

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

Факультет прикладної математики

Кафедра системного програмування

і спеціалізованих комп'ютерних систем

«На правах рукопису»
УДК 004.9:355.58

До захисту допущено:
Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ
« ____ » _____ 2025 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Системне програмування та
спеціалізовані комп'ютерні системи»
зі спеціальності 123 «Комп'ютерна інженерія»**

**на тему: «Спеціалізована багатокористувацька інформаційна система
реєстрування процесів розмінування земельних ділянок»**

Виконав:

студент II курсу, групи КВ-41мп
Холодар Антон Анатолійович _____

Науковий керівник:

старший викладач кафедри СПіСКС
кандидат технічних наук,
Коляда Костянтин Вячеславович _____

Рецензент:

доцент кафедри ОТ, кандидат технічних наук,
доцент, Марковський Олександр Петрович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент _____

Київ – 2025 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет прикладної математики
Кафедра системного програмування
і спеціалізованих комп'ютерних систем**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування та спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

«___» _____ 2025 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту
Холодару Антону Анатолійовичу**

1. Тема дисертації «Спеціалізована багатокористувацька інформаційна система реєстрування процесів розмінування земельних ділянок», науковий керівник дисертації к.т.н., ст. викладач Коляда Костянтин Вячеславович, затверджені наказом по університету від « 6 » листопада 2025 р. № 4836-с
2. Термін подання студентом дисертації: 5 грудня 2025
3. Об'єкт дослідження: інформаційні системи для реєстрування та відображення процесів розмінування земельних ділянок.
4. Вихідні дані: аналіз потреб населення в інформації про небезпечні території, аналіз існуючих рішень у сфері протимінної діяльності, вимоги до багатокористувацьких вебсистем з підтримкою геопросторових даних.
5. Перелік завдань, які потрібно розробити:
 - Провести аналіз предметної області гуманітарного розмінування та потреб різних категорій користувачів.
 - Виконати аналіз існуючих інформаційних рішень у сфері протимінної діяльності.
 - Обґрунтувати вибір технологічного стеку для розробки геоінформаційної системи.
 - Розробити багатокористувацьку модель доступу для різних категорій користувачів.

- Спроекувати модель даних для обліку секторів розмінування з підтримкою геопросторових типів.
- Реалізувати веборієнтовану систему з інтерактивною картою.
- Провести тестування та оцінку розробленої системи.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: презентація

7. Перелік публікацій:

- Підготувати тези до ПМК-25.
- Підготувати тези до конференції за узгодженням з керівником.

8. Дата видачі завдання 07.10.2024.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Вивчення літератури за тематикою магістерської дисертації	02.09.2024	
2.	Визначення структури магістерської дисертації	09.09.2024	
3.	Робота над вступом та змістом магістерської дисертації	08.09.2025	
4.	Робота над першим розділом магістерської дисертації	15.09.2025	
5.	Робота над другим розділом магістерської дисертації	29.09.2025	
6.	Підготовка матеріалів доповіді на конференції	13.10.2025	
7.	Робота над третім розділом магістерської дисертації	20.10.2025	
8.	Робота над четвертим розділом магістерської дисертації	10.11.2025	
9.	Оформлення документації магістерської дисертації	24.11.2025	
10.	Попередній розгляд магістерської дисертації	05.12.2025	

Студент

Антон ХОЛОДАР

Науковий керівник

Костянтин КОЛЯДА

РЕФЕРАТ

Актуальність теми. Після початку повномасштабного вторгнення у 2022 році проблема мінної забрудненості території України набула критичної гостроти. Значна частина території країни потребує обстеження та очищення від вибухонебезпечних предметів. Ця ситуація створює пряму загрозу для життя та здоров'я населення, а також блокує відновлення інфраструктури та повернення людей до звичного життя.

В умовах масштабного забруднення території виникає потреба у двох ключових можливостях. По-перше, населення потребує доступної та зрозумілої інформації про те, які території є небезпечними, які перебувають у процесі очищення, а які вже безпечні для перебування. По-друге, організації, що займаються розмінуванням – державні служби, гуманітарні місії, волонтерські структури – потребують зручного інструменту для реєстрації поточного стану робіт.

Існуючі інформаційні рішення у сфері протимінної діяльності переважно орієнтовані на професійних операторів та мають обмежену доступність для широкого загалу. Частина систем реалізована як спеціалізоване програмне забезпечення, що вимагає встановлення та спеціальної підготовки. Інші рішення, хоча й доступні через браузер, не забезпечують зручного інтерфейсу для реєстрації та управління даними.

Таким чином, існує потреба у створенні сучасної веборієнтованої інформаційної системи, що поєднує доступність для населення з функціональністю для операторів розмінування.

Метою даного дослідження є проектування та реалізація спеціалізованої багатокористувацької інформаційної системи для реєстрування процесів розмінування земельних ділянок, що забезпечує інформування населення про стан територій та надає зручний інструмент реєстрації для відповідних організацій.

Об'єктом дослідження є інформаційні системи для реєстрування та відображення процесів розмінування земельних ділянок.

Предметом дослідження є методи та технології побудови багатокористувацьких вебсистем з підтримкою геопросторових даних та інтерактивною картографічною візуалізацією.

Наукова новизна полягає в наступному:

1. Запропоновано спосіб побудови веборієнтованої геоінформаційної системи для гуманітарного розмінування, що поєднує публічний доступ населення до інформації про стан територій із функціоналом реєстрації процесів розмінування для авторизованих користувачів.
2. Розроблено багатокористувацьку модель доступу на основі RBAC для спеціалізованої системи протимінної діяльності, яка забезпечує різні рівні взаємодії (Guest, Operator, Administrator) та контрольований доступ до операційних функцій.
3. Надано архітектуру багатошарової веборієнтованої інформаційної системи для реєстрування процесів розмінування за принципами Clean Architecture з використанням технологічного стеку ASP.NET Core – PostgreSQL/PostGIS – React/TypeScript – Leaflet, що забезпечує тестованість бізнес-логіки, незалежність від конкретних фреймворків та можливість заміни окремих компонентів без зміни доменної моделі.

Практична цінність отриманих в роботі результатів полягає в тому, що розроблена інформаційна система може бути використана для:

- інформування населення про стан безпеки територій через зручний вебінтерфейс з інтерактивною картою;
- реєстрації процесів розмінування організаціями різного типу – державними службами, гуманітарними місіями, волонтерськими структурами;
- відстеження прогресу робіт з розмінування через візуалізацію статусів секторів.

Система реалізована на відкритому технологічному стеку, що забезпечує незалежність від комерційних постачальників та можливість подальшого розвитку.

Апробація результатів роботи. Положення даної роботи та проміжні результати доповідались і обговорювались на наступних конференціях:

1. Прикладна математика та комп'ютинг 2025, м. Київ, 2025.
2. IV Міжнародна науково-практична конференція «SCIENCE, TECHNOLOGY AND GLOBAL CHALLENGES», 4-6.12.2025, Токіо, Японія.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів, висновків до кожного розділу та загальних висновків по роботі в цілому, списку використаних літературних джерел та додатків.

У *вступі* подано загальну характеристику роботи, обґрунтовано актуальність створення системи для інформування населення та реєстрації процесів розмінування, сформульовано мету та задачі роботи.

У *першому* розділі проведено аналіз предметної області гуманітарного розмінування, розглянуто потреби різних категорій користувачів, проаналізовано існуючі інформаційні рішення та обґрунтовано вибір технологічного стеку.

У *другому* розділі описано методи та моделі системи: моделювання геопросторових даних, модель багатокористувацького доступу та методи візуалізації інформації на карті.

У *третьому* розділі представлено проєктування та реалізацію системи: структуру бази даних, серверну частину та клієнтський інтерфейс з інтерактивною картою.

У *четвертому* розділі проведено тестування розробленої системи та оцінку її відповідності поставленим вимогам.

У *висновках* підсумовано результати проведеної роботи та окреслено напрямки подальшого розвитку системи.

Повний обсяг дисертації – 142 сторінки, у тому числі 91 сторінка основного тексту, 20 рисунків, 9 таблиць, список використаних літературних джерел (20 найменувань), 24 слайди презентації.

Ключові слова: інформаційна система, гуманітарне розмінування, веборієнтована система, багатокористувацький доступ, геопросторові дані, картографічна візуалізація, земельні ділянки.

ABSTRACT

Topic relevance. Following the start of the full-scale invasion in 2022, the problem of mine contamination in Ukraine has become critically acute. A significant part of the country's territory requires survey and clearance of explosive remnants of war. This situation poses a direct threat to the life and health of the population and blocks infrastructure recovery and people's return to normal life.

In conditions of large-scale territorial contamination, two key capabilities are needed. First, the population needs accessible and understandable information about which territories are dangerous, which are being cleared, and which are already safe. Second, organizations involved in demining – government services, humanitarian missions, volunteer structures – need a convenient tool for registering the current state of work.

Existing information solutions in the field of mine action are mainly oriented towards professional operators and have limited accessibility for the general public. Some systems are implemented as specialized software that requires installation and special training. Other solutions, although accessible via browser, do not provide a convenient interface for data registration and management.

Thus, there is a need to create a modern web-oriented information system that combines accessibility for the population with functionality for demining operators.

The purpose of this research is to design and implement a specialized multi-user information system for registering land demining processes that provides information to the population about the state of territories and offers a convenient registration tool for relevant organizations.

The object of the research is information systems for registering and displaying land demining processes.

The subject of the research is methods and technologies for building multi-user web systems with geospatial data support and interactive cartographic visualization.

The scientific novelty of the work consists in the following:

1. A method is proposed for building a web-oriented geographic information system for humanitarian demining that combines public access for the population to information on the status of territories with registration functionality for demining processes available to authorized users.
2. A multi-user access model based on role-based access control (RBAC) has been developed for a specialized mine-action information system, providing different levels of interaction (Guest, Operator, Administrator) and controlled access to operational functions.
3. An architecture of a multi-layer web-oriented information system for registering demining processes has been presented, designed according to Clean Architecture principles and using the technology stack ASP.NET Core – PostgreSQL/PostGIS – React/TypeScript – Leaflet, which ensures testability of the business logic, independence from specific frameworks, and the possibility of replacing individual components without changing the domain model.

The practical value of the results obtained is that the developed information system can be used for:

- informing the population about the safety status of territories through a user-friendly web interface with an interactive map;
- registering demining processes by various types of organizations – government services, humanitarian missions, volunteer structures;
- tracking demining progress through visualization of sector statuses.

The system is implemented on an open technology stack, ensuring independence from commercial vendors and the possibility of further development.

Testing of the results of the work. The provisions of this work and intermediate results were reported and discussed at the following conferences:

1. Applied Mathematics and Computing 2025, Kyiv, Ukraine, 2025.

2. IV International Scientific and Practical Conference "SCIENCE, TECHNOLOGY AND GLOBAL CHALLENGES", December 4–6, 2025, Tokyo, Japan.

Structure and scope of work. The master's thesis consists of an introduction, four sections, conclusions to each section and general conclusions on the work as a whole, a list of used literature sources, and appendices.

The *introduction* provides a general description of the work, justifies the relevance of creating a system for informing the population and registering demining processes, and formulates the goal and objectives of the work.

The *first* section analyzes the subject area of humanitarian demining, examines the needs of different user categories, analyzes existing information solutions, and justifies the choice of technology stack.

The *second* section describes the system's methods and models: geospatial data modeling, multi-user access model, and methods for visualizing information on the map.

The *third* section presents the design and implementation of the system: database structure, server-side, and client interface with an interactive map.

The *fourth* section conducts testing of the developed system and evaluates its compliance with the stated requirements.

The *conclusions* summarize the results of the work and outline directions for further system development.

The full volume of the dissertation is 142 pages, including 91 pages of the main text, 20 figures, 9 tables, a list of used literary sources (20 titles), 24 presentation slides.

Keywords: information system, humanitarian demining, web-oriented system, multi-user access, geospatial data, cartographic visualization, land plots.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	4
ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ.....	8
1.1. Процеси гуманітарного розмінування.....	8
1.1.1. Типи вибухонебезпечних предметів та методи розмінування.....	8
1.1.2. Міжнародні стандарти IMAS та національне законодавство.....	10
1.1.3. Роль інформаційних систем у процесах розмінування	12
1.2. Аналіз існуючих інформаційних систем.....	13
1.2.1. IMSMA: функціональність та недоліки.....	13
1.2.2. Національна система ua.imsma.org: порівняльний аналіз.....	14
1.2.3. Міжнародний досвід (Хорватія, Камбоджа).....	17
1.2.4. Порівняльна таблиця існуючих рішень	18
1.3. Технології розробки геоінформаційних вебсистем.....	20
1.3.1. Backend-фреймворки: ASP.NET Core, Node.js, Django - порівняння	20
1.3.2. Геопросторові СУБД: PostgreSQL + PostGIS.....	21
1.3.3. Картографічні бібліотеки: Leaflet.js, OpenLayers, Mapbox GL JS	22
1.3.4. Обґрунтування вибору технологічного стеку	22
1.4. Визначення вимог до системи	23
1.4.1. Функціональні вимоги	24
1.4.2. Нефункціональні вимоги	25
Висновки до розділу 1	26
РОЗДІЛ 2. МЕТОДИ ТА МОДЕЛІ СИСТЕМИ.....	28
2.1. Моделювання геопросторових даних	28
2.1.1. Векторна модель представлення територій	28
2.1.2. Модель секторів розмінування	30

2.1.3. Геопросторові операції.....	31
2.2. Оптимізація роботи з геопросторовими даними	33
2.2.1. Просторова індексація	34
2.2.2. Тайлова система відображення карти	35
2.3. Модель багатокористувацького доступу.....	37
2.3.1. Рівні доступу користувачів	37
2.3.2. Матриця дозволів	39
2.4. Методи візуалізації даних на карті.....	40
2.4.1. Кольорове кодування статусів секторів	41
2.4.2. Кластеризація маркерів	42
2.4.3. Статистичні показники	43
Висновки до розділу 2	44
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ.....	46
3.1. Архітектура системи	46
3.1.1. Clean Architecture та компоненти високого рівня.....	46
3.1.2. Технологічний стек та обґрунтування вибору	49
3.2. Проєктування бази даних	51
3.2.1. Концептуальна модель даних.....	52
3.2.2. Логічна модель з геопросторовими типами	53
3.2.3. Фізична модель та індексація.....	55
3.3. Реалізація серверної частини	58
3.3.1. Структура проєкту та REST API	58
3.3.2. Геопросторові запити з NetTopologySuite	62
3.3.3. Автентифікація та авторизація	64
3.4. Реалізація клієнтської частини	67
3.4.1. Структура застосунку та компоненти	67
3.4.2. Інтеграція карти та форми реєстрації секторів.....	69
3.4.3. Адаптивний інтерфейс та скріншоти системи.....	72

	3
Висновки до розділу 3	76
РОЗДІЛ 4. ТЕСТУВАННЯ ТА ОЦІНКА ЯКОСТІ СИСТЕМИ	78
4.1. Методика тестування	78
4.1.1. Тестове середовище	78
4.1.2. Інструмент Google Lighthouse	78
4.2. Результати тестування продуктивності	79
4.2.1. Загальні показники	79
4.2.2. Детальний аналіз метрик продуктивності	80
4.2.3. Фактори оптимізації продуктивності	81
4.3. Аналіз доступності (Accessibility)	82
4.4. Аналіз найкращих практик (Best Practices)	82
4.5. Аналіз SEO-оптимізації	83
4.6. Тестування серверної частини	84
4.7. Функціональна верифікація	84
4.7.1. Верифікація картографічного модуля	85
4.7.2. Верифікація розмежування доступу	85
4.8. Порівняння з галузевими стандартами	85
Висновки до розділу 4	86
ВИСНОВКИ	87
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	90
ДОДАТОК А. ФРАГМЕНТИ ПРОГРАМНОГО КОДУ	92
ДОДАТОК Б. ТЕЗИ ДЛЯ КОНФЕРЕНЦІЇ ПМК-2025	98
ДОДАТОК В. ТЕЗИ ДЛЯ КОНФЕРЕНЦІЇ «SCIENCE, TECHNOLOGY AND GLOBAL CHALLENGES»	103
ДОДАТОК Г. ПРЕЗЕНТАЦІЯ	108

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

API (Application Programming Interface) – програмний інтерфейс застосунку

CRUD (Create, Read, Update, Delete) – базові операції роботи з даними

GeoJSON (Geographic JSON) – геопросторовий формат обміну даними

GiST (Generalized Search Tree) – узагальнене пошукове дерево

HTTPS (Hypertext Transfer Protocol Secure) – захищений протокол передачі гіпертексту

IMAS (International Mine Action Standards) – міжнародні стандарти протимінної діяльності

IMSMA (Information Management System for Mine Action) – інформаційна система управління протимінною діяльністю

JWT (JSON Web Token) – вебтокен у форматі JSON

OGC (Open Geospatial Consortium) – відкритий геопросторовий консорціум

PostGIS – геопросторове розширення для PostgreSQL

RBAC (Role-Based Access Control) – розмежування доступу на основі ролей

REST (Representational State Transfer) – архітектурний стиль передачі репрезентативного стану

SPA (Single Page Application) – односторінковий вебзастосунок

WCAG (Web Content Accessibility Guidelines) – настанови з доступності вебконтенту

WGS84 (World Geodetic System 1984) – світова геодезична система 1984

ВНП – вибухонебезпечні предмети

ГІС – геоінформаційна система

ДСНС – Державна служба України з надзвичайних ситуацій

СУБД – система управління базами даних

ВСТУП

Проблема гуманітарного розмінування територій є однією з найбільш гострих гуманітарних криз сучасної України. Після початку повномасштабного вторгнення у 2022 році значна частина території країни опинилася забрудненою вибухонебезпечними предметами різних типів. Ця ситуація створює пряму загрозу для життя та здоров'я мільйонів людей, блокує відновлення інфраструктури, повернення переселенців та нормальне функціонування економіки постраждалих регіонів.

В умовах такого масштабного забруднення виникають дві взаємопов'язані потреби. З одного боку, населення потребує доступної та зрозумілої інформації про стан безпеки територій – де є небезпека, де тривають роботи з очищення, а де вже безпечно перебувати. Люди мають знати, куди можна повертатися, де можна вести господарську діяльність, а яких ділянок слід уникати. З іншого боку, організації, що займаються розмінуванням – державні служби на кшталт ДСНС, міжнародні гуманітарні місії, волонтерські структури – потребують зручного інструменту для реєстрації поточного стану робіт та координації зусиль.

Існуючі інформаційні рішення у сфері протимінної діяльності створювалися переважно для потреб професійних операторів розмінування. Спеціалізовані системи, такі як IMSMA, забезпечують широкий функціонал для планування та обліку робіт, проте реалізовані як desktop-додатки, що вимагають встановлення та спеціальної підготовки користувачів. Це робить їх недоступними для широкого загалу та ускладнює використання в польових умовах. Водночас існуючі веборієнтовані рішення, хоча й доступні через браузер, не завжди забезпечують зручний інтерфейс для реєстрації даних та управління інформацією про стан територій.

Таким чином, існує розрив між потребами населення в інформації та можливостями існуючих систем. Населення не має простого способу дізнатися про стан безпеки конкретної території, а організації, що працюють над розмінуванням,

не мають єдиного зручного вебінструменту для реєстрації прогресу робіт, що був би доступний без встановлення додаткового програмного забезпечення.

Актуальність розробки нової інформаційної системи обумовлена необхідністю створення сучасного веборієнтованого рішення, що поєднує дві ключові можливості: надання населенню доступної інформації про стан територій через інтерактивну карту та забезпечення зручного інструменту реєстрації для операторів розмінування.

Об'єктом дослідження є інформаційні системи для реєстрування та відображення процесів розмінування земельних ділянок.

Предметом дослідження є методи та технології побудови багатокористувацьких вебсистем з підтримкою геопросторових даних та інтерактивною картографічною візуалізацією.

Мета роботи полягає у проєктуванні та реалізації спеціалізованої багатокористувацької веборієнтованої інформаційної системи для реєстрування процесів розмінування земельних ділянок, що забезпечує:

- інформування населення про стан безпеки територій через доступний вебінтерфейс;
- зручний інструмент реєстрації для організацій, що займаються розмінуванням;
- візуалізацію інформації на інтерактивній карті.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести аналіз предметної області гуманітарного розмінування та потреб різних категорій користувачів системи.
2. Виконати аналіз існуючих інформаційних рішень у сфері протимінної діяльності.
3. Обґрунтувати вибір технологічного стеку для розробки веборієнтованої системи.
4. Розробити багатокористувацьку модель доступу для різних категорій користувачів.

5. Спроекувати модель даних для обліку секторів розмінування з підтримкою геопросторових типів.
6. Реалізувати веборієнтовану систему з інтерактивною картою.
7. Провести тестування та оцінку розробленої системи.

Методи дослідження включають системний аналіз предметної області, аналіз існуючих рішень, об'єктно-орієнтоване проєктування та експериментальне тестування.

Наукова новизна роботи полягає у розробці підходу до побудови веборієнтованої інформаційної системи, що поєднує публічний доступ до інформації про стан територій з функціоналом реєстрації для авторизованих користувачів, а також у розробці багатокористувацької моделі доступу для різних категорій користувачів.

Практична цінність полягає у створенні системи, що може бути використана для інформування населення про безпеку територій та реєстрації процесів розмінування організаціями різного типу.

