**Software Requirements Specification**

**for**

**Програмна система моніторингу землі для сільськогосподарських потреб «SoilMonitoring»**

**Version 1.0 approved**

**Prepared by Ivan Shevchenko**

**PZPI-21-8**

**15.06.2025**

**ЗМІСТ**

1 ВСТУП 4

1.1 Огляд продукту 4

1.2 Мета 4

1.3 Межі 4

1.4 Посилання 4

1.5 Означення та абревіатури 4

2. ЗАГАЛЬНИЙ ОПИС 5

2.1 Перспективи продукту 5

2.2 Функції продукту 5

2.3 Характеристики користувачів 5

2.4 Загальні обмеження 5

2.5 Припущення й залежності 6

3. КОНКРЕТНІ ВИМОГИ 6

3.1 Вимоги до зовнішніх інтерфейсів 6

3.1.1 Інтерфейс користувача 6

3.1.2 Апаратний інтерфейс 10

3.1.3 Програмний інтерфейс 10

3.1.4 Комунікаційний протокол 10

3.1.5 Обмеження пам'яті 11

3.1.6 Операції 11

3.1.7 Функції продукту 11

3.1.8 Припущення й залежності 12

3.2 Властивості програмного продукту 12

3.3 Атрибути програмного продукту 12

3.3.1 Надійність 12

3.3.2 Доступність 12

3.3.3 Безпека 13

3.3.4 Супроводжуваність 13

3.3.5 Переносимість 13

3.3.6 Продуктивність 13

3.4 Вимоги бази даних 13

3.5 Інші вимоги 14

1. Вступ

1.1 Огляд продукту

Дана система призначена для аграріїв, фермерських господарств та організацій, що займаються вирощуванням культур на відкритому ґрунті.

Програмна система дозволяє автоматизовано отримувати дані з ґрунтових сенсорів (вологість, температура, рівень поживних речовин), відображати їх у веб-інтерфейсі, формувати аналітику, надавати рекомендації та планувати агротехнічні дії.

1.2 Мета

Метою створення даного ПЗ є спрощення процесу моніторингу стану полів, покращення планування аграрних робіт і підвищення ефективності прийняття рішень завдяки інтеграції IoT-технологій та геоінформаційних сервісів.

1.3 Межі

* Система не замінює агрономічну експертизу, але може її доповнювати.
* Система не здійснює контроль над дронами або поливальними системами — лише рекомендації.
* Система не виконує складних кліматичних прогнозів, а інтегрує базовий погодний API.

1.4 Посилання

Посилання на GitHub репозиторій проєкту:

https://github.com/NureShevchenkoIvan/soilmonitor

1.5 Означення та абревіатури

IoT – Internet of Things (Інтернет речей)

API – Application Programming Interface

ESP32 – Embedded Serial Peripheral 32-bit SoC

JSON – JavaScript Object Notation

JWT – JSON Web Token

UI/UX – User Interface / User Experience

CRUD – Create, Read, Update, Delete

NPK – Nitrogen, Phosphorus, Potassium

2. Загальний опис

2.1 Перспективи продукту

Розроблене програмне забезпечення є частиною ширшої тенденції цифровізації сільського господарства — концепції «розумного фермерства». У майбутньому система може бути розширена за рахунок:

* мобільного застосунку для iOS/Android;
* автоматизованої системи оповіщень (push/spam alerts);
* машинного навчання для прогнозування агротехнічних дій;
* інтеграції з супутниковими або дроновими даними.

2.2 Функції продукту

Основні функції, які виконує система:

* Авторизація/реєстрація користувача, доступ за роллю;
* Додавання, перегляд та редагування сільськогосподарських полів;
* Візуалізація меж полів на карті;
* Додавання сенсорів (GPS-локація, тип);
* Прийом даних із сенсорів: вологість, температура, NPK;
* Побудова графіків за показниками з можливістю вибору періоду;
* Створення нотаток та завантаження документів до полів;
* Планування аграрних завдань у календарі;
* Прогноз погоди по полю;
* Панель адміністратора: керування користувачами;
* Рекомендації на основі зчитувань.

