

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

ПОГОДЖЕНО
Декан факультету

_____ **Ігор БОЛБОТ**

«___» _____ **2026**
р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри
інформаційних систем і
технологій

_____ **Михайло ШВИДЕНКО**

«___» _____ **2026** р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«Система підтримки управління виробничою діяльністю
сільськогосподарських підприємств регіону»**

Спеціальність 126 «Інформаційні системи і технології»

Освітньо-професійна програма «Інформаційні системи і технології»

Гарант освітньої програми
к.е.н., доцент

Максим МОКРІЄВ

**Керівник бакалаврської
кваліфікаційної роботи**
к.е.н., доцент

Михайло САДКО

Виконав

Владислав БРЕЧКА

Київ – 2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних систем і
технологій

к.е.н, доцент _____ Михайло ШВИДЕНКО
“ _____ ” ____ 2026р.

З А В Д А Н Н Я

**ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧА**

Бречки Владислава Олександровича

Спеціальність «126 Інформаційні системи і технології»

Освітньо-професійна програма «Інформаційні системи і технології»

Тема бакалаврської кваліфікаційної роботи

«Система підтримки управління виробничою діяльністю
сільськогосподарських підприємств регіону (на прикладі Кам'янського району
Дніпропетровської області)»

затверджена наказом від «19» грудня 2025 року № 3205«С»

Термін подання завершеної роботи на кафедру «__» _____ 2026 року

Вихідні дані до бакалаврської кваліфікаційної роботи:

Бази даних показників первинних звітних матеріалів сільськогосподарських
підприємств України та їх обробка з допомогою створеної інформаційної
системи.

Перелік питань, що підлягають дослідженню:

Створити сховище даних, яке включає структуровані бази даних показників
первинних звітних матеріалів господарської діяльності сільськогосподарських
підприємств України; визначити ключові ознаки та забезпечити зв'язування джерел
інформації по ключовим полям, реалізувати можливості відбору необхідних порцій
даних згідно ключових ознак та формування відповідних результативних матеріалів;
використати методи та моделі MS ACCESS, MS EXCEL для аналізу та визначення
перспектив сталого розвитку сільськогосподарських підприємств регіону.

Дата видачі завдання «19» грудня 2025 р.

**Керівник бакалаврської
кваліфікаційної роботи**

Михайло САДКО

**Завдання взяв до
виконання**

Владислав БРЕЧКА

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, ІСНУЮЧИХ ІТ, ВИЗНАЧЕННЯ ВИМОГ ТА РОЗРОБКА ТЕХНІЧНОГО ЗАВДАННЯ.....	8
1.1 Характеристика об'єкта автоматизації та аналіз предметної області... ..	8
1.2 Аналіз існуючих інформаційних технологій та аналогів.....	9
1.3 Обґрунтування вибору інформаційних технологій та архітектури системи.....	15
РОЗДІЛ 2. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ВИРОБНИЧИМ ПОТЕНЦІАЛОМ ПІДПРИЄМСТВ РЕГІОНУ.....	26
2.1 Інфологічне та даталогічне проєктування реляційної бази даних системи.....	26
2.2 Проєктування клієнт-серверної архітектури веб-середовища.....	30
2.3 Проєктування людино-машинного інтерфейсу та інформаційного забезпечення користувача.....	36
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ІНСТРУКЦІЯ З ЕКСПЛУАТАЦІЇ ВЕБ-СИСТЕМИ АГРОМОНІТОРИНГУ.....	41
3.1 Структура програмного комплексу та обґрунтування вибору інструментарію розробки.....	41
3.2 Розробка інтерфейсу користувача та семантична розмітка (Файл index.html).....	44
3.3 Візуальне оформлення та адаптивна верстка веб-середовища (Файл style.css).....	47
3.4 Програмна реалізація бізнес-логіки та інтерактивної моделі взаємодії (Файл app.js).....	54
3.5 Інструкція з експлуатації та керівництво оператора інформаційної системи.....	58
3.6 Тестування програмного комплексу та валідація результатів.....	51
ВИСНОВКИ.....	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62
ДОДАТКИ.....	63
ДОДАТОК А.....	63
ДОДАТОК Б.....	86

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІС – інформаційна система

ІТ – інформаційна технологія

ІАС – інформаційно-аналітична система

СУБД – система управління базами даних

HTML – мова гіпертекстової розмітки веб-документів

PHP – скриптова мова програмування

CSS – каскадні таблиці стилів

JavaScript – мови програмування

SPSS – програмний засіб для статистичного аналізу даних

ВСТУП

На сучасному етапі розвитку цифрової економіки ефективне функціонування агропромислового комплексу (АПК) України та її регіонів безпосередньо залежить від рівня впровадження новітніх інформаційних систем і технологій (ІСТ). Аграрний сектор Кам'янського району Дніпропетровської області оперує значними масивами ретроспективної та поточної статистичної інформації, що характеризує площі посівів, валові збори, виробничі витрати та фінансові результати діяльності суб'єктів господарювання. Традиційні підходи до обробки державної статистичної звітності, які базуються на паперовому документообігу або використанні локальних ізольованих електронних таблиць, уже не відповідають сучасним вимогам оперативності, надійності та якості управління.

Ручна агрегація та аналіз розрізнених даних призводять до значних трудовитрат, підвищують ризик виникнення помилок через «людський фактор» та унеможливають швидке проведення складного багатовимірного аналізу, такого як статистичне групування підприємств. У зв'язку з цим виникає гостра потреба в проектуванні та реалізації спеціалізованих веб-орієнтованих інформаційних систем підтримки управління (СПУ). Такі системи дозволяють централізовано акумулювати дані, автоматизувати розрахунки макроекономічних показників та надавати управлінським структурам територіальних громад гнучкі інструменти моніторингу для прийняття обґрунтованих стратегічних рішень, що й зумовлює високу актуальність теми дипломної роботи.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконана відповідно до науково-дослідних тем кафедри інформаційних систем і технологій Національного університету біоресурсів і природокористування України, спрямованих на цифровізацію процесів управління та автоматизацію моніторингу в аграрному секторі економіки.

Метою бакалаврської кваліфікаційної роботи є теоретичне обґрунтування,

проектування та практична програмна реалізація веб-орієнтованої інформаційної системи підтримки управління виробничою діяльністю сільськогосподарських підприємств Кам'янського району Дніпропетровської області для підвищення оперативності та обґрунтованості управлінських рішень.

Для досягнення поставленої мети було сформульовано та вирішено такі завдання:

Проаналізувати теоретико-методологічні засади розробки та використання інформаційних систем і технологій в управлінні аграрним виробництвом;

- провести системний аналіз об'єкта автоматизації та дослідити існуючу технологію збору й аналізу статистичної звітності в регіоні;

- виконати концептуальне проектування реляційної бази даних СПУ та провести її послідовну нормалізацію для усунення надлишковості;

- спроектувати триланкову архітектуру веб-середовища системи на основі концепції «тонкого клієнта» та розробити макети інтерфейсу користувача;

- реалізувати розрахунково-аналітичні модулі системи для моніторингу ефективності рослинництва й фінансових результатів підприємств району;

- провести практичну апробацію системи на реальних статистичних масивах даних, автоматизувати метод аналітичного групування підприємств та оцінити техніко-економічну ефективність впровадження ІТ-продукту.

Об'єкт дослідження – процес управління виробничою діяльністю сільськогосподарських підприємств Кам'янського району Дніпропетровської області.

Предмет дослідження – інформаційне та програмне забезпечення системи підтримки управління виробничою діяльністю сільськогосподарських підприємств Кам'янського району Дніпропетровської області.

Методи дослідження. Для розв'язання поставлених завдань у роботі було використано комплекс методів наукового дослідження: метод системного аналізу для вивчення інформаційних потоків предметної області; концепції

об'єктно-орієнтованого проєктування та архітектури для розробки структури веб-середовища; метод математико-статистичного аналізу та аналітичного групування для реалізації модулів дослідження взаємозв'язку факторів у базі даних.

Практичне значення отриманих результатів. Розроблена інформаційна система та її аналітичні модулі (ZEEP, ZEOP, ZWAGOP) можуть безпосередньо впроваджуватися в практику роботи управлінь агропромислового розвитку, органів місцевого самоврядування та територіальних громад для автоматизації збору звітності, суттєвого скорочення трудовитрат фахівців (на 85-90%) та підвищення обґрунтованості рішень щодо підвищення ефективності аграрного сектора регіону.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СИСТЕМИ ПІДТРИМКИ УПРАВЛІННЯ

1.1 Характеристика об'єкта автоматизації та аналіз предметної області

Об'єктом автоматизації є процес управління виробничою діяльністю сільськогосподарських підприємств Кам'янського району Дніпропетровської області. Це управлінський процес, який охоплює збір інформації про виробництво, її подальшу обробку, аналіз та використання для прийняття рішень на рівні підприємств і органів управління аграрним сектором.

У практичній діяльності аграрного сектору управління виробництвом є складним і багаторівневим процесом. Це пов'язано з тим, що результати діяльності підприємств залежать не лише від організаційних рішень, а й від зовнішніх факторів, зокрема погодних умов, сезонності робіт, якості ґрунтів та інших природних чинників. У зв'язку з цим інформація, яка використовується в управлінні, має бути актуальною, повною та систематизованою.

Предметна область включає взаємодію різних учасників процесу: сільськогосподарських підприємств, фермерських господарств, аналітичних підрозділів та органів місцевого самоврядування, які здійснюють моніторинг стану аграрного сектору. Кожен із цих учасників працює з даними, що відображають різні аспекти виробничої діяльності.

До основних інформаційних показників, що використовуються в аналізі, належать площі посівів сільськогосподарських культур, обсяги виробництва продукції рослинництва і тваринництва, рівень урожайності, витрати ресурсів (паливо, добрива, насіння), а також фінансові результати діяльності підприємств. Саме ці дані дають змогу оцінювати ефективність виробництва та визначати напрями його розвитку.

На практиці такі дані формуються у вигляді виробничої та статистичної звітності, яку подають підприємства аграрного сектору. Надалі ця інформація використовується для узагальнення показників за районом, підготовки

аналітичних довідок та прийняття управлінських рішень.

Аналіз існуючої організації обробки даних показує, що значна частина інформації зберігається у вигляді окремих електронних файлів, найчастіше у форматі таблиць. Часто дані передаються між учасниками процесу без єдиного стандартизованого сховища, що ускладнює їх подальше використання.

Такий підхід має низку практичних недоліків. По-перше, відсутність централізованої бази даних призводить до дублювання інформації та втрати її цілісності. По-друге, обробка даних виконується переважно вручну, що збільшує витрати часу та підвищує ризик помилок. По-третє, аналітична інформація формується із затримкою, що знижує її актуальність для ухвалення рішень. Крім того, доступ до узагальнених даних часто є обмеженим і залежить від конкретних виконавців або форматів звітності.

З точки зору системного підходу, процес управління виробничою діяльністю можна розглядати як потік інформації між джерелами даних та користувачами. Вхідною інформацією є первинні показники діяльності підприємств, що надходять у вигляді звітів. Ці дані проходять етапи обробки, перевірки та узагальнення. Вихідними результатами є аналітичні показники, таблиці, графіки та звіти, які використовуються для підтримки ухвалення управлінських рішень.

Таким чином, наявна організація роботи з даними в аграрному секторі потребує вдосконалення. Основною проблемою є відсутність єдиної автоматизованої системи, яка б забезпечувала централізоване зберігання інформації, її швидку обробку та зручне подання результатів користувачам. Це створює передумови для розробки інформаційної системи підтримки управління виробничою діяльністю сільськогосподарських підприємств, яка дозволить підвищити ефективність роботи з даними та якість управлінських рішень.

1.2 Аналіз існуючих інформаційних технологій та аналогів

Ефективне проєктування інформаційної системи підтримки управління виробничою діяльністю в аграрному секторі потребує аналізу наявних

програмних рішень, які використовуються для збору, обробки та аналізу аграрних даних. Сучасний ринок інформаційних технологій для агропромислового комплексу представлений різними класами систем, що відрізняються функціональним призначенням, рівнем автоматизації та архітектурними підходами.

Існуючі рішення умовно поділяють на три основні групи:

- ERP-системи управління підприємством забезпечують облік фінансових, виробничих і кадрових ресурсів у межах підприємства або агрохолдингу.

- AgTech та IoT-платформи, що орієнтовані на оперативний моніторинг виробничих процесів у рослинництві з використанням супутникових даних, GPS-технологій і сенсорних систем.

BI-системи (Business Intelligence) призначені для консолідації даних із різних джерел, їх обробки та візуалізації у вигляді аналітичних панелей.

ERP-системи управління підприємством

ERP-системи (Enterprise Resource Planning) є класом корпоративних інформаційних систем, які забезпечують комплексне управління ресурсами підприємства. Вони призначені для автоматизації основних бізнес-процесів, зокрема фінансового обліку, управління виробництвом, логістикою, постачанням, складськими запасами, а також кадровими ресурсами. У контексті аграрного сектору ERP-системи найчастіше впроваджуються на рівні окремих підприємств або великих агрохолдингів, де існує потреба в централізованому управлінні великою кількістю внутрішніх процесів.

Функціонально такі системи забезпечують інтеграцію різних підсистем у єдине інформаційне середовище, що дає змогу отримувати узгоджені дані про діяльність підприємства в режимі реального часу. Наприклад, фінансові модулі відповідають за бухгалтерський облік, формування фінансової звітності та контроль витрат, виробничі модулі — за планування та облік виробничих операцій, а кадрові модулі — за управління персоналом, нарахування заробітної

плати та облік трудових ресурсів. Додатково ERP-системи можуть включати модулі управління постачанням і збутом, що дозволяють контролювати весь цикл руху ресурсів від закупівлі до реалізації продукції.

Серед основних переваг ERP-систем слід відзначити високий рівень інтеграції даних, що дозволяє уникати їх дублювання та забезпечує узгодженість інформації між різними підрозділами підприємства. Також важливою перевагою є можливість автоматизації великої кількості рутинних операцій, що зменшує навантаження на персонал і підвищує точність обліку. ERP-рішення забезпечують детальну аналітику внутрішніх процесів підприємства, що сприяє ефективнішому плануванню та контролю ресурсів.

Водночас ERP-системи мають і певні обмеження з точки зору задач регіонального або міжпідприємницького рівня. Насамперед вони орієнтовані на внутрішні процеси однієї організації або групи підприємств і не забезпечують повноцінного аналізу агрегованих статистичних даних на рівні району чи регіону. Крім того, впровадження таких систем потребує значних фінансових витрат, тривалого налаштування та адаптації до специфіки конкретного підприємства. У багатьох випадках ERP-системи є складними у супроводі та потребують наявності кваліфікованих ІТ-фахівців.

Таким чином, ERP-системи є ефективними інструментами для управління внутрішніми ресурсами підприємств аграрного сектору, однак вони не вирішують завдання узагальненого регіонального аналізу та підтримки управлінських рішень на рівні адміністративних одиниць.

AgTech та IoT-платформи

До сучасних цифрових рішень в аграрному секторі належать AgTech-платформи, призначені для оперативного моніторингу стану сільськогосподарських угідь і підтримки прийняття рішень у межах виробничих процесів. До найпоширеніших представників цього класу систем належать Cropio та Climate FieldView. Такі платформи функціонують у хмарному середовищі, що забезпечує користувачам доступ до даних у режимі реального часу через веб-інтерфейси або мобільні застосунки. Основна концепція їх роботи

полягає у зборі та аналізі геопросторових даних, що характеризують стан посівів і динаміку їх розвитку.

Функціональні можливості подібних систем включають відстеження стану сільськогосподарських культур на основі супутникових знімків, аналіз індексів вегетації, таких як NDVI та EVI, що дозволяє оцінювати розвиток рослин, а також моніторинг погодних умов і їх впливу на виробничі процеси. Крім того, за допомогою GPS-технологій і телеметрії здійснюється контроль руху сільськогосподарської техніки, що підвищує точність управління технологічними операціями. У результаті аграрні спеціалісти отримують можливість оперативно оцінювати стан посівів без необхідності безпосереднього виїзду на поля, що суттєво підвищує швидкість реагування на зміни виробничих умов.

Серед основних переваг AgTech-платформ слід відзначити автоматизований збір даних без ручного введення, високу точність геоінформаційного аналізу завдяки використанню супутникових технологій, зручну візуалізацію стану полів у вигляді картографічних шарів, а також застосування сучасних алгоритмів обробки даних, включаючи елементи прогнозування. Це робить такі системи ефективними інструментами для оперативного контролю аграрного виробництва.

Водночас, незважаючи на свої переваги, AgTech-рішення мають обмеження з точки зору їх використання для задач регіонального управління. Насамперед вони орієнтовані на рівень окремого поля або господарства і не призначені для узагальнення інформації на рівні району чи регіону. Крім того, такі системи не підтримують роботу зі статистичною та економічною звітністю підприємств, яка використовується органами управління аграрного сектору, зокрема показниками собівартості, рентабельності та загальних виробничих результатів у регіональному розрізі. Також вони не інтегровані з формами державної статистичної звітності, що ускладнює їх використання для комплексного аналізу діяльності великої кількості підприємств.

Додатковим обмеженням є залежність вартості впровадження від площі

моніторингу, що робить масштабне використання таких рішень економічно недоцільним для органів місцевого самоврядування або районних структур. Таким чином, AgTech-системи є ефективними для оперативного контролю виробничих процесів на рівні господарства, однак вони не забезпечують повного спектра завдань регіонального статистичного аналізу та підтримки управлінських рішень.

BI-системи (Business Intelligence)

Інший клас сучасних інформаційних технологій представлений аналітичними платформами бізнес-аналітики (BI-системами), до яких належать такі поширені рішення, як Microsoft Power BI та Tableau. Ці системи призначені для обробки великих обсягів даних, їх узагальнення та подальшої візуалізації у вигляді інтерактивних аналітичних панелей, що використовуються для підтримки прийняття управлінських рішень.

У практичній діяльності BI-платформи забезпечують підключення до різних джерел даних, зокрема реляційних баз даних, електронних таблиць або зовнішніх API-сервісів. Після підключення здійснюються трансформація, очищення та структурування даних — необхідні етапи підготовки до аналізу. На основі обробленої інформації користувач може будувати інтерактивні графіки, діаграми та зведені звіти, а також формувати аналітичні панелі (дашборди), які відображають ключові показники діяльності у зручному візуальному вигляді.

