

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ПОГОДЖЕНО
Декан факультету

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри інформаційних
систем і технологій

_____ **Ігор БОЛБОТ**
(підпис)
"___" _____ 2026 р.

_____ **Михайло ШВИДЕНКО**
(підпис)
"___" _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**на тему: «Інформаційна система оптимізації накладних витрат закупівлі та
реалізації сільськогосподарської продукції»**

Спеціальність «126 Інформаційні системи і технології»

Освітньо-професійна програма «Інформаційні системи і технології»

Гарант освітньої програми
к.е.н., доцент

_____ **Максим МОКРІСВ**
(підпис)

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

_____ **Сергій САЯПН**
(підпис)

Виконав

_____ **Назарій ЧЕРНЕЦЬ**
(підпис)

Київ – 2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**
Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри інформаційних систем і
технологій

к.е.н., доцент _____ Михайло ШВИДЕНКО
(підпис)

« ____ » _____ 2026 р.

З А В Д А Н Н Я
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ

Чернець Назарію Дмитрович

Спеціальність «126 Інформаційні системи і технології»
Освітньо-професійна програма «Інформаційні системи і технології»

Тема бакалаврської кваліфікаційної роботи (проєкту) «Інформаційна система оптимізації накладних витрат закупівлі та реалізації сільськогосподарської продукції» затверджена наказом від «19» грудня 2025 року № 3205 «С»

Термін подання завершеної роботи на кафедру «30» травня 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи:

відкриті веб-ресурси, що містять закупівельні пропозиції на сільськогосподарську продукцію; статистичні дані Міністерства аграрної політики та Державної служби статистики; нормативні показники якості зернових та олійних культур; технічна документація інструментів веб-скрапінгу та фреймворків для побудови веб-застосунків.

Перелік питань, що підлягають дослідженню:

- Аналіз предметної області та ринку сільськогосподарської продукції в Україні
- Дослідження каналів дистрибуції та джерел інформації
- Проєктування архітектури та структури даних ІС

Дата видачі завдання «19» січня 2026 р.

**Керівник бакалаврської
кваліфікаційної роботи**

(підпис)

Сергій САЯПІН

Завдання взяв до виконання

(підпис)

Назарій ЧЕРНЕЦЬ

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	6
1.1 Аналіз предметної області.....	6
1.2 Аналіз наявних інформаційних систем та платформ для моніторингу пропозицій.....	12
1.3 Визначення вимог до системи автоматизованого збору даних.....	13
1.4 Аналіз методів та інструментів реалізації системи	14
1.5 Аналіз джерел пропозицій для автоматизованого збору даних	16
РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ЗБОРУ ТА АГРЕГАЦІЇ ДАНИХ.....	20
2.1. Архітектура системи та проектні рішення	20
2.2. Логічна та фізична структура даних	21
2.3. Декомпозиція системи на модулі	23
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ	31
3.1 Реалізація модулів парсингу	31
3.2 Реалізація модулів розрахунку та геолокації	36
3.3 Веб-інтерфейс	41
3.4 Тестування	46
ВИСНОВКИ	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
Додатки А.....	62
Додатки Б.....	63

ВСТУП

У сучасних умовах динамічного розвитку аграрного сектору України малі та середні фермерські господарства стикаються з необхідністю оперативного пошуку найвигідніших пропозицій для реалізації вирощеної продукції. Інформація про закупівельні ціни розпорошена по численних джерелах: веб-сайтах трейдерських компаній, елеваторів, переробних підприємств, електронних торгових майданчиках та системах державних закупівель. Ручний моніторинг всіх цих джерел є надзвичайно трудомістким та часозатратним процесом, що призводить до втрати потенційних вигідних пропозицій.

Актуальність теми дослідження полягає у необхідності автоматизації процесу збору та систематизації інформації про пропозиції закупівлі сільськогосподарської продукції з різних джерел. Створення веб-орієнтованої інформаційної системи, здатної автоматично збирати дані про закупівельні ціни, обробляти їх та надавати фермерам зведену інформацію для порівняння, дозволить значно скоротити час на пошук вигідних каналів реалізації та підвищити ефективність прийняття рішень.

Основна проблема полягає у відсутності єдиного централізованого джерела інформації про всі доступні пропозиції закупівлі. Фермери змушені вручну відвідувати десятки різних веб-сайтів, аналізувати структуру кожного ресурсу, порівнювати умови та ціни. При цьому значна частина угод відбувається в тіньовому секторі без публікації офіційних пропозицій, що додатково ускладнює аналіз ринкової ситуації.

Об'єктом дослідження є процес автоматизованого збору та агрегації інформації про пропозиції закупівлі сільськогосподарської продукції з різномірних електронних джерел.

Предметом дослідження є методи, алгоритми та програмні засоби веб-скрапінгу для автоматизованого збору даних, технології агрегації та систематизації інформації з різних джерел, способи представлення зведених даних для ефективного порівняння пропозицій.

Наукова новизна роботи полягає у комплексному дослідженні джерел публікації пропозицій закупівлі сільськогосподарської продукції в Україні, систематизації цих джерел за критеріями доступності та можливості автоматизованого збору даних, а також розробці архітектури системи для агрегації інформації з різномірних джерел.

Практична цінність роботи полягає у створенні функціонуючого прототипу веб-орієнтованої системи, що автоматизує процес збору даних про закупівельні пропозиції та надає фермерам можливість швидкого порівняння умов від різних покупців без необхідності ручного моніторингу численних джерел.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Найпопулярніші культури для вирощування в Україні

Аграрний сектор України відіграє ключову роль у національній економіці, забезпечуючи значну частину експортних надходжень та продовольчої безпеки держави. За даними Міністерства аграрної політики та Державної служби статистики, у 2025 році загальна посівна площа під товарними культурами становила близько 23 мільйони гектарів.

Аналіз структури посівних площ дозволяє виявити найбільш поширені культури, які формують основу сільськогосподарського виробництва та, відповідно, є найбільш представленими у пропозиціях закупівлі. На основі статистичних даних було сформовано перелік десяти найпопулярніших культур за посівною площею у 2025 році.

1. Пшениця (озима та яра) – 5 464,2 тис. га (23,8% від загальної площі). Пшениця залишається стратегічною продовольчою культурою України. Озима пшениця займає переважну частину площ (5 241,5 тис. га), тоді як яра вирощується переважно для пересіву загиблих озимих посівів (222,7 тис. га).
2. Соняшник – 5 171,0 тис. га (22,5%). Соняшник є основною олійною культурою України та лідером за експортом олії. Україна забезпечує понад половину світового експорту соняшникової олії.
3. Кукурудза – 4 145,5 тис. га (18,0%). Кукурудза є ключовою фуражною та експортною культурою. У 2025 році виробництво досягло рекордних показників завдяки сприятливим погодним умовам.
4. Соя – 2 422,6 тис. га (10,5%). Соя цінується за подвійну цінність – високий вміст олії та білка. Українська соя має статус не-ГМО, що забезпечує преміум на європейському ринку.
5. Ячмінь (озимий та ярий) – 1 580,4 тис. га (6,9%). Ячмінь поділяється на фуражний (для кормів) та пивоварний (для солодування). Культура зберігає популярність як рання зернова для отримання швидких грошових надходжень.
6. Ріпак (озимий та ярий) – 1 167,2 тис. га (5,1%). Ріпак є важливою олійною культурою з високим експортним потенціалом. Практично весь експорт спрямовується до Європейського Союзу.
7. Горох – 216,5 тис. га (0,9%). Горох має нішевий ринок, але цінується за агротехнічне значення (попередник, азотфіксація). Великозерний зелений горох має попит на продовольчих ринках.

8. Овес – 161,6 тис. га (0,7%). Овес поділяється на харчовий (для пластівців) та фуражний. Харчовий овес високої якості має значну премію до базової ціни.

9. Гречка – 86,8 тис. га (0,4%). Гречка має обмежений світовий ринок, основне споживання зосереджене в Україні та країнах СНД. Культура характеризується непередбачуваною врожайністю.

10. Просо – 85,5 тис. га (0,4%). Просо має нішеву позицію на ринку, вирощується переважно в посушливих регіонах завдяки високій посухостійкості.

Окрім зернових та олійних культур, важливу роль відіграє цукровий буряк – технічна культура для виробництва цукру. Хоча його посівні площі менші (199 тисяч гектарів у 2025 році), він має високу додану вартість через переробку та специфічні канали реалізації.

Аналіз структури посівних площ показує, що три культури – пшениця, соняшник та кукурудза – разом займають понад 64% всіх посівних площ України. Для подальшого аналізу системи автоматизованого збору пропозицій закупівлі було обрано перші десять культур за посівними площами, оскільки саме ці культури формують основний обсяг товарного виробництва та, відповідно, генерують найбільшу кількість пропозицій закупівлі на ринку. Концентрація на цих культурах дозволяє охопити близько 88% всіх посівних площ та забезпечити максимальну практичну цінність системи для цільової аудиторії – малих та середніх фермерських господарств.

Особливості ціноутворення кожної культури

Ціноутворення на сільськогосподарську продукцію є складним багатофакторним процесом, що залежить як від внутрішніх характеристик культури (якість, вологість, вміст олії/білка, клейковини), так і від зовнішніх ринкових факторів (світові котирування, сезонність, логістичні витрати, курс валюти). Розуміння механізмів ціноутворення для кожної культури є важливим аспектом для прийняття обґрунтованих рішень про вибір оптимального каналу збуту.

СОНЯШНИК

Для соняшника головним показником якості є вміст олії в насінні, який безпосередньо визначає його цінність для переробних підприємств. Стандартні показники: 48-50% – базовий рівень. Вологість також важлива: стандарт становить 8%, кожен відсоток понад 8% вимагає додаткового сушіння. Підвищена вологість (більше 10%) призводить до розвитку цвілі, зниження виходу олії та, як наслідок, прибутку.

Засміченість насіння є важливим фактором: стандарт чистоти мінімум 97-98%. Домішки поділяються на органічні (частки стебел, листя), мінеральні (земля, камінчики) та зернові (насіння інших культур). Понад 5% домішок призводить до знижки зменшення вартості або відмови у прийманні. Механічні

пошкодження (колоті, биті насінини) призводять до втрати олії, швидкого прогіркання.

Найважливіші показники: Вміст олії, Вологість, Засміченість.

КУКУРУДЗА

Вологість є найважливішим показником при закупівлі кукурудзи. Базисна вологість становить 14%, за якої визначається базова закупівельна ціна. Свіжозібрана кукурудза зазвичай має вологість 25-35%, що вимагає сушіння. Витрати на сушіння та зберігання можуть становити 15-25% від вартості.

Класифікація кукурудзи поділяється на три класи: 1 клас (елітна) для експорту з високими вимогами; 2 клас (звичайна) для експорту та внутрішнього ринку; 3 клас переважно для кормів. Домішки впливають на ціну: 1 клас дозволяє не більше 2% загальних домішок, 2 клас – не більше 5%, 3 клас – не більше 10%.

Натура зерна (об'ємна маса) є показником виповненості: стандарт для 1 класу мінімум 700 г/л (грам на літр), для 2 класу – 650 г/л. Низька натура (менше 600 г/л) свідчить про недорозвинене зерно та призводить до зниження вартості. Вміст мікотоксинів критично важливий для експорту до ЄС – перевищення норм означає повну заборону.

Найважливіші показники: Класифікація, Вологість, Натура.

ПШЕНИЦЯ

Клас пшениці є найважливішим показником ціноутворення, різниця між класами може сягати 40%. Класифікація: 1 клас (елітна) – мінімум 14% білка, 28% клейковини; 2 клас – 12,5-14% білка, 25-28% клейковини; 3 клас (стандартна) – 11-12,5% білка, 23-25% клейковини; 4 клас (фуражна) – 10-11% білка, знижка -10-15%; 5 клас (фуражна) – менше 10% білка.

Вологість: стандарт 14%, підвищена 15-17% вимагає вентилявання, висока понад 17% створює ризик самонагрівання.

Зовнішні фактори: орієнтиром є Паризька біржа для європейського ринку та Чиказька біржа для глобального. Сезонність: мінімум цін липень-серпень (збір озимої), зростання грудень-квітень.

Найважливіші показники: Класифікація, Вологість.

СОЯ

Соя оцінюється за показниками - вміст олії та білка. Вміст олії: 19-20% – стандарт. Вміст білка: 36-38% - стандарт. Важливим є ГМО-статус: не-ГМО соя має перевагу на ринку ЄС перед ГМО.

Вологість: стандарт 12-14%, підвищення вологості понад 15% призводить до знижки 5-10%. Домішки: стандарт чистоти мінімум 98%, розколоті боби до

5% допустимо. Розмір бобів: великі (більше 7 мм) мають більшу вартість ніж дрібні (менше 6 мм).

Найважливіші показники: Вміст білка та олії, Вологість, ГМО-статус.

ЯЧМІНЬ

Поділяється на пивоварний та фуражний. Важливі параметри: вміст білка, екстрактивність зерна (мінімум 78-80%), енергія проростання (мінімум 95%). Натура: мінімум 600 г/л.

Найважливіші показники: Призначення (Пивоварний, Фуражний), Натура.

РІПАК

Вміст олії 40-42% - стандарт. Важливий показник рівня ерукової кислоти: для експорту до ЄС має бути менше 2%. Перевищення означає заборону імпорту. Вологість: стандарт 9-10%, ріпак дуже гігроскопічний.

Найважливіші показники: Вміст олії, Вологість.

ГОРОХ

Тип: жовтий (дешевший, для кормів), зелений (дорожчий, для продовольства). Діаметр зерна: великий понад 7 мм, дрібний менше 6 мм. Цілісність: допустимо до 5% розколотих.

Найважливіші показники: Призначення, Діаметр зерна.

Аналіз способів реалізації сільськогосподарської продукції та джерел публікації пропозицій

Розуміння механізмів реалізації сільськогосподарської продукції є важливим для визначення джерел інформації, які повинна охоплювати система автоматизованого збору даних. Реалізація може відбуватися через різні канали, кожен з яких має свої особливості публікації закупівельних пропозицій.

1. ТРЕЙДЕРИ

Трейдерські компанії здійснюють прямі закупівлі продукції у фермерів для подальшого експорту або перепродажу. Трейдери можуть мати власні закупівельні пункти в регіонах або працювати через агентів.

Великі трейдерські компанії (Kernel, Cargill, Louis Dreyfus, NIBULON) мають власні корпоративні веб-сайти, де можуть публікувати базові закупівельні ціни. Однак детальні умови закупівлі часто обговорюються індивідуально і не публікуються відкрито. Частина трейдерів публікує пропозиції на спеціалізованих аграрних порталах та біржах.

2. ЕЛЕВАТОРИ

Елеватори виконують подвійну функцію - зберігання зерна та посередництво між фермерами і покупцями. Фермер може здати зерно на елеватор для зберігання з подальшим продажем, або елеватор сам виступає покупцем.

Елеватори зазвичай публікують свої закупівельні ціни на власних веб-сайтах, якщо такі є, або на аграрних порталах, біржах. Публікуються базові ціни з урахуванням якості (вологість, засміченість).

3. ПЕРЕРОБНИКИ

Переробні підприємства (олійні заводи, цукрові заводи, комбікормові підприємства) закуповують сировину для виробництва. Часто використовуються форвардні контракти, коли ціна фіксується заздалегідь ще до збору врожаю.

Переробники рідко публікують відкриті пропозиції на широкому ринку. Закупівлі часто відбуваються за прямими договорами з постійними постачальниками. Базові ціни можуть з'являтися на корпоративних сайтах великих підприємств або в торгових пропозиціях на електронних майданчиках для форвардних контрактів.

