ДОДАТОК А

Звіт результатів перевірки на унікальність тексту

StrikePlagiarism.com

Дата звіту 5/29/2025
Дата редагування ---

Звіт не був оцінений

Звіт подібності

метадані

Назва організації
Kharkiv National University of Radio Electronics
Заголовок
2025_Б_ПІ_ПЗПІ-21-10_Островерхов_Є_А_скорочений
Автор
Островерхов Єгор Андрійович
Науковий керівник / Експерт
Євген Кардаш
підрозділ
каф. ПІ

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

1.14% 1.14% КП 1

0.38% 0.38% КЦ

25 Довжина фрази для коефіцієнта подібності 2

3425 Кількість слів

28148 Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		3

Подібності за списком джерел

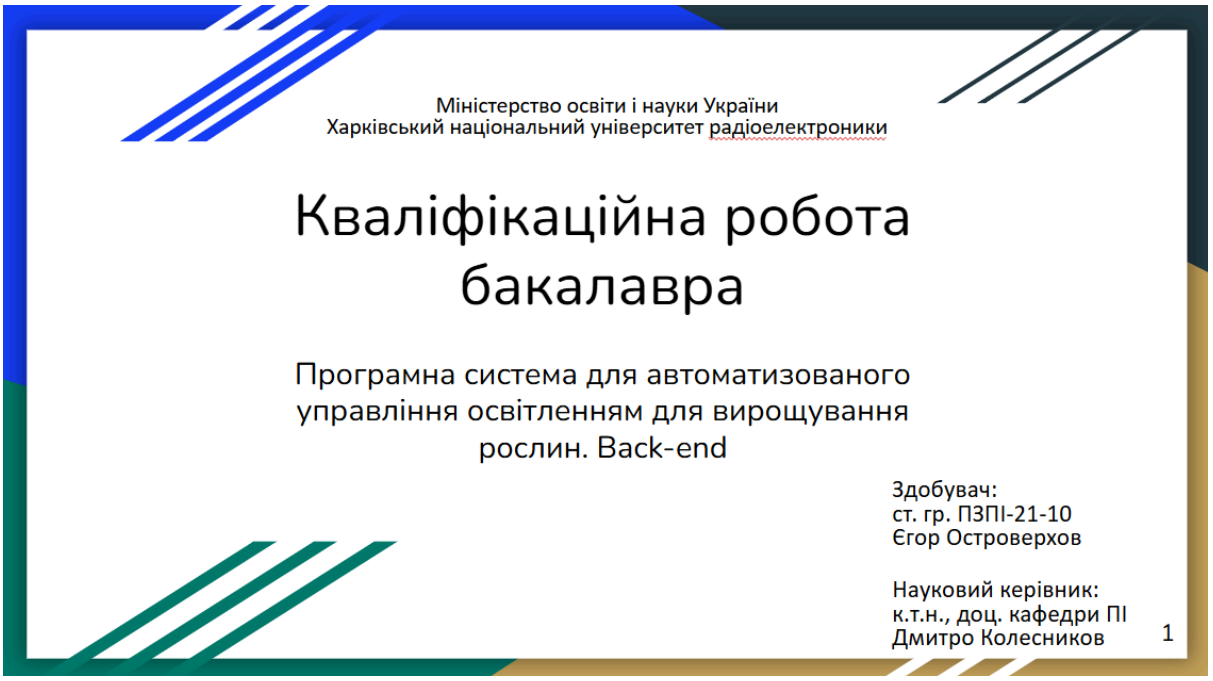
Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