BI-системи набули широкого поширення завдяки своїй універсальності та високому рівню зручності в користуванні. Вони надають потужні інструменти для візуалізації даних, підтримують роботу з великою кількістю різномірних джерел інформації, дозволяють створювати складні аналітичні моделі та забезпечують відносно просту взаємодію для кінцевих користувачів, які не обов'язково мають глибокі знання у сфері програмування або баз даних. Завдяки цьому такі системи є ефективними інструментами для бізнес-аналітики та аналізу даних у корпоративному середовищі.

Водночас BI-платформи мають низку обмежень, які важливо враховувати в контексті аграрної предметної області. Насамперед вони не мають

спеціалізованої галузевої адаптації для аграрного сектору, тому структуру даних та логіку аналізу потрібно попередньо сформувавши користувачем відповідно до конкретних завдань. Значна частина підготовчих операцій із даними виконується поза межами системи, що передбачає їх попередню очистку, нормалізацію та приведення до єдиного формату.

Сучасні інформаційні технології в аграрному секторі можна умовно поділити на кілька основних класів, кожен з яких виконує специфічні функції в межах управління та аналізу даних. До першого класу належать ERP-системи управління підприємством, які забезпечують комплексний облік і контроль фінансових, виробничих та кадрових ресурсів у межах окремого підприємства або агрохолдингу, а також дозволяють інтегрувати різні бізнес-процеси в єдине інформаційне середовище. Другий клас становлять AgTech та IoT-платформи, які орієнтовані на оперативний моніторинг виробничих процесів у рослинництві із використанням супутникових даних, GPS-технологій та сенсорних систем, що забезпечує можливість відстеження стану посівів і технологічних операцій у режимі реального часу. Третій клас представлений BI-системами (Business Intelligence), які призначені для консолідації даних із різних джерел, їх аналітичної обробки та візуалізації у вигляді інтерактивних панелей і звітів, що використовуються для підтримки прийняття управлінських рішень.

Разом з тим, аналіз можливостей BI-систем показує, що вони переважно орієнтовані на аналітичну обробку вже наявних даних і не забезпечують повноцінного функціоналу для введення та верифікації первинної інформації. У контексті регіонального управління це є суттєвим обмеженням, оскільки первинні дані, що надходять із різних підприємств, потребують попереднього контролю якості, уніфікації та структуризації ще на етапі їх внесення. Додатково слід враховувати, що реалізація спеціалізованих методів агростатистичного аналізу в BI-середовищах часто потребує використання додаткових інструментів або програмування на внутрішніх мовах платформ, що ускладнює налаштування та подальший супровід систем. Також певним обмеженням може бути ліцензійна модель таких рішень, яка у випадку масштабного використання є фінансово

витратною для органів місцевого самоврядування або районних адміністрацій. У результаті ВІ-системи залишаються ефективними засобами візуалізації та аналітики даних, однак не забезпечують повного циклу роботи з інформацією в аграрному секторі, особливо на рівні адміністративного району або регіону.

Аналогічно, AgTech-платформи забезпечують високий рівень оперативного моніторингу виробничих процесів і дозволяють отримувати детальну інформацію про стан посівів, однак вони здебільшого орієнтовані на рівень окремого поля або господарства та не призначені для узагальнення даних щодо великої кількості підприємств. ERP-системи, у свою чергу, ефективно вирішують завдання внутрішнього управління ресурсами підприємства, але також не орієнтовані на міжпідприємницький або регіональний рівень аналізу.

Таким чином, узагальнення показує, що жоден із розглянутих класів інформаційних систем не забезпечує повного циклу обробки даних на рівні адміністративного району — від введення первинної інформації, її перевірки та збереження до аналітичної обробки й формування управлінської звітності. Саме тому виникає потреба у створенні спеціалізованої інформаційної системи підтримки управління виробничою діяльністю сільськогосподарських підприємств, яка повинна забезпечувати централізоване зберігання даних, підтримку введення та верифікації первинної інформації, автоматизовану аналітичну обробку та формування звітності для різних рівнів управління. Реалізація такої системи дозволить інтегрувати дані різних підприємств у єдиному інформаційному середовищі та підвищити ефективність і обґрунтованість управлінських рішень на регіональному рівні.

1.3 Обґрунтування вибору інформаційних технологій та архітектури системи

Ефективність функціонування інформаційної системи підтримки управління виробничою діяльністю сільськогосподарських підприємств регіону визначається якістю обраних архітектурних рішень, рівнем технологічної інтеграції та здатністю системи забезпечувати повний цикл роботи з даними від

їх первинного введення до аналітичної обробки та формування управлінських висновків. З урахуванням специфіки предметної області, що передбачає роботу з великими масивами статистичної інформації аграрних підприємств на рівні Кам'янського району Дніпропетровської області, ключовими вимогами до системи є централізація даних, багатокористувацький доступ, узгодженість інформаційних потоків та можливість оперативного аналізу показників виробничої діяльності.

У процесі дослідження було проведено порівняння двох основних архітектурних підходів до реалізації подібних систем: локального (настільного) та веб-орієнтованого (клієнт-серверного). Локальні інформаційні системи, що базуються на використанні настільних систем керування базами даних або табличних процесорів, історично застосовуються для автоматизації облікових задач, однак у контексті регіонального управління демонструють суттєві обмеження. Насамперед це відсутність єдиного інформаційного середовища, що унеможлиблює одночасну роботу великої кількості користувачів на різних організаційних рівнях. У таких умовах виникають ризики дублювання інформації, втрати актуальних змін та порушення цілісності даних під час паралельного редагування.

Додатковим недоліком локальних рішень є низький рівень масштабованості. Будь-які зміни у структурі даних, алгоритмах розрахунку або формах звітності потребують ручного оновлення програмного забезпечення на кожному робочому місці користувача, що значно ускладнює супровід системи та підвищує адміністративні витрати. Також обмежені можливості інтеграції з сучасними веб- та аналітичними сервісами не дозволяють реалізувати повноцінну систему підтримки прийняття управлінських рішень на регіональному рівні.

Перехід до веб-орієнтованої клієнт-серверної архітектури є більш доцільним з точки зору сучасних вимог до інформаційних систем у сфері публічного управління та аграрної аналітики. Основною концептуальною перевагою такого підходу є централізація обробки та зберігання даних на

серверному рівні, що забезпечує єдину точку істини для всієї інформаційної системи. Це дозволяє гарантувати узгодженість даних, підвищити рівень їх захисту та забезпечити контроль доступу відповідно до ролей користувачів.

Важливою характеристикою веб-архітектури є можливість одночасної роботи багатьох користувачів у системі, що дозволяє одночасну роботу аналітиків, адміністраторів та представників аграрних підприємств без втрати продуктивності системи. Крім того, використання веб-технологій забезпечує природну кросплатформність, оскільки доступ до системи здійснюється через стандартний браузер без встановлення додаткового програмного забезпечення. Це особливо важливо для регіональних структур, де технічне забезпечення робочих місць може суттєво відрізнятися.

Окремо слід відзначити спрощення життєвого циклу системи під час використання клієнт-серверної архітектури. Усі оновлення, модифікації бізнес-логіки та вдосконалення функціоналу виконуються централізовано на сервері, що виключає необхідність ручного оновлення клієнтських компонентів і мінімізує ризики несумісності версій програмного забезпечення.

Для реалізації інформаційної системи обрано сучасний стек веб-технологій, який забезпечує баланс між продуктивністю, гнучкістю та простотою супроводу. Клієнтська частина системи реалізується з використанням HTML5 та CSS3, що дозволяє створити семантично структурований, адаптивний і інтуїтивно зрозумілий інтерфейс користувача. Такий підхід забезпечує коректне відображення інформації на різних типах пристроїв і підтримує принципи доступності та ергономічності.

Інтерактивна складова реалізується за допомогою JavaScript, який забезпечує динамічну взаємодію користувача з системою без необхідності перезавантажувати сторінки. Це дозволяє реалізувати функціонал швидкої фільтрації даних, пошуку підприємств, оновлення аналітичних показників у реальному часі та асинхронної взаємодії з серверною частиною. Використання JavaScript також дозволяє підвищити швидкість роботи системи та зменшити навантаження на сервер за рахунок часткової обробки логіки на стороні клієнта.

Для реалізації аналітичного блоку та візуалізації даних використовується бібліотека Chart.js, яка забезпечує побудову інтерактивних графіків та діаграм на основі технології HTML5 Canvas. Це дозволяє реалізувати наочне подання ключових показників діяльності аграрних підприємств, включаючи виробничі та економічні характеристики, що є критично важливим для підтримки управлінських рішень на регіональному рівні. Використання графічної аналітики підвищує інтерпретованість даних і сприяє швидкому виявленню тенденцій та аномалій у виробничих процесах.

Серверна частина системи базується на реляційній системі керування базами даних MySQL, яка забезпечує надійне збереження структурованих даних, підтримку складних SQL-запитів, індексацію та транзакційну обробку даних. Це дозволяє ефективно працювати з великими обсягами статистичних даних, забезпечуючи їхню цілісність, узгодженість і швидкий доступ.

На рис. 1.1 представлено узагальнену архітектуру інформаційної системи підтримки управління виробничою діяльністю сільськогосподарських підприємств регіону, яка реалізована за клієнт-серверним принципом і побудована на основі трирівневої моделі обробки даних.

Архітектура системи включає три основні рівні: рівень представлення (Frontend), рівень прикладної логіки (Backend / Application Layer) та рівень зберігання даних (Database Layer), що забезпечує структурований розподіл функцій та підвищує масштабованість і супровідність програмного рішення.

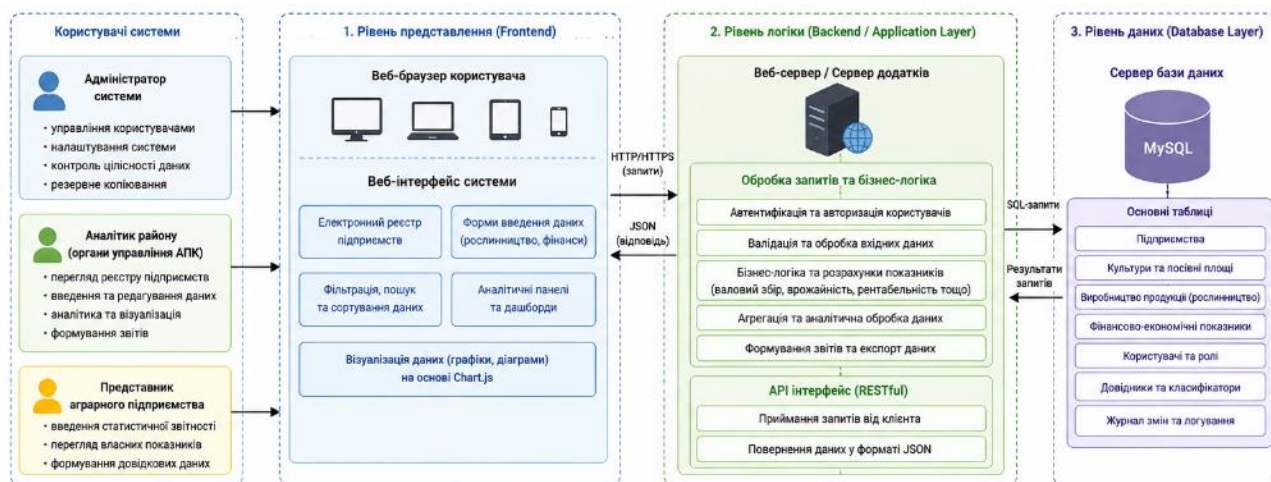


Рис. 1.1. Архітектура інформаційної системи підтримки управління

виробничою діяльністю сільськогосподарських підприємств регіону

Рівень представлення (Frontend) реалізує взаємодію користувача із системою через веб-браузер. На даному рівні забезпечується доступ до функціональних модулів системи, таких як електронний реєстр аграрних підприємств, форми введення статистичних та фінансових даних, інструменти фільтрації та пошуку, а також аналітичні панелі. Також реалізовано засоби візуалізації даних у вигляді графіків і діаграм, що формуються за допомогою бібліотеки Chart.js. Користувачами цього рівня є адміністратор системи, аналітики та представники аграрних підприємств.

Рівень прикладної логіки (Backend / Application Layer) виконує ключові функції обробки даних і реалізації бізнес-логіки системи. На цьому рівні здійснюється перевірка та валідація вхідних даних, обробка запитів користувачів, виконання аналітичних розрахунків, агрегація статистичних показників і формування вихідної інформації. Взаємодія між клієнтською частиною та сервером реалізується через REST API з використанням HTTP-запитів і формату обміну даними JSON. Цей рівень також забезпечує реалізацію алгоритмів розрахунку ключових показників ефективності виробництва.

Рівень зберігання даних (Database Layer) представлений реляційною базою даних MySQL, яка забезпечує централізоване збереження структурованої інформації про діяльність сільськогосподарських підприємств. У базі даних зберігаються основні сутності системи, зокрема довідники підприємств, дані щодо виробництва продукції рослинництва, фінансово-економічні показники, а також допоміжні класифікатори та службові журнали. Використання реляційної моделі дозволяє забезпечити цілісність даних, підтримку зв'язків між таблицями та ефективну обробку SQL-запитів.

Передача даних у системі здійснюється за принципом послідовної взаємодії: користувач вводить або запитує інформацію через веб-інтерфейс, запит передається на сервер, де відбувається його обробка та взаємодія з базою даних, після чого результат повертається користувачу у форматі JSON та відображається у вигляді таблиць або графічних елементів.

Запропонована архітектура забезпечує високу гнучкість, масштабованість та зручність супроводу системи, а також відповідає сучасним вимогам до веб-орієнтованих інформаційних систем у сфері аграрного управління та аналітики. Вона дозволяє ефективно інтегрувати різні джерела даних, забезпечувати їх актуальність та підтримувати прийняття управлінських рішень на регіональному рівні.

Таким чином, запропонована комбінація клієнт-серверної архітектури, реляційної СУБД MySQL, мови програмування JavaScript та сучасних засобів веб-візуалізації формує технологічно обґрунтовану основу для створення масштабованої, гнучкої та функціонально повної інформаційної системи. Реалізований підхід повністю відповідає вимогам автоматизації процесів управління аграрним сектором на регіональному рівні та забезпечує підвищення якості прийняття управлінських рішень завдяки оперативному доступу до аналітичної інформації.

Виходячи з визначених завдань предметної області та можливостей обраних технологій, сформульовано вимоги до функціональних характеристик системи. Система повинна забезпечувати виконання таких основних функцій:

- відображення інтерактивного електронного реєстру сільськогосподарських підприємств Кам'янського району (аналог таблиці-довідника DPIDPR);
- внесення та редагування звітних даних щодо рослинництва та фінансово-економічних показників;
- можливість миттєвої фільтрації підприємств за організаційною формою (ТОВ, СФГ) на стороні клієнта без перезавантаження сторінки;
- автоматичний розрахунок сумарних показників (валового збору, врожайності, рівня рентабельності) по всьому аналізованому регіону;
- графічна візуалізація аналітичних даних у вигляді динамічних стовпчикових та кругових діаграм для підтримки прийняття управлінських рішень.

Для формалізації функціональних вимог та визначення взаємодії

користувачів із системою побудовано UML-діаграму прецедентів (рис. 1.2.). Діаграма відображає основних акторів системи (адміністратора, оператора підприємства та аналітика) і перелік прецедентів, що реалізують заявлені функціональні можливості: перегляд електронного реєстру підприємств, фільтрацію за організаційною формою, внесення та редагування звітних даних, автоматичний розрахунок узагальнених показників регіону та візуалізацію аналітичної інформації у вигляді діаграм.

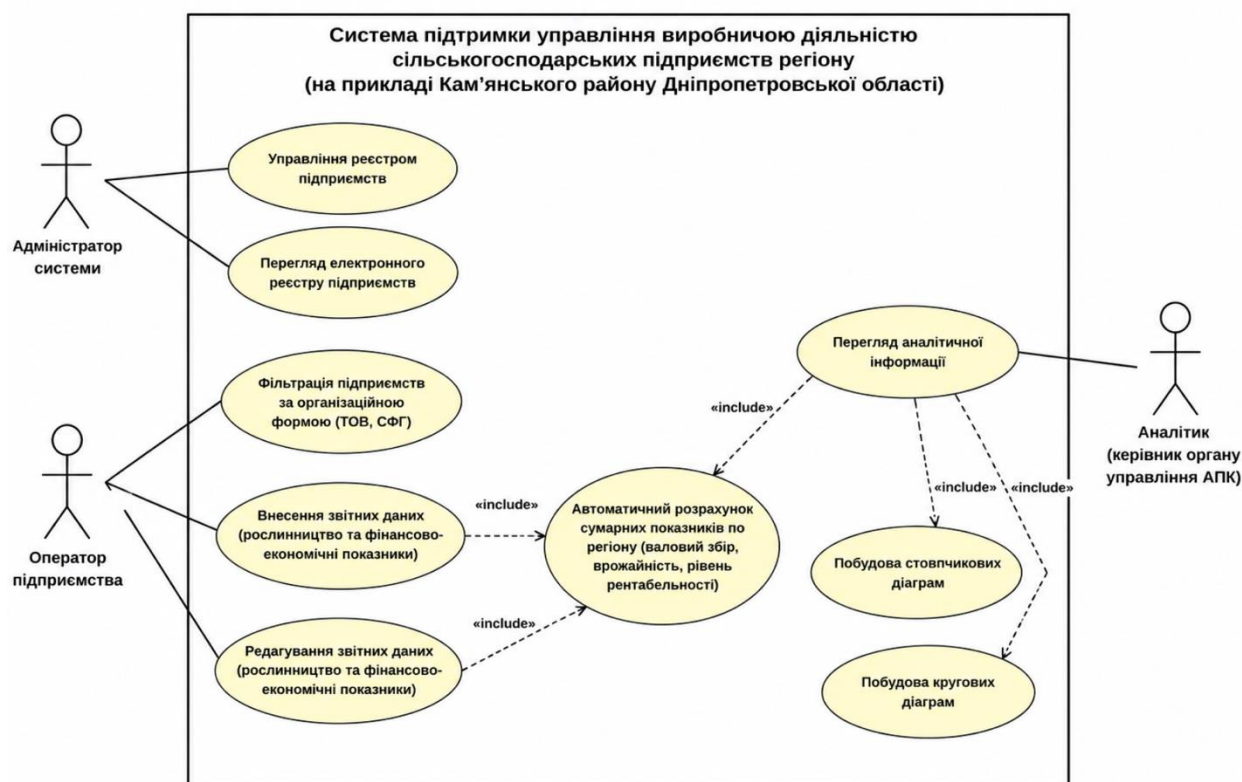


Рис. 1.2. Діаграма прецедентів системи підтримки управління виробничою діяльністю сільськогосподарських підприємств регіону.