4. ЕЛЕКТРОННІ ТОРГОВІ ПЛАТФОРМИ ТА БІРЖІ

Електронні майданчики забезпечують торгівлю через систему пропозицій купівлі та продажу. Учасники реєструються на платформі, розміщують пропозиції, укладають угоди онлайн.

Біржі та торгові платформи є найбільш прозорими джерелами інформації. Пропозиції публікуються в структурованому форматі з вказівкою, обсягу, якості, ціни, локації. Дані оновлюються в режимі реального часу. Це один з найбільш придатних джерел для автоматизованого збору даних.

5. ДЕРЖАВНІ ЗАКУПІВЛІ (Prozorro)

Державні установи здійснюють закупівлі через систему електронних закупівель Prozorro. Процедура регламентована законодавством та передбачає конкурсний відбір постачальників.

Всі державні закупівлі публікуються на порталі Prozorro у повністю структурованому форматі з детальними технічними специфікаціями, обсягами, цінами та термінами. Дані відкриті та доступні через API, що робить це джерело простим для реалізації автоматизованого збору. Однак обсяги державних закупівель сільгосппродукції є відносно невеликими.

6. КООПЕРАТИВИ ТА АСОЦІАЦІЇ

Фермерські кооперативи об'єднують продукцію своїх членів для збільшення обсягів та покращення умов реалізації. Кооператив виступає як єдиний постачальник для великих покупців.

Кооперативи рідко публікують відкриті пропозиції, оскільки працюють переважно за прямими контрактами або через посередників. Інформація може з'являтися на сайтах асоціацій або у профільних ЗМІ, але не в структурованому форматі.

ПРОБЛЕМА ТІНЬОВОГО РИНКУ

Значна частина реалізації сільськогосподарської продукції в Україні відбувається в тіньовому секторі. За експертними оцінками, від третини до половини всіх угод здійснюються за готівкові розрахунки без офіційного документування. Це створює проблему для будь-якої системи аналізу ринку.

Характеристики тіньового ринку:

- відсутність публічних пропозицій - угоди домовляються через особисті контакти, телефонні дзвінки, месенджери;
- нефіксована ціна - остаточна ціна може значно відрізнятись від заявленої;
- розрахунки готівкою – операції не проходять через банківську систему;
- відсутність контрактів або спрощені форми угод.

Системи автоматизованого збору даних можуть охопити лише офіційний сегмент ринку, де пропозиції публікуються у відкритому доступі. Тіньові угоди залишаються поза межами аналізу, що створює неповну картину ринку. Однак навіть часткова інформація про офіційні пропозиції має практичну цінність, оскільки дозволяє фермерам орієнтуватися у діапазоні ринкових цін та умовах закупівлі.

Запровадження обов'язкового використання РРО (реєстраторів розрахункових операцій), розвиток електронних торгових платформ, підвищення прозорості ринку поступово зменшують частку тіньових операцій. Це створює сприятливі умови для розвитку систем автоматизованого моніторингу ринку.

Реалізація сільськогосподарської продукції відбувається через різні канали з різним рівнем прозорості публікації пропозицій. Найбільш придатними джерелами для автоматизованого збору даних є електронні торгові платформи, державні закупівлі через Prozorro, веб-сайти великих трейдерів та елеваторів. При розробці системи автоматизованого збору пропозицій необхідно сконцентруватися саме на цих категоріях джерел: трейдери (корпоративні сайти та публікації на аграрних порталах), елеватори (власні сайти та регіональні майданчики), електронні біржі та торгові майданчики (найбільш структуровані дані). Врахування обмежень через тіньовий сектор є важливим при інтерпретації результатів аналізу.

1.2 Аналіз наявних інформаційних систем та платформ для моніторингу пропозицій

Для визначення вимог до розроблюваної системи та виявлення недоліків існуючих рішень було проведено аналіз інформаційних платформ, що надають доступ до даних про пропозиції закупівлі сільськогосподарської продукції. Основна увага приділялася можливості автоматизованого збору даних та наявності конкретних пропозицій у структурованому форматі.

UkrAgroConsult – Аналітична платформа та інформаційний сервіс

UkrAgroConsult є провідною аналітичною компанією України у сфері сільського господарства. Платформа надає широкий спектр послуг від новинної аналітики до глибоких ринкових досліджень.

Платформа публікує моніторинг цін на зерно від елеваторів, терміналів та переробників по регіонах України. Дані оновлюються щоденно та представлені у форматі таблиць з вказівкою культури, регіону, типу покупця та ціни. Однак це саме моніторинг цін, а не конкретні пропозиції з можливістю укладання угод.

UkrAgroConsult надає персоналізовані консультаційні послуги за окремою оплатою. Це включає індивідуальні ринкові дослідження, прогнози цін для конкретних господарств, аналіз ефективності каналів збуту.

Платформа не надає публічного API. Веб-сайт має структуровані таблиці з цінами, що дозволяє парсинг, але більша частина даних доступна за підпискою. Безкоштовно надаються обмежені дані.

АРК-Inform – Інформаційно-аналітичне агентство

АРК-Inform є спеціалізованим агентством, що надає інформацію про ринки зерна, олійних культур та продуктів їх переробки.

Можливості щодо пропозицій: Платформа публікує щоденний моніторинг закупівельних цін від трейдерів та елеваторів по всіх регіонах України. Інформація представлена у вигляді таблиць з розбивкою по культурах, базисах якості та локаціях. АРК-Inform також має розділ "Торгові пропозиції", де публікуються оголошення від продавців та покупців.

АРК-Inform надає API для корпоративних клієнтів за платною підпискою. Для веб-інтерфейсу доступ до повних даних також потребує платної реєстрації. Структура HTML-сторінок з таблицями дозволяє парсинг, але потребує аутентифікації.

GrainTrade UA – Електронна торгова платформа

GrainTrade UA позиціонується як електронна біржа для торгівлі зерном та олійними культурами.

Платформа публікує закупівельні ціни від елеваторів та терміналів по всій Україні. Користувачі можуть розмішувати власні пропозиції купівлі та продажу. Кожна пропозиція містить культуру, обсяг, якість, ціну, локацію.

Можливість автоматизованого збору: Доступ до перегляду пропозицій вимагає безкоштовної реєстрації. Повний функціонал (укладання угод) доступний після верифікації. Платформа не надає API. HTML-структура сайту дозволяє парсинг після автентифікації.

Висновок: GrainTrade UA має обмежену кількість активних пропозицій порівняно з іншими платформами, але надає структуровані дані. Вимога реєстрації ускладнює автоматизований збір.

Latifundist Analytics – Аналітичний портал

Latifundist є медіа-холдингом та аналітичним порталом для агробізнесу.

Можливості щодо пропозицій: Платформа не публікує конкретні пропозиції закупівлі. Фокус на аналітичних матеріалах, дослідженнях рентабельності, новинах ринку. Є калькулятори собівартості та рентабельності культур.

Висновок: Latifundist не є джерелом для збору пропозицій закупівлі, але надає корисну аналітичну інформацію про ринок.

Платформа	Конкретні пропозиції	API/Парсинг	Реєстрація	Оцінка придатності
UkrAgroConsult	Моніторинг цін	Ні / Складно	Платна	Низька
APK-Inform	Ціни оголошення	API платний	Платна	Середня
GrainTrade UA	Так	Ні / Середньо	Безкоштовна	Середня
Latifundist	Ні	Ні	Ні	Низька

Таблиця 1.1 Порівняльний аналіз інформаційних платформ

Аналіз показує, що існуючі рішення орієнтовані переважно на інформування користувачів про ринкові ціни та загальні тенденції ринку. Відсутність автоматизованого розрахунку чистого прибутку для кожного каналу збуту. Немає урахування логістичних витрат залежно від місцезнаходження господарства.

Таким чином, існує потреба у створенні спеціалізованої інформаційної системи для визначення оптимального каналу збуту з урахуванням логістики.

1.3 Визначення вимог до системи автоматизованого збору даних

На основі аналізу предметної області та існуючих платформ сформовано функціональні та нефункціональні вимоги до системи.

Функціональні вимоги:

- Система повинна забезпечувати автоматизований збір актуальних ринкових даних про ціни на сільськогосподарську продукцію з різних джерел.
- Система повинна підтримувати введення користувачем параметрів вирощеної продукції: тип культури, обсяг, якісні характеристики (вологість, олійність тощо), місцезнаходження продукції.
- Система повинна розраховувати чистий прибуток для кожного доступного каналу збуту з урахуванням логістичних витрат.
- Система повинна візуалізувати результати аналізу у зручній формі
- Система повинна надавати інформацію про особливості кожного каналу збуту.
- Система повинна підтримувати збереження пропозицій користувача та їх подальше редагування.

Нефункціональні вимоги:

- Продуктивність: час відповіді системи на запит користувача не повинен перевищувати 3 секунди при нормальному навантаженні.
- Доступність: система повинна бути доступна 24/7 з коефіцієнтом доступності не менше 95%.
- Масштабованість: архітектура системи повинна дозволяти збільшення кількості користувачів без зниження продуктивності.
- Безпека: система повинна захищати персональні дані користувачів, підтримувати HTTPS-протокол.
- Зручність використання: інтерфейс повинен бути інтуїтивно зрозумілим.
- Підтримка української мови: весь інтерфейс та контент повинні бути українською мовою.

1.4 Аналіз методів та інструментів реалізації системи

Для реалізації системи автоматизованого збору даних необхідно обрати відповідні технології та інструменти для кожного компонента: парсингу веб-сторінок, HTTP-комунікації, серверної частини, бази даних та клієнтського інтерфейсу.

ІНСТРУМЕНТИ ДЛЯ ВЕБ-СКРАПІНГУ

BeautifulSoup - бібліотека Python для парсингу HTML та XML. Переваги: проста та інтуїтивна у використанні, підтримує різні парсери (html.parser, lxml, html5lib), потужні селектори для пошуку елементів, відмінна документація. Недоліки: не виконує JavaScript (не підходить для динамічних сайтів), повільніша за низькорівневі альтернативи.

Scrapy - фреймворк для веб-скрапінгу. Переваги: вбудований HTTP-клієнт, асинхронна архітектура для високої продуктивності, middleware для обробки cookies/headers, вбудовані експортери даних (JSON, CSV, XML). Недоліки: надмірно складний для простих задач, важко налагоджувати.

Selenium - інструмент для автоматизації браузера. Переваги: виконує JavaScript, емулює реальний браузер, може взаємодіяти з елементами. Недоліки: дуже повільний, високе споживання ресурсів, нестабільність через оновлення браузерів.

Для системи обрано BeautifulSoup, оскільки цільові сайти мають статичний HTML без складного JavaScript. Простота BeautifulSoup дозволяє швидко адаптувати парсери під зміни структури сайтів. Для складніших сайтів можна додати Selenium як опцію.

БІБЛІОТЕКИ ДЛЯ HTTP-ЗАПИТІВ

Requests – стандартна бібліотека для HTTP. Переваги: проста та зрозуміла API, автоматична обробка cookies/редиректів, підтримка сесій, відмінна документація. Недоліки: синхронна (блокуюча), не підходить для масових паралельних запитів.

httpx – сучасна альтернатива Requests. Переваги: підтримує async/await, сумісна API з Requests, HTTP/2 підтримка. Недоліки: менша екосистема плагінів, відносно нова.

aiohttp – асинхронна бібліотека. Переваги: висока продуктивність для багатьох паралельних запитів, підтримка WebSocket. Недоліки: складніша у використанні, потребує розуміння asyncіо.

Обрано Requests через простоту та достатню продуктивність для послідовного парсингу 5-10 джерел. Для масштабування у майбутньому можна перейти на httpx з мінімальними змінами коду.

СЕРВЕРНІ ФРЕЙМВОРКИ

Flask – мікрофреймворк для веб-додатків. Переваги: мінімалістичний, легкий у вивченні, гнучкість архітектури, швидкий старт розробки. Недоліки: потребує ручного додавання компонентів (ORM, валідація), не масштабується "з коробки".

Django – повноцінний фреймворк. Переваги: вбудована ORM, адмін-панель, автентифікація, безпека "з коробки". Недоліки: надлишковість для простих задач, жорстка структура проекту, повільніший старт розробки.

FastAPI – сучасний фреймворк. Переваги: висока продуктивність (async), автоматична документація API, валідація через Pydantic. Недоліки: відносно новий, менша спільнота.

Обрано Flask через баланс між простотою та функціональністю. Система не потребує складної ORM чи автентифікації, Flask дозволяє швидко реалізувати REST API та інтегрувати парсери.

БАЗИ ДАНИХ

SQLite - вбудована реляційна БД. Переваги: не потребує окремого сервера, один файл на диску, достатня продуктивність для малих обсягів, вбудована у Python. Недоліки: обмежена паралельна робота, не підходить для великих навантажень.

PostgreSQL - потужна реляційна БД. Переваги: висока продуктивність, ACID-транзакції, JSON-підтримка, масштабованість. Недоліки: потребує окремого сервера, складніше налаштування.

MySQL - популярна реляційна БД. Переваги: швидкість читання, велика спільнота. Недоліки: менша підтримка складних запитів порівняно з PostgreSQL.

Обрано SQLite для простоти розгортання та достатньої продуктивності. Система зберігає кешовані дані парсингу, що не потребує складної БД. Для майбутнього масштабування можлива міграція на PostgreSQL.

На основі аналізу альтернативних інструментів обрано технологічний стек: BeautifulSoup для парсингу (простота адаптації), Requests для HTTP (достатня продуктивність), Flask для backend (швидка розробка API), SQLite для БД (простота розгортання). Цей стек забезпечує баланс між швидкістю розробки та функціональними можливостями для прототипу системи.

1.5 Аналіз джерел пропозицій для автоматизованого збору даних

Для реалізації системи автоматизованого збору необхідно детально проаналізувати потенційні джерела даних за критеріями придатності для парсингу. Кожне джерело оцінюється за чотирма ключовими параметрами: актуальність даних, доступність, тип пропозицій та технічна можливість автоматизованого збору.

1. Prozorro – Система державних закупівель

Актуальність: Інформація про закупівлі сільгосппродукції з'являється нерегулярно. За травень 2026 року нараховано два тендера, які стосуються закупівлі обраних культур.

Доступність: Повністю відкритий доступ без реєстрації. Надається публічний API для автоматизованого доступу.

Тип пропозицій: Виключно державні закупівлі від бюджетних установ, державних програм. Обсяги поодиноких тендерів можуть бути значними, але загальна частка державних закупівель у ринку є незначною.

Можливість парсингу: API з повною документацією. Структуровані дані у форматі JSON. Фільтрація по категоріях товарів за класифікатором ДК021. Ідеальне джерело для автоматизації. Prozorro має екосистему майданчиків (SmartTender, Zakupki.prom.ua, E-tender), але всі вони агрегуються в єдиному API.

Висновок: Відмінне джерело з технічної точки зору, але обмежене за обсягом даних.

2. Zernotorg.ua - Торгова платформа

Актуальність: Пропозиції оновлюються користувачами. За 2026 рік кількість пропозицій 6.

Доступність: Відкритий доступ без реєстрації до розділу "Ринок".

Тип пропозицій: Переважно пропозиції від трейдерів, елеваторів та окремих фермерів. Є як купівля, так і продаж.

Можливість парсингу: Немає API. HTML-сторінки з чіткою структурою - пропозиції у вигляді карток з класами CSS. Парсинг можливий через BeautifulSoup.

