегляду картографічного шару із відображенням поточного екологічного стану та запиту значень індексу у довільній географічній точці на карті (станції). Ці функції розраховані на широку аудиторію, яка потребує швидкого ознайомлення з ситуацією без заглиблення у деталі розрахунків. Еколог – авторизований фахівець із розширеним набором прав. Може формувати користувацькі зони спостереження для цільового відстеження окремих територій, ініціювати примусовий перерахунок індексів при оновленні вхідних даних, реєструвати нові вимірювальні станції різних типів середовища та генерувати аналітичні звіти з

динамікою показників. Адміністратор – відповідає за технічне супроводження системи. До його обов'язків належать управління обліковими записами (створення, редагування, блокування) та контроль коректності й регулярності надходження даних від підключених станцій. Наведена діаграма чітко розмежовує зони відповідальності між учасниками та слугує відправною точкою для формування переліку функціональних вимог до системи.

1.2.2 Діаграма послідовності. Діаграма послідовності відображає часовий порядок взаємодії між учасниками та компонентами системи під час виконання основних функціональних сценаріїв. Вона дає змогу простежити, які запити надсилаються користувачами, як веб-інтерфейс передає дані на серверну частину, які операції виконує Backend та яким чином система взаємодіє з базою даних.

На відміну від діаграми прецедентів, яка демонструє загальний набір можливостей системи та ролі користувачів, діаграма послідовності деталізує логіку виконання цих можливостей у часі. Для розроблюваної системи основними учасниками взаємодії є Адміністратор, Еколог, Користувач (гість), веб-інтерфейс, Backend та база даних.

На рис. 1.2 наведено діаграму послідовності роботи системи моніторингу екологічного стану територій.

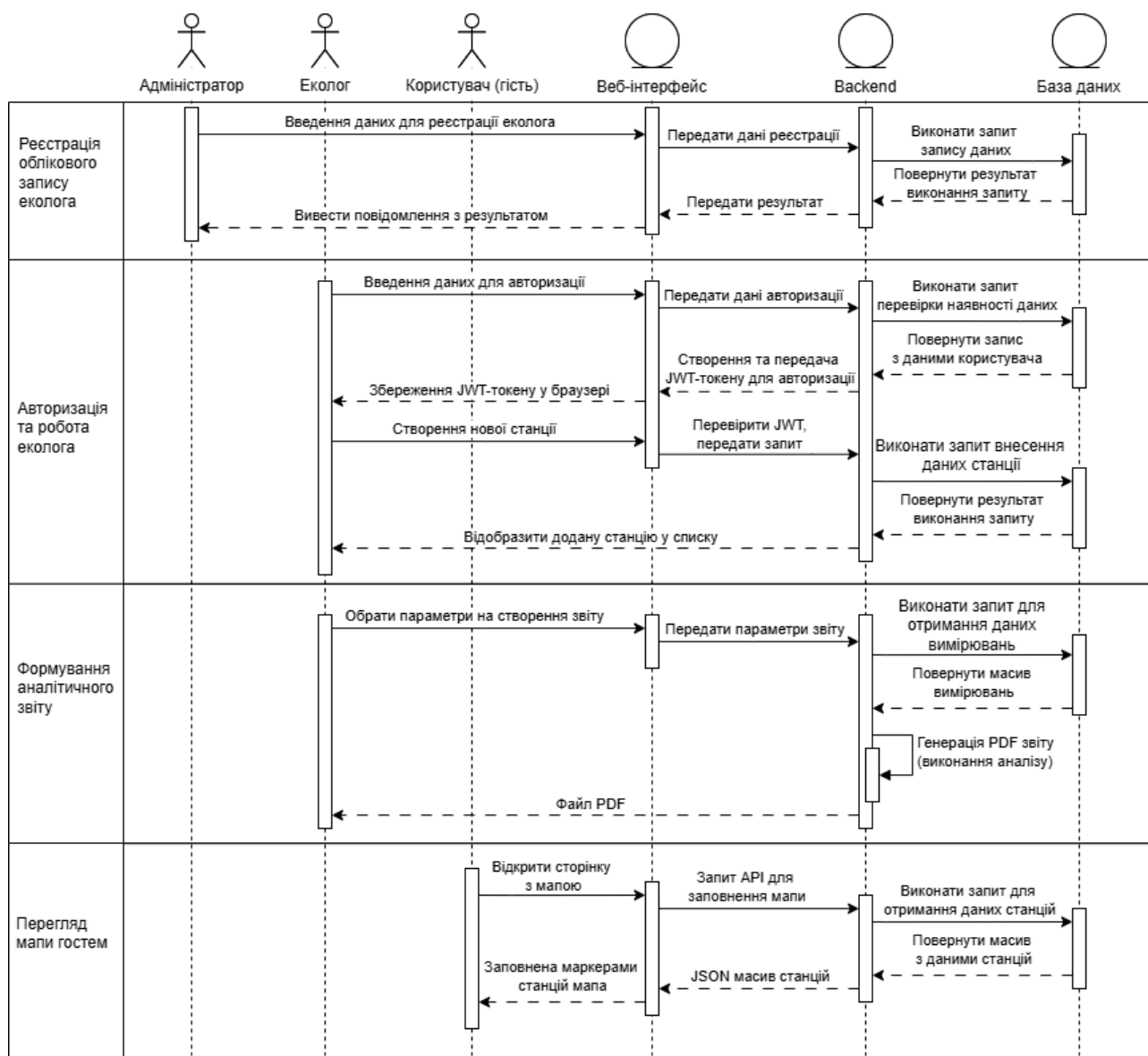


Рис. 1.2 Діаграма послідовності роботи системи

Перший сценарій описує реєстрацію облікового запису еколога. Адміністратор вводить дані для реєстрації еколога у веб-інтерфейсі. Після цього веб-інтерфейс передає реєстраційні дані до Backend, який виконує запит на запис інформації до бази даних. База даних повертає результат виконання запиту, після чого Backend передає відповідь веб-інтерфейсу. У результаті адміністратор отримує повідомлення про успішне створення або помилку реєстрації облікового запису.

Другий сценарій відображає авторизацію та подальшу роботу еколога. Еколог вводить дані для авторизації, які передаються через веб-інтерфейс до Backend. Серверна частина виконує перевірку наявності користувача в базі даних

і, у разі успішної авторизації, створює та передає JWT-токен. Токен зберігається у браузері й використовується для подальших захищених запитів. Після авторизації еколог може створити нову станцію: веб-інтерфейс передає запит до Backend разом із JWT-токеном, Backend перевіряє токен, виконує запит на внесення даних станції до бази даних і повертає результат. Після цього додана станція відображається у списку.

Третій сценарій стосується формування аналітичного звіту екологом. Користувач обирає параметри звіту у веб-інтерфейсі, після чого ці параметри передаються до Backend. Серверна частина виконує запит до бази даних для отримання необхідних вимірювань, проводить аналіз отриманих даних і генерує PDF-звіт. Сформований файл передається назад через веб-інтерфейс екологу для перегляду або завантаження.

Четвертий сценарій демонструє перегляд мапи користувачем-гостем. Користувач відкриває сторінку з мапою, після чого веб-інтерфейс надсилає API-запит до Backend для отримання даних станцій. Backend звертається до бази даних, отримує масив записів зі станціями та повертає його у форматі JSON. Після цього веб-інтерфейс заповнює карту маркерами станцій, що дозволяє користувачу переглядати екологічну інформацію без авторизації.

Таким чином, діаграма послідовності показує взаємодію між користувацькими ролями, клієнтською частиною, серверною логікою та базою даних. Вона підтверджує, що система підтримує як адміністративні функції, так і роботу еколога з даними та відкритий перегляд екологічної інформації для неавторизованих користувачів.

1.2.3 Діаграма активності. Для деталізації внутрішньої логіки двох найважливіших процесів системи розроблено діаграми активності. На відміну від діаграми послідовності, яка фокусується на обміні повідомленнями між компонентами, діаграма активності зосереджена на алгоритмічному потоці – кроках виконання, умовних розгалуженнях і можливих альтернативних шляхах.

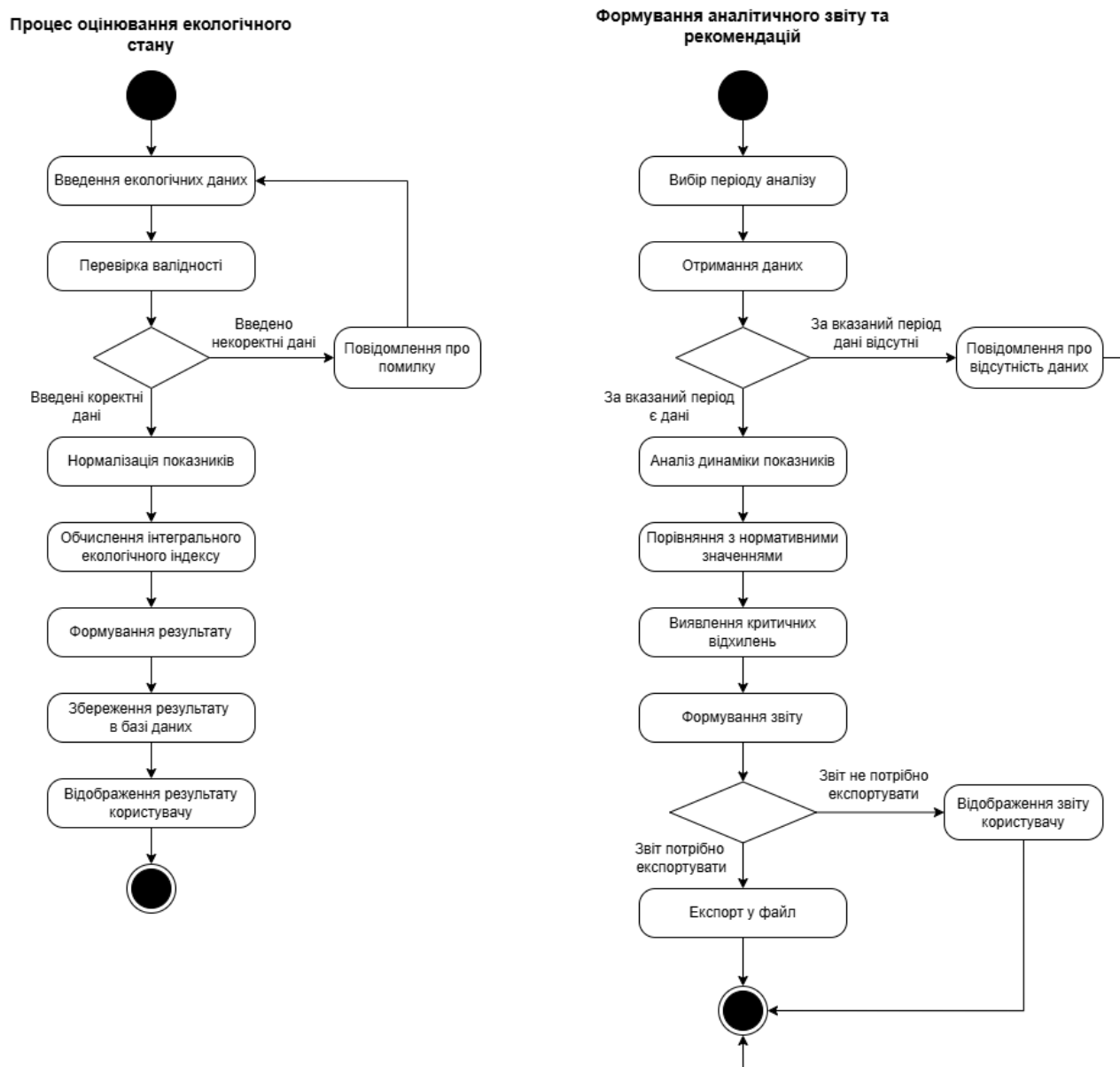


Рис. 1.4 Діаграми активності процесів оцінювання екологічного стану та формування аналітичного звіту

Процес оцінювання екологічного стану. Точкою входу є введення екологічних даних. Система негайно перевіряє їх коректність: якщо вхідні значення містять помилки або виходять за допустимі межі – генерується повідомлення про помилку, і процес повертається до кроку введення. Валідовані дані проходять нормалізацію відповідно до параметрів метрик, після чого обчислюється інтегральний екологічний індекс. Сформований результат зберігається у базі даних і виводиться користувачу у вигляді зрозумілого показника екологічного стану.

Процес формування аналітичного звіту та рекомендацій. Процес ініціюється вибором часового проміжку для аналізу. Система запитує відповідні дані: якщо за вказаний період вимірювань не зафіксовано – користувач отримує повідомлення про відсутність даних і процес завершується. За наявності даних виконується аналіз часової динаміки показників, їх зіставлення з нормативними значеннями та виявлення критичних відхилень. На основі цих результатів формується звіт. Далі система перевіряє, чи необхідний експорт: якщо так – звіт зберігається у зовнішній файл; якщо ні – відображається безпосередньо в інтерфейсі. Обидві діаграми активності розкривають алгоритмічну природу ключових процесів системи, виділяють критичні точки прийняття рішень та є безпосередньою основою для подальшого проектування бізнес-логіки програмного продукту.

1.3 Огляд аналогів та конкурентних рішень

Перед тим як формулювати вимоги до розроблюваної системи, доцільно проаналізувати наявні рішення у суміжних нішах. Це дозволяє виявити функціональні прогалини, уникнути повторення вже вирішених задач та чітко обґрунтувати оригінальність власного підходу.

Серед актуальних програмних продуктів і веб-сервісів у галузі екологічного моніторингу виділяються такі рішення:

IQAir (AirVisual) – комерційна платформа зі світовим охопленням, що надає дані про якість повітря через мобільний застосунок і веб-інтерфейс. Джерелами інформації слугують як офіційні станції моніторингу, так і датчики IoT-мережі компанії. Система дозволяє користувачам переглядати поточний індекс якості повітря, отримувати прогноз забруднення, аналізувати рівні основних забруднювачів та порівнювати екологічну ситуацію в різних містах.

На рис. 1.5 представлено інтерфейс системи IQAir (AirVisual).

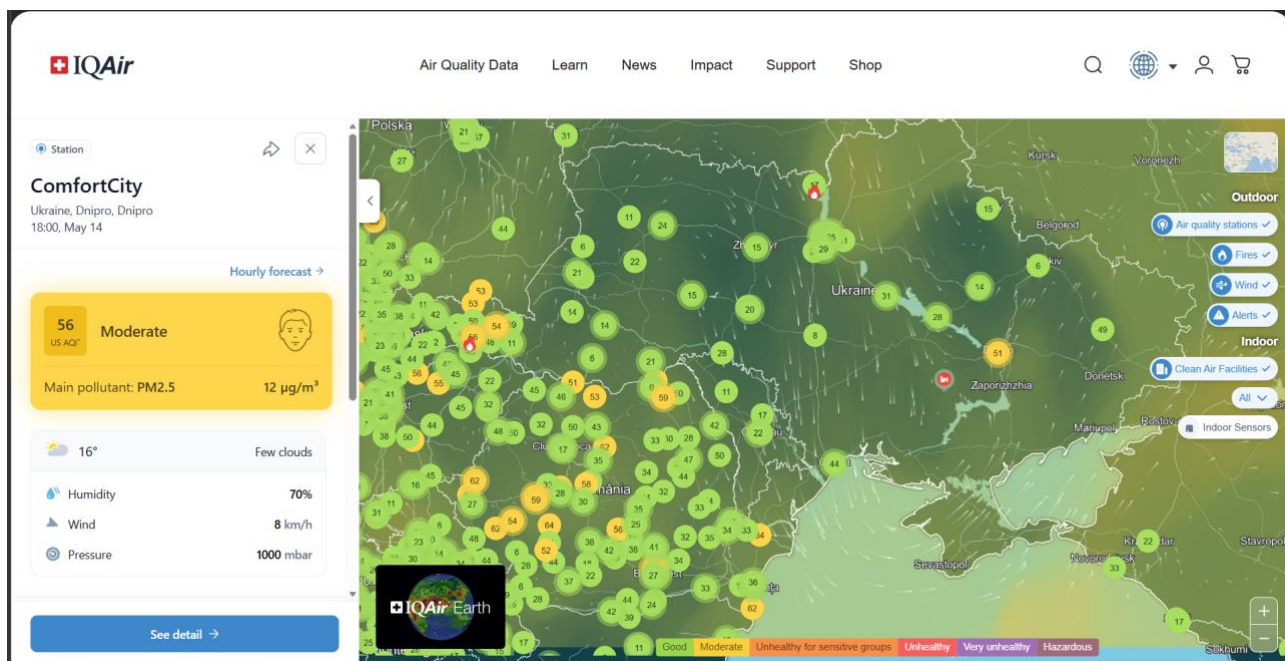


Рис. 1.5 Інтерфейс системи IQAir (AirVisual)

Переваги:

- широке міжнародне охоплення та велика кількість станцій моніторингу;
- наявність мобільного застосунку та зручного веб-інтерфейсу;
- відображення поточного стану якості повітря та прогнозних показників;
- використання як офіційних джерел даних, так і власної мережі датчиків.

Недоліки:

- основна увага приділяється саме якості повітря, без комплексного аналізу води та ґрунту;
- частина функцій і пристроїв є комерційними;
- точність даних у певних регіонах залежить від кількості доступних станцій;
- відсутня можливість створення користувацьких зон спостереження довільної форми.

Saveecobot – вітчизняний проєкт, що агрегує показники з мережі громадських датчиків якості повітря по всій Україні з картографічним відображенням. Система орієнтована на швидке інформування населення про стан атмосферного повітря в конкретних населених пунктах і районах. Користувач може переглядати інтерактивну карту, аналізувати показники забруднення та отримувати інформацію про розташування датчиків.

На рис. 1.6 показано інтерфейс системи Saveecobot.

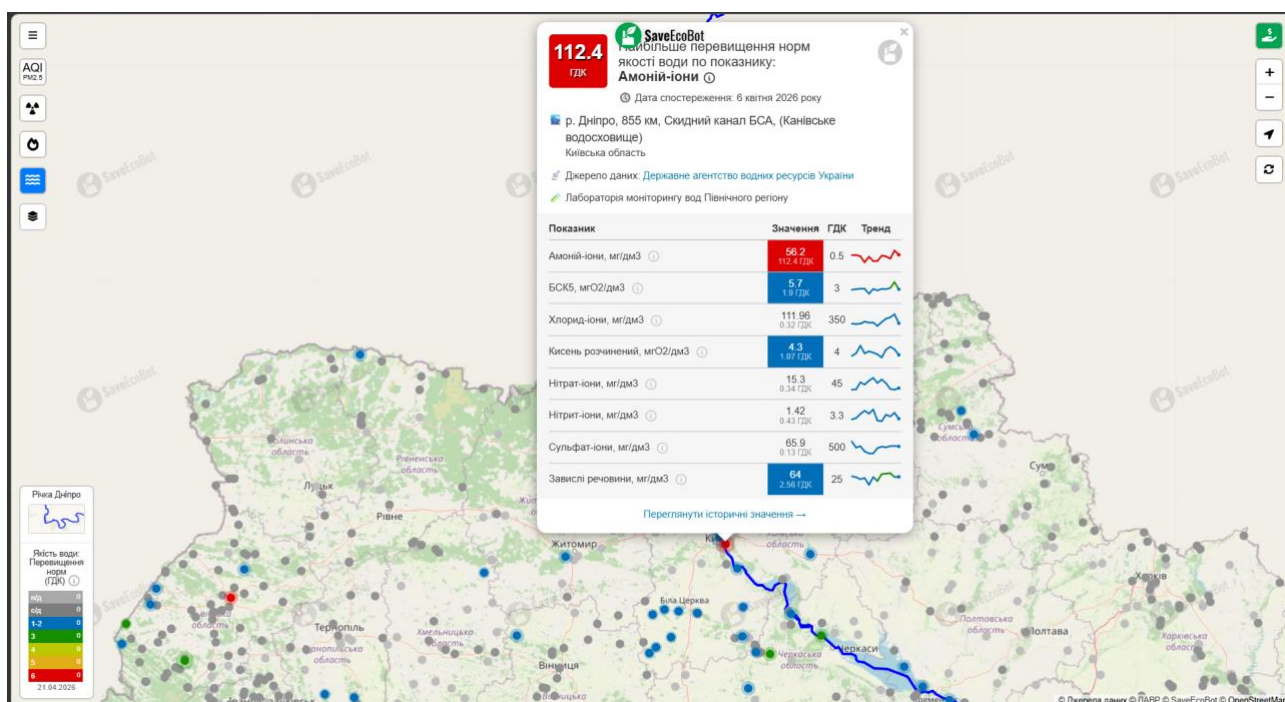


Рис. 1.6 Інтерфейс системи Saveecobot

Переваги:

- орієнтація на територію України;
- інтерактивне картографічне відображення даних;
- використання мережі громадських датчиків;
- наявні вимірювання стану якості води та радіаційного фону;
- простота доступу до інформації через веб-інтерфейс.

