


ДОДАТОК А

Слайди презентації



Кваліфікаційна робота бакалавра

Тема: «Програмна система для керування вирощуванням культур у сільському господарстві»

Виконала: ст. гр. ПЗПз-21-1 Яремчук К.О.
Керівник роботи: доц. Бабій А.С.

Рисунок А.1 – Слайд 1

2

Мета роботи та об'єкт розробки

Мета роботи
Розробка інтелектуальної програмної системи для автоматизації та оптимізації процесу вирощування сільськогосподарських культур, що забезпечує автоматичне планування догляду за рослинами, оптимізацію розміщення культур на полях, підбір оптимального набору рослин з урахуванням ґрунтово-кліматичних умов, підвищення врожайності, зниження витрат і зменшення впливу на довкілля.

Об'єкт розробки
Програмна система для керування вирощуванням культур, яка включає модулі для аналізу даних про ґрунт, клімат і культури, формування календаря агротехнічних заходів, оптимізації розміщення рослин, рекомендацій щодо вибору культур та зручного інтерфейсу для фермерів.

Рисунок А.2 – Слайд 2

Аналіз предметної області та актуальність теми ³

Аналіз предметної галузі

Сільське господарство – ключова галузь економіки, що забезпечує продовольчу безпеку, формує ВВП та підтримує екологічний і соціальний розвиток. Глобальні виклики, такі як зміна клімату, скорочення родючих земель і зростання населення, вимагають модернізації шляхом впровадження інформаційних технологій. Процес вирощування культур охоплює підготовку ґрунту, вибір культур, планування сівозміни, агротехнічні заходи та моніторинг. Традиційні методи, що базуються на досвіді фермерів, часто неефективні через нестабільний клімат і брак даних. Smart Farming (розумне фермерство) інтегрує GIS, IoT, Big Data, штучний інтелект і мобільні платформи для підвищення продуктивності.

Рисунок А.3 – Слайд 3

Аналіз предметної області та актуальність теми ⁴

Проблеми та рішення

- **Низька автоматизація:** ручне планування знижує врожайність і підвищує витрати.
- ✓ Рішення: автоматизована система для планування, моніторингу та аналізу.
- **Відсутність моніторингу:** брак даних про ґрунт і рослини призводить до втрат.
- ✓ Рішення: інтеграція з датчиками та супутниковими системами.
- **Складність документації:** розпорошеність даних ускладнює звітність.
- ✓ Рішення: автоматизований документообіг і журнали.
- **Недостатнє прогнозування:** відсутність аналізу даних підвищує ризики.
- ✓ Рішення: аналітичні модулі з машинним навчанням.
- **Людський фактор:** помилки в дозуванні та термінах.
- ✓ Рішення: автоматичні нагадування та контроль.
- **Економічна ефективність:** складність оцінки рентабельності.
- ✓ Рішення: модулі фінансової аналітики.

Рисунок А.4 – Слайд 4

Аналіз предметної області та актуальність теми 5

Актуальність теми

Розробка програмної системи для управління вирощуванням культур є актуальною, оскільки вона:

- автоматизує планування, моніторинг і облік;
- оптимізує розміщення культур і вибір рослин за умовами середовища;
- підвищує врожайність, знижує витрати та мінімізує вплив на довкілля;
- адаптована для малих і середніх господарств, забезпечуючи доступність і зручність.

Така система сприятиме цифровій трансформації агросектору, підвищенню конкурентоспроможності та сталому розвитку.

Рисунок А.5 – Слайд 5

Постановка задачі кваліфікаційної роботи 6

Мета

Розробка програмної системи для автоматизації планування, моніторингу та аналізу вирощування культур, що підвищує ефективність агровиробництва, знижує втрати та оптимізує ресурси.

Завдання

- Аналіз аграрного середовища та потреб фермерів.
- Створення бази знань про культури (біологічні особливості, потреби).
- Розробка модуля автоматичного календаря догляду (полив, підживлення, захист).
- Оптимізація розміщення культур за умовами ґрунту та світла.
- Рекомендації щодо вибору культур за параметрами середовища.
- Створення зручного веб-інтерфейсу з нагадуваннями.
- Забезпечення гнучкості та масштабованості системи.
- Автоматизований моніторинг і журнал історичних даних.

Результат

Цифровий помічник фермера, що автоматизує рутинні задачі, підвищує продуктивність і адаптується до умов агросектору.