Висновок: Оптимальне співвідношення доступності та обсягу даних.

3. GrainTrade UA – Електронна біржа

Актуальність: Пропозиції оновлюються користувачами. Активна торгова платформа з постійним потоком нових оголошень.

Доступність: Вимагає безкоштовної реєстрації для перегляду детальних пропозицій. Базова інформація доступна без авторизації. Базова інформація включає в себе перші 10 сторінок біржі.

Тип пропозицій: Пропозиції від елеваторів, трейдерів, переробників. Представлені як закупівельні ціни, так і пропозиції продажу від виробників.

Можливість парсингу: Немає публічного API. HTML-структура організована у вигляді таблиць з пропозиціями. Потребує реалізації механізму авторизації у парсері, але це технічно вирішувана задача через збереження сесії.

Висновок: Перспективне джерело через активність платформи. Вимога реєстрації ускладнює, але не виключає можливість автоматизованого збору.

4. Tripoli – Агротехнологічна платформа

Актуальність: Платформа позиціонується як комплексне рішення для фермерів, включаючи пошук покупців, публікація актуальної інформації про закупівельні ціни.

Доступність: Відкритий доступ до перегляду інформації про трейдерів. Реєстрація потрібна для використання повного функціоналу платформи (укладання угод, особистий кабінет).

Тип пропозицій: Інформація про трейдерські компанії з контактними даними, спеціалізацією по культурах, регіонами роботи.

Можливість парсингу: Немає API. Структура сторінки організована у вигляді карток трейдерів з чіткими CSS-класами. Інформація включає: назву компанії, спеціалізацію (які культури закупають), регіони елеваторів, контактні дані. Парсинг через BeautifulSoup є технічно простим завданням. Дані представлені у статичному HTML без динамічного завантаження через JavaScript.

Висновок: Цінне джерело для створення довідника трейдерів. Доповнює інші джерела, які надають конкретні цінові пропозиції.

5. Agrotender – Тендерна платформа

Актуальність: Регулярне оновлення тендерів від великих компаній. Платформа спеціалізується на організації закупівельних тендерів для агробізнесу.

Доступність: Базовий перегляд тендерів доступний без реєстрації. Повний функціонал (участь у тендерах) вимагає верифікації.

Тип пропозицій: Переважно тендери від великих компаній на закупівлю значних обсягів. Це структуровані закупівельні процедури з чітко визначеними умовами, термінами, обсягами та технічними вимогами.

Можливість парсингу: Немає публічного API. Частина контенту завантажуються динамічно через JavaScript, що потребує використання Selenium або аналогічних інструментів для емуляції браузера. HTML-структура сторінок змінюється періодично, що вимагає регулярного оновлення парсера. Технічна складність парсингу вища порівняно зі статичними сайтами.

Висновок: Джерело цінне для фермерів з великими обсягами продукції. Тендери надають можливість укладання довгострокових контрактів на вигідних умовах.

6. Agro-Ukraine.com, VAS.trade, GlenDeal, BuyMyCorn

Ці платформи мають схожі характеристики: публікують оголошення від учасників ринку, мають HTML-структуру придатну для парсингу, але вимагають реєстрації або мають менший обсяг даних порівняно з основними джерелами. Кожна платформа має свою специфіку: VAS.trade орієнтована на міжнародну торгівлю, GlenDeal на B2B-сектор, BuyMyCorn спеціалізується на зернових.

Висновок: Можливо розглянути для розширення системи у майбутньому після реалізації базового функціоналу з основними джерелами.

Джерело	Актуальність	Доступність	Тип	API/Парсинг
Prozorro	Висока	Відкрита	Держ.закупівлі	API повний
Zernotorg	Висока	Відкрита	Трейдери/елеватори	HTML
GrainTrade	Висока	Реєстрація	Біржа	HTML+auth
Tripoli	Середня	Відкрита	Каталог трейдерів	HTML
Agrotender	Середня	Частково відкрита	Тендери	JS динамічний
AgroUkraine/інші	Середня	Реєстрація	Оголошення	HTML

Таблиця 1.2 – Порівняльний аналіз джерел для автоматизованого збору

На основі комплексного аналізу для реалізації системи обрано чотири пріоритетні джерела, що забезпечують різні типи інформації та покривають основні сегменти ринку.

Zernotorg.ua обрано як основне джерело поточних пропозицій від трейдерів та елеваторів завдяки відкритому доступу без реєстрації та чіткій HTML-структурі, що забезпечує стабільність парсингу. Платформа надає актуальні ціни з щоденним оновленням.

GrainTrade UA включено як джерело біржових пропозицій та закупівельних цін від великих гравців ринку. Попри вимогу реєстрації, платформа є однією з найактивніших торгових майданчиків з високою ліквідністю пропозицій.

Tripoli додано як джерело структурованого каталогу великих трейдерів з інформацією про їх спеціалізацію та регіони роботи. Це дозволяє користувачам системи не лише бачити поточні ціни, але й формувати базу потенційних покупців для довгострокової співпраці.

Agrotender обрано для охоплення сегменту тендерних закупівель від великих компаній. Цей тип пропозицій важливий для фермерів з великими обсягами продукції, оскільки тендери часто передбачають укладання довгострокових контрактів на стабільних умовах. Попри технічну складність парсингу динамічного контенту, цінність інформації виправдовує зусилля.

У майбутньому можливе розширення на додаткові джерела після відпрацювання стабільної роботи з основними чотирма платформами та отримання зворотного зв'язку від користувачів щодо пріоритетності типів інформації.

РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ЗБОРУ ТА АГРЕГАЦІЇ ДАНИХ

2.1 Архітектура системи та проектні рішення

Архітектура розробленої системи базується на класичній трирівневій моделі. Дана архітектурна модель широко застосовується у веб-орієнтованих інформаційних системах завдяки своїй гнучкості, масштабованості та можливості незалежної модифікації окремих рівнів без впливу на інші частини системи.

Рівень представлення даних реалізовано у вигляді веб-інтерфейсу. Цей рівень відповідає за взаємодію з користувачем, відображення інформації та збір вхідних параметрів. Інтерфейс побудовано за принципом адаптивного дизайну, що забезпечує коректне відображення на різних пристроях від настільних комп'ютерів до мобільних телефонів. Ключовими елементами інтерфейсу є форми введення параметрів пошуку, таблиці з результатами, статистичні панелі та картографічний компонент для візуалізації географічного розташування пропозицій.

Рівень бізнес-логіки виконано як серверний додаток на базі мікрофреймворку Flask, що надає REST API для обробки запитів від клієнтської частини. Цей рівень інкапсулює всю логіку роботи системи. Використання REST API забезпечує слабку зв'язаність компонентів та можливість майбутньої інтеграції з іншими системами або розробки альтернативних клієнтських додатків.

Рівень даних реалізовано як гібридну систему зберігання, що поєднує файлове сховище та оперативну пам'ять. Агреговані дані про пропозиції зберігаються у CSV-файлі для забезпечення персистентності між перезапусками системи, кеш геокоординат та курсів валют тримається в оперативній пам'яті у вигляді словників Python для швидкого доступу, користувацькі пропозиції зберігаються в оперативній пам'яті протягом сесії роботи. Такий підхід обрано як компроміс між простотою реалізації та достатньою функціональністю.

Взаємодія між рівнями відбувається за таким принципом: користувач через веб-інтерфейс ініціює запит, наприклад, вибирає культуру для пошуку пропозицій, JavaScript-код формує HTTP-запит до відповідного API серверного додатку, Flask-сервер обробляє запит, виконує необхідну бізнес-логіку, звертається до даних у пам'яті або файлах, формує відповідь у JSON-форматі, клієнтська частина отримує дані та оновлює інтерфейс користувача. Такий підхід забезпечує асинхронність операцій.

Принцип модульності реалізовано через виділення окремих функціональних блоків з чітко визначеними інтерфейсами взаємодії, що дозволяє незалежно розвивати та тестувати окремі компоненти. Кожне джерело даних має власний модуль-парсер, що дозволяє легко додавати нові джерела або модифікувати логіку обробки існуючих без впливу на інші частини системи.

Принцип масштабованості закладено в архітектуру через використання REST API, що дозволяє у майбутньому розподілити навантаження між кількома серверами або перейти до мікросервісної архітектури. Модульна структура парсерів дозволяє запускати їх паралельно для прискорення збору даних з різних джерел. Система кешування зменшує кількість звернень до зовнішніх API геокодування, що важливо при масштабуванні кількості оброблюваних пропозицій.

Принцип відмовостійкості реалізовано через обробку винятків на всіх рівнях системи. Якщо парсинг одного джерела завершується помилкою, система продовжує роботу з іншими джерелами. Локальний кеш координат забезпечує функціонування розрахунків навіть при недоступності сервісу геокодування. Збереження даних у файли гарантує, що зібрана інформація не втрачається при перезапуску системи.

Архітектурний вибір на користь клієнт-серверної моделі з REST API обґрунтовується рядом переваг. Чітке розділення клієнтської та серверної логіки дозволяє розробляти їх незалежно. Також така модель природно підтримує множинних клієнтів, що у майбутньому дозволить створити мобільний додаток або інтегрувати систему з іншими сервісами без змін серверної частини.

2.2 Логічна та фізична структура даних

Структура даних системи спроектована з урахуванням специфіки задачі автоматизованого збору та агрегації інформації з гетерогенних джерел. На відміну від класичних реляційних баз даних, де структура визначається заздалегідь та забезпечується жорстка нормалізація, у розробленій системі застосовано гібридний підхід, що поєднує елементи структурованого та напівструктурованого зберігання даних. Це рішення обумовлено веб-скрапінгом, де схема даних може змінюватися разом зі структурою джерел, та необхідністю швидкого прототипування без створення складної інфраструктури бази даних.

Логічна модель даних системи включає три основні сутності, кожна з яких відповідає певному функціональному аспекту системи.

Сутність Пропозиції є центральною структурою даних, що містить інформацію про кожну окрему пропозицію закупівлі. Структура включає такі атрибути: `date` – дата публікації або останнього оновлення пропозиції у форматі ДД.ММ.РРРР; `contragent` – найменування компанії або організації, що опублікувала пропозицію; `culture` - назва сільськогосподарської культури; `volume` - обсяг закупівлі; `price` - ціна закупівлі за тонну у числовому форматі; `currency` - валюта ціни, зазвичай гривні або долари; `location` – текстовий опис локації або адреса точки поставки; `contact` - контактна інформація для зв'язку з контрагентом; `source` - ідентифікатор джерела даних для відстеження походження інформації. Додатково для деяких записів можуть зберігатися обчислені поля `lat` та `lon` - географічні координати після геокодування, що прискорює повторні розрахунки.

Сутність Кеш Геокоординат реалізує механізм прискорення геокодування шляхом збереження раніше отриманих координат. Структура: `location_name` - текстова назва локації як ключ пошуку; `latitude` - географічна широта у десятковому форматі; `longitude` - географічна довгота у десятковому форматі. Кеш реалізовано як Python-словник і включає як результати динамічних запитів до API Nominatim, так і заздалегідь визначений словник з координатами понад 200 типових локацій: обласних центрів України, великих міст, районів та портів.

Сутність Користувацькі Пропозиції має структуру ідентичну `Proposals`, але призначена для збереження пропозицій, доданих безпосередньо користувачем через інтерфейс системи для порівняння з ринковими. Ці записи зберігаються окремо для можливості їх виділення в інтерфейсі та відрізнення від спарсених даних. Користувацькі пропозиції існують лише в оперативній пам'яті протягом поточної сесії роботи сервера.

Фізична реалізація структури даних відрізняється від традиційних СУБД. Основні дані про пропозиції зберігаються у CSV-файлі. Формат CSV обрано через його простоту, читабельність людиною та широку підтримку інструментами аналізу даних. Структура файлу: перший рядок містить заголовки стовпців `date`, `contractor`, `culture`, `volume`, `price`, `currency`, `location`, `contact`, `source`. Кожен наступний рядок представляє одну пропозицію; значення розділені комами. Текстові поля, що можуть містити коми, беруться у подвійні лапки. Кодування UTF-8 з BOM для коректного відображення кирилиці у різних програмах.

Кеш геокоординат та користувацькі пропозиції зберігаються в оперативній пам'яті у вигляді структур даних Python. Змінна `geocache` реалізована як словник, де ключ – назва локації, значення – кортеж координат (`lat`, `lon`). Користувацькі пропозиції у змінній `user_offers` зберігаються у списку

словників, де кожен словник представляє одну пропозицію з тими самими полями, що й у CSV-файлі. Такий підхід забезпечує високу швидкість доступу.

Рішення використати файлове сховище замість повноцінної СУБД обумовлене специфікою прототипу та обсягами даних. Типовий обсяг даних від чотирьох реалізованих джерел становить 200-700 пропозицій, що займає близько 100-150 КБ у CSV-форматі. Такі обсяги цілком можна ефективно обробляти у пам'яті сучасних комп'ютерів. Файлове сховище забезпечує простоту розгортання системи без необхідності встановлення та налаштування СУБД, спрощує резервне копіювання та міграцію даних, дозволяє легко інспектувати дані стандартними текстовими редакторами або табличними процесорами, уникає накладних витрат на з'єднання з БД та виконання SQL-запитів для простих операцій.

Водночас архітектура системи не виключає майбутньої міграції на реляційну СУБД. Модульна структура дозволяє легко замінити функції читання/запису файлів на відповідні операції з базою даних без змін бізнес-логіки. У разі масштабування системи на десятки джерел з тисячами пропозицій доцільним стане перехід на PostgreSQL або MySQL з індексуванням полів *culture*, *price*, *location* для прискорення фільтрації та пошуку.

Цілісність даних забезпечується на декількох рівнях. При парсингу застосовується валідація отриманих значень: перевірка наявності обов'язкових полів, фільтрація записів без ціни або культури, видалення дублікатів за комбінацією ключових полів. При збереженні у CSV використовується бібліотека *csv* з *Python*, що автоматично обробляє екранування спеціальних символів. При читанні застосовується *try-except* для обробки можливих помилок формату або кодування. Користувацькі пропозиції валідуються на рівні API: перевірка типів даних полів, обов'язковість ключових полів.

2.3 Декомпозиція системи на модулі

Система декомпозована на функціонально незалежні модулі, кожен з яких відповідає за певний аспект роботи. Така модульна архітектура полегшує тестування окремих компонентів та дозволяє паралельний розвиток різних частин системи.

Модуль парсингу даних

Модуль парсингу є ключовим компонентом системи, що відповідає за автоматизований збір даних з різних веб-джерел. Архітектура модуля побудована за принципом адаптерів: є загальна логіка обробки HTTP-запитів та управління помилками, а для кожного конкретного джерела реалізовано спеціалізований парсер-адаптер, що розуміє специфіку HTML-структури цього

джерела. У системі реалізовано чотири парсери для джерел Tripoli, Agrofond, Agrotender та GrainTrade.

Парсер Tripoli реалізовано у функції `parse_tripoli()`. Джерело `tripoli.land/buyers` має статичну HTML-структуру без JavaScript-рендерингу. Алгоритм роботи: HTTP-запит до URL з заголовком User-Agent для емуляції браузера; отримана HTML-сторінка передається у BeautifulSoup для парсингу; пошук всіх блоків `div` з класом `trader-card`. Для кожної картки трейдера витягуються дані: назва компанії, спеціалізація по культурах, регіони роботи, контактні дані. Фільтруються лише ті картки, де фігурують зазначені вище культури. Дані форматуються у стандартну структуру з полями `date`, `contractor`, `culture`, `volume`, `price`, `currency`, `location`, `contact`, `source`.