Недоліки:

- система зосереджена переважно на моніторингу якості повітря;
- відсутній комплексний аналіз стану ґрунту;

- якість даних залежить від справності та розміщення громадських датчиків;
- обмежені можливості для побудови власних аналітичних зон і формування розширених звітів.

World Air Quality Index (WAQI) – відкрита платформа з інтерактивною картою станцій і публічним API. Вона охоплює більшість офіційних державних мереж моніторингу якості повітря у світі та дозволяє користувачам переглядати актуальні показники забруднення в різних країнах і містах. Платформа надає інформацію про індекс якості повітря, концентрації забруднювачів та місцезнаходження станцій моніторингу.

Інтерфейс системи WAQI наведено на рис. 1.7.

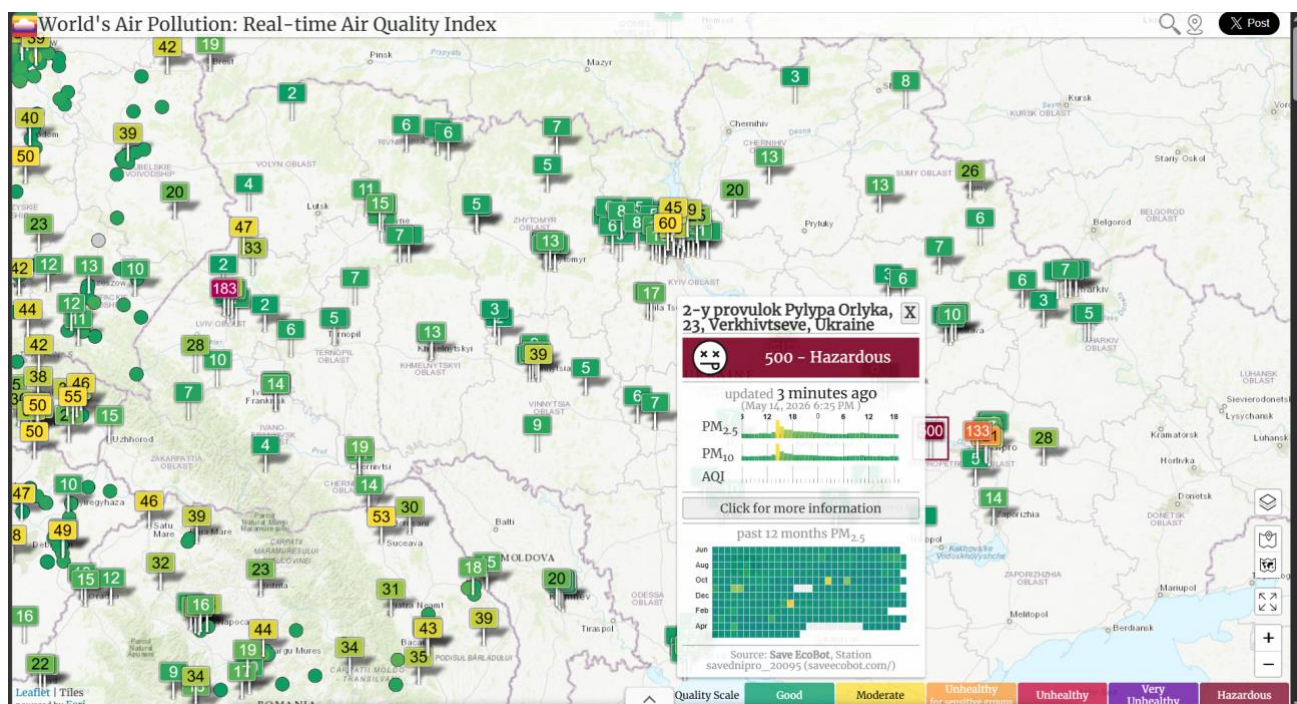


Рис. 1.7 Інтерфейс системи World Air Quality Index

Переваги:

- глобальне охоплення та велика кількість станцій;
- наявність інтерактивної карти;
- підтримка публічного API для отримання даних;
- використання інформації з офіційних державних мереж моніторингу.

Недоліки:

- платформа орієнтована переважно на якість повітря;
- не забезпечується комплексне оцінювання стану води та ґрунту;
- функціональність користувацького аналізу є обмеженою;
- відсутній механізм обчислення єдиного екологічного індексу для різних компонентів довкілля.

1.4 Постановка задачі та функціональні вимоги

Метою даного проєкту є розробка програмного забезпечення для визначення узгодженого екологічного індексу, яке дозволить комплексно оцінювати стан навколишнього середовища одночасно за трьома середовищами – повітря, вода, ґрунт – та надавати результати у зручному картографічному і звітному форматі. Система повинна автоматично збирати актуальні дані, підтримувати ручне введення польових вимірювань і забезпечувати розрахунок узгодженого індексу за методом аналізу ієрархій Сааті.

Вхідні дані системи

Система повинна обробляти такі вхідні дані:

- дані якості повітря (автоматично з Open-Meteo API): концентрація PM2.5, PM10, NO, NO2, CO, O3, SO2, NH3 (мкг/м³), пил (µg/m³), UV-індекс, а також метеопказники – температура (°C), вологість (%), тиск (гПа), швидкість вітру (км/год);
- дані якості води (ручне введення екологом): амоній (NH₄⁺), БСК₅, хлориди, розчинений кисень, нітрати, нітрити, сульфати, завислі речовини (мг/дм³);
- дані стану ґрунту (ручне введення екологом): рН, електропровідність, органічна речовина, макроелементи (NO₃-N, P, K, Ca, Mg, SO₄-S) та мікроелементи (Zn, Mn, Cu, Fe, Co, B, Mo) у ppm;

- параметри станцій та точок спостереження: назва, географічні координати (широта, довгота), тип (district / custom), дата та час вимірювання;
- параметри зон спостереження: назва зони, масив координат вершин полігону у форматі [lat, lng][[]], ідентифікатор власника;
- облікові дані користувачів: електронна пошта, пароль (передається у відкритому вигляді лише по захищеному каналу, зберігається виключно як bcrypt-хеш), роль (ecologist / admin);
- параметри запиту звіту: тип середовища, часовий діапазон (from, to), перелік станцій або зон.

Вихідні дані системи

Система повинна генерувати такі результати:

- субіндекси якості середовищ: European AQI для повітря (0–100, апроксимація за стандартом EEA), WQI для води, SQI для ґрунту;
- узгоджений екологічний індекс для зони: агрегована числова оцінка стану зони спостереження, розрахована за методом аналізу ієрархій Сааті з урахуванням вагових коефіцієнтів трьох середовищ;
- інтерактивна карта: кольорові маркери станцій (колір відповідає значенню AQI), теплова карта, кольорові полігони зон спостереження;
- аналітичні звіти: файли формату PDF та CSV з динамікою показників за обраний період, порівнянням із нормативними значеннями та переліком критичних відхилень;
- JSON-відповіді REST API [11]: структуровані дані станцій, вимірювань та індексів для програмної інтеграції зовнішніх застосунків.

Умовно-постійна інформація

Умовно-постійна інформація передбачає:

- таблиці breakpoint-значень для обчислення субіндексів (PM25_BP, PM10_BP, NO2_BP, O3_BP, SO2_BP) відповідно до стандарту ЕЕА;
- нормативні граничні значення показників якості води та ґрунту для порівняння з виміряними даними;
- перелік 215 районів України з географічними координатами для автоматичного моніторингу (districts.json);
- вагові коефіцієнти метрик для методу аналізу ієрархій Сааті (матриця попарних порівнянь середовищ).

Функціональні вимоги до системи

Авторизація та розмежування доступу. Система підтримує багаторівневу авторизацію на основі JWT-токенів [21] для трьох ролей: гість (неавторизований), еколог, адміністратор. Кожен користувач має доступ виключно до функцій, що відповідають його ролі.

Функції гостя:

- перегляд публічної інтерактивної карти з маркерами станцій;
- отримання значень індексів та детальних метрик у довільній точці карти.

Функції еколога:

- автоматичний збір актуальних показників якості повітря з Open-Meteo API за розкладом Cron;
- ручне введення та редагування вимірювань якості води та стану ґрунту;
- додавання, редагування та видалення станцій збору даних;
- створення, редагування та видалення зон спостереження у вигляді полігонів на карті;
- ініціювання перерахунку узгодженого екологічного індексу для зон;

- формування та вивантаження аналітичних звітів у форматах PDF та CSV.

Функції адміністратора:

- управління обліковими записами користувачів (створення, редагування, блокування);
- моніторинг стану та коректності завантаження вхідних даних;
- перегляд журналів роботи системи.

Нефункціональні вимоги до системи

- продуктивність: час відповіді API на стандартний запит – не більше 500 мс за умови щонайменше 50 одночасних з'єднань;
- надійність: коефіцієнт доступності – не менше 99 %. Збої Open-Meteo API ізолюються на рівні окремих батч-запитів і не зупиняють основний функціонал;
- безпека: паролі зберігаються виключно у вигляді bcrypt-хешів. Усі захищені ендпоінти вимагають дійсного JWT-токену. Персональні дані не передаються стороннім сервісам;
- масштабованість: серверна частина підтримує горизонтальне масштабування: кілька NestJS-контейнерів можуть працювати паралельно за reverse-проху без змін у коді;
- сумісність: клієнтський застосунок коректно відображається у двох останніх версіях Chrome, Edge, Firefox та Safari;
- локалізація: інтерфейс – українською мовою. Код і коментарі – англійською або українською відповідно до контексту.

Сукупність наведених вимог визначає архітектурні рішення та технологічний стек системи, детальний розгляд яких здійснюється у наступному розділі.

2 ПРОЄКТУВАННЯ БАЗИ ДАНИХ

2.1 Побудова ER-діаграми

Першим кроком проєктування бази даних є побудова концептуальної моделі – ER-діаграми (Entity-Relationship diagram), яка відображає сутності предметної області, їх атрибути та зв'язки між ними. Для системи комплексного екологічного моніторингу предметна область охоплює кілька взаємопов'язаних процесів: автоматизований збір показників якості атмосферного повітря, ручне внесення даних з водних об'єктів та ґрунтових станцій, управління спостережними зонами та точками, а також розмежування доступу між різними категоріями користувачів.

Аналіз функціональних вимог до системи, описаних у попередньому розділі, дозволив виявити дев'ять основних сутностей, які необхідно зберігати в базі даних. Кожна сутність відповідає певному аспекту моніторингової діяльності й характеризується власним набором атрибутів.

Сутність “Користувач” (user) зберігає облікові записи всіх зареєстрованих у системі осіб і містить електронну адресу, захищений хеш пароля та роль, що визначає права доступу. Передбачено дві ролі: адміністратор і еколог. Сутність “Станція” (station) описує автоматичні станції моніторингу якості повітря, дані яких надходять із зовнішнього API; атрибути station включають назву, джерело даних та географічні координати. З нею пов'язана сутність “Вимірювання повітря” (air_measurement), яка зберігає багатопараметричні дані: концентрації PM2.5, PM10, NO, NO₂, CO, O₃, SO₂, NH₃, запиленості, ультрафіолетовий індекс, розрахований Європейський індекс якості повітря (european_aqi), а також метеопоказники – температуру, вологість, тиск та швидкість вітру.

Сутності “Станція водного моніторингу” (water_station) та “Вимірювання води” (water_measurement) призначені для зберігання спостережень за якістю

водних об'єктів. До складу вимірювань входять показники амонію, БСК₅, хлоридів, розчиненого кисню, нітратів, нітритів, сульфатів, завислих речовин та інтегральний водний індекс якості (wqi). Аналогічно організовані сутності “Станція ґрунтового моніторингу” (soil_station) і “Вимірювання ґрунту” (soil_measurement), яка охоплює pH, електропровідність, вміст органічної речовини, макроелементи (NO₃-N, P, K, Ca, Mg, SO₄-S) та мікроелементи (Zn, Mn, Cu, Fe, Co, B, Mo), а також ґрунтовий індекс якості (sqi).

Сутність “Зона спостереження” (zone) дозволяє описувати довільні полігональні ділянки за допомогою масиву координат, що зберігається у форматі JSONB. Зони прив'язуються до конкретного користувача через атрибут ownerId. Сутність “Точка повітряного спостереження” (air_point) зберігає координати довільних місць, для яких еколог або адміністратор бажає отримувати дані з найближчих станцій; вона також належить певному користувачеві і може бути публічною або приватною.

Між сутностями встановлено такі зв'язки. Між station та air_measurement реалізовано зв'язок “один до багатьох” (1:N) з каскадним видаленням: при видаленні станції автоматично видаляються всі пов'язані вимірювання. Аналогічний зв'язок 1:N існує між water_station та water_measurement, а також між soil_station та soil_measurement. Сутність user пов'язана зв'язком 1:N із water_station, soil_station, air_point та zone через атрибут ownerId, що дозволяє ідентифікувати власника кожного об'єкта. Станції повітря (station) є загальносистемними й не прив'язані до конкретного користувача, оскільки їх перелік визначається автоматично сервісом планувальника.

На рис. 2.1 наведено ER-діаграму бази даних системи, що відображає всі виявлені сутності, їх ключові атрибути та зв'язки між ними.

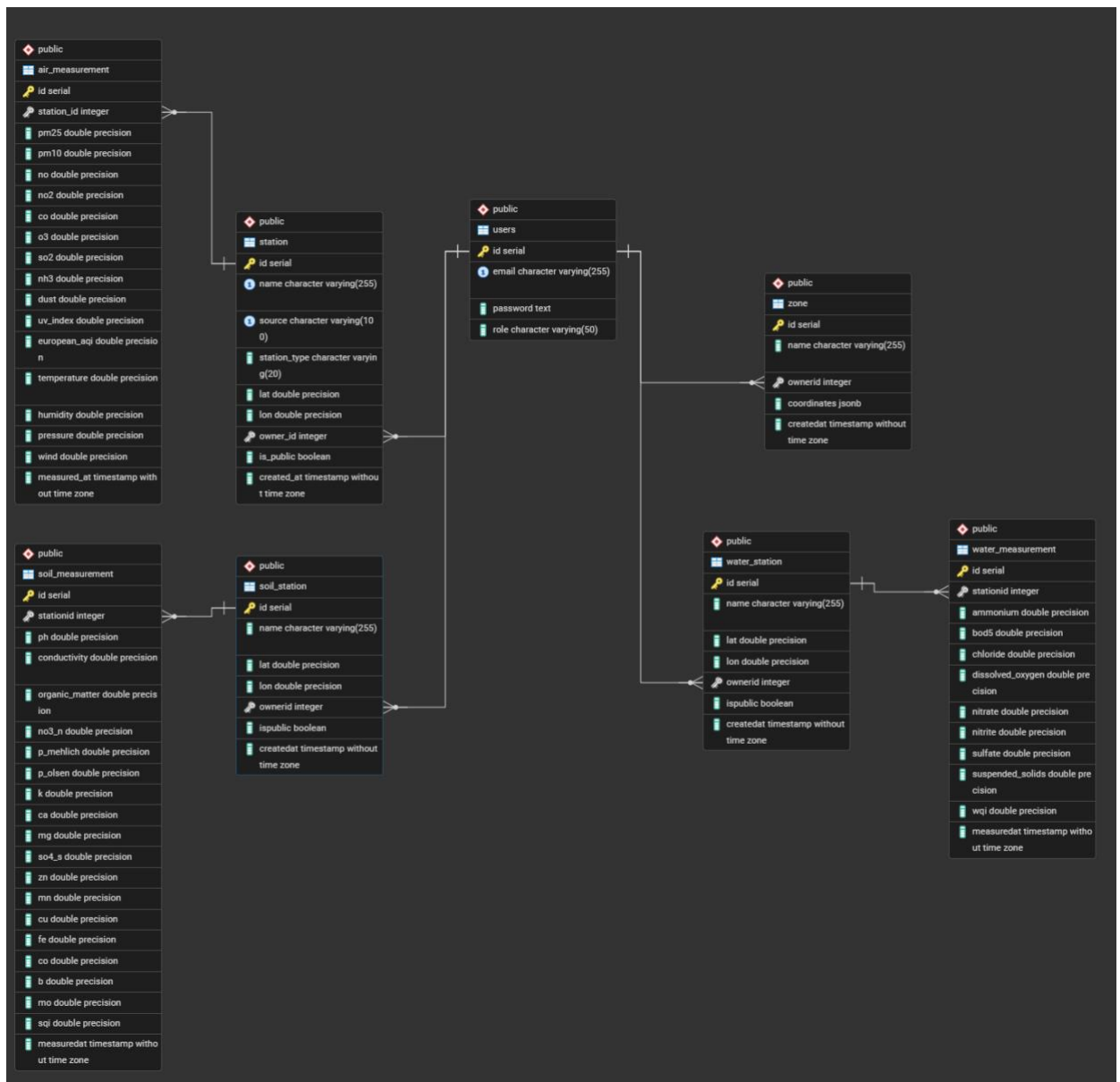


Рис. 2.1 ER-діаграма бази даних системи екологічного моніторингу

Нормалізація схеми виконана до третьої нормальної форми (3NF). Кожна таблиця має єдиний первинний ключ типу serial (автоінкрементний цілочисловий ідентифікатор). Відсутні транзитивні залежності: метеорологічні показники та індекс AQI зберігаються разом із вимірюванням, а не в окремій таблиці, оскільки вони функціонально залежать від конкретного часу спостереження. Таблиці вимірювань навмисно денормалізовані в частині параметрів для спрощення аналітичних запитів і зниження накладних витрат на JOIN-операції при побудові часових рядів.

2.2 Обґрунтування вибору СУБД

Вибір СУБД є одним із ключових архітектурних рішень, оскільки він безпосередньо впливає на продуктивність, надійність і масштабованість усієї системи. Для задачі комплексного екологічного моніторингу сформульовано такі вимоги до СУБД: підтримка повноцінних ACID-транзакцій для гарантії цілісності даних при паралельних записах від планувальника; можливість роботи з просторовими даними (координати, полігони); підтримка складних аналітичних запитів зі з'єднаннями типу LATERAL JOIN; наявність механізму часткових індексів та індексів на кількох стовпцях; відкрите програмне забезпечення з комерційною підтримкою; сумісність із ORM-бібліотекою TypeORM.

Для порівняльного аналізу розглянуто чотири системи: PostgreSQL, MySQL, MongoDB та SQLite. PostgreSQL – об'єктно-реляційна СУБД з відкритим кодом, яка активно розвивається з 1986 року і відрізняється широким набором розширень. MySQL – реляційна СУБД, орієнтована на веб-застосунки з простою схемою. MongoDB – документно-орієнтована NoSQL система без жорсткої схеми. SQLite – вбудована СУБД, призначена для локального зберігання даних у невеликих застосунках.

Таблиця 2.1

Порівняльний аналіз СУБД

Критерій порівняння	PostgreSQL	MySQL	MongoDB	SQLite
<i>ACID-транзакції</i>	Наявне	Наявне	Частково	Наявне
<i>Просторові розширення (PostGIS)</i>	Наявне	Частково	Відсутнє	Відсутнє

<i>Тип JSONB (двійковий JSON)</i>	Наявне	Відсутнє	Наявне	Відсутнє
<i>LATERAL JOIN / складні запити</i>	Наявне	Частоково	Відсутнє	Відсутнє
<i>Часткові та складені індекси</i>	Наявне	Частоково	Відсутнє	Відсутнє
<i>Підтримка TypeORM</i>	Наявне	Наявне	Наявне	Наявне
<i>Вбудована реплікація</i>	Наявне	Наявне	Наявне	Відсутнє
<i>Відкритий вихідний код</i>	Наявне	Наявне	Наявне	Наявне
<i>Придатність для великих об'ємів</i>	Наявне	Наявне	Наявне	Відсутнє

За результатами порівняльного аналізу для реалізації системи обрано PostgreSQL 15 [14] у поєднанні з розширенням PostGIS 3.4 [15]. Саме ця СУБД забезпечує повний набір необхідних можливостей: строгу відповідність ACID, тип даних JSONB для компактного зберігання масиву координат зон спостереження, підтримку LATERAL JOIN для ефективного вибирання останнього вимірювання по кожній станції, а також розширення PostGIS для роботи з геопросторовими об'єктами. Крім того, PostgreSQL добре інтегрується з ORM-бібліотекою TypeORM, що використовується у NestJS-бекенді системи. Для розгортання СУБД було застосовано офіційний Docker-образ postgis/postgis:15-3.4, що забезпечує відтворюваність середовища і спрощує перенесення системи між хостами. База даних отримала ім'я geo_monitoring.

