Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Вебсистема управління фермерським господарством з використанням штучного інтелекту
(тема)

Виконав:
здобувач четвертого року навчання,
групи ІТШІ-21-2
Владислав Путятин
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Штучний інтелект
(повна назва освітньої програми)
Керівник доц. Євген Павленко
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Штучний інтелект
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Путятину Владиславу Євгеновичу
(прізвище, ім'я, по батькові)

1. Тема роботи Вебсистема управління фермерським господарством з використанням штучного інтелекту

затверджена наказом університету від 19 травня 20 25 р. № 378Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25 червня 20 25 р.

3. Вихідні дані до роботи Python, HTML, CSS, JavaScript, Bootstrap, PyCharm, FastAPI, OpenWeatherAPI, ClaudeAI, OpenRouter, Leaflet, LLM, MlPRegressor, XGBRegressor, Інтернет ресурси, наукові статті

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі та постановка задачі
2) Теоретичні дослідження
3) Проектування та розробка вебсистеми
4) Вебсистема управління фермерським господарством з використанням штучного інтелекту

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	19.05.2025	виконано
2	Аналіз предметної галузі	20.05 – 24.05	виконано
3	Постановка задачі	25.05 – 26.05	виконано
4	Проведення теоретичних досліджень	27.05 – 29.05	виконано
5	Обґрунтування вибору використаних технологій	30.05 – 01.06	виконано
6	Реалізація проекту	02.06 – 10.06	виконано
7	Оформлення пояснювальної записки	10.06 – 18.06	виконано
8	Підготовка до захисту	18.06 – 24.06	виконано
9	Захист перед ЕК	25.06.2025	

Дата видачі завдання 19 травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Євген Павленко
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 89 с., 19 рис., 2 табл., 4 дод., 21 джерело.

АГРАРНА АНАЛІТИКА, ВЕБСИСТЕМА, ОБЛІК, ФЕРМЕРСЬКЕ ГОСПОДАРСТВО, ЧАТ-БОТ, ШТУЧНИЙ ІНТЕЛЕКТ, CLAUDE API, CSS, FLASK, HTML, JAVASCRIPT, MACHINE LEARNING, OPENROUTER, OPENWEATHERMAP API, PYTHON, SQLALCHEMY.

Об’єкт дослідження – цифрові інформаційні системи для управління фермерським господарством.

Предмет дослідження – вебсистема для управління фермерським господарством, яка використовує штучний інтелект для прогназування та інтерактивної взаємодії з користувачем.

Мета роботи – розробка та впровадження інтелектуальної вебсистеми, яка забезпечує ефективне управління аграрною діяльністю шляхом інтеграції штучного інтелекту, метеоаналізу, обліку та картографії.

Методи дослідження – аналіз предметної галузі, проектування архітектури інформаційної системи, використання методів машинного навчання, реалізація API-взаємодії з Claude AI та OpenWeatherMap, розробка клієнт-серверної інфраструктури з використанням Flask та Leaflet.js.

В результаті дослідження була створена інтелектуальна веб-система для управління фермерським господарством. Основна ідея полягає у створенні зручного середовища для фермерів, що дозволяє приймати обґрунтовані рішення на основі даних. Реалізовано компоненти машинного навчання для прогнозу урожайності та росту тварин, інтегровано погодне API та чат-бот для взаємодії з користувачем. Робота демонструє практичну користь цифрових технологій у сільському господарстві та доводить ефективність поєднання ШІ.

ABSTRACT

Bachelor’s thesis contains: 89 pp., 19 fig., 2 tabl., 4 ann., 21 references.

ACCOUNTING, AGRICULTURAL ANALYTICS, ARTIFICIAL INTELLIGENCE, CHATBOT, CLAUDE API, CSS, FARM MANAGEMENT, FLASK, HTML, JAVASCRIPT, MACHINE LEARNING, OPENROUTER, OPENWEATHERMAP API, PYTHON, SQLALCHEMY, WEB SYSTEM.

Object of the study is digital information systems for farm management.

Subject of the study is a web-oriented farm management system that uses artificial intelligence to forecast crop yields and livestock weight gain, automate accounting, support decision-making, and enable interactive user communication.

Goal of the project is the development and implementation of an intelligent web-based system that ensures efficient agricultural management through the integration of artificial intelligence, weather analysis, record-keeping, and mapping.

Research methods include analysis of the subject area, design of the information system architecture, use of machine learning methods, implementation of API integration with Claude AI and OpenWeatherMap, and development of a client-server infrastructure using Flask and Leaflet.js.

Results show that an intelligent web system for farm management was developed. The system combines tools for accounting, forecasting, weather analytics, and map-based visualization. The core idea is to create a convenient environment for farmers that supports data-driven decision-making. Machine learning components were implemented for yield and livestock growth prediction, and a weather API and chatbot were integrated for user interaction. The project demonstrates the practical value of digital technologies in agriculture and confirms the effectiveness of combining AI, cartography, and process automation in a single web application.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект;

AI – Artificial Intelligence – штучний інтелект;

API – Application Programming Interface – інтерфейс прикладного програмування, що дозволяє підключати сторонні сервіси;

CSS – Cascading Style Sheets – мова опису стилів для HTML-документів;

Flask – легкий фреймворк для розробки вебзастосунків мовою Python;

GIS – Geographic Information System – геоінформаційна система для роботи з просторовими даними;

HTML – HyperText Markup Language – мова гіпертекстової розмітки для створення вебсторінок;

IT – Information Technology – інформаційні технології;

LLM – Large Language Model – велика мовна модель штучного інтелекту;

ML – Machine Learning – машинне навчання, підмножина штучного інтелекту;

NDVI – Normalized Difference Vegetation Index – нормалізований індекс вегетації, показник стану рослинності;

ORM – Object-Relational Mapping – спосіб взаємодії між об'єктами програми та реляційною базою даних;

UI – User Interface – користувацький інтерфейс;

UX – User Experience – досвід користувача.

ВСТУП

Дивлячись у минуле, коли люди займалися фермерським господарством – це було здебільшого ручною справою. Все спочатку зберігалося у голові, потім на звичайних паперах. З часом з’явилися комп’ютери і інформація почала зберігатися в електронному виді. Але вже на даний момент цих технологій не вистачає, бо живемо у той час коли усе вдосконалюється завдяки новим технологіям.

Беручи ці застаріли технології, фермерські господарства становляться малоефективними та неприбутковими. Простого обліку більше недостатньо, так як росте конкуренція. З’являється потреба швидкого та точного рішення з урахуванням великої кількості факторів одночасно. Тому фермерське господарство треба робити сучасним за допомогою новітніх технологій.

Наприклад, ще років 20 назад фермерське господарство потребувало участі багатьох спеціалістів і тижні ручної праці для ведення усього обліку. Тепер можна все це реалізувати набагато швидше – онлайн та з будь-якої точки світу. Все за допомогою вебсистеми з використанням штучного інтелекту для усього агросектору. Її функціонал залежить не тільки в тому що відбувається цифрова трансформація, тобто не просто перепис даних у електронний вид. А створення такої розумної вебсистеми, яка повинна аналізувати та мати швидкий витяг потрібної інформації у складних ситуаціях, робити прогнозування, надавати рекомендації та допомагати у прийнятті рішення. Увесь перечисллений функціонал дуже актуальний на сьогоднішній день в умовах кліматичних змін, зростання витрат на ресурси та економічної нестабільності.

Раніше для фермерів такі речі, як штучний інтелект, прогнози погоди через API, карти з полями чи автоматичний облік фінансів здавалися чимось далеким, складним. Здавалося, що все це використовують лише великі компанії. Але часи змінилися – сьогодні технології розвиваються настільки

швидко, що навіть невелике фермерське господарство може дозволити собі сучасну систему, яка все це об’єднує й реально допомагає в роботі. Ця вебсистема з даним функціоналом не просто зменшує кількість зайвих дій, а може змінити сам підхід до аграрної діяльності. Так як у майбутньому її можна масштабувати, додавати нові технології, які набагато можуть спростити керування господарством. Але вже коли фермер отримує не просто звіт, а конкретну інформацію з прогнозуванням та порадою на основі оточуючих факторів – це вже новий рівень ефективності у сфері агро-діяльності.

Тому у рамках цієї роботи було поставлено завдання створити саме таку систему. Її суть буде складатися – в інтеграції сучасних інструментів, використання ШІ-компонентів та простому для користування інтерфейсі. Ця класифікаційна робота повинна бути прикладом того, що сучасні технології можуть бути адаптовані та штучний інтелект – це не щось далеке, а реально прикладний інструмент для фермерського господарства.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальні виклики для фермерських господарств

Виклики для фермерського господарства дуже багатосторонні. Як вид економічної діяльності охоплює дуже багато різних процесів: вирощування тварин, догляд за полями та посадженою культурою, облік ресурсів, техніки, працівників, фінансів, логістики, зберігання та збуту. З роками спостерігається, що обсяги аграрного виробництва збільшуються і водночас ускладняється управління господарством. Це велика проблема і тому з’являється потреба в автоматизованих засобах збору, обробки та аналізу даних.

З кожним днем зростає конкуренція, відбуваються кліматичні зміни, завжди є цінова нестабільність. Тому виникає необхідність приймати швидкі та точні рішення, щоб мінімізувати втрати. Однак традиційні методи для обліку та управління такі як, паперові журнали, неструктуровані Excel-таблиці або усні інструкції з цим не допоможуть у сучасному світі.

Виникають труднощі на багатьох підприємствах, включаючи фермерське господарство – це працівники. Багато погано виконують роботу або просто виїжджають за кордон, і фермери залишаються без рук. Через це господарі ризикують втратити не лише гроші, а й розвиток свого господарства. Без сучасної технічної підтримки це може призвести до зниження врожайності, зростання витрат і загалом – до неефективної роботи з ресурсами.

З’являються проблеми з вирощуванням культури на полях. По-перше, з розвитком технологій самі методи обробки, внесення добрив стають складнішими. Наприклад, раніше робилося стандартне внесення добрив, а зараз використовується диференційоване внесення. Тобто для кожної ділянки поля робиться індивідуальне дозування на основі аналізу ґрунту та інших факторів. Тепер щоб мінімізувати втрати культури обробка поля

проводиться з урахуванням силу вітру, температури та погоди.

Також використовують карти NDVI – це стандартизовані показники здорової рослинності. Вони кількісно визначають рослинність, вимірюючи різницю між ближнім інфрачервоним (NIR) і червоним світлом. Здорова рослинність відбиває більше NIR та зеленого світла, ніж інші довжини хвиль, але вона поглинає більше червоного та синього світла. Значення NDVI завжди коливаються від -1 до +1 [1]. Самі дані беруться із спутників або дронів з мультиспектральними камерами. Тобто усі ці методи підлягли модернізації і якщо їх використовувати, то вручну за цим потоком інформації нереально услідкувати.

По-друге, з кожним роком посилюється екологічний контроль та вводяться нові стандарти. Це ставиться до використання пестицидів, добрив, що потенційно шкідливі для ґрунту і підземних вод. Бо в законі України «Про Охорону навколишнього природного середовища» Стаття 35 зазначено, що суб’єкти господарювання повинні не перевищувати гранично допустимі концентрації забруднюючих речовин. Треба дотримуватися норми щодо забруднення нітратами, залишками хімічних засобів та важкими металами [2]. Тому вручну це все дуже складно контролювати і це загрожує порушенням закону.

Крім того, є труднощі до прозорості фінансів. Тобто банки, партнери або інвестори все частіше вимагають чіткої фінансової звітності. Без автоматизації завжди трапляється невідповідність. В результаті штрафи, втрати партнерів і ніякого інвестування. А це великий ризик збанкрутувати та позбутися бізнесу.

Всі ці труднощі створюють головну проблему для фермерів – відсутність якісної аналітики і нормального прогнозування. Саме від цього залежить робота на господарстві: якщо не бачити картину наперед, можна втратити і час, і гроші, і результат. Без прогнозу та аналізу будь-яке рішення фактично дія навмання, а не управління.

1.2 Обмеження традиційних методів

Окрім великих господарств, багато малих та середніх господарств досі продовжують опиратися на паперовий облік або електроні таблиці, які створюються вручну. Саме такі застарілі підходи призводять до невідповідності даних. Найчастіше це втрата даних через пошкодження або помилкове збереження, дублювання записів, неточності у підрахунках. Також користуванням такими способами утворюють складність у відстеженні змін в обліках.