Парсер Agrofond реалізовано у функції `parse_agrofond()`. Це джерело має структуру оголошень на сторінці `agrofond.com`. Пропозиції організовані у блоки з класом `offer-item` або аналогічним. Кожен блок містить інформацію про культуру, ціну, обсяг, локацію; ціни можуть бути вказані у діапазоні, що потребує додаткової обробки; локації часто містять скорочення назв областей, які нормалізуються. Парсер виконує витягування даних за CSS-селекторами, очищення та нормалізацію текстових полів, конвертацію цін у числовий формат, визначення валюти.

Парсер Agrotender реалізовано у функції `parse_agrotender()`. Особливість - частина контенту може завантажуватися динамічно через JavaScript, але базова інформація доступна у статичному HTML. Необхідна обробка структурованих блоків з множинними полями. Парсер обробляє таблиці або списки тендерів, витягує назву культури, обсяг, стартову ціну, локацію поставки, термін подання пропозицій.

Парсер GrainTrade реалізовано у функції `parse_graintrade()`. Особливість - сторінка може вимагати авторизації для повного доступу, але перші 10 сторінок доступні публічно. Структура включає таблиці з сортуванням. Парсер обробляє таблиці пропозицій, витягує дані про культуру, ціну купівлі, обсяг, локацію елеватора або трейдера.

Важливим сегментом є очищення та нормалізація отриманих даних. Ціни можуть містити текстові символи валюти, пробіли, тире для діапазонів. Застосовуються регулярні вирази для вилучення числових значень. Назви культур нормалізуються до стандартного написання через словник синонімів: "пшеница" - "Пшениця", "soя бобы" - "Соя". Локації очищуються від зайвих пробілів, скорочення розгортаються: "Київ. обл." - "Київська".

Обробка помилок потрібна для стабільності парсингу. Кожен парсер обгорнуто у конструкцію `try-except` для перехоплення можливих винятків. Основні типи помилок: мережеві помилки - перехоплюються та логуються, помилки парсингу через відсутність очікуваних елементів - обробляються через перевірки. Часткова збірка даних краще, ніж повна відмова роботи системи.

При помилці конкретного джерела система продовжує парсинг інших джерел, логує повідомлення про проблему, повертає порожній список результатів для проблемного джерела.

Функція `parse_all_proposals()` є координатором всіх парсерів. Вона послідовно викликає кожен спеціалізований парсер. Виконує видалення дублікатів за комбінацією усіх полів через множину `seen`; зберігає унікальні результати у глобальну змінну `prop_data`; записує дані у csv файл через спеціальний модуль. Оновлює глобальну мітку часу. Логує статистику: кількість зібраних пропозицій з кожного джерела, загальну кількість після видалення дублікатів.

Модуль геолокації та логістичних витрат

Модуль геолокації забезпечує перетворення текстових адрес у географічні координати та розрахунок відстаней між точками. Це потрібно для модуля розрахунку прибутковості, оскільки логістичні витрати впливають на вигідність пропозиції.

Геокодування реалізовано у функції `geocode_location()`. Алгоритм роботи: спочатку функція нормалізує назву локації, видаляючи зайві пробіли та приводячи до стандартного регістру. Перевіряється наявність локації у словнику, який містить координати понад 200 найпоширеніших українських міст, районів та обласних центрів; якщо локація знайдена у словнику, координати повертаються миттєво без звернення до API; якщо локації немає у словнику, виконується запит до API OpenStreetMap Nominatim з параметрами: URL <https://nominatim.openstreetmap.org/search>, з обмеженням пошуку по Україні. Отримана JSON-відповідь парситься для витягування полів `lat` та `lon`; результат додається до кешу `geocache` для подальшого використання. Якщо API недоступний або локацію не знайдено, повертається `None`.

Розрахунок відстаней виконується функцією `calculate_distance()`. Реалізовано використання OSRM API (Open Source Routing Machine) для розрахунку реальної відстані по автомобільних дорогах. Це надає точніші результати для розрахунку логістичних витрат, оскільки враховує дорожню мережу, а не пряму лінію. Алгоритм: формується HTTP-запит до публічного OSRM сервера з координатами у форматі `lon1,lat1;lon2,lat2` (OSRM використовує порядок довгота-широта). Отримана JSON-відповідь містить поле з відстанню у метрах, відстань конвертується у кілометри діленням на 1000. Результат повертається разом з прогнозованим часом у дорозі. У разі недоступності OSRM застосовується формула Haversine для розрахунку відстані по великому колу Землі між двома точками з відомими координатами. Ця відстань менша за реальну дорожню приблизно на 20-40%, але достатня як альтернатива при збоях API.

Розрахунок логістичних витрат враховує навантаженість та тарифи транспорту, яку користувач може ввести у формі. Алгоритм обчислення: для

заданого обсягу та типу транспорту визначається кількість необхідних рейсів. Повертається загальна сума витрат.

Для оптимізації продуктивності модуль застосовує кілька стратегій. Результати геокодування кешуються у словнику `geocache` у пам'яті, що дозволяє обробляти повторні запити без звернення до API. Словник `KNOWN_LOCATIONS` містить преобчислені координати для типових локацій, що покриває деякі випадки без зовнішніх запитів. При обробці багатьох пропозицій одночасно результати геокодування зберігаються безпосередньо у структурі пропозиції у полях `lat` та `lon`, що дозволяє уникати повторних обчислень при наступних викликах функції розрахунку.

Модуль розрахунку прибутковості

Модуль розрахунку прибутковості є інтегруючою ланкою, що поєднує дані про пропозиції, геолокацію, логістику та параметри користувача для визначення найвигідніших варіантів продажу продукції. Реалізація зосереджена у функції `calculate_profit()`, яка приймає на вхід пропозицію, координати користувача, обсяг продукції у тоннах та список доступних типів транспорту.

Алгоритм розрахунку складається з наступних етапів. Спочатку виконується геокодування локації пропозиції, якщо координати ще не відомі. Функція перевіряє наявність полів `lat` та `lon` у структурі пропозиції; якщо вони відсутні, викликається `geocode_location()`; отримані координати зберігаються у полях `'lat'` та `'lon'` для подальшого використання. Якщо геокодування невдале (координати `None`), пропозиція отримує позначку недоступності для розрахунку.

Наступним кроком є розрахунок відстані від локації користувача до локації пропозиції. Викликається функція `calculate_distance()`; результат містить відстань у кілометрах, що зберігається у полі `'distance'`. Ця відстань використовується для розрахунку логістичних витрат. Якщо відстань не вдалося обчислити через помилки API, пропозиція також позначається як нерозрахована.

Далі виконується конвертація валюти для уніфікації всіх цін у гривні. Перевіряється поле `currency` пропозиції. Якщо `currency = USD`, ціна множиться на актуальний курс з глобальної змінної `usd_rate`. Якщо `currency = UAH`, ціна залишається без змін. Результат зберігається у додаткове поле `price_uah` для відображення користувачу у гривнях.

Наступним етапом є розрахунок логістичних витрат з урахуванням доступних типів транспорту. Для кожного типу з масиву `vehicles` обчислюються витрати за формулою з попереднього модуля. Обирається варіант з мінімальними витратами як оптимальний. Результат зберігається у полі `logistics_cost`. Якщо масив `vehicles` порожній, використовується середній тип транспорту за замовчуванням.

Фінальний розрахунок прибутковості виконується за формулою: $\text{Прибуток} = (\text{Обсяг} \times \text{Ціна_грн}) - \text{Логістичні_витрати}$. Таким чином, пропозиція збагачується обчисленими полями `distance`, `logistics_cost`, `price_uah`, `profit`, які використовуються для ранжування та відображення.

Після обчислення прибутковості для всіх пропозицій виконується ранжування. Список пропозицій сортується за прибутковістю. Необчислені пропозиції відсуваються в кінець списку. При однаковому прибутку додатковим критерієм може бути відстань - ближчі пропозиції мають пріоритет. Результат повертається клієнту у вигляді відсортованого JSON-масиву, готового для відображення у таблиці інтерфейсу.

Модуль REST API

REST API є інтерфейсом взаємодії між клієнтською та серверною частинами системи. Реалізовано на базі мікрофреймворку Flask, API надає набір HTTP-ендпоінтів для виконання різних операцій над даними. Всі ендпоінти повертають дані у JSON-форматі для зручності обробки JavaScript-кодом клієнта.

Ендпоінт GET `/api/proposals` повертає список пропозицій з можливістю фільтрації за культурою. Якщо користувач вказує назву культури, система фільтрує дані та повертає лише релевантні пропозиції. Відповідь містить як спарсені пропозиції з веб-джерел, так і додані користувачем пропозиції, час останнього оновлення даних. Фільтрація виконується без урахування регістру символів.

Ендпоінт GET `/api/statistics` обчислює та повертає статистичні показники по обраній культурі. Система аналізує всі доступні пропозиції та визначає мінімальну, максимальну та середню ціну, а також підраховує загальну кількість пропозицій для цієї культури. Якщо назву культури не вказано або пропозицій не знайдено, повертається відповідне повідомлення про помилку. Ці дані використовуються для відображення статистичної панелі в інтерфейсі користувача.

Ендпоінт POST `/api/calculate` виконує комплексний розрахунок прибутковості кожної пропозиції з урахуванням логістичних витрат. Користувач надсилає свої координати, обсяг продукції, цільову культуру та доступні типи транспорту. Система для кожної пропозиції обчислює відстань до користувача, розраховує логістичні витрати та визначає потенційний прибуток. Результати сортуються від найбільш до найменш вигідних пропозицій. Для оптимізації передачі даних повертаються лише топ-100 найкращих варіантів.

Ендпоінт GET `/api/geocode` перетворює текстову адресу у географічні координати. Користувач надсилає назву локації, а система повертає широту та довготу або повідомлення про помилку, якщо координати знайти не вдалося.

Цей ендпоінт використовується клієнтською частиною для перетворення введеної адреси у числові координати перед виконанням розрахунків відстаней та логістики.

Ендпоінт `POST /api/add_offer` дозволяє користувачу додати власну пропозицію для порівняння з ринковими. Система автоматично доповнює пропозицію поточною датою та спеціальними позначками, що ідентифікують її як користувачку. Додана пропозиція стає доступною у всіх подальших запитах протягом поточної сесії роботи сервера, що дозволяє користувачу порівнювати свої умови з пропозиціями з веб-джерел.

Ендпоінт `POST /api/force_update` ініціює примусове оновлення даних з усіх джерел. Замість очікування планового оновлення через 24 години, адміністратор може вручну запустити процес парсингу. Система збирає нові дані з усіх підключених джерел, зберігає їх та повертає статистику оновлення. Це корисно для контролю актуальності інформації без необхідності перезапуску всього сервера.

Обробка CORS забезпечує можливість звернення до API з різних доменів. Використовується бібліотека `flask_cors`, яка автоматично додає необхідні заголовки до відповідей сервера. У розробці дозволені запити з будь-яких джерел для зручності тестування, проте рекомендується налаштувати білий список дозволених доменів для підвищення безпеки.

Валідація вхідних даних виконується на кожному ендпоінті перед обробкою запиту. Система перевіряє наявність усіх обов'язкових параметрів, коректність типів даних та діапазони значень. Наприклад, обсяг продукції має бути додатним числом, координати мають бути у допустимих географічних межах. При виявленні некоректних даних система повертає зрозуміле повідомлення про помилку замість спроби обробки неправильного запиту, що запобігає збоям та полегшує налагодження.

Модуль фонового оновлення

Модуль фонового оновлення забезпечує автоматичну актуалізацію даних без втручання користувача. Реалізовано на базі модуля `threading` стандартної бібліотеки Python, що дозволяє виконувати парсинг паралельно з обробкою HTTP-запитів.

Архітектура модуля складається з двох функцій. Функція `background_parsing()` працює у безкінечному циклі: оновлює курс валют, запускає парсинг всіх джерел, логує результати, очікує 24 години до наступного циклу. Усі операції обгорнуто механізмом обробки помилок.

Функція `start_background_parsing()` створює новий потік, встановлює його як `daemon`-потік та активує виконання. `Daemon`-потік автоматично завершується при закритті програми, що спрощує управління додатком. При некоректному завершенні незавершені операції можуть бути втрачені, тому

система зберігає результати у файл після кожного циклу. При запуску системи завантажуються збережені дані з файлу. Якщо файл відсутній, виконується повний синхронний парсинг. Після цього запускається фоновий потік та активується Flask-сервер. Автоматичне перезавантаження вимикається для уникнення дублювання потоків.

Стратегія обробки помилок забезпечує надійність системи. При виникненні помилки система записує її у лог, але продовжує роботу за розкладом. Це робить систему стійкою до тимчасових збоїв мережі або недоступності окремих джерел.

Інтервал оновлення у 24 години обраний на основі аналізу джерел даних. Більшість торгових платформ публікують нові пропозиції нерегулярно, ціни змінюються повільно. Щоденного оновлення достатньо для актуальності без надмірного навантаження на джерела.

Frontend-модулі

Клієнтська частина системи реалізована як односторінковий веб-додаток на чистому JavaScript без використання фреймворків типу React чи Vue. Таке рішення обумовлене відносною простотою інтерфейсу та бажанням мінімізувати залежності для полегшення розгортання. Весь код міститься в одному файлі.

Модуль інтерфейсу користувача відповідає за відображення форм введення та результатів. Основні компоненти: форма вибору культури з випадającym списком, форма параметрів з полями для адреси та обсягу продукції, чекбокси для вибору типів транспорту, панель статистики що динамічно оновлюється, таблиця результатів з усіма полями пропозицій.

Інтерактивність реалізована через обробники подій JavaScript. При зміні культури автоматично виконуються запити для оновлення статистики та завантаження пропозицій. При натисканні кнопки Розрахувати збираються дані з форми, виконується валідація, відправляється POST-запит до API, результати відображаються у таблиці з виділенням найвигідніших варіантів.

Модуль картографічної візуалізації реалізовано на базі бібліотеки Leaflet.js. При ініціалізації створюється карта з центром на Україні, використовується від OpenStreetMap, для кожної пропозиції створюються маркери. Користувач може взаємодіяти з картою для вказівки локації. Клік на карту встановлює маркер відправки, координати автоматично заповнюються у форму. Підтримується геолокація браузера для автоматичного визначення місцеположення та пошук за адресою через API геокодування.

Модуль розрахунків координує взаємодію форм, API та відображення результатів. При натисканні кнопки виконується валідація полів, формується запит, відображається індикатор завантаження, відправляється запит до API, результати додаються до таблиці з сортуванням та виділенням кольором.

Додаткові функції для покращення досвіду користувача: автозбереження параметрів у `localStorage` для повторного використання, адаптивний дизайн для коректного відображення на мобільних пристроях, індикатор завантаження під час виконання запитів, підказки для пояснення призначення полів.

Обробка помилок забезпечує зрозумілий зворотний зв'язок. Мережеві помилки відображаються, поля з помилками валідації підсвічуються червоною рамкою, помилки API показуються у модальному вікні, критичні помилки логуються у консоль для діагностики.

Стилізація виконана з використанням CSS3. Застосовано зелено-білу колірну схему аграрної тематики, веб-шрифти Segoe UI для читабельності, чергування кольорів рядків у таблицях, `hover`-ефекти для кнопок. Інтерфейс побудовано за принципами простоти та зрозумілості.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Реалізація модулів парсингу

Підготовка середовища розробки

Для реалізації системи використано мову програмування Python версії 3.10. Основним веб-фреймворком обрано Flask версії 3. Для парсингу HTML-сторінок застосовано бібліотеку BeautifulSoup4, що надає можливості для навігації та пошуку елементів у структурі веб-сторінки. Бібліотека Requests використовується для виконання HTTP-запитів до джерел даних.

