**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**
**КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ**

# КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інформаційний портал з інтеграцією картографічних даних на безкоштовних хмарних сервісах»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерної інженерії
*(код, найменування спеціальності)*
освітньо-професійної програми Комп'ютерна інженерія
*(назва)*

*Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело*

| | |
|---|---|
| _____ | *Владислав КРАСАВІН* |
| *(підпис)* | *Ім'я, ПРІЗВИЩЕ здобувача* |

Виконав: здобувач вищої освіти гр. КІД-41
Владислав КРАСАВІН
Ім'я, ПРІЗВИЩЕ

Керівник: Сергій КОРОТКОВ
*науковий ступінь, вчене звання*
Ім'я, ПРІЗВИЩЕ

Рецензент: _____
*науковий ступінь, вчене звання*
Ім'я, ПРІЗВИЩЕ

**Київ 2025**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**Навчально-науковий інститут Інформаційних технологій**

Кафедра 123 Комп'ютерної інженерії

Ступінь вищої освіти Бакалавр

Спеціальність 123 Комп'ютерної інженерії

Освітньо-професійна програма Комп'ютерна інженерія

**ЗАТВЕРДЖУЮ**
Завідувач кафедрою Наталія ЛАЩЕВСЬКА
_____Ім'я, ПРІЗВИЩЕ
«_____» _____ 2025р.

# ЗАВДАННЯ
# НА КВАЛІФІКАЦІЙНУ РОБОТУ

Красавін Владислав Олександрович
*(прізвище, ім'я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: Інформаційний портал з інтеграцією картографічних даних на безкоштовних хмарних сервісах
керівник кваліфікаційної роботи Сергій КОРОТКОВ,
*(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)*
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. № 56
2. Строк подання кваліфікаційної роботи «5» червня 2025р.
3. Вихідні дані до кваліфікаційної роботи:
Вебтехнології: HTML, CSS, JavaScript
Бібліотеки: Leaflet.js
Хмарні сервіси: Firebase, GitHub Pages
Формат геоданих: GeoJSON
Інструменти: Visual Studio Code, Git
Приклади подібних рішень та довідкова література
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Дослідження принципів побудови інформаційного вебпорталу з картографією
2. Аналіз хмарних сервісів для розміщення та зберігання даних
3. Реалізація інформаційного порталу з інтеграцією карти на основі безкоштовних API.
Перелік ілюстративного матеріалу: *презентація*
5. Дата видачі завдання «24» лютого 2025р.

# КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Збір даних | 28.02-22.03.2025 | Виконано |
| 2 | Обробка матеріалу | 23.03-05.04.2025 | Виконано |
| 3 | Розробка теоретичної частини | 06.04-10.04.2025 | Виконано |
| 4 | Розробка проєкту картографії | 11.04-30.04.2025 | Виконано |
| 5 | Впровадження деяких функцій захисту в проєкт | 01.05-03.05.2025 | Виконано |
| 6 | Висновки за результатами | 04.05-08.05.2025 | Виконано |
| 7 | Розробка презентації | 09.05-11.05.2025 | Виконано |
| 8 | Передзахист | 27.05-29.05.2025 | Виконано |
| 9 | Здача в деканат | 30.05-05.06.2025 | Виконано |

Здобувач вищої освіти _____ Владислав КРАСАВІН
*(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Сергій КОРОТКОВ
*(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

# РЕФЕРАТ

Кваліфікаційна робота: 52 стор., 21 рис., 17 джерел.

Мета: дослідити та реалізувати інтерактивний інформаційний вебпортал із візуалізацією картографічних даних на основі безкоштовних хмарних сервісів.

Об’єкт дослідження: хмарні сервіси для розгортання IT-систем.

Предмет дослідження: побудова інтерактивного вебпорталу з інтегрованою картографією.

Короткий зміст: розглянуто технології хмарних платформ, веброзробки та інтеграції картографічних даних. Реалізовано MVP-проект, що дозволяє користувачу переглядати карти, фільтрувати об’єкти, здійснювати навігацію.

Систему протестовано на відповідність функціональним вимогам.

Ключові слова: інформаційний портал, картографія, хмарні сервіси, веброзробка, Leaflet, Firebase.

**ЗМІСТ**

ВСТУП…….....
РОЗДІЛ1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ……..10
1.1 Поняття інформаційного порталу……..……....10
1.2 Системи інтеграції геоданих у веброзробці……..13
1.3 Огляд безкоштовних хмарних сервісів для розміщення порталу…….....17
1.4 Сучасні приклади реалізації картографічних проєктів……19
РОЗДІЛ 2. ВИБІР ІНСТРУМЕНТІВ ТА ПРОЄКТУВАННЯ АРХІТЕКТУРИ….25
2.1 Вибір стеку технологій……...……25
2.2 Побудова архітектури програмного забезпечення……27
2.3 Розробка базової структури вебпорталу……29
2.4 Організація бази даних і хмарного середовища…… 33
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ІНТЕГРАЦІЯ КАРТОГРАФІЇ…….....38
3.1 Підключення API картографії (Leaflet, OSM)…….……..38
3.2 Створення інтерфейсу для взаємодії з картою…...……..…….41
3.3 Візуалізація об’єктів на карті…….44
3.4 Тестування функціоналу……..47
3.5 Розгортання геопорталу у хмарному середовищі…….50
ВИСНОВКИ……53
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ…….……54
Додаток А.ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)……..……56

# ВСТУП

У сучасному цифровому середовищі інформаційні вебпортали є ключовими інструментами для ефективного збору, представлення та обміну даними між користувачами, державними установами, бізнесом та іншими організаціями. Значна кількість таких порталів інтегрує візуалізацію просторових даних, що відкриває нові можливості для аналізу, прийняття рішень та взаємодії з інформацією. Картографічні сервіси стали не лише способом навігації, а й потужним інструментом для демонстрації даних на географічній основі.

З іншого боку, використання хмарних технологій дозволяє мінімізувати витрати на інфраструктуру, підвищити масштабованість та забезпечити швидкий доступ до системи з будь-якої точки світу. Поєднання картографії з хмарними сервісами створює умови для побудови сучасних, зручних та надійних інформаційних рішень, які легко впроваджуються в різних сферах – від урбаністики та логістики до туризму та екологічного моніторингу.

Актуальність теми обумовлюється зростаючим попитом на інтерактивні онлайн-системи, що відображають дані на картах, зокрема у мобільних додатках, вебпорталах та IoT-платформах. Безкоштовні картографічні сервіси (такі як OpenStreetMap, Google Maps, Mapbox) у поєднанні з хмарними платформами (Firebase, Netlify, GitHub Pages тощо) надають потужний інструментарій для реалізації таких рішень без значних фінансових витрат.

Метою цієї кваліфікаційної роботи є проєктування та створення інформаційного вебпорталу, який інтегрує картографічні дані, використовуючи безкоштовні хмарні сервіси для розміщення і зберігання даних. У роботі досліджуються існуючі технології веброзробки, хмарні архітектури та API картографічних сервісів, виконується аналіз їх переваг, обмежень та сумісності. На основі цього проводиться розробка MVP-продукту з можливістю виведення об’єктів на мапу, фільтрації, інтерактивної взаємодії з елементами інтерфейсу.

Об’єктом дослідження є хмарні сервіси та вебінтерфейси, що забезпечують інтеграцію геоданих. Предметом дослідження – процес побудови вебпорталу з

підтримкою картографічних функцій.

Структура роботи включає три основні розділи. У першому розглядаються особливості побудови інформаційних порталів та аналізуються сучасні картографічні сервіси. У другому – обґрунтовується вибір інструментів розробки та описується архітектура системи. У третьому розділі подано результати реалізації функціоналу з картографічною інтеграцією та тестування розробленої системи.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

## 1.1 Поняття інформаційного порталу

*Інформаційні портали* сьогодні відіграють ключову роль у цифровій трансформації суспільства, забезпечуючи ефективний доступ до тематично згрупованих даних, сервісів і аналітичних інструментів. На відміну від звичайних сайтів, інформаційний портал не обмежується функцією презентації контенту – він виступає як інтерактивне середовище, в якому відбувається обмін, фільтрація та уніфікація інформації для широкого або спеціалізованого кола користувачів.

У найзагальнішому вигляді, *інформаційний портал* – це багаторівнева система доступу до цифрових ресурсів, що дозволяє здійснювати навігацію між різними розділами, використовувати пошук за ключовими параметрами, фільтрувати вміст, взаємодіяти з сервісами, а також іноді – створювати або змінювати дані. Сучасні портали забезпечують не лише пасивне споживання інформації, а й активну участь користувачів у процесах генерації контенту, завантаження документів, комунікації в реальному часі, обміну аналітичними висновками тощо .

*Інформаційні портали* класифікуються за типами вмісту (тематичні, галузеві, новинні), цільовою аудиторією (громадські, корпоративні, освітні) або ступенем інтеграції з іншими системами (локальні, інтегровані, агрегуючі). Наприклад, в умовах муніципального управління інформаційні портали використовуються для збирання відгуків громадян, публікації інтерактивних карт, виводу статистичних даних про рух транспорту, стан інфраструктури або рівень забруднення повітря. Така функціональність неможлива без сучасних картографічних рішень, інтегрованих у структуру порталу.

Однією з провідних тенденцій у розвитку порталів є використання просторової інформації (геоданих), що відображається на інтерактивних картах. Ці карти не лише орієнтують користувача у фізичному просторі, але й виступають як

засіб аналізу великих масивів інформації за критерієм географічного положення. Завдяки використанню бібліотек Leaflet.js, Mapbox GL JS, Google Maps API або OpenLayers, розробники можуть додавати до порталу шари з даними, налаштовувати масштабування, кластеризацію точок, стилізацію об'єктів, пошук, фільтрацію та інтеграцію з API третіх сторін.

Технічна реалізація сучасного інформаційного порталу включає фронтенд (користувацький інтерфейс), бекенд (логіка обробки даних), базу даних, системи безпеки, а також інфраструктурну платформу, на якій портал розміщується. Наприклад, хмарні сервіси, такі як Firebase чи GitHub Pages, дозволяють безкоштовно або з мінімальними витратами розмістити портал у мережі, забезпечивши безперервну доступність, автоматичне масштабування та базову систему безпеки.

Формальна модель інформаційного порталу містить такі компоненти:

*Інтерфейс користувача* – Інтуїтивно зрозумілий набір сторінок, що дозволяють здійснювати навігацію, пошук, перегляд і фільтрацію контенту. Сучасні фреймворки на JavaScript (React, Vue.js) дають змогу реалізовувати SPA-архітектуру (Single Page Application), що значно покращує взаємодію з користувачем;

*Серверна логіка* – Обробляє запити, взаємодіє з базами даних, забезпечує автентифікацію та маршрутизацію. Часто реалізується через Node.js, Express або інші подібні платформи;

*Сховище даних* – У випадку інформаційних порталів з картографією це можуть бути як SQL-бази, так і NoSQL (наприклад, Firestore);

*Картографічний модуль* – Відповідає за візуалізацію геопросторових даних, роботу з GeoJSON-об'єктами, побудову маршрутів тощо;

*Інфраструктура хостингу та CI/CD* – Платформи на зразок Firebase Hosting або Netlify забезпечують автоматичне розгортання та підтримку сайту у продакшн-середовищі.

Унікальною рисою сучасного інформаційного порталу є інтеграційність – здатність з'єднуватися з іншими системами через API. Наприклад, до системи

можна підключити погодні сервіси, відкриті державні реєстри, бази статистики, а також сервіси збору зворотного зв’язку. У сфері «розумного міста» така інтеграція дозволяє побудувати дашборди з актуальним станом освітлення вулиць, заповненістю парковок, навантаженням на транспортні вузли.

В освітньому або академічному середовищі інформаційні портали можуть виступати платформами для доступу до навчальних матеріалів, цифрових бібліотек, реєстраційних систем, систем електронного голосування або взаємодії між студентами й викладачами. Тут важливими є модулі особистого кабінету, управління доступом до контенту, історії перегляду та підписки на тематичні розділи.

Не менш важливою є роль безпеки у побудові порталу. Враховуючи, що система містить персональні дані користувачів або потенційно чутливу інформацію (наприклад, про об'єкти інфраструктури), обов’язковими є використання HTTPS-протоколу, автентифікації, контроль прав доступу та резервне копіювання.

Ще одним визначальним критерієм є адаптивність. Умовою сучасного порталу є коректне відображення на різних типах пристроїв – від смартфонів до десктопів, що забезпечується через адаптивну верстку або прогресивні вебдодатки (PWA).

Таким чином, інформаційний портал – це складна інформаційно-технічна система, яка виходить далеко за межі звичайного вебсайту. Її функціонування потребує ретельного проєктування, дотримання принципів масштабованості, безпеки та інтеграції з геоданими. Особливо цінною стає здатність таких систем до візуалізації інформації у просторовому (картографічному) вимірі, що дозволяє користувачам не лише читати або переглядати дані, а й розуміти їх у контексті географії, логістики та містобудування.

## 1.2 Системи інтеграції геоданих у веброзробці

У сучасному цифровому середовищі, де доступ до просторової інформації став звичним елементом багатьох сервісів, особливе місце займає інтеграція геоданих у веброзробку. Візуалізація простору, подій, маршрутів, об'єктів і явищ у географічному контексті перетворює звичайні вебпортали на динамічні аналітичні платформи з великою кількістю прикладних функцій. Інформаційні портали з підтримкою геоданих дозволяють користувачу не лише бачити, але й аналізувати дані в реальному просторі – що критично важливо в таких сферах як урбаністика, безпека, транспорт, екологія, логістика, туризм тощо.

Геодані – це дані, що мають географічне положення: координати точок, лінії маршрутів, полігони регіонів, зони покриття, межі районів, тощо. Для їх цифрового представлення у вебсередовищі використовуються стандартизовані формати, зокрема GeoJSON – найзручніший для роботи з JavaScript, оскільки заснований на форматі JSON. Інші формати – KML (використовується у Google Maps), GML, CSV з координатами, TopoJSON тощо – застосовуються в залежності від цілей і сумісності з конкретними інструментами. Дані у форматі GeoJSON можна зберігати на сервері або у хмарному сховищі й завантажувати динамічно у вебдодаток. Структура таких даних дозволяє не лише відобразити точку на мапі, а й додати до неї атрибути: назву, опис, тип, колір маркера, рівень небезпеки або статус ремонту.

У веброзробці широко застосовуються бібліотеки клієнтської частини, які дозволяють інтерактивно працювати з картою в браузері. Однією з найпопулярніших є Leaflet.js – легка й гнучка бібліотека, що дозволяє підключати фонові карти (базові тайли) з таких джерел як OpenStreetMap, Mapbox, Carto та інших. Вона підтримує відображення точок, ліній, полігонів, кластеризацію, інфо-вікна, події натискання й наведення, шари та підключення плагінів, наприклад, для heatmap або геолокації. Завдяки простоті й модульності Leaflet використовується навіть у великих проєктах із відкритим кодом.

Mapbox GL JS є більш складним рішенням, побудованим на WebGL. Він дозволяє рендерити векторні шари без використання растрових тайлів, а також надає розширений інтерфейс для стилізації, анімації об’єктів, 3D-візуалізації, маршрутизації та аналізу. Його використовують для створення кастомізованих карт із унікальним дизайном, які працюють швидко навіть з великою кількістю об’єктів на одному полотні. Крім того, Mapbox підтримує інтеграцію з Mapbox Directions API для побудови маршрутів, геокодування, автоматичної побудови ізохрон та heatmap.

З боку серверної частини головним завданням є підготовка, зберігання та передача геоданих клієнту. Найпоширеніше open-source рішення – GeoServer, що дозволяє публікувати геодані за стандартами OGC: WMS (вебмапи), WFS (векторні об’єкти), WCS (растрові дані). Він підтримує джерела даних у вигляді файлів (Shapefile), баз даних (PostgreSQL + PostGIS), а також дозволяє задавати стилі відображення за допомогою SLD (Styled Layer Descriptor). Завдяки цьому розробники можуть отримувати на стороні браузера або мобільного клієнта лише необхідні шари й об'єкти, фільтруючи за атрибутами або просторовими критеріями.

Існує також можливість повністю відмовитися від серверної частини в традиційному розумінні, якщо обсяг геоданих невеликий. Для цього використовуються безкоштовні хмарні сервіси, такі як GitHub Pages або Firebase Hosting, де можна зберігати GeoJSON-файли, JavaScript-код і HTML-інтерфейс. Цей варіант підходить для навчальних, дослідницьких, некомерційних або пілотних проєктів. Наприклад, карту з актуальним станом пішохідних доріжок міста або пошкодженими об’єктами інфраструктури можна реалізувати без жодного бекенду – достатньо просто завантажити файл з об’єктами та зв'язати його з картою в Leaflet.

