ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Проєктування комерційного вебресурсу підприємства з реалізації сільськогосподарської техніки»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_31[1]
Кривий Ілля Русланович
Керівник: Одарущенко Олена Борисівна
Рецензент: Стрюк Олексій Юрійович

Полтава – 2026 року

Навчально-науковий інститут економіки, управління, права та інформаційних технологій
Кафедра інформаційних систем та технологій

Освітня програма Інформаційні управляючі системи
Спеціальність 126 Інформаційні системи та технології
Рівень вищої освіти перший (бакалаврський)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Юрій УТКІН
«10» червня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Кривого Іллі Руслановича

1. Тема кваліфікаційної роботи:
«Проєктування комерційного вебресурсу підприємства з реалізації сільськогосподарської техніки»,
Керівник роботи: к. т. н., доцент, доцент кафедри інформаційних систем та технологій Одарущенко Олена Борисівна.
Затверджено наказом закладу вищої освіти від «10» квітня 2026 року № 384-ст

2. Строк подання здобувачем вищої освіти роботи «08» червня 2026 року.

3. Вихідні дані до роботи: стандарти проєктування та розробки вебдодатків (HTML5, CSS3, ECMAScript), офіційна технічна документація з сайтів мов програмування та середовищ розробки, дані офіційних сайтів компаній-конкурентів, аналітичні дані порталів https://agravery.com та https://agroportal.ua, статистична інформація Державної служби статистики України та Державного аграрного реєстру.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Розділ 1. Дослідження та аналіз предметної області.
Розділ 2. Проєктування вебресурсу.
Розділ 3. Розробка вебресурса.

5. Перелік графічного матеріалу: схеми, рисунки, діаграми за темою та об’єктом дослідження

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
Оцінювання економічної ефективності результатів дослідження	Загребельна І. Л., к. е. н., доцент, доцент кафедри економіки та публічного управління	01.04.2026 р.	29.05.2026 р.

7. Дата видачі завдання «10» червня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Вибір і затвердження теми роботи	02.06.2025 р. – 04.06.2025	
2	Складання і затвердження розгорнутого плану та завдання на кваліфікаційну роботу	05.06.2025 р. – 10.06.2025 р.	
3	Опрацювання джерел інформації	11.06.2025 р. – 15.06.2025 р.	
4	Збір, вивчення і обробка інформації, необхідної для виконання роботи	16.06.2025 р. – 20.06.2025 р.	
5	Виконання теоретико-методологічного розділу роботи	08.09.2025 р. – 01.11.2025 р.	
6	Виконання дослідницько-аналітичного розділу роботи	19.01.2026 р. – 14.02.2026 р.	
7	Виконання проєктно-рекомендаційного розділу роботи	16.02.2026 р. – 31.03.2026 р.	
8	Оцінювання економічної ефективності результатів дослідження	01.04.2026 р. – 29.05.2026 р.	
9	Оформлення тексту роботи	01.06.2026 р. – 06.06.2026р.	
10	Попередній захист роботи на кафедрі	08.06.2026 р.	
11	Доопрацювання роботи з урахуванням зауважень і пропозицій	09.06.2026 р. – 10.06.2026 р.	
12	Нормоконтроль	11.06.2026 р. – 15.06.2026 р.	
13	Захист кваліфікаційної роботи	16.06.2026 р. - 18.06.2026 р.	

Здобувач вищої освіти **Ілля КРИВИЙ**

Керівник роботи **Олена ОДАРУЩЕНКО**

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

HTML – Hypertext markup language

CSS – Cascading Style Sheets

JS – Java Script

UML – Universal Modeling Language

SQL – Structured query language

API – Application programming interface

URL – Uniform Resource Locator

JSON – JavaScript Object Notation

СУБД – Система управління базами даних

ВСТУП

Сільськогосподарський сектор України впродовж останніх десятиліть пройшов шлях від переважно ручної праці до широкого використання складної спеціалізованої техніки. Підвищення ефективності польових робіт, скорочення витрат і забезпечення конкурентоспроможності господарств на ринку стають можливими лише при систематичному оновленні машинного парку. Попит на сільськогосподарську техніку в Україні залишається стабільно високим: щороку аграрні підприємства різних форм власності закуповують тисячі тракторів, зернозбиральних комбайнів, обприскувачів та іншого обладнання.

Разом із тим стрімкий розвиток інформаційних технологій принципово змінив модель прийняття рішень про купівлю техніки. Якщо раніше більшість угод укладалася на галузевих виставках або в результаті особистих переговорів, то нині значна частина покупців здійснює попереднє дослідження ринку через інтернет задовго до будь-якого прямого контакту з продавцем. Потенційний покупець вивчає технічні характеристики моделей, порівнює пропозиції різних постачальників і формує коло вподобань уже на етапі онлайн-пошуку. Відповідно, компанія, що не має якісного цифрового представництва, фактично відсутня для значної частини своєї потенційної аудиторії.

Вебресурс постачальника агротехніки перестав бути просто електронною візиткою з переліком телефонів і адресою. Він має забезпечувати повний цикл взаємодії між постачальником і покупцем: від ознайомлення з каталогом і порівняння характеристик – до оформлення замовлення, подачі заявки на гарантійне обслуговування і отримання оперативного зворотного звязку. Компанії, що не реалізують цей повний цикл онлайн, поступово втрачають позиції навіть за умови конкурентної якості самої техніки..

Актуальність теми. Враховуючи наявну у відкритому доступі аналітичну інформацію, представлену у галузевих дослідженнях [1-2], станом на 2026 рік обсяг продажів нової техніки збільшився на 35% у порівнянні з аналогічним періодом 2025 року. Таке стрімке збільшення попиту свідчить про значні можливості для

швидкого виходу нових гравців на ринок, де якісний вебресурс може стати ключовим інструментом для збільшення обсягів продажів і охоплення більшої кількості клієнтів. Варто зауважити, що на даний момент лише 38% сайтів постачальників агротехніки реально адаптовані під сучасні стандарти зручності користувачів і відповідають актуальним вимогам до юзабіліті.

Мета роботи – провести комплексне дослідження теоретичних основ та практичних аспектів проєктування комерційних вебресурсів у сфері реалізації сільськогосподарської продукції. Створити детальну концептуальну модель та функціональну архітектуру вебресурсу для підприємства, що спеціалізується на реалізації сільськогосподарської техніки та супутнього обладнання.

Об'єкт дослідження – процес проєктування комерційних вебресурсів для підприємств, що займаються реалізацією сільськогосподарської техніки.

Предмет дослідження кваліфікаційної роботи – сучасні методи, актуальні технології та спеціалізовані інструменти проєктування функціональної архітектури вебплатформ, призначених для організації торгівлі агротехнікою.

Методи дослідження – системний аналіз ринкового середовища та конкурентного оточення (розділ 1), концептуальне моделювання бізнес-процесів (розділ 2), практичне проєктування архітектури системи (розділ 3).

Інформаційна база, що використовувалася: публікації в періодичних виданнях, веб-ресурси.

Практична значущість: спроєктовано та розроблено вебресурс для реалізації сільськогосподарської техніки. Сформовано універсальний робочий підхід до проєктування та створення вебресурсів згідно існуючих рекомендацій для подальшого використання.

Робота складається зі вступу, трьох розділів, висновків та списку літератури. Обсяг роботи становить 51 сторінку, 4 таблиці, 19 рисунків.

РОЗДІЛ 1
ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис об’єкту дослідження

Сільськогосподарська техніка є однією з найбільш капіталомістких категорій промислових товарів. Вартість сучасного трактора середнього класу потужності сягає декількох мільйонів гривень, а зернозбиральний комбайн нового покоління обходиться від п’яти до двадцяти і більше мільйонів. За таких умов рішення про купівлю ніколи не є спонтанним і завжди передбачає тривалий етап дослідження ринку: вивчення технічних характеристик, порівняння пропозицій, отримання консультацій, оцінку умов гарантії і сервісного обслуговування. Саме цей підготовчий процес все частіше переміщується в онлайн-простір.

Проєктування комерційного вебресурсу для реалізації агротехніки є комплексним завданням, що охоплює кілька взаємопов’язаних підсистем. По-перше, це публічна частина сайту: каталог продукції з можливістю пошуку та фільтрації, детальні картки товарів з технічними специфікаціями і фотоматеріалами, механізм оформлення замовлень, форми зворотного звязку і подачі заявок. По-друге, адміністративний інтерфейс, що дозволяє співробітникам компанії без технічних знань оперативно оновлювати асортимент, редагувати ціни і стежити за вхідними замовленнями. По-третє, серверна частина і база даних, що забезпечують надійне зберігання інформації і коректну роботу всього функціоналу.

Враховуючи, що на сьогоднішній день комерційні вебресурси остаточно перестали виконувати роль простої електронної візитки для компаній, а перетворилися на повноцінні розвинуті інформаційні системи з широким функціоналом, вебресурс має відповідати декільком критично важливим вимогам:

– забезпечувати зручний та інтуїтивно зрозумілий інтерфейс для максимального комфорту користувацького досвіду незалежно від рівня технічної підготовки відвідувачів;

– підтримувати повноцінну сумісність та безшовну інтеграцію з корпоративними ERP-системами, CRM-платформами та різноманітними зовнішніми сервісами через API;

– забезпечувати ефективний збір, систематизацію та якісну обробку великих масивів інформації про клієнтів та товари;

– надавати високорівневий та зручний пошук товарів за множинними категоріями з можливістю гнучкого фільтрування;

– нарантувати постійне оновлення актуальної інформації та забезпечувати оперативну підтримку клієнтів через різноманітні канали комунікації.

В процесі проєктування вебресурсу для реалізації сільськогосподарської техніки критично важливо враховувати специфічні особливості роботи агропромислового комплексу України. До таких особливостей належить необхідність роботи з великими обсягами детальних технічних даних та специфікацій, врахування виражених сезонних коливань навантаження на серверну інфраструктуру в період з кінця зимового сезону до середини осені, коли відбувається пік активності сільськогосподарських робіт.

Таким чином, можна дійти висновку, що проєктування комерційного вебресурсу для реалізації сільськогосподарської техніки охоплює не лише суто технічні аспекти розробки, а й економічні складові ефективності, маркетингові стратегії просування та логістичні компоненти організації процесу продажу. В таких специфічних умовах вебресурс має забезпечити максимально високу ефективність комерційної діяльності підприємства.

Основною метою даної кваліфікаційної роботи є створення повнофункціонального комерційного сайту для компанії, яка спеціалізується на реалізації широкого асортименту сільськогосподарської техніки та супутнього обладнання. Розроблений вебресурс має наочно демонструвати потенційним клієнтам можливості підприємства у сфері надання послуг з вибору, консультування та постачання сільськогосподарської техніки і обладнання різноманітного призначення.

У межах проведеного дослідження ми будемо комплексно розглядати об'єкт проєктування у частині організації бізнес-процесів, вимог до програмно-апаратного забезпечення і специфічних потреб різних категорій користувачів системи.

1.2 Аналіз сайтів за схожим спрямуванням

Для якісного проєктування сучасного комерційного сайту з реалізації сільськогосподарської техніки необхідно провести детальний порівняльний аналіз вебресурсів прямих конкурентів на ринку. Для проведення об'єктивного аналізу мною було вибрано два репрезентативні вебсайти провідних компаній галузі: Аякс Агро (https://ayax-agro.com.ua/ua/) і General Machines (https://general-machines.ua)

Головним завданням нашого системного аналізу є комплексне дослідження візуального оформлення та функціонального наповнення сайтів конкурентів з метою виявлення їх сильних та слабких сторін. В межах детального системного аналізу буде ретельно розглянуто рівень юзабіліті вебресурсів, ступінь комфорту роботи для різних категорій користувачів, особливості стилістичного оформлення і технічні можливості обраних для порівняння вебплатформ.