Встановлення залежностей виконано через менеджер пакетів pip. Створено файл requirements.txt з переліком необхідних бібліотек: Flask, flask-cors, requests, beautifulsoup4, lxml. Команда `pip install -r requirements.txt` автоматично встановлює всі залежності у віртуальне середовище.

Структура проекту складається з двох основних файлів: backend.py містить серверну логіку, парсери та API, frontend.html реалізує клієнтський інтерфейс з JavaScript-кодом. Додатково створюються файли prop_data.csv для зберігання спарсених даних.

Реалізація парсера Tripoli

Парсер для платформи tripoli реалізовано у функції parse_tripoli(). Сайт має статичну веб-структуру з інформацією про компанії-трейдерів. (Рисунок 3.1.1)

Алгоритм роботи парсера:

- у коді визначено список відомих компаній-трейдерів з їх назвами та URL профілів на платформі Tripoli (Агропросперіс, Кернел, МХП, Гленпорт Одеса, Нібулон);
- для кожного трейдера виконується HTTP-запит до його профілю на сайті;
- отримана HTML-сторінка обробляється через BeautifulSoup;
- з профілю компанії витягуються дані про місцезнаходження елеваторів, спеціалізацію по культурах, контактну інформацію;
- на основі цієї інформації дані структуруються у стандартний формат з полями date, contractor, culture, location, contact, source.

Реалізація парсера Agrofond

Особливістю цього джерела є різноманітні формати представлення цін: діапазони (наприклад, 8000-8500 грн/т), фіксовані значення (7800 грн), договірні ціни, ціни з додатковими умовами (8200 грн + доставка). Парсер використовує регулярні вирази для вилучення числових значень з текстових полів. Для діапазонів обчислюється середнє арифметичне значення: якщо знайдено два числа, ціна визначається як $(\text{min_price} + \text{max_price}) / 2$. Якщо вказано лише одне число, воно використовується як фіксована ціна. У випадку відсутності числових даних запис пропускається.

Структура оголошень представлена у вигляді окремих HTML-блоків типу `div` із класами `offer-item` або `card`. Кожен такий блок містить основну інформацію про пропозицію. Назва сільськогосподарської культури зазвичай розміщується у заголовках `h3` або тегах `strong`. Інформація про ціну знаходиться в елементах із класами `price` або `offer-price`. Дані щодо обсягу продукції можуть бути розташовані в окремих елементах `span` або `div`. Локація поставки вказується текстом, інколи із використанням скорочених назв областей, які нормалізуються. Також у структурі оголошення містяться дата публікації та контактні дані продавця, представлені у вигляді номера телефону або електронної пошти.

Алгоритм роботи парсера. Виконується HTTP-запит до URL <https://agrofond.net/zakupovujemo>. Отримана HTML-сторінка передається у BeautifulSoup з парсером lxml. За допомогою методу `soup.find_all()` виконується пошук усіх HTML-блоків, що містять інформацію про пропозиції. Для кожного знайденого блоку окремо витягуються вкладені елементи за допомогою CSS-селекторів або назв класів. Текстові дані очищуються від HTML-розмітки методом `get_text(strip=True)`, що дозволяє отримати структуровану текстову інформацію без непотрібних символів і пробілів. Після цього виконується перевірка наявності обов'язкових полів, зокрема назви культури та ціни. Далі система фільтрує лише ті оголошення, які містять культури зі списку `INTERESTED_CROPS`. На завершальному етапі оброблені дані формуються у вигляді словників Python та додаються до результуючого списку для подальшого аналізу й використання.

Механізм обробки помилок включає перехоплення винятків при відсутності очікуваних HTML-елементів. Якщо для конкретного оголошення не вдалося знайти ціну або культуру, це оголошення пропускається з логуванням попередження, але парсинг інших оголошень продовжується.

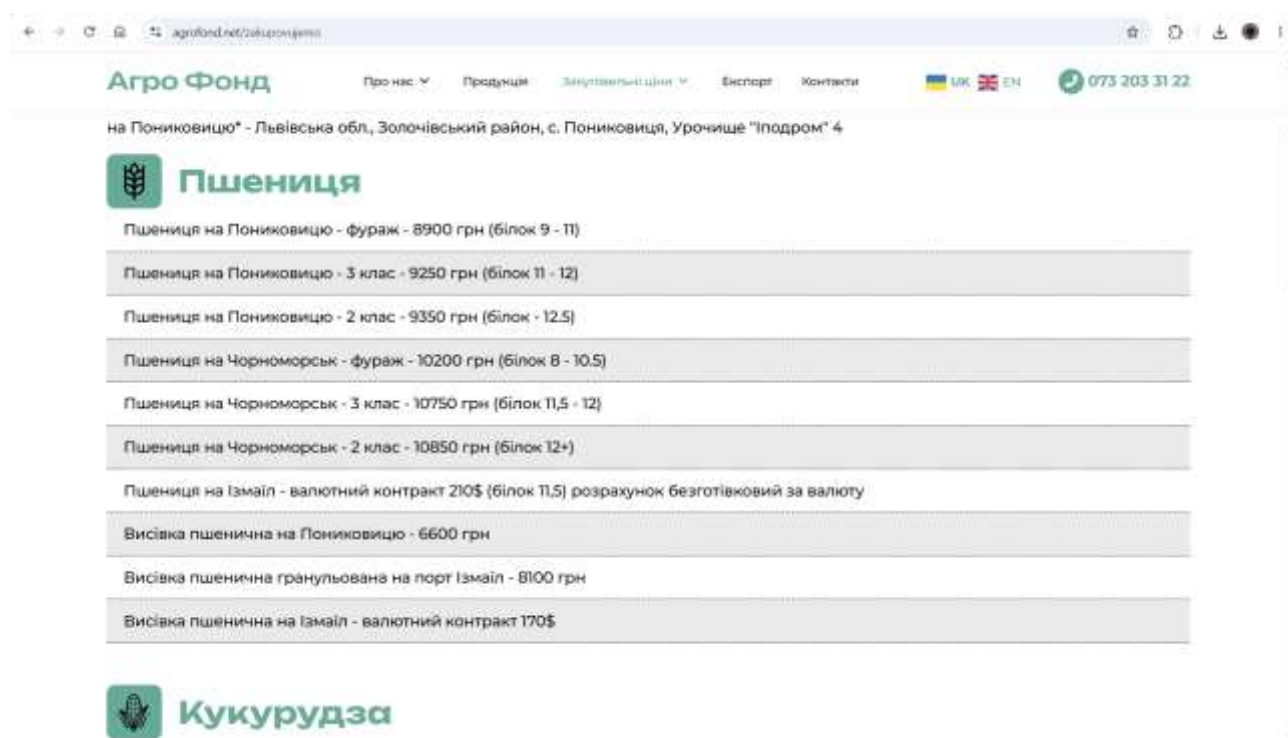


Рисунок 3.1.2. Оголошення на платформі Agrofond

Реалізація парсера Agrotender

Парсер для платформи agrotender.com.ua реалізовано у функції `parse_agrotender()`. Структура сторінки побудована як сітка карток трейдерів.

Кожна картка містить: назву компанії, список культур, валюту ціни, дату оновлення. (Рисунок 3.1.3)

Алгоритм роботи парсера. Виконується HTTP-запит до URL https://agrotender.com.ua/traders/region_ukraine. HTML-код обробляється через BeautifulSoup для пошуку всіх посилань. Знайдені URL зберігаються у множину `processed_urls`. Для кожного посилання витягується назва компанії з тегів `h2`, `h3`, `h4`, `h5`, `strong` або `b` всередині елемента. Якщо назву не знайдено, використовується перший текстовий рядок з картки.

Для кожної компанії робиться запит на її сторінку з затримкою 0.5 секунди між запитамі. Використовується словник `months_uk` для конвертації українських назв місяців у числа. Якщо знайдена дата, обчислюється кількість днів від неї до поточної дати. Якщо результат більше за 7 днів, трейдер пропускається.

На сторінці компанії виконується пошук всіх таблиць. Для кожної таблиці перевіряється перший рядок. Перша колонка має містити ключові слова регіон, елеватор, локація, місце, порт, переход для ідентифікації таблиці цін. Якщо таблиця не відповідає критеріям, вона пропускається.

Для кожного рядка обраної таблиці (крім заголовка) витягуються комірки. Перша комірка містить локацію (регіон, елеватор, порт). Наступні комірки містять ціни для відповідних культур з заголовків. З кожної комірки витягується числове значення.

Результатом роботи `parse_agrotender()` є список пропозицій. Кількість зібраних записів виводиться у консоль для контролю.



Рисунок 3.1.3. Каталог трейдерів на платформі Agrotender

Реалізація парсера GrainTrade

Парсер для біржової платформи graintrade.com.ua реалізовано у функції `parse_graintrade()`. Ця платформа публікує оголошення купівлі-продажу зерна у розділі біржа у вигляді таблиці. Парсер обробляє перші 5 сторінок результатів. (Рисунок 3.1.4)

Виконується цикл по 5 сторінках через `for page in range(1, 6)`. Для кожної сторінки формується URL `https://graintrade.com.ua/birzha?page={page}` з номером сторінки. Виконується HTTP-запит з заголовком `User-Agent` та перервою 15 секунд.

З кожного рядка витягуються дані за фіксованими індексами комірок: `cells[0]` - дата публікації оголошення, `cells[1]` - назва компанії або контрагента, `cells[2]` - тип оголошення (куплю/продам), `cells[3]` - назва культури, `cells[4]` - обсяг у тоннах або іншому форматі, `cells[5]` - ціна з валютою, `cells[7]` - локація. Текст витягується методом `get_text(strip=True)` для видалення зайвих пробілів. Перевіряється поле `offer_type` на наявність слова "куплю" через `if "куплю" not in offer_type.lower(): continue`. Оголошення типу "продам" пропускаються.

Дата	Компанія	Тип	Культура	Обсяг	Ціна	Локація
22.05.2020 17:38	ACE GRAIN	куплю	Корнуш	4000	233 дол	Рівне, Україна
22.05.2020 17:38	ACE GRAIN	куплю	Корнуш	3000	232 дол	Тернопіль, Україна
22.05.2020 17:37	ACE GRAIN	куплю	пшениця 2 клас	3000	221 дол	Рівне, Україна
22.05.2020 14:48	The Green International Limited	продам	Цукор	5000	402 дол	у вільному світі, будинок порт, Таїланд
22.05.2020 14:48	The Green International Limited	продам	Рис	30000	700 дол	у вільному світі, будинок порт, Таїланд
22.05.2020 14:48	The Green International Limited	продам	Пшениця	10000	410 дол	у вільному світі, будинок порт, Таїланд
22.05.2020 14:47	The Green International Limited	продам	Олія соняшникова	3000	800 дол	доставка по вільному світі, Таїланд

Рисунок 3.1.4. Таблиця пропозицій на платформі GrainTrade

Координатор парсингу та збереження даних

Функція `parse_all_proposals()` є стартером всіх парсерів. Алгоритм роботи:

- послідовно викликаються всі спеціалізовані парсери `parse_tripoli()`, `parse_agrofond()`, `parse_agrotender()`, `parse_graintrade()`;
- результати кожного парсера додаються до загального списку;
- виконується видалення дублікатів за комбінацією полів `contractor`, `culture`, `price`, `location`, `source` через множину `seen`;
- решта пропозиції зберігаються у глобальну змінну `prop_data`; дані записуються у файл `prop_data.csv` через модуль `csv` Python;
- оновлюється глобальна мітка часу `last_update`;
- логується статистика зібраних пропозицій.

Механізм видалення дублікатів використовує кортеж ключових полів як ідентифікатор пропозиції. Якщо така комбінація вже зустрічалася, запис пропускається. Це запобігає множинному додаванню однієї пропозиції, яка може з'явитися на кількох джерелах.

3.2 Реалізація модулів розрахунку та геолокації

Модуль геокодування локацій

Модуль геокодування відповідає за перетворення текстових назв локацій у географічні координати (широта, довгота). Реалізовано багаторівневу систему пошуку координат з резервними механізмами для забезпечення вдалого геокодування.

У коді визначено словник з обчисленими координатами більш ніж для 200 типових локацій України (`KNOWN_LOCATIONS`). Це включає всі обласні центри, великі міста, райони, порти, елеватори та логістичні центри. Наприклад: "Одеса": (46.4825, 30.7233), "Полтава": (49.5883, 34.5514). Використання словника виконується у випадку коли пошук координат за допомоги зовнішніх API не вдається.

Перед пошуком у словнику виконується функція нормалізації (`normalize_location`) текстової назви. Функція використовується у випадку коли не вдається попередньо знайти координати локації. Алгоритм нормалізації:

- видаляються слова ("елеватор", "термінал", "порт", "філіал", "комбінат", "МЕЗ");
- дефіси замінюються на пробіли;
- якщо у рядку є коми, береться остання частина після коми;
- використовуються регулярні вирази для витягування слів з великої літери, які зазвичай є назвами міст;
- виконується перевірка входження частини рядка у словнику для знаходження часткових співпадінь.

Функція `geocode_location()` є головною функцією для геокодування з каскадним алгоритмом пошуку:

Крок 1: Перевірка кешу geocache. Якщо локація вже була геокодована раніше у поточній сесії, координати повертаються з кешу одразу без повторних обчислень.

Крок 2: Формується HTTP GET-запит до <https://nominatim.openstreetmap.org/search> з параметрами: q - текстова назва локації, format=json для отримання структурованої відповіді, limit=1 для отримання лише одного найкращого результату. Встановлюється заголовок User-Agent з назвою додатку та email для ідентифікації, як вимагає політика Nominatim. Таймаут встановлено 5 секунд для запобігання довгим очікуванням. Якщо API повертає результат, координати витягуються з полів lat та lon, зберігаються у кеш і повертаються. В іншому випадку алгоритм продовжується.

Крок 3: Викликається `normalize_location()` для обробки назви. Нормалізований рядок шукається у словнику. Якщо знайдено, координати зберігаються у кеш та повертаються.

Крок 4: Якщо попередні кроки не дали результату, формується новий запит з додаванням ", Ukraine" для підвищення точності пошуку. Встановлюється затримка в 1 секунду перед запитом для дотримання ліміту Nominatim. Якщо координати знайдені, вони зберігаються у кеш як для оригінальної, так і для нормалізованої назви.

Крок 5: Якщо всі методи не дали результату, повертається (None, None) і локація позначається у кеші як така, що не знайдена, щоб уникнути повторних спроб у поточній сесії.

Кожен запит до API обгорнуто у try-ехсерт блок. При мережевих помилках, затримках або некоректній відповіді логується повідомлення з деталями помилки, але система продовжує роботу. Це забезпечує часткове функціонування навіть при недоступності зовнішніх сервісів.

Модуль розрахунку відстаней

модуль розрахунку відстаней визначає відстань між локацією користувача та локацією пропозиції. Реалізовано два методи: основний через OSRM API для точних дорожніх маршрутів та резервний через формулу Гаверсинусів для відстані по прямій.

Функція `get_road_distance_osrm()` використовує публічний API Open Source Routing Machine (OSRM) для розрахунку оптимального маршруту автомобільними дорогами. Алгоритм роботи:

Формується URL-запит до <http://router.project-osrm.org/route/v1/driving/{lon1},{lat1};{lon2},{lat2}>. OSRM використовує порядок координат довгота, широта. Параметр `overview=false` вимикає детальну геометрію маршруту для зменшення розміру відповіді, оскільки системі потрібні лише відстань та час.