Потужним рішенням у комерційній і муніципальній сфері є ArcGIS Online – хмарна геоінформаційна платформа, що дозволяє не лише створювати карти, а й розробляти повноцінні додатки, ділитися даними з громадськістю, керувати рівнями доступу, працювати зі статистикою, часовими рядами, 3D-візуалізацією, інтегрувати карти у сайти або портали через iframe чи API. Цей інструмент широко

використовується державними структурами, дослідницькими інституціями та бізнесом завдяки багатому функціоналу, підтримці мобільних пристроїв, автоматичному бекапу та гнучкій системі доступу.

При реалізації інтеграції геоданих важливо враховувати не лише функціонал, а й продуктивність. Якщо портал працює з великими обсягами даних (наприклад, десятками тисяч точок або полігонів), необхідно застосовувати оптимізацію: кластеризацію, векторизацію, тайли, асинхронне завантаження. Для цього застосовуються бібліотеки з підтримкою кластерів, розбиття даних на сторінки або локації, кешування на клієнті, обмеження області відображення за масштабом.

Крім відображення, інтеграція геоданих передбачає взаємодію: зворотний зв’язок від користувачів (маркування місць на мапі), фільтри, пошук за координатами або назвою, натискання для отримання додаткової інформації, оновлення статусу об'єкта. Усе це реалізується через JavaScript, з використанням DOM-подій, AJAX-запитів або WebSocket для оновлень у реальному часі.

Таким чином, сучасна система інтеграції геоданих у веброзробці – це сукупність відкритих та комерційних інструментів, що дозволяють працювати з просторовими даними у браузері, забезпечуючи високий рівень інтерактивності, наочності та ефективності інформаційного порталу. Залежно від потреб і масштабу проєкту, можна вибрати як прості рішення на основі Leaflet та GeoJSON, так і повноцінну ГІС–платформу на основі ArcGIS або Mapbox. Основною перевагою таких систем є їх доступність, адаптивність і сумісність із сучасними технологіями веброзробки.

Таблиця 1.1 – Порівняльна таблиця систем інтеграції геоданих у веброзробці

| Назва | Тип | Призначення | Переваги | Недоліки / Обмеження |
|---|---|---|---|---|
| Leaflet.js | Бібліотека JS | Відображення інтерактивних карт, додавання шарів, точок | Легка, проста в освоєнні, багато плагінів, мобільна підтримка | Не підтримує векторну тайл-рендеризацію, обмежений 3D-функціонал |
| Mapbox GL JS | Бібліотека JS | Векторні карти з кастомізацією, анімацією та API | Підтримка WebGL, високопродуктивна візуалізація, 3D-об’єкти, власні стилі | Безкоштовна версія має обмеження, складніше у використанні |
| GeoServer | Серверна система | Видача геоданих за стандартами WMS/WFS, стилізація, фільтрація | Open-source, сумісність із PostGIS, широкі можливості налаштування | Потребує налаштування сервера, складний у розгортанні |
| ArcGIS Online | Хмарна ГІС-платформа | Хостинг, візуалізація, аналітика, публікація геоданих | Хмарне розгортання, інтеграція в портал, підтримка 3D, аналітичні функції | Комерційна ліцензія, обмеження у безкоштовному режимі |
| GeoJSON | Формат даних | Зберігання векторних об’єктів (точки, лінії, полігони) | Прозорий, сумісний із JS, зручний для REST/API | Не підходить для великих наборів даних без оптимізації |
| Firebase Hosting | Хмарний хостинг | Розміщення статичних сайтів та файлів (наприклад GeoJSON) | Безкоштовно, просто у використанні, CI/CD, інтеграція з Google Cloud | Лише для статичного контенту, немає серверної логіки |
| PostGIS | База даних | Просторове розширення до PostgreSQL, зберігання геоданих | Висока точність, підтримка SQL-запитів, інтеграція з GeoServer | Потрібні знання SQL та налаштування серверної частини |

## 1.3 Огляд безкоштовних хмарних сервісів для розміщення порталу

Вибір інфраструктури для розміщення інформаційного порталу є ключовим етапом у процесі веброзробки. У випадках, коли проєкт має обмежений бюджет, створюється як навчальне або пілотне рішення, надзвичайно актуальними стають безкоштовні хмарні сервіси. Вони дозволяють розгорнути статичні або динамічні вебресурси без необхідності налаштовувати та адмініструвати власні сервери, зберігаючи при цьому гнучкість у масштабуванні та інтеграції сторонніх сервісів.

Одним із найбільш популярних рішень у цьому сегменті є *Firebase Hosting* – платформа, що належить Google і надає інструменти для швидкого розміщення статичних вебсайтів. Firebase підтримує безкоштовний тариф, що ідеально підходить для невеликих проєктів. Основними перевагами є надзвичайно проста система розгортання (через CLI), підтримка HTTPS за замовчуванням, інтеграція з іншими сервісами Firebase (Realtime Database, Firestore, Authentication) та підтримка динамічного роутінгу без потреби у серверному бекенді. Firebase Hosting також надає можливість використовувати власний домен або отримати безкоштовний субдомен .

Ще одне поширене рішення – *GitHub Pages*, що дозволяє безкоштовно хостити статичні сайти безпосередньо з репозиторіїв GitHub. Особливо зручним є автоматичне розгортання при кожному оновленні коду, що підтримує сучасні DevOps–практики. Цей сервіс широко використовується в освітніх, наукових і громадських проєктах, адже не вимагає окремого інтерфейсу для керування – усі зміни відбуваються безпосередньо через git-коміти. GitHub Pages підтримує HTML, CSS, JavaScript, а також генератори сайтів на кшталт Jekyll, що дозволяє створювати навіть блогоподібні чи документальні портали без серверного коду.

Ще одним хмарним сервісом, що забезпечує безкоштовний хостинг статичних сайтів, є *Netlify*. Цей сервіс пропонує низку розширених функцій навіть на безкоштовному плані: автоматичне CI/CD, підтримка кастомних доменів, безкоштовні сертифікати HTTPS, форми для збору даних та навіть простий

серверний функціонал через Serverless Functions. Netlify має зручну інтеграцію з GitHub, GitLab і Bitbucket, що дозволяє розробнику отримати робочий сайт одразу після завантаження коду у репозиторій. Завдяки цьому платформа ідеально підходить для MVP та SPA-проєктів.

У деяких випадках, коли портал потребує більшого контролю над середовищем, розробники використовують безкоштовні VPS або контейнерні платформи з обмеженим ресурсом, зокрема *Render* або *Railway*. Хоча безкоштовні плани таких сервісів мають обмеження за обсягом оперативної пам'яті чи кількістю запитів, вони дозволяють запускати повноцінні серверні додатки (наприклад, на Node.js або Python), підключати бази даних, зберігати файли, а також масштабувати проєкт у разі потреби. Це відкриває можливості для реалізації порталів з бекенд-логікою, авторизацією, інтеграцією з API та обробкою форм.

Ключовим чинником у виборі хмарного сервісу для розміщення порталу є тип застосування. Якщо це повністю статичний портал з інтерактивною картою на Leaflet.js і геоданими у форматі GeoJSON, достатньо буде Firebase Hosting або GitHub Pages. Якщо ж необхідна авторизація, взаємодія з базою даних, завантаження користувацьких файлів, – доцільніше обирати Firebase з Realtime Database або Render. Для проектів, що передбачають швидке оновлення контенту та зручне CI/CD, найкращим варіантом буде Netlify або GitHub Pages у зв'язці з GitHub Actions.

Загалом, безкоштовні хмарні сервіси надають достатньо функціональних можливостей для побудови навчального, інформаційного чи соціального порталу з інтеграцією картографічних даних. Вони усувають бар'єри на старті, дозволяючи розробникам зосередитися на логіці застосунку, інтерфейсі користувача та якості інформаційного наповнення, не витрачаючи ресурси на розгортання та обслуговування серверної інфраструктури.

## 1.4 Сучасні приклади реалізації картографічних проєктів

Сфера вебкартографії на сьогодні охоплює широкий спектр рішень – від урядових геопорталів до громадських ініціатив, від логістичних систем до мобільних додатків з картографічною візуалізацією даних у реальному часі. Розвиток відкритих стандартів, геоінформаційних технологій і доступність хмарних сервісів створили передумови для реалізації багатьох масштабних та інноваційних проєктів у різних країнах, включаючи Україну.

Одним із найвідоміших прикладів є *національний геопортал публічної кадастрової карти України*, який реалізований на базі технологій Leaflet та OpenLayers і забезпечує доступ до відомостей про земельні ділянки, їх межі, кадастрові номери, цільове призначення та форму власності. Інтерфейс дозволяє здійснювати пошук за номером ділянки, координатами, адресою, а також переглядати інформацію про стан розгляду заяв і звернень. Публічність таких даних сприяє прозорості у сфері земельних відносин і контролю з боку громадськості.

У сфері міської інфраструктури прикладом може слугувати портал *Lviv Open Data*, на якому публікуються актуальні геопросторові дані міста Львова. Зокрема, реалізовано інтерактивну мапу з даними про паркувальні майданчики, комунальні об'єкти, міську мережу транспорту, укриття, заклади освіти та охорони здоров’я. Дані доступні у форматах GeoJSON, CSV, JSON і можуть бути повторно використані розробниками та аналітиками. Портал працює на основі CKAN-платформи з використанням Leaflet для відображення картографічної інформації.

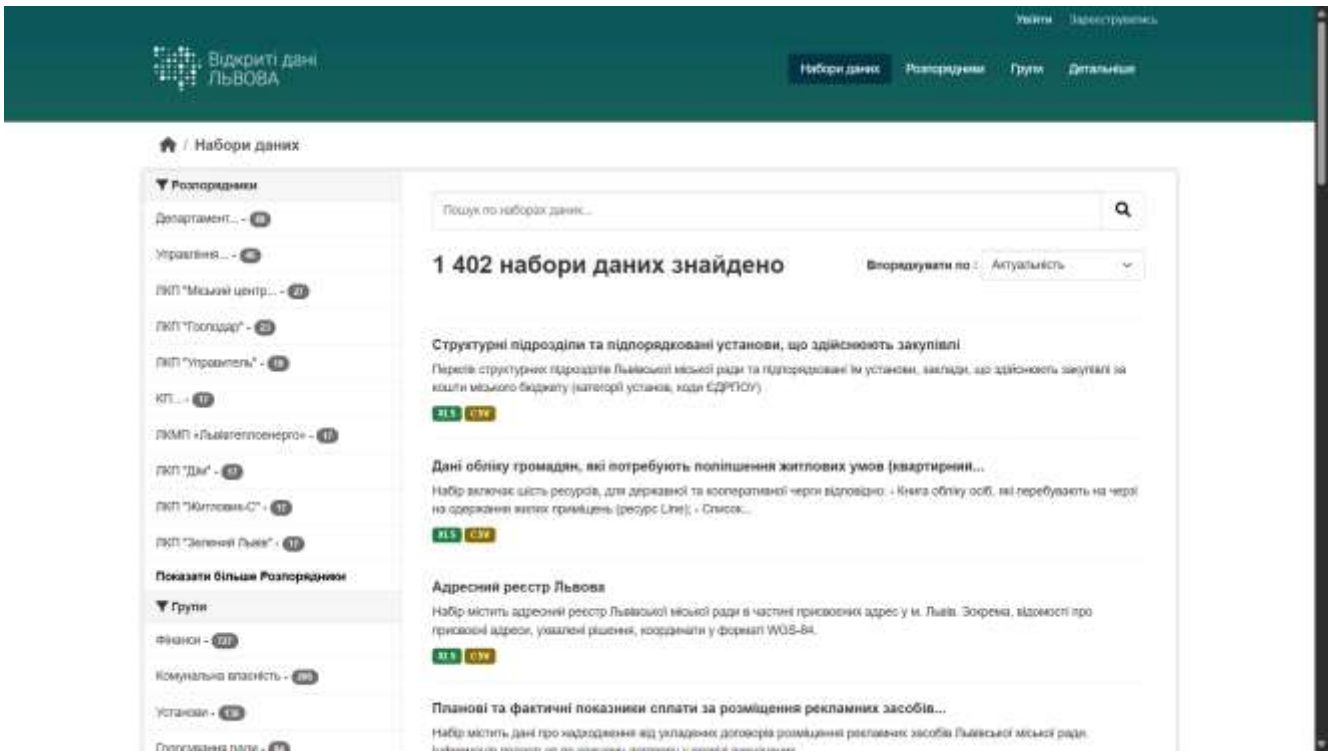



Рисунок 1.1 – Екологічна карта SaveEcoBot

Окремо варто відзначити ініціативу *SaveEcoBot* – екологічну карту, що показує в реальному часі рівень забруднення повітря у різних населених пунктах України. Проєкт збирає дані з понад 300 станцій, які використовують недорогі сенсори, і візуалізує їх на інтерактивній мапі. Користувачі можуть бачити індекс якості повітря (AQI), вміст PM2.5, PM10 та інших шкідливих речовин. Карта створена на основі Leaflet, має адаптивний інтерфейс і працює з API реального часу.

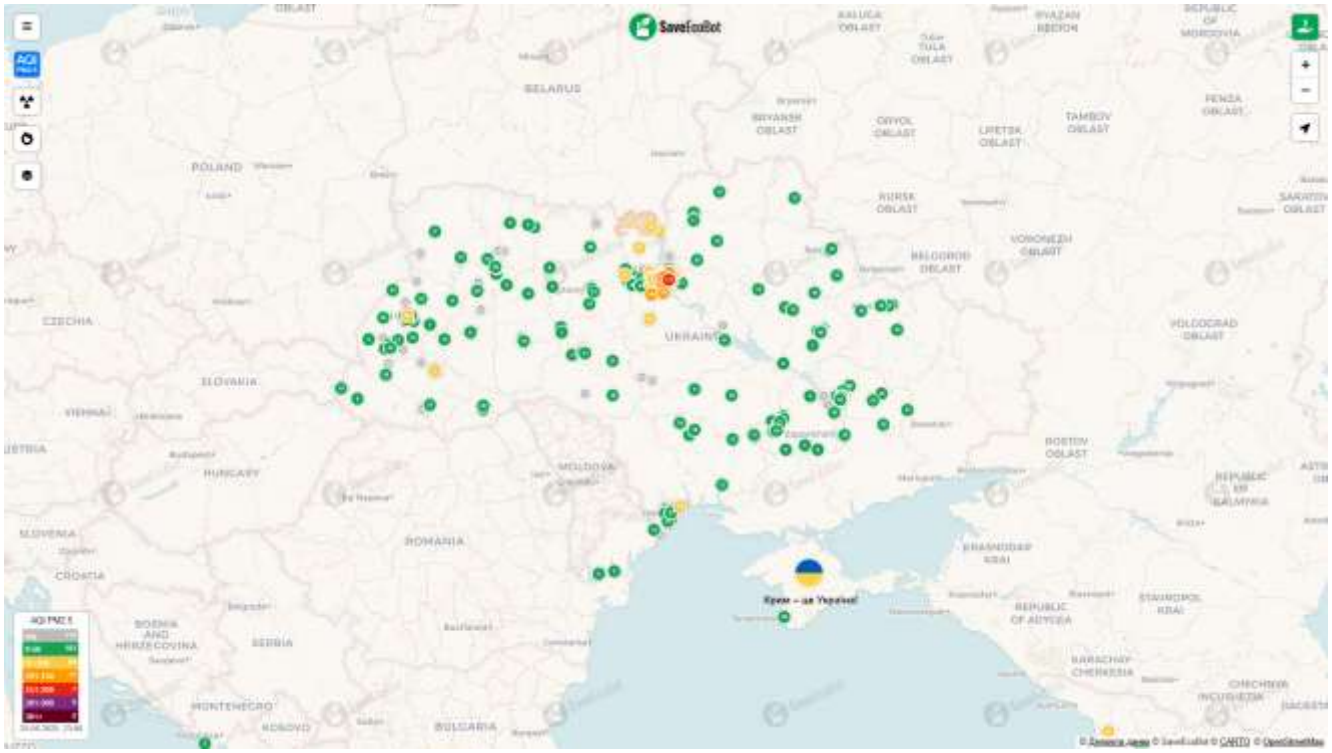



Рисунок 1.2 – Екологічна карта SaveEcoBot

Міжнародним прикладом масштабного картографічного рішення є *Humanitarian Data Exchange (HDX)* – платформа, що створена під егідою ООН і надає доступ до гуманітарних даних у вигляді візуалізацій, таблиць, мап. Багато з картографічних матеріалів інтегруються через Mapbox або ArcGIS Online, що забезпечує точність і деталізацію. HDX підтримує завантаження наборів геоданих у відкритих форматах та використовується для координації допомоги в надзвичайних ситуаціях у різних регіонах світу.




