

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
магістра

(назва освітньо-кваліфікаційного рівня)

студента	<i>Юдіна Олександра Сергійовича</i>		
	(ПІБ)		
академічної групи	<i>121М-22-3</i>		
	(шифр)		
спеціальності	<i>121 Інженерія програмного забезпечення</i>		
	(код і назва спеціальності)		
освітньої програми	<i>«121 Інженерія програмного забезпечення»</i>		
	(назва освітньої програми)		
на тему:	<i>Розробка та дослідження веб-інтерфейсу системи моніторингу мікроклімату промислових теплиць</i>		

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинг овою	інституційною	
розділів кваліфікаційної роботи				
спеціальний	<i>проф. Лактіонов І.С.</i>			
Рецензент				
Нормоконтролер	<i>проф. Лактіонов І. С.</i>			

Дніпро 2023

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

Завідувач кафедри

Програмного забезпечення комп'ютерних систем

(повна назва)

М.О. Алексєєв

(підпис)

(прізвище, ініціали)

« »

20 23 Року

ЗАВДАННЯ

на виконання кваліфікаційної роботи

спеціальності

121 Інженерія програмного забезпечення

(код і назва спеціальності)

студенту

121М-22-3

Юдіну Олександр Сергійовичу

(група)

(прізвище та ініціали)

Тема кваліфікаційної роботи

Розробка та дослідження веб-інтерфейсу

системи моніторингу мікроклімату промислових теплиць

ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від 09.10.2023р. № 1227 -с

МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – процес автоматизації системи моніторингу мікроклімату

промислових теплиць на основі платформи .NET.

Предмет досліджень – моделі, методи та інформаційні технології серверної та UI складових для забезпечення контролю мікроклімату промислових теплиць.

Мета НДР – підвищення якості та надійності контролю мікроклімату промислових теплиць.

Вихідні дані для проведення роботи – теоретичні та експериментальні дослідження для вирішення задач проектування інформаційної системи застосунку та підходи до вибору технологій реалізації.

ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Новизна запропонованих рішень полягає в розробці та застосуванні сучасних технологій програмного забезпечення для створення надійного додатку за рахунок використання ефективних патернів проектування.

Практична цінність результатів полягає у тому, що отримані в ході дослідження результати роботи можуть застосовуватися як для інтеграції з існуючими рішеннями так і окремим проектом.

ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень повинні бути подані у вигляді, який дозволяє побачити та оцінити безпосереднє використання технологій серверної та UI сторін розробки. В результаті роботи повинен бути розроблений програмний комплекс для візуалізації отримання даних в реальному часі.

ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Аналіз теми та постановка задачі	10.09.2023-24.09.2023
Збір, дослідження та систематизація інформації щодо проектування та реалізації веб інтерфейсу контролю мікроклімату теплиць	25.09.2023-05.10.2023
Розробка і тестування системи для контролю мікроклімату промислових теплиць	06.10.2023-10.12.2023

Завдання видав

(підпис)

Лактіонов І. С.

(прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Юдін О.С.

(прізвище, ініціали)

Дата видачі завдання: 15.09.2023 р.

Термін подання кваліфікаційної роботи до ЕК 16.12.2023 р.

РЕФЕРАТ

Пояснювальна записка: 84 с., 25 рис., 5 дод., 32 джерел.

Об'єкт розробки: процес автоматизації системи моніторингу мікроклімату промислових теплиць на основі платформи .NET.

Предмет дослідження: методи, схеми та засоби автоматизації системи моніторингу мікроклімату промислових теплиць.

Мета кваліфікаційної роботи: підвищення якості та надійності контролю мікроклімату промислових теплиць.

Методи дослідження.

Наукова новизна полягає в створенні надійного додатку на основі новітніх технологій та патернів проектування.

Практичне значення роботи. Отримані в ході дослідження результати роботи можуть застосовуватися як для інтеграції з існуючими рішеннями так і окремим проектом.

Ключові слова: REST API, веб-застосунок, фреймворк, клієнт, сервер, інтерфейс, SQL, http-запит, http-відповідь, SignalR, CQRS, Blazor, .NET.

ABSTRACT

Explanatory Note: 84 pages, 25 figures, 5 appendices, 32 sources.

Development Object: The process of automating the microclimate monitoring system of industrial greenhouses based on the .NET platform.

Research Subject: Methods, schemes, and means of automating the microclimate monitoring system of industrial greenhouses.

Purpose of the Qualification Work: To improve the quality and reliability of microclimate control in industrial greenhouses.

Research Methods: To accomplish the set tasks, the following methods were utilized: object-oriented programming, design patterns, a system for managing object storage in relational database tables, HTML5 template mechanisms.

Scientific Novelty lies in the creation of a reliable application based on modern technologies and design patterns.

Practical Significance of the Work: The results obtained during the research can be applied both for integration with existing solutions and as a standalone project.

Keywords: REST API, web application, framework, client, server, interface, SQL, HTTP request, HTTP response, SignalR, CQRS, Blazor, .NET.

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних;

SQL – Structured Query Language;

API – Application Programming Interface;

HTTP – Hypertext Transfer Protocol;

HTML – HyperText Markup Language;

SCADA – Supervisory Control and Data Acquisition;

MSSQL – Microsoft SQL Server;

SSMS – SQL Server Management Studio;

CSV – Comma-separated values;

BE – Back End;

UI – User Interface;

CQRS – Command and Query Responsibility Segregation;

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1 ОГЛЯД СИСТЕМ МОНІТОРИНГУ МІКРОКЛІМАТУ ПРОМИСЛОВИХ ТЕПЛИЦЬ	13
1.1. Загальні відомості.....	13
1.2. Аналіз сучасних веб-засобів моніторингу кліматичних параметрів.....	16
1.3. Аналіз техніко-функціональних характеристик систем-аналогів.....	21
1.3.1. Автоматизовані системи моніторингу теплиць Appvales.....	21
1.3.2. Автоматизована теплична система Intellias	25
1.3.3. Система моніторингу теплиць Webhmi	27
1.3.4. Система контролю клімату Priva	31
1.4. Висновки за розділом	33
РОЗДІЛ 2 МАТЕРІАЛИ, МЕТОДИ І МОДЕЛІ ДОСЛІДЖЕНЬ	35
2.1. Узагальнена структурно-функціональна організація системи	35
2.1.1. Оновлення даних в реальному часі	35
2.1.2. Збереження даних до бази даних	37
2.1.3. Зчитування вхідних даних.....	40
2.2. Використані методи досліджень	42
2.3. Інформаційна модель досліджуваної системи.....	45
2.3.1. Загальна модель комунікації компонентів	45
2.3.2. Обробка вхідних даних.....	46
2.3.3. Зберігання даних.....	47
2.3.4. Оновлення даних	49
2.4. Опис алгоритмів роботи досліджуваної системи	51
2.4.1. Масштабованість.....	51
2.4.2. Надійність	53
2.4.3. Ефективність.....	53
2.5. Висновки за розділом	55

РОЗДІЛ 3 РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ.....	57
3.1. Опис розробленої системи програмного забезпечення	57
3.1.1. Серверна частина	57
3.1.2. Клієнтська частина	59
3.2. Результати тестування розробленої системи програмного забезпечення 61	
3.3. Перспективні напрямки подальших досліджень	65
3.3.1. Покращення існуючого функціоналу	65
3.3.2. Впровадження нових функціональних можливостей	65
3.4. Порівняльний аналіз отриманих результатів із раніше відомими	66
3.5. Висновки за розділом	67
ВИСНОВКИ	69
Список літератури.....	70
ДОДАТОК А.....	74
ВИХІДНИЙ КОД ЗАВАНТАЖЕННЯ ФАЙЛІВ	74
ДОДАТОК Б.....	77
ОТРИМАННЯ ДАНИХ З БАЗИ ДАНИХ	77
ДОДАТОК В.....	79
ООНОВЛЕННЯ ДАНИХ В РЕАЛЬНОМУ ЧАСІ.....	79
ДОДАТОК Г	82
ПРОГРАМНИЙ КОД СТОРІНКИ ЗАВАНТАЖЕННЯ ФАЙЛУ	82
ДОДАТОК Д.....	84
ПЕРЕЛІК ФАЙЛІВ НА ОПТИЧНОМУ НОСІЇ.....	84

ВСТУП

Актуальність дослідження. Мікроклімат у промислових теплицях є одним з найважливіших факторів, що впливають на врожайність та якість продукції. Оптимальний мікроклімат забезпечує рослинам комфортні умови для росту та розвитку, що дозволяє їм повноцінно реалізувати свій потенціал.

Системи моніторингу мікроклімату дозволяють автоматизувати процес контролю та управління мікрокліматом у теплицях, що дозволяє значно підвищити ефективність управління тепличним господарством, а також зменшити витрати на енергоносії.

В сучасних умовах розвитку сільського господарства важливо вдосконалювати технології вирощування, зокрема управління мікрокліматом у промислових теплицях. Розробка веб-інтерфейсу системи моніторингу цих умов є актуальною з огляду на кілька ключових аспектів.

По-перше, ефективне управління тепличними умовами дозволяє сільськогосподарським підприємствам оперативно реагувати на зміни середовища та оптимізувати параметри мікроклімату. Це має безпосередній вплив на зростання врожайності, якість продукції і зниження витрат ресурсів.

По-друге, забезпечення оптимальних умов вирощування рослин у теплицях є ключовим для економії енергії та води. Система моніторингу дозволяє точно налаштовувати параметри освітлення і поливу, зменшуючи таким чином витрати енергії і води.

По-третє, в умовах зростаючого екологічного тиску та необхідності стійкості до змін клімату, важливо впроваджувати технології, спрямовані на раціональне використання ресурсів.

Отже, розробка веб-інтерфейсу для системи моніторингу мікроклімату промислових теплиць не тільки сприяє підвищенню ефективності вирощування, але й відповідає викликам сучасного сільського господарства, спрямованого на сталість та оптимізацію ресурсного використання.

У кваліфікаційній роботі актуальність розробки веб-інтерфейсу для моніторингу мікроклімату вирішується шляхом створення інтуїтивно зрозумілого та функціонального інтерфейсу для фермерів. Система забезпечує надання даних в режимі реального часу та використання автоматизованих можливостей для оптимізації параметрів мікроклімату.

Мета дослідження: підвищення якості та надійності контролю мікроклімату промислових теплиць.

Завдання дослідження: для досягнення поставленої мети в роботі були сформульовані та вирішені наступні завдання:

1. Створення інтуїтивно зрозумілого та функціонального веб-інтерфейсу для системи моніторингу мікроклімату в теплицях;
2. Розробка та впровадження системи моніторингу, що здійснює збір та аналіз даних про умови в теплицях в реальному часі;
3. Проведення тестувань розробленого веб-інтерфейсу та системи моніторингу для виявлення можливих недоліків та оптимізації їхньої роботи;
4. Проведення аналізу отриманих результатів для визначення ефективності та можливості подальшого вдосконалення системи;

Об'єкт дослідження: процес автоматизації системи моніторингу мікроклімату промислових теплиць на основі платформи .NET.

Предмет дослідження: методи, схеми та засоби автоматизації системи моніторингу мікроклімату промислових теплиць

Методи дослідження. Для виконання поставлених завдань були використані методи: об'єктно-орієнтоване програмування, патерни проектування, система

управління збереженням об'єктів у таблиці реляційних баз даних, механізм шаблонів HTML5.

Наукова новизна полягає в створенні надійного додатку на основі новітніх технологій та патернів проектування.

Практичне значення. Проведене дослідження буде корисним для фермерів та агротехніків, щоб ефективніше контролювати та оптимізувати умови вирощування рослин у теплицях.

Результати досліджень можуть знайти застосування в сільському господарстві для оптимізації вирощування рослин та підвищення врожайності. Агропромислові підприємства можуть впроваджувати розроблені технології для підвищення ефективності та економії ресурсів.

Особистий внесок автора:

1. Наукові результати роботи отримані автором самостійно.
2. Вибір методів досліджень і технологій реалізації;
3. Проектування і розробка веб-інтерфейсу системи моніторингу;
4. Програмна реалізація системи моніторингу мікроклімату теплиць;
5. Оцінка отриманих результатів.

Структура і обсяг роботи. Робота складається з вступу, трьох розділів і висновків. Містить 84 сторінок, в тому числі 60 сторінок тексту основної частини з 25 рисунками, списку використаних джерел з 32 найменуваннями на 4 сторінках, 5 додатків на 11 сторінках.

РОЗДІЛ 1

ОГЛЯД СИСТЕМ МОНІТОРИНГУ МІКРОКЛІМАТУ ПРОМИСЛОВИХ ТЕПЛИЦЬ

1.1. Загальні відомості

Мікроклімат теплиці - це сукупність фізичних і біологічних умов, що впливають на ріст і розвиток рослин у теплиці. До основних компонентів мікроклімату теплиці відносяться [1]:

- Температура - це найважливіший фактор мікроклімату теплиці. Оптимальна температура для росту і розвитку різних рослин становить від 15 до 25 градусів Цельсія.
- Вологість - Оптимальна вологість для росту і розвитку різних рослин становить від 50 до 70 відсотків.
- Освітлення - використовується для фотосинтезу рослин. Оптимальна освітленість для росту і розвитку різних рослин становить від 500 до 1000 люксів.
- Вміст вуглекислого газу - вміст вуглекислого газу також є важливим фактором для фотосинтезу рослин. Оптимальний вміст вуглекислого газу для росту і розвитку різних рослин становить від 300 до 500 ppm.

Промислові теплиці - це великі закриті споруди, призначені для вирощування рослин у промислових масштабах. Промислові теплиці забезпечують оптимальні умови для росту і розвитку рослин, що дозволяє отримувати високі врожаї.

Системи моніторингу мікроклімату теплиць - це системи, які призначені для контролю і реєстрації параметрів мікроклімату теплиці. Системи моніторингу мікроклімату теплиць використовуються для забезпечення оптимальних умов для росту і розвитку рослин, а також для виявлення проблем з мікрокліматом теплиці.

Системи моніторингу мікроклімату теплиць є невід'ємною частиною сучасного сільськогосподарського виробництва, спрямованого на оптимізацію умов вирощування рослин. Вони складаються з ряду ключових компонентів, кожен з яких виконує важливі функції для забезпечення оптимального мікроклімату в теплиці [2]:

- Датчики мікроклімату - датчики вимірюють різноманітні параметри мікроклімату в теплиці, такі як температура, вологість, освітленість і інші. Ці дані є основою для подальшого аналізу та прийняття рішень щодо регулювання умов вирощування рослин.
- Контролер - контролер виконує функцію обробки і аналізу отриманих даних від датчиків. Він відповідає за управління різними системами теплиці, такими як система поливу, система освітлення та інші, з метою створення оптимального середовища для росту рослин.
- Інтерфейс користувача - інтерфейс користувача надає можливість фермерам та операторам теплиці отримувати доступ до даних з датчиків мікроклімату та взаємодіяти з системами управління. Це може бути веб-інтерфейс, мобільний додаток або інший зручний спосіб взаємодії.
- Система моніторингу мікроклімату для промислових теплиць – Ця технологія є важливим інструментом для керування та контролювання умов навколишнього середовища в промислових теплицях. Забезпечення оптимальних параметрів мікроклімату допомагає підтримувати врожайність і якість продукції на високому рівні.

Промислові теплиці використовуються для вирощування рослин з метою забезпечення оптимальних умов для їхнього росту та розвитку. Мікроклімат у теплицях є ключовим фактором, який впливає на врожайність та якість продукції.

На рис.1.1 представлено оснащення промислової теплиці.



Рис. 1.1. Оснащення промислової теплиці

Системи моніторингу мікроклімату теплиць можуть бути використані для таких цілей [3]:

- Контроль мікроклімату теплиці - системи моніторингу мікроклімату теплиць дозволяють контролювати параметри мікроклімату теплиці і забезпечувати їх оптимальні значення.
- Виявлення проблем з мікрокліматом теплиці - системи моніторингу мікроклімату теплиць дозволяють виявляти проблеми з мікрокліматом теплиці, такі як перепади температури, вологості або освітлення.
- Аналіз мікроклімату теплиці - системи моніторингу мікроклімату теплиць можуть використовуватися для аналізу мікроклімату теплиці, наприклад, для визначення впливу різних факторів на ріст і розвиток рослин.
- Системи моніторингу мікроклімату теплиць є важливим інструментом для забезпечення оптимальних умов для росту і розвитку рослин у промислових теплицях.

Використання систем моніторингу мікроклімату мають такі вигоди:

- Підвищення врожайності: Оптимальне управління мікрокліматом сприяє підвищенню врожайності та якості продукції.
- Енергозбереження: Автоматичні системи дозволяють ефективно використовувати енергію для утримання оптимальних умов.
- Мінімізація ризиків: Моніторинг дозволяє швидко виявляти та вирішувати проблеми, такі як хвороби чи зміни в екологічних умовах.
- Автоматизація процесів: Системи моніторингу можуть бути пов'язані з автоматичними системами управління, що забезпечує повну автоматизацію тепличного господарства.

У сфері управління мікрокліматом промислових теплиць методи, що використовуються для моніторингу та контролю, відіграють ключову роль у визначенні успіху сільського господарства. Виділяються три основних підходи моніторингу мікроклімату:

- Мануальний
- За допомогою веб-інтерфейсу
- Автоматизований

1.2. Аналіз сучасних веб-засобів моніторингу кліматичних параметрів

Веб-засоби моніторингу кліматичних параметрів промислових теплиць можна класифікувати за такими критеріями [4]:

За способом отримання даних:

- Автономні:
 - Працюють без підключення до Інтернету.
 - Мають власні сенсори для отримання даних і власні пристрої для їх зберігання і обробки.

- Дані з цих пристроїв можна переглядати на локальних пристроях, таких як планшети або комп'ютери.
- Централізовані:
 - Працюють з датчиками, розташованими в різних теплицях.
 - Дані з цих датчиків передаються в централізований сервер, де вони зберігаються і обробляються.
 - Дані з центру можна переглядати в Інтернеті з будь-якого пристрою, підключеного до мережі.

За способом передачі даних:

- Безпроводні - використовують бездротові технології, такі як Wi-Fi або Bluetooth, для передачі даних. Це дозволяє їм бути більш мобільними, ніж дротові системи.
- Дротові - використовують дротові технології, такі як Ethernet або RS-485, для передачі даних, що дозволяє їм бути більш надійними, ніж бездротові системи.
- За способом відображення даних:
 - На локальних пристроях - дані з пристроїв моніторингу можна переглядати на локальних пристроях, таких як планшети або комп'ютери.
 - В Інтернеті - дані з пристроїв моніторингу можна переглядати в Інтернеті з будь-якого пристрою, підключеного до мережі.