2.3 Характеристики користувачів

* Фермер/Агроном - Основний користувач. Має доступ до полів, сенсорів, календаря, прогнозів, рекомендацій.
* Адміністратор - Має додаткові повноваження: керування користувачами, блокування, видалення.

2.4 Загальні обмеження

* Система реалізована як веб-застосунок і потребує наявності стабільного інтернет-з’єднання.
* IoT-модуль не має автономного джерела живлення — тестування відбувалося у віртуальному середовищі Wokwi.
* Максимальна кількість полів на одного користувача — до 100 (можна масштабувати).
* Сенсорні дані не проходять сертифіковану калібровку — результати орієнтовні.

2.5 Припущення й залежності

* Користувач вводить достовірні координати та інформацію про поля.
* Пристрій ESP32 успішно підключається до Wi-Fi та має доступ до інтернету.
* Клієнт запускається у сучасному браузері (Chrome, Firefox, Edge).
* Сервер працює на хостингу або локально (Node.js >= v16, MongoDB >= v5).

3. Конкретні вимоги

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

Інтерфейс реалізовано у вигляді SPA (single-page application) з використанням React.js та Material UI.

Основні вимоги:

1. Меню навігації у вигляді бічної панелі (див. рис. 1), яка адаптивно ховається на малих екранах.
2. Головна сторінка (див. рис. 1) повинна містити:

* список полів;
* їхні статуси (зелений, жовтий, червоний);
* картки аналітики: активні сенсори, поля з критичними показниками, заплановані завдання;
* інтерактивну мапу.

