

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО”**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

“До захисту допущено”

Завідувач кафедри

\_\_\_\_\_ Наталія АУШЕВА

“\_\_\_” \_\_\_\_\_ 2026 р.

**Дипломна робота**

**на здобуття ступеня бакалавра**

За освітньою програмою “Цифрові технології в енергетиці”

Спеціальності 122 “Комп’ютерні науки”

на тему: “Вебсистема кореляційного аналізу екологічних та економічних  
показників території”

Виконала: студентка 4 курсу, групи ТР-22

\_\_\_\_\_ Головаченко Ксенія Володимирівна

(прізвище, ім’я, по батькові)

\_\_\_\_\_ (підпис)

Керівник: доктор техн. наук, проф. каф. цифрових технологій

в енергетиці, Сліпченко Володимир Георгійович

(посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові)

\_\_\_\_\_ (підпис)

Рецензент: к.т.н., доц. каф. цифрових технологій в енергетиці,

Пешко Віталій Анатолійович

(посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові)

\_\_\_\_\_ (підпис)

Н.контроль доцент каф. цифрових технологій в енергетиці,

д.ф., Гурін Артем Леонідович

(посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові)

\_\_\_\_\_ (підпис)

Засвідчую, що у цій дипломній роботі  
немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Київ — 2026

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО”**

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра

ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти — перший (бакалаврський)

Спеціальність 122 “Комп’ютерні науки”

Освітньо-професійна програма “Цифрові технології в енергетиці”

ЗАТВЕРДЖУЮ

Завідувач кафедри ЦТЕ

\_\_\_\_\_ Наталія АУШЕВА

“\_\_” \_\_\_\_\_ 2026 р.

**З А В Д А Н Н Я**  
**на дипломну роботу студенту**

Головченко Ксенії Володимирівні

(прізвище, ім’я, по батькові)

1. Тема роботи “Вебсистема кореляційного аналізу екологічних та економічних показників території”

керівник роботи доктор техн. наук, проф., Сліпченко Володимир Георгійович

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “28” травня 2026 р. № 1992 -с

2. Термін подання роботи студентом 08.06.2026 р.

3. Вихідні дані до роботи мова програмування Python, фреймворк FastAPI, система управління базами даних PostgreSQL, ORM-бібліотека SQLAlchemy, інструмент міграцій Alembic, бібліотеки наукових обчислень NumPy та SciPy, зовнішнє API джерело даних SaveEcoBot, система контролю версій Git, середовище розробки VS Code.

4. Перелік питань, які потрібно розробити 1) визначити мету й задачі роботи та проаналізувати предметну область кореляційного аналізу екологічних та економічних показників; 2) проаналізувати існуючі системи моніторингу та аналізу екологічних і економічних даних; 3) обрати засоби розробки для реалізації серверної частини системи та аналітичних модулів; 4) визначити методи статистичного аналізу, які використовуються в системі: кореляційний, регресійний аналіз, аналіз часових рядів та порівняльний аналіз територій; 5) спроектувати архітектуру програмного забезпечення та схему бази даних; 6) розробити серверну

частину системи та аналітичні модулі обробки даних; 7) описати методику роботи користувача з системою.

5. Орієнтовний перелік ілюстративного матеріалу діаграма USE-CASE, компонентна схема архітектури системи, діаграма послідовності взаємодії компонентів системи, скріншоти інтерфейсу користувача.

6. Дата видачі завдання 01.04.2026 р.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів виконання дипломної роботи                                                                | Термін виконання етапів дипломної роботи | Примітка |
|-------|--------------------------------------------------------------------------------------------------------|------------------------------------------|----------|
| 1     | Затвердження теми роботи                                                                               | 18.09.2025                               | Виконано |
| 2     | Дослідження існуючих рішень для аналізу екологічних та економічних даних і визначення вимог до системи | 11-19.04.26                              | Виконано |
| 3     | Розробка архітектури системи та вибір технологій для реалізації                                        | 15-21.04.26                              | Виконано |
| 4     | Проектування бази даних та розробка серверної частини системи                                          | 23-27.04.26                              | Виконано |
| 5     | Реалізація аналітичних модулів: кореляційний, регресійний аналіз, аналіз часових рядів                 | 27-28.04.26                              | Виконано |
| 6     | Програмна реалізація веб-системи кореляційного аналізу екологічних та економічних показників           | 29.04-06.05.26                           | Виконано |
| 7     | Оформлення пояснювальної записки                                                                       | 06-12.05.26                              | Виконано |
| 8     | Захист програмного продукту                                                                            | 14.05.26                                 | Виконано |
| 9     | Передзахист                                                                                            | 04.06.26                                 | Виконано |
| 10    | Захист                                                                                                 | 19.06.26                                 | Виконано |

Студент

\_\_\_\_\_  
(підпис)

Керівник роботи

\_\_\_\_\_  
(підпис)

Ксенія ГОЛОВЧЕНКО

(ім'я, ПРІЗВИЩЕ)

Володимир СЛІПЧЕНКО

(ім'я, ПРІЗВИЩЕ)

# АНОТАЦІЯ

Дипломна робота виконана на 81 сторінці, містить 47 ілюстрацій, 2 таблиці, 2 додатки, 20 джерел в переліку посилань.

Метою роботи є розробка веб-системи для кореляційного аналізу екологічних та економічних показників територій. Система збирає, зберігає та обробляє дані, щоб виявляти зв'язки між станом навколишнього середовища і соціально-економічним розвитком регіонів.

Робота охоплює аналіз предметної області та наявних рішень з моніторингу екологічних й економічних показників, вибір методів статистичного аналізу і засобів розробки, проектування архітектури та бази даних, реалізацію серверної частини з аналітичними модулями, а також опис роботи користувача з системою.

Методи та засоби: кореляційний аналіз (Пірсон, Спірмен, часткова та крос-кореляція), регресійний аналіз, аналіз часових рядів, порівняльний аналіз територій; мова програмування Python, фреймворк FastAPI, СУБД PostgreSQL, бібліотеки NumPy та SciPy, зовнішній API SaveEcoBot.

Результат: застосунок для кореляційного аналізу екологічних та економічних показників територій з автоматичним збором даних, статистичними розрахунками і генерацією PDF-звітів.

Застосунок може використовуватись органами місцевого самоврядування, аналітичними центрами та дослідницькими організаціями для прийняття управлінських рішень на основі аналізу зв'язків між екологічним станом і економікою регіонів.

**Ключові слова:** ВЕБ-ЗАСТОСУНОК, КОРЕЛЯЦІЙНИЙ АНАЛІЗ, ЕКОЛОГІЧНІ ПОКАЗНИКИ, ЕКОНОМІЧНІ ПОКАЗНИКИ, СТАТИСТИЧНИЙ АНАЛІЗ, FASTAPI, POSTGRESQL.

# ABSTRACT

This thesis consists of 81 pages and includes 47 figures, 2 tables, 2 appendices, and 20 references in the bibliography.

The aim of this work is to develop a web-based system for the correlation analysis of environmental and economic indicators of territories. The system collects, stores, and processes data to identify relationships between the state of the environment and the socio-economic development of regions.

The work covers an analysis of the subject area and existing solutions for monitoring environmental and economic indicators, the selection of statistical analysis methods and development tools, the design of the architecture and database, the implementation of the server-side with analytical modules, as well as a description of user interaction with the system.

Methods and tools: correlation analysis (Pearson, Spearman, partial and cross-correlation), regression analysis, time series analysis, comparative analysis of territories; Python programming language, FastAPI framework, PostgreSQL DBMS, NumPy and SciPy libraries, SaveEcoBot external API.

Result: an application for the correlation analysis of environmental and economic indicators of territories with automatic data collection, statistical calculations, and the generation of PDF reports.

The application can be used by local governments, think tanks, and research organizations to make management decisions based on the analysis of relationships between the environmental condition and the economy of regions.

**Keywords:** WEB APPLICATION, CORRELATION ANALYSIS, ENVIRONMENTAL INDICATORS, ECONOMIC INDICATORS, STATISTICAL ANALYSIS, FASTAPI, POSTGRESQL.

# ЗМІСТ

|                                                      |    |
|------------------------------------------------------|----|
| ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ ..... | 6  |
| ВСТУП.....                                           | 7  |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....                     | 9  |
| 1.1 Постановка прикладної задачі .....               | 9  |
| 1.2 Аналіз наявних рішень у сфері .....              | 10 |
| 1.2.1 STATISTICA .....                               | 11 |
| 1.2.2 IBM SPSS Statistics.....                       | 12 |
| 1.2.3 OpenAQ .....                                   | 13 |
| 1.2.4 DataWrapper.....                               | 14 |
| 1.2.5 Результат аналізу сучасних систем.....         | 16 |
| 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ .....        | 17 |
| 2.1 Технології для розробки серверної частини .....  | 17 |
| 2.2 Технології для розробки клієнтської частини..... | 20 |
| 3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ .....                   | 22 |
| 3.1 Архітектура системи .....                        | 22 |
| 3.2 Схема зберігання даних в системі .....           | 27 |
| 3.3 Методи статистичного аналізу .....               | 32 |
| 3.4 Реалізація серверної частини .....               | 34 |
| 3.5 Реалізація клієнтської частини .....             | 38 |
| 4 МЕТОДИКА РОБОТИ КОРИСТУВАЧА З СИСТЕМОЮ .....       | 49 |
| 4.1 Системні вимоги.....                             | 49 |
| 4.2 Сценарій роботи користувача з системою .....     | 51 |
| ВИСНОВКИ .....                                       | 62 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                     | 63 |
| ДОДАТОК А .....                                      | 66 |
| ДОДАТОК Б.....                                       | 76 |

# **ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ**

API — Application Programming Interface.

CSV — Comma-Separated Values.

HTTP — HyperText Transfer Protocol.

JWT — JSON Web Token.

ORM — Object-Relational Mapping.

PDF — Portable Document Format.

REST — Representational State Transfer.

SQL — Structured Query Language.

АКФ — автокореляційна функція.

БД — база даних.

ВВП — валовий внутрішній продукт.

ВЕБ-СИСТЕМА — програмний застосунок, що функціонує через мережу Інтернет і доступний через браузер.

КОРЕЛЯЦІЙНИЙ АНАЛІЗ — статистичний метод для виявлення та оцінки взаємозв'язків між змінними.

РЕГРЕСІЙНИЙ АНАЛІЗ — статистичний метод для моделювання залежності між змінними.

ТЕРИТОРІЯ — адміністративно-географічна одиниця, для якої збираються та аналізуються показники.

## ВСТУП

Екологічний стан території і економічні показники її розвитку на перший погляд живуть окремо, але між ними є залежності, і доволі стійкі. Промислові підприємства впливають на якість повітря і води, забруднення довкілля тягне за собою витрати на охорону здоров'я, падіння продуктивності сільського господарства, відтік населення. Проблема в тому, що ці зв'язки рідко фіксуються системно, адже дані є, але вони розкидані по різних відомствах і форматах, і зробити з них щось осмислене вручну практично нереально. Фахівець, який хоче зрозуміти, чи впливає концентрація промислових підприємств на динаміку захворюваності або продуктивність аграрного сектору, змушений вручну зводити таблиці, рахувати кореляції в Excel і будувати графіки окремо. Кореляційний аналіз дозволяє кількісно оцінити силу та напрямок таких залежностей, але існуючі інструменти або занадто складні для нетехнічного користувача, або не мають жодної прив'язки до екологічно-економічної предметної області.

У цьому контексті актуальним є створення вебсистеми, яка автоматизує процес від завантаження даних до інтерпретації кореляційних залежностей у зручному інтерфейсі, орієнтованому на роботу з показниками українських регіонів.

Метою даної роботи є розробка веб-системи кореляційного аналізу екологічних та економічних показників території з підтримкою кількох статистичних методів, автоматизованою підготовкою даних і можливістю генерації аналітичних звітів.

Для досягнення поставленої мети були сформовані наступні завдання:

- 1) проаналізувати наявні рішення та сервіси для статистичного аналізу екологічних та економічних даних;
- 2) обрати технології для створення серверної і клієнтської частин системи;
- 3) створити серверну частину додатку з реалізацією методів кореляційного, регресійного аналізу і аналізу часових рядів;
- 4) розробити структуру та дизайн користувацького інтерфейсу для



взаємодії з аналітичними функціями системи.

Дана дипломна робота складається з 4 розділів. Загальна кількість рисунків — 47, таблиць — 2, найменувань у списку використаних джерел — 20, основна частина викладена на 53 сторінках, повний обсяг роботи — 81 сторінок.

У першому розділі проведено аналіз предметної області, сформульовано постановку прикладної задачі та виконано огляд наявних рішень у сфері аналізу екологічних та економічних даних.

У другому розділі обґрунтовано вибір програмних засобів для розробки, а саме технологій серверної і клієнтської частин, системи управління базою даних і допоміжних бібліотек.

У третьому розділі описано програмну реалізацію системи: архітектуру, схему зберігання даних, реалізовані методи статистичного аналізу, а також особливості побудови серверної і клієнтської частин.

У четвертому розділі описано методику роботи користувача з системою, системні вимоги і покроковий сценарій взаємодії з інтерфейсом.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Статистичні дані про стан довкілля збираються роками: рівні викидів, забруднення водойм, якість ґрунтів. Паралельно існують економічні показники — ВРП, зайнятість, обсяги виробництва по галузях. Проблема не у відсутності даних, а в тому, що вони розрізнені, зберігаються в різних форматах і ніяк між собою не порівнюються. Фахівець, який хоче зрозуміти, чи впливає концентрація промислових підприємств на динаміку захворюваності або продуктивність аграрного сектору, змушений вручну зводити таблиці, рахувати кореляції вручну і будувати графіки окремо, а це незручно, повільно. Якраз під таку задачу і розроблялась описувана система — автоматизувати процес як і завантаження даних, так і інтерпретації кореляційних залежностей у зрозумілому та зручному вебінтерфейсі.

## 1.1 Постановка прикладної задачі

Взаємозв'язок між екологічним станом і економічним розвитком території — тема, яка активно досліджується з кінця 1970-х, коли стало очевидно, що індустріальне зростання дає не лише приріст ВВП, а й накопичені екологічні збитки, які врешті-решт самі починають гальмувати економіку. Деградація земель знижує врожайність, забруднення водойм б'є по рибальству і туризму, погіршення якості повітря збільшує витрати на охорону здоров'я — і все це врешті відбивається на регіональних економічних показниках. Такі ефекти розтягнуті в часі й не завжди очевидні без кількісного аналізу [1].

Кореляційний аналіз — це один із базових інструментів для виявлення таких залежностей. Він дозволяє встановити, наскільки тісно пов'язані два показники, і в якому напрямку ця залежність діє. Коефіцієнт Пірсона підходить для лінійних зв'язків між кількісними змінними, коефіцієнт Спірмена для випадків, коли дані не відповідають нормальному розподілу або мають порядковий характер. На практиці для аналізу екологічно-економічних даних частіше доводиться комбінувати обидва

підходи, бо вхідні набори даних рідко бувають ідеальними [2].

Мета роботи: розробити вебсистему, яка автоматизує кореляційний аналіз екологічних та економічних показників для вибраної території. Система повинна забезпечити завантаження та попередню обробку різнорідних наборів даних, розрахунок парних і множинних кореляцій, візуалізацію результатів у вигляді матриць, теплових карт і діаграм розсіювання, а також надати користувачеві зрозумілий інтерфейс без вимог до статистичного бекграунду.

Для досягнення цієї мети вирішуються такі завдання. Перше — проаналізувати предметну область і визначити ключові екологічні та економічні показники, між якими має сенс шукати залежності. Друге — розробити архітектуру вебсистеми з розподілом відповідальності між серверною і клієнтською частинами. Третє — реалізувати модуль завантаження й попередньої обробки даних, включно з фільтрацією аномалій і нормалізацією. Четверте — реалізувати обчислювальний модуль кореляційного аналізу з підтримкою кількох статистичних методів. П'яте — розробити інтерактивний інтерфейс для відображення результатів і навігації по наборах даних. Шосте — провести тестування системи на реальних даних і оцінити коректність результатів.

Об'єкт дослідження: процес кореляційного аналізу екологічних та економічних показників території. Предмет: методи і засоби реалізації вебсистеми, що автоматизує цей процес.

Практична цінність роботи полягає в тому, що готова система може використовуватись аналітиками, екологами або представниками місцевих органів влади для оцінки взаємозалежностей між станом довкілля і показниками розвитку регіону без необхідності вручну працювати зі статистичними пакетами.

## **1.2 Аналіз наявних рішень у сфері**

Перш ніж проектувати власне рішення, варто розібратись, що вже існує в цій ніші — і чому наявні інструменти не закривають задачу повністю. Ринок програмного забезпечення для роботи з екологічними та економічними даними

доволі широкий, але більшість продуктів або занадто вузькоспеціалізовані, або орієнтовані на зовсім інший рівень користувача. Нижче розглянуто чотири характерних представники.

### 1.2.1 STATISTICA

STATISTICA — статистичний пакет, розроблений StatSoft у 1984 році, пізніше придбаний TIBCO Software (рисунок 1.1). Десятиліттями використовується в академічному і промисловому середовищі як один із найбільш функціонально насичених інструментів для статистичного аналізу. Підтримує коефіцієнти Пірсона і Спірмена, часткові кореляції, матриці кореляцій для довільної кількості змінних, множинну регресію і окремий модуль для роботи з часовими рядами — з автокореляційними функціями, згладжуванням і виявленням трендів [3].

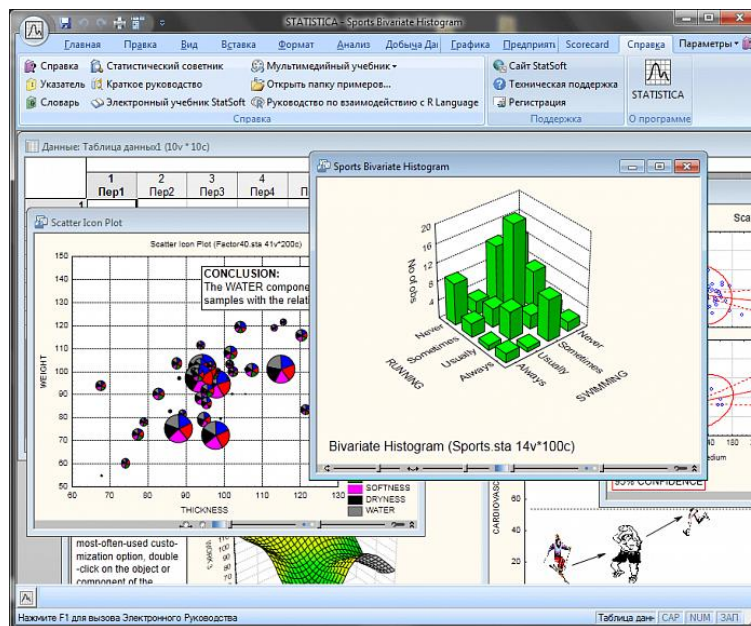


Рисунок 1.1 — Приклад інтерфейсу та візуалізації даних у STATISTICA

Для задачі цієї роботи STATISTICA має кілька принципових обмежень. Система повністю десктопна: немає вебінтерфейсу, немає спільного доступу, результати існують локально на одній машині. Крім того, пакет предметно нейтральний: він не знає нічого про екологічні показники, регіони чи структуру

даних моніторингу. Щоб порахувати кореляцію між PM2.5 і рівнем безробіття по конкретній області, потрібно спочатку вручну зібрати дані з різних джерел, синхронізувати часові ряди, обробити пропуски і привести все до формату, який система розуміє. Ніякої автоматизації підготовчого етапу немає. Географічна прив'язка даних теж не підтримується, тобто регіон кодується як звичайна категоріальна змінна, порівняти кілька територій одночасно без ручного доопрацювання не вийде. Після переходу під TIBCO ліцензування стало корпоративним: повноцінна версія коштує від кількох сотень доларів і вище, що робить її фактично недоступною для бюджетних держустанов [3].

### 1.2.2 IBM SPSS Statistics

IBM SPSS традиційно орієнтований на соціальні науки і прикладну статистику, інструмент широко використовується в академічному середовищі (рисунок 1.2). Він підтримує кореляції Пірсона, Спірмена і Кендалла, часткові кореляції, регресійний аналіз і окремий модуль Forecasting для роботи з часовими рядами [4].

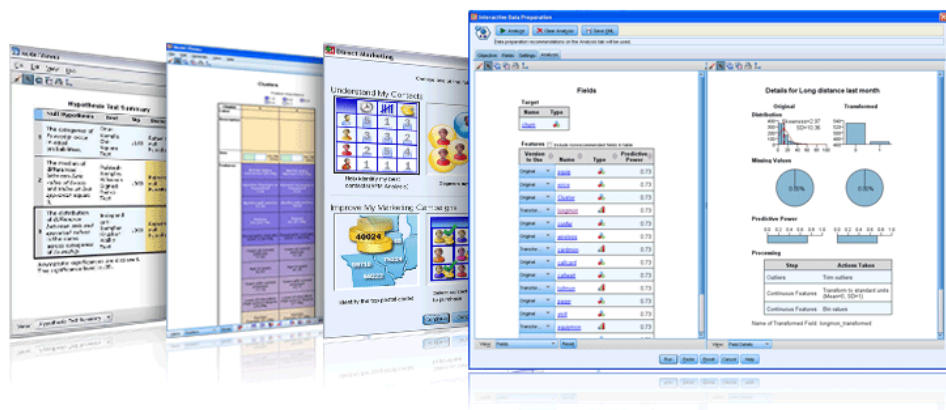


Рисунок 1.2 — Інтерфейс IBM SPSS Statistics з модулями кластерного аналізу та підготовки даних

Основна проблема SPSS: висока складність зрозумілості для нетехнічного користувача. Інтерфейс не інтуїтивний: глибока вкладеність меню і власна мова

команд (SPSS Syntax), для нетривіального аналізу потрібно або знати синтаксис, або витратити значний час на пошук потрібних опцій. При імпорті зовнішніх даних регулярно виникають проблеми з кодуванням і форматами дат — особливо при роботі з кириличними назвами і нестандартними часовими мітками. Обробка пропущених значень обмежена: лінійна інтерполяція доступна тільки в модулі часових рядів і не може застосовуватись як загальний крок підготовки даних. Як і STATISTICA, SPSS є ізольованим десктопним інструментом без підтримки вебдоступу і без будь-якої інтеграції з зовнішніми джерелами. Модульна структура ліцензування передбачає, що повний набір функцій, включно з Forecasting і Advanced Statistics, потребуватиме окремих підписок, а загальна вартість витрат для організації може становити кілька тисяч доларів на рік [4].