Структура роботи. Магістерська дисертація складається зі вступу, чотирьох розділів, висновків до кожного розділу та загальних висновків, списку використаних літературних джерел та додатків.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1. Процеси гуманітарного розмінування

Гуманітарне розмінування є комплексним процесом, що охоплює виявлення, ідентифікацію та знешкодження вибухонебезпечних предметів на територіях, які постраждали від збройних конфліктів. Цей процес має критичне значення для відновлення нормального життя на постраждалих територіях, оскільки наявність вибухонебезпечних предметів унеможливорює сільськогосподарську діяльність, будівництво інфраструктури та безпечне проживання населення. Розуміння специфіки цих процесів є необхідною передумовою для проєктування ефективної інформаційної системи, здатної забезпечити координацію зусиль різних організацій та облік виконаних робіт.

1.1.1. Типи вибухонебезпечних предметів та методи розмінування

Проблема мінної забрудненості залишається однією з найгостріших гуманітарних криз сучасності. За даними Landmine Monitor, десятки країн світу потерпають від наслідків мінування територій [1]. Вибухонебезпечні предмети, що залишаються на територіях після завершення активних бойових дій, продовжують становити загрозу для цивільного населення протягом десятиліть після закінчення конфліктів.

Протипіхотні міни становлять найбільшу загрозу для цивільного населення через свою масовість та різноманітність конструкцій. Ці пристрої спеціально розроблені для ураження людей і можуть спрацювати від мінімального тиску або руху. Протитанкові міни призначені для ураження бронетехніки та мають значно більшу потужність вибухової речовини, для їх спрацювання потрібне значне механічне зусилля. Проте вони також становлять небезпеку для цивільного населення, особливо при використанні важкої сільськогосподарської техніки.

Невибухлі боєприпаси (UXO) становлять окрему категорію вибухонебезпечних предметів, до якої належать артилерійські снаряди, авіаційні бомби, реактивні снаряди та гранати, що з тих чи інших причин не спрацювали під час бойового застосування. Ці предмети є особливо небезпечними через свою непередбачуваність – вони можуть детонувати від найменшого зовнішнього впливу після тривалого перебування в ґрунті.

Саморобні вибухові пристрої набули широкого розповсюдження в сучасних конфліктах через свою технологічну доступність та різноманітність конструкцій. Їх виявлення є особливо складним завданням через відсутність стандартизованих характеристик та можливість використання нестандартних матеріалів і механізмів спрацювання.

Методи розмінування території відрізняються залежно від типу місцевості та характеру забруднення. Кожен метод має свої переваги та обмеження, що визначає сферу його оптимального застосування. Механічне розмінування передбачає використання спеціалізованої техніки – мінних тралів, котків та роторних культиваторів, що забезпечує значну продуктивність обробки території. Цей метод є найефективнішим на відкритих рівнинних територіях з однорідним ґрунтом.

Ручне розмінування забезпечує найвищий рівень контролю при виявленні та знешкодженні вибухонебезпечних предметів, особливо у складних умовах місцевості. Сапери використовують металодетектори, щупи та інші інструменти для послідовного обстеження території. Цей метод є незамінним у забудованих районах, на схилах та в місцях з густою рослинністю, де застосування техніки неможливе або обмежене.

Біологічний метод розмінування з використанням собак-міношукачів застосовується для пошуку вибухових речовин на великих територіях. Тварини здатні виявляти вибухові речовини за запахом, що дозволяє швидко обстежувати значні площі. Цей метод часто використовується для первинного обстеження території перед застосуванням інших методів очищення.

Метод	Продуктивність	Точність	Застосування
Ручне	20-100 м²/день	99.6%	Всі типи місцевості
Механічне	1000+ м²/день	~95%	Відкриті території
Біологічне (собаки)	до 1000 м²/день	>95%	Різноманітні умови

Рисунок 1.1 – Порівняння методів розмінування

Сучасна практика гуманітарного розмінування базується на комбінованому підході, де різні методи застосовуються послідовно залежно від етапу робіт та характеристик території. На початковому етапі нетехнічного обстеження території збирається інформація про ймовірні місця розташування мінних полів шляхом опитування місцевого населення та аналізу історичних даних. Технічне обстеження передбачає детальне вивчення підозрілих ділянок з використанням міношукачів та георадарів. Завершальним етапом є безпосереднє очищення території з використанням найбільш доцільних методів залежно від виявлених загроз.

1.1.2. Міжнародні стандарти IMAS та національне законодавство

Міжнародні стандарти протимінної діяльності IMAS являють собою комплексну систему вимог, рекомендацій та найкращих практик, що охоплюють усі аспекти гуманітарного розмінування [3]. Ці стандарти розроблені для забезпечення безпеки персоналу, ефективності операцій та якості результатів робіт з очищення територій. Перша редакція IMAS була випущена у жовтні 2001 року в системі ООН (UNMAS) за підтримки Женевського міжнародного центру гуманітарного розмінування (GICHD) [3]. З того часу стандарти постійно оновлюються з урахуванням накопиченого досвіду та технологічного розвитку.

Стандарт IMAS 08.10 регламентує процедури нетехнічного обстеження – збір та аналіз інформації про підозрілі території без безпосереднього фізичного втручання [4]. Цей етап є критично важливим для планування подальших робіт та

оптимального розподілу ресурсів. Стандарт IMAS 08.20 визначає вимоги до технічного обстеження з використанням різних методів (ручного, механічного, із застосуванням тварин) [5]. Технічне обстеження дозволяє підтвердити або спростувати наявність вибухонебезпечних предметів на території та визначити необхідність проведення очищення. Стандарт IMAS 09.10 визначає "очищену" територію як таку, з якої видалено всі міни та вибухонебезпечні залишки війни до заданої глибини [6]. Дотримання цього стандарту є обов'язковою умовою для сертифікації території як безпечної.

Геопросторовий компонент займає особливе місце в системі IMAS, оскільки точна географічна прив'язка інформації є критично важливою для безпеки та ефективності операцій. Стандарти вимагають, щоб національний орган визначав і документував вимоги до точності позиційних даних та забезпечував їх досягнення [4]. Це включає визначення системи координат, точності вимірювань та процедур верифікації географічних даних.

Національне законодавство України базується на Законі "Про протимінну діяльність в Україні", прийнятому у грудні 2018 року та доповненому у 2019–2023 роках [7]. Закон визначає інституційну архітектуру протимінної діяльності, включаючи національний орган з питань протимінної діяльності, центр протимінної діяльності при Міністерстві оборони та центр гуманітарного розмінування при органі у сфері цивільного захисту. Ця структура забезпечує координацію зусиль державних органів, міжнародних організацій та приватних операторів розмінування.

Технічний регламент з гуманітарного розмінування адаптує положення IMAS до специфічних умов України, враховуючи особливості географії, клімату та характер мінного забруднення. Регламент визначає процедури акредитації операторів, вимоги до обладнання та кваліфікації персоналу, а також порядок документування та звітності.

1.1.3. Роль інформаційних систем у процесах розмінування

Масштаби мінної забрудненості України після початку повномасштабного вторгнення створили безпрецедентні виклики для системи управління протимінною діяльністю. За оцінками міжнародних експертів, процес повного розмінування території країни може тривати десятиліття. Така тривалість робіт вимагає надійної системи обліку та координації, здатної функціонувати протягом багатьох років та забезпечувати наступність інформації при зміні персоналу та організацій.

Множинність операторів розмінування створює критичну потребу в єдиній системі обліку виконаних робіт. На території України одночасно працюють державні служби, міжнародні гуманітарні організації та приватні компанії. Без централізованої бази даних виникають ризики дублювання зусиль, коли різні організації незалежно обстежують одну й ту саму територію, або навпаки – прогалини у покритті, коли певні ділянки залишаються необстеженими через відсутність координації.

Функціональність сучасної інформаційної системи охоплює весь життєвий цикл робіт від первинного виявлення підозрілої території до остаточного підтвердження її безпечності. Система має забезпечувати реєстрацію повідомлень про виявлення вибухонебезпечних предметів, планування обстежень та очищення, облік виконаних робіт та формування звітності. Геопросторовий компонент є фундаментальною основою – на відміну від традиційних інформаційних систем, тут координати визначають саму суть інформації, оскільки кожен запис пов'язаний з конкретним місцем на місцевості.

Вимоги до точності координат є суворішими порівняно з багатьма іншими застосуваннями геоінформаційних систем. Помилка у визначенні місця розташування виявленого боєприпасу навіть на кілька метрів може мати фатальні наслідки, оскільки сапери можуть не знайти об'єкт під час повернення для

знешкодження. Аналогічно, неточне позначення меж очищеної території може призвести до виходу людей за межі безпечної зони.

1.2. Аналіз існуючих інформаційних систем

Аналіз існуючих рішень у сфері інформаційного забезпечення протимінної діяльності дозволяє виявити сильні сторони та обмеження кожного підходу, а також сформулювати вимоги до розроблюваної системи. Дослідження охоплює як глобальні платформи, що використовуються в багатьох країнах світу, так і національні системи, адаптовані до специфічних умов окремих держав.

1.2.1. IMSMA: функціональність та недоліки

Інформаційна система IMSMA є одним з найпоширеніших програмних рішень для управління протимінною діяльністю у світі [8]. Розробка системи була ініційована Женевським міжнародним центром гуманітарного розмінування (GICHD) з метою створення універсального інструменту для координації зусиль з розмінування на глобальному рівні. IMSMA має кілька поколінь: ранні версії (IMSMA NG) були desktop-орієнтованими та вимагали встановлення на локальні комп'ютери, тоді як сучасна версія IMSMA Core, запущена у лютому 2018 року, є вебплатформою на базі ArcGIS завдяки стратегічному партнерству GICHD з компанією Esri.

Функціональна структура IMSMA організована у вигляді модулів, кожен з яких відповідає за окремий аспект управління протимінною діяльністю: Land Release для управління процесом звільнення територій від загрози вибухонебезпечних предметів, Victim Assistance для обліку постраждалих та координації допомоги їм, Quality Management для контролю якості виконаних робіт та забезпечення відповідності стандартам. Така модульна архітектура дозволяє

національним програмам обирати необхідні компоненти залежно від специфіки їх потреб.

Суттєві обмеження ранніх версій системи включали відсутність вебінтерфейсу, що ускладнювало доступ до даних з польових умов, складність розгортання та підтримки, що вимагало наявності кваліфікованого ІТ-персоналу, та тривалий курс підготовки операторів через складність інтерфейсу. Ці обмеження стимулювали розробку нової версії IMSMA Core з веборієнтованою архітектурою.

Попри недоліки, IMSMA залишається одним з найповніших та найпоширеніших рішень для управління протимінною діяльністю, що використовується в десятках країн світу. Проблеми IMSMA пов'язані радше з технологічною платформою реалізації, ніж з концептуальними помилками в моделюванні предметної області. Накопичений досвід експлуатації системи є цінним джерелом інформації для розробки нових рішень.

1.2.2. Національна система ua.imsma.org: порівняльний аналіз

Національна інформаційна система протимінної діяльності України ua.imsma.org функціонує як вебдодаток – значний крок вперед порівняно з desktop-версією IMSMA. Користувачі можуть отримувати доступ до системи з будь-якого пристрою з браузером, що суттєво знижує технічні бар'єри для участі в загальнонаціональній системі обліку.

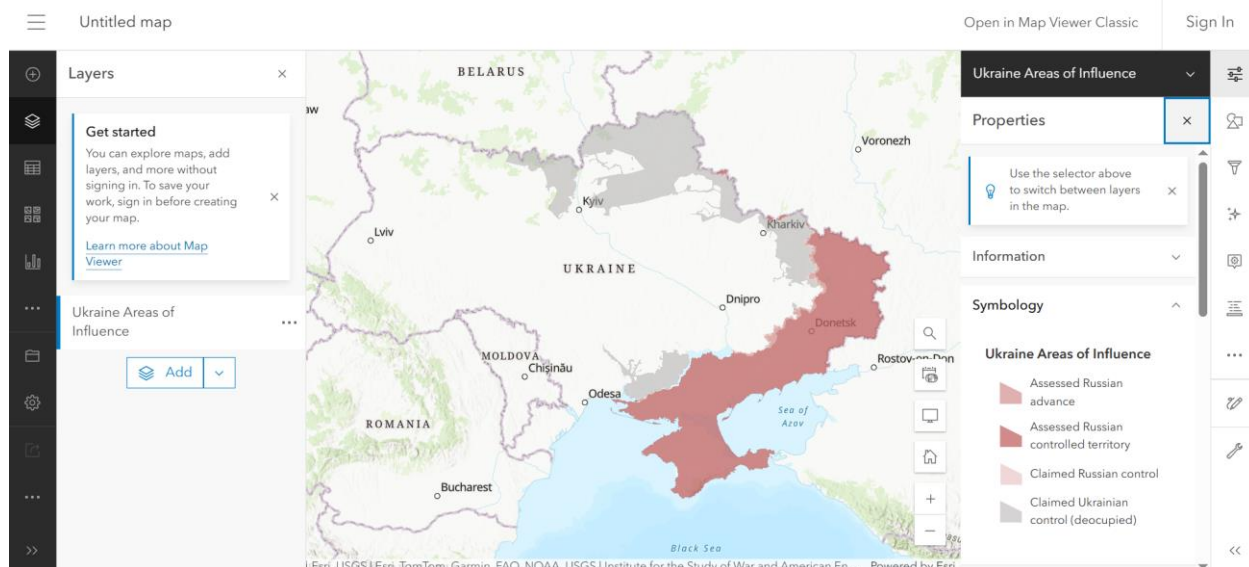


Рисунок 1.2 – Графічний інтерфейс ua.imsma.org

Веборієнтована архітектура ua.imsma.org надає суттєві переваги порівняно з традиційним desktop-додатком IMSMA. Користувачі можуть отримувати доступ до системи з будь-якого пристрою з браузером та підключенням до інтернету, що різко знижує технічні бар'єри для участі в загальнонаціональній системі обліку. Оператори розмінування можуть вносити дані безпосередньо зі своїх польових офісів без необхідності встановлення спеціалізованого програмного забезпечення чи підтримки складної IT-інфраструктури. Це особливо важливо для невеликих організацій з обмеженими технічними ресурсами.

Інтеграція з OpenStreetMap як картографічної основи є ще однією важливою перевагою ua.imsma.org. OpenStreetMap надає детальну топографічну інформацію для території України, включаючи дорожню мережу, населені пункти, водойми та інші об'єкти інфраструктури. Використання відкритої картографічної платформи забезпечує актуальність даних та незалежність від комерційних постачальників картографічних сервісів.

Водночас практичне використання ua.imsma.org виявило певні обмеження, що впливають на ефективність системи як інструменту координації національних зусиль з розмінування. Серед виявлених аспектів, що потребують уваги, – неповне покриття території даними та варіативність географічної точності інформації

залежно від джерела даних. Ці обмеження є типовими для систем, що знаходяться на етапі активного розвитку та наповнення даними.

Обмежені можливості роботи з секторами території ускладнюють процес планування та обліку робіт. Міжнародна практика гуманітарного розмінування передбачає поділ великих підозрілих територій на менші сектори для систематичного обстеження та очищення. Такий підхід дозволяє точно відстежувати прогрес робіт, розподіляти завдання між різними бригадами та забезпечувати повноту покриття території. Важливо, щоб інтерфейс системи дозволяв зручно вводити координати секторів вручну з можливістю редагування їхніх меж, а також відображав поточний статус кожного сектору.

Швидкість оновлення інформації в системі є ще одним аспектом, що впливає на користувацький досвід. За окремими даними, інтервал між внесенням інформації та її відображенням у публічному інтерфейсі може варіюватися, що пов'язано з особливостями архітектури та процесами верифікації даних. Для забезпечення безпеки населення критично важливо, щоб інформація про виявлені небезпечні об'єкти та очищені території оновлювалася максимально оперативно.

Інтерфейс ua.imsma.org є більш сучасним порівняно з desktop-версією IMSMA, хоча зберігає потенціал для подальшого спрощення. Процес реєстрації нового об'єкта передбачає заповнення численних полів, що забезпечує повноту даних, але може ускладнювати роботу для користувачів без спеціальної підготовки. Розширення контекстної довідки та впровадження покрокових майстрів могло б полегшити освоєння системи новими користувачами.

Мобільний доступ є ще одним напрямком для потенційного вдосконалення. Вебінтерфейс доступний зі смартфонів та планшетів, проте адаптація макету для малих екранів могла б суттєво покращити досвід користувачів у польових умовах, де мобільні пристрої часто є єдиним доступним засобом зв'язку.

Аналіз публічно доступних матеріалів та звітів організацій, що працюють у сфері гуманітарного розмінування, вказує на наявність потенціалу для

вдосконалення системи в напрямках зручності інтерфейсу, оперативності оновлення даних та функціональності роботи з секторами території.

1.2.3. Міжнародний досвід (Хорватія, Камбоджа)

Landmine Monitor [1] відзначає прогрес низки країн у вирішенні проблеми мінної забрудненості. Вивчення досвіду країн, що успішно справляються з проблемою мінного забруднення, дозволяє виявити найкращі практики та уникнути типових помилок при розробці власних рішень.

Хорватія, де війна 1991–1995 років залишила значні площі замінованих територій, досягла значних результатів у розмінуванні та є прикладом системного підходу до вирішення проблеми. Для інформаційного супроводу використовується система CROMAC GIS – вебплатформа з геопросторовими можливостями (за оглядом відкритого інтерфейсу системи). Ключовою особливістю хорватської системи є детальність картографічної інформації та тісна інтеграція з кадастровими даними, що дозволяє відстежувати статус очищення на рівні окремих земельних ділянок.



Рисунок 1.3 – Приклад інтерфейсу CROMAC GIS

Камбоджа, яка має одне з найбільших мінних забруднень у світі внаслідок тривалих конфліктів протягом другої половини XX століття, використовує адаптовану версію IMSMA (E-IMSMA). Особливістю камбоджійського досвіду є інтеграція системи з мобільними додатками для збору даних у польових умовах, що дозволяє оперативно реєструвати інформацію безпосередньо на місці виявлення вибухонебезпечних предметів.

Порівняння досвіду різних країн виявляє універсальні закономірності: критично важлива адаптація системи до специфічного контексту країни, врахування особливостей інституційного середовища та довгострокова інституційна підтримка впровадження. Успішні системи характеризуються поєднанням функціональної повноти з простотою використання, що забезпечує їх прийняття широким колом користувачів.

1.2.4. Порівняльна таблиця існуючих рішень

На основі аналізу документації та публікацій складено порівняльну характеристику розглянутих систем. Оцінки є авторськими узагальненнями на основі доступної інформації та можуть варіюватися залежно від конкретних умов впровадження. Таблиця надає структурований огляд ключових характеристик, що допомагає виявити переваги та обмеження кожного рішення.

Таблиця 1.1 – Порівняльний аналіз існуючих інформаційних систем протимінної діяльності

Критерій	IMSMA	ua.imsma.org	CROMAC GIS	E-IMSMA
Тип доступу	Desktop / Веб (Core)	Вебінтерфейс	Вебінтерфейс	Desktop + мобільний
Оперативність оновлення	Залежить від версії	Н/д	Н/д	Н/д

Складність інтерфейсу	Висока	Висока	Середня	Середня
Мобільний доступ	Обмежений	Обмежений	Адаптивний	Нативний додаток
Підтримка стандартів IMAS	Повна	Часткова	Повна	Повна
Картографічна основа	Esri / ArcGIS	OpenStreetMap	OpenStreetMap	Google Maps
Ліцензування	Через GICHD/Esri	Безкоштовна	Закрита	Безкоштовна
Інтеграція з іншими системами	Через Esri	Обмежена	Є	Є
Підтримка відкритих форматів	Shapefile, CSV	Обмежена	GeoJSON, GML, KML	Shapefile, KML

Аналіз виявляє закономірність між архітектурним підходом та практичною ефективністю системи. Веборієнтовані рішення демонструють переваги з точки зору доступності та простоти розгортання, що особливо важливо для організацій з обмеженими технічними ресурсами. Оперативність оновлення даних є критичною для безпеки – затримки в оновленні інформації про небезпечні території можуть призвести до небезпечних ситуацій. Жодна з розглянутих систем не є універсально оптимальною: IMSMA Core є найповнішою з функціональної точки зору, але залежить від комерційної інфраструктури Esri; CROMAC GIS демонструє позитивний досвід та зручний інтерфейс, але є закритою системою, недоступною для використання іншими країнами.

1.3. Технології розробки геоінформаційних вебсистем

Вибір технологічного стеку є критичним етапом проєктування геоінформаційної системи, який значною мірою визначає можливості, продуктивність та вартість підтримки рішення протягом усього життєвого циклу. Сучасний ринок пропонує різноманітні рішення для серверної та клієнтської частин, кожне з яких має свої переваги та обмеження. Обґрунтований вибір технологій вимагає врахування специфічних вимог предметної області, наявних ресурсів та довгострокових перспектив розвитку системи.

1.3.1. Backend-фреймворки: ASP.NET Core, Node.js, Django - порівняння

ASP.NET Core є кросплатформною реінкарнацією класичного .NET Framework, що дозволяє розгортати застосунки на Windows, Linux та macOS. Продуктивність ASP.NET Core є високою серед сучасних вебфреймворків завдяки оптимізованому runtime та підтримці асинхронного програмування [9]. Типобезпека C# забезпечує виявлення помилок на етапі компіляції, що зменшує кількість runtime-помилки та спрощує рефакторинг коду. Інтеграція з Entity Framework Core надає потужні засоби для роботи з базами даних.

Node.js представляє підхід, побудований навколо JavaScript як єдиної мови для клієнтської та серверної частин, що дозволяє використовувати одну мову програмування та спільні бібліотеки на всіх рівнях застосунку. Продуктивність прийнятна завдяки V8 engine від Google, проте однопотокова природа означає, що процесорно-інтенсивні геопросторові розрахунки можуть блокувати обробку інших запитів. Велика екосистема npm-пакетів забезпечує швидкий старт розробки, хоча якість пакетів може значно варіюватися.