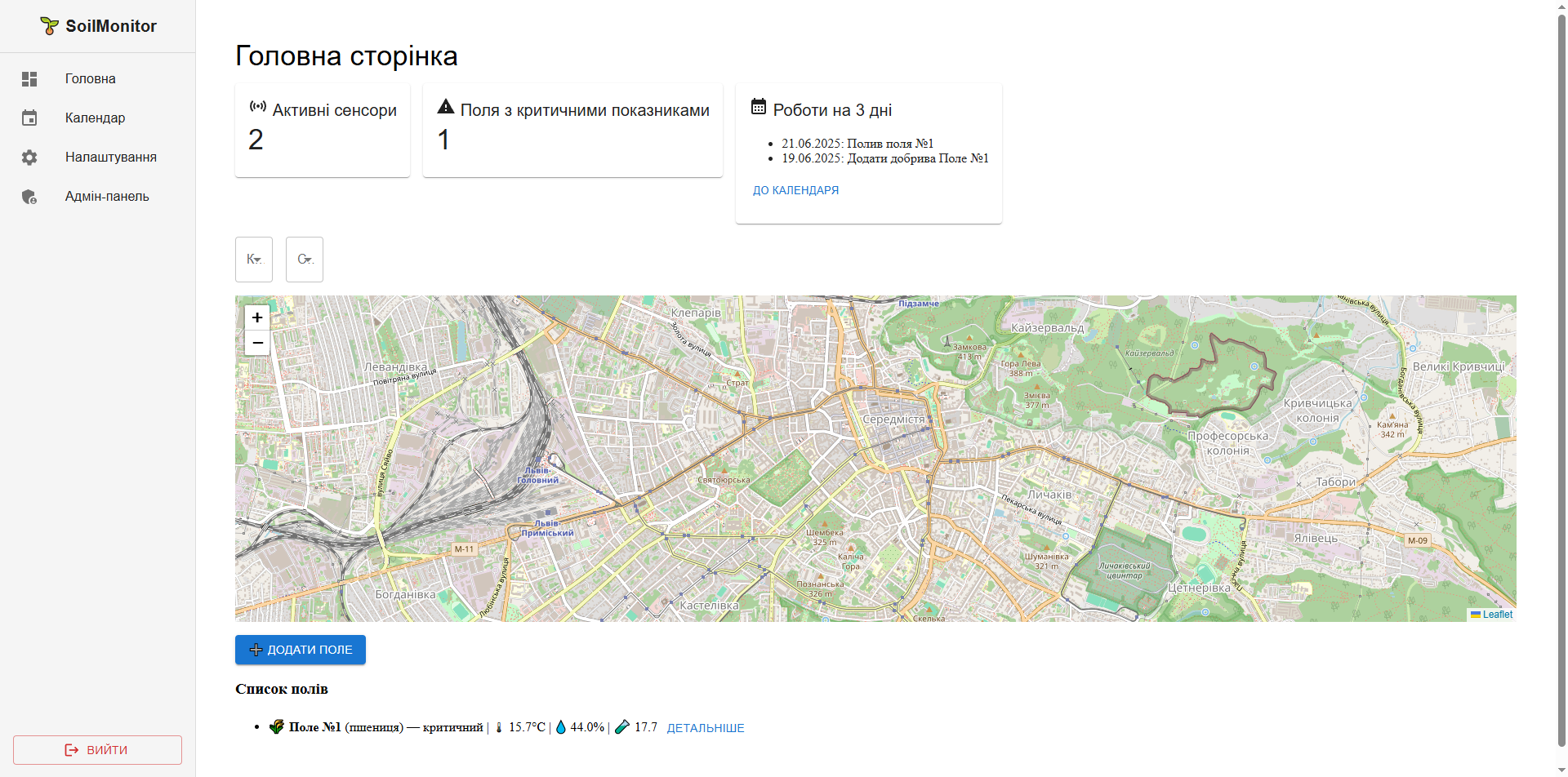


Рисунок. 1 – Інтерфейс головної сторінки з картою та списком полів

1. Форма додавання поля (див. рис. 2) дозволяє намалювати межі на карті, обрати культуру та дату посіву.