Веб-засоби моніторингу кліматичних параметрів промислових теплиць мають ряд переваг, у тому числі:

- Покращення контролю - користувачі можуть відстежувати умови в теплицях в режимі реального часу. Це дозволяє їм своєчасно реагувати на зміни і підтримувати оптимальні умови для росту і розвитку рослин.
- Збільшення ефективності – оптимізація використання ресурсів, таких як енергія і вода. Це може привести до зниження витрат і підвищення продуктивності.
- Збільшення прибутку – отримання врожаю вищої якості, що призводить до підвищення ціни на продукцію і збільшення прибутку.

Мікрокліматичний моніторинг теплиць дозволяє відстежувати різні параметри мікроклімату, такі як температура, вологість, освітленість, вміст діоксиду вуглецю та інші. Ці дані можуть бути використані для управління теплицями, оптимізації використання ресурсів та підвищення продуктивності.

Сучасний ринок пропонує комплексні рішення для автоматизації всіх основних процесів у промислових теплицях великої площі: опалення, вентиляції, контролю вологості, поливу, контролю вмісту в атмосфері вуглекислоти та підживлення рослин мінеральними добривами.

На рис.1.2 приведено приклад комплексного рішення, що забезпечує автоматизацію основних кліматичних параметрів.

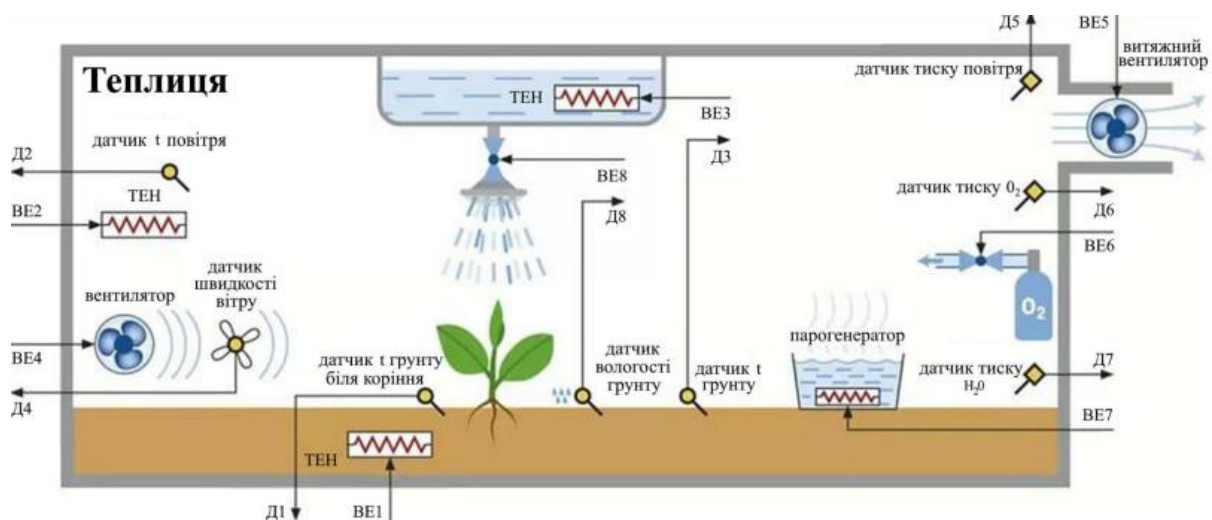


Рис. 1.2. Приклад комплексного рішення, що забезпечує автоматизацію основних кліматичних параметрів

Обробка даних мікрокліматичного моніторингу є важливим етапом, який дозволяє отримати корисну інформацію з цих даних. Існує кілька підходів до обробки даних мікрокліматичного моніторингу, які можна розділити на дві основні групи:

- Статистична обробка

– Моделювання.

Статистична обробка дозволяє описати дані мікроклімату за допомогою статистичних величин, таких як середнє значення, дисперсія, медіана, мода та інші. Ці величини можуть бути використані для оцінки поточних умов в теплиці, а також для виявлення тенденцій і закономірностей в зміні даних. Наприклад, за допомогою статистичної обробки можна визначити середню температуру в теплиці протягом дня або тижня. Це може бути використано для контролю температури в теплиці та забезпечення оптимальних умов для росту і розвитку рослин.

Моделювання дозволяє створювати математичні моделі мікроклімату теплиць [5]. Ці моделі можуть бути використані для прогнозування змін мікроклімату, а також для управління теплицями. Наприклад, можна створити модель, яка прогнозує температуру в теплиці в залежності від погодних умов. Ця модель може бути використана для автоматичного регулювання температури в теплиці.

Вибір підходу до обробки даних залежить від конкретних завдань, які потрібно вирішити. Якщо потрібно описати дані мікроклімату, то можна використовувати статистичну обробку. Якщо потрібно прогнозувати зміни мікроклімату або управляти теплицями, то можна використовувати моделювання.

У додаток до статистичної обробки та моделювання, існують інші підходи до обробки даних мікрокліматичного моніторингу, які можна успішно використовувати. Зокрема, це включає аналіз трендів, аналіз часових рядів та використання методів машинного навчання. Ці підходи дозволяють отримати глибше розуміння даних мікроклімату та використовувати їх для вирішення більш складних завдань, таких як передбачення змін у мікрокліматі чи виявлення небезпечних тенденцій. Ці підходи дозволяють отримувати більш глибоке розуміння даних мікроклімату та використовувати їх для вирішення більш складних завдань.

Обробка даних мікрокліматичного моніторингу є важливим етапом, який дозволяє отримати корисну інформацію з цих даних. Вибір підходу до обробки даних залежить від конкретних завдань, які потрібно вирішити.

Обробка даних мікрокліматичного моніторингу є важливим етапом, який дозволяє отримати корисну інформацію з цих даних. Вибір підходу до обробки даних залежить від конкретних завдань, які потрібно вирішити.

Функціональність веб-інтерфейсу моніторингу кліматичних параметрів є важливим фактором, який впливає на ефективність використання таких систем. Веб-інтерфейс повинен бути простим у використанні і інтуїтивно зрозумілим, щоб користувачі могли легко отримувати доступ до інформації та керувати системою.

Основні функції веб-інтерфейсу моніторингу кліматичних параметрів включають [6]:

- Перегляд даних в реальному часі - можливість переглядати поточні значення параметрів мікроклімату в теплиці.
- Відображення даних в історичному вигляді. перегляд даних про мікроклімат за певний період часу.
- Створення звітів - створення звітів про отримані дані мікроклімату теплиці.
- Налаштування сповіщень - отримання оповіщення про відхилення від заданих параметрів мікроклімату.

Крім цих основних функцій, веб-інтерфейс моніторингу кліматичних параметрів може включати також такі додаткові функції:

- Аналіз даних - аналіз даних про мікроклімат за допомогою таких інструментів, як статистична обробка, моделювання та машинне навчання.
- Управління обладнанням теплиць - можливість керувати обладнанням теплиць, таким як обігрівальні системи, кондиціонери, поливальні системи.

- Інтеграція з іншими системами - веб-інтерфейс повинен дозволяти інтегрувати систему моніторингу мікроклімату з іншими системами, такими як системи управління теплицями, системи управління виробництвом та інші.

Розширення функціональності веб-інтерфейсу моніторингу кліматичних параметрів є перспективним напрямком розвитку. Це дозволить фермерам отримувати більше користі від використання таких систем і підвищити ефективність виробництва. Додаткові можливості, такі як аналіз даних та управління обладнанням, роблять систему більш гнучкою і адаптованою до різних потреб користувачів. Це не лише сприятиме отриманню більшої користі від використання системи, але й підвищить ефективність виробництва в сфері сільського господарства.

1.3. Аналіз техніко-функціональних характеристик систем-аналогів

1.3.1. Автоматизовані системи моніторингу теплиць Appvales

Appvales - це передова система моніторингу теплиць, яка впроваджує комплексні технології для автоматизації контролю та управління тепличним господарством. Це інноваційне рішення спрямоване на забезпечення ефективного вирощування рослин шляхом оптимізації умов середовища в теплиці.

Проект був побудований для контролю загального стану в групі теплиць. Вбудована система складалася з 4 груп датчиків, які спілкувалися з хабом теплиці, звідки інформація надсилалася до API (інтерфейс прикладного програмування). Одна група датчиків контролювала вологість, температуру та знала про небажану вологість. Чотири датчики створювали віртуальну кімнату, яка контролювалася хабом і API. API відповідав за віртуальні кімнати (кожна кімната вказує на окрему

теплицю), мав інформацію від 4 датчиків і міг віддалено регулювати вологість і температуру за допомогою дистанційного керування віртуальною кімнатою [7].

На рис.1.3 показано сторінки головного екрану додатку.

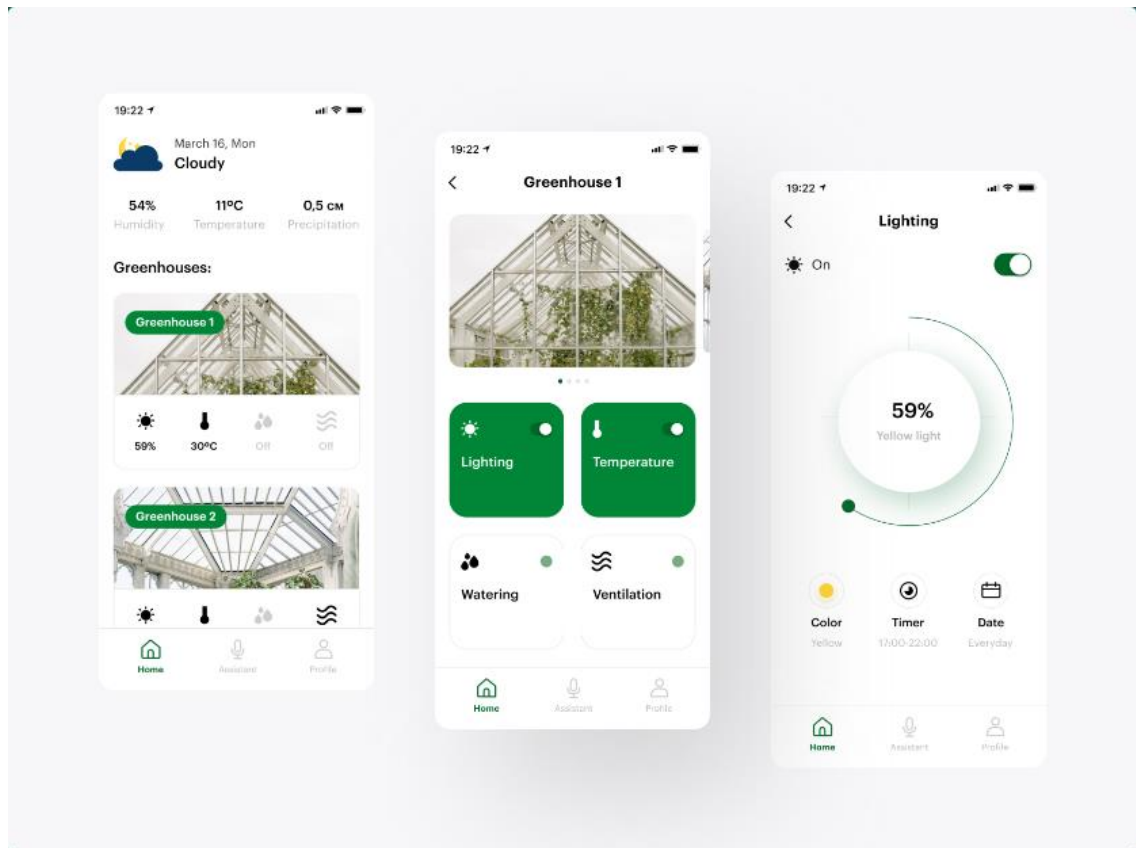


Рис. 1.3. Сторінки додатку Appvales

Проект пропонує наступні функції:

- Захист рослин від екстремальних температур
- Збільшення врожайності
- Автоматичне регулювання атмосферних умов
- Аналіз у веб-або мобільному додатку
- Запобігає крадіжкам і підвищує безпеку.

Система складається з наступних компонентів:

- Контролери - це основні пристрої системи, які збирають дані з датчиків і передають їх на сервер. Контролери можуть бути встановлені на будь-якій відстані від сервера, оскільки вони працюють на бездротовому зв'язку.
- Датчики - Appvales використовує передові датчики для вимірювання різноманітних параметрів мікроклімату. Ці дані надходять в реальному часі, що дозволяє операторам теплиці миттєво реагувати на зміни та вносити необхідні корективи. До датчиків, які використовуються в системі Appvales, відносяться:
 - Датчики температури та вологості повітря
 - Датчики температури та вологості ґрунту
 - Датчики освітленості
 - Датчики CO₂
 - Датчики вологості листя
- Сервер - це центральний вузол системи, який обробляє дані з контролерів і датчиків. Сервер також дозволяє користувачам отримувати доступ до даних та керувати системою.

Система Appvales дозволяє контролювати такі параметри тепличного господарства:

- Температура та вологість повітря - ці параметри є критичними для росту рослин. Система Appvales дозволяє підтримувати оптимальні умови температури та вологості повітря в теплиці.
- Температура та вологість ґрунту - ці параметри також важливі для росту рослин. Система Appvales дозволяє контролювати вологість ґрунту, щоб запобігти пересиханню або перезволоженню.
- Освітленість - освітлення є важливим фактором для фотосинтезу рослин. Система Appvales дозволяє контролювати освітленість у теплиці, щоб забезпечити рослинам оптимальний світловий режим.

- CO₂ - вуглекислий газ є компонентом для фотосинтезу. Система Appvales дозволяє контролювати рівень CO₂ у теплиці, щоб забезпечити рослинам оптимальний вміст цього газу.
- Вологість листя - вологість листя є важливим показником здоров'я рослин. Система Appvales дозволяє контролювати вологість листя, щоб запобігти зневодненню або надмірному зволоженню рослин.

Система Appvales має ряд переваг, які роблять її цінним інструментом для тепличників:

- Автоматизація - система Appvales дозволяє автоматизувати контроль та управління тепличним господарством, що звільняє час і ресурси для інших завдань.
- Ефективність - система Appvales дозволяє підвищити ефективність роботи тепличного господарства, забезпечивши оптимальні умови для росту рослин.
- Зручність - система Appvales проста у використанні та дозволяє користувачам отримувати доступ до даних і керувати системою з будь-якого місця.

Система Appvales може використовуватися в теплицях різного розміру та призначення. Система є економічно доступною і може бути встановлена та налаштована в короткий термін.

Система Appvales дозволяє:

- підтримувати оптимальні умови температури та вологості повітря в теплиці;
- запобігати пересиханню або перезволоженню ґрунту;
- забезпечувати рослинам оптимальний світловий режим;

Appvales - це не просто система моніторингу, але й інтегроване рішення для підвищення продуктивності та оптимізації виробничих процесів у тепличному господарстві. Завдяки високотехнологічним можливостям цієї системи, фермери мають змогу досягти максимального врожаю та забезпечити високу якість сільськогосподарської продукції.

1.3.2. Автоматизована теплична система Intellias

Команда Intellias розпочала розробку веб-додатку для зручного моніторингу та керування фермою в приміщенні, збираючи вимоги для вирішення конкретних потреб виробників. Під час дослідження вони визнали необхідність автоматично вставляти зібрані дані з нового програмного забезпечення для моніторингу теплиць прямо в уніфіковану платформу управління фермою їхнього клієнта, а також обмінюватися даними між виробниками, щоб прискорити реагування на зміни клімату, покращити розвиток насіння та знайти оптимальні умови для зростання врожаю [7].

Основні характеристики автоматизованої тепличної системи

- Веб-програма для спрощення введення, візуалізації та аналізу тепличних даних.
- Обов'язкове щорічне введення кліматичних даних і реєстрація рослин, а також порівняння з іншими членами групи.
- Календар для додавання нових періодів і введення усереднених тижневих даних у єдину електронну таблицю, щоб надати користувачам уніфікований перегляд усіх показників.
- Звіти та аналіз гістограм із шістьма обов'язковими гістограмами та конструктором для додаткових гістограм за потреби.
- Широкі можливості експорту.

Система моніторингу теплиць Intellias — це комплексне рішення, яке дозволяє контролювати всі параметри теплиці в режимі реального часу. Система складається з наступних компонентів:

- Прилади для збору даних: датчики температури, вологості, освітлення, вуглекислого газу, а також інші параметри, які можуть бути важливими для конкретного типу теплиці.

- Контролер: пристрій, який обробляє дані від датчиків і передає їх на сервер.
- Сервер: пристрій, який зберігає дані від датчиків і забезпечує доступ до них через веб-інтерфейс або мобільний додаток.
- Веб-інтерфейс або мобільний додаток: інструменти для перегляду даних і управління системою.

Система Intellias має широкий спектр функцій, які дозволяють автоматизувати та оптимізувати процес управління теплицею. До цих функцій належать:

- Контроль параметрів середовища: система Intellias дозволяє стежити за такими параметрами, як температура, вологість, освітлення, вуглекислий газ, а також іншими факторами, які можуть впливати на ріст рослин.
- Автоматичне управління обладнанням: система Intellias може управляти таким обладнанням, як системи опалення, вентиляції, освітлення та зрошення.
- Управління поливом: система Intellias дозволяє контролювати частоту, тривалість та обсяг поливу.
- Обробка даних: система Intellias дозволяє зберігати та аналізувати дані від датчиків. Це може бути використано для підвищення ефективності управління теплицею.

Система Intellias може використовуватися для поліпшення управління теплицею в різних аспектах. Наприклад, вона може бути використана для:

- Контролю температури в теплиці: допомагає запобігти заморозкам або перегріву рослин.
- Управління освітленням теплиці: забезпечує рослинам оптимальний рівень освітлення.
- Контролю вологості в теплиці: запобігає перезволоженню або пересиханню рослин.

- Управління поливом теплиці: допомагає забезпечити рослинам оптимальний рівень вологи.
- Аналіз даних від датчиків: виявляє проблеми з рослинами або обладнанням на ранніх стадіях.

Система Intellias має ряд переваг, які роблять її цінним інструментом для власників теплиць. До цих переваг належать:

- Автоматизація та оптимізація процесу управління теплицею: система Intellias може автоматизувати багато завдань, пов'язаних з управлінням теплицею. Це може звільнити час власника теплиці для інших завдань.
- Покращення ефективності управління теплицею: система Intellias може допомогти підвищити ефективність управління теплицею за рахунок забезпечення оптимальних умов для росту рослин.
- Покращення прибутковості: система Intellias може допомогти власникам теплиць підвищити прибутковість за рахунок підвищення продуктивності та зменшення втрат.

1.3.3. Система моніторингу теплиць Webhmi

Webhmi - це високотехнологічна система моніторингу теплиць, спроектована для оптимізації управління та контролю за умовами вирощування рослин у тепличному середовищі. Ця система впроваджує передові технології для забезпечення точного та ефективного моніторингу параметрів мікроклімату, а також для автоматизації процесів управління.