Адміністратор системи відповідає за підтримання актуальності електронного реєстру сільськогосподарських підприємств. Він має доступ до функцій перегляду реєстру та управління записами підприємств, що забезпечує коректність і повноту даних, які використовуються іншими користувачами системи.

Оператор підприємства здійснює роботу з первинними даними. Йому доступні функції перегляду електронного реєстру підприємств, фільтрації

підприємств за організаційною формою (товариство з обмеженою відповідальністю або селянське фермерське господарство), а також внесення та редагування звітних даних щодо виробничої діяльності підприємств. Під час додавання або коригування інформації система автоматично ініціює перерахунок узагальнених показників регіону.

Аналітик використовує систему для аналізу зведеної інформації та підтримки процесу прийняття управлінських рішень. Йому доступні функції перегляду електронного реєстру, застосування фільтрів та перегляду аналітичної інформації. Для формування аналітичних матеріалів система автоматично виконує розрахунок сумарних показників за регіоном, зокрема валового збору, врожайності та рівня рентабельності.

Прецедент «Перегляд аналітичної інформації» включає функції побудови стовпчикових і кругових діаграм, що забезпечують наочне представлення результатів аналізу. Використання зв'язків типу <<include>> свідчить про те, що побудова аналітичних звітів неможлива без попереднього автоматичного розрахунку узагальнених показників, а також що процес внесення або редагування звітних даних супроводжується автоматичним оновленням розрахункових значень.

Таким чином, діаграма прецедентів демонструє основні сценарії використання системи, визначає ролі користувачів та відображає взаємозв'язки між ключовими функціями, необхідними для забезпечення ефективного управління виробничою діяльністю сільськогосподарських підприємств Кам'янського району Дніпропетровської області.

Для більш детального відображення логіки виконання ключових функцій системи побудовано діаграму діяльності, яка описує послідовність дій під час внесення звітних даних та автоматичного розрахунку сумарних показників за регіоном. Діаграма відображає взаємодію користувача із системою, процес перевірки та збереження даних, а також механізм автоматичного оновлення аналітичних показників після внесення змін. Діаграма діяльності для прецедентів «Внесення звітних даних» та «Автоматичний розрахунок сумарних показників

по регіону» наведена на рис. 1.3.

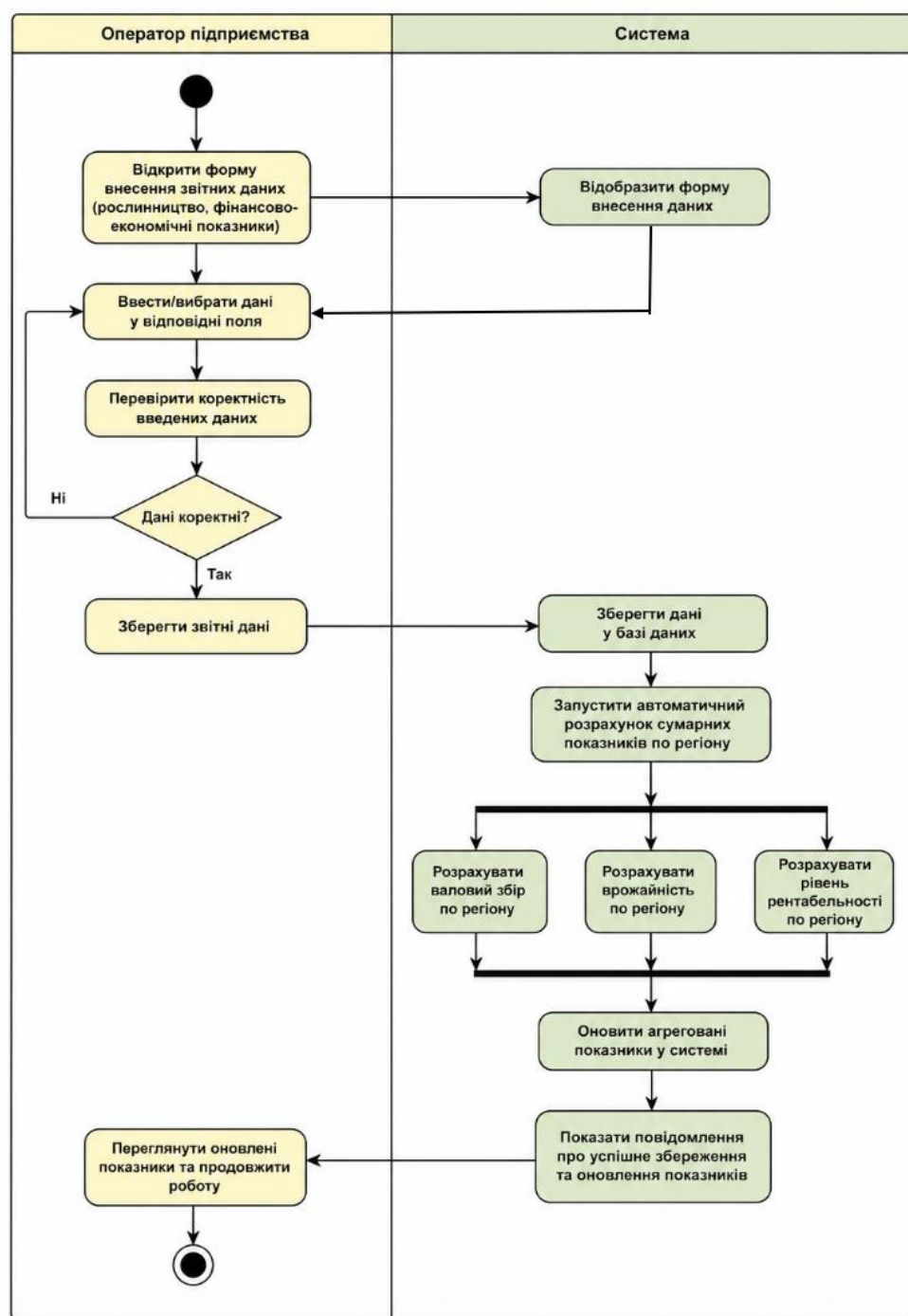


Рис. 1.3. Діаграма діяльності для прецедентів «Внесення звітних даних» та «Автоматичний розрахунок сумарних показників по регіону».

Представлена діяльність розпочинається з відкриття оператором форми для внесення звітних даних щодо виробничих і фінансово-економічних показників підприємства. У відповідь система відображає форму введення даних, після чого користувач заповнює необхідні поля відповідними значеннями.

Наступним етапом є перевірка коректності введених даних. На діаграмі цей процес представлено вузлом прийняття рішення. Якщо виявлено помилки або неповноту інформації, користувач повертається до етапу введення даних для їх виправлення. У разі успішного проходження перевірки оператор підтверджує збереження звітної інформації.

Після збереження даних у базі система автоматично запускає процедуру розрахунку агрегованих показників по всьому регіону. На цьому етапі виконуються паралельні обчислення основних аналітичних характеристик: валового збору сільськогосподарської продукції, середньої врожайності та рівня рентабельності діяльності підприємств. Використання паралельних гілок на діаграмі відображає можливість незалежного розрахунку цих показників.

Після завершення всіх розрахунків результати об'єднуються та використовуються для оновлення узагальнених показників у системі. Далі користувач отримує повідомлення про успішне збереження даних та актуалізацію аналітичної інформації. Завершальним етапом є перегляд оновлених показників, після чого оператор може продовжити роботу із системою або завершити сеанс.

Таким чином, діаграма діяльності демонструє повний цикл обробки звітної інформації від введення даних користувачем до автоматичного формування оновлених аналітичних показників регіону. Це забезпечує оперативність обробки інформації, мінімізує ризик виникнення помилок та створює основу для прийняття обґрунтованих управлінських рішень у сфері сільськогосподарського виробництва.

Отже, у результаті проведеного аналізу обґрунтовано вибір веб-орієнтованої клієнт-серверної архітектури як найбільш доцільної для реалізації інформаційної системи підтримки управління виробничою діяльністю сільськогосподарських підприємств регіону. Запропонована трирівнева архітектура забезпечує централізоване зберігання даних, багатокористувацький доступ, масштабованість і зручність супроводу системи. Визначено основні функціональні можливості програмного рішення, ролі користувачів і сценарії

їхньої взаємодії із системою, що формалізовано за допомогою UML-діаграми прецедентів та діаграми діяльності. Розроблені моделі відображають логіку обробки даних, механізми автоматичного розрахунку аналітичних показників і процес формування інформації для підтримки управлінських рішень. Отримані результати є основою для подальшого проектування структури бази даних і реалізації програмних компонентів інформаційної системи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ УПРАВЛІННЯ ВИРОБНИЧОЮ ДІЯЛЬНІСТЮ АГРАРНИХ ПІДПРИЄМСТВ РЕГІОНУ

2.1 Проєктування та реалізація реляційної бази даних для інформаційної системи підтримки управління аграрним виробництвом

Побудова веб-орієнтованої інформаційної системи підтримки управління агропромисловим комплексом (АПК) Кам'янського району Дніпропетровської області базується на проєктуванні оптимальної архітектури реляційної бази даних. База даних є централізованим сховищем інформації та функціонує під керуванням сучасної системи керування базами даних (СУБД) MySQL. Процес розробки інформаційного забезпечення складався з двох взаємопов'язаних етапів: інфологічного (концептуального) та даталогічного (фізичного) проєктування. На рис. 2.1 представлено ER-діаграму бази даних системи підтримки управління виробничою діяльністю СГП.

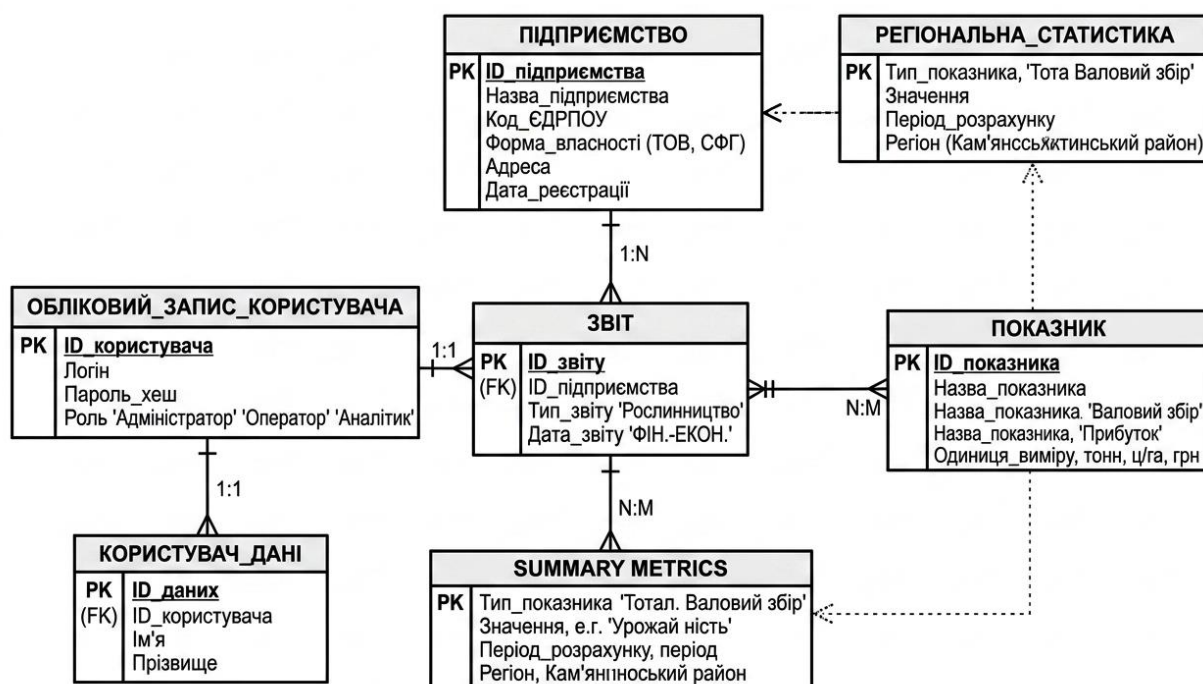


Рис. 2.1. ER-діаграма бази даних системи підтримки управління виробничою діяльністю СГП

На етапі інфологічного проектування було проведено системний аналіз сутностей предметної області на основі структури та показників форм державної статистичної звітності («Звіт про площі та валові збори сільськогосподарських культур») та («Основні економічні показники роботи сільськогосподарських підприємств»). Головною метою цього етапу було створення абстрактної моделі даних, яка б адекватно відображала бізнес-процеси моніторингу виробничої діяльності, але не залежала б від конкретної програмної реалізації СУБД.

Для побудови логічно узгодженої моделі предметної області інформаційної системи було визначено три основні сутності, що відображають ключові аспекти діяльності сільськогосподарських підприємств регіону.

Сутність DPIDPR (довідник підприємств району) виконує роль базового довідкового компонента системи та забезпечує унікальну ідентифікацію кожного аграрного підприємства. Вона містить основні реєстраційні та класифікаційні дані про суб'єкти господарювання, зокрема їх назву, ідентифікаційні коди та організаційно-правову форму. Використання цієї сутності дозволяє уникнути дублювання інформації та забезпечити цілісність довідкових даних у межах бази даних.

Сутність T29SG (показники виробництва рослинництва) призначена для зберігання статистичних даних про результати сільськогосподарського виробництва у рослинницькій галузі. Вона акумулює інформацію за роками у розрізі окремих культур, включаючи посівні площі, обсяги зібраного врожаю та розрахункові показники ефективності виробництва. Дана сутність відображає динаміку виробничих процесів підприємств у часі та є основою для подальшого аналітичного опрацювання.

Сутність T50SG (фінансово-економічні показники) використовується для структурованого збереження фінансових результатів діяльності підприємств. Вона містить дані щодо виробничих витрат, обсягів реалізації продукції, отриманого доходу та узагальненого фінансового результату, що дає змогу оцінювати економічну ефективність господарської діяльності в регіональному розрізі.

Під час проєктування бази даних було виконано нормалізацію структури даних до третьої нормальної форми, що дозволило мінімізувати надлишковість інформації та усунути потенційні аномалії під час вставки, оновлення й видалення даних. Це забезпечило підвищення цілісності даних і спростило подальший супровід інформаційної системи.

Між суттю DPIDPR та аналітичними таблицями T29SG і T50SG встановлено зв'язки типу «один до багатьох». Це означає, що кожне підприємство може бути пов'язане з множиною записів у таблицях виробничих та фінансово-економічних показників, які формуються за різні звітні періоди або за види продукції. У свою чергу, кожен запис аналітичних таблиць однозначно належить до одного конкретного підприємства, що забезпечує коректність структурної організації даних.

На етапі даталогічного проєктування концептуальна ER-схема була трансформована у фізичну структуру таблиць реляційної СУБД MySQL. На цьому рівні для кожного атрибуту було жорстко визначено типи даних, встановлено обмеження на порожні значення (NOT NULL), виділено сурогатні первинні ключі (PRIMARY KEY) із властивістю автоматичного нарощування (AUTO_INCREMENT) та налаштовано обмеження зовнішніх ключів (FOREIGN KEY).

Нижче наведено розгорнуту низькорівневу специфікацію та опис фізичної структури розроблених таблиць бази даних:

1. Таблиця DPIDPR (Загальносистемний довідник сільськогосподарських підприємств):

о id_pred (Тип: INT, специфікація: PRIMARY KEY, AUTO_INCREMENT) – внутрішній системний унікальний код підприємства, що використовується для зв'язку з іншими таблицями.

о kod_edrpou (Тип: VARCHAR(8), специфікація: UNIQUE, NOT NULL) – офіційний восьмизначний ідентифікаційний код юридичної особи в Єдиному державному реєстрі. Поле має обмеження унікальності для запобігання дублювання компаній.

- о name_pred (Тип: VARCHAR(150), специфікація: NOT NULL) – повне найменування агропідприємства.

- о ownership_form (Тип: VARCHAR(50), специфікація: NOT NULL) – організаційно-правова форма господарювання (наприклад: «ТОВ», «СФГ»). Використовується як ключовий параметр для фільтрації клієнтів.

- о address (Тип: VARCHAR(255)) – юридична та фактична адреса реєстрації суб'єкта в межах Кам'янського району.

2. Таблиця T29SG (Статистична звітність - Рослинництво):

- о id_record_29 (Тип: INT, специфікація: PRIMARY KEY, AUTO_INCREMENT) – унікальний номер запису у системі.

- о id_pred (Тип: INT, специфікація: FOREIGN KEY REFERENCES DPIDPR(id_pred)) – зовнішній ключ, який логічно прив'язує показники врожаю до конкретної фірми.

- о year_report (Тип: INT, специфікація: NOT NULL) – календарний рік, за який подано звіт.

- о culture_name (Тип: VARCHAR(100), специфікація: NOT NULL) – назва сільськогосподарської культури (пшениця, соняшник, кукурудза).

- о area_gektar (Тип: DECIMAL(10,2), специфікація: NOT NULL) – загальна посівна (або зібрана) площа. Використання точного типу DECIMAL замість FLOAT гарантує відсутність помилок округлення під час математичних розрахунків сумарних площ.

- о val_zbir_centner (Тип: DECIMAL(12,2), специфікація: NOT NULL) – обсяг валового збору культури у центнерах.

3. Таблиця T50SG (Економічні результати):

- о id_record_50 (Тип: INT, specification: PRIMARY KEY, AUTO_INCREMENT) – унікальний ідентифікатор фінансового запису.

- о id_pred (Тип: INT, специфікація: FOREIGN KEY REFERENCES DPIDPR(id_pred)) – зовнішній ключ для зв'язку з довідником компаній.

- о year_report (Тип: INT, специфікація: NOT NULL) – звітний фінансово-економічний рік.

- о production_cost (Тип: DECIMAL(15,2)) – повна собівартість реалізованої аграрної продукції (тис. грн).
- о revenue (Тип: DECIMAL(15,2)) – чистий дохід (виручка) від реалізації продукції (тис. грн).
- о profit_loss (Тип: DECIMAL(15,2)) – чистий прибуток або збиток підприємства (тис. грн).
- о rentability_percent (Тип: DECIMAL(5,2)) – відсотковий рівень рентабельності, що вираховується на основі відношення прибутку до собівартості.