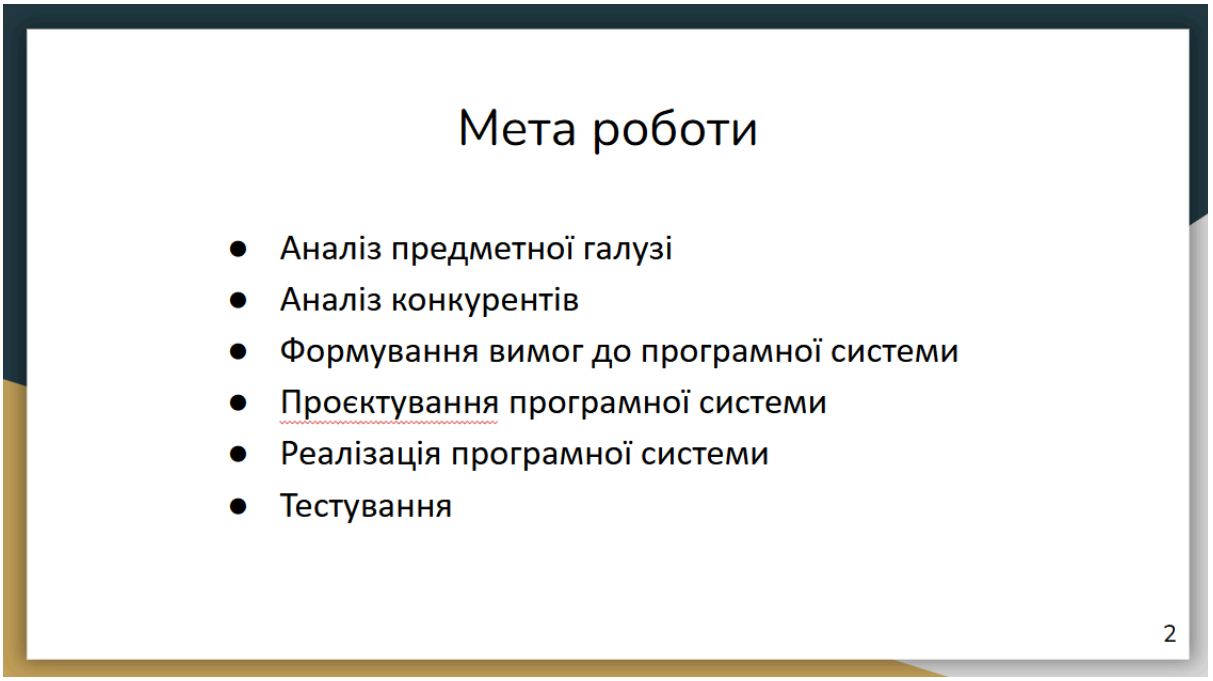
10 найдовших фраз — Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://openarchive.nure.ua/bitstreams/dcb865f8-766f-4683-a4bb-5cb937468862/download	11 0.32 %
2	https://ela.kpi.ua/bitstreams/5d896f18-05a7-4655-a184-eda579e3e2fe/download	10 0.29 %
3	ЗАХИСТ ІНТЕРНЕТ-МАГАЗИНУ ГАДЖЕТІВ ВІД КІБЕРАТАК [redacted] 5/29/2024 International University of Economics and Humanities named after Academician Stepan Demianchuk (International University of Economics and Humanities named after Academician Stepan Demianchuk)	10 0.29 %

ДОДАТОК Б

Слайди презентації





Аналіз предметної галузі

- Зростання інтересу до автоматизованих систем освітлення
- Переважно статичне керування освітленням
- Відсутність адаптації до змін у середовищі вирощування рослин

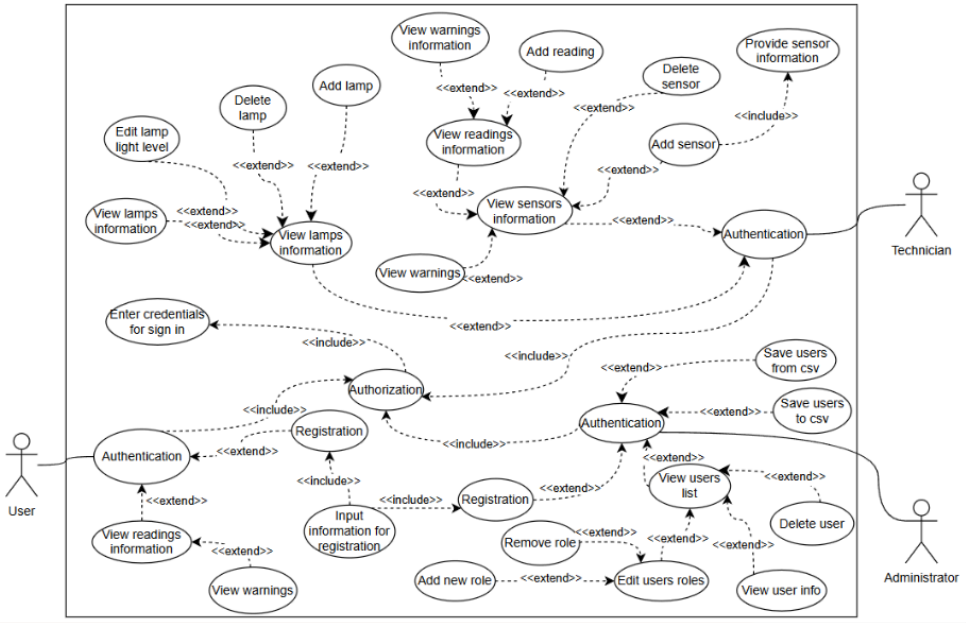
Існуючі рішення

Характеристика	Sollum Technologies	Heliospectra	Bioled	Kroptek
Інноваційні, сучасні рішення	+	+	–	+
Енергоефективність	–	+	+	+
Автоматизація освітлення	+	+	–	–
Інтеграція з сенсорами	+	–	–	–
Підтримка мобільного/веб-інтерфейсу	+	+	–	+
Динамічна зміна освітлення за кліматичними параметрами	+	+	–	–