### 1.2.3 OpenAQ

OpenAQ — відкрита платформа агрегації даних про якість повітря, запущена у 2015 році як некомерційний проєкт (рисунок 1.3). Система збирає вимірювання з урядових і дослідницьких станцій моніторингу по всьому світу: станом на 2024 рік база охоплює понад 100 країн і десятки тисяч сенсорів. Доступ до даних реалізований через REST API і вебінтерфейс із базовою картою розміщення станцій [5].

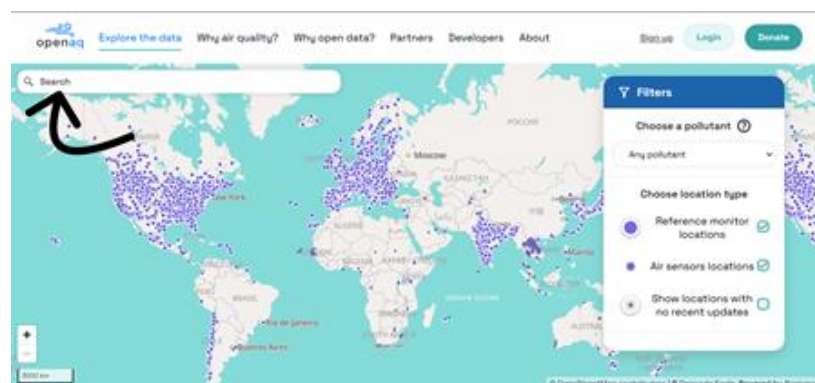


Рисунок 1.3 — Інтерфейс платформи OpenAQ Explorer з картою станцій моніторингу якості повітря

З технічного боку OpenAQ вирішує проблему — централізує розрізнені дані моніторингу в одному місці з єдиним форматом відповіді. Для розробника це зручно: API повертає JSON із вимірюваннями, координатами станції, часовою міткою і одиницями вимірювання. Однак як аналітичний інструмент платформа має суттєві обмеження, бо OpenAQ — це виключно сховище сирих екологічних даних. Жодного аналітичного функціоналу: немає розрахунку кореляцій, немає статистичних тестів, немає навіть базової агрегації по регіонах у вигляді, зручному для подальшого аналізу. Дані повертаються у вигляді точкових вимірювань по конкретній станції — щоб отримати середньомісячне значення PM2.5 по Харківській області, потрібно самостійно вибрати всі станції регіону, зробити кілька запитів до API, об'єднати результати і порахувати агрегат [6].

Станції моніторингу регулярно виходять з ладу, не передають дані або змінюють одиниці вимірювання без попередження. OpenAQ ніяк не сигналізує про наявність прогалин у ряді і не пропонує механізму їх заповнення. При завантаженні даних за кілька років по одній станції цілком реально отримати ряд із місяцями без жодного вимірювання — і якщо не перевірити це вручну, результати подальшого аналізу будуть спотвореними. Економічних даних у системі немає взагалі — платформа зосереджена виключно на якості повітря, тому для завдання кореляційного аналізу еколого-економічних показників OpenAQ може виступати лише одним із джерел вхідних даних, але не повноцінним рішенням [5].

#### **1.2.4 DataWrapper**

DataWrapper — це вебінструмент для створення інтерактивних візуалізацій, розроблений у 2012 році (рисунок 1.4). Платформа дозволяє завантажити таблицю у форматі CSV або Excel і за кілька кроків отримати готовий інтерактивний графік, хорошу карту чи таблицю без особливих знань. Підтримуються лінійні і стовпчасті діаграми, діаграми розсіювання, хороплет-карти, таблиці з умовним форматуванням. Результат можна вбудувати на сторінку або поділитись посиланням [7].

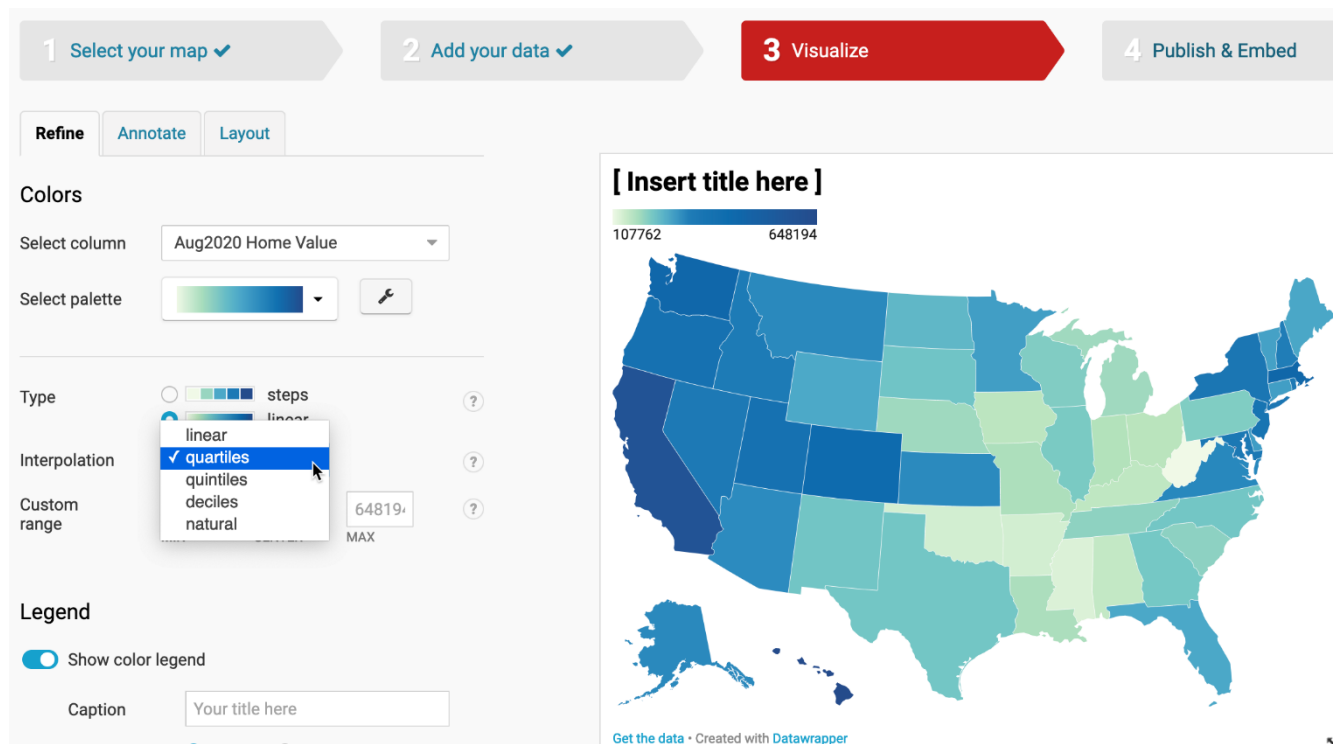


Рисунок 1.4 — Інтерфейс DataWrapper: налаштування хороплет-карти з вибором методу інтерполяції кольору

Для швидкої публікації вже підготовлених даних DataWrapper справді зручний, але в контексті кореляційного аналізу екологічних та економічних показників інструмент має обмеження: він займається виключно відображенням даних, а не їх аналізом. Scatter plot у DataWrapper покаже розкид точок двох змінних, але ніякого коефіцієнта кореляції, чи наприклад оцінки статистичної значущості система не розраховує і не виводить. Користувач бачить картинку, але не отримує числового підтвердження чи спростування залежності [7].

Препроцесинг даних у DataWrapper відсутній повністю. Платформа очікує чисту таблицю на вході і якщо у даних є пропуски, викиди або різні часові розрізи для різних показників, це потрібно виправити до завантаження, у зовнішньому інструменті. Ніякої інтеграції з джерелами даних теж немає: підключити API SaveEcoBot або автоматично підтягнути свіжу статистику Держстату неможливо. Кожне оновлення візуалізації вимагає ручного перезавантаження файлу. Безкоштовний тариф накладає обмеження на кількість графіків і прибирає брендинг лише в платних версіях — від 149 євро на місяць для організацій [8].



### 1.2.5 Результат аналізу сучасних систем

Аналіз розглянутих рішень показує два полюси: потужні статистичні пакети, які вміють усе, але вимагають високої кваліфікації і не мають предметної прив'язки; та зручні вебінструменти з гарною візуалізацією, але без серйозної аналітики і без інтеграції між джерелами даних. Таблиця 1.1 узагальнює основні характеристики розглянутих систем.

Таблиця 1.1 — Порівняльна характеристика наявних рішень

| Характеристика               | STATISTICA<br>SPSS                 | OpenAQ<br>DataWrapper     |
|------------------------------|------------------------------------|---------------------------|
| Кореляційний аналіз          | Повний (Пірсон, Спірмен, часткова) | Відсутній або поверхневий |
| Обробка пропусків і викидів  | Ручна                              | Ручна                     |
| Вебінтерфейс                 | Ні (десктоп)                       | Частково                  |
| Інтеграція еко+економ. даних | Ні                                 | Ні                        |
| Доступність                  | Платна ліцензія                    | Безкоштовно / freemium    |

Система, що розроблена в ході роботи усуває ключові недоліки обох підходів: автоматизує попередню обробку даних і вбудовує це в єдиний пайплайн разом із розрахунком кореляцій і візуалізацією результатів у вебінтерфейсі.

## 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ

Вибір технологічного стеку для вебсистеми кореляційного аналізу визначався кількома ключовими вимогами: серверна частина повинна ефективно виконувати обчислювально навантажені статистичні розрахунки, а клієнтська забезпечувати візуалізацію результатів у реальному часі; загальна архітектура має підтримувати розширення набору показників і територій без переписування основної логіки.

### 2.1 Технології для розробки серверної частини

Серверна частина системи реалізована на мові програмування Python з використанням фреймворку FastAPI. Вибір Python у контексті цієї задачі очевидний: мова має найбільш зрілу екосистему бібліотек для наукових обчислень (NumPy, SciPy, pandas, statsmodels) і фактично є стандартом у задачах аналізу даних. Альтернативи на кшталт Node.js або Java технічно дозволяють реалізувати REST API, але для статистичних розрахунків вимагають або підключення зовнішніх сервісів, або використання значно менш зрілих бібліотек. Писати кореляційний аналіз із нуля на Java, коли в Python є `scipy.stats` з готовою реалізацією коефіцієнтів Пірсона, Спірмена і часткових кореляцій не виправдано [9].

Серед Python-фреймворків для побудови REST API основними варіантами були Django REST Framework і FastAPI. Django — це широко використовуваний фреймворк із великою кількістю вбудованих інструментів: ORM, адмін-панель, автентифікація. Але для цієї системи його надмірність є скоріше недоліком. Django проєктувався навколо концепції повноцінного вебдодатку з шаблонами і сесіями, і його архітектура накладає обмеження на гнучкість при побудові API-сервісу. FastAPI ж розроблявся саме як інструмент для створення асинхронних API [9].

Ключова перевага FastAPI для цієї задачі є в підтримці асинхронного виконання через `async/await`. Запити на кореляційний аналіз, особливо при роботі з великими часовими рядами або матрицями кількох індикаторів одночасно, є

обчислювально затратними. Асинхронна модель дозволяє серверу обробляти кілька таких запитів паралельно, не блокуючи виконання на час очікування результату від бази даних або зовнішнього API. Flask, ще одна популярна альтернатива, є синхронним за замовчуванням і вимагає додаткових зусиль для досягнення аналогічної поведінки [9].

Окрема перевага FastAPI — автоматична валідація вхідних даних через Pydantic і автогенерація документації OpenAPI. Pydantic перевіряє типи, наявність обов'язкових полів і відповідність значень переліченим варіантам автоматично [9].

Для роботи з базою даних використовується SQLAlchemy як ORM у поєднанні з Alembic для керування міграціями. SQLAlchemy дозволяє описувати моделі даних як Python-класи і генерувати SQL-запити. Alembic забезпечує версіонування схеми бази — при додаванні нового індикатора або зміні структури таблиці міграція фіксується у файлі і може бути застосована або відкочена без ручних змін у базі.

Як СУБД обрано PostgreSQL. Порівняно з SQLite, який часто використовується на етапі прототипування, PostgreSQL краще справляється з конкурентними запитами і підтримує індексування по числових полях, що важливо при вибірці часових рядів по конкретній території і діапазону дат. MySQL є повноцінною альтернативою, але PostgreSQL має кращу підтримку JSON-полів (у системі результати аналізу зберігаються у полі типу JSONB).

Збір екологічних даних реалізований через інтеграцію з SaveEcoBot API (рисунок 2.1). Це публічний API, який надає доступ до даних українських станцій моніторингу якості повітря — PM2.5, PM10, та інших показників. Сервіс DataImportService відповідає за геолокацію станцій відносно збережених територій, вибір найближчої станції при відсутності даних по конкретному місту і нормалізацію одиниць вимірювання. Для економічних показників передбачено завантаження CSV-файлів вручну (дані Держстату по ВВП, зайнятості, інвестиціях і охороні здоров'я не мають публічного API і завантажуються адміністратором системи) [10].

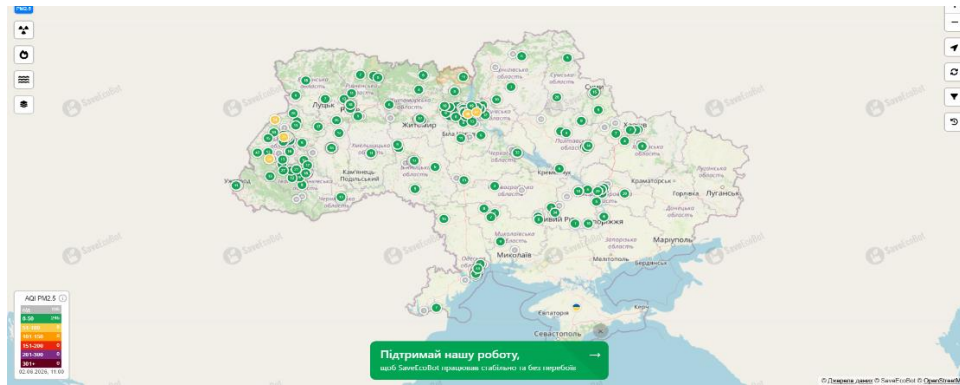


Рисунок 2.1 — Інтерфейс платформи SaveEcoBot з даними моніторингу якості повітря

Архітектура серверної частини організована за принципом розподілу відповідальності: маршрутизатори API (routers) приймають запити і викликають сервіси, сервіси містять бізнес-логіку і звертаються до модулів аналізу або бази даних, модулі аналізу (correlation, partial, timeseries, regression, comparison) реалізують статистичні обчислення і не залежать від решти системи. Така структура дозволяє тестувати аналітичні модулі ізольовано і додавати нові методи аналізу без зміни логіки маршрутизації чи роботи з базою даних. Загальна архітектура системи з розподілом на рівні Frontend, Backend і зовнішні сервіси наведена на рисунку 2.2.

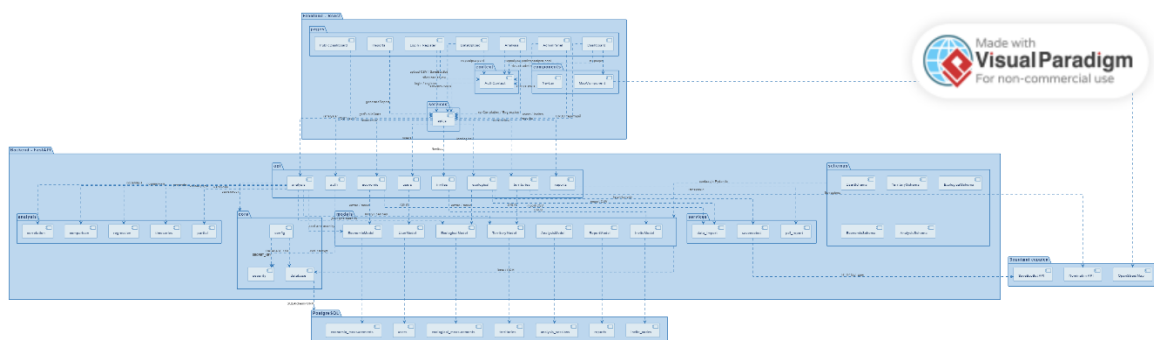


Рисунок 2.2 — Компонентна діаграма архітектури системи

Архітектура серверної частини організована за принципом розподілу відповідальності: маршрутизатори API (routers) приймають запити і викликають сервіси, сервіси містять бізнес-логіку і звертаються до модулів аналізу або бази даних, модулі аналізу (correlation, partial, timeseries, regression, comparison) реалізують статистичні обчислення і не залежать від решти системи. Така структура дозволяє тестувати аналітичні модулі ізольовано і додавати нові методи аналізу без зміни логіки.

## **2.2 Технології для розробки клієнтської частини**

Клієнтська частина системи реалізована на бібліотеці React.

Альтернативами React у цьому контексті були Vue.js і Angular. Vue має нижчий поріг входу і простіший синтаксис шаблонів, але менш розвинену екосистему для складних аналітичних інтерфейсів — зокрема, бібліотеки для побудови інтерактивних графіків під Vue значно поступаються React-аналогам за функціональністю і документацією. Angular — це повноцінний фреймворк із жорсткою структурою і вбудованою типізацією через TypeScript. Для великих корпоративних додатків це перевага, але для системи середнього масштабу Angular додає надлишкову складність: обов'язкові декоратори, модулі, сервіси і dependency injection збільшують кількість коду без пропорційної вигоди. React у цьому сенсі займає середню позицію, бо він достатньо гнучкий, щоб не нав'язувати зайву структуру, і достатньо ефективний, щоб мати готові рішення для всіх типових задач інтерфейсу [11].

Для управління станом застосування використовується React Context API через AuthContext. Контекст зберігає стан автентифікації користувача (токен доступу, роль, базову інформацію профілю) і забезпечує доступ до цих даних з будь-якого компонента без передачі через props [20]. Альтернативою міг би бути Redux, але для цієї системи його використання надлишкове: Redux виправданий при складній глобальній логіці з багатьма взаємозалежними частинами стану, тоді як тут більшість даних є локальними для конкретної сторінки і не потребують

централізованого сховища [11].

Комунікація з серверною частиною реалізована через бібліотеку Axios. Усі запити до FastAPI-бекенду, а саме: отримання списку територій, запуск аналізу, завантаження CSV, генерація звіту — виконуються через єдиний модуль `api.js`, де налаштовані базова URL-адреса і перехоплювач запитів для автоматичного додавання JWT-токена, аби не дублювати логіку автентифікації в кожному виклику окремо. Fetch API, вбудований у браузер, міг би виконати ту ж функцію, але Axios зручніший завдяки автоматичній серіалізації JSON, обробці HTTP-помилки і підтримці перехоплювачів запитів і відповідей без додаткового коду.

Для маршрутизації між сторінками застосовується React Router. Система має кілька окремих сторінок — публічний дашборд, сторінка аналізу, завантаження даних, звіти, адмін-панель і React Router забезпечує навігацію між ними без перезавантаження сторінки. Захищені маршрути реалізовані через перевірку стану автентифікації з `AuthContext`: неавторизований користувач автоматично перенаправляється на сторінку входу [11].

Візуалізація результатів аналізу побудована на бібліотеці Recharts. Вона надає набір готових компонентів для побудови графіків (`LineChart`, `BarChart`, `ScatterChart`), які інтегруються як звичайні елементи і підтримують інтерактивність: підказки при наведенні, масштабування, фільтрацію через легенду. Для відображення матриць кореляцій з тепловим кодуванням значень використовується власна реалізація на основі SVG з розрахунком кольору комірки залежно від величини коефіцієнта. D3.js міг би забезпечити більшу гнучкість у побудові нестандартних візуалізацій, але вимагає значно більше коду для базових сценаріїв і погано інтегрується з React-моделлю [12].

Для відображення географічної карти з позначенням територій використовується OpenStreetMap. OpenStreetMap обрано як безкоштовну альтернативу Google Maps API, яка не вимагає ключа з платіжними лімітами і повністю покриває потреби системи у відображенні географічних об'єктів на рівні регіонів України [19].

## 3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Цей розділ описує архітектурні рішення системи, структуру зберігання даних, реалізовані методи статистичного аналізу та особливості побудови серверної і клієнтської частин. Кожне технічне рішення обґрунтовується вимогами до функціональності системи і специфікою предметної області.

### 3.1 Архітектура системи

Система побудована за клієнт-серверною архітектурою з чітким розподілом відповідальності між рівнями. Фронтенд на React відповідає виключно за відображення даних і взаємодію з користувачем, бекенд на FastAPI за обробку запитів, виконання обчислень і доступ до бази даних, PostgreSQL відповідає за зберігання вимірювань, результатів аналізу і службової інформації. Зовнішні сервіси: SaveEcoBot API і OpenStreetMap підключаються на рівні бекенду і клієнта відповідно.

Для розуміння функціональних вимог до системи розроблено діаграму варіантів використання, яка відображає взаємодію акторів із системою (рисунк 3.1).