Для забезпечення жорсткої посилальної цілісності даних (Referential Integrity) на рівні фізичної схеми СУБД MySQL, для всіх зовнішніх ключів було активовано правила каскадного контролю: **ON DELETE RESTRICT** (забороняє видаляти підприємство з довідника DPIDPR, якщо в базі даних присутні його звіти за формою 29-сг або 50-сг, щоб уникнути появи «втрачених» даних) та **ON UPDATE CASCADE** (автоматично оновлює ідентифікатори у підпорядкованих таблицях при зміні id_pred у головній).

Окрему увагу в процесі фізичного проєктування було приділено оптимізації швидкодії веб-додатка при виконанні аналітичних SQL-запитів. Оскільки користувач системи (аналітик) постійно здійснює сортування, пошук та групування підприємств Кам'янського району за їх типами власності, на рівні СУБД MySQL було згенеровано вторинні B-tree індекси (INDEX) для полів id_pred та ownership_form. Наявність цих індексів знижує часову складність пошукових операцій на сервері з лінійної до логарифмічної, що забезпечує миттєвий відгук веб-інтерфейсу системи навіть у разі значного масштабування бази даних.

2.2 Проєктування клієнт-серверної архітектури веб-середовища

Для реалізації інформаційної системи підтримки управління агропромисловим комплексом Кам'янського району Дніпропетровської області обрано клієнт-серверну архітектуру, що базується на трирівневій моделі

побудови веб-додатків. Такий підхід дозволяє забезпечити централізоване зберігання даних, багатокористувацький доступ та масштабованість системи при обробці значних обсягів аграрної статистичної інформації.

Використання веб-орієнтованої архітектури дозволяє подолати обмеження традиційних локальних систем, які пов'язані з відсутністю віддаленого доступу, складністю синхронізації даних та низькою масштабованістю при роботі з багатьма користувачами.

Запропонована архітектура включає три логічно розділені рівні:

Рівень представлення (Presentation Layer) – реалізований за допомогою HTML5, CSS3 та JavaScript і відповідає за користувацький інтерфейс, відображення реєстрів підприємств та інтерактивну візуалізацію аналітичних даних;

Рівень бізнес-логіки (Application Layer) – забезпечує обробку запитів користувача, виконання розрахунків, фільтрацію та агрегацію даних, а також взаємодію між клієнтською частиною та базою даних через серверні механізми;

Рівень збереження даних (Data Layer) – реалізований на основі реляційної СУБД MySQL, яка забезпечує централізоване зберігання структурованої інформації про діяльність аграрних підприємств.

Взаємодія між рівнями здійснюється за допомогою HTTP/HTTPS-запитів та API-інтерфейсів, що забезпечує стандартизований обмін даними та незалежність компонентів системи. Така структура зменшує зв'язність між модулями (low coupling) та підвищує їх внутрішню узгодженість (high cohesion), що позитивно впливає на супровід і масштабування системи.

На рис. 2.2 наведено схему трирівневої клієнт-серверної архітектури веб-орієнтованої інформаційної системи, яка відображає взаємодію між користувачем, серверною частиною та базою даних.

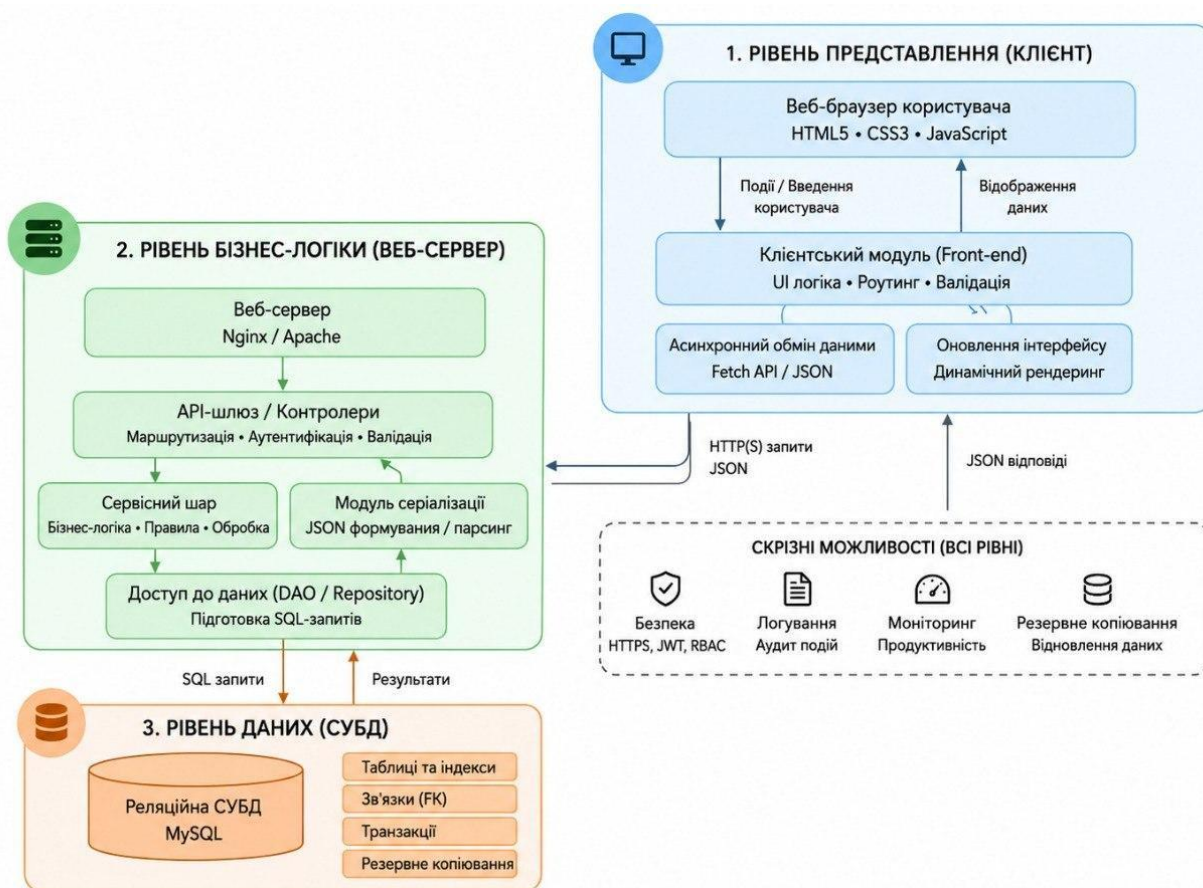


Рис. 2.2. Схема трирівневої клієнт-серверної архітектури веб-орієнтованої інформаційної системи

У межах розроблюваної інформаційної системи кожен рівень клієнт-серверної архітектури виконує чітко визначені функції, що забезпечують розподіл відповідальності між компонентами системи та підвищують її структурованість і масштабованість.

1. Рівень представлення (Presentation Tier / Frontend / Клієнтська частина)

Рівень представлення є верхнім шаром архітектури веб-середовища і виступає безпосереднім людино-машинним інтерфейсом, який забезпечує візуалізацію даних та навігацію користувача (аналітика Районної державної адміністрації чи керівника сільськогосподарського підприємства) у просторі функціональних можливостей системи. Цей рівень функціонує виключно всередині середовища веб-браузера на кінцевому робочому місці користувача і є повністю кросплатформним, оскільки не потребує встановлення стороннього залежного програмного забезпечення.

Для проектування та програмної реалізації Frontend-складової було обрано сучасний та ефективний стек технологій:

- HTML5 (HyperText Markup Language): використовується для опису логічної, семантичної та ієрархічної структури веб-сторінок, розмітки табличних реєстрів, інтерактивних форм введення і коригування показників звітності № 29-сг та № 50-сг, а також елементів керування сесією.
- CSS3 (Cascading Style Sheets): забезпечує каскадне стилювання, побудову візуальної ієрархії компонентів інтерфейсу, налаштування кольірних схем та реалізацію принципів адаптивної верстки (*Responsive Web Design*). Це гарантує стабільне та коректне відображення робочих аналітичних панелей системи (дашбордів) на пристроях з будь-якою роздільною здатністю екрана від широкоформатних моніторів стаціонарних ПК до екранів мобільних планшетів чи ноутбуків.
- JavaScript (стандарт ECMAScript 6): є базовим інструментом реалізації динамічної обробки подій та бізнес-логіки на стороні клієнта. Програмні модулі на базі JavaScript перехоплюють події користувацького інтерфейсу, здійснюють первинну валідацію введених даних (перевірку на заповненість полів, відповідність типів даних), асинхронно взаємодіють із сервером додатків та безпосередньо керують роботою аналітичної бібліотеки Chart.js. Дана бібліотека відповідає за динамічний рендеринг інтерактивних стовпчикових та кругових діаграм врожайності й рентабельності суб'єктів господарювання в полі графічного елемента <canvas>.

2. Рівень бізнес-логіки (Application Tier / Backend / Серверна частина)

Рівень бізнес-логіки (сервер додатків) виступає інтелектуальним обчислювальним ядром системи та посередником, що координує і спрямовує потоки даних між користувацьким інтерфейсом (Frontend) та фізичним сховищем даних (СУБД MySQL). Він розгортається і функціонує на віддаленому сервері під керуванням високопродуктивного веб-сервера Apache або Nginx.

Основними функціональними завданнями Backend-рівня в межах системи агромоніторингу є:

- Перехоплення, дешифрація та маршрутизація вхідних HTTP/HTTPS запитів, які надходять від клієнтських браузерів за допомогою технології асинхронних запитів *AJAX* через прикладний інтерфейс *Fetch API*. Це дозволяє оновлювати вміст екрана без повного перезавантаження сторінки.
- Повне очищення, фільтрація та екранування вхідних параметрів для захисту системи від поширених вразливостей веб-додатків, зокрема від ін'єкцій шкідливого SQL-коду на рівні СУБД, що підвищує загальну кібербезпеку системи.
- Виконання розрахункових та аналітичних алгоритмів системи підтримки прийняття управлінських рішень (СППР). Серверна частина здійснює математичне агрегування загальних показників виробництва по Кам'янському району, калькуляцію середньої врожайності зернових чи технічних культур, розрахунок інтегральних індексів рентабельності та збитковості на основі первинних даних форми № 50-сг, а також підготовку аналітичних зрізів.
- Сerialізація та трансляція обчислених аналітичних результатів у текстовий кросплатформний формат JSON (JavaScript Object Notation), який є сучасним стандартом для передачі структурованих об'єктів та масивів у веб-середовищі завдяки своїй компактності та швидкості десerialізації.

3. Рівень даних (Data Tier / Сервер баз даних)

Нижній шар трирівневої архітектури представлений безпосередньо сервером реляційної системи керування базами даних MySQL. Цей рівень є повністю ізольованим від прямого доступу з боку зовнішніх клієнтських браузерів, що є фундаментальною вимогою безпеки: клієнт не знає реальної структури таблиць і не може надіслати прямий запит до бази даних.

Рівень даних відповідає за виконання таких критично важливих операцій:

- Надійне фізичне збереження, структурування та індексація інформаційних масивів у межах розроблених таблиць DPIDPR, T29SG та T50SG.
- Забезпечення сувого дотримання вимог транзакційності ACID (атомарність, узгодженість, ізольованість, довговічність) при виконанні будь-

яких операцій модифікації, додавання або видалення фінансово-виробничих показників агропідприємств. Це гарантує, що база даних завжди перебуватиме в узгодженому стані.

- Інтерпретація, оптимізація та безпосереднє виконання низькорівневих SQL-запитів на вибірку даних, що надходять від сервера додатків, з мінімальним часом відгуку за рахунок використання раніше спроектованих індексів.

Взаємодія між рівнями побудована на базі клієнт-серверного протоколу HTTP/HTTPS за подійною моделлю. Коли аналітик змінює критерій фільтрації на дашборді (наприклад, обирає перегляд лише підприємств форми власності «ТОВ»), Frontend-модуль на JavaScript генерує асинхронний запит `Fetch()`. Веб-сервер приймає цей запит, звертається до СКБД MySQL за допомогою SQL-інструкції `SELECT`, отримує результуючу вибірку, формує з неї об'єкт JSON і надсилає його назад клієнту. Браузер, отримавши JSON, парсить дані та оновлює лише цільову таблицю і графік `Chart.js`.

Впровадження такої трирівневої клієнт-серверної архітектури забезпечує розроблюваній системі низку стратегічних переваг порівняно з дволанковими або локальними рішеннями:

1. Висока масштабованість: за необхідності збільшення обчислювальної потужності, кожен рівень може бути масштабований незалежно (наприклад, перенесення бази даних MySQL на окремий виділений сервер без зміни коду Frontend та Backend).

2. Гнучкість супроводу та модернізації: будь-які зміни в дизайні, стилістиці чи інтерфейсних сценаріях на рівні представлення можуть здійснюватися розробником автономно і жодним чином не вимагатимуть зупинки сервера або реструктуризації фізичних таблиць бази даних.

3. Посилена безпека: рівень бізнес-логіки повністю приховує структуру бази даних від кінцевого користувача, що унеможливлює здійснення

прямих недеструктивних або зловмисних маніпуляцій зі сховищем аграрної статистики.

2.3. Проєктування людино-машинного інтерфейсу та інформаційного забезпечення користувача

Проєктування людино-машинного інтерфейсу (UI) та детальна розробка сценаріїв користувацького досвіду (UX) для веб-орієнтованої інформаційної системи агромоніторингу здійснювалися з суворим дотриманням сучасних стандартів ергономіки, інженерної психології, принципів інтуїтивності інтерфейсів (*User-Centered Design*) та концепції мінімізації когнітивного навантаження на оператора. Оскільки кінцевими користувачами розробленого програмного продукту є профільні фахівці управління агропромислового комплексу (АПК) Кам'янського району Дніпропетровської області, які виконують щоденні рутинні операції з консолідації звітності, веб-інтерфейс було спроектовано як єдине високоінтерактивне робоче середовище.

Для дослідження динаміки функціонування системи та моделювання логіки взаємодії компонентів під час виконання користувацьких запитів було розроблено модель поведінки. Процес асинхронного обміну даними за допомогою Fetch API між клієнтським браузером, обчислювальним сервером на базі PHP та СУБД MySQL під час фільтрації підприємств чи генерації аналітичних графіків Chart.js проілюстровано на рис. 2.3.

В основу UX-стратегії системи покладено класичну ергономічну концепцію «у три кліки». Цей принцип передбачає, що користувач повинен мати можливість отримати будь-який аналітичний зріз, сформулювати підсумковий звіт або переглянути техніко-економічні показники конкретного господарства за допомогою не більше ніж трьох послідовних інтерактивних дій. Для реалізації цієї концепції було відкинуто ідею використання класичних складних багатосторінкових меню з глибоким рівнем вкладеності. Замість цього інтерфейс системи спроектовано за принципом Single Page Dashboard (єдиної аналітичної

панелі), де вся критично важлива інформація та елементи керування сфокусовані в межах одного робочого екрана.

При компонованні візуальних блоків інтерфейсу було враховано природні психофізіологічні особливості сприйняття людиною текстової та графічної інформації на цифрових дисплеях, зокрема F-паттерн зчитування контенту. Згідно з цим паттерном, погляд користувача спочатку рухається горизонтально у верхній частині екрана, потім опускається трохи нижче і здійснює коротку горизонтальну траєкторію, і наостанок сканує ліву вертикальну сторону дисплея.

Керуючись цим правилом, було розроблено структурно-компоновальний макет-карту інтерфейсу, яка фіксує просторове розташування та логічні зв'язки між чотирма основними функціональними зонами веб-додатка:

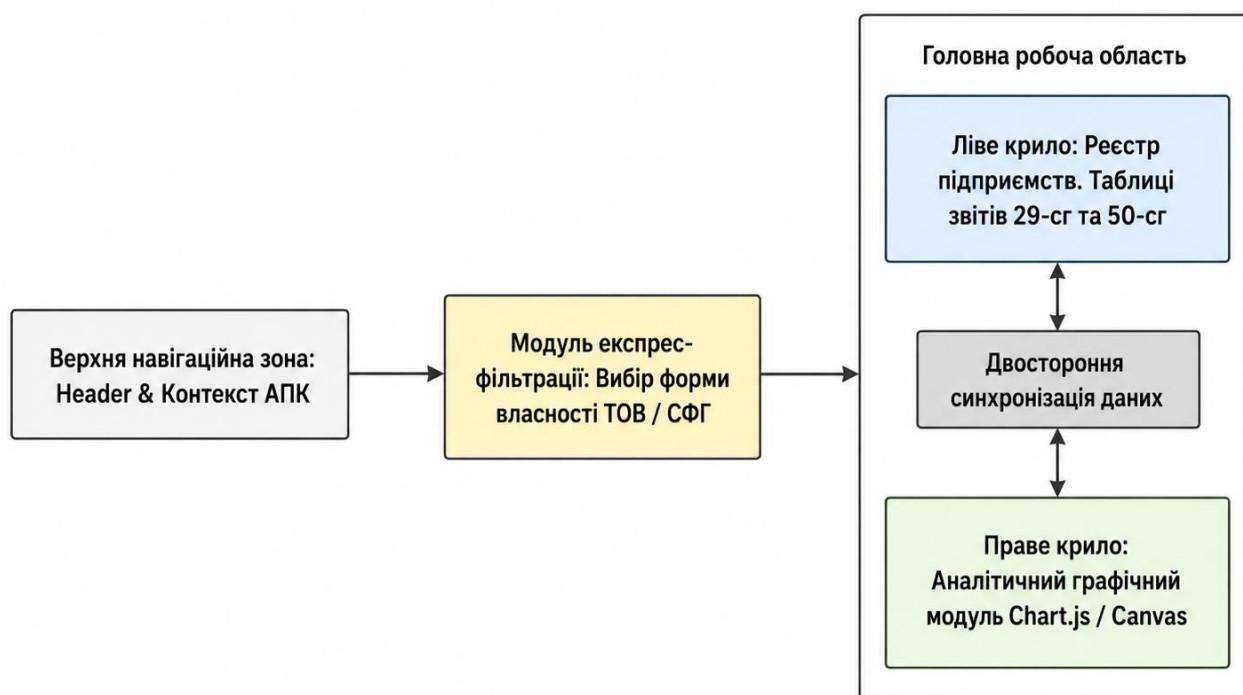


Рис. 2.3. Структурно-компоновальний макет людино-машинного інтерфейсу аналітичної панелі

Кожна з чотирьох спроектованих зон виконує унікальну роль у процесі людино-машинної взаємодії:

1. Верхня інформаційно-навігаційна зона (Header): Статичний блок у верхній частині сторінки, який виконує роль глобального ідентифікатора

системи. Він містить назву розробки та фіксує географічний контекст моніторингу («Кам'янський район Дніпропетровської області»). Це дозволяє оператору з першої секунди чітко розуміти поточне призначення веб-додатка.