Цей проект був створений для теплиць, де були об'єкти моніторингу та контролю, такі як: датчики температури та вологості, клапани зрошення/туману та приводи вікон. Також, для кожного клапана та приводу потрібен був контроль за

розкладом, а для приводів контроль температури [8] . Однією з головних переваг є можливість віддаленого моніторингу та управління теплицею через інтернет.

Інтерфейс керування системою показано на рис.1.4

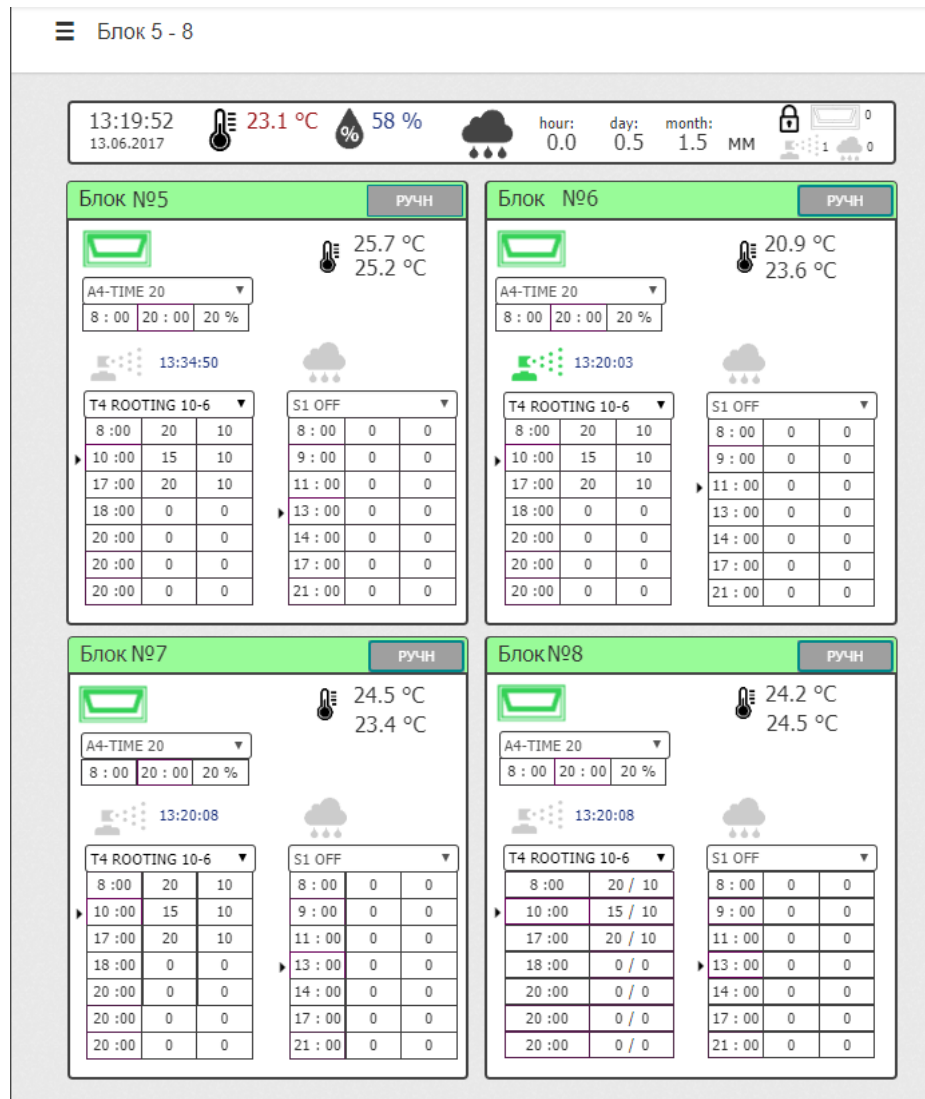


Рис. 1.4. Інтерфейс керування системою

Webhmi має такі особливості:

- Компактна апаратна платформа.
- Проста топологія та кабельне підключення.
- Масштабованість.
- Доступність.

Система моніторингу теплиць WebHMI є SCADA-системою з вбудованим веб-сервером, що дозволяє збирати, накопичувати та обробляти дані з датчиків, розташованих у теплиці. Система також дозволяє керувати обладнанням теплиці, таким як клапанами поливу, туману тощо.

Основні характеристики системи:

- Можливість підключення до будь-яких датчиків: система підтримує широкий спектр датчиків, включаючи датчики температури, вологості, світла, CO₂, pH тощо.
- Можливість керування будь-яким обладнанням: система дозволяє керувати будь-яким обладнанням теплиці, включаючи клапани поливу, туману, освітлення тощо.
- Вбудована веб-інтерфейс: система має вбудований веб-інтерфейс, який дозволяє відстежувати стан теплиці та керувати обладнанням з будь-якого комп'ютера або мобільного пристрою, підключеного до Інтернету.
- Історія даних: система зберігає історію даних з датчиків, що дозволяє аналізувати стан теплиці в динаміці.
- Резервне копіювання даних: система дозволяє робити резервне копіювання даних з датчиків, що дозволяє захистити дані від втрати.

Компоненти системи:

Система WebHMI складається з наступних компонентів:

- Сервер: сервер є центральним компонентом системи, він відповідає за збір, зберігання та обробку даних з датчиків.
- Давачі: датчики розташовані в теплиці і збирають дані про стан середовища.

Апаратні модулі: апаратні модулі забезпечують зв'язок між датчиками та сервером.

- Веб-інтерфейс: веб-інтерфейс дозволяє відстежувати стан теплиці та керувати обладнанням.

Приклади використання системи:

Система WebНМІ може використовуватися для моніторингу та керування теплицями різного розміру та призначення. Система може використовуватися для:

- Контролю температури та вологості: система може використовуватися для контролю температури та вологості в теплиці, що є важливим для забезпечення оптимальних умов для росту рослин.
- Керування поливом та туманом: забезпечує рослини необхідною кількістю вологи.
- Керування освітленням: дозволяє забезпечити рослини необхідним рівнем світла.
- Керування вентиляцією: система може використовуватися для керування вентиляцією, що дозволяє забезпечити в теплиці оптимальну циркуляцію повітря.
- Захисту рослин від шкідників та хвороб: захищає рослини від шкідників та хвороб, наприклад, шляхом автоматичного розпилення інсектицидів або фунгіцидів.

Переваги системи:

Система WebНМІ має ряд переваг, у тому числі:

- Широкий спектр підтримуваних датчиків: система підтримує широкий спектр датчиків, що дозволяє контролювати різні параметри середовища в теплиці.
- Можливість керування будь-яким обладнанням: система дозволяє керувати будь-яким обладнанням теплиці, що дозволяє автоматизувати процеси в теплиці.
- Вбудований веб-інтерфейс: вбудований веб-інтерфейс дозволяє відстежувати стан теплиці та керувати обладнанням з будь-якого комп'ютера або мобільного пристрою.
- Історія даних: система зберігає історію даних з датчиків, що дозволяє аналізувати стан теплиці в динаміці.

- Резервне копіювання даних: система дозволяє робити резервне копіювання даних з датчиків, що дозволяє захистити дані від втрати.

Система WebHMI є потужним інструментом для моніторингу та керування теплицями. Система має широкий спектр можливостей, що дозволяє автоматизувати процеси в теплиці та забезпечити оптимальні умови для росту рослин.

1.3.4. Система контролю клімату Priva

Система контролю клімату Priva є інноваційним рішенням у сфері садівничої автоматизації, спрямованою на мінімізацію виробничих витрат та оптимізацію управління різними аспектами сільськогосподарського виробництва. Зокрема, система дозволяє ефективно контролювати температуру, вентиляцію, екрани, вентилятори, туман, освітлення, опалення та управління котлом [10].

Головні функції системи Priva включають:

- Моніторинг та регулювання параметрів клімату: Система надає можливість швидко переглядати звіти про стан та регулювати температуру, вентиляцію та інші параметри для створення оптимальних умов для рослин.
- Зрошення та запліднення: Priva дозволяє встановлювати параметри для зрошення та запліднення, щоб забезпечити оптимальне водопостачання та живлення рослин.
- Інтеграція даних: Обробний комп'ютер системи є ключовою частиною вирощування, забезпечуючи інтеграцію всіх даних про клімат та зрошення. Це дозволяє отримати максимальну віддачу від систем та установок.

- Гнучкість та адаптивність: Система повинна бути гнучкою та здатною адаптуватися до змін та вимог, забезпечуючи ефективне управління всіма аспектами виробництва.
- Централізоване управління: Системи Priva Nutri-line можуть працювати як автономний блок або бути централізовано управляється за допомогою обробного комп'ютера, також відомого як "кліматичний регулятор".

На рис.1.5 показано інтерфейс системи контролю клімату Priva.



Рис. 1.5. інтерфейс системи контролю клімату Priva

Зображений інтерфейс мобільного додатку дозволяє більше зв'язувати системи для перехресного посилання та управління параметрами, що включає воду, рецептури поживних речовин, клімат та споживання енергії. Це забезпечує широке розуміння та рішення на основі даних для оптимального виробництва рослинництва.

Система Priva не лише підвищує ефективність вирощування рослин, але й сприяє сталому виробництву, дозволяючи оптимізувати використання ресурсів, ефективно

використовуючи воду та енергію. Завдяки високотехнологічним функціям, система не лише спрощує контроль за параметрами клімату та зрошенням, але і забезпечує централізоване та гнучке управління усіма важливими аспектами сільськогосподарського виробництва. Це інноваційне рішення відкриває перспективи для сучасного сільського господарства, де технології і аналітика об'єднуються для досягнення найкращих результатів у вирощуванні рослин.

1.4. Висновки за розділом

У даному розділі проведено огляд систем моніторингу мікроклімату для промислових теплиць. Визначено актуальність та важливість моніторингу мікроклімату в промислових теплицях. З'ясовано, що системи моніторингу мікроклімату промислових теплиць є важливим інструментом для підвищення ефективності та прибутковості виробництва. Зроблено огляд сучасних веб-засобів, які використовуються для моніторингу кліматичних умов в теплицях. Огляд сучасних веб-засобів, використовуваних для моніторингу кліматичних умов в теплицях, розкриває ряд важливих аспектів. Зокрема, зазначається, що сучасні веб-засоби моніторингу відрізняються високою зручністю використання, доступністю та можливістю віддаленого доступу. Ці характеристики роблять їх ефективним інструментом для сільськогосподарських підприємств, де важливо оперативно реагувати на зміни у кліматичних умовах. Описано основні аспекти, які враховуються при виборі систем моніторингу. Цей аналіз дозволяє визначити переваги та недоліки веб-засобів та їхню відповідність потребам сучасного сільськогосподарського сектору. Виявлено, що сучасні веб-засоби моніторингу кліматичних параметрів мають ряд переваг, у тому числі зручність використання, доступність та можливість віддаленого доступу. У контексті вибору систем моніторингу, були враховані різні аспекти, серед яких можна виділити технічні

характеристики, функціональні можливості та ефективність. Проведено детальний аналіз технічних та функціональних характеристик трьох систем-аналогів: "Appvales" "Intellias" та "Webhmi". Забезпечуючи надійний контроль за кліматичними умовами, ці системи дозволяють оптимізувати ріст і розвиток рослин, що в свою чергу призводить до збільшення врожаю та поліпшення якості продукції. Висвітлено загальні відомості про ці системи, їхні основні параметри та ключові функції. Описано функціонал та переваги системи "Appvales" Зазначено його особливості та можливості для оптимізації управління мікрокліматом в теплицях. Її особливості відображаються в надійності та ефективності, що робить її однією з перспективних систем для використання в сільському господарстві. Надано висновки щодо технічних можливостей та ефективності "Intellias" Визначено, як ця система може внести вагомий внесок у покращення умов для рослин. Високоточний моніторинг та аналіз кліматичних параметрів дозволяють максимально використовувати можливості теплиці для досягнення оптимальних результатів. Проаналізовано основні характеристики "Webhmi" і визначено його переваги та обмеження. Такий огляд систем відкриває можливості для подальшого вдосконалення та розвитку системи моніторингу мікроклімату для промислових теплиць. У підсумку, аналіз розглянутих систем відкриває перспективи для подальшого вдосконалення та розвитку систем моніторингу мікроклімату для промислових теплиць. Поглиблений розгляд їхніх характеристик надає підстави для вдосконалення та адаптації цих систем до конкретних потреб сучасного сільськогосподарського сектору.

РОЗДІЛ 2

МАТЕРІАЛИ, МЕТОДИ І МОДЕЛІ ДОСЛІДЖЕНЬ

2.1. Узагальнена структурно-функціональна організація системи

2.1.1. Оновлення даних в реальному часі

Одна з основних функцій системи моніторингу мікроклімату промислових теплиць - це надання актуальних даних про мікроклімат в режимі реального часу [11]. Це дозволяє оперативно реагувати на зміни в мікрокліматі та підтримувати оптимальні умови для росту рослин.

Для оновлення даних в реальному часі в системах моніторингу мікроклімату промислових теплиць використовується бібліотека SignalR [12]. Бібліотека SignalR дозволяє створювати веб-додатки, які підтримують двосторонню зв'язок між сервером і клієнтом. Це означає, що сервер може надсилати повідомлення клієнтам в реальному часі, а клієнти можуть надсилати повідомлення серверу.

У системах моніторингу мікроклімату промислових теплиць бібліотека SignalR використовується наступним чином:

- Датчики мікроклімату передають свої дані на контролер.
- Контролер обробляє отримані дані і передає їх на сервер.
- Сервер зберігає дані в базі даних і передає їх клієнтам.

За рахунок використання бібліотеки SignalR фактори, такі як точність, швидкість і складність оновлення даних в реальному часі, покращуються наступним чином:

- Точність: безперебійний зв'язок між сервером і клієнтами, що дозволяє гарантувати точність даних, отриманих від датчиків.
- Швидкість: забезпечується швидка передача даних між сервером і клієнтами, що дозволяє отримувати найсвіжіші дані про мікроклімат.

- Складність: бібліотека SignalR є порівняно простою в використанні, що полегшує розробку і експлуатацію систем моніторингу мікроклімату, що використовують оновлення даних в реальному часі.

На рис.2.1 зображено приклад структурної схеми моніторингу мікроклімату промислової теплиці.

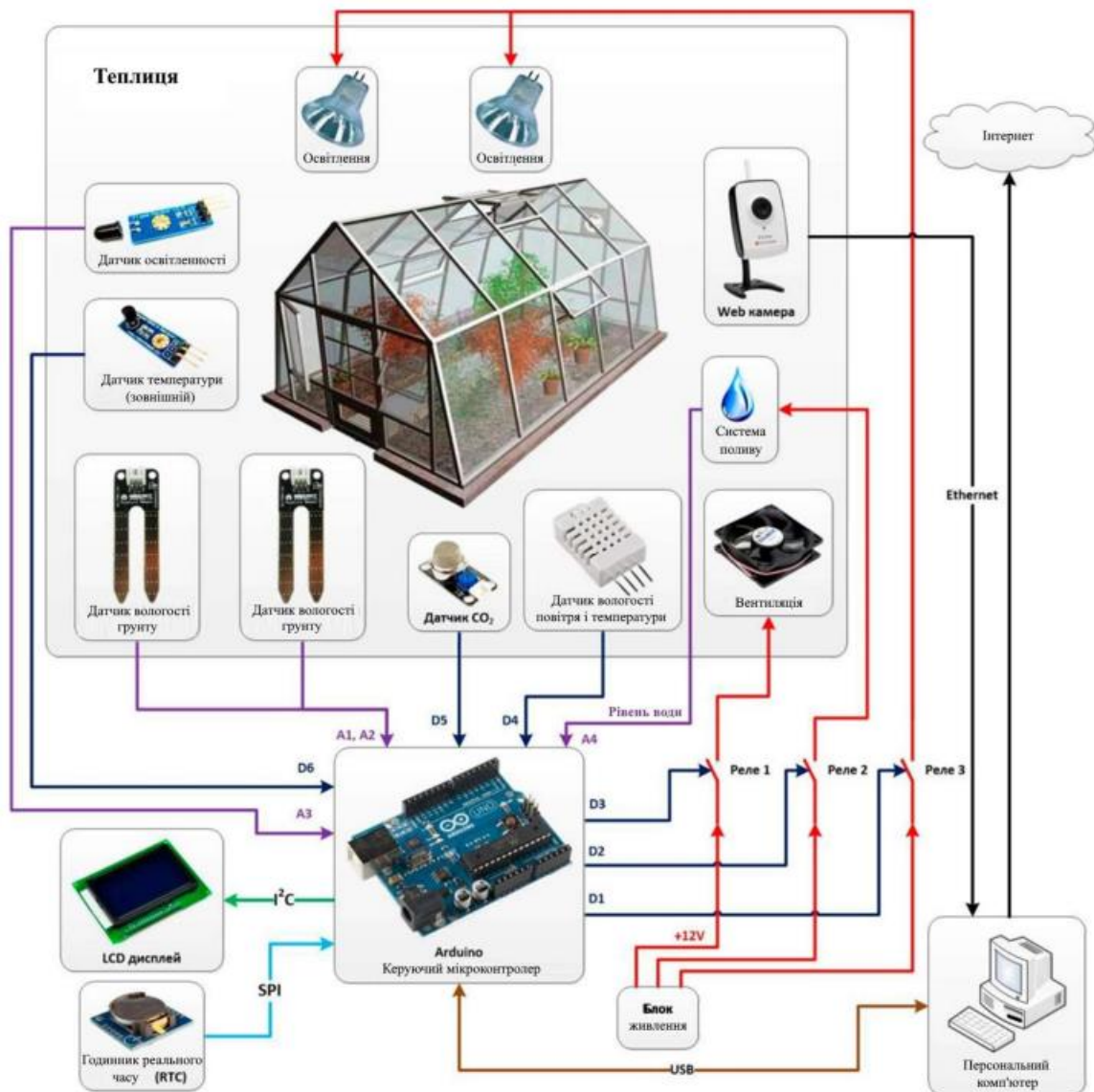


Рис. 2.1. Структурна схема моніторингу мікроклімату промислової теплиці

Переваги оновлення даних в реальному часі в системах моніторингу мікроклімату промислових теплиць:

- Швидке реагування на зміни: завдяки оновленню даних в реальному часі можна швидко реагувати на зміни в мікрокліматі, наприклад, на підвищення температури або вологості. Це дозволяє підтримувати оптимальні умови для росту рослин і запобігти їх пошкодженню.
- Покращена ефективність: оновлення даних в реальному часі дозволяє підвищити ефективність використання енергії та ресурсів. Наприклад, якщо температура в теплиці підвищується, можна автоматично включити систему вентиляції. Це дозволить заощадити енергію, яка витрачається на охолодження теплиці.
- Зменшення ризиків: Оновлення даних в реальному часі дозволяє фермерам зменшити ризики, пов'язані з несприятливими погодними умовами, шкідниками та хворобами. Наприклад, якщо в прогнозі погоди очікується похолодання, фермер може заздалегідь підготувати теплицю до цього, наприклад, включити систему опалення.