Django пропонує філософію "batteries included" з великою кількістю вбудованої функціональності, включаючи адміністративний інтерфейс, систему автентифікації та ORM. GeoDjango є вбудованим модулем для геоінформаційних

вебдодатків, що надає інтеграцію з PostGIS та GDAL. Продуктивність може поступатися ASP.NET Core через інтерпретовану природу Python, хоча для більшості застосувань це не є критичним обмеженням.

1.3.2. Геопросторові СУБД: PostgreSQL + PostGIS

PostgreSQL є об'єктно-реляційною СУБД з понад тридцятирічною історією розвитку та репутацією надійної платформи для критичних застосувань. Система підтримує складні типи даних, транзакції з повною підтримкою ACID та розширюваність через механізм розширень. PostGIS – розширення PostgreSQL, що додає повноцінну підтримку геопросторових об'єктів відповідно до специфікації OGC Simple Features та стандарту SQL/MM Part 3: Spatial [2]. Розширення надає сотні функцій для роботи з просторовими об'єктами та є де-факто стандартом для геопросторових баз даних у світі відкритого програмного забезпечення.

Просторові операції PostGIS включають широкий спектр функціональності: топологічні предикати (ST_Contains, ST_Intersects, ST_Within) для визначення просторових відношень між об'єктами, просторовий аналіз (ST_Buffer, ST_Union) для створення та маніпулювання геометриями, вимірювання (ST_Area, ST_Distance) для обчислення площ та відстаней. Індexсація реалізована через GiST з R-tree алгоритмом, що забезпечує ефективний пошук об'єктів за їх просторовим розташуванням.

Вибір PostGIS базується на відповідності стандартам OGC, що гарантує сумісність з широким спектром геоінформаційних інструментів, відкритому коді та безкоштовному ліцензуванні, що знижує вартість володіння системою, а також на високій продуктивності при роботі з великими обсягами геопросторових даних.

1.3.3. Картографічні бібліотеки: Leaflet.js, OpenLayers, Mapbox GL JS

Leaflet.js є легковісною JavaScript-бібліотекою з фокусом на простоту використання та мобільні пристрої [11]. Бібліотека має невеликий розмір та швидко завантажується навіть при повільному з'єднанні, що є критично важливим для польових умов. Екосистема плагінів включає Leaflet.draw для малювання геометрій користувачем, Leaflet.markercluster для кластеризації великої кількості маркерів, Leaflet.heat для створення теплових карт. Документація є зрозумілою та містить численні приклади, що спрощує освоєння бібліотеки.

OpenLayers є більш потужною бібліотекою з підтримкою векторних тайлів та нативною інтеграцією з OGC-сервісами (WMS, WFS, WMTS). Ця бібліотека краще підходить для складних картографічних застосувань, що вимагають роботи з різноманітними джерелами геопросторових даних. Проте більша складність бібліотеки підвищує поріг входження для розробників.

Mapbox GL JS використовує WebGL для апаратного прискорення та тривимірної візуалізації з найкращою продуктивністю при відображенні складних карт. Бібліотека забезпечує плавну анімацію та можливість відображення великої кількості об'єктів без втрати продуктивності. Проте використання комерційної платформи Mapbox може створювати залежність від постачальника.

Leaflet є оптимальним вибором для проектів, де пріоритетом є швидкість розробки, легковісність та простота підтримки. Бібліотека забезпечує всю необхідну функціональність для відображення інтерактивних карт з мінімальними накладними витратами.

1.3.4. Обґрунтування вибору технологічного стеку

Для серверної платформи вибрано ASP.NET Core 8.0 завдяки найвищій продуктивності серед розглянутих фреймворків, надійності промислового рівня та інтеграції з геопросторовими бібліотеками через NetTopologySuite.

Екосистема .NET забезпечує довгострокову підтримку та регулярні оновлення безпеки.

PostgreSQL 16 з PostGIS 3.4 обрано як геопросторову СУБД завдяки відповідності стандартам OGC, що гарантує сумісність з іншими геоінформаційними системами, відкритій ліцензії, що знижує вартість володіння, та високій продуктивності при роботі з просторовими запитам.

Frontend-фреймворк React 18 з TypeScript обрано завдяки найширшій екосистемі компонентів серед сучасних JavaScript-фреймворків та статичній типізації для роботи зі складними геопросторовими структурами даних [12]. TypeScript забезпечує виявлення типових помилок на етапі компіляції та покращує підтримуваність коду.

Leaflet.js 1.9 обрано як оптимальний баланс між функціональністю та простотою, з легковісністю, критичною для польових умов з обмеженою пропускну здатністю каналів зв'язку.

Обраний стек ASP.NET Core + PostgreSQL/PostGIS + React/TypeScript + Leaflet.js представляє зрілу, продуктивну та повністю відкриту платформу, що забезпечує незалежність від комерційних постачальників та можливість адаптації до специфічних вимог проекту.

1.4. Визначення вимог до системи

На основі проведеного аналізу предметної області та існуючих рішень сформульовано вимоги до розроблюваної системи реєстрування процесів розмінування. Вимоги структуровано за категоріями для забезпечення повноти охоплення всіх аспектів функціонування системи.

1.4.1. Функціональні вимоги

Система має забезпечувати дві основні функції: інформування населення про стан безпеки територій та надання інструменту реєстрації для організацій розмінування. Диференційований підхід до рівнів доступу балансує відкритість інформації з необхідністю захисту даних про незавершені операції.

Рівень доступу Guest призначений для широкої громадськості та дозволяє переглядати інформацію про стан безпеки територій без можливості внесення змін. Користувачі цього рівня можуть переглядати карту з позначенням небезпечних та очищених територій, отримувати інформацію про поточний статус секторів та планувати безпечні маршрути.

Рівень доступу Operator надається акредитованим організаціям, що виконують роботи з розмінування. Реєстрація процесу очищення є центральною функцією: система має підтримувати зручний інтерфейс для введення координат секторів на інтерактивній карті або у текстовому форматі з можливістю редагування введених даних. Кожен сектор отримує унікальний ідентифікатор та проходить життєвий цикл зі станами: небезпечний (Hazardous), в процесі (InProgress), очищений (Cleared), верифікований (Verified). Система забезпечує перегляд історії змін кожного об'єкту з повним аудит-логом, що фіксує всі операції з датою, часом та ідентифікатором користувача.

Рівень доступу Administrator призначений для координаторів програм розмінування та передбачає розширені можливості управління системою: створення облікових записів операторів з призначенням ролей та дозволів, перегляд системних логів для моніторингу активності користувачів, налаштування параметрів системи та управління довідниками.

1.4.2. Нефункціональні вимоги

Нефункціональні вимоги визначають якісні характеристики системи, що є критично важливими для практичної придатності рішення в умовах реальної експлуатації. Ці вимоги охоплюють продуктивність, безпеку, доступність та сумісність системи.

Вимоги до продуктивності передбачають забезпечення комфортного досвіду взаємодії навіть за умов нестабільного інтернет-з'єднання, типового для польових умов роботи операторів розмінування. Цільові показники швидкодії: допоміжна метрика First Contentful Paint (FCP) – до 1,8 секунди, Core Web Vital метрика Largest Contentful Paint (LCP) – до 2,5 секунди [10]. Система має забезпечувати плавну взаємодію з картографічним інтерфейсом без відчутних затримок при масштабуванні та переміщенні, що є критичним для комфортної роботи з картою.

Вимоги до безпеки базуються на необхідності захисту чутливої інформації про розташування вибухонебезпечних предметів від несанкціонованого доступу та модифікації. Автентифікація користувачів має реалізовуватися через механізм JSON Web Tokens (JWT), що забезпечує безстанову перевірку ідентичності та природне масштабування серверної частини. Всі з'єднання мають використовувати шифрування HTTPS для захисту даних під час передачі. Розмежування доступу за ролями (RBAC) повинно унеможливлювати несанкціоновану модифікацію даних, забезпечуючи, щоб кожен користувач мав доступ лише до функцій, передбачених його роллю.

Вимоги до доступності передбачають відповідність стандартам WCAG 2.1 для забезпечення можливості використання системи людьми з обмеженими можливостями. Інтерфейс має підтримувати навігацію за допомогою клавіатури для користувачів, що не можуть використовувати миш, використовувати семантичну HTML-розмітку для коректної роботи з екранними читачами та забезпечувати достатню контрастність кольорів для користувачів з порушеннями зору.

Вимоги до сумісності охоплюють підтримку актуальних версій браузерів Chrome, Firefox, Safari та Edge, що забезпечує доступність системи для переважної більшості користувачів. Геопросторовий компонент має працювати з системою координат WGS84 (EPSG:4326) для зберігання та обміну даними, оскільки це стандарт GPS-приймачів та більшості геоінформаційних систем, тоді як візуалізація на вебкартах використовує проєкцію Web Mercator (EPSG:3857) як стандарт вебкартографії. Експорт даних у форматі GeoJSON забезпечує сумісність з широким спектром ГІС-інструментів та можливість інтеграції з іншими системами.

Висновки до розділу 1

Проведений аналіз предметної області та існуючих технологічних рішень дозволив сформулювати цілісне розуміння вимог до геоінформаційної системи протимінної діяльності. Значна частина території України класифікована як потенційно небезпечна внаслідок мінного забруднення, що створює гостру потребу в ефективних інструментах координації зусиль з розмінування.

Аналіз міжнародних стандартів IMAS виявив ключові вимоги до структури даних та процесів: обов'язкове документування всіх етапів процесу розмінування, однозначне зазначення системи координат для забезпечення точності позиціонування, збереження даних у форматах, сумісних зі стандартами OGC, для забезпечення інтероперабельності з іншими геоінформаційними системами.

Аналіз існуючих систем виявив сильні сторони та обмеження кожного рішення, що дозволяє врахувати накопичений досвід при розробці власної системи. Ранні версії IMSMA (до Core) були desktop-орієнтованими та мали обмежену доступність, тоді як IMSMA Core є сучасною вебплатформою з розширеними можливостями. Українська ua.imsma.org має переваги вебдоступу з потенціалом для вдосконалення в напрямках зручності інтерфейсу та функціональності роботи з секторами. CROMAC GIS демонструє оптимальний користувацький досвід та тісну інтеграцію з кадастровими даними.

Порівняльний аналіз технологій обґрунтував вибір стеку: ASP.NET Core 8.0 для серверної частини завдяки високій продуктивності та типобезпеці, PostgreSQL з PostGIS для зберігання геопросторових даних завдяки відповідності стандартам OGC, React з TypeScript для клієнтської частини завдяки широкій екосистемі та статичній типізації, Leaflet.js для картографічної візуалізації завдяки легковісності та простоті використання.

Сформульовані функціональні вимоги охоплюють три рівні доступу: Guest для публічного інформування населення про стан безпеки територій, Operator для реєстрації процесів розмінування акредитованими організаціями, Administrator для повного контролю над системою та управління користувачами. Результати розділу формують основу для проєктування системи, орієнтованої на публічну доступність інформації про безпеку територій та зручність реєстрації процесів розмінування для операторів.

РОЗДІЛ 2. МЕТОДИ ТА МОДЕЛІ СИСТЕМИ

Другий розділ присвячено опису методів та моделей, що лежать в основі спеціалізованої багатокористувацької інформаційної системи реєстрування процесів розмінування. Розглядаються підходи до моделювання геопросторових даних, методи оптимізації роботи з просторовою інформацією, модель розмежування доступу користувачів та методи візуалізації даних на інтерактивній карті.

2.1. Моделювання геопросторових даних

Ефективне представлення геопросторової інформації є фундаментальною основою інформаційної системи реєстрування процесів розмінування. Оскільки вся інформація про стан територій нерозривно пов'язана з конкретними локаціями на місцевості, вибір моделі геопросторових даних безпосередньо впливає на можливості системи щодо зберігання, обробки та візуалізації інформації.

2.1.1. Векторна модель представлення територій

Для представлення геопросторових даних у системі обрано векторну модель, що є стандартним підходом для інформаційних систем, де потрібно працювати з дискретними об'єктами на карті. На відміну від растрової моделі, що представляє простір як регулярну сітку комірок з присвоєними значеннями, векторна модель описує географічні об'єкти через їхні геометричні характеристики – координати точок, ліній та полігонів.

У контексті системи реєстрування процесів розмінування використовуються два основних типи геометричних примітивів:

- Полігони (Polygon) – замкнені області, що визначаються послідовністю координат вершин. Полігони використовуються для представлення

секторів розмінування – ділянок території з певним статусом. Кожен полігон описує межі сектору і може мати довільну форму, що дозволяє точно відображати реальні межі територій незалежно від їхньої конфігурації.

- Точки (Point) – окремі локації, що визначаються парою координат (довгота, широта). Точки використовуються для позначення місць виявлення вибухонебезпечних предметів. Кожна точка відповідає конкретному місцю знахідки та може містити додаткову атрибутивну інформацію.

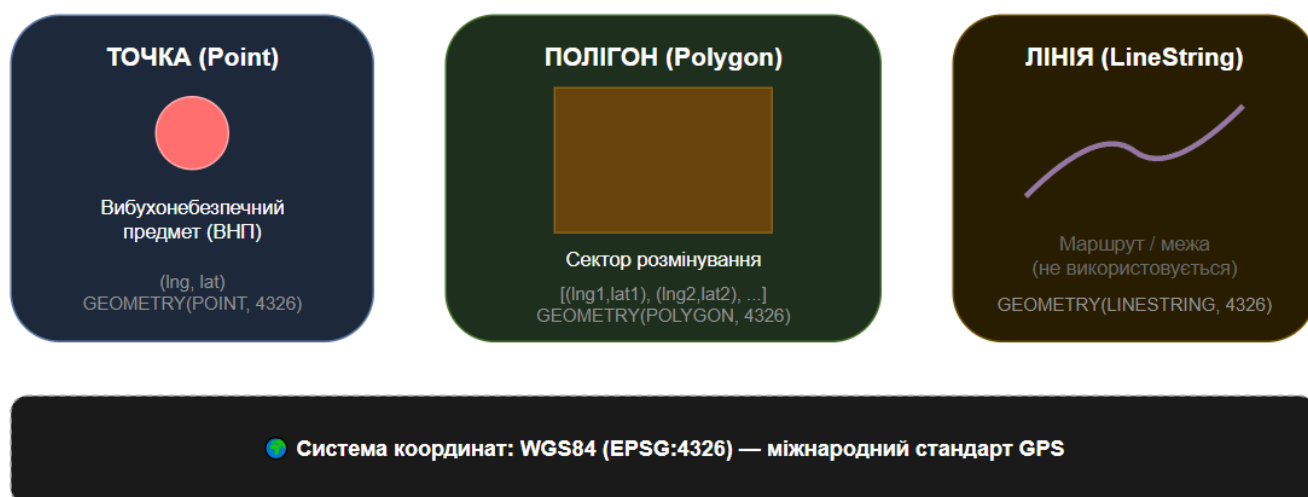


Рисунок 2.1 – Приклад векторного представлення даних: полігони секторів та точки ВНП на карті

Для зберігання геометричних даних використовується формат, сумісний зі стандартами Open Geospatial Consortium (OGC) [13]. Це забезпечує можливість обміну даними з іншими геоінформаційними системами та використання стандартних інструментів для роботи з просторовою інформацією.

Система координат WGS84 (World Geodetic System 1984) обрана як основна для зберігання всіх геопросторових даних. WGS84 є глобальним стандартом, що використовується системами GPS та більшістю картографічних сервісів, що

забезпечує сумісність з різними джерелами даних та спрощує інтеграцію з навігаційним обладнанням.

2.1.2. Модель секторів розмінування

Сектор є центральною сутністю системи, що представляє ділянку території з певним статусом щодо процесу розмінування. Модель сектору є спрощеною моделлю станів для обліку робіт, що узгоджується з загальною логікою процесу звільнення території (Land Release), визначеного міжнародними стандартами протимінної діяльності, але не є дослівним копіюванням формальних етапів Land Release.

Кожен сектор характеризується такими основними атрибутами:

- Геометрія – полігон, що визначає межі сектору. Координати вершин полігону зберігаються у системі координат WGS84. Геометрія сектору може мати довільну форму, що дозволяє точно відображати межі реальних ділянок незалежно від їхньої конфігурації.
- Статус – поточний стан сектору в процесі розмінування. Модель передбачає чотири основних статуси:

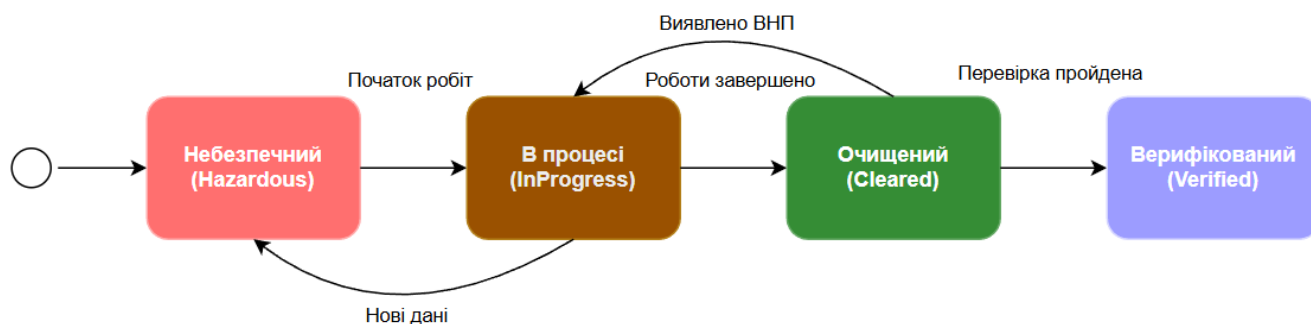


Рисунок 2.2 – Діаграма станів сектору

Небезпечний (Hazardous) – територія визначена як потенційно небезпечна та потребує обстеження або очищення. Це початковий статус для новостворених секторів.

В процесі (InProgress) – на території тривають роботи з обстеження або очищення. Перехід до цього статусу фіксує початок активних робіт на ділянці.

Очищений (Cleared) – роботи з очищення завершені командою, що виконувала розмінування. Територія очікує незалежної перевірки.

Верифікований (Verified) – безпечність території підтверджена незалежною перевіркою. Це кінцевий статус, що означає можливість безпечного використання території.

Послідовність переходів між статусами відображає реальний процес розмінування та забезпечує чіткий облік на якому етапі перебуває кожна ділянка.

Додатково сектор може містити такі атрибути:

- унікальний ідентифікатор;
- назва або код сектору;
- дата створення та останньої модифікації;
- інформація про користувача, що створив або модифікував запис;
- площа (обчислюється автоматично на основі геометрії);
- додаткові примітки.

2.1.3. Геопросторові операції

Для роботи з геопросторовими даними система використовує набір просторових операцій PostGIS [2], що забезпечують аналіз та маніпуляцію геометричними об'єктами.

Топологічні предикати – функції, що перевіряють просторові відношення між геометриями:

- `ST_Covers` – перевіряє, чи повністю покриває одна геометрія іншу, включаючи випадки коли об'єкт знаходиться на межі. Використовується, наприклад, для визначення, чи знаходиться точка ВНП всередині або на межі певного сектору. На відміну від `ST_Contains`, який повертає `false` для точок на межі полігона, `ST_Covers` коректно обробляє такі граничні випадки.
- `ST_Intersects` – перевіряє, чи перетинаються дві геометрії. Застосовується для пошуку секторів, що потрапляють у видиму область карти.
- `ST_Within` – перевіряє, чи знаходиться одна геометрія повністю всередині іншої. Використовується для фільтрації об'єктів за географічним регіоном.

Функції вимірювання:

- `ST_Area` – обчислює площу полігону. Оскільки геометрії зберігаються у WGS84 (EPSG:4326), для отримання площі в метричних одиницях використовується перетворення до типу `geography`: `ST_Area(geom::geography)` повертає площу в квадратних метрах, яка потім конвертується в гектари або квадратні кілометри. Використовується для автоматичного розрахунку площі секторів та формування статистики.
- `ST_Distance` – обчислює відстань між двома геометриями. Аналогічно до `ST_Area`, для отримання відстані в метрах використовується приведення до `geography`: `ST_Distance(a::geography, b::geography)`. Може застосовуватися для пошуку найближчих об'єктів.

Функції перетворення:

- `ST_AsGeoJSON` – конвертує геометрію у формат GeoJSON для передачі клієнтській частині системи.

- **ST_GeomFromGeoJSON** – створює геометрію з GeoJSON-представлення при отриманні даних від клієнта. Після парсингу GeoJSON SRID встановлюється як EPSG:4326 за допомогою **ST_SetSRID** для забезпечення коректної роботи просторових операцій.
- **ST_Transform** – перетворює координати між різними системами координат. Коректна робота цієї функції вимагає правильно встановленого SRID у вхідній геометрії.