2.3 Структура бази даних

У цьому підрозділі наведено детальний опис кожної таблиці бази даних geo_monitoring. Для кожної таблиці вказано призначення, перелік полів з типами даних, обмеженнями цілісності та семантикою. Схема бази даних формується

автоматично завдяки механізму TypeORM synchronize, який під час запуску бекенду аналізує декоратори entity-класів і виконує необхідні DDL-операції.

2.3.1 Таблиця облікових записів користувачів. Таблиця user зберігає облікові дані всіх зареєстрованих користувачів системи. Автентифікація реалізована за схемою JWT: при вході користувач надає email і пароль, сервер верифікує хеш і повертає підписаний токен. Поле password ніколи не зберігає відкритий пароль – лише bcrypt-хеш з коефіцієнтом складності 10. Поле role визначає набір дозволених операцій: значення 'admin' надає право управляти станціями та користувачами, тоді як 'ecologist' обмежує доступ операціями введення вимірювань і перегляду даних. Таблиця 2.2 описує поля таблиці user.

Таблиця 2.2

Структура таблиці user

Поле	Тип даних	Обмеження	Призначення
id	serial	PK, NOT NULL	Унікальний автоінкрементний ідентифікатор запису
email	varchar	UNIQUE, NOT NULL	Електронна адреса, що використовується як логін; унікальна в межах системи
password	varchar	NOT NULL	Бcrypt-хеш пароля (cost factor 10); відкритий пароль не зберігається
role	varchar	DEFAULT 'ecologist'	Роль: 'admin' – повний доступ, 'ecologist' – обмежений доступ

2.3.2 Таблиці станцій та вимірювань якості повітря. Таблиця station описує автоматичні станції моніторингу атмосферного повітря, перелік яких формується автоматично сервісом-планувальником при першому зверненні до API. Складений унікальний індекс на полях (source, name) унеможливорює дублювання станцій з одного джерела. Таблиця 2.3 містить опис полів.

Таблиця 2.3

Структура таблиці station

Поле	Тип даних	Обмеження	Призначення
id	serial	PK, NOT NULL	Автоінкрементний ідентифікатор станції
source	varchar	NOT NULL	Ідентифікатор джерела даних (наприклад, 'open-meteo')
name	varchar	NOT NULL	Назва або код станції відповідно до джерела
lat	float8	NOT NULL	Географічна широта станції у десяткових градусах
lon	float8	NOT NULL	Географічна довгота станції у десяткових градусах

Таблиця `air_measurement` акумулює результати вимірювань, що надходять щогодини від планувальника через API Open-Meteo. Кожен запис відповідає одному часовому зрізу для конкретної станції. Усі числові поля оголошено як `nullable`, оскільки певні параметри можуть бути відсутні в конкретному джерелі даних. Поле `european_aqi` розраховується бекендом за кусково-лінійною функцією відповідно до стандарту Європейського агентства з охорони навколишнього середовища (EEA) на основі значень PM2.5, PM10, NO₂, O₃ і SO₂. Складений індекс на (`stationId`, `measuredAt`) забезпечує швидкодію при вибірках часових рядів. Таблиця 2.4 описує поля таблиці `air_measurement`.

Таблиця 2.4

Структура таблиці air_measurement

Поле	Тип даних	Обмеження	Призначення
id	serial	PK, NOT NULL	Автоінкрементний ідентифікатор запису вимірювання
stationId	int4	FK station(id), CASCADE	Зовнішній ключ на станцію; при видаленні станції – каскадне видалення

pm25	float8	NULL	Концентрація дрібнодисперсних частинок PM2.5, мкг/м³
pm10	float8	NULL	Концентрація завислих частинок PM10, мкг/м³
no	float8	NULL	Концентрація оксиду азоту NO, мкг/м³
no2	float8	NULL	Концентрація діоксиду азоту NO ₂ , мкг/м³
co	float8	NULL	Концентрація чадного газу CO, мкг/м³
o3	float8	NULL	Концентрація озону O ₃ , мкг/м³
so2	float8	NULL	Концентрація діоксиду сірки SO ₂ , мкг/м³
nh3	float8	NULL	Концентрація аміаку NH ₃ , мкг/м³
dust	float8	NULL	Концентрація пилу, мкг/м³
uv_index	float8	NULL	Ультрафіолетовий індекс (безрозмірна величина 0–11+)
european_aqi	float8	NULL	Розрахований Європейський AQI (0–500); значення до 100 вважається прийнятним
temperature	float8	NULL	Температура повітря, °C
humidity	float8	NULL	Відносна вологість повітря, %
pressure	float8	NULL	Атмосферний тиск, гПа
wind	float8	NULL	Швидкість вітру, м/с
measuredAt	timestamp	DEFAULT CURRENT_TIMESTAMP	Час фіксації вимірювання; формує часовий ряд спостережень

2.3.3 Таблиці станцій та вимірювань якості води. Таблиця water_station описує пункти спостереження за водними об'єктами, які створюються вручну користувачами системи. На відміну від станцій повітря, пункти водного

моніторингу прив'язані до конкретного власника через поле `ownerId` і можуть бути помічені як публічні або приватні. Публічні пункти доступні для перегляду всім зареєстрованим користувачам, приватні – лише власнику. Структуру таблиці наведено в таблиці 2.5.

Таблиця 2.5

Структура таблиці `water_station`

Поле	Тип даних	Обмеження	Призначення
id	serial	PK, NOT NULL	Автоінкрементний ідентифікатор пункту спостереження
name	varchar	NOT NULL	Назва пункту водного моніторингу
lat	float8	NOT NULL	Географічна широта пункту, десяткові градуси
lon	float8	NOT NULL	Географічна довгота пункту, десяткові градуси
ownerId	int4	NULL, FK user	Ідентифікатор власника; nullable для системних пунктів
isPublic	bool	DEFAULT true	Ознака публічності: true – доступно всім, false – лише власнику
createdAt	timestamp	DEFAULT CURRENT_TIMESTAMP	Дата та час реєстрації пункту в системі

Таблиця `water_measurement` фіксує результати лабораторних аналізів проб води, які вносяться екологами вручну. Набір параметрів відповідає стандарту ДСТУ 4808:2007 [8] щодо якості питної та поверхневої води. На основі вимірюваних концентрацій обчислюється інтегральний водний індекс якості `wqi`. Структуру таблиці наведено в таблиці 2.6.

Таблиця 2.6

Структура таблиці water_measurement

Поле	Тип даних	Обмеження	Призначення
id	serial	PK, NOT NULL	Автоінкрементний ідентифікатор запису
stationId	int4	FK water_station, CASCADE	Зовнішній ключ на пункт спостереження
ammonium	float8	NULL	Вміст амонію NH_4^+ , мг/дм ³
bod5	float8	NULL	Біологічне споживання кисню за 5 діб (БСК ₅), мг O_2 /дм ³
chloride	float8	NULL	Вміст хлоридів Cl^- , мг/дм ³
dissolved_oxygen	float8	NULL	Концентрація розчиненого кисню, мг/дм ³
nitrate	float8	NULL	Вміст нітратів NO_3^- , мг/дм ³
nitrite	float8	NULL	Вміст нітритів NO_2^- , мг/дм ³
sulfate	float8	NULL	Вміст сульфатів SO_4^{2-} , мг/дм ³
suspended_solids	float8	NULL	Концентрація завислих речовин, мг/дм ³
wqi	float8	NULL	Інтегральний водний індекс якості (Water Quality Index, 0–100)
measuredAt	timestamp	DEFAULT CURRENT_TIMESTAMP	Момент відбору проби або внесення результату

2.3.4 Таблиці станцій та вимірювань якості ґрунту. Таблиця soil_station за своєю структурою аналогічна water_station і описує ділянки, де проводиться агрохімічний аналіз ґрунту. Кожен об'єкт може належати конкретному користувачеві або бути загальнодоступним. Структуру таблиці наведено в таблиці 2.7.

Таблиця 2.7

Структура таблиці soil_station

Поле	Тип даних	Обмеження	Призначення
id	serial	PK, NOT NULL	Автоінкрементний ідентифікатор ділянки ґрунтового моніторингу
name	varchar	NOT NULL	Назва ділянки або агрохімічного полігону
lat	float8	NOT NULL	Географічна широта місця відбору зразків, десяткові градуси
lon	float8	NOT NULL	Географічна довгота місця відбору зразків, десяткові градуси
ownerId	int4	NULL	Ідентифікатор власника ділянки; nullable для системних об'єктів
isPublic	bool	DEFAULT true	Ознака доступності ділянки для інших користувачів системи
createdAt	timestamp	DEFAULT CURRENT_TIMESTAMP	Дата реєстрації ділянки в системі

Таблиця soil_measurement містить результати агрохімічного аналізу ґрунтових зразків [9]. Параметри згруповано у три блоки: основні фізико-хімічні показники (рН, питома електропровідність, вміст органічної речовини), макроелементи (азот нітратний, рухомий фосфор за методами Мелліха та Олсена, обмінний калій, кальцій, магній, сульфат-сірка) та мікроелементи (цинк, марганець, мідь, залізо, кобальт, бор, молібден). На основі цих даних розраховується ґрунтовий індекс якості sqi. Структуру таблиці наведено в таблиці 2.8.

Таблиця 2.8

Структура таблиці soil_measurement

Поле	Тип даних	Обмеження	Призначення
id	serial	PK, NOT NULL	Автоінкрементний ідентифікатор запису
stationId	int4	FK soil_station, CASCADE	Зовнішній ключ на ділянку ґрунтового моніторингу
ph	float8	NULL	Реакція ґрунтового розчину (pH у водній суспензії)
conductivity	float8	NULL	Питома електропровідність, мСм/см
organic_matter	float8	NULL	Вміст органічної речовини (гумусу), %
no3_n	float8	NULL	Вміст нітратного азоту NO ₃ -N, мг/кг (ppm)
p_mehlich	float8	NULL	Рухомий фосфор за методом Мелліха-3, мг/кг
p_olsen	float8	NULL	Рухомий фосфор за методом Олсена, мг/кг
k	float8	NULL	Обмінний калій K, мг/кг
ca	float8	NULL	Обмінний кальцій Ca, мг/кг
mg	float8	NULL	Обмінний магній Mg, мг/кг
so4_s	float8	NULL	Сульфат-сірка SO ₄ -S, мг/кг
zn	float8	NULL	Цинк Zn, мг/кг
mn	float8	NULL	Марганець Mn, мг/кг
cu	float8	NULL	Мідь Cu, мг/кг
fe	float8	NULL	Залізо Fe, мг/кг
co	float8	NULL	Кобальт Co, мг/кг

b	float8	NULL	Бор В, мг/кг
mo	float8	NULL	Молібден Мо, мг/кг
sqi	float8	NULL	Інтегральний ґрунтовий індекс якості (Soil Quality Index, 0–100)
measuredAt	timestamp	DEFAULT CURRENT_TIMESTAMP	Момент відбору зразка або внесення результату аналізу

2.3.5 Таблиці зон спостереження. Таблиця zone зберігає геопросторові полігони, що визначають зони спостереження, для яких формуються аналітичні звіти. Межі зони описуються масивом координатних пар `[[lat, lng], ...]` і зберігаються у стовпці типу JSONB, що забезпечує компактне двійкове представлення та підтримку оператора індексування. Кожна зона належить конкретному користувачеві. Структуру таблиці наведено в таблиці 2.9.

Таблиця 2.9

Структура таблиці zone

Поле	Тип даних	Обмеження	Призначення
id	serial	PK, NOT NULL	Автоінкрементний ідентифікатор зони спостереження
name	varchar	NOT NULL	Довільна назва зони, що задається користувачем
ownerId	int4	NOT NULL	Ідентифікатор власника зони (посилання на user.id)
coordinates	jsonb	NOT NULL	Масив пар <code>[[lat, lng], ...]</code> – вершини полігону зони
createdAt	timestamp	DEFAULT CURRENT_TIMESTAMP	Дата та час створення зони в системі

2.4 Фізична схема бази даних

Фізична схема бази даних є конкретизацією логічної схеми відносно обраної СУБД – PostgreSQL 15. Вона включає визначення типів стовпців, обмежень цілісності, індексів та каскадних правил. У рамках проєкту схему не потрібно створювати вручну: TypeORM у режимі `synchronize:true` автоматично генерує та застосовує відповідні DDL-команди при кожному запуску сервера. Нижче наведено SQL-представлення ключових таблиць бази даних для ілюстрації фізичної структури.

Таблиця користувачів `user` реалізує систему автентифікації та рольового розмежування доступу. Унікальне обмеження на поле `email` запобігає реєстрації дублікатів, а значення за замовчуванням для ролі гарантує, що новий користувач отримає мінімально необхідні права. Нижче наведено DDL-опис таблиці `user`.

```
CREATE TABLE user (
  id    SERIAL PRIMARY KEY,
  email VARCHAR NOT NULL UNIQUE,
  password VARCHAR NOT NULL,
  role  VARCHAR NOT NULL DEFAULT 'ecologist'
);
```

Таблиця `station` зберігає перелік зовнішніх станцій моніторингу повітря. Складений унікальний індекс `UNIQUE(source, name)` забезпечує ідемпотентність операції `UPSERT` при щогодинному оновленні даних планувальником. Нижче наведено DDL-опис таблиці `station`.

```
CREATE TABLE station (
  id    SERIAL PRIMARY KEY,
  source VARCHAR NOT NULL,
  name  VARCHAR NOT NULL,
  lat   DOUBLE PRECISION NOT NULL,
  lon   DOUBLE PRECISION NOT NULL,
  CONSTRAINT uq_station UNIQUE (source, name)
);
```

Таблиця `air_measurement` акумулює почасові вимірювання від усіх станцій. Зовнішній ключ з правилом `ON DELETE CASCADE` гарантує видалення осиротілих записів при видаленні батьківської станції. Складений індекс на `(stationId, measuredAt)` забезпечує ефективне виконання `LATERAL JOIN` запитів, що вибирають останнє вимірювання по кожній станції. Записи, старші за 48 годин, видаляються автоматично планувальником, що обмежує зростання обсягу таблиці. Нижче наведено DDL-опис таблиці `air_measurement`.

```
CREATE TABLE air_measurement (
  id      SERIAL PRIMARY KEY,
  "stationId" INTEGER NOT NULL
    REFERENCES station(id) ON DELETE CASCADE,
  pm25    DOUBLE PRECISION,
  pm10    DOUBLE PRECISION,
  no      DOUBLE PRECISION,
  no2     DOUBLE PRECISION,
  co      DOUBLE PRECISION,
  o3      DOUBLE PRECISION,
  so2     DOUBLE PRECISION,
  nh3     DOUBLE PRECISION,
  dust    DOUBLE PRECISION,
  uv_index DOUBLE PRECISION,
  european_aqi DOUBLE PRECISION,
  temperature DOUBLE PRECISION,
  humidity DOUBLE PRECISION,
  pressure DOUBLE PRECISION,
  wind    DOUBLE PRECISION,
  "measuredAt" TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE INDEX idx_air_meas_station_time
  ON air_measurement ("stationId", "measuredAt");
```

Таблиця `water_station` зберігає пункти водного моніторингу, що реєструються користувачами. Поле `ownerId` є `nullable`, що дозволяє адміністратору створювати загальносистемні пункти без прив'язки до конкретного власника. Нижче наведено DDL-опис таблиці `water_station`.

```
CREATE TABLE water_station (
  id      SERIAL PRIMARY KEY,
  name    VARCHAR NOT NULL,
  lat     DOUBLE PRECISION NOT NULL,
  lon     DOUBLE PRECISION NOT NULL,
  "ownerId" INTEGER,
  "isPublic" BOOLEAN NOT NULL DEFAULT true,
  "createdAt" TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Таблиця `water_measurement` фіксує результати аналізу якості проб води. Зовнішній ключ з правилом `ON DELETE CASCADE` забезпечує автоматичне видалення вимірювань при видаленні батьківського пункту. Складений індекс на `(stationId, measuredAt)` прискорює вибірки часових рядів. Нижче наведено DDL-опис таблиці `water_measurement`.

```
CREATE TABLE water_measurement (
  id      SERIAL PRIMARY KEY,
  "stationId" INTEGER NOT NULL
    REFERENCES water_station(id) ON DELETE CASCADE,
  ammonium DOUBLE PRECISION,
  bod5     DOUBLE PRECISION,
  chloride  DOUBLE PRECISION,
  dissolved_oxygen DOUBLE PRECISION,
  nitrate   DOUBLE PRECISION,
  nitrite   DOUBLE PRECISION,
  sulfate   DOUBLE PRECISION,
  suspended_solids DOUBLE PRECISION,
  wqi       DOUBLE PRECISION,
  "measuredAt" TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE INDEX idx_water_meas_station_time
  ON water_measurement ("stationId", "measuredAt");
```

Таблиця `soil_station` описує ділянки агрохімічного моніторингу ґрунту. Структурно вона ідентична `water_station` і дозволяє задати публічні або приватні

ділянки з координатами та прив'язкою до власника. Нижче наведено DDL-опис таблиці soil_station.

```
CREATE TABLE soil_station (
  id      SERIAL PRIMARY KEY,
  name    VARCHAR NOT NULL,
  lat     DOUBLE PRECISION NOT NULL,
  lon     DOUBLE PRECISION NOT NULL,
  "ownerId" INTEGER,
  "isPublic" BOOLEAN NOT NULL DEFAULT true,
  "createdAt" TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Таблиця soil_measurement зберігає результати повного агрохімічного аналізу: основні показники ґрунту, макро- та мікроелементи, а також розрахований ґрунтовий індекс якості sqi. Каскадне видалення та складений індекс реалізовані аналогічно до water_measurement. Нижче наведено DDL-опис таблиці soil_measurement.

```
CREATE TABLE soil_measurement (
  id      SERIAL PRIMARY KEY,
  "stationId" INTEGER NOT NULL
    REFERENCES soil_station(id) ON DELETE CASCADE,
  ph      DOUBLE PRECISION,
  conductivity DOUBLE PRECISION,
  organic_matter DOUBLE PRECISION,
  no3_n   DOUBLE PRECISION,
  p_mehlich DOUBLE PRECISION,
  p_olsen DOUBLE PRECISION,
  k       DOUBLE PRECISION,
  ca      DOUBLE PRECISION,
  mg      DOUBLE PRECISION,
  so4_s   DOUBLE PRECISION,
  zn      DOUBLE PRECISION,
  mn      DOUBLE PRECISION,
  cu      DOUBLE PRECISION,
  fe      DOUBLE PRECISION,
  co      DOUBLE PRECISION,
```

```

b      DOUBLE PRECISION,
mo     DOUBLE PRECISION,
sqi    DOUBLE PRECISION,
"measuredAt"  TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE INDEX idx_soil_meas_station_time
ON soil_measurement ("stationId", "measuredAt");

```

Таблиця `zone` використовує тип `JSONB` для зберігання масиву координат полігону зони спостереження. Це дозволяє описувати довільні полігональні ділянки без необхідності в окремій нормалізованій таблиці координат, що спрощує запити та знижує кількість `JOIN`. Нижче наведено DDL-опис таблиці `zone`.

```

CREATE TABLE zone (
  id      SERIAL PRIMARY KEY,
  name    VARCHAR NOT NULL,
  "ownerId"  INTEGER NOT NULL,
  coordinates JSONB NOT NULL,
  "createdAt"  TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

Таблиця `air_point` зберігає довільні точки спостереження за якістю повітря. При запиті даних для точки система автоматично знаходить географічно найближчу станцію і повертає її останнє вимірювання. Поле `ownerId` є `nullable`, що дозволяє системним точкам існувати без прив'язки до конкретного користувача. Нижче наведено DDL-опис таблиці `air_point`.

```

CREATE TABLE air_point (
  id      SERIAL PRIMARY KEY,
  name    VARCHAR NOT NULL,
  lat     DOUBLE PRECISION NOT NULL,
  lon     DOUBLE PRECISION NOT NULL,
  "ownerId"  INTEGER,
  "isPublic"  BOOLEAN NOT NULL DEFAULT true,
  "createdAt"  TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

Наведені DDL-визначення формуються автоматично бібліотекою TypeORM на підставі декораторів entity-класів. Параметр `synchronize:true` у конфігурації модуля TypeORM у файлі `app.module.ts` забезпечує автоматичне застосування схеми до бази даних `geo_monitoring` при кожному запуску NestJS-сервера, що спрощує процес розгортання та підтримки актуальності структури бази даних без виконання ручних міграцій.