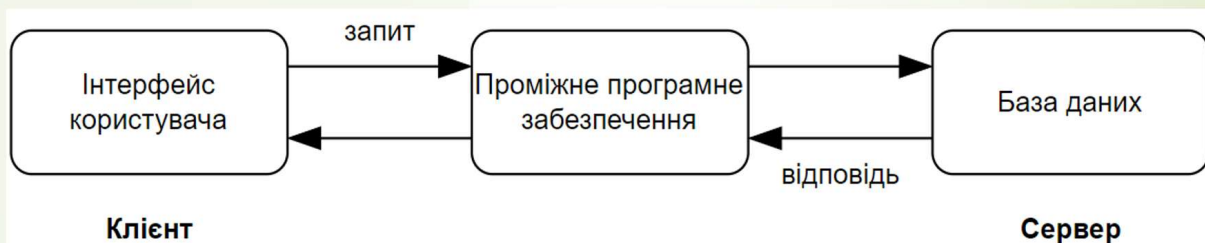
Рисунок А.6 – Слайд 6

Функціональні можливості програмної системи 7

- **Реєстрація та автентифікація:** створення облікового запису, авторизація, скидання пароля, ролі користувачів (фермер, агроном, адміністратор).
- **Керування ділянками:** додавання/редагування ділянок (площа, геолокація, тип ґрунту, освітленість).
- **Додавання культур:** вибір із бази знань, налаштування параметрів (вегетація, полив, добрива, умови).
- **Календар догляду:** автоматичне створення графіку агрозаходів (полив, підживлення, захист) із нагадуваннями.
- **Оптимізація розміщення:** пропозиція раціонального розташування культур за параметрами ґрунту, світла, сумісності.
- **Підбір культур:** рекомендації культур за типом ґрунту, кліматом, зрошенням.
- **Моніторинг операцій:** відстеження виконання, адаптація графіку, аналітика.
- **База знань:** оновлення/редагування культур адміністратором, перегляд користувачами.
- **Звіти:** генерація звітів про роботи, експорт у PDF/CSV.

Рисунок А.7 – Слайд 7

Архітектура програмної системи 8



Трирівнева клієнт-серверна архітектура

- **Клієнтський рівень:** інтерфейс користувача (React.js) для введення даних і відображення результатів.
- **Серверний рівень:** FastAPI (Python) обробляє запити, реалізує бізнес-логіку, перевіряє доступ.
- **Рівень даних:** MongoDB зберігає гнучкі дані (ділянки, культури, календарі).

Логіка взаємодії

Запит → Обробка (FastAPI) → База даних (MongoDB) → Відповідь (JSON).

Рисунок А.8 – Слайд 8

Приклад найцікавішого фрагменту програмного коду:
оптимальне розміщення рослин на полі

```
def place_plants_on_field(field_grid, plant_list, compatibility_matrix):
    field = [[None for _ in range(len(field_grid[0]))] for _ in range(len(field_grid))]
    for plant in plant_list:
        for i in range(len(field_grid)):
            for j in range(len(field_grid[0])):
                cell = field_grid[i][j]
                if cell['soil'] == plant['soil'] and cell['light'] >= plant['light']:
                    neighbors = get_neighbors(field, i, j)
                    if all(compatibility_matrix.get((plant['type'], n), True) for n in neighbors):
                        field[i][j] = plant['type']
                        break
    return field

def get_neighbors(grid, x, y):
    neighbors = []
    for dx in [-1, 0, 1]:
        for dy in [-1, 0, 1]:
            if dx == 0 and dy == 0:
                continue
            nx, ny = x + dx, y + dy
            if 0 <= nx < len(grid) and 0 <= ny < len(grid[0]) and grid[nx][ny]:
                neighbors.append(grid[nx][ny])
    return neighbors
```