Веб сайт компанії АЯКС Агро являє собою повноцінний комерційний вебресурс з розвинутим функціоналом. Інтерфейс даного сайту представлений у витриманому мінімалістичному стилі і професійно виконаний у корпоративній стилістиці компанії з використанням фірмових кольорів та елементів айдентики. На ньому реалізований структурований каталог продукції з можливістю перегляду детальних характеристик, розділ з детальною інформацією про компанію та її історію. До суттєвих недоліків даного вебресурсу можна обґрунтовано віднести повну відсутність можливості створення персонального кабінету користувача для відстеження історії замовлень та збереження обраних товарів. Зовнішній вигляд головної сторінки вебсайту Аякс Агро наочно продемонстровано на рисунку 1.1.

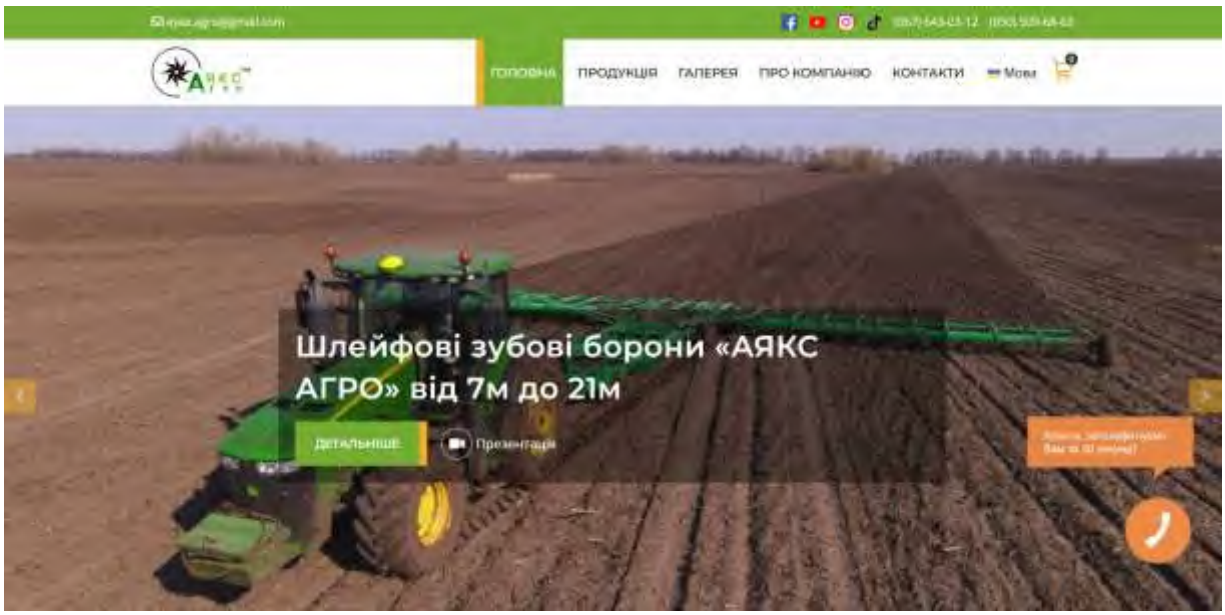

Рисунок 1.1 – Сайт Аякс Агро

Функціональні можливості аналізованого сайту, представлені в основному у вигляді каталогу товарів з різноманітним асортиментом, детально наведено на рисунку 1.2. Вебресурс пропонує користувачам великий обсяг різноманітної техніки і обладнання для різних сфер застосування у сільськогосподарському виробництві. До безперечних переваг даної платформи можна віднести унікальну можливість детально ознайомитися з широким рядом представленої техніки за допомогою інтерактивних мультимедійних презентацій та відео оглядів. Проте серед недоліків функціональних можливостей варто відзначити обмежену кількість критеріїв фільтрування товарів, що ускладнює швидкий пошук необхідного обладнання.

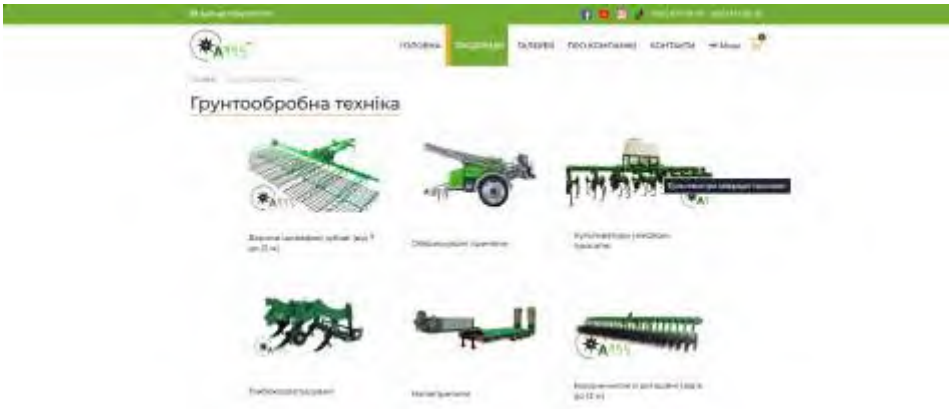

Рисунок 1.2 – Функціональні можливості сайту Аякс Агро

Наступним об'єктом детального дослідження став вебсайт компанії General Machines, що представляє альтернативний підхід до організації онлайн-продажів. Даний вебресурс демонструє витриманий мінімалістичний стиль оформлення, пропонує розширений каталог продукції з детальними описами, демонструє посилену орієнтацію на покращення користувацького досвіду і забезпечення якісного зворотного зв'язку з клієнтами через різноманітні канали комунікації. Вагомою перевагою даного вебресурсу є реалізована можливість організації доставки придбаної техніки провідними транспортними компаніями у будь-який регіон України, що значно збільшує зручність для клієнтів з віддалених областей. Зовнішній вигляд головної сторінки представлений на рисунку 1.3.

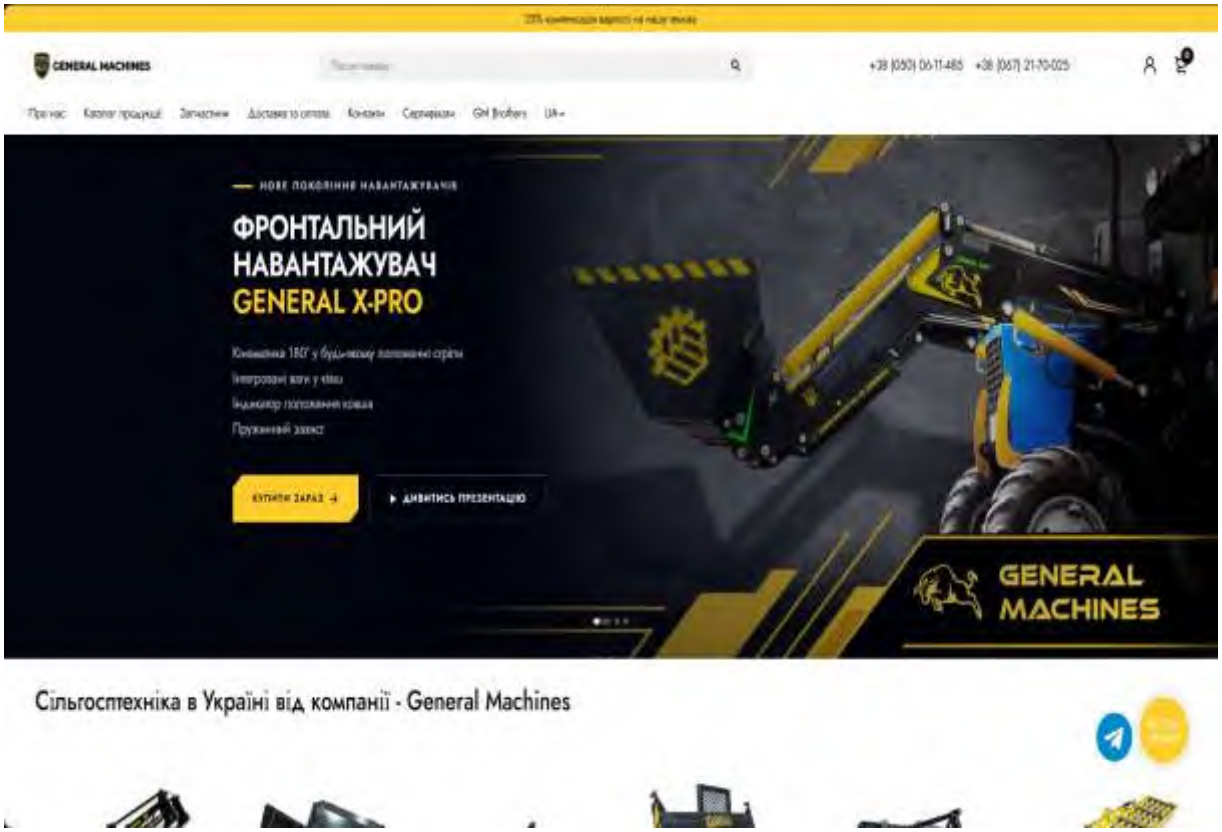

Рисунок 1.3 – Сайт general-machines

З точки зору функціональних можливостей на аналізованому сайті представлений добре структурований каталог товарів з логічною організацією категорій. Візуальне представлення каталогу сайту детально зображено на рисунку 1.4.

Рисунок 1.4 – Функціональні можливості сайту general-machines

Для проведення об'єктивного порівняльного аналізу досліджуваних вебресурсів були застосовані перевірені методи системного аналізу інформаційних систем. Узагальнені результати проведеного дослідження систематизовано та наочно продемонстровано в таблиці 1.1.

Таблиця 1.1 – Порівняння сайтів

Характеристики	Аякс Агро	General-machines
Стиль	Мінімалістичний	Мінімалістичний
Наявність особистого кабінету	-	+
Якість технічного контенту	Висока, Подана стисло	Середня, Подана текстом без виділення основних характеристик
Адаптація під мобільні пристрої	+	+
Навігація	Верхнє меню з вибором категорій	Класичне верхнє меню
Візуальний контент	Фото і демонстраційні відео	Фото і банери
Каталог продукції	Обмежений	Обмежений

Беручи до уваги всю досліджену та систематизовану інформацію, ми можемо зробити обґрунтовані висновки, що проаналізовані вебресурси конкурентів досить суттєво обмежені у своїх функціональних можливостях та не використовують весь потенціал сучасних веб-технологій. Недостатня кількість критеріїв фільтрування

товарів негативно впливає на загальний користувацький досвід взаємодії з сайтом і значно погіршує ефективність пошуку необхідного товару, що може призводити до втрати потенційних клієнтів.

1.3 Методології проєктування сучасних вебресурсів

Обґрунтований вибір оптимальної методології для проєктування комерційних вебресурсів є критично важливим етапом у загальному процесі розробки професійних сайтів. Кожен з існуючих підходів має свої характерні переваги і специфічні недоліки, які необхідно враховувати при виборі стратегії реалізації проєкту. Розглянемо детальніше основні методології, що застосовуються у сучасній веб-розробці:

1. Waterfall – це класична каскадна модель управління проєктами [3], яка протягом багатьох років залишається популярною у певних сегментах розробки програмного забезпечення. Дана методологія передбачає, що кожен наступний етап розробки слідує строго послідовно і має глибоку функціональну залежність від результатів попереднього етапу. Таким чином формується каскадний одно направлений рух вперед без можливості повернення, який становить концептуальну основу всієї моделі. Візуальне представлення принципу роботи методології Waterfall наочно продемонстровано на рисунку 1.5.