Виконується GET-запит з затримкою 10 секунд. Якщо API відповідає, з відповіді витягуються дані першого маршруту про відстань і час. Відстань подається у метрах, яка конвертується у кілометри діленням на 1000. Час повертається у секундах, що конвертується у хвилини діленням на 60. Дані логуються у консоль для контролю: "OSRM розрахував відстань: 340.25 км, 245 хв".

При будь-якій помилці (мережева помилка, некоректна відповідь, маршрут не знайдено) функція повертає None, що є сигналом для використання резервного методу.

Функція `haversine()` є резервним методом розрахунку відстані по прямій лінії між двома точками на сфері. Функція використовує формулу гаверсинусів.

Константа $R = 6371$ км - середній радіус Землі. Координати конвертуються з градусів у радіани. Обчислюються різниці широт та довгот у радіанах.

Застосовується формула:

$$a = \sin(\Delta lat/2)^2 + \cos lat1 * \cos lat2 + \sin(\frac{\Delta lon}{2})^2, \quad (1.1)$$

Це проміжне значення для кута між точками.

Обчислюється кутова відстань:

$$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{(1-a)}), \quad (1.2)$$

Фінальна відстань: $\text{distance} = R * c$. Результат повертається у кілометрах. Цей метод дає відстань по прямій, яка зазвичай на 20-40% менша за реальну дорожню, але достатня для оцінки при недоступності OSRM.

Модуль розрахунку логістичних витрат та прибутковості

Це центральний модуль системи, що включає всі попередні компоненти для визначення вигідних пропозицій. Реалізовано у функції `calculate_profit()`.

Вхідні параметри:

- `offer (dict)` - словник з пропозиціями, що містить сформовані поля пропозицій.
- `user_lat, user_lon (float)` - координати локації користувача.
- `user_volume (float)` - обсяг продукції користувача у тоннах.
- `vehicles (list)` - список доступних транспортних засобів.

Алгоритм розрахунку:

Етап 1: Геокодування локації пропозиції. Перевіряються поля `offer['lat']` та `offer['lon']`. Якщо координати вже є (пропозиція була оброблена раніше), використовуються вони. Якщо координати відсутні, викликається функція `geocode_location()` для їх отримання. Якщо геокодування не вдалося

(повернуто None), встановлюються порожні значення для відстані, логістичних витрат та прибутку і пропозиція повертається як необроблена. Така пропозиція відобразатиметься у кінці списку результатів.

Етап 2: Розрахунок відстані. Викликається `get_road_distance_osrm()` для отримання відстані. Якщо OSRM повертає результат, використовуються отримана відстань. Якщо OSRM не працює (повернуто None), логується повідомлення "Використовую резервний розрахунок" і викликається функція `haversine()`. Отримана відстань округлюється до 2 знаків після коми та зберігається.

Етап 3: Оптимізація вибору транспорту. У випадку, якщо список транспортних засобів порожній, значення логістичних витрат та прибутку встановлюються рівними нулю. Після цього пропозиція повертається без виконання подальших розрахунків. Якщо транспорт доступний виконується цикл по всіх доступних транспортних засобах. Для кожної машини витягуються параметри: вантажопідйомність у тоннах та ставка оплати за кілометр. Обчислюється необхідна кількість рейсів за формулою округлення в більшу сторону, оскільки навіть неповний рейс потребує оплати.

Розраховуються загальні витрати на перевезення за формулою: кількість рейсів * відстань * 2 * ставка за кілометр. Множення на 2 враховує повернення транспорту у зворотному напрямку без вантажу. Наприклад, для перевезення 100 тонн продукції вантажівкою місткістю 20 тонн на відстань 300 км при ставці 35 грн/км необхідно виконати 5 рейсів. Загальні витрати становитимуть 105000 грн. Під час перебору всіх транспортних засобів система порівнює отримані значення витрат і зберігає найменшу знайдену вартість перевезення. Після завершення обчислень у структурі пропозиції фіксується мінімальна логістична вартість, округлена до двох знаків після коми.

Етап 4: Конвертація валюти. Витягується ціна та валюта. Якщо валюта визначається як долар США, виконується конвертація вартості у гривню за актуальним курсом. Змінна `usd_rate` містить актуальний курс долара, отриманий з API ПриватБанку при запуску системи. Логується процес конвертації для контролю. Якщо ціна вказана у гривнях, значення використовується без змін.

Етап 5: Розрахунок прибутку. Обчислюється дохід від продажу: Прибуток = к-сть тонн * ціна 1 тонни. Наприклад: 100 т * 8000 грн/т = 800000 грн. Обчислюється чистий прибуток: Чистий прибуток = Прибуток - логістичні витрати. Продовжуючи приклад: 800000 - 105000 = 695000 грн. Результат округлюється до 2 знаків після коми та зберігається. Ціна у гривнях зберігається для відображення у інтерфейсі незалежно від оригінальної валюти.

Функція повертає список пропозиції з додатковими полями: відстань, логістичні витрати, прибуток. Ці дані використовуються для сортування та відображення у таблиці результатів.

API ендпоінт розрахунку

Функція `calculate()` реалізує REST API ендпоінт `POST /api/calculate`, який координує весь процес розрахунку для всіх пропозицій.

З тіла запиту витягуються параметри: координати користувача, обсяг продукції у тоннах, назва культури, список доступних транспортних засобів. Виконується валідація обов'язкових параметрів. Якщо координати відсутні, повертається повідомлення про помилку з кодом відмови у обробці. Аналогічно перевіряється наявність назви культури. З загального списку даних створюються окремі копії пропозицій для уникнення зміни оригінальних записів. Відбираються лише ті пропозиції, де назва культури відповідає запиту користувача. Пошук виконується без урахування реєстру символів.

У консоль виводиться кількість знайдених пропозицій для контролю процесу обробки. Виконується цикл по всіх відфільтрованих пропозиціях, де для кожної викликається функція обчислення прибутку з параметрами користувача. Цей процес додає до кожної пропозиції нові поля з розрахованими показниками відстані, логістичних витрат та прогнозованого прибутку. Список впорядковується за величиною прибутку від найбільшого до найменшого. Пропозиції, для яких не вдалося розрахувати прибуток через відсутність координат або інші проблеми, отримують найнижчий пріоритет і розміщуються в кінці списку. Аналогічно обробляється окремий список пропозицій, доданих самим користувачем через інтерфейс системи.

Повертається структурована відповідь з двома масивами даних: відсортовані пропозиції з веб-джерел та відсортовані пропозиції користувача. Кожен елемент містить всі початкові дані про пропозицію плюс розраховані показники відстані, витрат на доставку, прибутковості та ціни у гривнях. Успішний статус обробки повідомляє клієнтську частину про готовність даних до відображення. Інтерфейс отримує ці дані та показує у таблиці.

Кешування та оптимізація

Для забезпечення кращої продуктивності при обробці пропозицій застосовано кілька механізмів оптимізації.

Спеціальне сховище зберігає результати визначення координат протягом роботи сервера. При першому зверненні до локації координати отримуються через зовнішній сервіс або внутрішній словник, після чого результат зберігається у кеші з назвою локації як ключем. При наступних зверненнях координати повертаються з кешу миттєво без повторних обчислень або запитів.

При першому розрахунку прибутковості отримані координати зберігаються безпосередньо у структурі даних пропозиції. При повторному розрахунку, коли користувач змінив параметри і запустив перерахунок, координати беруться з цієї структури без повторного звернення до сервісів геокодування. Це дозволяє швидше реагувати на зміну параметрів користувача.

Актуальний курс долара до гривні запитується з інтерфейсу банку один раз при запуску сервера. Результат зберігається у спеціальній змінній і використовується для всіх конвертацій протягом сесії. Оновлення курсу виконується при перезапуску системи або автоматично через фоновий процес кожні двадцять чотири години.

При масштабуванні на тисячі пропозицій доцільно повертати клієнту лише сотню найкращих результатів після сортування. Це зменшує обсяг переданих даних та прискорює відображення таблиці у браузері користувача, оскільки перегляд сотень записів одночасно не є практичним.

Завдяки цим оптимізаціям система здатна обробляти великий масив пропозицій за менший проміжок часу.

3.3 Веб-інтерфейс

Загальна структура інтерфейсу

Веб-інтерфейс системи реалізовано у файлі `frontend.html` як односторінковий додаток. Структура сторінки включає: заголовок з назвою системи, форму вибору культури з випадаючим списком, панель статистики з основними показниками, форму введення параметрів користувача, інтерактивну карту для визначення координат, таблицю результатів з розрахунками прибутковості.

Дизайн виконано у зелено-білій палітрі. Використовуються веб-шрифти Segoe UI. Адаптивний дизайн реалізовано через CSS media queries, що забезпечує коректне відображення на екранах різних розмірів. (Рисунок 3.3.1)

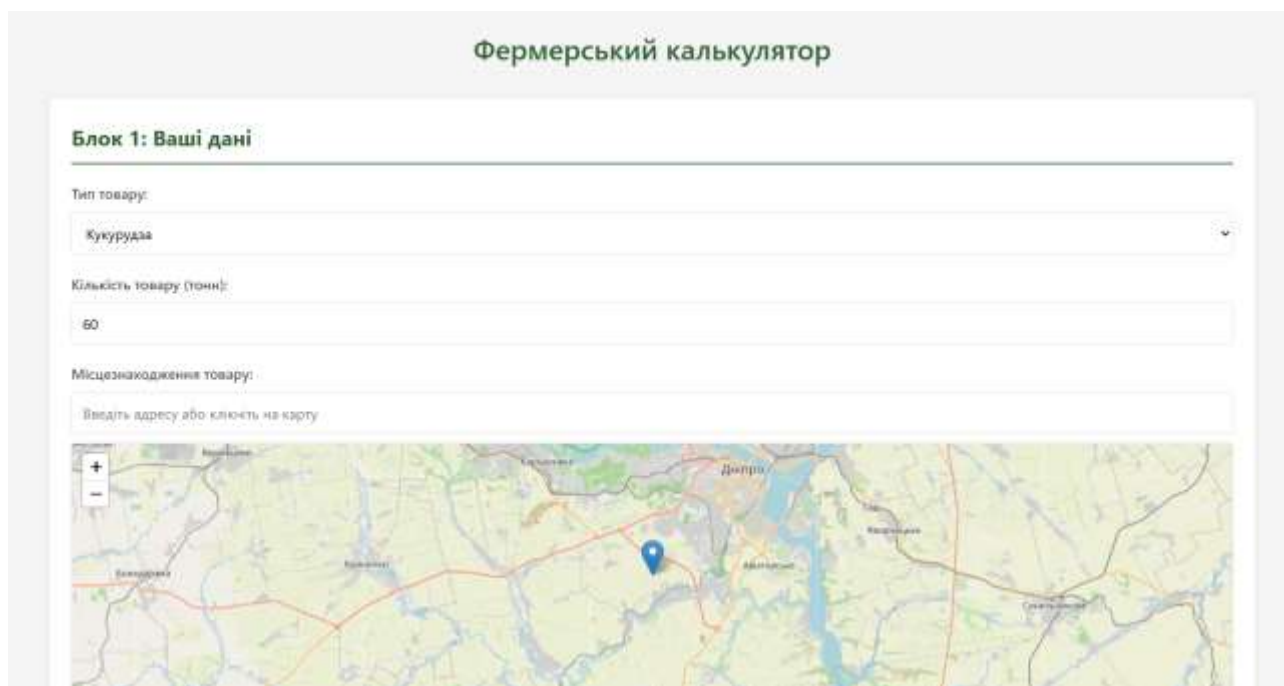


Рисунок 3.3.1. Сайт при першому запуску

Форма вибору культури та панель статистики

Форма вибору культури реалізована через HTML-елемент select з попередньо визначеним переліком культур зі списку INTERESTED_CROPS. При виборі культури автоматично виконуються два запити: до /api/statistics для отримання статистичних показників та до /api/proposals для завантаження списку пропозицій. (Рисунок 3.3.2)

Тип товару:

Кукурудза

Оберіть культуру

Соняшник

Кукурудза

Пшениця

Соя

Ячмінь

Ріпак

Горох

Овес

Гречка

Цукровий буряк

Рисунок 3.3.2. Форма вибору культури

Панель статистики динамічно оновлюється та відображає: мінімальну ціну серед усіх пропозицій для обраної культури, максимальну ціну, середню ціну з округленням до двох знаків, загальну кількість знайдених пропозицій. Дані пропозиції представлені у табличному форматі з виділенням ключових показників. (Рисунок 3.3.3)



Рисунок 3.3.3. Панель статистики

Форма параметрів користувача

Форма введення параметрів містить поля для всіх необхідних даних розрахунку прибутковості. Поле адреси відправки дозволяє ввести текстову назву локації, яка автоматично геокодується у координати через /api/geocode. Поле обсягу продукції приймає числове значення у тоннах з валідацією на додатне значення. (Рисунок 3.3.4)

Блок 1: Ваші дані

Тип товару:

Соя

Кількість товару (тонн):

60

Місцезнаходження товару:

Введіть адресу або клікніть на карту



Рисунок 3.3.4. Форма введення параметрів користувача

Кнопка Розрахувати ініціює процес обчислення прибутковості для всіх пропозицій обраної культури. При натисканні виконується валідація введених даних, формується об'єкт запиту з усіма параметрами, відправляється POST-запит до `/api/calculate`, відображається індикатор завантаження під час обробки, результати відображаються у таблиці з сортуванням за прибутковістю.

Картографічний модуль

Картографічна візуалізація реалізована на базі бібліотеки Leaflet.js версії 1.9.4, яка підключається через CDN з unpkg.com. У системі використовуються дві окремі карти: основна карта для вибору локації користувача та додаткова карта у модальному вікні для додавання власних пропозицій.

При кліку на карту створюється маркер витягуються координати точки. Якщо маркер вже існує, він видаляється, створюється новий маркер і додається на карту. Координати відображаються у текстовому блоці з округленням до 6 знаків після коми. Дані зберігаються у localStorage для повторного використання.

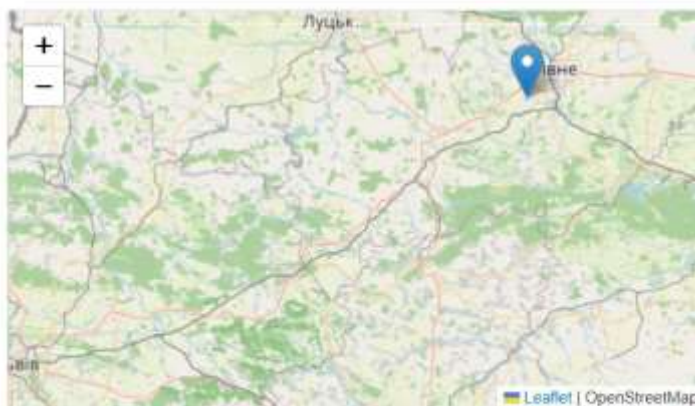
При натисканні кнопки додавання власної пропозиції відкривається модальне вікно з другою картою. Використовується затримка на 100 мс перед ініціалізацією, щоб контейнер карти встиг відобразитися. Ця карта має власний маркер та координати, незалежні від основної карти. (Рисунок 3.3.5)

Нова пропозиція на покупку

Ціна за тону (грн):

Кількість товару (тонн):

Місцезнаходження:



Координати: 50.540665, 26.138888

Рисунок 3.3.5. Модальне вікно додавання власної пропозиції

Після вибору координат викликається функція `saveData()`, яка зберігає дані у `localStorage`. При розрахунку прибутковості координати та передаються у POST-запит до `/api/calculate` разом з іншими параметрами. API використовує ці координати для обчислення відстаней до кожної пропозиції та логістичних витрат. (Рисунок 3.3.6)

Ваші пропозиції на покупку:

Соя: 8923 грн/т, 43 т.