Загалом фізична схема бази даних `geo_monitoring` включає дев'ять таблиць: `user`, `station`, `air_measurement`, `air_point`, `zone`, `water_station`, `water_measurement`, `soil_station` та `soil_measurement`. Для оптимізації часових вибірок створено три складені індекси на парах (`stationId`, `measuredAt`) – для таблиць `air_measurement`, `water_measurement` та `soil_measurement`. Додатково на таблиці `station` визначено складений унікальний індекс `UNIQUE(source, name)` для ідемпотентного UPSERT. Тип `JSONB` у таблиці `zone` забезпечує компактне двійкове представлення масиву координат полігонів зон спостереження. Сукупність первинних ключів, зовнішніх ключів з каскадним видаленням, унікальних обмежень та індексів гарантує цілісність даних і достатню продуктивність системи за умов щогодинного автоматичного оновлення та ручного введення вимірювань.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Виділення абстракцій та опис класів

Об'єктно-орієнтований аналіз предметної області передбачає виявлення ключових сутностей, що беруть участь у функціонуванні системи, та їх формалізацію у вигляді абстракцій. Кожна абстракція фіксує лише ті властивості реального об'єкта або процесу, які є значущими для реалізації програмного продукту, відкидаючи несуттєві деталі. Для розроблюваної системи визначення узгодженого екологічного індексу сформовано десять базових абстракцій, які охоплюють усі ролі учасників, об'єкти моніторингу та процеси обробки даних.

3.1.1 Абстракції предметної області. Першу групу складають абстракції, що моделюють взаємодію з картографічним інтерфейсом. Абстракція “Карта” відповідає за географічне відображення інформації: вона містить властивості географічної основи, поточного масштабу, маркерів станцій, шарів даних і видимої області перегляду. Її обов'язки полягають у відображенні станцій моніторингу на тайловому підкладі, візуалізації екологічних індексів та забезпеченні навігації картою. Абстракція “Користувач-гість” описує неавтентифікованого відвідувача: він має сесійний ідентифікатор і може переглядати карту, отримувати публічні екологічні показники та обирати маркери станцій, але не може вносити зміни до системи.

До групи сутностей моніторингу відносяться чотири абстракції. “Станція моніторингу” – це фізична або логічна точка збору екологічних даних, що характеризується назвою, типом, координатами та статусом активності; її обов'язок – надавати дані вимірювань і відображатися на карті. “Вимірювання” фіксує одиничний замір параметра навколишнього середовища: дату та час, тип показника, числове значення, одиницю виміру і джерело даних; воно бере участь у розрахунках індексів і формуванні звітів. “Зона спостереження” задає

геопросторовий контур аналізу – полігон із координатами меж, площею і типом – і дозволяє групувати станції та вимірювання для екологічної оцінки певної території. “Екологічний індекс” являє собою розрахований інтегральний показник стану середовища: він несе числове значення, дату обчислення, тип середовища і метод розрахунку та є основою для оцінки рівня забруднення.

Аналітичний аспект системи відображає абстракція “Звіт”, що узагальнює результати моніторингу за певний період: містить назву, діапазон аналізу, результати, графіки і дату формування, а також підтримує експорт у файл. “Обліковий запис” зберігає аутентифікаційні дані – логін, захищений пароль, роль та дату створення – і забезпечує підтвердження прав доступу та ідентифікацію кожного зареєстрованого учасника системи. Рольовий рівень утворюють дві абстракції. “Адміністратор” – привілейований суб'єкт з акаунтом і розширеними правами: він створює облікові записи екологів, керує даними станцій і контролює доступ до системи. “Еколог” – основний робочий користувач, що авторизується в системі, реєструє нові станції моніторингу, вносить вимірювання та формує аналітичні звіти.

На рис. 3.1 наведено схему абстракцій предметної області з переліком важливих властивостей та обов'язків кожної абстракції.

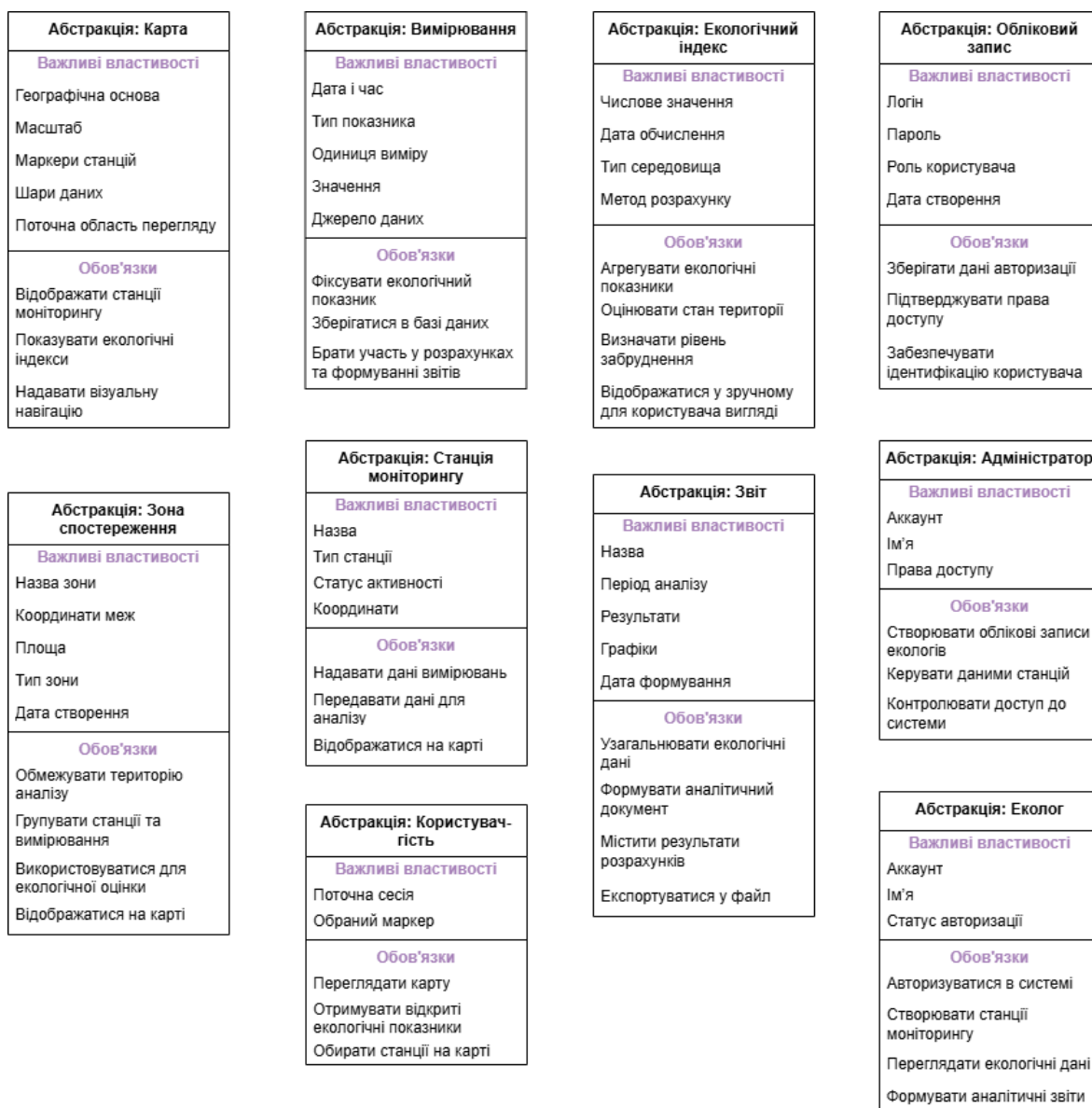


Рис. 3.1 Схема абстракцій предметної області

3.1.2 Діаграма класів. Діаграма класів формалізує виявлені абстракції у програмну структуру: кожна абстракція перетворюється на клас з визначеним набором атрибутів та методів, а смислові зв'язки між абстракціями конкретизуються у відносини асоціації, агрегації, залежності або узагальнення. Діаграму класів із асоціаціями наведено на рис. 3.2.

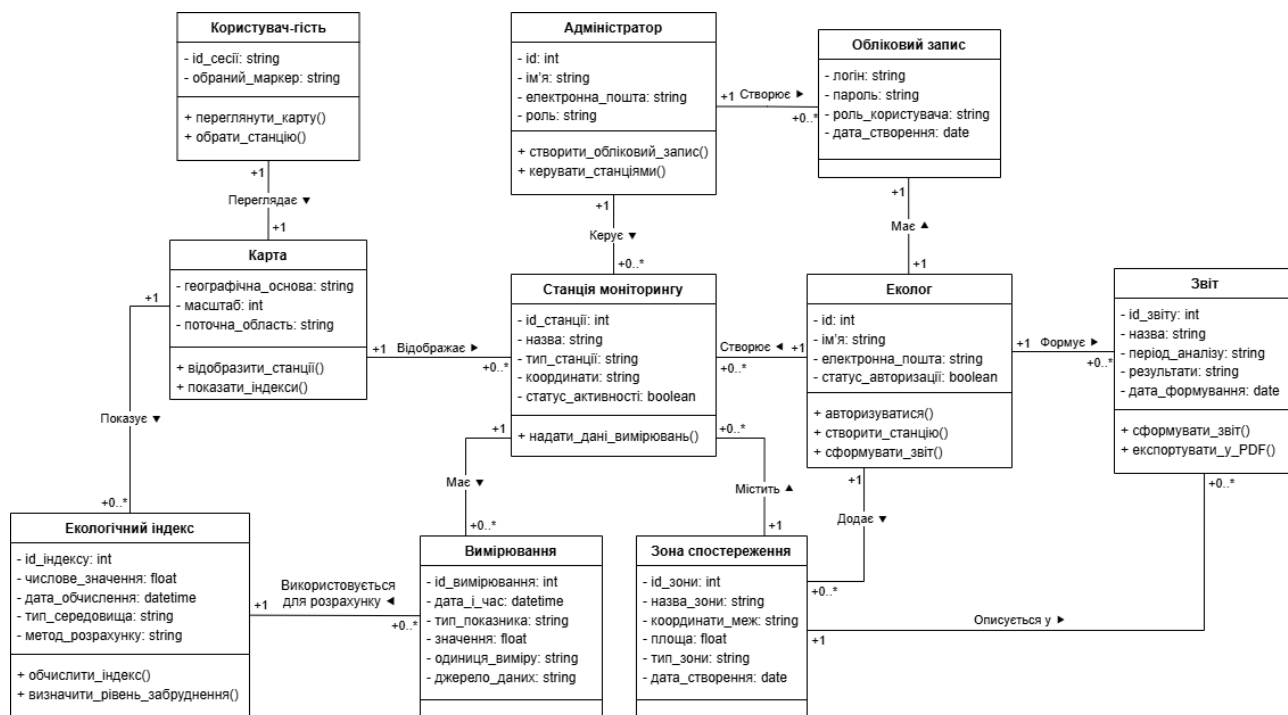


Рис. 3.2 Діаграма класів системи екологічного моніторингу з асоціаціями

Клас “Адміністратор” пов’язаний із класом “Обліковий запис” відношенням асоціації “Створює” (1 до 0..*): один адміністратор може зареєструвати довільну кількість облікових записів. Зв’язок “Адміністратор – Станція моніторингу” типу “Керує” (1 до 0..*) відображає право адміністратора редагувати та видаляти будь-які станції. Клас “Еколог” пов’язаний із “Обліковим записом” відношенням “Має” з кардинальністю 1 до 1 – кожен еколог має рівно один акаунт. Крім того, еколог може реєструвати нові станції (асоціація “Створює”, 1 до 0..*), формувати звіти (“Формує”, 1 до 0..*) та додавати зони спостереження.

Клас “Карта” пов’язаний зі “Станцією моніторингу” відношенням “Відображає” (1 до 0..*) і показує “Екологічний індекс” через залежність. “Користувач-гість” пов’язаний із “Картою” асоціацією “Переглядає” (1 до 1). Між “Станцією моніторингу” та “Вимірюванням” визначено агрегацію “Має” (1 до 0..*): вимірювання належать станції, але можуть зберігатися і після її видалення (якщо це не каскадне видалення). “Вимірювання” пов’язане з “Екологічним індексом” залежністю “Використовується для розрахунку”, а

“Зона спостереження” охоплює вимірювання зі своєї географічної ділянки та пов'язана зі “Звітом” через асоціацію “Описується у”.

3.1.3 Діаграми кооперації. Діаграма кооперації – це різновид UML-діаграми класів, що зосереджується на взаємодії обмеженої групи об'єктів у межах конкретного сценарію. На відміну від повної діаграми класів, кооперація демонструє лише ті класи й відносини, які безпосередньо задіяні у виконанні певної функції, не перевантажуючи читача зайвими зв'язками. Для розроблюваної системи побудовано п'ять діаграм кооперації, що охоплюють введення даних, обчислення індексів, формування звітів та рольову ієрархію.

Перша кооперація “Введення екологічних даних” (рис. 3.3) об'єднує три класи: “Еколог”, “Станція моніторингу” та “Вимірювання”. Клас “Еколог” несе методи створити_станцію() та ввести_вимірювання() і пов'язаний зі “Станцією моніторингу” відношенням залежності (пунктирна стрілка): еколог тимчасово використовує станцію для внесення даних, не набуваючи права власності на неї. “Станція моніторингу” надає метод надати_дані_вимірювань() і пов'язана з класом “Вимірювання” агрегацією (порожній ромб з боку станції): вимірювання є частиною станції, але зберігаються самостійно після її видалення. Клас “Вимірювання” містить атрибути дата_і_час, тип_показника, значення та джерело_даних, що фіксують кожен одиничний замір середовища.

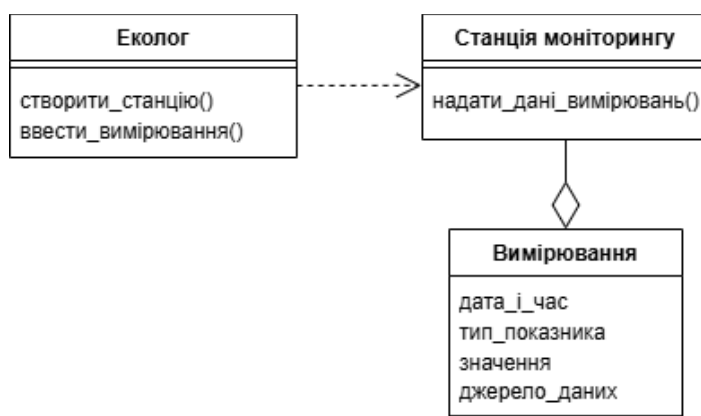


Рис. 3.3 Діаграма кооперації “Введення екологічних даних”

Друга кооперація “Обчислення екологічного індексу” (рис. 3.4) розкриває механізм агрегування вимірювань у розрахований показник. Центральний клас “Екологічний індекс” несе атрибути `числове_значення`, `тип_середовища` і `метод_розрахунку`, а також методи `обчислити_індекс()` та `визначити_рівень_забруднення()`. Він пов'язаний із класом “Вимірювання” (атрибути `дата_і_час`, `тип_показника`, `значення`, `одиниця_виміру`) агрегацією: індекс агрегує набір вимірювань, що залишаються незалежними об'єктами і можуть входити до кількох розрахунків. Клас “Екологічна метрика” – з атрибутами `назва_метрики`, `ваговий_коефіцієнт` та `граничне_значення` – пов'язаний із “Екологічним індексом” узагальненням (пунктирна лінія з порожнім трикутником): метрика є спеціалізацією індексу й успадковує його структуру, конкретизуючи параметри зважування для кожного виду забруднювача.

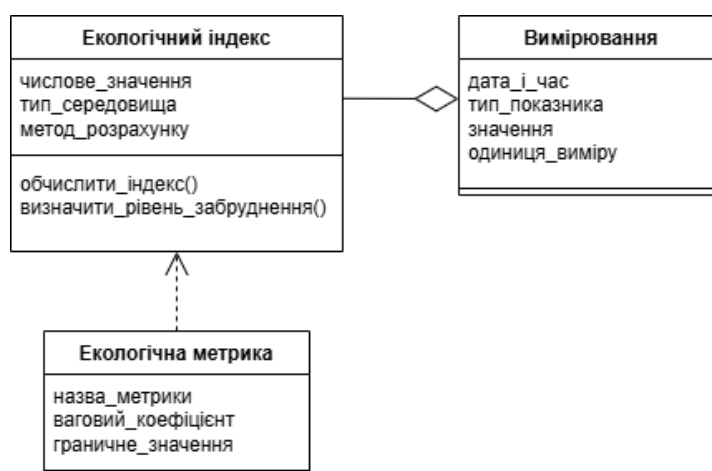


Рис. 3.4 Діаграма кооперації “Обчислення екологічного індексу”

Третя кооперація “Формування звіту” (рис. 3.5) включає класи “Еколог”, “Звіт” та “Екологічний індекс”. “Звіт” містить атрибути `назва`, `період_аналізу` і `дата_формування` та метод `експортувати_у_PDF()`. Між “Звітом” та “Екологічним індексом” (атрибути `числове_значення`, `дата_обчислення`, `тип_середовища`; метод `визначити_рівень_забруднення()`) встановлено композицію (заповнений ромб з боку звіту): результати обчислень є невід’ємною складовою документа – без них звіт не має сенсу, і при видаленні звіту пов’язані

дані вилучаються разом з ним. Клас “Еколог” з методом `сформувати_звіт()` пов'язаний зі “Звітом” реалізацією (пунктирна лінія з порожнім трикутником): еколог зобов'язаний надати конкретну реалізацію операції формування документа.



Рис. 3.5 Діаграма кооперації “Формування звіту”

Четверта кооперація “Формування звіту через інтерфейс” (рис. 3.6) демонструє поліморфний підхід до тієї самої операції. У центрі – інтерфейс “ІФормуванняЗвіту”, що оголошує єдиний контракт: метод `сформувати_звіт()`. Клас “Еколог” (методи `сформувати_звіт()` та `експортувати_у_PDF()`) пов'язаний з інтерфейсом залежністю (пунктирна стрілка): еколог звертається до інтерфейсу під час виконання, не зберігаючи постійного зв'язку. Клас “Звіт” (атрибути `назва`, `результати`, `дата_формування`) пов'язаний з “ІФормуванняЗвіту” реалізацією: він конкретно виконує контракт інтерфейсу, забезпечуючи фактичне наповнення документа. Така конструкція дозволяє у майбутньому підключити альтернативну реалізацію без зміни коду, що залежить від інтерфейсу.

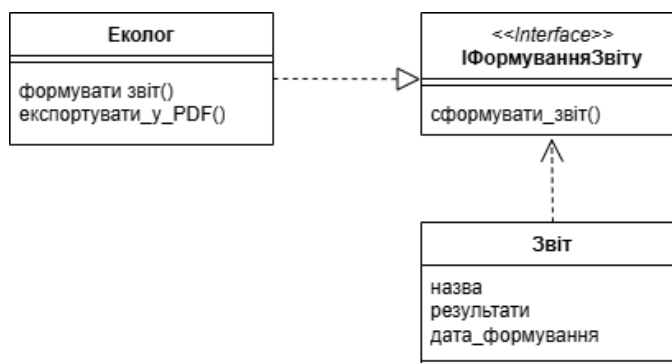


Рис. 3.6 Діаграма кооперації “Формування звіту” через інтерфейс

П'ята кооперація “Ієрархія ролей” (рис. 3.7) відображає структуру прав доступу через механізм узагальнення (наслідування). Базовий клас “Користувач” задає спільний контракт двома методами – переглянути_результати() та переглянути_карту() – доступними для всіх ролей. Від нього успадковуються три спеціалізованих класи. “Еколог” розширює базову поведінку атрибутом статус_авторизації і методами ввести_дані() та сформувати_звіт(), що забезпечують введення вимірювань і побудову аналітичних документів. “Адміністратор” набуває атрибут права_доступу і методи керувати_метриками() та створити_обліковий_запис() – для конфігурування системи та реєстрації нових учасників. “Користувач-гість” має лише атрибут id_сесії та метод обрати_станцію(), що відображає обмежені можливості неавтентифікованого відвідувача. Усі три підкласи пов'язані з “Користувачем” відношенням узагальнення (суцільна лінія з порожнім трикутником).



Рис. 3.7 Діаграма кооперації “Ієрархія ролей”

3.2 Структура пакетів та компонентів

Для відображення логічного поділу системи на функціональні блоки та фізичного розміщення компонентів у середовищі виконання розроблено дві діаграми: пакетів і компонентів. Перша описує архітектурні модулі та залежності між ними на рівні вихідного коду, друга – виконувані артефакти, бібліотеки, сервіси та зв'язки між ними під час роботи розгорнутої системи.

3.2.1 Діаграма пакетів. Діаграма пакетів системи (рис. 3.8) розділяє програмний продукт на два верхньорівневі пакети: “Client” і “Server”, – що відповідає архітектурному поділу на клієнтську і серверну частини.