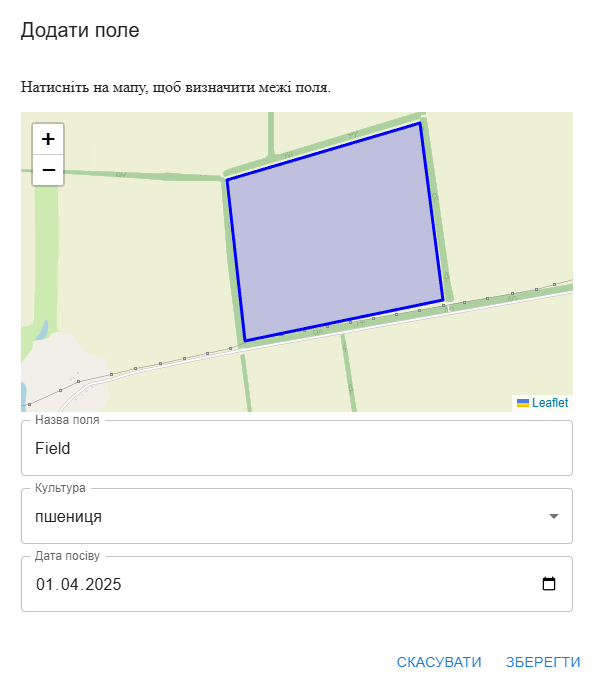
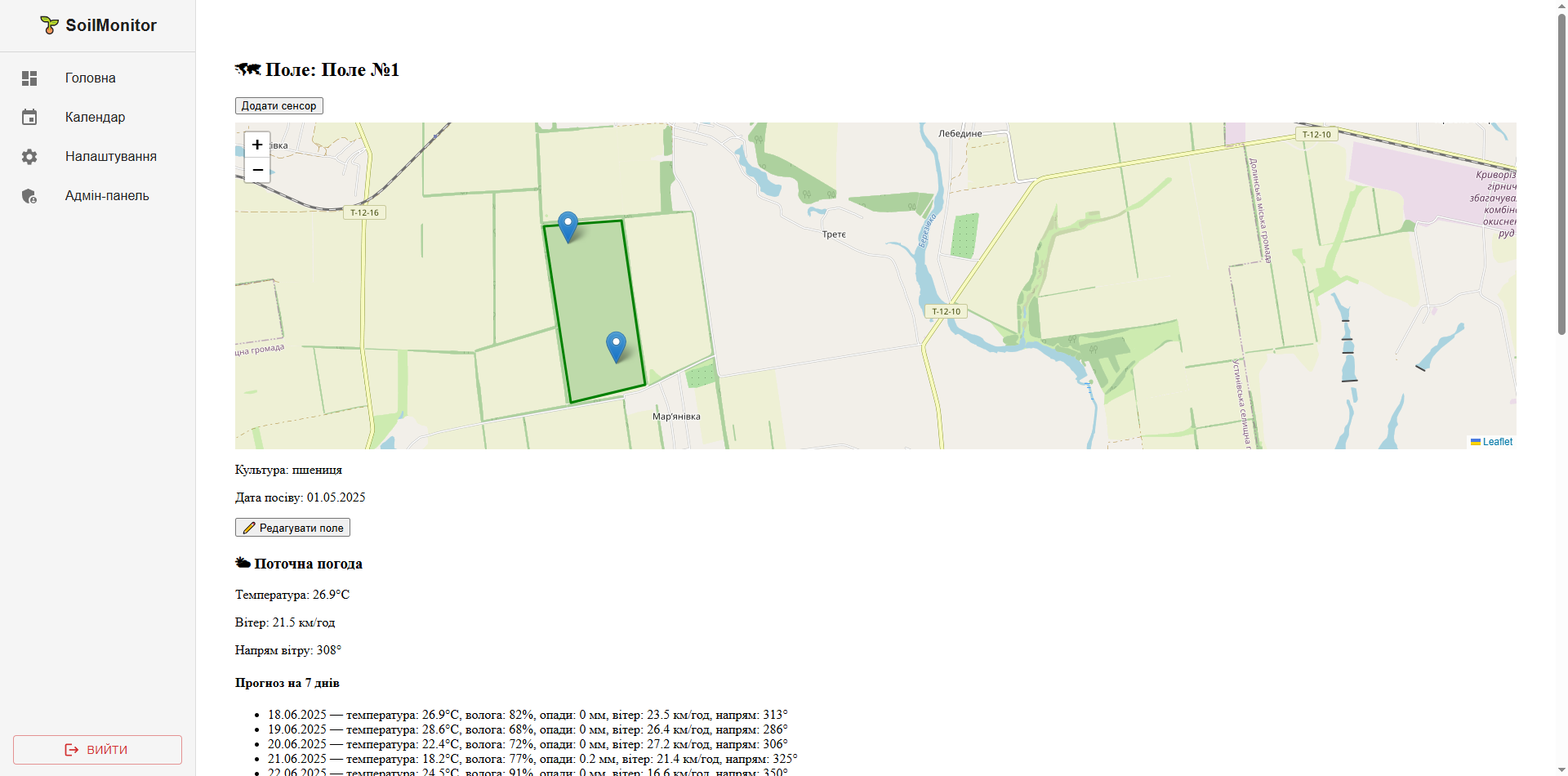