2. Керуючий модуль експрес-фільтрації: Розміщений безпосередньо під заголовком сторінки, що відповідає першій фазі F-паттерну. Модуль містить набір інтерактивних елементів вибору (селекторів `<select>`), за допомогою яких аналітик може миттєво відфільтрувати весь масив підприємств району за організаційно-правовою формою господарювання (наприклад, відокремити товариства з обмеженою відповідальністю від селянських фермерських господарств). Також тут спроектовано тригерну кнопку швидкого скидання фільтрів для повернення до загального переліку об'єктів.

3. Робоча зона представлення реєстрів (Ліве крило дашборда): Займає ліву вертикальну частину екрана. Вона реалізована у вигляді динамічної інтерактивної HTML-таблиці, яка виводить ідентифікаційні коди ЄДРПОУ, офіційні назви суб'єктів АПК, а також агреговані показники валового збору культур і рівень рентабельності. Для покращення читабельності великих масивів цифрових даних спроектовано ефект ховерингу (`:hover`) динамічне підсвічування рядка таблиці при наведенні курсору миші, що мінімізує втому очей оператора.

4. Аналітичний графічний модуль (Праве крило дашборда): Розташований у правій частині робочої області й виступає інтерактивною системою підтримки прийняття рішень (СППР). Тут розміщено графічний елемент `<canvas>`, в якому клієнтська бібліотека Chart.js будує кольорові стовпчикові діаграми. Графіки відображають порівняльний аналіз врожайності та фінансової ефективності. При наведенні курсору на будь-який стовпчик діаграми спрацьовує сценарій відображення спливаючих підказок (*Tooltips*), які виводять точні числові значення показників, забезпечуючи високу наочність аналізу.

Окрему увагу під час проєктування людино-машинного інтерфейсу було приділено принципам адаптивності та кросбраузерності. Завдяки використанню гнучкої сітки CSS Media Queries, геометрія візуальних блоків автоматично

підлаштовується під параметри дисплея користувача. При роботі на широкоформатних моніторах таблиця та графік відображаються паралельно у дві колони, а при зменшенні роздільної здатності (наприклад, на екранах ноутбуків або планшетів) інтерфейс автоматично трансформується в одноколонну структуру, де графічні діаграми плавно зміщуються під табличний реєстр, зберігаючи 100% функціональності та зручності.

Взаємодія людини з інтерфейсом системи спроектована на базі подійної архітектури (*Event-Driven Architecture*) клієнтського JavaScript. Традиційні веб-системи вимагають повного перезавантаження сторінки при кожному запиті до бази даних, що призводить до втрати фокусу користувача, тривалого очікування відгуку та перевитрати серверних ресурсів. У спроектованій системі цей недолік повністю усунено завдяки використанню асинхронного обміну даними за технологією AJAX через Fetch API.

Сценарій людино-машинної взаємодії при реалізації аналітичного запиту складається з таких етапів:

- Користувач взаємодіє з елементом фільтрації, обираючи у випадальному списку параметр «ТОВ».
- Браузер реєструє подію зміни стану елемента onChange і миттєво передає керування відповідній функції-обробнику на JavaScript.
- JavaScript-сценарій без перезавантаження сторінки здійснює лінійну фільтрацію DOM-структури таблиці, приховуючи рядки тих підприємств, які не відповідають обраному критерію.
- Одночасно запускається внутрішній математичний модуль, який перераховує підсумкові суми (наприклад, загальний валовий збір зернових по Кам'янському району лише для відфільтрованих товариств).
- Наприкінці викликається метод оновлення об'єкта графіка Chart.js, який за допомогою внутрішніх анімаційних ефектів плавно змінює висоту стовпчиків діаграми відповідно до оновленого масиву чисел.

Такий підхід забезпечує оперативну реакцію інтерфейсу на дії користувача, що відповідає вимогам сучасних інтерактивних веб-додатків і

створює комфортні та безперервні умови роботи для аналітика під час обробки та візуалізації даних.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ЕКСПЛУАТАЦІЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ ПІДТРИМКИ УПРАВЛІННЯ АГРОВИРОБНИЦТВОМ

3.1 Структура програмного комплексу та обґрунтування вибору засобів розробки

На етапі фізичного втілення спроектованої інформаційної системи підтримки управління агропромисловим комплексом (АПК) Кам'янського району Дніпропетровської області було сформовано чітку модульну структуру програмного комплексу. Керуючись принципами трирівневої архітектури, розробленої та обґрунтованої у другому розділі, вихідний код системи було декомпоновано та розподілено за окремими файлами й каталогами. Такий підхід дозволив мінімізувати взаємозалежність окремих компонентів системи (забезпечити низький рівень *coupling*), максимізувати внутрішню згуртованість коду всередині кожного модуля (високий рівень *cohesion*), а також створити надійне підґрунтя для подальшого супроводу, рефакторингу та масштабування програмного забезпечення.

Логіко-функціональна структура програмного додатка на рівні файлової системи має модульний вигляд і складається з таких базових елементів:

- `index.html` – головний ініціалізуючий файл, який формує каркас рівня представлення (Frontend). Він містить семантичну HTML5-розмітку інтерфейсу користувача, структуру розробленого аналітичного дашборда, інтерактивні елементи фільтрації та блоки для динамічного виведення інформації. Також цей файл виконує роль координатора, підключаючи зовнішні каскадні таблиці стилів, графічні бібліотеки та скрипти бізнес-логіки.
- `style.css` – каскадна таблиця стилів, яка повністю відповідає за візуалізацію, колірне кодування, типографіку, мікроінтеракції (ефекти наведення) та просторове розташування елементів інтерфейсу. У цьому модулі реалізовано адаптивні медіа-запити (*CSS Media Queries*), що забезпечує кросплатформенність та стабільне відображення системи на дисплеях із різною роздільною здатністю.

- `app.js` – програмне ядро клієнтської бізнес-логіки, написане мовою JavaScript. Цей модуль безпосередньо реалізує модель взаємодії, перехоплює дії користувача, виконує асинхронні запити для ініціалізації вибірок із бази даних, реалізує алгоритми фільтрації та обчислює підсумкові техніко-економічні показники системи підтримки прийняття рішень (СППР). Крім того, він здійснює оперативне керування графічним об'єктом візуалізації `Chart.js`.

3.1.1 Обґрунтування вибору технологічного стеку Frontend-складової

Під час вибору інструментарію для програмної реалізації інтерфейсної частини інформаційної системи аналізувався широкий спектр сучасних веб-технологій. Основний акцент було зроблено на забезпеченні максимальної швидкодії, кросплатформенності, відсутності потреби у встановленні сторонніх плагінів на робочих місцях користувачів та мінімальному часі відгуку системи.

Для розробки клієнтської частини було обрано нативний технологічний стек HTML5 / CSS3 / JavaScript (стандарт ECMAScript 6+). Обґрунтування цього вибору базується на таких критеріях:

1. Нативна підтримка та кросплатформенність: Усі компоненти цього стеку інтерпретуються безпосередньо веб-браузером користувача (Google Chrome, Mozilla Firefox, Microsoft Edge тощо) незалежно від встановленої операційної системи (Windows, Linux, macOS). Це повністю усуває проблему сумісності програмного забезпечення, яка є критичною для державних установ та підрозділів РДА.

2. Висока обчислювальна ефективність мови JavaScript: Використання стандарту ES6+ дозволило застосувати сучасні конструкції мови (стрілкові функції, деструктуризацію, класи та модулі), що підвищило читабельність коду та оптимізувало роботу з пам'яттю. Завдяки високій швидкості сучасних двигунів інтерпретації (таких як V8), обробка великих масивів аграрної статистики, їх фільтрація та сортування відбуваються безпосередньо на стороні клієнта за лічені мілісекунди.

3. Асинхронний обмін даними (Fetch API): Вибір JavaScript зумовлений вбудованим інструментарієм для побудови асинхронних клієнт-серверних систем. Інтерфейс Fetch API дозволяє надсилати мережеві запити на сервер у фоновому режимі та отримувати структуровані відповіді у форматі JSON. Це виключає потребу у повному перезавантаженні веб-сторінки, зберігає стан інтерфейсу та забезпечує безперервний користувацький досвід (UX).

3.1.2. Обґрунтування вибору графічного двигуна візуалізації

Одним із ключових завдань розроблюваної ІС є наочне представлення результатів фінансово-виробничої діяльності підприємств Кам'янського району для прийняття управлінських рішень. Для реалізації графічного модуля аналізувалися такі рішення, як D3.js, Highcharts та Chart.js.

У результаті порівняльного аналізу було обрано відкриту JavaScript-бібліотеку Chart.js. Вибір обґрунтовано такими перевагами:

- Висока продуктивність рендерингу: Бібліотека використовує елемент стандарту HTML5 <canvas>, малюючи графіки на низькому рівні за допомогою апаратного прискорення GPU. Це забезпечує плавні анімації перемальовування діаграм під час зміни фільтрів, на відміну від важких SVG-рішень (наприклад, D3.js), які перевантажують дерево DOM великою кількістю елементів.
- Вбудована адаптивність: Chart.js має інтегровані механізми автоматичного масштабування (*responsive: true*). При зміні розмірів вікна браузера або перегляді дашборда на планшеті, графічні діаграми самостійно перераховують свої геометричні параметри, зберігаючи пропорції та чіткість ліній.
- Інтерактивність «із коробки»: Бібліотека містить готові алгоритми обробки подій миші, реалізуючи плавні спливаючі підказки (*tooltips*) та динамічне підсвічування секторів діаграми при наведенні курсору, що суттєво підвищує ергономічність людино-машинного інтерфейсу.

3.1.3 Обґрунтування вибору формату обміну даними

Для організації інформаційної взаємодії між рівнем бізнес-логіки та рівнем представлення було обрано текстовий формат JSON (JavaScript Object Notation). На відміну від громіздкого та складного для парсингу формату XML, JSON має мінімальну надлишковість (не містить важких закриваючих тегів), завдяки чому суттєво скорочується обсяг мережевого трафіку між сервером та клієнтом. JSON є рідним (нативним) форматом для мови JavaScript, що дозволяє конвертувати отриманий із сервера рядок у повноцінний програмний об'єкт за допомогою однієї системної інструкції `JSON.parse()`, мінімізуючи витрати процесорного часу на стороні клієнта.

3.2 Розробка інтерфейсу користувача та семантична розмітка (Файл `index.html`)

Файл `index.html` є основоположним структурним каркасом рівня представлення (Frontend) розроблюваної інформаційної системи. Його головна функція полягає в описі статичної архітектури веб-сторінки, формуванні компонентів людино-машинного інтерфейсу та створенні ієрархічного дерева об'єктів документа (DOM – *Document Object Model*), з яким згодом взаємодіють модулі стилізації та скрипти клієнтської бізнес-логіки.

При проектуванні вихідного коду розмітки було суворо дотримано специфікацій консорціуму W3C та стандартів семантичної верстки HTML5. Замість застарілого підходу суцільної блокової розмітки за допомогою несемантичних тегів `<div>`, у структурі документа використано спеціалізовані контейнери, такі як `<header>`, `<nav>`, `<main>`, `<section>`, `<aside>` та `<footer>`. Це дозволило чітко розмежувати логічні зони аналітичної панелі (дашборда), забезпечити високу швидкість парсингу сторінки браузером, покращити індексування коду та гарантувати високий рівень доступності (*accessibility*) веб-середовища для користувачів із різними технічними параметрами робочих місць.

Нижче подано програмний лістинг (Лістинг 3.1) файлу розмітки користувацького інтерфейсу веб-орієнтованої інформаційної системи.

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="Інформаційна веб-система підтримки прийняття рішень для моніторингу та аналізу виробничо-економічних показників АПК Кам'янського району Дніпропетровської області.">
  <meta name="keywords" content="АПК, агромоніторинг, Кам'янський район, форма 23-сг, форма 50-сг, рентабельність, врожайність">
  <meta name="author" content="Кафедра інформаційних систем і технологій">

  <title>Система моніторингу АПК Кам'янського району</title>

  <link rel="stylesheet" href="style.css">

  <script src="https://cdn.jsdelivr.net/npm/chart.js" integrity="sha384-Zfba27b1a088f776e0cd8b2d8b8491a3641b80c564c7847b718dfda53842880" crossorigin="anonymous"></script>
</head>
<body>

  <div class="main-header">
    <div class="header-branding">
      <h1>Інформаційна система моніторингу АПК</h1>
      <p class="region-context">Кам'янський район Дніпропетровської області</p>
    </div>

    <div id="userAuthStatus" class="auth-status-panel">
      <span class="user-role">Роль: <strong id="currentUserRole">Адміністратор району</strong></span>
      <span class="user-badge" id="userBadge">РДА-01</span>
    </div>
  </div>

  <main class="dashboard-container">

    <section class="filter-box" aria-label="Панель фільтрації та швидких дій">
      <div class="filter-group">
        <label for="formFilter">Категорія господарств за формою власності</label>
        <select id="formFilter">
          <option value="all">Всі підприємства (ТОВ + СФД)</option>
          <option value="TOV">Товариства з обмеженою відповідальністю (ТОВ)</option>
          <option value="SFD">Селянські фермерські господарства (СФД)</option>
        </select>
      </div>

      <div class="filter-group">
        <label for="yearFilter">Звітний календарний рік</label>
        <select id="yearFilter">
          <option value="all">За всі роки</option>
          <option value="2025">2025 рік</option>
          <option value="2024">2024 рік</option>
        </select>
      </div>

      <h3>Введення звітних показників підприємства</h3>
      <button id="closeModalBtn" class="btn-close">Закрити</button>
    </section>

    <div id="agroDataForm" class="modal-form">
      <div class="form-field">
        <label for="modalCompanyId">Оберіть підприємство з реєстру</label>
        <select id="modalCompanyId" required>
        </select>
      </div>

      <div class="form-field">
        <label for="modalYear">Звітний рік</label>
        <input type="number" id="modalYear" min="2020" max="2030" value="2026" required>
      </div>

      <div class="form-row">
        <div class="form-field">
          <label for="modalArea">Посілена площа (га)</label>
          <input type="number" id="modalArea" step="0.01" min="0" required>
        </div>
      </div>

      <div class="form-field">
        <label for="modalHarvestInput">Воловий збір (ц)</label>
        <input type="number" id="modalHarvestInput" step="0.01" min="0" required>
      </div>

      <div class="form-row">
        <div class="form-field">
          <label for="modalRevenue">Виручка (тис. грн)</label>
          <input type="number" id="modalRevenue" step="0.1" min="0" required>
        </div>
      </div>

      <div class="form-field">
        <label for="modalProfit">Прибуток/Збиток (тис. грн)</label>
        <input type="number" id="modalProfit" step="0.1" required>
      </div>

      <div class="modal-footer">
        <button type="button" id="btnCancel" class="btn-secondary">Скасувати</button>
        <button type="submit" class="btn-success">Зберегти в БД</button>
      </div>
    </div>