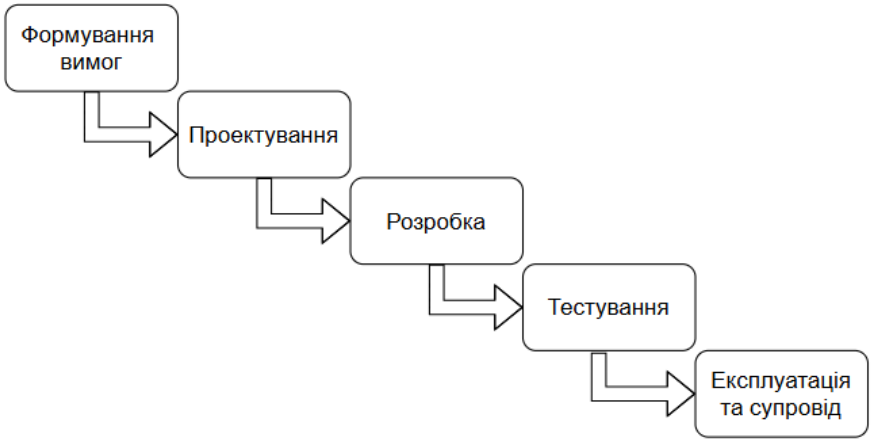

Рисунок 1.5 – Принцип роботи Waterfall

Основні послідовні етапи класичної моделі Waterfall включають наступні обов'язкові фази реалізації проєкту:

– основні послідовні етапи класичної моделі Waterfall включають наступні обов'язкові фази реалізації проєкту;

– комплексне проєктування загальної архітектури майбутньої системи з урахуванням всіх компонентів;

– безпосередня розробка програмного коду (етап активного програмування);

– всебічне тестування функціоналу на виявлення помилок та несправностей;

– поетапне введення системи в промислову експлуатацію та організація подальшого технічного супроводу.

Ключовою безперечною перевагою даної моделі є її висока структурованість процесів, абсолютна прозорість для всіх учасників проєкту і значна зручність довгострокового планування ресурсів. Вона дозволяє створити повну детальну проєктну документацію на самих ранніх етапах розробки, що суттєво полегшує ефективний контроль за дотриманням встановлених термінів та затвердженого бюджету проєкту. Водночас методологія має ряд суттєвих недоліків, серед яких особливо виділяється дуже низька гнучкість системи та значна складність внесення будь-яких змін після завершення етапу детального проєктування. Критичні помилки та недоліки, виявлені вже на етапі комплексного тестування, часто призводять до необхідності повернення на початкові стадії, що спричиняє значні непередбачені витрати часових та фінансових ресурсів.

2. Agile – це гнучка ітеративна та інкрементальна методологія підходу до розроблення програмного забезпечення різного рівня складності. Вона базується на фундаментальних принципах максимальної адаптивності до змін, швидкої оперативної реакції на нові вимоги замовника та тісної постійної співпраці всієї команди розробників з представниками бізнесу.

Основні концептуальні ідеї та принципи методології детально викладені у відомому Маніфесті Agile [4], який проголошує наступні пріоритети:

– люди та їх ефективна взаємодія є значно важливішими за формалізовані процеси та технічні інструменти розробки;

– реально працюючий функціональний програмний продукт є важливішим за створення вичерпної детальної документації;

– продуктивна співпраця із замовником на всіх етапах є важливішою за формальне узгодження умов контракту;

– гнучка готовність до прийняття змін є важливішою за жорстке дотримання початкового затвердженого плану.

Agile передбачає розбиття проєкту на короткі ітерації (спринти), після кожної з яких замовник отримує робочу версію продукту. Такий підхід особливо ефективний в умовах високої невизначеності вимог та швидкозмінного ринкового середовища.

Використовуючи методи системного аналізу мною було проведено порівняння даних методології, результати порівняння наведено у таблиці 1.2.

Таблиця 1.2 – Порівняння методології Waterfall та Agile

Критерій	Waterfall	Agile
Структура процесу	Лінійна, послідовна	Ітеративна, гнучка
Гнучкість до змін	Низька	Висока
Документація	Дуже детальна	Мінімально необхідна
Залучення замовника	Обмежене	Постійне
Ризик виявлення помилок	На пізніх етапах	На ранніх етапах
Придатність для проєктів	З чіткими та стабільними вимогами	З високою невизначеністю
Швидкість отримання результату	Низька	Висока
Прогнозованість бюджету	Висока	Середня

Таким чином для вебресурсу підприємства, що спеціалізується на реалізації сільськогосподарської техніки, найбільш оптимальним та обґрунтованим вибором буде свідоме використання гнучкої методології Agile. Традиційна методологія

Waterfall категорично не підходить для реалізації нашого проєкту, оскільки у випадку виявлення критичних помилок або необхідності змін на етапі розроблення вона вимагає практично повного перезапуску проєкту із самого початку, що є економічно недоцільним. На відміну від неї, сучасна методологія Agile надає унікальну можливість оперативно вносити необхідні зміни у будь-яку частину проєкту без глобального перероблення всієї системи, що критично важливо для динамічного ринку агротехніки.

РОЗДІЛ 2
ПРОЄКТУВАННЯ ВЕБРЕСУРСУ

2.1 Визначення цільової аудиторії та переліку вимог до вебресурса

Перед початком безпосереднього проєктування архітектури системи критично важливо чітко визначити цільову аудиторію, яка буде активно користуватися нашим комерційним вебресурсом для вирішення своїх бізнес-завдань. Основною категорією користувачів є різноманітні фермерські господарства України, які для зручності аналізу можна класифікувати на декілька сегментів: малі господарства з площею оброблюваних земель до 50 гектарів, середні господарства з площею від 50 до 500 гектарів і великі агрохолдинги з площею понад 500 гектарів орної землі. Для формування об'єктивної картини розподілу господарств була використана офіційна статистична інформація з державного порталу статистики "Держстат" (https://stat.gov.ua/) і дані Державного Аграрного Реєстру України (https://www.dar.gov.ua/), які надають достовірну інформацію про структуру аграрного сектору. Систематизовані результати проведеного дослідження детально наведено в таблиці 2.1.

Таблиця 2.1 – Сегменти цільової аудиторії

Параметр	Малі господарства	Середні господарства	Великі господарства
Розмір господарства	До 50 га	50–500 га	Більше 500 га
Основний пристрій	Смартфон	Планшет + ПК	ПК + корпоративна мережа
Ключові критерії вибору	Ціна, окупність	Технічні параметри	Інтеграція, сервіс
Частота покупок	1–2 рази на 5 років	1–2 одиниць на рік	Глобальні оновлення
Рівень цифрової обізнаності	Середній	Високий	Дуже високий

Відповідно до детально розглянутої аналітичної інформації та проведеного сегментування ринку, нами було свідомо обрано як пріоритетну цільову аудиторію саме середні фермерські господарства. Даний сегмент було вибрано на основі

ретельного економічного розрахунку, який враховує той факт, що малі господарства не оновлюють технічний парк достатньо часто через обмежені фінансові можливості і зазвичай віддають перевагу придбанню техніки, що була у використанні на вторинному ринку. З іншого боку, великі агрохолдинги у переважній більшості випадків укладають прямі багаторічні договори безпосередньо з виробниками техніки або з офіційними дилерськими центрами, що робить їх менш доступними для середніх постачальників.

Визначившись з пріоритетною цільовою аудиторією та її специфічними потребами, ми можемо приступити до формування детального технічного завдання і структурованого переліку функціональних та нефункціональних вимог до проєктованого вебресурсу. На початковому етапі важливо чітко визначити, чим саме за своєю суттю та призначенням є комерційний вебресурс у контексті нашого проєкту

Комерційний вебресурс [5] – це спеціалізована вебплатформа, створена для ефективної підтримки та розвитку бізнесу підприємств у цифровому просторі. Його основними стратегічними завданнями є максимально широке поширення структурованої інформації про компанію та її послуги, встановлення і підтримка стабільного контакту з потенційними та існуючими клієнтами, а також організація повноцінної електронної комерції з можливістю онлайн-замовлень.

Враховуючи детально проаналізовані сайти-аналоги конкурентів та виявлені в них недоліки, ми можемо сформувати наступний комплексний перелік вимог до нашого вебресурсу:

– навігаційне меню має розміщуватися у верхній частині сторінки для забезпечення інтуїтивної навігації;

– вебресурс має включати доцільно організований багаторівневий фільтр товарів і логічно структурований каталог продукції;

– всі представлені товари на сайті мають бути систематизовані та чітко поділені за релевантними категоріями та підкатегоріями;

– візуальний дизайн інтерфейсу має бути максимально простим, інтуїтивним та зрозумілим для користувачів різного рівня;

– система має підтримувати адаптивний дизайн для коректного відображення на різних типах пристроїв;

– необхідно забезпечити швидке завантаження сторінок навіть при повільному інтернет-з'єднанні;

– реалізувати зручну систему пошуку товарів за ключовими словами та артикулами.

Таким чином мною було визначено цільову аудиторію вебресурса та основні вимоги до нього.

2.2 Засоби для розроблення вебресурса

У даному підрозділі нами буде детально розглянуто сучасні мови програмування та технології, які активно використовуються професійними розробниками для створення функціональних вебресурсів різного рівня складності. Правильний та обґрунтований вибір технологічного стеку безпосередньо впливає на якість відображення сторінок у різних браузерах, потенційні можливості інтеграцій з зовнішніми системами і загальну стабільність роботи вебресурсу під навантаженням.

HyperText Markup Language (HTML) [6] являє собою стандартизовану універсальну технологію створення структурованої розмітки для документів у глобальній мережі Інтернет. Сучасні веббраузери здійснюють інтерпретацію HTML-коду згідно з міжнародними стандартами, після чого повністю оброблена сторінка виводиться на дисплей персонального комп'ютера або екран портативного мобільного гаджету користувача.

До п'ятої редакції стандарту HTML розглядався як спеціалізований компонент SGML (Standard Generalized Markup Language) відповідно до міжнародного стандарту ISO 8879. Сучасна специфікація HTML5 базується на концепції DOM (Document Object Model), яка забезпечує об'єктно-орієнтований підхід до роботи з елементами документа.

XHTML представляє строгий та формалізований різновид HTML, що послідовно наслідує синтаксичні правила XML і функціонує як його практичне застосування для створення розмітки гіпертекстових структур з підвищеними вимогами до валідності коду.

Сучасні веб браузери зазвичай завантажують HTML-документи, активно взаємодіючий із серверним обладнанням через стандартизовані протоколи HTTP чи HTTPS, передаючи дані у форматі звичайного незахищеного тексту або із обов'язковим застосуванням криптографічного захисту для забезпечення безпеки передачі інформації.

HTML функціонує як добре структурована тегова система розмітки документів. Кожен HTML-документ логічно складається з множини взаємопов'язаних елементів, чіткі межі яких визначаються спеціальними текстовими маркерами – тегами. Певні специфічні елементи можуть бути порожніми, не вміщуючи всередині себе текстового наповнення чи додаткових вкладених даних. Для таких самозакривних елементів закриваючий тег зазвичай повністю відсутній у синтаксисі. Більшість елементів можуть містити різноманітні атрибути, що детально визначають їхні специфічні характеристики та поведінку.

Cascading Style Sheets (CSS) - формалізована спеціалізована мова точної специфікації візуального представлення документів, створених засобами різноманітних мов розмітки. CSS також широко застосовується до XML-документів різного функціонального призначення, зокрема векторної графіки SVG та інтерфейсів XUL.

CSS3 надає потужні інструменти для стилізації та анімації елементів інтерфейсу. Flexbox використовувався для створення гнучких макетів, що автоматично адаптуються до ширини контейнера. Grid Layout застосовувався для побудови складних сіток каталогу товарів, що дозволяє легко контролювати розташування елементів у двовимірному просторі. CSS-змінні (custom properties) забезпечували централізоване управління кольоровою схемою та розмірами елементів, що полегшує підтримку та модифікацію стилів.

Професійні веб розробники активно використовують потужні можливості CSS для детального визначення колірних схем інтерфейсу, параметрів типографії та шрифтового оформлення, стилістичного візуального оформлення елементів, точного позиціонування структурних блоків на сторінці та багатьох інших важливих характеристик візуального представлення сучасних веб сторінок. Ключове стратегічне завдання створення технології CSS полягало у фундаментальному відокремленні логічної архітектури веб документа (що реалізується через HTML чи альтернативні мови розмітки) від детальної специфікації його зовнішнього візуального вигляду. Подібне чітке розмежування суттєво підвищує загальну доступність документації для користувачів, забезпечує широкі гнучкі можливості централізованого керування представленням контенту, а також значно знижує складність підтримки та рівень дублювання коду у структурному наповненні системи.