Гість має доступ до публічного дашборду і перегляду загальної статистики без автентифікації. Аналітик даних після входу в систему отримує повний набір функцій: управління даними, завантаження екологічних і економічних CSV-файлів, збір даних із SaveEcoBot API, запуск аналізу і перегляд результатів із можливістю генерації PDF-звіту. Адміністратор додатково має доступ до управління акаунтами користувачів, перегляду системних логів і налаштування API-ключів. Особа, що приймає рішення (ЛПР), взаємодіє з публічним дашбордом і може переглядати агреговану статистику по регіонах без необхідності розбиратись у деталях аналітичного процесу.



Рисунок 3.1 — Use case діаграма

Компонентна архітектура системи деталізує внутрішню структуру кожного рівня і зв'язки між модулями (рисунок 3.2).



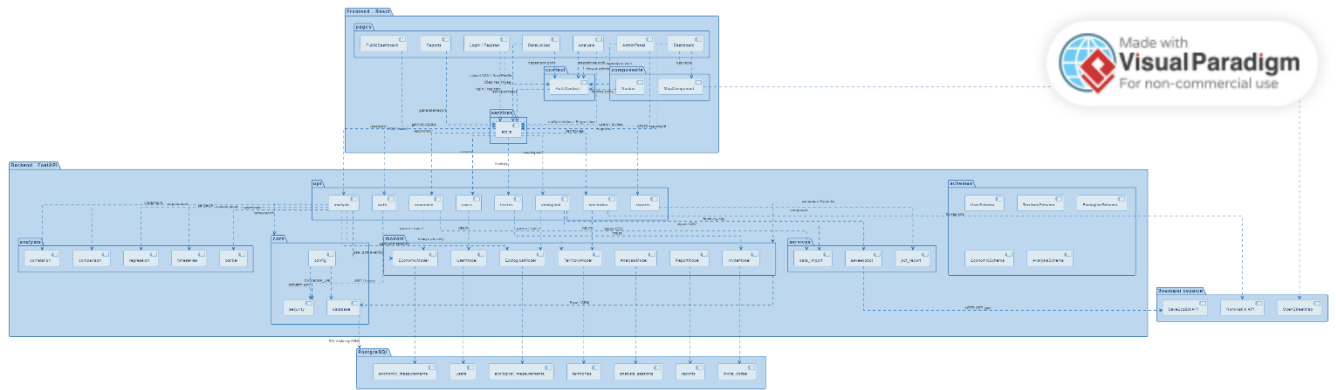


Рисунок 3.2 — Компонентна діаграма архітектури вебсистеми кореляційного аналізу

На рівні фронтенду виділяються три групи елементів: сторінки (pages) — PublicDashboard, Analysis, DataUpload, Reports, AdminPanel, Dashboard; компоненти (components) — AuthContext, Navbar, MapComponent; і сервісний шар (services) — модуль api.js, через який проходять усі HTTP-запити до бекенду. Розподіл дозволяє змінювати логіку окремої сторінки без впливу на решту інтерфейсу.

Бекенд організований за шаровою структурою. Верхній шар — маршрутизатори API (routers), які приймають HTTP-запити і делегують їх відповідним сервісам. Маршрутизатори розбиті за функціональними областями: analysis, auth, economic, ecological, territories, users, reports, invited. Кожен маршрутизатор обробляє тільки свою частину API і не знає про існування інших. Середній шар — це сервіси, що містять бізнес-логіку: DataImportService відповідає за імпорт і нормалізацію даних, SaveEcoBotService за взаємодію із зовнішнім API моніторингу, PDFReportService за генерацію звітів, SecurityService за автентифікацію і роботу з JWT-токенами. Нижній аналітичний шар — це модулі обчислень (correlation, partial, regression, timeseries, comparison), які реалізують статистичні алгоритми і не мають залежностей від решти системи, що дозволяє тестувати їх ізольовано.

База даних PostgreSQL містить десять таблиць (рисунок 3.3): users, territories, ecological\_measurements, economic\_measurements, analysis\_sessions, analysis\_results,

reports, invite\_codes, datasets, system\_logs. Зв'язок між таблицями вимірювань і territories реалізований через зовнішній ключ territory\_id, щоб вибирати дані по конкретному регіону без повного сканування таблиці. Результати кожного сеансу аналізу зберігаються в таблиці analysis\_sessions у полі типу JSONB — аби зберігати довільно структуровані об'єкти результатів для різних типів аналізу без створення окремих таблиць під кожен метод. Таблиця analysis\_results виступає як окреме сховище фінальних результатів з можливістю швидкого доступу, system\_logs фіксує системні події для адміністративного моніторингу, datasets зберігає метадані завантажених наборів даних.

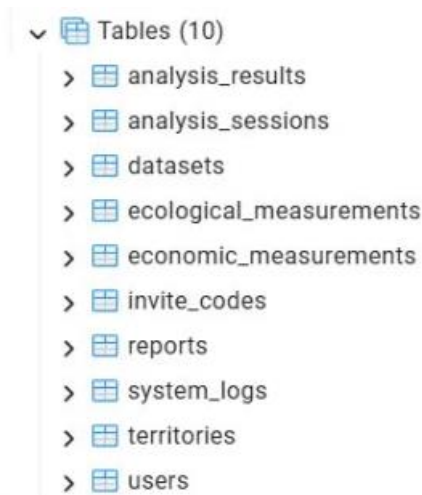


Рисунок 3.3 — Перелік таблиць бази даних системи у pgAdmin

Взаємодія між компонентами системи у типовому сценарії виглядає так: користувач обирає територію і показники на сторінці Analysis, фронтенд формує POST-запит до /api/analysis/correlation із параметрами, FastAPI-маршрутизатор передає запит у CorrelationService, сервіс отримує дані з БД, передає їх у модуль correlation.py для обчислень, результат зберігається в analysis\_sessions і повертається клієнту у вигляді JSON. Весь цикл від запиту до відображення результату відбувається без перезавантаження сторінки. Послідовність взаємодії компонентів при виконанні типових сценаріїв детально відображена на діаграмі послідовності (рисунок 3.4).

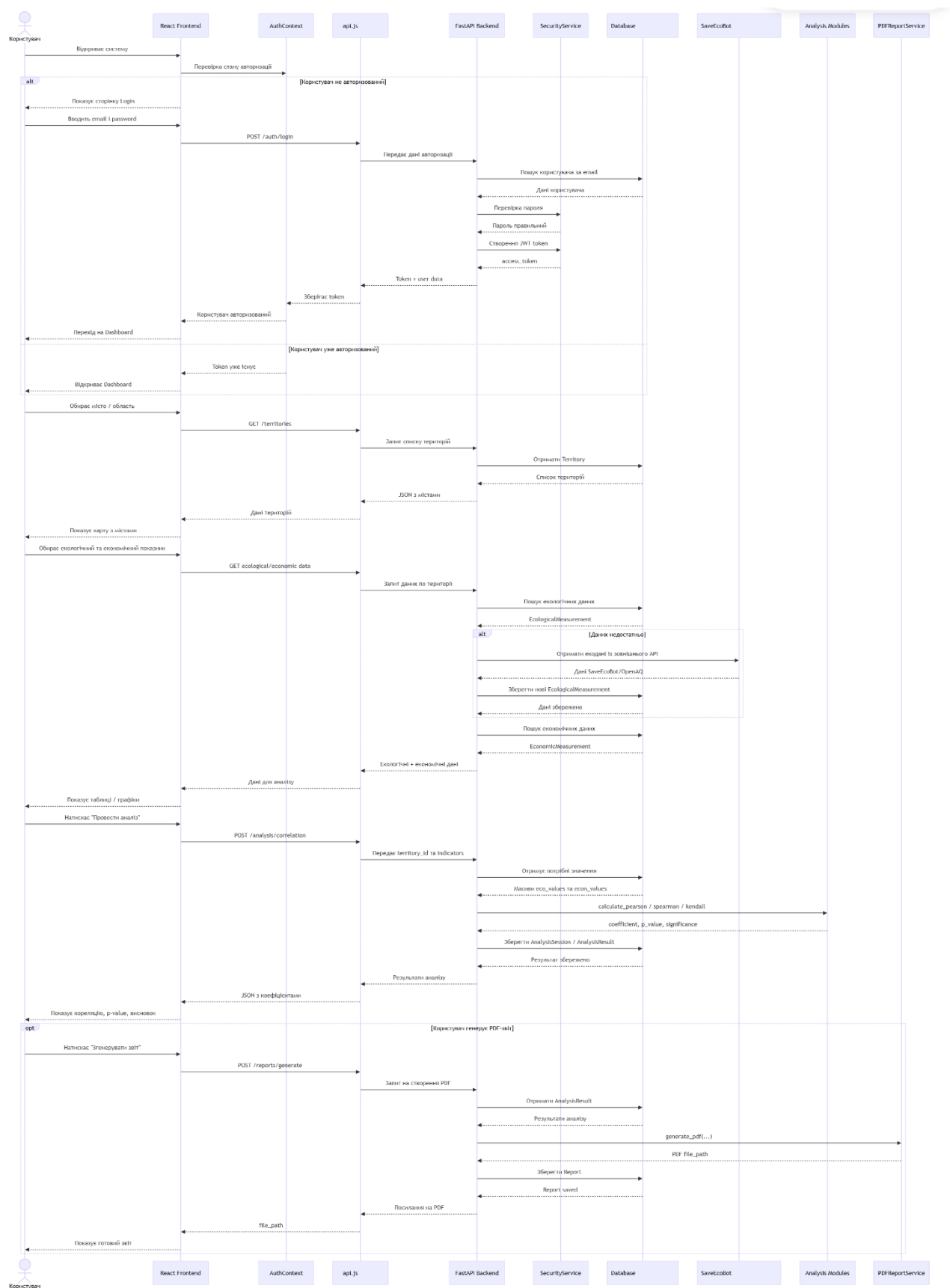


Рисунок 3.4 — Діаграма послідовності

Тож така архітектура забезпечує незалежність рівнів системи: заміна фронтенд-фреймворку або міграція на іншу СУБД не вимагає глобальних змін, а додавання нового методу аналізу зводиться до написання окремого модуля і реєстрації маршруту без втручання в існуючий код.

## 3.2 Схеми зберігання даних в системі

Структура бази даних проєктувалась під дві основні вимоги: ефективна вибірка часових рядів по конкретній території і гнучке зберігання результатів різнотипних аналітичних сесій без розростання схеми. PostgreSQL обрано як основну СУБД. Схеми містять десять таблиць, між якими встановлені зовнішні ключі для забезпечення цілісності даних.

Таблиця `territories` зберігає назву регіону, його код, координати (`latitude`, `longitude`) і опис. Координати використовуються сервісом `SaveEcoBotService` для пошуку найближчої станції моніторингу при відсутності даних безпосередньо по обраному місту — відстань між точками розраховується через формулу гаверсинуса. Всі таблиці вимірювань і аналітичних сесій прив'язані до конкретної території через `territory_id`, що дозволяє вибирати дані одним запитом з індексом замість повного сканування таблиці (рисунок 3.5).

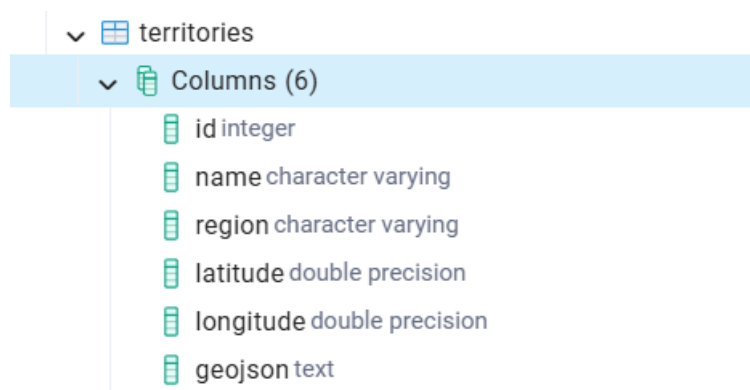


Рисунок 3.5 — Структура таблиці `territories` у pgAdmin

Екологічні вимірювання зберігаються в таблиці `ecological_measurements`

(рисунок 3.6). Кожен запис містить `territory_id`, `indicator_type` (перелічувальний тип: `turbidity`, `PM2.5`, `PM10`, `AQI`, `temperature`, `humidity`, `pressure`), числове значення `value`, одиницю вимірювання `unit`, часову мітку `year` і джерело даних `source` (`EcoDataSource`: `csv` або `saveecobot`). Окреме поле `source` важливе для аудиту: при розбіжності результатів аналізу можна перевірити, чи не змішуються дані з різних джерел з різною методологією вимірювання. Аналогічну структуру має таблиця `economic_measurements`, де `indicator_type` приймає значення `gdp`, `salary`, `unemployment`, `enterprises`, `healthcare`, `investments`, а `source` розрізняє `csv` і `manual`.

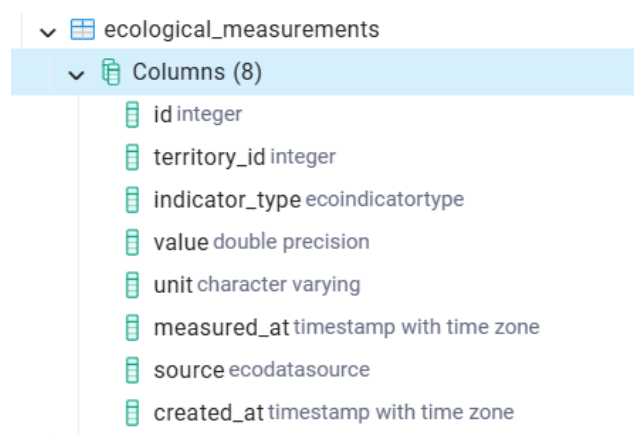


Рисунок 3.6 — Структура таблиці `ecological_measurements` у pgAdmin

Таблиця `users` зберігає облікові дані користувачів: `email`, хешований пароль (`bcrypt`), повне ім'я, роль (`guest`, `analyst`, `admin`) і часові мітки створення та оновлення запису (рисунок 3.7). Паролі зберігаються виключно у хешованому вигляді — `SecurityService` використовує `bcrypt` з автоматичним солінням, що унеможливорює відновлення оригінального пароля навіть при компрометації бази. Роль визначає набір доступних маршрутів API через систему залежностей `FastAPI`.

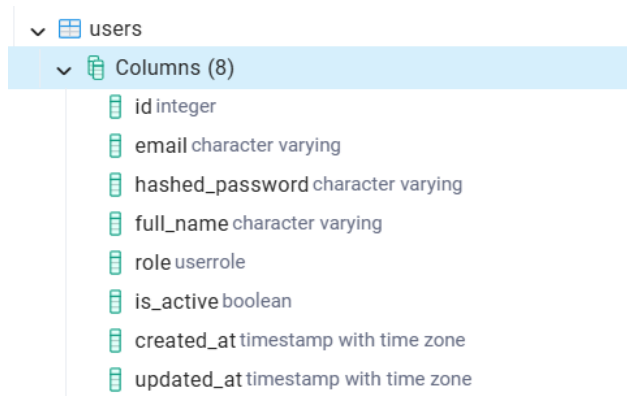


Рисунок 3.7 — Структура таблиці users у pgAdmin

Таблиця `invite_codes` реалізує запрошувальну модель реєстрації (рисунок 3.8). Новий користувач може зареєструватись тільки з дійсним кодом запрошення, який генерує адміністратор. Запис містить сам код, прапорець `is_used` і посилання на акаунт, який його використав (`used_by_email`), щоб обмежувати несанкціонований доступ до системи без повного закриття реєстрації.

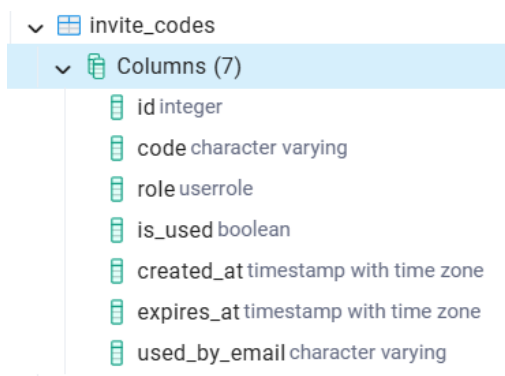


Рисунок 3.8 — Структура таблиці invite\_codes у pgAdmin

Центральна аналітична таблиця — `analysis_results` (рисунок 3.9). Кожен запуск аналізу створює новий запис із полями: `session_id`, `user_id`, `method` (тип аналізу: `pearson`, `spearman`, `partial`, `timeseries`, `regression`, `comparison`). Таблиця `analysis_sessions` виступає як окреме сховище фінальних результатів для швидкого доступу без розбору JSONB-поля (рисунок 3.10) [18].

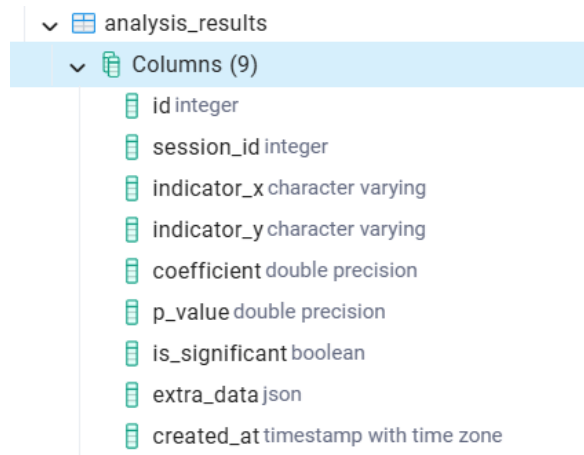


Рисунок 3.9 — Структура таблиці analysis\_results у pgAdmin

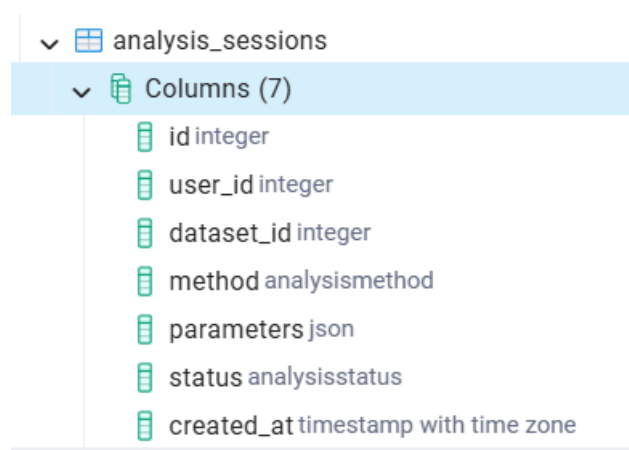
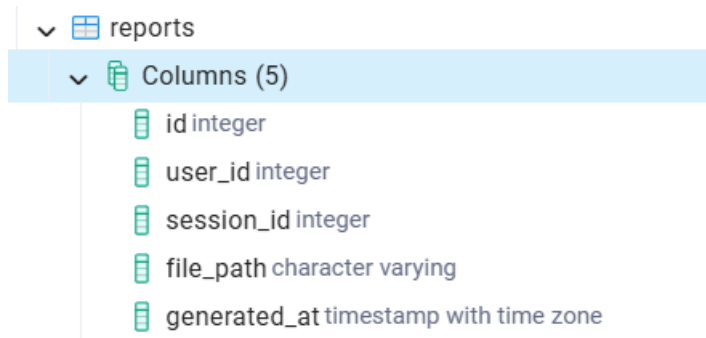


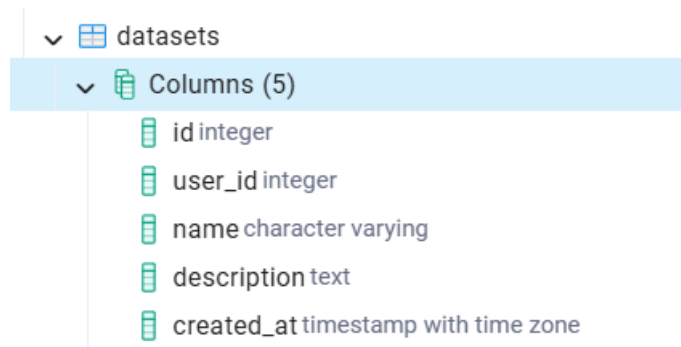
Рисунок 3.10 — Структура таблиці analysis\_sessions у pgAdmin

Таблиця reports пов'язана з analysis\_sessions і зберігає метадані згенерованих PDF-звітів: session\_id, user\_id, шлях до файлу на диску (file\_path) і часову мітку генерації (рисунок 3.11). Самі файли зберігаються у директорії reports/ на сервері, а не в базі даних, бо зберігання бінарних файлів у БД суттєво збільшує її розмір і уповільнює резервне копіювання без відчутних переваг для цього сценарію. Таблиця datasets (рисунок 3.12) зберігає метадані завантажених наборів даних, system\_logs (рисунок 3.13) фіксує системні події для адміністративного моніторингу.



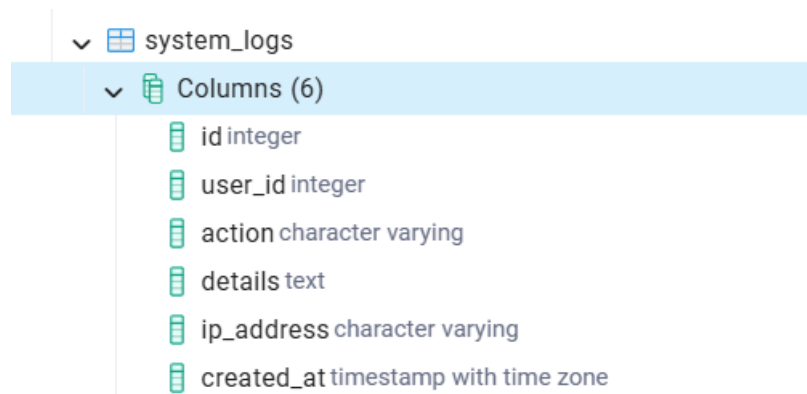
|   |                                       |
|---|---------------------------------------|
| ▼ | reports                               |
| ▼ | Columns (5)                           |
|   | id integer                            |
|   | user_id integer                       |
|   | session_id integer                    |
|   | file_path character varying           |
|   | generated_at timestamp with time zone |

Рисунок 3.11 — Структура таблиці reports у pgAdmin



|   |                                     |
|---|-------------------------------------|
| ▼ | datasets                            |
| ▼ | Columns (5)                         |
|   | id integer                          |
|   | user_id integer                     |
|   | name character varying              |
|   | description text                    |
|   | created_at timestamp with time zone |

Рисунок 3.12 — Структура таблиці datasets у pgAdmin



|   |                                     |
|---|-------------------------------------|
| ▼ | system_logs                         |
| ▼ | Columns (6)                         |
|   | id integer                          |
|   | user_id integer                     |
|   | action character varying            |
|   | details text                        |
|   | ip_address character varying        |
|   | created_at timestamp with time zone |

Рисунок 3.13 — Структура таблиці system\_logs у pgAdmin

Міграції схеми керуються через Alembic. Кожна зміна структури фіксується як окремий файл міграції з функціями `upgrade()` і `downgrade()`. Початкова міграція створює всі десять таблиць і встановлює індекси на `territory_id` у таблицях вимірювань і на поле `year` для прискорення вибірки по діапазону дат.