Рисунок 1.3 – Humanitarian Data Exchange (HDX)

Ще один показовий приклад – *Global Forest Watch*, інтерактивна карта вирубки лісів, що використовує супутникові знімки й алгоритми машинного навчання для виявлення змін у лісовому покриві. Користувачі можуть переглядати зменшення або приріст лісів у будь-якому регіоні світу, порівнювати історичні дані, експортувати результати для подальшого аналізу. Карта реалізована за допомогою Google Earth Engine, Mapbox GL JS і CartoDB. Такі сервіси є важливими інструментами для екологічних організацій та науковців.

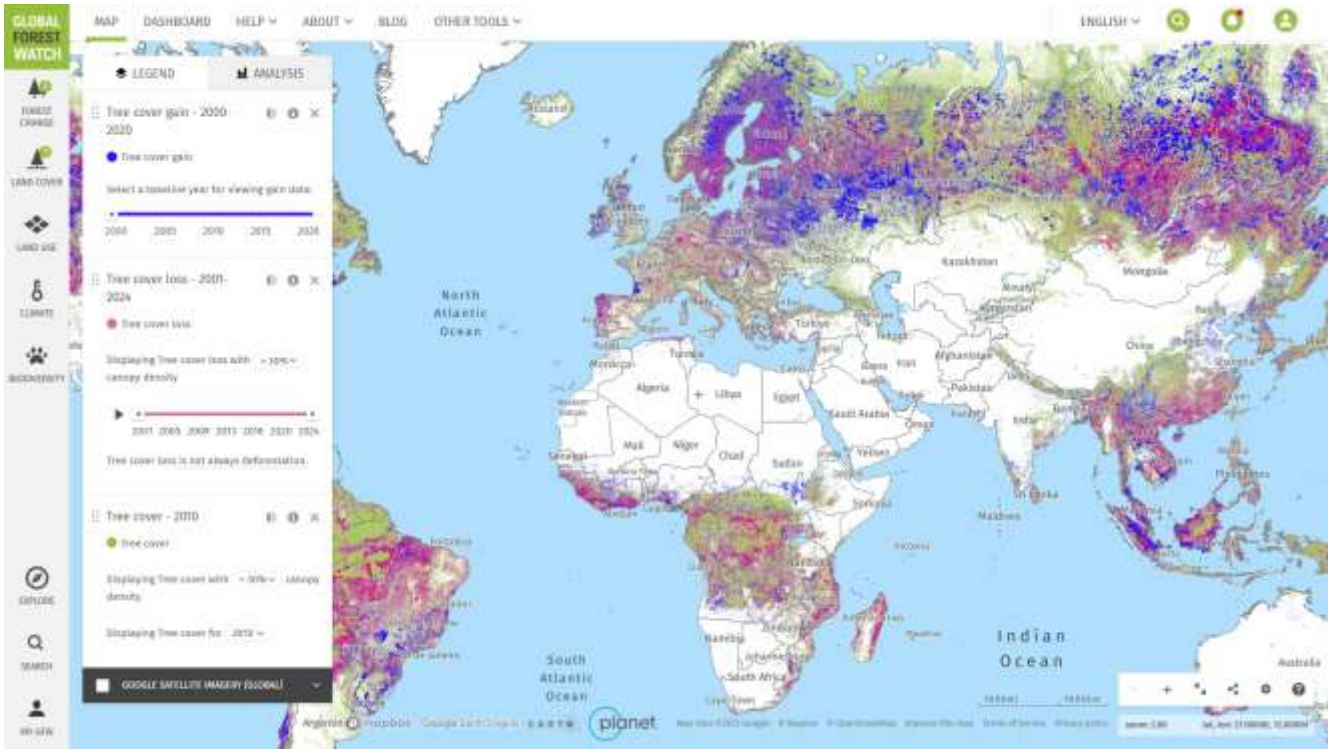



Рисунок 1.4 – Global Forest Watch

Картографічні проєкти широко використовуються також у сфері транспорту. Наприклад, у Києві реалізовано систему *EasyWay*, яка відображає розташування громадського транспорту в реальному часі. На мапі видно маршрути, розклади руху, пересування автобусів, тролейбусів і трамваїв з GPS-трекерами. Це забезпечує зручність для пасажирів і дає змогу планувати поїздки. Технічно реалізація базується на Leaflet.js і відкритому API Київпастрансу.

Варто також згадати про численні освітні та культурні мапи. Наприклад, проєкт *«Мапа пам'яті»* дозволяє побачити місця масових репресій, таборів та поховань у період СРСР. Інформація зібрана на основі архівних джерел і представлена на інтерактивній карті з фільтрацією за типом об'єкта, хронологією, регіонами. Візуалізація здійснена через Mapbox, а дані оброблялися за допомогою GIS-систем.

Загалом, сучасні картографічні проєкти не обмежуються лише відображенням об'єктів на мапі. Вони виконують функцію аналітичних інструментів, платформ для зворотного зв'язку, каналів комунікації, інтеграції даних, а також є важливою складовою цифрової трансформації у різних сферах.

Поєднання відкритих даних, геоінформаційних систем, хмарних сервісів і картографічних бібліотек дає змогу створювати гнучкі, доступні й масштабовані рішення, що відповідають вимогам часу.

# 2 ВИБІР ІНСТРУМЕНТІВ ТА ПРОЄКТУВАННЯ АРХІТЕКТУРИ

## 2.1 Вибір стеку технологій

У процесі розробки інформаційного вебпорталу з інтеграцією картографічних даних вирішальну роль відіграє правильний вибір технологічного стеку. Платформа має забезпечити одночасно інтерактивність, легкість розгортання, підтримку просторових даних та гнучке масштабування без істотних фінансових витрат. В умовах реалізації MVP (Minimum Viable Product) основний акцент робився на використанні безкоштовних, відкритих технологій, сумісних із популярними хмарними сервісами.

На рівні користувацького інтерфейсу (frontend) застосовано класичний стек веброзробки: *HTML5, CSS3 та JavaScript*. HTML5 виконує функцію основи структури вебпорталу, забезпечуючи логічну ієрархію елементів, що легко адаптується до подальшої стилізації та динаміки. CSS3 використовується для оформлення інтерфейсу та забезпечення адаптивного дизайну, який коректно відображається як на стаціонарних комп'ютерах, так і на мобільних пристроях різного формату. JavaScript, у свою чергу, забезпечує логіку взаємодії з елементами інтерфейсу, роботу з картографічними даними та підключення до хмарних сервісів.

Ключовим інструментом для візуалізації просторової інформації стала бібліотека *Leaflet.js.* Це легковагове та високопродуктивне рішення з відкритим кодом, яке ідеально підходить для створення інтерактивних карт у браузері. Leaflet підтримує інтеграцію з відкритими джерелами картографії, зокрема OpenStreetMap, що дозволяє створювати мапи без обмежень комерційних ліцензій. Перевагами вибору Leaflet є простота впровадження, велика кількість плагінів, активна спільнота підтримки, а також висока сумісність з мобільними пристроями. Особливо важливою є його здатність до роботи з форматом *GeoJSON*, що є стандартом для подання геоданих у вебсередовищі.

Додатково було розглянуто інтеграцію *Google Maps API*, який забезпечує розширені функції маршрутизації, геокодування та деталізованих мап. Проте, враховуючи обмеження безкоштовного тарифу та бажання уникнути прив'язки до платних сервісів, основний акцент у проєкті зроблено саме на Leaflet. Це дозволяє зберегти незалежність від комерційних платформ та забезпечити повний контроль над даними.

Зберігання і обробка даних здійснюються у хмарному середовищі *Firebase,* що надає вбудовану *NoSQL-базу даних Firestore*. Firestore дозволяє зберігати просторові об'єкти у вигляді структурованих JSON-документів, де кожен документ містить координати, назву, опис та тип об'єкта. Такий підхід ідеально підходить для роботи з геопросторовими даними, які відображаються на мапі. Перевагою Firestore є підтримка асинхронного доступу до даних, можливість масштабування у реальному часі, а також інтеграція з JavaScript SDK без потреби у серверному API.

Для розміщення самого порталу використано *Firebase Hosting*, який забезпечує безпечне HTTPS-з'єднання, високу швидкість завантаження, автоматичну оптимізацію контенту та підтримку CI/CD процесів. Як альтернатива або доповнення використовувався також *GitHub Pages* – платформа для безкоштовного хостингу статичних сайтів, яка тісно інтегрується з системами контролю версій та автоматизацією розгортання.

Таким чином, вибір стеку технологій у цьому проєкті був зумовлений такими критеріями: доступність, відкритість, легкість в освоєнні, мобільність, можливість масштабування, мінімальні вимоги до інфраструктури та підтримка картографічної інтеграції. Обрана сукупність інструментів дозволила створити повноцінний інформаційний портал, який ефективно працює у хмарному середовищі, має високу продуктивність і може бути легко розширений у майбутньому за рахунок додаткових модулів або API.

## 2.2 Побудова архітектури програмного забезпечення

Розробка інформаційного порталу передбачає побудову архітектури, яка дозволить ефективно реалізувати користувацький інтерфейс, зберігати географічні дані, відображати карту та забезпечувати швидкий доступ до інформації. Основними елементами архітектури є клієнтська частина, база даних і сервіси, що розміщуються в хмарному середовищі.

У рамках цієї роботи побудована архітектура односторінкового додатку (SPA), який завантажується з хостингу Firebase або GitHub Pages і далі взаємодіє з картографічним API та базою даних через асинхронні запити.

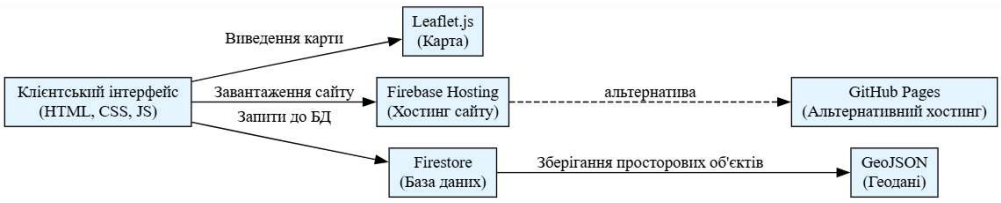



Рисунок 2.1 – Архітектура клієнт-серверної взаємодії

*Клієнтська частина* реалізується засобами HTML5, CSS3 та JavaScript. Вона відповідає за побудову інтерфейсу, завантаження карти, керування маркерами, фільтрацію та виведення даних користувачу.

*Картографічний модуль* базується на бібліотеці Leaflet, яка взаємодіє з відкритими джерелами мап, такими як OpenStreetMap. Інтерактивна карта дозволяє масштабувати область перегляду, відображати об’єкти з бази даних, створювати кастомізовані маркери та виводити інформаційні вікна.

*База даних* реалізована на Firestore (NoSQL) – хмарному сервісі, що дозволяє зберігати документи у вигляді структурованих об’єктів JSON. Кожен об’єкт містить координати (latitude, longitude), назву, опис, тип об’єкта, а також додаткові властивості. Для взаємодії з даними використовується JavaScript SDK Firebase.

*Хостинг* реалізовано за допомогою Firebase Hosting або GitHub Pages. Обидва сервіси надають можливість безкоштовного розміщення статичних сайтів, які не

потребують серверної обробки. Firebase Hosting забезпечує також захищене з'єднання через HTTPS, що підвищує безпеку взаємодії.

Таблиця 2.1 – Порівняльна характеристика основних інструментів

| Інструмент | Призначення | Переваги | Обмеження |
|---|---|---|---|
| Leaflet.js | Відображення інтерактивної карти | Легкий, безкоштовний, підтримує OSM, простий у використанні | Не підтримує 3D, обмежена візуалізація |
| Google Maps API | Платна картографічна бібліотека | Багата функціональність, маршрути, геокодування | Обмежений безкоштовний тариф |
| Mapbox GL JS | Векторна мапа на базі WebGL | Стильна візуалізація, 3D підтримка | Складність налаштування, ліміти |
| Firebase Hosting | Хостинг вебпорталу | HTTPS, швидке розгортання, CI/CD | Обмеження на розмір проєкту |
| GitHub Pages | Безкоштовний хостинг статичних сторінок | Простота, інтеграція з Git | Лише статичний контент |
| Firestore (Firebase) | Хмарна NoSQL-база даних | JSON-структура, онлайн-запити, інтеграція з JS | Обмеження на обсяг зберігання |

Таким чином, побудована архітектура дозволяє забезпечити автономність клієнтської частини, мінімізувати серверні ресурси, зберігати дані в хмарі, а також відображати карту та об'єкти у реальному часі. Це забезпечує масштабованість та високу доступність порталу при мінімальних витратах на розгортання.

## 2.3 Розробка базової структури вебпорталу

Проєктування структури вебпорталу – це не лише вибір візуального компонування інтерфейсу, а передусім визначення логіки взаємодії користувача з системою, способів доступу до картографічної інформації та механізмів обробки даних у режимі реального часу. Основна ціль – створити вебдодаток, який є інтуїтивно зрозумілим, адаптивним, технологічно простим у реалізації та водночас достатньо функціональним для подальшого розширення.

Архітектурно проєкт базується на концепції *односторінкового застосунку (SPA)*, що означає: при першому завантаженні вся логіка та компоненти клієнтської частини передаються браузеру, після чого подальша взаємодія з мапою, базою даних чи користувацьким інтерфейсом відбувається *без повного перезавантаження сторінки*. Такий підхід забезпечує високу швидкодію, знижує навантаження на сервери, а також дозволяє зручно реалізовувати інтерактивність через JavaScript.

Центральним елементом інтерфейсу виступає *інтерактивна карта*, побудована за допомогою бібліотеки Leaflet.js. Вона займає основну площу екрану і виконує функцію не лише візуального представлення просторових об’єктів, а й навігаційного інструменту. Усі взаємодії – пошук, вибір об’єкта, перегляд деталей – відбуваються саме через карту, що виступає ядром логіки порталу.

Кожен об’єкт, що зберігається в базі даних, має *географічні координати* та низку атрибутів: назву, опис, тип, дату оновлення тощо. Ці дані структуруються у форматі *GeoJSON* і автоматично трансформуються у *візуальні маркери* на мапі. При натисканні на маркер відкривається спливаюче вікно з детальною інформацією – що створює ефект взаємодії з живими об’єктами в реальному просторі.

Окрім карти, користувач має доступ до *панелі керування*, яка може бути реалізована у вигляді бічного або верхнього меню. Вона надає засоби фільтрації даних за категоріями, виконує пошук об’єктів, перемикає тематичні шари або

оновлює вміст карти. Основний функціонал панелі полягає у тому, щоб дати змогу швидко знаходити потрібну інформацію без перевантаження візуального простору.

Особливу увагу в структурі порталу приділено *адаптивності*. Завдяки використанню CSS3 із media-запитами, вебпортал однаково коректно відображається на широких моніторах, планшетах і мобільних пристроях. Усі елементи інтерфейсу масштабуються, а навігація залишається доступною при будь-якій роздільній здатності екрану. Це критично важливо, зважаючи на те, що велика частина користувачів взаємодіє з подібними порталами саме з мобільних пристроїв.

Внутрішня структура вебпорталу реалізована на основі модульного підходу. Кожен функціональний блок (від завантаження карти до обробки подій користувача) відповідає окремому логічному модулю, що дозволяє легко масштабувати або оновлювати систему в майбутньому. Усього реалізовано чотири ключові модулі: *мапи, даних, інтерактивності та панелі керування*.

Модуль *мапи* ініціалізує Leaflet-карту, підключає шари з відкритих джерел (OpenStreetMap), обробляє параметри виводу тайлів, дозволяє масштабування, переміщення карти та перехід у певні координати. Саме тут створюється контейнер для всіх маркерів, полігонів та ліній, що завантажуються із бази.

Модуль *даних* відповідає за отримання об'єктів із бази Firestore. Дані завантажуються у форматі GeoJSON, обробляються асинхронно за допомогою JavaScript та одразу передаються у візуальний компонент карти. Для покращення швидкодії використовуються кешування в сесії браузера, а також оптимізація рендерингу великих масивів точок через групування (кластеризацію).

Модуль *інтерактивності* обробляє всі дії користувача: натискання на маркер, відкриття інфо-вікна, натискання на кнопки керування, використання фільтрів тощо. Усі події обробляються через JavaScript API Leaflet, а також за допомогою DOM-елементів.