Додатково CSS надає унікальну можливість гнучкого представлення ідентичного за змістом документа у принципово різноманітних форматах виведення - звичайному екранному відображенні для моніторів, оптимізованій друкованій версії для паперу, спеціалізованому аудіальному відтворенні для людей з вадами зору.

До широкого впровадження технології CSS стилістичне оформлення веб документів реалізовувалося виключно обмеженим інструментарієм HTML безпосередньо в межах структурного контенту сторінки. Революційна поява CSS уможливила фундаментальне концептуальне розділення змісту та оформлення документації на різні рівні абстракції. Ця технологічна інновація дозволила максимально ефективно застосовувати єдину узгоджену стилістичну концепцію одночасно до численних подібних документів та оперативно модифікувати їхнє візуальне представлення шляхом редагування лише одного центрального файлу стилів.

JavaScript являє собою потужну мову програмування, що підтримує множинні парадигми сучасної розробки програмного забезпечення. Мова органічно інтегрує об'єктно-орієнтовані, імперативні та функціональні підходи до

написання ефективного коду. JavaScript представляє практичну повноцінну імплементацію міжнародної специфікації ECMAScript.

Переважно JavaScript функціонує як універсальна вбудована технологія, що забезпечує гнучкий програмний доступ до внутрішніх компонентів різноманітних застосунків та їх функціоналу. Найширше практичне розповсюдження мова отримала саме у сучасних веб браузерах, де вона ефективно виконує важливу роль скриптової технології для створення інтерактивних динамічних елементів на веб сторінках та забезпечення відгуку на дії користувача.

Ключові фундаментальні архітектурні характеристики JavaScript включають: динамічну гнучку систему типізації даних, нестрогу слабку типізацію змінних, повністю автоматизоване управління оперативною пам'яттю через механізм збирання сміття, прототипну об'єктно-орієнтовану модель програмування, трактування функцій як повноцінних первинних об'єктів першого класу.

JavaScript концептуально побудований на класичних об'єктно-орієнтованих принципах програмування, проте практичне застосування специфічної прототипної моделі спричиняє певні суттєві відмінності у роботі з об'єктами порівняно з традиційними класичними мовами, орієнтованими на використання класів як основи. Водночас JavaScript яскраво демонструє характерні характеристики, притаманні функціональним мовам програмування: функції як повноцінні первинні об'єкти, об'єкти у якості структур даних типу списків, підтримка керування функцій, можливість створення безіменних анонімних функцій, потужний механізм замикань. Всі ці унікальні особливості надають мові JavaScript значну гнучкість та виразність у практичному застосуванні для вирішення складних завдань.

JavaScript ES6+ використовувався для реалізації інтерактивної функціональності. Модульна структура коду з використанням import/export дозволяє розділити логіку на окремі файли згідно з принципом відповідальності. Асинхронні операції з використанням async/await забезпечують зручну роботу з API без блокування інтерфейсу. Деструктуризація об'єктів та масивів, стрілкові функції, шаблонні рядки роблять код більш читабельним та лаконічним.

Розглянувши детально основні мови для професійної розроблення верстки і динамічних сторінок сайту, ми можемо перейти до ретельного розгляду спеціалізованих мов роботи з базами даних, які є критично важливими для зберігання інформації.

SQL – декларативна спеціалізована мова програмування для роботи з даними, що широко застосовується для створення структури, модифікації існуючих даних та управління інформацією в реляційній базі даних, керованої відповідною потужною системою управління базами даних (СУБД).

Дана універсальна мова слугує основним стандартизованим інструментом взаємодії користувачів та програмних систем з базою даних. До її ключових функціональних можливостей відносять: формування нових таблиць у структурі бази даних, внесення нових записів у існуючі таблиці, гнучка модифікація та оновлення існуючих записів, вилучення непотрібних записів з дотриманням цілісності даних, складні вибірки інформації з однієї або декількох взаємопов'язаних таблиць згідно з заданими складними умовами.

Python являє собою інтерпретовану об'єктно-орієнтовану мову програмування високого рівня абстракції, що характеризується строгою динамічною типізацією змінних під час виконання програми. Створена у 1990 році талановитим нідерландським програмістом Гвідо ван Россумом, мова швидко здобула популярність у різних галузях. Вона забезпечує надійну підтримку модулів та модульних пакетів, що активно стимулює модульність загальної архітектури проєктів та максимальне повторне застосування готового програмного коду у різних контекстах.

Python володіє ефективними високорівневими структурами даних та лаконічним, водночас надзвичайно потужним підходом до об'єктно-орієнтованого програмування з підтримкою успадкування. Елегантний та виразний синтаксис Python, динамічна інтелектуальна обробка типів даних під час виконання, а також інтерпретований характер мови роблять її оптимальним вибором для швидкого створення автоматизаційних скриптів та оперативної розробки прикладного

програмного забезпечення у різноманітних галузях застосування на переважній більшості сучасних обчислювальних платформ.

Python реалізує гнучку динамічну типізацію, що практично означає визначення конкретного типу змінної виключно під час фактичного виконання програмного коду. Серед базових вбудованих типів даних варто особливо відзначити надійну підтримку цілих чисел необмеженої розрядності (обмежених лише доступною пам'яттю) та комплексних чисел для математичних обчислень. Python володіє надзвичайно потужною стандартною бібліотекою для ефективного маніпулювання рядковими даними, зокрема текстом, закодованим у сучасному форматі Unicode для підтримки інтернаціоналізації. Гнучка система класів забезпечує повноцінне множинне успадкування та розвинуте мета програмування для створення складних абстракцій. Усі типи даних без винятку, включаючи навіть базові примітивні типи, органічно інтегровані у єдину систему класів, що за потреби дозволяє гнучке успадкування навіть від фундаментальних вбудованих типів для розширення їх функціоналу.

Після детального ознайомлення з найбільш популярними та широко використовуваними мовами програмування, які активно застосовуються професіоналами для розробки сучасних вебресурсів різного рівня складності, ми можемо зробити остаточний обґрунтований вибір технологічного стеку для нашого проєкту. Для успішного виконання поставлених завдань кваліфікаційної роботи мною було свідомо обрано перевірені мови HTML і CSS для створення якісної верстки та стилізації інтерфейсу, потужну мову SQL для проєктування та створення надійної реляційної бази даних. Даний технологічний вибір вирізняється відносною простотою розуміння синтаксису для початківців і водночас надає просторний функціонал для реалізації складних завдань професійного рівня.

Flask був обраний як backend-фреймворк завдяки своїй простоті та гнучкості. На відміну від більш масивних фреймворків, Flask надає мінімальний набір інструментів, дозволяючи розробнику самостійно обирати додаткові бібліотеки згідно з потребами проєкту. Це особливо важливо для освітніх проєктів, де необхідно розуміти кожен аспект роботи системи.

Маршрутизація у Flask реалізується через декоратори, що дозволяє оголошувати endpoint-и безпосередньо перед функціями-обробниками. Це робить код інтуїтивно зрозумілим – одразу видно, яка функція обробляє який URL. Підтримка різних HTTP-методів (GET, POST, PUT, DELETE) дозволяє реалізувати REST згідно API з загальноприйнятими стандартами.

Архітектура бази даних була спроєктована згідно з принципами нормалізації для мінімізації дублювання даних та забезпечення цілісності. Використовувались зовнішні ключі для підтримки зв'язків між таблицями, індекси для прискорення пошукових запитів, тригери для автоматичного оновлення службових полів.

2.3 Архітектура вебресурса

Архітектура розробленого вебресурсу побудована за принципом клієнт-серверної взаємодії з чітким розділенням обов'язків між клієнтською та серверною частинами. Така архітектура забезпечує гнучкість, масштабованість та полегшує подальшу підтримку системи.

Клієнтська частина являє собою Multi-Page Applications (MPA), яка завантажується при першому відвідуванні сайту і далі завантажує нові сторінки без повного перезавантаження. Це забезпечує швидку реакцію на дії користувача та створює відчуття роботи з нативним застосунком. HTML-структура визначає семантичну розмітку сторінок, CSS відповідає за візуальне представлення, JavaScript реалізує логіку взаємодії та комунікацію з сервером.

Для організації JavaScript-коду використовується модульна структура. Кожен функціональний блок (каталог, кошик, форми, API-клієнт) винесений в окремий модуль з власним простором імен. Це запобігає конфліктам імен змінних та функцій, полегшує налагодження та тестування окремих компонентів. Головний модуль main.js координує роботу всіх інших модулів та керує життєвим циклом застосунку.

Система маршрутизації на клієнті відслідковує зміни URL та відображає відповідний вміст без запитів до сервера. Використовується History API браузера для управління навігацією. При переході між розділами сайту URL змінюється, що дозволяє користувачам використовувати кнопки "Назад" та "Вперед" браузера, а також створювати закладки на конкретні сторінки.

Управління станом застосунку здійснюється через централізоване сховище. Всі дані, що впливають на відображення інтерфейсу (вміст кошика, активні фільтри, дані користувача), зберігаються в єдиному об'єкті стану. При зміні стану автоматично відбувається завантаження відповідних компонентів інтерфейсу. Це забезпечує цілісність даних та спрощує налагодження.

Серверна частина побудована на фреймворку Flask, що надає мінімальний набір інструментів для створення вебресурсів. Flask відповідає за обробку HTTP-запитів, маршрутизацію, взаємодію з базою даних, генерацію відповідей. Легкість фреймворку дозволяє досягти високої продуктивності навіть на недорогому хостингу.

Архітектура серверної частини організована за принципом контроля представлення моделі. Моделі відповідають за роботу з даними та бізнес-логіку. Кожна модель представляє сутність системи (товар, категорію, замовлення) та інкапсулює методи для роботи з нею. Представлення відповідають за генерацію відповідей – у нашому проєкті це JSON-дані для API. Контроль приймає запити, викликає необхідні методи моделей та формують відповіді.

База даних SQLite зберігає всі збережені дані системи. Вибір SQLite обумовлений простотою розгортання та відсутністю необхідності в окремому серверному процесі. База представлена набором нормалізованих таблиць з визначеними зв'язками. Індекси створені для полів, що часто використовуються в запитах (ID товарів, назви категорій, статуси замовлень).

Взаємодія між клієнтом та сервером здійснюється через REST API. Кожен ресурс системи має свій набір endpoint, що підтримують операції CRUD. Використовуються стандартні HTTP-методи: GET для отримання даних, POST для створення нових записів, PUT для оновлення існуючих, DELETE для видалення.

Форматом обміну даними вибрано JSON. Всі відповіді API мають узгоджену структуру: поле status містить код результату операції, data – корисне навантаження, message – текстове повідомлення для відображення користувачу, errors – масив помилок валідації, якщо вони виникли.

Обробка помилок організована на кількох рівнях. На рівні клієнта перевіряється коректність введення даних у форми перед відправкою на сервер. На рівні сервера виконується додаткове узгодження вхідних даних, перевірка прав доступу, обробка виключних ситуацій. Всі помилки зберігаються в логі для подальшого аналізу.

Кешування застосовується для оптимізації продуктивності. Статичні ресурси (CSS, JS) віддаються з довгим терміном кешування. Відповіді API для даних, що рідко змінюються (список категорій, виробників), кешуються на стороні клієнта. При оновленні даних кеш узгоджуються.