### 3.3 Методи статистичного аналізу

Аналітичне ядро системи реалізує п'ять методів статистичного аналізу, вибір між якими залежить від характеру вхідних даних і дослідницької задачі. Усі методи реалізовані в окремих модулях серверної частини з використанням бібліотек NumPy і SciPy.

**Коефіцієнт кореляції Пірсона** вимірює силу і напрямок лінійного зв'язку між двома кількісними змінними. Значення коефіцієнта знаходиться в діапазоні від  $-1$  до  $+1$ , де  $-1$  відповідає точній оберненій лінійній залежності,  $+1$  — точній прямій,  $0$  — відсутності лінійного зв'язку. Коефіцієнт розраховується за формулою:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \cdot \sum_{i=1}^n (y_i - \bar{y})^2}},$$

де  $x_i$ ,  $y_i$  — значення двох аналізованих показників;  $\bar{x}$ ,  $\bar{y}$  — їх середні значення;  $n$  — кількість спостережень.

Статистична значущість коефіцієнта перевіряється через  $t$ -статистику:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}},$$

яка при виконанні нульової гіпотези про відсутність кореляції підпорядковується  $t$ -розподілу з  $n - 2$  ступенями свободи. При аналізі матриці кореляцій для кількох пар показників одночасно система застосовує поправку Бонфероні — рівень значущості  $\alpha$  ділиться на кількість порівнянь  $m$ , що знижує ймовірність хибнопозитивних результатів при множинному тестуванні [13, 15].

**Коефіцієнт кореляції Спірмена** є непараметричним аналогом коефіцієнта Пірсона. Замість вихідних значень показників використовуються їх ранги, що робить метод стійким до викидів і придатним для даних, які не відповідають нормальному розподілу. Для екологічних часових рядів, де окремі аномальні вимірювання є нормою, Спірмен дає надійніший результат. Коефіцієнт розраховується як:

$$r_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2-1)},$$

де  $d_i = rg(x_i) - rg(y_i)$  — різниця рангів  $i$ -го спостереження для двох змінних;  $n$  — кількість спостережень. При наявності зв'язаних рангів

застосовується загальна формула через матрицю рангів, аналогічна формулі Пірсона, але обчислена для рангових значень [2].

**Часткова кореляція** дозволяє оцінити зв'язок між двома змінними при фіксованому впливі третьої — контрольної. У контексті еколого-економічного аналізу це важливо: наприклад, кореляція між рівнем PM2.5 і безробіттям може бути зумовлена спільним впливом загального рівня індустріалізації регіону, а не прямим причинно-наслідковим зв'язком. Часткова кореляція між  $x_i$  упри контролі зрозраховується як:

$$r_{xy \cdot z} = \frac{r_{xy} - r_{xz} \cdot r_{yz}}{\sqrt{(1 - r_{xz}^2)(1 - r_{yz}^2)}},$$

де  $r_{xy}$ ,  $r_{xz}$ ,  $r_{yz}$  — парні коефіцієнти кореляції між відповідними парами змінних. Реалізація в модулі `partial.py` підтримує як коефіцієнт Пірсона, так і Спірмена як базу для розрахунку парних кореляцій [14].

**Лінійна регресія** будує математичну модель залежності між залежною змінною  $y$  і одним або кількома предикторами. У системі реалізовано просту і множинну лінійну регресію методом найменших квадратів. Для простої регресії рівняння має вигляд:

$$y = \beta_0 + \beta_1 x + \varepsilon,$$

де  $\beta_0$  — вільний член (зміщення),  $\beta_1$  — коефіцієнт нахилу,  $\varepsilon$  — залишковий член. Коефіцієнти оцінюються як:

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \beta_0 = \bar{y} - \beta_1 \bar{x}.$$

Якість моделі оцінюється через коефіцієнт детермінації  $R^2$ , який показує частку дисперсії залежної змінної, поясненої моделлю:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

де  $\hat{y}_i$  — значення, передбачені моделлю. Значення  $R^2$  близьке до 1 свідчить про високу пояснювальну здатність моделі [2, 17].

**Аналіз часових рядів** реалізований у модулі `timeseries.py` і включає декомпозицію ряду на компоненти і розрахунок автокореляційної функції. Лінійний тренд оцінюється методом найменших квадратів як функція від часу  $t$ :

$$\hat{y}_t = \beta_0 + \beta_1 t.$$

Для виявлення автокореляції — залежності між значеннями ряду, зсунутими на  $k$  кроків — розраховується автокореляційна функція:

$$r_k = \frac{\sum_{t=1}^{n-k} (y_t - \bar{y})(y_{t+k} - \bar{y})}{\sum_{t=1}^n (y_t - \bar{y})^2},$$

де  $k$  — лаг (зсув у часі). Висока автокореляція при малих лагах свідчить про наявність тренду або сезонності в ряді, що необхідно враховувати при інтерпретації кореляцій між двома часовими рядами — некоректне порівняння двох трендових рядів може давати хибно високі кореляції без реального причинно-наслідкового зв'язку [1].

Перед застосуванням будь-якого з описаних методів дані проходять попередню обробку: лінійна інтерполяція заповнює пропущені значення, метод міжквартильного розмаху (IQR) виявляє і замінює викиди медіаною вибірки, щоб була забезпечена коректність статистичних розрахунків незалежно від якості вхідних даних [16].

### 3.4 Реалізація серверної частини

Серверна частина системи розгорнута як FastAPI-додаток із точкою входу в модулі `main.py`. При ініціалізації застосунок реєструє дев'ять маршрутизаторів, налаштовує `middleware` для обробки міждоменних запитів і створює таблиці бд через SQLAlchemy, якщо вони ще не існують. Дозволені джерела запитів визначені явно: `localhost:3000` для локальної розробки і задеплоєний фронтенд на Render.

Кожен маршрутизатор відповідає за окрему функціональну область і реєструється з префіксом: `/auth`, `/users`, `/territories`, `/ecological`, `/economic`, `/datasets`, `/analysis`, `/reports`, `/invites`. Одразу по URL видно, яка частина системи обробляє запит, і відповідальність між модулями розмежована без конфліктів маршрутів (рисунок 3.14).

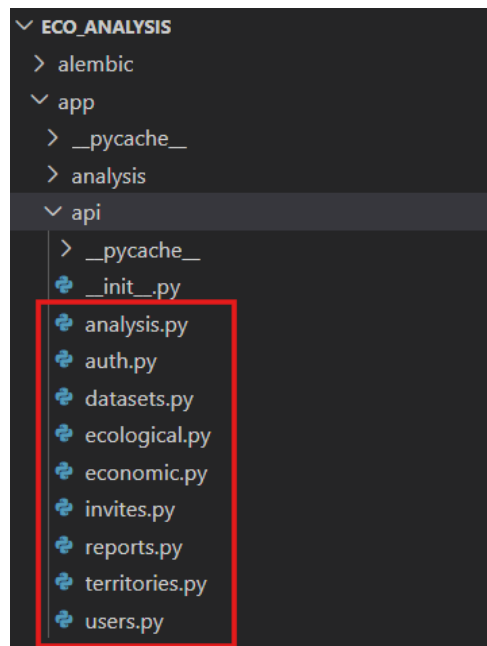


Рисунок 3.14 — Структура серверної частини проекту

Автентифікація реалізована через токени доступу у форматі JWT — підписані рядки, що містять зашифровану інформацію про користувача: його ідентифікатор, електронну пошту і роль у системі. При успішному вході маршрутизатор `/auth/login` перевіряє пароль через `SecurityService`, який порівнює введені значення із захешованим записом у базі, і повертає токен із вбудованими даними користувача та терміном дії. На захищених маршрутах FastAPI-залежність `get_current_user` розшифровує токен із заголовка запиту, витягує користувача з бази і передає його об'єкт у функцію-обробник. Якщо токен прострочений або підроблений, то повертається помилка 401 без виконання будь-якої бізнес-логіки.

Реєстрація нових користувачів захищена інвайт-кодами. Маршрут `/auth/register` приймає необов'язковий код запрошення, перевіряє його наявність у базі, факт використання і термін дії. Роль нового користувача визначається роллю, вбудованою в код запрошення: адміністратор може видавати запрошення як для аналітиків, так і для інших адміністраторів.

Центральний маршрутизатор системи — `analysis.py`. Він реєструє шість ендпоінтів: `/analysis/correlation`, `/analysis/partial`, `/analysis/timeseries`, `/analysis/regression`, `/analysis/multiple_regression` і `/analysis/comparison`. Кожен

приймає схему запиту з обов'язковими полями: ідентифікатор території, назва екологічного і економічного показника. Рудantic автоматично перевіряє типи і наявність полів до виконання будь-якого коду — некоректний запит повертає помилку 422 з детальним описом того, що саме не так (рисунок 3.15).

### Веб-система кореляційного аналізу 1.0.0 QAS 3.1

openapi.json

Аналіз екологічних та економічних показників території

#### Авторизація

POST /auth/register Register

POST /auth/login Login

GET /auth/me Get Me

#### Користувачі

GET /users/ Get Users

GET /users/{user\_id} Get User

PATCH /users/{user\_id}/role Update User Role

PATCH /users/{user\_id}/deactivate Deactivate User

PATCH /users/{user\_id}/activate Activate User

#### Території

GET /territories/ Get Territories

POST /territories/ Create Territory

Рисунок 3.15 — Автоматично згенерована документація API у Swagger UI

Перед передачею даних в аналітичний модуль маршрутизатор вибирає вимірювання з бази з фільтром по ідентифікатору території і типу показника. Для деяких територій дані можуть бути відсутні безпосередньо, тому реалізована функція пошуку суміжних територій, вона шукає записи через поле регіону і розширює вибірку, що покриває випадки, коли дані завантажені по регіону в цілому, а запит виконується по конкретному місту в межах цього регіону.

Збір екологічних даних з SaveEcoBot реалізований у відповідному сервісі. Сервіс надсилає запит до публічного API SaveEcoBot в асинхронному режимі і отримує масив станцій моніторингу з поточними вимірюваннями. Для кожної станції витягуються координати через допоміжну функцію, яка перебирає кілька можливих назв полів (це необхідно через непослідовність формату відповіді зовнішнього сервісу). Пошук станцій по обраній території реалізований двоетапно:

спочатку шукаються збіги по назві міста в полях назви станції, а якщо нічого не знайдено — розраховується відстань за формулою гаверсинуса до всіх станцій і повертається найближча. Знайдені вимірювання нормалізуються і зберігаються в таблиці екологічних даних із позначкою джерела saveecobot.

Імпорт даних із CSV-файлів обробляє сервіс імпорту даних. При завантаженні екологічного файлу сервіс читає його через pandas, нормалізує назви стовпців через словник відповідностей, який містить понад 30 варіантів написання кожного показника, і зіставляє рядки з відповідними записами територій через функцію нормалізації назв регіонів. Аналогічно працює імпорт економічних даних. Така нормалізація дозволяє завантажувати файли Держстату без попереднього перейменування стовпців вручну.

Генерація PDF-звітів реалізована через бібліотеку ReportLab. Звіт будується як послідовність параграфів, таблиць і відступів із кастомними стилями у вигляді зеленої кольорової схеми, шрифту DejaVuSans для підтримки кирилиці, кольорового кодування рядків таблиці залежно від значущості результату: зелений фон для статистично значущих кореляцій, жовтий для граничних, червоний для незначущих. Готовий файл зберігається на диску, шлях до нього фіксується в таблиці звітів (рисунок 3.16).

#### PDF-звіт кореляційного аналізу

|                        |                  |
|------------------------|------------------|
| Територія              | Львів            |
| Екологічний показник   | Якість води      |
| Економічний показник   | Підприємства     |
| Кількість спостережень | 36               |
| Дата генерації         | 02.06.2026 14:37 |

##### 1. Кореляційний аналіз

| Метод   | Коефіцієнт | p-value | Значущість | Інтерпретація                   |
|---------|------------|---------|------------|---------------------------------|
| Пірсон  | 0.1095     | 0.525   | Ні         | дуже слабка позитивна кореляція |
| Спірмен | 0.0774     | 0.6538  | Ні         | дуже слабка позитивна кореляція |
| Кендалл | 0.079      | 0.5325  | Ні         | дуже слабка позитивна кореляція |

| Показник     | Результат                                                          | p-value |
|--------------|--------------------------------------------------------------------|---------|
| Екологічний  | розподіл не нормальний                                             | 0.0     |
| Економічний  | розподіл не нормальний                                             | 0.0055  |
| Рекомендація | Обидва розподіли не нормальні - рекомендується Спірмен або Кендалл |         |

##### 2. Часткова кореляція

| Показник          | Значення |
|-------------------|----------|
| Контрольна змінна | PM2.5    |
| Коефіцієнт        | 0.1207   |

Рисунок 3.16 — Приклад згенерованого PDF-звіту системи

Підключення до бази даних реалізоване через SQLAlchemy з патерном передачі залежностей: функція `get_db` створює сесію, передає її в обробник запиту і гарантує закриття сесії після завершення незалежно від результату через блок захищеного виконання.

### 3.5 Реалізація клієнтської частини

Клієнтська частина системи реалізована як односторінковий застосунок на React із маршрутизацією між сторінками без перезавантаження браузера. Інтерфейс побудований у зеленій кольоровій схемі з темним навігаційним рядком угорі, що містить посилання на основні розділи: Дашборд, Дані, Аналіз, Звіти (рисунок 3.17). Для авторизованих користувачів із роллю адміністратора відображається додаткова кнопка Адмін, виділена жовтим кольором. Усі сторінки доступні через єдиний навігаційний рядок, що залишається незмінним між переходами.

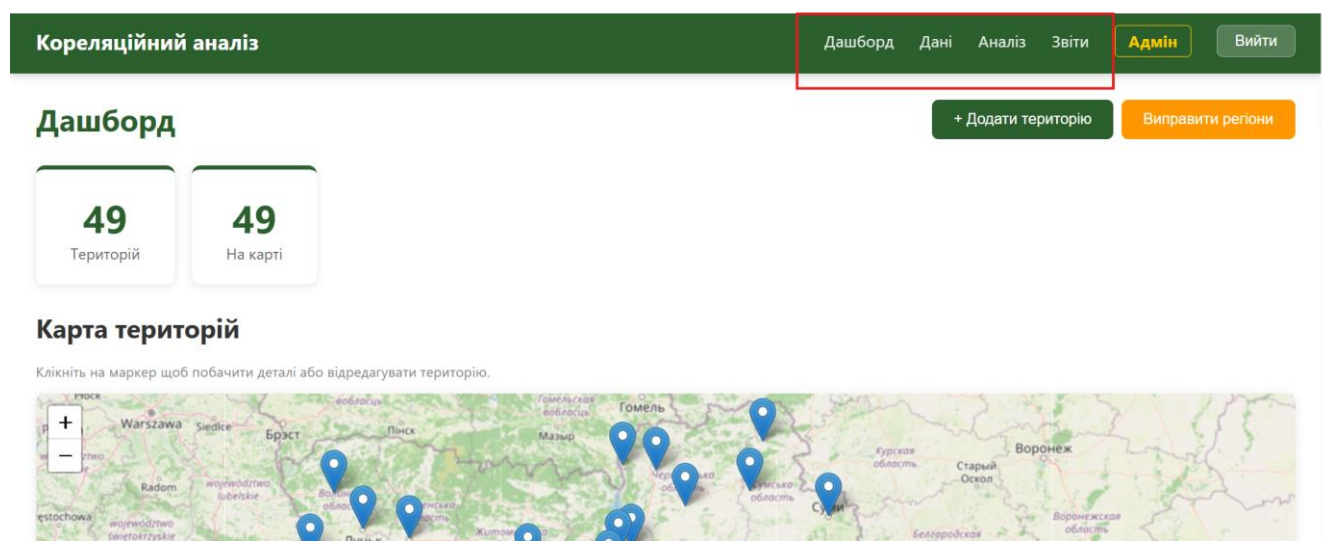


Рисунок 3.17 — Навігаційний рядок системи

Стан автентифікації зберігається в контексті застосунку через `AuthContext`. Після успішного входу токен доступу записується в пам'ять застосунку і автоматично додається до заголовка кожного запиту до серверу через перехоплювач у модулі `api.js`. Захищені маршрути перевіряють наявність токена і

перенаправляють на сторінку входу, якщо користувач не авторизований. Роль користувача визначає, які елементи інтерфейсу відображаються, як от наприклад кнопка адміністративної панелі, що видима тільки адміністраторам.

Публічний дашборд — це перша сторінка, яку бачить неавторизований відвідувач (рисунок 3.18). У верхній частині розташований банер із назвою системи і двома кнопками: увійти та зареєструватись. Нижче відображаються три лічильники: кількість територій у базі, кількість екологічних вимірювань і економічних показників. Далі йде блок із переліком можливостей системи у вигляді карток: кореляційний аналіз, часткова кореляція, часові ряди, регресійний аналіз, геовізуалізація, генерація PDF-звітів. Завершує сторінку сітка доступних територій, а саме картки з назвою міста, областю і координатами. Сторінка повністю доступна без реєстрації і виконує роль публічного інформаційного ресурсу для ознайомлення з можливостями системи.

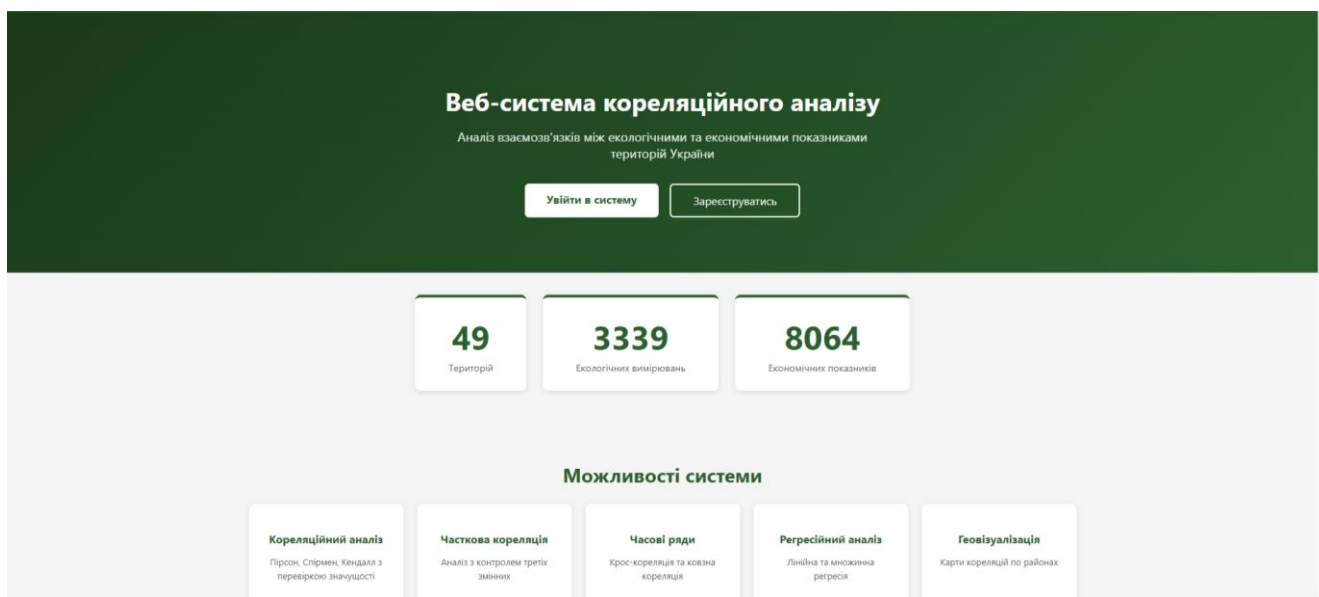


Рисунок 3.18 — Публічний дашборд системи з лічильниками і переліком можливостей

Дашборд аналітика відкривається після входу в систему. Він містить форму додавання нової території, туди користувач вводить назву міста, після чого система автоматично підставляє область і координати через геокодування. Нижче



розташована інтерактивна карта на основі OpenStreetMap із маркерами всіх збережених територій — кожен маркер можна натиснути для перегляду або редагування деталей. Під картою відображається список усіх територій у вигляді карток із кнопками редагування і видалення. (рисунок 3.19)

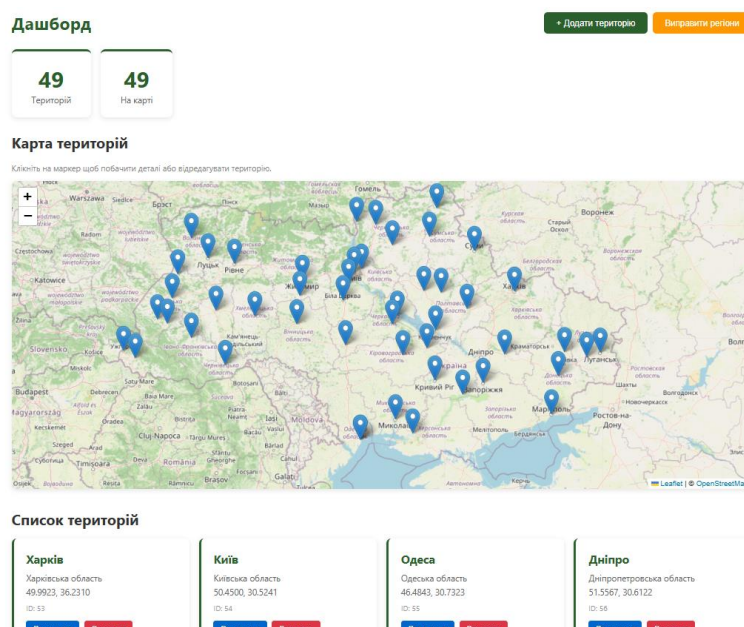


Рисунок 3.19 — Дашборд аналітика з інтерактивною картою територій України

Сторінка завантаження даних розділена на три функціональні блоки. Перший блок — завантаження екологічних даних у форматі CSV, де підтримуються показники якості води, PM2.5, PM10, індекс якості повітря, температура, вологість і тиск. Другий блок — завантаження економічних даних CSV із показниками ВРП, зарплати, безробіття, кількості підприємств, охорони здоров'я та інвестицій. Третій блок — збір даних через SaveEcoBot API: користувач вводить назву міста, і система автоматично знаходить найближчу станцію моніторингу і завантажує актуальні вимірювання. Нижче відображається зведена таблиця наявних даних у базі з розбивкою по територіях; для кожної території окремо показані екологічні і економічні показники з кількістю записів і датою останнього оновлення. Записи можна видаляти по одному прямо з цієї таблиці. (рисунок 3.20)

## Завантаження даних

### Екологічні дані (CSV)

Якість води, PM2.5, PM10, індекс якості повітря, температура, вологість, тиск

Вибрати CSV

### Економічні дані (CSV)

ВРП, зарплата, безробіття, підприємства, охорона здоров'я, інвестиції

Вибрати CSV

### SaveEcoBot API

Якість повітря: PM2.5, PM10, AQI, температура, вологість, тиск

Введіть будь-яке місто України. Якщо станції немає — система знайде найближчу і залише дані до всіх міст цієї області.