Крім того, проблематика цих методів ще залежить в тому, що інформація майже завжди зберігається локально на комп’ютері або в роздрукованому виді. Це обмежує доступ до неї для інших працівників.

З’являється проблема в відсутності аналітики в реальному часі. Так як в цих традиційних системах закладена обробка даних вручну – все робиться з великою затримкою. Через керівник не бачить поточної ситуації і не може оперативно реагувати на зміни. Наприклад, поломка техніки або великий сплеск витрат.

Також є проблема цих методів – це цілісність даних, так як захищеність низька. Що стосується Excel-файлів – дані можуть легко втратитися через збій комп’ютера. До того ж, можна завантажити файл з вірусом і трапиться витік даних або навіть просто випадково видалити інформацію. З паперовими журналами теж є ризики – їх можна загубити, пошкодити чи навіть випадково викинути разом з непотрібними документами.

Також проблема залежить у відсутності автоматичної обробки даних. Фермер або адміністратор повинні аналізувати великі обсяги інформації. З цього виявляється, що не виходить швидко виявляти проблеми та закономірності на фермі. У випадку збільшення масштабів ферми ці способи ведення обліків стають дуже затратними та практично непридатними.

До того ж, є розуміння – втручання людського фактора. Проблема залежить у тому, що при зміні відповідального працівника велика частина знань або обліку може бути втрачена. А це впливає на стабільність бізнесу, так і на подальше планування господарства.

Ще одна важлива проблема – це висока трудомісткість традиційного обліку. Всі дані потрібно вносити вручну – або у зошити, або в таблиці. На це йде дуже багато часу, особливо коли господарство має кілька десятків полів, техніку, персонал і велику кількість дій щодня. При цьому вся відповідальність за правильність записів лежить на конкретному працівнику.

Коли облік не автоматизовано, фермер або адміністратор змушений постійно заповнювати, переносити або перевіряти інформацію, що не тільки займає час, а й збільшує ризик помилок. Наприклад, при копіюванні цифр можна випадково переплутати дані, не помітити зсув рядка або стерти щось важливе. І таких дрібних помилок за місяць накопичується багато.

Ще одна проблема – це постійне повторення однакових дій. У багатьох випадках потрібно кілька разів вносити ті самі дані в різні таблиці або вручну заповнювати звіти кожного місяця. Це незручно, займає багато часу і просто перевантажує працівників. У результаті це тільки ускладнює управління фермою і забирає сили, які можна було б направити на щось корисніше.

1.3 Можливості сучасних вебсистем

Сучасні вебсистеми дають користувачу багато зручностей у роботі. Власник або працівник може повністю керувати господарством – дивитись дані, вносити зміни і приймати рішення прямо з телефона або комп’ютера, не перебуваючи на місці. Усе це відбувається в реальному часі.

У таких системах забезпечується єдиний доступ до обліків про тварин, поля, техніку, фінанси, працівників. Це дозволяє уникати дублювання та

втрати даних, забезпечує прозору та зрозумілу звітність. Інтерфейс надає можливість фільтрації, пошуку, аналітики та швидкого редагування або видалення записів.

Також системи мають можливість створювати готову звітність на основі збережених даних у базі. Це дозволяє отримувати картину поточної ситуації на фермі. Наприклад, кілька разів натиснув мишкою і вже є інформація про те яка була врожайність поля за минулі роки.

Перевагою вебсистем також вважається їх масштабованість. Нові модулі можуть оновлюватися або додаватися без зупинки роботи системи. Бо це забезпечується завдяки Rest Api [3], які дозволяють інтегрувати зовнішні сервіси. Наприклад, сервіси за метеоданими для прогнозування або модулі штучного інтелекту.

Одна з важливих переваг цієї системи полягає в безпеці даних і правильному розмежуванні доступу для працівників. Кожен користувач бачить тільки ту інформацію, яка стосується його обов’язків. Наприклад, бухгалтер працює лише з фінансовими показниками, а агроном бачить тільки дані по полях. Завдяки цьому вся важлива інформація залишається захищеною.

Також ще однією з переваг вебсистем є можливість легко змінювати або доповнювати структуру даних без необхідності переписувати всю систему. Наприклад, якщо з часом у господарстві з’являється нова культура, техніка або роль працівника – відповідні поля можна додати до бази даних без зупинки всієї роботи.

Також зручність вебсистеми – це користування з будь-якого пристроя. Не обов’язково працювати з комп’ютера, можна за телефона, планшета або ноутбука. Так легше взаємодіяти та керувати процесами з будь-якого місця.

Ще, використоння ORM-бібліотек, допомагають просто працювати з моделями даних. Це особливо зручно, коли потрібно додати нову культуру, техніку чи змінити структуру обліку. Система легко підлаштовується під

нові потреби і при цьому не втрачає стабільності. Можна додавати нові функції без переробки всього проекту.

У реальному часі можна зафіксувати нову агрооперацію, перевірити історію поля, подивитись дані про техніку або передати завдання іншому працівнику. Таким чином, зникає необхідність вести облік у зошиті, а потім переносити його в систему – вся інформація з’являється одразу. Це підвищує оперативність, точність і загальну продуктивність праці.

1.4 Штучний інтелект у фермерському господарстві

Завдяки використанню інтелектуальних технологій у фермерському управлінні з’являється можливість суттєво оптимізувати багато щоденних завдань. Замість того щоб покладатися на застарілі підходи, власник господарства взаємодіє з вебсистемою, який проводить аналіз даних, надає корисні поради та підтримує процес ухвалення рішень у режимі реального часу. Це реально працює не лише в теорії, а й на практиці – як у рослинництві, так і в тваринництві, обліку ресурсів чи фінансовому плануванні [4, с. 71].

Алгоритми машинного навчання можуть аналізувати велику кількість даних – наприклад, погоду за останні роки, типи ґрунтів, які культури де вирощувались, скільки було витрачено коштів. І на основі цього система вже робить прогнози: яка буде врожайність, як виросте тварина, які витрати очікуються або де можуть виникнути ризики.

Також дуже корисне для таких систем використання мовних моделей, або по іншому LLM. Коли такі моделі ШІ вбудовані у вигляді чат-бота, фермер може просто задати питання звичайною мовою – наприклад, «які були витрати цього місяця?» або «що порадите з посівом кукурудзи?». Система одразу відповідає на основі реальних даних господарства. Це дуже зручно, особливо для тих, хто не має глибоких знань у технологіях. Завдяки

цьому навіть малі фермерські господарства можуть користуватись сучасними інструментами штучного інтелекту без складнощів.

Крім того модулі машинного навчання, які використовуються в аграрних системах, можуть з часом вдосконалюватися. Система постійно оновлюється, враховуючи нові дані з господарства: погоду, врожайність, витрати. Навіть якщо умови кожного сезону різні, модель підлаштовується під реальну ситуацію і з часом дає точніші прогнози.

У майбутньому систему можна доповнити новими джерелами інформації, такими як сенсори на полі, дрони, супутникові знімки або результати аналізів ґрунту з лабораторій. Наприклад, сенсор може показувати вологість, дрон виявляє бур’яни, а система одразу пропонує, як краще діяти. З огляду на це, з’являється можливість працювати не лише з минулими даними, а й бачити реальну ситуацію – прямо з поля, у реальному часі.

У результаті ШІ в системі не замінює агроспеціаліста, а працює як цифровий помічник. Він підказує, допомагає швидше ухвалювати рішення та зменшує ризик помилок, які можуть виникнути через людський фактор.

1.5 Інтеграція зовнішніх даних та візуалізація

Система може відображати графіки або звичайну інформацію про погодні умови завдяки сторонніх API. Зміни температури, вологість, кількість опадів та силу вітру можна відображати на діаграмах. З цього з’являється можливість швидко відслідковувати погодну ситуацію та приймати рішення що до перенесення аграрних робіт на сприятливі дня. Тобто користувач буде бачити не просто числа, а й додаткову аналітику.

Крім поточних даних, система дозволяє створювати план дій на майбутнє. На мапі можна вказати, яке поле потребує обробки, хто за це відповідає і що саме має бути зроблено. Завдяки цьому карта стає не просто

засобом перегляду, а справжнім центром керування польовими роботами. Працювати з нею зручно як з комп’ютера, так і з мобільного пристрою.

1.6 Перспективи застосування

Після вивчення предметної галузі можна зробити висновок, що сучасне господарство потребує змін. Старі методи управління вже не працюють так ефективно, як раніше. Натомість з'являються нові цифрові інструменти, які дають змогу організувати роботу набагато простіше і точніше.

Розробка подібної системи відкриває широкі можливості. Вона спрощує щоденні процеси, об'єднуючи все в одному інтерфейсі – облік культур, контроль техніки, працівників, витрат, прогнозування. Усе це доступне з будь-якого пристрою, тому користуватися системою зручно як в офісі, так і на полі.

Однією з ключових переваг такої системи є її модульність. Це означає, що користувачі можуть розпочати з базового функціоналу а згодом – підключити додаткові модулі: аналітику, прогнози, інтерактивну мапу, чат-бота. Таким чином, система адаптується під масштаби та потреби конкретного господарства.

У перспективі така система може стати частиною єдиної інформаційної структури, яка об’єднуватиме фермерів, обласні центри агростатистики, логістичні компанії, кооперативи та інші служби. Всі ці елементи зможуть обмінюватися даними та працювати як єдине середовище.

Тому платформа перестає бути просто місцем для обліку і перетворюється на повноцінне цифрове середовище. Вона підлаштовується під реальні умови господарства, навчається на базі зібраних даних і може масштабуватись при потребі. Це відповідає сучасному підходу до

«розумного фермерства» і допомагає залишатись конкурентоспроможним у швидкозмінному середовищі.

У разі розширення функціоналу можлива також підтримка електронного документообігу: формування рахунків, актів, звітів для контролюючих органів, автоматична генерація звітності відповідно до стандартів МінАПК [5], ДПС [6] або міжнародних сертифікаційних систем.

Наявність таких інструментів підвищує шанси господарства на участь у державних та міжнародних програмах підтримки аграріїв. Багато грантів, програм технічної допомоги або експортних ініціатив вимагають цифрової прозорості управління. Наявність вебсистеми з аналітикою – це конкурентна перевага.

Далі така система цілком може стати частиною великої цифрової мережі, яка об'єднує фермерів, аграрні центри, страхові компанії, логістичні сервіси чи кооперативи. Тобто формується єдина інформаційна структура, де всі елементи пов’язані між собою.

Зрештою, з простого рішення для обліку система може перетворитись на повноцінну цифрову платформу, яка не просто зберігає дані, а сама навчається, масштабується і підлаштовується під умови конкретного господарства. Це відповідає новому підходу до «розумного фермерства» і допомагає залишатися конкурентоспроможним у сучасному агросекторі.

1.7 Постановка задачі

На основі проведеного аналізу предметної галузі було сформульовано задачу розробки сучасної веборієнтованої системи, яка об’єднує засоби аграрного обліку з можливостями аналітики, візуалізації та прогнозування на основі технологій штучного інтелекту. Така система має вирішувати не лише завдання збереження й структурування інформації, але й допомагати користувачу в ухваленні рішень на основі аналізу накопичених даних.

Ключовим функціональним напрямом є реалізація інструментів прогнозування, які дозволяють визначати потенційний приріст ваги тварин та очікувану врожайність сільськогосподарських культур. У першому випадку система працює на основі внутрішніх даних користувача. У другому додатково враховуються зовнішні фактори, передусім метеорологічні умови, які отримуються за допомогою відповідних онлайн-сервісів. Це дозволяє зробити прогноз точнішим і наближеним до реального стану речей на полі.

Окрім прогнозів, важливою особливістю системи є можливість візуалізації географічної інформації. Зокрема, користувач має змогу переглядати місцезнаходження своїх полів на інтерактивній карті, що спрощує навігацію та дозволяє краще оцінити просторову структуру господарства.

Додатково до цього передбачено впровадження інтелектуального віртуального помічника, який допомагає користувачу орієнтуватися в системі, отримувати рекомендації, пояснення до прогнозів і відповіді на запитання у зручній формі. Такий елемент робить взаємодію з платформою більш природною і доступною навіть для користувачів без спеціальної технічної підготовки.