2.4 Створення структури вебресурса

Грамотне проєктування логічної структури є одним з найважливіших етапів у процесі створення функціональних вебресурсів будь-якого рівня складності. Саме структура відповідає за те, наскільки доступним та зрозумілим буде юзабіліті вебресурсу для кінцевих користувачів, які можливості модифікації та розширення вмісту будуть доступні адміністраторам системи, а також як ефективно користувачі зможуть знаходити необхідну їм інформацію. У класичній теорії створення вебресурсів професіонали розрізняють декілька основних типів архітектурних структур організації контенту:

1. Ієрархічна структура – найпоширеніший тип організації великих сайтів, що нагадує деревоподібну систему з чітко визначеними рівнями вкладеності. У такій структурі навігаційні посилання з головної сторінки ведуть на окремі тематичні розділи верхнього рівня, які у свою чергу містять посилання на менші тематичні підрозділи, ті можуть ділитися на ще менші спеціалізовані сторінки з конкретним

контентом. Дана структура активно застосовується у сайтах-каталогах з великим асортиментом, інформаційних порталах з різноманітним контентом, великих маркетплейсах з численними категоріями товарів, корпоративних інтернет-магазинах та багатьох інших типах складних вебресурсів з великою кількістю сторінок.

2. Блокова мережева структура припускає, що абсолютно всі сторінки сайту містять прямі перехресні посилання один на одного для забезпечення максимальної зв'язності. Дана специфічна структура практично не використовується для великих вебресурсів з значною кількістю сторінок через складність підтримки, і в основному знаходить застосування для невеликих спеціалізованих сайтів, присвячених одному конкретному типу товарів або вузькій темі, де важлива швидка навігація між всіма матеріалами.

3. Лінійна послідовна структура представляє з себе набір логічно впорядкованих сторінок, що йдуть строго одна за одною у заздалегідь визначеній послідовності. Така організація ідеально підходить для навчальних курсів, покрокових інструкцій, презентацій продуктів, де важливо, щоб користувач послідовно проходив матеріал від початку до кінця без пропусків.

Для виконання нашого проєкту було вибрано ієрархічну структуру. Вона найкраще підходить для створення професійних сайтів комерційного напрямку з великим каталогом продукції, де товари логічно групуються за категоріями та підкатегоріями. Детальна структура нашого вебресурсу з усіма рівнями вкладеності наочно зображена на рисунку 2.1.

Рисунок 2.1 – Структура сайту

Не менш важливим аспектом проєктування є чітке візуальне представлення всіх можливих дій та операцій, які будуть виконуватися користувачами різних типів на вебресурсі в процесі його використання. Для наочного та структурованого відображення функціональних можливостей системи найкращим професійним вибором буде створення спеціалізованого зображення у формі стандартизованої UML діаграми прецедентів, яка дозволяє побачити систему очима користувача.

UML (Universal Modeling Language) – універсальна мова моделювання, який був розроблений компанією Rational Software з метою створення найбільш оптимального та універсальної мови для опису як предметної області, так і конкретного завдання в програмуванні. Візуальне моделювання в UML можна уявити як певний процес порівневого спуску від найбільш загальної і абстрактної концептуальної моделі системи до логічної, а потім і до фізичної моделі відповідної системи. Будь-яке завдання, таким чином, моделюється за допомогою деякого набору ієрархічних діаграм, кожна з яких представляє собою деяку проекцію системи.

Детальна діаграма прецедентів нашого вебресурсу з усіма акторами та варіантами використання наочно продемонстрована на рисунку 2.2.

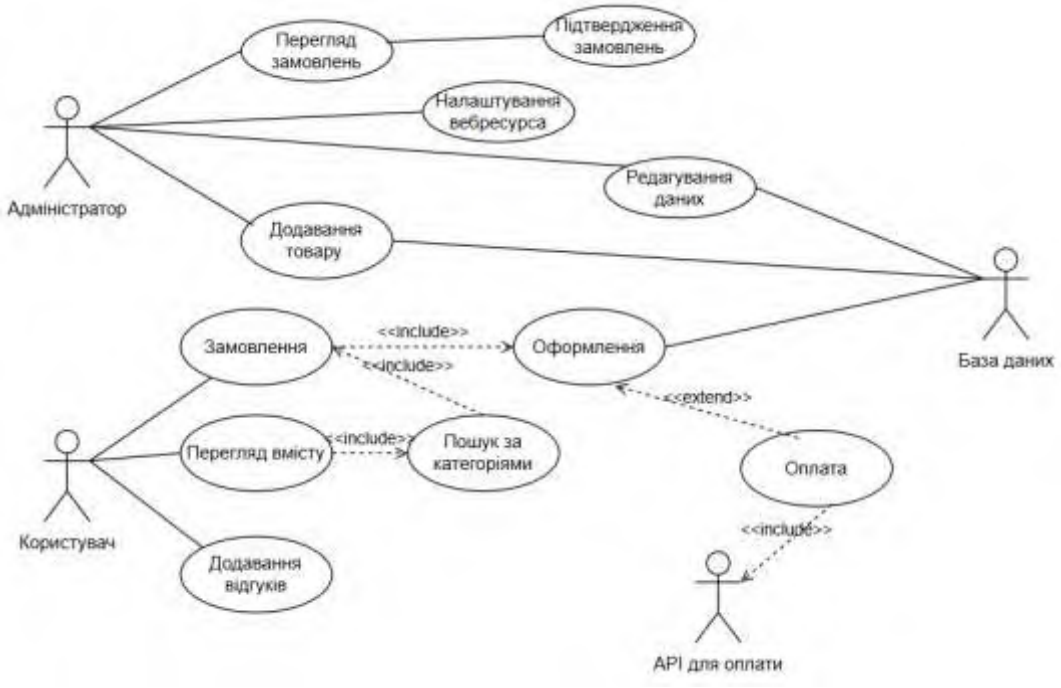

Рисунок 2.2 – Діаграма прецедентів

Відповідно до створеної діаграми ми можемо зрозуміти дії користувачів вебресурса. Таким чином в даному розділу мною було проведено дослідження цільової аудиторії майбутнього вебресурса, створено список вимог, яким має відповідати створений продукт, розглянуто можливі взаємодії користувачів з ресурсом. Беручи до уваги розглянуту інформацію ми можемо перейти до етапу розроблення самого вебресурса.

РОЗДІЛ 3
РОЗРОБКА ВЕБРЕСУРСА

3.1 Визначення робочих інструментів

Перед розробкою програмної частини важливо визначитися з робочим простарнством розробки. Від даної частини залежить складність розробки, необхідні навички і можливості готового сайту. Основними вважаються:

1. Фреймворки – це готовий комплексний інструментарій для прискореного створення якісного програмного забезпечення з використанням перевірених практик. У сфері веб-розробки фреймворк забезпечує розробників готовими архітектурними рішеннями щодо створення надійної backend-архітектури вебресурсів. Він вирізняється цілим рядом безперечних переваг, такими як суттєве пришвидшення процесу розробки за рахунок готових компонентів, стандартизація та уніфікація коду згідно з кращими практиками галузі, вбудована інтеграція сучасних інструментів забезпечення безпеки, гнучка можливість горизонтального та вертикального масштабування системи і висока загальна продуктивність роботи. Для професійної розробки веб-ресурсів фреймворки класифікують за типом архітектурного патерну, який вони підтримують та реалізують:

– Контролер представлення моделі – даний широко розповсюджений тип архітектури логічно ділить всю систему на три чітко розділених рівня відповідальності: модель для роботи з даними та бізнес-логікою, контролер для обробки запитів користувача та керування потоком виконання, представлення для відображення інформації користувачу. Такий поділ забезпечує високий рівень модульності та можливість незалежної розробки компонентів.

– Model-View-ViewModel (MVVM) – структурована організація коду з допомогою готових шаблонів проєктування.В ній розділені бізнес логіка і відображення даних. Дана модель ділиться на рівні: ViewModel, представлення та модель.

– На основі Push & Pull – відрізняється тим фактом, що ініціація процесу надсилання даних починається на рівні виконання дії контролера і тільки потім система надсилає оброблені дані на рівень візуального відображення для користувача, що дозволяє краще контролювати потік даних.

– Трирівнева структура – даний тип фреймворків розділяє клієнт сервер на три фізичних та логічних рівня. Дані рівні називаються прикладний, презентаційний рівень та рівня бази даних.

Також даний тип інструмента розділяють за видом розробки:

– Інтерфейсні веб-фреймворки – даний тип фрейм ворків відповідає за створення видимої частини сайту. Вони включають попередньо створені частини коду і шаблони, які можна використовувати як основу. Зазвичай в таких фреймворках є готові шрифти, компоненти і елементи дизайну. Акценти при його використанні ставляться на інтерактивність та функціональність програми.

– Серверні веб-фреймворки – дані фреймворки спрощують створення backend сторони вебресурсів, таких як маршрутизація, взаємодія з базою даних, підвищення безпеки і все інше, що знаходиться на стороні сервера і невидиме для звичайних користувачів. В ній зазвичай серверні мови програмування, такі як Python, C++, JavaScript, PHP та інші.

2. Другим варіантом розробки вебресурсів є використання інструментів на основі штучного інтелекту. Даний тип розробки є відносно новим методом. Цей метод надає можливість створити власний ресурс знань верстки і коду. В даному сегменті можна виділити наступні рішення:

– Генератор за текстовим описом – даний вид дозволяє створити вебресурс за описом всього кількома реченнями. Його вважать найшвидшим рішенням, де необхідно всього кілька дій.

– ШІ–конструктори – в даному рішенні використовується збір сайту з уже готових шаблонів «блоків». В ньому користувач створює розташування блоків, а ШІ підбирає зображення, дизайн, шрифти і кольори.

– Гібридні рішення дане рішення дозволить зберегти час на ручному налаштуванні і зосередитися на деталях.

Даний варіант підходить для швидкого створення вебресурсів, але має глобальні недоліки при розробці сайтів зі складною логікою і унікальними функціями чи дизайном.

3. Третім варіантом є створення вебресурса вручну від дизайну до функцій. Даний вид є самий довгий у реалізації, але дає повний контроль над проєктом. Він виконується у середовищах розробки IDE використовуючи мови програмування.

Для нашого проєкту було вибрано варіант номер 3. Такий вибір відрізняється низькою вартістю програмного забезпечення, але і одночасно важкістю запуска веб платформи.

Основним середовищем розробки (IDE) обрано Visual Studio Community – безкоштовну версію інтегрованого середовища Microsoft, що надає повний набір інструментів для роботи з різними мовами. Середовище забезпечує підсвічування синтаксису для HTML, CSS, JavaScript і Python, автодоповнення коду, вбудований термінал для запуску сервера, підтримку Git. Перевагою є можливість відкрити цілу папку як проєкт і переміщатись між файлами у єдиному робочому просторі без перемикання між вікнами.

Система контролю версій Git використовувалась для фіксації стану проєкту на кожному значущому кроці. Регулярні коміти з описовими коментарями забезпечували можливість повернутись до будь-якого попереднього стану без втрати роботи. Ця можливість виявилась особливо цінною при рефакторингу складних компонентів – кошика і системи фільтрів.

Тестування веб інтерфейсу здійснювалось у Google Chrome з використанням вбудованих засобів DevTools. Вкладка Network дозволяла відстежувати HTTP-запити між клієнтом і сервером, вкладка Console – помилки JavaScript у реальному часі, інструмент Emulation – симулювати перегляд з різних мобільних пристроїв. Тестування API проводилось через тестовий клієнт Flask, що дозволяє надсилати запити різних типів безпосередньо з Python-коду і перевіряти відповіді сервера.

Структура проєкту організована так: HTML-файли – в кореневій директорії, стилі – у папці css/, JavaScript-модулі – у папці js/, файли серверної частини і база

даних – також у кореневій директорії. Така організація спрощує налаштування Flask для роздачі статичних файлів і мінімізує кількість маршрутів.

3.2 Розробка сторінок для вебресурса

Розробка розпочалась із проєктування і реалізації загального шаблону сторінок. Усі сторінки поділяють три спільних компоненти: верхню інформаційну смугу, футер і підвал. Верхня смуга містить email, телефон і швидкі посилання. Футер – логотип, поле пошуку і кнопку кошика з лічильником. Навігаційна смуга синього кольору зафіксована під шапкою і залишається видимою при прокрутці сторінки.