Постановка задачі

- Розробка серверної частини
- Реалізація REST API
- Авторизація користувачів
- Зберігання даних
- Обробка даних в залежності від показників
- Резервне копіювання

Інструменти розробки



Приклад коду контролера

```java
@RestController  Cheese
@RequestMapping("/lamps")
@RequiredArgsConstructor
public class LampController {
    Service handling lamp-related operations.

    private final LampService lampService;

    Endpoint to retrieve all lamps.

    Returns: ResponseEntity containing the list of all lamps

    @GetMapping  Cheese
    public ResponseEntity<List<LampEntity>> getAllLamps() {
        return ResponseEntity.ok(lampService.getAllLamps());
    }
```

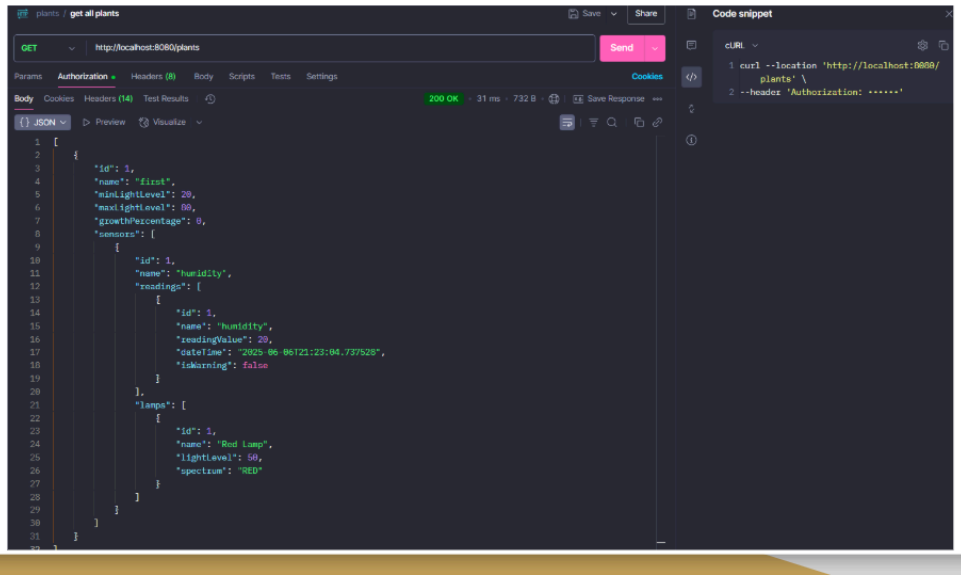
Приклад коду з сервісу

```java
    Retrieves all lamps associated with a specific sensor.
    Params: sensorId – The ID of the sensor.
    Returns: A list of LampEntity objects associated with the specified sensor ID.
public List<LampEntity> getAllSensorLamps(Long sensorId) {  1 usage   Cheese
    return lampRepository.findAllBySensorId(sensorId);
}

    Saves a new lamp associated with a specific sensor.
    Params: lampDto – The LampDto object containing lamp details.
    Returns: A set of LampEntity objects associated with the sensor after saving the new lamp.
public Set<LampEntity> saveLamp(LampDto lampDto) {  1 usage   Cheese
    SensorEntity sensor = sensorService.getSensorById(lampDto.sensorId());

    LampEntity lamp = new LampEntity(lampDto.name(), lampDto.lightLevel(), lampDto.spectrum());

    lamp.setSensor(sensor);
    sensor.addLamp(lamp);

    return sensorService.saveSensor(sensor).getLamps();
}
```