Назва міста (Чернігів, Тростянець, Жашків...)

Завантажити дані

### Дані в базі

3 339  
Екологічних

8 064  
Економічних

47  
Регіонів

Сортування:  
За назвою (А-Я)

Пошук регіону:  
Харків, Одеська...

Екологічні дані

Економічні дані

#### Алчевськ

|                       |    |            |   |                  |    |         |   |
|-----------------------|----|------------|---|------------------|----|---------|---|
| Температура           | 5  | 2022-02-28 | × | Підприємства     | 9  | 2023    | × |
| Індекс якості повітря | 3  | 2022-02-28 | × | Інвестиції       | 44 | 2025-Q4 | × |
| Тиск                  | 4  | 2020-05-22 | × | Зарплата         | 85 | 2022 01 | × |
| Вологість             | 5  | 2022-02-28 | × | Безробіття       | 7  | 2021    | × |
| PM10                  | 4  | 2022-02-28 | × | Охорона здоров'я | 4  | 2020    | × |
| Якість води           | 14 | 2026-04-01 | × | ВРП              | 19 | 2021    | × |
| PM2.5                 | 4  | 2022-02-28 | × |                  |    |         |   |

#### Бахмут

|                       |   |            |   |              |    |      |   |
|-----------------------|---|------------|---|--------------|----|------|---|
| Індекс якості повітря | 3 | 2022-03-13 | × | Підприємства | 9  | 2023 | × |
| PM2.5                 | 4 | 2022-03-13 | × | ВРП          | 19 | 2021 | × |

Рисунок 3.20 — Сторінка завантаження даних із трьома блоками імпорту

Сторінка аналізу — це центральна частина системи. У верхній частині розташована панель налаштувань: випадаючий список для вибору території, вибір методу аналізу, вибір екологічного і економічного показника. Доступні методи: кореляція (Пірсон, Спирмен, Кендалл), часткова кореляція, аналіз часових рядів, лінійна регресія, множинна регресія і порівняння методів. При виборі часткової кореляції з'являється додатковий елемент, а саме вибір контрольної змінної. При множинній регресії замість одного екологічного показника відображається набір прапорців для вибору кількох предикторів. Для деяких показників над панеллю відображається інформаційне повідомлення про особливості наявності даних (наприклад, що дані по якості води є лише для річкових постів моніторингу і при відсутності по конкретному місту система використає найближчий регіон) (рисунок 3.21).

Кореляційний аналіз

Увага щодо наявності даних:  
Якість води: Дані по водним ресурсам є не для всіх міст — лише для річкових постів моніторингу. Якщо для обраного міста даних немає, система використає найближчий регіон.  
Охорона здоров'я: Дані по Україні в цілому розподілені рівномірно між регіонами. Є для всіх 25 регіонів, роки 2017-2020.

Територія

Метод аналізу

Екологічний показник

Економічний показник

— Оберіть —

Кореляція (Пірсон/Спірмен/Кендалл)

Якість води \*

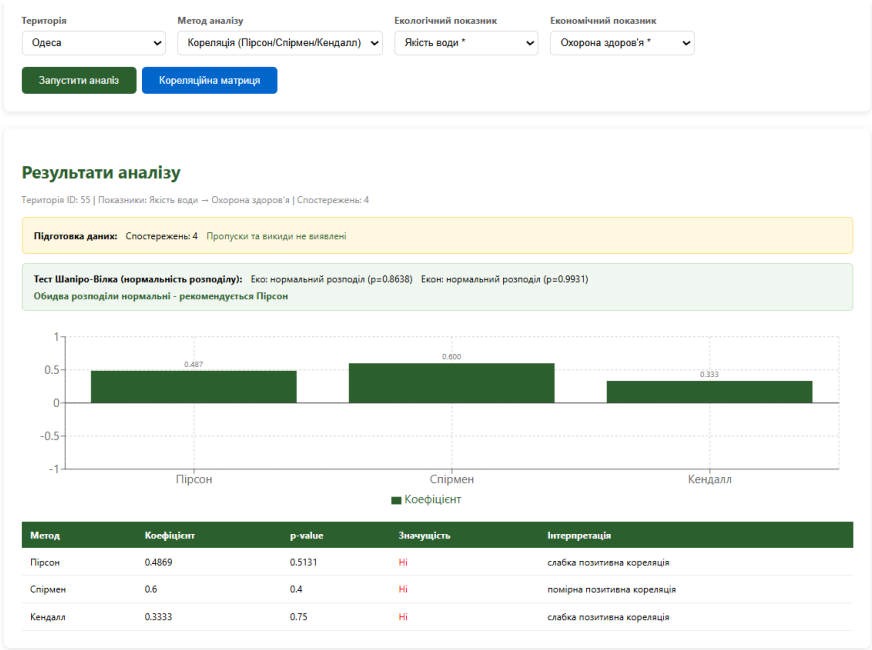
Охорона здоров'я \*

Запустити аналіз

Кореляційна матриця

Рисунок 3.21 — Панель налаштувань сторінки аналізу

Після запуску аналізу результати відображаються безпосередньо під панелью налаштувань без переходу на іншу сторінку. Для кореляційного аналізу результат містить: інформацію про підготовку даних (кількість спостережень, кількість заміненних викидів методом міжквартильного розмаху), результати тесту Шапіро-Вілка на нормальність розподілу з рекомендацією щодо вибору методу, стовпчасту діаграму коефіцієнтів для трьох методів одночасно і таблицю з числовими значеннями коефіцієнтів, рівнів значущості і текстовою інтерпретацією (рисунок 3.22).



Результати аналізу

Територія ID: 55 | Показники: Якість води → Охорона здоров'я | Спостережень: 4

Підготовка даних: Спостережень: 4 Пропуски та викиди не виявлені

Тест Шапіро-Вілка (нормальність розподілу): Еко: нормальний розподіл (p=0.8638) Еконс: нормальний розподіл (p=0.9931)  
Обидва розподіли нормальні - рекомендується Пірсон

| Метод   | Коефіцієнт | p-value | Значущість | Інтерпретація               |
|---------|------------|---------|------------|-----------------------------|
| Пірсон  | 0.4869     | 0.5131  | Ні         | слабка позитивна кореляція  |
| Спірмен | 0.6        | 0.4     | Ні         | помірна позитивна кореляція |
| Кендалл | 0.3333     | 0.75    | Ні         | слабка позитивна кореляція  |

Рисунок 3.22 — Результати кореляційного аналізу (Пірсон, Спірмен, Кендалл)

Для часткової кореляції результат компактніший: коефіцієнт, рівень значущості, прапорець значущості, назва контрольної змінної і текстова інтерпретація (рисунок 3.23).

### Кореляційний аналіз

**Увага щодо наявності даних:**  
**Якість води:** Дані по водним ресурсам є не для всіх міст — лише для річкових постів моніторингу. Якщо для обраного міста даних немає, система використає найближчий регіон.  
**Охорона здоров'я:** Дані по Україні в цілому розподілені рівномірно між регіонами. Є для всіх 25 регіонів, роки 2017-2020.

Територія

Метод аналізу

Екологічний показник

Економічний показник

Контрольна змінна

Харків

Часткова кореляція

Якість води \*

Охорона здоров'я \*

PM10

Запустити аналіз

Кореляційна матриця

**Результати аналізу**  
Територія ID: 53 | Показники: Якість води → Охорона здоров'я | Спостережень: 4

**Метод:** Часткова кореляція  
**Коефіцієнт:** -0.4628  
**p-value:** 0.5372  
**Значущість:** Ні  
**Контрольна змінна:** PM10  
**Інтерпретація:** помірна негативна часткова кореляція

Рисунок 3.23 — Результати часткової кореляції з контрольною змінною

Аналіз часових рядів повертає два графіки: крос-кореляційну функцію, що показує силу зв'язку при різних часових зсувах між рядами, і ковзну кореляцію, що відображає як змінюється сила зв'язку в часі при заданому розмірі вікна. Над графіками виводяться числові результати: оптимальний лаг і значення крос-кореляції при цьому лагу, середня ковзна кореляція (рисунок 3.24).

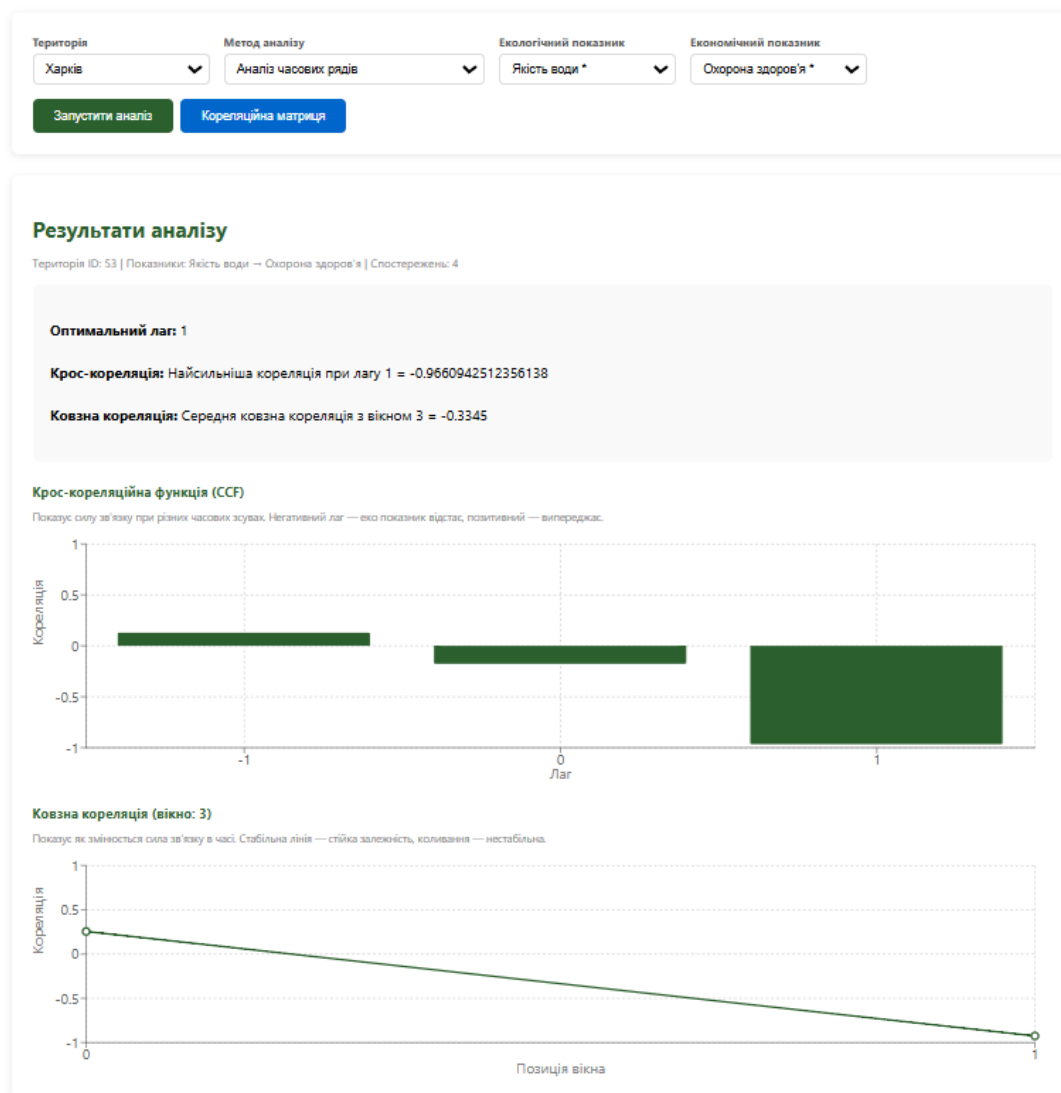


Рисунок 3.24 — Результати аналізу часових рядів: крос-кореляційна функція і ковзна кореляція

Лінійна регресія виводить рівняння залежності, коефіцієнт детермінації, F-статистику, рівень значущості моделі і стандартну похибку. Нижче відображаються два графіки: діаграма розсіювання з нанесеною лінією регресії і графік залишків для візуальної оцінки якості моделі (рисунок 3.25).

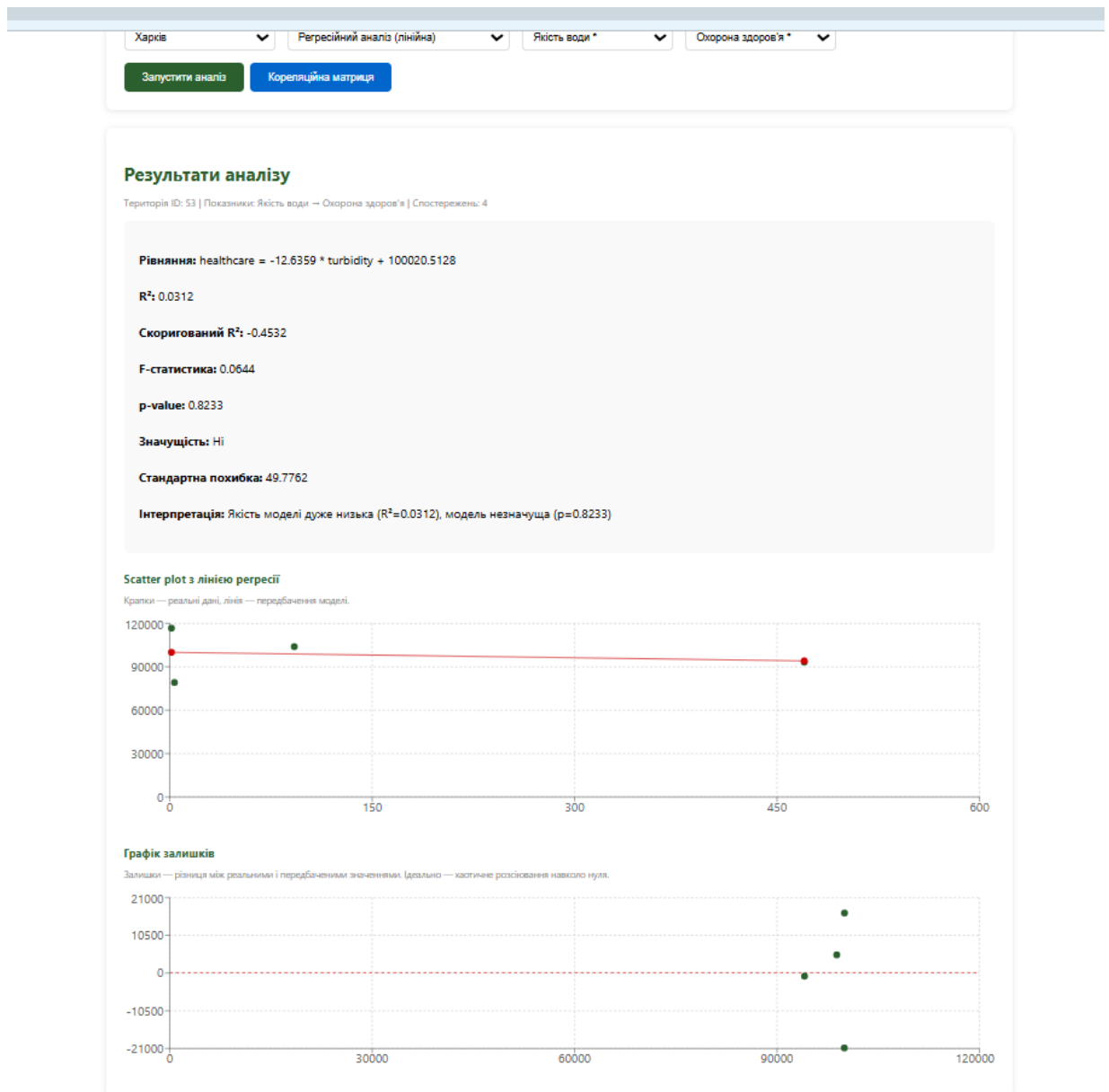


Рисунок 3.25 — Результати лінійного регресійного аналізу з діаграмою розсіювання та графіком залишків

Множинна регресія виводить  $R^2$ , скоригований  $R^2$ , кількість спостережень і таблицю коефіцієнтів для кожного предиктора окремо (рисунок 3.26).

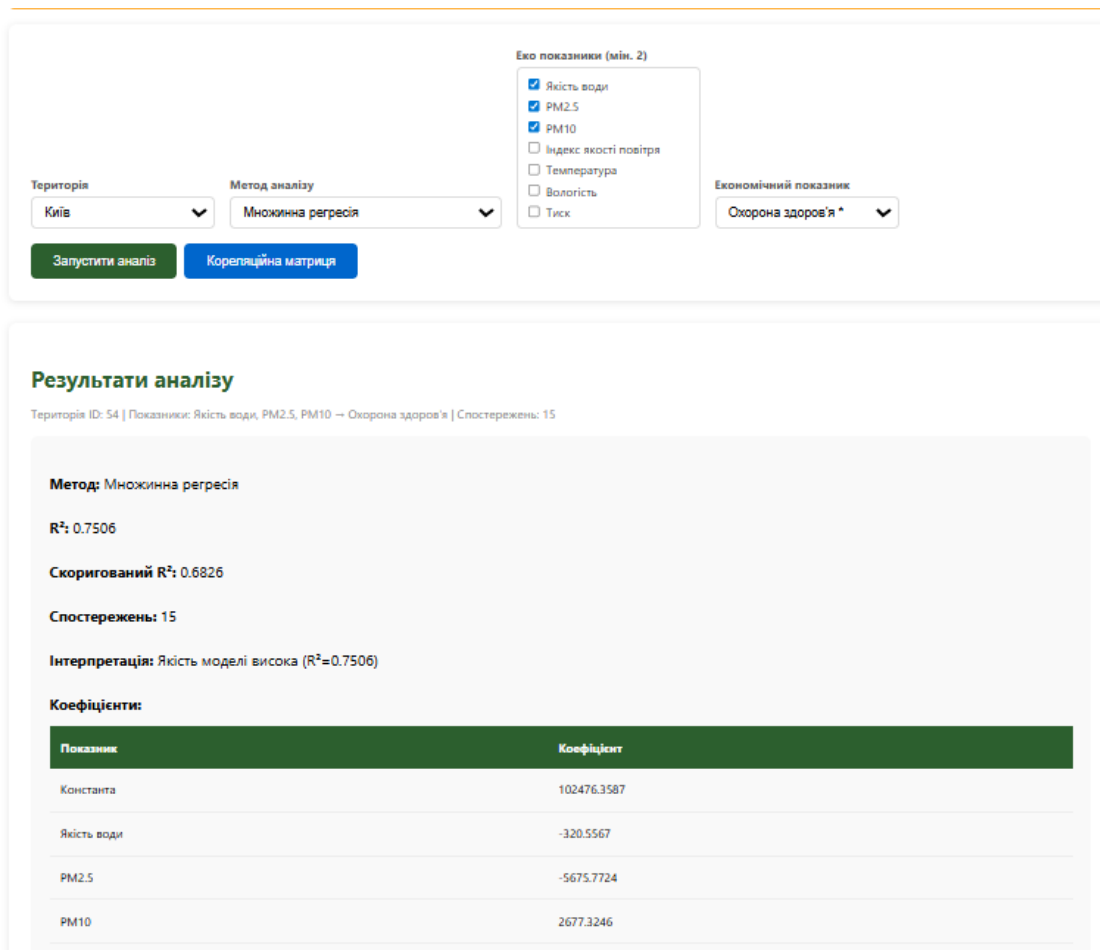


Рисунок 3.26 — Результати множинної регресії з таблицею коефіцієнтів

Порівняння методів повертає оцінку узгодженості між Пірсоном, Спірменом і Кендаллом та текстову рекомендацію. Кнопка «Кореляційна матриця» поруч із запуском аналізу відкриває повну матрицю кореляцій для всіх можливих пар екологічних і економічних показників по обраній території. Матриця відображається у вигляді таблиці з сортуванням за значенням коефіцієнта, фільтрами по групах показників і прапорцем для відображення тільки статистично значущих пар. Рядки підсвічуються зеленим для значущих позитивних кореляцій і рожевим для значущих негативних (рисунок 3.27).