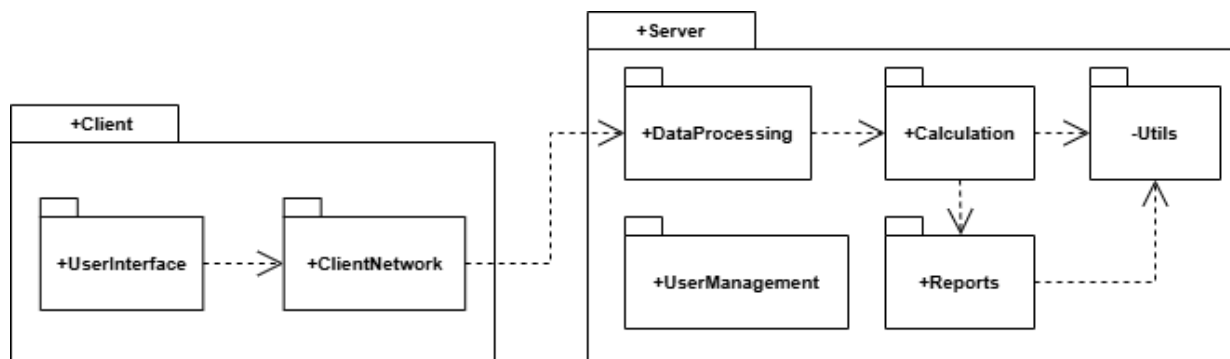


Рис. 3.8 Діаграма пакетів системи екологічного моніторингу

Пакет “Client” містить два підпакети. “+UserInterface” охоплює всі React/Next.js-компоненти, сторінки маршрутизатора та UI-елементи, що відповідають за побудову інтерфейсу користувача. “+ClientNetwork” реалізує рівень мережових запитів: сервіс `dataService.ts` із функціями-обгортками над `fetch()` та логіку обробки відповідей API. “+UserInterface” залежить від “+ClientNetwork” – сторінки викликають мережові функції для отримання та відправки даних.

Пакет “Server” складається з п'яти підпакетів. “+DataProcessing” відповідає за прийом і збереження вхідних даних: отримання вимірювань від зовнішнього API Open-Meteo та обробка запитів на додавання водних і ґрунтових даних. “+Calculation” реалізує обчислювальне ядро: алгоритми розрахунку AQI, WQI, SQI та узгодженого індексу; цей пакет залежить від “+DataProcessing” (бере звідти дані) та від “-Utils” (допоміжні функції). Пакет “-Utils” є закритим і містить загальні утиліти – парсери, математичні функції та допоміжні типи; він не залежить від жодного іншого пакета сервера. “+UserManagement” реалізує автентифікацію, управління обліковими записами та рольовий контроль доступу. “+Reports” відповідає за генерацію аналітичних документів і залежить від “+Calculation” (потребує розрахованих індексів) та “-Utils”.

3.2.2 Діаграма компонентів. Діаграма компонентів (рис. 3.9) відображає архітектуру системи у вигляді п'яти функціональних зон: “Client Side”, “Server Infrastructure”, “Application Logic”, “External Services” та “Database Tier”.

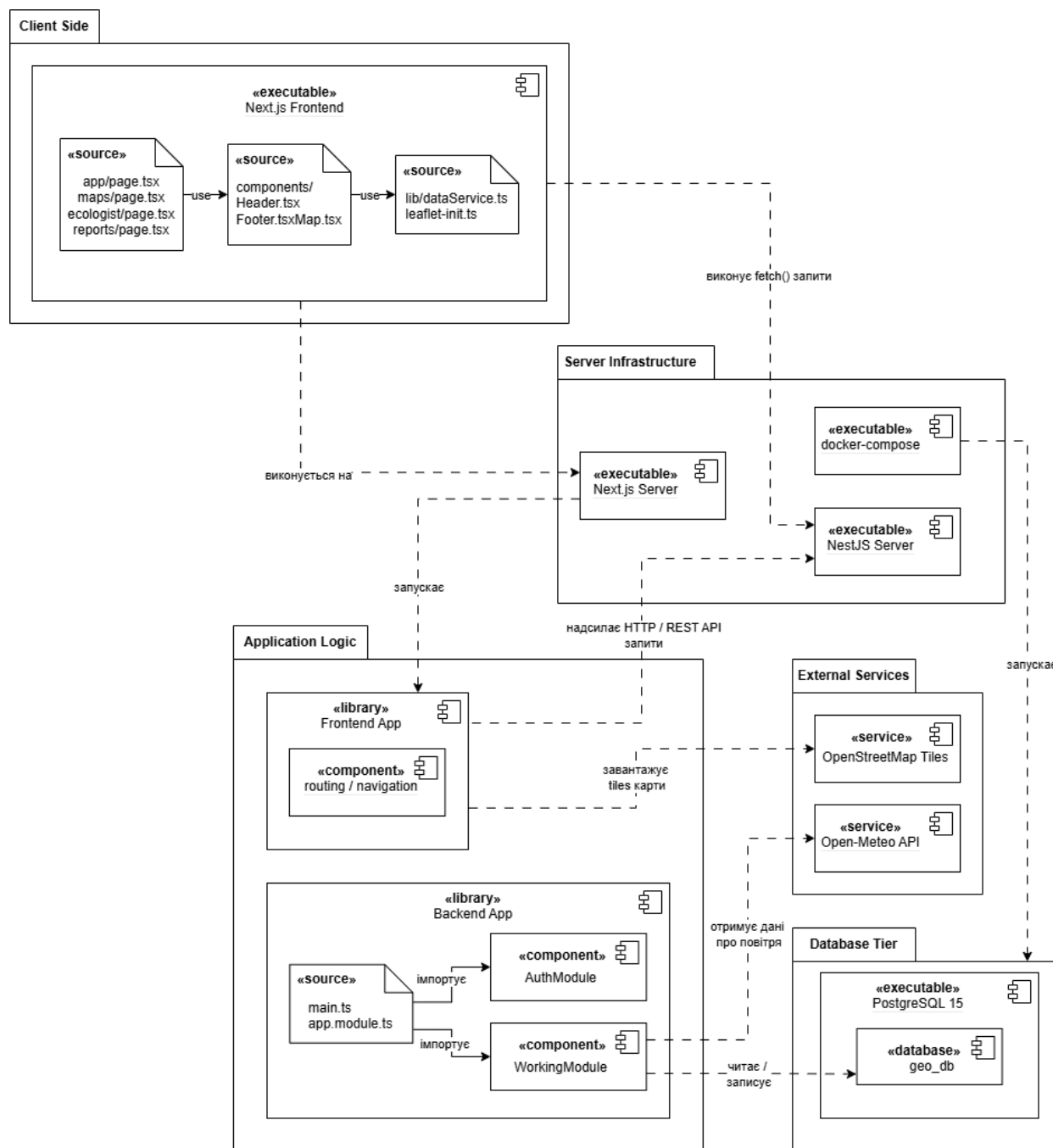


Рис. 3.9 Діаграма компонентів системи екологічного моніторингу

Зона “Client Side” містить виконуваний артефакт “Next.js Frontend” та три групи файлів-джерел. Перша – сторінки маршрутизатора: app/page.tsx, maps/page.tsx, ecologist/page.tsx, reports/page.tsx. Вони залежать від другої групи

– спільних компонентів `components/Header.tsx`, `Footer.tsx`, `Map.tsx` – за зв'язком “use”. Компоненти, у свою чергу, залежать від третьої групи – утиліт `lib/dataService.ts` та `lib/leaflet-init.ts`, які здійснюють `fetch()`-запити до бекенду та ініціалізують бібліотеку `Leaflet.js` для відображення тайлової карти.

Зона “Server Infrastructure” визначає виконване середовище: `docker-compose` запускає два сервери – “Next.js Server” (рендеринг клієнтської частини) і “NestJS Server” (REST API). Клієнтська частина виконується на “Next.js Server”, а `fetch()`-запити прямують до “NestJS Server” по внутрішній мережі Docker. Зона “Application Logic” поділяється на дві бібліотеки. “Frontend App” містить компонент `routing/navigation`, що забезпечує клієнтську маршрутизацію Next.js. “Backend App” включає стартові файли `main.ts` та `app.module.ts`, а також два компоненти: “AuthModule” для автентифікації та “WorkingModule”, що консолідує всі функціональні NestJS-модулі системи.

Зона “External Services” містить два зовнішніх сервіси. “OpenStreetMap Tiles” постачає тайли карти – до нього звертається компонент `Map.tsx` у Frontend App. “Open-Meteo API” надає дані про якість повітря і метеопокказники – його споживає “WorkingModule” у межах щогодинного планового оновлення станцій. “Database Tier” представлено виконуваним компонентом “PostgreSQL 15”, що керує базою даних “geo_db” (`geo_monitoring`). “WorkingModule” читає та записує дані до “geo_db” через TypeORM, а `docker-compose` запускає PostgreSQL у контейнері з розширенням PostGIS.

3.3 Програмний інструментарій та бібліотеки

Технологічний стек системи екологічного моніторингу вибирався з урахуванням таких критеріїв: підтримка TypeScript як єдиної мови для фронтенду та бекенду, наявність зрілої екосистеми модулів, сумісність із PostgreSQL та PostGIS, а також можливість контейнеризованого розгортання. Стек охоплює серверний фреймворк, ORM, СУБД, клієнтський фреймворк,

картографічну бібліотеку, інфраструктурні інструменти та ряд спеціалізованих бібліотек.

Серверна частина побудована на основі NestJS [12] – фреймворку для Node.js, що реалізує модульну архітектуру на базі декораторів TypeScript і принципу впровадження залежностей (Dependency Injection). Кожен функціональний блок системи (автентифікація, станції, вода, ґрунт, зони) оформлено у вигляді окремого NestJS-модуля (`@Module`) з власним контролером (`@Controller`) та сервісом (`@Injectable`). Планування щогодинного оновлення станцій реалізовано за допомогою вбудованого пакета `@nestjs/schedule` з декоратором `@Cron`.

Взаємодія з базою даних організована через TypeORM [18] – об'єктно-реляційний маппер для TypeScript, що дозволяє описувати структуру таблиць безпосередньо у класах-сутностях за допомогою декораторів `@Entity`, `@Column`, `@PrimaryGeneratedColumn`, `@ManyToOne` тощо. У режимі `synchronize:true` TypeORM автоматично застосовує схему до бази даних при кожному старті сервера. Для виконання складних аналітичних запитів використовується QueryBuilder із підтримкою конструкцій LATERAL JOIN.

Як СУБД обрано PostgreSQL 15 з розширенням PostGIS 3.4, що розгортається у Docker-контейнері на базі образу `postgis/postgis:15-3.4`. PostgreSQL забезпечує підтримку ACID-транзакцій, типу JSONB для зберігання масивів координат зон, складених індексів на парах (`stationId`, `measuredAt`) та повний набір просторових функцій PostGIS для майбутнього розширення геопросторового функціоналу. База даних отримала назву `geo_monitoring`.

Клієнтська частина розроблена на Next.js 14 [13] з використанням App Router – файлової системи маршрутизації, де кожен підкаталог `app/` відповідає маршруту і містить файл `page.tsx` з React-компонентом сторінки. Next.js забезпечує серверний рендеринг (SSR) і статичну генерацію (SSG) за потреби, а також оптимізацію зображень та автоматичне розбиття бандлу. Весь клієнтський

код написано на TypeScript, що дозволяє перевіряти відповідність типів між API-відповідями та React-компонентами.

Відображення інтерактивної карти реалізовано за допомогою бібліотеки Leaflet.js [16] у поєднанні з React-обгорткою react-leaflet. Тайловий підклад завантажується з OpenStreetMap [17]. Для уникнення помилки “document is not defined” при серверному рендерингу Next.js бібліотеку Leaflet підключено динамічно через lib/leaflet-init.ts з директивою dynamic import і параметром ssr:false.

Автентифікація реалізована за схемою JWT (JSON Web Token): при успішному вході сервер генерує підписаний токен через пакет @nestjs/jwt; наступні запити перевіряються декоратором @UseGuards(JwtAuthGuard). Паролі зберігаються виключно у вигляді bcrypt-хешів (cost factor 10) завдяки бібліотеці bcrypt. Зовнішні запити до Open-Meteo Air Quality API [6] та Weather API виконуються через вбудований HttpModule NestJS; для зменшення навантаження на API реалізовано пакетну обробку по 5 станцій із затримкою 1500 мс між пакетами.

Для генерації аналітичних звітів у форматі Word використовується JavaScript-бібліотека docx, що дозволяє програмно формувати документи з таблицями, графіками та форматованим текстом без залежності від встановленого офісного пакету. Конвертація SVG-діаграм у растрові зображення для вбудовування у звіт здійснюється засобами Python-бібліотеки cairosvg. Вся інфраструктура розгортання описана у файлі docker-compose.yml і включає чотири контейнери: api, web, db та proxy (Cloudflare Tunnel).

3.4 Алгоритми обчислення екологічних індексів

Для кожного з трьох компонентів середовища – повітря, вода, ґрунт – у системі реалізовано окремий алгоритм обчислення індексу якості. Алгоритми

відрізняються природою вхідних даних та методом агрегування часткових оцінок, але всі повертають числовий результат, що далі нормується до єдиної шкали для обчислення узгодженого індексу.

3.4.1 Індекс якості повітря (AQI). Обчислення індексу якості повітря реалізовано відповідно до методики Європейського агентства з охорони довкілля (ЕЕА). Алгоритм включає три послідовні кроки: отримання вимірюваних концентрацій PM2.5, PM10, NO₂, O₃ та SO₂ з зовнішнього API; обчислення часткового індексу для кожного забруднювача методом кусково-лінійної інтерполяції за граничними точками; вибір максимального з часткових значень як підсумкового AQI станції.

Кусково-лінійна інтерполяція реалізована функцією `piecewise(value, pts)`, де `pts` – масив відрізків виду `[cL, cH, iL, iH]` (нижня та верхня межі концентрації, нижня та верхня межі індексу). Для кожного забруднювача знаходиться відрізок, до якого потрапляє вимірювана концентрація, після чого виконується лінійна інтерполяція:

$$AQI_{part} = iL + (value - cL) / (cH - cL) \times (iH - iL)$$

Граничні точки для PM2.5 визначаються чотирма відрізками: `[0;10]→[0;25]`, `[10;20]→[25;50]`, `[20;25]→[50;75]`, `[25;50]→[75;100]`. Аналогічно задано відрізки для PM10 (`[0;20]`, `[20;40]`, `[40;50]`, `[50;100]` мкг/м³), NO₂ (`[0;40]`, `[40;90]`, `[90;120]`, `[120;230]` мкг/м³), O₃ (`[0;50]`, `[50;100]`, `[100;130]`, `[130;240]` мкг/м³) та SO₂ (`[0;100]`, `[100;200]`, `[200;350]`, `[350;500]` мкг/м³). Підсумковий AQI станції обчислюється як максимум часткових значень: $AQI = \max(AQI_PM2.5, AQI_PM10, AQI_NO_2, AQI_O_3, AQI_SO_2)$.

Шкала інтерпретації AQI: 0–25 – гарна якість повітря; 26–50 – добра; 51–75 – помірна; 76–100 – погана; понад 100 – дуже погана якість (надзвичайно шкідливий стан атмосфери).

3.4.2 Індекс якості води (WQI). Індекс якості води обчислюється як максимальне кратне гранично допустимої концентрації (ГДК) серед усіх вимірних параметрів. Така підхід дозволяє виявити найнебезпечніше перевищення навіть за умови, що більшість параметрів знаходяться в нормі. Формула WQI для n параметрів має вигляд:

$$WQI = \max\{ratio_i\}, \quad i = 1..n$$

де $ratio_i = c_i / ГДК_i$ для більшості показників (амоній, БСК5, хлориди, нітрати, нітроти, сульфати, завислі речовини). Для розчиненого кисню застосовується зворотна формула: $ratio = ГДК / c$, якщо $c < ГДК$ (нестача кисню відповідає забрудненню). Параметри та їхні ГДК: амоній – 0.5 мг/л, БСК5 – 3 мг/л, хлориди – 350 мг/л, розчинений кисень – 4 мг/л, нітрати – 45 мг/л, нітроти – 3.3 мг/л, сульфати – 500 мг/л, завислі речовини – 25 мг/л.

Значення WQI інтерпретується за п'ятиступінчастою шкалою: 0–3 – допустима якість; 3–4 – помірно забруднена; 4–5 – забруднена; 5–6 – сильно забруднена; понад 6 – критично забруднена вода.

3.4.3 Індекс деградації ґрунту (SDI). Для оцінки стану ґрунту реалізовано індекс деградації SDI (Soil Degradation Index), значення якого зростає зі збільшенням деградованості: 0 відповідає ідеальному стану, 100 – повній деградації. SDI обчислюється як зважена сума п'яти компонентів деградації:

$$SDI = 0.25 \times D_{pH} + 0.20 \times D_{EC} + 0.20 \times D_{OM} + 0.25 \times D_{macro} + 0.10 \times D_{micro}$$

Компонент D_{pH} обчислюється як $\text{clamp}(|pH - 7| \times 20, 0, 100)$: відхилення pH від нейтрального значення 7 масштабується до шкали 0–100. Компонент електропровідності $D_{EC} = \text{clamp}(EC \times 25, 0, 100)$ відображає засоленість ґрунту. Деградація органічної речовини $D_{OM} = \text{clamp}((5 - OM) \times 20, 0, 100)$, де оптимальним вважається вміст 5%. Для макроелементів (NO_3-N , P, K, Ca, Mg, SO_4-S) та мікроелементів (Zn, Mn, Cu, Fe, Co, B, Mo) деградація кожного

елемента визначається як $\text{clamp}((\text{оптимум} - \text{факт}) / \text{оптимум} \times 100, 0, 100)$, а D_{macro} та D_{micro} – середні значення по групах елементів.

Шкала інтерпретації SDI: 0–15 – високоякісний ґрунт; 15–30 – добрий; 30–50 – задовільний; 50–70 – деградований; 70–90 – сильно деградований; понад 90 – критичний стан ґрунту.

3.5 Реалізація узгодженого індексу за методом Сааті

Узгоджений екологічний індекс E є інтегральним показником стану навколишнього середовища зони спостереження, що об'єднує оцінки повітря, води та ґрунту з урахуванням відносної важливості кожного компонента. Для визначення вагових коефіцієнтів застосовано метод аналізу ієрархій (Analytic Hierarchy Process, АНР) Т. Сааті [1].

3.5.1 Визначення вагових коефіцієнтів методом АНР. Метод Сааті передбачає побудову матриці парних порівнянь A розмірності 3×3 , елемент a_{ij} якої показує, у скільки разів компонент i важливіший за компонент j за дев'ятибальною шкалою. Для трьох критеріїв – повітря (air), вода (water), ґрунт (soil) – на підставі експертного аналізу сформовано матрицю:

$$A = \begin{bmatrix} 1 & 5/3 & 5/2 \\ 3/5 & 1 & 3/2 \\ 2/5 & 2/3 & 1 \end{bmatrix}$$

Власний вектор матриці A нормується так, щоб сума компонент дорівнювала одиниці. Отримані значення є ваговими коефіцієнтами: $w_{\text{air}} = 0.50$, $w_{\text{water}} = 0.30$, $w_{\text{soil}} = 0.20$. Консистентність матриці перевіряється через індекс узгодженості $CI = (\lambda_{\text{max}} - n) / (n - 1)$ та коефіцієнт $CR = CI / RI$, де $RI = 0.58$ для $n = 3$. Отримане значення $CR < 0.1$ підтверджує узгодженість експертних суджень. Обчислені ваги зафіксовано в реалізації константою $W = \{ \text{air: } 0.50, \text{water: } 0.30, \text{soil: } 0.20 \}$.

3.5.2 Нормування часткових індексів. Перед обчисленням E часткові індекси різної природи приводяться до єдиної шкали від 0 до 100, де 100

відповідає найкращій якості середовища. Для індексу якості повітря: $I_{air} = \max(0, \min(100, 100 - AQI))$, тобто $AQI = 0$ дає $I_{air} = 100$, $AQI = 100$ дає $I_{air} = 0$. Для індексу якості води: $I_{water} = \max(0, \min(100, 100 - WQI \times 20))$, де множник 20 масштабує шкалу WQI (0–5+) до шкали 0–100. Для індексу деградації ґрунту: $I_{soil} = \max(0, \min(100, 100 - SDI))$, оскільки SDI вже має шкалу 0–100 зростаючої деградації – інверсія переводить його у шкалу якості.

3.5.3 Формула узгодженого індексу. Узгоджений екологічний індекс для зони спостереження обчислюється як зважена сума наявних нормованих компонентів:

$$E = (w_{air} \times I_{air} + w_{water} \times I_{water} + w_{soil} \times I_{soil}) / \sum w_k$$

де сума $\sum w_k$ включає лише ваги тих компонентів, для яких є фактичні дані вимірювань у зоні. Ця нормалізація дозволяє коректно обчислювати індекс навіть за часткової відсутності даних – наприклад, якщо у зоні відсутні водні або ґрунтові станції, система використовує лише наявні компоненти, перерозподіляючи їхні ваги пропорційно. Значення E на шкалі 0–100 інтерпретується аналогічно нормованим компонентам: чим вище значення, тим кращий загальний стан середовища зони спостереження.