Приклад репозиторію

```java
    Repository interface for LampEntity to perform CRUD operations.
@Repository  2 usages   Cheese
public interface LampRepository extends JpaRepository<LampEntity, Long> {
        Retrieves a lamp by its name.
        Params: name – The name of the lamp to retrieve.
        Returns: The LampEntity with the specified name.
    LampEntity findByName(String name);  1 usage   Cheese

        Retrieves all lamps associated with a specific sensor ID.
        Params: sensorId – The ID of the sensor to retrieve lamps for.
        Returns: A list of LampEntity objects associated with the specified sensor ID.
    List<LampEntity> findAllBySensorId(Long sensorId);  1 usage   Cheese
}
```

Тестування


plants / get all plants
Save
Share
GET
http://localhost:8080/plants
Send
Params
Authorization
Headers (8)
Body
Scripts
Tests
Settings
Cookies
Body
Cookies
Headers (14)
Test Results
200 OK
31 ms
732 B
Save Response
{ } JSON
Preview
Visualize
Code snippet
cURL
curl --location 'http://localhost:8080/plants' \
--header 'Authorization: ******'
[
{
"id": 1,
"name": "first",
"minLightLevel": 20,
"maxLightLevel": 80,
"growthPercentage": 0,
"sensors": [
[
"id": 1,
"name": "humidity",
"readings": [
[
"id": 1,
"name": "humidity",
"readingValue": 20,
"dateTime": "2025-06-06T21:23:04.737525",
"isWarning": false
]
],
"lamps": [
[
"id": 1,
"name": "Red Lamp",
"lightLevel": 50,
"spectrum": "RED"
]
]
]
]
]

ДОДАТОК В

Специфікація ПЗ

1 ВСТУП

1.1 Огляд продукту

Розроблений програмний продукт є серверною системою управління освітленням для вирощування рослин. Він забезпечує автоматичне регулювання освітлення на основі даних сенсорів, стану рослин та параметрів навколишнього середовища. Система включає функції авторизації, управління користувачами, пристроями та базою даних, з можливістю резервного копіювання і масштабування.

1.2 Мета

Метою даного продукту є автоматизація керування освітленням для оптимального вирощування рослин на основі даних з сенсорів.

1.3 Межі

Серверна частина:

– розроблена з використанням Java 17 та Spring Boot;

– реалізоване REST API для обміну даними між клієнтами, сенсорами та пристроями освітлення;

– використовується PostgreSQL для зберігання даних про рослини, сенсори, джерела світла та користувачів;

– реалізована авторизація користувачів із підтримкою ролей (адміністратор, технічний персонал, звичайний користувач);

– забезпечено обробку сенсорних даних у реальному часі для динамічного керування освітленням;

– впроваджено механізми резервного копіювання та відновлення даних;

– використано Maven для керування залежностями, збірки;

Інтеграція з пристроями:

– прийом та збереження показників від сенсорів (вологість, температура тощо);

– взаємодія із джерелами світла для зміни яскравості та спектру згідно з отриманими параметрами;

1.4 Посилання

В проєкті були використані такі посилання:

– https://www.geeksforgeeks.org/spring-boot/

– https://swagger.io/resources/articles/best-practices-in-api-design/

– https://www.baeldung.com/swagger-2-documentation-for-spring-rest-api

1.5 Означення та абревіатури

Користувач – особа, яка взаємодіє з програмною системою через веб-інтерфейс або пристрої введення даних (сенсори). Може мати одну з ролей: адміністратор, технічний персонал або звичайний користувач.

Сервер – програмна система, що обробляє запити, виконує бізнес-логіку системи та зберігає інформацію у базі даних.

Сенсор – пристрій, що зчитує параметри навколишнього середовища (вологість, температура, освітленість) і надсилає їх до серверної частини системи.

Джерело світла або Лампа – електронний пристрій, який забезпечує освітлення для рослин. Може керуватися системою автоматично на основі показників сенсорів.

API (Application Programming Interface) – інтерфейс програмування застосунків, через який інші системи та пристрої взаємодіють із сервером.

CRUD (Create, Read, Update, Delete) – базові операції для створення, зчитування, оновлення та видалення даних у системі.

CSV (Comma-Separated Values) – формат табличного зберігання та обміну даними, що використовується для імпорту та експорту користувачів у системі.

REST (Representational State Transfer) – архітектурний стиль для розробки веб-сервісів, використаний у створенні серверного API.

DB (Database) – база даних PostgreSQL, яка зберігає всю інформацію про користувачів, рослини, сенсори та зчитування.

Spring Boot – фреймворк Java для створення серверних застосунків, використаний у розробці серверної частини системи та IoT симулятору.

Maven – система управління проєктами, що використовується для автоматизації білдів, тестування й залежностей у проєкті.