Таким чином, розроблювана система повинна забезпечити не лише автоматизацію облікових процесів, а й надати інструменти підтримки прийняття рішень, що є актуальним завданням для сучасного аграрного сектору.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Теоретичні основи цифрового фермерства

Зараз усе більше фермерів переходять на цифрові інструменти. Це вже не якась новинка, а звичайна частина щоденної роботи. В господарстві з’являються програми для обліку, карти для перегляду полів, підказки про погоду чи витрати. Усе це значно полегшує роботу – більше не потрібно все тримати в голові або записувати в зошит. Вся інформація зберігається в одному місці, і її легко перевірити в будь-який момент. Так економиться час, зменшуються помилки, і врешті-решт – витрати. Коли все під контролем, планувати справи набагато легше. Тому цифрове фермерство – це вже не щось далеке, а цілком реальний спосіб покращити управління в фермерському господарстві.

2.1.1 Поняття розумне фермерство та цифровізація господарства

Саме за раз аграрна галузь здебільшого намагається переходити на цифрові рішення. Бо є розуміння, що користуватися старими традиційними методами вже неефективно, в порівнянні з інструментами, які дозволяють управляти фермерами значно краще без зайвих дій та втрати даних. Тому багато вже років формується таке поняття, як розумне або точне фермерство.

Це схема управління фермерським господарством, яка приймає рішення на основі сирих даних. Таким чином, дані про сільське господарство можуть бути отримані з облікових систем, погодних служб, спеціальних датчиків і супутників. Наприклад, власник знає, що поле засіяно. Але крім цього отримує ще інформацію про стан поля, прогнозування та картину на майбутнє.

Тобто ідея розумного фермерства – це зібрати усю інформацію у купу і дати господарю ферми можливість не тільки просто переглядати дані, а і побачити аналітику, прогноз та поради для покращення процесу на фермі. Стосується в більшість випадках система такими процесами, як вирощування культури та тварин, управління фінансами та введення звітності.

Дійсно, розумне фермерство – це не тільки IT-програмування. А про те, що економить багато робочого часу, мінімізує втрати ресурсів та зменшує людський чинник, через який багато створюються помилок із за навмання прийнятого рішення.

2.1.2 Сучасні тренди в автоматизації господарства

На даний момент в аграрному виробництві з’явилося багато трендів на використання сучасних технологій, які дозволяють економити ресурси та краще планувати роботу.

Перший такий тренд – це використання дронів. В основному використовують для моніторингу полів. Завдяки дронам відстежується стан культури та вноситься точне внесення добрива і пестицидів на поле. Тому власник може переглядати, які зони потребують уваги.

Другий тренд – це використання GPS-навігації. Так як, вже є сучасні трактори, які можуть обробляти поле за заданим маршрутом. Це знижує людський чинник та економить фінанси.

Також є тренд на IoT-пристрої. IoT – це концепція, яка передбачає об’єднання різних фізичних пристроїв через інтернет, дозволяючи їм взаємодіяти та обмінюватися даними. Суть ідеї в тому, щоб зробити об’єкти «розумними» й пов’язаними між собою, створюючи єдиний інтегрований простір для збирання та оброблення інформації. Все це для поліпшення якості життя, оптимізації процесів і підвищення ефективності [7].

У фермерському господарстві ці сенсори використовуються здебільшого для полей та тварин. Наприклад, відстежують вологість ґрунту на полі або стан тварини. Таким образом, господар має поний контроль над цим процесами.

І головний трендом є перехід на вебсистеми. Які збирають інформацію зі всіх нових використаних технологій. Є змога завжди керувати даними у реальному часі та з будь-якої частини світу. А за допомогою штучного інтелекту, вебсистема становиться набагато ефективнішою. Бо тоді з’являється автоматична аналітика та прогнозування на майбутнє. Також швидкий витяг інформації у складній ситуації. Все це дозволяє приймати рішення не наосліп, а на основі реальної аналітики.

Тому судячи за цими трендами, відображається така картина, що аграрна галузь прагне стати розумною та сучасною. І це лише початок, бо господарства тільки набирають оберти.

2.1.3 Вплив IT-рішень на ефективність фермерського господарства

Вибір правильної цифрової системи, дасть змогу повністю змінити спосіб управління господарством. У цьому випадку власник не тільки записує просто дані, але й керує всім процесом і вчасно бере ситуацію під контроль. Тому виконує роботу швидше і зберігає конкурентоспроможність господарства на ринку.

Один із основних плюсів таких рішень – економія. Завдяки системі стає легше керувати витратами на добрива, пальне, обробку полів. Зменшуються зайві витрати, бо вся інформація під рукою, і рішення приймаються точніше.

Ще один важливий момент – це контроль і прозорість. Усі дії працівників фіксуються в системі, процеси відображаються автоматично. Кожен знає, за що відповідає. У разі проблем можна швидко знайти причину і вирішити ситуацію без затримок.

Більше того, ці системи дуже динамічні за своєю природою. Наприклад, якщо змінюється погода або щось відбувається в полі, фермер одразу бачить це в системі і може скоригувати план дій. Це особливо корисно в ту пору року, коли кожен день на полі є цінним.

Також важливо, що інформація в системі не просто зберігається. Вона аналізується. Це дозволяє побачити, що працює краще, де були проблеми, як змінилась ситуація за сезон і що можна покращити в наступному. Такий підхід дає змогу будувати роботу на основі фактів, а не припущень.

2.2 Архітектура вебсистем в агросфері

Зазвичай архітектура вебсистем вистроюється на класичному підході. Це frontend, backend та база даних.

Frontend – це публічна частина web-додатків з якою користувач може взаємодіяти і контактувати напряму. У Frontend входить відображення функціональних завдань призначеного для користувача інтерфейсу, що виконуються на стороні клієнта, а також обробка запитів користувачів. По суті, фронтенд – це все те, що бачить користувач при відкритті сторінки [8]. А backend – створення внутрішнього механізму вебсистем, який обробляє дані, керує базами даних і забезпечує працездатність [9]. База даних – це простір, де зберігаються дані з усієї системи.

2.2.1 Принципи побудови вебсистем

Проектування вебсистеми для ферми є специфічним завданням. Ця система повинна бути зручною, простою та відповідати практиці ведення сільського господарства.

Тому одним з основних принципів є відкритість та підтримка багатьох платформ. Система повинна бути незалежною від пристрою, тобто

користувач повинен мати можливість використовувати систему з будь-якого пристрою.

Саме тому вебтехнології, такі як HTML, CSS та JavaScript, дозволяють створити інтерфейс, який не може бути більш зручним для користувача, без необхідності завантаження стороннього програмного забезпечення.

Так як, господарство містить багато напрямків, то без модульності система буде руйнуватися. Додаток має легко дозволяти розширювати функціонал. Тобто додавання або видалення нових модулів повинно не руйнувати архітектуру.

Крім того, існує цікавий принцип – включення зовнішніх сервісів. Вони також є джерелами додаткової інформації для моніторингу або рефакторингу залишків старих модулів у системі. Усі ці дані потрібно враховувати, щоб побачити поточний стан на фермі.

Що до бази даних, вони повинні бути добре структуровані, бо господарство завжди збирає багато інформації. Для цього в багатьох випадках використовуються ORM-системи, наприклад SQLAlchemy. Також треба враховувати безпеку бази даних. Так як, у системі завжди зберігається кондиційна інформація. Тому реалізуються права доступу для різних працівників, додаються захисти від несанкціонованого доступу та резервне копіювання у хмарне середовище.

І останній важливий принцип – це швидкість та стабільність додатку. Система повинна працювати плавно, без багів та затримок. Навіть при великої кількості записів. Бо завжди є проблеми з мобільним інтернетом та сезонні навантаження.

2.2.2 Робота компонентів у вебсистемі

Із компонентів є вебклієнт, сервер, база даних, API. Все працює разом, як єдиний механізм. Вебклієнт – це те, що бачить користувач у браузері. Через нього він взаємодіє з інтерфейсом і відправляються запити на сервер.

А сервер – це мозок вебсистеми. Саме він обробляє ці запити, шукає потрібні дані в базі даних, в якій зберігаються всі дані. Тобто, база даних, як електронний архів усієї системи. Потім сервер знаходить потрібні дані, формує відповідь і повертає її назад користувачу.

На прикладі, вебсистеми для фермерського господарства. Власник вибирає пункт «показати витрати за сезон». Цей запит відправляється на сервер. Сервер знаходить потрібну інформацію в базі даних. Обробляє, формує відповіді і назад відправляє на сайт. Потім власник бачить відповідну інформацію за пункту.

Також вебсайти можуть бути зв’язані с іншими онлайн-сервісами. Їх використовують для того, щоб додавати новий функціонал без зміни всієї системи. Це дуже гнучкий та зручний спосіб.

2.3 Застосування штучного інтелекту

Так як, штучний інтелект розвивається і вже в багатьох галузях використовується – в фермерському господарстві теж повинен ефективно реалізовуватися. Завдяки технологіям аналізу даних та автоматизованого прийняття рішень в аграрній галузі зменшується об’єм ручної праці. Також, ШІ може підказувати, як краще поступити в ситуації, щоб зменшити кількість дій та фізичні сили.

Штучний інтелект поєднує такі ключові технології: машинне навчання, мовні моделі, нейронні мережі та прогностичні алгоритми. Ці компоненти можуть навчатись на реальних даних та з кожним циклом давати все точніше і точніше результати. Тому вебсистема для фермерського господарства буде становитися з часом більш розумною і практичною.

2.3.1 Основи машинного навчання, нейронних мереж та LLM

У фермерстві з кожним роком накопичується все більше інформації: про врожаї, погоду, ґрунти, техніку. Щоб краще з нею працювати, використовують машинне навчання – це коли система самостійно вчиться на прикладах і виявляє зв’язки в даних, без чітко заданих інструкцій. Це зручно, бо всі фактори постійно змінюються, і складно передбачити їх наперед.

Для складніших завдань застосовують нейронні мережі – це спеціальні математичні моделі, що «наслідують» логіку роботи людського мозку. Вони корисні там, де потрібно розпізнавати візуальні об’єкти, аналізувати фото з дронів або дані про здоров’я тварин.

Також з’явились так звані великі мовні моделі (LLM), наприклад Claude чи GPT. Вони здатні працювати з текстами: відповідати на запитання, пояснювати дані, надавати поради.

2.3.2 Приклади прогнозування які можна реалізувати в вебсистемі

Штучний інтелект відкриває багато варіантів для практичного застосування в фермерському господарстві. Один із прикладів – прогноз урожайності. Якщо система отримає дані про тип ґрунту, площу поля, історію врожаїв і погоду, вона може передбачити, який буде результат цього сезону. Це допоможе краще планувати витрати, збут і навіть залучення техніки.

Інший напрям – контроль за ростом тварин. Якщо регулярно фіксувати дані про вагу, вік, раціон і стан здоров’я, модель зможе передбачити, скільки ще часу потрібно до досягнення цільової ваги. Так фермер бачить, чи йде все за планом, і вчасно виявляє відхилення.

Також можна реалізувати аналіз ризиків. Наприклад, якщо система бачить, що на певному полі вже кілька років поспіль виникають проблеми з

вологістю або хворобами культур – вона попередить про ймовірність повторення ситуації. Це дає змогу заздалегідь змінити план: підготувати інше насіння, скоригувати обробку ґрунту або перенести терміни посіву.

2.3.3 Інтелектуальні рекомендіції та роль діалогових моделей у підтримці рішень

Один із найбільш корисних напрямів у застосуванні штучного інтелекту – це можливість отримувати рекомендації у зручній і зрозумілій формі. Сучасні діалогові моделі, такі як Claude або GPT, здатні працювати не просто як чат-бот, а як повноцінний цифровий помічник, що знає всі дані господарства і може дати поради, які справді мають сенс.

Такий підхід особливо зручний для користувача, бо не потрібно розбиратись у складних таблицях чи шукати потрібну інформацію по різних модулях. Можна просто написати в чаті запит типу: «Покажи витрати на пальне за останній місяць» або «Коли краще сіяти кукурудзу на полі №4 з урахуванням погоди?» – і система одразу видає відповідь, враховуючи всю актуальну інформацію.

Це працює завдяки тому, що діалогова модель під’єднується до бази даних і використовує її як основу для відповідей. Фактично, вона вміє не тільки говорити, а й думати на основі того, що вже відомо про господарство. Якщо фермер хоче оцінити ризики чи отримати прогноз – достатньо сформулювати запит своїми словами, і система надасть відповідь без зайвої тяганини.

Також такі моделі можуть підказати наступні кроки: наприклад, після аналізу даних система може написати, що *«*Витрати на обприскування зросли на 30% – рекомендовано перевірити технічний стан обприскувача або змінити норму внесення*»*. Це дозволяє реагувати швидко й точково, без потреби залучати аналітика.