Модуль *панелі керування* реалізує інтерфейсну логіку: побудову фільтрів, оновлення карти, взаємодію з параметрами запитів. Залежно від вибраного типу

фільтрації, система в режимі реального часу оновлює відображення об'єктів на мапі без перезавантаження сторінки.




Рисунок 2.2 – Схема модульної структури вебпорталу

Для зручності узагальнимо функціональні ролі модулів у таблиці:

Таблиця 2.2 – Модульна структура вебпорталу

| Назва модуля | Функціональність |
|---|---|
| Модуль мапи | Ініціалізація Leaflet, підключення тайлів, виведення об'єктів, управління шаром |
| Модуль даних | Запит до Firestore, обробка GeoJSON, передача даних у карту |
| Модуль інтерактивності | Обробка кліків, відкриття інфо-вікон, реагування на дії користувача |
| Модуль керування | Побудова фільтрів, пошук, оновлення вмісту, перемикання тематичних шарів |




Рисунок 2.3 – Інтерфейсна схема користувацьких сценаріїв

Дані оновлюються в реальному часі завдяки тому, що Firestore підтримує live-підписку. Це дозволяє, за потреби, реалізувати відображення нових об'єктів на мапі одразу після їх внесення в базу – без перезапуску сторінки. Така функція може бути надзвичайно корисною для громадських ініціатив, наприклад, моніторингу аварій або об’єктів інфраструктури.

З технічної точки зору, важливо підкреслити, що всі дії відбуваються *на стороні клієнта*. Це дозволяє зменшити обсяг серверних ресурсів до мінімуму, забезпечити швидке завантаження інтерфейсу та створити стабільний досвід використання навіть на слабких пристроях. Саме такий підхід робить запропоновану структуру універсальною для широкого спектру задач – від муніципальних систем до освітніх чи екологічних порталів.

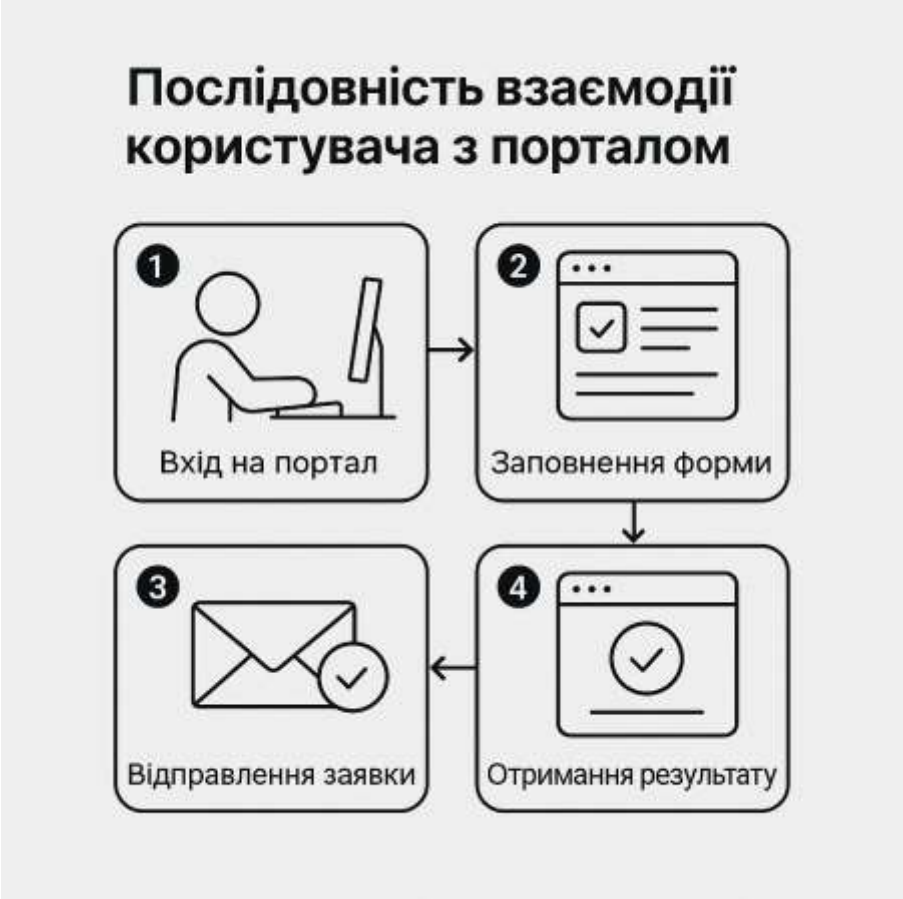



Рисунок 2.4 – Послідовність взаємодії користувача з порталом

У підсумку, створена структура порталу забезпечує повноцінну роботу з геоданими, високу продуктивність і масштабованість, що дозволяє реалізовувати як базовий функціонал, так і перспективу розширення у більш складні картографічні та аналітичні системи.

## 2.4 Організація бази даних і хмарного середовища

Ефективне зберігання, структуризація та подальша інтеграція просторових даних є невід’ємною складовою побудови сучасного вебпорталу з картографічними функціями. Враховуючи архітектурні особливості реалізованої інформаційної системи, для зберігання географічних об'єктів обрано формат GeoJSON – гнучкий,

універсальний і стандартизований спосіб представлення геоданих у веборієнтованих застосунках.

Формат GeoJSON забезпечує збереження геометричних характеристик об’єктів (координати точок, ліній, полігонів) у поєднанні з їх описовими атрибутами (назва, тип, статус, категорія тощо). На відміну від традиційних реляційних баз даних, дані у GeoJSON представлені у вигляді структури JavaScript Object Notation, що дозволяє обійтися без серверної частини та забезпечити повну інтеграцію з клієнтським інтерфейсом.

У процесі проєктування було розроблено концепцію логічної організації бази даних, яка передбачає тематичне розділення просторової інформації на окремі шари. Кожен тип об’єктів – природні території, інфраструктурні елементи, історико-культурні об'єкти, тощо – представлено окремим файлом GeoJSON. Такий підхід дозволяє легко оновлювати окремі набори даних без необхідності перезавантаження усієї системи, а також забезпечує модульність, що є особливо важливим для подальшої масштабованості проєкту.

Зберігання даних у формі GeoJSON дозволяє:

– забезпечити оперативне завантаження об’єктів у клієнтському браузері;

– реалізувати фільтрацію, кластеризацію та інтерактивну взаємодію з кожним елементом карти;

– знизити вимоги до апаратних ресурсів та зменшити навантаження на канал зв’язку.

Завдяки підтримці даного формату бібліотекою Leaflet.js, інтеграція просторової інформації у карту відбувається автоматизовано, без проміжних перетворень чи конвертацій. Дані зчитуються безпосередньо з відповідного шляху у файловій структурі порталу, після чого обробляються клієнтським скриптом та виводяться у вигляді маркерів, полігонів або підписів.

Для спрощення адміністрування, файли GeoJSON зберігаються у підкаталозі структури сайту, що дозволяє швидко вносити зміни навіть без доступу до програмного коду. Усі шари можуть оновлюватися вручну або автоматизовано, за

умови дотримання стандартів структури формату. Додатково, така модель відкриває можливості для інтеграції з відкритими державними реєстрами, геопорталами та іншими джерелами просторової інформації.

Варто зазначити, що GeoJSON є не лише технічним рішенням, а й інструментом підтримки принципів відкритих даних, що дозволяє у перспективі здійснити публічну візуалізацію або повторне використання інформації у сторонніх інформаційних системах.

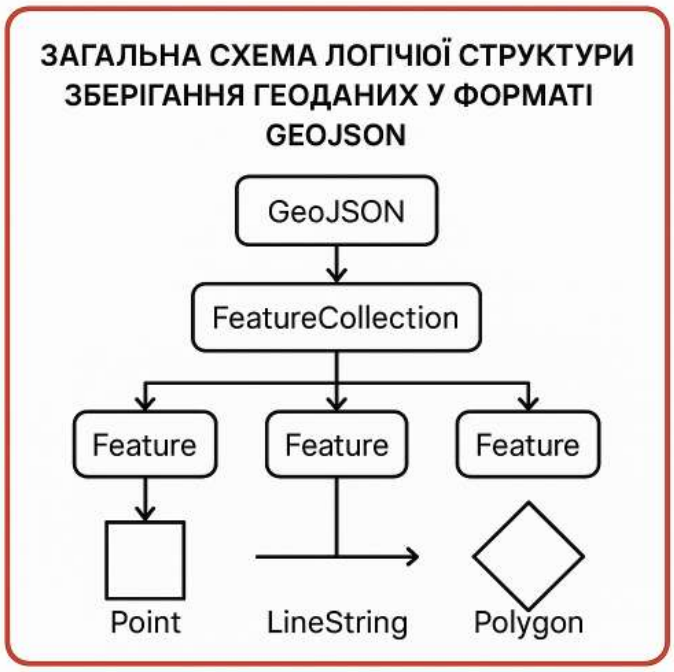



Рисунок 2.5 – Загальна схема логічної структури зберігання геоданих у форматі GeoJSON

Загалом, запропонована модель побудови бази даних забезпечує дотримання сучасних стандартів розробки вебГІС, підвищує прозорість і гнучкість системи, а також повністю відповідає технічним і функціональним вимогам поставленого завдання.

Сучасна розробка вебзастосунків, особливо таких, що передбачають інтерактивну взаємодію з геопросторовими даними, неможлива без надійного та стабільного хостингового середовища. Основними вимогами до вибору хмарної платформи у рамках дипломного проєкту були: висока швидкість доступу,

мінімальні витрати на підтримку, простота розгортання, наявність захищеного з'єднання, а також можливість масштабування у разі розширення функціональності.

Зважаючи на ці критерії, у якості хостингу для реалізованого вебпорталу було обрано *Firebase Hosting* – хмарну платформу від компанії Google, яка спеціалізується на обслуговуванні статичних і односторінкових застосунків. Це рішення повністю відповідає вимогам поставленого завдання, оскільки дозволяє безпосередньо розмістити HTML-сторінки, стилі, скрипти та зовнішні файли даних, включно з GeoJSON.

Firebase Hosting відзначається низкою ключових переваг:

- *Гарантована швидкодія* завдяки географічно розподіленій інфраструктурі та автоматичному кешуванню вмісту;
- *Захищене з'єднання HTTPS*, що забезпечує довіру користувачів і відповідність сучасним стандартам безпеки;
- *Інтегрована система розгортання* – проєкт можна оновлювати в один клік за допомогою інтерфейсу командного рядка без складної конфігурації;
- *Безкоштовний тариф* для невеликих та середніх проєктів, що робить платформу особливо привабливою для навчальних і соціальних ініціатив;
- *Підтримка гнучкого керування вмістом*, з можливістю подальшого розширення за рахунок інших сервісів Firebase (наприклад, Firestore, Functions, Authentication).

У рамках проєкту структура розгортання передбачає розміщення всіх статичних компонентів вебпорталу в окремому каталозі (public), що включає основну HTML-сторінку, стилі оформлення, сценарії взаємодії з картою, а також підкаталог з просторовими даними. Така конфігурація дозволяє максимально ізолювати логіку візуалізації від вмісту даних, що спрощує як підтримку системи, так і потенційні оновлення.

Процес первинного розгортання ресурсу у хмарі не вимагає особливих технічних навичок і виконується за допомогою офіційного інструментарію Firebase CLI. У разі зміни вмісту або додавання нових даних, оновлення ресурсу

виконується за лічені секунди, без необхідності повторного налаштування конфігурації. Цей підхід дозволяє забезпечити гнучкість у роботі з ресурсом, скорочуючи витрати часу на адміністрування.

Окрім того, Firebase Hosting гарантує стабільність роботи вебпорталу, що особливо важливо для інформаційних систем з відкритим публічним доступом. Ресурс доступний у будь-який час доби, незалежно від навантаження, а його масштабованість дозволяє оперативно реагувати на зміну кількості користувачів або розширення вмісту.




Рисунок 2.5 – Схема хмарного розміщення інформаційного порталу у середовищі Firebase

Таким чином, використання платформи Firebase як хмарного середовища для розміщення реалізованого вебпорталу дозволяє досягти оптимального балансу між функціональністю, стабільністю, безпекою та зручністю керування. Такий підхід повністю відповідає цілям проєкту, забезпечуючи просте та масштабоване розгортання інформаційної системи з картографічною складовою.

# 3 РЕАЛІЗАЦІЯ ТА ІНТЕГРАЦІЯ КАРТОГРАФІЇ

## 3.1 Підключення API картографії (Leaflet, OSM)

Однією з ключових складових реалізації інформаційного порталу є інтерактивна карта, що дозволяє відображати просторові об’єкти безпосередньо у вебінтерфейсі. Для побудови картографічного шару було використано *бібліотеку Leaflet.js* – сучасний JavaScript-інструмент із відкритим вихідним кодом, що забезпечує високий рівень інтерактивності, масштабованість, а також гнучкість у підключенні до різних картографічних сервісів. Основною картоосновою в проєкті виступає *OpenStreetMap*, що надає вільний доступ до глобальних геоданих.

Підключення Leaflet реалізується через додавання посилань на офіційні стилі та скрипти бібліотеки за допомогою CDN у заголовкову частину HTML-документу. Після цього ініціалізується об'єкт карти, вказуються координати центру та масштаб, а також додається базовий шар тайлів OSM, який відповідає за візуальну основу.

Створення контейнера  для карти із фіксованим розміром дозволяє браузеру зарезервувати простір на сторінці. Ініціалізація карти виконується за допомогою методу L.map(...), а підключення тайлів відбувається через функцію L.tileLayer(...), яка забезпечує завантаження плиток на основі URL-шаблону та визначає атрибуцію. Це дозволяє гарантувати відповідність умовам відкритої ліцензії, згідно з якою має зберігатись посилання на OpenStreetMap.

Карта відображається миттєво при завантаженні сторінки і дозволяє користувачу взаємодіяти з мапою – змінювати масштаб, рухатися в межах глобального полотна, а також переглядати об’єкти, які додаються у наступних кроках. З точки зору UX/UI, карта є основним візуальним елементом порталу, і на її основі надалі розгортатиметься вся логіка роботи з геопросторовими даними.

Крім того, Leaflet підтримує завантаження об’єктів у форматі *GeoJSON*, що є

оптимальним для веброзробки. Це дозволяє безпосередньо зчитувати координати точок, ліній або полігонів і автоматично конвертувати їх у маркери, що будуть виводитись на мапу. Кожен об’єкт може містити додаткові атрибути, зокрема: назву, опис, категорію, статус або навіть URL до фото. У рамках даної реалізації також передбачено використання кастомних popup-вікон для кожного маркера, які містять детальну інформацію з бази даних.

Такий підхід дозволяє досягти високого рівня універсальності – картографічна частина порталу може відображати будь-які об’єкти: інфраструктурні, екологічні, культурні, адміністративні тощо. У подальших підрозділах буде розглянуто механізми взаємодії користувача з картою, включно з фільтрацією даних, додаванням нових точок та обробкою подій натискання.