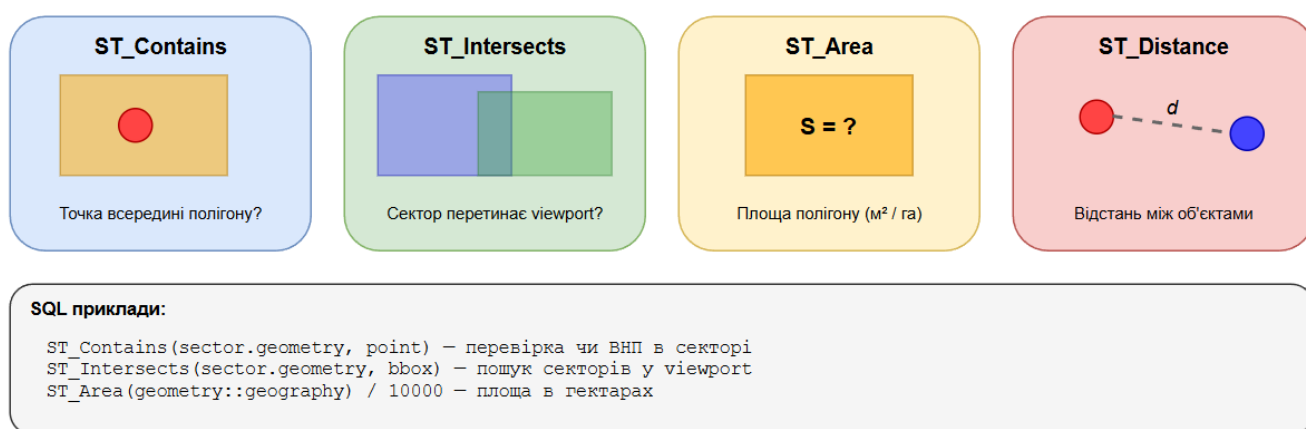


Рисунок 2.3 – Схема застосування геопросторових операцій у системі

Використання стандартних геопросторових операцій забезпечує надійність обчислень та сумісність з іншими ГІС-інструментами.

2.2. Оптимізація роботи з геопросторовими даними

Ефективна робота з геопросторовими даними вимагає застосування спеціальних методів оптимізації, що враховують особливості просторових запитів. На відміну від звичайних запитів до бази даних, просторові запити часто охоплюють діапазони координат та вимагають обчислення геометричних відношень, що є ресурсомісткими операціями.

2.2.1. Просторова індексація

Просторова індексація є ключовим механізмом забезпечення продуктивності запитів до геопросторових даних. Без індексів кожен запит вимагав би перевірки кожного об'єкта в базі даних, що є неприйнятним при великій кількості записів.

Для індексації геопросторових даних у PostGIS використовується індекс GiST (Generalized Search Tree) [2], який реалізує концепцію R-tree (Rectangle tree) [14] – збалансованого дерева, оптимізованого для індексації багатовимірних даних. GiST організовує просторові об'єкти ієрархічно, групуючи близькі об'єкти у вузли та будуючи дерево обмежуючих прямокутників (Minimum Bounding Rectangle, MBR).

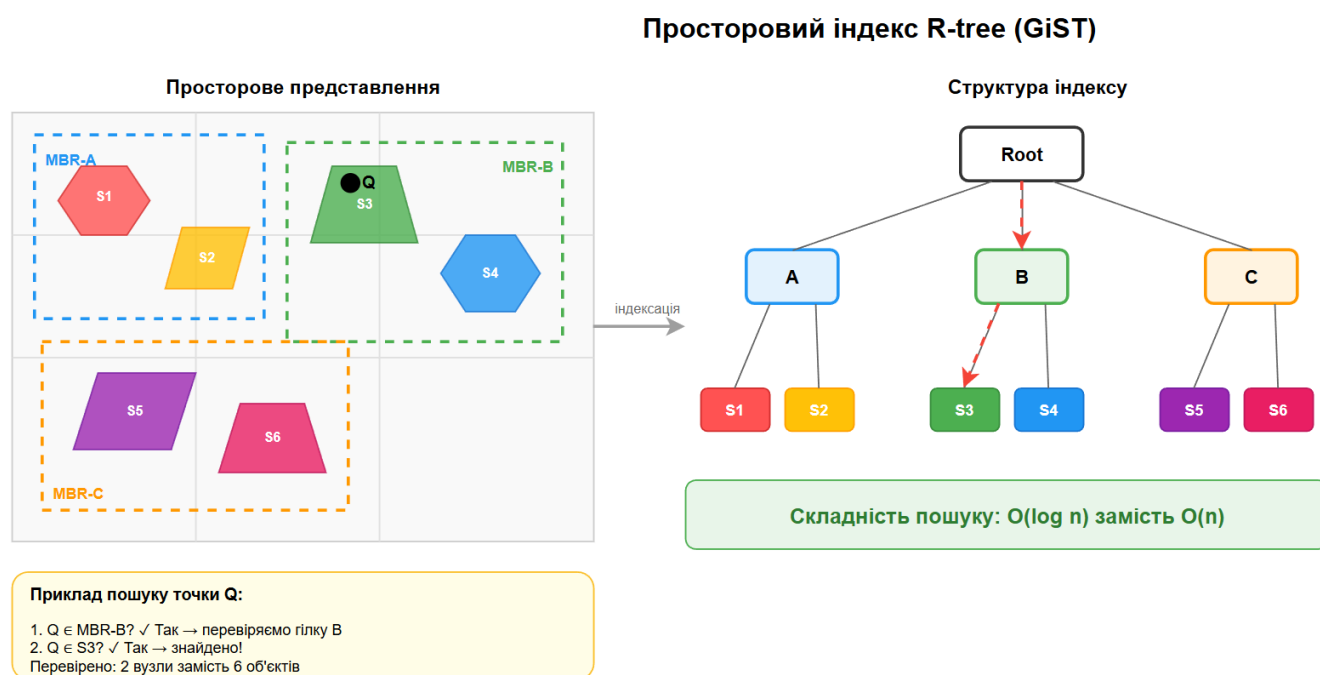


Рисунок 2.4 – Принцип роботи R-tree індексу: ієрархія обмежуючих прямокутників

Принцип роботи R-tree полягає у наступному:

1. Кожен геопросторовий об'єкт (полігон сектору або точка ВНП) охоплюється мінімальним обмежуючим прямокутником (MBR).

2. Об'єкти групуються у вузли дерева так, щоб MBR вузла охоплював усі об'єкти, що до нього належать.
3. Вузли організовуються ієрархічно – вузли нижнього рівня групуються у вузли вищого рівня.
4. При виконанні запиту відбувається обхід дерева зверху вниз з відсіканням гілок, MBR яких не перетинається з областю запиту.

Такий підхід дозволяє швидко звузити пошук до невеликої підмножини об'єктів, що потенційно відповідають запиту, без необхідності перевірки всіх записів.

Ефективність просторового індексу особливо помітна при запитах типу "знайти всі сектори у видимій області карти", де без індексу довелося б перевіряти перетин з кожним сектором у базі.

2.2.2. Тайлова система відображення карти

Для відображення картографічної основи використовується тайлова система [15], що є стандартним підходом у вебкартографії. Тайли (tiles) – це невеликі зображення фіксованого розміру (зазвичай 256×256 пікселів), що разом формують повну карту.

Тайлова піраміда (Tile Pyramid)

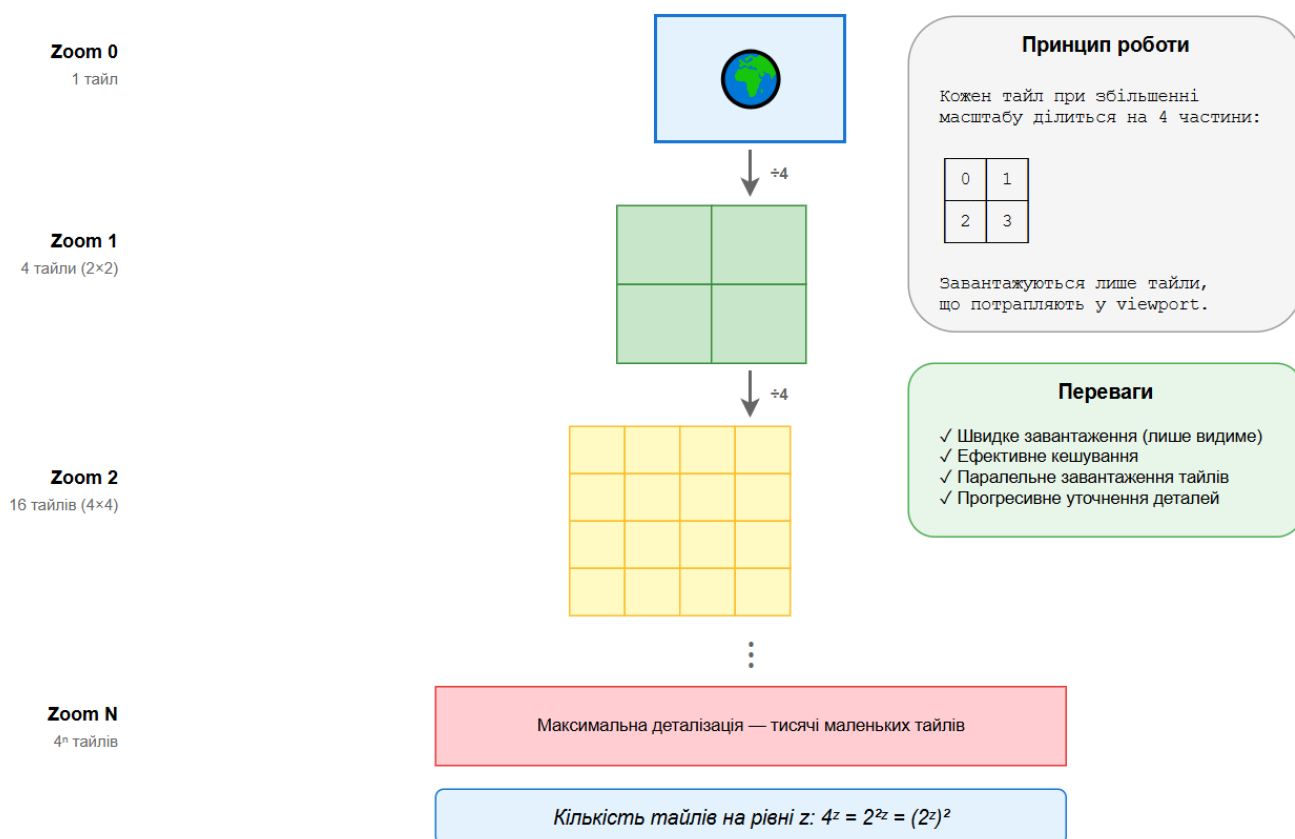


Рисунок 2.5 – Схема тайлової піраміди: рівні масштабування та розбиття на тайли.

Принцип тайлової системи:

- Карта розбивається на сітку тайлів для кожного рівня масштабування (zoom level). На мінімальному масштабі вся карта світу міститься в одному тайлі. При збільшенні масштабу кількість тайлів зростає експоненціально – на кожному наступному рівні кількість тайлів збільшується вчетверо.
- Тайли ідентифікуються трьома параметрами: рівень масштабування (z), номер колонки (x) та номер рядка (y). Це дозволяє клієнту запитувати лише ті тайли, що потрапляють у видиму область екрану.

Переваги тайлового підходу:

- завантажуються лише видимі частини карти, що зменшує обсяг переданих даних;
- тайли ефективно кешуються на стороні клієнта та проміжних серверах;
- стандартизований підхід дозволяє використовувати різні джерела картографічних даних.

Для відображення тайлів на клієнтській частині використовується бібліотека Leaflet.js [11], що забезпечує плавну навігацію картою та автоматичне завантаження необхідних тайлів при зміні положення або масштабу.

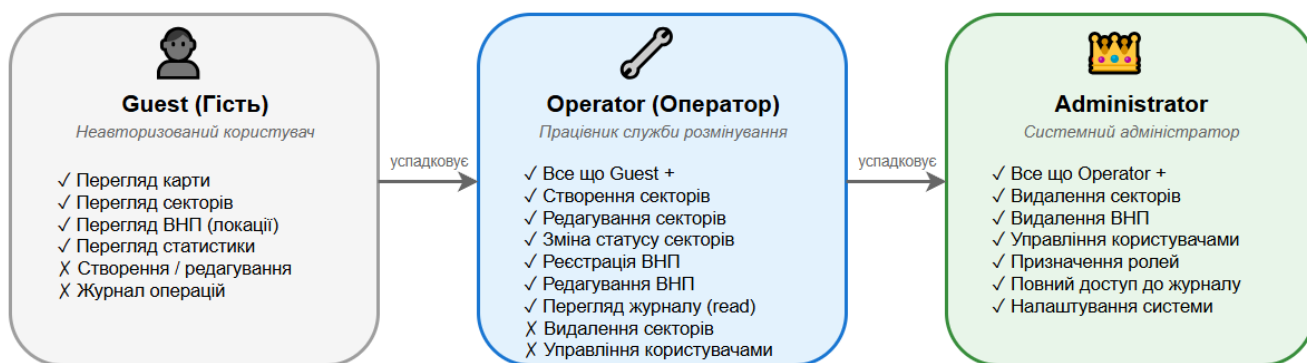
2.3. Модель багатокористувацького доступу

Система реєстрування процесів розмінування призначена для різних категорій користувачів з різними потребами та рівнями довіри. Модель багатокористувацького доступу забезпечує розмежування функціональності залежно від ролі користувача, балансуючи між відкритістю інформації для населення та захистом чутливих даних.

2.3.1. Рівні доступу користувачів

Модель доступу базується на підході RBAC (Role-Based Access Control) – управління доступом на основі ролей. Замість призначення прав окремим користувачам, права групуються у ролі, а користувачам призначаються відповідні ролі.

Модель контролю доступу RBAC



Матриця доступу:

Роль	Сектори	ВВП	Користувачі	Журнал	Статистика
Guest	R	R	—	—	R
Operator	CRU	CRU	—	R	R
Admin	CRUD	CRUD	CRUD	R	R

Позначення:

C — Create (створення)
 R — Read (перегляд)
 U — Update (редагування)
 D — Delete (видалення)
 — — немає доступу

Рисунок 2.6 – Схема рівнів доступу: Guest, Operator, Administrator

Система передбачає три основні ролі:

- **Guest (Гість)** – неавторизований користувач, що представляє широку громадськість. Роль Guest забезпечує публічний доступ до інформації про стан територій без можливості внесення змін. Основне призначення цієї ролі – надати населенню можливість дізнатися, які території є небезпечними, де тривають роботи з очищення, а де вже безпечно перебувати. Користувачі з роллю Guest можуть переглядати карту з візуалізацією секторів, бачити загальну статистику, але не мають доступу до детальної інформації про окремі записи та не можуть вносити зміни.
- **Operator (Оператор)** – авторизований користувач, що представляє організацію, яка займається розмінуванням. Це може бути працівник державної служби, гуманітарної місії або іншої акредитованої структури. Оператори мають розширений доступ порівняно з гостями: вони можуть створювати нові сектори, змінювати статуси існуючих секторів, реєструвати виявлені вибухонебезпечні предмети та переглядати детальну

інформацію. Роль Operator є основною робочою роллю для реєстрації процесів розмінування.

- Administrator (Адміністратор) – користувач з повним доступом до системи. Адміністратори мають усі можливості операторів, а також додаткові функції управління: створення та редагування облікових записів користувачів, перегляд журналу операцій, налаштування параметрів системи. Роль Administrator призначена для координаторів, що відповідають за загальне управління системою та контроль за діяльністю операторів.

2.3.2. Матриця дозволів

Для формалізації розмежування прав складено матрицю дозволів, що визначає які дії доступні для кожної ролі.

Таблиця 2.1 – Матриця дозволів за ролями

Дія	Guest	Operator	Administrator
Перегляд карти	✓	✓	✓
Перегляд загальної статистики	✓	✓	✓
Перегляд базової інформації секторів	✓	✓	✓
Перегляд детальної інформації секторів	—	✓	✓
Створення секторів	—	✓	✓
Редагування секторів	—	✓	✓

Зміна статусу секторів	—	✓	✓
Видалення секторів	—	—	✓
Реєстрація ВНП	—	✓	✓
Редагування ВНП	—	✓	✓
Видалення ВНП	—	—	✓
Перегляд журналу власних операцій	—	✓	✓
Управління користувачами	—	—	✓
Налаштування системи	—	—	✓

Така модель забезпечує:

- відкритість базової інформації для населення (роль Guest);
- функціональність для реєстрації процесів розмінування (роль Operator);
- контроль та управління системою (роль Administrator).

Авторизація користувачів здійснюється з використанням токенів JWT (JSON Web Token) [17], що забезпечує безстанову перевірку ідентичності та масштабованість системи.

2.4. Методи візуалізації даних на карті

Візуалізація є ключовим аспектом системи, оскільки саме через карту користувачі отримують інформацію про стан територій. Методи візуалізації

розроблені з урахуванням потреб різних категорій користувачів та принципів ефективного представлення геопросторової інформації.

2.4.1. Кольорове кодування статусів секторів

Для швидкого візуального розрізнення секторів з різними статусами використовується система кольорового кодування. Кожному статусу відповідає окремий колір, що дозволяє користувачу миттєво оцінити стан території без необхідності читати текстові підписи.

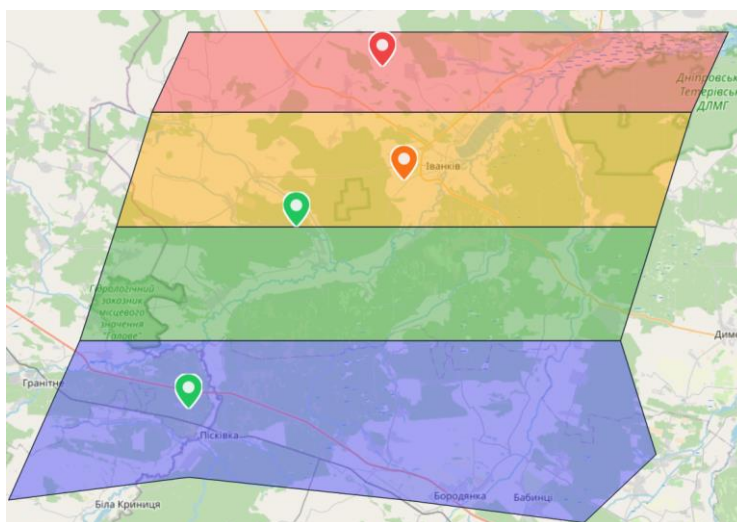


Рисунок 2.7 – Кольорове кодування статусів секторів на карті.

Обрана кольорова схема:

Статус	Колір	Обґрунтування
Небезпечний	Червоний	Асоціюється з небезпекою, привертає увагу
В процесі	Жовтий	Проміжний стан, сигналізує про активні роботи
Очищений	Зелений	Асоціюється з безпекою
Верифікований	Синій	Відрізняється від "Очищений", підкреслює підтвердження

Кольорова схема базується на загальноприйнятих асоціаціях кольорів (червоний – небезпека, зелений – безпека) та забезпечує достатній контраст для розрізнення різних статусів. Для забезпечення доступності відповідно до WCAG 2.1 [16], статус сектору дублюється не лише кольором, а й типом контуру (суцільний, штриховий, пунктирний) та іконкою в спливаючій підказці, що дозволяє користувачам з порушеннями сприйняття кольорів коректно ідентифікувати статус.

Полігони секторів відображаються з напівпрозорим заливанням, що дозволяє бачити картографічну основу під ними, та виразним контуром для чіткого визначення меж.

2.4.2. Кластеризація маркерів

При великій кількості зареєстрованих вибухонебезпечних предметів відображення кожного окремим маркером може призвести до перевантаження карти – маркери накладаються один на одного, що ускладнює сприйняття інформації та уповільнює роботу інтерфейсу.

Для вирішення цієї проблеми застосовується метод кластеризації маркерів [18]. Кластеризація автоматично групує близько розташовані маркери у кластери при віддаленому масштабі карти.

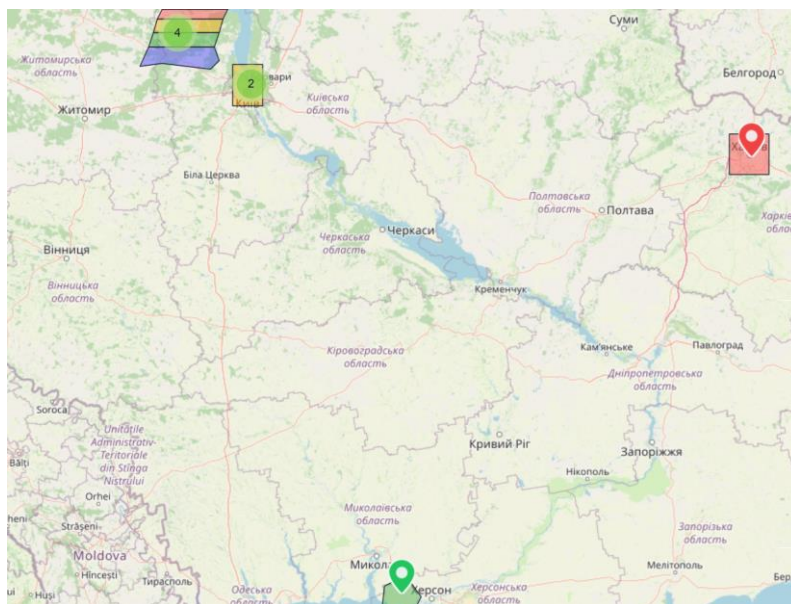


Рисунок 2.8 – Приклад кластеризації маркерів

Принцип роботи:

1. При віддаленому масштабі маркери, що знаходяться близько один до одного на екрані, об'єднуються у кластер.
2. Кластер відображається як один маркер з числом, що показує кількість об'єктів у групі.
3. При наближенні карти кластери поступово розпадаються на менші групи або окремі маркери.
4. При достатньому наближенні кожен об'єкт відображається окремим маркером.

Переваги кластеризації:

- карта залишається читабельною навіть при великій кількості об'єктів;
- забезпечується краща продуктивність інтерфейсу;
- кластери дають швидке уявлення про густоту розташування ВВП у різних районах.