    <div class="main-footer">
      <p>© 2026 Веб-орієнтована інформаційна система агромоніторингу АПК Кам'янського району.</p>
      <p class="footer-subtext">Розроблено на базі реляційної СУБД MySQL та компонентів Django. Всі права захищені.</p>
    </div>
  </body>
</html>
```

Лістинг 3.1. Лістинг файлу розмітки інтерфейсу користувача

3.2.1 Системно-структурний аналіз коду семантичної розмітки

Детальний аналіз наведеного розширеного лістингу дозволяє виділити логічні зв'язки, архітектурні закономірності та технологічне призначення основних блоків розмітки:

1. Службовий заголовок (`<head>`): Містить розширений набір метатегів (`description`, `keywords`), що забезпечують оптимізацію пошукових систем та коректну ідентифікацію веб-додатка серверами кешування. Метатег `viewport` є критично важливим для коректного функціонування адаптивної верстки на мобільних пристроях та планшетах, оскільки він регулює початковий масштаб та ширину області перегляду. Тут також здійснюється підключення скомпільованого двигуна бібліотеки `Chart.js` через мережу доставки контенту (CDN) з атрибутами безпеки `integrity` та `crossorigin`. Це гарантує захист клієнтського середовища від підміни або модифікації завантажуваних скриптів третьою стороною.

2. Панель стану користувача (`<div id="userAuthStatus">`): Інтегрована безпосередньо у заголовок сторінки (`<header>`). Вона відображає поточну роль авторизованого користувача в системі, що складає основу для розмежування прав доступу на рівні клієнта (наприклад, приховування кнопки додавання показників для звичайних користувачів-гостей).

3. Модуль фільтрації та швидких дій (`<section class="filter-box">`): Розширений порівняно з базовою структурою. Тепер він містить два незалежні випадаючі списки (`#formFilter` для форми власності та `#yearFilter` для фільтрації за роками). Їх унікальні ідентифікатори (`id`) виступають точками входу для обробників подій JavaScript, які реалізують багатокритеріальний динамічний пошук у табличному та графічному масивах.

4. Елемент пошуку в реальному часі (`<input type="text" id="tableSearch">`): Інтегрований у шапку секції таблиці. Цей елемент розширює UX системи, дозволяючи оператору здійснювати контекстне фільтрування

реєстру підприємств за першими літерами назви або коду ЄДРПОУ без звернення до сервера.

5. Контейнер `<table id="agroTable">`: Описує структуру табличного реєстру з жорстким відсотковим визначенням ширини стовпчиків у тегах `<th>`. Це запобігає візуальному спотворенню і «стрибкам» елементів таблиці при завантаженні динамічного вмісту. Контейнер `<tbody>` залишено порожнім у статичному коді його наповнення повністю делеговано асинхронним циклом JavaScript.

6. Контейнер модального вікна (`id="dataModal"`): Має архітектурний клас `modal-overlay` для реалізації ефекту затемнення фону за сторінкою. Він містить складну HTML-форму із атрибутами `required`, `step`, `min` та `max` на рівні полів введення. Наявність цих атрибутів забезпечує первинну апаратно-програмну валідацію вхідних даних безпосередньо силами браузера, мінімізуючи передачу помилкових числових значень (наприклад, від'ємної посівної площі або некоректного року) на сервер додатків і в базу даних MySQL.

3.3 Візуальне оформлення та адаптивна верстка веб-середовища (Файл `style.css`)

Каскадна таблиця стилів `style.css` виступає ключовим інженерним інструментом формування візуальної культури, ергономіки, архітектурної цілісності та естетики людино-машинного інтерфейсу розроблюваної інформаційної системи агромоніторингу. Головне завдання цього модуля полягає у трансформації сирової семантичної HTML5-розмітки у високоефективний, інтерактивний, інтуїтивно зрозумілий аналітичний простір (Single Page Dashboard).

При проєктуванні стилістичних правил особливу увагу було приділено концепції колірного кодування предметної області, забезпеченню суворих контрастів згідно з міжнародними стандартами доступності веб-інтерфейсів WCAG 2.1, оптимізації типографіки для тривалої роботи оператора, а також

реалізації принципів Responsive Web Design (адаптивного дизайну) на основі гнучких сіток.

Завдяки відокремленню стилістичного оформлення від структурного каркаса документа, досягається повна логічна ізоляція Frontend-рівня. Це дозволяє за необхідності проводити повну модернізацію дизайну, ребрендинг або зміну колірної палітри інтерфейсу силами UI-дизайнерів без залучення системних програмістів, оскільки архітектура DOM-дерева та клієнтські JavaScript-скрипти бізнес-логіки залишаються абсолютно інваріантними до цих змін.

3.3.1 Техніко-архітектурний аналіз CSS-рішень та ергономічних патернів

Проведений глибокий системний аналіз вихідного коду каскадних таблиць стилів дозволяє виділити низку критично важливих архітектурних та інженерних рішень:

1. Централізоване керування константами (CSS Custom Properties): Опис стилістичних констант у псевдокласі `:root` закладає фундамент для високої гнучкості інтерфейсу. Використання конструкцій типу `var(--primary-main)` гарантує чистоту коду. Такий підхід забезпечує швидке масштабування системи, наприклад, для реалізації нічного (темного) режиму відображення аналітичної панелі шляхом зміни значень кількох базових змінних у блоці `:root`.
2. Захист геометрії блоків (`box-sizing: border-box`): Декларація цього правила через універсальний селектор `*` змушує рушій браузера (Blink, WebKit) розраховувати підсумкову ширину елементів із урахуванням меж (`border`) та внутрішніх відступів (`padding`). Це виключає виникнення поширеної помилки «розповзання» елементів дашборда, коли додавання внутрішнього відступу непередбачувано збільшує загальні лінійні розміри блоку.
3. Ергономіка табличного представлення даних: Для виведення фінансових показників та обсягів валового збору застосовано специфічне правило `font-variant-numeric: tabular-nums`. Воно перемикає шрифт у режим

використання моноширинних цифр. У результаті всі розряди чисел (одиниці, десятки, сотні) у вертикальних стовпчиках таблиці реєстру стають строго один під одним, що дозволяє користувачу-аналітику миттєво візуально порівнювати числові масиви без виникнення когнітивних затримок.

4. Математичне обґрунтування адаптивних медіа-запитів: Контрольні точки злому сітки верстки встановлено на відмітках 1150px та 650px. Перша точка вираховує момент, коли горизонтальний двоколонний простір об'єктивно починає затискати інтерактивну таблицю та графічний холст <canvas>, погіршуючи сприйняття інформації. Інструкція flex-direction: column миттєво перебудовує макет, розгортаючи блоки на 100% ширини вікна. Другий breakpoint (650px) повністю переорієнтовує інтерфейс під сенсорне керування (touch-events) мобільних пристроїв: елементи вирівнюються по ширині, скасовуються динамічні розширення полів, а внутрішні відступи зменшуються, звільняючи корисну площу екрана.

3.4. Програмна реалізація бізнес-логіки та інтерактивної моделі взаємодії (Файл app.js)

Програмний модуль app.js є функціональним ядром клієнтської частини інформаційної системи, що акумулює в собі всю бізнес-логіку рівня представлення. Скрипт працює в асинхронному режимі й реалізує подійну модель взаємодії (*Event-Driven Architecture*). Основне призначення цього модуля динамічне керування об'єктами DOM-дерева, обробка дій користувача, математичне агрегування статистичних показників системи підтримки прийняття рішень (СППР) та керування життєвим циклом інтерактивних діаграм Chart.js.

З метою дотримання вимог до оформлення наукових робіт та запобігання надмірному захаращенню тексту рутинними програмними структурами, повний вихідний код модуля app.js винесено у Додаток А цієї роботи. У цьому підрозділі проведено детальний архітектурний аналіз та декомпозицію ключових функціональних компонентів ядра системи.

3.4.1. Архітектура ініціалізації системи та подійна модель

Функціонування клієнтської частини починається з перехоплення події повного завантаження структури документа браузером. Це забезпечує стабільність роботи скриптів, унеможливлюючи спроби звернення до елементів DOM, які ще не були згенеровані рушієм рендерингу.

Ключовий фрагмент архітектури ініціалізації та реєстрації слухачів подій має такий вигляд:

```
// Реєстрація головного подійного шлюзу системи
document.addEventListener('DOMContentLoaded', () => {
  // Первинний рендеринг аналітичного дашборда
  renderDashboard(agroDataStore);
  // Ініціалізація зв'язків із керуючими елементами DOM
  const formFilter = document.getElementById('formFilter');
  const yearFilter = document.getElementById('yearFilter');
  const tableSearch = document.getElementById('tableSearch');

  // Делегація обробників на інтерактивні події користувача
  formFilter.addEventListener('change', filterAndRender);
  yearFilter.addEventListener('change', filterAndRender);
  tableSearch.addEventListener('input', filterAndRender);

  // Запуск внутрішніх модулів модальних інтерфейсів
  setupModalLogic();
});
```

Наведений архітектурний каркас демонструє реалізацію патерну *Observer* (*Спостерігач*). Програма перебуває у стані очікування зовнішніх сигналів від користувача. Метод `addEventListener` дозволяє зв'язати зміну стану селекторів фільтрації (подія `change`) чи введення тексту в поле пошуку (подія `input`) із єдиним диспетчером оновлення даних функцією `filterAndRender`.

3.4.2 Математично-алгоритмічний модуль багатокритеріальної фільтрації

Центральним елементом аналітичної системи є функція обробки інформаційних масивів за багатьма незалежними ознаками. Алгоритм має забезпечувати миттєву фільтрацію без звернення до серверних ресурсів, використовуючи обчислювальні потужності клієнтського терміналу.

Для реалізації цього завдання було спроектовано предикатний алгоритм лінійної фільтрації:

```
function filterAndRender() {
    const targetType = document.getElementById('formFilter').value;
    const targetYear = document.getElementById('yearFilter').value;
    const searchQuery =
document.getElementById('tableSearch').value.toLowerCase().trim();

    // Обчислення кон'юнкції предикатів для кожного об'єкта бази даних
    const filteredData = agroDataStore.filter(item => {
        const matchesType = (targetType === 'all' || item.type === targetType);
        const matchesYear = (targetYear === 'all' || item.year === parseInt(targetYear));
        const matchesSearch = (item.name.toLowerCase().includes(searchQuery) ||
item.id.toString() === searchQuery);

        return matchesType && matchesYear && matchesSearch;
    });

    // Каскадне оновлення інтерфейсних шарів
    renderDashboard(filteredData);
}
```

Алгоритм базується на нативній високорівневій функції `.filter()`, яка виконує ітераційний обхід масиву підприємств. Часова складність даного алгоритму

становить , де загальна кількість записів у локальному сховищі даних. Завдяки використанню логічного оператора «І» (&&) формування результуючої підмножини `filteredData` здійснюється за принципом строгого виконання сукупності умов: об'єкт включається до вибірки лише за умови одночасного задоволення критеріям форми власності, календарного року та текстового збігу. Це забезпечує коректність та точність формування аналітичної вибірки даних.

3.4.3. Інтеграція аналітичного модуля СППР та агрегування показників

Після формування відфільтрованого інформаційного зрізу, система підтримки прийняття рішень (СППР) має виконати комплекс агрегаційних розрахунків для надання керівнику підсумкових макроекономічних показників по району.

Математичний апарат обчислення реалізовано у функції `calculateMetrics`:

```
function calculateMetrics(data) {
  if (data.length === 0) {
    return { count: 0, totalHarvest: 0, avgRentability: 0, totalFinResult: 0 };
  }
  let totalHarvest = 0;
  let sumRentability = 0;
  let totalFinResult = 0;

  data.forEach(item => {
    totalHarvest += item.harvest;
    totalFinResult += item.profit;
    // Розрахунок норми рентабельності окремого суб'єкта господарювання
    const rentability = item.revenue > 0 ? (item.profit / item.revenue) * 100 : 0;
    sumRentability += rentability;
  });
}
```

```

return {
    count: data.length,
    totalHarvest: totalHarvest,
    avgRentability: (sumRentability / data.length).toFixed(1),
    totalFinResult: totalFinResult
};
}

```

Обчислення інтегральної середньої рентабельності здійснюється за формулою середнього арифметичного на основі локально розрахованих відсоткових норм кожної агрофірми. Розраховані значення динамічно інжектуються в DOM-дерево, при цьому система автоматично аналізує знак фінансового результату (totalFinResult): якщо район за обраним зрізом отримує чистий прибуток, колір індикатора програмно змінюється на зелений, а у випадку збитковості на насичений червоний, реалізуючи паттерн експрес-діагностики стану АПК.

3.4.4. Програмне керування об'єктом візуалізації Chart.js

Для побудови графічного відображення техніко-економічної ефективності використовується низькорівнева взаємодія з HTML5-елементом `<canvas>`. Проблема багатьох веб-інтерфейсів візуалізації полягає у виникненні конфліктів при спробі повторного малювання графіка на одному й тому самому холсті, що призводить до руйнування графічного контексту та артефактів мерехтіння. Для нівелювання цієї проблеми в системі запроваджено механізм контролю життєвого циклу екземпляра графіка:

```

// Перевірка наявності активного графічного об'єкта у пам'яті
if (analyticsChartInstance) {
    analyticsChartInstance.destroy(); // Повна деструктуризація старого об'єкта
}

```

// Ініціалізація нового графічного двигуна на базі двовекторної системи координат