Інтелектуальна підтримка у вигляді діалогу є простим і ефективним спосібом взаємодії з цифровою системою. Її може освоїти навіть той, хто не має досвіду роботи з комп’ютерними програмами, бо все відбувається у звичному форматі питання-відповідь. Це значно знижує бар’єр входу та робить систему корисною для малих і середніх господарств.

2.4 Геоінформаційні системи (GIS) та картографія

Сучасне фермерство активно використовує просторові дані для планування і контролю. Карти допомагають не лише бачити територію, а й ефективно керувати ресурсами, планувати роботи і проводити аналіз.

2.4.1 Основи використання Leaflet.js, OpenStreetMap

Коли потрібно показати карту прямо в браузері, зручно використовувати бібліотеку Leaflet.js [10]. Вона проста в налаштуванні і не потребує складних знань. Завдяки їй можна швидко додати на карту поля, об’єкти чи позначки, які потрібні для роботи. Як основу для самої карти часто беруть OpenStreetMap – це безкоштовна відкрита мапа, яку можна використовувати без ліцензії [11].

Разом ці інструменти дозволяють зробити зручний візуальний інтерфейс, де все видно в реальному масштабі: поле, техніка або завдання. Це важливо в аграрній сфері, бо надає можливість бачити ситуацію прямо на місці – не просто читати дані, а реально орієнтуватися по карті.

2.4.2 Інтеграція геоданих

Інтеграція геоданих є важливим кроком у розвитку цифрових інструментів для агросфери. Один із найкорисніших інструментів – це підключення кадастрових меж, які дозволяють чітко розмежувати території,

визначити, де закінчується власна ділянка, і навіть перевірити статус сусідніх полів.

Окрім кадастру, у сучасних платформах також можна реалізувати відображення меж полів, які можуть не збігатися з офіційними кадастровими даними. У таких межах часто зберігається додаткова інформація: назва культури, площа, історія обробки. Вся ця інформація легко прив’язується до карти, і користувач бачить не просто полігони, а «живу» базу даних на карті.

Інтеграція NDVI у вебсистему дає змогу зменшити кількість виїздів у поле, точніше планувати внесення добрив, а також швидко орієнтуватись у великому господарстві. Всі дані відображаються у зрозумілому графічному вигляді, що значно полегшує аналіз і прийняття рішень.

Таким чином, використання кадастрових шарів, меж полів і індексу NDVI дозволяє перетворити вебсистему на справжній геоінформаційний центр управління фермою. Це не просто мапа, а повноцінний інструмент моніторингу та прийняття рішень на основі просторових даних.

2.4.3 Переваги візуалізації компонентів

Візуалізація об'єктів на фермі додає цінності управлінню фермою. Саме це полегшує не лише зберігання даних у таблицях, але й їх візуалізацію на реальній карті з географічною прив'язкою. З одного погляду користувачі можуть побачити, що відбувається на полі, де саме в полі, де знаходиться обладнання або хто з персоналу чи працівників може виконувати яку роботу.

Зручність і простота цього підходу вважаються його сильною стороною. Наприклад, адміністратор може за лічені секунди побачити, що завершено, що залишилося і які ресурси зайняті. Це також має особливе значення при високих навантаженнях, коли багато процесів потрібно контролювати одночасно.

Крім того, дії працівників і обладнання можуть бути легко синхронізовані за допомогою візуалізації. Якщо система використовує GPS, можна фактично бачити рух об'єктів у реальному часі. У разі несправності або затримок менеджер може одразу дізнатися місцезнаходження інструментів і вжити швидких заходів.

Планування завдань також менш ускладнене. Користувачам потрібно лише натиснути на поле на карті, описати роботу, яку потрібно виконати, вказати виконавця та термін. Все буде чітко видно і легко читатися, мінімізуючи помилки та залишаючись організованими.

Система також може зберігати історичні дані, щоб користувачі могли бачити, що відбувалося на полях і на фермі в минулому. Це корисно для розслідування врожайності, повторення проблем або змін в управлінні землею. Все це сприяє більшій прозорості та кращому прийняттю рішень.

Тому можливість карти в управлінні фермою не лише для перегляду інформації, але й для взаємодії з нею. Це формує повноциклове рішення для управління, організації та аналізу всіх процесів на фермі.

2.5 Метеоаналітика та погодні API в агросистемах

В аграрному виробництві погодні умови мають вирішальне значення. Для того щоб краще реагувати на зміни клімату, сучасні фермерські системи починають використовувати автоматичне підключення до погодних сервісів. Завдяки цьому з’являється можливість бачити не тільки поточну ситуацію, а й прогноз на кілька днів наперед. Це особливо корисно під час планування посівів, зрошення чи збирання врожаю. Вебсистеми можуть самостійно отримувати погодні дані, оновлювати їх у реальному часі та враховувати при формуванні підказок для користувача. Такий підхід допомагає приймати рішення з урахуванням реальної обстановки на полі.

2.5.1 Принципи роботи OpenWeatherMap API

Принцип роботи OpenWeatherMap API полягає в тому, що вебсистема надсилає запит на сервер OpenWeatherMap із вказаними параметрами, наприклад географічними координатами або назвою населеного пункту. У відповідь сервіс надає структуровані погодні дані у форматі JSON. Ці дані можуть містити поточну температуру повітря, вологість, атмосферний тиск, силу і напрям вітру, рівень опадів, хмарність, а також детальний погодний прогноз на кілька днів наперед.

API підтримує багато типів запитів. Наприклад, можна отримати дані лише про поточну погоду, або окремо прогноз на 24 години чи на 5 днів. Окрім базових показників, можна підключити додаткові параметри, такі як рівень ультрафіолетового випромінювання або ймовірність опадів у певні години.

OpenWeatherMap постійно оновлює свої дані, що дозволяє системі працювати із максимально свіжою погодною інформацією. Завдяки автоматичним оновленням фермер не потребує вручну вводити дані – система сама підтягує і обробляє погодну ситуацію.

Отримані дані можна зберігати у базі для подальшого аналізу, використовувати для побудови графіків, візуалізацій або виводити безпосередньо у вебінтерфейс. Це відкриває широкі можливості для автоматизації процесів: наприклад, агросистема може сповіщати про ризик посухи або рекомендувати найкращий час для посіву певної культури, зважаючи на прогноз температури і опадів.

2.5.2 Значення погодного фактору в агропланування

Погода є важливим фактором, який потрібно враховувати при прийнятті рішень щодо планування на фермі, оскільки вона впливає не лише на врожайність, але й на безпеку, своєчасність та економічну життєздатність

багатьох видів діяльності на фермі. Температура повітря, кількість опадів, вологість, сила і напрямок вітру, інтенсивність сонячних променів безпосередньо впливають на стадії розвитку рослин, час сівби, обробку ґрунту або захист врожаю та збирання, а також транспортування.

Із за вітряних умов захисні засоби можуть бути розподілені неправильно або ефективність продукту може зменшитися під час обприскування. Посів у занадто вологий ґрунт призведе до гниття насіння, а збирання врожаю під дощем може спричинити втрати та псування врожаю. Уважно стежачи за прогнозом погоди, багато з цих ризиків можна зменшити. Використовуючи погодні дані, фермер може збільшити кількість днів, які можна використовувати для конкретної операції, та мінімізувати витрати на переробку або втрату матеріалу.

Крім того, регулярне спостереження за погодними умовами є хорошим способом оцінити, як змінюється сезонність, і скоригувати стратегію ферми, щоб відобразити «нову норму» клімату.

Отже, погодний параметр не є просто вхідними даними для отримання або очікуваною інформацією, а стратегічною змінною, яка значно впливає на прийняття рішень, розподіл ресурсів та фінансовий результат сільськогосподарського бізнесу.

2.5.3 Приклади автоматичних рішень на основі погодних даних

Автоматизовані дії, які ґрунтуються на метеоданих, суттєво покращують процес прийняття рішень у фермерському управлінні. Система може самостійно аналізувати прогноз погоди та попередні кліматичні умови і пропонувати оптимальні дії без потреби в постійному контролі з боку людини.

Якщо прогноз показує дощ, програма може попередити, що поливати поля зараз недоцільно. При різкому зниженні температури фермер отримає повідомлення з порадою про укриття культур, які можуть постраждати. А у

випадку спеки система підкаже перенести заплановані роботи на інший день.

Також може бути реалізоване автоматичне оновлення графіка польових завдань. Наприклад, календар буде коригуватись відповідно до погодних умов, а відповідальні працівники отримають повідомлення про зміни. Це знижує ризики простоїв і втрат через несприятливу погоду.

Крім поточної ситуації, система може аналізувати, як погодні фактори впливали на врожай у минулих роках, і враховувати це під час формування рекомендацій. У результаті фермер не тільки бачить прогноз, а й отримує підказку, що саме варто зробити з огляду на власну практику.

Такий підхід допомагає швидко реагувати на зовнішні зміни і планувати дії, спираючись не на інтуїцію, а на факти з попереднього досвіду та актуальної інформації. Це особливо корисно у пікові періоди сезону, коли помилки коштують дорого.

2.6 Безпека та права доступу в аграрних IT-системах

У феремерській IT-системи, зазвичай зберігається значна кількість важливої інформації, такої як дані про вирощування, витрати або осіб. Тому важливо не лише знайти зручний спосіб роботи з цією інформацією, але й забезпечити її безпеку. Для цього система повинна мати обмеження прав доступу для всіх користувачів до певних частин, і дані навіть у разі збою не повинні бути втрачені. Добре визначений контроль доступу та якісні резервні копії можуть підтримувати порядок і запобігати втраті інформації.

2.6.1 Моделі ролей користувачів

У фермерському господарстві працюють різні люди з різними обов’язками. Один керує технікою, інший займається полями, ще один відповідає за фінанси. Щоб у системі не було плутанини і щоб ніхто

випадково не видалив або не змінив щось важливе, впроваджуються ролі користувачів. Це означає, що кожен бачить тільки ту частину інформації, яка йому дійсно потрібна. Наприклад, адміністратор має повний доступ і може змінювати налаштування або створювати нових користувачів. Агроном працює лише з мапою, посівами, даними про поля. Бухгалтер бачить тільки фінансову частину – витрати, доходи, зарплати.

Розподіл доступів не тільки підвищує безпеку системи, але й робить її більш зручною. Працівник не витрачає час на те, що йому не потрібно, а також не має змоги випадково змінити важливі дані. Це особливо важливо в умовах, коли працівники мають різний рівень досвіду роботи з цифровими системами. Крім того, розмежування прав дає змогу краще контролювати діяльність – завжди можна побачити, хто і коли вніс якусь зміну.

2.6.2 Захист даних, хмарне зберігання, резервне копіювання

Управління інформацією в аграрних системах є важливим. Будь-яка інформація, що знаходиться в такій системі, може бути критичною для підприємства. Втрата запису врожаю, фінансової документації або записів про роботу співробітників може бути як дорогою, так і шкідливою для організаційної інфраструктури.

Потрібна хороша система хмарного зберігання, бажано надійна понад усе. Дані зберігаються не лише на одному пристрої, але й передаються на віддалені сервери, розташовані в безпечних дата-центрах. Це добре для спокою в разі виходу з ладу обладнання або втрати локального доступу.

Також система має надзвичайно важливу функцію резервного копіювання. Основна інформація автоматично зберігається як дублікати в регулярних місцях. У разі випадкового видалення або зміни є можливість повернутися до старої версії.

Передавання даних між користувачем і сервером відбувається по зашифрованих каналах зв’язку. Це означає, що сторонні особи не можуть

перехопити інформацію під час її передачі. Додатково реалізується розмежування прав доступу. Наприклад, працівник бачить лише ту частину системи, яка йому потрібна, і не може змінювати важливі параметри, якщо не має відповідного рівня прав.

Усе це разом забезпечує не просто збереження даних, а й довіру користувача до системи, що важливо для стабільної роботи будь-якого фермерського підприємства.

2.7 Інтеграція зовнішніх сервісів у вебсистему

Підключення до зовнішніх сервісів робить вебсистему більш гнучкою та здатною підлаштовуватись під потреби користувача. Замість того щоб зберігати всі дані всередині, система може взаємодіяти з іншими джерелами. Наприклад, отримувати прогнози погоди, фінансову інформацію або аналітичні відповіді від мовних моделей. Це дає змогу не тільки оновлювати інформацію автоматично, а й розширювати функціонал без кардинальних змін у структурі самої програми.