Головна сторінка виконує роль вітрини компанії. Центральним елементом є автоматичний слайдер з трьома банерами, кожен з яких присвячений окремому напряму: каталог нової техніки, зернозбиральна техніка, гарантійний сервіс. Слайдер реалізований засобами JavaScript з автопрокруткою кожні 5 секунд. Індикатори поточного слайда реалізовані як набір кнопок-точок внизу слайдера. При кліку на точку відбувається миттєвий перехід до відповідного слайда. Активна точка виділяється кольором та збільшеним розміром. Стрілки переходу розміщені по боках слайдера та з'являються при наведенні, щоб не відволікати увагу від самого контенту. Нижче розміщено блок показників компанії, динамічне прев’ю чотирьох товарів каталогу і заклик до дії з посиланням на гарантійний сервіс. Вигляд сторінки показано на рисунку 3.1.

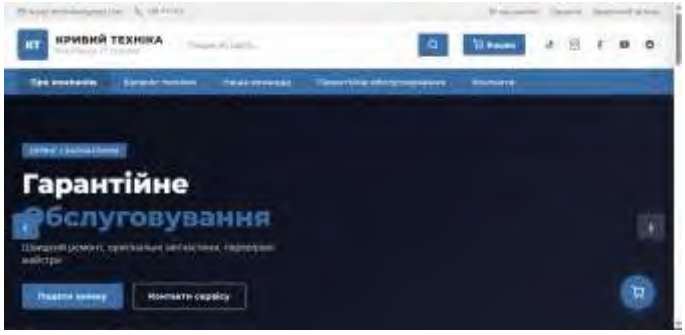

Рисунок 3.1 – Головна сторінка вебресурса

Прев’ю товарів на головній сторінці формується динамічно через запит до API. Завантажуються чотири випадкових товари зі статусом "в наявності". Кожна картка товару містить зображення, назву, модель, ціну та кнопку швидкого додавання в кошик. При наведенні на картку з'являється короткий опис та додаткова інформація про статус наявності. Клік на будь-яку частину картки, окрім кнопки кошика, веде на детальну сторінку товару.

Зміну кольору кнопок реалізовано за допомогою CSS коду, та доєднано до html файлу. Її реалізація представлена на рисунку 3.2

```
.btn {
  display: inline-flex;
  align-items: center;
  gap: .4rem;
  font-family: var(--font-h);
  font-weight: 700;
  font-size: .85rem;
  letter-spacing: .04em;
  padding: .75rem 1.6rem;
  border-radius: 4px;
  cursor: pointer;
  border: 2px solid transparent;
  transition: all var(--tr);
}
.btn--primary { background: var(--red); color: var(--white); border-color: var(--red); }
.btn--primary:hover { background: var(--red-dark); border-color: var(--red-dark); }
.btn--ghost { background: transparent; color: var(--white); border-color: rgba(255,255,255,.5); }
.btn--ghost:hover { background: rgba(255,255,255,.15); border-color: var(--white); }
.btn--outline { background: transparent; color: var(--red); border-color: var(--red); }
.btn--outline:hover { background: var(--red); color: var(--white); }
.btn--dark { background: var(--dark); color: var(--white); border-color: var(--dark); }
.btn--dark:hover { background: var(--black); }
.btn--lg { padding: 1rem 2.2rem; font-size: .95rem; }
.btn--sm { padding: .5rem 1.1rem; font-size: .8rem; }
```

Рисунок 3.2 – Реалізація роботи кнопок

Сторінка каталогу є функціонально найнасиченішим розділом. Ліва бічна панель містить кілька блоків фільтрів: дерево категорій і підкатегорій, список виробників, поля діапазону ціни і вибір статусу наявності. Кожен блок може розгортатись і згортатись, а при накопиченні більше 5 пунктів – з’являється кнопка. Активні фільтри відображаються тегами над сіткою товарів, кожен тег можна прибрати кнопкою «×». Фільтрація відбувається на стороні клієнта без перезавантаження сторінки, що забезпечує миттєву реакцію інтерфейсу. Сторінка каталогу продемонстрована на рисунку 3.3.

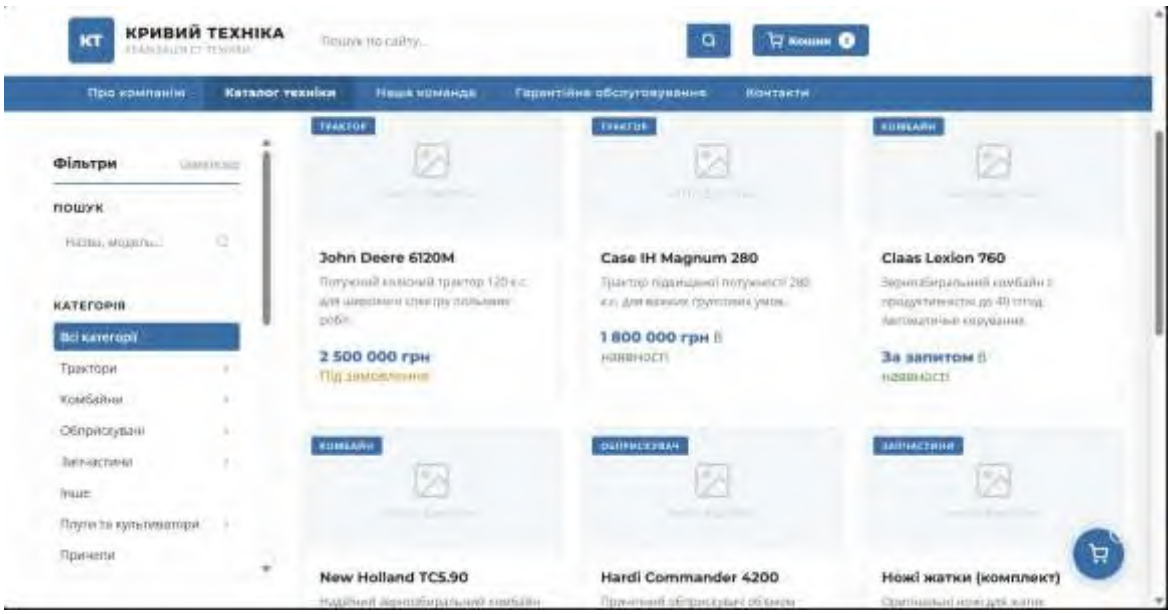

Рисунок 3.3 – Сторінка каталогу

Фільтри працюють у режимі реального часу без перезавантаження сторінки. При зміні будь-якого фільтра запускається функція перерахунку видимих товарів. Спочатку формується набір активних фільтрів, потім кожен товар перевіряється на відповідність всім активним умовам.

Дерево категорій реалізоване як вкладений список з можливістю розгортання підкатегорій. При виборі батьківської категорії автоматично відображаються всі товари з цієї категорії та її підкатегорій. При виборі підкатегорії фільтруються лише товари цієї конкретної підкатегорії. Активна категорія виділяється кольором та жирним шрифтом.

Реалізація пошуку була виконана з допомогою JavaScript і виділення необхідної категорії реалізовано у CSS форматі. Код зображено на рисунку 3.4.

Рисунок 3.4 – Реалізація вибору фільтра

```
.active-filters {
  display: flex;
  flex-wrap: wrap;
  gap: .4rem;
  margin-bottom: .8rem;
  min-height: 0;
}

.filter-tag {
  display: flex;
  align-items: center;
  gap: .3rem;
  background: var(--red-light);
  color: var(--red);
  border: 1px solid rgba(58,110,165,.2);
  border-radius: 20px;
  padding: .25rem .7rem;
  font-size: .78rem;
  font-weight: 600;
}

.filter-tag__remove {
  background: none;
  border: none;
  color: var(--red);
  cursor: pointer;
  font-size: .9rem;
  padding: 0;
  line-height: 1;
  margin-left: .1rem;
}
```

Рисунок 3.5 – Реалізація зміни кольору активного фільтра

Сторінка оформлення замовлення відображає повний список кошика у вигляді таблиці з колонкою рядкових сум і загальним підсумком. Паралельно формується «підсумок замовлення» – окрема таблиця праворуч із товарами, кількостями і рядковими сумами. Форма оформлення містить: прізвище та ім’я, телефон, email, адресу, спосіб доставки і оплати, коментар. Логіка доставки: якщо кошик містить великогабаритну техніку (трактори, комбайни, обприскувачі, причепи, двигуни) – список доставки блокується і залишається лише «Самовивіз» з відповідним попередженням. Зовнішній вигляд сторінки замовлення зображено на рисунку 3.6.

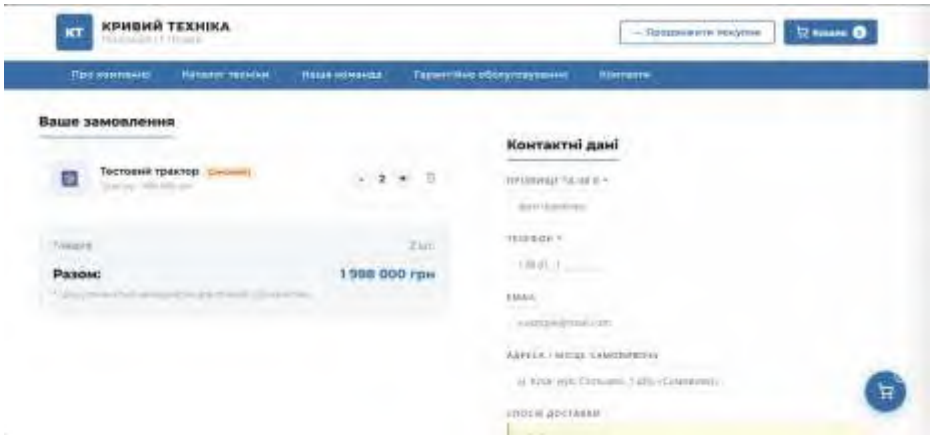

Рисунок 3.6 – Сторінка замовлення

Наступним було створено вікно «Наша Команда». Даний пункт сайту презентує керівництво і головних співробітників підприємства. Даний розділ показано на рисунку 3.7.

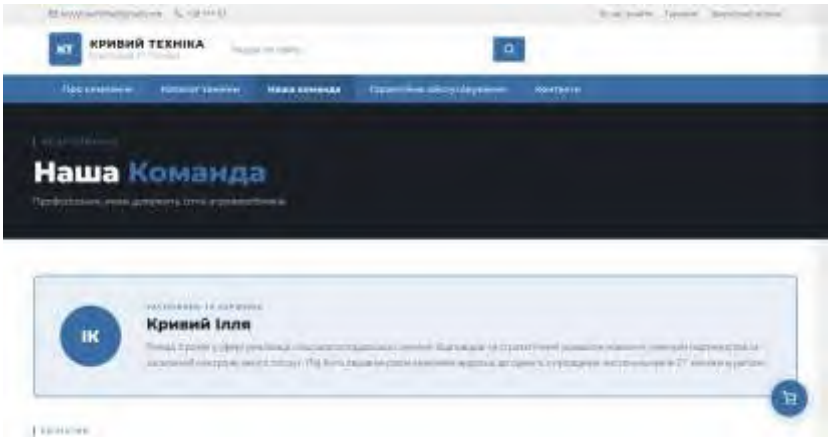

Рисунок 3.7 – Розділ сайту «наша команда»

Сторінка гарантійного обслуговування поєднує інформаційну і функціональну частини. Інформаційна – чотирикроковий опис процесу і перелік умов гарантії. Форма заявки включає: ім’я, телефон, email, марку і модель техніки, серійний номер, дату купівлі, тип несправності зі списку, детальний опис і місцезнаходження техніки. Після відправки дані зберігаються у базі і стають видимими в адмін-панелі. Вигляд сторінки показано на рисунку 3.8.