### Кореляційний аналіз

**Увага щодо наявності даних:**

**Якість води:** Дані по водним ресурсам є не для всіх міст — лише для річкових постів моніторингу. Якщо для обраного міста даних немає, система використовує найближчий регіон.

**Охорона здоров'я:** Дані по Україні в цілому розподілені рівномірно між регіонами. Є для всіх 25 регіонів, роки 2017-2020.

Територія

Метод аналізу

Екологічний показник

Економічний показник

Київ

Порівняння методів

Якість води \*

Охорона здоров'я \*

Запустити аналіз

Кореляційна матриця

#### Результати аналізу

Територія ID: 54 | Показники: Якість води → Охорона здоров'я | Спостережень: 24

**Узгодженість методів:** низька

**Деталі:** Є помітні розбіжності між методами

**Рекомендація:** Є суперечності між методами. Варто перевірити викиди, форму залежності та інтерпретувати результат обережно.

Рисунок 3.27 — Результати порівняння методів кореляційного аналізу

Сторінка звітів дозволяє налаштувати і згенерувати PDF-звіт. Користувач обирає територію, основний екологічний і економічний показник, а також набір екологічних показників для множинної регресії. Перед генерацією відображається перелік розділів, що увійдуть у звіт: кореляції Пірсона, Спірмена і Кендалла, часткова кореляція, аналіз часових рядів, лінійна і множинна регресія, порівняння методів (рисунок 3.28).

### Генерація PDF звітів

#### Налаштування звіту

Територія

Екологічний показник для основного аналізу

Економічний показник

Оберть територію

Якість води

ВРП

Екологічні показники для множинної регресії

☐ Якість води

☒ PM2.5

☒ PM10

☐ Індекс якості повітря

☐ Температура

☐ Вологість

☐ Тиск

**Що буде в звіті:**

- Інформація про територію та вибрані показники
- Кореляція Пірсона, Спірмена і Кендалла
- Часткова кореляція з контрольною змінною
- Аналіз часових рядів: крос-кореляція і ковзна кореляція
- Лінійна регресія з рівнянням,  $R^2$ , p-value і залишками
- Множинна регресія з коефіцієнтами,  $R^2$  і VIF
- Порівняння методів, узгодженість, суперечності і ranking

Згенерувати PDF звіт

Рисунок 3.28 — Сторінка налаштування та генерації PDF-звіту



Адміністративна панель має дві вкладки. Перша — управління користувачами: таблиця з електронною поштою, іменем, роллю і статусом кожного акаунта, можливість змінити роль через випадаючий список і заблокувати користувача. Друга вкладка — це генерація запрошувальних кодів: адміністратор обирає роль для нового користувача і термін дії коду в днях, після чого система генерує унікальний рядок. Усі наявні коди відображаються в таблиці зі статусом використання (рисунок 3.29).

### Адміністративна панель

Користувачі (6)

Інвайт-коди (8)

Управління користувачами

| ID | Email                 | Ім'я            | Роль     | Статус   | Дії                        |
|----|-----------------------|-----------------|----------|----------|----------------------------|
| 1  | admin@test.com        | Адміністратор   | Адмін    | Активний | Адмін <span>Блок</span>    |
| 2  | krasotka123@gmail.com | Ванільна Аліна  | Аналітик | Активний | Аналітик <span>Блок</span> |
| 4  | vasil@gmail.com       | Пупкін Василь   | Аналітик | Активний | Аналітик <span>Блок</span> |
| 3  | zhyk@gmail.com        | Жук Вікторія    | ЛПР      | Активний | ЛПР <span>Блок</span>      |
| 5  | kotik@gmail.com       | Кіт Муркіт      | ЛПР      | Активний | ЛПР <span>Блок</span>      |
| 6  | analitik@gmail.com    | Янковський Влад | Аналітик | Активний | Аналітик <span>Блок</span> |

Генерація інвайт-кодів

Роль для коду

Термін дії (днів)

Згенерувати код

Всі коди

| Код                | Роль     | Статус      | Використав | Дії     |
|--------------------|----------|-------------|------------|---------|
| LPR-r1w4ThxRO3U    | ЛПР      | Використано | —          |         |
| ANALYST-3LQnQ5QJkK | Аналітик | Використано | —          |         |
| LPR-9F7dK25k6Y     | ЛПР      | Активний    | —          | Відняти |
| LPR-VXyPUERSQ4Q    | ЛПР      | Активний    | —          | Відняти |
| ADMIN-Cn6CdG2s2cI  | Адмін    | Активний    | —          | Відняти |
| LPR-VBT73NEJ3s6    | ЛПР      | Активний    | —          | Відняти |
| LPR-yFOB3eFhLas    | ЛПР      | Активний    | —          | Відняти |
| LPR-XFw_Zyzn6xE    | ЛПР      | Активний    | —          | Відняти |

Рисунок 3.29 — Адміністративна панель: управління користувачами та інвайт-кодами

Інтерфейс повністю адаптований для роботи в браузері без встановлення додаткового програмного забезпечення. Усі взаємодії з сервером виконуються асинхронно і результати аналізу з'являються на тій самій сторінці без переходів, що скорочує час між запитом і отриманням результату та спрощує порівняння різних методів для одних і тих самих даних.

## 4 МЕТОДИКА РОБОТИ КОРИСТУВАЧА З СИСТЕМОЮ

Цей розділ описує практичну сторону роботи з системою: мінімальні вимоги до середовища, в якому вона запускається, і покроковий сценарій взаємодії користувача з інтерфейсом в деталях (від реєстрації до отримання результатів аналізу і генерації звіту).

### 4.1 Системні вимоги

Система не потребує встановлення жодного додаткового програмного забезпечення на стороні користувача, адже весь функціонал доступний через браузер. Це принципово знижує поріг входу: достатньо пристрою з підключенням до інтернету і будь-яким сучасним браузером.

З боку клієнта мінімальні вимоги такі. Операційна система: будь-яка, що підтримує сучасний браузер: Windows 7 і вище, macOS 10.12 і вище, будь-який актуальний дистрибутив Linux. Браузер: Google Chrome версії 90+, Mozilla Firefox версії 88+, Microsoft Edge версії 90+, Safari версії 14+. Браузери повинні мати увімкнений JavaScript — без нього React-застосунок не запуститься. Оперативна пам'ять: мінімум 2 ГБ, рекомендовано 4 ГБ і більше, оскільки при відображенні великих кореляційних матриць і інтерактивних графіків браузер може споживати до 500 МБ. Роздільна здатність екрану: від 1280×720, інтерфейс адаптований під десктопне відображення. Швидкість інтернет-з'єднання: від 1 Мбіт/с для комфортної роботи; запити до серверу повертають JSON-відповіді обсягом до кількох десятків кілобайт, тому вимоги до каналу невисокі.

З боку серверного середовища, якщо система розгортається локально, вимоги такі. Python версії 3.10 і вище — для запуску FastAPI-бекенду і аналітичних бібліотек. Node.js версії 16 і вище та менеджер пакетів npm для збірки і запуску React-фронтенду. PostgreSQL версії 13 і вище — як основна база даних. Оперативна пам'ять сервера: мінімум 1 ГБ, рекомендовано 2 ГБ і більше, оскільки при розрахунку матриці кореляцій для великої кількості пар показників одночасно

навантаження на пам'ять зростає. Дисковий простір: від 500 МБ для зберігання коду, залежностей і початкових даних; при активному завантаженні CSV-файлів і генерації PDF-звітів потреба в дисковому просторі зростає пропорційно обсягу даних. Docker і Docker Compose опціонально, але рекомендовано для спрощення розгортання: у репозиторії наявний файл `docker-compose.yml`, який автоматично піднімає бекенд, фронтенд і базу даних в ізольованих контейнерах без ручного налаштування середовища.

Для хмарного розгортання, яке використовується у продакшн-версії системи, бекенд розміщений на платформі Render як вебсервіс, фронтенд також на Render як статичний сайт. База даних PostgreSQL розгорнута як керований сервіс. Змінні середовища: URL бази даних, секретний ключ для підпису токенів доступу, ключ SaveEcoBot API передаються через панель налаштувань платформи і не зберігаються у коді. При такому варіанті розгортання користувач взагалі не взаємодіє з серверним середовищем, достатньо лише браузера.

Таблиця 4.1 узагальнює мінімальні і рекомендовані системні вимоги для клієнтської частини.

Таблиця 4.1 — Системні вимоги до клієнтського середовища

| Компонент           | Мінімум                         | Рекомендовано                    |
|---------------------|---------------------------------|----------------------------------|
| Операційна система  | Windows 7 / macOS 10.12 / Linux | Windows 10+ / macOS 12+          |
| Браузер             | Chrome 90+, Firefox 88+         | Chrome або Edge останньої версії |
| Оперативна пам'ять  | 2 ГБ                            | 4 ГБ і більше                    |
| Роздільна здатність | 1280×720                        | 1920×1080                        |
| Інтернет-з'єднання  | 1 Мбіт/с                        | 10 Мбіт/с і більше               |
| JavaScript          | Увімкнений                      | Увімкнений                       |

## 4.2 Сценарій роботи користувача з системою

Реєстрація і вхід у систему. При першому відкритті системи користувач потрапляє на публічний дашборд, де бачить загальну статистику по базі і перелік доступних функцій. Для отримання доступу до аналітичного функціоналу необхідно зареєструватись. Реєстрація потребує коду запрошення, який видає адміністратор системи, електронної пошти і пароля. Після успішної реєстрації користувач автоматично перенаправляється на дашборд аналітика. При повторному вході, вводить електронну пошту і пароль на сторінці входу, після чого токен доступу зберігається в пам'яті застосунку і діє протягом встановленого адміністратором терміну (рисунок 4.1).

The image shows two side-by-side login and registration forms for a system titled 'Кореляційний аналіз' (Correlation Analysis). The left form is for login, with fields for 'Email' (admin@test.com) and 'Пароль' (password), and a 'Увійти' (Login) button. The right form is for registration, with fields for 'Повне ім'я' (Full name: Іван Іванов), 'Email' (email@example.com), 'Пароль' (password), and 'Інвайт-код' (Invitation code: ADMIN-xxxxxxx or LPR-xxxxxxx). It includes a 'Зареєструватись' (Register) button and a note about roles: 'Без коду - роль Аналітика. З кодом від адміністратора - роль Адміна або ЛПР.' (Without code - role Analyst. With code from administrator - role Admin or LPR).

Рисунок 4.1 — Форми входу та реєстрації в системі

Перегляд дашборду і карти територій. Після входу відкривається дашборд із картою, на якій відображені всі доступні території у вигляді маркерів. Клік на маркер відкриває картку з назвою міста, областю і координатами. Тут же можна додати нову територію, достатньо ввести назву міста, координати і область

підставляться автоматично. Нижче карти відображається повний список територій із можливістю редагування і видалення (рисунок 4.2).

**Редагувати: Олександрія**

Введіть назву міста і перейдіть до наступного поля — координати і область підставляється автоматично

| Назва міста * | Область                | Широта  | Довгота |
|---------------|------------------------|---------|---------|
| Олександрія   | Кіровоградська область | 48.6706 | 33.117  |

**Зберегти зміни** **Скасувати**

**49** Територій **49** На карті

**Карта територій**

Клікніть на маркер щоб побачити деталі або відредагувати територію.

Олександрія  
Кіровоградська область  
48.6706, 33.1170  
**Редагувати** **Видалити**

Рисунок 4.2 — Демонстрація редагування території з автоматичним підставленням координат на карті

Завантаження даних. Перед запуском аналізу необхідно переконатись, що для обраної території в базі є дані. Сторінка завантаження даних відкривається через пункт меню «Дані». Якщо є готовий CSV-файл із екологічними або економічними даними — він завантажувється через відповідний блок. Система автоматично розпізнає назви стовпців через словник синонімів і зіставляє рядки з таблицею територій. Якщо файлу немає, але місто є в Україні, то можна скористатись блоком SaveEcoBot API: ввести назву міста і натиснути «Завантажити

дані», після чого система знайде найближчу станцію моніторингу і запише вимірювання якості повітря прямо в базу. Результат одразу відображається в таблиці нижче, де видно кількість записів по кожному показнику і дату останнього оновлення. (рисунок 4.3).

### Завантаження даних

**Екологічні дані (CSV)**  
Якість води, PM2.5, PM10, індекс якості повітря, температура, вологість, тиск  
**Вибрати CSV**

**Економічні дані (CSV)**  
ВРП, зарплата, безробіття, підприємства, охорона здоров'я, інвестиції  
**Вибрати CSV**

**SaveEcoBot API**  
Якість повітря: PM2.5, PM10, AQI, температура, вологість, тиск  
Введіть будь-яке місто України. Якщо станції немає — система знайде найближчу і запише дані до всіх міст цієї області.  
**Червоноград**  
**Завантажити дані**

Завантажено 80 записів для Червоноград (дані з найближчої станції: Chervonohrad). Оновлено міст: 4

**Дані в базі** Оновити

**3 419**  
Екологічних

**8 064**  
Економічних

**47**  
Регіонів

Сортування: За назвою (А-Я) Пошук регіону: червоноград Скинути

| Екологічні дані       |    |            |   | Економічні дані  |    |         |   |
|-----------------------|----|------------|---|------------------|----|---------|---|
| Червоноград           |    |            |   | Червоноград      |    |         |   |
| PM10                  | 9  | 2026-06-02 | x | Підприємства     | 9  | 2023    | x |
| Якість води           | 46 | 2026-04-06 | x | Інвестиції       | 44 | 2025-Q4 | x |
| PM2.5                 | 9  | 2026-06-02 | x | ВРП              | 19 | 2021    | x |
| Вологість             | 9  | 2026-06-02 | x | Зарплата         | 85 | 2022 Q1 | x |
| Температура           | 9  | 2026-06-02 | x | Безробіття       | 7  | 2021    | x |
| Індекс якості повітря | 9  | 2026-06-02 | x | Охорона здоров'я | 4  | 2020    | x |
| Тиск                  | 2  | 2020-05-18 | x |                  |    |         |   |

тільки що завантажені дані

Рисунок 4.3 — Сторінка завантаження даних із результатом імпорту через SaveEcoBot API

Запуск кореляційного аналізу. На сторінці аналізу користувач обирає територію, метод і два показники для порівняння. При виборі методу «Кореляція»

система одночасно розраховує коефіцієнти Пірсона, Спірмена і Кендалла і відображає їх поруч для порівняння. Перед результатами відображається блок підготовки даних: скільки спостережень знайдено, скільки викидів замінено медіаною, і результат тесту Шапіро-Вілка на нормальність розподілу, і якщо розподіл ненормальний, система рекомендує орієнтуватись на Спірмена або Кендалла, а не на Пірсона. Результати відображаються у вигляді стовпчастої діаграми і таблиці з числовими значеннями, рівнями значущості і текстовою інтерпретацією типу «помірна негативна кореляція» (рисунк 4.4).

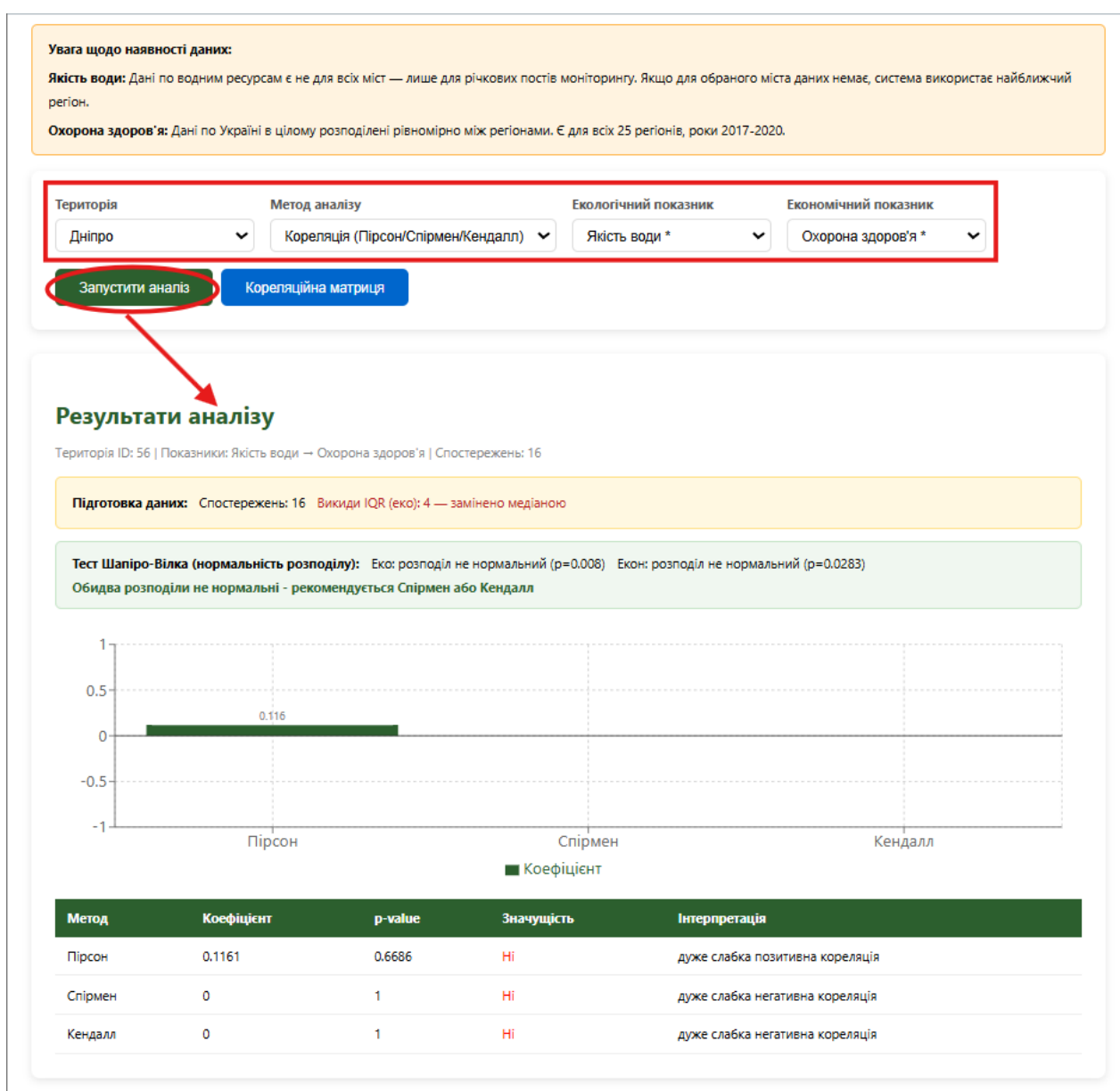


Рисунок 4.4 — Запуск кореляційного аналізу та результати для території Дніпро

Кореляційна матриця. Кнопка «Кореляційна матриця» поруч із запуском аналізу будує повну матрицю по всіх можливих парах показників для обраної території при повному наборі даних. Матриця сортується за абсолютним значенням коефіцієнта Пірсона за замовчуванням, але сортування і фільтри можна змінити. Статистично значущі пари після поправки Бонфероні підсвічуються кольором — зелений для позитивних кореляцій, рожевий для негативних. Прапорець «Тільки значущі» залишає в таблиці лише ті пари, які пройшли поріг значущості (рисунок 4.5).

**Кореляційна матриця**

Проаналізовано пар: 42 | Еко: Якість води, PM2.5, PM10, Індекс якості повітря, Температура, Вологість, Тиск | Екон: ВРП, Зарплата, Безробіття, Підприємства, Охорона здоров'я, Інвестиції

Зелений фон — значуща позитивна кореляція. Рожевий — значуща негативна. Поправка Бонферроні застосована.

Сортування: **За |Пірсон| (сильніші вгорі)** | Еко показник: **Всі** | Екон показник: **Всі** | ☒ Тільки значущі (Бонф.)

| Еко показник | Екон показник    | N   | Пірсон  | p (Бонф.) | Значущість | Спірмен | Інтерпретація                   |
|--------------|------------------|-----|---------|-----------|------------|---------|---------------------------------|
| Тиск         | ВРП              | 8   | 1       | 0         | Так        | 1       | дуже сильна позитивна кореляція |
| Тиск         | Зарплата         | 8   | 1       | 0         | Так        | 1       | дуже сильна позитивна кореляція |
| Тиск         | Безробіття       | 8   | 1       | 0         | Так        | 1       | дуже сильна позитивна кореляція |
| Тиск         | Підприємства     | 8   | -1      | 0         | Так        | -1      | дуже сильна негативна кореляція |
| Тиск         | Охорона здоров'я | 8   | 1       | 0         | Так        | 1       | дуже сильна позитивна кореляція |
| Тиск         | Інвестиції       | 8   | 1       | 0         | Так        | 1       | дуже сильна позитивна кореляція |
| Температура  | Безробіття       | 20  | -0.8545 | 0         | Так        | -0.9    | сильна негативна кореляція      |
| PM2.5        | Інвестиції       | 20  | -0.8289 | 0         | Так        | -0.7    | сильна негативна кореляція      |
| Якість води  | Охорона здоров'я | 16  | 0.7619  | 0.0252    | Так        | 0.4     | сильна позитивна кореляція      |
| Вологість    | Підприємства     | 20  | 0.7426  | 0.0084    | Так        | 0.7     | сильна позитивна кореляція      |
| Якість води  | Зарплата         | 340 | 0.1795  | 0.0378    | Так        | 0.0341  | дуже слабка позитивна кореляція |

Показано рядків: 11 з 42

Рисунок 4.5 — Кореляційна матриця зі значущими парами показників після поправки Бонфероні



Часткова кореляція і аналіз часових рядів. Якщо після перегляду матриці виникає питання, чи не зумовлений виявлений зв'язок третьою змінною, то запускається часткова кореляція з вибором контрольного показника (рисунок 4.6). Результат показує коефіцієнт і рівень значущості вже з урахуванням впливу контрольної змінної. Аналіз часових рядів дозволяє перевірити, чи існує зсув у часі між двома показниками: крос-кореляційна функція показує при якому лагу зв'язок найсильніший, а ковзна кореляція — чи є цей зв'язок стабільним у часі чи змінюється (рисунок 4.7).