2.7.1 API як інструмент підключення сторонніх даних

Коли йдеться про сучасне фермерське господарство, вже не достатньо мати тільки власну базу даних. Система має взаємодіяти з іншими цифровими сервісами. Це дозволяє працювати швидше, точніше й з меншими зусиллями. Саме для цього і потрібні API, які можна уявити як міст між вебдодатком і зовнішніми джерелами інформації.

Наприклад, якщо фермер хоче дізнатись точний прогноз погоди або отримати пораду від штучного інтелекту, не потрібно створювати окрему програму – достатньо звернутись до зовнішнього сервісу через API. Так система отримує потрібні дані і миттєво показує результат у зручному вигляді. Це можуть бути не лише погодні сервіси, а й інтеграції з мовними

моделями. Тобто фермер може поставити запитання в системі, і вона сформулює відповідь не гірше за консультанта.

Інша ситуація – коли потрібна інформація про витрати, фінансові звіти чи інші економічні показники. Завдяки підключенню до банківських чи бухгалтерських платформ, дані можуть оновлюватись автоматично. Це допомагає уникнути помилок, пов’язаних із ручним введенням, і зменшує ризик фінансових прорахунків.

Ще один важливий момент – робота з картами. Наприклад, на базі відкритої карти місцевості можна будувати власну мапу полів, техніки чи завдань. За допомогою бібліотек з картами система може показувати реальне положення об’єктів, дозволяючи користувачу бачити всю ситуацію на екрані: що, де і коли відбувається. Це значно зручніше, ніж працювати з таблицями або паперовими нотатками.

Підключення до зовнішніх сервісів дає системі нове життя. Вона вже не виглядає як закритий інструмент для обліку, а перетворюється на гнучку платформу, яка може змінюватися, доповнюватися й пристосовуватись до конкретних потреб. І головне усе це не потребує глобального переписування коду або складних оновлень. Якщо потрібна нова функція – додається ще одне підключення і система вже вміє більше, ніж учора.

2.7.2 Переваги модульної архітектури

Коли мова йде про створення складної цифрової системи для фермерського господарства, дуже важливо зробити її такою, щоб зміни та розширення не спричиняли труднощів у роботі. Для цього найкраще підходить модульний підхід. У такій архітектурі вся система поділяється на незалежні частини. Кожна з них відповідає за окрему функцію. Наприклад, одна частина може зберігати інформацію про поля, інша – працювати з фінансами, ще інша – надавати погодні прогнози.

Це дає можливість зручно змінювати лише ту частину, яка потребує вдосконалення. Наприклад, якщо виникла потреба додати облік витрат на паливо, це можна зробити без втручання в інші блоки. У результаті система оновлюється швидко і без ризику для вже наявних даних.

Такий підхід також дозволяє розробляти різні частини системи паралельно. Коли одна команда працює над обліком працівників, інша може в цей самий час займатись інтерактивною картою. Це значно економить час, особливо якщо дедлайни жорсткі.

Модульна структура також зручна для самого користувача. Якщо господарству не потрібен розділ тваринництва, його просто не активують. Якщо з часом з’являється новий напрямок діяльності, його можна додати без переробки всієї платформи.

У підсумку можна сказати, що такий підхід дозволяє створити гнучку і надійну систему. Вона легко розвивається, швидко адаптується під нові умови і не вимагає складного технічного обслуговування. Це робить її справді ефективним інструментом для сучасного фермера.

2.8 Проблеми впровадження ІТ-рішень в невеликих господарствах

Хоча цифрові технології відкривають великі можливості для аграрного бізнесу, саме малі та середні господарства найчастіше зіштовхуються з труднощами під час їхнього впровадження. Обмежений доступ до ресурсів, нестача ІТ-компетенцій та застаріла матеріальна база ускладнюють перехід до сучасних систем управління. Саме тому важливо враховувати специфіку таких підприємств при розробці цифрових рішень. Системи мають бути не тільки функціональними, а й простими у використанні, доступними за вартістю і здатними працювати навіть у мінімальних технічних умовах.

2.8.1 Бар’єри: фінансові, освітні, технічні

Коли малі та середні ферми намагаються впровадити цифрові технології, часто стикаються з різними труднощами. По-перше, це фінанси. Не кожен фермер може дозволити собі сучасне обладнання чи платні сервіси. Тому доводиться шукати безкоштовні рішення, які часто мають обмежений функціонал.

Друга проблема – це нестача знань. Багато людей просто не мають достатнього досвіду в роботі з комп’ютером чи інтернетом. І навіть якщо система хороша, не всі вміють нею користуватися. Це уповільнює процес цифровізації, бо користувачі бояться щось зламати або не розуміють, з чого почати.

Також є технічні обмеження. У сільських районах часто слабкий інтернет або застарілі комп’ютери. Іноді немає доступу до смартфонів чи планшетів. Це ускладнює запуск навіть найпростіших онлайн-сервісів.

Щоб цифрові рішення дійсно працювали, потрібно враховувати ці проблеми ще на етапі розробки. Інакше система залишиться тільки на папері, а фермери й далі користуватимуться блокнотом і ручкою.

2.8.2 Роль простих інтерфейсів і адаптивних рішень

Для малих і середніх фермерських господарств простота цифрових рішень має особливо велике значення. У таких підприємствах часто немає окремого ІТ-відділу або навіть спеціаліста, який би міг займатись налаштуванням складної системи. Більшість процесів виконує сам власник або кілька найманих працівників, які здебільшого не мають технічної освіти. Тому інтерфейс повинен бути інтуїтивно зрозумілим, без складних навігаційних меню чи зайвих елементів. Людина має одразу бачити, куди натискати, як додати новий запис, знайти потрібну інформацію або сформувати звіт.

Наявність адаптивного дизайну, що дозволяє користуватись системою як з комп’ютера, так і з мобільного телефону або планшета, значно полегшує доступ до даних без прив’язки до офісу. Це особливо важливо під час роботи в полі або при виїздах на віддалені ділянки. Якщо додаток працює однаково добре на різних пристроях, це забезпечує зручність для користувача в будь-яких умовах.

Окрім зручного доступу, адаптивність означає і здатність системи змінюватися відповідно до потреб господарства. Наприклад, якщо в господарстві з’являється новий вид культур, розширюється автопарк або змінюється структура персоналу, користувач повинен мати змогу легко оновити систему без допомоги програміста. Це економить час і ресурси, а також дозволяє уникнути затримок у роботі.

Загалом простота в інтерфейсі, разом із можливістю адаптувати систему до змін, відіграє ключову роль у впровадженні цифрових інструментів на малих фермах. Саме завдяки таким рішенням фермери можуть без зайвих витрат і складнощів користуватись сучасними технологіями, що підвищують ефективність і конкурентоспроможність їх господарства.

Крім того, важливо враховувати мовну та культурну адаптацію інтерфейсу, оскільки багато фермерів віддають перевагу роботі українською мовою та звикли до певної термінології. Якщо система відображає знайомі слова й процеси, це значно знижує бар’єр входу. Таке локалізоване рішення не лише полегшує навчання користувача, а й сприяє довірі до продукту, що особливо важливо в аграрній сфері, де рішення часто приймаються на інтуїтивному рівні.

3 ПРОЕКТУВАННЯ ТА РОЗРОБКА ВЕБСИСТЕМИ

3.1 Архітектура проекту

У процесі було вирішено створити проєкт у формі вебсистеми. Вебсистема була розгорнута на основі багатошарової архітектури, що зменшує зв'язки між логічними компонентами системи та може полегшити обслуговування, масштабованість і подальший розвиток. Архітектурна модель базується на моделі «модель–вид–контролер» (MVC), яка прийнята більшістю інженерів-програмістів у сучасній веброзробці. Цей підхід передбачає поділ програмної системи на три взаємопов’язані, але чітко ізольовані компоненти:

– model, що відповідає за структуру та зберігання даних. У даному вебдодатку модель реалізує об’єктно-реляційне відображення (ORM), що дозволяє зручно взаємодіяти з базою даних через Python-класи. Саме в моделі описується, які об’єкти існують у системі (наприклад, тварини, прогнози росту), які атрибути вони мають, як зберігаються, пов’язуються та оновлюються. Крім того, модель включає логіку взаємодії з базою даних, зокрема запити на створення, читання, оновлення та видалення (CRUD-операції);

– view, що відповідає за візуальне представлення даних, отриманих від моделі. У проєкті роль view виконують HTML-шаблони з використанням шаблонізатора, який дозволяє виводити динамічну інформацію, включати умови, цикли та інші логічні конструкції прямо в HTML. Таким чином, інтерфейс користувача залишається гнучким, адаптивним і чітко відокремленим від логіки програми;

– controller, що забезпечує взаємозв’язок між моделлю та уявленням. Саме контролер приймає вхідні запити користувача (HTTP-запити), виконує необхідні обчислення або обробку (наприклад, передає дані до моделі або викликає прогноз моделі ШІ), та повертає відповідний шаблон з

результатами. У цьому компоненті також реалізується маршрутизація запитів, перевірка введених даних, обробка форм і інтеграція з модулями машинного навчання.

Клієнтська частина системи реалізована за класичною схемою серверної генерації сторінок, у якій інтерфейс формується на сервері та передається користувачу у вигляді HTML. Реалізацію коду для клієнтської частини можна побачити у лістингу Б.1 додатку Б.

Основу клієнтської частини становлять:

– HTML-шаблони, що зберігаються в директорії templates/ та відповідають за структуру і вміст вебсторінок. Вони генеруються динамічно за допомогою шаблонізатора, що дозволяє виводити дані з бази, створювати форми, списки та результати обчислень;

– файли стилів (CSS), що розміщені у static/css/ і відповідають за зовнішній вигляд сторінок: кольори, розмітку, шрифти, а також анімації;

– скрипти JavaScript, що у static/js/ реалізують базову динаміку клієнтської частини: обробку фонових ефектів, інтерактивні елементи, повідомлення та взаємодію з картою або іншими інтерактивними блоками;

– мультимедійні ресурси, що зображення та анімації, які використовуються для візуального оформлення інтерфейсу, зберігаються у static/images/.

Загалом клієнтська частина забезпечує зручну та інтуїтивну взаємодію користувача із системою.

Серверна частина системи розроблена мовою Python із використанням фреймворку Flask. Вона відповідає за обробку запитів користувача, взаємодію з базою даних, запуск моделей машинного навчання, формування відповідей і підтримку логіки системи. Архітектура дотримується принципів модульності та розділення відповідальностей, що забезпечує легкість у підтримці та масштабуванні. Реалізацію сереверної частини можна побачити у лістингу А.1 додатку А.

Основні файли серверної частини:

– app.py, що є головним файлом запуску застосунку. Відповідає за ініціалізацію Flask-додатку, імпорт конфігурацій, підключення розширень, реєстрацію маршрутів та запуск локального сервера;

– config.py, що є централізованим місцем зберігання налаштувань системи: URI бази даних, режим виконання (Debug/Production), секретні ключі, шляхи до моделей тощо;

– extensions.py, що є підключенням та ініціалізацією сторонніх бібліотек, включаючи SQLAlchemy (ORM), WTForms (обробка форм), та інші Flask-компоненти;

– models.py є файлом визначення моделей бази даних за допомогою SQLAlchemy. Тут описано такі сутності як Animal, GrowthPrediction, Field тощо, з атрибутами, зв’язками та методами взаємодії з БД;

– forms.py є файлом оголошення форм, що використовуються у HTML-шаблонах. Забезпечує валідацію введених користувачем даних на серверному рівні.

Файли routes.py – це модулі, що реалізують маршрутизацію HTTP-запитів. Вони містять логіку додавання, редагування, перегляду тварин, а також викликів прогнозних моделей.

Файл init_db.py створює базу даних та всі необхідні таблиці відповідно до моделей.

Файл .env – простір середовищних змінних, який зазвичай містить конфіденційні налаштування (ключі, URI бази даних, шляхи до моделей тощо).

Файл requirements.txt – список залежностей проєкту, необхідних для встановлення середовища розробки. Він дозволяє швидко розгорнути систему на будь-якому пристрої, встановивши всі потрібні бібліотеки однією командою.

3.2 Використанні технології

3.2.1 Python

Для реалізації серверної частини була обрана мова програмування – Python. Python є однією за найпопулярнішою та універсальною мовою у світі, бо поєднала в собі простоту, зрозумілий синтаксис та зручність. Має велику кількість бібліотек для вирішення багатьох задач. Основні переваги Python, можна побачити в таблиці 3.1.