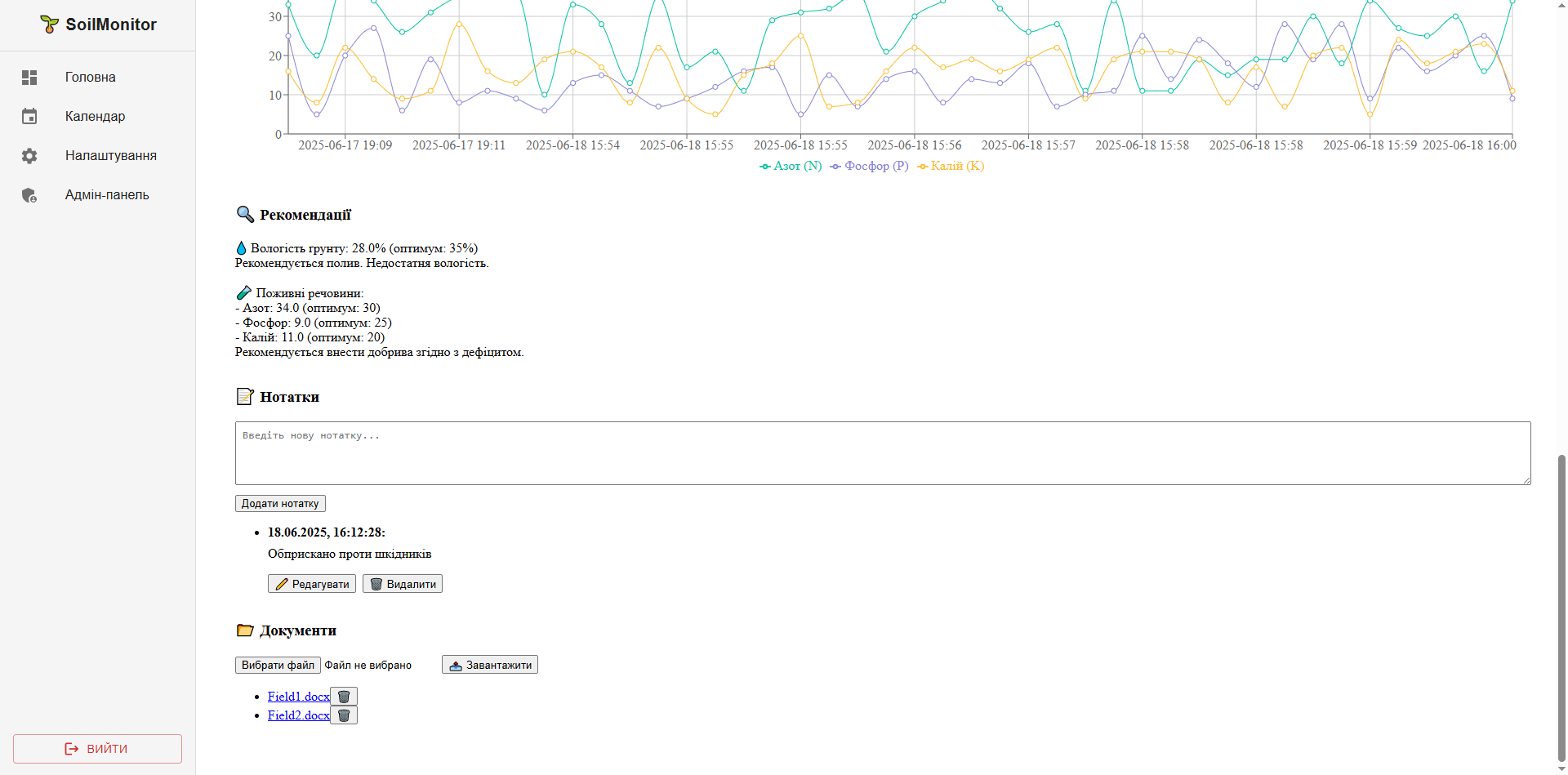