Оновлення даних в реальному часі є важливою функцією систем моніторингу мікроклімату промислових теплиць. Воно дозволяє користувачам системи отримувати найсвіжіші дані про мікроклімат теплиці, що може призвести до підвищення ефективності її роботи.

2.1.2. Збереження даних до бази даних

Збереження даних до бази даних є важливим етапом роботи будь-якої системи моніторингу мікроклімату промислових теплиць [13]. Це дозволяє зберігати дані для подальшого аналізу та використання, а також забезпечує надійний доступ до даних для користувачів системи.

Вибір методу збереження даних до бази даних може впливати на такі фактори, як:

- Точність: Точність даних залежить від того, як дані передаються від датчиків до бази даних.
- Швидкість: Рівень швидкості збереження даних залежить від обраного методу збереження.
- Складність: Складність збереження даних залежить від вибраного методу збереження.

На рис.2.2 показано функціональну схему системи для контролю мікроклімату промислових теплиць.

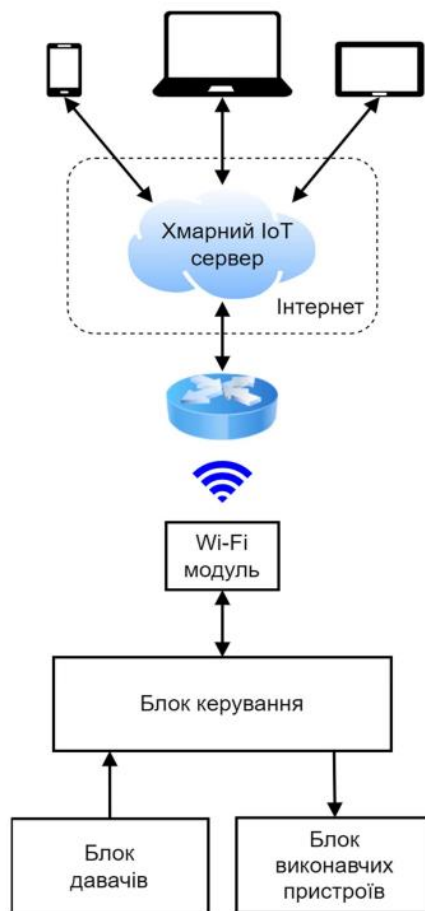


Рис. 2.2. Функціональна схема системи для контролю мікроклімату промислових теплиць

У системах моніторингу мікроклімату промислових теплиць використовуються датчики для вимірювання різних параметрів мікроклімату, таких

як температура, вологість, освітленість та інші. Дані з датчиків періодично передаються на контролер, який, у свою чергу, передає їх на сервер. Сервер зберігає дані в базі даних [15].

Найбільш поширеними типами баз даних для систем моніторингу мікроклімату є:

- Реляційні бази даних: Реляційні бази даних є найбільш поширеним типом баз даних. Вони складаються з таблиць, які містять дані про різні атрибути.
- Нереляційні бази даних: Нереляційні бази даних не використовують таблиці для зберігання даних. Замість цього вони використовують інші структури даних, такі як графіки та дерева.
- Кеш-бази даних: Кеш-бази даних використовуються для зберігання часто використовуваних даних. Вони дозволяють швидко отримувати доступ до даних, що може бути важливо для систем моніторингу мікроклімату.

Збереження даних до бази даних в системах моніторингу мікроклімату має ряд переваг:

- Довгострокове зберігання даних: Дані, що зберігаються в базі даних, можуть бути використані для подальшого аналізу та використання.
- Зручний доступ до даних: Дані, що зберігаються в базі даних, можуть бути легко доступні для користувачів системи.
- Розширюваність: База даних може бути легко розширена для зберігання додаткових даних.

Багато платформ програмного забезпечення для систем моніторингу мікроклімату включають в себе функції для збереження даних до бази даних. Ці функції зазвичай є простими у використанні і дозволяють користувачам легко зберігати дані в базі даних.

Збереження даних в базу даних відіграє ключову роль у забезпеченні надійності та доступності інформації в системі моніторингу мікроклімату для

промислових теплиць. Вибір реляційної бази даних MSSQL є обґрунтованим рішенням, оскільки цей тип баз даних визначається високою продуктивністю, надійністю та масштабованістю, що важливо для сучасних систем моніторингу. Для взаємодії з нею обрано SQL Server Management Studio(SSMS), яка є інтегрованим середовищем для керування будь-якою інфраструктурою SQL. SSMS надає інструменти для налаштування, моніторингу та адміністрування екземплярів SQL Server і баз даних [14]. Бази даних MSSQL є одним з найпопулярніших типів баз даних, що використовуються в системах моніторингу та управління. Вони пропонують високу продуктивність, надійність та масштабованість.

При виборі методу зберігання даних до бази даних MSSQL необхідно враховувати наступні фактори [16]:

- Ефективність. Метод повинен бути ефективним, щоб не спричиняти затримок у роботі системи.
- Надійність. Метод повинен бути надійним, щоб гарантувати безпеку даних.
- Доступність. Метод повинен бути доступним, щоб його можна було легко реалізувати.

У системі моніторингу мікроклімату промислових теплиць, що розглядається в даній роботі, для збереження даних до бази даних використовується бібліотека Dapper [32].

Вибір методу зберігання даних до бази даних MSSQL є важливим рішенням, яке може вплинути на ефективність і надійність системи моніторингу та управління мікрокліматом теплиць. Саме тому було обрано Dapper, бо він дозволяє мапити один рядок на кілька об'єктів. Це ключова функція, для того, щоб уникнути зайвих запитів та навантаження.

2.1.3. Зчитування вхідних даних

Зчитування вхідних даних - це етап роботи системи моніторингу та управління мікрокліматом теплиць, на якому дані, що надходять з датчиків або інших джерел, читаються та обробляються.

Для зчитування використовується бібліотека `CsvHelper`, яка містить велику кількість конфігурації будь-якого типу файлу CSV та будь-якого типу класу. Крім того, вона швидка, гнучка і проста у використанні [17].

Для зчитування вхідних даних можна використовувати різні методи, в тому числі:

- Читання даних з файлу. Цей метод є найбільш поширеним, оскільки дозволяє зберігати дані в файлі для подальшого використання.
- Читання даних з мережі. Цей метод дозволяє отримувати дані з інших систем або пристроїв через мережу.
- Читання даних з датчиків. Цей метод дозволяє отримувати дані безпосередньо з датчиків, встановлених в теплиці.

Вибір методу зчитування вхідних даних залежить від конкретних потреб системи.

У випадку з `CsvHelper`, це бібліотека, яка дозволяє зчитувати дані з файлів у форматі CSV. CSV - це текстовий формат, який використовується для зберігання даних у вигляді таблиці.

Для зчитування вхідних даних за допомогою `CsvHelper` необхідно виконати наступні кроки:

- Імпортувати бібліотеку. Спочатку необхідно імпортувати бібліотеку `CsvHelper` в проект.
- Створити об'єкт `CsvReader`. Для читання даних з файлу CSV необхідно створити об'єкт `CsvReader`.
- Налаштувати об'єкт `CsvReader`. Можна налаштувати об'єкт `CsvReader`, щоб вказати такі параметри, як роздільник полів, кодування файлу та інші.

- Читання даних з файлу. Дані з файлу CSV можна читати за допомогою методу Read() об'єкта CsvReader.

Після того, як дані будуть прочитані, їх можна обробити за допомогою відповідних методів і процедур.

Отже, бібліотека CsvHelper має ряд переваг, які роблять її зручним інструментом для зчитування вхідної інформації в системах моніторингу мікроклімату промислових теплиць.

2.2. Використані методи досліджень

У процесі створення системи моніторингу та управління мікрокліматом теплиць були використані наступні методи досліджень [18]:

- Аналіз існуючих систем. На початку проекту був проведений аналіз існуючих систем моніторингу та управління мікрокліматом теплиць. Це дозволило визначити основні вимоги до нової системи та найкращі практики її реалізації.
- Дослідження технологій. Було проведено дослідження різних технологій, які могли бути використані для створення системи. Це дозволило вибрати оптимальні технології для вирішення поставлених завдань.
- Тестування. Система була ретельно протестована на різних тестових сценаріях. Це дозволило виявити і усунути можливі помилки та недоліки.

Розробка системи моніторингу мікроклімату для промислових теплиць ґрунтується на використанні сучасних технологій та методів досліджень. Основні технології та методи, задіяні у процесі створення системи, включають:

- .NET - крос платформний фреймворк для створення сучасних програм і потужних хмарних сервісів [19].
- Blazor - це інтерфейс веб-фреймворку .NET, який підтримує рендеринг на стороні сервера та інтерактивність клієнта в одній моделі програмування [20].

- SQL - мова програмування для зберігання та обробки інформації в реляційній базі даних [21].
- Visual Studio – це додаток, який використовується для редагування, налагодження, створення коду та деплою програми [22].

.NET - це потужний і гнучкий фреймворк, який дозволяє розробляти високопродуктивне програмне забезпечення. Він пропонує широкий спектр можливостей, включаючи підтримку різних мов програмування, об'єктно-орієнтовану модель програмування, а також багато іншого.

У процесі створення системи моніторингу та управління мікрокліматом теплиць .NET був використаний для реалізації наступних функцій:

- Зчитування даних з файлу.
- Обробка даних.
- Зберігання даних у базі даних.
- Виконання запитів до бази даних.
- Відображення даних на веб-сторінках.

Blazor - це новий фреймворк від Microsoft, який дозволяє розробляти веб-додатки з використанням технології JavaScript. Він пропонує ряд переваг у порівнянні з традиційними веб-додатками, включаючи:

- Більшу продуктивність. Blazor дозволяє виконувати код на сервері, а не на клієнті. Це може значно підвищити продуктивність веб-додатків.
- Кращу безпеку. Код, який виконується на сервері, краще захищений від атак, ніж код, який виконується на клієнті.
- Більшу простоту розробки. Blazor спрощує розробку веб-додатків, оскільки дозволяє використовувати традиційні мовні конструкції, такі як класифікатори і методи.

У ході розробки системи моніторингу та управління мікрокліматом промислових теплиць великий акцент був покладений на використання технології

Blazor для реалізації ключових функцій. Blazor використовувався для створення веб-сторінок системи та забезпечення взаємодії з користувачем. Ця технологія дозволила ефективно відображати дані на веб-сторінках та забезпечити інтуїтивно зрозумілу взаємодію з системою.

SQL - це мова запитів до баз даних, яка дозволяє отримувати, зберігати і змінювати дані в базі даних. Вона є стандартною мовою для взаємодії з базами даних, що робить її легкою для вивчення і використання.

У процесі створення системи моніторингу та управління мікрокліматом теплиць SQL був використаний для реалізації наступних функцій:

- Зберігання даних.
- Виконання запитів до бази даних.

Використання SQL дозволило забезпечити надійність та швидкодійність операцій з даними, роблячи систему більш ефективною та функціональною.

Visual Studio - це потужне середовище розробки програмного забезпечення від Microsoft. Воно пропонує широкий спектр можливостей, включаючи підтримку різних мов програмування, інтеграцію з іншими інструментами Microsoft, а також багато іншого.

У процесі створення системи моніторингу та управління мікрокліматом теплиць Visual Studio був використаний для реалізації наступних функцій:

- Розробка серверної частини системи.
- Розробка клієнтської частини системи.

Вибрані технології та методи дозволяють створити ефективну і надійну систему моніторингу мікроклімату промислових теплиць. Вони забезпечують високу продуктивність, безпеку та простоту використання системи.

2.3. Інформаційна модель досліджуваної системи

2.3.1. Загальна модель комунікації компонентів

Процес обробки даних складається з декількох етапів. Перш за все дані за допомогою HTTP-запитів потрапляють до BE сервіса, який зчитує дані та відправляє до бази даних. Після запису даних до бд, вони відображаються на UI. Схема обробки даних показано на рис. 2.3. Такий підхід гарантує ефективну та надійну роботу між усіма процесами.

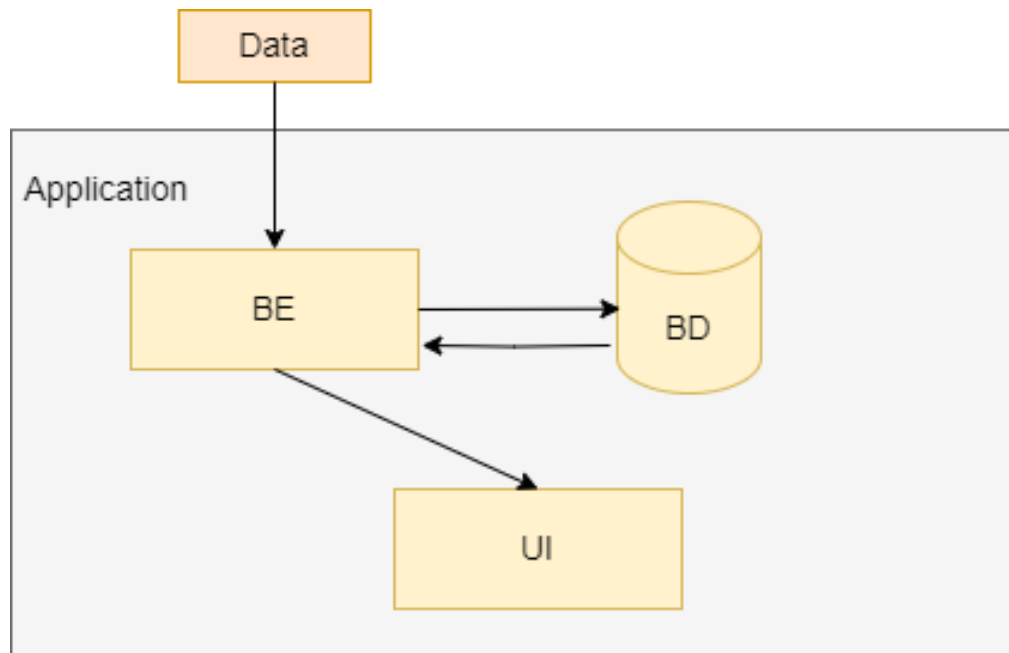


Рис. 2.3. Обробка вхідних даних

Система моніторингу та управління мікрокліматом теплиць складається з наступних компонентів :

- Датчики. Датчики збирають дані про мікроклімат теплиці.
- База даних. База даних зберігає дані, зібрані датчиками.
- Сервер. Сервер обробляє дані з бази даних і виконує команди управління.
- Клієнт. Клієнт дозволяє користувачам взаємодіяти з системою.

Компоненти системи взаємодіють один з одним за допомогою наступних каналів:

- Збір даних. Датчики збирають дані і передають їх на сервер.
- Зберігання даних. Сервер зберігає дані в базі даних.
- Обробка даних. Сервер обробляє дані з бази даних.
- Виконання команд. Сервер виконує команди управління, які надходять від клієнта.
- Відображення даних. Клієнт відображає дані, які надходять від сервера.

2.3.2. Обробка вхідних даних

Для забезпечення належного функціонування системи, виконується кілька ключових процесів. Бекенд отримує вхідні дані за допомогою HTTP протоколу [23]. HTTP протокол є стандартним протоколом для передачі даних в Інтернеті. Він використовується для обміну інформацією між клієнтом і сервером.

В системах моніторингу мікроклімату HTTP протокол може використовуватися для обробки вхідних даних від датчиків, користувачів і інших систем. Ці дані включають інформацію про теплиці, параметри мікроклімату та різноманітні запити на оновлення. Після чого проходить обробка й інтерпретація даних, виконання необхідних обчислень і підготовка їх для зберігання в базі даних. Сервер обмінюється даними з базою даних за допомогою технології Dapper.

Dapper - це бібліотека, яка дозволяє виконувати SQL-запити до баз даних без використання ORM. Dapper є простим у використанні і ефективним способом доступу до баз даних. Синхронізація даних між сервісами підтримується за допомогою технології SignalR. В цій роботі Dapper використовується для отримання та збереження даних для моніторингу мікроклімату.

SignalR - це бібліотека, яка дозволяє створювати веб-додатки, які підтримують двосторонній зв'язок між сервером і клієнтом. SignalR використовується для відправки даних користувачам в реальному часі. Це забезпечує плавну передачу даних у структурованому форматі. На рис. 2.4 представлено блок-схему роботи BE сервісу.

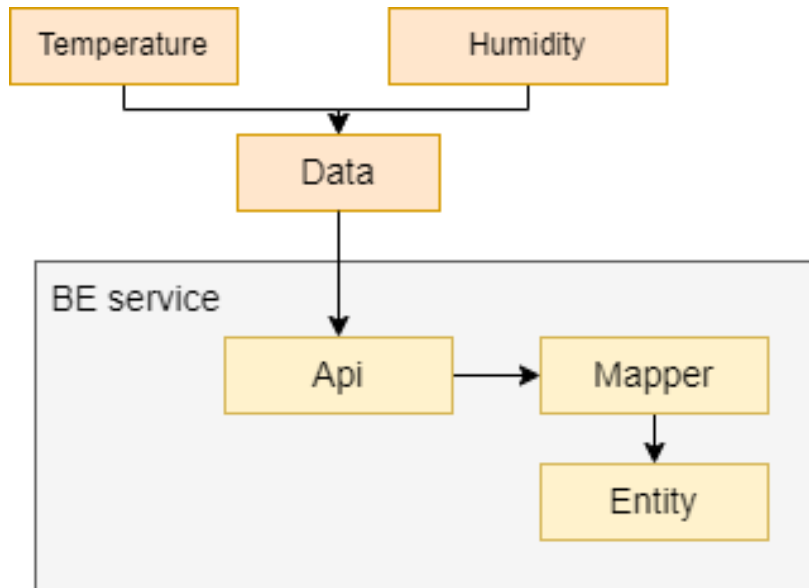


Рис. 2.4. Блок-схема роботи BE сервісу

HTTP, Dapper і SignalR є ефективними інструментами, які допомагають оброблювати вхідні дані в системі моніторингу мікроклімату.

2.3.3. Зберігання даних

В контексті системи моніторингу мікроклімату промислових теплиць, зберігання даних є важливою функцією систем моніторингу мікроклімату. Воно дозволяє зберігати дані про мікроклімат теплиці для подальшого аналізу та використання. Зберігання даних відбувається за допомогою SQL. SQL - це мова

програмування, яка використовується для роботи з базами даних. SQL дозволяє виконувати такі операції з базами даних, як [24]:

- Створення баз даних.
- Створення таблиць.
- Вставлення даних.
- Оновлення даних.
- Видалення даних.
- Запити до баз даних.