Таблиця 3.1 – Переваги Python

Переваги	Пояснення
Швидкість розробки	Завдяки високому рівню абстракції та численним готовим бібліотекам
Гнучкість	Python підтримує процедурне, об’єктно-орієнтоване та функціональне програмування
Багата екосистема	Наявність фреймворків Flask, Django для веброзробки, бібліотек scikit-learn, pandas для аналізу даних та машинного навчання
Зрозуміла інтеграція	Легка взаємодія з базами даних, API, зовнішніми службами
Широке ком’юніті	Забезпечує активну підтримку, оновлення документації, та велику кількість прикладів коду

У даному проекті Python використовувався для реалізації створення:

– реалізації бізнес-логіки вебдодатку;
– створення маршрутизації HTTP-запитів;
– обробки форм і валідації введених даних;
– взаємодії з базою даних через ORM;

– підключення та запуск моделей машинного навчання;
– побудови інтерфейсів взаємодії з чат-ботами.

3.2.2 HTML

Для створення клієнтської частини було використано HTML. HTML є стандартним засобом для подання інформації так, щоб її можна було побачити незалежно від типу використовуваного комп'ютера та операційної системи. Дві найважливіших властивості HTML− простота і платформна незалежність [12, с.5].

За допомогою HTML позначають текст, вказуючи своєму вебпереглядачу, як він має розуміти позначений текст, так само як і на жорсткому диску інформація зберігається в блоках, кластерах, секторах, доріжках і тільки за допомогою, такої, визначеної структури твій комп'ютер розуміє, що треба, а що не треба зчитувати [13].

Також HTML має свої переваги та недоліки. Це можна побачити в порінянні з іншими мовами та технологями в таблиці 3.2.

Таблиця 3.2 – Порівняння HTML з іншими мовами та технологіями

Мова	Призначення	Основні переваги	Недоліки
HTML	Створення структури вебсторінок	Стандарт вебу, підтримка всіма браузерами	Не підтримує складну логіку сам по собі
Markdown	Форматування простого тексту	Простота, швидкість написання	Обмежені можливості в дизайні та верстці

Продовження таблиці 3.2

Мова	Призначення	Основні переваги	Недоліки
XML	Обмін та зберігання структурованих даних	Гнучка структура, валідність	Не призначений для відображення в браузері
React/Vue	Створення SPA (односторінкових застосунків)	Динамічні інтерфейси, компонентна архітектура	Вища складність, потрібна збірка, більше ресурсів
Tkinter/Qt	Десктопні графічні інтерфейси	Не залежить від браузера, багата графіка	Не підходить для вебдоступу

У рамках даного проекту HTML використовувався для:

– формування шаблонів вебсторінок;
– побудови форм введення даних;
– створення таблиць, списків, кнопок та інших елементів UI;
– інтеграції з CSS та JavaScript для покращення вигляду та взаємодії.

3.2.3 CSS та JavaScript

Для стилізації клієнтської частини було використано CSS (абревіатура від Cascading Style Sheets, що в перекладі означає каскадні таблиці стилів). CSS – це спеціальна мова (мова стилів), за допомогою якої описують вигляду документів (як і де відображати елементи вебсторінки), написаних мовами розмітки даних. Найчастіше CSS використовується для документів, котрі розмічені мовою HTML, XHTML та XML [14].

За допомогою CSS була відокремлена візуальна складова інтерфейса від його структури, що сприяло кращій підтримці коду та повторному використанню стилів. На рисунку 3.1 можна побачити файли з використанням CSS для конкретних модулів проекту.

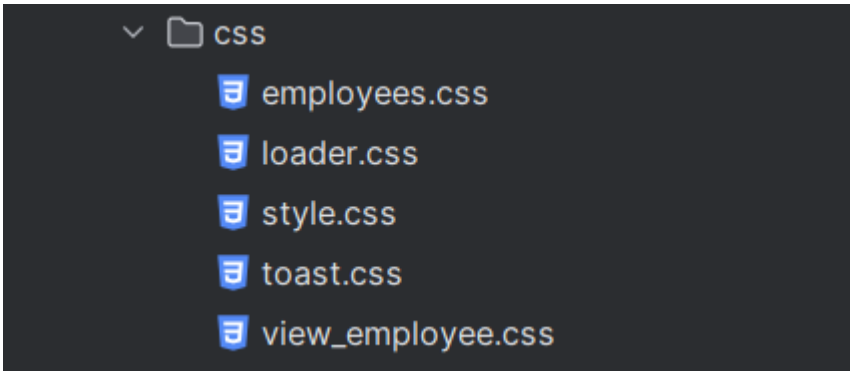

Рисунок 3.1 – Файли з використанням CSS для конкретних модулів проекту

У межах даного проекту CSS використовувався для таких задач:

– визначення кольорової гами інтерфейсу;
– налаштування шрифтів, розмірів тексту, відступів і розмітки;
– оформлення кнопок, форм, таблиць та повідомлень;
– реалізації анімацій;
– адаптивного відображення контенту на різних пристроях.

Щоб досягнути зробити інтерфейс інтерактивним та приємнішим для користувача було використано JavaScript. Коли вебсторінка завантажується, вбудований інтерпретатор JavaScript отримує та виконує JavaScript-інструкції, записані на сторінці. Інтерпретатор читає та виконує JavaScript-інструкції по черзі, перетворюючи їх на машинний код. Це дозволяє JavaScript взаємодіяти зі структурою та вмістом вебсторінки, змінювати HTML та CSS, обробляти події, виконувати асинхронні запити на сервер та оновлювати сторінку без перезавантаження [15]. На рисунку 3.2 можна побачити файли з використанням JavaScript.

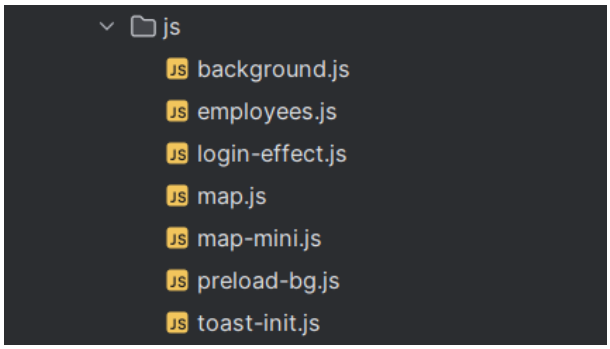

Рисунок 3.2 – файли з використанням JavaScript

Основне застосування JavaScript у системі:

– показ та приховування динамічних блоків;
– обробка фонових ефектів;
– вивід підказок, повідомлень про успіх або помилки;
– анімація під час завантаження сторінки;
– обробка карти;
– ефекти входу на сайт.

Використовуваючи JavaScript з’явилася змога зробити інтерфейс інтерактивним. Та досягти взаємодію користувача із системою реалізувати швидшою та приємнішою.

3.3 Робота з базою даних

Вебсистема працює з реляційною базою даних для зберігання інформації. В проекті для забезпечення швидкої інтеграції та легкого доступу було використано SQLite як основну систему управління бази даних, у поєднанні з ORM-бібліотекою SQLAlchemy. На рисунках 3.3, 3.4 та 3.5 можна побачити по частинам структуру бази даних.

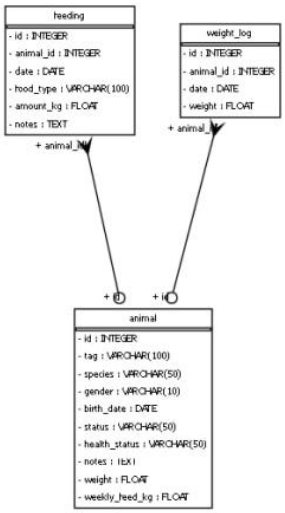

Рисунок 3.3 – Частина БД з таблицею animal, до якої прив’язуються записи з таблиць feeding та weith_log

Таблиця animal є центральною, до якої прив’язуються записи з таблиць feeding та weith_log. Тут реалізується зв'язок багато-до-одного. Це дозволяє вести історію годування до кожної тварини, зберігати динаміку ваги та зроблено для прогнозів приросту.

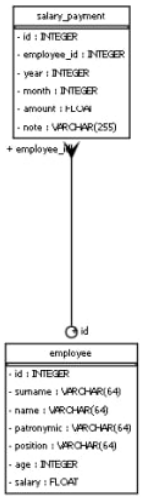

Рисунок 3.4 – Частина БД з таблицею employee, що підв’язує таблицю salary_payment

Таблиця employee відповідає за облік працівників. Підв’язує таблицю salary_payment, в якій міститься інформація про заробітну зарплатню кожного працівника за місяця та роки. Тут працює зв'язок один до багатьох.

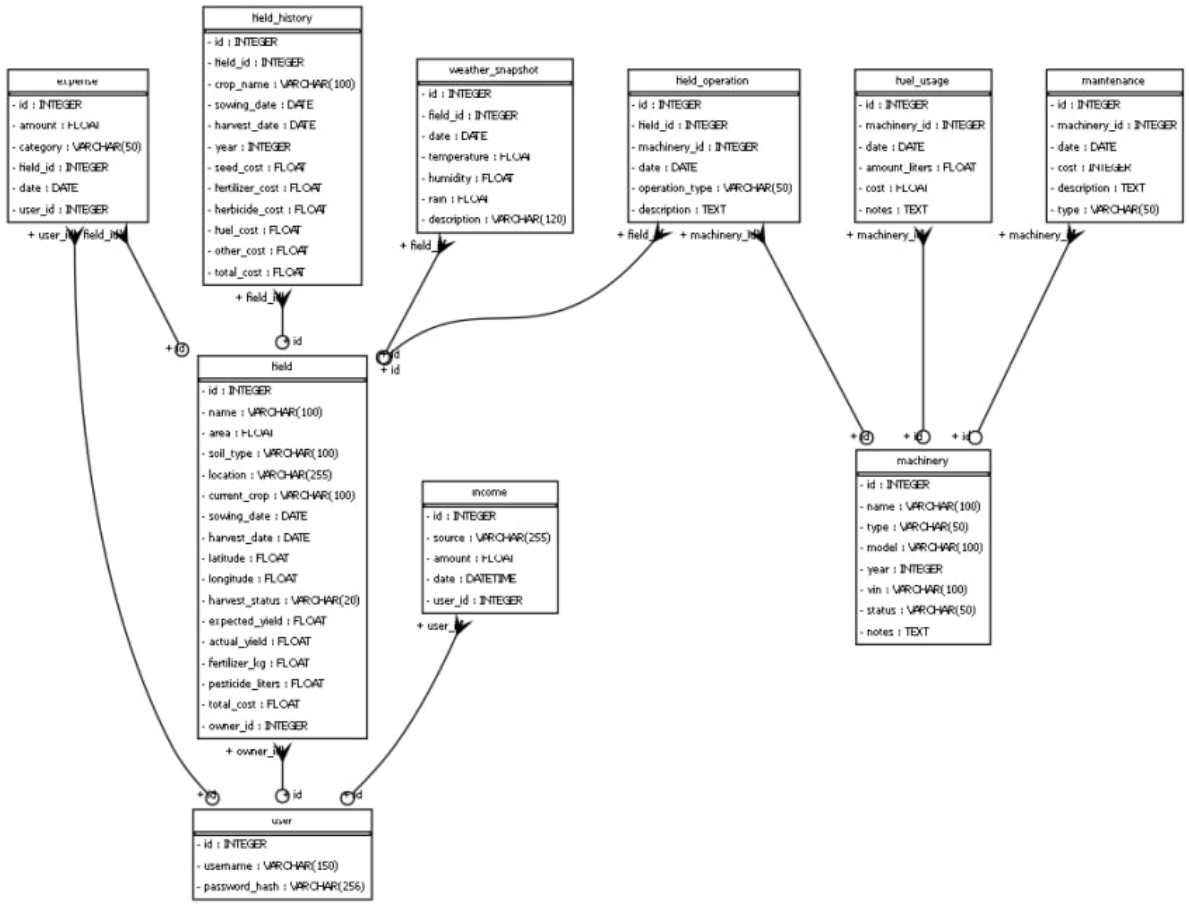

Рисунок 3.5 – Частина БД, яка охоплює усі інші класи

На даному рисунку відображається 70% бази даних, який охоплює усі інші класи:

– user відповідає за зберігання облікових даних користувача;

– field містить інформацію про поля: назву, площу, культуру, координати, тип ґрунту, витрати на добрива та пестициди, статус збору врожаю;

– fieldhistory зберігає історичні дані про поле;

– weathersnapshot містить погодні дані для кожного поля в задану дату. Також дані використовуються для прогнозування;

– expense фіксує витрати користувача за відповідні категорії;

– income місить записи про доходи користувача;

– machinery зберігає дані про техніки;

– maintenance відповідає за інформацію про технічне обслуговування фермерської техніки;

– fuelusage містить записи про витрати пального;

– fieldoperation зберігає дані про агротехнічні операції.