Рисунок А.9 – Слайд 9

Діаграма прецедентів

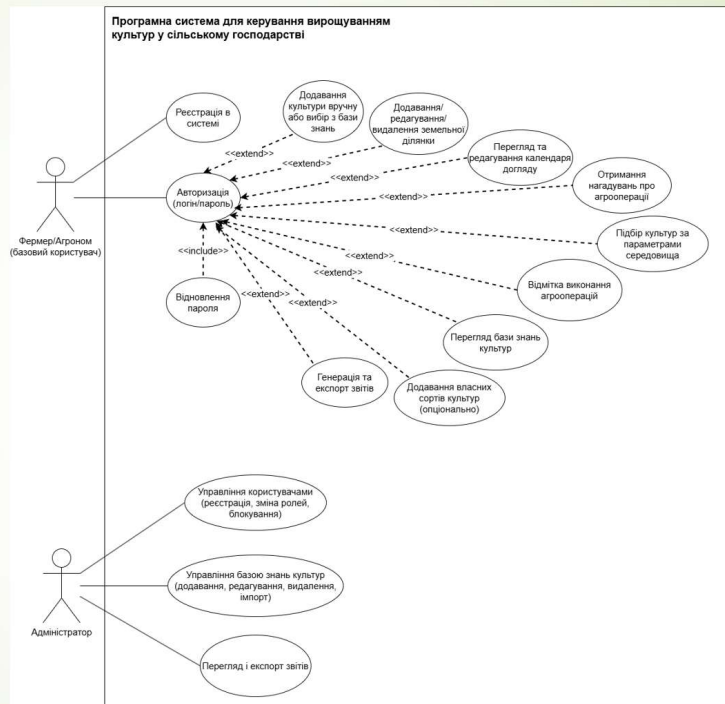


Рисунок А.10 – Слайд 10

Entity-Relationship Diagram структури даних системи управління агровиросуванням

11

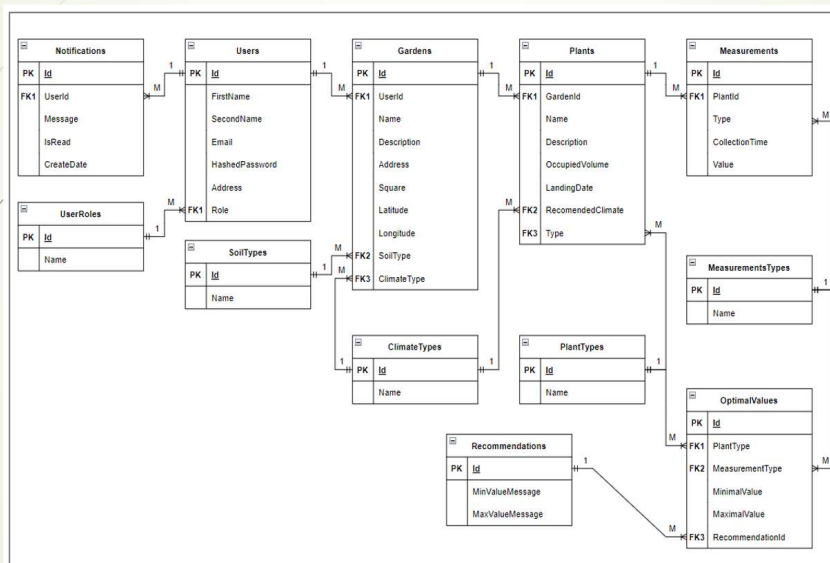


Рисунок А.11 – Слайд 11

Інтерфейс користувача: сторінка оптимального розміщення культур на полі

12

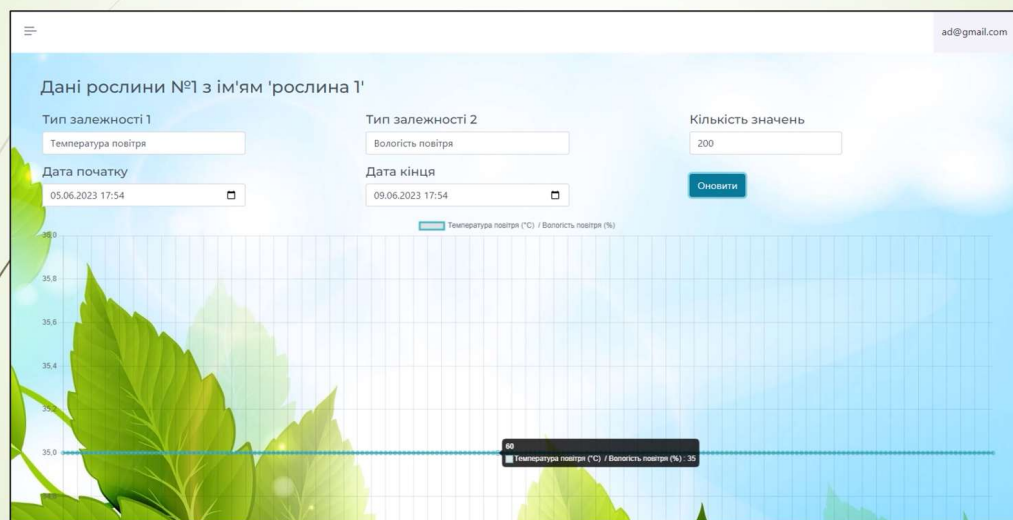


Рисунок А.12 – Слайд 12

Інтерфейс користувача: сторінка підбору культур за параметрами середовища

13

№	Назва	Опис	Займана площа (см ²)	Висота (см)	Графіки
	<input type="text" value="рослина 1"/>	<input type="text" value="опис рослин 1"/>			<input type="button" value="Згорнути"/>
Тип рослини		Рекомендований клімат	Дата посадки		
<input type="text" value="Польовий"/>		<input type="text" value="Континентальний тропічний"/>	<input type="text" value="07.06.2023, 17:52:00"/>		
Займана площа		Висоота	Користувач		
<input type="text" value="10 см²"/>		<input type="text" value="13 см"/>	<input type="text" value="user1 user1"/>		
Сад					
<input type="text" value="№1 Назва: сад1"/>					
<input type="button" value="Список вимірювань"/> <input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>					

Рисунок А.13 – Слайд 13

Інтерфейс користувача: сторінка відміток у календарі та звітності