Блок-схема збереження даних представлена на рис. 2.5.

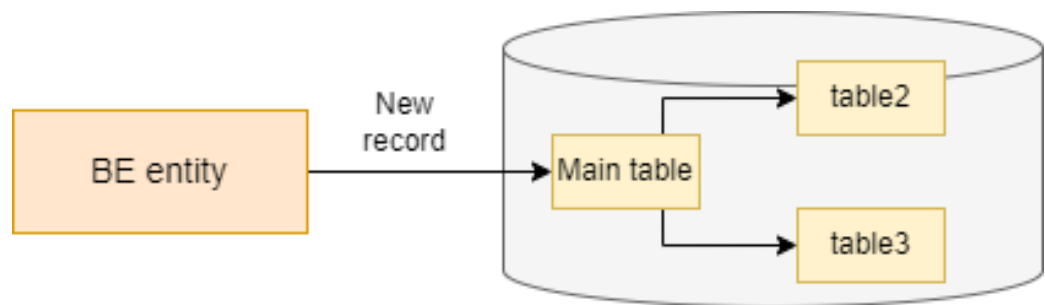


Рис. 2.5. Блок-схема збереження даних

База даних складається з набору таблиць, кожна з яких містить певні дані.

У даному випадку існує 6 таблиць:

- Greenhouse - містить загальну інформацію про теплицю.
- Air - містить дані про повітря в теплиці
- Atmosphere - містить дані про атмосферу в теплиці, наприклад, рівень вуглекислого газу.
- Sensors - містить інформацію про датчики, які використовуються для вимірювання даних.
- Soil - містить дані про ґрунт в теплиці, наприклад, температуру та вологість.

- Water - містить дані про воду в теплиці, наприклад, температуру.

ER-діаграма, представлена на рис. 2.6 , показує зв'язки між цими таблицями.

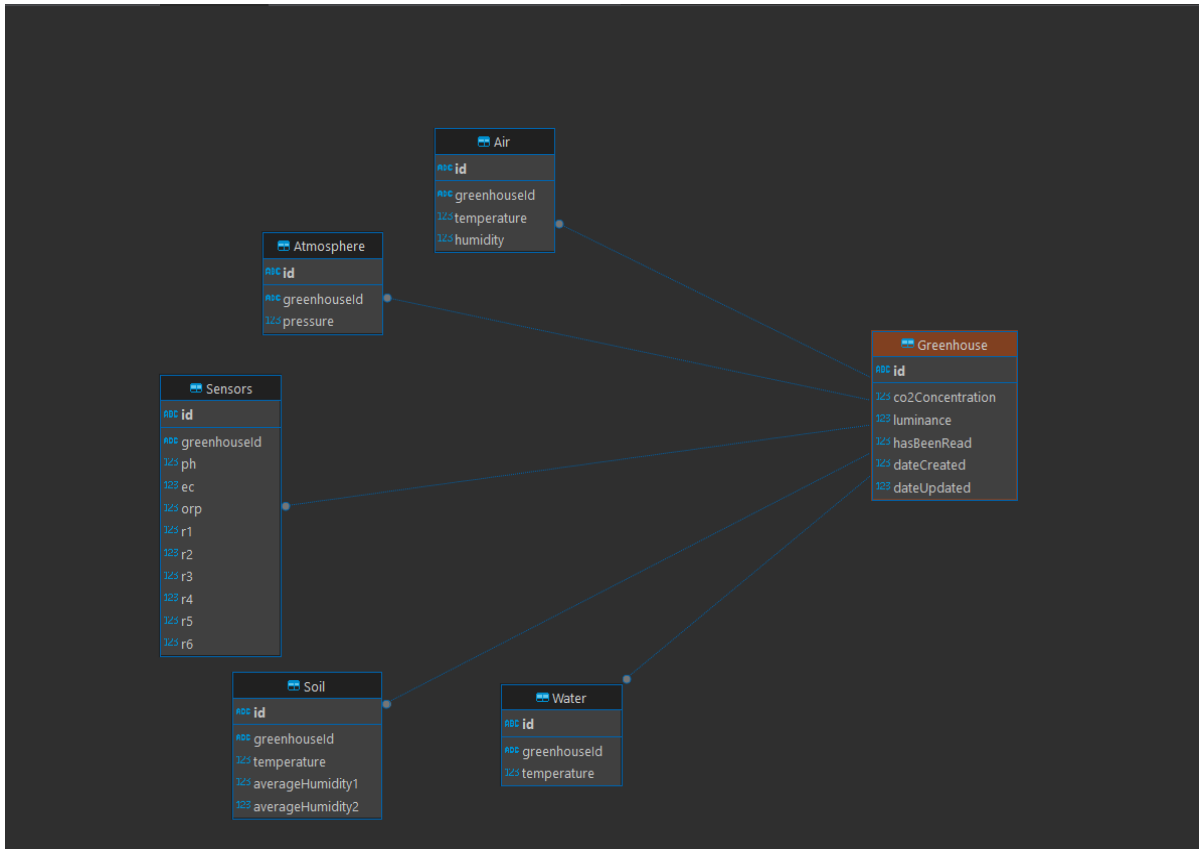


Рис. 2.6. ER зв'язки між таблицями

2.3.4. Оновлення даних

В системі моніторингу мікроклімату промислових теплиць використовується технологія SignalR. SignalR - це бібліотека для розробки веб-додатків в режимі реального часу. Ця технологія дозволяє забезпечити миттєвий обмін даними між сервером і клієнтами в режимі реального часу, що робить її ідеальним інструментом для обміну даними. SignalR дозволяє синхронізувати дані між різними компонентами системи, що є важливим для забезпечення оперативного реагування на зміни мікроклімату [25].

У процесі створення програми для моніторингу мікроклімату промислових теплиць використовується потужний механізм обміну даними в реальному часі, а саме SignalR. Цей механізм дозволяє ефективно передавати дані від сервера до клієнтів, забезпечуючи актуальну інформацію про мікроклімат теплиць у режимі реального часу. Клієнти отримують дані та відображають їх на веб-сторінці.

Для обміну даними в реальному часі в системі моніторингу мікроклімату промислових теплиць використовуються наступні компоненти:

- Сервер. Сервер відповідає за зберігання даних про мікроклімат теплиці та їх передачу клієнтам.
- Хаб. Хаб є каналом зв'язку між сервером та клієнтами.
- Клієнт. Клієнт отримує дані про мікроклімат теплиці від сервера за допомогою SignalR.

В розроблюваній програмі обмін даними відбувається за наступними етапами:

На серверній частині:

- Отримуються нові дані про мікроклімат теплиці від датчиків.
- Створюється повідомлення SignalR, що містить нові дані.
- Сервер надсилає повідомлення SignalR клієнтам.

На клієнтській частині:

- Отримується повідомлення SignalR від сервера.
- Клієнт десеріалізує дані з повідомлення SignalR.
- Оновлення та відображення даних на веб-сторінці.

Цей механізм обміну даними в реальному часі дозволяє клієнтам отримувати актуальну інформацію про мікроклімат теплиці в будь-який момент часу.

На рис.2.7 показано процес комунікації між клієнтом та сервером.

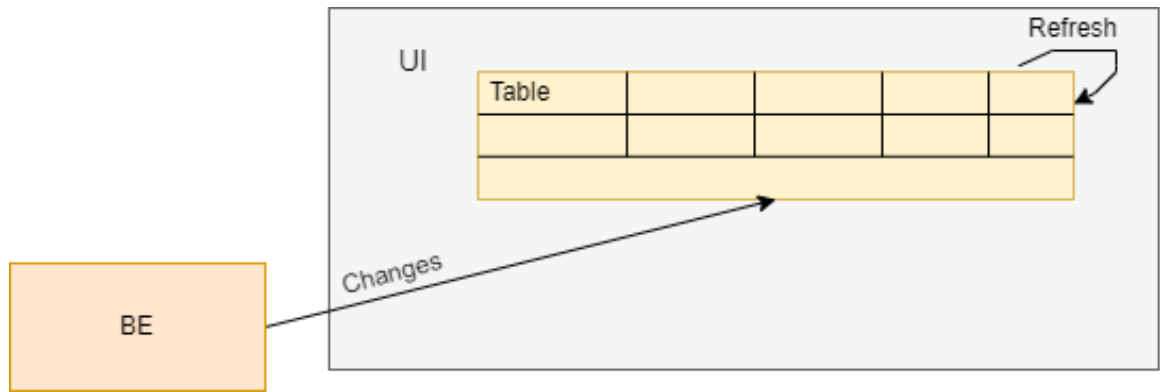


Рис. 2.7. Процес комунікації між клієнтом та сервером

2.4. Опис алгоритмів роботи досліджуваної системи

2.4.1. Масштабованість

У розроблюваній програмі для забезпечення масштабованості використовується мікросервісна архітектура. Крім того, очікується використання хмарних технологій.

Масштабованість системи моніторингу мікроклімату може бути забезпечена за допомогою таких механізмів [26]:

- Розподілені сервери - система може використовувати кілька серверів, які розподіляють навантаження між собою. Це дозволяє масштабувати систему відповідно до потреб, не вимагаючи придбання і обслуговування великої кількості серверів.
- Хмарні обчислення - система може використовувати хмарне обчислення для зберігання даних і виконання обчислень. Це дозволяє масштабувати систему відповідно до потреб, не вимагаючи придбання і обслуговування власної інфраструктури.

Хмарні обчислення дозволяють масштабувати систему відповідно до потреб, не вимагаючи придбання і обслуговування власної інфраструктури. Це дозволяє системам моніторингу мікроклімату регулювати потреби в обчисленнях

Хмарні обчислення можуть бути використані для зберігання даних, виконання обчислень і надання інших послуг, які необхідні для роботи системи моніторингу мікроклімату.

Щоб забезпечити ефективну та гнучку зростаючу систему з плином часу, використовується мікросервісна архітектура [27]. Кожен мікросервіс представляє окремий функціональний модуль, що спрощує розробку та дозволяє нам масштабувати лише ті частини системи, які потребують більшого обсягу ресурсів чи функціональних можливостей.

Мікросервісна архітектура має ряд переваг, включаючи:

- Масштабованість - системи з мікросервісною архітектурою можна легко масштабувати, додаючи або видаляючи сервіси.
- Надійність - системи з мікросервісною архітектурою більш стійкі до відмов, оскільки відмова одного сервісу не призводить до відмови всієї системи.
- Ефективність - системи з мікросервісною архітектурою можуть бути більш ефективними, оскільки вони дозволяють розробникам оптимізувати окремі сервіси.

Загальна схема мікросервісної архітектури представлена на рис.2.6

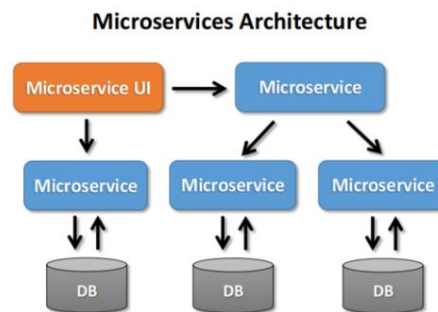


Рис. 2.8. Загальна схема мікросервісної архітектури

Така архітектура дозволяє масштабувати систему відповідно до потреб, не вимагаючи придбання і обслуговування великої кількості серверів.

В розроблюваній програмі, використовується фреймворк .NET, що дозволяє легко створювати і розгортати мікросервіси.

Для зберігання даних використовується база даних MSSQL, що є високопродуктивною і надійною базою даних, яка підходить для використання в масштабованих системах.

2.4.2. Надійність

Надійність - це здатність системи працювати без збоїв протягом заданого періоду часу. Надійність системи моніторингу мікроклімату є важливою характеристикою, оскільки вона дозволяє системі забезпечувати безперервний моніторинг мікроклімату теплиці.

Надійність системи моніторингу мікроклімату підвищується за допомогою таких заходів [28]:

- Вибір надійних компонентів - система побудована з надійних компонентів, таких як сервери і мережеве обладнання. Для зберігання даних використовується база даних MSSQL, яка є високопродуктивною і надійною базою даних. Для обробки даних використовується мову програмування .NET, яка має широкий набір бібліотек для обробки даних.
- Застосування надійних методів проектування - система спроектована з використанням надійних методів проектування, таких як резервування та відмовостійкість.

Резервування - це метод, який використовується для підвищення надійності системи шляхом створення резервних копій компонентів системи. Резервні компоненти можуть бути активовані, якщо основний компонент вийде з ладу.

2.4.3. Ефективність

Для підвищення ефективності системи використовується патерн CQRS [30]. Він допомагає розділити відповідальність між операціями команд та запитів. Ось як CQRS сприяє ефективності:

По перше, CQRS дозволяє оптимізувати операції читання та запису, розділяючи їхні відповідальності. Це дозволяє кожній частині системи спеціалізуватися на своїй роботі, що призводить до ефективнішої роботи.

По-друге, Запити виконуються на оптимізованому для зчитування шляху, що дозволяє ефективно та швидко отримувати дані. Ми можемо використовувати оптимізовані моделі для представлення даних для операцій читання.

Оскільки операції читання та запису розділені, ми можемо масштабувати їх незалежно одна від одної [29]. Це дозволяє ефективно масштабувати та розподіляти навантаження відповідно до потреб системи.

CQRS дозволяє використовувати асинхронний підхід для операцій читання та запису, що може покращити швидкодію системи в умовах великого обсягу операцій.

В цілому, використання CQRS дозволяє оптимізувати та прискорювати роботу системи, забезпечуючи ефективне використання ресурсів для потреб користувачів.

Для впровадження цього патерну використовується бібліотека MediatR [31]. MediatR допомагає визначити і розподілити обробку команд (Commands) і запитів (Queries). Він надає централізований механізм для відправлення команд та запитів до їхніх обробників. Завдяки MediatR, логіка обробки команд і запитів розділена на окремі обробники. Це полегшує управління кодом і дозволяє дотримуватися принципу єдиного обов'язку. MediatR дозволяє асинхронну обробку команд і запитів, що покращує продуктивність системи. Враховуючи ці переваги, MediatR відмінно підходить для використання в архітектурі, заснованій на CQRS, допомагаючи забезпечити чистоту, гнучкість та ефективність коду. На рис. 2.7 показано схему роботи патерну CQRS.

CQRS with MediatR

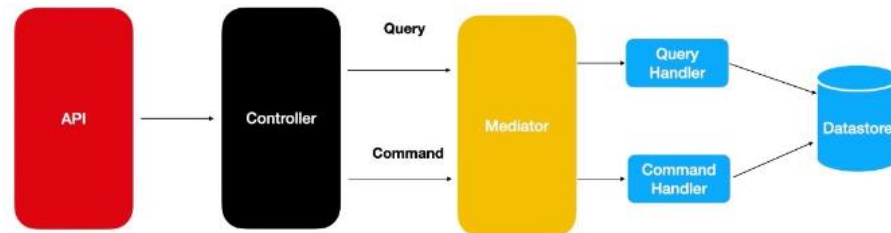


Рис. 2.9. Схема роботи патерну CQRS

2.5. Висновки за розділом

У даному розділі було виконано аналіз ключових аспектів системи моніторингу мікроклімату промислових теплиць. Детальний огляд структурно-функціональної організації системи дозволив виокремити основні характеристики, що впливають на її роботу. Проведено детальний аналіз структурно-функціональної організації системи моніторингу мікроклімату промислових теплиць. Це створює цілісний погляд на процеси, які відбуваються у системі моніторингу мікроклімату, підкреслюючи важливість правильного управління інформацією для досягнення мети системи. Визначено основні характеристики досліджуваної системи. Розроблено інформаційну модель системи, яка включає загальну комунікацію компонентів, обробку вхідних даних, їх зберігання та оновлення. Було використано різноманітні методи досліджень, спрямовані на розуміння та оптимізацію функціональності системи моніторингу. Визначено, як дані взаємодіють в реальному часі та як їх можна ефективно зберігати для подальшого використання. Ця модель надає високорівневий огляд процесів, що відбуваються у системі

моніторингу, що створює комплексний погляд на взаємодію складових системи та підкреслює важливість правильного управління інформацією для досягнення поставлених цілей. Описані алгоритми роботи системи дозволяють оцінити її масштабованість, надійність та ефективність. Все це становить фундаментальну базу для подальшого дослідження та вдосконалення системи моніторингу мікроклімату в промислових теплицях. Це важливий крок у забезпеченні оптимального функціонування та відповіді на вимоги реального виробничого середовища. Використання платформи .NET для серверної частини та Blazor для клієнтської вказує на сучасні та ефективні рішення в галузі розробки програмного забезпечення. Це важливо не лише для функціональності, але й для забезпечення стабільності та безпеки системи. Показано процес функціонування даних всередині системи. Зазначено використані технології для розробки системи. Описано використані методи досліджень при розробці системи моніторингу мікроклімату. Детально розглянуті методи досліджень при розробці системи моніторингу мікроклімату вказують на науковий та технічний підхід до створення системи. Використання мікросервісної архітектури виявляється як важливий фактор, що сприяє масштабованості, надійності та ефективності проекту. Узагальнюючи висновки, можна сказати, що проведений аналіз системи моніторингу мікроклімату промислових теплиць визначив її ключові переваги та аспекти, які слід враховувати для подальшого вдосконалення та розвитку. Важливість правильної інформаційної моделі, використання сучасних технологій та науково-дослідницький підхід створюють основу для створення ефективних та інноваційних систем моніторингу мікроклімату в агропромисловому секторі.

РОЗДІЛ 3

РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

3.1. Опис розробленої системи програмного забезпечення

3.1.1. Серверна частина

Для розробки серверної частини було використано технологію .NET. Було застосовано принцип мікросервісної архітектури для ефективного масштабування та обслуговування різних компонентів системи. Використання .NET з мікросервісною архітектурою має ряд переваг для розробки систем моніторингу мікроклімату теплиць:

- Модульність - система може бути поділена на невеликі, незалежні модулі, які можна розробляти і тестувати окремо. Це полегшує розробку, обслуговування та масштабування системи.
- Еластичність - система може бути легко масштабована, додаючи або видаляючи модулі. Це дозволяє системі адаптуватися до змін у потребах теплиці.
- Надійність - мікросервісна архітектура може підвищити надійність системи шляхом використання таких методів, як резервування і відмовостійкість.

Використана архітектура показана на рис. 3.1.

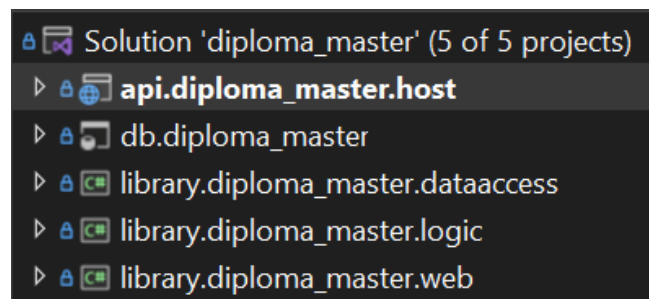


Рис. 3.1. Мікросервісна архітектура системи

Для того, щоб мати можливість розширювати компоненти, дані були розділені на окремі моделі. Структура моделей показана на рис. 3.2.