3.3.1 SQLite

Для база даних СУБД під назвою SQLite. SQLite – це компактна, вбудована система керування реляційними базами даних (СУБД). Вона відрізняється від багатьох інших СУБД, таких як MySQL або PostgreSQL, тим, що вона вбудовується безпосередньо в додаток, не вимагаючи окремого сервера баз даних [16]. У проекті база даних ініціалізується за допомогою скрипту init_db.py.

Основні переваги SQLite це:

– простота в налаштувані;

– швидкість для середніх обсягів даних;

– зручність перенесення між системами;

– сумісність із SQL-стандартами.

3.3.2 SQLAlchemy

Для взаємодії вебсистеми з базою даних використано SQLAlchemy. SQLAlchemy – це бібліотека для роботи з реляційними базами даних в Python. Вона також підтримує транзакції для гарантії цілісності даних [17].

Ініціалізація SQLAlchemy здійснюється у проекті завдяки файлу extensions.py, а моделі визначаються у models.py.

Переваги використання SQLAlchemy це:

– повна абстракція від SQL-запитів;
– зручність у створенні, оновленні та видаленні записів;
– сумісність з різними СУБД;
– можливість легкої міграції до інших СУБД у майбутньому.

3.4 Інтеграція штучного інтелекту

Головні функції реалізованої вебсистеми це прогнозування за допомогою регресійних нейронних мереж та створення чат-боту з інтеграцією ClaudeAI. Було реалізовано дві моделі для прогнозування росту тварин та врожайності полів. Обидві побудовані з використанням методів машинного навчання та бібліотеки scikit-learn.

Використано саме scikit-learn, бо це безплатна бібліотека машинного навчання, написана на Python. Вона надає широкий вибір алгоритмів навчання з учителем і без нього. Одна з основних переваг бібліотеки полягає в тому, що вона працює на основі декількох поширених математичних бібліотек і легко інтегрує їх одна з одною. Ще однією перевагою є широка спільнота й докладна документація [18].

3.4.1 Навчання моделей

Ці дві моделі використовують підхід регресії для побудови прогнозування. Регресія – алгоритм, який забезпечує лінійний зв’язок між незалежною змінною та залежною змінною [19]. На відмінну класифікації, де визначають категорію, регресійні моделі прогнозують значення.

По-перше реалізовано модель прогнозування приросту ваги тварин. Для цього використовувалась нейрона мережа MLPRegressor. У лістингу 3.1 надано програмний код навчання моделі прогнозування приросту ваги.

Лістинг 3.1 – Програмний код навчання моделі прогнозування приросту ваги

```
df = pd.read_csv("animal_test.csv", encoding="utf-8")

required_columns = ["species", "age_months",
"food_kg_week", "current_weight", "health_status",
"weight_gain"]

missing = set(required_columns) - set(df.columns)
if missing:
    raise ValueError(f"Відсутні стовпці: {missing}")

y = df["weight_gain"]
X = df.drop(columns=["weight_gain"])

categorical_features = ["species", "health_status"]
numerical_features = ["age_months", "food_kg_week",
"current_weight"]

preprocessor = ColumnTransformer([
    ("cat", OneHotEncoder(handle_unknown="ignore"),
categorical_features),
    ("num", StandardScaler(), numerical_features)
])

model = Pipeline([
    ("preprocess", preprocessor),
    ("regressor", MLPRegressor(hidden_layer_sizes=(64,
32), max_iter=500, random_state=42))
```

Продовження лістингу 3.1

```
])
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
model.fit(X_train, y_train)

score = model.score(X_test, y_test)
print(f"R² score на тесті: {score:.2f}")

joblib.dump(model,
"animal_growth_predictor_retrained.pkl")
print(" Модель збережено у
'animal_growth_predictor_retrained.pkl'")
```

Для цієї моделі був створений за допомогою скрипту набір даних animal_test.csv з усіма можливими випадками. У файлі train_growth_model.py проходило навчання моделі регресією.

Спочатку відбувалося завантаження даних, щоб переконатися у наявності всіх необхідних колонок. Далі дані розділяються на вхідні ознаки та цільовіу змінну. Тому що в наборі присутні як числові, так і текстові поля. Потім дані було поділено на тренувальні та тестові, а для якості моделі використано коефіцієнт детермінації. Після завершення навчання, модель зберіглася у файл animal_growth_predictor_retrained.pkl, що довзволило користуватися прогнозуванням у вебсистемі.

По-друге впроваджено модель прогнозування врожайності полів. Ця модель теж регресійна, але зроблена за допомогою нейроної моделі градієнтного бустингу на основі дерев рішень – XGBRegressor. Та підтягує кліматичні дані з стороннього сервісу OpenWeather. У лістингу 3.2 надано програмний код навчання моделі прогнозування врожайності полів.

Лістинг 3.2 – Програмний код навчання моделі прогнозування врожайності полів

```
df = pd.read_csv("crop_yield_data.csv")

features = ["crop", "soil_type", "area", "sowing_month",
"avg_temp", "avg_humidity", "total_rain"]
X = pd.get_dummies(df[features])
y = df["yield_kg"]

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
model = XGBRegressor(n_estimators=100)
model.fit(X_train, y_train)

from math import sqrt
mse = mean_squared_error(y_test, model.predict(X_test))
rmse = sqrt(mse)

print(f" RMSE: {rmse:.2f} кг")
joblib.dump(model, "crop_yield_predictor.pkl")
print(" Модель збережена у crop_yield_predictor.pkl")
```

Для цієї моделі було створено набір даних crop_yield_data.csv за допомогою скрипта. Далі датасет був поділен на тренувальні та тестові вибірки. Навчання проводилося на основі 100 дерев рішень, які постопупово вдосканаляють прогноз з кожним новим деревом. Для оцінки точності такод було використано коефіцієнт детермінації. Після цього модель була збережена у файл crop_yield_predictor.pkl.

3.4.2 Інтеграція чат-бота на базі ClaudeAI

У проекті також реалізований інтелектуальний чат-бот, який функціонує на основі штучного інтелекту ClaudeAI від компанії Anthropic.

Сама компанія Anthropic – це стартап зі штучним інтелектом, заснований колишніми членами OpenAI, включаючи братів і сестер Даріо та Даніелу Амодей, які зіграли ключову роль у розробці GPT-3. Стурбовані безпекою штучного інтелекту, вони залишили OpenAI і запустили Anthropic у 2021 році, зосередившись на етичній розробці штучного інтелекту. Одним із творінь Anthropic є Claude, помічник зі штучним інтелектом, призначений для виконання різноманітних завдань, таких як написання, аналіз і кодування. З моменту свого дебюту в березні 2023 року Claude зазнав кілька оновлень. Claude 3, випущений у березні 2024 року, навіть додав можливість аналізувати зображення. Доступний через API, вебпрограми та мобільні додатки, Claude створений для природної розмовної взаємодії, відмінно справляючись із такими завданнями, як узагальнення, запитання та відповіді, прийняття рішень і написання коду [20].

У системі бот виконує роль віртуального асистента. В його функціоналі закладено такі задачі:

– надавати фермерам потрібну інформацію по господарству;
– допомагати з прогнозами;
– надавати поради щодо годівлі тварин та економії ресурсів.

У лістингу 3.3 надано програмний код чат бота з інтеграцію Claude AI.

Лістинг 3.3 – Програмний код чат бота з інтеграцію Claude AI

```
claude_chatbot_bp = Blueprint('claude_chatbot', __name__)
def months_old(birth_date):
    if not birth_date:
   return "?"
today = date.today()
return (today.year - birth_date.year) * 12 +
today.month - birth_date.month

@claude_chatbot_bp.route('/claude-chatbot',
methods=['POST'])
```

Продовження лістингу 3.3

```
def handle_claude_chat():
    user_message = request.json.get("message", "").strip()

    animals = Animal.query.all()
    animal_info = "\n".join([
        f"{a.species}, {months_old(a.birth_date)} міс.,
{a.weight} кг, стан: {a.health_status}"
        for a in animals
    ]) or "Немає тварин у базі."

    fields = Field.query.all()
    field_info = "\n".join([
        f"Поле {f.name}: культура {f.current_crop}, площа
{f.area} га"
        for f in fields
    ]) or "Немає полів у базі."

{animal_info}
{field_info}

    headers = {
        "Authorization": f"Bearer
{os.getenv('OPENROUTER_API_KEY')}",

        "Content-Type": "application/json"
    }
   payload = {
        "model": "anthropic/claude-3-haiku",
        "messages": [

            {"role": "system", "content": system_prompt},
            {"role": "user", "content": user_message}
        ]
    }
```

Продовження лістингу 3.3

```
    try:
        response =
requests.post("https://openrouter.ai/api/v1/chat/completio
ns", json=payload, headers=headers)
        reply =
response.json()["choices"][0]["message"]["content"].strip(
)
        return jsonify({"response": reply})
    except Exception as e:
        return jsonify({"response": f"Виникла помилка
Claude/OpenRouter: {e}"}), 500
```

Чат-бот Claude інтегровано у вебдодаток через окремий модуль, який створює спеціальний маршрут /claude-chatbot. Коли користувач надсилає запит у чаті, повідомлення обробляється сервером: спочатку зчитуються дані з бази формується контекст для Claude, після чого сервер надсилає запит до хмарного API Claude AI через платформу OpenRouter [21].

Бот повертає сформовану відповідь, яка одразу ж з'являється в інтерфейсі користувача. У разі помилки система повідомляє про це через відповідне повідомлення. Така інтеграція дозволяє надавати фермеру персоналізовані поради на основі реальних даних з господарства.

4 ВЕБСИСТЕМА УПРАВЛІННЯ ФЕРМЕРСЬКИМ ГОСПОДАРСТВОМ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

4.1 Облік тварин

У вебсистемі фермерського господарства облік тварин є ключовою частиною. Для обліку реалізовані необхідні функції:

– переглядання повної історії по кожній тварині;
– редагування та оновлювання даних;
– пошук за ідентифікаційним номером;
– сортування та фільтрація записів;
– додавання нових записів та видалення непотрібних.

При перегляданні кожної тварини, додано функціонал прогнозування ваги та AI-рекомендацій щодо годівлі. На рисунку 4.1 можна побачити розроблений інтерфейс для обліку тварин, а на рисунку В.8 додатку В, можна побачити прогноз та поради від реалізованої моделі.

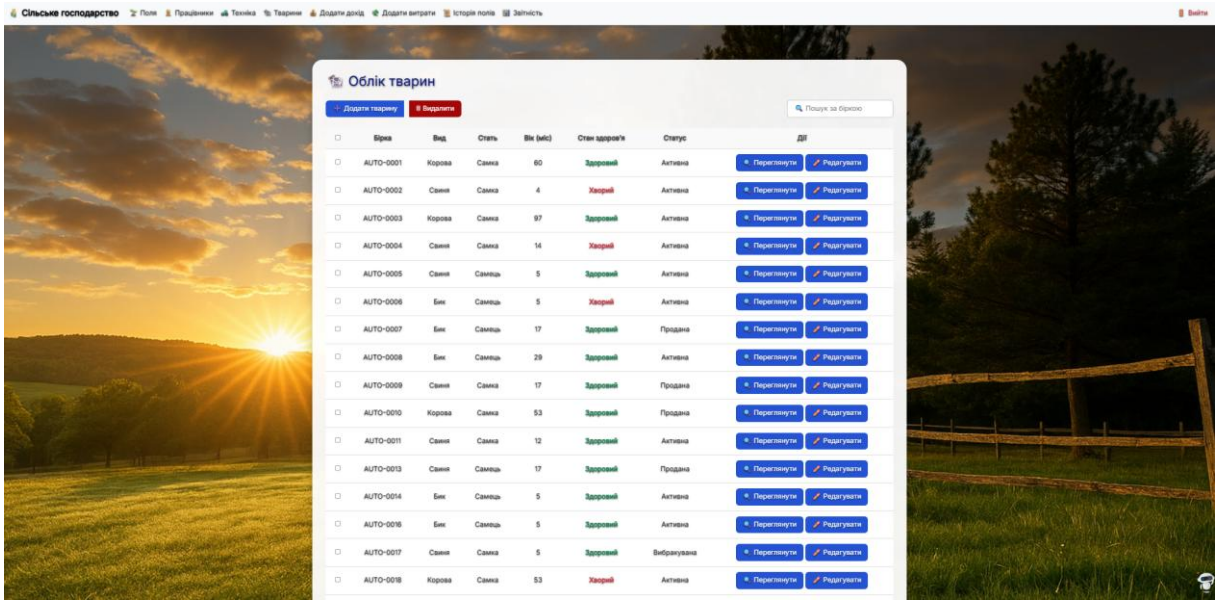

Рисунок 4.1– Розроблений інтерфейс для обліку тварин

4.2 Облік полів

У вебсистемі фермерського господарства підсистема обліку полів дозволяє зберігати, переглядати та редагувати ключові агрономічні параметри кожної земельної ділянки. Для роботи з полями реалізовані наступні функції:

– додавання нового поля із зазначенням площі, типу ґрунту, культури, координат;
– перегляд усіх зареєстрованих полів у вигляді модулів;
– редагування та оновлення даних по кожному полю;
– інтеграція з метеоданими для подальшого аналізу;
– пошук за назвою або культурою, а також фільтрація записів.