Для зони спостереження середні значення AQI, WQI та SDI обчислюються як арифметичні середні по всіх станціях відповідного типу, розташованих у межах геопросторового контуру зони. Перевірка належності станцій до зони здійснюється запитом до бази даних з використанням координат центроїду або LATERAL JOIN за координатами станцій.

У таблиці 3.4 наведено узагальнену характеристику трьох часткових індексів та їх роль у формуванні узгодженого показника.

Таблиця 3.4

Характеристика компонентів узгодженого екологічного індексу

Компонент	Індекс	Формула / метод	Нормування	Вага
Повітря	AQI	$\max(\text{piecewise})$ по PM2.5, PM10, NO ₂ , O ₃ , SO ₂	$I_{\text{air}} = 100 - \text{AQI}$	0.50
Вода	WQI	$\max(c_i / \text{ГДК}_i)$ по 8 параметрам	$I_{\text{water}} = 100 - \text{WQI} \times 20$	0.30
Ґрунт	SDI	Зважена сума 5 компонентів деградації	$I_{\text{soil}} = 100 - \text{SDI}$	0.20

4 ТЕСТУВАННЯ ТА РОЗГОРТАННЯ СИСТЕМИ

4.1 Системні вимоги та інструкція з інсталяції

Для коректного функціонування системи екологічного моніторингу необхідно дотримуватися певних вимог до апаратного та програмного забезпечення. Правильна організація технічної бази дозволяє системі ефективно обробляти запити користувачів, забезпечувати щогодинне автоматичне оновлення даних від зовнішніх API та надавати зручний інтерфейс для взаємодії з екологічною інформацією в режимі реального часу.

Відповідно до діаграми розгортання системи, наведеної на рис. 4.1, програмні компоненти розміщено на окремих логічних вузлах, що відображає фізичну структуру реалізації.

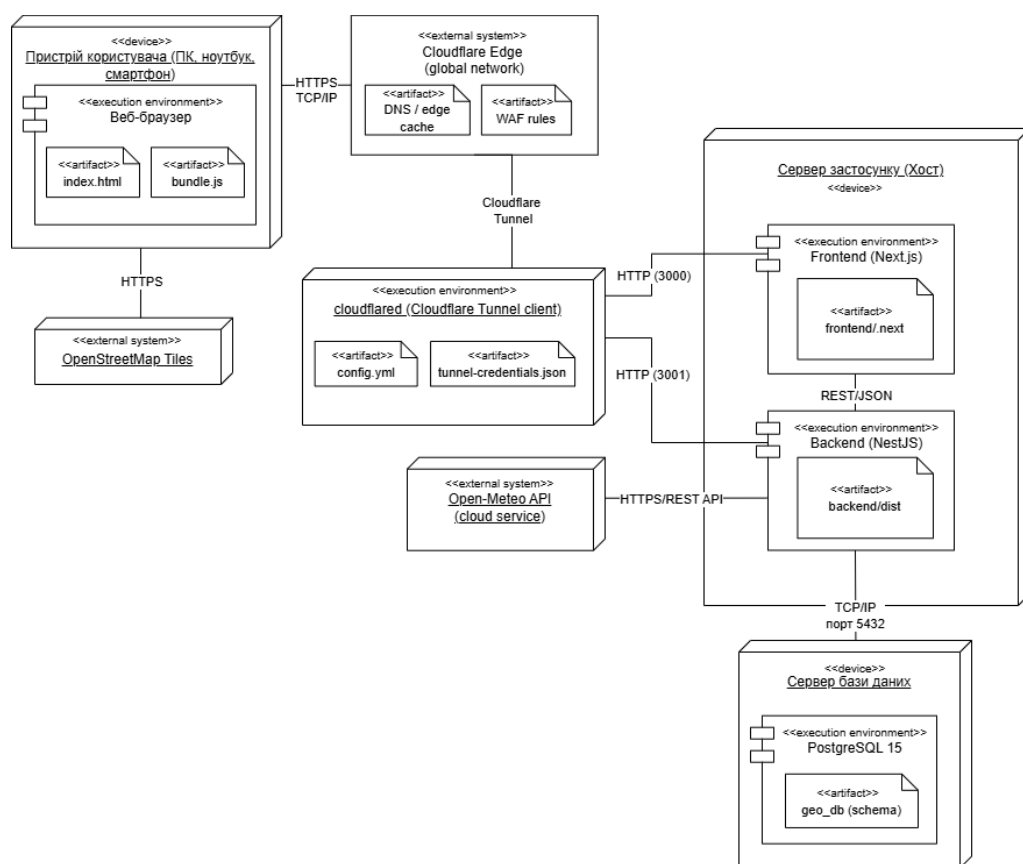


Рис. 4.1 Діаграма розгортання системи екологічного моніторингу

Система реалізована за архітектурою клієнт-сервер з трирівневим розподілом компонентів. Клієнтська частина представлена веб-браузером на пристрої кінцевого користувача, який завантажує з сервера Next.js-застосунок (index.html, bundle.js). Серверна частина розгорнута на хост-машині та складається з двох незалежних процесів: Frontend на порту 3000 (Next.js) та Backend на порту 3001 (NestJS). База даних PostgreSQL 15 розгорнута у Docker-контейнері на окремому вузлі і доступна для Backend по TCP/IP на порту 5432.

Публічний доступ до системи забезпечується через Cloudflare Tunnel [20]: клієнт звертається до домену lupa.living по HTTPS, запит проходить через Cloudflare Edge (DNS, кешування, WAF), далі по захищеному тунелю надходить до демона cloudflared на сервері, який маршрутизує його до Frontend (порт 3000) або Backend (порт 3001). Зовнішній сервіс Open-Meteo API (хмарний сервіс) опитується безпосередньо Backend по HTTPS щогодини для отримання даних якості повітря та метеопоказників.

Нижче наведено основні вимоги до апаратного та програмного забезпечення системи екологічного моніторингу.

Апаратне забезпечення

Сервер застосунку (хост-машина):

- процесор з тактовою частотою не менше 2,4 ГГц і кількістю ядер не менше двох;
- оперативна пам'ять – не менше 4 ГБ (рекомендовано 8 ГБ);
- вільний обсяг дискового простору – не менше 20 ГБ;
- мережеве з'єднання зі швидкістю не менше 100 Мбіт/с.

Клієнтська машина (пристрій користувача):

- будь-який персональний комп'ютер, ноутбук або смартфон з доступом до мережі Інтернет;
- мережеве підключення зі швидкістю не менше 5 Мбіт/с для коректного завантаження тайлів карти.

Програмне забезпечення

Сервер застосунку:

- операційна система Linux (Ubuntu 22.04 LTS або Debian 12) або Windows 10/11 з підсистемою WSL 2;
- Docker Engine версії 24 або новіше та Docker Compose v2 [19] – для запуску контейнера бази даних;
- Node.js версії 20 LTS – для виконання серверних процесів NestJS та Next.js;
- Git версії 2.x – для клонування репозиторію та оновлення вихідного коду.

Клієнтська машина:

- сучасний веб-браузер: Chrome 120+, Edge 120+, Firefox 120+ або Safari 17+;
- увімкнений JavaScript у налаштуваннях браузера;
- доступ до мережі Інтернет для завантаження тайлів OpenStreetMap та даних системи.

Дотримання наведених вимог забезпечить стабільну та ефективну роботу системи, зручність для користувачів і надійне підключення до всіх зовнішніх сервісів.

4.2 Склад інсталяційного пакету

Інсталяційний пакет програмного забезпечення розроблено для зручного розгортання серверної частини системи екологічного моніторингу. Пакет містить усі необхідні компоненти для повноцінної роботи як серверних процесів, так і бази даних.

До складу інсталяційного пакету входять такі компоненти:

- Репозиторій вихідного коду – містить каталоги backend/ (серверна частина NestJS) та frontend/ (клієнтська частина Next.js) з усіма файлами вихідного коду, файлами залежностей package.json та конфігурацією TypeScript. Клонується командою `git clone <repo-url>`.
- Файл docker-compose.yml – описує конфігурацію Docker-контейнера бази даних на основі образу postgres/postgis:15-3.4 з базою geo_monitoring. Запускається командою `docker compose up -d`.
- Конфігураційні файли змінних середовища (.env) – містять параметри підключення до бази даних (хост, порт, ім'я БД, логін, пароль) та інші параметри середовища виконання. Заповнюються на основі шаблону .env.example перед першим запуском.
- Конфігурація Cloudflare Tunnel – файли config.yml та tunnel-credentials.json, що налаштовують захищений тунель від сервера до мережі Cloudflare і маршрутизацію трафіку до Frontend (порт 3000) та Backend (порт 3001).
- Документація – інструкція з розгортання серверної частини та опис конфігураційних параметрів системи.

Особливості інсталяції:

- схема бази даних створюється автоматично при першому запуску Backend завдяки налаштуванню TypeORM synchronize:true – окремого виконання SQL-скриптів або міграцій не потрібно;

- перший обліковий запис адміністратора створюється через маршрут POST /auth/seed, який доступний лише за відсутності будь-якого адміністратора в системі та автоматично блокується після першого успішного виклику;
- Open-Meteo Air Quality API є безкоштовним публічним сервісом – реєстрація та отримання API-ключа не потрібні;
- після розгортання Backend автоматично виконує перше завантаження даних по всіх станціях через 10 секунд після старту, подальші оновлення – щогодини.

Таким чином, інсталяційний пакет забезпечує комплексне розгортання та налаштування всіх компонентів системи, що гарантує їхню коректну взаємодію та готовність до експлуатації в робочому середовищі.

4.3 Тестування функціональності системи

Перевірку коректності розробленого програмного забезпечення проведено на двох рівнях: модульному та інтеграційному. Для обох рівнів використовується фреймворк Jest у поєднанні з модулем тестування NestJS. Запуск тестів виконується командами `npm test` (модульні) та `npm run test:e2e` (інтеграційні).

4.3.1 Модульне тестування. Модульне тестування реалізовано з використанням NestJS Testing Module, який дозволяє ізолювати окремий сервіс або контролер, замінюючи його залежності заглушками (mocks). Тестовий модуль формується у блоці `beforeEach` через `Test.createTestingModule()`, що гарантує чистий стан перед кожним тест-кейсом. Специфікації тестів розміщуються поруч із відповідними файлами вихідного коду у файлах `*.spec.ts`.

Модульними тестами охоплено такі компоненти: логіку контролера застосунку (`AppController`) – перевіряється повернення коректного рядка вітання; алгоритми обчислення часткових індексів (`computeEuropeanAQI`,

computeWQI, computeSQI) – тестуються граничні значення концентрацій та поведінка при відсутніх параметрах; сервіс автентифікації (AuthService) – перевіряється генерація та валідація JWT-токенів, хешування паролів bcryptjs, відхилення некоректних облікових даних.

4.3.2 Інтеграційне тестування (e2e). Інтеграційне тестування виконується бібліотекою supertest, яка відправляє реальні HTTP-запити до NestJS-застосунку, запущеного у тестовому середовищі. Конфігурація тестового середовища описана у файлі test/jest-e2e.json. У блоці beforeEach повний модуль AppModule компілюється та ініціалізується, а у afterEach – завершується, що запобігає витокам з'єднань між тест-кейсами.

Перелік перевірених інтеграційних сценаріїв наведено у таблиці 4.1. Кожен сценарій відповідає одному або декільком варіантам використання, виявленим під час аналізу вимог.

Таблиця 4.1

Результати інтеграційного тестування REST API

№	Маршрут / дія	Мета перевірки	Очікуваний результат	Фактичний результат
1	GET /	Доступність кореневого маршруту сервера	HTTP 200 OK	Успішно
2	POST /auth/login – коректні дані	Успішна автентифікація з поверненням JWT	HTTP 200, access_token	Успішно
3	POST /auth/login – некоректний пароль	Відхилення неправильних облікових даних	HTTP 401 Unauthorized	Успішно
4	GET /stations – без токена	Захист маршруту переліку станцій	HTTP 401 Unauthorized	Успішно
5	GET /stations – з токеном еколога	Отримання масиву станцій повітря	HTTP 200, масив JSON	Успішно

6	POST /zones – еколог, валідні дані	Створення зони спостереження	HTTP 201, об'єкт зони	Успішно
7	DELETE /zones/:id – власна зона	Видалення власної зони екологом	HTTP 200, deleted: id	Успішно
8	DELETE /zones/:id – чужа зона	Заборона видалення зони іншого користувача	HTTP 403 Forbidden	Успішно
9	POST /water-stations – екологічні дані	Збереження водного вимірювання з WQI	HTTP 201, wqi в об'єкті	Успішно
10	POST /soil-stations – ґрунтові дані	Збереження ґрунтового вимірювання з SDI	HTTP 201, sqi в об'єкті	Успішно

Усі десять тест-кейсів виконано успішно. У процесі тестування виявлено та усунуто такі дефекти: некоректна обробка null-значень у полі carbon_monoxide для окремих станцій при виклику computeEuropeanAQI; відсутність обмеження довжини поля email у DTO реєстрації; помилка округлення в алгоритмі AQI на межі кусково-лінійних відрізків при значенні концентрації рівно сН. Після виправлення перелічених дефектів усі тести повторно пройдено успішно.

4.3.3 Ручне функціональне тестування. Ручне функціональне тестування клієнтського застосунку проведено відповідно до сценаріїв, побудованих на основі діаграми варіантів використання. Перевірено такі сценарії: перегляд інтерактивної мапи зі станціями моніторингу; отримання значення AQI у конкретній точці через спливаюче вікно; створення зони спостереження (полігон) на мапі; перегляд та видалення власних зон в особистому кабінеті еколога; перегляд звітів з можливістю відбору за датою; авторизація та вихід з

системи. Всі сценарії виконано без виявлення критичних дефектів у фінальній версії.

4.4 Результати експлуатації клієнтського застосунку

Клієнтський застосунок реалізовано у вигляді односторінкового Next.js-додатка з чотирма основними розділами: головна сторінка з інтерактивною картою, особистий кабінет еколога, розділ звітів та сторінка авторизації. Нижче наведено результати перевірки ключових екранних форм.

Головна сторінка застосунку (рис. 4.2) відображає тайлову карту на базі OpenStreetMap з маркерами станцій моніторингу. Колір маркера відповідає поточному рівню AQI: зелений – гарна якість (0–25), жовтий – добра (26–50), помаранчевий – помірна (51–75), червоний – погана або дуже погана (76+). Карта підтримує масштабування, переміщення та фільтрацію шарів (повітря, вода, ґрунт).

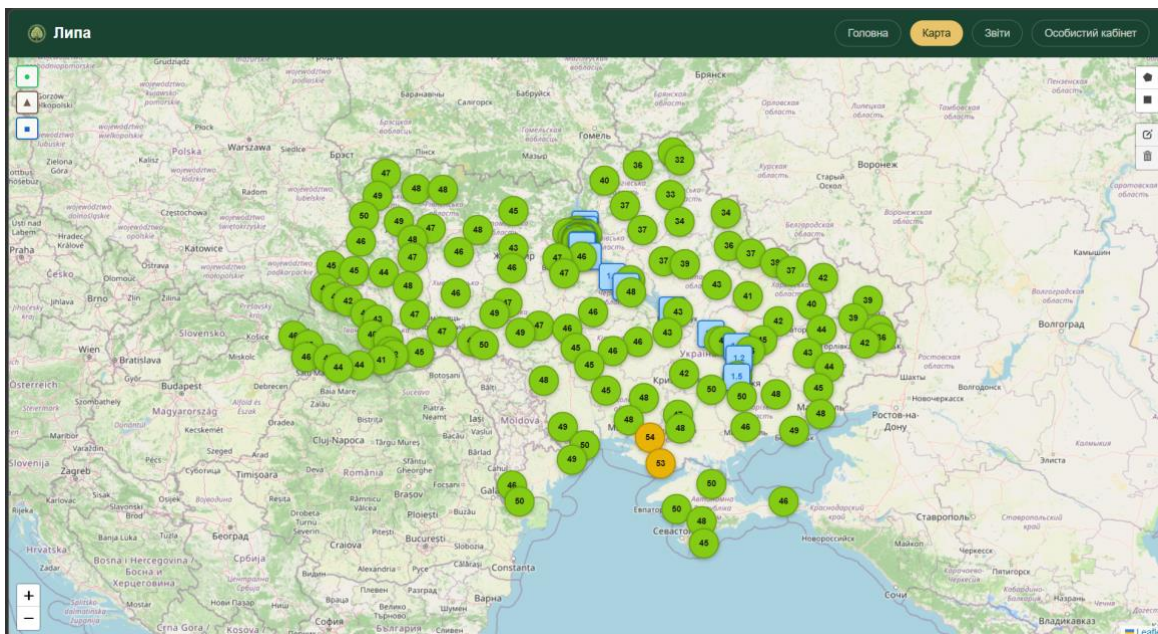


Рис. 4.2 Головна сторінка клієнтського застосунку з картою станцій

Вибір маркера станції відкриває спливаюче вікно (рис. 4.3) з детальною інформацією: значення European AQI, концентрації PM2.5, PM10, NO₂, O₃, SO₂,

метеопоказники (температура, вологість, тиск, швидкість вітру) та кольорова індикація категорії якості повітря. Дані оновлюються автоматично щогодини завдяки плановому завданню @Cron сервера.

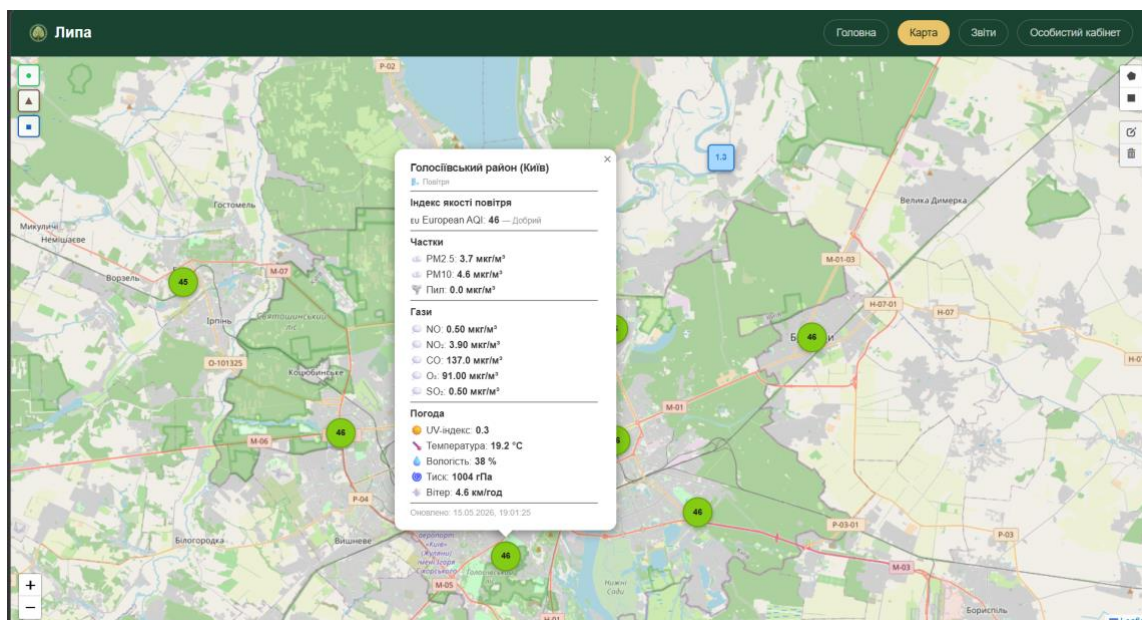


Рис. 4.3 Спливаюче вікно з показниками якості повітря

Авторизований еколог отримує доступ до інструментів малювання зон на карті (рис. 4.4). Доступні типи зон: довільний полігон, прямокутник і коло. Після завершення малювання користувач задає назву зони й зберігає її натисканням кнопки підтвердження. Збережені координати полігону передаються на сервер у форматі масиву пар [lat, lon] та зберігаються у полі JSONB таблиці zone бази даних.