2.4.3. Статистичні показники

Для надання узагальненої інформації про стан процесів розмінування система відображає статистичні показники. Статистика дозволяє швидко оцінити загальну картину без необхідності детального вивчення карти.

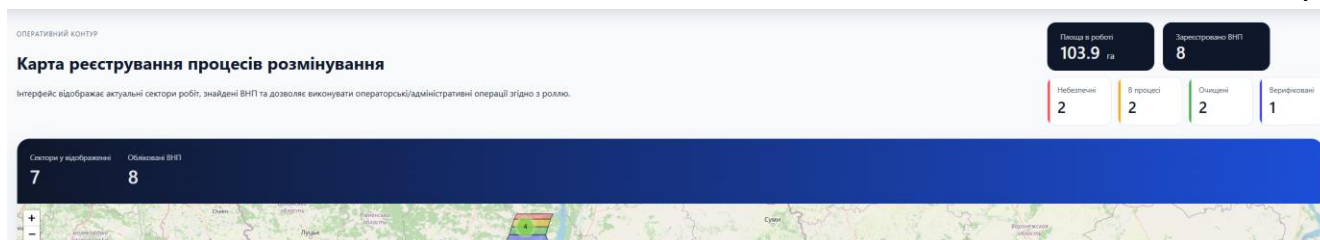


Рисунок 2.9 – Базова панель статистичних показників, загальнодоступна

Основні статистичні показники:

- Кількість секторів за статусами – показує скільки секторів має кожен з чотирьох статусів. Дозволяє оцінити загальний прогрес робіт.
- Площа територій за статусами – показує сумарну площу секторів у кожному статусі (у квадратних кілометрах або гектарах). Більш інформативний показник, ніж просто кількість секторів.
- Загальна кількість зареєстрованих ВВП – показує скільки вибухонебезпечних предметів виявлено та зареєстровано в системі.

Статистика може відображатися у вигляді числових значень, діаграм або комбінованих віджетів залежно від контексту використання.

Для ролі Guest статистика відображається в агрегованому вигляді без деталізації по окремих секторах. Для ролей Operator та Administrator доступна більш детальна статистика з можливістю фільтрації за періодом, регіоном або оператором.

Висновки до розділу 2

У другому розділі описано методи та моделі, що лежать в основі спеціалізованої багатокористувацької інформаційної системи реєстрування процесів розмінування.

Для моделювання геопросторових даних обрано векторну модель з використанням полігонів для представлення секторів розмінування та точок для

позначення місць виявлення вибухонебезпечних предметів. Система координат WGS84 забезпечує сумісність з GPS-обладнанням та картографічними сервісами.

Розроблено модель секторів розмінування з чотирма статусами (Небезпечний, В процесі, Очищений, Верифікований), що узгоджується з логікою процесу звільнення території. Послідовність переходів між статусами забезпечує чіткий облік прогресу робіт.

Для оптимізації роботи з геопросторовими даними описано застосування просторової індексації на основі GiST (що реалізує концепцію R-tree) та тайлової системи відображення карти, що забезпечують продуктивність системи при роботі з великими обсягами даних.

Модель багатокористувацького доступу на основі RBAC передбачає три ролі: Guest для публічного доступу населення до інформації, Operator для реєстрації процесів розмінування, Administrator для управління системою. Матриця дозволів формалізує розмежування функціональності між ролями.

Методи візуалізації включають кольорове кодування статусів секторів для швидкого візуального розрізнення, кластеризацію маркерів для ефективного відображення великої кількості об'єктів та статистичні показники для узагальненої оцінки прогресу робіт.

Описані методи та моделі формують основу для проєктування та реалізації системи, що буде представлено в наступному розділі.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

Третій розділ присвячено детальному опису архітектурних рішень та практичної реалізації спеціалізованої багатокористувацької інформаційної системи реєстрування процесів розмінування. Розглянуто архітектуру системи на основі принципів Clean Architecture, проєктування геопросторової бази даних з використанням PostgreSQL та PostGIS, реалізацію серверної частини на платформі ASP.NET Core та клієнтського інтерфейсу на базі React з картографічною бібліотекою Leaflet.js.

3.1. Архітектура системи

Архітектура системи є фундаментальним аспектом, що визначає її масштабованість, підтримуваність та можливість розвитку. У цьому підрозділі розглянуто застосування принципів Clean Architecture для організації програмного коду та обґрунтовано вибір технологічного стеку.

3.1.1. Clean Architecture та компоненти високого рівня

Для побудови системи обрано архітектурний підхід Clean Architecture, запропонований Робертом Мартіном [19]. Цей підхід забезпечує чіткий розподіл відповідальності між компонентами системи та мінімізує залежності від конкретних технологій та фреймворків.

Ключовим принципом Clean Architecture є правило залежностей: залежності в коді спрямовані від зовнішніх шарів до внутрішніх. Внутрішні шари не знають про існування зовнішніх, що забезпечує незалежність бізнес-логіки від деталей реалізації інфраструктури.

Архітектура системи складається з чотирьох концентричних шарів (рис. 3.1):

1. Domain (Доменний шар) – центральний шар, що містить бізнес-сутності та бізнес-правила. У контексті системи реєстрування процесів розмінування цей шар включає:
 - Sector – сутність сектору розмінування з геометрією полігону, статусом та метаданими;
 - ExplosiveItem – сутність вибухонебезпечного предмета з точковою геометрією та характеристиками;
 - User – сутність користувача з роллю та організаційною приналежністю;
 - Value Objects – незмінні об'єкти значень (SectorStatus, UserRole).
2. Application (Шар застосунку) – містить бізнес-логіку та сценарії використання системи. Ключові сервіси:
 - ISectorService – операції з секторами (створення, оновлення статусу, геопросторовий пошук);
 - IExplosiveItemService – реєстрація та управління записами про ВНП;
 - IDashboardService – агрегація статистичних даних для аналітичної панелі;
 - IAuditService – логування операцій користувачів.
3. Infrastructure (Інфраструктурний шар) – реалізація технічних деталей та зовнішніх залежностей:
 - Репозиторії на базі Entity Framework Core з підтримкою NetTopologySuite для геопросторових типів;
 - Інтеграція з PostgreSQL та PostGIS для зберігання та обробки геоданих;
 - Реалізація JWT-автентифікації та авторизації на основі ролей;
 - Сервіси для роботи з файлами та зовнішніми API.

4. Presentation (Шар представлення) – точка входу та взаємодії з користувачами:

- ASP.NET Core Web API контролери для обробки HTTP-запитів;
- Middleware для обробки помилок, логування та авторизації;
- Конфігурація CORS для взаємодії з клієнтським застосунком.

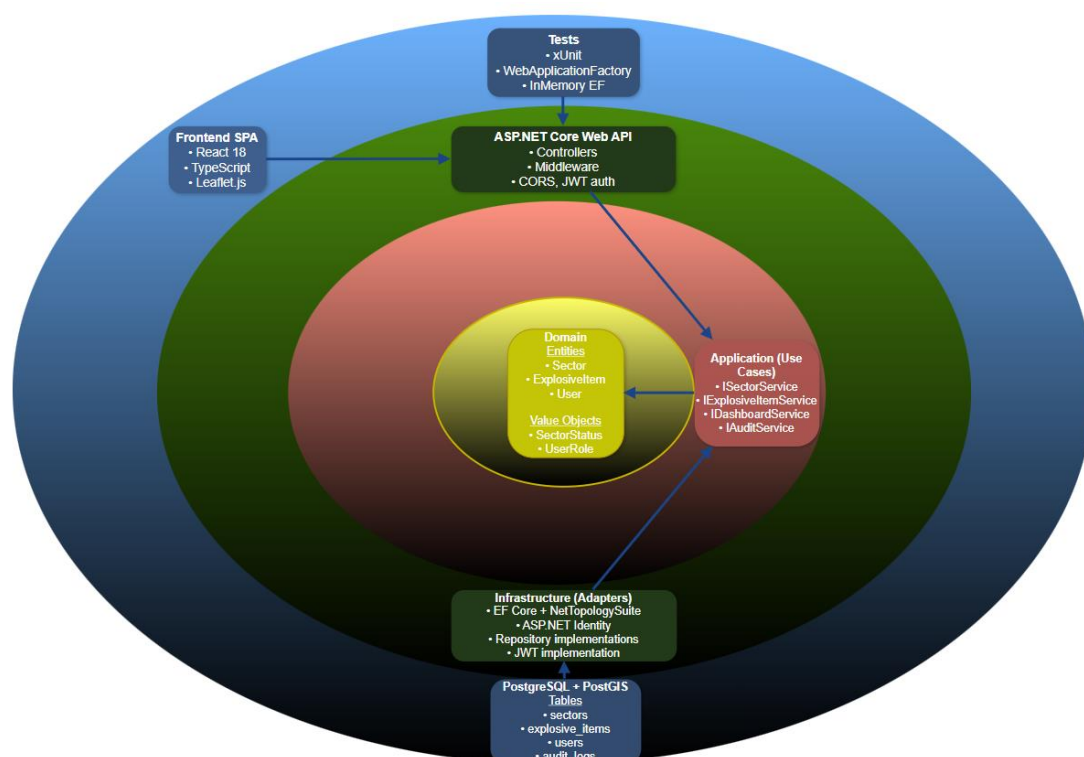


Рисунок 3.1 – Діаграма архітектури системи за принципами Clean Architecture

Переваги обраного архітектурного підходу:

- Тестованість – бізнес-логіка в доменному шарі та шарі застосунку не залежить від інфраструктури, що дозволяє писати модульні тести без необхідності підключення до бази даних чи зовнішніх сервісів.
- Незалежність від фреймворків – заміна Entity Framework Core на інший ORM або ASP.NET Core на альтернативний вебфреймворк зводить зміни у доменному шарі та бізнес-логіці до мінімуму за умови дотримання принципів Clean Architecture.

- Незалежність від бази даних – доменні сутності не прив'язані до конкретної СУБД. Теоретично можлива міграція з PostgreSQL на іншу геопросторову базу даних, хоча на практиці це потребуватиме адаптації інфраструктурного шару та перевірки сумісності специфічних функцій.
- Чіткі межі відповідальності – кожен шар має визначену роль, що спрощує розуміння системи та її підтримку командою розробників.

3.1.2. Технологічний стек та обґрунтування вибору

Вибір технологічного стеку ґрунтувався на вимогах до системи, визначених у розділі 1.4, та результатах порівняльного аналізу технологій. Обраний стек забезпечує баланс між продуктивністю, надійністю та зручністю розробки.

Таблиця 3.1 – Компоненти технологічного стеку системи

Компонент	Технологія	Версія	Призначення
Серверна платформа	.NET	8.0 LTS	Середовище виконання серверного коду
Вебфреймворк	ASP.NET Core	8.0	Побудова REST API
ORM	Entity Framework Core	8.0	Об'єктно-реляційне відображення
Геопросторова бібліотека	NetTopologySuite	2.5	Робота з геометриями в .NET
Автентифікація	ASP.NET Identity	8.0	Управління користувачами
Токени доступу	JWT Bearer	—	Автентифікація API-запитів
СУБД	PostgreSQL	16	Реляційне сховище даних

Геопросторове розширення	PostGIS	3.4	Геопросторові операції в БД
Клієнтський фреймворк	React	18	Побудова інтерфейсу користувача
Мова програмування (клієнт)	TypeScript	5.0	Типізований JavaScript
Картографічна бібліотека	Leaflet.js	1.9	Інтерактивні карти
Кластеризація маркерів	Leaflet.markercluster	1.5	Групування маркерів на карті

Обґрунтування вибору серверної платформи .NET 8:

- Продуктивність – .NET 8 демонструє високу продуктивність у бенчмарках TechEmpower, поступаючись лише низькорівневим рішенням на C++ та Rust. Для вебдодатків з інтенсивним використанням бази даних .NET забезпечує оптимальний баланс швидкості та зручності розробки.
- Екосистема – наявність зрілих бібліотек для роботи з геопросторовими даними (NetTopologySuite), вбудована підтримка автентифікації (ASP.NET Identity), потужний ORM (Entity Framework Core).
- Довгострокова підтримка – версія .NET 8 має статус LTS (Long Term Support) з гарантованою підтримкою до листопада 2026 року.
- Крос-платформність – можливість розгортання на Windows, Linux та macOS без змін у коді.

Обґрунтування вибору PostgreSQL + PostGIS:

- Стандарт галузі – PostGIS є де-факто стандартом для геопросторових застосунків з відкритим кодом. Використовується в QGIS, GeoServer та багатьох інших ГІС-рішеннях.

- Відповідність стандартам – повна підтримка стандартів OGC Simple Features [13] та SQL/MM Spatial.
- Продуктивність – ефективна реалізація просторових індексів на основі R-tree [14] (GiST), що забезпечує швидкий пошук у великих наборах геоданих.
- Інтеграція з .NET – бібліотека Npgsql.EntityFrameworkCore.PostgreSQL з розширенням NetTopologySuite забезпечує прозору інтеграцію геопросторових типів PostGIS з Entity Framework Core.

Обґрунтування вибору React та Leaflet.js:

- React – декларативний підхід до побудови інтерфейсу спрощує створення інтерактивних компонентів. Велика екосистема та активна спільнота забезпечують доступність готових рішень.
- TypeScript – статична типізація зменшує кількість помилок та покращує підтримуваність коду, особливо при роботі зі складними структурами даних (GeoJSON).
- Leaflet.js – легковісна бібліотека (~42 КБ gzip) з простим API, достатнім для потреб проєкту. На відміну від OpenLayers, не перевантажена функціоналом. На відміну від Mapbox GL JS, не потребує комерційної ліцензії для продакшн-використання.

3.2. Проєктування бази даних

База даних є критичним компонентом системи, що забезпечує надійне зберігання геопросторових даних та ефективне виконання просторових запитів. У цьому підрозділі представлено концептуальну модель даних, логічну схему з геопросторовими типами та SQL-скрипти створення структури бази даних.

3.2.1. Концептуальна модель даних

Концептуальна модель визначає основні сутності системи та зв'язки між ними. На основі аналізу предметної області та функціональних вимог виділено чотири ключові сутності (рис. 3.2):

- **Sector** (Сектор розмінування) – центральна сутність системи, що представляє територію з визначеними межами та статусом безпеки. Кожен сектор характеризується унікальним ідентифікатором, назвою, описом, поточним статусом, геометрією полігону та обчисленою площею. Також фіксується інформація про створення та останнє оновлення запису.
- **ExplosiveItem** (Вибухонебезпечний предмет) – сутність для реєстрації виявлених ВНП. Містить інформацію про тип предмета (категорія, тип, модель), точкові координати виявлення, статус (виявлено, нейтралізовано), дати виявлення та нейтралізації, а також посилання на фотодокументацію.
- **User** (Користувач) – сутність для управління обліковими записами та розмежування доступу. Включає ім'я користувача, електронну пошту, хеш пароля, роль у системі (Operator, Administrator), організаційну приналежність та службову інформацію. Публічний доступ (роль Guest на концептуальному рівні) реалізується без створення запису в таблиці users – анонімні користувачі працюють без JWT-токена.
- **AuditLog** (Журнал операцій) – сутність для забезпечення відстежуваності змін у системі. Кожен запис фіксує тип дії, сутність та її ідентифікатор, попередні та нові значення у форматі JSON, IP-адресу та час операції.

Зв'язки між сутностями:

- **Sector – ExplosiveItem** (один-до-багатьох): один сектор може містити множину зареєстрованих ВНП. Зв'язок реалізовано через зовнішній ключ

sector_id у таблиці explosive_items. При видаленні сектору зв'язок встановлюється у NULL, зберігаючи історичні дані про ВНП.

- User – Sector (один-до-багатьох): користувач може створювати та оновлювати множину секторів. Зв'язки created_by та updated_by забезпечують відстежуваність авторства змін.
- User – ExplosiveItem (один-до-багатьох): користувач реєструє виявлення ВНП (found_by) та може фіксувати їх нейтралізацію (neutralized_by).
- User – AuditLog (один-до-багатьох): кожна операція у журналі прив'язана до користувача, що її виконав.

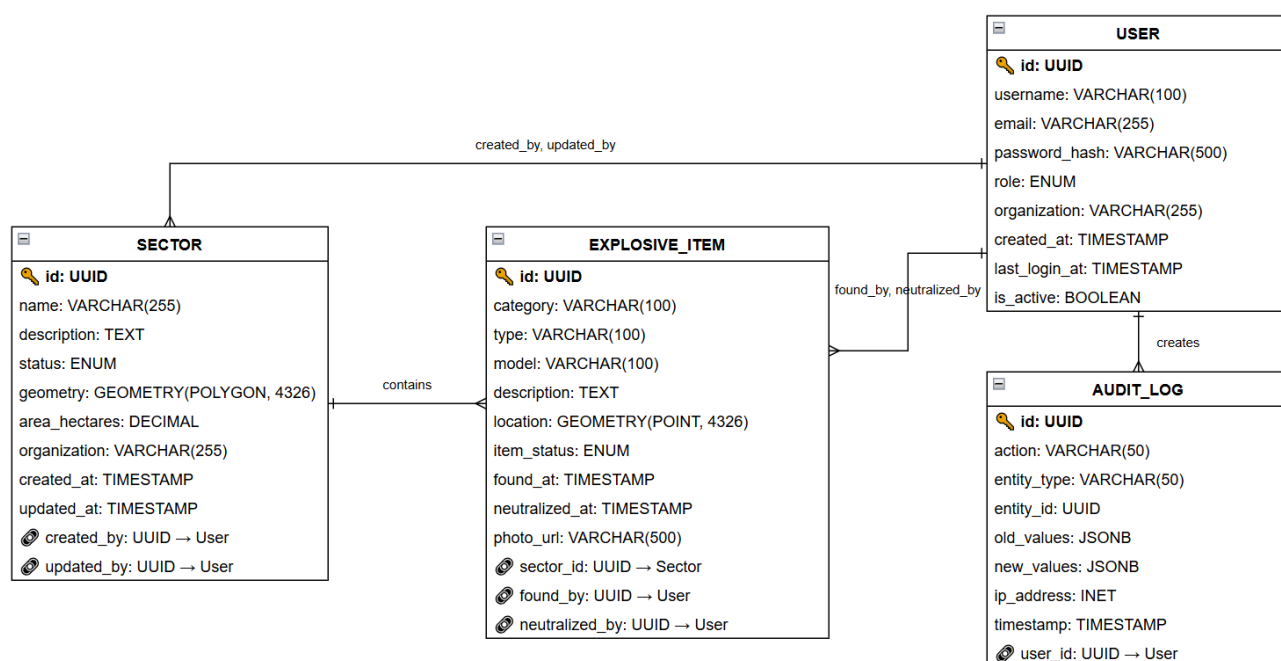


Рисунок 3.2 – ER-діаграма бази даних системи

3.2.2. Логічна модель з геопросторовими типами

Логічна модель деталізує структуру таблиць з урахуванням специфіки геопросторових даних та можливостей PostgreSQL з розширенням PostGIS. У таблиці 3.2 наведено спеціалізовані типи даних, що використовуються в системі.

Таблиця 3.2 – Спеціалізовані типи даних PostgreSQL/PostGIS

Тип даних	Таблиця	Поле	Призначення
GEOMETRY(POLYGON, 4326)	sectors	geometry	Межі сектору (полігон у WGS84)
GEOMETRY(POINT, 4326)	explosive_items	location	Координати виявлення ВНП
GENERATED (обчислюване)	sectors	area_hectares	Площа сектору в гектарах
TIMESTAMP WITH TIME ZONE	всі таблиці	created_at, updated_at	Часові позначки в UTC
JSONB	audit_logs	old_values, new_values	Історія змін у JSON-форматі
TEXT[]	explosive_items	photo_urls	Масив посилань на фото

Числовий ідентифікатор 4326 (SRID – Spatial Reference System Identifier) у типах GEOMETRY вказує на систему координат WGS84, що є стандартом для GPS-обладнання та вебкартографії.

PostgreSQL підтримує обчислювані (generated) колонки, що автоматично розраховуються на основі інших полів. У таблиці sectors використовується обчислювана площа за формулою:

$$\text{area_hectares} = \text{ST_Area(geometry::geography)} / 10000$$

Функція ST_Area з приведенням до типу geography повертає площу в квадратних метрах з урахуванням кривизни Землі. Ділення на 10000 конвертує результат у гектари.

Для забезпечення цілісності даних використовуються обмеження CHECK з переліком допустимих значень:

- sectors.status: 'Hazardous', 'InProgress', 'Cleared', 'Verified'
- users.role: 'Operator', 'Administrator'
- explosive_items.status: 'Found', 'Neutralized'

Такий підхід не потребує окремих довідникових таблиць та гарантує, що в базу потраплять лише валідні значення.

Усі часові поля використовують тип `TIMESTAMP WITH TIME ZONE`, що зберігає час у UTC. Значення за замовчуванням `NOW()` автоматично встановлює поточний час при створенні запису. Оновлення поля `updated_at` виконується в прикладному коді: `Entity Framework Core interceptor` автоматично встановлює поточний час перед збереженням зміненої сутності.

Поля `old_values` та `new_values` у таблиці `audit_logs` використовують тип `JSONB` – бінарний формат JSON з підтримкою індексації, що дозволяє зберігати довільну структуру даних для різних типів сутностей.