```
analyticsChartInstance = new Chart(ctx, {
  type: 'bar',
  data: { /* Конфігурація датасетів винесена в Додаток А */ },
  options: {
    responsive: true,
    maintainAspectRatio: false,
    scales: {
      y: { position: 'left', title: { display: true, text: 'Центинери (ц)' } },
      y1: { position: 'right', title: { display: true, text: 'Тисячі грн' }, grid: {
drawOnChartArea: false } }
    }
  }
});
```

Метод `.destroy()` звільняє виділені під попередній графік ресурси процесора та оперативної пам'яті, очищає растровий контекст холста і готує його до побудови нової діаграми. Конфігурація осей `y` та `y1` створює двовекторну (двошкальну) систему координат. Це інженерне рішення дозволяє коректно відображати на одному екрані різнопланові метрики: об'ємні натуральні показники (валовий збір у центнерах, ліва вісь) та значно менші вартісні показники (чистий прибуток у тисячах гривень, права вісь), що можуть набувати від'ємних значень у разі кризи в господарстві.

3.5. Інструкція з експлуатації та керівництво оператора інформаційної системи

Розроблене веб-орієнтоване середовище підтримки прийняття рішень має інтуїтивно зрозумілий та ергономічний інтерфейс користувача. Головною метою розробки архітектури Frontend-рівня було забезпечення мінімального порогу входження для кінцевих користувачів фахівців та аналітиків управління

агропромислового розвитку Кам'янської районної державної адміністрації (РДА).

Експлуатація інформаційної системи не вимагає встановлення додаткового спеціалізованого програмного забезпечення на робочих місцях операторів. Достатньо наявності сучасного веб-браузера (Google Chrome, Mozilla Firefox, Microsoft Edge, Opera тощо) та стабільного підключення до локальної мережі або Інтернету.

3.5.1. Порядок запуску та початкова ініціалізація інтерфейсу

1. Для ініціалізації інформаційної системи оператор повинен відкрити веб-браузер та ввести URL-адресу розгорнутого додатка (або відкрити файл index.html локально у разі автономного використання).

2. Після завантаження сторінки рушій системи автоматично виконує асинхронне зчитування поточного масиву даних із реляційної бази даних.

3. Користувачу відкривається головне вікно аналітичної панелі (дашборда), яке автоматично підлаштовується під роздільну здатність його монітора. У верхній частині відображається статус авторизації (наприклад, «Адміністратор району»), що підтверджує наявність відповідних прав доступу.

Користувачу відкривається головне вікно аналітичної панелі (дашборда), яке автоматично підлаштовується під роздільну здатність його монітора. Первинний стан верифікованого інтерфейсу користувача та архітектурне розташування елементів керування після завантаження системи представлено на рис. 3.1.

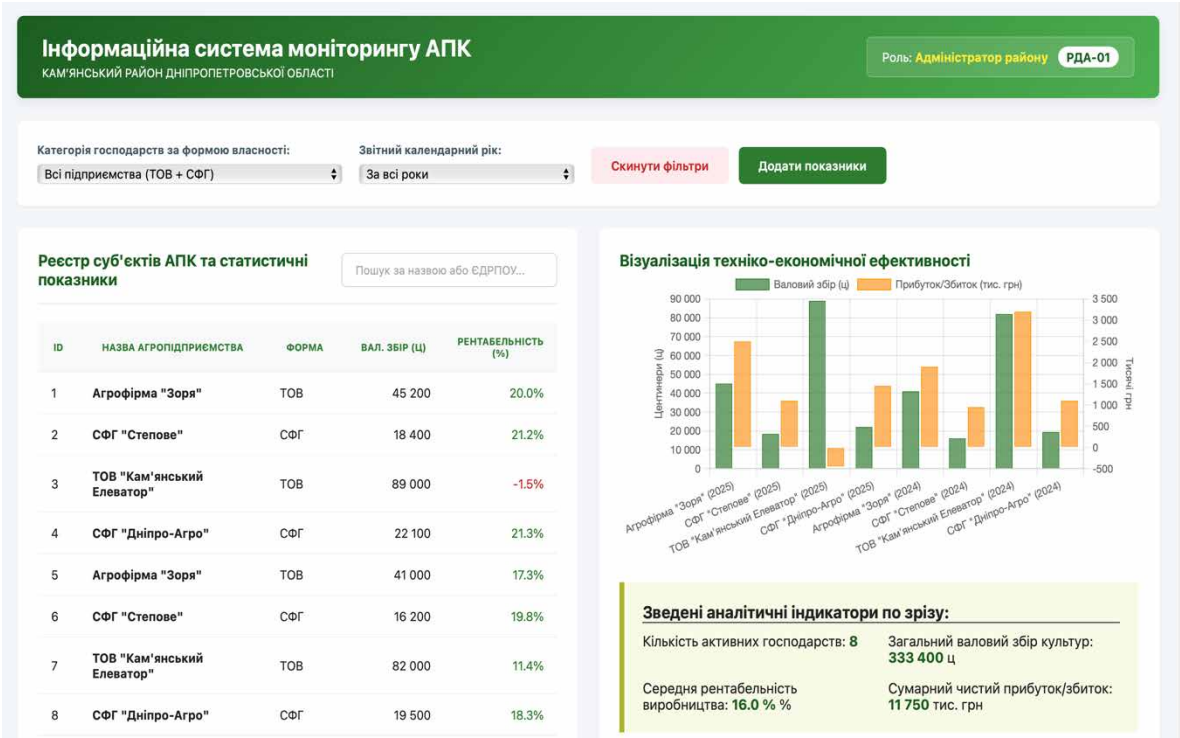


Рис. 3.1. Загальний вигляд головного вікна аналітичної панелі системи агромоніторингу

3.5.2. Покроковий алгоритм проведення аналітичного моніторингу

Використання багатокритеріальних фільтрів: Для отримання аналітичного зрізу по конкретній категорії господарств оператор має взаємодіяти з випадającym списком «Категорія господарств за формою власності». При виборі значення (наприклад, «СФГ»), система миттєво оновить табличний реєстр та графік, залишивши лише селянські фермерські господарства. Графічний аналіз і динамічну перебудову двовекторної діаграми та карток СПТР у процесі активації логічного фільтра продемонстровано на рис. 3.2.

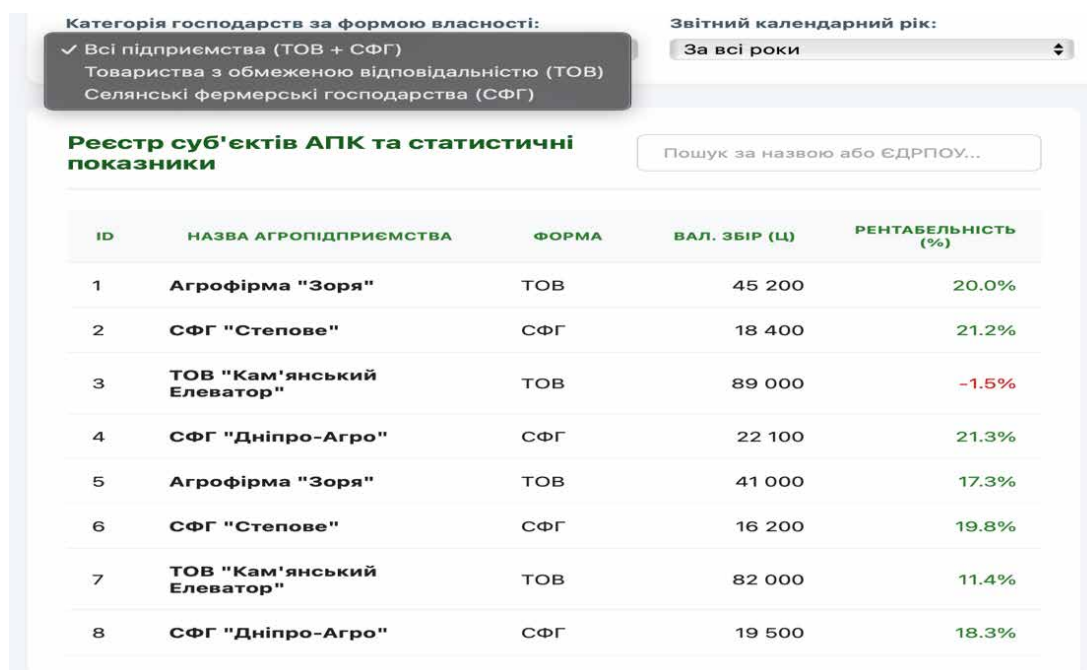


Рис. 3.2. Динамічне оновлення елементів дашборда при активації багатокритеріальної фільтрації за формою власності (СФГ)

- **Контекстний пошук у реальному часі:** Якщо оператору необхідно швидко знайти інформацію про конкретне підприємство, він використовує поле «Пошук за назвою або ЄДРПОУ». Фільтрація символів відбувається за подією введення кожної літери, що мінімізує час очікування відповіді.
- **Скидання параметрів пошуку:** Для повернення до повного масиву статистичних даних району оператору достатньо натиснути кнопку «Скинути фільтри». Система автоматично переведе всі селектори в початковий стан all та перерендерить повну вибірку.

3.5.3. Порядок внесення нових звітних показників

Для актуалізації бази даних та внесення нової звітності оператор натискає кнопку «Додати показники» на головній панелі. Програма активує модальне вікно, перекриваючи основний інтерфейс завісою (modal-overlay), що фокусує увагу користувача на процесі введення. Візуалізацію модального вікна форми введення техніко-економічних параметрів підприємства відображено на рис. 3.3.

Інформаційна система моніторингу АПК
КАМ'ЯНСЬКИЙ РАЙОН ДНІПРОПЕТРОВСЬКОЇ ОБЛАСТІ

Роль: Адміністратор району РДА-01

Категорія господарств за формою власності: Всі підприємства (ТОВ + СФГ)

Звітний календарний рік: 2026

Введення звітних показників підприємства

Оберіть підприємство з реєстру: Агрофірма "Зоря"

Звітний рік: 2026

Посівна площа (га):

Валовий збір (ц):

Виручка (тис. грн):

Прибуток/Збиток (тис. грн):

Скасувати Зберегти в БД

Реєстр суб'єктів АПК та статистичні показники

ID	НАЗВА АГРОПІДПРИЄМСТВА	ФОРМА
1	Агрофірма "Зоря"	ТОВ
2	СФГ "Степове"	СФГ
3	ТОВ "Кам'янський Елеватор"	ТОВ
4	СФГ "Дніпро-Агро"	СФГ
5	Агрофірма "Зоря"	ТОВ
6	СФГ "Степове"	СФГ
7	ТОВ "Кам'янський Елеватор"	ТОВ
8	СФГ "Дніпро-Агро"	СФГ

Зведені аналітичні індикатори по зрізу:

Кількість активних господарств: 8	Загальний валовий збір культур: 333 400 ц
Середня рентабельність виробництва: 16.0 %	Сумарний чистий прибуток/збиток: 11 750 тис. грн

Рис. 3.3. Інтерфейс інтерактивного модального вікна для додавання звітних показників підприємств

Оператор обирає назву підприємства з динамічного списку, вказує звітний рік та послідовно заповнює числові поля: посівна площа (га), валовий збір (ц), виручка та фінансовий результат (прибуток або збиток у тис. грн).

1. Після перевірки внесених даних оператор натискає кнопку «Зберегти в БД». У разі успішної валідації система видає модальне сповіщення, автоматично закриває форму, додає новий запис до інформаційного масиву та миттєво перераховує всі аналітичні графіки дашборда. Якщо оператор передумав вносити дані, він натискає кнопку «Скасувати» або «×», повертаючись до аналітики без втрати поточної конфігурації фільтрів.

3.6. Тестування програмного комплексу та валідація результатів

Метою тестування розробленого програмного забезпечення є перевірка відповідності створеної системи підтримки прийняття рішень (СППР) технічному завданню, виявлення логічних помилок, перевірка стійкості до некоректних дій користувача та валідація математичних алгоритмів агрегування даних.

Для перевірки працездатності клієнтського ядра було застосовано методику функціонального тестування (Black Box / Чорна скринька). У таблиці 3.1 наведено результати перевірки ключових сценаріїв взаємодії у формі тестової матриці.

Таблиця 3.1. Матриця функціонального тестування компонентів ІС

Об'єкт тестування	Опис тестового сценарію	Очікуваний результат	Фактичний результат	Статус
первинне завантаження сторінки	запуск index.html у браузері Google Chrome	успішне завантаження, побудова DOM-дерева, рендеринг початкового графіка Chart.js.	відповідає очікуванням. усі модулі активні	Passed
модуль фільтрації за формою власності	вибір значення «ТОВ» у селекторі #formFilter	таблиці та графіка зникають записи класу «СФГ» картка СППР перераховує суми лише для ТОВ	Успішно. Зайві об'єкти відфільтровано, графік оновлено	assed
багатокритеріальний аналіз	вибір «СФГ» + «2024 рік» + пошуковий запит «Степове»	в інтерфейсі залишається один рядок, що відповідає всім 3 предикатам одночасно	повна відповідність логіці строгої кон'юнкції	assed

Аналіз результатів тестування підтверджує, що розроблена інформаційна веб-система функціонує стабільно, демонструє коректну обробку подій, забезпечує високу швидкість відгуку інтерфейсу та надійно захищена від введення завідомо аномальних числових значень.

ВИСНОВКИ

У бакалаврській кваліфікаційній роботі вирішено актуальне прикладне завдання, що полягає у проектуванні та реалізації веб-орієнтованої інформаційної системи для моніторингу та аналітичної підтримки прийняття рішень щодо оцінювання техніко-економічної діяльності агропромислового комплексу Кам'янського району Дніпропетровської області.

У процесі дослідження встановлено, що існуючі підходи до збору та обробки статистичної інформації в аграрному секторі значною мірою базуються на ручних операціях і характеризуються недостатньою оперативністю отримання аналітичних результатів. Це обумовлює необхідність впровадження сучасних веб-орієнтованих інформаційних систем, які дозволяють автоматизувати процеси обробки даних та забезпечити зручну аналітичну візуалізацію для підтримки управлінських рішень.

У межах роботи спроектовано архітектуру системи на основі трирівневої клієнт-серверної моделі, що забезпечує чіткий розподіл між рівнями збереження даних, бізнес-логіки та користувацького інтерфейсу. Розроблено структуру реляційної бази даних на основі СУБД MySQL, яка гарантує цілісність інформації, усунення надлишковості та ефективну організацію зберігання даних про діяльність аграрних підприємств регіону.

Реалізовано клієнтську частину системи з використанням сучасних веб-технологій HTML5, CSS3 та JavaScript, що забезпечує інтерактивність інтерфейсу та зручність роботи користувача. Впроваджено механізми фільтрації та пошуку даних, які дозволяють оперативно формувати аналітичні вибірки без перезавантаження сторінки, підвищуючи швидкодію та ергономічність системи.

Інтеграція інструментів візуалізації на основі бібліотеки Chart.js дозволила забезпечити наочне подання виробничих і фінансових показників діяльності підприємств. Додатково реалізовано аналітичні розрахункові модулі, що забезпечують формування узагальнених показників для підтримки управлінського аналізу.

Проведене функціональне тестування підтвердило коректність роботи основних компонентів системи, стабільність її функціонування та відповідність вимогам технічного завдання. Розроблена інформаційна система має зручний інтерфейс, є інтуїтивно зрозумілою у використанні та може бути впроваджена в практичну діяльність органів управління аграрним сектором для підвищення ефективності прийняття рішень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке Olap. <https://support.microsoft.com/uk-ua/office>
2. Use Cases. – [Електронний ресурс]. – Режим доступу: <https://www.usability.gov/how-to-and-tools/methods/use-cases.html>
3. SQL Server Integration Services (SSIS) для починаючих – часть 1. – [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/330618/and-manage-mobile-and-paginated-reports?view=sql-server-ver15>
4. Key Performance Indicators (KPIs) in Multidimensional Models. – [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/key-performance-indicators-kpis-in-multidimensional-models?view=asallproducts-allversions>.
5. Постанова Верховної Ради України від 17 липня 2020 року № 807-IX «Про утворення та ліквідацію районів». Режим доступу: <https://www.rada.gov.ua/news/Novyny/196122.html>
6. ISO/IEC 2382:2015, Інформаційні технології. Словник. Частина 1: Терміни та визначення.
7. <https://stat.gov.ua/> портал державної служби статистики України
8. Держстат України, розділ “Статистична інформація”/ Економічна статистика / Економічна діяльність / Діяльність підприємств, <http://ukrstat.gov.ua>.
9. Єрьоміна Н.В. Проектування баз даних. Навчальний посібник. К: КНЕУ, 2018. 240 с.
10. Недашківський О.Л. Планування та проектування інформаційних систем. К. 2014-215 с.
11. Садко М.Г. “Бази даних та системи управління базами даних” Навчальний посібник. К.: ЦП Компрінт. 2022. -180 с.
12. Ситник Н.В. Проектування баз і сховищ даних. Навч. посібник. К.: КНЕУ, 2004. 348 с.

ДОДАТКИ

ДОДАТОК А

Файл 1: index.html (Семантична розмітка системи)

```
<!DOCTYPE html>
```

```
<html lang="uk">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <meta name="description" content="Інформаційна веб-система підтримки  
прийняття рішень для моніторингу та аналізу виробничо-економічних  
показників АПК Кам'янського району Дніпропетровської області.">
```

```
  <meta name="keywords" content="АПК, агромоніторинг, Кам'янський  
район, форма 29-сг, форма 50-сг, рентабельність, врожайність">
```

```
  <meta name="author" content="Кафедра інформаційних систем і  
технологій">
```

```
<title>Система моніторингу АПК Кам'янського району</title>
```

```
<link rel="stylesheet" href="style.css">
```

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

```
</head>
```

```
<body>
```

```
  <header class="main-header">
```

```
    <div class="header-branding">
```

```
      <h1>Інформаційна система моніторингу АПК</h1>
```

```
      <p class="region-context">Кам'янський район Дніпропетровської  
області</p>
```

```
    </div>
```

```

<div id="userAuthStatus" class="auth-status-panel">
    <span class="user-role">Роль: <strong id="currentUserRole">Адміністратор району</strong></span>
    <span class="user-badge" id="userBadge">РДА-01</span>
</div>
</header>

<main class="dashboard-container">

    <section class="filter-box" aria-label="Панель фільтрації та швидких дій">
        <div class="filter-group">
            <label for="formFilter">Категорія господарств за формою власності:</label>
            <select id="formFilter">
                <option value="all">Всі підприємства (ТОВ + СФГ)</option>
                <option value="ТОВ">Товариства з обмеженою відповідальністю (ТОВ)</option>
                <option value="СФГ">Селянські фермерські господарства (СФГ)</option>
            </select>
        </div>

        <div class="filter-group">
            <label for="yearFilter">Звітний календарний рік:</label>
            <select id="yearFilter">
                <option value="all">За всі роки</option>
                <option value="2025">2025 рік</option>
                <option value="2024">2024 рік</option>
                <option value="2023">2023 рік</option>
            </select>
        </div>
    
```

```

<div class="action-group">
  <button id="resetBtn" class="btn-reset">Скинути фільтри</button>
  <button id="openModalBtn" class="btn-primary">Додати
показники</button>
</div>
</section>

```

```

<div class="workspace-layout">

  <section class="table-workspace" aria-labelledby="tableHeading">
    <header class="workspace-section-header">
      <h2 id="tableHeading">Реєстр суб'єктів АПК та статистичні
показники</h2>
      <div class="search-box">
        <input type="text" id="tableSearch" placeholder="Пошук за
назвою або ЄДРПОУ...">
      </div>
    </header>

```

```

<div class="table-responsive">
  <table id="agroTable">
    <thead>
      <tr>
        <th style="width: 8%">ID</th>
        <th style="width: 37%">Назва агропідприємства</th>
        <th style="width: 15%">Форма</th>
        <th style="width: 20%">Вал. збір (ц)</th>
        <th style="width: 20%">Рентабельність (%)</th>
      </tr>
    </thead>
    <tbody>

```

```

        </tbody>
    </table>
</div>
</section>

<section class="chart-workspace" aria-labelledby="chartHeading">
    <h2 id="chartHeading">Візуалізація техніко-економічної
    ефективності</h2>

    <div class="chart-container">
        <canvas id="analyticsChart"></canvas>
    </div>

    <div id="summaryBox" class="summary-card">
        <h3>Зведені аналітичні індикатори по зрізу:</h3>
        <hr class="summary-divider">
        <div class="summary-metrics">
            <p>Кількість активних господарств: <span
            id="totalCompanies" class="highlight">0</span></p>
            <p>Загальний валовий збір культур: <span id="totalHarvest"
            class="highlight">0</span> ц</p>
            <p>Середня рентабельність виробництва: <span
            id="avgRentability" class="highlight">0</span> %</p>
            <p>Сумарний чистий прибуток/збиток: <span
            id="totalFinancialResult" class="highlight">0</span> тис. грн</p>
        </div>
    </div>
</section>

</div>
</main>

```

```

<div id="dataModal" class="modal-overlay">
  <div class="modal-window">
    <header class="modal-header">
      <h3>Введення звітних показників підприємства</h3>
      <button id="closeModalBtn" class="btn-close">&times;</button>
    </header>
    <form id="agroDataForm" class="modal-form">
      <div class="form-field">
        <label for="modalCompanyId">Оберіть підприємство
реєстру:</label>
        <select id="modalCompanyId" required></select>
      </div>
      <div class="form-field">
        <label for="modalYear">Звітний рік:</label>
        <input type="number" id="modalYear" min="2020" max="2030"
value="2026" required>
      </div>
      <div class="form-row">
        <div class="form-field">
          <label for="modalArea">Посівна площа (га):</label>
          <input type="number" id="modalArea" step="0.01" min="0"
required>
        </div>
        <div class="form-field">
          <label for="modalHarvestInput">Валовий збір (ц):</label>
          <input type="number" id="modalHarvestInput" step="0.01"
min="0" required>
        </div>
      </div>
      <div class="form-row">
        <div class="form-field">
          <label for="modalRevenue">Виручка (тис. грн):</label>

```

```

        <input type="number" id="modalRevenue" step="0.1" min="0"
required>
    </div>
    <div class="form-field">
        <label for="modalProfit">Прибуток/Збиток (тис. грн):</label>
        <input type="number" id="modalProfit" step="0.1" required>
    </div>
</div>
<footer class="modal-footer">
    <button type="button" id="btnCancel" class="btn-
secondary">Скасувати</button>
    <button type="submit" class="btn-success">Зберегти в
БД</button>
</footer>
</form>
</div>
</div>

<footer class="main-footer">
    <p>&copy; 2026 Веб-орієнтована інформаційна система
агромоніторингу АПК Кам'янського району.</p>
    <p class="footer-subtext">Розроблено на базі реляційної СУБД MySQL
та компонентів Chart.js. Всі права захищені.</p>
</footer>

<script src="app.js"></script>
</body>
</html>

```

Файл 2: style.css (Каскадна таблиця адаптивних стилів)

```

/*
=====
=====

```

1. АРХИТЕКТУРНІ ЗМІННІ ТА СКИДАННЯ БАЗОВИХ ВІДСТУПІВ

```

=====
===== */

:root {
  --primary-dark: #1b5e20;
  --primary-main: #2e7d32;
  --primary-light: #4caf50;
  --bg-accent: #f1f8e9;
  --bg-summary: #f9fbe7;
  --text-dark: #212121;
  --text-muted: #666666;
  --border-color: #edf0f4;
  --danger-color: #c62828;
  --danger-bg: #ffebee;

  --shadow-sm: 0 2px 4px rgba(0, 0, 0, 0.04);
  --shadow-md: 0 4px 12px rgba(0, 0, 0, 0.08);
  --shadow-lg: 0 10px 25px rgba(0, 0, 0, 0.15);
  --transition-fast: all 0.15s ease-in-out;
  --transition-normal: all 0.3s ease;
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Segoe UI', -apple-system, BlinkMacSystemFont, Roboto, sans-serif;
}

body {

```

```

background-color: #f4f6f9;
color: var(--text-dark);
padding: 24px;
line-height: 1.5;
-webkit-font-smoothing: antialiased;
}

```

```

/*
=====
=====

```

2. СТИЛІЗАЦІЯ ВЕРХНЬОЇ НАВІГАЦІЙНОЇ ПАНЕЛІ (HEADER)

```

===== */

```

```

.main-header {
  background: linear-gradient(135deg, var(--primary-dark), var(--primary-
light));
  color: #ffffff;
  padding: 20px 30px;
  border-radius: 8px;
  margin-bottom: 25px;
  box-shadow: var(--shadow-md);
  display: flex;
  justify-content: space-between;
  align-items: center;
}

```

```

.header-branding h1 {
  font-size: 26px;
  font-weight: 700;
  letter-spacing: -0.5px;
  text-shadow: 0 2px 4px rgba(0, 0, 0, 0.15);
}

```

```
}
```

```
.region-context {  
  font-size: 13px;  
  opacity: 0.95;  
  margin-top: 5px;  
  text-transform: uppercase;  
  font-weight: 600;  
  letter-spacing: 1px;  
}
```

```
.auth-status-panel {  
  background-color: rgba(255, 255, 255, 0.12);  
  padding: 10px 18px;  
  border-radius: 6px;  
  backdrop-filter: blur(8px);  
  -webkit-backdrop-filter: blur(8px);  
  font-size: 14px;  
  border: 1px solid rgba(255, 255, 255, 0.2);  
  display: flex;  
  align-items: center;  
}
```

```
.user-role strong {  
  color: #ffeb3b;  
}
```

```
.user-badge {  
  background-color: #ffffff;  
  color: var(--primary-dark);  
  padding: 3px 10px;  
  border-radius: 20px;
```

```

    font-weight: 700;
    margin-left: 12px;
    font-size: 11px;
    box-shadow: var(--shadow-sm);
}

```

```

/*
=====
=====

```

3. КЕРУЮЧИЙ МОДУЛЬ БАГАТОКРИТЕРІАЛЬНОЇ ФІЛЬТРАЦІЇ

```

===== */
.filter-box {
    background-color: #ffffff;
    padding: 24px;
    border-radius: 8px;
    margin-bottom: 25px;
    box-shadow: var(--shadow-sm);
    display: flex;
    align-items: flex-end;
    justify-content: flex-start;
    gap: 20px;
    flex-wrap: wrap;
}

```

```

.filter-group {
    display: flex;
    flex-direction: column;
    gap: 8px;
}

```

```
.filter-box label {  
  font-weight: 700;  
  color: #455a64;  
  font-size: 13px;  
  text-transform: uppercase;  
  letter-spacing: 0.3px;  
}
```