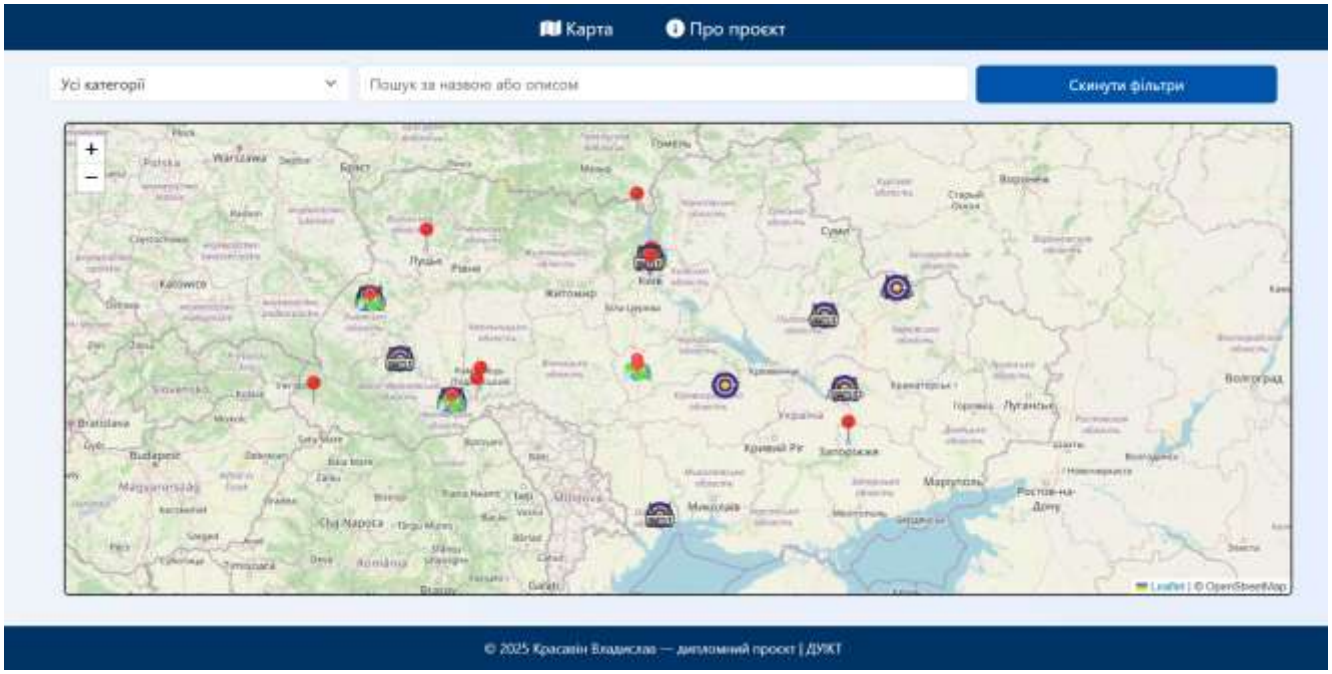



Рисунок 3.1 – Загальне підключення та ініціалізація карти Leaflet.js з базовими тайлами OpenStreetMap

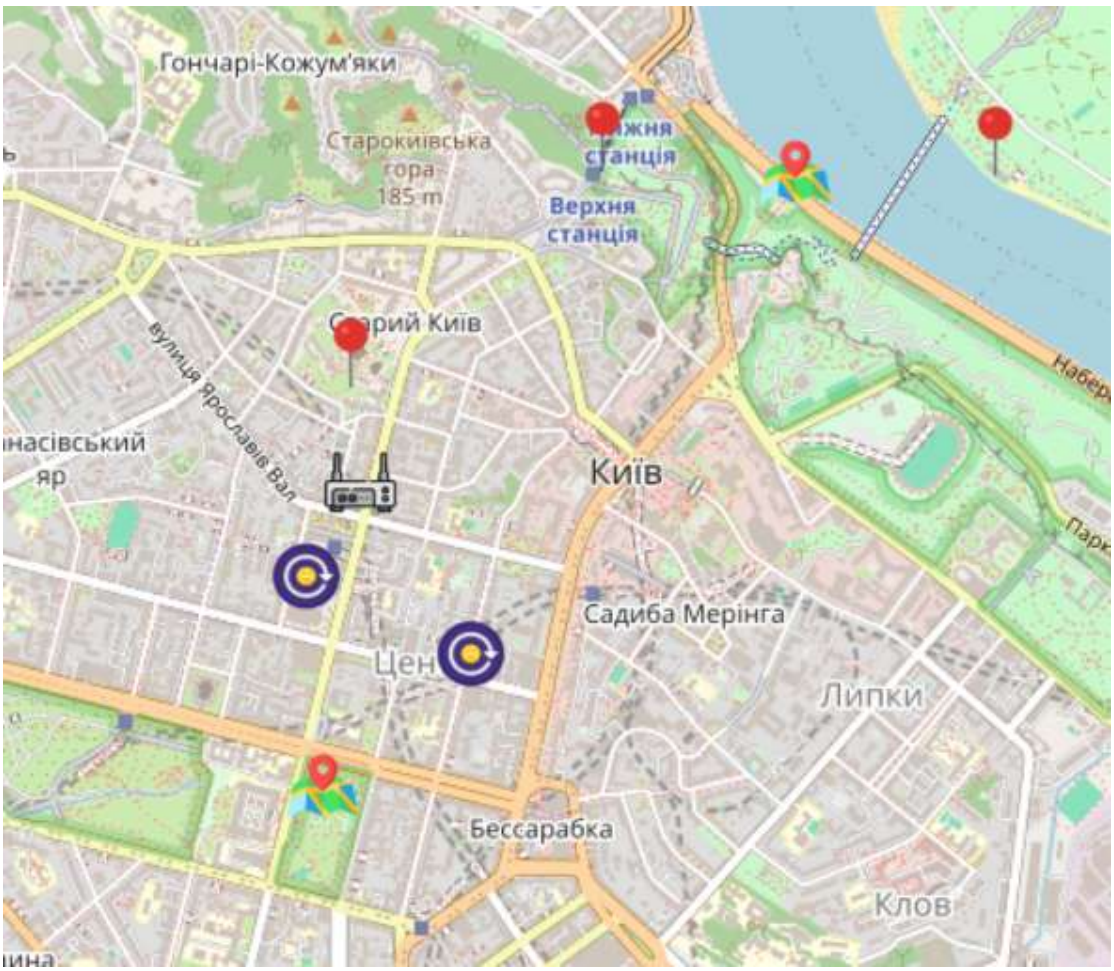



Рисунок 3.2 – Виведення маркерів із GeoJSON-файлу на карту

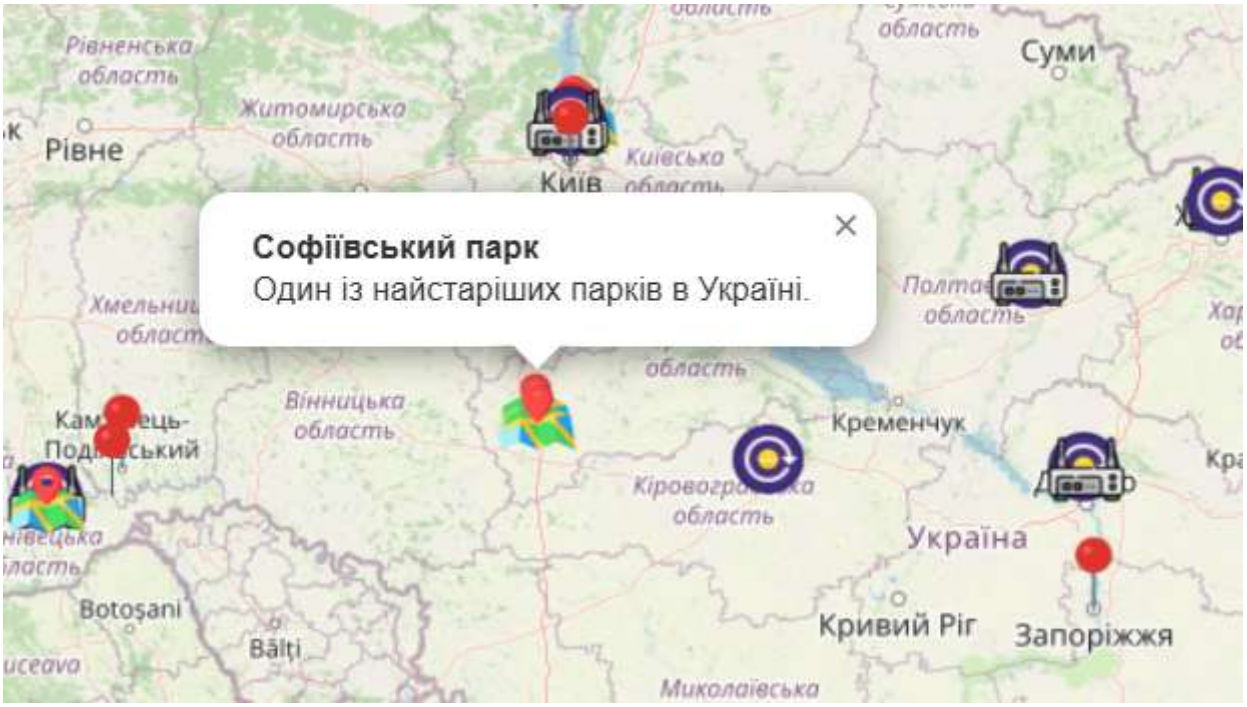



Рисунок 3.3 – Приклад інформативного спливаючого вікна (popup) при натисканні на об’єкт карти

З технічної точки зору, завдяки клієнтській реалізації карти, весь процес рендерингу виконується в браузері користувача, що дозволяє зменшити навантаження на сервер, не потребує складної серверної архітектури, і водночас забезпечує інтерактивність та динамічність вебпорталу.

### 3.2 Створення інтерфейсу для взаємодії з картою

Після реалізації базового підключення картографічного API наступним етапом стала побудова повноцінного користувацького інтерфейсу (UI), який дозволяє ефективно взаємодіяти з картою та геоданими. Основна мета цього підрозділу – організувати зручну структуру керування, яка забезпечить користувачеві простий доступ до інформації, виведеної на мапі, та дозволить фільтрувати, додавати й переглядати об’єкти.

Інтерфейс було реалізовано з урахуванням принципів *інтуїтивності, адаптивності та мінімалізму*, що дозволяє забезпечити комфортну роботу як з настільних комп’ютерів, так і з мобільних пристроїв. Основними компонентами UI стали:

– *панель керування* (sidebar або верхній control bar), яка містить фільтри, кнопки перемикання режимів, а також поле для пошуку;

– *контекстні спливаючі вікна* (popup), що відкриваються при натисканні на об’єкти на мапі;

– п*лаваючі кнопки дій*, які з’являються або активуються під час взаємодії з картою (наприклад, у режимі додавання об’єкта або перегляду детальної інформації);

– *модальні вікна* або інформаційні блоки з розширеною інформацією.

Для реалізації фільтрації передбачено окремі категорії об’єктів (типи, теми, стани), що відповідають структурі атрибутів у базі GeoJSON або Firestore. Користувач має можливість обирати потрібну категорію з випадаючого списку,

після чого карта автоматично оновлюється, відображаючи лише відповідні маркери. Ця логіка базується на оновленні шару GeoJSON у Leaflet з урахуванням фільтраційної функції.

Ключову роль у взаємодії з картою відіграють *popup-вікна*, які з'являються при кліку на об'єкт. У межах реалізації вони містять такі елементи: назву, короткий опис, дату додавання та за потреби – посилання на розширену інформацію або фото. Стилістика оформлення витримана у темній кольоровій гамі з піксельним ефектом (як вказано у стилістиці всього сайту), що забезпечує візуальну узгодженість. У разі використання Firebase Storage або інших хмарних джерел, popup може також включати прев'ю зображення або QR-код для переходу.

Окремо було реалізовано функцію перемикання в «режим вибору об'єкта», при якому змінюється вигляд курсора, з'являється анімована графіка (наприклад, GIF або SVG-елемент), а карта тимчасово блокує інші дії. Це дозволяє зручно реалізовувати сценарії додавання нових точок або взаємодії з новими шарами. Після вибору координат запускається модальне вікно для введення додаткової інформації.

Адаптивність була реалізована на основі *CSS media-запитів*. Усі елементи інтерфейсу – від панелі керування до popup-вікон – масштабуються, автоматично змінюють розміщення (наприклад, із горизонтального меню у вертикальне) при зменшенні ширини екрану. Таким чином, карта залишається центральним елементом незалежно від типу пристрою, а взаємодія з нею зберігає повний функціонал.

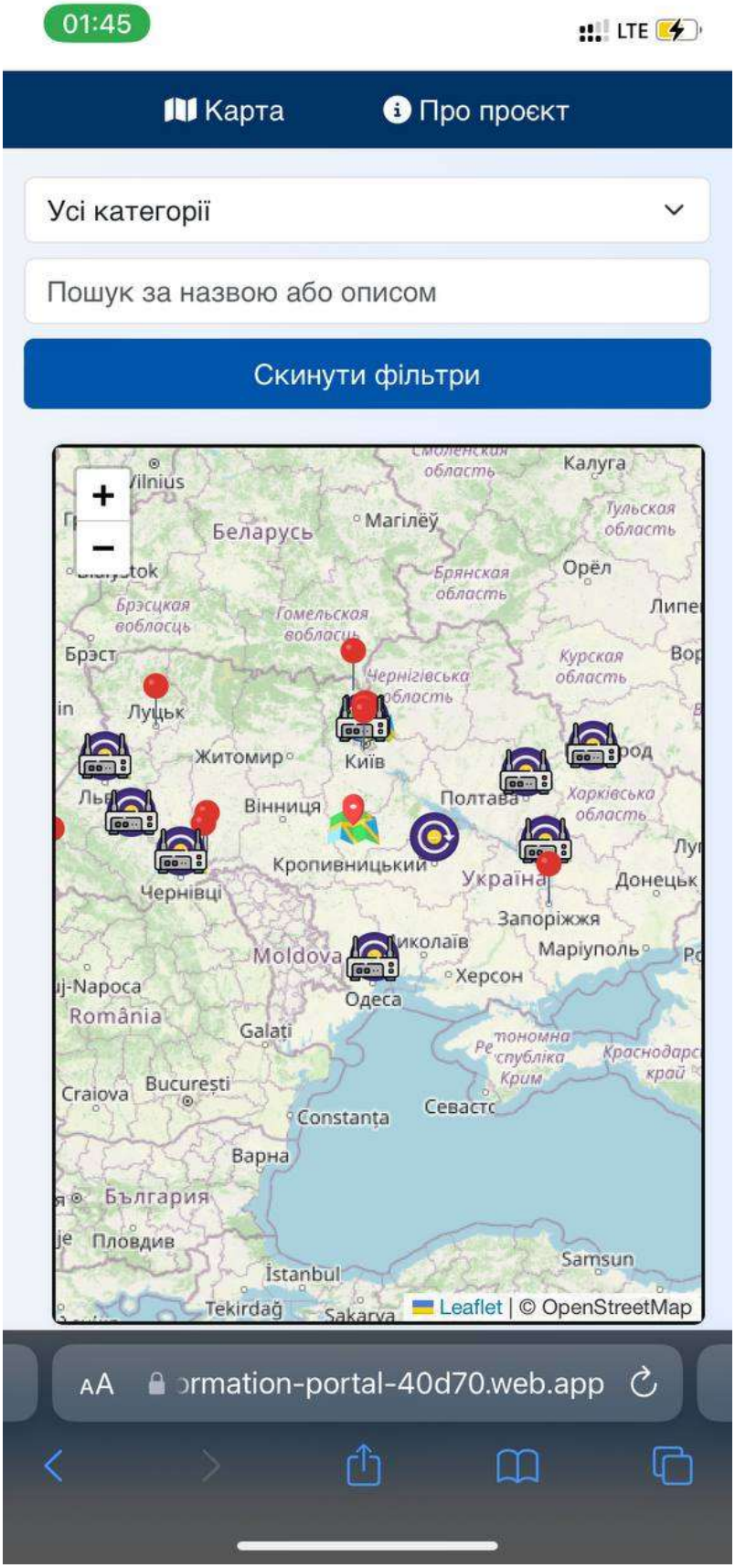



Рисунок 3.4 – Відображення мобільної версії інтерфейсу з адаптивними елементами керування

Створення такого інтерфейсу забезпечує не лише зручність роботи, а й підвищує ефективність подачі просторової інформації, що є критичним для інформаційного порталу. У наступному підрозділі 3.3 буде розглянуто безпосередню візуалізацію об’єктів, їх типізацію, стилізацію маркерів, застосування кластеризації, а також обробку подій на карті.

### 3.3 Візуалізація об’єктів на карті

Після розробки інтерфейсу для взаємодії з картою було реалізовано механізм візуалізації об’єктів, що мають просторову прив’язку. Основною метою цього етапу стало представлення об'єктів у вигляді маркерів на інтерактивній мапі з урахуванням їх категорій, описових характеристик та подальшої можливості взаємодії з ними через спливаючі вікна.

Для організації даних об’єкти були структуровані у форматі GeoJSON. Це популярний відкритий формат, який дозволяє описати просторові об’єкти (точки, лінії, полігони) разом з їхніми атрибутами у вигляді текстового JSON-файлу. Кожен об’єкт у GeoJSON представлявся як геометрична точка (Point) з координатами та полями: назва (name), опис (description), категорія (category).

Файл із даними завантажується асинхронно на стороні клієнта за допомогою функції fetch(...), що забезпечує читання зовнішнього джерела без оновлення сторінки. Після завантаження дані обробляються, і з них формуються маркери, які додаються на мапу за допомогою засобів бібліотеки Leaflet.js.

Особливу увагу було приділено категоріям об’єктів. Для зручності сприйняття та візуального розрізнення різних типів елементів (наприклад, парки, навчальні заклади, об'єкти інфраструктури тощо), кожній категорії було призначено унікальну іконку. Іконки завантажуються з локальної директорії сайту (img/) і прив’язуються до відповідних категорій через об’єкт icons, який містить асоціації типу "park" → іконка зеленого кольору, "university" → маркер з емблемою закладу освіти, "default" → базовий маркер.