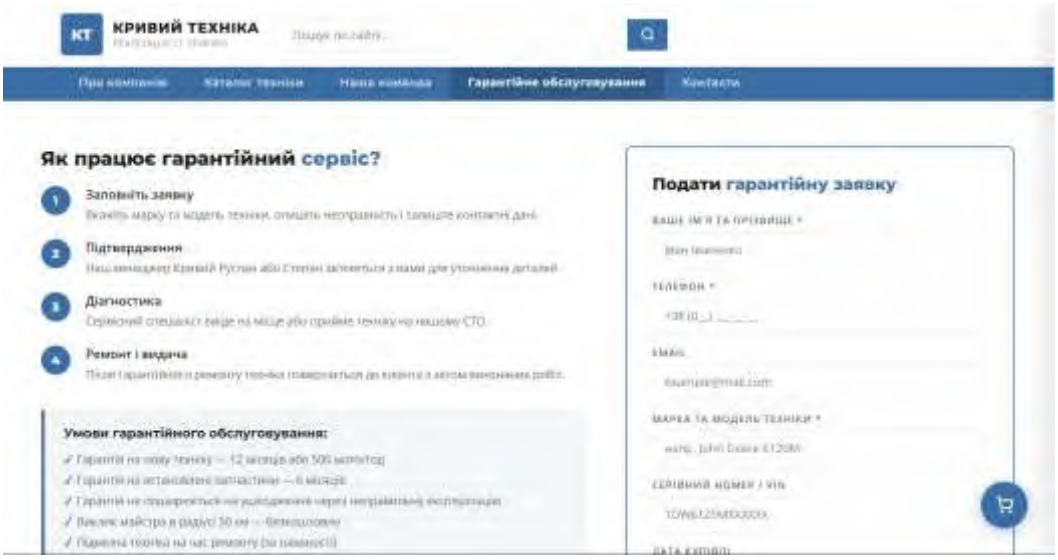

Рисунок 3.8 – Сторінка Гарантійне обслуговування

Логіка роботи даної сторінки виконана мовою JavaScript і показана на рисунку 3.9. Всі подані заявки видні з панелі адміністратора.

Рисунок 3.9 – Фрагмент коду сторінки «Гарантійне обслуговування»

Адмін-панель захищена PIN-кодом і містить: форму додавання товару з усіма полями, список товарів з вбудованим редагуванням ціни і статусу, списки замовлень і гарантійних заявок. Дизайн вебресурса показано на рисунку 3.10.

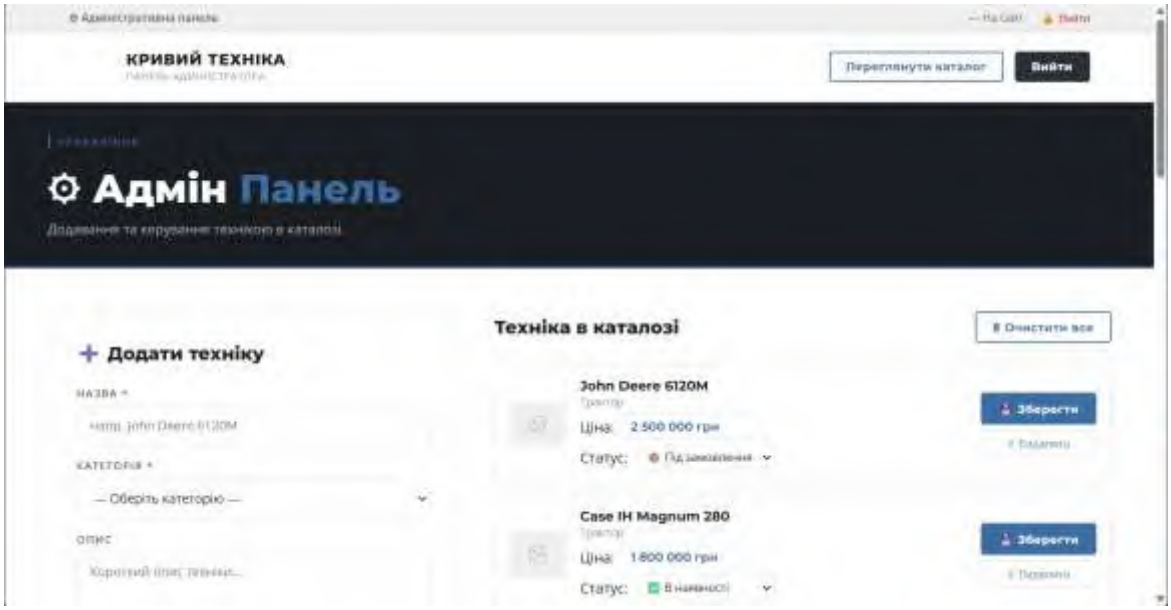

Рисунок 3.10 – Сторінка адміністратора

Додавання нової техніки і обладнання реалізовано за допомогою JavaScript. Код даної функції продемонстровано на рисунку 3.11.

```
async function addItem() {
  const get = id => document.getElementById(id)?.value.trim();
  const msg = document.getElementById('adminFormMsg');

  const name = get('aName');
  const cat  = get('aCat');

  if (!name) return flash(msg, 'Введіть назву техніки.', 'err');
  if (!cat)  return flash(msg, 'Оберіть категорію.', 'err');

  const res = await addProduct({
    name,
    cat,
    desc:   get('aDesc') || 'Деталі уточнюйте у менеджера.',
    price:  get('aPrice') || 'За запитом',
    status: get('aStatus') || 'available',
    photo:  photoBase64
  });
```

Рисунок 3.11 – Добавлення товару

В даному підрозділі нам вдалося створити дизайн і інтерактивні компоненти сайту.

3.3 Створення бази даних вебресурса

Серверна частина на Flask виконує дві функції: роздачу статичних файлів і обробку REST API запитів. Взаємодія з базою організована через три допоміжні функції у db.py: q() – для SELECT, що повертає список; one() – для SELECT, що повертає один запис; run() – для INSERT/UPDATE/DELETE, що повертає lastrowid. Всі функції використовують контекстний менеджер для автоматичного управління з'єднанням і транзакціями.

База даних SQLite містить 13 таблиць. Таблиця categories зберігає ієрархію категорій з полем parent_id для підкатегорій. Таблиця manufacturers – довідник

виробників. Таблиця products – центральне сховище каталогу: посилання на категорію і виробника, назва, модель, опис, числова ціна, відображуваний рядок ціни, статус, ознака велико габаритності, фото у base64. Тригер products_updated_at автоматично оновлює дату зміни. Таблиця clients формується автоматично: новий запит перевіряє наявність клієнта за телефоном і або використовує існуючий запис, або створює новий. Таблиці orders і order_items зберігають замовлення та їх позиції із знімком даних товару на момент замовлення. Таблиці warranty_requests і contact_requests – гарантійні та контактні заявки. Додатково: employees, admins, admin_logs, settings.

REST API включає 20 ендпоінтів. Для каталогу: GET /api/products, GET /api/products/<id>, POST /api/products, PATCH /api/products/<id>, DELETE /api/products/<id>. Для пошуку: GET /api/search (з параметрами q, cat, price_min, price_max, maker, status). Довідники: GET /api/categories (дерево з дочірніми), GET /api/makers. Для замовлень: POST /api/orders, GET /api/orders, PATCH /api/orders/<id>/status. Для гарантії: POST /api/warranty, GET /api/warranty, PATCH /api/warranty/<id>/status. Для контактів: POST /api/contact, GET /api/contact. Налаштування: GET /api/settings. Усі ендпоінти повертають JSON.

3.4 Тестування вебресурсу та його розгортання

Важливим етапом у розробці вебресурсів є тестування продукту з метою перевірки функціональності, стабільності роботи та відповідності вимогам. В даному розділі визначаються можливі помилки, недоліки інтерфейсу, проблеми сумісності і потенційні ризики системи.

В першу чергу виконалося функціональне тестування вебресурса. Головна увага була приділена ключовим модулям системи, таким як каталог, фільтрація товарів, сторінці оформлення замовлень, форми зворотного зв’язку та форми гарантійного обслуговування. Під час цього було розібрано правильність

відображення даних і відображення передачі між клієнтом і хостингом, стабільність роботи API.

Наступним кроком було протестовано користувацький інтерфейс. Дане тестування проводилося у браузерах Google Chrome, Microsoft Edge і Opera. Даний етап дозволив переконатися у коректному відображенні сторінок на різних платформах.

Після завершення проєктування і розробки важливим етапом є вибір хостинга для розміщення вебресурса. Дана частина відповідає за розміщення вебресурса в мережі інтернет і відображенні на стороні користувачів.

В основному хостинги розділяють на наступні типи:

– віртуальний хостинг представляє собою один фізичний сервер на якому працюють одночасно десятки різних сайтів. Даний варіант є дешевим, але сайти на ньому можуть працювати повільно;

– віртуальний виділений сервер даний тип хостингу відрізняється тим що один фізичний сервер ділять на декілька ізольованих віртуальних машин. В даному типі кожен сайт отримує гарантовані ресурси системи;

– виділений сервер є найдорожчим з усіх типів хостингу сайтів, так як потребує оренду фізичного сервера і спеціальних знань для його адміністрування.

– хмарний хостинг являє собою кілька об’єднаних серверів, які надають безвідмовну роботу сайту.

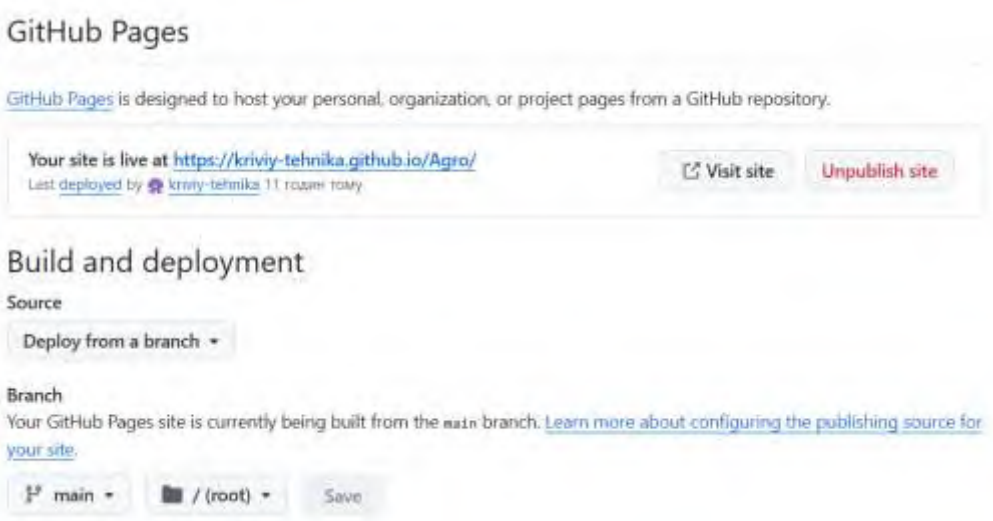

Рисунок 3.12 – Запуск сайту на GitHub Pages

Для виконання кваліфікаційної роботи було вибрано розміщення сайту на платформі GitHub. Дана платформа дозволяє розміщувати статичні сайти за допомогою GitHub Pages. Сторінка з даної платформи показана на рисунку 3.12. Даний сайт опубліковано за URl: https://kriviy-tehnika.github.io/Agro/.

3.5 Економічне обґрунтування

Економічне обґрунтування проєктування комерційного вебресурсу підприємства з реалізації сільськогосподарської техніки виконується з метою оцінювання доцільності його створення та впровадження. Основний економічний ефект від реалізації вебресурсу полягає у збільшенні обсягу продажів, скороченні витрат на залучення клієнтів, зменшенні навантаження на менеджерів з продажу та автоматизації процесу оформлення замовлень.

Економічний ефект від впровадження вебресурсу визначається як різниця між отриманою економією та понесеними витратами на розробку і підтримку і розраховується за формулою:

$$E = \Delta Q \times C_{зам} \times R \times 12 – B_{розр} - B_{підтр}, \quad (3.1)$$

де E – економічний ефект від впровадження вебресурсу, грн;

ΔQ – приріст кількості замовлень на місяць, од.;

$C_{зам}$ – середня вартість одного замовлення, грн;

R – рентабельність продажів підприємства, частки од.;

$B_{розр}$ – одноразові витрати на розробку вебресурсу, грн;

$B_{підтр}$ – річні витрати на підтримку (хостинг), грн.