Територія: Дніпро

Метод аналізу: Часткова кореляція

Екологічний показник: Якість води \*

Економічний показник: Охорона здоров'я \*

Контрольна змінна: PM10

Запустити аналіз

Кореляційна матриця

### Результати аналізу

Територія ID: 56 | Показники: Якість води → Охорона здоров'я | Спостережень: 16

**Метод:** Часткова кореляція

**Коефіцієнт:** 0.764

**p-value:** 0.0006

**Значущість:** Так

**Контрольна змінна:** PM10

**Інтерпретація:** сильна позитивна часткова кореляція

Рисунок 4.6 — Результати часткової кореляції з контрольною змінною PM10

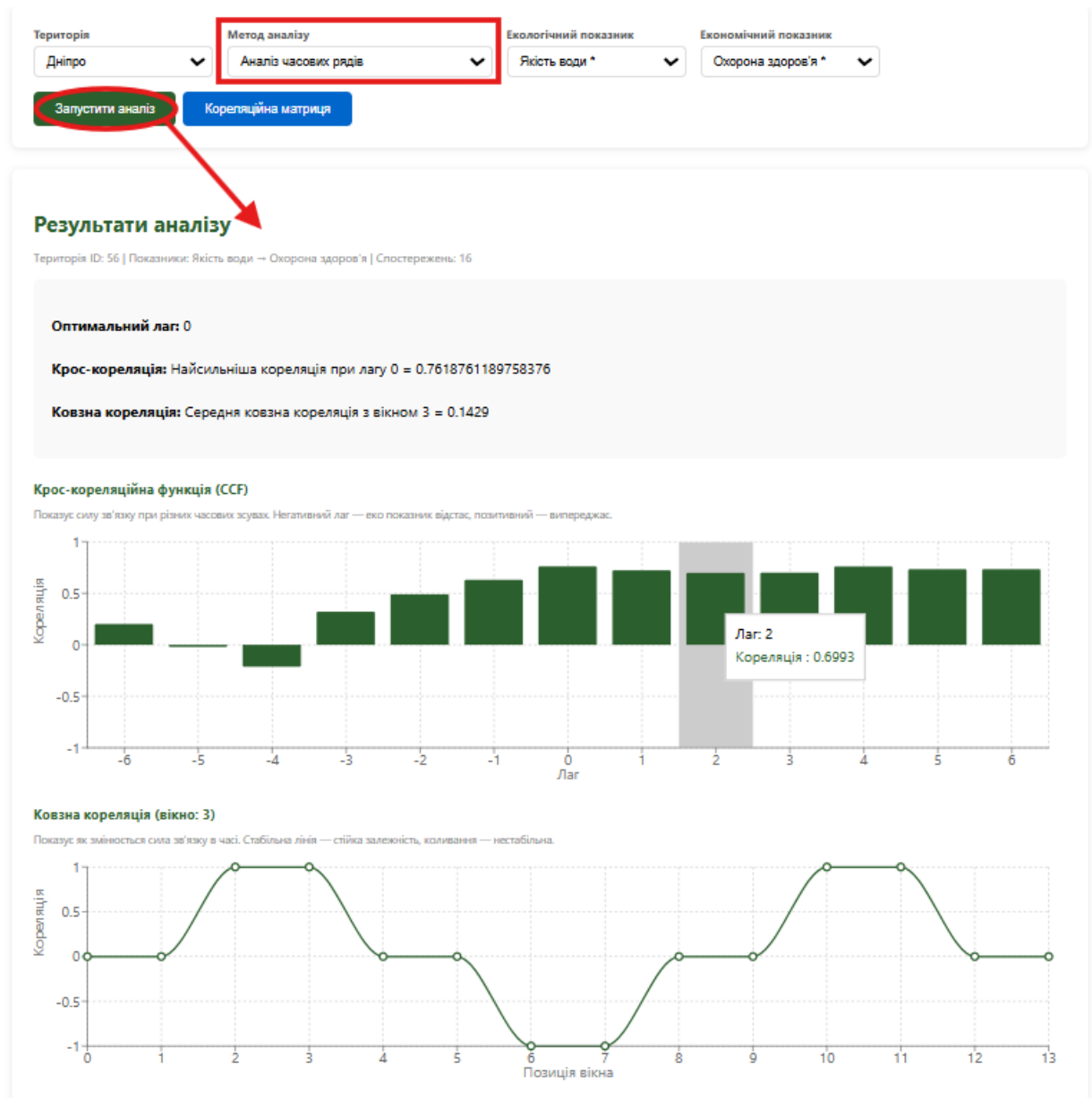


Рисунок 4.7 — Результати аналізу часових рядів: крос-кореляційна функція і ковзна кореляція

Регресійний аналіз (рисунок 4.8). Для побудови математичної моделі залежності обирається метод лінійної або множинної регресії. При лінійній регресії система виводить рівняння залежності, F-статистику, рівень значущості і два графіки: діаграму розсіювання з лінією регресії і графік залишків. При множинній — додатково відображається таблиця коефіцієнтів для кожного предиктора (рисунок 4.9).

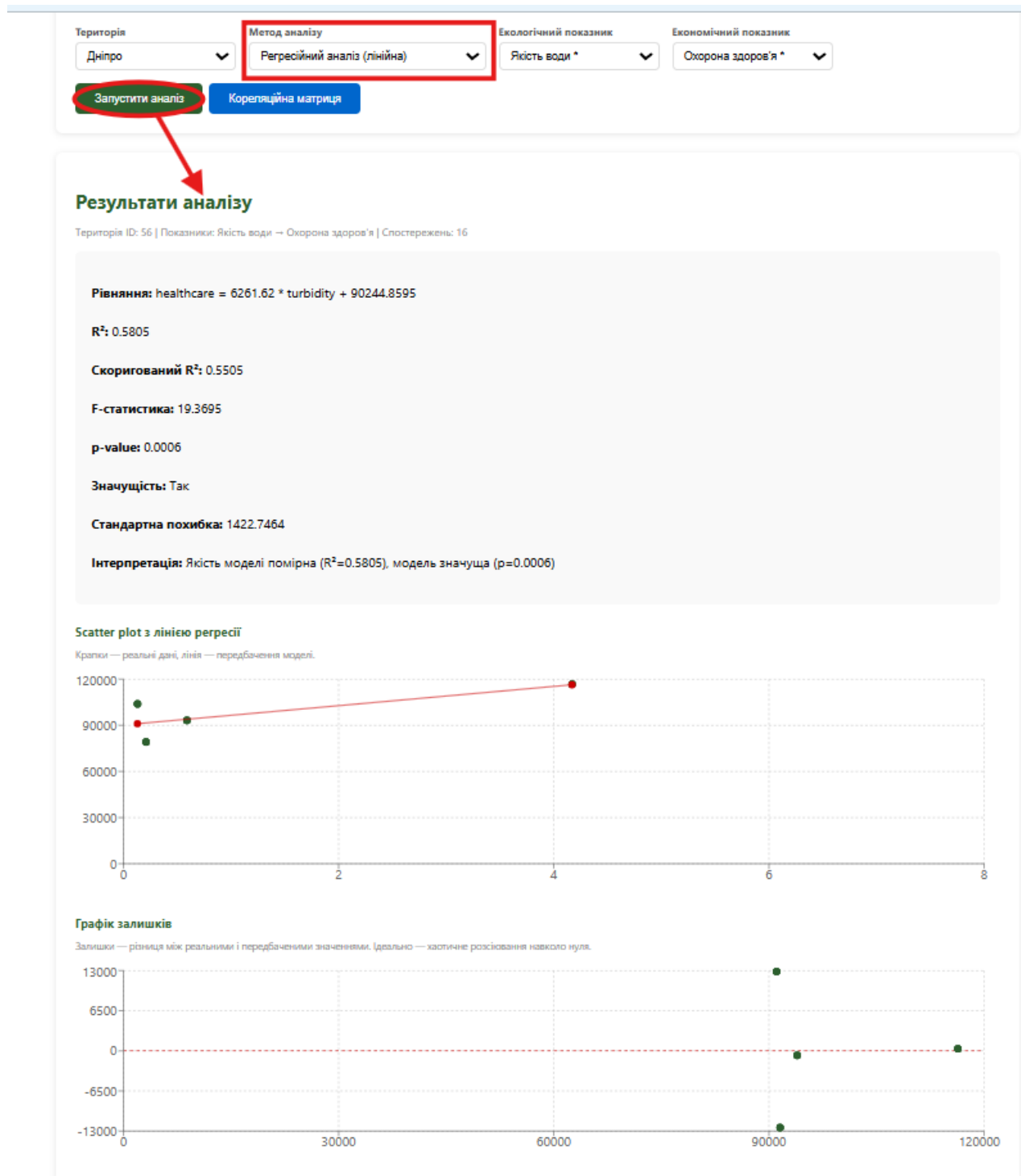


Рисунок 4.8 — Результати лінійного регресійного аналізу з діаграмою розсіювання та графіком залишків

Територія

Дніпро

Метод аналізу

Множинна регресія

Еко показники (мін. 2)

☒ Якість води  
☒ PM2.5  
☒ PM10  
☒ Індекс якості повітря  
☐ Температура  
☐ Вологість  
☐ Тиск

Економічний показник

Охорона здоров'я \*

Запустити аналіз

Кореляційна матриця

Результати аналізу

Територія ID: 56 | Показники: Індекс якості повітря, PM10, PM2.5, Якість води → Охорона здоров'я | Спостережень: 16

Метод: Множинна регресія

R<sup>2</sup>: 1

Скоригований R<sup>2</sup>: 1

Спостережень: 16

Інтерпретація: Якість моделі дуже висока (R<sup>2</sup>=1.0)

Коефіцієнти:

| Показник              | Коефіцієнт  |
|-----------------------|-------------|
| Константа             | 5019.006    |
| Індекс якості повітря | 3254.8408   |
| PM10                  | 11989.789   |
| PM2.5                 | -25890.7801 |
| Якість води           | -2999.8251  |

Рисунок 4.9 — Результати множинної регресії з таблицею коефіцієнтів предикторів

Генерація PDF-звіту. Після завершення аналізу результати можна зафіксувати у вигляді PDF-звіту. На сторінці звітів користувач обирає територію, основний екологічний і економічний показник і набір показників для множинної регресії. Перед генерацією відображається перелік розділів, що увійдуть у звіт. Після натиснення кнопки «Згенерувати PDF звіт» файл формується на сервері і повертається для завантаження. Звіт містить усі розраховані коефіцієнти, графіки і текстові інтерпретації у структурованому форматі з кольоровим кодуванням значущості результатів (рисунок 4.10).

### Налаштування звіту

Територія

Дніпро

Екологічний показник для основного аналізу

Якість води

Економічний показник

Охорона здоров'я

Екологічні показники для множинної регресії

☒ Якість води
 ☐ Температура
 ☒ PM2.5
 ☐ Вологість
 ☒ PM10
 ☐ Тиск
 ☒ Індекс якості повітря

Що буде в звіті:

- Інформація про територію та вибрані показники
- Кореляція Пірсона, Спірмена і Кендалла
- Часткова кореляція з контрольною змінною
- Аналіз часових рядів: крос-кореляція і ковзна кореляція
- Лінійна регресія з рівнянням,  $R^2$ , p-value і залишками
- Множинна регресія з коефіцієнтами,  $R^2$  і VIF
- Порівняння методів, узгодженість, суперечності і ranking

Згенерувати PDF звіт

Рисунок 4.10 — Налаштування параметрів PDF-звіту

Адміністративні функції. Для користувача з роллю адміністратора доступна панель управління акаунтами. Тут відображається таблиця всіх зареєстрованих користувачів із можливістю змінити роль через випадаючий список або заблокувати акаунт. На вкладці запрошувальних кодів адміністратор генерує нові коди із вказаною роллю і терміном дії, а після використання код автоматично позначається як використаний і більше не приймається при реєстрації (рисунок 4.11).

### Управління користувачами

| ID | Email                 | Im'я           | Роль     | Статус   | Дії      |
|----|-----------------------|----------------|----------|----------|----------|
| 1  | admin@test.com        | Адміністратор  | Адмін    | Активний | Адмін    |
| 2  | krasotka123@gmail.com | Ваніліна Аліна | Аналітик | Активний | Аналітик |
| 4  | vasil@gmail.com       | Пупкін Василь  | Аналітик | Активний | Аналітик |
| 3  | zhyh@gmail.com        | Жук Вікторія   | ДПР      | Активний | ДПР      |
| 5  | kotik@gmail.com       | Кіт Мурзик     | ДПР      | Активний | ДПР      |
| 6  | anastasiya@gmail.com  | Яковський Влад | Аналітик | Активний | Аналітик |

### Генерація інвайт-кодів

Роль для коду

ДПР

Термін дії (днів)

7

Згенерувати код

Всі коди

| Код               | Роль     | Статус       | Використання | Дії          |
|-------------------|----------|--------------|--------------|--------------|
| LPR-19641048000   | ДПР      | Використаний | —            | Використаний |
| ANALYST-3LQHQDQAK | Аналітик | Використаний | —            | Використаний |
| LPR-91794823320   | ДПР      | Активний     | —            | Використаний |
| LPR-1007689900    | ДПР      | Активний     | —            | Використаний |
| ADMIN-CHC8023232  | Адмін    | Активний     | —            | Використаний |
| LPR-18177963230   | ДПР      | Активний     | —            | Використаний |
| LPR-1708349544    | ДПР      | Активний     | —            | Використаний |
| LPR-1745794008    | ДПР      | Активний     | —            | Використаний |

Рисунок 4.11 — Адміністративна панель: управління користувачами та генерація інвайт-кодів

Розмежування ролей між адміністратором, аналітиком і особою, що приймає рішення, дозволяє гнучко керувати доступом до функціоналу залежно від рівня технічної підготовки користувача. Інтерфейс спроектований таким чином, що кожен крок логічно впливає з попереднього — користувач не може запустити аналіз без попередньо завантажених даних, а результати одразу доступні для подальшої інтерпретації або збереження у звіті.

## ВИСНОВКИ

У ході виконання дипломної роботи досягнуто поставленої мети: розроблено вебсистему кореляційного аналізу екологічних та економічних показників території, яка автоматизує повний цикл від збору даних до інтерпретації результатів і генерації аналітичних звітів.

Для серверної частини обрано Python із фреймворком FastAPI, який забезпечує асинхронну обробку обчислювально навантажених запитів і автоматичну валідацію вхідних даних. Як СУБД обрано PostgreSQL із підтримкою JSONB-полів. Клієнтська частина реалізована на React із бібліотекою Recharts для інтерактивної візуалізації і Leaflet на основі OpenStreetMap для географічного відображення територій.

Серверну частину реалізовано за шаровою архітектурою з розподілом на маршрутизатори, сервіси і аналітичні модулі. Реалізовано п'ять методів статистичного аналізу: кореляція Пірсона, Спірмена і Кендалла з поправкою Бонфероні, часткова кореляція з контрольною змінною, аналіз часових рядів із крос-кореляційною функцією і ковзною кореляцією, лінійна і множинна регресія. Перед кожним аналізом виконується автоматична підготовка даних через лінійну інтерполяцію пропусків і заміну викидів медіаною методом IQR. Збір екологічних даних інтегровано через SaveEcoBot API з геолокаційним пошуком найближчої станції моніторингу.

Розроблено інтерактивний вебінтерфейс із публічним дашбордом, сторінками завантаження даних, аналізу, генерації PDF-звітів і адміністративною панеллю. Система підтримує три ролі користувачів: адміністратор, аналітик і особа, що приймає рішення — з розмежуванням доступу через токени і запрошувальні коди. Розгорнута як вебзастосунок, вона доступна через браузер без встановлення додаткового програмного забезпечення, що робить інструмент придатним для практичного використання аналітиками, екологами і представниками органів місцевого самоврядування.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Єріна А., Мазуренко О . СТАТИСТИЧНИЙ АНАЛІЗ ЧАСОВИХ РЯДІВ : НАВЧАЛЬНИЙ ПОСІБНИК. Київ : КНУШ, 2022. С. 164. URL: [https://idss.org.ua/arhiv/Yerina\\_Mazurenko\\_colour.pdf](https://idss.org.ua/arhiv/Yerina_Mazurenko_colour.pdf) (дата звернення: 16.04.2026).
2. Грицевич В. С . КОРЕЛЯЦІЙНИЙ ТА РЕГРЕСІЙНИЙ АНАЛІЗ В СУСПІЛЬНІЙ ГЕОГРАФІЇ. ТЕКСТИ ЛЕКЦІЙ. Львів : Малий видавничий центр. Лабораторія тематичного картографування географічного факультету, 2016. С. 24. URL: <https://geography.lnu.edu.ua/wp-content/uploads/2017/11/50272620.pdf> (дата звернення: 16.04.2026).
3. TIBCO. TIBCO Product Documentation. URL: [https://docs.tibco.com/pub/stat/14.1.0/doc/pdf/TIB\\_stat\\_14.1.0\\_r-integration.pdf?id=1](https://docs.tibco.com/pub/stat/14.1.0/doc/pdf/TIB_stat_14.1.0_r-integration.pdf?id=1) (дата звернення: 17.04.2026).
4. IBM SPSS Statistics. URL: <https://www.ibm.com/products/spss-statistics> (дата звернення: 18.04.2026).
5. Openaq. About Us. URL: <https://openaq.org/about/> (дата звернення: 18.04.2026).
6. OpenAQ Docs. About the API. URL: <https://docs.openaq.org/about/about> (дата звернення: 18.04.2026).
7. Datawrapper. URL: <https://www.datawrapper.de/> (дата звернення: 19.04.2026).
8. Datawrapper. Pricing. URL: <https://www.datawrapper.de/pricing> (дата звернення: 19.04.2026).
9. FastAPI. URL: <https://fastapi.tiangolo.com/#typer-the-fastapi-of-clis> (дата звернення: 20.04.2026).
10. SaveEcoBot. URL: <https://www.saveecobot.com/> (дата звернення: 20.04.2026).
11. React. the library for web and native user interfaces. react.dev.. URL: <https://react.dev/> (дата звернення: 21.04.2026).
12. Recharts. a composable charting library. recharts.org.. URL:



<https://recharts.org/> (дата звернення: 21.04.2026).

13. Перегуда О. В., Капустян О. А., Курилко О. Б. СТАТИСТИЧНА ОБРОБКА ДАНИХ : Навчальний посібник. Київ : МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА, 2022. С. 103. URL: [https://mechmat.knu.ua/wp-content/uploads/2022/02/navch\\_pos\\_perehuda.pdf](https://mechmat.knu.ua/wp-content/uploads/2022/02/navch_pos_perehuda.pdf) (дата звернення: 24.04.2026).

14. Яровий А. Т., Страхов Є. М. Аналіз часових рядів : навчально-методичний посібник. Одеса : Освіта України, 2019. С. 109. URL: [https://onu.edu.ua/pub/bank/userfiles/files/fmfit/nachalni\\_materiali/navch\\_mat\\_111\\_mag/analiz\\_chasovykh\\_ryadiv.pdf](https://onu.edu.ua/pub/bank/userfiles/files/fmfit/nachalni_materiali/navch_mat_111_mag/analiz_chasovykh_ryadiv.pdf) (дата звернення: 24.06.2026).

15. Куц Ю. В., Лисенко Ю. Ю. Статистичні методи визначення залежностей між випадковими величинами : навчальний посібник. Київ : КПІ ім. Ігоря Сікорського, 2023. URL: <https://ela.kpi.ua/server/api/core/bitstreams/5499b838-167b-4360-8c3d-57eab355a88d/content> (дата звернення: 28.04.2026).

16. Бідюк П. І., Данилов В. Я., Жиров О. Л. Прикладна статистика : навчальний посібник. Київ : КПІ ім. Ігоря Сікорського, 2023. URL: <https://ela.kpi.ua/server/api/core/bitstreams/678e0b39-a61d-4b34-9c61-f275eb358ed8/content> (дата звернення: 27.04.2026).

17. Коннова Л. О., Чорноус Л. В., Скопенко Н. С. Економетрія : конспект лекцій. Київ : НУХТ, 2010. 144 с. URL: <https://dspace.nuft.edu.ua/server/api/core/bitstreams/63914a57-1d45-4b40-867f-837977a0998b/content> (дата звернення: 29.04.2026).

18. Штельма О. М., Макогон Н. В. Методичні вказівки до виконання лабораторних робіт з дисципліни «Комп'ютерне моделювання економічних систем та процесів». Харків : ХНУМГ ім. О. М. Бекетова, 2014. URL: [https://eprints.kname.edu.ua/53814/1/2014\\_348M-репоз-KMES.pdf](https://eprints.kname.edu.ua/53814/1/2014_348M-репоз-KMES.pdf) (дата звернення: 30.04.2026).

19. OpenStreetMap — відкрита карта світу [Електронний ресурс]. URL: <https://www.openstreetmap.org/about> (дата звернення: 21.04.2026).

20. Passing Data Deeply with Context. react.dev [Електронний ресурс]. URL:

<https://react.dev/learn/passing-data-deeply-with-context> (дата звернення: 21.04.2026).