2 ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Основні перспективи продукту:

– розширення функціональності системи шляхом покращення можливостей сенсорів;

– покращення алгоритмів обробки зчитувань та відповідної зміни освітлення;

– додавання інших пристроїв, що будуть покращувати зростання рослин, як, до прикладу, пристрої поливу;

– створення мобільного застосунку для зручного моніторингу й керування системою в реальному часі;

– забезпечення підтримки хмарної синхронізації для масштабування на великі агропромислові об’єкти;

2.2 Функції продукту

Користувачі матимуть такі функції:

– реєстрація;

– аутентифікація;

– перегляд усіх користувачів;

– перегляд усіх ролей;

– надавання, або скасування ролі користувачу;

– експорт та імпорт даних користувачів;

– управління рослинами, CRUD операції;

– управління сенсорами, CRUD операції;

– управління освітлювальними пристроями, CRUD операції;

– перегляд зчитувань та попереджень;

2.3 Характеристики користувачів

2.3.1 Адміністратори

– відповідають за управління користувачами та ролями;

– добре орієнтуються в налаштуваннях системи та безпеці;

– володіють базовими або розширеними технічними знаннями;

– працюють із CSV-експортом/імпортом даних;

2.3.2 Технічний персонал

– працюють із рослинами, сенсорами та освітленням;

– мають базові комп’ютерні навички;

– залучені до внесення даних, моніторингу та обслуговування обладнання;

– очікують простого та зрозумілого інтерфейсу.

2.3.3 Кінцеві користувачі (агровиробники або фермери)

– використовують систему для моніторингу стану рослин;

– цікавляться простими візуалізаціями та попередженнями;

– можуть не мати технічних вмінь, але зацікавлені в підвищенні ефективності виробництва;

– потребують інтуїтивно зрозумілого функціоналу та мобільності;

2.4 Загальні обмеження

На даному етапі присутні такі обмеження:

– присутній веб-інтерфейс, мобільного застосунку наразі не передбачено;

– доступ до різних ресурсів доступний користувачам лише при наявності необхідної ролі;

– присутні українська та англійська мови;

2.5 Припущення й залежності

На даному етапі припущення та залежності такі:

– система потребує локальної мережі, щоб передавати дані з сенсорів;

– серверна частина має бути розгорнута на системі із встановленою Java 17 та PostgreSQL;

– додаткові підключені пристрої мають передавати дані у форматі, сумісному з розробленим REST API;

– автоматичне керування освітленням залежить від наявності актуальних даних від сенсорів;

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

Користувач має такий інтерфейс:

– взаємодія з користувачем: Swagger або Postman;

– доступні ресурси: реєстрація, автентифікація, рослини, сенсори, освітлювальні пристрої, зчитування, користувачі;

– простий інтерфейс, небагато вхідних параметрів;

3.1.2 Апаратний інтерфейс

Вимоги до апаратного інтерфейсу:

– підключення до локальної мережі;

– 1 ГБ оперативної пам’яті;

– 500 МБ вільного місця для зберігання даних;

3.1.3 Програмний інтерфейс

Серверна частина:

– технології: Java 17, Spring Boot 3, PostgreSQL;

– API: RESTful API для обміну даними між сервером, клієнтами та сенсорами;

3.1.4 Комунікаційний протокол

Для безпечної передачі даних між сенсорами, клієнтською та серверною частинами використовуватиметься протокол HTTPS.

Для управління сесіями користувачів та забезпечення контролю доступу використовується JWT. Це дозволяє реалізувати безпечну авторизацію з можливістю масштабування.

3.1.5 Обмеження пам’яті

Для коректної роботи програмного забезпечення діють такі обмеження:

– 500 МБ вільного місця на диску для встановлення програми та збереження даних;

– використовується реляційна база даних для зручної взаємодії різних об’єктів між собою;

3.1.6 Операції

Основні операції програмної системи:

– створення нового облікового запису користувача;

– вхід до системи з перевіркою даних (авторизація);

– можливість перегляду списку всіх зареєстрованих користувачів;

– доступ до інформації про всі наявні ролі в системі;

– призначення або видалення ролей у користувачів;

– можливість збереження інформації про користувачів у файл (експорт) або завантаження з файлу в систему (імпорт);

– додавання, редагування, перегляд і видалення інформації про рослини;

– повне керування даними сенсорів (додавання, оновлення, перегляд, видалення);

– налаштування параметрів освітлювальних пристроїв: їх створення, оновлення, видалення та перегляд;