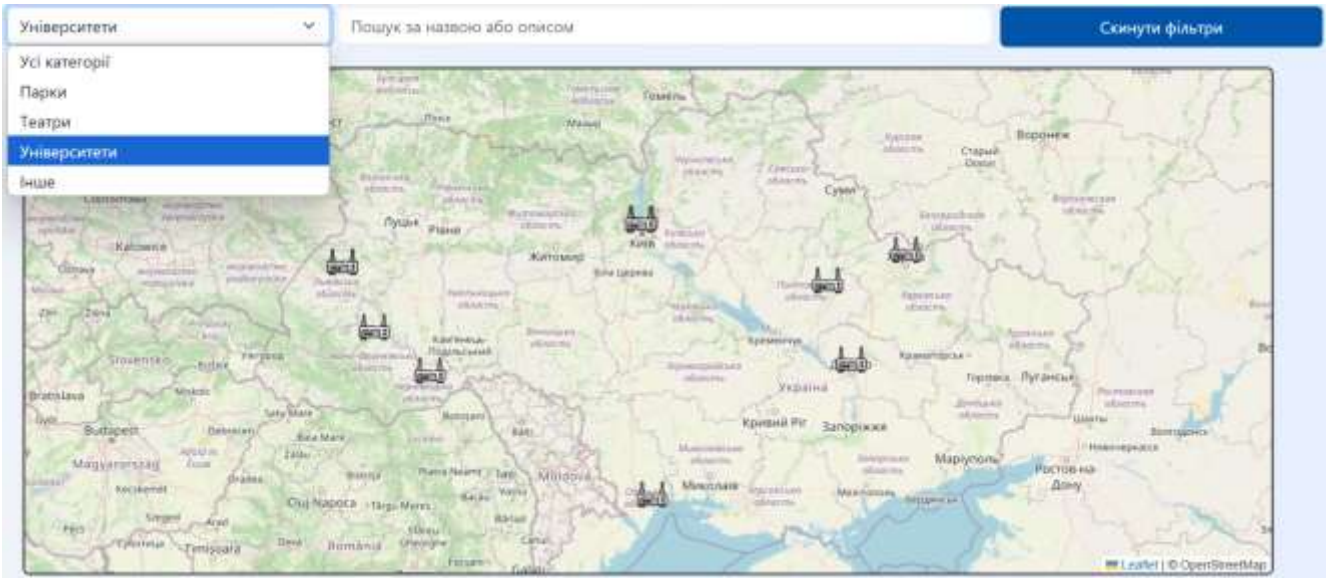



Рисунок 3.5 – Візуальне відображення маркерів на мапі згідно з категоріями об’єктів

Додатково в процесі побудови кожного маркера вказуються параметри відображення, такі як розмір, положення іконки, зміщення pop-up вікна. Таким чином забезпечується правильне розташування маркера на мапі та його візуальна стабільність при масштабуванні або переміщенні карти.

Кожен маркер супроводжується контекстним вікном (popup), яке автоматично з’являється при кліку на відповідну точку. Вміст вікна формується динамічно і включає назву об'єкта, його описову частину, а також інші допоміжні дані, якщо вони передбачені в структурі даних. Така реалізація дозволяє користувачеві швидко отримувати детальну інформацію про об’єкт без потреби переходу на інші сторінки.

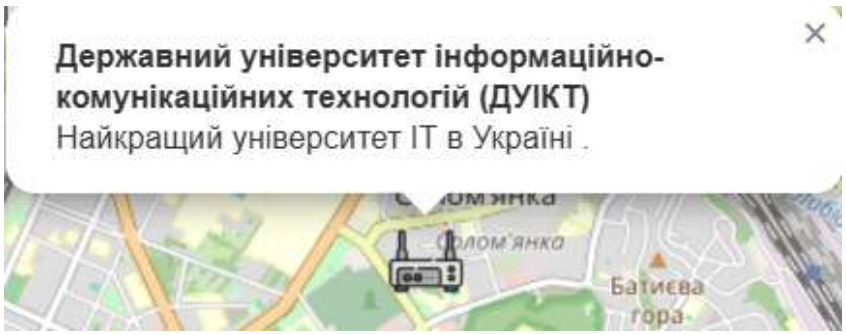



Рисунок 3.6 – Приклад відображення інформації про об’єкт у спливаючому вікні (popup)

Ще одним важливим аспектом візуалізації є забезпечення актуальності та адаптивності маркерів. Для цього було реалізовано можливість фільтрації об’єктів за категоріями, а також пошуку за ключовими словами, які можуть зустрічатися як у назві, так і в описі. При зміні параметрів фільтрації відповідні об’єкти автоматично перезавантажуються та відображаються на мапі у скоригованому складі. Це стало можливим завдяки функції updateMap() (або renderFilteredData()), яка очищає попередній шар маркерів і створює новий відповідно до вхідних критеріїв.

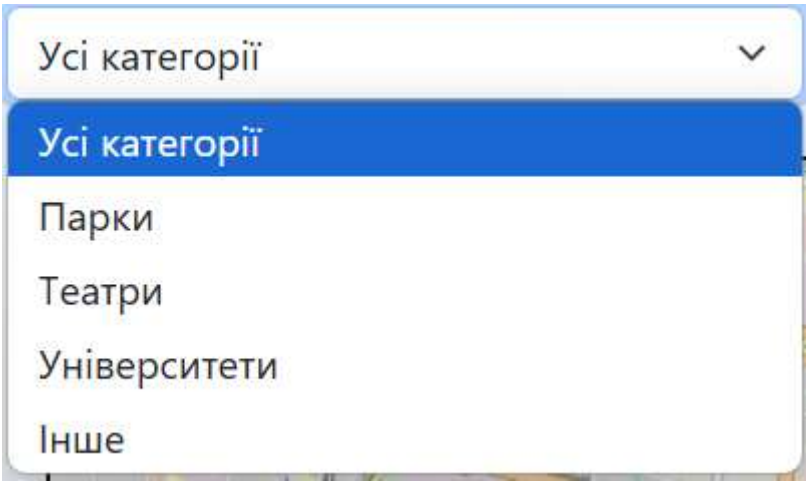



Рисунок 3.7 – Інтерфейс фільтрації та пошуку на карті

Візуалізація об’єктів у такий спосіб не лише покращує сприйняття просторової інформації, а й забезпечує високу інтерактивність у роботі з картою. Користувач може оперативно дослідити розміщення об’єктів, дізнатися їх характеристику, швидко змінити область фокусування або тип показаних даних. Це дозволяє перетворити карту на повноцінний інструмент аналітики, доступний у зручному вебінтерфейсі.

### 3.4 Розгортання геопорталу у хмарному середовищі

Завершальним етапом реалізації інформаційного геопорталу стало його розгортання у хмарній інфраструктурі, що дало змогу забезпечити його безперервну доступність, незалежну від робочої машини розробника. У сучасних підходах до створення IT-систем, особливо у сфері комп'ютерної інженерії, хмарні обчислення займають ключову роль. Вони дозволяють не лише зменшити витрати на підтримку фізичної інфраструктури, а й реалізувати автоматизовану доставку, моніторинг, масштабування та віддалене керування розгорнутими сервісами.

Для розміщення даного проєкту використано платформу Firebase, яка надає інструменти для швидкого розгортання, інтеграції з іншими сервісами Google, та забезпечення мережевої безпеки. Важливо зазначити, що в рамках проєкту використовувалась не лише можливість публікації статичних файлів, а й реалізовано повноцінний інженерний підхід до автоматизації розгортання, конфігурації та подальшого обслуговування.

Firebase Hosting є частиною більш широкої хмарної екосистеми, яка підтримує інфраструктуру як код, аналітику, авторизацію, зберігання даних, а також можливість CI/CD-інтеграції. У контексті комп'ютерної інженерії, вибір саме такого рішення дозволяє моделювати процеси, які раніше були притаманні лише серверним або DevOps-середовищам: керування версіями, контроль над середовищем, оптимізація мережевої взаємодії та безпечна взаємодія між користувачем і системою.

Розгортання геопорталу здійснювалось через офіційний CLI-інтерфейс платформи, що дозволило гнучко керувати всіма етапами: від ініціалізації до публікації. Процес супроводжувався генерацією службових конфігураційних файлів, які описують маршрутизацію, політику кешування, налаштування безпеки та базову логіку роботи із запитами. Такий підхід відповідає принципам масштабованої архітектури: код розгортання зберігається у проєкті та може бути повторно використаний або змінений без втрати працездатності системи.

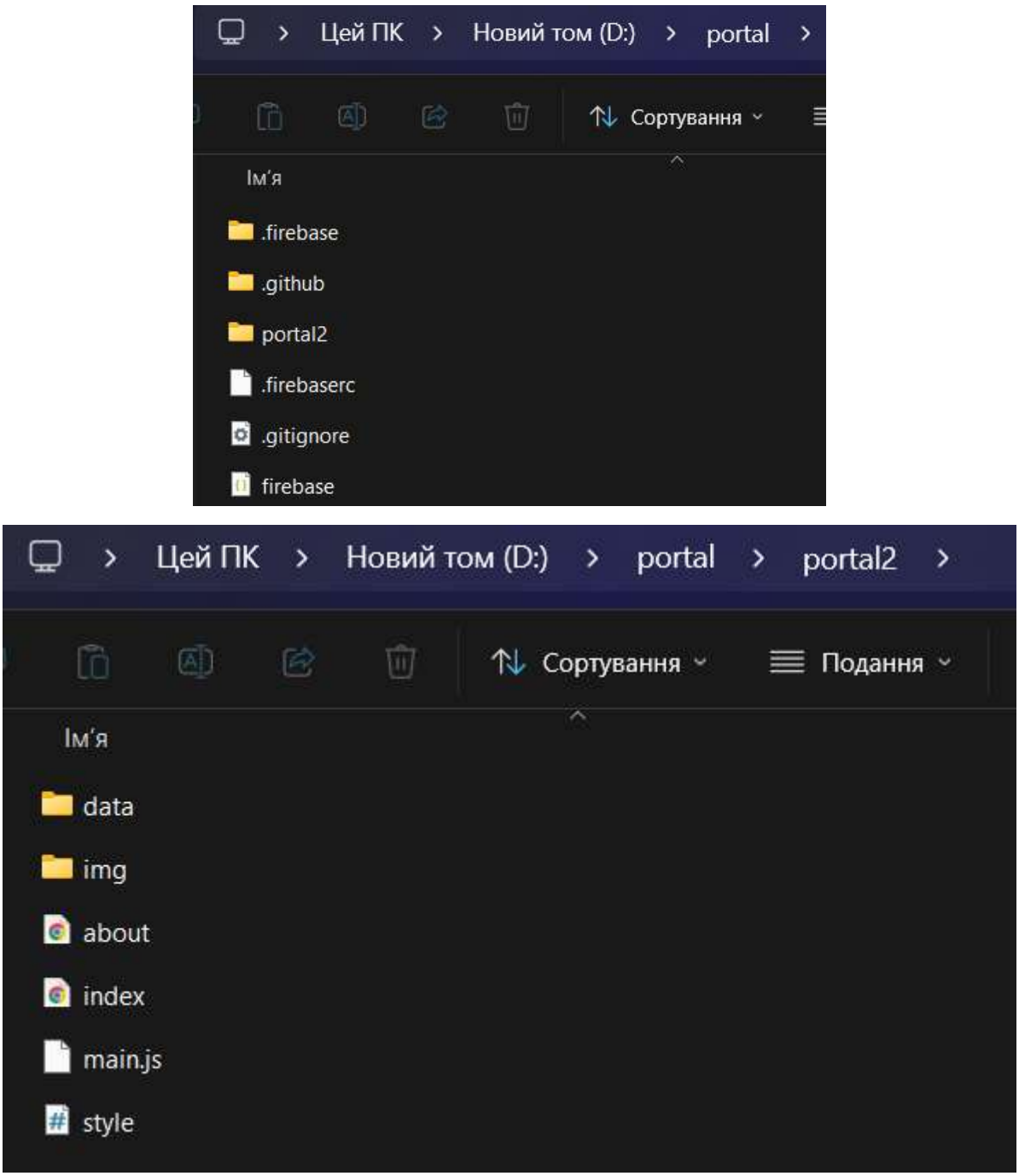



Рисунок 3.8 – Структура файлів, що були розгорнуті на Firebase

Після налаштування середовища та вибору функціональних компонентів (у цьому випадку – лише модуль хостингу) було здійснено генерацію базових конфігурацій, таких як firebase.json. Далі виконано деплой – публікацію всіх компонентів у публічному просторі. У результаті сформовано URL-адресу, за якою портал став доступним для будь-якого користувача.

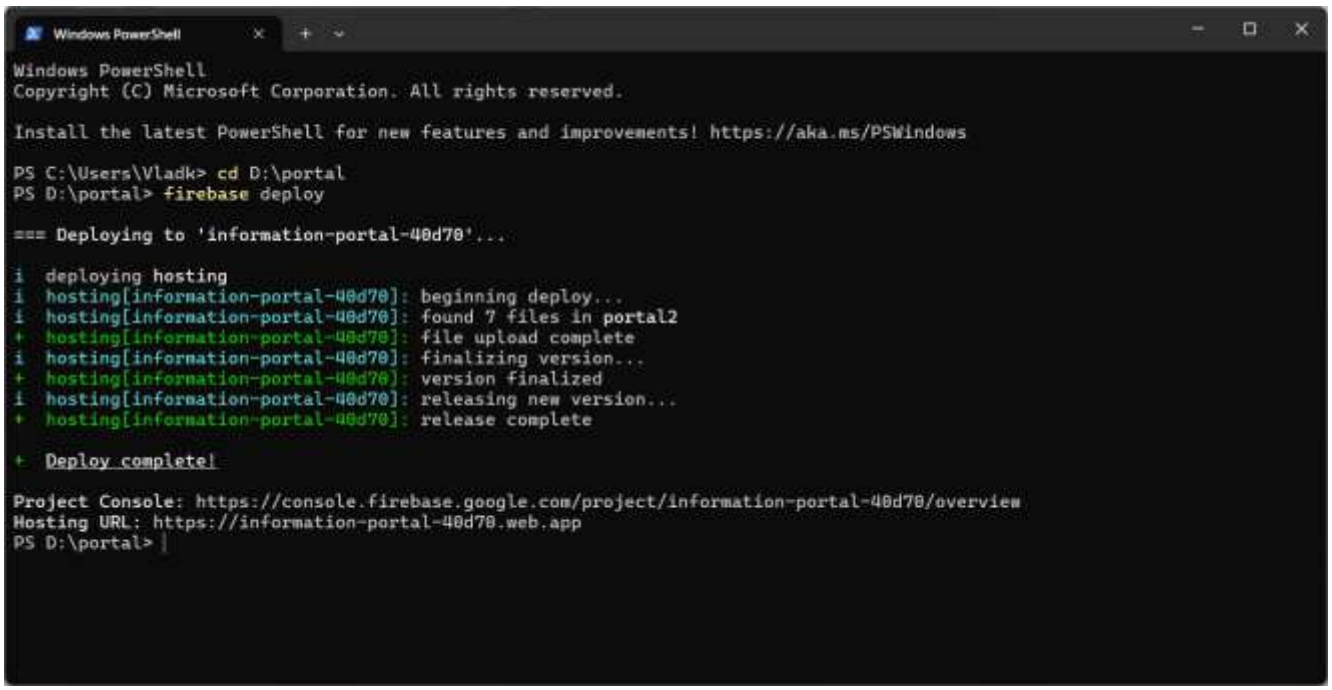



Рисунок 3.9 – Процес розгортання через Firebase CLI

З технічної точки зору, цей підхід ілюструє впровадження концепції «DevOps для фронтенду»: швидке тестування змін, їх перевірка в реальному середовищі, спрощене оновлення без зупинки сервісу та підтримка логів і моніторингу. Наявність базових метрик у консолі Firebase Hosting дозволяє оцінити ефективність та навантаження на систему, що сприяє прийняттю інженерних рішень щодо подальшої оптимізації.

Таке середовище також зручне для реалізації мобільних або десктопних додатків, що взаємодіють із геопорталом через API – оскільки Firebase підтримує CORS, HTTPS і дозволяє легко реалізовувати REST-запити.