## **ДОДАТОК А**

### **Веб-система кореляційного аналізу екологічних та економічних показників території**

Текст програми

НТУУ «КПІ ім. Ігоря Сікорського»

Аркушів 10

Київ — 2026

## Додаток А

Програмні засоби:

- мова програмування — Python 3.11
- серверний фреймворк — FastAPI 0.104
- ORM та міграції — SQLAlchemy 2.0, Alembic
- СУБД — PostgreSQL 15
- аналітичні бібліотеки — NumPy 1.26, SciPy 1.11, pandas 2.1
- генерація звітів — ReportLab 4.0
- HTTP-клієнт — httpx 0.25
- клієнтська частина — React 18, Recharts 2.8, Leaflet 1.9

### Лістинг аналітичних модулів системи

#### Модуль підготовки даних та кореляційного аналізу (correlation.py)

```
import numpy as np
from scipy import stats
from typing import List, Dict, Any, Tuple

def interpolate_missing(values: List[float]) -> List[float]:
    """
    Лінійна інтерполяція пропущених значень (None/NaN).
    Крайні пропуски заповнюються найближчим наявним значенням.
    """
    arr = np.array(values, dtype=float)
    nans = np.isnan(arr)
    if not nans.any():
        return values
    indices = np.arange(len(arr))
    arr[nans] = np.interp(indices[nans], indices[~nans], arr[~nans])
    return arr.tolist()
```

```

def remove_outliers_iqr(values: List[float],
                        factor: float = 1.5) -> Tuple[List[float], int]:
    """
    Виявлення та заміна викидів методом міжквартильного розмаху (IQR).
    Значення за межами [Q1 - factor*IQR, Q3 + factor*IQR]
    замінюються на медіану вибірки.
    """
    arr = np.array(values, dtype=float)
    q1, q3 = np.percentile(arr, 25), np.percentile(arr, 75)
    iqr = q3 - q1
    if iqr == 0:
        return values, 0
    lower, upper = q1 - factor * iqr, q3 + factor * iqr
    median = np.median(arr)
    outlier_mask = (arr < lower) | (arr > upper)
    arr[outlier_mask] = median
    return arr.tolist(), int(outlier_mask.sum())

def check_normality(x: List[float]) -> Dict[str, Any]:
    """Тест Шапіро-Вілка на нормальність розподілу"""
    if len(x) < 3:
        return {"is_normal": None, "note": "Замало даних"}
    stat, p_value = stats.shapiro(x)
    is_normal = bool(p_value > 0.05)
    return {
        "statistic": round(float(stat), 4),
        "p_value": round(float(p_value), 4),
        "is_normal": is_normal
    }

```

```
def calculate_pearson(x, y) -> Dict[str, Any]:
    coef, p_value = stats.pearsonr(x, y)
    return {
        "method": "pearson",
        "coefficient": round(float(coef), 4),
        "p_value": round(float(p_value), 4),
        "is_significant": bool(p_value < 0.05),
        "interpretation": interpret_correlation(coef)
    }
```

```
def calculate_spearman(x, y) -> Dict[str, Any]:
    coef, p_value = stats.spearmanr(x, y)
    return {
        "method": "spearman",
        "coefficient": round(float(coef), 4),
        "p_value": round(float(p_value), 4),
        "is_significant": bool(p_value < 0.05),
        "interpretation": interpret_correlation(coef)
    }
```

```
def calculate_kendall(x, y) -> Dict[str, Any]:
    coef, p_value = stats.kendalltau(x, y)
    return {
        "method": "kendall",
        "coefficient": round(float(coef), 4),
        "p_value": round(float(p_value), 4),
        "is_significant": bool(p_value < 0.05),
        "interpretation": interpret_correlation(coef)
    }
```

```
def apply_bonferroni_correction(p_values: List[float]) -> List[float]:
    n = len(p_values)
    return [round(min(float(p) * n, 1.0), 4) for p in p_values]
```

### Модуль часткової кореляції (partial.py)

```
import numpy as np
from scipy import stats
from typing import List, Dict, Any

def calculate_partial_correlation(
    x: List[float], y: List[float], z: List[float]
) -> Dict[str, Any]:
    if len(x) < 4:
        return {"error": "Потрібно мінімум 4 спостереження"}

    def residuals(a, b):
        slope, intercept = np.polyfit(b, a, 1)
        return [a[i] - (slope * b[i] + intercept) for i in range(len(a))]

    res_x = residuals(x, z)
    res_y = residuals(y, z)
    coef, p_value = stats.pearsonr(res_x, res_y)

    return {
        "method": "partial_correlation",
        "coefficient": round(float(coef), 4),
        "p_value": round(float(p_value), 4),
        "is_significant": bool(p_value < 0.05),
```

```

        "interpretation": interpret_partial(coef)
    }

```

## Модуль аналізу часових рядів (timeseries.py)

```

import numpy as np
from scipy import stats
from typing import List, Dict, Any

def calculate_cross_correlation(
    x: List[float], y: List[float], max_lag: int = 6
) -> Dict[str, Any]:
    if len(x) < 4:
        return {"error": "Потрібно мінімум 4 спостереження"}

    x = np.array(x, dtype=float)
    y = np.array(y, dtype=float)
    x = (x - np.mean(x)) / (np.std(x) + 1e-10)
    y = (y - np.mean(y)) / (np.std(y) + 1e-10)

    safe_max_lag = min(max_lag, len(x) // 2 - 1)
    lags = range(-safe_max_lag, safe_max_lag + 1)
    correlations = []

    for lag in lags:
        if lag < 0:
            xa, ya = x[:lag], y[-lag:]
        elif lag > 0:
            xa, ya = x[lag:], y[:-lag]

```



```

else:
    xa, ya = x, y
    corr, p_val = stats.pearsonr(xa, ya)
    correlations.append({
        "lag": int(lag),
        "correlation": float(corr),
        "p_value": float(p_val),
        "is_significant": bool(p_val < 0.05)
    })

```

```

best = max(correlations, key=lambda c: abs(c["correlation"]))
return {
    "method": "cross_correlation",
    "correlations_by_lag": correlations,
    "optimal_lag": best["lag"],
    "optimal_correlation": best["correlation"]
}

```

```

def calculate_rolling_correlation(
    x: List[float], y: List[float], window: int = 3
) -> Dict[str, Any]:
    x = np.array(x, dtype=float)
    y = np.array(y, dtype=float)
    rolling = []

    for i in range(len(x) - window + 1):
        x_w = x[i:i + window]
        y_w = y[i:i + window]
        corr, _ = stats.pearsonr(x_w, y_w)
        rolling.append({

```

```

        "window_start": i,
        "correlation": round(float(corr), 4)
    })

avg = float(np.mean([r["correlation"] for r in rolling]))
return {
    "method": "rolling_correlation",
    "window_size": window,
    "rolling_correlations": rolling,
    "average_correlation": round(avg, 4)
}

```

### Модуль регресійного аналізу (regression.py)

```

import numpy as np
from scipy import stats
from typing import List, Dict, Any

def calculate_linear_regression(
    x: List[float], y: List[float],
    x_label: str = "x", y_label: str = "y"
) -> Dict[str, Any]:
    if len(x) < 3:
        return {"error": "Потрібно мінімум 3 спостереження"}

    x_arr = np.array(x, dtype=float)
    y_arr = np.array(y, dtype=float)
    slope, intercept, r_value, p_value, std_err = stats.linregress(x_arr, y_arr)

    r_squared = float(r_value ** 2)

```

```

n = len(x_arr)
adj_r_squared = 1 - (1 - r_squared) * (n - 1) / (n - 2)
y_pred = slope * x_arr + intercept
residuals = y_arr - y_pred

return {
    "method": "linear_regression",
    "equation": f"{y_label} = {round(float(slope),4)} * {x_label} +
                {round(float(intercept),4)}",
    "r_squared": round(r_squared, 4),
    "adj_r_squared": round(adj_r_squared, 4),
    "f_statistic": round(float((r_squared/(1-r_squared))*(n-2)), 4),
    "p_value": round(float(p_value), 4),
    "std_error": round(float(std_err), 4),
    "is_significant": bool(p_value < 0.05),
    "predicted": [round(float(v), 4) for v in y_pred],
    "residuals": [round(float(r), 4) for r in residuals]
}

```

```

def calculate_multiple_regression(
    X: List[List[float]], y: List[float],
    x_labels: List[str] = None
) -> Dict[str, Any]:
    X_arr = np.array(X, dtype=float).T
    y_arr = np.array(y, dtype=float)
    n, k = X_arr.shape
    X_const = np.column_stack([np.ones(n), X_arr])
    coefs = np.linalg.lstsq(X_const, y_arr, rcond=None)[0]

    y_pred = X_const @ coefs

```

```

ss_res = float(np.sum((y_arr - y_pred) ** 2))
ss_tot = float(np.sum((y_arr - np.mean(y_arr)) ** 2))
r_squared = 1 - ss_res / ss_tot if ss_tot > 1e-10 else 0.0
adj_r_sq = 1 - (1 - r_squared) * (n - 1) / (n - k - 1)

coef_dict = {"intercept": round(float(coefs[0]), 4)}
for i, label in enumerate(x_labels):
    coef_dict[label] = round(float(coefs[i + 1]), 4)

return {
    "method": "multiple_regression",
    "coefficients": coef_dict,
    "r_squared": round(r_squared, 4),
    "adj_r_squared": round(adj_r_sq, 4),
    "n_observations": n
}

```

## ДОДАТОК Б

Веб-система кореляційного аналізу екологічних та економічних  
показників території

### АПРОБАЦІЯ

3-тя Міжнародна науково-практична конференція «Science and Technology: New Horizons of Development»: збірник наукових праць / International Scientific Unity. – Прага, Чехія, 3–5 червня 2026 року. – Секція: Комп'ютерна інженерія

Аркушів 6

Київ — 2026

## ВЕБ-СИСТЕМА КОРЕЛЯЦІЙНОГО АНАЛІЗУ ЕКОЛОГІЧНИХ ТА ЕКОНОМІЧНИХ ПОКАЗНИКІВ ТЕРИТОРІЇ

Головченко Ксенія  
студентка 4 курсу  
cheryxso@gmail.com

Кафедра цифрових технологій в енергетиці  
НТУУ «Київський політехнічний інститут імені Ігоря Сікорського», Україна

Взаємозалежність між екологічним станом і економічним розвитком регіонів є актуальною задачею прикладного аналізу даних. Існуючі статистичні пакети, такі як STATISTICA і IBM SPSS, вимагають високої кваліфікації та не мають предметної прив'язки до регіональних даних України, тоді як вебплатформи на кшталт DataWrapper і OpenAQ обмежені або візуалізацією без розрахунків, або збором даних без аналізу [1, 2]. У цих умовах розроблено спеціалізовану вебсистему, що автоматизує повний цикл від завантаження даних до інтерпретації кореляційних залежностей.

**Архітектура системи.** Система побудована за клієнт-серверною архітектурою з чітким розподілом відповідальності між рівнями. Серверна частина реалізована на Python із фреймворком FastAPI, що забезпечує асинхронну обробку обчислювально навантажених запитів і автоматичну валідацію вхідних даних через Pydantic. Як СУБД обрано PostgreSQL для гнучкого зберігання результатів різноманітних аналізів. Клієнтська частина реалізована на React із бібліотекою Recharts для інтерактивної візуалізації і Leaflet на основі OpenStreetMap для географічного відображення територій. Бекенд організовано за шаровою структурою: маршрутизатори приймають HTTP-запити і делегують їх сервісам, сервіси містять бізнес-логіку, аналітичні модулі реалізують статистичні обчислення незалежно від решти системи Рис.1 [3].

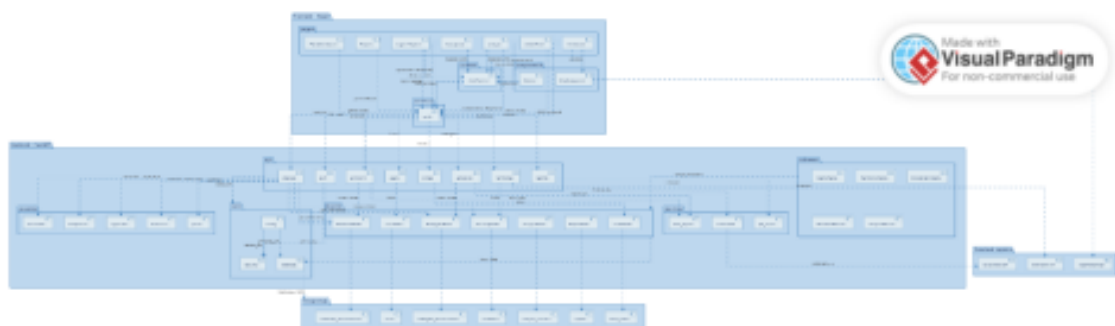


Рисунок 1. Компонентна діаграма архітектури системи.

**Методи статистичного аналізу.** Аналітичне ядро системи реалізує п'ять методів статистичного аналізу. Коефіцієнт кореляції Пірсона вимірює силу і напрямок лінійного зв'язку між двома кількісними змінними. Статистична значущість перевіряється через t-статистику з  $n - 2$  ступенями свободи. При аналізі матриці кореляцій для кількох пар показників одночасно застосовується поправка Бонфероні, що знижує ймовірність хибнопозитивних результатів [4]:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \cdot \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

Коефіцієнт кореляції Спірмена є непараметричним аналогом, стійким до викидів і придатним для даних без нормального розподілу [4]:

$$r_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)} \quad (2)$$

де  $d_i$  – різниця рангів  $i$  – го спостереження. Для екологічних часових рядів, де аномальні вимірювання є нормою, Спірмен дає надійніший результат, ніж Пірсон.

Часткова кореляція дозволяє оцінити зв'язок між двома змінними при фіксованому впливі третьої – контрольної, що є важливим в еколого-економічному аналізі: кореляція між PM2.5 і безробіттям може бути зумовлена спільним впливом рівня індустріалізації регіону, а не прямим причинно-наслідковим зв'язком [5]:

$$r_{xy \cdot z} = \frac{r_{xy} - r_{xz}r_{yz}}{\sqrt{(1 - r_{xz}^2)(1 - r_{yz}^2)}} \quad (3)$$

Лінійна регресія будує модель залежності методом найменших квадратів. Якість моделі оцінюється через коефіцієнт детермінації [4]:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4)$$

Аналіз часових рядів включає розрахунок крос-кореляційної функції для виявлення часового зсуву між рядами і ковзної кореляції для оцінки стабільності зв'язку в часі [6]. Перед застосуванням будь-якого методу дані проходять автоматичну підготовку: лінійна інтерполяція заповнює пропущені значення, метод міжквартильного розмаху (IQR) виявляє і замінює викиди медіаною вибірки Рис.2.

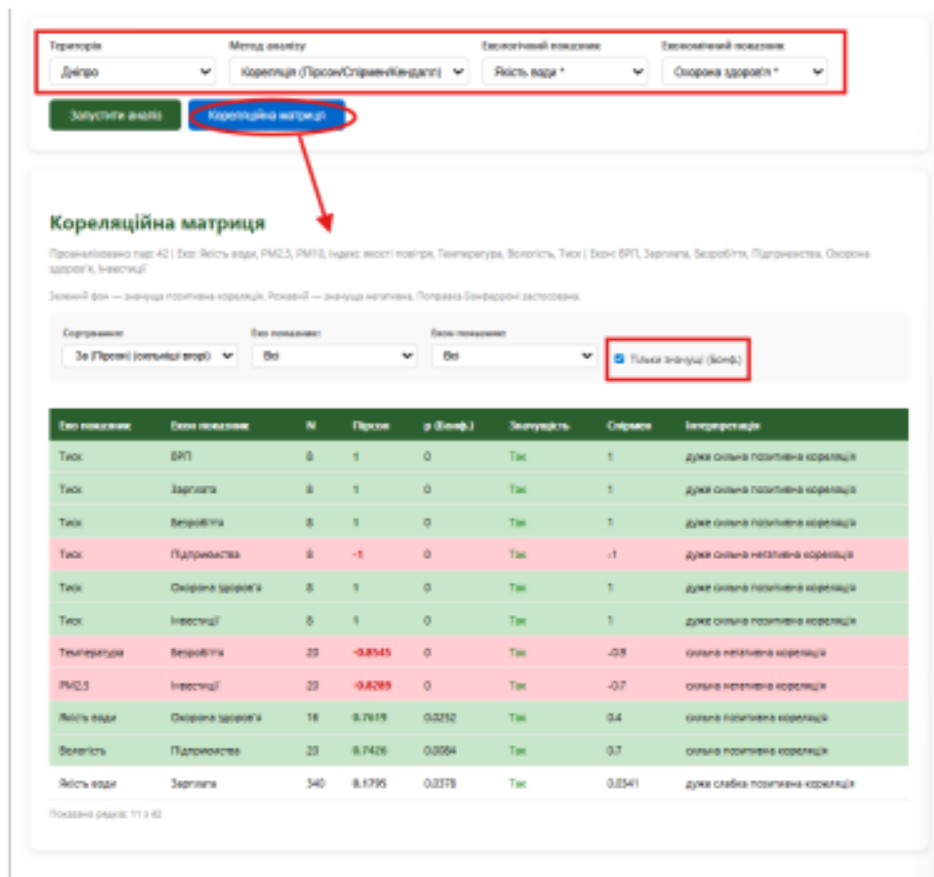


Рисунок 2. Приклад кореляційної матриці з кольоровим кодуванням.

**Ключові технічні рішення.** Збір екологічних даних реалізовано через інтеграцію з SaveEcoBot API – публічним сервісом моніторингу якості повітря в Україні. Сервіс виконує двоетапний пошук станцій: спочатку за назвою міста, а при відсутності збігів – за мінімальною відстанню через формулу гаверсінуса. Для економічних показників реалізовано імпорту CSV із автоматичною нормалізацією назв стовпців через словник із понад 30 варіантами написання кожного показника, що дозволяє завантажувати файли Держстату без попереднього перейменування. Результати аналізу зберігаються в полі типу JSONB, що дає змогу зберігати довільно структуровані об'єкти для різних типів аналізу без створення окремих таблиць під кожен метод. Генерація PDF-звітів реалізована через бібліотеку ReportLab із кольоровим кодуванням рядків за рівнем статистичної значущості результатів Рис.3 [3].



## PDF-звіт кореляційного аналізу

|                        |                  |
|------------------------|------------------|
| Територія              | Львів            |
| Екологічний показник   | Якість води      |
| Економічний показник   | Підприємства     |
| Кількість спостережень | 36               |
| Дата генерації         | 02.06.2026 14:37 |

### 1. Кореляційний аналіз

| Метод   | Коефіцієнт | p-value | Значущість | Інтерпретація                   |
|---------|------------|---------|------------|---------------------------------|
| Пірсон  | 0.1095     | 0.525   | Ні         | дуже слабка позитивна кореляція |
| Спірмен | 0.0774     | 0.6538  | Ні         | дуже слабка позитивна кореляція |
| Кендалл | 0.079      | 0.5325  | Ні         | дуже слабка позитивна кореляція |

| Показник     | Результат                                                          | p-value |
|--------------|--------------------------------------------------------------------|---------|
| Екологічний  | розподіл не нормальний                                             | 0.0     |
| Економічний  | розподіл не нормальний                                             | 0.0055  |
| Рекомендація | Обидва розподіли не нормальні - рекомендується Спірмен або Кендалл |         |

### 2. Часткова кореляція

| Показник          | Значення                             |
|-------------------|--------------------------------------|
| Контрольна змінна | PM2.5                                |
| Коефіцієнт        | 0.4297                               |
| p-value           | 0.0586                               |
| Значущість        | Ні                                   |
| Інтерпретація     | помірна позитивна часткова кореляція |

### 3. Аналіз часових рядів

| Метод          | Результат                                             |
|----------------|-------------------------------------------------------|
| Крос-кореляція | Найсильніша кореляція при лагу -3 = 0.317583810721835 |

Рисунок 3. Скріншот фрагменту сторінки PDF-звіту.

**Результати.** Систему розгорнуто як вебзастосунок. Інтерфейс підтримує три ролі користувачів – адміністратор, аналітик і особа, що приймає рішення (з розмежуванням доступу через JWT-токени і запрошувальні коди). Система доступна через браузер без встановлення додаткового програмного забезпечення.

**Висновки.** Розроблена вебсистема автоматизує повний аналітичний цикл: від збору даних через SaveEcoBot API та імпорту CSV до розрахунку кореляцій, регресії, аналізу часових рядів і генерації PDF-звітів. На відміну від існуючих рішень, вона має предметну прив'язку до регіональних даних України, вбудований препроцесинг і доступний вебінтерфейс для нетехнічних користувачів.

### Список використаних джерел

1. TIBCO Software Inc. TIBCO Statistica – product overview [Електронний ресурс]. – Режим доступу: <https://docs.tibco.com/products/tibco-statistica-14-1-0> (дата звернення: 01.06.2026).
2. OpenAQ. About OpenAQ [Електронний ресурс] // [openaq.org](https://openaq.org/about/). – Режим доступу: <https://openaq.org/about/> (дата звернення: 01.06.2026).
3. FastAPI. FastAPI – high performance Python web framework [Електронний ресурс] // [fastapi.tiangolo.com](https://fastapi.tiangolo.com). – Режим доступу: <https://fastapi.tiangolo.com> (дата звернення: 01.06.2026).
4. Перегуда О. В., Капустян О. А., Курилко О. Б. Статистична обробка даних : навч. посіб. / О. В. Перегуда, О. А. Капустян, О. Б. Курилко. – Київ : КНУ імені Тараса Шевченка, 2022. – 103 с.
5. Вислоух С. І. Перевірка зв'язку між перемінними [Електронний ресурс] // [vysloukh.in.ua](https://vysloukh.in.ua). – Режим доступу: [https://vysloukh.in.ua/4\\_Prov\\_sv\\_perem.pdf](https://vysloukh.in.ua/4_Prov_sv_perem.pdf) (дата звернення: 01.06.2026).
6. Яровий А. Т., Страхов Є. М. Аналіз часових рядів : навч.-метод. посіб. / А. Т. Яровий, Є. М. Страхов. – Одеса : Освіта України, 2019.