3.2.3. Фізична модель та індексація

На основі логічної моделі розроблено DDL-скрипти для створення таблиць бази даних. Для генерації UUID-ідентифікаторів використовується функція `gen_random_uuid()`, що потребує розширення `pgcrypto` (`CREATE EXTENSION IF NOT EXISTS pgcrypto`). Нижче наведено структуру центральної таблиці `sectors`, що демонструє використання геопросторових типів `PostGIS`:

```
CREATE TABLE sectors (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  name VARCHAR(255) NOT NULL,
  description TEXT,
  status VARCHAR(50) NOT NULL DEFAULT 'Hazardous'
  CHECK (status IN ('Hazardous', 'InProgress', 'Cleared', 'Verified')),
```

```

geometry GEOMETRY(POLYGON, 4326) NOT NULL,
area_hectares DECIMAL(10,2)
    GENERATED ALWAYS AS (ST_Area(geometry::geography) / 10000) STORED,
organization VARCHAR(255),
created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
created_by UUID NOT NULL REFERENCES users(id),
updated_by UUID REFERENCES users(id)
);
CREATE INDEX idx_sectors_geometry ON sectors USING GIST (geometry);

```

Структура DDL демонструє ключові аспекти геопросторової моделі: тип `GEOMETRY(POLYGON, 4326)` для зберігання полігональних меж секторів, обчислювана колонка `area_hectares` для автоматичного розрахунку площі та GiST-індекс для прискорення просторових запитів.

Аналогічно, таблиця `explosive_items` використовує точкову геометрію для зберігання координат виявлення ВНП:

```

CREATE TABLE explosive_items (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    sector_id UUID REFERENCES sectors(id) ON DELETE SET NULL,
    category VARCHAR(100) NOT NULL,
    type VARCHAR(100) NOT NULL,
    model VARCHAR(100),
    location GEOMETRY(POINT, 4326) NOT NULL,
    status VARCHAR(50) NOT NULL DEFAULT 'Found'
    CHECK (status IN ('Found', 'Neutralized')),
    found_at TIMESTAMP WITH TIME ZONE NOT NULL,

```

```

found_by UUID NOT NULL REFERENCES users(id),
neutralized_at TIMESTAMP WITH TIME ZONE,
neutralized_by UUID REFERENCES users(id),
photo_urls TEXT[],
notes TEXT
);
CREATE INDEX idx_items_location ON explosive_items USING GIST (location);
CREATE INDEX idx_items_sector ON explosive_items (sector_id);
CREATE INDEX idx_items_status ON explosive_items (status);

```

Тип `GEOMETRY(POINT, 4326)` зберігає точкові координати у системі WGS84. Масив `photo_urls` типу `TEXT[]` дозволяє зберігати посилання на кілька фотографій без окремої таблиці. Індекс на `sector_id` прискорює вибірку ВНП, що належать конкретному сектору.

Первинні ключі типу `UUID` генеруються функцією `gen_random_uuid()`, що забезпечує унікальність ідентифікаторів без централізованого лічильника та спрощує реплікацію даних.

Геопросторові колонки використовують типи `GEOMETRY(POLYGON, 4326)` для секторів та `GEOMETRY(POINT, 4326)` для ВНП з обов'язковим зазначенням `SRID` для забезпечення коректних просторових операцій.

Обчислювана колонка `area_hectares` автоматично розраховує площу сектору в гектарах за формулою `ST_Area(geometry::geography) / 10000`, що звільняє прикладний код від цих обчислень.

Зовнішні ключі з політикою `ON DELETE SET NULL` зберігають історичні дані про ВНП навіть при видаленні пов'язаного сектору.

Система використовує три типи індексів для оптимізації запитів:

1. GiST-індекси на геометричних колонках – забезпечують у середньому випадку складність просторових запитів, наближену до $O(\log n)$, замість

лінійного сканування $O(n)$. Фактична ефективність залежить від розподілу геометрій та ступеня перекриття MBR. Особливо критичні для операцій пошуку секторів у viewport та перевірки належності точки до полігону.

2. B-tree індекси на часто використовуваних полях – прискорюють фільтрацію за статусом, пошук за email, сортування за часом.
3. Складені індекси – оптимізують запити з кількома умовами у WHERE, наприклад пошук історії змін конкретного сектору.

3.3. Реалізація серверної частини

Серверна частина системи реалізована як RESTful Web API на платформі ASP.NET Core. У цьому підрозділі розглянуто структуру проєкту, специфікацію API-ендпоінтів, реалізацію геопросторових запитів та механізм автентифікації.

3.3.1. Структура проєкту та REST API

Структура серверного рішення відповідає принципам Clean Architecture та організована у вигляді чотирьох .NET-проєктів:

DeminingSystem.sln

```

├── src/
│   ├── DeminingSystem.Domain/      – доменні сутності та інтерфейси
│   ├── DeminingSystem.Application/  – бізнес-логіка та сценарії використання
│   ├── DeminingSystem.Infrastructure/ – реалізація репозиторіїв та сервісів
│   └── DeminingSystem.WebApi/       – контролери та конфігурація API
└── tests/
    ├── DeminingSystem.UnitTests/    – модульні тести
    └── DeminingSystem.IntegrationTests/ – інтеграційні тести
  
```

Проект DeminingSystem.Domain містить РОСО-класи сутностей без залежностей від Entity Framework чи інших інфраструктурних бібліотек. Це забезпечує чистоту доменної моделі.

Проект DeminingSystem.Application визначає інтерфейси сервісів та DTO (Data Transfer Objects) для обміну даними між шарами. Залежить лише від Domain.

Проект DeminingSystem.Infrastructure реалізує інтерфейси з Application, містить DbContext для Entity Framework Core та конфігурацію підключення до PostgreSQL.

Проект DeminingSystem.WebApi є точкою входу застосунку, містить контролери, middleware та конфігурацію Dependency Injection.

API побудовано за принципами REST з використанням стандартних HTTP-методів та кодів відповідей. Усі ендпоінти використовують формат JSON для обміну даними.

Таблиця 3.3 – Специфікація REST API системи

Метод	Ендпоінт	Опис	Авторизація
GET	/api/sectors	Отримати сектори у viewport (базова інформація)	Public
GET	/api/sectors/{id}	Отримати деталі сектору за ID	Operator+
POST	/api/sectors	Створити новий сектор	Operator+
PUT	/api/sectors/{id}	Оновити сектор	Operator+
PUT	/api/sectors/{id}/status	Змінити статус сектору	Operator+
DELETE	/api/sectors/{id}	Видалити сектор	Administrator
GET	/api/explosive-items	Отримати ВВП у viewport (базова інформація)	Public
GET	/api/explosive-items/{id}	Отримати деталі ВВП за ID	Operator+

POST	/api/explosive-items	Зареєструвати новий ВНП	Operator+
PUT	/api/explosive-items/{id}	Оновити інформацію про ВНП	Operator+
DELETE	/api/explosive-items/{id}	Видалити запис про ВНП	Administrator
GET	/api/dashboard/stats	Отримати статистику для панелі	Public
GET	/api/audit-logs	Отримати журнал операцій (пагінація)	Operator+
POST	/api/auth/login	Автентифікація користувача	Public
POST	/api/auth/refresh	Оновлення токена доступу	Public
GET	/api/users/me	Отримати профіль поточного користувача	Авторизований
GET	/api/users	Отримати список користувачів	Administrator
POST	/api/users	Створити нового користувача	Administrator
PUT	/api/users/{id}	Оновити користувача	Administrator

Позначення в колонці «Авторизація»: Public – ендпоінт доступний без автентифікації (анонімний доступ); Авторизований – будь-який користувач з валідним JWT-токеном; Operator+ – користувач з роллю Operator або Administrator; Administrator – тільки користувач з роллю Administrator.

Публічні ендпоінти (/api/sectors, /api/explosive-items) повертають базову інформацію для відображення на карті: ідентифікатор, статус, геометрію. Детальна інформація (опис, організація, примітки, історія змін) доступна тільки авторизованим користувачам через ендпоінти з параметром {id}. Координати ВНП

публічно доступні для забезпечення інформування населення про небезпечні знахідки; на клієнтській стороні вони відображаються у кластеризованому вигляді.

Контролери реалізовано за стандартним шаблоном ASP.NET Core Web API з використанням Dependency Injection для отримання залежностей та атрибутів [Authorize] для розмежування доступу за ролями. Кожна операція модифікації даних супроводжується записом до журналу аудиту.

Усі ендпоінти повертають дані у форматі JSON. Геопросторові об'єкти серіалізуються у формат GeoJSON для сумісності з картографічними бібліотеками. Нижче наведено приклад відповіді для запиту GET /api/sectors/{id}:

```
{
  "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "name": "Сектор Харківська обл. - 127",
  "description": "Східна ділянка поблизу с. Липці",
  "status": "InProgress",
  "areaHectares": 45.72,
  "geoJson": {
    "type": "Polygon",
    "coordinates": [[[36.123, 50.456], [36.234, 50.456],
                     [36.234, 50.567], [36.123, 50.567],
                     [36.123, 50.456]]]
  },
  "organization": "ДСНС Харківської області",
  "createdAt": "2025-03-15T09:30:00Z",
  "updatedAt": "2025-11-28T14:22:00Z",
  "explosiveItemsCount": 12
}
```

Геометрія полігону передається у полі `geoJson` як об'єкт `GeoJSON Polygon`. Клієнтський застосунок може безпосередньо передавати цей об'єкт у бібліотеку `Leaflet.js` для відображення на карті.

API використовує стандартні HTTP-коди відповідей: 200 OK для успішних запитів, 201 Created для створення ресурсів, 400 Bad Request для помилок валідації, 401 Unauthorized для неавторизованих запитів, 403 Forbidden для відмови в доступі, 404 Not Found для неіснуючих ресурсів. Повідомлення про помилки повертаються у структурованому форматі з полями `code` та `message`.

3.3.2. Геопросторові запити з `NetTopologySuite`

Бібліотека `NetTopologySuite` забезпечує підтримку геопросторових типів у `.NET` та інтегрується з `Entity Framework Core` через пакет `Npgsql.EntityFrameworkCore.PostgreSQL.NetTopologySuite`. Це дозволяє використовувати геопросторові операції безпосередньо в LINQ-запитах, які транслюються у відповідні функції `PostGIS`.

Для активації підтримки геопросторових типів необхідно налаштувати `DbContext` та підключення до бази даних. У конфігурації підключення вказується `UseNetTopologySuite()`, що реєструє обробники геопросторових типів. У моделі сутності `Sector` властивість `Geometry` має тип `NetTopologySuite.Geometries.Polygon`, що автоматично мапиться на тип `GEOMETRY PostgreSQL`.

`Entity Framework Core` коректно обробляє `SRID` (`Spatial Reference System Identifier`) – при створенні геометрії вказується `SRID 4326`, що відповідає системі координат `WGS84`. Це забезпечує узгодженість з `GPS`-координатами та вебкартографічними сервісами.

Пошук секторів у `viewport` – отримання секторів, що потрапляють у видиму область карти. Клієнт передає координати обмежувального прямокутника (`bounding box`), сервер формує геометрію `Envelope` та виконує запит з умовою `Intersects()`, що транслюється у функцію `ST_Intersects()` `PostgreSQL`.

Визначення сектору за координатами – при реєстрації ВНП система автоматично визначає належність точки до сектору за допомогою операції `Covers()`, що відповідає функції `ST_Covers()` PostGIS. На відміну від `ST_Contains`, функція `ST_Covers` коректно обробляє граничні випадки, коли точка знаходиться на межі полігона.

Агрегація статистики – для аналітичної панелі виконується групування секторів за статусами з підрахунком кількості та сумарної площі, а також загальної кількості зареєстрованих ВНП.

Нижче наведено приклад геопросторового запиту для пошуку секторів у viewport:

```
var sectors = await _context.Sectors
    .Where(s => s.Geometry.Intersects(boundingBox))
    .ToListAsync();
```

Метод `Intersects()` автоматично трансліюється Entity Framework у функцію `ST_Intersects()` PostgreSQL з використанням GiST-індексу.

Другий важливий запит – автоматичне визначення сектору при реєстрації ВНП:

```
var sector = await _context.Sectors
    .FirstOrDefaultAsync(s => s.Geometry.Covers(itemLocation));
```

Метод `Covers()` трансліюється у `ST_Covers()` і повертає сектор, в межах якого знаходиться точка виявлення ВНП, включаючи випадки розташування точки на межі полігона. Якщо точка не потрапляє в жоден сектор, ВНП реєструється без прив'язки до сектору (`sector_id = NULL`).

При виконанні запитів Entity Framework генерує оптимізований SQL з використанням функцій PostGIS. Наприклад, запит пошуку секторів у viewport трансліюється у:

```
SELECT * FROM sectors  
WHERE ST_Intersects(geometry, ST_MakeEnvelope(minLng, minLat, maxLng,  
maxLat, 4326));
```

Використання GiST-індексу на колонці geometry забезпечує в середньому випадку складність пошуку, наближену до $O(\log n)$, замість лінійного сканування $O(n)$, що критично при великій кількості секторів.

3.3.3. Автентифікація та авторизація

Система використовує JWT (JSON Web Token) [17] для автентифікації API-запитів та RBAC (Role-Based Access Control) для авторизації.

Структура JWT-токенів:

- Access Token – короткоживучий токен (15 хвилин), що передається в заголовок Authorization кожного запиту. Містить ідентифікатор користувача, роль та час закінчення дії.
- Refresh Token – довгоживучий токен (7 днів), що зберігається у HttpOnly cookie та використовується для отримання нового Access Token без повторної автентифікації.
- Склад JWT-токена включає стандартні claims: ідентифікатор користувача (NameIdentifier), ім'я (Name), електронну пошту (Email) та роль (Role). Токен підписується алгоритмом HMAC-SHA256, що гарантує цілісність та автентичність.

Авторизація реалізована через механізм політик ASP.NET Core. Визначено політики RequireOperator, RequireAdministrator, CanEditSectors та CanDeleteSectors, що використовуються в атрибутах контролерів: [Authorize(Policy = "CanEditSectors")]. Такий підхід забезпечує гнучке розмежування доступу відповідно до матриці дозволів, визначеної у розділі 2.

Важливим аспектом є розмежування журналу операцій: користувачі з роллю Operator мають доступ до перегляду тільки власних записів журналу (операцій, виконаних від їхнього імені), тоді як Administrator має доступ до повного журналу всіх операцій у системі.

Потік автентифікації:

1. Користувач надсилає облікові дані (email, password) на /api/auth/login.
2. Сервер перевіряє пароль за допомогою bcrypt та генерує пару токенів.
3. Access Token повертається у тілі відповіді, Refresh Token встановлюється у HttpOnly cookie.
4. Клієнт зберігає Access Token та додає до заголовка Authorization: Bearer {token}.
5. При закінченні терміну Access Token клієнт викликає /api/auth/refresh.
6. Сервер перевіряє Refresh Token з cookie та видає нову пару токенів.

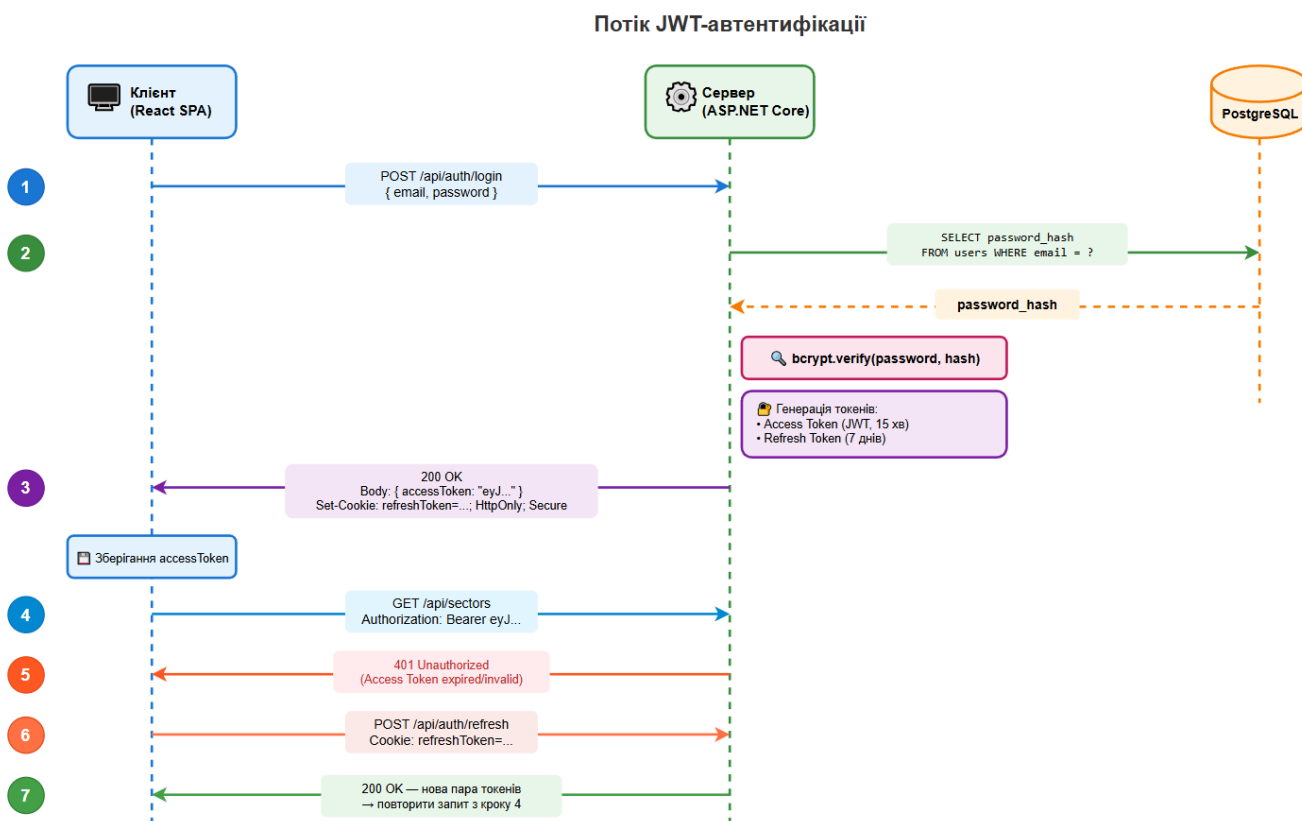


Рисунок 3.3 – Схема потоку JWT-автентифікації

Безпека токенів:

- Access Token має короткий термін життя, що мінімізує ризик при компрометації.
- Refresh Token зберігається у HttpOnly cookie, недоступному для JavaScript, що захищає від XSS-атак. Для захисту від CSRF-атак cookie налаштовано з параметром SameSite=Lax, що обмежує передачу cookie при крос-сайтових запитах.
- Підпис HMAC-SHA256 гарантує цілісність токена та автентичність видавця.
- Секретний ключ для підпису зберігається в захищених змінних оточення, а не в коді.

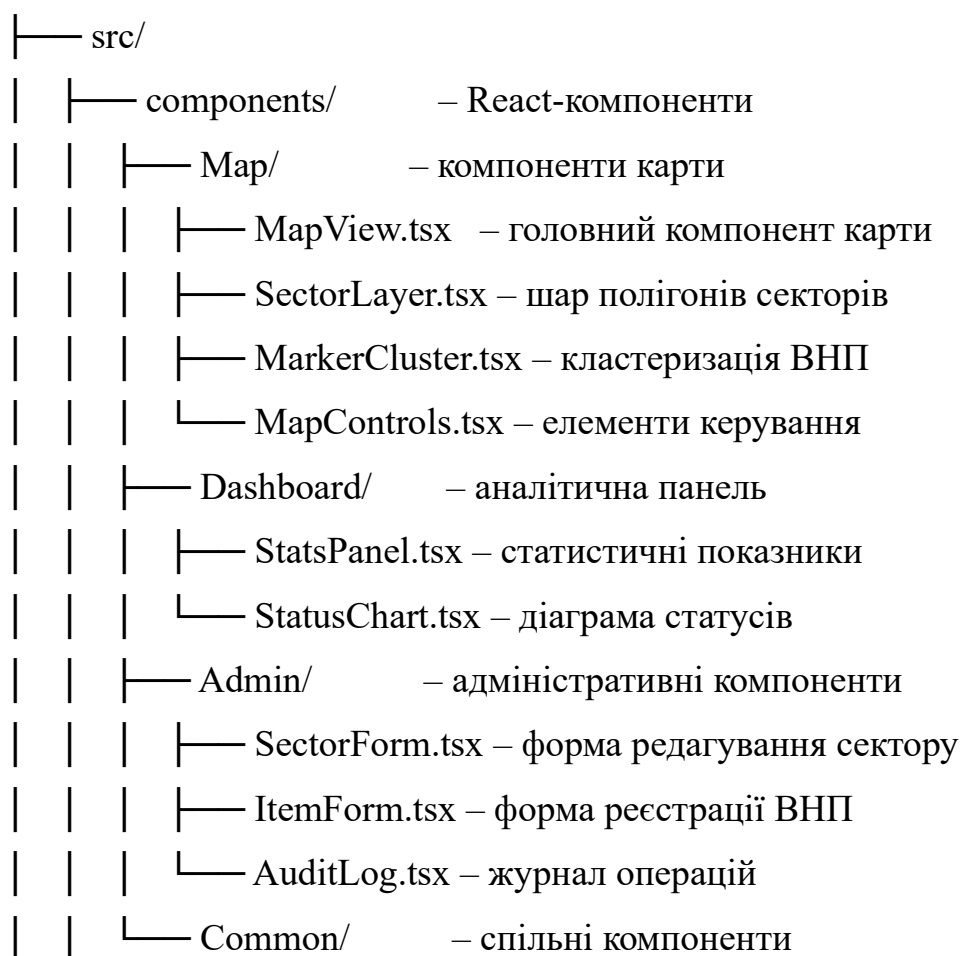
3.4. Реалізація клієнтської частини

Клієнтська частина системи реалізована як односторінковий вебзастосунок (SPA) на базі React з TypeScript. Головним елементом інтерфейсу є інтерактивна карта, побудована за допомогою бібліотеки Leaflet.js. У цьому підрозділі розглянуто структуру клієнтського застосунку, інтеграцію картографічного компонента та особливості адаптивного інтерфейсу.