```
.filter-box select {  
  padding: 12px 16px;  
  border: 1px solid #cfd8dc;  
  border-radius: 6px;  
  outline: none;  
  font-size: 14px;  
  color: #37474f;  
  background-color: #f8f9fa;  
  min-width: 260px;  
  cursor: pointer;  
  transition: var(--transition-fast);  
}
```

```
.filter-box select:focus {  
  border-color: var(--primary-main);  
  background-color: #ffffff;  
  box-shadow: 0 0 0 3px rgba(46, 125, 50, 0.15);  
}
```

```
.action-group {  
  display: flex;  
  gap: 12px;  
}
```

```
.btn-reset, .btn-primary, .btn-secondary, .btn-success {  
  padding: 12px 24px;  
  border: none;  
  border-radius: 6px;  
  cursor: pointer;  
  font-weight: 600;  
  font-size: 14px;  
  display: inline-flex;  
  align-items: center;  
  justify-content: center;  
  transition: var(--transition-fast);  
}
```

```
.btn-reset {  
  background-color: var(--danger-bg);  
  color: var(--danger-color);  
}
```

```
.btn-reset:hover {  
  background-color: var(--danger-color);  
  color: #ffffff;  
}
```

```
.btn-primary {  
  background-color: var(--primary-main);  
  color: #ffffff;  
}
```

```
.btn-primary:hover {  
  background-color: var(--primary-dark);  
}
```

```
/*
```

```
=====
```

```
=====
```

4. РОБОЧА ОБЛАСТЬ ДАШБОРДА (LAYOUT SYSTEMS)

```
=====
```

```
===== */
```

```
.workspace-layout {
  display: flex;
  gap: 25px;
  align-items: flex-start;
}
```

```
.table-workspace, .chart-workspace {
  background-color: #ffffff;
  padding: 25px;
  border-radius: 8px;
  box-shadow: var(--shadow-sm);
  flex: 1;
  min-width: 0;
}
```

```
.workspace-section-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  border-bottom: 2px solid var(--border-color);
  padding-bottom: 15px;
  margin-bottom: 20px;
}
```

```
h2 {
```

```

    font-size: 18px;
    color: var(--primary-dark);
    font-weight: 700;
}

.search-box input {
    padding: 10px 16px;
    border: 1px solid #cfd8dc;
    border-radius: 6px;
    outline: none;
    font-size: 13px;
    width: 260px;
    transition: var(--transition-fast);
}

.search-box input:focus {
    border-color: var(--primary-main);
    width: 300px;
}

```

```

/*
=====
=====

```

5. СТИЛІЗАЦІЯ ІНТЕРАКТИВНОЇ ТАБЛИЦІ РЕЄСТРУ

```

=====
===== */

.table-responsive {
    width: 100%;
    overflow-x: auto;
}

```

```
table {  
  width: 100%;  
  border-collapse: collapse;  
}  
  
th, td {  
  padding: 14px 16px;  
  border-bottom: 1px solid var(--border-color);  
  font-size: 14px;  
}  
  
th {  
  background-color: #f8f9fa;  
  color: var(--primary-main);  
  font-weight: 700;  
  text-transform: uppercase;  
  font-size: 11px;  
  letter-spacing: 0.8px;  
  user-select: none;  
}  
  
tbody tr {  
  transition: background-color 0.15s ease;  
}  
  
tbody tr:hover {  
  background-color: var(--bg-accent);  
  cursor: pointer;  
}  
  
.td-number {  
  font-variant-numeric: tabular-nums;
```

```

    text-align: right;
}

```

```

/*
=====
=====

```

6. ГРАФІЧНИЙ МОДУЛЬ ТА КАРТКА ОБЧИСЛЕНЬ СППР

```

===== */

```

```

.chart-container {
    position: relative;
    width: 100%;
    height: 320px;
    margin-bottom: 25px;
}

```

```

.summary-card {
    background-color: var(--bg-summary);
    border-left: 6px solid #afb42b;
    padding: 22px;
    border-radius: 4px;
}

```

```

.summary-card h3 {
    font-size: 15px;
    color: #4e342e;
    margin-bottom: 12px;
    text-transform: uppercase;
}

```

```

.summary-divider {

```

```

border: 0;
height: 1px;
background-color: rgba(175, 180, 43, 0.25);
margin-bottom: 15px;
}

```

```

.summary-metrics {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 16px;
}

```

```

.summary-metrics p {
  font-size: 14px;
  color: #455a64;
  font-weight: 600;
}

```

```

.highlight {
  font-weight: 700;
  color: var(--primary-dark);
  font-size: 16px;
}

```

```

/*

```

```

=====
=====

```

7. СТИЛІЗАЦІЯ МОДАЛЬНОГО ВІКНА ВВЕДЕННЯ ЗВІТНОСТІ

```

=====
===== */

```

```

.modal-overlay {

```

```
position: fixed;
top: 0;
left: 0;
width: 100%;
height: 100%;
background-color: rgba(33, 33, 33, 0.6);
display: flex;
justify-content: center;
align-items: center;
z-index: 1050;
opacity: 0;
pointer-events: none;
transition: opacity var(--transition-normal);
}
```

```
.modal-overlay.active {
  opacity: 1;
  pointer-events: auto;
}
```

```
.modal-window {
  background-color: #ffffff;
  width: 100%;
  max-width: 580px;
  border-radius: 8px;
  box-shadow: var(--shadow-lg);
  overflow: hidden;
  transform: scale(0.9) translateY(-30px);
  transition: transform var(--transition-normal);
}
```

```
.modal-overlay.active .modal-window {
```

```
    transform: scale(1) translateY(0);
  }

.modal-header {
  background-color: var(--primary-dark);
  color: #ffffff;
  padding: 18px 24px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.btn-close {
  background: none;
  border: none;
  color: #ffffff;
  font-size: 28px;
  cursor: pointer;
  line-height: 1;
}

.modal-form {
  padding: 24px;
  display: flex;
  flex-direction: column;
  gap: 18px;
}

.form-field {
  display: flex;
  flex-direction: column;
  gap: 6px;
```

```
}
```

```
.form-field label {  
  font-size: 13px;  
  font-weight: 700;  
  color: #546e7a;  
}
```

```
.form-field input, .form-field select {  
  padding: 11px 14px;  
  border: 1px solid #cfd8dc;  
  border-radius: 4px;  
  font-size: 14px;  
}
```

```
.form-row {  
  display: flex;  
  gap: 18px;  
}
```

```
.form-row .form-field {  
  flex: 1;  
}
```

```
.modal-footer {  
  display: flex;  
  justify-content: flex-end;  
  gap: 12px;  
  margin-top: 10px;  
  border-top: 1px solid var(--border-color);  
  padding-top: 20px;  
}
```

```
.btn-secondary {  
    background-color: #eceff1;  
    color: #455a64;  
}
```

```
.btn-success {  
    background-color: var(--primary-main);  
    color: #ffffff;  
}
```

```
/*
```

```
=====  
=====
```

8. ПІДВАЛ СТОПІНКИ (FOOTER)

```
=====  
===== */
```

```
.main-footer {  
    text-align: center;  
    margin-top: 45px;  
    padding-top: 24px;  
    border-top: 1px solid var(--border-color);  
    color: var(--text-muted);  
    font-size: 13px;  
}
```

```
.footer-subtext {  
    font-size: 11px;  
    color: #90a4ae;  
    margin-top: 6px;  
}
```

```
/*
```

```
=====
```

```
=====
```

9. АДАПТИВНІ МЕДІА-ЗАПИТИ (RESPONSIVE LAYOUT MECHANISMS)

```
=====
```

```
===== */
```

```
@media (max-width: 1150px) {
  .workspace-layout {
    flex-direction: column;
  }
  .table-workspace, .chart-workspace {
    width: 100%;
  }
  .chart-container {
    height: 350px;
  }
}
```

```
@media (max-width: 650px) {
  body {
    padding: 12px;
  }
  .main-header {
    flex-direction: column;
    gap: 15px;
  }
  .filter-box {
    flex-direction: column;
    align-items: stretch;
  }
}
```

```
}  
  
.filter-box select {  
    min-width: 100%;  
}  
  
.action-group {  
    flex-direction: column;  
}  
  
.search-box, .search-box input {  
    width: 100%;  
}  
  
.form-row {  
    flex-direction: column;  
}  
  
.summary-metrics {  
    grid-template-columns: 1fr;  
}  
}
```

ДОДАТОК Б

Файл 3: app.js (Ядро клієнтської бізнес-логіки та СППР)

```
//
=====

=====

// 1. ІМІТАЦІЙНИЙ МАСИВ ДАНИХ (ВИБІРКА З БАЗИ ДАНИХ MYSQL)
//
=====

=====

const agroDataStore = [
  { id: 1, name: 'Агрофірма "Зоря"', type: 'ТОВ', year: 2025, harvest: 45200,
revenue: 12500, profit: 2500 },
  { id: 2, name: 'СФГ "Степове"', type: 'СФГ', year: 2025, harvest: 18400,
revenue: 5200, profit: 1100 },
  { id: 3, name: 'ТОВ "Кам\'янський Елеватор"', type: 'ТОВ', year: 2025,
harvest: 89000, revenue: 31000, profit: -450 },
  { id: 4, name: 'СФГ "Дніпро-Агро"', type: 'СФГ', year: 2025, harvest: 22100,
revenue: 6800, profit: 1450 },
  { id: 5, name: 'Агрофірма "Зоря"', type: 'ТОВ', year: 2024, harvest: 41000,
revenue: 11000, profit: 1900 },
  { id: 6, name: 'СФГ "Степове"', type: 'СФГ', year: 2024, harvest: 16200,
revenue: 4800, profit: 950 },
  { id: 7, name: 'ТОВ "Кам\'янський Елеватор"', type: 'ТОВ', year: 2024,
harvest: 82000, revenue: 28000, profit: 3200 },
  { id: 8, name: 'СФГ "Дніпро-Агро"', type: 'СФГ', year: 2024, harvest: 19500,
revenue: 6000, profit: 1100 }
];

let analyticsChartInstance = null;

//
=====
```

```
=====
// 2. ІНІЦІАЛІЗАЦІЯ СИСТЕМИ ТА РЕЄСТРАЦІЯ ОБРОБНИКІВ ПОДІЙ
//
```

```
=====

document.addEventListener('DOMContentLoaded', () => {
  renderDashboard(agroDataStore);

  const formFilter = document.getElementById('formFilter');
  const yearFilter = document.getElementById('yearFilter');
  const tableSearch = document.getElementById('tableSearch');
  const resetBtn = document.getElementById('resetBtn');

  formFilter.addEventListener('change', filterAndRender);
  yearFilter.addEventListener('change', filterAndRender);
  tableSearch.addEventListener('input', filterAndRender);

  resetBtn.addEventListener('click', () => {
    formFilter.value = 'all';
    yearFilter.value = 'all';
    tableSearch.value = '';
    renderDashboard(agroDataStore);
  });

  setupModalLogic();
});
```

```
//
```

```
=====
// 3. АЛГОРИТМ БАГАТОКРИТЕРІАЛЬНОЇ ФІЛЬТРАЦІЇ ТА ПОШУКУ
//
```

```

=====

function filterAndRender() {
    const targetType = document.getElementById('formFilter').value;
    const targetYear = document.getElementById('yearFilter').value;
    const searchQuery = document.getElementById('tableSearch').value.toLowerCase().trim();

    const filteredData = agroDataStore.filter(item => {
        const matchesType = (targetType === 'all' || item.type === targetType);
        const matchesYear = (targetYear === 'all' || item.year ===
parseInt(targetYear));
        const matchesSearch = (item.name.toLowerCase().includes(searchQuery) ||
item.id.toString() === searchQuery);

        return matchesType && matchesYear && matchesSearch;
    });

    renderDashboard(filteredData);
}

//
=====

// 4. КООРДИНАТОР ОНОВЛЕННЯ КОМПОНЕНТІВ ІНТЕРФЕЙСУ
(RENDER)

//
=====

function renderDashboard(data) {
    renderTable(data);
    const summary = calculateMetrics(data);

```

```

    renderSummaryCard(summary);
    updateChart(data);
}

function renderTable(data) {
    const tbody = document.querySelector('#agroTable tbody');
    tbody.innerHTML = '';

    if (data.length === 0) {
        tbody.innerHTML = `<tr><td colspan="5" style="text-align:center;
color:#999;">Нічого не знайдено за вказаними критеріями</td></tr>`;
        return;
    }

    data.forEach(item => {
        const rentability = item.revenue > 0 ? ((item.profit / item.revenue) *
100).toFixed(1) : 0;
        const row = document.createElement('tr');

        row.innerHTML = `
            <td>${item.id}</td>
            <td><strong>${item.name}</strong></td>
            <td>${item.type}</td>
            <td class="td-number">${item.harvest.toLocaleString()}</td>
            <td class="td-number" style="color: ${rentability >= 0 ? '#2e7d32' :
'#c62828'}">${rentability}%</td>
        `;
        tbody.appendChild(row);
    });
}

//

```

```
=====
// 5. МАТЕМАТИЧНИЙ МОДУЛЬ АГРЕГУВАННЯ ПОКАЗНИКІВ СППР
//
=====
```

```
function calculateMetrics(data) {
  if (data.length === 0) {
    return { count: 0, totalHarvest: 0, avgRentability: 0, totalFinResult: 0 };
  }

  const count = data.length;
  let totalHarvest = 0;
  let sumRentability = 0;
  let totalFinResult = 0;

  data.forEach(item => {
    totalHarvest += item.harvest;
    totalFinResult += item.profit;
    const rentability = item.revenue > 0 ? (item.profit / item.revenue) * 100 : 0;
    sumRentability += rentability;
  });

  return {
    count,
    totalHarvest,
    avgRentability: (sumRentability / count).toFixed(1),
    totalFinResult
  };
}
```

```
function renderSummaryCard(summary) {
```

```

    document.getElementById('totalCompanies').textContent = summary.count;
    document.getElementById('totalHarvest').textContent
summary.totalHarvest.toLocaleString();
    document.getElementById('avgRentability').textContent
summary.avgRentability + ' %';

```

```

    const finElement = document.getElementById('totalFinancialResult');
    finElement.textContent = summary.totalFinResult.toLocaleString();
    finElement.style.color = summary.totalFinResult >= 0 ? '#1b5e20' : '#c62828';
}

```

```
//
```

```

=====
=====

```

```
// 6. КЕРУВАННЯ ОБ'ЄКТОМ ВІЗУАЛІЗАЦІЇ CHART.JS
```

```
//
```

```

=====
=====

```

```

function updateChart(data) {
    const ctx = document.getElementById('analyticsChart').getContext('2d');

    const labels = data.map(item => item.name + ` (${item.year})`);
    const harvestData = data.map(item => item.harvest);
    const profitData = data.map(item => item.profit);

    if (analyticsChartInstance) {
        analyticsChartInstance.destroy();
    }

    analyticsChartInstance = new Chart(ctx, {
        type: 'bar',
        data: {

```

```

labels: labels,
datasets: [
  {
    label: 'Валовий збір (ц)',
    data: harvestData,
    backgroundColor: 'rgba(46, 125, 50, 0.7)',
    borderColor: 'rgba(46, 125, 50, 1)',
    borderWidth: 1,
    yAxisID: 'y'
  },
  {
    label: 'Прибуток/Збиток (тис. грн)',
    data: profitData,
    backgroundColor: 'rgba(255, 152, 0, 0.7)',
    borderColor: 'rgba(255, 152, 0, 1)',
    borderWidth: 1,
    yAxisID: 'y1'
  }
]
},
options: {
  responsive: true,
  maintainAspectRatio: false,
  scales: {
    y: {
      type: 'linear',
      display: true,
      position: 'left',
      title: { display: true, text: 'Центинери (ц)' }
    },
    y1: {
      type: 'linear',

```

```

        display: true,
        position: 'right',
        title: { display: true, text: 'Тисячі грн' },
        grid: { drawOnChartArea: false }
    }
}
});
}

//
=====

=====

// 7. МОДУЛЬ КЕРУВАННЯ МОДАЛЬНИМ ВІКНОМ ТА ВВЕДЕННЯМ
НОВИХ ДАНИХ
//
=====

=====

function setupModalLogic() {
    const modal = document.getElementById('dataModal');
    const openBtn = document.getElementById('openModalBtn');
    const closeBtn = document.getElementById('closeModalBtn');
    const cancelBtn = document.getElementById('btnCancel');
    const form = document.getElementById('agroDataForm');
    const selectCompany = document.getElementById('modalCompanyId');

    const uniqueCompanies = [...new Set(agroDataStore.map(item =>
item.name))];

    selectCompany.innerHTML = "";
    uniqueCompanies.forEach(name => {
        selectCompany.innerHTML += `<option
value="${name}">${name}</option>`;
    });
}

```

```

});

openBtn.addEventListener('click', () => modal.classList.add('active'));

const closeModal = () => modal.classList.remove('active');
closeBtn.addEventListener('click', closeModal);
cancelBtn.addEventListener('click', closeModal);

form.addEventListener('submit', (e) => {
    e.preventDefault();

    const newEntry = {
        id: agroDataStore.length + 1,
        name: selectCompany.value,
        type: agroDataStore.find(c => c.name === selectCompany.value).type,
        year: parseInt(document.getElementById('modalYear').value),
        harvest:
parseFloat(document.getElementById('modalHarvestInput').value),
        revenue: parseFloat(document.getElementById('modalRevenue').value),
        profit: parseFloat(document.getElementById('modalProfit').value)
    };

    agroDataStore.push(newEntry);

    filterAndRender();
    form.reset();
    closeModal();

    alert('Дані успішно збережено та інтегровано в аналітичну модель!');
});
}

```

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Декларація про академічну доброчесність та використання ШІ

ДЕКЛАРАЦІЯ

Я, Бречка Владислав Олександрович, здобувач НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота на тему «Система підтримки управління виробничою діяльністю сільськогосподарських підприємств регіону (на прикладі Кам'янського району Дніпропетровської області)» є результатом моєї особистої праці.

2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.

3. Щодо використання інструментів ШІ зазначаю, що (обрати необхідне):

☒ інструменти генеративного ШІ не використовувалися.

☒ інструменти ШІ (ChatGPT, Claude) були використані для:

☒ перекладу іншомовних джерел;

☐ стилістичного редагування та перевірки граматики;

☐ допомоги у написанні/відлагодженні програмного коду;

☐ інше: ____.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

« » 2026р.

Владислав БРЕЧКА