14

№	Тип	Значення	Назва рослини	Редагувати	Видалити
1	Температура повітря	10 °C	рослина 1	<input type="button" value="Редагувати"/>	<input type="button" value="Видалити"/>

Рисунок А.14 – Слайд 14

Використані технології

15

Технологічний стек

Фронтенд:

- React.js – для створення динамічного, компонентно-орієнтованого веб-інтерфейсу (SPA) з React Router, Redux Toolkit, Axios, Bootstrap.

Бекенд:

- Python, FastAPI – для реалізації RESTful API з асинхронною обробкою запитів, JWT-автентифікацією.

База даних:

- MongoDB – документо-орієнтована NoSQL для гнучкого зберігання даних (користувачі, ділянки, культури, календарі).

Тестування:

- Selenium, Postman, Jest, React Testing Library для юніт-тестів і API-тестування.

UI/UX

- Flat design, адаптивність (Bootstrap), локалізація (i18next), WCAG 2.1.
- Інтерактивні компоненти: календарі (react-calendar), карти (Leaflet/Mapbox), звіти (PDF/CSV).

Рисунок А.15 – Слайд 15

Результати першого функціонального тестування

16

Перше тестування:
83.33% успіху,
помилки в календарі
та розміщенні.

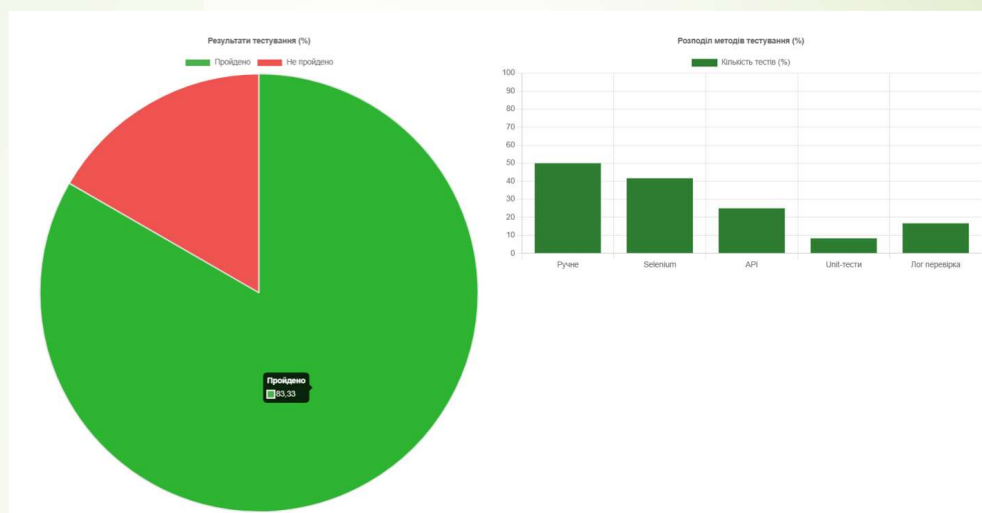


Рисунок А.16 – Слайд 16

Результати другого функціонального тестування після виправлення

17

Друге тестування:
100% успіху,
стабільність
підтверджена.