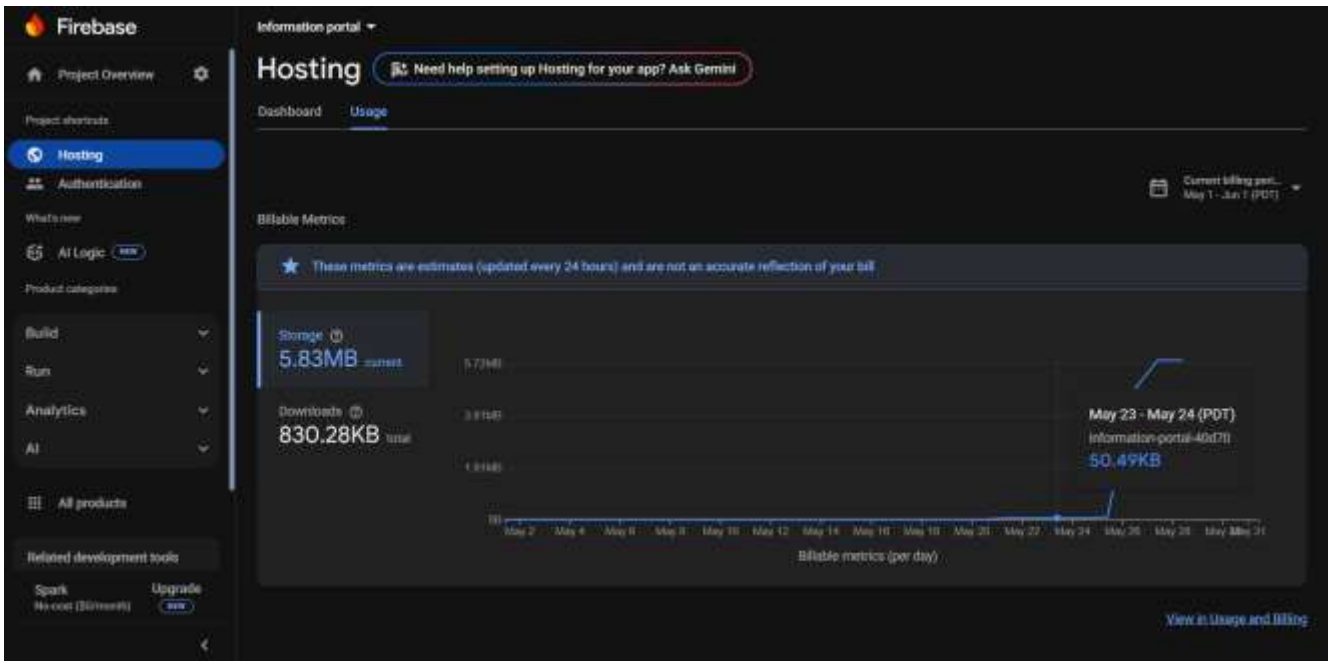



Рисунок 3.10 – Хостинг в Firebase

Загалом, розгортання у хмарному середовищі продемонструвало практичне застосування ключових понять комп'ютерної інженерії: ефективне використання хмарних обчислень, забезпечення інформаційної безпеки, масштабованості та підтримки життєвого циклу IT-системи з орієнтацією на майбутнє розширення.

### 3.5 Тестування функціоналу

Після завершення розробки основних модулів інформаційного порталу було проведено комплексне тестування функціоналу з метою перевірки коректності роботи окремих компонентів, стабільності інтерактивних елементів та відповідності реалізованої системи попередньо визначеним функціональним вимогам. Тестування проводилось у кілька етапів: модульне тестування, інтеграційне тестування, перевірка юзер-інтерфейсу (UI) та продуктивності при взаємодії з картою.

У процесі *модульного тестування* перевірялись окремі функції: завантаження карти, ініціалізація маркерів, обробка подій натискання, відкриття popup-вікон, фільтрація за категоріями. Для цього використовувались як тестові GeoJSON-файли, так і реальні дані. Помилок в обробці даних не виявлено, всі функції працювали відповідно до очікуваної логіки.

На етапі *інтеграційного тестування* було перевірено взаємодію між модулями: фільтрація динамічно змінювала відображення об’єктів, оновлення даних у Firestore відображалося на мапі в реальному часі без перезавантаження сторінки.

*UI-тестування* проводилось на ПК, планшетах і смартфонах. Завдяки використанню Bootstrap і CSS3, інтерфейс коректно відображався на будь-якому екрані. Навігація, спливаючі вікна та фільтри працювали стабільно.

*Продуктивність* перевірялась шляхом завантаження великого обсягу маркерів (>500). Завдяки кластеризації та асинхронному рендерингу, карта залишалась чутливою до дій користувача навіть при великій кількості об'єктів.

Таблиця 3.1 – Результати тестування функціональних компонентів геопорталу

| Компонент | Очікувана поведінка | Результат | Статус |
|---|---|---|---|
| Ініціалізація карти | Відображення OSM-шару | Успішно | Пройдено |
| Завантаження GeoJSON | Вивід маркерів на карту | Успішно | Пройдено |
| Popup-вікна | Вивід інформації по об’єкту | Успішно | Пройдено |
| Фільтрація об’єктів | Динамічна зміна вмісту карти | Успішно | Пройдено |
| Адаптивність | Коректне відображення на різних екранах | Успішно | Пройдено |
| Швидкодія при 50+ маркерах | Завантаження без затримок | Успішно | Пройдено |
| Хмарне оновлення (Firestore) | Автоматичне оновлення карти після зміни даних | Успішно | Пройдено |

Таким чином, результати тестування підтвердили стабільну роботу порталу відповідно до вимог, хорошу продуктивність та зручність для користувача. Система готова до розгортання або розширення з додатковими функціями.

# ВИСНОВКИ

У межах даної кваліфікаційної роботи було успішно реалізовано інформаційний вебпортал, що інтегрує картографічні дані з використанням сучасних вебтехнологій і безкоштовних хмарних сервісів. Результати дослідження та розробки підтвердили можливість створення ефективного, адаптивного та масштабованого рішення з мінімальними витратами на інфраструктуру.

Проведений аналіз предметної області дозволив визначити ключові технології, підходи до інтеграції геоданих та обґрунтувати вибір інструментів: Leaflet.js як бібліотека для інтерактивної візуалізації просторових об’єктів, формат GeoJSON для зберігання координованих даних, Firebase Hosting як середовище хмарного розміщення. Усі компоненти інтегровано в єдину архітектуру, побудовану за моделлю "клієнт-хостинг-дані".

Під час реалізації порталу створено базову SPA-структуру з адаптивним інтерфейсом, підтримкою фільтрації, пошуку та взаємодії з об’єктами на мапі. Здійснено тестування функціоналу на відповідність вимогам щодо швидкодії, відображення геооб’єктів, коректної роботи фільтрів та інтерактивності карти. Система продемонструвала стабільну роботу в умовах реального використання.

Практичне значення розробки полягає у можливості застосування даного підходу для побудови освітніх, муніципальних або екологічних порталів без значних фінансових затрат. Обрана архітектура дозволяє легко масштабувати систему, додавати нові шари даних, функції зворотного зв’язку або авторизації користувачів.

Таким чином, мета роботи досягнута повністю: створено функціональний інформаційний вебпортал з картографічною візуалізацією на основі безкоштовних хмарних сервісів. Отримані результати можуть бути використані як основа для подальшого вдосконалення системи, зокрема впровадження реального часу, мобільної оптимізації та інтеграції з відкритими державними даними.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. What is Information Portals. [Електронний ресурс] / IGI Global. – Режим доступу: https://www.igi-global.com/dictionary/web-portal-research-issues/37241
2. Technical Explanation of Information Portals. [Електронний ресурс] / Goong – Режим доступу: https://goong.com/word/information-portal-meaning/
3. GeoNetwork opensource. [Електронний ресурс] / Wikipedia – Режим доступу: https://en.wikipedia.org/wiki/GeoNetwork_opensource
4. Student Portal Features and Advantages. [Електронний ресурс] / Buddy4Study – Режим доступу: https://www.buddy4study.com/article/student-portals
5. Leaflet.js documentation. [Електронний ресурс] – Режим доступу: https://leafletjs.com
6. Firebase Hosting Overview. [Електронний ресурс] – Режим доступу: https://firebase.google.com/docs/hosting
7. Leaflet.js – JavaScript library for interactive maps. [Електронний ресурс] – Режим доступу : https://leafletjs.com
8. ArcGIS Online. [Електронний ресурс] / Web GIS mapping software – Режим доступу: https://www.esri.com/en-us/arcgis/products/arcgis-online/overview
9. GeoServer [Електронний ресурс] / Open source server for geospatial data – Режим доступу: https://geoserver.org
10. Mapbox GL JS [Електронний ресурс] / Vector maps on the web – Режим доступу: https://docs.mapbox.com/mapbox-gl-js
11. GitHub Pages Documentation [Електронний ресурс] – Режим доступу: https://pages.github.com
12. Netlify Hosting & Deployment Platform [Електронний ресурс] – Режим доступу: https://www.netlify.com
13. Публічна кадастрова карта України [Електронний ресурс] – Режим доступу: https://map.land.gov.ua
14. Lviv Open Data Portal [Електронний ресурс] – Режим доступу: https://opendata.city-adm.lviv.ua

15. SaveEcoBot – інтерактивна карта якості повітря [Електронний ресурс] – Режим доступу: https://www.saveecobot.com

16. Humanitarian Data Exchange [Електронний ресурс] – Режим доступу: https://data.humdata.org

17. Global Forest Watch [Електронний ресурс] – Режим доступу: https://www.globalforestwatch.org

Додадот А

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**



**Навчально-науковий інститут Інформаційних технологій**
**Кафедра Комп'ютерної інженерії**

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

# на тему: «Інформаційний портал з інтеграцією картографічних даних на безкоштовних хмарних сервісах»

Виконав студент групи КІД-41
Красавін Владислав Олександрович

Керівник кваліфікаційної роботи:
Коротков Сергій Станіславович, доцент кафедри, phD

**Київ – 2025**

**Мета** – розробити вебпортал, який відображає об’єкти на карті, з можливістю інтерактивної взаємодії та без використання платних сервісів.

**Об`єкт дослідження** – хмарні сервіси для розгортання ІТ-систем

**Предмет дослідження** – інтерактивний вебпортал з геоданими

# Актуальність обраної теми

Інтерактивні карти стали ключовим компонентом сучасних цифрових рішень. Вони дозволяють не просто бачити об’єкти на місцевості, а взаємодіяти з ними – знаходити, аналізувати, фільтрувати. Поєднання картографічних інструментів з безкоштовними хмарними платформами відкриває шлях до ефективної реалізації інформаційних сервісів навіть без значних ресурсів. Це особливо актуально в таких сферах, як урбаністика, екологічний моніторинг, транспортна аналітика та локальні ініціативи.

# Використанітехнології

- **HTML, CSS, JavaScript** – основа інтерфейсу
- **Leaflet.js** – бібліотека для роботи з картами
- **Firebase** – хмарне середовище та база даних
- **GeoJSON** – формат просторових даних
- **GitHub Pages** – розгортання проєкту

Таблиця 1.1 – Порівняльна таблиця систем інтеграції геоданих у веброзробці

| Назва | Тип | Призначення | Переваги | Недоліки / Обмеження |
|---|---|---|---|---|
| Leaflet.js | Бібліотека JS | Відображення інтерактивних карт, додавання шарів, точок | Легка, проста в освоєнні, багато плагінів, мобільна підтримка | Не підтримує векторну тайл-рендеризацію, обмежений 3D-функціонал |
| Mapbox GL JS | Бібліотека JS | Векторні карти з кастомізацією, анімацією та API | Підтримка WebGL, високопродуктивна візуалізація, 3D-об’єкти, власні стилі | Безкоштовна версія має обмеження, складніше у використанні |
| GeoServer | Серверна система | Видача геоданих за стандартами WMS/WFS, стилізація, фільтрація | Open-source, сумісність із PostGIS, широкі можливості налаштування | Потребує налаштування сервера, складний у розгортанні |
| ArcGIS Online | Хмарна ГІС-платформа | Хостинг, візуалізація, аналітика, публікація геоданих | Хмарне розгортання, інтеграція в портал, підтримка 3D, аналітичні функції | Комерційна ліцензія, обмеження у безкоштовному режимі |
| GeoJSON | Формат даних | Зберігання векторних об’єктів (точки, лінії, полігони) | Прозорий, сумісний із JS, зручний для REST/API | Не підходить для великих наборів даних без оптимізації |
| Firebase Hosting | Хмарний хостинг | Розміщення статичних сайтів та файлів (наприклад GeoJSON) | Безкоштовно, просто у використанні, CI/CD, інтеграція з Google Cloud | Лише для статичного контенту, немає серверної логіки |
| PostGIS | База даних | Просторове розширення до PostgreSQL, зберігання геоданих | Висока точність, підтримка SQL-запитів, інтеграція з GeoServer | Потрібні знання SQL та налаштування серверної частини |

# Архітектура клієнт-серверної взаємодії

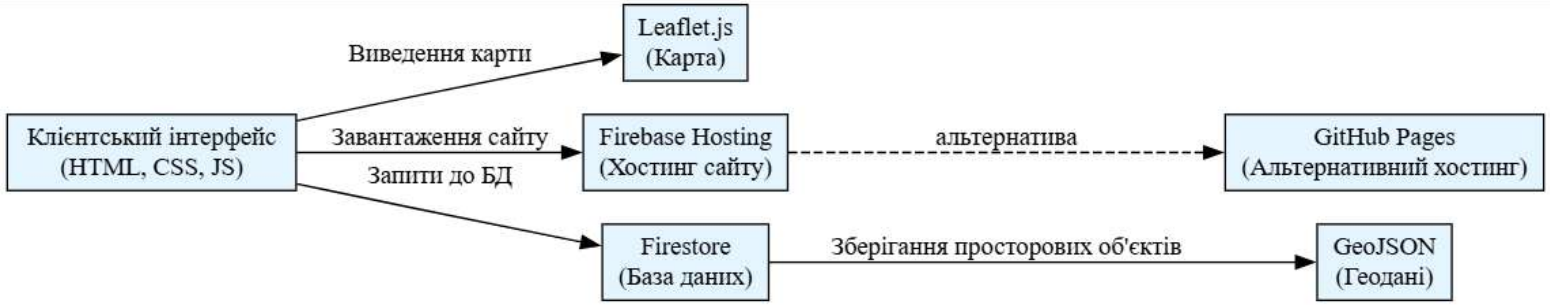



Рисунок 2.1 – Архітектура клієнт-серверної взаємодії

# Модульна структура вебпорталу

Таблиця 2.2 – Модульна структура вебпорталу

| Назва модуля | Функціональність |
|---|---|
| Модуль мапи | Ініціалізація Leaflet, підключення тайлів, виведення об'єктів, управління шаром |
| Модуль даних | Запит до Firestore, обробка GeoJSON, передача даних у карту |
| Модуль інтерактивності | Обробка кліків, відкриття інфо-вікон, реагування на дії користувача |
| Модуль керування | Побудова фільтрів, пошук, оновлення вмісту, перемикання тематичних шарів |




Рисунок 2.2 – Схема модульної структури вебпорталу

# Сценарії користувацької взаємодії




Рисунок 2.3 – Інтерфейсна схема користувацьких сценаріїв

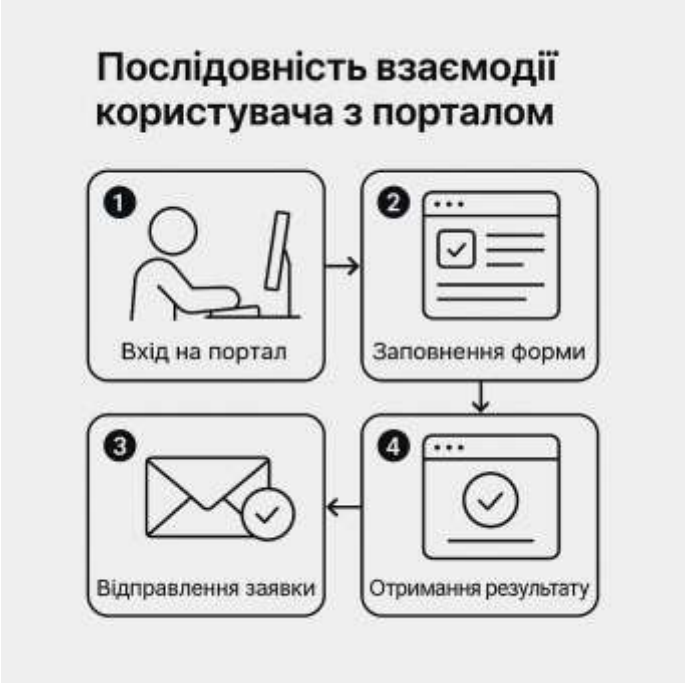



Рисунок 2.4 – Послідовність взаємодії користувача з порталом

# Зберігання та обробка просторових даних

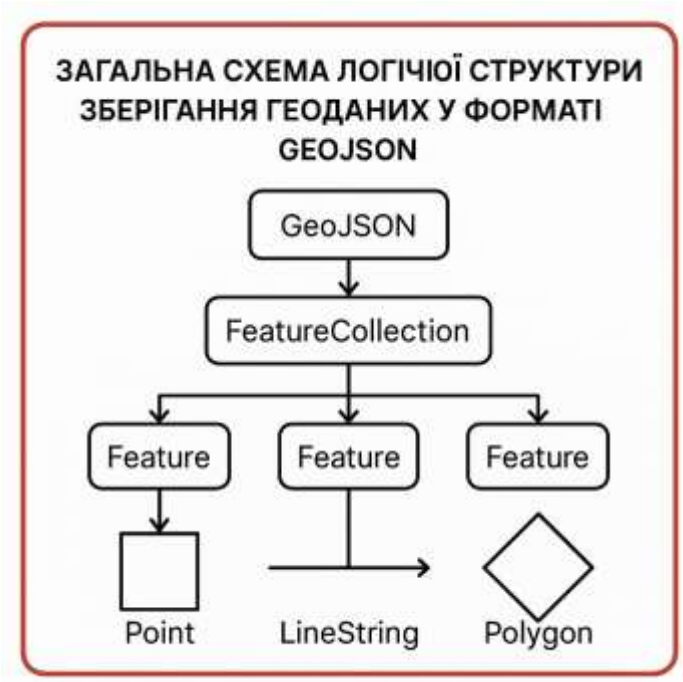



Рисунок 2.5 – Загальна схема логічної структури зберігання геоданих у форматі GeoJSON