Рисунок 2 – Інтерфейс форми додавання поля на головній сторінці

1. Сторінка деталей поля (див. рис. 3-4) включає:

* карту з сенсорами;
* віджет погоди;
* блок інформації з можливістю редагування;
* блок з графіками за обраний період (день/тиждень/місяць);
* нотатки та документи.





Рисунки 3-4 – Фрагменти інтерфейсу сторінки деталей поля

1. Календар робіт (див. рис. 5) реалізований через компонент типу FullCalendar з кольоровим маркуванням типів робіт.

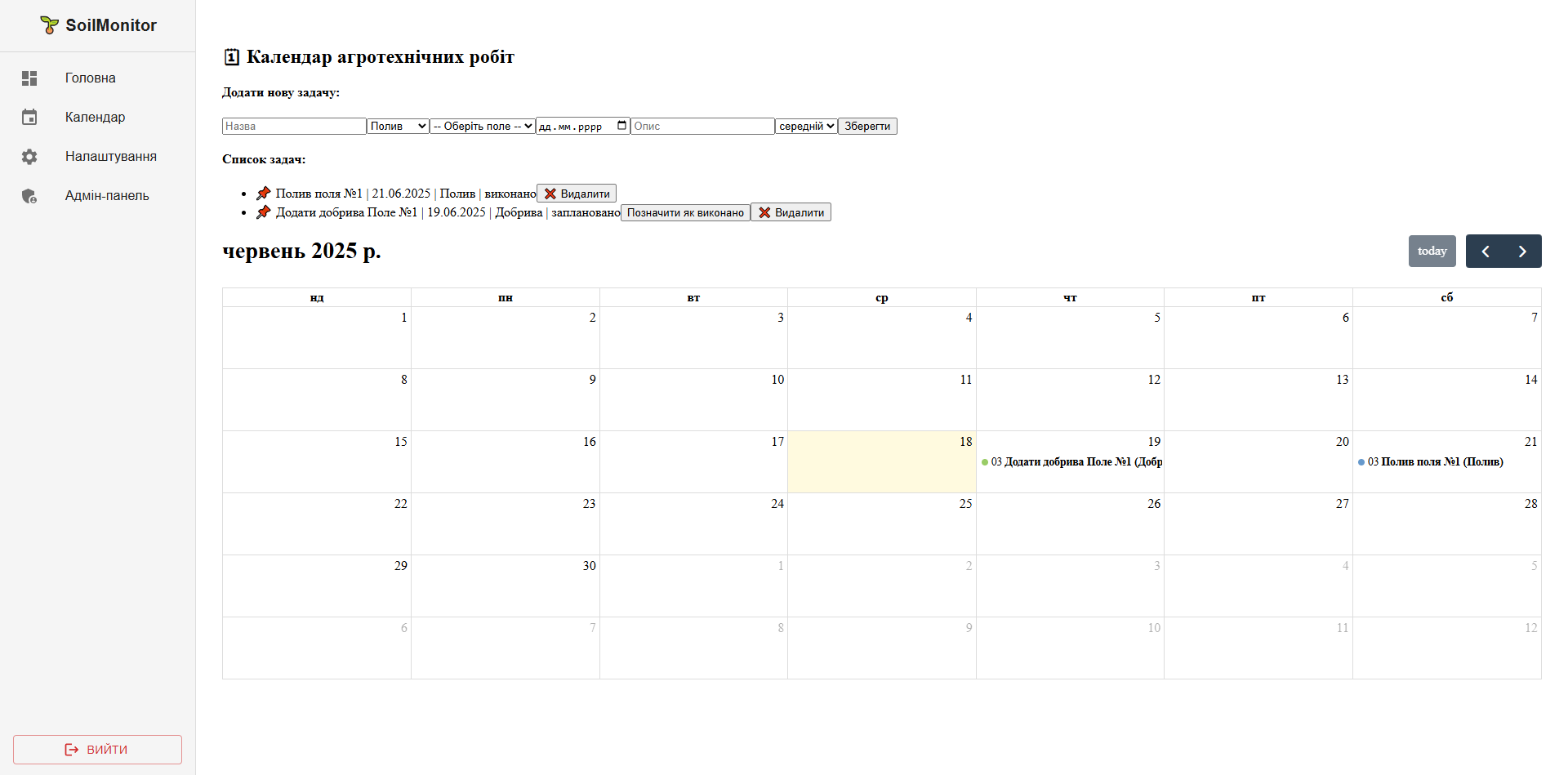


Рисунок 5 – Інтерфейс сторінки календаря агротехнічних робіт

1. Сторінка налаштувань (див. рис. 6) дає змогу змінювати персональні дані та пароль із валідацією.