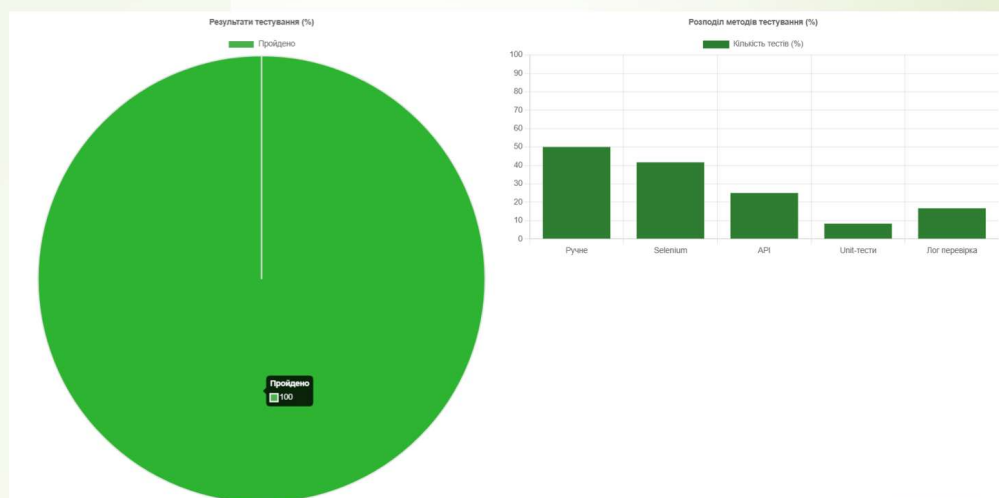


Рисунок А.17 – Слайд 17

Підсумки

18

Реалістичність та корисність отриманих результатів

- Розроблена система повністю відповідає концепції Smart Farming та актуальним трендам цифровізації агросектору.
- Автоматизація ключових агротехнічних операцій і зниження впливу людського чинника підтверджено функціональним тестуванням (Selenium, Postman, unit-тести, ручна перевірка).
- Інтелектуальні модулі (календар догляду, рекомендаційна система k-NN, моніторинг) демонструють реальну користь для планування та оптимізації ресурсів.

Можливості використання

- Фермерські господарства різного масштабу – від невеликих приватних до великих агропідприємств.
- Освітні та наукові установи – як демонстраційний та аналітичний інструмент для студентів і дослідників.
- Інтеграція з IoT-пристроями та погодними сервісами – для розширеного моніторингу та адаптивного управління.
- Гнучка адаптація під локальні ґрунтово-кліматичні умови завдяки модульній архітектурі на базі React.js, FastAPI та MongoDB.

Можливий розвиток програмного забезпечення

- Інтеграція сенсорних пристроїв (датчики вологості, освітленості, pH ґрунту).
- Алгоритми прогнозування врожайності на основі машинного навчання (Time-series, Deep Learning).
- Мобільний додаток для оперативного доступу та сповіщень.
- Розширення бази знань: додавання нових культур, агротехнічних практик, фінансових модулів для планування витрат.
- Аналітична панель з дашбордами й інтерактивними звітами в реальному часі.

Отже, програмна система забезпечує підвищення продуктивності, зниження витрат і підтримку сталого розвитку аграрного виробництва.

Рисунок А.18 – Слайд 18

ДОДАТОК Б

Звіт результатів перевірки на унікальність тексту



Дата звіту 6/7/2025

Дата редагування ---



Звіт не був оцінений

Звіт подібності

метадані

Назва організації

Kharkiv National University of Radio Electronics

Заголовок

2025_Б_ПІ_ПЗПіз-21-1_Яремчук_К_О_ скорочений

Автор

Науковий керівник / Експерт

Яремчук Катерина Олександрівна Олена Олійник

підрозділ

каф. ПІ

Обсяг знайдених подібностей

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



КП 1



КЦ

25

Довжина фрази для коефіцієнта подібності 2

11550

Кількість слів

96324

Кількість символів

Тривога

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

Заміна букв		0
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		1

Подібності за списком джерел

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

10 найдовших фраз

Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://openarchive.nure.ua/server/api/core/bitstreams/e84c9972-be3f-41bb-a499-69f315a08e4c/content	11 0.10 %
2	«Управління кадровим резервом підприємства: інструменти і практичні аспекти» 1/20/2025 Institute of Economics and Technology named after Robert Elworthy (Institute of Economics and Technology named after Robert Elworthy)	10 0.09 %

з бази даних RefBooks (0.00 %)