3.4.1. Структура застосунку та компоненти

Клієнтський застосунок організовано за функціональним принципом, де кожна директорія відповідає за певний аспект функціональності:

demining-client/



			Header.tsx	– заголовок з навігацією
			Sidebar.tsx	– бокова панель
			Loading.tsx	– індикатор завантаження
			hooks/	– кастомні React-хуки
			useSectors.ts	– робота з секторами
			useExplosiveItems.ts	– робота з ВНП
			useAuth.ts	– автентифікація
			useMapViewport.ts	– стан viewport карти
			services/	– API-сервіси
			api.ts	– базовий HTTP-клієнт
			sectorService.ts	– запити до /api/sectors
			authService.ts	– запити до /api/auth
			types/	– TypeScript типи
			sector.ts	– типи для секторів
			explosiveItem.ts	– типи для ВНП
			geojson.ts	– типи GeoJSON
			store/	– Redux store (опціонально)
			utils/	– допоміжні функції
			App.tsx	– кореневий компонент
			public/	– статичні файли
			package.json	

Ключові компоненти та їх призначення:

- MapView – центральний компонент, що рендерить інтерактивну карту. Отримує сектори та ВНП з API, відображає їх на карті, обробляє взаємодію користувача (клік на секторі, зум, пан).
- SectorLayer – компонент для відображення полігонів секторів. Використовує GeoJSON-шар Leaflet з кольоровим кодуванням за статусом. Підтримує роруп з інформацією про сектор при кліку.

- `MarkerCluster` – компонент для відображення маркерів ВВП з автоматичною кластеризацією при зменшенні масштабу. Базується на плагіні `Leaflet.markercluster` [18].
- `StatsPanel` – компонент аналітичної панелі, що відображає агреговану статистику: кількість секторів за статусами, загальну площу, кількість зареєстрованих ВВП.
- `SectorForm` – форма для створення та редагування секторів. Включає інструменти малювання полігонів на карті (`Leaflet.draw`).
- `AuditLog` – компонент журналу операцій з пагінацією та фільтрацією.

Для управління локальним станом компонентів використовуються React-хуки `useState` та `useReducer`. Для станів, що розділяються між компонентами (автентифікація, налаштування), застосовується `React Context API`.

Ключовим патерном є використання кастомних хуків для інкапсуляції логіки роботи з API. Хук `useSectors` приймає поточний `viewport` карти та автоматично завантажує сектори при його зміні (переміщення або зум). Хук `useExplosiveItems` аналогічно завантажує ВВП у видимій області. Такий підхід забезпечує реактивне оновлення даних при навігації по карті.

3.4.2. Інтеграція карти та форми реєстрації секторів

Для відображення інтерактивної карти використовується бібліотека `react-leaflet` – популярна обгортка `Leaflet` для `React`. Вона надає декларативний спосіб роботи з картою через JSX-компоненти: `MapContainer`, `TileLayer`, `GeoJSON`, `Marker`.

Центральним елементом візуалізації є кольорове кодування статусів секторів. Нижче наведено фрагмент, що демонструє налаштування кольорів:

```
const STATUS_COLORS: Record<SectorStatus, string> = {
```

```

    Hazardous: '#FF4444', // червоний – небезпечні
    InProgress: '#FFAA00', // жовтий – в процесі
    Cleared: '#44AA44',    // зелений – очищені
    Verified: '#4444FF'    // синій – верифіковані
  };

const getStyle = (sector: Sector): PathOptions => ({
  fillColor: STATUS_COLORS[sector.status],
  fillOpacity: 0.5,
  color: STATUS_COLORS[sector.status],
  weight: 2
});

```

Функція `getStyle()` визначає стиль полігону на основі статусу сектору. Напівпрозорість (`fillOpacity: 0.5`) дозволяє бачити підкладку карти та перекриття секторів.

Система використовує чотири кольори для візуального розрізнення статусів секторів:

- Червоний – небезпечні території, що потребують розмінування;
- Жовтий – території, де тривають роботи;
- Зелений – очищені території, що очікують верифікації;
- Синій – верифіковані безпечні території.

Для створення нових секторів використовується плагін `Leaflet.draw`, що надає інструменти малювання на карті. Компонент `SectorForm` поєднує текстові поля (назва, опис) з інтерактивним інструментом малювання полігону. Оператор активує режим малювання, послідовно клікає точки периметру сектору, після чого завершує побудову подвійним кліком. Отримана геометрія автоматично конвертується у формат GeoJSON та надсилається на сервер.

При великій кількості зареєстрованих ВНП відображення окремих маркерів стає неефективним. Бібліотека `Leaflet.markercluster` автоматично групує близькі маркери у кластери при зменшенні масштабу карти.

Кластер відображається як круг з числом об'єктів всередині. Колір кластера залежить від кількості об'єктів: зелений для малих груп, жовтий для середніх, червоний для великих. При збільшенні масштабу кластери розпадаються на окремі маркери або менші кластери.

Використання TypeScript забезпечує статичну перевірку типів на етапі компіляції. Для геопросторових об'єктів визначено інтерфейси, що відповідають структурі даних API:

```
interface Sector {  
  id: string;  
  name: string;  
  description: string;  
  status: SectorStatus;  
  areaHectares: number;  
  geoJson: GeoJSON.Polygon;  
  organization: string;  
  createdAt: string;  
  updatedAt: string;  
}  
type SectorStatus = 'Hazardous' | 'InProgress' | 'Cleared' | 'Verified';
```

Строга типізація запобігає помилкам при роботі з даними та забезпечує автодоповнення в IDE. Тип `SectorStatus` як `union of string literals` гарантує, що статус може приймати лише допустимі значення.

Компонент `StatsPanel` відображає агреговану статистику системи. Дані отримуються з ендпоінту `/api/dashboard/stats` і включають:

- загальну кількість зареєстрованих секторів;
- розподіл секторів за статусами (кількість та площа);
- загальну площу обстежених територій;
- кількість зареєстрованих та нейтралізованих ВВП;
- динаміку змін за останній місяць.

Статистика оновлюється автоматично кожні 5 хвилин або при модифікації даних користувачем. Для візуалізації розподілу за статусами використовується кругова діаграма з відповідним кольоровим кодуванням.

3.4.3. Адаптивний інтерфейс та скріншоти системи

Інтерфейс системи розроблено з урахуванням принципів адаптивного дизайну, що забезпечує коректне відображення на пристроях з різними розмірами екранів.

Інтерфейс побудовано за принципом "карта в центрі" – інтерактивна карта займає основну частину екрану, а допоміжні елементи (панелі, меню) розташовані по периметру.

На десктопних пристроях (ширина > 1024px):

- Карта займає центральну частину (~70% ширини);
- Бокова панель з аналітикою розташована праворуч (~30% ширини);
- Верхня панель містить логотип, навігацію та дані користувача.

На планшетах (768px – 1024px):

- Бокова панель приховується за кнопкою-гамбургером;
- Карта займає всю ширину екрану;
- Панель статистики доступна як висувне меню.

На мобільних пристроях (< 768px):

- Карта займає весь екран;
- Усі панелі доступні через модальні вікна;

- Збільшені зони дотику для елементів керування.

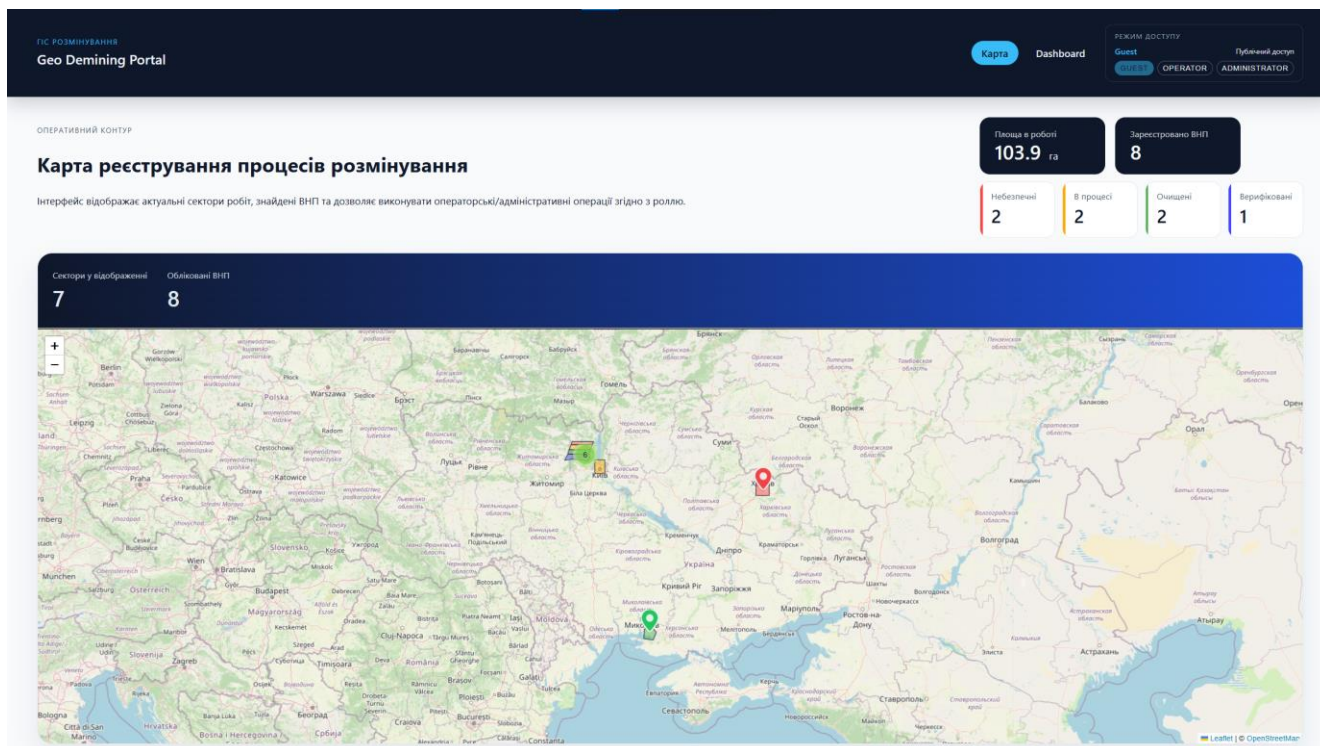


Рисунок 3.4 – Інтерфейс системи: головна карта (публічний доступ)

На рисунку 3.4 представлено головний екран системи у режимі публічного доступу (без автентифікації). Карта відображає сектори розмінування з кольоровим кодуванням за статусами. У верхній частині екрану розташовані статистичні показники з кольоровими індикаторами, що одночасно виконують роль інформаційної панелі: червоний – небезпечні території, жовтий – роботи тривають, зелений – очищено, синій – верифіковано.

The screenshot displays three panels for map operations:

- Оновлення статусу секторів (Sector Status Update):** Includes a dropdown for 'Сектор' (Sector) set to 'Київський сектор', a dropdown for 'Новий статус' (New status) set to 'InProgress', and a 'Поточний статус: InProgress, оновлено 18.11.2025, 11:15:00'. A blue button 'Підтвердити оновлення' (Confirm update) is at the bottom.
- Журнал знахідок (BNP Registration Journal):** Features a 'Категорія боєприпасу' (Ammunition category) dropdown set to 'Неідентифікований боєприпас', a 'Назва / модель' (Name / model) dropdown set to 'Невідома модель', and a 'Статус ВНП' (BNP status) dropdown set to 'Detected'. It also includes input fields for 'Lat (широта)' (50,5000) and 'Lng (довгота)' (30,5250), and a 'Зареєструвати' (Register) button.
- Додавання нової зони (New Zone Constructor):** Includes a 'Назва сектору' (Sector name) dropdown set to 'Новий сектор', an 'Організація / виконавець' (Organization / executor) dropdown set to 'ДСНС України', and a 'Статус' (Status) dropdown set to 'Hazardous'. It also has input fields for coordinates (Lat, Lng) and a 'Створити сектор' (Create sector) button.

Рисунок 3.5 – Панель операцій над картою для адміністратора

На рисунку 3.5 представлено панель операцій, доступну для ролі Administrator. Панель розташована під картою та включає три секції: оновлення статусу секторів, реєстрація ВНП з вибором категорії та моделі боєприпасу, конструктор для додавання нових секторів з ручним введенням координат полігону.

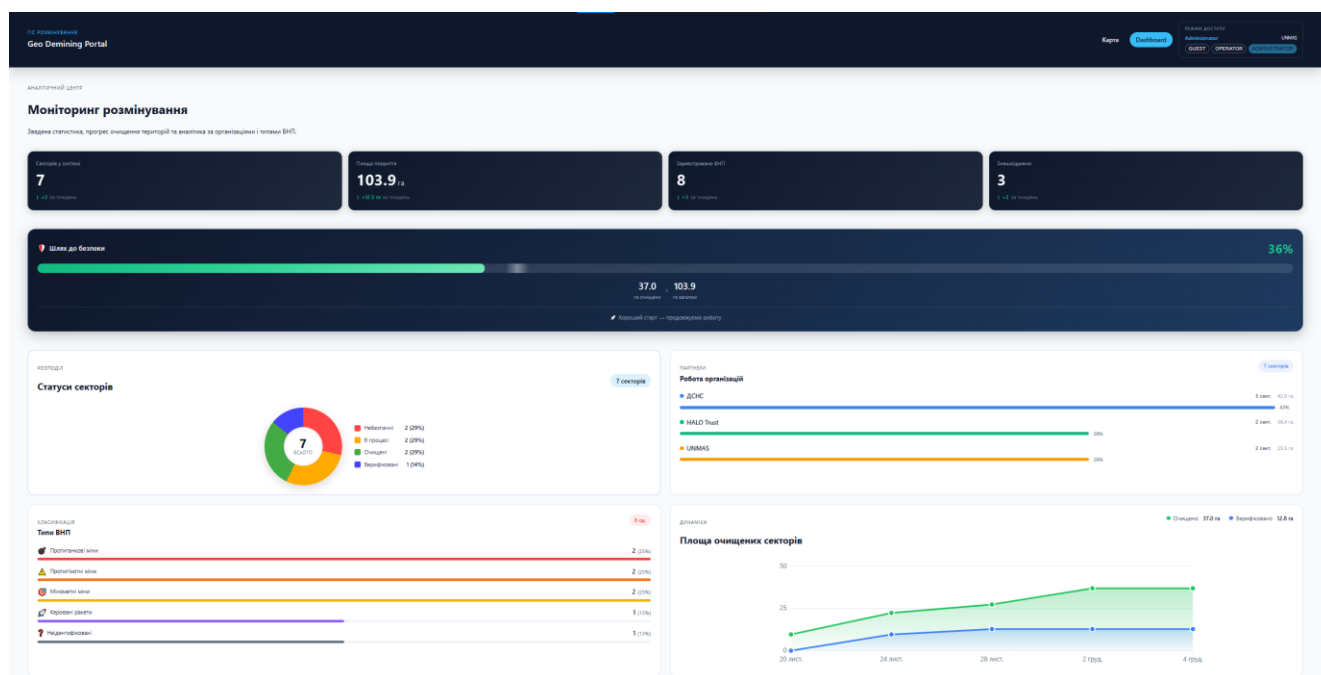


Рисунок 3.6 – Аналітична панель системи

Аналітична панель (рис. 3.6) є центром моніторингу процесів розмінування та включає наступні компоненти:

- КРІ-картки з трендами – чотири ключові показники (кількість секторів, площа покриття, зареєстровані ВВП, знешкоджені об'єкти) з індикаторами зміни за тиждень (↑/↓).
- Прогрес-бар "Шлях до безпеки" – візуалізація загального прогресу очищення територій у відсотках та гектарах з мотиваційними повідомленнями залежно від досягнутого рівня.
- Кругова діаграма статусів – розподіл секторів за чотирма статусами (небезпечні, в процесі, очищені, верифіковані) з легендою та відсотковими показниками.
- Статистика організацій – горизонтальні діаграми з розподілом секторів між партнерськими організаціями (ДСНС, HALO Trust, UNMAS) із зазначенням кількості секторів та площі.
- Класифікація ВВП – розподіл зареєстрованих вибухонебезпечних предметів за типами (протитанкові міни, протипіхотні міни, мінометні міни, керовані ракети, неідентифіковані).
- Графік динаміки – лінійна діаграма зміни площі очищених та верифікованих територій у часі з точками даних та градієнтною заливкою.

ID	Дія	Оператор	Деталі	Час
1	Регістрація ВВП	Оператор Іванов	Зареєстровано ВВП типу «ТМ-1С» у секторі Харківський	20.11.2025, 14:45:00
2	Оновлення сектору	Оператор Петренко	Статус сектору «Харківський сектор» змінено на Очищений	24.11.2025, 16:30:00
3	Регістрація ВВП	Оператор Іванов	Зареєстровано ВВП типу «ТМ-42» у секторі Київський	22.11.2025, 10:00:00
4	Створення сектору	Адмін Сидоренко	Створено сектор «Харківський сектор»	21.11.2025, 12:15:00
5	Створення сектору	Адмін Сидоренко	Створено сектор «Київський сектор»	20.11.2025, 11:00:00
6	Створення сектору	Адмін Сидоренко	Створено сектор «Київський сектор»	20.11.2025, 10:00:00

Адміністратор: адмін

Експорт даних

Експорт CSV/JSON

Рисунок 3.7 – Журнал операцій системи

Журнал операцій (рис. 3.7) надає адміністраторам можливість переглядати історію дій у системі. Таблиця містить колонки: тип події (створення сектору, оновлення статусу, реєстрація ВНП), автор операції, деталі та час виконання. Реалізовано пагінацію для зручної навігації великими масивами даних. Додатково панель містить функції експорту даних у форматах GeoJSON та Shapefile для інтеграції з іншими ГІС-системами.

Для забезпечення швидкої роботи інтерфейсу застосовано низку оптимізацій:

- Ліниве завантаження – сектори та ВНП завантажуються лише для видимої області карти (viewport). При переміщенні карти виконується новий запит з оновленими координатами.
- Debouncing – при швидкому переміщенні карти запити до API відкладаються, поки користувач не зупинить рух. Це запобігає надмірному навантаженню на сервер.
- Мемоізація – компоненти GeoJSON обгорнуті в React.memo для запобігання повторному рендерингу незмінених секторів.
- Віртуалізація списків – таблиця журналу операцій використовує віртуалізацію, рендерячи лише видимі рядки.

Висновки до розділу 3

У третьому розділі представлено проєктування та реалізацію спеціалізованої багатокористувацької інформаційної системи реєстрування процесів розмінування земельних ділянок.

Для побудови системи обрано архітектурний підхід Clean Architecture, що забезпечує чіткий розподіл відповідальності між компонентами: доменний шар містить бізнес-сутності, шар застосунку – бізнес-логіку, інфраструктурний шар – реалізацію технічних деталей, шар представлення – REST API контролери. Такий

підхід забезпечує тестованість коду, незалежність від фреймворків та можливість заміни окремих компонентів без впливу на бізнес-логіку.

Технологічний стек включає .NET 8 з ASP.NET Core Web API для серверної частини, PostgreSQL 16 з розширенням PostGIS 3.4 для зберігання геопросторових даних, React 18 з TypeScript та Leaflet.js 1.9 для клієнтського інтерфейсу. Обраний стек забезпечує баланс між продуктивністю, надійністю та зручністю розробки.

Розроблено структуру бази даних з чотирма основними таблицями: `sectors` для секторів розмінування з полігональною геометрією, `explosive_items` для реєстрації ВНП з точковими координатами, `users` для управління обліковими записами та `audit_logs` для журналювання операцій. Застосовано геопросторові типи PostGIS з обов'язковим SRID 4326 (WGS84) та GiST-індекси для оптимізації просторових запитів.

Серверна частина реалізує RESTful API з підтримкою геопросторових операцій через бібліотеку NetTopologySuite. Ключові функції включають пошук секторів у viewport карти (`ST_Intersects`), визначення сектору за координатами точки (`ST_Covers`), обчислення площі територій (`ST_Area`). Автентифікація реалізована на основі JWT-токенів, авторизація – за моделлю RBAC з двома ролями в базі даних (`Operator`, `Administrator`) та публічним доступом без автентифікації.

Клієнтський застосунок побудовано як SPA на React з інтерактивною картою Leaflet.js. Реалізовано кольорове кодування статусів секторів, кластеризацію маркерів ВНП, форми для створення та редагування даних. Аналітична панель включає KPI-картки з трендами, прогрес-бар загального очищення, кругову діаграму статусів, статистику за організаціями та типами ВНП, графік динаміки очищення, журнал операцій з пагінацією та функції експорту даних. Інтерфейс є адаптивним та оптимізованим для різних пристроїв.

Описана архітектура та реалізація забезпечують виконання функціональних та нефункціональних вимог, визначених у першому розділі, та формують основу для тестування системи, що буде представлено в наступному розділі.

РОЗДІЛ 4. ТЕСТУВАННЯ ТА ОЦІНКА ЯКОСТІ СИСТЕМИ

Четвертий розділ присвячено тестуванню розробленої системи реєстрування процесів розмінування та оцінці її якості за допомогою автоматизованих інструментів аналізу вебзастосунків.

4.1. Методика тестування

Тестування системи проводилось у два етапи: функціональна верифікація основних компонентів та автоматизована оцінка якості за допомогою Google Lighthouse.