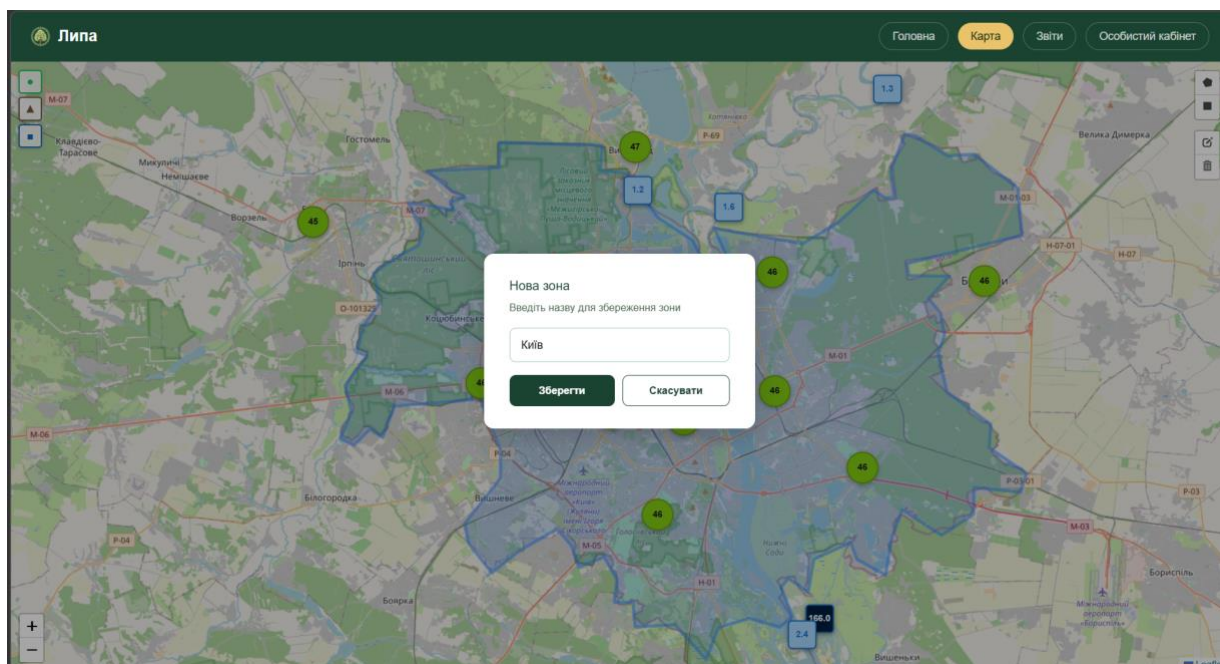


Рис. 4.4 Створення користувацької зони спостереження на карті

Після збереження зони активується кнопка обчислення узгодженого екологічного індексу. Сервер збирає середні значення AQI, WQI та SDI по всіх станціях у межах зони, нормує їх до шкали 0–100 та обчислює зважену суму з коефіцієнтами 0.50 / 0.30 / 0.20. Результат відображається у вигляді кругових діаграм для кожного компонента і загального значення E (рис. 4.5).

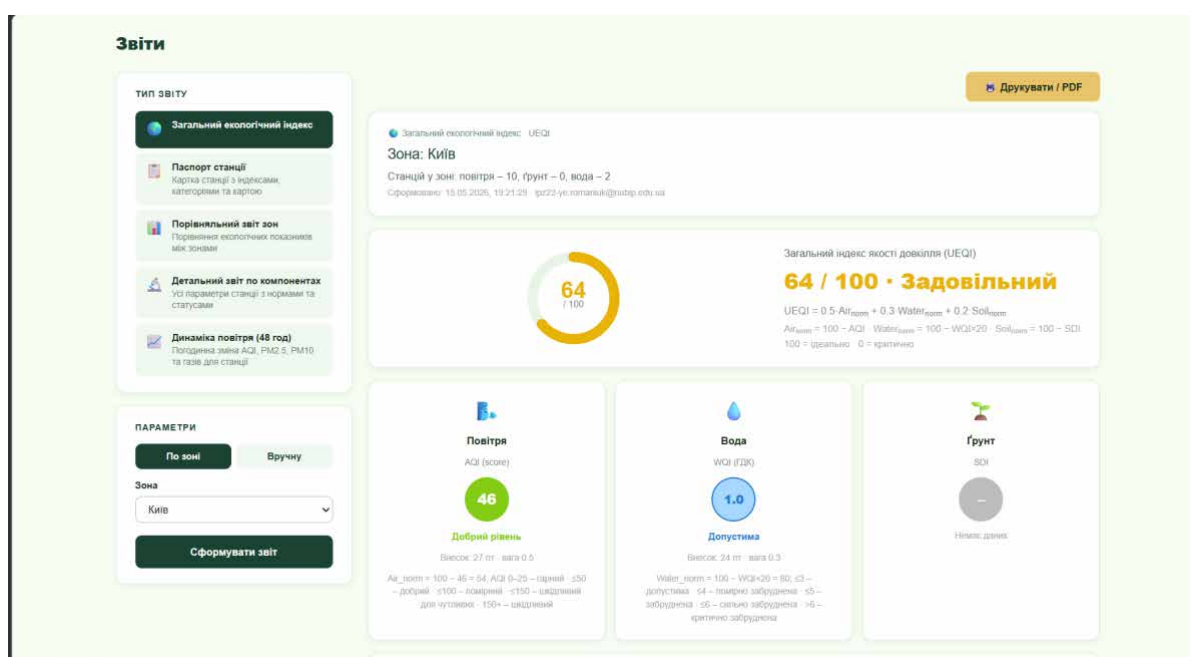


Рис. 4.5 Відображення узгодженого екологічного індексу для зони

Особистий кабінет еколога (рис. 4.6) відображає перелік власних зон, водних і ґрунтових станцій з можливістю перегляду, редагування та видалення. Адміністратор додатково бачить усі зони всіх користувачів і має доступ до управління обліковими записами: створення нових екологів, зміна пароля, видалення облікового запису.

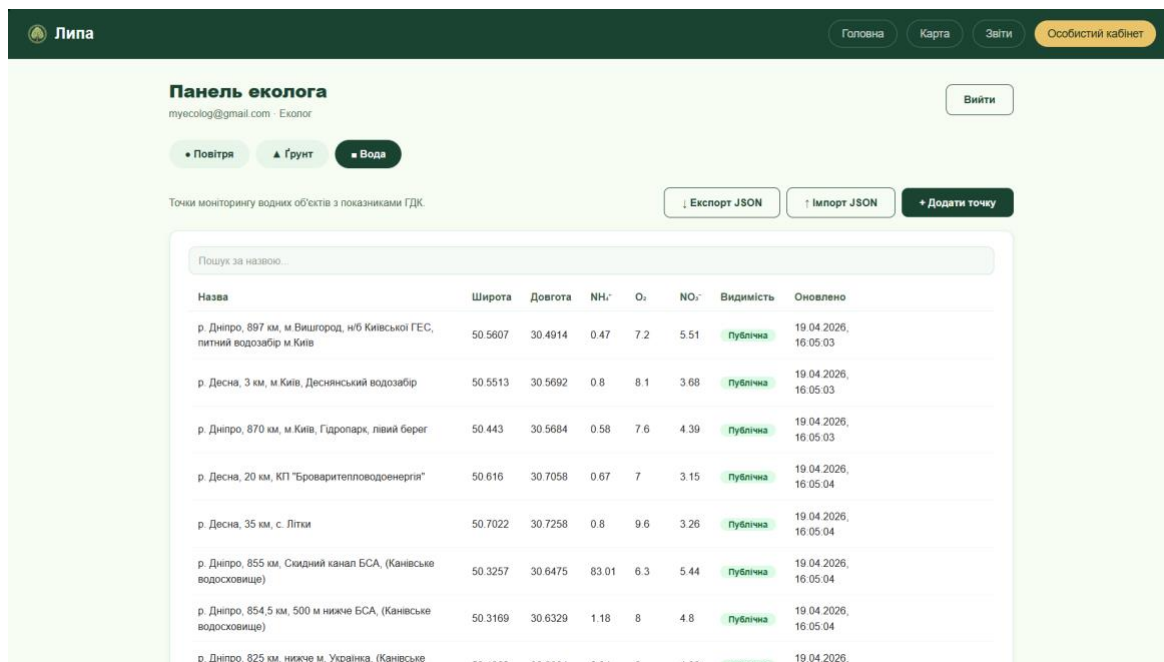


Рис. 4.6 Особистий кабінет з переліком зон і станцій

Розділ звітів (рис. 4.7) дозволяє обирати тип звіту (повітря, вода, ґрунт або узгоджений індекс за зоною) і переглядати результати у табличній формі з можливістю фільтрації за датою. Для кожної станції відображаються поточне значення індексу, категорія якості та динаміка змін. Передбачено можливість експорту результатів у форматі Word-документа (.docx) через бібліотеку docx для подальшої обробки або долучення до офіційних звітів.

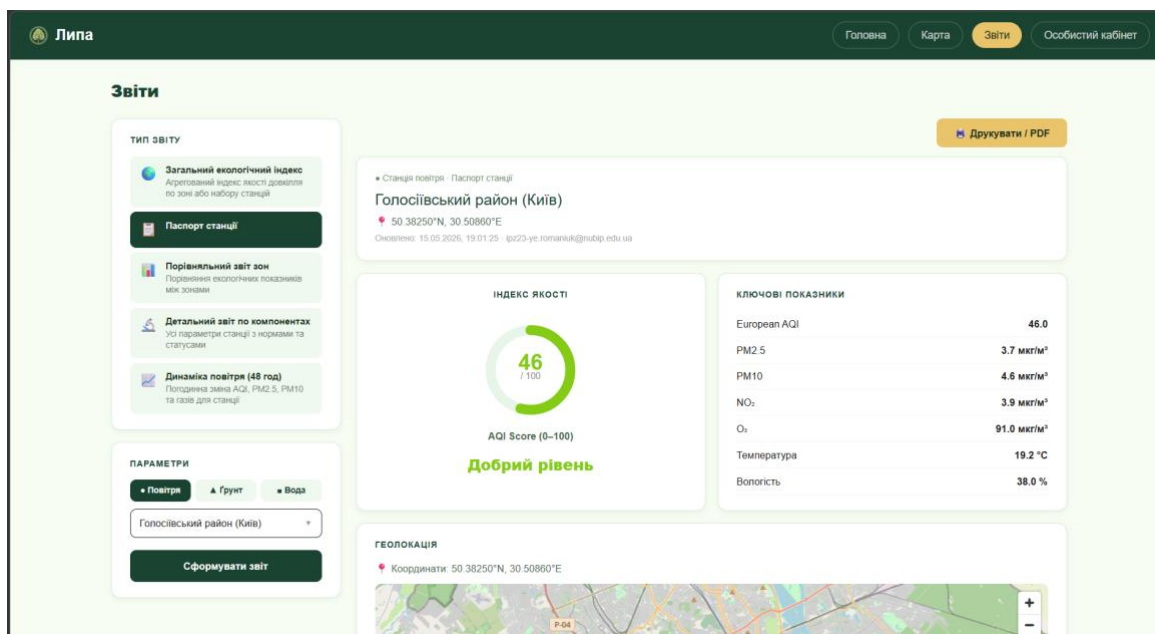


Рис. 4.7 Сторінка звітів та експорту результатів моніторингу

ВИСНОВКИ

У дипломній роботі було вирішено актуальну науково-практичну задачу – розроблено клієнт-серверне програмне забезпечення для визначення узгодженого екологічного індексу з урахуванням окремих метрик якості повітря, води та ґрунту. Систему було розгорнуто у відкритому доступі під назвою “Липа – Екологічний моніторинг України”. У процесі виконання роботи отримано такі результати.

- 1) Проведено аналіз предметної області – моніторингу якості навколишнього середовища. Розглянуто стандартизовані підходи до оцінювання стану повітря, води та ґрунту: European AQI за методикою Європейського агентства з охорони довкілля, WQI на основі гранично допустимих концентрацій та SQI з урахуванням деградації ґрунтових показників. Встановлено обмеженість наявних підходів у частині агрегування різнорідних індексів в єдину інтегральну оцінку для визначеної геопросторової зони.
- 2) Виконано огляд та порівняльний аналіз трьох конкурентних рішень – IQAir, Saveecobot та WAQI. З'ясовано, що жодна з платформ не надає комплексної оцінки стану довкілля, що включає одночасно повітря, воду і ґрунт, не підтримує створення довільних зон спостереження із обчисленням узгодженого індексу та не дозволяє екологам вносити власні вимірювання. Встановлена ніша стала підставою для формулювання функціональних вимог до розроблюваної системи.
- 3) Спроектовано архітектуру програмного продукту як трирівневої клієнт-серверної системи з окремою базою даних. Розроблено концептуальну та логічну моделі бази даних, що охоплюють дев'ять таблиць. Обґрунтовано вибір PostgreSQL 15 з розширенням PostGIS 3.4 як СУБД; побудовано фізичну схему з повним набором DDL-сценаріїв,

складеними індексами та JSONB-полем для зберігання координат зон спостереження.

- 4) Виконано об'єктно-орієнтоване проектування системи: виділено десять базових абстракцій предметної області, побудовано діаграму класів з відношеннями агрегації, залежності, реалізації та узагальнення. Розроблено п'ять діаграм кооперацій, що охоплюють ключові сценарії взаємодії: введення даних, обчислення індексу, формування звіту, взаємодію через інтерфейс та ієрархію ролей. Побудовано діаграми пакетів та компонентів, що відображають модульну структуру системи та її розгортання.
- 5) Реалізовано серверну частину системи у формі модульного NestJS-застосунку з REST API, що налічує понад 20 кінцевих точок. Кожен функціональний блок – автентифікація, повітряні станції, водні станції, ґрунтові станції, зони спостереження, довільні точки спостереження – оформлено у вигляді окремого NestJS-модуля з контролером та сервісом. Взаємодія з базою даних організована через TypeORM з декораторами @Entity та QueryBuilder. Автентифікацію реалізовано за схемою JWT з хешуванням паролів bcryptjs. Щогодинне оновлення даних автоматизовано декоратором @Cron.
- 6) Реалізовано алгоритми обчислення трьох часткових індексів. AQI обчислюється за кусково-лінійною інтерполяцією ЕЕА по п'яти забруднювачах (PM2.5, PM10, NO₂, O₃, SO₂) як максимум часткових значень. WQI визначається як максимальне кратне ГДК серед восьми параметрів якості води. SDI обчислюється як зважена сума п'яти компонентів деградації ґрунту з фіксованими вагами: рН (25%), електропровідність (20%), органічна речовина (20%), макроелементи (25%), мікроелементи (10%).
- 7) Реалізовано метод Сааті (АНР) для визначення вагових коефіцієнтів узгодженого екологічного індексу. На основі матриці парних порівнянь

отримано ваги $w_{\text{air}} = 0,50$, $w_{\text{water}} = 0,30$, $w_{\text{soil}} = 0,20$; консистентність матриці підтверджено ($CR < 0,1$). Узгоджений індекс E обчислюється як нормована зважена сума приведених до шкали 0–100 компонентів: $I_{\text{air}} = 100 - AQI$, $I_{\text{water}} = 100 - WQI \times 20$, $I_{\text{soil}} = 100 - SDI$. Часткова нормалізація забезпечує коректне обчислення індексу навіть за відсутності окремих типів станцій у зоні.

- 8) Реалізовано клієнтську частину системи на основі Next.js 14 (App Router), React та бібліотеки Leaflet. Розроблено інтерактивну карту з маркерами станцій і кольоровою індикацією рівня AQI, спливаючі вікна з детальними показниками, інструменти малювання зон спостереження (полігон, прямокутник, коло) та особистий кабінет для управління зонами й станціями. Реалізовано сторінку звітів з можливістю експорту у форматі .docx через бібліотеку docx.
- 9) Здійснено інтеграцію системи із зовнішнім сервісом Open-Meteo Air Quality API для автоматичного отримання погодинних даних якості повітря та метеопоказників. Запити виконуються пакетами по 5 станцій із затримкою 1500 мс між пакетами для дотримання обмежень API. Дані 48-годинної історії зберігаються в базі і автоматично очищуються за розкладом.
- 10) Виконано контейнеризацію бази даних за допомогою Docker і Docker Compose та налаштовано публікацію системи у мережу Інтернет через Cloudflare Tunnel з автоматичним TLS-сертифікатом. Проведено модульне тестування сервісів NestJS (Jest + NestJS Testing Module) та інтеграційне тестування REST API (supertest, 10 тест-кейсів). Усі заплановані тестові сценарії успішно пройдено; виявлені дефекти виправлено у процесі розробки.

Розроблений програмний продукт може використовуватися екологічними службами, органами місцевого самоврядування, проєктними та науково-дослідними установами для моніторингу стану довкілля, підготовки аналітичних

звітів та підтримки управлінських рішень у сфері охорони навколишнього середовища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Saaty T. L. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. New York: McGraw-Hill, 1980. 287 p.
- 2) Saaty T. L. Decision making with the analytic hierarchy process // *International Journal of Services Sciences*. 2008. Vol. 1, № 1. P. 83–98.
- 3) European Environment Agency. Air quality index. – [Електронний ресурс] – режим доступу: <https://www.eea.europa.eu/themes/air/air-quality-index> (дата звернення: 05.03.2026).
- 4) Brown R. M., McClelland N. I., Deininger R. A., Tozer R. G. A water quality index — do we dare? // *Water and Sewage Works*. 1970. Vol. 117, № 10. P. 339–343.
- 5) Karlen D. L., Mausbach M. J., Doran J. W., Cline R. G., Harris R. F., Schuman G. E. Soil quality: a concept, definition, and framework for evaluation // *Soil Science Society of America Journal*. 1997. Vol. 61, № 1. P. 4–10.
- 6) Open-Meteo Air Quality API. – [Електронний ресурс] – режим доступу: <https://open-meteo.com/en/docs/air-quality-api> (дата звернення: 12.03.2026).
- 7) ДСТУ 4174-2003. Якість атмосферного повітря. Загальні положення. Київ: Держспоживстандарт України, 2003. 28 с.
- 8) ДСТУ 4808:2007. Джерела централізованого питного водопостачання. Гігієнічні та технічні вимоги щодо якості води і контроль якості. Київ: Держспоживстандарт України, 2007. 24 с.
- 9) ДСТУ 4287:2004. Якість ґрунту. Відбирання проб. Київ: Держспоживстандарт України, 2004. 32 с.
- 10) Booch G., Rumbaugh J., Jacobson I. *Unified Modeling Language User Guide*. 2nd ed. Boston: Addison-Wesley, 2005. 496 p.

- 11) Fielding R. T. *Architectural Styles and the Design of Network-based Software Architectures* : dissertation. Irvine: University of California, 2000. 162 p.
- 12) NestJS Documentation. – [Электронный ресурс] – режим доступа: <https://docs.nestjs.com> (дата звернения: 02.04.2026).
- 13) Next.js Documentation. – [Электронный ресурс] – режим доступа: <https://nextjs.org/docs> (дата звернения: 02.04.2026).
- 14) PostgreSQL 15 Documentation. – [Электронный ресурс] – режим доступа: <https://www.postgresql.org/docs/15/> (дата звернения: 02.04.2026).
- 15) PostGIS Documentation. – [Электронный ресурс] – режим доступа: <https://postgis.net/documentation/> (дата звернения: 02.04.2026).
- 16) Leaflet — a JavaScript library for interactive maps. – [Электронный ресурс] – режим доступа: <https://leafletjs.com> (дата звернения: 02.04.2026).
- 17) OpenStreetMap Foundation. – [Электронный ресурс] – режим доступа: <https://www.openstreetmap.org> (дата звернения: 02.04.2026).
- 18) TypeORM Documentation. – [Электронный ресурс] – режим доступа: <https://typeorm.io> (дата звернения: 02.04.2026).
- 19) Docker Documentation. – [Электронный ресурс] – режим доступа: <https://docs.docker.com> (дата звернения: 04.04.2026).
- 20) Cloudflare Tunnel. – [Электронный ресурс] – режим доступа: <https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/> (дата звернения: 20.04.2026).
- 21) RFC 7519. JSON Web Token (JWT) / М. Jones, J. Bradley, N. Sakimura. 2015. – [Электронный ресурс] – режим доступа: <https://datatracker.ietf.org/doc/html/rfc7519> (дата звернения: 20.04.2026).