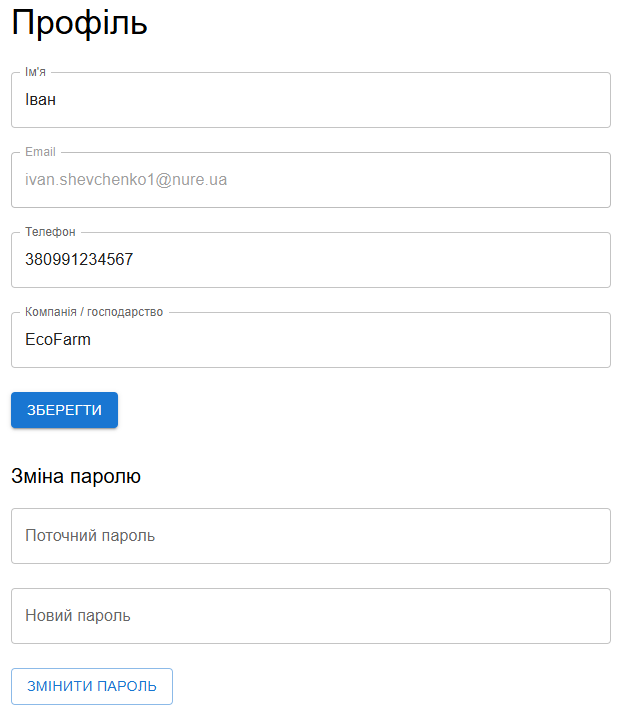


Рисунок 6 – Інтерфейс сторінки налаштувань

1. Адміністративна панель (для ролі admin) (див. рис. 7) — список усіх користувачів із можливістю редагування, блокування, видалення.

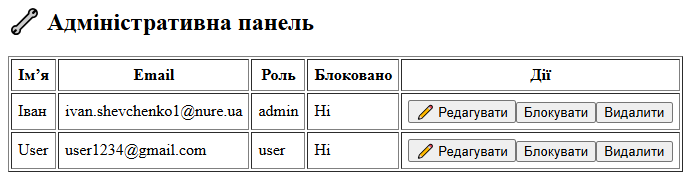


Рисунок 7 – Інтерфейс адміністративної панелі

3.1.2 Апаратний інтерфейс

Система працює із зовнішнім пристроєм ESP32, який:

* має підключений датчик DHT22 (температура, вологість);
* потенціометр, що імітує показники NPK (рівень поживних речовин);
* передає дані у вигляді HTTP POST запиту на відповідний серверний ендпоінт.

ESP32 взаємодіє з сервером, використовуючи Wi-Fi-з'єднання, без потреби додаткових драйверів або апаратних шлюзів.

3.1.3 Програмний інтерфейс

Серверна частина реалізує RESTful API. Основні ендпоінти:

* /api/auth/register, /api/auth/login — автентифікація;
* /api/fields, /api/fields/:id — CRUD-операції з полями;
* /api/sensors, /api/sensor-readings — робота з сенсорами та зчитуваннями;
* /api/tasks — додавання агротехнічних завдань;
* /api/users — робота з користувачами (admin).

Передбачено повернення кодов стану HTTP, обробку помилок, повідомлення на фронтенд. Дані передаються у форматі JSON.

3.1.4 Комунікаційний протокол

Для взаємодії між IoT-пристроєм і сервером використовується HTTP/1.1 через Wi-Fi. Передбачено:

* формат JSON для запиту;
* метод POST;
* заголовки Content-Type: application/json.

Фронтенд використовує HTTPS-з’єднання при розгортанні у продуктивному середовищі.

3.1.5 Обмеження пам’яті

* Серверна частина не повинна перевищувати 512 МБ оперативної пам’яті при стандартному навантаженні.
* IoT-пристрій ESP32 має обмеження на розмір JSON (не більше 1 КБ на один запит).
* Зображення та документи обмежуються 5 МБ для одного файлу (при завантаженні до поля).

3.1.6 Операції

Операції, які повинні бути доступні користувачеві:

* Реєстрація / вхід / вихід;
* Додавання, редагування, видалення полів;
* Додавання / переміщення / видалення сенсорів;
* Перегляд історії вимірів;
* Генерація графіків;
* Додавання нотаток та документів;
* Створення завдань;
* Перегляд прогнозу погоди;
* Зміна налаштувань;
* Управління користувачами (admin).