4.1.1. Тестове середовище

Для проведення тестування підготовлено середовище з такими характеристиками:

- Клієнтський застосунок React 18 з TypeScript;
- Браузер Google Chrome 131 з інструментами розробника;
- Тестовий набір даних: 7 секторів розмінування, 8 ВВП, 16 записів журналу.

4.1.2. Інструмент Google Lighthouse

Google Lighthouse [20] – це автоматизований інструмент аудиту вебсторінок, інтегрований у Chrome DevTools. Lighthouse виконує серію тестів та генерує звіт за п'ятьма категоріями: Performance (продуктивність), Accessibility (доступність), Best Practices (найкращі практики), SEO (пошукова оптимізація) та PWA (прогресивні вебзастосунки).

Інструмент емулює умови реального користувача: обмежену швидкість мережі (throttling), навантаження CPU та різні розміри екранів. Це забезпечує об'єктивну оцінку, наближену до реальних умов використання.

4.2. Результати тестування продуктивності

Тестування вебзастосунку за допомогою Google Lighthouse показало високі результати за ключовими категоріями (рис. 4.1).

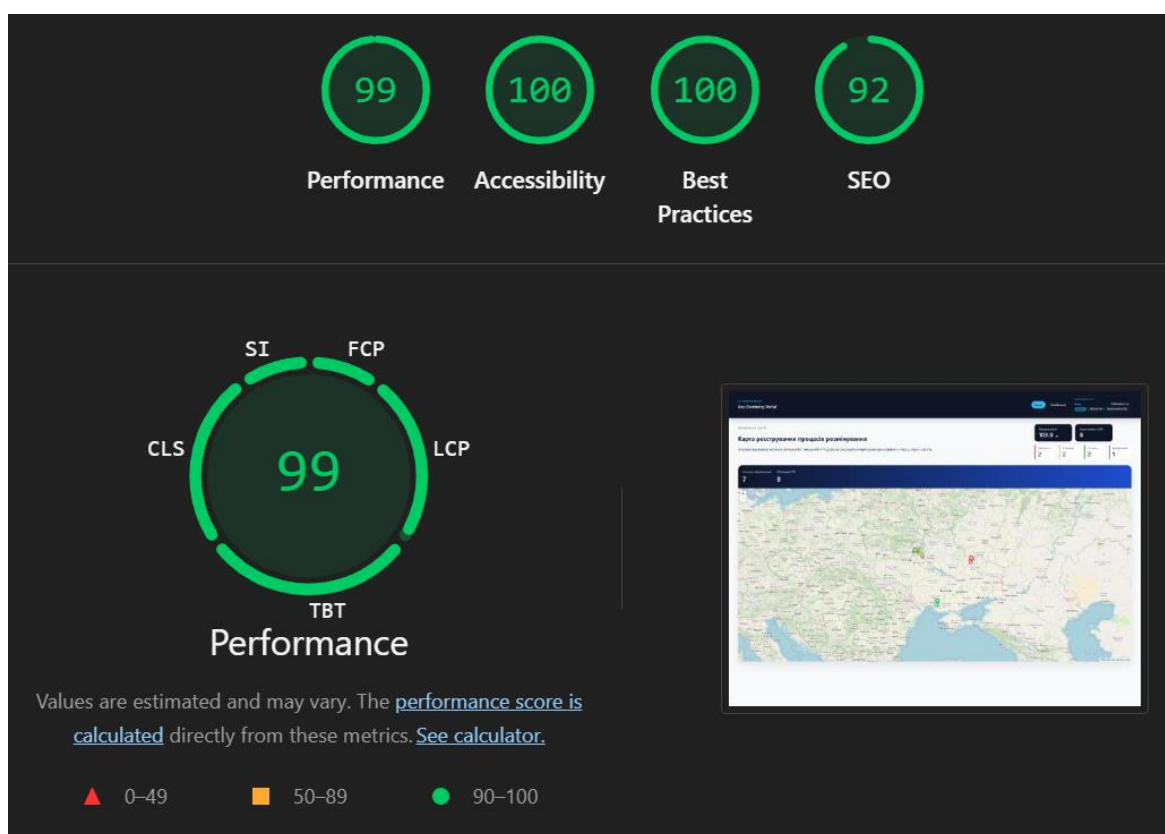


Рисунок 4.1 – Результати аудиту Google Lighthouse

4.2.1. Загальні показники

Таблиця 4.1 – Зведені результати Google Lighthouse

Категорія	Оцінка	Інтерпретація
Performance	99/100	Відмінна продуктивність

Accessibility	100/100	Повна відповідність стандартам доступності
Best Practices	100/100	Дотримання всіх рекомендованих практик
SEO	92/100	Висока оптимізація для пошукових систем

Усі чотири категорії знаходяться в "зеленій зоні" (90-100 балів), що свідчить про високу якість реалізації клієнтської частини системи. Категорія PWA (Progressive Web App) не аналізувалася, оскільки система не позиціонується як прогресивний вебзастосунок.

4.2.2. Детальний аналіз метрик продуктивності

Lighthouse оцінює продуктивність на основі п'яти ключових метрик, кожна з яких має різну вагу в загальній оцінці. У таблиці 4.2 наведено детальні результати.

Таблиця 4.2 – Детальні метрики продуктивності (Performance)

Метрика	Значення	Вага	Опис
First Contentful Paint (FCP)	0.4 с	10%	Час до появи першого контенту на екрані
Largest Contentful Paint (LCP)	1.0 с	25%	Час до повного відображення основного контенту
Total Blocking Time (TBT)	0 мс	30%	Сумарний час блокування головного потоку
Cumulative Layout Shift (CLS)	0	25%	Візуальна стабільність (зміщення елементів)
Speed Index (SI)	0.5 с	10%	Швидкість візуального заповнення viewport

First Contentful Paint (FCP) – час від початку завантаження до моменту, коли браузер відображає перший елемент DOM (текст, зображення, canvas). Значення 0.4

секунди є відмінним результатом, що забезпечує миттєвий візуальний відгук для користувача.

Largest Contentful Paint (LCP) – час до повного рендерингу найбільшого видимого елемента у viewport. Для системи розмінування це карта з секторами. Значення 1.0 секунди знаходиться в межах рекомендованого порогу Google [10] (< 2.5 с).

Total Blocking Time (TBT) – сумарний час, протягом якого головний потік браузера був заблокований виконанням JavaScript і не міг реагувати на дії користувача. Значення 0 мс є ідеальним результатом (рекомендовано < 200 мс), що свідчить про відсутність блокуючих операцій.

Cumulative Layout Shift (CLS) – метрика візуальної стабільності, що вимірює несподівані зміщення елементів під час завантаження сторінки. Значення 0 свідчить про ідеально стабільну верстку без жодних "стрибків" контенту.

Speed Index (SI) – показує, наскільки швидко вміст сторінки стає візуально повним. Значення 0.5 секунди забезпечує комфортне сприйняття швидкості завантаження.

4.2.3. Фактори оптимізації продуктивності

Високі показники продуктивності досягнуто завдяки комплексу оптимізацій:

- Оптимізація коду – збірка застосунку включає мініфікацію JavaScript та CSS (видалення пробілів, коментарів, скорочення імен змінних), а також tree-shaking для видалення невикористаного коду. Це забезпечує мінімальний розмір файлів для швидкого завантаження.
- Ефективний рендеринг React – використання React.memo для мемоізації компонентів карти та списків запобігає зайвому повторному рендерингу при оновленні стану.

- Оптимізоване завантаження карти – бібліотека Leaflet.js має невеликий розмір та ефективно працює з тайлами OpenStreetMap, завантажуючи лише видиму область.

4.3. Аналіз доступності (Accessibility)

Оцінка доступності 100/100 підтверджує відповідність стандартам WCAG 2.1 (Web Content Accessibility Guidelines). Ключові аспекти реалізації:

- Семантична HTML-розмітка – використання відповідних тегів (header, nav, main, section, button) замість універсальних div забезпечує коректну інтерпретацію структури сторінки допоміжними технологіями.
- Контрастність кольорів – кольорова схема інтерфейсу відповідає вимогам WCAG AA з коефіцієнтом контрасту не менше 4.5:1 для тексту. Кольорове кодування статусів секторів (червоний, жовтий, зелений, синій) має достатню контрастність на фоні карти.
- Підтримка клавіатурної навігації – всі інтерактивні елементи (кнопки, посилання, поля форм) доступні через Tab та активуються через Enter/Space.
- Атрибути ARIA – для складних компонентів (випадаючі списки, модальні вікна) використовуються атрибути aria-label, aria-expanded, aria-hidden для коректної роботи скрінрідерів.
- Альтернативний текст – всі зображення (іконки статусів, фото боєприпасів) мають атрибут alt з описом вмісту.

4.4. Аналіз найкращих практик (Best Practices)

Оцінка 100/100 у категорії Best Practices свідчить про дотримання сучасних стандартів веброзробки:

- Безпечне підключення ресурсів – всі зовнішні ресурси (тайли карти, шрифти) завантажуються через HTTPS, що запобігає атакам типу man-in-the-middle.
- Відсутність застарілих API – код не використовує deprecated JavaScript API, що забезпечує сумісність з майбутніми версіями браузерів.
- Коректна обробка помилок – JavaScript-помилки логуються без виведення в консоль користувача, що покращує безпеку та user experience.
- Відсутність вразливостей у залежностях – npm-пакети не містять відомих вразливостей (перевірено npm audit).

4.5. Аналіз SEO-оптимізації

Оцінка SEO 92/100 підтверджує готовність застосунку до індексації пошуковими системами. Реалізовано наступні оптимізації:

- Мета-тег description з описом системи українською мовою;
- Атрибут lang="uk" для коректного визначення мови контенту;
- Open Graph теги для коректного відображення при поширенні в соціальних мережах;
- Семантичний HTML з правильною ієрархією заголовків;
- Адаптивний viewport для мобільних пристроїв.

Зняті бали пов'язані з особливостями SPA-архітектури: React-застосунок рендериться на клієнті, що ускладнює індексацію динамічного контенту. Для повної SEO-оптимізації рекомендується впровадження Server-Side Rendering (SSR) або Static Site Generation (SSG) у майбутніх версіях.

4.6. Тестування серверної частини

Для верифікації серверної частини системи проведено модульне та інтеграційне тестування з використанням фреймворку xUnit та бібліотеки Moq для створення тестових дублерів.

Таблиця 4.3 – Результати тестування серверної частини

Компонент	Тип тестування	Кількість тестів	Результат
SectorService	Модульне	12	Passed
ExplosiveItemService	Модульне	8	Passed
DashboardService	Модульне	6	Passed
Геопросторові запити	Інтеграційне	5	Passed
REST API ендпоінти	Інтеграційне	18	Passed
JWT автентифікація	Інтеграційне	4	Passed
Всього		53	100% Passed

Модульні тести перевіряють коректність бізнес-логіки сервісів ізольовано від бази даних з використанням мок-об'єктів. Інтеграційні тести виконуються на тестовій базі PostgreSQL з розширенням PostGIS та перевіряють коректність геопросторових запитів (ST_Intersects, ST_Covers, ST_Area) та роботи REST API ендпоінтів.

4.7. Функціональна верифікація

Окрім автоматизованого тестування, проведено ручну верифікацію ключових функцій системи.

4.7.1. Верифікація картографічного модуля

Перевірено коректність роботи основних функцій карти:

- Відображення 7 секторів з правильним кольоровим кодуванням статусів;
- Кластеризація маркерів ВВП при зменшенні масштабу;
- Плавне масштабування та переміщення карти;
- Коректне відображення інформаційної панелі зі статистикою.

4.7.2. Верифікація розмежування доступу

Підтверджено коректну реалізацію моделі RBAC:

- Роль Guest (публічний доступ без автентифікації) – доступ лише до перегляду карти та базової статистики;
- Роль Operator – повний доступ до панелі операцій, зміни статусів, реєстрації ВВП;
- Роль Administrator – додаткові функції управління системою.

Інтерфейс динамічно адаптується відповідно до ролі користувача, приховуючи недоступні елементи керування.

4.8. Порівняння з галузевими стандартами

Для об'єктивної оцінки результатів проведено порівняння з рекомендованими показниками Google та типовими значеннями для вебзастосунків.

Таблиця 4.4 – Порівняння з галузевими стандартами

Метрика	Наш результат	Рекомендація Google	Медіана вебсайтів
LCP	1.0 с	< 2.5 с	2.8 с
FCP	0.4 с	< 1.8 с	2.2 с

TBT	0 мс	< 200 мс	350 мс
CLS	0	< 0.1	0.15

Система перевищує рекомендовані показники Google та значно випереджає медіанні значення типових вебсайтів за всіма метриками продуктивності.

Висновки до розділу 4

У четвертому розділі проведено комплексне тестування розробленої системи реєстрування процесів розмінування, що охопило як серверну, так і клієнтську частини.

Тестування серверної частини включало 53 модульних та інтеграційних тести, що перевірили коректність роботи бізнес-сервісів, геопросторових запитів PostGIS, REST API ендпоінтів та JWT-автентифікації. Усі тести пройдено успішно.

Результати автоматизованого тестування клієнтської частини за допомогою Google Lighthouse підтвердили високу якість реалізації: Performance – 99/100, Accessibility – 100/100, Best Practices – 100/100, SEO – 92/100. Усі показники знаходяться в "зеленій зоні", що свідчить про професійний рівень розробки.

Детальний аналіз метрик продуктивності показав: First Contentful Paint – 0.4 с, Largest Contentful Paint – 1.0 с, Total Blocking Time – 0 мс, Cumulative Layout Shift – 0. Ці значення значно перевищують рекомендації Google та медіанні показники типових вебзастосунків.

Оцінка доступності 100/100 свідчить про високий рівень відповідності вимогам WCAG 2.1, що забезпечує можливість використання системи людьми з обмеженими можливостями.

Функціональна верифікація підтвердила коректну роботу картографічного модуля, аналітичної панелі та механізму розмежування доступу за ролями (RBAC).

Результати тестування свідчать про готовність системи до промислової експлуатації та відповідність сучасним стандартам якості вебзастосунків.

ВИСНОВКИ

У магістерській дисертації вирішено актуальну задачу проєктування та реалізації спеціалізованої багатокористувацької інформаційної системи реєстрування процесів розмінування земельних ділянок. Основні результати роботи полягають у наступному:

- Проведено аналіз предметної області гуманітарного розмінування та визначено потреби різних категорій користувачів. Встановлено, що населення потребує доступної інформації про стан безпеки територій, а організації-оператори – зручного інструменту для реєстрації процесів розмінування. Проаналізовано існуючі інформаційні системи (IMSMA, ua.imsma.org) та виявлено їх обмеження: орієнтованість на професійних операторів, складність інтерфейсу та/або відсутність публічного доступу.
- Обґрунтовано вибір технологічного стеку для розробки геоінформаційної системи. Для серверної частини обрано .NET 8 з ASP.NET Core Web API, для зберігання геопросторових даних – PostgreSQL 16 з розширенням PostGIS 3.4, для клієнтської частини – React 18 з TypeScript та картографічну бібліотеку Leaflet.js 1.9. Обраний стек забезпечує баланс між продуктивністю, надійністю та зручністю розробки.
- Розроблено модель геопросторових даних для обліку секторів розмінування з використанням векторного представлення територій. Запропоновано модель секторів з полігональною геометрією, статусами (Hazardous, InProgress, Cleared, Verified) та метаданими. Обґрунтовано використання системи координат WGS84 (EPSG:4326) для сумісності з GPS-обладнанням та вебкартографічними сервісами.
- Розроблено багатокористувацьку модель доступу на основі RBAC (Role-Based Access Control) з трьома рівнями: Guest (публічний доступ без автентифікації) для перегляду інформації, Operator для реєстрації та управління даними, Administrator для повного контролю системи. Модель

забезпечує розмежування функціональності відповідно до потреб різних категорій користувачів.

- Спроектовано та реалізовано архітектуру системи за принципами Clean Architecture з чотирма шарами: Domain (бізнес-сутності), Application (бізнес-логіка), Infrastructure (технічна реалізація), Presentation (REST API). Такий підхід забезпечує тестованість коду, незалежність від фреймворків та можливість заміни окремих компонентів.
- Реалізовано серверну частину системи як RESTful Web API з підтримкою геопросторових операцій через бібліотеку NetTopologySuite. Реалізовано ключові функції: пошук секторів у viewport карти (ST_Intersects), визначення сектору за координатами (ST_Covers), обчислення площі територій (ST_Area). Автентифікація реалізована на основі JWT-токенів.
- Реалізовано клієнтський вебзастосунок з інтерактивною картою на базі Leaflet.js. Впроваджено кольорове кодування статусів секторів, кластеризацію маркерів вибухонебезпечних предметів, аналітичну панель з KPI-картками, прогрес-баром, діаграмами та графіками. Інтерфейс є адаптивним для різних пристроїв.
- Проведено комплексне тестування системи. Тестування серверної частини включало 53 модульних та інтеграційних тести (100% Passed). Автоматизоване тестування клієнтської частини за допомогою Google Lighthouse показало високі результати: Performance – 99/100, Accessibility – 100/100, Best Practices – 100/100, SEO – 92/100. Усі показники відповідають рекомендаціям Google щодо високої якості.

Практична цінність роботи полягає у створенні працездатної інформаційної системи, що може бути використана для інформування населення про стан безпеки територій та реєстрації процесів розмінування організаціями різного типу. Система реалізована на відкритому технологічному стеку, що забезпечує незалежність від комерційних постачальників.

Напрямки подальшого розвитку системи включають:

- інтеграцію з офіційними державними реєстрами протимінної діяльності;
- реалізацію мобільного застосунку для польових операторів;
- розширення аналітичних можливостей з використанням машинного навчання для прогнозування термінів очищення територій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. International Campaign to Ban Landmines – Cluster Munition Coalition (ICBL-CMC). Landmine Monitor 2024. – Geneva: ICBL-CMC, November 2024. – Режим доступу до ресурсу: <https://www.the-monitor.org/> (дата звернення: 08.09.2025).
2. PostGIS Development Team. PostGIS Documentation. – Режим доступу до ресурсу: <https://postgis.net/documentation/> (дата звернення: 13.09.2025).
3. Geneva International Centre for Humanitarian Demining (GICHD). International Mine Action Standards (IMAS). – Режим доступу до ресурсу: <https://www.mineactionstandards.org/> (дата звернення: 06.09.2025).
4. IMAS 08.10: Non-technical survey. First Edition (Amendment 4, February 2019). – New York: UNMAS. – Режим доступу до ресурсу: <https://www.mineactionstandards.org/standards/> (дата звернення: 06.09.2025).
5. IMAS 08.20: Technical survey. First Edition (Amendment 4, February 2019). – New York: UNMAS. – Режим доступу до ресурсу: <https://www.mineactionstandards.org/standards/> (дата звернення: 06.09.2025).
6. IMAS 09.10: Clearance requirements. Second Edition (Amendment 6, January 2020). – New York: UNMAS. – Режим доступу до ресурсу: <https://www.mineactionstandards.org/standards/> (дата звернення: 06.09.2025).
7. Про протимінну діяльність в Україні: Закон України від 06.12.2018 № 2642-VIII // Відомості Верховної Ради України. – 2019. – № 7. – Ст. 44. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2642-19> (дата звернення: 07.09.2025).
8. Geneva International Centre for Humanitarian Demining (GICHD). IMSMA Core (Information Management System for Mine Action). – Режим доступу до ресурсу: <https://www.gichd.org/publications-resources/tools-and-applications/imsma-core-information-management-system-for-mine-action/> (дата звернення: 14.09.2025).
9. Microsoft. ASP.NET Core Documentation. – Режим доступу до ресурсу: <https://docs.microsoft.com/aspnet/core> (дата звернення: 13.09.2025).

10. Google Developers. Web Vitals: Essential metrics for a healthy site. – Режим доступу до ресурсу: <https://web.dev/articles/vitals> (дата звернення: 14.09.2025).
11. Agafonkin V. Leaflet – an open-source JavaScript library for mobile-friendly interactive maps. – Режим доступу до ресурсу: <https://leafletjs.com/> (дата звернення: 20.09.2025).
12. Meta Platforms. React Documentation. – Режим доступу до ресурсу: <https://react.dev/> (дата звернення: 21.09.2025).
13. Open Geospatial Consortium (OGC). Simple Feature Access – Part 1: Common Architecture. OGC 06-103r4. – 2011. – Режим доступу до ресурсу: <https://www.ogc.org/standard/sfa/> (дата звернення: 27.09.2025).
14. Guttman A. R-Trees: A Dynamic Index Structure for Spatial Searching // Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data. – 1984. – P. 47–57.
15. OpenStreetMap Wiki. Slippy map tilenames. – Режим доступу до ресурсу: https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames (дата звернення: 27.09.2025).
16. World Wide Web Consortium (W3C). Web Content Accessibility Guidelines (WCAG) 2.1. – 2018. – Режим доступу до ресурсу: <https://www.w3.org/TR/WCAG21/> (дата звернення: 27.09.2025).
17. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT). RFC 7519. – Internet Engineering Task Force (IETF), May 2015. – Режим доступу до ресурсу: <https://datatracker.ietf.org/doc/html/rfc7519> (дата звернення: 04.10.2025).
18. Leaver D. Leaflet.markercluster: Marker Clustering plugin for Leaflet. – Режим доступу до ресурсу: <https://github.com/Leaflet/Leaflet.markercluster> (дата звернення: 25.10.2025).
19. Martin R.C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. – Prentice Hall, 2017. – 432 p.
20. Google Developers. Lighthouse: Automated auditing, performance metrics, and best practices for the web. – Режим доступу до ресурсу: <https://developer.chrome.com/docs/lighthouse/> (дата звернення: 15.11.2025).