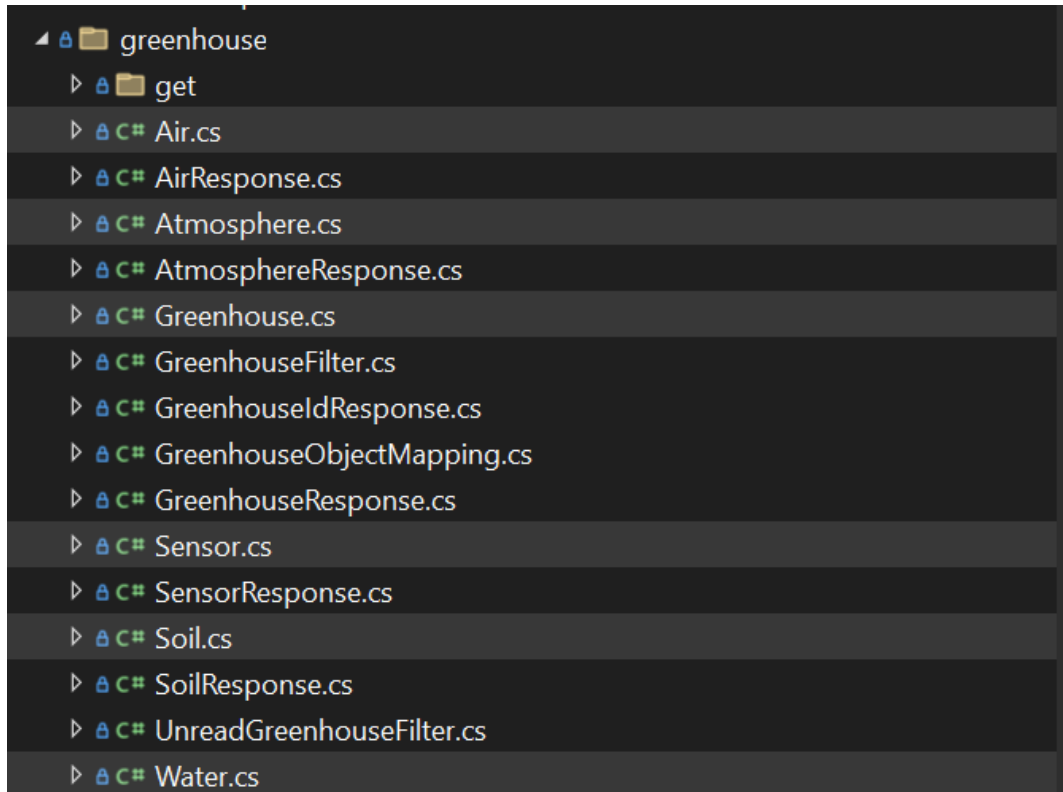


Рис. 3.2. Структура моделей даних

Комунікація між сервісами проводиться за допомогою патернів CQRS та MediatR.

CQRS (Command Query Responsibility Segregation) - це архітектурний шаблон, який розділяє додаток на два основні потоки: потік команд і потік запитів. Це дозволяє краще структурувати додаток і підвищити його масштабованість.

MediatR дозволяє створювати централізовані точки входу для обробки команд і запитів. Це полегшує тестування коду і підвищує його надійність.

Код використання патернів на прикладі завантаження файлів наведений в додатку А.

У якості бази даних використовується MSSQL. Для комунікації з бд використовується Dapper [20]. Dapper є бібліотекою, що дозволяє виконувати SQL-запити без необхідності писати код для створення об'єктів ADO.NET. Це спрощує розробку коду і підвищує його продуктивність. Код отримання даних для теплиці показано в додатку Б. Серверна частина системи програмного забезпечення для моніторингу мікроклімату теплиці забезпечує всі необхідні функції для ефективного моніторингу. Серверна частина написана на популярних і надійних технологіях, що забезпечує її надійність і масштабованість.

3.1.2. Клієнтська частина

Клієнтська частина системи моніторингу мікроклімату теплиць розроблена на основі технології Blazor. Blazor - це технологія, яка дозволяє створювати інтерактивні веб-додатки, які працюють на сервері. Це означає, що клієнтська частина додатка не вимагає установки додаткових плагінів або розширень.

Клієнтська частина системи моніторингу мікроклімату теплиць була оцінена за такими критеріями:

- Простота використання - додаток має простий і інтуїтивно зрозумілий інтерфейс.
- Функціональність - додаток забезпечує всі необхідні функції для моніторингу мікроклімату теплиці.
- Ефективність - додаток працює швидко і без збоїв.

Вибір технології Blazor для розробки клієнтської частини системи був обумовлений такими факторами:

- Мобільність - Blazor Server дозволяє веб-додаткам взаємодіяти з датчиками мікроклімату і системами теплиці безпосередньо через сервер.

- Простота використання - технологія Blazor має простий і інтуїтивно зрозумілий синтаксис.

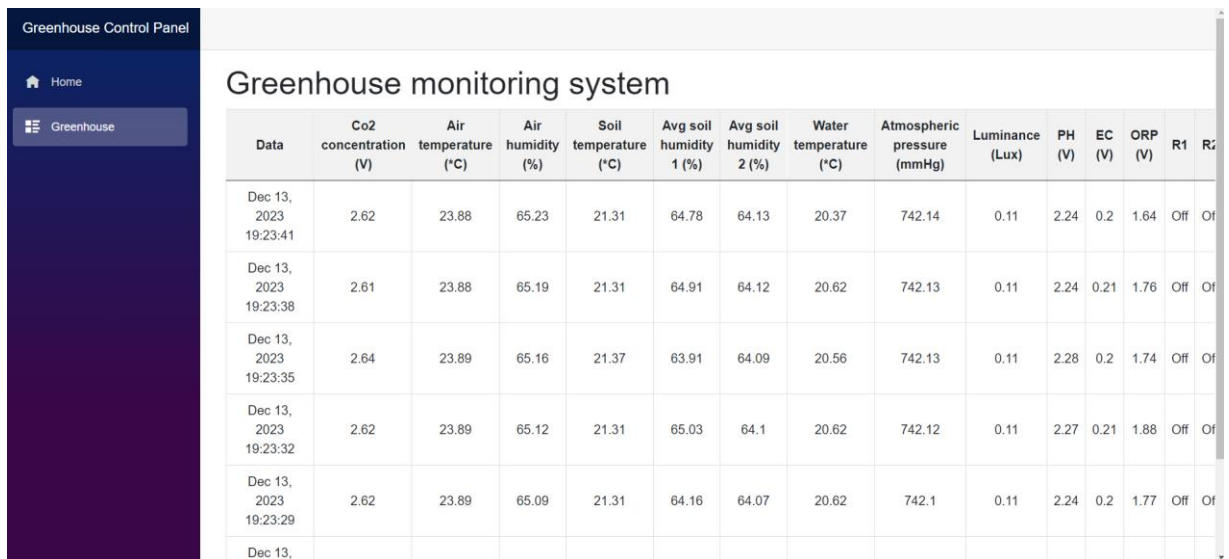
При розробці веб-додатку були використані такі технології:

.NET 6 - це фреймворк, який використовується для розробки веб-додатків на основі Blazor.

Microsoft SQL Server - це реляційна база даних, яка використовується для зберігання даних мікроклімату.

Було створено декілька основних функцій. Одна з них – оновлення даних для теплиці в реальному часі, що є важливою складовою системи моніторингу та керування тепличним вирощуванням рослин. Основна мета цього розділу - забезпечити користувачам доступ до актуальної інформації про умови в теплиці в реальному часі. Код цього функціоналу показаний у додатку В.

Веб інтерфейс сторінки показаний на рис.3.3.



The screenshot displays a web application titled "Greenhouse Control Panel" with a sidebar menu containing "Home" and "Greenhouse". The main content area is titled "Greenhouse monitoring system" and features a table with the following data:

Data	Co2 concentration (V)	Air temperature (°C)	Air humidity (%)	Soil temperature (°C)	Avg soil humidity 1 (%)	Avg soil humidity 2 (%)	Water temperature (°C)	Atmospheric pressure (mmHg)	Luminance (Lux)	PH (V)	EC (V)	ORP (V)	R1	R2
Dec 13, 2023 19:23:41	2.62	23.88	65.23	21.31	64.78	64.13	20.37	742.14	0.11	2.24	0.2	1.64	Off	Of
Dec 13, 2023 19:23:38	2.61	23.88	65.19	21.31	64.91	64.12	20.62	742.13	0.11	2.24	0.21	1.76	Off	Of
Dec 13, 2023 19:23:35	2.64	23.89	65.16	21.37	63.91	64.09	20.56	742.13	0.11	2.28	0.2	1.74	Off	Of
Dec 13, 2023 19:23:32	2.62	23.89	65.12	21.31	65.03	64.1	20.62	742.12	0.11	2.27	0.21	1.88	Off	Of
Dec 13, 2023 19:23:29	2.62	23.89	65.09	21.31	64.16	64.07	20.62	742.1	0.11	2.24	0.2	1.77	Off	Of
Dec 13,														

Рис. 3.3. Сторінка моніторингу змін мікроклімату теплиці

Ще однією функцією додатку є завантаження даних з файлу. Користувач обирає файл, який містить всередині дані для теплиці, після чого починається

процес завантаження. Відповідний програмний код знаходиться в додатку Г. Інтерфейс сторінки завантаження показаний на рис. 3.4.

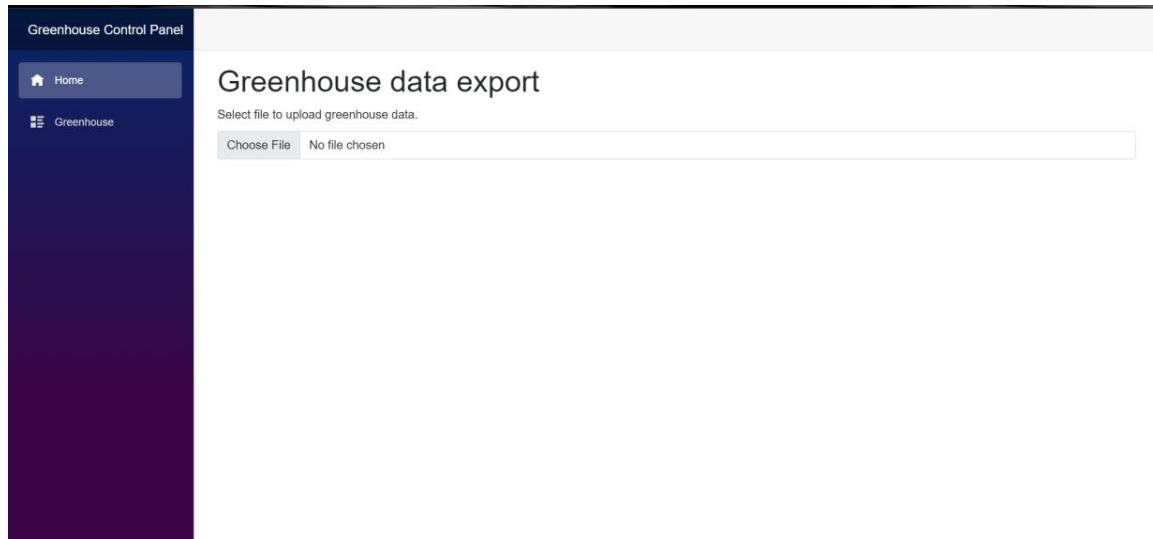


Рис. 3.4. Сторінка завантаження файлу

3.2. Результати тестування розробленої системи програмного забезпечення

Першим етапом є завантаження даних з файлу. На рис. 3.5 показано файл з вхідними параметрами.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	U_CDCor	2.63 V	AirTemper	23.96 °C	AirHumidit	65.32 %	SoilTempe	21.44 °C	SoilHumidit	64.41 %	SoilHumidit	64.46 %	WaterTem	20.69 °C	Lumiance	0.11 lux							
2	U_CDCor	2.62 V	AirTemper	23.97 °C	AirHumidit	65.29 %	SoilTempe	21.5 °C	SoilHumidit	64.16 %	SoilHumidit	64.2 %	WaterTem	20.69 °C	Lumiance	0.11 lux							
3	U_CDCor	2.67 V	AirTemper	23.97 °C	AirHumidit	65.24 %	SoilTempe	21.44 °C	SoilHumidit	63.29 %	SoilHumidit	64.4 %	WaterTem	20.41 °C	Lumiance	0.11 lux							
4	U_CDCor	2.64 V	AirTemper	23.98 °C	AirHumidit	65.2 %	SoilTempe	21.44 °C	SoilHumidit	64.54 %	SoilHumidit	64.39 %	WaterTem	20.75 °C	Lumiance	0.11 lux							
5	U_CDCor	2.64 V	AirTemper	23.97 °C	AirHumidit	65.17 %	SoilTempe	21.44 °C	SoilHumidit	63.67 %	SoilHumidit	64.37 %	WaterTem	20.56 °C	Lumiance	0.11 lux							
6	U_CDCor	2.68 V	AirTemper	23.98 °C	AirHumidit	65.16 %	SoilTempe	21.17 °C	SoilHumidit	63.04 %	SoilHumidit	64.32 %	WaterTem	20.69 °C	Lumiance	0.11 lux							
7	U_CDCor	2.64 V	AirTemper	23.98 °C	AirHumidit	65.11 %	SoilTempe	21.37 °C	SoilHumidit	63.03 %	SoilHumidit	64.34 %	WaterTem	20.75 °C	Lumiance	0.11 lux							
8	U_CDCor	2.64 V	AirTemper	23.98 °C	AirHumidit	65.07 %	SoilTempe	21.44 °C	SoilHumidit	64.16 %	SoilHumidit	64.31 %	WaterTem	20.69 °C	Lumiance	0.11 lux							
9	U_CDCor	2.64 V	AirTemper	23.98 °C	AirHumidit	65.04 %	SoilTempe	21.17 °C	SoilHumidit	64.54 %	SoilHumidit	64.32 %	WaterTem	20.75 °C	Lumiance	0.11 lux							
10	U_CDCor	2.63 V	AirTemper	23.98 °C	AirHumidit	64.99 %	SoilTempe	21.5 °C	SoilHumidit	64.41 %	SoilHumidit	64.33 %	WaterTem	20.69 °C	Lumiance	0.11 lux							
11	U_CDCor	2.66 V	AirTemper	23.98 °C	AirHumidit	64.96 %	SoilTempe	21.25 °C	SoilHumidit	63.66 %	SoilHumidit	64.31 %	WaterTem	20.9 °C	Lumiance	0.11 lux							
12	U_CDCor	2.61 V	AirTemper	23.98 °C	AirHumidit	64.94 %	SoilTempe	21.37 °C	SoilHumidit	66.03 %	SoilHumidit	64.41 %	WaterTem	20.75 °C	Lumiance	0.11 lux							
13	U_CDCor	2.61 V	AirTemper	23.98 °C	AirHumidit	64.91 %	SoilTempe	21.25 °C	SoilHumidit	65.66 %	SoilHumidit	64.41 %	WaterTem	20.69 °C	Lumiance	0.11 lux							
14	U_CDCor	2.6 V	AirTemper	23.95 °C	AirHumidit	64.92 %	SoilTempe	21.17 °C	SoilHumidit	65.66 %	SoilHumidit	64.5 %	WaterTem	20.75 °C	Lumiance	0.11 lux							
15	U_CDCor	2.6 V	AirTemper	23.95 °C	AirHumidit	64.9 %	SoilTempe	21.37 °C	SoilHumidit	65.9 %	SoilHumidit	64.52 %	WaterTem	20.69 °C	Lumiance	0.11 lux							
16	U_CDCor	2.61 V	AirTemper	23.98 °C	AirHumidit	64.81 %	SoilTempe	21.44 °C	SoilHumidit	64.54 %	SoilHumidit	64.55 %	WaterTem	20.69 °C	Lumiance	0.11 lux							
17	U_CDCor	2.61 V	AirTemper	23.98 °C	AirHumidit	64.9 %	SoilTempe	21.44 °C	SoilHumidit	65.9 %	SoilHumidit	64.62 %	WaterTem	20.69 °C	Lumiance	0.11 lux							
18	U_CDCor	2.6 V	AirTemper	23.98 °C	AirHumidit	64.9 %	SoilTempe	21.44 °C	SoilHumidit	65.78 %	SoilHumidit	64.66 %	WaterTem	20.69 °C	Lumiance	0.11 lux							
19	U_CDCor	2.62 V	AirTemper	23.97 °C	AirHumidit	64.91 %	SoilTempe	21.17 °C	SoilHumidit	65.53 %	SoilHumidit	64.7 %	WaterTem	20.9 °C	Lumiance	0.11 lux							
20	U_CDCor	2.62 V	AirTemper	23.97 °C	AirHumidit	64.93 %	SoilTempe	21.44 °C	SoilHumidit	65.9 %	SoilHumidit	64.76 %	WaterTem	20.75 °C	Lumiance	0.11 lux							
21	U_CDCor	2.62 V	AirTemper	23.97 °C	AirHumidit	64.95 %	SoilTempe	21.11 °C	SoilHumidit	65.28 %	SoilHumidit	64.79 %	WaterTem	20.9 °C	Lumiance	0.11 lux							
22	U_CDCor	2.62 V	AirTemper	23.97 °C	AirHumidit	64.97 %	SoilTempe	21.44 °C	SoilHumidit	65.53 %	SoilHumidit	64.81 %	WaterTem	20.9 °C	Lumiance	0.11 lux							
23	U_CDCor	2.63 V	AirTemper	23.97 °C	AirHumidit	64.99 %	SoilTempe	21.44 °C	SoilHumidit	65.28 %	SoilHumidit	64.83 %	WaterTem	20.75 °C	Lumiance	0.11 lux							

Рис. 3.5. Файл з вхідними параметрами

Запускаємо додаток. На рис. 3.6 показана сторінка з вибором файлу.