Виділити

Додати пропозицію

Рисунок 3.3.6. Список пропозицій користувача

Таблиця результатів розрахунків

Таблиця результатів відображає пропозиції з обчисленими показниками прибутковості після натискання кнопки Розрахувати. Структура таблиці включає колонки: дата публікації пропозиції, назва контрагента, культура, обсяг закупівлі, ціна за тону у гривнях після конвертації валюти, локація поставки, відстань від користувача у кілометрах, логістичні витрати з

урахуванням обраного транспорту, прогнозований прибуток, контактна інформація. Результати автоматично сортуються за прибутковістю у порядку спадання. Рядки таблиці мають чергування кольорів для легшого сприйняття. (Рисунок 3.3.7)

Блок 3: Пропозиції на покупку

Ваші пропозиції:

Контрагент	Культура	Обсяг	Ціна	Відстань (км)	Прибуток (грн)
Моя пропозиція	Соя	43	8923 грн	835.56	485246.6

Інші пропозиції:

Дата	Контрагент	Культура	Обсяг	Ціна	Локація	Відстань (км)	Прибуток (грн)	Джерело
22.05.2026	МНО ТРАНС-СЕРВІС	Соя	-	22500 грн	Дніпропетровська обл.Марганець	22.23	1348666.03	agrotender.com.ua
22.05.2026	КІВБ ОЛІМП	Соя	-	22200 грн	Дніпропетровська обл.Богучинка	22.23	1330666.03	agrotender.com.ua
22.05.2026	ТОВ «ФЕРМЕР І ПРІВАТ...	Соя	-	22000 грн	Дніпропетровська обл.Дніпровський район	22.23	1318666.03	agrotender.com.ua
22.05.2026	МНО ТРАНС-СЕРВІС	Соя	-	22000 грн	Полтавська обл.Власівка	199.49	1308030.4	agrotender.com.ua

Рисунок 3.3.7. Таблиця результатів розрахунків

Адаптивність інтерфейсу

Адаптивний дизайн веб-застосунку забезпечує коректне відображення інтерфейсу на різних типах пристроїв - від персональних комп'ютерів до смартфонів і планшетів. Для цього використовуються CSS media queries, які автоматично змінюють структуру та оформлення сторінки при ширині екрана менше 768 пікселів.

На мобільних пристроях елементи інтерфейсу перебудовуються у вертикальну структуру з розташуванням блоків в одну колонку, що покращує зручність взаємодії користувача з системою. Інтерактивна карта автоматично зменшується до мінімального розміру або може приховуватися для економії простору екрана. Таблиці з даними стають горизонтально прокручуваними, що дозволяє переглядати великі обсяги інформації без порушення структури сторінки. Крім того, розмір шрифтів та відступів автоматично збільшується для покращення читабельності інтерфейсу на малих екранах. (Рисунок 3.3.8)

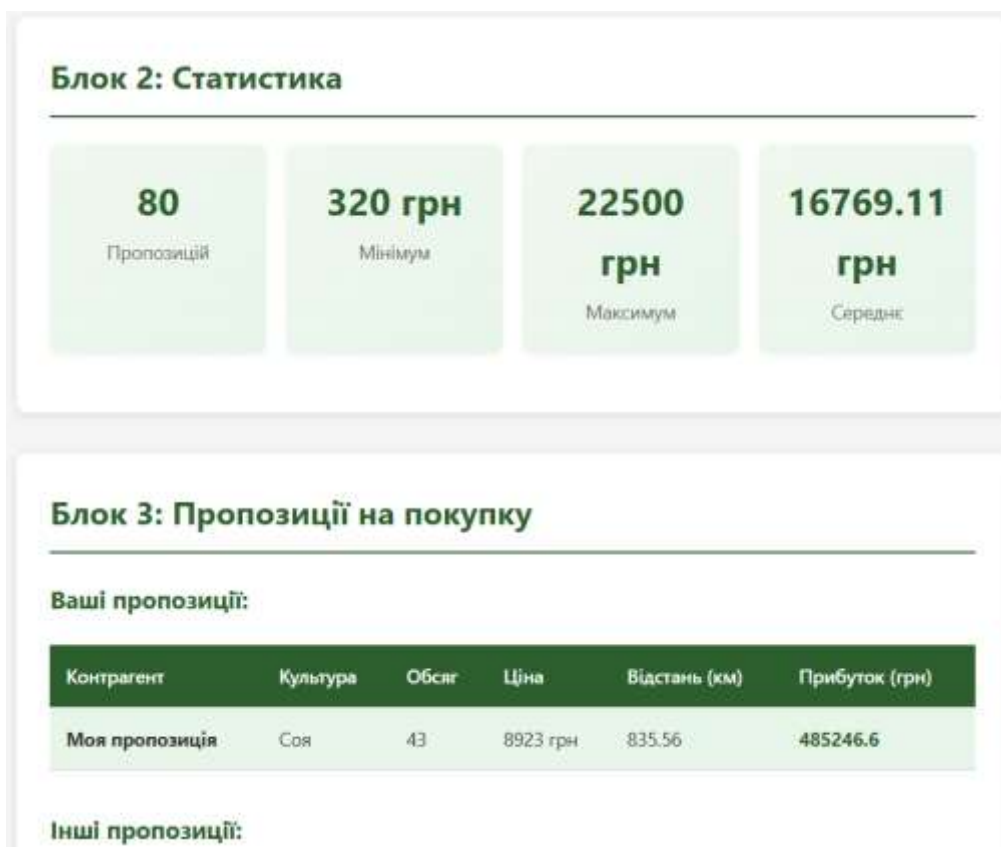


Рисунок 3.3.8. Демонстрація сайту у меншому вікні

3.4 Тестування

Стратегія тестування

Для перевірки коректності роботи системи застосовано метод наскрізного тестування, що імітує реальний сценарій використання системи від запуску серверної частини до отримання результатів у веб-інтерфейсі. Такий підхід дозволяє перевірити взаємодію всіх компонентів системи в їх робочому середовищі та виявити потенційні проблеми інтеграції, які можуть бути непомітні при ізольованому тестуванні окремих модулів. Тестування виконується локально через веб-інтерфейс з використанням тестових даних.

Тестовий сценарій

Мета тестування: перевірити повний цикл роботи системи від збору даних з веб-джерел до розрахунку прибутковості та відображення результатів для конкретної культури з урахуванням логістичних параметрів користувача.

Тестові дані:

- Культура для аналізу: Пшениця
- Локація користувача: Полтава

- Обсяг продукції: 100 тонн
- Доступний транспорт: Вантажівка до 20 тонн

Очікуваний результат: система має зібрати актуальні пропозиції закупівлі пшениці з чотирьох джерел, визначити координати всіх локацій, розрахувати відстані від Полтави до кожної пропозиції, обчислити логістичні витрати з урахуванням обраного транспорту, визначити прибутковість кожної пропозиції, відсортувати результати за прибутковістю та відобразити у зручному для аналізу форматі.

Етапи виконання тестування

Етап 1: Запуск серверної частини

Дія: Запуск файлу `backend.py` через інтерпретатор Python у терміналі командою `python backend.py`. (Рисунок 3.4.1)

Що відбувається:

- Завантаження курсу валют з API ПриватБанку з виводом у консоль актуального курсу долара
- Перевірка наявності збереженого файлу `prop_data.csv` з попередніми даними
- Якщо файл існує, завантаження даних з нього у пам'ять з повідомленням про кількість завантажених пропозицій
- Якщо файл відсутній, виконання повного циклу парсингу всіх чотирьох джерел з поетапними повідомленнями про процес
- Запуск фонового потоку для автоматичного оновлення даних кожні 24 години
- Запуск веб-сервера на порту 5000 з повідомленням про готовність приймати запити

Що перевіряється:

- Успішне підключення до API ПриватБанку
- Коректне завантаження збережених даних або виконання парсингу
- Відсутність критичних помилок при запуску
- Доступність веб-сервера на вказаному порту

```

PS D:\NAZa\Yniver\4 курс\Дипломна\AllSite> python backend.py
Курс USD: 44.44444 грн
=====
ФЕРМЕРСЬКИЙ КАЛЬКУЛЯТОР
=====

Завантаження збережених даних...
Завантаження prop_data.csv...
    Завантажено 916 пропозицій

=====
Сервер запущено: http://localhost:5000
=====

```

Рисунок 3.4.1. Запуск файлу backend.py

Етап 2: Відкриття веб-інтерфейсу

Дія: перехід за адресою <http://localhost:5000>. (Рисунок 3.4.2)

Що відбувається:

- Завантаження HTML-розмітки сторінки
- Підключення бібліотеки Leaflet для відображення карти
- Ініціалізація інтерактивної карти з центром на Україні
- Відображення форм введення параметрів
- Підготовка обробників подій для взаємодії користувача

Що перевіряється:

- Коректне відображення всіх елементів інтерфейсу
- Завантаження та відображення карти
- Доступність випадаючого списку з культурами
- Відсутність помилок у консолі розробника браузера

- Паралельно відправляється другий запит GET до `/api/proposals?culture=Пшениця`
- Отримані пропозиції відображаються у таблиці нижче форми

Що відбувається при введенні кількості:

- Значення 100 зберігається у полі форми з типом `number`
- Якщо користувач намагається ввести від'ємне число, HTML-валідація з атрибутом `min="0"` блокує введення
- Дані зберігаються у локальне сховище браузера через `localStorage` для збереження між сесіями

Що відбувається при кліку на карті:

- Обробник події перехоплює координати точки з об'єкта події через властивості `e.latlng.lat` та `e.latlng.lng`
- Якщо на карті вже існує попередній маркер, він видаляється методом `map.removeLayer()`
- Створюється новий маркер у точці кліку через `L.marker()` та додається на карту методом `addTo(map)`
- Координати зберігаються у глобальні змінні `userLat` та `userLon` для подальшого використання у розрахунках
- Координати формуються методом `toFixed(6)` для точності до шести знаків після коми
- Відформатовані координати відображаються у текстовому блоці під картою у форматі "широта, довгота"
- Дані автоматично зберігаються у `localStorage` для збереження при наступному відвідуванні

Що відбувається при додаванні машини:

- Відкривається окрема форма для введення параметрів транспорту
- Користувач вводить характеристики транспортного засобу: вантажопідйомність у тоннах, тариф оплати за кілометр у гривнях
- Натискання кнопки "Додати машину" валідує введені дані на коректність
- Створюється об'єкт транспорту з полями `capacity` (вантажопідйомність) та `rate` (тариф)
- Об'єкт додається до масиву `vehicles` у пам'яті браузера
- Інформація про додану машину відображається у списку доступного транспорту
- Користувач може додати декілька різних транспортних засобів
- При розрахунку система автоматично обере найвигідніший варіант з доданих машин

Що перевіряється на цьому етапі:

- Коректна робота випадаючого списку культур
- Обчислення статистичних показників
- Коректне введення числових значень у поле обсягу
- Працездатність валідації поля на додатність
- Визначення координат при кліку на карті
- Видимість маркера на карті після встановлення
- Правильність відображення координат у текстовому блоці
- Збереження всіх введених даних у локальне сховище
- Коректна робота форми додавання транспорту
- Відображення доданих машин у списку доступного транспорту (Рисунок 3.4.7)

Тип товару:

Соя

Оберіть культуру

Соняшник

Кукурудза

Пшениця

Соя

Ячмінь

Ріпак

Горіх

Овес

Гречка

Цукровий буряк

Рисунок 3.4.3. Список культур



Рисунок 3.4.4. Статистика пропозицій

Блок 1: Ваші дані

Тип товару:

Пшениця

Кількість товару (тонн):

100

Місцезнаходження товару:

Введіть адресу або клікніть на карту



Координати: 49.591218, 34.447906

*Рисунок 3.4.5. Введені данні користувачем***Додати машину**

Вантажопідйомність (тонн):

20

Ставка за км (грн):

15

Зберегти машину

Скасувати

Рисунок 3.4.6. Форма додавання машини

Ваші машини:

Машини 1: 20 т, 15 грн/км

Виділити

Додати машину

*Рисунок 3.4.7. Список машин користувача***Етап 4: Виконання розрахунку прибутковості**

Дія: Натискання великої зеленої кнопки "Розрахувати прибутковість".

Що відбувається на клієнті:

- Виконується валідація всіх обов'язкових полів форми. Перевіряється наявність координат користувача, обсягу продукції, обраної культури
- Якщо валідація не пройдена, відображається повідомлення про помилку
- Збираються всі параметри у JSON-об'єкт з полями lat, lon, volume, culture, vehicles
- Відображається індикатор завантаження для інформування користувача про процес обробки (Рисунок 3.4.9)
- Відправляється POST-запит до ендпоінту /api/calculate з JSON-тілом

Що відбувається на сервері:

- Приймається та парситься JSON-тіло запиту
- Витягуються параметри координат, обсягу, культури, транспорту
- Виконується валідація обов'язкових параметрів з поверненням помилки при їх відсутності
- З глобального списку prop_data створюються копії всіх пропозицій
- Фільтруються пропозиції за культурою через перевірку входження рядка без урахування регістру
- Логується кількість знайдених пропозицій

Що перевіряється:

- Коректність валідації введених даних
- Швидкість виконання розрахунків для всіх пропозицій
- Геокодування локацій
- Точність розрахунку відстаней через OSRM (Рисунок 3.4.8)
- Коректність обчислення логістичних витрат
- Конвертація валюти (Рисунок 3.4.11)
- Точність розрахунку прибутку за формулою
- Коректність сортування результатів
- Відображення результатів у таблиці (Рисунок 3.4.10)

Розрахунок для пропозиції: Пшениця 4 кл. в Дніпропетровська обл.м. Самар
✓ OSRM розрахував відстань: 193.19 км, 181 хв

Розрахунок для пропозиції: Пшениця 4 кл. в Полтавська обл.сmt. Диканька
✓ OSRM розрахував відстань: 10.25 км, 17 хв

Розрахунок для пропозиції: Пшениця 4 кл. в Хмельницька обл.сmt. Летичів
✓ OSRM розрахував відстань: 678.17 км, 569 хв

Розрахунок для пропозиції: Пшениця 2 кл. в Київська обл.м. Сквир
✓ OSRM розрахував відстань: 334.69 км, 259 хв

Розрахунок для пропозиції: Пшениця тверда в Київська обл.м. Сквир
✓ OSRM розрахував відстань: 334.69 км, 259 хв

Рисунок 3.4.8. Обчислення прибутковості у терміналі



Рисунок 3.4.9. Інформування користувача про процес обробки

Блок 3: Пропозиції на покупку

Дата	Контрагент	Культура	Обсяг	Ціна	Локація	Відстань (км)	Прибуток (грн)	Джерело
22.05.2026	МКО ТРАНС-СЕРВІС	Пшениця тверда	-	12800 грн	Київська обл.Київ	334.69	1229796.11	agrotender.com.ua
22.05.2026	Скви́рський комбінат хлібопродуктів	Пшениця тверда	-	11800 грн	Київська обл.м. Сквир	334.69	1129796.11	agrotender.com.ua
22.05.2026	Трейдер з agrotender	Пшениця тверда	-	11800 грн	Одеський МППЗП	558.21	1096267.89	agrotender.com.ua
22.05.2026	УКРАЇНЬКА АГРОПРОМИСЛОВА ТРАНСПОРТНА ГРУПА	Пшениця тверда	-	11000 грн	Київська обл.с.Русанів,Броварський р-н	325.28	1051287.34	agrotender.com.ua
22.05.2026	Трейдер з agrotender	Пшениця 2 кл.	-	11200 грн	Одеський МПБерукіне-Київ	558.21	1036867.89	agrotender.com.ua
22.05.2026	Агропроектис (NCH)	Пшениця 2 клас	-	11200 грн	ПС-МічудівкаОдеськая, Лиманський	558.21	1036267.89	trpobland
22.05.2026	ПРОДНВЕСТ СЕРВІС	Пшениця 2 кл.	-	11200 грн	Південний МП	558.21	1036267.89	agrotender.com.ua
22.05.2026	ТАС АГРО ТРЕЙД	Пшениця 2 кл.	-	11200 грн	Південний МП"ДС-Зерно"	558.21	1036267.89	agrotender.com.ua

Рисунок 3.4.10. Готова таблиця пропозицій

22.05.2026	Трейдер з agrotender	Пшениця тверда	-	245 дол (10889 грн)	Одеський МППЗП	558.21	1005156.67	agrotender.com.ua
------------	----------------------	----------------	---	---------------------	----------------	--------	------------	-------------------

Рисунок 3.4.11. Обчислена пропозиція у доларах

Етап 5: Додавання власної пропозиції

Дії:

- Натискання кнопки "Додати свою пропозицію".
- Заповнення форми з полями власної пропозиції. (Рисунок 3.4.12)
- Натискання кнопки "Зберегти пропозицію".