# Інтерфейс інформаційного порталу

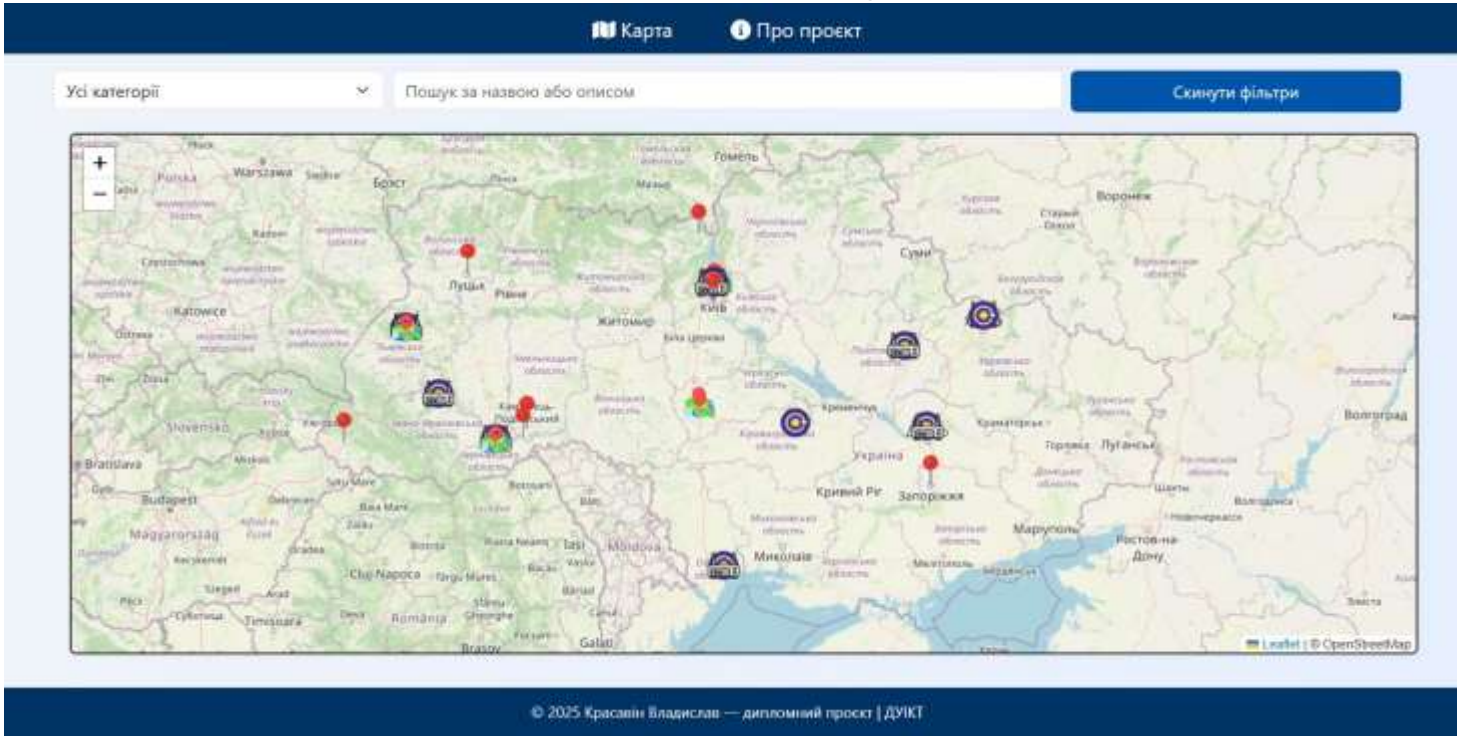



Рисунок 3.1 – Загальне підключення та ініціалізація карти Leaflet.js з базовими тайлами OpenStreetMap

# Тестування функціоналу

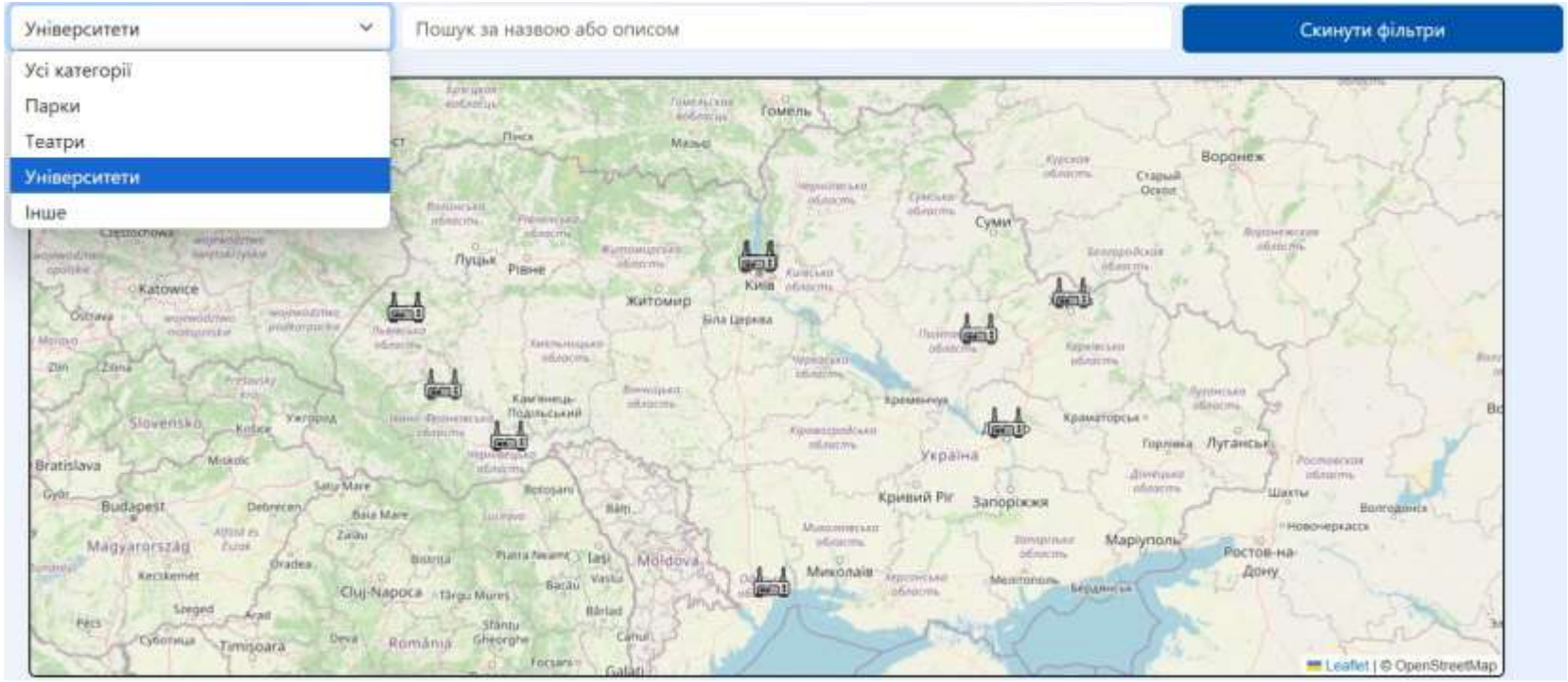



Рисунок 3.5 – Візуальне відображення маркерів на мапі згідно з категоріями об’єктів

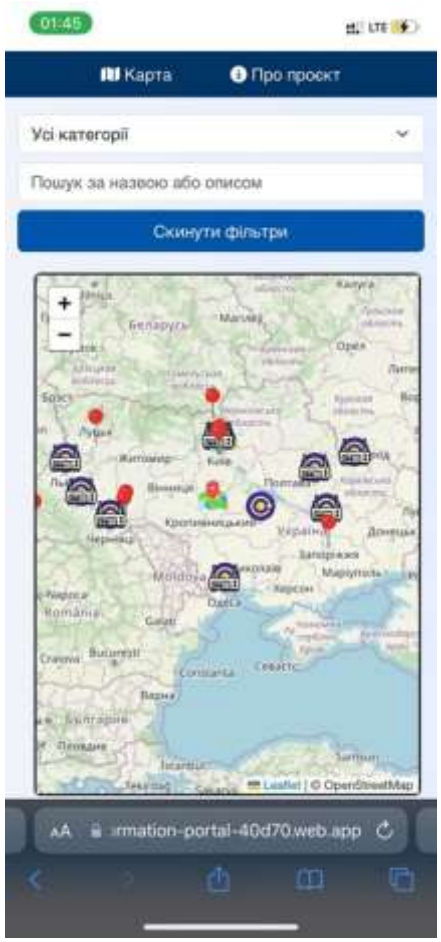



Рисунок 3.4 – Відображення мобільної версії інтерфейсу з адаптивними елементами керування

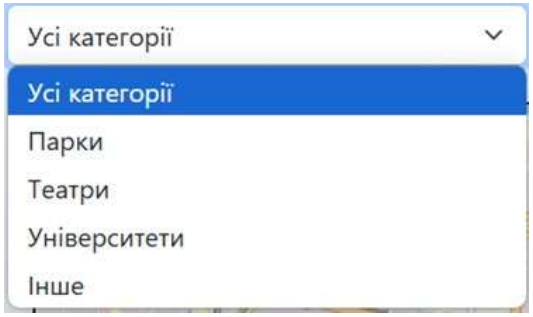



Рисунок 3.7 – Інтерфейс фільтрації та пошуку на карті

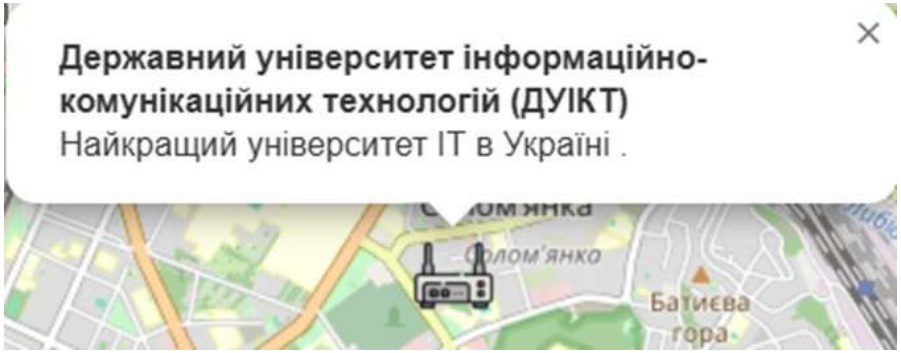



Рисунок 3.6 – Приклад відображення інформації про об’єкт у спливаючому вікні (popup)

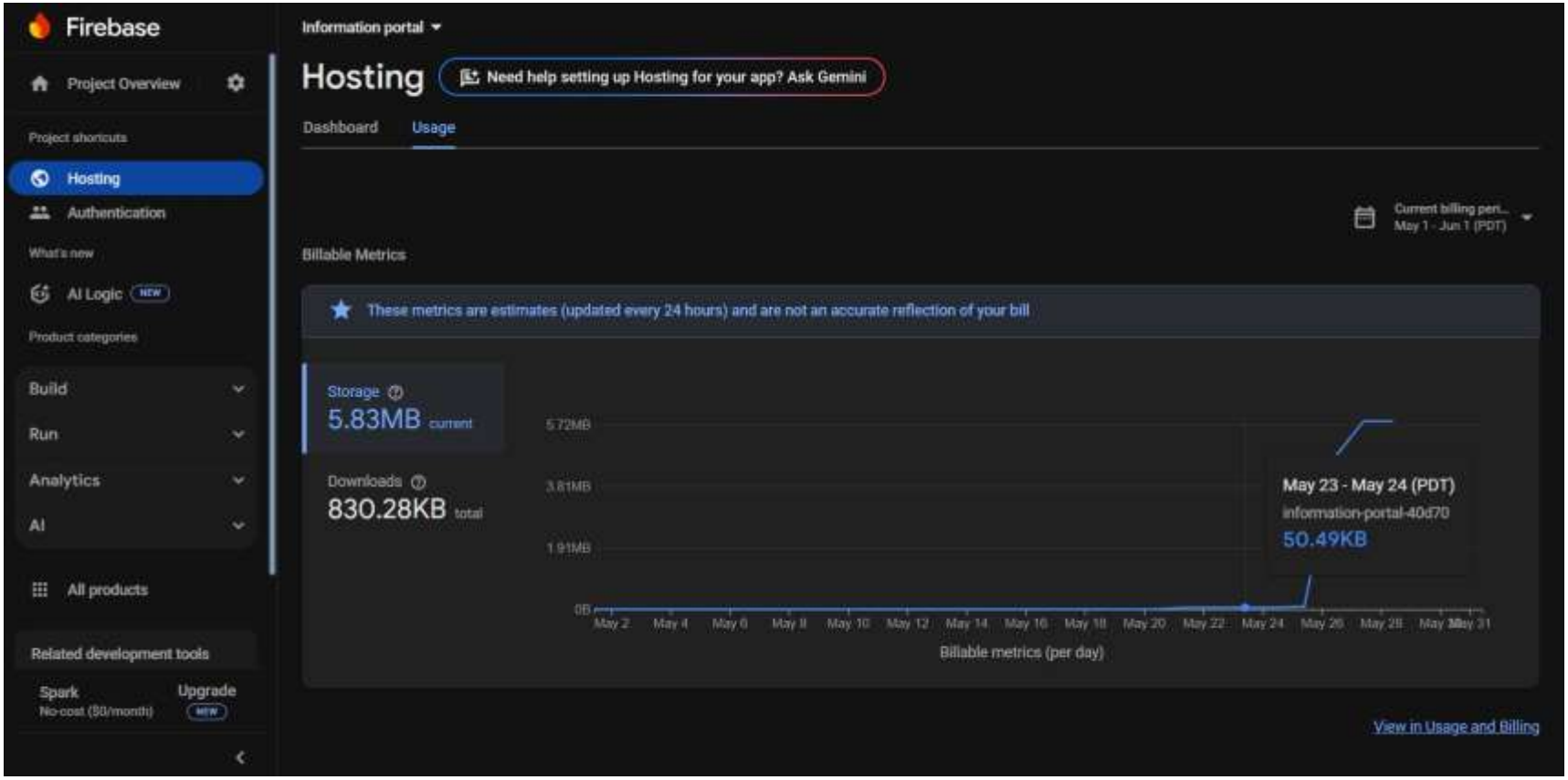



Рисунок 3.9 – Хостинг в Firebase

# Висновки

У результаті виконання кваліфікаційної роботи створено повноцінний інформаційний вебпортал з інтерактивною картою, який працює на основі безкоштовних хмарних сервісів. Було здійснено аналіз сучасних вебтехнологій, обґрунтовано вибір інструментів розробки та реалізовано архітектуру односторінкового застосунку. Портал забезпечує динамічне завантаження та візуалізацію геоданих, підтримує взаємодію користувача з картою, включає функції фільтрації та роботи з об’єктами у реальному часі. Систему розгорнуто у хмарному середовищі Firebase, що забезпечує високу доступність, масштабованість та простоту адміністрування. Проведене тестування підтвердило функціональну придатність розробленого рішення, яке може бути використане як основа для подальших прикладних або дослідницьких проєктів у сфері геоінформаційних технологій.