Приріст кількості замовлень на місяць від вебресурса:

$$\Delta Q = 8 \times 0{,}20 = 1{,}6 \approx 2.$$

Додатковий прибуток від приросту замовлень за місяць:

$$R_{міс} = 2 \times 1\,200\,000 \times 0{,}08 = 192000 \text{ грн.}$$

Щомісячна вартість хостингу після передачі вебресурсу підприємству становить 250 грн/міс, що за рік складає:

$$В_{підтр} = 250 \times 12 = 3000 \text{ грн.}$$

Додатковий прибуток від вебресурсу за перший рік:

$$R_{рік} = 192000 \times 12 = 2304000 \text{ грн.}$$

Витрати на розробку визначаються трудомісткістю виконаних робіт. Загальний обсяг розробки склав 120 годин. За погодинної ставки розробника 350 грн/год витрати на створення вебресурсу становлять:

$$В_{розр} = 120 \times 350 = 42000 \text{ грн.}$$

Результати розрахунку витрат на розробку наведено в таблиці 3.1.

Таблиця 3.1 – Вартість розробки вебресурса

Вид роботи	Трудомісткість, год	Ставка, грн/год	Сума, грн
Проєктування архітектури та структури	20	350	7000
Розробка клієнтської частини	50	350	17500
Розробка серверної частини	35	350	12250
Тестування та налагодження	15	350	5250
Разом	120	–	42000

Економічний ефект за перший рік з урахуванням витрат на розробку та підтримку:

$$Е = 2304000 – 42000 – 3000 = 2259000 \text{ грн.}$$

Термін окупності:

$$Т_{ок} = 42000 / 192000 \approx 0{,}2 \text{ місяця.}$$

Наведений розрахунок підтверджує високу економічну доцільність розробки вебресурсу. Починаючи з другого року єдиною статтею витрат залишається хостинг вартістю 3000 грн на рік.

Крім прямого економічного ефекту, вебресурс розширює географію залучення покупців, забезпечує цілодобову доступність каталогу та підвищує впізнаваність підприємства на ринку сільськогосподарської техніки. Ці фактори створюють передумови для подальшого зростання обсягу продажів у наступних періодах.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проведено комплексне дослідження теоретичних основ та практичних аспектів проєктування комерційних вебресурсів для підприємств, що спеціалізуються на реалізації сільськогосподарської техніки. Досягнуто всі поставлені цілі та вирішено комплекс завдань, що стояли перед дослідженням.

На етапі аналізу предметної області вивчено специфіку ринку сільськогосподарської техніки в Україні, визначено ключові тренди цифровізації торгівлі агротехнікою, проаналізовано понад двадцять конкурентних вебресурсів вітчизняних та міжнародних постачальників. Виявлено основні недоліки існуючих рішень: недостатню адаптивність для мобільних пристроїв, слабку інтеграцію з обліковими системами, відсутність якісних систем фільтрації та порівняння товарів.

На основі проведеного аналізу сформульовано вимоги до функціоналу проєктованого вебресурсу. Визначено, що ресурс має включати: детальний каталог техніки з потужною системою фільтрації, функціонал електронної комерції з кошиком та оформленням замовлень, систему прийому та обробки гарантійних заявок, адміністративну панель для управління контентом, повну адаптивність для всіх типів пристроїв.

В результаті проєктування розроблено інформаційну архітектуру вебресурсу, що включає шість основних розділів: головна сторінка, каталог, сторінка товару, оформлення замовлення, гарантійне обслуговування, адміністративна панель. Для кожного розділу визначено структуру, основні елементи інтерфейсу, логіку взаємодії з користувачем.

Розроблено архітектуру бази даних, що включає тринадцять взаємопов'язаних таблиць для зберігання інформації про товари, категорії, виробників, клієнтів, замовлення, гарантійні заявки. Застосовано принципи нормалізації для забезпечення цілісності даних та мінімізації дублювання інформації.

Спроєктовано REST API, що включає двадцять endpoint для взаємодії клієнтської та серверної частин. API забезпечує повний CRUD-функціонал для управління товарами, обробку замовлень та гарантійних заявок, пошук та фільтрацію товарів, управління налаштуваннями системи.

На етапі практичної реалізації створено повнофункціональний вебресурс з використанням сучасного технологічного стеку: HTML5, CSS3, JavaScript для клієнтської частини; Python, Flask, SQLite для серверної частини. Реалізовано всі заплановані функції, проведено тестування на коректність роботи та сумісність з різними браузерами та пристроями.

Особливу увагу приділено юзабіліті інтерфейсу. Застосовано принципи інтуїтивної навігації, узгодженості елементів управління, миттєвого зворотного зв'язку на дії користувача. Реалізовано адаптивний дизайн з використанням підходу Mobile First, що забезпечує комфортну роботу з сайтом на смартфонах, планшетах та десктопах.

Практична значущість роботи полягає в тому, що розроблений вебресурс може бути безпосередньо використаний реальним підприємством для організації онлайн-продажів сільськогосподарської техніки. Запропоновані технічні рішення та методичні підходи можуть бути застосовані при проєктуванні вебресурсів для інших сегментів ринку агротехніки або суміжних галузей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. В Україні у 2025 році зареєстрували рекордну кількість сільгосптехніки URL: https://agravery.com/uk/posts/show/v-ukraini-u-2025-roci-zareestruvali-rekordnu-kilkist-silgosptehniki (дата звернення 22.04.2026)

2. Ринок сільгосптехніки 2025: що найчастіше купували та продавали українські аграрії. URL: https://agroportal.ua/publishing/infografika/rinok-silgosptehniki-2025-shcho-naychastishe-kupuvali-ta-prodavali-ukrajinski-agrariji(дата звернення 22.04.2026)

3. Цибульник С. О, Барандич К. С. Технології розроблення програмного забезпечення Частина 1. Життєвий цикл програмного забезпечення: навч.посіб. Київ : КПІ ім. Ігоря Сікорського, 2022. 268 с.

4. Agile-маніфест розробки програмного забезпечення URL: https://agilemanifesto.org/iso/uk/manifesto.html (дата звернення 22.04.2026)

5. Пустюльга С. І. Самчук В. П. Технології вебдизайну навч.посіб. Луцьк: Вежа, 2023. 304 с.

6. Босько В.В., Константинова Л.В., Марченко К.М., Улічев О.С. Web-програмування. Частина 1 (frontend) навч.посіб Кропивницький: ЦНТУ, 2022. 210 с.

7. Що таке фронтенд розробка: складові, етапи та технології : веб-сайт. URL: https://wezom.com.ua/ua/blog/chto-takoe-front-end-razrabotka (дата звернення 22.04.2026)

8. Туташвілі Ю. Web-програмування навч.посіб Луцьк: РВВ ЛНТУ, 2023. 242 с.

9. Баран С.В Основи Web-програмування навч.посіб. Кривий Ріг: Державний університет економіки і технологій, 2023. 316 с.

10. 10 найкращих фреймворків веб-розробки URL: https://www.poptin.com/blog/uk/top-web-development-frameworks/ (дата звернення 22.04.2026)

11. Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн : навч. посіб. Чернівці: нац. ун-т ім. Ю. Федьковича, 2022. 472 с.

12. Краус К.М., Краус Н.М., Манжура О.В. Електронна комерція та Інтернет-торгівля: навч.посіб. Київ: Аграр Медіа Груп, 2021. 454 с.

13. Чемерис Г. Ю. UX/UI дизайн Навч.посіб. Запоріжжя: ЗНУ, 2021. 290 с.

14. Основи UI/UX: розуміння основ дизайну інтерфейсу користувача та досвіду URL: https://ua.linkedin.com/pulse/uiux-essentials-understanding-basics-user-interface-design-olaniyan-rg4tf?tl=uk (дата звернення 22.04.2026)

15. Дакетт Д. HTML та CSS. Розробка і дизайн веб-сайтів. Київ : Ексмо, 2021. 480 с.

16. 10 етапів створення сайту – від ідеї до запуску URl: https://wezom.com.ua /ua/blog/etapy-razrabotki-sajta (дата звернення 22.04.2026)

17. Rakan S. Rashid, Ahmed A. I. The Role of Web Programming in Modern IT Solutions: Trends and Challenges. *Journal of Information Systems Engineering and Management*. 2025 Vol. 10 № 27 P.80–93 DOI: 10.52783/jisem.v10i27s.4380

18. Hall, J. The Basics of HTML. In: *Technical Writing for Developers*. *Apress Pocket Guides*. Apress, Berkeley, CA. 2025, P.64–68. DOI :10.1007/979-8-8688-2111-0_1

19. Pavle P. Designing Accessible Websites and Applications. Apress Berkeley, CA. 2025. P.145

20. Fedorchuk A. Usata O. Nakonechna O. Web design and web programming in the modern internet world. *Municipal Economy of Cities*. 2023. Vol.6, №180. P.12–20. DOI: 10.33042/2522-1809-2023-6-180-12-20

21. Jimmy Molina-Ríos, Nieves Pedreira-Souto. Comparison of development methodologies in web applications. *Information and Software Technology*. 2020. Vol.119. P. 15– 26. DOI: 10.1016/j.infsof.2019.106238

22. Attardi J. Modern CSS. Book. Apress Berkeley, CA. 2025. 356 p.

23. Manhas J. Initial framework for website design and development. *International Journal of Information Technology.* 2017. Vol. 9 P.363-375. DOI: 10.1007/s41870-017-0045-4

24. Lucas W. Nunes R. Bonifácio R. Understanding the adoption of modern Javascript features: An empirical study on open-source systems. *Empirical Software Engineering.* Vol.30 № 107. P.30–32. DOI: 10.1007/s10664-025-10663-9

25. Rio A. Abreu F. Mendes D. Causal inference of server- and client-side code smells in web apps evolution. *Empirical Software Engineering.* Vol.29 № 133.P.30–35 DOI: 10.1007/s10664-024-10478-0

26. Jinat A. Sik-Lányi C. Kelemen A. Accessibility engineering in web evaluation process: a systematic literature review. *Universal Access in the Information Society*. 2023 Vol. 23. P.654 – 663. DOI: 10.1007/s10209-023-00967-2

27. Nokeri. T.C. Essentials of HTML. In: Web App Development and Real-Time Web Analytics with Python. Apress, Berkeley, CA. 2021. P.63-78. DOI: 10.1007/978-1-4842-7783-6_4

28. Singh H. Kaur P. An Effective Clustering-Based Web Page Recommendation Framework for E-Commerce Websites. *SN Computer Science*. 2021 Vol. 2 № 339. P.21–32. DOI: 10.1007/s42979-021-00736-z

29. Malallah H. Innovative Service Design of Online Agricultural Product Platforms: Improving User Experience and Trust. *Journal of Innovation and Development.* Vol. 9 № 1. P.40-42 DOI: 10.54097/27zmm960.

30. Романюк О. Н., Катєльніков Д. І., Косовець О. П. Веб-дизайн і комп’ютерна графіка. Навч.посіб. Вінниця ВНТУ 2007. 145 с.

31. Трофименко О. Г., Козін О. Б., Задерейко О. В., Плачінда О. Є. Веб-технології та веб-дизайн : навч. посіб. Одеса : Фенікс, 2019. 284 с.

32. Стеценко І. В., Пасічник О. В., Пасічник О. Г. Основи веб-дизайну : навч.посіб. Київ: Видавнича група BHV, 2009. 337 с.

33. Цеслів О. В. Web-програмування : навч.посіб. Київ: НТУУ «КПІ», 2011. 298 с.

34. Мельник Р. А. Програмування веб-застосувань (фронт-енд та бек-енд): навч.посіб. Львів : Видавництво Львівської політехніки, 2018. 248 с.

35. HTML Підручник URL: https://w3schoolsua.github.io/html (дата звернення 22.04.2026)