Окремо реалізовано прогнозування врожайності на основі навченої моделі ШІ, яка враховує погодні та агрономічні фактори. На рисунку 4.2 зображено інтерфейс поля з даними поля, яке знаходиться в обліку, а на рисунку В.10 додатку В – приклад роботи модуля прогнозування врожаю.

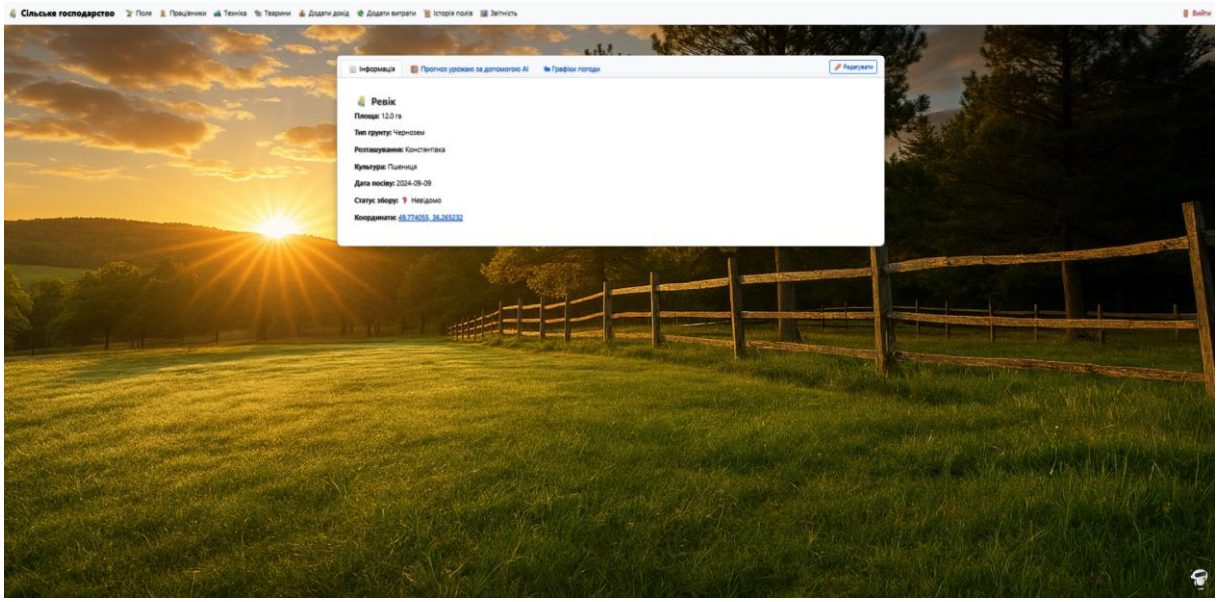

Рисунок 4.2 – Інтерфейс з даними поля, яке знаходиться в обліку

4.3 Чат-бот з інтеграцією Claude AI

Для підвищення зручності користування вебсистемою у проекті реалізовано чат-бота з інтеграцією штучного інтелекту ClaudeAI. Чат-бот працює безпосередньо у вікні браузера та доступний з будь-якої сторінки системи. Бот здатен відповідати на запитання користувача в реальному часі, реагує на введені повідомлення, аналізує контекст і надає поянеснення. Основний функціона чат-бота це:

– відповіді на аграрні запити;
– пояснення прогнозів моделей, якщо вони незрозумілі користувачу;
– поради щодо годівлі, стану здоров’я, посівів;
– підтримка діалогу природною мовою українською.

Інтеграція виконана через OpenRouter API, а весь запит передбачає підстановку поточних даних ферми. На рисунку 4.3 можна побачити інтерефейс чат-бота, а на рисунку В.11 додатку В – приклад роботи чат-бота.

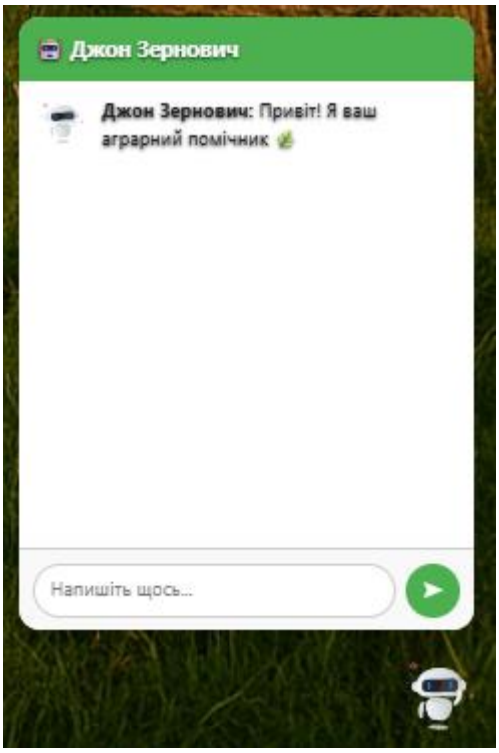

Рисунок 4.3 – Інтерфейс чат-бота

ВИСНОВКИ

У рамках даної кваліфікаційної роботи було успішно розроблено та реалізовано сучасну вебсистему управління фермерським господарством з використанням компонентів штучного інтелекту. Під час дослідження було проаналізовано поточні проблеми в аграрній сфері, зокрема неефективність традиційних методів обліку, відсутність аналітики в реальному часі, ризики втрат даних та необхідність прийняття швидких і точних рішень в умовах змінного середовища.

У ході реалізації було створено масштабовану вебсистему, яка включає модулі для обліку тварин, полів, техніки, витрат, а також прогнозування врожайності та приросту ваги тварин. У проекті застосовано дві регресійні моделі машинного навчання: MLPRegressor для прогнозу ваги тварин та XGBRegressor для прогнозування врожайності полів, що дозволяє користувачу отримувати точні оцінки на основі наданих параметрів.

Однією з ключових особливостей стало впровадження інтелектуального чат-бота на базі Claude AI, який взаємодіє з користувачем у режимі реального часу та надає персоналізовані поради на основі поточних даних з бази.

Система побудована з урахуванням принципів модульності, безпеки, гнучкості та адаптивності, що дозволяє легко масштабувати її під потреби конкретного господарства. Для реалізації було використано сучасні інструменти розробки – Python, Flask, HTML/CSS, JavaScript, SQLAlchemy, бібліотеки машинного навчання, API для погоди та хмарних ШІ-моделей.

У результаті розробки вдалося створити вебсистему, яка не лише автоматизує облік та щоденні операції, а й забезпечує аналітичну підтримку прийняття рішень, покращуючи ефективність господарювання. Система демонструє практичну реалізацію концепції розумного фермерства та є прикладом успішного впровадження штучного інтелекту в аграрну діяльність.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Що таке NDVI, як його використовують у сільському господарстві - з якими камерами - agtecher: The Agri Tech Place. *agtecher: The Agri Tech Place*. URL: https://agtecher.com/uk/what-is-ndvi-how-is-it-used-in-agriculture-with-which-cameras/ (дата звернення: 21.05.2025).

2. Про охорону навколишнього природного середовища. *Офіційний вебпортал парламенту України*. URL: https://zakon.rada.gov.ua/laws/show/1264-12#Text (дата звернення: 21.05.2025).

3. Що таке rest api: основні принципи та практики застосування. *FoxmindEd*. URL: https://foxminded.ua/shcho-take-rest-api/ (дата звернення: 21.05.2025).

4. Kuchmiiova T., Moroz T., Sheshunova A. Use of artificial intelligence in agriculture. *Modern economics*. 2023. Vol. 39, no. 1. P. 69–74. URL: https://doi.org/10.31521/modecon.v39(2023)-10 (date of access: 03.05.2025).

5. Мінагрополітики. *Elevatorist.com – Elevatorist.com*. URL: https://elevatorist.com/kompanii/238-minagroprod (дата звернення: 22.05.2025).

6. Положення про Державну податкову службу України. *tax.gov.ua*. URL: https://tax.gov.ua/pro-sts-ukraini/pologennya (дата звернення: 23.05.2025).

7. Інтернет речей (IoT): що це та його використання у сільському господарстві - WEAGRO. *WEAGRO*. URL: https://weagro.com.ua/blog/internet-rechej-iot-shho-cze-ta-jogo-vykorystannya-v-silskomu-gospodarstvi/ (дата звернення: 24.05.2025).

8. Розробка з боку Front end. *dan it*. URL: https://dan-it.com.ua/uk/blog/rozrobka-z-boku-front-end-shho-ce-take-i-chim-vidriznjaietsja-vid-back-end/ (дата звернення: 25.05.2025).

9. Frontend, backend чи full stack: в чому відмінності. *dan it*. URL: https://dan-it.com.ua/uk/blog/frontend-backend-ili-full-stack-v-chem-razlichija-i-chto-luchshe-vybrat-dlja-starta-karery-v-it-sfere/ (дата звернення: 25.05.2025).

10. Створення інтерактивної картографії на вебсайті з використанням Leaflet.js - IT рейтинг UA. *IT рейтинг України*. URL: https://it-rating.ua/stvorennya-interaktivnoi-kartografii-na-veb-sayti-z-vikoristannyam-leafletjs (дата звернення: 28.05.2025).

11. OpenStreetMap Україна – вільна мапа, покращувати яку може кожен!. *OpenStreetMap Україна – вільна мапа, покращувати яку може кожен!*. URL: https://openstreetmap.org.ua/ (дата звернення: 01.06.2025).

12. Павленко Є. П. Інженерія програмного забезпечення. WEB-програмування : навч. посіб. / Є. П. Павленко, В. М. Бутенко, О. В. Головко. – Харків : УкрДУЗТ, 2019. – 120 с. (date of access: 02.06.2025).

13. Що таке html?. *Український вебдовідник*. URL: https://css.in.ua/article/shcho-take-css_3 (дата звернення: 05.06.2025).

14. Що таке CSS. *Український вебдовідник*. URL: https://css.in.ua/article/shcho-take-html_10 (дата звернення: 05.06.2025).

15. Що таке JavaScript і для чого він потрібен. *Goit*. URL: https://goit.global/ua/articles/shcho-take-javascript-i-dlia-choho-vin-potriben/ (дата звернення: 07.06.2025).

16. Програмування баз даних на SQLite: перші кроки в освоєнні. *FoxmindEd*. URL: https://foxminded.ua/prohramuvannia-baz-danykh-na-sqlite/ (дата звернення: 08.06.2025).

17. Робота з транзакціями в SQLAlchemy. *K2 Cloud ERP: WMS, CRM, document management, CMS, Education system, VDoc, K2. Web System Python, PHP*.URL: https://corp2.eu/p/docs/item/64b2797b6b1502876e76579d45920ada0 (дата звернення: 09.06.2025).

18. Перші кроки в NLP: розглядаємо Python-бібліотеку scikit-learn в реальному завданні. *Dou*. URL: https://dou.ua/lenta/articles/first-steps-in-nlp-scikit-learn/ (дата звернення: 10.06.2025).

19. Лінійна регресія – IT Master - електроніка та програмування. *Головна – IT Master - електроніка та програмування*. URL: https://itmaster.biz.ua/programming/vision/linear-regression.html (дата звернення: 11.06.2025).

20. Що таке Anthropic, компанія, що стоїть за Claude AI?. *Oksim*. URL: https://www.oksim.ua/2024/10/14/shho-take-anthropic-kompaniya-shho-stoït-za-claude-ai/ (дата звернення: 12.06.2025).

21. OpenRouter. *OpenRouter*. URL: https://openrouter.ai/ (date of access: 13.06.2025).