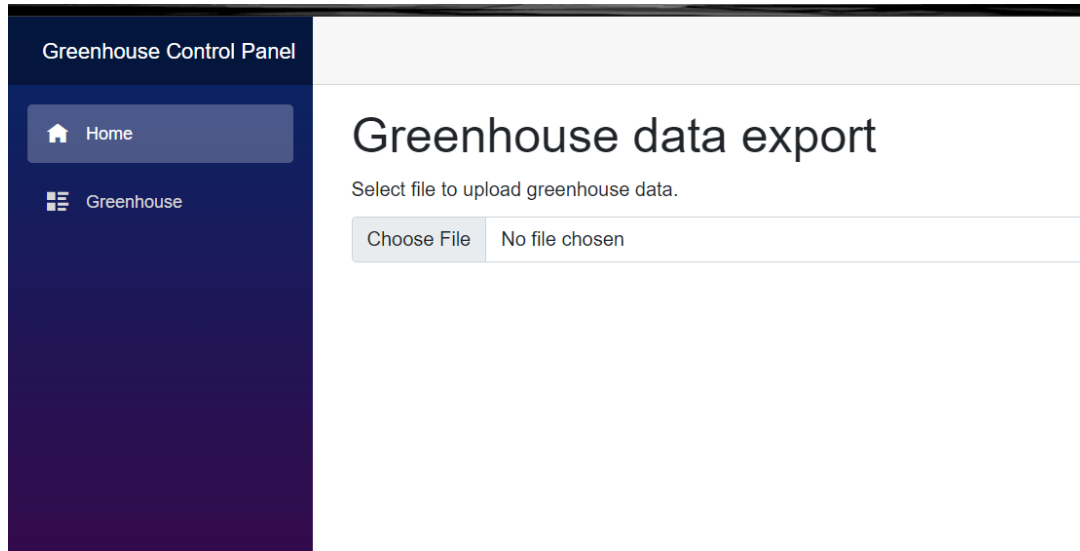


Рис. 3.6. Сторінка вибору файлу

Клікаючи на кнопку Choose file відкривається файловий менеджер, де з'являється можливість вибрати потрібний документ. На рис. 3.7 відображено вибір файлу з файлової системи.

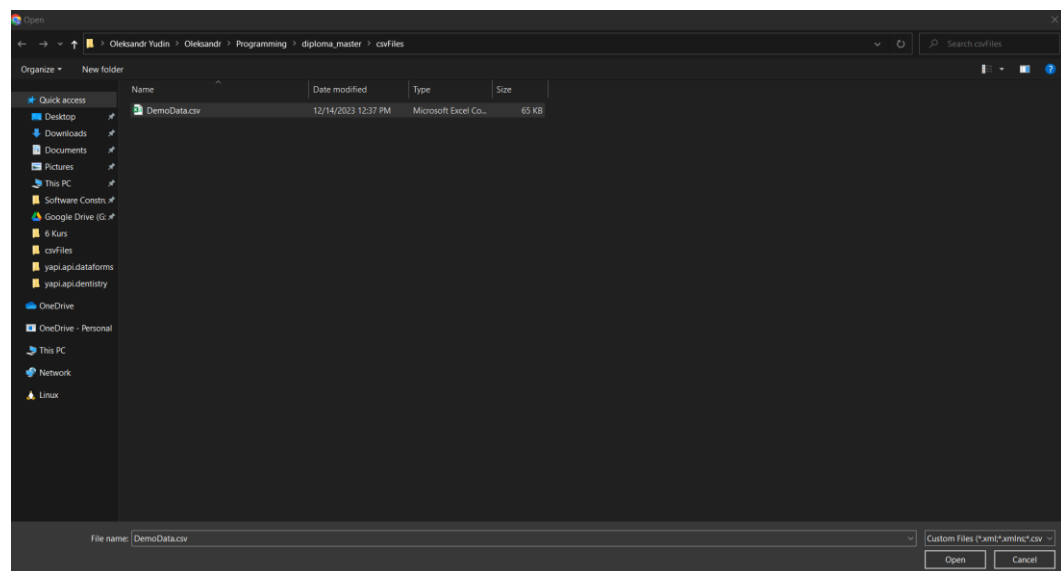


Рис. 3.7. Вибір файлу з файлової системи

Після того як файл обраний починається завантаження. Якщо файл валідний, то після завершення завантаження буде відображено відповідне повідомлення. На рис. 3.8 показано успішний результат завантаження файлу.

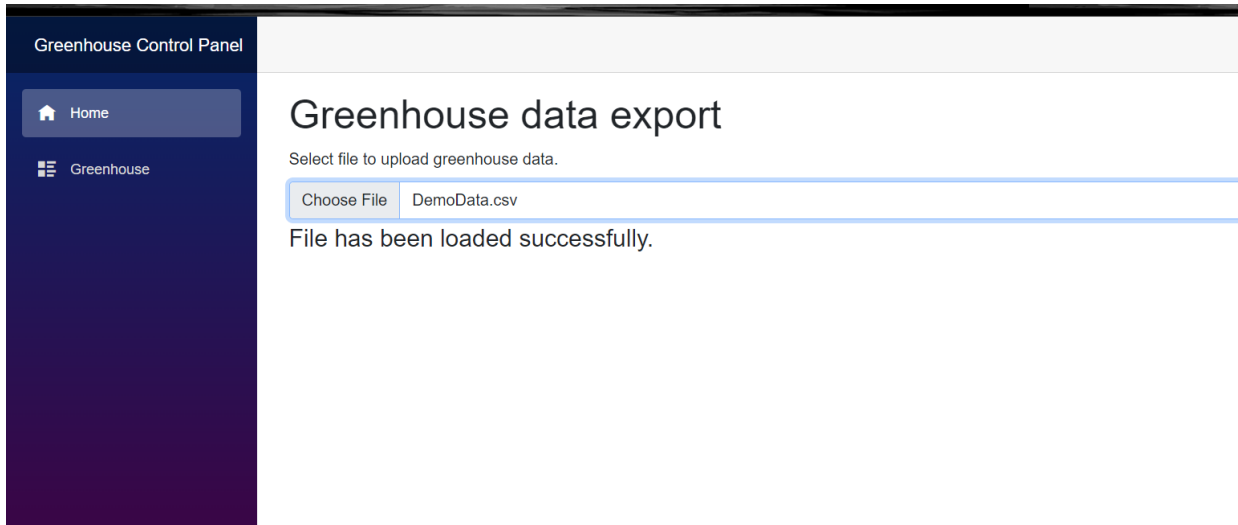


Рис. 3.8. Успішний результат завантаження файлу

Після успішного завантаження дані можна знайти в базі даних у відповідних таблицях. На рис. 3.9 видно збережені з файлу.

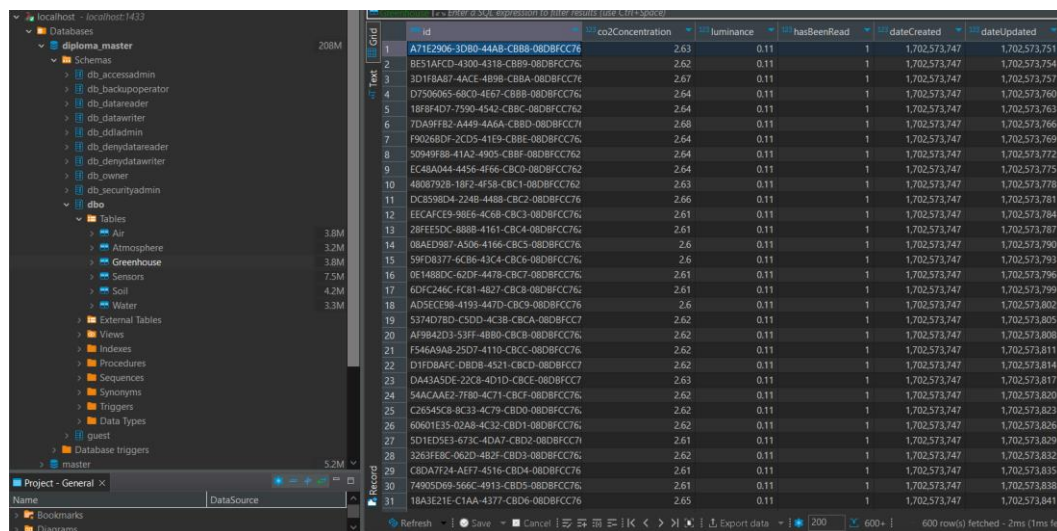


Рис. 3.9. Збережені дані в базі даних

Якщо під час вибору файлу обрати файл з невалідними даними або відкритий файл, то користувач побачить повідомлення з помилкою на сторінці інтерфейсу. На рис. 3.10 показано приклад завантаження файлу з помилкою.

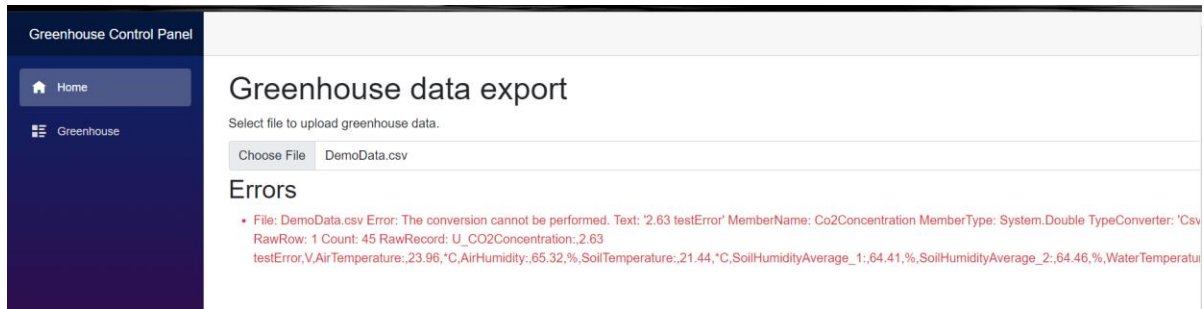


Рис. 3.10 Завантаження файлу з помилкою

Для того, щоб показати роботу оновлення даних в реальному часі, була створена затримка, яка дозволяє побачити роботу програми в реальному часі. На рис. 3.11Рис. 3.11. Відображення отриманих записів показано відображення записів з інтервалом в 3с.

The screenshot shows the 'Greenhouse monitoring system' section of the 'Greenhouse Control Panel'. It displays a table with the following data:

Data	Co2 concentration (V)	Air temperature (°C)	Air humidity (%)	Soil temperature (°C)	Avg soil humidity 1 (%)	Avg soil humidity 2 (%)	Water temperature (°C)	Atmospheric pressure (mmHg)	Luminance (Lux)	PH (V)	EC (V)	ORP (V)	R1	R2
Dec 14, 2023 19:06:17	2.64	23.98	65.11	21.37	65.03	64.34	20.75	742.52	0.11	2.25	0.22	1.88	Off	Of
Dec 14, 2023 19:06:14	2.68	23.98	65.16	21.37	63.04	64.32	20.69	742.51	0.11	2.28	0.21	1.74	Off	Of
Dec 14, 2023 19:06:11	2.64	23.97	65.17	21.44	63.67	64.37	20.56	742.53	0.11	2.27	0.2	1.94	Off	Of
Dec 14, 2023 19:06:08	2.64	23.98	65.2	21.44	64.54	64.39	20.75	742.52	0.11	2.28	0.22	1.71	Off	Of
Dec 14, 2023 19:06:05	2.67	23.97	65.24	21.44	63.29	64.4	20.81	742.51	0.11	2.25	0.22	1.92	Off	Of
Dec 14,														

Рис. 3.11. Відображення отриманих записів

3.3. Перспективні напрямки подальших досліджень

3.3.1. Покращення існуючого функціоналу

Одним з потенційних напрямків для подальших досліджень є розширення функціоналу за допомогою додавання графіків та більше ефективного сортування існуючої таблиці.

Додавання графіків:

- Розробка графічних відображень даних дозволить користувачам більш зручно аналізувати зміни в мікрокліматі протягом певного періоду часу.
- Графіки можуть включати в себе різноманітні параметри, такі як температура, вологість, тиск тощо. Це може покращити розуміння стану системи.

Ефективне сортування таблиці:

- Вдосконалення сортування існуючої таблиці дозволить користувачам швидко знаходити необхідну інформацію та порівнювати значення різних параметрів.
- Можливість сортування за різними критеріями, такими як час, місцезнаходження, тип показника мікроклімату, може полегшити роботу та аналіз.
- Розробка механізму оповіщення

3.3.2. Впровадження нових функціональних можливостей

Основні аспекти цього напрямку можуть включати:

- Розробка зручної та інтуїтивно зрозумілої панелі керування, яка дозволяє користувачам взаємодіяти з усіма аспектами системи моніторингу та контролю мікроклімату.
- Додавання можливостей для миттєвого реагування та коригування параметрів мікроклімату в реальному часі на основі отриманих даних.

- Впровадження автоматизованих систем регулювання, які можуть адаптувати параметри, такі як температура, вологість чи освітленість, залежно від вимог рослин або конкретних завдань користувача.
- Розширення інтерфейсу для взаємодії з іншими сучасними технологіями, такими як голосові асистенти, мобільні додатки або системи "розумний дім"
- Включення в інтерфейс аналітичних інструментів та можливостей створення звітів для детального аналізу ефективності та результатів управління.
- Додавання можливостей інтерактивного взаємодії, наприклад, візуалізація змін параметрів через графічні елементи чи використання технологій віртуальної реальності.

Реалізація цих напрямків досліджень може значно покращити способи взаємодії та управління мікрокліматичною системою, забезпечуючи користувачам більш широкі можливості для персоналізації та оптимізації умов оточуючого середовища.

3.4. Порівняльний аналіз отриманих результатів із раніше відомими

Було проведено порівняльний аналіз розробленої системи моніторингу мікроклімату теплиць з трьома іншими системами аналогів: Webhmi, Intellias та Appvales. Порівняння було проведено за такими критеріями:

- Інтерфейс користувача - розроблена система уступає іншим системам у зручному та інтуїтивно зрозумілому інтерфейсі через відсутність зворотного зв'язку від користувачів і потребує деяких покращень.
- Функціональність - розроблена система на даному етапі виконує лише базові функції моніторингу мікроклімату теплиці. Інші системи пропонують вже готовий продукт та забезпечують більш широкий спектр функцій, таких як управління теплицею, прогнозування врожаю та контроль якості продукції.

- Цілісність – існуючі системи пропонують комплексне рішення, що включає в собі як встановлення датчиків, так і керування всією системою. В цій роботі представлено тільки веб-інтерфейс, що відповідає за моніторинг.
- Інтеграція з іншими системами – існуючі продукти пропонують готові рішення, тому їх складно інтегрувати з іншими системами. Створена програма може бути інтегрована з іншими частинами, які будуть доповнювати систему для повноцінної роботи.

3.5. Висновки за розділом

У даному розділі було висвітлено ключові аспекти розробленої системи програмного забезпечення для моніторингу мікроклімату в промислових теплицях, що дозволяє визначити її потужності та перспективи подальшого розвитку.

Описано та проаналізовано серверну частину системи, де була надана детальна інформація про структуру та функціональність серверного компонента. Розглянута його здатність до зберігання та обробки даних, а також забезпечення стійкості та ефективності в роботі. Особливий акцент зроблено на тому, як серверна частина взаємодіє з даними мікроклімату, що є важливим аспектом для забезпечення надійності та точності інформації, що надходить до системи. Зазначено, що серверний компонент системи має важливі характеристики, такі як здатність до зберігання та обробки даних, а також забезпечення стійкості та ефективності в роботі.

Описано та вивчено аспекти клієнтської частини системи, яка відіграє важливу роль у взаємодії з користувачем. Висвітлено функціонал клієнтської частини та її взаємодію з користувачем, що визначає зручність та доступність для кінцевого користувача. Аналіз клієнтської частини системи розкрив функціонал та взаємодію з користувачем, яка відіграє ключову роль у забезпеченні зручності та

доступності. Відзначено, як користувач може ефективно взаємодіяти з системою, контролювати та моніторити мікроклімат у теплицях, що сприяє покращенню управління агропродукцією.

Результати тестування свідчать про стабільну роботу системи та її здатність ефективно обробляти та візуалізувати дані мікроклімату.

У контексті перспективних напрямків подальших досліджень висвітлено можливості покращення існуючого функціоналу системи та впровадження нових функціональних можливостей. Це визначає базу для подальшого розвитку системи та її адаптації до зростаючих вимог користувачів.

Перспективні напрямки подальших досліджень вказують на можливості покращення функціоналу системи через вдосконалення та створення нових графічних та функціональних можливостей. Це створить умови для подальшого розвитку системи та забезпечить користувачам більш широкий функціонал у сфері моніторингу та управління мікрокліматом в теплицях. Також система може бути інтегрована з іншими системами управління теплицями, такими як системи освітлення, поливу та вентиляції. Це дозволить забезпечити більш комплексний контроль мікроклімату в теплицях.

Підсумовуючи, висвітлено, що аналіз ключових аспектів розробленої системи свідчить про її високу ефективність та готовність до подальшого вдосконалення. Перспективні напрямки подальших досліджень вказують на великий потенціал для росту та розвитку системи, що важливо в умовах швидко розвиваючого агропромислового сектору.

ВИСНОВКИ

У першому розділі був проведений огляд існуючих систем моніторингу мікроклімату в промислових теплицях. Розглянуті загальні відомості, проведений аналіз сучасних веб-засобів та техніко-функціональних характеристик аналогічних систем. Зокрема, розглянуто автоматизовані системи Appvales, Intellias та Webhmi. Зазначені висновки в цьому розділі дозволяють сформуванню зрозумілу картину сучасного стану систем моніторингу мікроклімату в теплицях і визначити переваги та обмеження різних підходів.

У другому розділі була представлена узагальнена структурно-функціональна організація розробленої системи, включаючи оновлення даних в реальному часі та їх збереження до бази даних. Крім того, розглянуто зчитування вхідних даних, використані методи досліджень, інформаційну модель та алгоритми роботи системи. Висновки цього розділу визначають основні аспекти функціональності, алгоритмів та характеристик досліджуваної системи.

У третьому розділі представлено результати досліджень, включаючи опис розробленої системи програмного забезпечення, результати тестування, перспективні напрямки подальших досліджень та порівняльний аналіз з відомими аналогами. Загальні висновки за цим розділом визначають досягнуті результати та окреслюють шляхи подальшого розвитку системи.

Загалом, розроблена система є цінним доповненням до існуючих систем моніторингу мікроклімату промислових теплиць, однак, важливо продовжувати дослідження та розвиток системи для вдосконалення функціоналу та адаптації до змінних потреб користувачів. Вона може бути використана для підвищення ефективності виробництва та якості продукції в тепличному господарстві.