ДОДАТОК А

Програмний код обчислення екологічних індексів

А.1 Обчислення індексу якості повітря (European AQI)

Функція `computeEuropeanAQI` реалізована у файлі `backend/src/stations/stations.service.ts`. Алгоритм базується на методиці Європейського агентства з охорони навколишнього середовища (ЕЕА): для кожного забруднювача обчислюється суб-індекс методом кусково-лінійної інтерполяції між граничними точками, а остаточне значення AQI дорівнює максимуму серед усіх суб-індексів.

```
// Файл: backend/src/stations/stations.service.ts
// Обчислення European AQI (EEA — European Environment Agency)
```

```
type BP = [number, number, number, number];
// Формат:
// [conclow, conchigh, idxlow, idxhigh]

// -----
// Лінійна інтерполяція між точками розбиття
// -----
function piecewise(value: number, pts: BP[]): number {
  for (const [cL, cH, iL, iH] of pts) {
    if (value <= cH) {
      return iL + ((value - cL) / (cH - cL)) * (iH - iL);
    }
  }

  // Якщо значення вище останнього діапазону —
  // використовується лінійна екстраполяція
  return pts[pts.length - 1][3] +
    (value - pts[pts.length - 1][1]);
}
```

```
// -----
// Граничні точки для European AQI
// Одиниці вимірювання: мкг/м³
// -----
```

```
// PM2.5
const PM25_BP: BP[] = [
  [0, 10, 0, 25],
  [10, 20, 25, 50],
  [20, 25, 50, 75],
  [25, 50, 75, 100]
];
```

```
// PM10
```

```

const PM10_BP: BP[] = [
  [0, 20, 0, 25],
  [20, 40, 25, 50],
  [40, 50, 50, 75],
  [50, 100, 75, 100]
];

// NO2
const NO2_BP: BP[] = [
  [0, 40, 0, 25],
  [40, 90, 25, 50],
  [90, 120, 50, 75],
  [120, 230, 75, 100]
];

// O3
const O3_BP: BP[] = [
  [0, 50, 0, 25],
  [50, 100, 25, 50],
  [100, 130, 50, 75],
  [130, 240, 75, 100]
];

// SO2
const SO2_BP: BP[] = [
  [0, 100, 0, 25],
  [100, 200, 25, 50],
  [200, 350, 50, 75],
  [350, 500, 75, 100]
];

// -----
// Обчислення загального European AQI
// -----
// Підсумковий індекс визначається як
// максимальне значення серед усіх під-індексів
// -----

function computeEuropeanAQI(s: {
  pm25?: number | null;
  pm10?: number | null;
  no2?: number | null;
  o3?: number | null;
  so2?: number | null;
}): number | null {
  const subs: number[] = [];

  if (s.pm25 != null && s.pm25 >= 0) {
    subs.push(piecewise(s.pm25, PM25_BP));
  }

  if (s.pm10 != null && s.pm10 >= 0) {
    subs.push(piecewise(s.pm10, PM10_BP));
  }

  if (s.no2 != null && s.no2 >= 0) {

```

```

    subs.push(piecewise(s.no2, NO2_BP));
  }

  if (s.o3 != null && s.o3 >= 0) {
    subs.push(piecewise(s.o3, O3_BP));
  }

  if (s.so2 != null && s.so2 >= 0) {
    subs.push(piecewise(s.so2, SO2_BP));
  }

  // Якщо дані відсутні
  if (!subs.length) {
    return null;
  }

  // Повертається максимальний під-індекс
  return Math.round(Math.max(...subs));
}

```

A.2 Обчислення індексу якості води (WQI)

Функція computeWQI реалізована у файлі backend/src/water/water-stations.service.ts. Індекс розраховується як найбільше кратне ГДК (гранично допустимої концентрації) серед усіх контрольованих параметрів. Для розчиненого кисню застосовується інверсний критерій: нестача кисню нижче ГДК трактується як забруднення.

```

// Файл: backend/src/water/water-stations.service.ts
// Обчислення Water Quality Index (WQI)
// Відносний індекс забруднення води відносно ГДК

// -----
// ГДК показників якості води
// (згідно з ДСТУ / СанПіН)
// -----

const WQI_PARAMS: Record<string, {
  gdk: number;
  inverse?: boolean;
}> = {

  // Амоній-іони, мг/дм³
  ammonium: {
    gdk: 0.5
  },

  // БСК₅, мгО₂/дм³
  bod5: {
    gdk: 3
  },

  // Хлорид-іони, мг/дм³

```

```

chloride: {
  gdk: 350
},

// Розчинений кисень, мгО2/дм3
// Для кисню використовується inverse,
// оскільки менше значення = гірша якість
dissolved_oxygen: {
  gdk: 4,
  inverse: true
},

// Нітрат-іони, мг/дм3
nitrate: {
  gdk: 45
},

// Нітрит-іони, мг/дм3
nitrite: {
  gdk: 3.3
},

// Сульфат-іони, мг/дм3
sulfate: {
  gdk: 500
},

// Завислі речовини, мг/дм3
suspended_solids: {
  gdk: 25
}
};

// -----
// Обчислення Water Quality Index (WQI)
// -----
// WQI = максимальне кратне перевищення ГДК
// серед усіх доступних параметрів
//
// Інтерпретація:
// <= 3 → Допустима
// <= 4 → Помірно забруднена
// <= 5 → Забруднена
// <= 6 → Сильно забруднена
// > 6 → Критично забруднена
// -----

export function computeWQI(
  s: Partial<WaterMeasurement>
): number | null {

  const ratios: number[] = [];

  for (const [key, meta] of Object.entries(WQI_PARAMS)) {

    const val = (s as any)[key];

```

```

// Пропуск відсутніх або некоректних значень
if (val == null || val < 0) {
  continue;
}

let ratio: number;

// Для розчиненого кисню:
// нестача кисню = забруднення
if (meta.inverse) {

  ratio =
    val > 0 && val < meta.gdk
      ? meta.gdk / val
      : 0;

} else {

  // Звичайне відношення до ГДК
  ratio = val / meta.gdk;
}

ratios.push(ratio);
}

// Якщо немає даних
if (!ratios.length) {
  return null;
}

// Повертається максимальний коефіцієнт
// перевищення ГДК
return Math.round(Math.max(...ratios) * 100) / 100;
}

```

A.3 Обчислення індексу деградації ґрунту (SDI)

Функція `computeSQI` реалізована у файлі `backend/src/soil/soil-stations.service.ts`. Індекс деградації ґрунту (SDI) є зваженою сумою нормованих відхилень ключових агрохімічних параметрів від оптимальних значень: рН (25%), електропровідність (20%), органічна речовина (20%), макроелементи (25%) та мікроелементи (10%).

```

// Файл: backend/src/soil/soil-stations.service.ts
// Обчислення Soil Degradation Index (SDI)
// Індекс деградації ґрунту

// -----
// Обмеження значення у заданому діапазоні
// -----

function clamp(
  v: number,

```

```

    lo: number,
    hi: number
  ): number {

    return Math.max(lo, Math.min(hi, v));
  }

  // -----
  // Оптимальні значення макроелементів
  // Одиниці вимірювання: ppm
  // -----

  const MACRO_OPTIMAL: Record<string, number> = {

    // Нітратний азот
    no3_n: 30,

    // Фосфор (Mehlich)
    p_mehlich: 50,

    // Фосфор (Olsen)
    p_olsen: 20,

    // Калій
    k: 200,

    // Кальцій
    ca: 2000,

    // Магній
    mg: 200,

    // Сульфатна сірка
    so4_s: 20
  };

  // -----
  // Оптимальні значення мікроелементів
  // Одиниці вимірювання: ppm
  // -----

  const MICRO_OPTIMAL: Record<string, number> = {

    // Цинк
    zn: 3,

    // Марганець
    mn: 30,

    // Мідь
    cu: 2,

    // Залізо
    fe: 100,

    // Кобальт
    co: 0.5,

```

```

// Бор
b: 1,

// Молібден
mo: 0.3
};

// -----
// Обчислення Soil Degradation Index (SDI)
// -----
// SDI — зважена сума відхилень від норми
//
// Інтерпретація:
// 0–15 → Високоякісний ґрунт
// 16–30 → Добрий стан
// 31–50 → Задовільний стан
// 51–70 → Деградований
// 71–90 → Сильно деградований
// >90 → Критичний стан
// -----

export function computeSQI(
  s: Partial<SoilMeasurement>
): number | null {

  let sdi = 0;
  let hasAny = false;

  // -----
  // pH ґрунту (вага 25%)
  // Відхилення від нейтрального значення pH = 7
  // -----

  if (s.ph != null && s.ph >= 0) {

    sdi +=
      0.25 *
      clamp(
        Math.abs(s.ph - 7) * 20,
        0,
        100
      );

    hasAny = true;
  }

  // -----
  // Електропровідність (вага 20%)
  // Високе значення = засоленість ґрунту
  // -----

  if (
    s.conductivity != null &&
    s.conductivity >= 0
  ) {

```

```

sdi +=
  0.20 *
  clamp(
    s.conductivity * 25,
    0,
    100
  );

hasAny = true;
}

// -----
// Органічна речовина (вага 20%)
// Дефіцит відносно оптимального рівня 5%
// -----

if (
  s.organic_matter != null &&
  s.organic_matter >= 0
) {

  sdi +=
    0.20 *
    clamp(
      (5 - s.organic_matter) * 20,
      0,
      100
    );

  hasAny = true;
}

// -----
// Макроелементи (вага 25%)
// Середнє відхилення від оптимальних значень
// -----

const macroD: number[] = [];

for (const [field, optimal] of Object.entries(MACRO_OPTIMAL)) {

  const val = (s as any)[field];

  if (val != null && val >= 0) {

    macroD.push(
      clamp(
        ((optimal - val) / optimal) * 100,
        0,
        100
      )
    );
  }
}

if (macroD.length > 0) {

```

```

sdi +=
  0.25 *
  (
    macroD.reduce((a, b) => a + b, 0) /
    macroD.length
  );

  hasAny = true;
}

// -----
// Мікроелементи (вага 10%)
// Середнє відхилення від оптимальних значень
// -----

const microD: number[] = [];

for (const [field, optimal] of Object.entries(MICRO_OPTIMAL)) {

  const val = (s as any)[field];

  if (val != null && val >= 0) {

    microD.push(
      clamp(
        ((optimal - val) / optimal) * 100,
        0,
        100
      )
    );
  }
}

if (microD.length > 0) {

  sdi +=
    0.10 *
    (
      microD.reduce((a, b) => a + b, 0) /
      microD.length
    );

  hasAny = true;
}

// Якщо відсутні будь-які дані
if (!hasAny) {
  return null;
}

// Повернення підсумкового індексу
return Math.round(sdi * 100) / 100;
}

```

ДОДАТОК Б

SQL-схема бази даних

Нижче наведено повний SQL-скрипт створення реляційної схеми бази даних системи екологічного моніторингу. СУБД – PostgreSQL 15 з розширенням PostGIS. Скрипт охоплює таблиці користувачів, автоматичних повітряних станцій з вимірюваннями, ґрунтових та водних станцій, а також зон спостереження.

```
-- =====
```

```
-- SQL-схема бази даних системи екологічного моніторингу
```

```
-- СУБД: PostgreSQL 15 з розширенням PostGIS
```

```
-- =====
```

```
-- Очистка таблиць (порядок враховує зовнішні ключі)
```

```
DROP TABLE IF EXISTS
```

```
    air_measurement,
```

```
    soil_measurement,
```

```
    water_measurement,
```

```
    soil_station,
```

```
    water_station,
```

```
    station,
```

```
    zone,
```

```
    users
```

```
CASCADE;
```

```
-- =====
```

```
-- Користувачі системи
```

```
-- =====
```

```
CREATE TABLE users (
```

```
    id    SERIAL PRIMARY KEY,
```

```
    email VARCHAR(255) UNIQUE NOT NULL,
```

```
    password TEXT NOT NULL,
```

```

role    VARCHAR(50)
        DEFAULT 'ecologist'
        -- 'ecologist' | 'admin'
);

-- =====
-- Автоматичні повітряні станції (Open-Meteo API)
-- =====

CREATE TABLE station (

    id      SERIAL PRIMARY KEY,

    name     VARCHAR(255) NOT NULL,

    source   VARCHAR(100),
            -- 'district' | 'custom'

    station_type VARCHAR(20) NOT NULL,

    lat      FLOAT,
    lon      FLOAT,

    owner_id INT NULL REFERENCES users(id),

    is_public BOOLEAN DEFAULT TRUE,

    created_at  TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    UNIQUE (source, name)
);

```

```
-----
-- Вимірювання якості повітря
-----
```

```
CREATE TABLE air_measurement (
```

```
    id          SERIAL PRIMARY KEY,
```

```
    station_id  INT REFERENCES station(id)
```

```
        ON DELETE CASCADE,
```

```
    -- Тверді частинки
```

```
    pm25        FLOAT, -- мкг/м³
```

```
    pm10        FLOAT, -- мкг/м³
```

```
    dust        FLOAT, -- мкг/м³
```

```
    -- Газові забруднювачі
```

```
    no          FLOAT, -- мкг/м³
```

```
    no2         FLOAT, -- мкг/м³
```

```
    co          FLOAT, -- мкг/м³
```

```
    o3          FLOAT, -- мкг/м³
```

```
    so2         FLOAT, -- мкг/м³
```

```
    nh3         FLOAT, -- мкг/м³
```

```
    -- Індекси
```

```
    european_aqi FLOAT, -- European AQI (EEA)
```

```
    uv_index     FLOAT,
```

```
    -- Метеорологічні показники
```

```
    temperature  FLOAT, -- °C
```

```
    humidity     FLOAT, -- %
```

```
    pressure     FLOAT, -- гПа
```

```
    wind         FLOAT, -- км/год
```

```

    measured_at    TIMESTAMP
                  DEFAULT CURRENT_TIMESTAMP
);

```

```

CREATE INDEX idx_air_measurement_station_time
  ON air_measurement (
    station_id,
    measured_at
  );

```

```

-- =====
-- Ґрунтові станції (ручне введення)
-- =====

```

```

CREATE TABLE soil_station (

  id          SERIAL PRIMARY KEY,

  name        VARCHAR(255),

  lat         FLOAT,
  lon         FLOAT,

  "ownerId"   INT NULL,

  "isPublic"  BOOLEAN DEFAULT TRUE,

  "createdAt" TIMESTAMP
              DEFAULT CURRENT_TIMESTAMP
);

```

```

-- -----

```

-- Вимірювання ґрунту

CREATE TABLE soil_measurement (

id SERIAL PRIMARY KEY,

"stationId" INT REFERENCES soil_station(id)

ON DELETE CASCADE,

-- Базові показники

ph FLOAT, -- одиниці pH

conductivity FLOAT, -- mS/cm (засоленість)

organic_matter FLOAT, -- %

-- Макроелементи (ppm)

no3_n FLOAT, -- Нітратний азот

p_mehlich FLOAT, -- Фосфор (метод Мехліха)

p_olsen FLOAT, -- Фосфор (метод Олсена)

k FLOAT, -- Калій

ca FLOAT, -- Кальцій

mg FLOAT, -- Магній

so4_s FLOAT, -- Сірка (сульфат)

-- Мікроелементи (ppm)

zn FLOAT, -- Цинк

mn FLOAT, -- Марганець

cu FLOAT, -- Мідь

fe FLOAT, -- Залізо

co FLOAT, -- Кобальт

b FLOAT, -- Бор

```

mo          FLOAT, -- Молібден

-- Індекс деградації ґрунту
sqi          FLOAT, -- Soil Degradation Index (SDI)

"measuredAt"  TIMESTAMP
              DEFAULT CURRENT_TIMESTAMP
);

CREATE INDEX idx_soil_measurement_station_time
ON soil_measurement (
    "stationId",
    "measuredAt"
);

-- =====
-- Водні станції (ручне введення)
-- =====

CREATE TABLE water_station (

    id          SERIAL PRIMARY KEY,

    name         VARCHAR(255),

    lat          FLOAT,
    lon          FLOAT,

    "ownerId"    INT NULL,

    "isPublic"   BOOLEAN DEFAULT TRUE,

    "createdAt"  TIMESTAMP

```

```

        DEFAULT CURRENT_TIMESTAMP
    );

-----
-- Вимірювання якості води
-----

CREATE TABLE water_measurement (

    id          SERIAL PRIMARY KEY,

    "stationId" INT REFERENCES water_station(id)
                ON DELETE CASCADE,

    ammonium    FLOAT, -- Амоній-іони, мг/дм³

    bod5        FLOAT, -- БСК₅, мгО₂/дм³

    chloride    FLOAT, -- Хлорид-іони, мг/дм³

    dissolved_oxygen FLOAT, -- Кисень розчинений, мгО₂/дм³

    nitrate     FLOAT, -- Нітрат-іони, мг/дм³

    nitrite     FLOAT, -- Нітрит-іони, мг/дм³

    sulfate     FLOAT, -- Сульфат-іони, мг/дм³

    suspended_solids FLOAT, -- Завислі речовини, мг/дм³

    wqi         FLOAT, -- Water Quality Index

    "measuredAt"  TIMESTAMP

```

```

        DEFAULT CURRENT_TIMESTAMP
    );

CREATE INDEX idx_water_measurement_station_time
    ON water_measurement (
        "stationId",
        "measuredAt"
    );

-- =====
-- Зони спостереження (полігони)
-- =====

CREATE TABLE zone (

    id          SERIAL PRIMARY KEY,

    name        VARCHAR(255),

    "ownerId"   INT,

    coordinates JSONB,
        -- масив координат:
        -- [[lat, lon], ...]

    "createdAt" TIMESTAMP
        DEFAULT CURRENT_TIMESTAMP
);

-- =====
-- Зовнішні ключі
-- =====

```

```
ALTER TABLE soil_station  
  ADD FOREIGN KEY ("ownerId")  
  REFERENCES users(id);
```

```
ALTER TABLE water_station  
  ADD FOREIGN KEY ("ownerId")  
  REFERENCES users(id);
```

```
ALTER TABLE zone  
  ADD FOREIGN KEY ("ownerId")  
  REFERENCES users(id);
```

ДОДАТОК В

Таблиця маршрутів REST API системи

Таблиця В.1 – Маршрути REST API системи екологічного моніторингу

№	Метод	Маршрут	Призначення	Авторизація
Автентифікація та управління користувачами /auth				
1	POST	/auth/login	Вхід в систему; повертає JWT-токен, роль та email	Публічний
2	POST	/auth/seed	Створення першого адміністратора (блокується після першого виклику)	Публічний
3	POST	/auth/register	Створення нового облікового запису еколога адміністратором	JWT - admin
4	GET	/auth/users	Отримання переліку всіх користувачів системи	JWT - admin
5	PUT	/auth/users/ :id/password	Зміна пароля облікового запису користувача	JWT - admin
6	DELETE	/auth/users/:id	Видалення облікового запису (заборонено для адміністраторів)	JWT - admin
Повітряні станції моніторингу (автоматичні) /stations				
7	GET	/stations	Перелік усіх станцій з останнім вимірюванням AQI та метеопоказниками	Публічний
8	GET	/stations/ :id/history	Погодинна 48-годинна історія вимірювань конкретної станції	Публічний

9	GET	/stations/update	Ручний запуск оновлення даних від Open-Meteo API у фоновому режимі	JWT - ecologist/admin
10	GET	/stations/update/status	Статус поточного оновлення (чи виконується, час останнього запуску)	JWT - ecologist/admin
Водні станції моніторингу /water-stations				
11	GET	/water-stations	Перелік водних станцій (публічні + власні для авторизованих)	Публічний / JWT
12	POST	/water-stations	Створення станції з першим вимірюванням; WQI обчислюється автоматично	JWT - ecologist/admin
13	PUT	/water-stations/:id	Оновлення даних вимірювання; зберігається нова запис у хронології	JWT - власник/admin
14	DELETE	/water-stations/:id	Видалення станції разом з усіма вимірюваннями	JWT - власник/admin
Ґрунтові станції моніторингу /soil-stations				
15	GET	/soil-stations	Перелік ґрунтових станцій (публічні + власні для авторизованих)	Публічний / JWT
16	POST	/soil-stations	Створення станції з першим вимірюванням; SDI обчислюється автоматично	JWT - ecologist/admin
17	PUT	/soil-stations/:id	Оновлення ґрунтових показників з новим записом у хронології	JWT - власник/admin

18	DELETE	/soil-stations/:id	Видалення станції разом з усіма вимірюваннями	JWT - власник/admin
Довільні точки спостереження повітря /air-points				
19	GET	/air-points	Перелік користувацьких точок (публічні + власні)	Публічний / JWT
20	POST	/air-points	Створення нової точки; дані AQI завантажуються при наступному @Crop	JWT - ecologist/admin
21	PUT	/air-points/:id	Зміна назви, координат або видимості точки спостереження	JWT - власник/admin
22	DELETE	/air-points/:id	Видалення точки та пов'язаних вимірювань	JWT - власник/admin
Зони спостереження /zones				
23	GET	/zones	Перелік зон (власні для еколога, всі для адміністратора)	JWT
24	POST	/zones	Створення зони; координати полігону передаються масивом [lat, lon]	JWT
25	DELETE	/zones/:id	Видалення зони спостереження (адмін може видаляти будь-яку)	JWT - власник/admin

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

1. Моя бакалаврська кваліфікаційна робота на тему «Програмне забезпечення з визначення узгодженого екологічного індексу з урахуванням окремих метрик» є результатом моєї особистої праці.

3. Щодо використання інструментів ШІ зазначаю, що (обрати необхідне):

[+] інструменти III (ChatGPT, Gemini, Claude) були використані для:

- **[+]** стилістичного редагування та перевірки граматики;

- [+] допомоги у написанні/відлагодженні програмного коду;

- **[+]** інше: генерації зображень логотипу_____.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

« » 2026 p. _____

Євгеній РОМАНЮК