– перегляд зчитувань, отриманих від сенсорів, та повідомлень про потенційні проблеми з рослинами;

3.1.7 Функції продукту

Програмне забезпечення має такі функції:

– регулювання освітлення;

– реєстрація та авторизація користувачів;

– управління користувачами з панелі адміністратора;

– моніторинг зчитувань з сенсорів;

– управління об’єктами програмної системи (сенсори, рослини, освітлювальні пристрої);

3.1.8 Припущення й залежності

Припущення:

– користувачі (фермери) мають базові навички роботи з веб-застосунками;

– усі сенсори та освітлювальні пристрої правильно під’єднані та знаходяться в робочому стані;

– дані, що надходять від сенсорів, є коректними та не містять суттєвих похибок;

Залежності:

– залежність від безперебійної роботи серверної частини, що відповідає за зберігання, обробку даних та логіку управління;

– залежність від точності показників, отриманих із сенсорів (вологість, температура, освітлення);

– залежність від енергопостачання для функціонування сенсорів, освітлення та серверної частини;

– залежність від зовнішніх бібліотек та фреймворків (Spring Boot, PostgreSQL тощо) для роботи системи;

– залежність від механізмів безпеки (JWT, HTTPS) для захисту конфіденційної інформації користувачів;

3.2 Властивості програмного продукту

Програмний продукт для автоматизованого вирощування рослин на плантаціях та фермах забезпечує автоматизований моніторинг і керування умов росту рослин, орієнтуючись на показники з сенсорів.

Основні властивості продукту:

– користувачі можуть додавати та редагувати інформацію про рослини, включаючи відсоток росту та вимоги до освітлення;

– система зчитує показники з сенсорів у реальному часі (температура, вологість, освітленість тощо) та реагує на зміни;

– автоматичне керування освітленням на основі отриманих кліматичних даних та встановлених параметрів для конкретних рослин;

– можливість перегляду зчитування параметрів середовища та попереджень про відхилення від норми;

– керування сенсорами та освітлювальними пристроями (додавання, редагування, видалення);

– підтримка багаторівневої рольової системи з можливістю призначення та скасування ролей користувачам;

– експорт та імпорт даних користувачів;

– захищений доступ до системи через реєстрацію та автентифікацію з використанням імені користувача і пароля;

3.3 Атрибути програмного продукту

3.3.1 Надійність

Ціль – система має стабільно працювати і видавати результати у режисі реального часу, обробляючи дані сенсорів і виконувати дії без збоїв.

Метрика – час простою не повинен перевищувати 0.1% на місяць при стандартному навантаженні.

3.3.2 Доступність

Ціль – програмна система повинна бути доступною користувачам в будь-який час доби, включно з обробкою даних з сенсорів та можливістю віддаленого управління пристроями.

Метрики:

– доступність системи має становити не менше 99.9% на місяць;

– система має бути доступною з будь-якого пристрою, що підключений до локальної мережі;

3.3.3 Безпека

Ціль – захистити дані програмної системи, такі як дані користувачів, параметри рослин, налаштування пристроїв, історію зчитувань.

Метрики:

– паролі мають бути захешовані;

– автентифікація відбувається з використанням JWT;

– всі попередження мають зберігатися у системі;

3.3.4 Супроводжуваність

Ціль – система повинна бути легкою для оновлення, тестування і масштабування.

Метрики:

– код має бути достатньо чистим та легким в розумінні;

– стандартизоване оформлення коду;

3.3.5 Переносимість

Ціль – можливість розгортання серверної частини на різних середовищах (локальних або хмарних).

Метрики:

– серверна частина побудована з використанням кросплатформеного стеку технологій;

– мінімальна кількість залежностей від специфічного середовища;

3.3.6 Продуктивність

Ціль – обробка вхідних даних та прийняття рішень щодо освітлення повинні виконуватись із мінімальною затримкою.

Метрики:

– час реакції системи на зміну сенсорних даних – не більше 1 секунди;

– підтримка обробки не менше 1000 запитів на хвилину у серверній частині;

3.4 Вимоги бази даних

Тип бази даних – реляційна база даних PostgreSQL.

Призначення – зберігання структурованих даних системи, зокрема:

– облікові записи користувачів і їхні ролі;
– інформація про рослини (стадії росту, потреби);
– дані сенсорів (вологість, температура, освітлення тощо);
– параметри освітлювальних пристроїв;
– зчитування;

3.5 Інші вимоги

Інші вимоги включають:
– підтримка локалізації української та англійської мов;