Що відбувається:

- Відкривається модальне вікно з формою додавання пропозиції
- Ініціалізується друга карта у модальному вікні для вибору локації
- Користувач заповнює поля ціни та обсягу
- Клік на карті встановлює маркер та зберігає координати
- При натисканні "Зберегти" формується JSON-об'єкт з усіма даними
- Відправляється POST-запит до `/api/add_offer` з пропозицією
- Сервер додає пропозицію до глобального списку `user_offers`
- Модальне вікно закривається
- Автоматично виконується повторний розрахунок для відображення власної пропозиції у результатах

Що перевіряється:

- Коректна робота модального вікна
- Функціонування другої карти для вибору локації
- Валідація введених даних у формі
- Успішне додавання пропозиції через API
- Автоматичне оновлення таблиці результатів
- Виділення власної пропозиції іншим кольором фону (світло-зелений) (Рисунок 3.4.14)
- Правильність розрахунків для власної пропозиції

Нова пропозиція на покупку

Ціна за тону (грн):

9500

Кількість товару (тонн):

100

Місцезнаходження:

Введіть адресу



Координати: 49.405076, 31.954404

Рисунок 3.4.12. Форма додавання користувачем пропозиції

Ваші пропозиції на покупку:

Пшениця: 9500 грн/т, 100 т.

Відалити

Додати пропозицію

Рисунок 3.4.13. Список пропозицій

Блок 3: Пропозиції на покупку

Ваші пропозиції:

Контрагент	Культура	Обсяг	Ціна	Відстань (км)	Прибуток (грн)
Моя пропозиція	Пшениця	100	9500 грн	235.55	914667.21

Інші пропозиції:

Дата	Контрагент	Культура	Обсяг	Ціна	Локація	Відстань (км)	Прибуток (грн)	Джерело
22.05.2026	МКО ТРАНС-СЕРВІС	Пшениця тверда	-	12000 грн	Київська обл. Київ	334.69	1229796.11	agrotender.com.ua
22.05.2026	Селянський клубініт хлібопродуктів	Пшениця тверда	-	11800 грн	Київська обл. м. Остер	334.69	1129796.11	agrotender.com.ua
22.05.2026	Трейдер з agrotender	Пшениця тверда	-	11800 грн	Одеський МПЗ	558.21	1096267.89	agrotender.com.ua
	УКРАЇНСЬКА АГРОПРОМИСЛОВА	Пшениця						agrotender.com.ua

Рисунок 3.4.14. Обчислена пропозиція користувача

Результати тестування

№	Етап тестування	Результат
1	Запуск backend.py	Пройдено
2	Відкриття frontend.html	Пройдено
3	Введення даних	Пройдено
4	Розрахунок прибутковості	Пройдено
5	Відображення результатів	Пройдено
6	Додавання власної пропозиції	Пройдено
7	Перевірка валідації	Пройдено
8	Точність розрахунків	Пройдено
9	Збереження в localStorage	Пройдено

Таблиця 3.1 Результати тестування

Під час тестування було виявлено наступні проблеми:

- При великій кількості пропозицій (більше 200) розрахунок займав понад 10 секунд
- Нестандартні назви локацій з деяких джерел
- Розрахунок пропозицій починається при кожній зміні даних користувачем, що збільшує навантаження на сервер.

ВИСНОВКИ

У даній бакалаврській кваліфікаційній роботі досліджено проблему автоматизації збору та агрегації інформації про пропозиції закупівлі сільськогосподарської продукції з різних електронних джерел та розроблено веб-орієнтовану інформаційну систему для вирішення цієї задачі.

Виконано аналіз предметної області, що включав дослідження десяти найпопулярніших сільськогосподарських культур України за посівними площами. Встановлено, що пшениця, соняшник та кукурудза разом займають понад 64% всіх посівних площ та генерують найбільшу кількість торгових пропозицій.

Проведено аналіз способів реалізації продукції через різні канали: трейдери, елеватори, переробники, електронні біржі, державні закупівлі. Виявлено, що значна частина угод відбувається в тіньовому секторі без публікації пропозицій, що обмежує можливості автоматизованого аналізу офіційним сегментом ринку. Встановлено, що найбільш прозорими джерелами є електронні торгові платформи та система Prozorro.

Виконано аналіз існуючих інформаційних платформ (UkrAgroConsult, APK-Inform, Zernotorg, GrainTrade UA, Latifundist). Виявлено, що аналітичні сервіси надають моніторинг цін, але не агрегують конкретні пропозиції, а торгові майданчики публікують пропозиції, але не об'єднують дані з різних джерел. Це обґрунтовує необхідність розробки системи автоматизованої агрегації.

Проведено детальний аналіз десяти потенційних джерел даних для автоматизованого збору (Prozorro, Zernotorg, GrainTrade, Propozitsiya, Agrotender, AgroUkraine, VAS.trade, GlenDeal) за критеріями актуальності, доступності, типу пропозицій та можливості парсингу. Встановлено, що переважна більшість досліджених платформ не надає публічних API для програмного доступу до даних: ні Zernotorg.ua, ні Propozitsiya.com, ні GrainTrade UA, ні більшість інших аграрних торгових майданчиків не реалізували відкритих інтерфейсів для машинного зчитування інформації. Винятком серед відібраних джерел є Prozorro, що надає повноцінний відкритий API. Відсутність API на комерційних платформах обумовлює необхідність застосування технологій веб-скрапінгу як єдиного практично доступного методу автоматизованого збору даних.

Обґрунтовано вибір технологічного стеку через порівняльний аналіз альтернативних інструментів: BeautifulSoup обрано для парсингу через простоту адаптації під зміни структури сайтів; Requests для HTTP-комунікації через достатню продуктивність; Flask для backend через баланс простоти та функціональності; SQLite для БД через легкість розгортання.

Розроблено архітектуру системи за трирівневою моделлю з модульною структурою, де кожне джерело даних має окремий парсер. Це забезпечує незалежність компонентів та можливість легкого додавання нових джерел без зміни значної частини логіки системи.

Реалізовано функціонуючий прототип системи з модулями парсингу для чотирьох джерел, модулем агрегації даних, REST API та веб-інтерфейсом. Проведено комплексне тестування, яке показало успішність парсингу.

Практична цінність роботи полягає у створенні інструменту, що автоматизує трудомісткий процес моніторингу множинних джерел пропозицій. Система дозволяє фермерам за кілька секунд отримати зведену інформацію про доступні пропозиції закупівлі замість багатогодинного ручного пошуку.

Напрямки подальшого розвитку: розширення переліку джерел (GrainTrade UA, Agrotender); впровадження інтелектуального аналізу для виявлення найвигідніших пропозицій з урахуванням якості та логістики; додавання системи сповіщень про нові пропозиції; розробка мобільного додатку; інтеграція з системами обліку фермерських господарств.

Розроблена система демонструє ефективність технологій веб-скрапінгу для вирішення практичних задач агрегації розпорошеної ринкової інформації та може бути впроваджена у діяльність малих та середніх фермерських господарств для підвищення ефективності пошуку каналів збуту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Міністерство аграрної політики та продовольства України. Офіційний веб-сайт. URL: <https://minagro.gov.ua/> (дата звернення: 24.04.2026).
2. Державна служба статистики України. Сільське господарство України 2025. URL: <https://stat.gov.ua/> (дата звернення: 04.05.2026).
3. Prozorro. Система електронних державних закупівель. URL: <https://prozorro.gov.ua/> (дата звернення: 21.04.2026).
4. Prozorro API Documentation. URL: <https://api-docs.prozorro.gov.ua/> (дата звернення: 30.04.2026).
5. Zerotorg.ua. Торгова платформа для зернових культур. URL: <https://zerotorg.ua/uk/> (дата звернення: 06.05.2026).
6. Propozitsiya.com. Аграрний інформаційний портал. URL: <https://propozitsiya.com/> (дата звернення: 01.05.2026).
7. UkrAgroConsult. Аналітика аграрних ринків. URL: <https://ukragroconsult.com/> (дата звернення: 06.05.2026).
8. APK-Inform. Інформаційно-аналітичне агентство. URL: <https://www.apk-inform.com/uk/> (дата звернення: 20.04.2026).
9. GrainTrade UA. Електронна торгова платформа. URL: <https://graintrade.com.ua/> (дата звернення: 13.04.2026).
10. Latifundist. Аналітичний портал агробізнесу. URL: <https://latifundist.com/> (дата звернення: 27.04.2026).
11. Beautiful Soup Documentation. Beautiful Soup 4.12.0 documentation. URL: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (дата звернення: 13.04.2026).
12. Python Requests. HTTP for Humans. URL: <https://requests.readthedocs.io/> (дата звернення: 05.05.2026).
13. Flask Documentation. Welcome to Flask. URL: <https://flask.palletsprojects.com/> (дата звернення: 03.05.2026).
14. SQLite. SQL Database Engine. URL: <https://www.sqlite.org/> (дата звернення: 08.05.2026).
15. Scrapy Documentation. URL: <https://docs.scrapy.org/> (дата звернення: 10.05.2026).
16. Mitchell R. Web Scraping with Python: Collecting More Data from the Modern Web. O'Reilly Media, 2018. 306 p.
17. Українська зернова асоціація. Аналіз ринку зерна 2025. URL: <https://uga.ua/> (дата звернення: 07.05.2026).

18. Agrotender. Тендерна платформа. URL: <https://agrotender.com.ua/> (дата звернення: 07.05.2026).
19. VAS Trade. Торгова платформа сільгосппродукції. URL: <https://ua.vas.trade/> (дата звернення: 19.04.2026).
20. Всеукраїнська аграрна рада. Позиційні документи. URL: <https://var.org.ua/> (дата звернення: 03.05.2026).
21. Tripoli. Агротехнологічна платформа. URL: <https://tripoli.land/> (дата звернення: 10.05.2026).
22. GlenDeal. Торгова платформа агропродукції. URL: <https://glendeal.com/> (дата звернення: 06.05.2026).
23. Agro-Ukraine. Платформа для торгівлі агропродукцією. URL: <https://agro-ukraine.com/> (дата звернення: 30.04.2026).
24. BuyMyCorn. Платформа купівлі зерна. URL: <https://buymycorn.com/> (дата звернення: 24.04.2026).
25. Vegoil Club. Аналітика ринку олійних культур. URL: <https://vegoil.club/> (дата звернення: 07.05.2026).
26. SuperAgronom. Новини та аналітика врожаїв. URL: <https://superagronom.com/> (дата звернення: 11.05.2026).
27. Prozorro BI. Аналітика державних закупівель. URL: <https://bi.prozorro.org/> (дата звернення: 20.04.2026).
28. Кабінет Міністрів України. Посівні площі 2025 року. URL: <https://www.kmu.gov.ua/> (дата звернення: 28.04.2026).
29. Верховна Рада України. Новини Комітету з питань аграрної політики. URL: https://www.rada.gov.ua/news/news_kom/258168.html (дата звернення: 05.05.2026).
30. Django Documentation. Django Web Framework. URL: <https://docs.djangoproject.com/> (дата звернення: 07.05.2026).
31. FastAPI Documentation. URL: <https://fastapi.tiangolo.com/> (дата звернення: 14.05.2026).
32. PostgreSQL. The World's Most Advanced Open Source Relational Database. URL: <https://www.postgresql.org/> (дата звернення: 05.05.2026).
33. Selenium Documentation. URL: <https://www.selenium.dev/documentation/> (дата звернення: 04.05.2026).
34. httpx Documentation. URL: <https://www.python-httpx.org/> (дата звернення: 07.05.2026).
35. aiohttp Documentation. URL: <https://docs.aiohttp.org/> (дата звернення: 20.04.2026).

Додаток А

Посилання на репозиторій: <https://github.com/cherven1302-ops/diplom>

Додаток Б

Узагальнений вигляд веб-інтерфейсу

Фермерський калькулятор

Блок 1: Ваші дані

Тип товару:

Соя

Кількість товару (тонн):

20

Місцезнаходження товару:

Введіть адресу або клікніть на карту

Координати: 48.436490, 33.998291

Ваші машини:

Машини 1: 20 т, 15 грн/км

Виділити

Додати машину

Ваші пропозиції на покупку:

Соя: 31664 грн/т, 14 т.

Виділити

Додати пропозицію

Блок 2: Статистика

80

Пропозицій

320 грн

Мінімум

22500 грн

Максимум

16768.84 грн

Середнє

Блок 3: Пропозиції на покупку

Дата	Компанія	Культура	Обсяг	Ціна	Локація	Відстань (км)	Прибуток (грн)	Джерело
24.05.2026	МКО ТРАНС-СЕРВІС	Соя	—	22500 грн	Дніпропетровська обл.Марганець	104.69	446859.16	agrotender.com.ua
24.05.2026	ЮІВ ОЛІМП	Соя	—	22200 грн	Дніпропетровська обл.Лозуватка	104.69	440859.16	agrotender.com.ua
24.05.2026	ТОВ ФЕРМЕРІВ І ПРИВАТНИХ ЗЕМЛЕВЛАСНИКІВ	Соя	—	22000 грн	Дніпропетровська обл.Дніпровський район	104.69	436859.16	agrotender.com.ua

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) Інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

ДЕКЛАРАЦІЯ ЗДОБУВАЧА

Я, Чернець Назарій Дмитрович, здобувач НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота (проект) на тему «Інформаційна система оптимізації накладних витрат закупівлі та реалізації сільськогосподарської продукції» 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:

○ [ChatGPT, Gemini, Claude, Qwen] інструменти ШІ були використані для:

- [] перекладу іншомовних джерел;
- [] стилістичного редагування та перевірки граматики;
- [] допомоги у написанні/відлагодженні програмного коду;
- [] допомоги знаходження джерел інформації;

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

«__» _____ 2026р.

_____ (підпис)

_____ (Ім'я та ПРІЗВИЩЕ)