3.1.7 Функції продукту

|  |  |  |
| --- | --- | --- |
| № | Назва функції | Опис |
| 1 | |  | | --- | | Візуалізація полів |  |  | | --- | |  | | |  | | --- | | Інтерактивна карта з межами полів та статусами |  |  | | --- | |  | |
| 2 | |  | | --- | | Робота з сенсорами |  |  | | --- | |  | | |  | | --- | | Прив’язка сенсорів до поля, відображення показників |  |  | | --- | |  | |
| 3 | Аналітика | |  | | --- | | Побудова графіків, рекомендації |  |  | | --- | |  | |
| 4 | |  | | --- | | Календар робіт |  |  | | --- | |  | | |  | | --- | | Планування поливу, внесення добрив, інших дій |  |  | | --- | |  | |
| 5 | |  | | --- | | Сторінка налаштувань |  |  | | --- | |  | | |  | | --- | | Редагування особистих даних, зміна пароля |  |  | | --- | |  | |
| 6 | Адмінпанель | Керування акаунтами, блокування, видалення користувачів |

3.1.8 Припущення і залежності

* Сторонні API (погода, мапи) працюють стабільно та без помилок.
* ESP32 має стабільне підключення до мережі Wi-Fi.
* Користувач вводить достовірну інформацію при реєстрації.

3.2 Властивості програмного продукту

Програмна система повинна відповідати наступним функціональним та нефункціональним властивостям:

* Функціональна повнота: усі заявлені функції реалізовані згідно з ТЗ.
* Інтуїтивність: інтерфейс повинен бути зрозумілим без потреби у додатковому навчанні.
* Модульність: система складається з незалежних частин (IoT, сервер, клієнт), що дозволяє їх масштабування чи заміну.
* Візуальна доступність: ключові показники (стан полів, прогноз, рівень вологості) мають бути виведені одразу на дашборді.
* Чутливість до контексту: дані, що відображаються, залежать від вибраного поля, культури та сенсорів.

3.3 Атрибути програмного продукту

3.3.1 Надійність

* Система має коректно обробляти невалідні вхідні дані та помилки API.
* У випадку втрати інтернету дані не надсилаються, а система відображає повідомлення.

3.3.2 Доступність

* Система доступна через браузер 24/7.
* IoT-модуль має працювати циклічно та передавати дані кожні 30 секунд (при наявності живлення).

3.3.3 Безпека

* Захист маршрутів за допомогою JWT-токенів.
* Шифрування паролів користувачів за допомогою bcrypt.
* Перевірка прав доступу при кожному запиті.

3.3.4 Супроводжуваність

* Код організовано за принципами REST, модулі відокремлені.
* Вся логіка структурується у вигляді окремих контролерів, сервісів та моделей.

3.3.5 Переносимість

* Програмна система може бути розгорнута на будь-якому сервері з підтримкою Node.js та MongoDB.
* ESP32 може бути перепрошитий для надсилання даних на інший сервер, змінюючи лише URL.

3.3.6 Продуктивність

* Відповідь API при звичайному навантаженні не перевищує 300 мс.
* Запит з ESP32 до сервера обробляється менш ніж за 1 секунду.
* Рендеринг списку полів та мапи – <1 сек.

3.4 Вимоги бази даних

Система використовує MongoDB як основну базу даних. Основні вимоги:

1. Колекції (див. рис. 8): users, fields, sensors, sensorreadings, tasks
2. Індексація: email користувача, fieldId у сенсорах і зчитуваннях.
3. Типи даних:

* координати меж полів – масив об'єктів типу [{lat, lng}];
* зчитування – об'єкт data: { temperature, moisture, npk: {n, p, k} }

1. Обмеження:

* Максимум 1000 зчитувань на сенсор за місяць;
* Максимум 10 МБ на завантажені документи на поле.

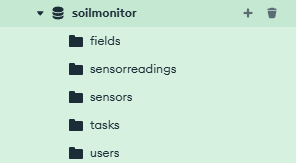


Рисунок 8 – Схема колекцій у MongoDB Compass

3.5 Інші вимоги

* Мова користувацького інтерфейсу: українська.
* Мова кваліфікаційної роботи та документації: українська.
* Мови програмування: JavaScript, HTML, CSS (React.js, Node.js), C++ (ESP32).
* Платформи розгортання: Windows/Linux-сервер, браузери Chrome/Firefox/Edge.
* Обмеження щодо третіх сторін: Використання API OpenWeatherMap, Wokwi, Google Maps (або Leaflet).