Список літератури

1. Sensor Systems for Greenhouse Microclimate Monitoring and Control / URL: https://www.researchgate.net/publication/347396578_Sensor_Systems_for_Greenhouse_Microclimate_Monitoring_and_Control_a_Review (дата звернення: 27.09.2023).
2. Mukazhanov, Y., Kamshat, Z., Assel, O., Shayhmetov, N., & Alimbaev, Microclimate control in greenhouses. In: International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management, 2017. – 704с.
3. N. Katsoulas, K.P. Ferentinos, A. Tzounis, T. Bartzanas, C. Kittas, Spatially distributed greenhouse climate control based on wireless sensor network measurements, Model. Conf. Wageningen, Netherlands. 2015. – 1154с.
4. Azaza, M.; Tanougast, C.; Fabrizio, E.; Mami, A. Smart greenhouse fuzzy logic-based control system enhanced with wireless data monitoring. ISA Trans., 2016. – 497с.
5. Zhang, K.S.; Zhang, X.W.; Zhou, Y.; Tang, W. Design of agricultural greenhouse environment monitoring system based on internet of things technology. In Advanced Materials Research; Trans Tech Publications Ltd.: Beijing, China, 2013. – 1655с.
6. Danita, M.; Mathew, B.; Shereen, N.; Sharon, N.; Paul, J.J. IoT Based Automated Greenhouse Monitoring System. In Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018. – 1937с.
7. Automated Greenhouse Monitoring Systems / URL: <https://appvales.com/blog/automated-greenhouse-monitoring-systems/> (дата звернення: 30.09.2023).

8. Automated Greenhouse System with Profound Analytics / URL: <https://intellias.com/automated-greenhouse-system/> (дата звернення: 05.10.2023).
9. Monitoring system for greenhouses / URL: http://webhmi.com.ua/en/portfolio_page/monitoring-system-for-greenhouses/ (дата звернення: 05.10.2023).
10. Priva Nutri-Line / URL: <https://agropeak.com.ua/priva-nutri-line/> (дата звернення: 05.10.2023).
11. Ullah, I.; Fayaz, M.; Aman, M.; Kim, D. An optimization scheme for IoT based smart greenhouse climate control with efficient energy consumption. *Computing* 2022. – 437с.
12. Real-time ASP.NET with SignalR / URL: <https://dotnet.microsoft.com/en-us/apps/aspnet/signalr> (дата звернення: 06.10.2023).
13. Kitouni, I.; Benmerzoug, D.; Lezzar, F. Smart agricultural enterprise system based on integration of internet of things and agent technology. *J. Organ. End User Comput.* 2018. – 398с.
14. SQL Server Management Studio / URL: <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16> (дата звернення: 08.10.2023).
15. Filatov, V., & Doskalenko, S. On the approach to searching for functional dependences of data in relational systems. *Innovative Technologies and Scientific Solutions for Industries*, 2018.– 128с
16. Kosenko, V. (2017), "Principles and structure of the methodology of risk-adaptive management of parameters of information and telecommunication networks of critical application systems", *Innovative Technologies and Scientific Solutions for Industries*, 2017.– 561с.
17. CsvHelper / URL: <https://joshclose.github.io/CsvHelper/> (дата звернення: 09.10.2023).

18. Druma A.M. Dynamic Climate Model of a Greenhouse. Report Geothermal Training Programme. Reykjavik, 2019. – 309с.

19. Build with .NET / URL: <https://dotnet.microsoft.com/en-us/> (дата звернення: 10.10.2023).

20. ASP.NET Core Blazor / URL: https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-8.0&WT.mc_id=dotnet-35129-website (дата звернення: 10.10.2023).

21. What is SQL / URL: [https://aws.amazon.com/what-is/sql/#:~:text=Structured%20query%20language%20\(SQL\)%20is,relationships%20between%20the%20data%20values.](https://aws.amazon.com/what-is/sql/#:~:text=Structured%20query%20language%20(SQL)%20is,relationships%20between%20the%20data%20values.) (дата звернення: 03.10.2023).

22. Visual Studio section / URL: <https://visualstudio.microsoft.com/#vs-section> (дата звернення: 10.10.2023).

23. A. P. Kalogeras, J. V. Gialelis, C. E. Alexakos, M. J. Georgoudakis and S. A. Koubias, "Vertical integration of enterprise industrial systems utilizing web services", 2006. – 1028с.

24. H. Hacigümüş, B. Iyer, C. Li and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model", 2002. – 512с.

25. G. S. Seyboth, D. V. Dimarogonas, K. H. Johansson, P. Frasca, and F. Allgwer, "On robust synchronization of heterogeneous linear multiagent systems with static couplings," *Automatica*, 2018. – 736с.

26. Montoya, A.P.; Obando, F.A.; Osorio, J.A.; Morales, J.G.; Kacira, M. Design and Implementation of a Low-Cost Sensor Network to Monitor Environmental and Agronomic Variables in a Plant Factory. *Comput. Electron. Agric.* 2020. – 178с.

27. Microservices Using ASP.NET Core / URL: <https://www.c-sharpcorner.com/article/microservice-using-asp-net-core/> (дата звернення: 11.10.2023).

28. Ferrag, M.A.; Shu, L.; Yang, X.; Derhab, A.; Maglaras, L. Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges, 2020. – 1349с.

29. Manoharan, S. Supervised Learning for Microclimatic parameter Estimation in a Greenhouse environment for productive Agronomics. J. Artif. Intell. Capsul. Netw. 2020. – 509с.

30. CQRS pattern / URL: <https://learn.microsoft.com/en-us/azure/architecture/patterns/cqrs> (дата звернення: 14.10.2023).

31. MediatR in ASP.NET Core / URL: <https://medium.com/dotnet-hub/use-mediatr-in-asp-net-or-asp-net-core-cqrs-and-mediator-in-dotnet-how-to-use-mediatr-cqrs-aspnetcore-5076e2f2880c> (дата звернення: 14.10.2023).

32. Learn Dapper / URL: <https://www.learnmapper.com/> (дата звернення: 15.10.2023).

ВИХІДНИЙ КОД ЗАВАНТАЖЕННЯ ФАЙЛІВ

```
[ApiController]
[Route("api/greenhouseFile")]
public class GreenhouseFileController : ControllerBase
{
    private readonly IMediator _mediator;

    public GreenhouseFileController(IMediator mediator)
    {
        _mediator = mediator;
    }

    [HttpPost]
    public async Task<IActionResult> UploadData(byte[] fileBytes)
    {
        var result = await _mediator.Send(new UploadGreenhouseDataCommand
        {
            FileBytes = fileBytes
        });

        return Ok(result);
    }
}

public record UploadGreenhouseDataCommand : ICommand<Unit>
{
    public byte[] FileBytes { get; init; }
}

internal class GetGreenhouseCsvTestCommandHandler : IRequestHandler<UploadGreenhouseDataCommand,
Result<Unit>>
{
    private readonly ICsvImporter _csvImporter;
```

```

public GetGreenhouseCsvTestCommandHandler(ICsvImporter csvImporter)
{
    _csvImporter = csvImporter;
}

public async Task<Result<Unit>> Handle(UploadGreenhouseDataCommand request, CancellationToken
cancellationToken)
{
    await _csvImporter.ImportCsv(request.FileBytes, cancellationToken);

    return Unit.Value;
}
}

public class GreenHouseCsvParser : ICsvImporter
{
    private readonly IEntityRepository<Greenhouse> _greenhouseEntityRepository;

    public GreenHouseCsvParser(IEntityRepository<Greenhouse> greenhouseEntityRepository)
    {
        _greenhouseEntityRepository = greenhouseEntityRepository;
    }

    public async Task ImportCsv(byte[] fileBytes, CancellationToken cancellationToken)
    {
        var config = new CsvConfiguration(CultureInfo.InvariantCulture)
        {
            HasHeaderRecord = false
        };

        try
        {
            using var memoryStream = new MemoryStream(fileBytes);
            using var streamReader = new StreamReader(memoryStream);
            using var csvReader = new CsvReader(streamReader, config);

```

```
csvReader.Context.RegisterClassMap<GreenhouseScvMapper>();

var records = csvReader
    .GetRecords<Greenhouse>()
    .ToList();

foreach (var record in records)
{
    await _greenhouseEntityRepository.AddAsync(record, cancellation token);
}
catch (Exception e)
{
    Console.WriteLine(e);
    throw;
}
}
```

ОТРИМАННЯ ДАНИХ З БАЗИ ДАНИХ

```

public class GreenhouseRepository :
    IReadOnlyRepository<GreenhouseFilter, IReadOnlyCollection<GreenhouseResponse>>,
    IReadOnlyRepository<UnreadGreenhouseFilter, GreenhouseIdResponse?>
{
    private readonly IDbConnection _dbConnection;

    public GreenhouseRepository(IDbConnection dbConnection)
    {
        _dbConnection = dbConnection;
    }

    public async Task<IReadOnlyCollection<GreenhouseResponse>> ReadAsync(GreenhouseFilter filter,
        CancellationToken cancellationToken)
    {
        string where = filter.Id is null
            ? "[Greenhouse].[hasBeenRead] = 1"
            : "[Greenhouse].[id] = @id";

        string query = @$"SELECT TOP(@take)
            [Greenhouse].[id],
            [Greenhouse].[co2Concentration],
            [Greenhouse].[luminance],
            [Greenhouse].[dateCreated],
            [Greenhouse].[dateUpdated],
            [Air].[id],
            [Air].[temperature],
            [Air].[humidity],
            [Atmosphere].[id],
            [Atmosphere].[pressure],
            [Sensors].[id],
            [Sensors].[ph],
            [Sensors].[ec],
            [Sensors].[orp],
            [Sensors].[r1],
            [Sensors].[r2],
            [Sensors].[r3],
            [Sensors].[r4],
            [Sensors].[r5],
            [Sensors].[r6],
            [Soil].[id],
            [Soil].[temperature],
            [Soil].[averageHumidity1],
            [Soil].[averageHumidity2],
            [Water].[id],
            [Water].[temperature]
        FROM [Greenhouse]
        INNER JOIN [Air]
            ON [Greenhouse].[id] = [Air].[greenhouseId]
        INNER JOIN [Atmosphere]
            ON [Greenhouse].[id] = [Atmosphere].[greenhouseId]
        INNER JOIN [Sensors]

```

```

        ON [Greenhouse].[id] = [Sensors].[greenhouseId]
        INNER JOIN [Soil]
        ON [Greenhouse].[id] = [Soil].[greenhouseId]
        INNER JOIN [Water]
        ON [Greenhouse].[id] = [Water].[greenhouseId]
        WHERE {where}
        ORDER BY [Greenhouse].[dateUpdated] DESC, [Greenhouse].[dateCreated] DESC";

var responses = (await _dbConnection
    .QueryAsync<
        GreenhouseResponse,
        AirResponse,
        AtmosphereResponse,
        SensorResponse,
        SoilResponse,
        WaterResponse,
        GreenhouseResponse>(
        query,
        (greenhouse, air, atmosphere, sensor, soil, water) =>
            new GreenhouseResponse
            {
                Air = air,
                Atmosphere = atmosphere,
                Sensor = sensor,
                Soil = soil,
                Water = water,
                Co2Concentration = greenhouse.Co2Concentration,
                Luminance = greenhouse.Luminance,
                DateCreated = greenhouse.DateCreated,
                DateUpdated = greenhouse.DateUpdated
            },
            filter))
    .ToList();

return responses;
}

public async Task<GreenhouseIdResponse?> ReadAsync(UnreadGreenhouseFilter filter, CancellationToken
cancellationToken)
{
    const string query = @"SELECT TOP(1)
        [Greenhouse].[id]
        FROM [Greenhouse]
        WHERE [Greenhouse].[hasBeenRead] = 0
        ORDER BY [Greenhouse].[dateUpdated] ASC, [Greenhouse].[dateCreated] ASC";

    var response = await _dbConnection.QueryFirstOrDefaultAsync<GreenhouseIdResponse>(query,
cancellationToken);

    return response;
}
}

```

ОНОВЛЕННЯ ДАНИХ В РЕАЛЬНОМУ ЧАСІ

```

@if (_greenhouses.Any())
{
  <div class="table-container">
    <div class="table-scroll">
      <table class="table table-bordered">
        <thead>
          <tr>
            <th style="min-width: 110px">Data</th>
            <th>Co2 concentration (V)</th>
            <th>Air temperature (*C)</th>
            <th>Air humidity (%)</th>
            <th>Soil temperature (*C)</th>
            <th>Avg soil humidity 1 (%)</th>
            <th>Avg soil humidity 2 (%)</th>
            <th>Water temperature (*C)</th>
            <th>Atmospheric pressure (mmHg)</th>
            <th>Luminance (Lux)</th>
            <th>PH (V)</th>
            <th>EC (V)</th>
            <th>ORP (V)</th>
            <th>R1</th>
            <th>R2</th>
            <th>R3</th>
            <th>R4</th>
            <th>R5</th>
            <th>R6</th>
          </tr>
        </thead>
        <tbody>
          @foreach (var greenhouse in _greenhouses.OrderByDescending(x => x.DateUpdated ?? x.DateCreated))
          {
            <tr>
              <td>
                @(DateTimeOffset
                  .FromUnixTimeSeconds(greenhouse.DateUpdated ?? greenhouse.DateCreated)
                  .ToString("MMM d, yyyy HH:mm:ss"))
              </td>
              <td>@greenhouse.Co2Concentration</td>
              <td>@greenhouse.Air?.Temperature</td>
              <td>@greenhouse.Air?.Humidity</td>
              <td>@greenhouse.Soil?.Temperature</td>
              <td>@greenhouse.Soil?.AverageHumidity1</td>
              <td>@greenhouse.Soil?.AverageHumidity2</td>
              <td>@greenhouse.Water?.Temperature</td>
              <td>@greenhouse.Atmosphere?.Pressure</td>
              <td>@greenhouse.Luminance</td>
              <td>@greenhouse.Sensor?.Ph</td>
              <td>@greenhouse.Sensor?.Ec</td>
              <td>@greenhouse.Sensor?.Orp</td>
              <td>@(greenhouse.Sensor?.R1 ?? false ? EnabledSensor : DisabledSensor)</td>
              <td>@(greenhouse.Sensor?.R2 ?? false ? EnabledSensor : DisabledSensor)</td>
              <td>@(greenhouse.Sensor?.R3 ?? false ? EnabledSensor : DisabledSensor)</td>
            </tr>
          }
        </tbody>
      </table>
    </div>
  </div>
}

```

```

        <td>@(greenhouse.Sensor?.R4 ?? false ? EnabledSensor : DisabledSensor)</td>
        <td>@(greenhouse.Sensor?.R5 ?? false ? EnabledSensor : DisabledSensor)</td>
        <td>@(greenhouse.Sensor?.R6 ?? false ? EnabledSensor : DisabledSensor)</td>
    </tr>
    }
</tbody>
</table>
</div>
</div>
}
@code {
    const string EnabledSensor = "On";
    const string DisabledSensor = "Off";

    private readonly CancellationTokenSource _cancellationToken = new();
    private HubConnection? _hubConnection;

    private List<GreenhouseResponse> _greenhouses = new();

    protected override async Task OnInitializedAsync()
    {
        _greenhouses = (await GreenhouseRepository.ReadAsync(new GreenhouseFilter
            {
                Take = 50
            },
            _cancellationToken.Token))
            .ToList();

        var baseUrl = ApplicationConfig.BaseUrl.TrimEnd('/');

        _hubConnection = new HubConnectionBuilder()
            .WithUrl($"{baseUrl}{GreenhouseNotificationsHub.HubUrl}")
            .WithAutomaticReconnect()
            .Build();

        _hubConnection.On<GreenhouseIdResponse>(nameof(IGreenhouseNotificationsClient.ReceiveNotification), async
message =>
        {
            var greenhouseFilter = new GreenhouseFilter
            {
                Id = message.Id,
                Take = 1
            };

            var greenhouses = (await GreenhouseRepository.ReadAsync(greenhouseFilter, _cancellationToken.Token))
                .ToList();

            _greenhouses.AddRange(greenhouses);

            await InvokeAsync(StateHasChanged);
        });

        await _hubConnection.StartAsync();
    }

    public void Dispose()
    {
        _cancellationToken.Cancel();
    }
}

```



```
        _cancellationToken.Dispose();
    }

    public async ValueTask DisposeAsync()
    {
        if (_hubConnection is not null)
        {
            await _hubConnection.DisposeAsync();
        }
    }
}
```

ДОДАТОК Г

ПРОГРАМНИЙ КОД СТОРІНКИ ЗАВАНТАЖЕННЯ ФАЙЛУ

```
<h1>Greenhouse data export</h1>
```

Select file to upload greenhouse data.

```
<InputFile OnChange="@UploadGreenhouseData"
  style="margin-top:10px"
  class="form-control"
  accept=".xml, .xmlns, .csv" />
```

```
@if (_resultIsSuccess)
{
  <div style="margin-top:5px">
    <h4>File has been loaded successfully.</h4>
  </div>
}
```

```
@if (_errors.Count > 0)
{
  <div style="margin-top:5px">
    <h2>Errors</h2>
    <ul class="text-danger">
      @foreach (var error in _errors)
      {
        <li>@error</li>
      }
    </ul>
  </div>
}
```

```
@code{
  private List<string> _errors = new();
  private long _maxFileSize = 1024 * 1024 * 50; // represents 50MB
  private HubConnection? _hubConnection;
  private bool _resultIsSuccess;

  protected override async Task OnInitializedAsync()
  {
    var baseUrl = ApplicationConfig.BaseUrl.TrimEnd('/');

    _hubConnection = new HubConnectionBuilder()
      .WithUrl($"{baseUrl}{GreenhouseNotificationsHub.HubUrl}")
      .WithAutomaticReconnect()
      .Build();

    _hubConnection.On<bool>(nameof(IGreenhouseNotificationsClient.ReceiveStatusNotification), message =>
    {
      _resultIsSuccess = message;
      StateHasChanged();
      StartTimerAsync(3);
    });

    await _hubConnection.StartAsync();
  }
}
```

```

}

private async Task UploadGreenhouseData(InputFileChangeEventArgs e)
{
    _errors.Clear();

    try
    {
        var file = e.File.OpenReadStream(_maxFileSize);
        var fileBytes = await GetBytesFromStream(file);

        await GreenhouseFileController.UploadData(fileBytes);
    }
    catch (Exception ex)
    {
        _errors.Add($"File: {e.File.Name} Error: {ex.Message}");
        _resultIsSuccess = false;
    }
}

private static async Task<byte[]> GetBytesFromStream(Stream file)
{
    using var memoryStream = new MemoryStream();
    await file.CopyToAsync(memoryStream);

    return memoryStream.ToArray();
}

private async void StartTimerAsync(int time)
{
    while (time > 0)
    {
        time--;
        await Task.Delay(1000);
    }

    _resultIsSuccess = false;
    StateHasChanged();
}
}

```

ДОДАТОК Д

ПЕРЕЛІК ФАЙЛІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файлу	Опис
Пояснювальні документи	
Диплом_Юдін_О.С.doc	Пояснювальна записка до проекту. Документ Word.
Диплом_Юдін_О.С.pdf	Пояснювальна записка до проекту в форматі PDF.
Програма	
Program.rar	Архів, що містить коди програми для запуску проекту.
Презентація	
Презентація_Юдін.ppt	Презентація для проекту.